**Oracle® Database**

Hướng dẫn bảo mật

11*g* Release 2 (11.2)

**E36292-09**

Tháng 1 năm 2017



Oracle Database Security Guide 11*g* Release 2 (11.2) E36292-09

Copyright © 2006, 2017, Oracle and/or its affiliates. All rights reserved. Primary Author: Patricia Huey

Contributors: Tammy Bednar, Naveen Gopal, Don Gosselin, Sumit Jeloka, Peter Knaggs, Sergei Kucherov, Nina Lewis, Bryn Llewellyn, Rahil Mir, Narendra Manappa, Gopal Mulagund, Janaki Narasinghanallur, Paul Needham, Deb Owens, Robert Pang, Preetam Ramakrishna, Vipin Samar, Digvijay Sirmukaddam, Richard Smith, Sachin Sonawane, James Spiller, Ashwini Surpur, Srividya Tata, Kamal Tbeileh, Rodney Ward, Daniel Wong

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and

adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

## Contents

[Preface](#_bookmark0) xxiii

[Audience](#_bookmark1) xxiii

[Documentation Accessibility](#_bookmark2) xxiii

[Related Documents](#_bookmark3) xxiv

[Conventions](#_bookmark4) xxiv

[What's New in Oracle Database Security?](#_bookmark5) xxv

[Oracle Database 11](#_bookmark7)*g* Release 2 (11.2.0.2) New Security Features xxv

[Oracle Database 11](#_bookmark14)*g* Release 2 (11.2.0.1) New Security Features xxvii

[Oracle Database 11](#_bookmark33)*g* Release 1 (11.1) New Security Features xxxi

1. [Introducing Oracle Database Security](#_bookmark50)

[About Oracle Database Security](#_bookmark51) 1-1

[Additional Database Security Resources](#_bookmark53) 1-2

1. [Managing Security for Oracle Database Users](#_bookmark55)

[About User Security](#_bookmark57) 2-1

[Creating User Accounts](#_bookmark59) 2-1

[Creating a New User Account](#_bookmark61) 2-2

[Specifying a User Name](#_bookmark65) 2-2

[Assigning the User a Password](#_bookmark68) 2-3

[Assigning a Default Tablespace for the User](#_bookmark71) 2-4

[Assigning a Tablespace Quota for the User](#_bookmark75) 2-5

[Restricting the Quota Limits for User Objects in a Tablespace](#_bookmark81) 2-5

[Granting Users the UNLIMITED TABLESPACE System Privilege](#_bookmark84) 2-5

[Assigning a Temporary Tablespace for the User](#_bookmark89) 2-6

[Specifying a Profile for the User](#_bookmark92) 2-7

[Setting a Default Role for the User](#_bookmark95) 2-7

[Altering User Accounts](#_bookmark97) 2-7

[About Altering User Accounts](#_bookmark98) 2-7

[Using the ALTER USER Statement to Alter a User Account](#_bookmark100) 2-8

[Changing Non-SYS User Passwords](#_bookmark103) 2-8

[Changing the SYS User Password](#_bookmark107) 2-9

[Configuring User Resource Limits](#_bookmark113) 2-9

[About User Resource Limits](#_bookmark114) 2-10

[Types of System Resources and Limits](#_bookmark117) 2-10

[Limiting the User Session Level.](#_bookmark119) 2-10

[Limiting Database Call Levels](#_bookmark121) 2-11

[Limiting CPU Time](#_bookmark123) 2-11

[Limiting Logical Reads](#_bookmark125) 2-11

[Limiting Other Resources](#_bookmark127) 2-11

[Determining Values for Resource Limits of Profiles](#_bookmark136) 2-12

[Managing Resources with Profiles](#_bookmark139) 2-12

[Creating Profiles](#_bookmark144) 2-13

[Dropping Profiles](#_bookmark148) 2-14

[Deleting User Accounts](#_bookmark152) 2-14

[Finding Information About Database Users and Profiles](#_bookmark164) 2-15

[Using Data Dictionary Views to Find Information About Users and Profiles](#_bookmark166) 2-15

[Listing All Users and Associated Information](#_bookmark171) 2-16

[Listing All Tablespace Quotas](#_bookmark173) 2-17

[Listing All Profiles and Assigned Limits](#_bookmark175) 2-17

[Viewing Memory Use for Each User Session](#_bookmark178) 2-18

1. [Configuring Authentication](#_bookmark181)

[About Authentication](#_bookmark182) 3-1

[Configuring Password Protection](#_bookmark184) 3-1

[What Are the Oracle Database Built-in Password Protections?](#_bookmark186) 3-2

[Minimum Requirements for Passwords](#_bookmark196) 3-3

[Using a Password Management Policy](#_bookmark200) 3-3

[About Managing Passwords](#_bookmark202) 3-4

[Finding User Accounts That Have Default Passwords](#_bookmark206) 3-4

[Configuring Password Settings in the Default Profile](#_bookmark209) 3-4

[Disabling and Enabling the Default Password Security Settings](#_bookmark221) 3-6

[Automatically Locking a User Account After a Failed Login](#_bookmark227) 3-6

[Controlling User Ability to Reuse Previous Passwords](#_bookmark239) 3-7

[Controlling Password Aging and Expiration](#_bookmark249) 3-8

[Password Change Life Cycle](#_bookmark256) 3-9

[Setting the PASSWORD\_LIFE\_TIME Profile Parameter to a Low Value](#_bookmark265) 3-10

[Enforcing Password Complexity Verification](#_bookmark268) 3-11

[Enabling or Disabling Password Case Sensitivity](#_bookmark274) 3-13

[Ensuring Against Password Security Threats by Using the SHA-1 Hashing Algorithm](#_bookmark289) 3-15

[Managing the Secure External Password Store for Password Credentials](#_bookmark297) 3-16

[About the Secure External Password Store](#_bookmark299) 3-17

[How Does the External Password Store Work?](#_bookmark304) 3-17

[Configuring Clients to Use the External Password Store](#_bookmark307) 3-18

[Managing External Password Store Credentials](#_bookmark311) 3-20

[Authenticating Database Administrators](#_bookmark316) 3-22

[Strong Authentication and Centralized Management for Database Administrators](#_bookmark318) 3-22

[Configuring Directory Authentication for Administrative Users](#_bookmark323) 3-22

[Configuring Kerberos Authentication for Administrative Users](#_bookmark325) 3-23

[Configuring Secure Sockets Layer Authentication for Administrative Users](#_bookmark327) 3-24

[Authenticating Database Administrators by Using the Operating System](#_bookmark329) 3-25

[Authenticating Database Administrators by Using Their Passwords](#_bookmark332) 3-25

[Using the Database to Authenticate Users](#_bookmark339) 3-26

[About Database Authentication](#_bookmark341) 3-26

[Advantages of Database Authentication](#_bookmark345) 3-26

[Creating a User Who Is Authenticated by the Database](#_bookmark347) 3-27

[Using the Operating System to Authenticate Users](#_bookmark349) 3-27

[Using the Network to Authenticate Users](#_bookmark356) 3-28

[Authentication Using Secure Sockets Layer](#_bookmark357) 3-28

[Authentication Using Third-Party Services](#_bookmark361) 3-28

[Configuring Global User Authentication and Authorization](#_bookmark374) 3-30

[Creating a User Who Is Authorized by a Directory Service](#_bookmark381) 3-31

[Creating a Global User Who Has a Private Schema](#_bookmark383) 3-31

[Creating Multiple Enterprise Users Who Share Schemas](#_bookmark387) 3-31

[Advantages of Global Authentication and Global Authorization](#_bookmark390) 3-31

[Configuring an External Service to Authenticate Users and Passwords](#_bookmark393) 3-32

[About External Authentication](#_bookmark394) 3-32

[Advantages of External Authentication](#_bookmark397) 3-33

[Creating a User Who Is Authenticated Externally](#_bookmark399) 3-33

[Authenticating User Logins Using the Operating System](#_bookmark401) 3-34

[Authentication User Logins Using Network Authentication](#_bookmark405) 3-34

[Using Multitier Authentication and Authorization](#_bookmark407) 3-34

[Administration and Security in Clients, Application Servers, and Database Servers](#_bookmark410) 3-35

[Preserving User Identity in Multitiered Environments](#_bookmark413) 3-36

[Using a Middle Tier Server for Proxy Authentication](#_bookmark415) 3-36

[About Proxy Authentication](#_bookmark417) 3-36

[Advantages of Proxy Authentication](#_bookmark423) 3-37

[Who Can Create Proxy User Accounts?](#_bookmark426) 3-38

[Creating Proxy User Accounts and Authorizing Users to Connect Through Them](#_bookmark430) 3-38

[Using Proxy Authentication with the Secure External Password Store](#_bookmark440) 3-40

[Passing Through the Identity of the Real User by Using Proxy Authentication](#_bookmark445) 3-40

[Limiting the Privilege of the Middle Tier](#_bookmark452) 3-41

[Authorizing a Middle Tier to Proxy and Authenticate a User](#_bookmark455) 3-42

[Authorizing a Middle Tier to Proxy a User Authenticated by Other Means](#_bookmark458) 3-42

[Reauthenticating the User Through the Middle Tier to the Database](#_bookmark461) 3-43

[Using Client Identifiers to Identify Application Users Not Known to the Database](#_bookmark466) 3-44

[About Client Identifiers](#_bookmark468) 3-44

[How Client Identifiers Work in Middle Tier Systems](#_bookmark471) 3-44

[Using the CLIENT\_IDENTIFIER Attribute to Preserve User Identity](#_bookmark473) 3-45

[Using CLIENT\_IDENTIFIER Independent of Global Application Context](#_bookmark475) 3-45

[Using the DBMS\_SESSION PL/SQL Package to Set and Clear the Client Identifier](#_bookmark481) 3-46

[Finding Information About User Authentication](#_bookmark488) 3-47

1. [Configuring Privilege and Role Authorization](#_bookmark494)

[About Privileges and Roles](#_bookmark496) 4-1

[Who Should Be Granted Privileges?](#_bookmark500) 4-2

[Granting the SYSDBA and SYSOPER Administrative Privileges to Users](#_bookmark504) 4-2

[Managing System Privileges](#_bookmark506) 4-2

[About System Privileges](#_bookmark508) 4-3

[Why Is It Important to Restrict System Privileges?](#_bookmark511) 4-3

[Restricting System Privileges by Securing the Data Dictionary](#_bookmark514) 4-3

[Allowing Access to Objects in the SYS Schema](#_bookmark521) 4-4

[Granting and Revoking System Privileges](#_bookmark528) 4-4

[Who Can Grant or Revoke System Privileges?](#_bookmark531) 4-5

[About ANY Privileges and the PUBLIC Role](#_bookmark533) 4-5

[Managing User Roles](#_bookmark539) 4-6

[About User Roles.](#_bookmark542) 4-6

[The Functionality of Roles](#_bookmark545) 4-6

[Properties of Roles and Why They Are Advantageous](#_bookmark553) 4-7

[Common Uses of Roles](#_bookmark559) 4-8

[How Roles Affect the Scope of a User’s Privileges](#_bookmark566) 4-9

[How Roles Work in PL/SQL Blocks](#_bookmark569) 4-9

[How Roles Aid or Restrict DDL Usage](#_bookmark575) 4-9

[How Operating Systems Can Aid Roles](#_bookmark579) 4-10

[How Roles Work in a Distributed Environment](#_bookmark581) 4-11

[Predefined Roles in an Oracle Database Installation](#_bookmark582) 4-11

[Creating a Role](#_bookmark643) 4-16

[Specifying the Type of Role Authorization](#_bookmark656) 4-17

[Authorizing a Role by Using the Database](#_bookmark658) 4-18

[Authorizing a Role by Using an Application](#_bookmark663) 4-18

[Authorizing a Role by Using an External Source](#_bookmark666) 4-18

[Global Role Authorization by an Enterprise Directory Service](#_bookmark677) 4-19

[Granting and Revoking Roles](#_bookmark685) 4-20

[About Granting and Revoking Roles](#_bookmark686) 4-20

[Who Can Grant or Revoke Roles?](#_bookmark690) 4-20

[Dropping Roles](#_bookmark693) 4-21

[Restricting SQL\*Plus Users from Using Database Roles](#_bookmark697) 4-21

[Potential Security Problems of Using Ad Hoc Tools](#_bookmark699) 4-21

[Limiting Roles Through the PRODUCT\_USER\_PROFILE Table](#_bookmark703) 4-22

[Using Stored Procedures to Encapsulate Business Logic](#_bookmark706) 4-22

[Securing Role Privileges by Using Secure Application Roles](#_bookmark707) 4-22

[Managing Object Privileges](#_bookmark712) 4-23

[About Object Privileges](#_bookmark713) 4-23

[Granting or Revoking Object Privileges](#_bookmark715) 4-24

[Managing Object Privileges](#_bookmark719) 4-24

[Granting and Revoking Object Privileges](#_bookmark723) 4-25

[Who Can Grant Object Privileges?](#_bookmark725) 4-25

[Using Object Privileges with Synonyms](#_bookmark726) 4-25

[Managing Table Privileges](#_bookmark728) 4-26

[How Table Privileges Affect Data Manipulation Language Operations](#_bookmark730) 4-26

[How Table Privileges Affect Data Definition Language Operations](#_bookmark733) 4-27

[Managing View Privileges](#_bookmark735) 4-27

[About View Privileges](#_bookmark736) 4-27

[Privileges Required to Create Views](#_bookmark739) 4-27

[Increasing Table Security with Views](#_bookmark742) 4-28

[Managing Procedure Privileges](#_bookmark745) 4-29

[Using the EXECUTE Privilege for Procedure Privileges](#_bookmark746) 4-29

[Procedure Execution and Security Domains](#_bookmark752) 4-29

[How Procedure Privileges Affect Definer’s Rights](#_bookmark753) 4-29

[How Procedure Privileges Affect Invoker’s Rights](#_bookmark760) 4-30

[System Privileges Required to Create or Replace a Procedure](#_bookmark763) 4-31

[System Privileges Required to Compile a Procedure](#_bookmark766) 4-31

[How Procedure Privileges Affect Packages and Package Objects](#_bookmark770) 4-31

[Managing Type Privileges](#_bookmark775) 4-33

[System Privileges for Named Types](#_bookmark777) 4-33

[Object Privileges](#_bookmark781) 4-33

[Method Execution Model](#_bookmark782) 4-34

[Privileges Required to Create Types and Tables Using Types](#_bookmark783) 4-34

[Example of Privileges for Creating Types and Tables Using Types](#_bookmark785) 4-34

[Privileges on Type Access and Object Access](#_bookmark787) 4-35

[Type Dependencies](#_bookmark790) 4-36

[Granting a User Privileges and Roles](#_bookmark791) 4-37

[Granting System Privileges and Roles](#_bookmark793) 4-37

[Granting the ADMIN Option](#_bookmark800) 4-38

[Creating a New User with the GRANT Statement](#_bookmark805) 4-38

[Granting Object Privileges](#_bookmark809) 4-38

[Specifying the GRANT OPTION Clause](#_bookmark813) 4-39

[Granting Object Privileges on Behalf of the Object Owner](#_bookmark816) 4-39

[Granting Privileges on Columns](#_bookmark819) 4-40

[Row-Level Access Control](#_bookmark821) 4-41

[Revoking Privileges and Roles from a User](#_bookmark822) 4-41

[Revoking System Privileges and Roles](#_bookmark823) 4-41

[Revoking Object Privileges](#_bookmark827) 4-42

[Revoking Object Privileges on Behalf of the Object Owner](#_bookmark830) 4-42

[Revoking Column-Selective Object Privileges](#_bookmark833) 4-43

[Revoking the REFERENCES Object Privilege](#_bookmark835) 4-43

[Cascading Effects of Revoking Privileges](#_bookmark837) 4-43

[Cascading Effects When Revoking System Privileges](#_bookmark839) 4-44

[Cascading Effects When Revoking Object Privileges](#_bookmark841) 4-44

[Granting to and Revoking from the PUBLIC Role](#_bookmark843) 4-45

[Granting Roles Using the Operating System or Network](#_bookmark846) 4-45

[About Granting Roles Using the Operating System or Network](#_bookmark848) 4-45

[Using Operating System Role Identification](#_bookmark849) 4-46

[Using Operating System Role Management](#_bookmark852) 4-47

[Granting and Revoking Roles When OS\_ROLES Is Set to TRUE](#_bookmark854) 4-47

[Enabling and Disabling Roles When OS\_ROLES Is Set to TRUE](#_bookmark856) 4-47

[Using Network Connections with Operating System Role Management](#_bookmark859) 4-47

[When Do Grants and Revokes Take Effect?](#_bookmark862) 4-48

[How the SET ROLE Statement Affects Grants and Revokes](#_bookmark865) 4-48

[Specifying a Default Role](#_bookmark870) 4-48

[The Maximum Number of Roles That a User Can Enable](#_bookmark874) 4-49

[Managing Fine-Grained Access in PL/SQL Packages and Types](#_bookmark877) 4-49

[About Fine-Grained Access Control to External Network Services](#_bookmark881) 4-50

[About Access Control to Wallets](#_bookmark885) 4-50

[Upgrading Applications That Depend on Packages That Use External Network Services](#_bookmark888) 4-51

[Creating an Access Control List for External Network Services](#_bookmark891) 4-51

[Step 1: Create the Access Control List and Its Privilege Definitions](#_bookmark894) 4-51

[Step 2: Assign the Access Control List to One or More Network Hosts](#_bookmark897) 4-53

[Configuring Access Control to a Wallet](#_bookmark900) 4-55

[Step 1: Create an Oracle Wallet](#_bookmark910) 4-55

[Step 2: Create an Access Control List that Grants the Wallet Privileges](#_bookmark911) 4-56

[Step 3: Assign the Access Control List to the Wallet](#_bookmark912) 4-57

[Step 4: Make the HTTP Request with the Passwords and Client Certificates](#_bookmark913) 4-57

[Examples of Creating Access Control Lists](#_bookmark914) 4-59

[Example of an Access Control List for a Single Role and Network Connection](#_bookmark917) 4-60

[Example of an Access Control List with Multiple Roles Assigned to Multiple Hosts ..](#_bookmark919) 4-61

[Example of an Access Control List for Using Passwords in a Non-Shared Wallet](#_bookmark923) 4-62

[Example of an Access Control List for Wallets in a Shared Database Session](#_bookmark925) 4-64

[Specifying a Group of Network Host Computers](#_bookmark927) 4-64

[Precedence Order for a Host Computer in Multiple Access Control List Assignments](#_bookmark929) 4-65

[Precedence Order for a Host in Access Control List Assignments with Port Ranges](#_bookmark931) 4-66

[Checking Privilege Assignments That Affect User Access to a Network Host](#_bookmark933) 4-66

[How a DBA Can Check User Network Connection and Domain Privileges](#_bookmark936) 4-67

[How Users Can Check Their Network Connection and Domain Privileges](#_bookmark942) 4-68

[Setting the Precedence of Multiple Users and Roles in One Access Control List.](#_bookmark948) 4-69

[Finding Information About Access Control Lists Configured for User Access](#_bookmark951) 4-71

[Finding Information About User Privileges and Roles](#_bookmark960) 4-71

[Listing All System Privilege Grants](#_bookmark968) 4-73

[Listing All Role Grants](#_bookmark971) 4-73

[Listing Object Privileges Granted to a User](#_bookmark974) 4-73

[Listing the Current Privilege Domain of Your Session](#_bookmark978) 4-74

[Listing Roles of the Database](#_bookmark982) 4-74

[Listing Information About the Privilege Domains of Roles](#_bookmark985) 4-75

1. [Managing Security for Application Developers](#_bookmark991)

[About Application Security Policies](#_bookmark992) 5-1

[Considerations for Using Application-Based Security](#_bookmark995) 5-1

[Are Application Users Also Database Users?](#_bookmark997) 5-2

[Is Security Better Enforced in the Application or in the Database?](#_bookmark1007) 5-2

[Securing Passwords in Application Design](#_bookmark1012) 5-3

[General Guidelines for Securing Passwords in Applications](#_bookmark1015) 5-3

[Platform-Specific Security Threats](#_bookmark1016) 5-3

[Designing Applications to Handle Password Input](#_bookmark1018) 5-4

[Configuring Password Formats and Behavior](#_bookmark1021) 5-5

[Handling Passwords in SQL\*Plus and SQL Scripts](#_bookmark1022) 5-5

[Securing Passwords Using an External Password Store](#_bookmark1026) 5-7

[Securing Passwords Using the orapwd Utility](#_bookmark1027) 5-7

[Example of Reading Passwords in Java](#_bookmark1028) 5-7

[Managing Application Privileges](#_bookmark1031) 5-11

[Creating Secure Application Roles to Control Access to Applications](#_bookmark1036) 5-12

[Step 1: Create the Secure Application Role](#_bookmark1038) 5-12

[Step 2: Create a PL/SQL Package to Define the Access Policy for the Application](#_bookmark1040) 5-13

[Associating Privileges with User Database Roles](#_bookmark1050) 5-14

[Why Users Should Only Have the Privileges of the Current Database Role](#_bookmark1052) 5-14

[Using the SET ROLE Statement to Automatically Enable or Disable Roles](#_bookmark1056) 5-15

[Protecting Database Objects by Using Schemas](#_bookmark1058) 5-15

[Protecting Database Objects in a Unique Schema](#_bookmark1060) 5-15

[Protecting Database Objects in a Shared Schema](#_bookmark1066) 5-16

[Managing Object Privileges in an Application](#_bookmark1071) 5-16

[What Application Developers Need to Know About Object Privileges](#_bookmark1074) 5-16

[SQL Statements Permitted by Object Privileges](#_bookmark1080) 5-17

[Parameters for Enhanced Security of Database Communication](#_bookmark1094) 5-17

[Reporting Bad Packets Received on the Database from Protocol Errors](#_bookmark1099) 5-17

[Terminating or Resuming Server Execution After Receiving a Bad Packet](#_bookmark1105) 5-18

[Configuring the Maximum Number of Authentication Attempts](#_bookmark1108) 5-19

[Controlling the Display of the Database Version Banner](#_bookmark1112) 5-19

[Configuring Banners for Unauthorized Access and Auditing User Actions](#_bookmark1115) 5-20

1. [Using Application Contexts to Retrieve User Information](#_bookmark1123)

[About Application Contexts](#_bookmark1124) 6-1

[What Is an Application Context?](#_bookmark1125) 6-1

[Components of the Application Context](#_bookmark1127) 6-1

[Where Are the Application Context Values Stored?](#_bookmark1130) 6-2

[Benefits of Using Application Contexts](#_bookmark1138) 6-2

[How Editions Affects Application Context Values](#_bookmark1141) 6-3

[Types of Application Contexts](#_bookmark1143) 6-3

[Using Database Session-Based Application Contexts](#_bookmark1147) 6-4

[About Database Session-Based Application Contexts](#_bookmark1149) 6-4

[Creating a Database Session-Based Application Context](#_bookmark1155) 6-5

[Creating a PL/SQL Package to Set the Database Session-Based Application Context](#_bookmark1162) 6-6

[About the Package That Manages the Database Session-Based Application Context](#_bookmark1164) 6-7

[Using SYS\_CONTEXT to Retrieve Session Information](#_bookmark1168) 6-7

[Using Dynamic SQL with SYS\_CONTEXT](#_bookmark1176) 6-8

[Using SYS\_CONTEXT in a Parallel Query](#_bookmark1179) 6-9

[Using SYS\_CONTEXT with Database Links](#_bookmark1183) 6-9

[Using DBMS\_SESSION.SET\_CONTEXT to Set Session Information](#_bookmark1186) 6-9

[Creating a Logon Trigger to Run a Database Session Application Context Package](#_bookmark1193) 6-11

[Tutorial: Creating and Using a Database Session-Based Application Context](#_bookmark1206) 6-12

[About This Tutorial](#_bookmark1207) 6-13

[Step 1: Create User Accounts and Ensure the User SCOTT Is Active](#_bookmark1210) 6-13

[Step 2: Create the Database Session-Based Application Context](#_bookmark1211) 6-13

[Step 3: Create a Package to Retrieve Session Data and Set the Application Context](#_bookmark1212) 6-14

[Step 4: Create a Logon Trigger for the Package](#_bookmark1215) 6-15

[Step 5: Test the Application Context](#_bookmark1216) 6-15

[Step 6: Remove the Components for This Tutorial](#_bookmark1217) 6-16

[Initializing Database Session-Based Application Contexts Externally](#_bookmark1218) 6-16

[Obtaining Default Values from Users](#_bookmark1221) 6-16

[Obtaining Values from Other External Resources](#_bookmark1222) 6-17

[Initializing Application Context Values from a Middle-Tier Server](#_bookmark1224) 6-17

[Initializing Database Session-Based Application Contexts Globally](#_bookmark1227) 6-18

[About Initializing Database Session-Based Application Contexts Globally](#_bookmark1228) 6-18

[Using Database Session-Based Application Contexts with LDAP](#_bookmark1230) 6-18

[How Globally Initialized Database Session-Based Application Contexts Work](#_bookmark1232) 6-19

[Example of Initializing a Database Session-Based Application Context Globally](#_bookmark1234) 6-19

[Using Externalized Database Session-Based Application Contexts](#_bookmark1235) 6-21

[Using Global Application Contexts](#_bookmark1237) 6-22

[About Global Application Contexts](#_bookmark1239) 6-22

[Using Global Application Contexts in an Oracle Real Application Clusters Environment.](#_bookmark1254) 6-23

[Creating a Global Application Context](#_bookmark1256) 6-23

[Creating a PL/SQL Package to Manage a Global Application Context](#_bookmark1260) 6-23

[About the Package That Manages the Global Application Context](#_bookmark1262) 6-24

[How Editions Affects the Results of a Global Application Context PL/SQL Package .](#_bookmark1264) 6-24

[Setting the DBMS\_SESSION.SET\_CONTEXT username and client\_id Parameters](#_bookmark1269) 6-25

[Sharing Global Application Context Values for All Database Users](#_bookmark1274) 6-26

[Setting a Global Context for Database Users Who Move Between Applications](#_bookmark1278) 6-27

[Setting a Global Application Context for Nondatabase Users](#_bookmark1282) 6-28

[Clearing Session Data When the Session Closes](#_bookmark1290) 6-32

[Embedding Calls in Middle-Tier Applications to Manage the Client Session ID](#_bookmark1292) 6-32

[About Managing Client Session IDs Using a Middle-Tier Application](#_bookmark1293) 6-32

[Retrieving the Client Session ID Using a Middle-Tier Application](#_bookmark1294) 6-32

[Setting the Client Session ID Using a Middle-Tier Application](#_bookmark1296) 6-33

[Clearing Session Data Using a Middle-Tier Application](#_bookmark1298) 6-34

[Tutorial: Creating a Global Application Context That Uses a Client Session ID](#_bookmark1299) 6-35

[About This Tutorial](#_bookmark1301) 6-35

[Step 1: Create User Accounts](#_bookmark1306) 6-35

[Step 2: Create the Global Application Context](#_bookmark1307) 6-36

[Step 3: Create a Package for the Global Application Context](#_bookmark1308) 6-36

[Step 4: Test the Global Application Context](#_bookmark1309) 6-37

[Step 5: Remove the Components for This Tutorial](#_bookmark1311) 6-39

[Global Application Context Processes](#_bookmark1312) 6-39

[Simple Global Application Context Process](#_bookmark1313) 6-39

[Global Application Context Process for Lightweight Users](#_bookmark1315) 6-40

[Using Client Session-Based Application Contexts](#_bookmark1317) 6-42

[About Client Session-Based Application Contexts](#_bookmark1318) 6-42

[Setting a Value in the CLIENTCONTEXT Namespace](#_bookmark1323) 6-43

[Retrieving the CLIENTCONTEXT Namespace](#_bookmark1325) 6-43

[Clearing a Setting in the CLIENTCONTEXT Namespace](#_bookmark1328) 6-44

[Clearing All Settings in the CLIENTCONTEXT Namespace](#_bookmark1330) 6-44

[Finding Information About Application Contexts](#_bookmark1332) 6-45

1. [Using Oracle Virtual Private Database to Control Data Access](#_bookmark1342)

[About Oracle Virtual Private Database](#_bookmark1343) 7-1

[What Is Oracle Virtual Private Database?](#_bookmark1344) 7-1

[Benefits of Using Oracle Virtual Private Database Policies](#_bookmark1348) 7-2

[Basing Security Policies on Database Objects Rather Than Applications](#_bookmark1350) 7-2

[Controlling How Oracle Database Evaluates Policy Functions](#_bookmark1353) 7-3

[Which Privileges Are Used to Run Oracle Virtual Private Database Policy Functions?](#_bookmark1357) 7-3

[Using Oracle Virtual Private Database with an Application Context](#_bookmark1364) 7-3

[Components of an Oracle Virtual Private Database Policy](#_bookmark1370) 7-4

[Creating a Function to Generate the Dynamic WHERE Clause.](#_bookmark1372) 7-4

[Creating a Policy to Attach the Function to the Objects You Want to Protect](#_bookmark1376) 7-5

[Configuring an Oracle Virtual Private Database Policy](#_bookmark1377) 7-5

[About Oracle Virtual Private Database Policies](#_bookmark1380) 7-6

[Attaching a Policy to a Database Table, View, or Synonym](#_bookmark1398) 7-7

[Enforcing Policies on Specific SQL Statement Types](#_bookmark1401) 7-7

[Controlling the Display of Column Data with Policies](#_bookmark1406) 7-8

[Adding Policies for Column-Level Oracle Virtual Private Database](#_bookmark1407) 7-8

[Displaying Only the Column Rows Relevant to the Query](#_bookmark1412) 7-9

[Using Column Masking to Display Sensitive Columns as NULL Values](#_bookmark1414) 7-10

[Working with Oracle Virtual Private Database Policy Groups](#_bookmark1419) 7-11

[About Oracle Virtual Private Database Policy Groups](#_bookmark1420) 7-11

[Creating a New Oracle Virtual Private Database Policy Group](#_bookmark1425) 7-12

[Designating a Default Policy Group with the SYS\_DEFAULT Policy Group](#_bookmark1427) 7-12

[Establishing Multiple Policies for Each Table, View, or Synonym](#_bookmark1430) 7-13

[Validating the Application Used to Connect to the Database](#_bookmark1432) 7-13

[Optimizing Performance by Using Oracle Virtual Private Database Policy Types](#_bookmark1435) 7-14

[About Oracle Virtual Private Database Policy Types](#_bookmark1436) 7-14

[Using the Dynamic Policy Type to Automatically Rerun Policy Functions](#_bookmark1438) 7-15

[Using a Static Policy to Prevent Policy Functions from Rerunning for Each Query](#_bookmark1443) 7-16

[Using a Shared Static Policy to Share a Policy with Multiple Objects](#_bookmark1446) 7-16

[When to Use Static and Shared Static Policies](#_bookmark1449) 7-17

[Using a Context-Sensitive Policy for Predicates That Do Not Change After Parsing ...](#_bookmark1452) 7-17

[Using a Shared Context Sensitive Policy to Share a Policy with Multiple Objects](#_bookmark1455) 7-18

[When to Use Context-Sensitive and Shared Context-Sensitive Policies](#_bookmark1458) 7-18

[Summary of the Five Oracle Virtual Private Database Policy Types](#_bookmark1461) 7-19

[Tutorials: Creating Oracle Virtual Private Database Policies](#_bookmark1465) 7-19

[Tutorial: Creating a Simple Oracle Virtual Private Database Policy](#_bookmark1467) 7-19

[About This Tutorial](#_bookmark1468) 7-20

[Step 1: Ensure That the OE User Account Is Active](#_bookmark1470) 7-20

[Step 2: Create a Policy Function](#_bookmark1471) 7-20

[Step 3: Create the Oracle Virtual Private Database Policy](#_bookmark1472) 7-21

[Step 4: Test the Policy](#_bookmark1473) 7-21

[Step 5: Remove the Components for This Tutorial](#_bookmark1474) 7-22

[Tutorial: Implementing a Policy with a Database Session-Based Application Context](#_bookmark1475) 7-22

[About This Tutorial](#_bookmark1476) 7-23

[Step 1: Create User Accounts and Sample Tables](#_bookmark1478) 7-23

[Step 2: Create a Database Session-Based Application Context](#_bookmark1479) 7-24

[Step 3: Create a PL/SQL Package to Set the Application Context](#_bookmark1480) 7-24

[Step 4: Create a Logon Trigger to Run the Application Context PL/SQL Package](#_bookmark1481) 7-25

[Step 5: Create a PL/SQL Policy Function to Limit User Access to Their Orders](#_bookmark1482) 7-26

[Step 6: Create the New Security Policy](#_bookmark1483) 7-26

[Step 7: Test the New Policy](#_bookmark1484) 7-27

[Step 8: Remove the Components for This Tutorial](#_bookmark1487) 7-28

[Tutorial: Implementing an Oracle Virtual Private Database Policy Group](#_bookmark1488) 7-28

[About This Tutorial](#_bookmark1489) 7-28

[Step 1: Create User Accounts and Other Components for This Tutorial](#_bookmark1493) 7-29

[Step 2: Create the Two Policy Groups](#_bookmark1494) 7-29

[Step 3: Create PL/SQL Functions to Control the Policy Groups](#_bookmark1495) 7-30

[Step 4: Add the PL/SQL Functions to the Policy Groups](#_bookmark1496) 7-31

[Step 5: Create the Driving Application Context](#_bookmark1497) 7-32

[Step 6: Test the Policy Groups](#_bookmark1498) 7-32

[Step 7: Remove the Components for This Tutorial](#_bookmark1499) 7-33

[How Oracle Virtual Private Database Works with Other Oracle Features](#_bookmark1500) 7-34

[Using Oracle Virtual Private Database Policies with Editions](#_bookmark1501) 7-34

[Using SELECT FOR UPDATE in User Queries on VPD-Protected Tables](#_bookmark1503) 7-34

[How Oracle Virtual Private Database Policies Affect Outer or ANSI Join Operations](#_bookmark1505) 7-34

[How Oracle Virtual Private Database Security Policies Work with Applications](#_bookmark1508) 7-35

[Using Automatic Reparsing for Fine-Grained Access Control Policy Functions](#_bookmark1511) 7-35

[Using Oracle Virtual Private Database Policies and Flashback Query](#_bookmark1514) 7-35

[Using Oracle Virtual Private Database and Oracle Label Security](#_bookmark1516) 7-36

[Using Oracle Virtual Private Database to Enforce Oracle Label Security Policies](#_bookmark1517) 7-36

[Oracle Virtual Private Database and Oracle Label Security Exceptions](#_bookmark1520) 7-36

[Exporting Data Using the EXPDP Utility access\_method Parameter](#_bookmark1527) 7-37

[User Models and Oracle Virtual Private Database](#_bookmark1530) 7-38

[Finding Information About Oracle Virtual Private Database Policies](#_bookmark1543) 7-39

1. [Developing Applications Using the Data Encryption API](#_bookmark1548)

[Security Problems That Encryption Does Not Solve](#_bookmark1549) 8-1

[Principle 1: Encryption Does Not Solve Access Control Problems](#_bookmark1552) 8-1

[Principle 2: Encryption Does Not Protect Against a Malicious Database Administrator](#_bookmark1554) 8-2

[Principle 3: Encrypting Everything Does Not Make Data Secure](#_bookmark1558) 8-3

[Data Encryption Challenges](#_bookmark1560) 8-4

[Encrypting Indexed Data](#_bookmark1562) 8-4

[Generating Encryption Keys](#_bookmark1564) 8-4

[Transmitting Encryption Keys](#_bookmark1566) 8-5

[Storing Encryption Keys](#_bookmark1568) 8-5

[Storing the Encryption Keys in the Database](#_bookmark1570) 8-5

[Storing the Encryption Keys in the Operating System](#_bookmark1571) 8-7

[Users Managing Their Own Encryption Keys](#_bookmark1573) 8-7

[Using Transparent Database Encryption and Tablespace Encryption](#_bookmark1575) 8-7

[Changing Encryption Keys](#_bookmark1578) 8-7

[Encrypting Binary Large Objects](#_bookmark1580) 8-7

[Storing Data Encryption by Using the DBMS\_CRYPTO Package](#_bookmark1583) 8-8

[Examples of Using the Data Encryption API](#_bookmark1591) 8-10

[Example of a Data Encryption Procedure](#_bookmark1592) 8-10

[Example of AES 256-Bit Data Encryption and Decryption Procedures](#_bookmark1594) 8-11

[Example of Encryption and Decryption Procedures for BLOB Data](#_bookmark1596) 8-12

[Finding Information About Encrypted Data](#_bookmark1598) 8-15

1. [Verifying Security Access with Auditing](#_bookmark1603)

[About Auditing](#_bookmark1604) 9-1

[What Is Auditing?](#_bookmark1605) 9-2

[Why Is Auditing Used?](#_bookmark1608) 9-2

[Protecting the Database Audit Trail](#_bookmark1609) 9-3

[Activities That Are Always Written to the Standard and Fine-Grained Audit Records](#_bookmark1614) 9-3

[Activities That Are Always Audited for All Platforms](#_bookmark1619) 9-4

[Auditing in a Distributed Database.](#_bookmark1627) 9-4

[Best Practices for Auditing](#_bookmark1629) 9-4

[Selecting an Auditing Type](#_bookmark1631) 9-5

[Auditing SQL Statements, Privileges, and Other General Activities](#_bookmark1635) 9-5

[Auditing Commonly Used Security-Relevant Activities](#_bookmark1636) 9-5

[Auditing Specific, Fine-Grained Activities](#_bookmark1637) 9-6

[Auditing General Activities with Standard Auditing](#_bookmark1639) 9-6

[About Standard Auditing](#_bookmark1640) 9-7

[What Is Standard Auditing?](#_bookmark1641) 9-7

[Who Can Perform Standard Auditing?](#_bookmark1645) 9-7

[When Are Standard Audit Records Created?](#_bookmark1648) 9-8

[Configuring Standard Auditing with the AUDIT\_TRAIL Initialization Parameter](#_bookmark1654) 9-8

[Enabling or Disabling the Standard Audit Trail](#_bookmark1656) 9-8

[Settings for the AUDIT\_TRAIL Initialization Parameter](#_bookmark1666) 9-9

[What Do the Operating System and Database Audit Trails Have in Common?.](#_bookmark1686) 9-11

[Using the Operating System Audit Trail](#_bookmark1695) 9-12

[About the Operating System Trail](#_bookmark1696) 9-13

[What Do Operating System Audit Trail Records Look Like?](#_bookmark1700) 9-13

[Advantages of the Operating System Audit Trail](#_bookmark1707) 9-16

[How the Operating System Audit Trail Works](#_bookmark1711) 9-17

[Specifying a Directory for the Operating System Audit Trail](#_bookmark1713) 9-17

[Using the Syslog Audit Trail on UNIX Systems](#_bookmark1719) 9-18

[About the Syslog Audit Trail](#_bookmark1721) 9-18

[Format of the Information Stored in the Syslog Audit Trail](#_bookmark1724) 9-19

[What Does the Syslog Audit Trail Look Like?](#_bookmark1726) 9-19

[Configuring Syslog Auditing](#_bookmark1729) 9-19

[How the AUDIT and NOAUDIT SQL Statements Work](#_bookmark1732) 9-20

[Enabling Standard Auditing with the AUDIT SQL Statement](#_bookmark1734) 9-20

[Auditing Statement Executions: Successful, Unsuccessful, or Both](#_bookmark1739) 9-21

[How Standard Audit Records Are Generated](#_bookmark1742) 9-21

[How Do Cursors Affect Standard Auditing?](#_bookmark1745) 9-22

[Benefits of Using the BY ACCESS Clause in the AUDIT Statement](#_bookmark1747) 9-22

[Auditing Actions Performed by Specific Users](#_bookmark1750) 9-23

[Removing the Audit Option with the NOAUDIT SQL Statement](#_bookmark1753) 9-23

[Auditing SQL Statements](#_bookmark1758) 9-23

[About SQL Statement Auditing](#_bookmark1760) 9-23

[Types of SQL Statements That Are Audited](#_bookmark1764) 9-23

[Configuring SQL Statement Auditing](#_bookmark1767) 9-24

[Removing SQL Statement Auditing](#_bookmark1778) 9-25

[Auditing Privileges](#_bookmark1784) 9-26

[About Privilege Auditing](#_bookmark1786) 9-26

[Types of Privileges That Can Be Audited](#_bookmark1789) 9-26

[Configuring Privilege Auditing](#_bookmark1791) 9-26

[Removing Privilege Auditing](#_bookmark1795) 9-27

[Auditing SQL Statements and Privileges in a Multitier Environment.](#_bookmark1798) 9-27

[Auditing Schema Objects](#_bookmark1807) 9-28

[About Schema Object Auditing](#_bookmark1809) 9-29

[Types of Schema Objects That Can Be Audited](#_bookmark1814) 9-29

[Using Standard Auditing with Editioned Objects](#_bookmark1819) 9-29

[Schema Object Audit Options for Views, Procedures, and Other Elements](#_bookmark1821) 9-29

[Configuring Schema Object Auditing](#_bookmark1823) 9-30

[Removing Object Auditing](#_bookmark1833) 9-31

[Setting Audit Options for Objects That May Be Created in the Future](#_bookmark1837) 9-31

[Auditing Directory Objects](#_bookmark1839) 9-32

[About Directory Object Auditing](#_bookmark1841) 9-32

[Configuring Directory Object Auditing](#_bookmark1843) 9-32

[Removing Directory Object Auditing](#_bookmark1846) 9-32

[Auditing Functions, Procedures, Packages, and Triggers](#_bookmark1848) 9-32

[About Auditing Functions, Procedures, Packages, and Triggers](#_bookmark1849) 9-32

[Configuring the Auditing of Functions, Procedures, Packages, and Triggers](#_bookmark1860) 9-33

[Removing the Auditing of Functions, Procedures, Packages, and Triggers](#_bookmark1866) 9-33

[Auditing Network Activity](#_bookmark1867) 9-33

[About Network Auditing](#_bookmark1868) 9-34

[Configuring Network Auditing](#_bookmark1874) 9-34

[Removing Network Auditing](#_bookmark1876) 9-35

[Using Default Auditing for Security-Relevant SQL Statements and Privileges](#_bookmark1880) 9-35

[About the Default Auditing Settings](#_bookmark1882) 9-35

[Privileges That Oracle Database Audits by Default](#_bookmark1887) 9-35

[Disabling and Enabling Default Audit Settings](#_bookmark1890) 9-36

[Auditing Specific Activities with Fine-Grained Auditing](#_bookmark1893) 9-36

[About Fine-Grained Auditing](#_bookmark1895) 9-37

[Where Are Fine-Grained Audit Records Stored?](#_bookmark1902) 9-37

[Advantages of Fine-Grained Auditing](#_bookmark1904) 9-38

[What Permissions Are Needed to Create a Fine-Grained Audit Policy?](#_bookmark1907) 9-38

[Activities That Are Always Audited in Fine-Grained Auditing](#_bookmark1909) 9-38

[Using Fine-Grained Audit Policies with Editions](#_bookmark1912) 9-39

[Creating an Audit Trail for Fine-Grained Audit Records](#_bookmark1914) 9-39

[How the Fine-Grained Audit Trail Generates Records](#_bookmark1916) 9-39

[Using the DBMS\_FGA Package to Manage Fine-Grained Audit Policies](#_bookmark1918) 9-39

[About the DBMS\_FGA PL/SQL Package](#_bookmark1919) 9-39

[Creating a Fine-Grained Audit Policy](#_bookmark1921) 9-40

[Disabling and Enabling a Fine-Grained Audit Policy](#_bookmark1934) 9-43

[Dropping a Fine-Grained Audit Policy](#_bookmark1939) 9-44

[Tutorial: Adding an Email Alert to a Fine-Grained Audit Policy](#_bookmark1942) 9-44

[About This Tutorial](#_bookmark1944) 9-44

[Step 1: Install and Configure the UTL\_MAIL PL/SQL Package](#_bookmark1950) 9-45

[Step 2: Create User Accounts](#_bookmark1952) 9-46

[Step 3: Configure an Access Control List File for Network Services](#_bookmark1953) 9-47

[Step 4: Create the Email Security Alert PL/SQL Procedure](#_bookmark1954) 9-47

[Step 5: Create and Test the Fine-Grained Audit Policy Settings](#_bookmark1955) 9-48

[Step 6: Test the Alert](#_bookmark1958) 9-48

[Step 7: Remove the Components for This Tutorial](#_bookmark1959) 9-49

[Tutorial: Auditing Nondatabase Users](#_bookmark1960) 9-50

[About This Tutorial](#_bookmark1961) 9-50

[Step 1: Create the User Account and Ensure the User HR Is Active](#_bookmark1963) 9-50

[Step 2: Create the Fine-Grained Audit Policy](#_bookmark1964) 9-51

[Step 3: Test the Policy](#_bookmark1966) 9-51

[Step 4: Remove the Components for This Tutorial](#_bookmark1967) 9-52

[Auditing SYS Administrative Users](#_bookmark1968) 9-53

[Auditing User SYSTEM](#_bookmark1970) 9-53

[Auditing User SYS and Users Who Connect as SYSDBA and SYSOPER](#_bookmark1973) 9-53

[Using Triggers to Write Audit Data to a Separate Table](#_bookmark1986) 9-55

[Managing Audit Trail Records](#_bookmark1995) 9-57

[About Audit Records](#_bookmark1996) 9-57

[Managing the Database Audit Trail](#_bookmark1999) 9-58

[Database Audit Trail Contents](#_bookmark2000) 9-58

[Controlling the Size of the Database Audit Trail](#_bookmark2011) 9-59

[Moving the Database Audit Trail to a Different Tablespace](#_bookmark2015) 9-60

[Auditing the Database Audit Trail](#_bookmark2018) 9-61

[Archiving the Database Audit Trail](#_bookmark2021) 9-61

[Managing the Operating System Audit Trail](#_bookmark2029) 9-62

[If the Operating System Audit Trail Becomes Full](#_bookmark2031) 9-62

[Setting the Size of the Operating System Audit Trail](#_bookmark2033) 9-62

[Setting the Age of the Operating System Audit Trail](#_bookmark2036) 9-64

[Archiving the Operating System Audit Trail](#_bookmark2039) 9-65

[Purging Audit Trail Records](#_bookmark2043) 9-65

[About Purging Audit Trail Records](#_bookmark2045) 9-66

[Selecting an Audit Trail Purge Method](#_bookmark2047) 9-66

[Scheduling an Automatic Purge Job for the Audit Trail](#_bookmark2051) 9-67

[Step 1: If Necessary, Tune Online and Archive Redo Log Sizes](#_bookmark2054) 9-68

[Step 2: Plan a Timestamp and Archive Strategy](#_bookmark2055) 9-68

[Step 3: Initialize the Audit Trail Cleanup Operation](#_bookmark2056) 9-68

[Step 4: Optionally, Set an Archive Timestamp for Audit Records](#_bookmark2058) 9-69

[Step 5: Create and Schedule the Purge Job](#_bookmark2063) 9-70

[Step 6: Optionally, Configure the Audit Trail Records to be Deleted in Batches](#_bookmark2065) 9-71

[Manually Purging the Audit Trail](#_bookmark2069) 9-72

[Purging a Subset of Records from the Database Audit Trail](#_bookmark2072) 9-74

[Other Audit Trail Purge Operations](#_bookmark2075) 9-75

[Verifying That the Audit Trail Is Initialized for Cleanup](#_bookmark2076) 9-75

[Setting the Default Audit Trail Purge Interval for Any Audit Trail Type](#_bookmark2078) 9-76

[Cancelling the Initialization Cleanup Settings](#_bookmark2080) 9-76

[Enabling or Disabling an Audit Trail Purge Job](#_bookmark2082) 9-77

[Setting the Default Audit Trail Purge Job Interval for a Specified Purge Job](#_bookmark2084) 9-77

[Deleting an Audit Trail Purge Job](#_bookmark2086) 9-78

[Clearing the Archive Timestamp Setting](#_bookmark2088) 9-78

[Clearing the Database Audit Trail Batch Size](#_bookmark2090) 9-78

[Example: Directly Calling a Database Audit Trail Purge Operation](#_bookmark2092) 9-79

[Finding Information About Audited Activities](#_bookmark2095) 9-80

[Using Data Dictionary Views to Find Information About the Audit Trail](#_bookmark2097) 9-80

[Using Audit Trail Views to Investigate Suspicious Activities](#_bookmark2102) 9-81

[Listing Active Statement Audit Options](#_bookmark2104) 9-82

[Listing Active Privilege Audit Options](#_bookmark2107) 9-82

[Listing Active Object Audit Options for Specific Objects](#_bookmark2109) 9-83

[Listing Default Object Audit Options](#_bookmark2111) 9-83

[Listing Audit Records](#_bookmark2113) 9-83

[Listing Audit Records for the AUDIT SESSION Option](#_bookmark2114) 9-83

[Deleting the Audit Trail Views](#_bookmark2115) 9-84

1. [Keeping Your Oracle Database Secure](#_bookmark2119)

[About the Security Guidelines in This Chapter](#_bookmark2120) 10-1

[Downloading Security Patches and Contacting Oracle Regarding Vulnerabilities](#_bookmark2121) 10-2

[Applying Security Patches and Workaround Solutions](#_bookmark2123) 10-2

[Contacting Oracle Security Regarding Vulnerabilities in Oracle Database](#_bookmark2128) 10-2

[Guidelines for Securing User Accounts and Privileges](#_bookmark2130) 10-2

[Guidelines for Securing Roles](#_bookmark2159) 10-6

[Guidelines for Securing Passwords](#_bookmark2166) 10-7

[Guidelines for Securing Data](#_bookmark2194) 10-10

[Guidelines for Securing the ORACLE\_LOADER Access Driver](#_bookmark2215) 10-11

[Guidelines for Securing a Database Installation and Configuration](#_bookmark2219) 10-12

[Guidelines for Securing the Network](#_bookmark2226) 10-13

[Securing the Client Connection](#_bookmark2228) 10-13

[Securing the Network Connection](#_bookmark2239) 10-14

[Securing a Secure Sockets Layer Connection](#_bookmark2269) 10-16

[Guidelines for Auditing](#_bookmark2284) 10-18

[Auditing Sensitive Information](#_bookmark2287) 10-18

[Keeping Audited Information Manageable](#_bookmark2291) 10-18

[Auditing Typical Database Activity](#_bookmark2294) 10-19

[Auditing Suspicious Database Activity](#_bookmark2298) 10-20

[Recommended Audit Settings](#_bookmark2300) 10-21

[Addressing the CONNECT Role Change](#_bookmark2304) 10-22

[Why Was the CONNECT Role Changed?](#_bookmark2306) 10-22

[How the CONNNECT Role Change Affects Applications](#_bookmark2308) 10-22

[How the CONNECT Role Change Affects Database Upgrades](#_bookmark2310) 10-22

[How the CONNECT Role Change Affects Account Provisioning](#_bookmark2312) 10-23

[How the CONNECT Role Change Affects Applications Using New Databases](#_bookmark2314) 10-23

[How the CONNECT Role Change Affects Users](#_bookmark2316) 10-23

[How the CONNECT Role Change Affects General Users](#_bookmark2318) 10-23

[How the CONNECT Role Change Affects Application Developers](#_bookmark2320) 10-23

[How the CONNECT Role Change Affects Client Server Applications](#_bookmark2322) 10-23

[Approaches to Addressing the CONNECT Role Change](#_bookmark2324) 10-24

[Approach 1: Create a New Database Role](#_bookmark2325) 10-24

[Approach 2: Restore CONNECT Privileges](#_bookmark2326) 10-25

[Approach 3: Conduct Least Privilege Analysis](#_bookmark2329) 10-25

##### [Glossary](#_bookmark2330) [Index](#_bookmark2353)

**List of Examples**

[2–1](#_bookmark64) Creating a User Account with the CREATE SESSION Privilege 2-2

[2–2](#_bookmark102) Altering a User Account 2-8

[2–3](#_bookmark112) Using ORAPWD to Change the SYS User Password 2-9

[2–4](#_bookmark156) Querying V$SESSION for the Session ID of a User 2-14

[2–5](#_bookmark158) Killing a User Session 2-14

[2–6](#_bookmark160) Finding Objects Owned by a User 2-15

[2–7](#_bookmark163) Dropping a User Account 2-15

[3–1](#_bookmark199) Password Creation SQL Statements 3-3

[3–2](#_bookmark234) Locking an Account with the CREATE PROFILE Statement 3-7

[3–3](#_bookmark255) Setting Password Aging and Expiration with CREATE PROFILE 3-9

[3–4](#_bookmark287) Enabling Password Case Sensitivity 3-14

[3–5](#_bookmark310) Sample SQLNET.ORA File with Wallet Parameters Set 3-20

[4–1](#_bookmark519) Setting O7\_DICTIONARY\_ACCESSIBILITY to FALSE 4-3

[4–2](#_bookmark649) Creating a User Role Authorized by a Password 4-17

[4–3](#_bookmark654) Altering a Role to be Authorized by an External Source 4-17

[4–4](#_bookmark661) Using SET ROLE for a Password-Authenticated Role 4-18

[4–5](#_bookmark665) Creating a Role Authorized by a PL/SQL Package for an Application 4-18

[4–6](#_bookmark670) Creating a Role Authorized by an External Source 4-19

[4–7](#_bookmark682) Creating a Global Role 4-19

[4–8](#_bookmark718) Revoking All Object Privileges Using CASCADE CONSTRAINTS 4-24

[4–9](#_bookmark769) Compiling a Procedure 4-31

[4–10](#_bookmark773) Package Objects Affected by Procedure Privileges 4-32

[4–11](#_bookmark797) Granting a System Privilege and a Role to a User 4-37

[4–12](#_bookmark799) Granting the EXECUTE Privilege on a Directory Object 4-37

[4–13](#_bookmark804) Granting the ADMIN Option 4-38

[4–14](#_bookmark808) Creating a New User with the GRANT Statement 4-38

[4–15](#_bookmark812) Granting Object Privileges to Users 4-39

[4–16](#_bookmark868) Using SET ROLE to Grant a Role and Specify a Password 4-48

[4–17](#_bookmark869) Using SET ROLE to Disable All Roles 4-48

[4–18](#_bookmark873) Using ALTER USER to Set Default Roles 4-49

[4–19](#_bookmark918) Creating an Access Control List for a Single Role and Network Connection 4-60

[4–20](#_bookmark920) Creating an Access Control List for Multiple Roles and Network Connections 4-61

[4–21](#_bookmark921) Using the DBA\_NETWORK\_ACL\_PRIVILEGES View to Show Granted Privileges ... 4-62 [4–22](#_bookmark922) Using the DBA\_NETWORK\_ACLS View to Show Host Assignments 4-62

[4–23](#_bookmark924) Configuring ACL Access Using Passwords in a Non-Shared Wallet 4-63

[4–24](#_bookmark926) Configuring ACL Access for a Wallet in a Shared Database Session 4-64

[4–25](#_bookmark939) Administrator Checking User Permissions for Network Host Connections 4-67

[4–26](#_bookmark941) Administrator Checking Permissions for Domain Name Resolution 4-68

[4–27](#_bookmark945) User Checking Permissions for Network Host Connections 4-69

[4–28](#_bookmark947) User Checking Privileges for Domain Name Resolution 4-69

[5–1](#_bookmark1030) Java Code for Reading Passwords 5-7

[6–1](#_bookmark1159) Creating a Database Session-Based Application Context 6-6

[6–2](#_bookmark1174) Finding SYS\_CONTEXT Values 6-8

[6–3](#_bookmark1192) Simple Procedure to Create an Application Context Value 6-10

[6–4](#_bookmark1197) Creating a Simple Logon Trigger 6-11

[6–5](#_bookmark1201) Creating a Logon Trigger for a Production Environment 6-12

[6–6](#_bookmark1203) Creating a Logon Trigger for a Development Environment 6-12

[6–7](#_bookmark1213) Package to Retrieve Session Data and Set a Database Session Context 6-14

[6–8](#_bookmark1223) Creating an Externalized Database Session-based Application Context 6-17

[6–9](#_bookmark1259) Creating a Global Application Context 6-23

[6–10](#_bookmark1277) Package to Manage Global Application Values for All Database Users 6-26

[6–11](#_bookmark1281) Package to Manage Global Application Context Values for a User Moving Between Applications 6-28

[6–12](#_bookmark1289) Package to Manage Global Application Context Values for Nondatabase Users 6-30

[6–13](#_bookmark1295) Using OCIStmtExecute to Retrieve a Client Session ID Value 6-33

[6–14](#_bookmark1327) Retrieving a Client Session ID Value for Client Session-Based Contexts 6-44

[7–1](#_bookmark1400) Attaching a Simple Oracle Virtual Private Database Policy to a Table 7-7

[7–2](#_bookmark1404) Specifying SQL Statement Types with DBMS\_RLS.ADD\_POLICY 7-8

[7–3](#_bookmark1411) Creating a Column-Level Oracle Virtual Private Database Policy 7-9

[7–4](#_bookmark1417) Adding a Column Masking to an Oracle Virtual Private Database Policy 7-10

[7–5](#_bookmark1442) Creating a DYNAMIC Policy with DBMS\_RLS.ADD\_POLICY 7-15

[7–6](#_bookmark1445) Creating a STATIC Policy with DBMS\_RLS.ADD\_POLICY 7-16

[7–7](#_bookmark1448) Creating a SHARED\_STATIC Policy with DBMS\_RLS.ADD\_POLICY 7-17

[7–8](#_bookmark1454) Creating a CONTEXT\_SENSITIVE Policy with DBMS\_RLS.ADD\_POLICY 7-17

[7–9](#_bookmark1457) Creating a SHARED\_CONTEXT\_SENSITIVE Policy with DBMS\_RLS.ADD\_POLICY 7-18

[9–1](#_bookmark1664) Checking the Current Value of the AUDIT\_TRAIL Initialization Parameter 9-9

[9–2](#_bookmark1665) Enabling the Standard Audit Trail 9-9

[9–3](#_bookmark1703) Text File Operating System Audit Trail 9-13

[9–4](#_bookmark1706) XML File Operating System Audit Trail 9-15

[9–5](#_bookmark1728) Syslog Audit Trail for SYS User 9-19

[9–6](#_bookmark1752) Using AUDIT to Audit User Actions 9-23

[9–7](#_bookmark1770) Using AUDIT to Enable SQL Statement Auditing 9-24

[9–8](#_bookmark1771) Auditing Unsuccessful Statements 9-24

[9–9](#_bookmark1782) Using NOAUDIT to Remove Session and SQL Statement Auditing 9-25

[9–10](#_bookmark1783) Using NOAUDIT to Remove ALL STATEMENTS Auditing 9-26

[9–11](#_bookmark1794) Using AUDIT to Configure Privilege Auditing 9-27

[9–12](#_bookmark1802) Using AUDIT to Audit a SQL Statement for a User 9-27

[9–13](#_bookmark1826) Configuring Auditing for a Schema Table 9-30

[9–14](#_bookmark1827) Auditing Successful Statements on a Schema Table 9-30

[9–15](#_bookmark1830) Configuring Auditing for Any New Objects Using the DEFAULT Clause 9-30

[9–16](#_bookmark1832) Auditing the Execution of a Procedure or Function 9-31

[9–17](#_bookmark1845) Auditing a Directory Object 9-32

[9–18](#_bookmark1863) Auditing All Functions, Procedures, Packages, and Triggers 9-33

[9–19](#_bookmark1864) Auditing a User’s Execution of Functions, Procedures, Packages, and Triggers 9-33

[9–20](#_bookmark1865) Auditing the Execution of a Procedure or Function within a Schema 9-33

[9–21](#_bookmark1931) Using DBMS\_FGA.ADD\_POLICY to Create a Fine-Grained Audit Policy 9-42

[9–22](#_bookmark1936) Disabling a Fine-Grained Audit Policy 9-43

[9–23](#_bookmark1938) Enabling a Fine-Grained Audit Policy 9-44

[9–24](#_bookmark1941) Dropping a Fine-Grained Audit Policy 9-44

[9–25](#_bookmark1972) Auditing Table Insert Operations by User SYSTEM 9-53

[9–26](#_bookmark1994) Audit Trigger to Record Before and After Changes to a Table 9-56

[9–27](#_bookmark2094) Directly Calling a Database Audit Trail Purge Operation 9-79

##### List of Tables

[2–1](#_bookmark169) Data Dictionary Views That Display Information about Users and Profiles 2-15

[3–1](#_bookmark212) Password-Specific Settings in the Default Profile 3-5

[3–2](#_bookmark243) Parameters Controlling Reuse of a Previous Password 3-7

[3–3](#_bookmark491) Data Dictionary Views That Describe User Authentication 3-47

[4–1](#_bookmark523) Roles to Allow Access to SYS Schema Objects 4-4

[4–2](#_bookmark555) Properties of Roles and Their Description 4-7

[4–3](#_bookmark584) Oracle Database Predefined Roles 4-11

[4–4](#_bookmark778) System Privileges for Named Types 4-33

[4–5](#_bookmark788) Privileges for Object Tables 4-35

[4–6](#_bookmark954) Data Dictionary Views That Display Information about Access Control Lists 4-71

[4–7](#_bookmark964) Data Dictionary Views That Display Information about Privileges and Roles 4-71

[5–1](#_bookmark1000) Features Affected by the One Big Application User Model 5-2

[5–2](#_bookmark1077) How Privileges Relate to Schema Objects 5-16

[5–3](#_bookmark1084) SQL Statements Permitted by Database Object Privileges 5-17

[6–1](#_bookmark1145) Types of Application Contexts 6-4

[6–2](#_bookmark1271) Setting the DBMS\_SESSION.SET\_CONTEXT username and client\_id Parameters 6-25

[6–3](#_bookmark1335) Data Dictionary Views That Display Information about Application Contexts 6-45

[7–1](#_bookmark1383) DBMS\_RLS Procedures 7-6

[7–2](#_bookmark1463) DBMS\_RLS.ADD\_POLICY Policy Types 7-19

[7–3](#_bookmark1541) Oracle Virtual Private Database in Different User Models 7-39

[7–4](#_bookmark1545) Data Dictionary Views That Display Information about VPD Policies 7-39

[8–1](#_bookmark1588) DBMS\_CRYPTO and DBMS\_OBFUSCATION\_TOOLKIT Feature Comparison 8-8

[8–2](#_bookmark1600) Data Dictionary Views That Display Information about Encrypted Data 8-15

[9–1](#_bookmark1669) AUDIT\_TRAIL Initialization Parameter Settings 9-10

[9–2](#_bookmark1691) Common Audited Actions in the Operating System and Database Audit Trails 9-12

[9–3](#_bookmark1737) Standard Auditing Levels and Their Effects 9-21

[9–4](#_bookmark1872) Auditable Network Error Conditions 9-34

[9–5](#_bookmark1992) Comparison of Built-in Auditing and Trigger-Based Auditing 9-56

[9–6](#_bookmark2049) Selecting an Audit Trail Purge Method 9-67

[9–7](#_bookmark2100) Data Dictionary Views That Display Information about the Database Audit Trail 9-80

[10–1](#_bookmark2327) Columns and Contents for DBA\_CONNECT\_ROLE\_GRANTEES 10-25

##### List of Figures

[3–1](#_bookmark258) Chronology of Password Lifetime and Grace Period 3-10

[3–2](#_bookmark412) Multitier Authentication 3-35

[4–1](#_bookmark561) Common Uses for Roles 4-8

[6–1](#_bookmark1231) Location of Application Context in LDAP Directory Information Tree 6-19

[9–1](#_bookmark1804) Auditing Proxy Users 9-28

[9–2](#_bookmark1806) Auditing Client Identifier Information Across Sessions 9-28

## Lời nói đầu

### Độc giả

Chào mừng đến với *Oracle Database Security Guide*. Cuốn sách này mô tả làm thế nào bạn có thể cấu hình bảo mật cho cơ sở dữ liệu Oracle bằng cách sử dụng những tính năng mặc định trong cơ sở dữ liệu.

Lời mở đầu chứa những chủ đề sau:

* Về độc giả
* Khả năng tiếp cận đến tài liệu
* Những tài liệu liên quan
* Các quy ước

*Oracle Database Security Guide* được dành cho các nhà quản trị cơ sở dữ liệu (DBAs), các nhà quản trị hệ thống, các lập trình viên ứng dụng, và những công việc khác được thực hiện sau đây một cách an toàn và hiệu quả:

* Thiết kế và cài đặt những chính sách bảo mật để bảo vệ dữ liệu của một tổ chức, những ngườI dùng, và các ứng dụng khỏi các hành động ngẫu nhiên, không thích hợp hoặc trái phép
* Tạo và thực hiện các chính sách và các ví dụ thực tiễn về kiểm tra, các biện pháp để đối phó với các hành động không thích hợp hoặc trái phép
* Tạo ra, bảo trì và xóa các tài khoản người dùng, mật khẩu, vai trò (role), và các quyền
* Việc pháp triển các ứng dụng cung cấp các dịch vụ mong muốn một cách bảo mật trong các mô hình tính toán đa dạngDeveloping applications that provide desired services securely in a variety of computational models, việc tận dụng các dịch vụ về cơ sở dữ liệu và thư mục để tối đa hóa hiệu quả và dễ sử dụng

Để sử dụng tài liệu này, bạn cần một sự hiểu biết căn bản về cách và tại sao một cơ sở dữ liệu được sử dụng, và sự quen thuộc với ngôn ngữ SQL.

* **Khả năng tiếp cận đến tài liệu**

Để biết thông tin về cam kết của Oracle khả năng tiếp cận, truy cập trang web Chương trình Tiếp cận Oracle tại <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.>

Tiếp cận đến sự hỗ trợ của Oracle

Những khách hàng của Oracle có quyền tiếp cận với sự hỗ trợ điện tử thông qua My Oracle Support. Thông tin chi tiết, truy cập [http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info](http://www.oracle.com/pls/topic/lookup?ctx=acc&amp;id=info) hoặc

Truy cập [http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs](http://www.oracle.com/pls/topic/lookup?ctx=acc&amp;id=trs)  nếu bạn bị điếc.

### Những tài liệu liên quan

Để biết them thông tin về những thứ liên quan đến bao mật, tìm đọc các nguồn tài nguyên của Oracle:

* + *Oracle Database Administrator's Guide*
  + *Oracle Database 2 Day DBA*
  + *Oracle Database 2 Day + Security Guide*
  + *Oracle Database Concepts*
  + *Oracle Database Reference*
  + *Oracle Database Vault Administrator's Guide*

Có rất nhiều ví dụ trong cuốn sách này sử dụng các lược đồ mẫu của các cơ sở dữ liệu gốc (seed database), cái mà bạn có thể tạo khi bạn cài đặt cơ sở dữ liệu Oracle. Tìm đọc Oracle Database Sample Schemas để biết them thông tin về các lược đồ này, cách mà nó được tạo và cách mà bạn có thể sử dụng chúng.

Oracle Technology Network (OTN) (Công nghệ mạng Oracle)

Bạn có thể tải miễn phí các bản ghi chú phát hành, tài liệu cài đặt, phiên bản mới nhất của cuốn sách này, sách giấy, hoặc các tài nguyên khác từ OTN. truy cập

<http://www.oracle.com/technetwork/index.html>

Để biết thông tin bảo mật cụ thể trên OTN, truy cập <http://www.oracle.com/technetwork/topics/security/whatsnew/index.html> FĐể có được tài liệu Oracle mới nhất, bao gồm cuốn sách này, truy cập <http://www.oracle.com/technetwork/documentation/index.html>

My Oracle Support (Sự hỗ trợ của Oracle)

Bạn có thể tìm thông tin về các bản vá lỗi bảo mật, chứng chỉ và sự hỗ trợ về kiến thiết bằng cách truy cập My Oracle Support (trước kia là Oracle*MetaLink*) tại

https://support.oracle.com

### Các quy ước

Các quy ước văn bản sau được sử dụng trong tài liệu này:

**Quy ước Ý nghĩa**

**in đậm** Kiểu in đập chỉ ra rằng các thành phần giao diện người dùng liên kết với một thao tác, hoặc các thuật ngữ được định nghĩa bằng văn bản hoặc bảng chú giải.

*in nghiêng* Kiểu in nghiêng chỉ ra rằng các tựa đề của sách, sự nhấn mạnh hoặc các biến mà bạn có thể cung cấp các giá trị cụ thể.

monospace Kiểu Monospace chỉ ra rằng các lệnh nằm trong một đoạn văn, đường dẫn, mã trong các ví dụ, chữ xuất hiện trên màn hình mà bạn nhập vào.

## Có điểm gì mới trong Oracle Database Security?

Oracle Database 11*g* Release 2 (11.2) các tính năng bảo mật và các cải tiến mô tả trong phần này bao gồm nỗ lực chung để cung cấp kiểm soát truy cập vượt trội, chính sách, và các chú giải với bản phần hát này.

Các phần dưới đây mô tả các tính năng bảo mật mới của cơ sở dữ liệu Oracle 11g Release 2 (11.2) và cung cấp các link đến thông tin bổ sung:

* + - [Oracle Database 11g Release 2 (11.2.0.2) New Security Features](#_bookmark7)
    - [Oracle Database 11g Release 2 (11.2.0.1) New Security Features](#_bookmark14)
    - [Oracle Database 11g Release 1 (11.1) New Security Features](#_bookmark33)

### Oracle Database 11*g* Release 2 (11.2.0.2) New Security Features

This section contains:

* + - [Enhancements to Fine-Grained Access to External Services and Wallets](#_bookmark8)
    - [Support for MERGE INTO Statements for Virtual Private Database Policies](#_bookmark9)
    - [BY ACCESS Audit Trail Option Now the Default for AUDIT Statements](#_bookmark10)
    - [Enhancements for the UTL\_SMTP PL/SQL Package](#_bookmark11)
    - [New DBMS\_SCHEDULER PL/SQL Package Global Scheduler Attributes](#_bookmark12)
    - [Change to the UNLIMITED TABLESPACE System Privilege](#_bookmark13)

**Enhancements to Fine-Grained Access to External Services and Wallets**

In this release, when you use fine-grained access control to configure external network services and wallets, you now can control access to the DBMS\_LDAP PL/SQL package. In a default database installation, this package is created with the EXECUTE privilege granted to PUBLIC users. This release enhances the security of this package by enabling you to control access to applications in the database that use this package. As part of this enhancement, the DBMS\_LDAP package is now an invoker's rights package. Before a user can connect to a remote network host, he or she must be granted the connect privilege in the access control list that was assigned to the remote network host.

See *Oracle Database PL/SQL Packages and Types Reference* for more information about the

DBMS\_LDAP package.

**Support for MERGE INTO Statements for Virtual Private Database Policies**

In previous releases of Oracle Database, when you created an Oracle Virtual Private Database policy on an application that included the MERGE INTO statement, the MERGE INTO statement would be prevented with an ORA-28132: Merge into syntax does

not support security policies error, due to the presence of the Virtual Private Database policy. In this release, you can create policies on applications that include MERGE INTO operations. To do so, in the DBMS\_RLS.ADD\_POLICY statement\_types parameter, include the INSERT, UPDATE, and DELETE statements, or just omit the statement\_types parameter altogether.

See ["Enforcing Policies on Specific SQL Statement Types" on page 7-7](#_bookmark1402) for more information.

**BY ACCESS Audit Trail Option Now the Default for AUDIT Statements**

Starting with this release, the standard audit records will by default be generated using the BY ACCESS clause functionality of the AUDIT statement. Both the BY ACCESS and BY SESSION clauses write an individual audit record for each audited event, but BY ACCESS captures more detail about the audited event.

See ["Benefits of Using the BY ACCESS Clause in the AUDIT Statement" on page 9-22](#_bookmark1748) for more information.

**Enhancements for the UTL\_SMTP PL/SQL Package**

Starting with this release, the UTL\_SMTP PL/SQL package has the following new functionality:

* + You now can configure the UTL\_SMTP PL/SQL package for use on both Transport Layer Security (TLS) and Secure Sockets Layer (SSL) servers.
  + UTL\_SMTP now provides support for the PLAIN, LOGON and CRAM\_MD5 password authentication schemes.

See *Oracle Database PL/SQL Packages and Types Reference* for more information about the

UTL\_SMTP package.

**New DBMS\_SCHEDULER PL/SQL Package Global Scheduler Attributes**

The DBMS\_SCHEDULER PL/SQL package the following two new global scheduler attributes, which are used to control encryption for connections to a mail server:

* + email\_server\_credential, which enables you to specify the schema and name of an existing credential object on which user SYS has the EXECUTE object privilege
  + email\_server\_encryption, which enables you to set one of three encryption settings for your mail server:
* ssl\_tls, which uses SSL or TLS to encrypt the connection to the mail server form the beginning of the connection
* starttls, in which the connection to the mail server starts as unencrypted but switches to an encrypted connection
* none, in which no encryption is used to connect to the mail server

See *Oracle Database Administrator's Guide* for more information about Scheduler preferences.

**Change to the UNLIMITED TABLESPACE System Privilege**

In previous releases, when you revoked the UNLIMITED TABLESPACE system privilege from users, then the explicit quotas again took effect. Starting with this release, after you revoke the UNLIMITED TABLESPACE system privilege, you must explicitly grant quotas to individual tablespaces.

See ["Granting Users the UNLIMITED TABLESPACE System Privilege" on page 2-5](#_bookmark85) for more information about the UNLIMITED TABLESPACE system privilege.

### Oracle Database 11*g* Release 2 (11.2.0.1) New Security Features

This section contains:

* + [Enhancements to Fine-Grained Access to External Services and Wallets](#_bookmark15)
  + [Support for MERGE INTO Statements for Virtual Private Database Policies](#_bookmark9)
  + [Global Application Contexts Available Across Oracle RAC Instances](#_bookmark16)
  + [Secure Sockets Layer (SSL) Version 2 Support Change](#_bookmark17)
  + [Enhancements to Directory Objects](#_bookmark18)
  + [Enhancements to Transparent Data Encryption](#_bookmark21)
  + [Enhancements to the Audit Trail Cleanup Process](#_bookmark26)
  + [Deprecated Security-Related Features](#_bookmark27)

**Enhancements to Fine-Grained Access to External Services and Wallets**

The previous release of Oracle Database introduced the ability to create fine-grained access control to external network services and wallets. In this release, the following enhancements are available:

* + **Updates to the UTL\_HTTP PL/SQL package.** You now can configure network services to use the Amazon Simple Storage Service (S3) scheme, which configures access to the Amazon.com Web site. In addition, an individual application can make HTTP requests by using its private wallet and HTTP cookie table that will not be shared with other applications in the same database session. This feature also offers protection of the wallet using the access control list (ACL) privileges in place of the password credential.
  + **Support for IP Version 6 (IPv6) addresses.** The DBMS\_NETWORK\_ACL\_ADMIN and DBMS\_NETWORK\_ACL\_UTILTIY packages, and the PL/SQL network utility packages (such as UTL\_TCP, UTL\_SMTP, UTL\_MAIL, UTL\_HTTP, and UTL\_INADDR), now support both IP Version 4 (IPv4) and IPv6 addresses.

See ["Managing Fine-Grained Access in PL/SQL Packages and Types" on page 4-49](#_bookmark878) for more information.

**Global Application Contexts Available Across Oracle RAC Instances**

In this release, changes to global application context values are automatically accessible across all Oracle Real Application Clusters (Oracle RAC) instances.

See ["Using Global Application Contexts" on page 6-22](#_bookmark1238) for more information about creating a global application context.

**Secure Sockets Layer (SSL) Version 2 Support Change**

Starting with Oracle Database 11*g* Release 2 (11.2), SSL version 2 is no longer included in the default list of default supported protocols. If your applications must use SSL version 2, then you can do so by explicitly setting SSL version 2 while maintaining the connection.

See *Oracle Database Advanced Security Administrator's Guide* for more information.

**Enhancements to Directory Objects**

This section contains:

* + [EXECUTE Privilege Available for Directory Objects](#_bookmark19)
  + [Ability to Audit Directory Objects](#_bookmark20)

EXECUTE Privilege Available for Directory Objects

You now can grant users the EXECUTE privilege on directory objects that contain a user-supplied preprocessor program for use by the ORACLE\_LOADER access driver. This prevents the user from accidentally or maliciously corrupting the preprocessor

program. The SQL statements that are affected by the EXECUTE privilege are GRANT and REVOKE. The ORACLE\_LOADER access parameters now include the PREPROCESSOR clause, which you can use to specify the name and location of a preprocessor program that modifies the contents of a data file so that the ORACLE\_LOADER access driver can read it.

For more information about using the ORACLE\_LOADER access driver preprocessor, see the following:

* + *Oracle Database Utilities* for more information about the ORACLE\_LOADER access driver
  + ["Granting System Privileges and Roles" on page 4-37](#_bookmark794) for the syntax of granting the

EXECUTE privilege for a directory object

* + ["Guidelines for Securing the ORACLE\_LOADER Access Driver" on page 10-11](#_bookmark2216)
  + *Oracle Database SQL Language Reference* for updates to the GRANT and REVOKE SQL statements

Ability to Audit Directory Objects

You now can audit the EXECUTE privilege on directory objects. This enables you to monitor users who run a preprocessor program (which is used by the ORACLE\_LOADER access driver) that has been added to a directory object.

See ["Auditing Directory Objects" on page 9-32](#_bookmark1840) for more information.

**Enhancements to Transparent Data Encryption**

This section contains:

* + [Unified Master Encryption Key](#_bookmark22)
  + [Tablespace Master Key Rekey: Changing the Encryption Key Password](#_bookmark23)
  + [Transparent Data Encryption Support for Oracle Exadata](#_bookmark24)
  + [Automatic Wallet Management Across Oracle RAC Instances](#_bookmark25)

Unified Master Encryption Key

In this release, the master encryption key for transparent tablespace encryption and transparent column encryption are now combined to one unified master encryption key. Combining these keys enables transparent re-key operations for both of these transparent data encryption features, regardless of whether the master encryption key is stored in the Oracle Wallet or in one of the certified Hardware Security Modules offered by RSA, SafeNet, Thales (including nCipher), and Utimaco.

For more information about transparent data encryption, see *Oracle Database Advanced Security Administrator's Guide*.

Tablespace Master Key Rekey: Changing the Encryption Key Password

In this release, Oracle Advanced Security enables you to change the master key that protects the encryption keys used to encrypt Oracle Database tablespaces. Industry initiatives, such as the Payment Card Industry Data Security Standard (PCI DSS), mandate periodic rotation of encryption keys associated with credit card data.

For more information about tablespace encryption, see *Oracle Database Advanced Security Administrator's Guide*.

Transparent Data Encryption Support for Oracle Exadata

Starting with this release, the master encryption key is copied to the intelligent storage cells, where data encrypted with transparent tablespace encryption or transparent column encryption is now decrypted before the pre-filtering of the result set takes place. This feature improves performance in databases that use transparent data encryption.

For more information about Oracle Exadata, see *Oracle Database High Availability Overview*.

Automatic Wallet Management Across Oracle RAC Instances

When you now open or close an Oracle wallet or re-key the master encryption key on one Oracle RAC instance, then the changes you make automatically are propagated to all other Oracle RAC instances.

For more information, see *Oracle Database Advanced Security Administrator's Guide*.

**Enhancements to the Audit Trail Cleanup Process**

Oracle Database 11*g* Release 2 (11.2) introduces several enhancements to the audit trail cleanup process. In this release, you can:

* + - **Timestamp audit trail records based on their archive date.** Later on, you can purge all records that were created before this archive date.

See ["Step 4: Optionally, Set an Archive Timestamp for Audit Records" on page 9-69](#_bookmark2059) for more information.

* + - **Purge audit trail records in one operation or create a purge job.** You can purge all audit trail records in the system, or audit trail records of an individual type, such as all fine-grained audit trail records within the database audit trail. The purge operation will either remove audit trail records that were created before their timestamped archive date, or it will remove all audit trail records of the specified audit trail type. The purge job enables you to purge records based on a time interval, and also can remove records based on their timestamped archive date.

See the following sections:

* + - * ["Scheduling an Automatic Purge Job for the Audit Trail" on page 9-67](#_bookmark2052)
      * ["Manually Purging the Audit Trail" on page 9-72](#_bookmark2070)
    - **Move the database audit trail table from the SYSTEM tablespace to a different tablespace.** You can move the standard audit trail table, the fine-grained audit trail table, or both standard and fine-grained audit trail tables together. Consider moving the database audit trail from the SYSTEM tablespace if it is too busy.

See ["Moving the Database Audit Trail to a Different Tablespace" on page 9-60](#_bookmark2016) for more information.

* + - **Set a batch size for the database audit trail records so that when they are purged, the purge operation deletes each batch.** In a purge operation, you remove all or some of the audit trail records. Typically, you do this after you archive the audit trail. Afterwards, the audit trail will resume collecting audit data. The batching process enables you remove the records in groups, for example, 10,000 records at a time, rather than deleting all records at a time.

See ["Step 6: Optionally, Configure the Audit Trail Records to be Deleted in](#_bookmark2066) [Batches" on page 9-71](#_bookmark2066) for more information.

* + **Set a maximum size and age for the operating system audit trail.** When the current audit file reaches this maximum, Oracle Database stops populating the current file and creates a new file for the subsequent audit trail records.

See the following sections:

* ["Setting the Size of the Operating System Audit Trail" on page 9-62](#_bookmark2034)
* ["Setting the Age of the Operating System Audit Trail" on page 9-64](#_bookmark2037)

**Deprecated Security-Related Features**

This section contains:

* + [DB\_EXTENDED Setting for the AUDIT\_TRAIL Parameter Deprecated](#_bookmark28)
  + [WKUSER Role and Ultra Search Schemas Deprecated](#_bookmark29)
  + [Database Configuration Assistant No Longer Provides Default Security Settings](#_bookmark30)
  + [ALTER USER Clause AUTHENTICATED USING PASSWORD Deprecated](#_bookmark31)
  + [Password for the listener.ora File Deprecated](#_bookmark32)

DB\_EXTENDED Setting for the AUDIT\_TRAIL Parameter Deprecated

The DB\_EXTENDED setting in the AUDIT\_TRAIL initialization parameter has been deprecated. Instead, use the DB, EXTENDED setting in its place.

See ["Configuring Standard Auditing with the AUDIT\_TRAIL Initialization Parameter"](#_bookmark1655) [on page 9-8](#_bookmark1655) for more information.

WKUSER Role and Ultra Search Schemas Deprecated

The WKUSER role and the WKSYS, WKTEST, WKPROXY schemas have been deprecated. For more information about Oracle Ultra Search, see *Oracle Ultra Search Administrator's Guide*.

Database Configuration Assistant No Longer Provides Default Security Settings

In the previous release of Oracle Database, you could use Database Configuration Assistant (DBCA) to add password security and audit options to a new database. This option is not available in this release. In this release, DBCA automatically adds audit options and password policies to new databases.

See the following sections for more information:

* + ["Configuring Password Settings in the Default Profile" on page 3-4](#_bookmark210)
  + ["Using Default Auditing for Security-Relevant SQL Statements and Privileges" on page 9-35](#_bookmark1881)

ALTER USER Clause AUTHENTICATED USING PASSWORD Deprecated

The AUTHENTICATED USING PASSWORD clause of the ALTER USER statement has been deprecated for this release. If you use this clause, Oracle Database converts it to the AUTHENTICATION REQUIRED clause. If you do not specify the AUTHENTICATION REQUIRED clause, then Oracle Database uses either the AUTHENTICATED USING CERTIFICATE clause or the AUTHENTICATED USING DISTINGUISHED NAME clause.

See *Oracle Database SQL Language Reference* for more information about the ALTER USER

statement options.

Password for the listener.ora File Deprecated

Setting a password for the listener.ora file has been deprecated for this release, because it is no longer needed. In the next release, the listener password will not be supported.

### Oracle Database 11*g* Release 1 (11.1) New Security Features

This section contains:

* + - [Automatic Secure Configuration](#_bookmark34)
    - [New Password Protections](#_bookmark35)
    - [SYSDBA and SYSOPER Strong Authentication](#_bookmark36)
    - [SYSASM Privilege for Automatic Storage Management](#_bookmark37)
    - [Encryption Enhancements](#_bookmark38)
    - [Fine-Grained Access Control on Network Services on the Database](#_bookmark43)
    - [Change to AUDIT BY SESSION](#_bookmark44)
    - [Oracle XML DB Security Enhancements](#_bookmark45)
    - [Directory Security Enhancements](#_bookmark48)
    - [Oracle Call Interface Security Enhancements](#_bookmark49)

**Automatic Secure Configuration**

When you create a new database, you can use Database Configuration Assistant (DBCA) to automatically create a more secure configuration than in previous releases of Oracle Database. You can enable the following secure configuration settings in one operation:

* + - **Password-specific settings in the default profile.** This feature enables you to enforce password expiration and other password policies. See ["Configuring Password Settings in the Default Profile" on page 3-4](#_bookmark210) for more information.
    - **Auditing.** This feature enables auditing for specific events such as database connections. See ["Using Default Auditing for Security-Relevant SQL Statements and Privileges" on page 9-35](#_bookmark1881) for more information.

To configure your database for greater security, follow the guidelines in [Chapter 10,](#_bookmark2118) ["Keeping Your Oracle Database Secure"](#_bookmark2118).

**New Password Protections**

Oracle Database now includes the following new password protections:

* + - **Easy ability to find default passwords.** If you have upgraded from an earlier release of Oracle Database, you may have user accounts that still have default passwords. For greater security, you should find and change these passwords. See ["Finding User Accounts That Have Default Passwords" on page 3-4](#_bookmark207) for more information.
    - **Password verification.** Password verification ensures that users set complex passwords when setting or resetting passwords. You can enforce password by using the default settings provided by Oracle Database, or create custom requirements to further secure the password requirements for your site.

["Enforcing Password Complexity Verification" on page 3-11](#_bookmark269) describes built-in password verification.

* + **Enforced case sensitivity.** See ["Enabling or Disabling Password Case Sensitivity" on page 3-13](#_bookmark275) for more information.
  + **Stronger password hashing algorithm.** This enhancement enables users to create passwords that contain mixed case or special characters. See ["Ensuring Against Password Security Threats by Using the SHA-1 Hashing Algorithm" on page 3-15](#_bookmark290) for more information.

**SYSDBA and SYSOPER Strong Authentication**

You can now use the Secure Sockets Layer (SSL) and Kerberos strong authentication methods to authenticate users who have the SYSDBA and SYSOPER privileges.

See ["Strong Authentication and Centralized Management for Database](#_bookmark319) [Administrators" on page 3-22](#_bookmark319) for more information.

**SYSASM Privilege for Automatic Storage Management**

The SYSASM system privilege has been added to Oracle Database 11*g* Release 2 (11.2), to be used exclusively to administer Automatic Storage Management (ASM). Use the SYSASM privilege instead of the SYSDBA privilege to connect to and administer ASM instances.

See *Oracle Automatic Storage Management Administrator's Guide* for more information about the SYSASM privilege.

**Encryption Enhancements**

This section describes the following enhancements in encryption:

* + [Intelligent LOB Compression, Deduplication, and Encryption with SecureFiles](#_bookmark39)
  + [Compressed and Encrypted Dump File Sets](#_bookmark40)
  + [Transparent Data Encryption with Hardware Security Module Integration](#_bookmark41)
  + [Transparent Tablespace Encryption](#_bookmark42)

Intelligent LOB Compression, Deduplication, and Encryption with SecureFiles

Oracle Database supports a new, faster, and scalable Large Object (LOB) storage paradigm called SecureFiles. SecureFiles, in addition to performance, supports efficient compression, deduplication (that is, coalescing duplicate data), and encryption. LOB data can now be encrypted with Oracle Database, and is available for random reads and writes.

For more information about SecureFiles, see *Oracle Database SecureFiles and Large Objects Developer's Guide*. See also *Oracle Database SQL Language Reference* for updates in the CREATE TABLE and ALTER TABLE statements to support this feature.

Compressed and Encrypted Dump File Sets

In this release, you can use Oracle Data Pump to compress and encrypt an entire dump file set. You can optionally compress and encrypt the data, metadata, or complete dump file set during an Oracle Data Pump export.

For more information, see *Oracle Database Utilities*.

Transparent Data Encryption with Hardware Security Module Integration

Transparent data encryption (TDE) stores the master key in an encrypted software wallet and uses this key to encrypt the column keys, which in turn encrypt column data. While this approach to key management is sufficient for many applications, it

may not be sufficient for environments that require stronger security. TDE has been extended to use hardware security modules (HSMs). This enhancement provides high assurance requirements of protecting the master key.

This release enables you to store the TDE master encryption key within a hardware security module (HSM) at all times, leveraging its key management capabilities. Only the table keys (for TDE column encryption) and tablespace keys (for TDE tablespace encryption) are decrypted on the HSM, before they are returned to the database; the encryption and decryption of application data remains with the database. Oracle recommends that you encrypt the traffic between HSM device and databases. This new feature provides additional security for transparent data encryption, because the master encryption key cannot leave the HSM, neither in clear text nor in encrypted format. Furthermore, it enables the sharing of the same key between multiple databases and instances in an Oracle Real Applications Clusters (Oracle RAC) or Data Guard environment.

To configure transparent data encryption with hardware security module integration, see *Oracle Database Advanced Security Administrator's Guide*.

Transparent Tablespace Encryption

Transparent tablespace encryption enables you to encrypt entire application tablespaces, encrypting all the data within these tablespaces. When a properly authorized application accesses the tablespace, Oracle Database transparently decrypts the relevant data blocks for the application.

Transparent tablespace encryption provides an alternative to TDE column encryption: It eliminates the need for granular analysis of applications to determine which columns to encrypt, especially for applications with a large number of columns containing personally identifiable information (PII), such as Social Security numbers or patient health care records. If your tables have small amounts of data to encrypt, then you can continue to use the TDE column encryption solution.

For an introduction to transparent encryption, see *Oracle Database 2 Day + Security Guide*. For detailed information about transparent tablespace encryption, see *Oracle Database Advanced Security Administrator's Guide*.

**Fine-Grained Access Control on Network Services on the Database**

Oracle Database provides a set of PL/SQL utility packages, such as UTL\_TCP, UTL\_SMTP, UTL\_MAIL, UTL\_HTTP, and UTL\_INADDR, that are designed to enable database users to access network services on the database. *Oracle Database PL/SQL Packages and Types Reference* describes the PL/SQL utility packages in detail.

In a default database installation, these packages are created with EXECUTE privileges granted to PUBLIC users. This release enhances the security of these packages by providing database administrators the ability to control access to applications in the database that use these packages.

See ["Managing Fine-Grained Access in PL/SQL Packages and Types" on page 4-49](#_bookmark878) for more information.

**Change to AUDIT BY SESSION**

The BY SESSION clause of the AUDIT statement now writes one audit record for every audited event. In previous releases, BY SESSION wrote one audit record for all SQL statements or operations of the same type that were executed on the same schema objects in the same user session. Now, both BY SESSION and BY ACCESS write one audit record for each audit operation. In addition, there are separate audit records for LOGON

and LOGOFF events. If you omit the BY ACCESS clause, then BY SESSION is used as the default.

The audit record that BY SESSION generates is different from the BY ACCESS audit record. Oracle recommends that you include the BY ACCESS clause for all AUDIT statements, which results in a more detailed audit record. In the case of LOGOFF events, the timestamp for the audit record has a greater precision than in previous releases.

Be aware that this change applies to schema object audit options, statement options, and system privileges that audit SQL statements other than data definition language (DDL) statements. Oracle Database has always audited using the BY ACCESS clause on all SQL statements and system privileges that audit a DDL statement.

See the following sections for more information:

* + ["How Standard Audit Records Are Generated" on page 9-21](#_bookmark1743)
  + ["Benefits of Using the BY ACCESS Clause in the AUDIT Statement" on page 9-22](#_bookmark1748)

**Oracle XML DB Security Enhancements**

This section contains:

* + [XML Translation Support for Oracle Database XML](#_bookmark46)
  + [Support for Web Services](#_bookmark47)

XML Translation Support for Oracle Database XML

Security objects are now stored in the Oracle XML DB repository as XMLType objects. These security objects can contain strings that need to be translated to different languages so that they can be searched or displayed in those languages. Developers can store translated strings with the XMLType and retrieve and operate on these strings depending on the language settings of the user. The advantage of this feature is that it reduces the costs associated with developing applications that are independent of the target preferred language of the user.

To configure security for XMLType objects, see *Oracle XML DB Developer's Guide*.

Support for Web Services

You can now use the Oracle XML DB HTTP server for service-oriented architecture (SOA) operations. This allows the database to be treated as simply another service provider in an SOA environment. Security administrators can control user access to Oracle Database Web services and their associated database objects by using the XDB\_ WEBSERVICES, XDB\_WEBSERVICES\_OVER\_HTTP, and XDB\_WEBSERVICES\_WITH\_PUBLIC

predefined roles.

To configure Oracle Database Web services, see *Oracle XML DB Developer's Guide*.For information on this feature’s predefined roles, see [Table 4–3, " Oracle Database](#_bookmark585) [Predefined Roles" on page 4-11](#_bookmark585).

**Directory Security Enhancements**

In this release, administrators can now disallow anonymous access to database service information in a directory and require clients to authenticate when performing LDAP directory-based name look-ups. If you are using Microsoft Active Directory-based name lookups, then Oracle Database uses the native operating system-based authentication. If you are using Oracle Internet Directory (OID)-based name lookups, then Oracle Database performs authentication by using wallets.

To configure directory security, see *Oracle Database Net Services Reference*.

**Oracle Call Interface Security Enhancements**

The following security enhancements are available for Oracle Call Interface (OCI):

* + - Reporting bad packets that may come from malicious users or intruders
    - Terminating or resuming the client or server process on receiving a bad packet
    - Configuring the maximum number of authentication attempts
    - Controlling the display of the Oracle database version banner, to prevent intruders from finding information about the security vulnerabilities present in the database software based on the version
    - Adding banner information, such as "Unauthorized Access" and "User Actions Audited," to server connections so that clients can display this information

Database administrators can manage these security enhancements for Oracle Call Interface developers by configuring a set of new initialization parameters. See [Section ,](#_bookmark1095) ["Parameters for Enhanced Security of Database Communication"](#_bookmark1095) for more information. See also *Oracle Call Interface Programmer's Guide* for detailed information on Oracle Call Interface.

## Giới thiệu về Oracle Database Security

Chương này chứa:

* + - [Thông tin về Oracle Database Security](#_bookmark51)
    - Các tài nguyên Database Security bổ sung

### Thông tin về Oracle Database Security

Bạn có thể sử dụng các tính năng mặc định của cơ sở dữ liệu Oracle để cấu hình bảo mật trong các phần dưới đây cho việc cài đặt cơ sở dữ liệu Oracle của bạn:

* + - **Các tài khoản người dùng.** Khi bạn tạo các tài khoản người dùng, bạn có thể bảo mật chúng bằng nhiều cách. Bạn cũng có thể tạo password profiles để bảo mật chính sách mật khẩu tốt hơn cho trang web của bạn. [Chapter 2, "Managing Security for Oracle Database Users"](#_bookmark56) mô tả cách để quản lý các tài khoản người dùng.
    - **Các phương pháp xác thực.** Oracle Database cung cấp vài cách để cấu hình xác thực cho các người dùng và các nhà quản trị cơ sở dữ liệu. Ví dụ,bạn có thể xác thực các người dùng trên cấp độ cơ sở dữ liệu, từ hệ điều hành, và trên network. [Chapter 3, "Configuring Authentication"](#_bookmark180) mô tả cách để cách xác thực hoạt động như thế nào trong cơ sở dữ liệu Oracle.
    - **Các quyền và các vai trò.** Bạn có thể sử dụng các quyền và các vai trò để hạn chế người dùng truy cập vào dữ liệu. [Appendix 4, "Configuring Privilege and Role Authorization"](#_bookmark493) mô tả cách làm thế nào để tạo và quản lý quyền và vai trò của người dùng.
    - **Bảo mật ứng dụng.** Tbước đầu để tạo một cơ sở dữ liệu ứng dụng phải đảm bảo nó được bảo mật một cách hợp lý. [Appendix 5, "Managing Security for Application Developers"](#_bookmark990) dithảo luận làm thế nào để kết hợp bảo mật ứng dụng vào các chính sách bảo mật ứng dụng của bạn.
    - **User session information using application context.** Một ngữ cảnh ứng dụng (application context) là một cặp tên-giá trị cái mà giữ về thông tin của phiên làm việc đó. Bạn có thể thu thập thông tin phiên làm việc về một người dùng, như là username hoặc thiết bị đầu cuối và hạn chế quyền truy cập cơ sở dữ liệu, ứng dụng đối với người dùng dựa trên những thông tin này. [Chapter 6, "Using Application Contexts to Retrieve User Information"](#_bookmark1122) mô tả làm thể nào để sử dụng application context.
    - **Database access on the row and column level using Virtual Private Database.** A Virtual Private Database policy dynamically imbeds a WHERE predicate into SQL statements the user issues. [Chapter 7, "Using Oracle Virtual Private Database](#_bookmark1341) [to Control Data Access"](#_bookmark1341) describes how to create and manage Virtual Private Database policies.
  + **Mã hóa.** Bạn có thể ngụy trang dữ liệu trên network để ngăn chặn truy cập trái phép vào dữ liệu đó. [Appendix 8, "Developing Applications Using the Data Encryption API"](#_bookmark1547) giải thích làm thế nào để sử dụng DBMS\_CRYPTO và gói QL/SQL để mã hóa dữ liệu.
  + **Nhật ký hóa (auditing) các hoạt động của cơ sở dữ liệu.** Bạn có thể nhật ký hóa các hoạt động của cơ sở dữ liệu bằng các phương pháp như sau: logging các câu lệnh SQL, các quyền SQL, schema objects, và hoạt động của network. Hoặc, bạn có thể logging một cách chi tiết như là khi các địa chỉ IP từ bên ngoài công ty đang được sử dụng. Chương này cũng giải thích cách để dọn dẹp logging cơ sở dữ liệu. [Appendix 9, "Verifying Security Access with Auditing"](#_bookmark1602) mô tả cách để mở và cấu hình database auditing.

Ngoài ra, [Chapter 10, "Keeping Your Oracle Database Secure"](#_bookmark2118) cung cấp các hướng dẫn mà bạn nên làm theo khi bạn cài đặt bảo mật cơ sở dữ liệu Oracle.

### Các tài nguyên bảo mật cơ sở dữ liệu bổ sung

Ngoài các tài nguyên bảo mật được mô tả trong cuốn sách này, cơ sở dữ liệu Oracle cung cấp các sản phẩm bảo mật cơ sở dữ liệu:

* + **Advanced security features.** See *Oracle Database Advanced Security Administrator's Guide* for information about advanced features such as transparent data encryption, wallet management, network encryption, and the RADIUS, Kerberos, Secure Sockets Layer authentication.
  + **Oracle Label Security.** Oracle Label Security secures database tables at the row level, allowing you to filter user access to row data based on privileges. See *Oracle Label Security Administrator's Guide* for detailed information about Oracle Label Security.
  + **Oracle Database Vault.** Oracle Database Vault provides fine-grained access control to your sensitive data, including protecting data from privileged users. *Oracle Database Vault Administrator's Guide* describes how to use Oracle Database Vault.
  + **Oracle Audit Vault.** Oracle Audit Vault collects database audit data from sources such as Oracle Database audit trail tables, database operating system audit files, and database redo logs. Using Oracle Audit Vault, you can create alerts on suspicious activities, and create reports on the history of privileged user changes, schema modifications, and even data-level access. *Oracle Audit Vault Administrator's Guide* explains how to administer Oracle Audit Vault.
  + **Oracle Enterprise User Security.** Oracle Enterprise User Security enables you to manage user security at the enterprise level. *Oracle Database Enterprise User Security Administrator's Guide* explains how to configure Oracle Enterprise User Security.

In addition to these products, you can find the latest information about Oracle Database security, such as new products and important information about security patches and alerts, by visiting the Security Technology Center on Oracle Technology Network at

<http://www.oracle.com/technetwork/topics/security/whatsnew/index.html>

**2**

## Quản lý bảo mật các người dùng cho cơ sở dữ liệu Oracle

Chương này chứa:

* + - Thông tin về bảo mật người dùng
    - Tạo các tài khoản người dùng
    - Chỉnh sửa các tài khoản người dùng
    - Cấu hình các giới hạn tài nguyên người dùng
    - Xóa các tài khoản người dùng [Deleting User Accounts](#_bookmark152)
    - [Finding Information About Database Users and Profiles](#_bookmark164)

### Thông tin về bảo mật người dùng

Mỗi một cơ sở dữ liệu Oracle có một danh sách các cơ sở dữ liệ người dùng hợp lệ. Để truy cập một cơ sở dữ liệu , một người dùng phải chạy một ứng dụng cơ sở dữ liệu, và kết nối đến thể hiện của cơ sở dữ liệu bằng cách sử dụng một tên khoản hợp lệ được định nghĩa trong cơ sở dữ liệu. Cơ sở dữ liệu Oracle cho phép bạn thiết lập bảo mật cho những người dùng của bạn bằng nhiều cách.Khi bạn tạo các tài khoản người dùng, bạn có thể chỉ ra các giới hạn cho tài khoản người dùng đó. Bạn cũng có thể thiết lập các giới hạn trên một lượng tài nguyên hệ thống khác nhau khả dụng cho mỗi người dùng nhưng là một phần miền bảo mật của người dùng đó. Cơ sở dữ liệu Oracle cung cấp một tập hợp các khung nhìn (view) cơ cở dữ liệu mà bạn có thể truy vấn để tìm thông tin như là thông tin tài nguyên, phiên làm việc. Chương này cũng mô tả profiles. Một profile là một tập hợp các thuộc tính mà áp dụng đến một người dùng. Nó cho phép một điểm tham chiếu duy nhất cho bất kỳ người dùng nào có nhiều thuộc tính chính xác.

Một cách khác để quản lý bảo mật người dùng là gán cho những người dùng các quyền và các vai trò (role). [Chapter 4,](#_bookmark493) ["Configuring Privilege and Role Authorization"](#_bookmark493) cung cấp thông tin chi tiết.

### Tạo các tài khoản người dùng

Phần này chứa:

* + - Tạo một tài khoản người dùng mới
    - Chỉ ra một tên tài khoản
    - Gán cho tài khoản đó một mật khẩu
    - Gán Tablespace mặc định cho người dùng
    - Gán Tablespace Quota cho người dùng
    - Gán Tablespace tạm cho người dùng
    - Chỉ ra một profile cho người dùng
  + Thiết lập vai trò mặc định cho người dùng (default role)

Để biết hướng dẫn về cách tạo và quản lý tài khoản người dùng và mật khẩu, hãy xem các phần sau:

* + ["Guidelines for Securing User Accounts and Privileges" on page 10-2](#_bookmark2131)
  + ["Guidelines for Securing Passwords" on page 10-7](#_bookmark2167)

#### Tạo mới một tài khoản

Bạn có thể tạo một tài khoản cơ sở dữ liệu với câu lệnh CREATE USER. Để tạo một người dùng, bạn phải có quyền system trên câu lệnh create user. Bởi vì đó là một quyền cực kỳ mạnh, một người quản trị cơ sở dữ liệu hoặc một người quản trị bảo mật thường là người duy nhất có quyền hệ thống để thực thi câu lệnh đó.

[Example 2–1](#_bookmark64) tạo ra một người dùng và chỉ ra mật khẩu người dùng, default tablespace, tablespace tạm nơi mà các segments tạm được tạo, tablespace quotas, và profile. Nó cũng cấp quyền tối thiếu cho người dùng đó, CREATE SESSION, để đăng nhập vào phiên làm việc cơ sở dữ liệu.

***Example 2–1 Tạo một tài khoản người dùng với quyền CREATE SESSION***

CREATE USER jward IDENTIFIED BY *password* DEFAULT TABLESPACE data\_ts

QUOTA 100M ON test\_ts QUOTA 500K ON data\_ts

TEMPORARY TABLESPACE temp\_ts

PROFILE clerk;

GRANT CREATE SESSION TO jward;

Thay thế *password*  với một mật khẩu an toàn. Đọc ["Minimum Requirements for](#_bookmark197) [Passwords" trên trang 3-3](#_bookmark197) để biết thêm chi tiết.

Một người dùng vừa được tạo ra không thể kết nối đén cơ sở dữ liệu cho đến khi bạn cấp quyền cho người dùng đó quyền hệ thống CREATE SESSION. Vì vậy, ngay lập tức sau khi bạn tạo tài khoản người dùng, sử dụng lệnh GRANT để cấp quyền cho người các quyền đó. Nếu người dùng phải truy cập Oracle Enterpise Manager, bạn cũng nên cấp quyền cho người dùng quyền SELECT ANY DICTIONARY.

**Lưu ý:** Đối với người quản trị bảo mật, bạn nên tạo các vai trò riêng của bạn và chỉ gán những quyền cần thiết. Ví dụ, nhiều người dùng, trước đây được cấp quyền CONNECT đã không cần quyền bổ sung này. Thay vào đó, chỉ có quyền CREATE SESSION là thật sự cần thiết, và trong thực tế, quyền CONNECT hiện vẫn còn được giữ lại

Tạo ra các vai trò dựa trên nghiệp vụ của tổ chức cho phép tổ chức đó quyền điều khiển chi tiết mà đã được gán.

[Chapter 4, "Configuring Privilege and Role Authorization"](#_bookmark493) giải thích làm sao để tạo và quản lý các vai trò.

#### Chỉ ra một tên người dùng

Bên trong mỗi cơ sở dữ liệu, một tên người dùng phải duy nhất tươnng tự như là với các tên vai trò. Một người và một vai trò không thể có cùng tên giống nhau. Hơn nữa, mỗi người dùng được liên kết với một schema (lược đồ), Bên trong một lược đồ, mỗi đói tượng lược đồ phải có một tên duy nhất. Trong phần dưới đây, phần chữ được **in** đậm cho thấy cách để tạo một tên người dùng.

Người dùng jward is được lưu truễ trong cơ sở dữ liệu theo kiểu chữ viết hoa, ví dụ:

**CREATE USER jward** IDENTIFIED BY *password* DEFAULT TABLESPACE data\_ts

QUOTA 100M ON test\_ts QUOTA 500K ON data\_ts

TEMPORARY TABLESPACE temp\_ts

PROFILE clerk CONTAINER = CURRENT;

SELECT USERNAME FROM ALL\_USERS; USERNAME

---------

JWARD

...

Tuy nhiên, Nếu bạn đặt tên người dùng trong dấu ngoặc kép, sau đó tên đó sẽ được lưu trữ phân biệt hoa thường. ví dụ:

CREATE USER "jward" IDENTIFIED BY *password*;

So, when you query the ALL\_USERS data dictionary view, you will find that the user account is stored using the case that you used to create it.

SELECT USERNAME FROM ALL\_USERS;

USERNAME

---------

jward

...

Người dùng JWARD và người dùng jward đều được lưu trữ trong cơ sơ dữ liệu như là 2 tài khoản người dùng riêng biệt. Sau đó, nếu bạn phải chỉnh sửa hoặc drop người dùng mà bạn đã tạo thông qua dấu ngoặc kép, thì bạn phải đặt tên người dùng trong dấu ngoặc kép.

Ví dụ:

DROP USER "jward";

#### Gán mật khẩu cho người dùng

Trong [Ví dụ 2–1 trên trang 2-2](#_bookmark64), người dùng mới sẽ được xác thực thông qua cơ sở dữ liệu. Trong trường hợp này, người dùng đang kết nối phải cung cấp mật khẩu đúng để kết nối đến cơ sở dữ liệu. Để đặt mật khẩu cho người dùng, sử dụng mệnh đề IDENTIFIED BY trong câu lệnh CREATE USER.

CREATE USER jward **IDENTIFIED BY *password*** DEFAULT TABLESPACE data\_ts

QUOTA 100M ON test\_ts QUOTA 500K ON data\_ts

TEMPORARY TABLESPACE temp\_ts

PROFILE clerk;

**Xem thêm:**

* + - ["Các yêu cầu tối thiểu để đặt mật khẩu" trên trang 3-3](#_bookmark197) các yêu cầu tối thiểu để tạo nên mật khẩu
    - ["Nguyên tắc bảo mật mật khẩu" trên trang 10-7](#_bookmark2167) các cách bổ sung để bảo vệ mật khẩu
    - [Chương 3, "Cấu hình xác thực "](#_bookmark180) để biết thêm thông tin về các phương pháp xác thực khả dụng cho người dùng cơ sở đữ liệu Oracle

#### Gán một Default Tablespace cho người dùng

Mỗi người dùng nên có một default tablespace. Khi một đối tượng schema được tạo ra trong schema của user và câu lệnh DDL không chỉ ra tablespace để chứa đối tượng, Oracle Database lưu trữ đối tượng trong tablespace mặc định của người dùng.

Thiết lập mặc định cho tất cả các tablespace mặc định của mọi người dùng là SYSTEM tablespace. Nếu một người dùng không tạo các đối tượng, và cũng không có quyền để làm như thế, thì cài đặt mặc định là khá ổn. Tuy nhiên, nếu mọt người dùng có khả năng tạo bất kỳ kiểu đối tượng, thì bạn nên chỉ định cụ thể default tablspace, như là USERS tablespace. Sử dụng một tablespace khác, không phải là SYSTEM làm giảm sự tranh chấp giữa các đối tượng từ điển dữ liệu và các đối tượng người dùng trên cùng một data files. Nhìn chung, không được lưu trữ dữ liệu người dùng trên SYSTEM tablespace.

Bạn có thể sử dụng câu lệnh sql CREATE TABLESPACE Sđể tạo một default tablespace cố định hơn là SYSTEM tại thời điểm tạo ra cơ sở dữ liệu, được sử dụng làm cơ sở dữ liệu mặc định cho các đối tượng cố định. Bằng cách tách dữ liệu người dùng khỏi dữ liệu hệ thống, bạn giảm khả năng xảy ra sự cố với tablespace SYSTEM, cái mà có thể trong một số trường hợp gây ra cho toàn bộ cơ sở dữ liệu không hoạt động. Các tablespace mặc định sau đây luôn luôn không được sử dụng bởi các người dùng hệ thống, đó là, SYS, SYSTEM, và OUTLN, mà trong đó tablespace mặc định vĩnh viễn là s SYSTEM. Một tablspace được thiết kế như là mặc định vĩnh viễn thì không thể bị drop. Để đạt được mục đích này, bạn đầu tiên phải thiết kế một tablespace khác như là tablespace vĩnh viễn mặc định. Bạn có thể sử dụng câu lệnh sql ALTER TABLESPACE Sđể chỉnh sửa tablespace mặc định vĩnh viễn với một tablespace khác. Lưu ý rằng việc thay đổi này sẽ ảnh hưởng đến tất cả người dùng hoặc các đối tượng đã được tạo sau khi câu lệnh ALTER DDL được commits.

Bạn cũng có thể thiết lập default tablespace cho người dùng trong quá trình tạo người dùng, và thay đổi nó sau này với câu lệnh ALTER USER. Thay đổi tablespace mặc định của người dùng chỉ ảnh hưởng đến các đối tượng được tạo ra sau khi mọi thiết lập đã thay đổi.

Khi bạn chỉ ra tablespace mặc định cho người dùng, bạn cũng phải chỉ ra quota trên tablspace đó.

Trong câu lệnh CREATE USER dưới đây, ttablspace mặc dịnh cho người dùng jward là data\_ts, và quota của anh ta trên tablespace đó là 500K:

CREATE USER jward IDENTIFIED BY *password* **DEFAULT TABLESPACE data\_ts**

QUOTA 100M ON test\_ts

**QUOTA 500K ON data\_ts**

TEMPORARY TABLESPACE temp\_ts

PROFILE clerk;

#### Đặt tablespace quota cho người dùng

Bạn có thể gán mỗi người dùng một tablespace quota đối với bất kỳ tablespace (ngoại trừ một tablespace tạm). Việc đặt quota phải đạt những điều sau:

* Người dùng phải có quyền tạo một số loại đối tượng trong một tablespace nhất định nào đó.
* Cơ sở dữ liệu Oracle giới hạn số lượng không gian có thể được cấp phát để lưu trữ các đối tượng của người dùng trong tablespace được chỉ định cho số lượng quota.

Mặc địnht,một người dùng khong có quota trên bất kỳ tablespace nào trong cơ sở dữ liệu. Nếu người dùng có quyền để tạo đối tượng schema, thì bạn phải đặt một quota để cho phép người dùng đó tạo những đối tượng.Tối thiếu nhất, gán cho các người dùng một quota cho tablespace mặc định, và ngoài ra các quota cho những tablespace khác để chúng có thể tạo ra các đối tượng.

Trong câu lệnh CREATE USER dưới đây, đã đặt các quota cho test\_ts và data\_ts tablespace:

CREATE USER jward IDENTIFIED BY *password* DEFAULT TABLESPACE data\_ts

**QUOTA 100M ON test\_ts QUOTA 500K ON data\_ts**

TEMPORARY TABLESPACE temp\_ts

PROFILE clerk;

Bạn có thể đặt quota cho người dùng bằng một trong hai cách, một là sử dụng quota nào đó chỉ định dung lượng trên đĩa cứng trong mỗi tablespace, hoặc không giới hạn quota cho mọi tablespace. Quota chỉ định phòng tránh việc các đối tượng của người dụng sử dụng quá nhiều không gian trong cơ sở dữ liệu. Dung lượng tối đa mà bạn có thể đặt cho một tablespace là 2TB. Nếu bạn cần thêm dung lượng, thì hãy chỉ định UNLIMITED trong việc tạo QUOTA.

Bạn có thể gán các quota cho một tablespace người dùng khi bạn tạo người dùng đó, hoặc thêm và thay đổi các quota sau này. (bạn có thể tìm thấy các quota cho người dùng tồn tại bằng cách truy vấn bằng câu lệnh đến khung nhìn USER\_TS\_QUOTAS ). Nếu một quota mới nhỏ hơn cái cũ thì để thực hiện được điều đó các điều kiện dưới đây vẫn đúng:

* Nếu một người dùng đã sử dụng vượt quá mức của dung lượng quota mới, thì các đối tượng của người dùng trong tablespace không thể được cấp phát thêm dung lượng cho đến khi dung lượng cộng lại của các đối tượng là nhỏ hơn dung lượng của quota mới.
* Nếu một người dùng chưa sử dụng vượt quá mức của quota mới hoặc nếu không gian được sử dụng bởi các đối tượng của người dùng trong tablespace nhỏ hơn tablespace quota mới, thì các đối tượng của người dùng có thể được cấp phát dung lượng ở mức quota mới.

###### Hạn chế dung lượng giới hạn quota đối với các đối tượng người dùng trong một tablespace

Bạn có thể hạn chế giới hạn quota đối với các đối tượng người dùng trong một tablespace bằng cách sử dụng lệnh SQL ALTER USER để thây đổi quota hiện tại của người dùng thành 0. Sau khi quota 0 được đặt, các đối tượng của người dùng trong tablespace được giữ lại, người dùng vẫn có thể tạo được các đối tượng, nhưng các đối tượng đã tồn tại sẽ không được cấp phát dung lượng. Ví dụ, bạn không thể chèn dữ liệu vào một trong những bảng đã tồn tại của người dùng. Hành động này sẽ thất bại với một mã lỗi ORA-1536 dung lượng quota vượt mức cho phép.

###### Cấp cho người dùng UNLIMITED TABLESPACE dưới quyền hệ thống

Để cho phép người dùng sử dụng không giới hạn dung lượng của bất kỳ tablespace nào trong cơ sở dữ liệu, cấp cho người dùng đó bằng lệnh UNLIMITED TABLESPACE dưới quyền hệ.Lệnh này sẽ ghi đè các quota chỉ định cho người dùng trước đó trong tablespace. Nếu bạn thu hồi lại quyền này, thì bạn phải chỉ rõ dung lượng quota cấp cho từng tablespace. Bạn chỉ có thể cấp quyền này cho người dùng, không cấp được cho các vai trò.

Trước khi cấp quyền UNLIMITED TABLESPACE dưới quyền hệ thống, Bạn phải xem xét các hệ quả sau nếu làm việc đó.

Ưu điểm:

Bạn có thể cấp cho người dùng quyền truy cập không giới hạn vào tất cả các không gian bảng của cơ sở dữ liệu bằng một câu lệnh.

Nhược điểm:

* + Quyền này ghi đè mọi quota đã được chỉ định trong tablespace của người dùng.
  + Bạn không thể thu hồi có chọn lọc truy cập tablespace từ người dùng có đặc quyền UNLIMITED TABLESPACE. Bạn chỉ có thể cấp quyền truy cập có chọn lọc hoặc giới hạn sau khi hủy bỏ đặc quyền..

#### Đặt một tablespace tạm cho người dùng

Bạn nên gán cho mỗi người dùng một tablespace tạm. khi một người dùng thực hiện một câu lệnh SQL mà nó yêu cầu các segment tạm, Oracle database lưu trữ các segment trong tablespace tạm của người dùng. Các segment tạm này được tạo ra bởi hệ thống khi thực hiện các toán tử sắp xếp hoặc kết. Các segment tạm được sở hữu bởi SYS, cái mà có quyền tài nguyên trong mọi tablespace.

Trong phần dưới đây, tablespace của jward là temp\_ts, một tablespace đã tạo ra một cách tường minh chỉ ddể chứa các segment tạm

CREATE USER jward IDENTIFIED BY *password* DEFAULT TABLESPACE data\_ts

QUOTA 100M ON test\_ts QUOTA 500K ON data\_ts

**TEMPORARY TABLESPACE temp\_ts**

PROFILE clerk;

Để tạo ra một tablespace tạm, sử dụng lệnh CREATE TEMPORARY TABLESPACE .

Nếu bạn không chỉ rõ ra một tablespace tạm thời cho người dùng, thì cơ sở dữ liệu Oracle tự đặt cho người dùng tablespace tạm mặc định cái mà đã được mô tả ở thời điểm tạo cơ sở dữ liệu, hoặc bởi câu lệnh ALTER DATABASE ở thời điểm sau đó.Nếu không có tablespace tạm mặc định được chỉ định một cách rõ rằng, thì mặc định sẽ là SYSTEM tablespace hoặc một cài đặt mặc định khác được thiết lập bởi người quản trị hệ thống. Không được lưu trữ dứ liệu người dùng trong SYSTEM. Việc chỉ định một tablespace được sử dụng cụ thể như một tablespace tạm thời sẽ loại bỏ sự tranh chấp tệp giữa các phân đoạn tạm thời và các loại phân đoạn khác.

**Lưu ý:** Nếu SYSTEM tablespace của bạn được quản lý cục bộ, thì các người dùng buộc phải chỉ ra cụ thể tablespace mặc định tạm thời (quản lý cục bộ). Chúng không được phép mặc định để sử dụng SYSTEM tablespace bởi vì các đối tượng tạm thời không thể được đặt trong các permanen tablespace được quản lý cục bộ.

Bạn có thể thiết lập tablespace tạm thời cho một người dùng ở thời điểm tạo người dùng, và thay đổi nó sau này bằng cách sử dụng câu lệnh ALTER USER. YBạn cũng có thể thiết lập các nhóm tablespace thay vì gán các tablespace đơn lẻ.

**Đọc thêm:**

* + - "Temporary Tablespaces" trong *Oracle Database Administrator's Guide*
    - "Multiple Temporary Tablespaces: Using Tablespace Groups" trong

*Oracle Database Administrator's Guide*

#### Thiết lập một profile cho người dùng

Bạn có thể thiết lập một profile khi bạn tạo một người dùng. Một profile là một tập hợp của các giới hạn trên các tài nguyên cơ sở dữ liệu và mật khẩu truy cập vào cơ sở dữ liệu. Nếu bạn không thiết lập một profile, thì cơ sở dữ liệu Oracle tự đặt cho người dùng đó một profile mặc định.

Ví dụ dưới đây mô tả cách để thiết lập một profile người dùng.

CREATE USER jward IDENTIFIED BY *password* DEFAULT TABLESPACE data\_ts

QUOTA 100M ON test\_ts QUOTA 500K ON data\_ts

TEMPORARY TABLESPACE temp\_ts

**PROFILE clerk;**

**Đọc thêm:** ["Managing Resources with Profiles" trêng trang 2-12](#_bookmark139)

#### Thiết lập một vai trò mặc định cho người dùng

Một vai trò là trên của một nhóm các quyền liên quan mà bạn muốn cấp cho một nhóm người dùng hoặc các vai trò khác. Một vai trò mặc định được cho phép một cách tự động đối với người dùng khi khi người dụng tạo ra một phiên làm việc. Bạn có thể gán cho một người dùng không hoặc nhiều vai trò mặc định.

Bạn không thể thiết lập các vai trò mặc định cho người dùng trong câu lệnh CREATE USER. When you first create a user, the default role setting for the user is ALL, làm cho tất cả các vai trò sau đó được cấp cho người dùng là vai trò mặc định. Sử dụng câu lệnh ALTER USER để thay đổi các vai trò mặc định đối với người dùng, ví dụ:

GRANT USER jward clerk\_role;

ALTER USER jward DEFAULT ROLE clerk\_role;

Trước khi một vai trò có thể được thực hiện vai trò mặc định cho người dùng, người dùng đó phải đã được cấp vai trò đó.

**Đọc thêm:** ["Managing User Roles" trên trang 4-6](#_bookmark540)

### Chỉnh sửa các tài khoản người dùng

Phần này chứa:

* [Về](#_bookmark98) việc chỉnh sửa các tài khoản người dùng
* [Sửu dụng lệnh ALTER USER Statement để](#_bookmark100) chỉnh sửa tài khoản người dùng
* Thay đổi mật khẩu người dùng không phải là sys
* [Thay](#_bookmark107) đổi mật khẩu người dùng là sys

#### Về việc chỉnh sửa các tài khoản người dùng

Các người dùng có thể thay đổi mật khẩu của riêng họ. Tuy nhiên, để thay đổi bất kỳ tùy chọn nào khác của miền bảo mật người dùng, Bạn phải có quyền hệ thống ALTER USER . Các nhà quản trị bảo mật thường là những người duy nhất có quyền hệ thống, và cái quyền này cho phép sửa đổi trên bất kỳ miền bảo mật người dùng nào. Quyền này bao gồm khả năng để thiết lập tablespace quota cho người dùng trên bất kỳ tablespace nào trong cơ sở dữ liệu, ngay cả khi người dùng thực hiện sửa đổi không có một quota đối với tablespace nhất định.

#### Sử dụng ALTER USER Statement để thay đổi tài khoản người dùng

Bạn có thể chỉnh sửa thiết lập bảo mật của người với câu lệnh sql ALTER USER. Việc thay đổi thiết lập bảo mật người dùng ảnh hướng đến các phiên làm việc tương lai, không phải các phiên làm việc hiện tay.

[Ví dụ 2–2](#_bookmark102) cho thấy cách để sử dụng lệnh ALTER USER để chỉnh sửa thiết lập bảo mật đối với người dùng avyrros:

***Ví dụ 2–2 Chỉnh sửa một tài khoản người dùng***

ALTER USER avyrros IDENTIFIED EXTERNALLY DEFAULT TABLESPACE data\_ts

TEMPORARY TABLESPACE temp\_ts

QUOTA 100M ON data\_ts QUOTA 0 ON test\_ts PROFILE clerk;

Câu lệnh ALTER USER sở đây thay đổi thiết lập bảo mật đối với người dùng avyrros as như sau:

* + Xác thực được thay đổi để sử dụng tài khoản hệ điều hành đối với người dùng avyrros.
  + Tablespace mặc định và tạm thời được thiết lập rõ ràng đối với người dùng AVYRROS.
  + Người dùng avyrros is được trao cho 100M quota đối với DATA\_TS tablespace.
  + The quota của người dùng avyrros trên test\_ts is bị thu hồi.
  + Người dùng avyrros is được gán clerk profile.

#### Changing Non-SYS User Passwords

Hầu hết mọi người dùng có thể thay đổi mật khẩu của chính họ với câu lệnh PASSWORD, như sau:

PASSWORD andy

Changing password for andy New password: *password*

Retype new password: *password*

Không cần quyền đặc biệt (ngoài các kết nối vào cơ sở dữ liệu và tạo phiên) bị bắt buộc đối với một người dùng để thay đổi mật khẩu của chính họ. Khuyến khích người dùng thay đổi mật khẩu của họ thường xuyên. ["Guidelines for Securing Passwords" trên trang 10-7](#_bookmark2167) cung cấp lời khuyên cách tốt nhất để bảo mật mật khẩu. Bạn có thể tìm thấy các người dùng tồn tại trong thể hiện của cơ sở dữ liệu hiện tại bằng cách truy vấn khung nhìn ALL\_USERS.

Các người dùng cũng có thể sử dụng câu lệnh SQL ALTER USER thay đổi mật khẩu của họ. Ví dụ:

ALTER USER andy IDENTIFIED BY *password*

Tuy nhiên, để bảo mật tốt hơn, sử dụng câu lệnh PASSWORD để thay đổi mật khẩu của tài khoản. Câu lệnh ALTER USER shiển thị mật khẩu mới trên màn hình, nơi mà nó có thể bị nhìn thấy bởi các đồng nghiệp quá tò mò. Câu lệnh PASSWORD command không hiển thị mật khẩu mới, nên chỉ có bạn mới biết, không phải các đồng nghiệp của bạn. Trong hai trường hợp, mật khẩuu được mã hóa trên đường mạng.

Các người dùng phải có quyền PASSWORD và ALTER USER để chuyển đổi giữa các phương pháp xác thực. Thông thường, Chỉ có nhà quả trị mới có quyền này.

**Đọc thêm:**

* + - ["Minimum Requirements for Passwords" on page 3-3](#_bookmark197) for the minimum requirements for creating passwords
    - ["Guidelines for Securing Passwords" on page 10-7](#_bookmark2167) for additional ways to secure passwords
    - [Chapter 3, "Configuring Authentication"](#_bookmark180) for information about authentication methods that are available for Oracle Database users

#### Changing the SYS User Password

Nếu bạn muốn thay đổi mật khẩu của người dùng SYS, thì bạn nên sử dụng tiện ích dòng lệnh ORAPWD để tạo mới một tập tin mật khẩu cái mà chứa mật khẩu mà bạn muốn sử dụng. Không được sử dụng câu lệnh ALTER USER s hoặc câu lệnh PASSWORD để thay đổi mật khẩu của người dùng SYS. Lưu ý những điều sau đây:

* Tài khoản người dùng SYS được sử dụng bởi hầu hết SQL đệ quy nội bộ. Do đó, nếu bạn thử sử dụng câu lệnh ALTER USER để thay đổi mật khẩu trong khi cơ sở dữ liệu đang được mở, thì sẽ có một cơ hội mà deadlock sẽ diễn ra.
* Nếu bạn thử sử dụng ALTER USER để thay đổi mật khẩu người dùng SYS, và nếu có thể hiện tham số khởi tạo REMOTE\_LOGIN\_PASSWORDFILE hđã được thiết lập thành SHARED, tthì bạn không thể thay đổi mật khẩu của SYS.Câu lệnh ALTER USER sẽ thất bại với một lỗi

ORA-28046: Lỗi thay đổi mật khẩu cho SYS không được phép.

[Ví dụ 2–3](#_bookmark112) scho thấy cách để sử dụng ORAPWD để tạo một file mật khẩu cái mà có một mật khẩu mới cho SYS. Trong ví dụ này, mật khẩu mới sẽ được lưu trữ trong file mật khẩu được gọi là orapworcl. ( Nếu file mật khẩu đã tồn tại, thì mỗi lỗi OPW-00005: File with same name exists - please delete or rename elỗi này cảnh báo bạn vì thế bạn có thể chọn một tên khác. Nếu bạn muốn ghi đè lên file mật khẩu hiện tại, thì thêm vào argument force=y vào lệnh ORAPWD.)

***Ví dụ 2–3 Sử dụng ORAPWD để thay đổi mật khẩu của người dùng SYS***

orapwd file='orapworcl'

Enter password for SYS: *new\_password*

**Đọc thêm:** *Oracle Database Administrator's Guide* for detailed information about the orapwd command syntax and arguments

### Cấu hình các giới hạn tài nguyên người dùng

This section contains:

* [About User Resource Limits](#_bookmark114)
* [Types of System Resources and Limits](#_bookmark117)
* [Determining Values for Resource Limits of Profiles](#_bookmark136)
* [Managing Resources with Profiles](#_bookmark139)

#### About User Resource Limits

Bạn có thể thiết lập các giới hạn trên mốt số lượng các tài nguyên khả dụng hệ thống khác nhau đối với mỗi người dùng như là một phần của miền bảo mật của người dùng đó. Bằng cách làm như thế, bạn có thể tránh khỏi sự tiêu sự không kiểm soát các tài nguyên hệ thống quý giá như là CPU time. Để thiết lập các giới hạn tài nguyên, bạn sử dụng Database Resource Manager, cái mà được mô tả trong *Oracle Database Administrator's Guide*.

Tính năng giới hạn tài nguyên là rất hữu dụng trong các hệ thống lớn, đa người dùng, nơi mà các tài nguyên hệ thống là rất đắt đỏ. Sự tiêu thục quá mức các tài nguyên bởi một hoặc nhiều người dùng có thể ảnh hưởng bất ngờ đến các người dùng khác trong cơ sở dữ liệu. Trong các hệ thống cơ sở dữ liệu một người dùng hoặc đa người dùng quy mô nhỏ, tính năng tài nguyên hệ thống là không quan trọng, bởi vì sự tiêu thụ tài nguyên hệ thống của người là rất ít để dẫn đến ảnh hưởng bất ngờ..

Bạn quản lý các giới hạn tài nguyên người dùng bằng cách sử dụng Database Resource Manager. Bạn có thể thiết lập tùy chọn quản lý mật khẩu sử dụng profile, hoặc thiết lập riêng lẻ hoặc sử dụng profile mặc định cho rất nhiều người dùng. Mỗi cơ sở dữ liệu Oracle có thể có một số lượng không giới hạn các profile.Cơ sở dữ liệu Oracle cho phép nhà quản trị bảo mật cho phép hoặc không cho phép các giới hạn thực thi của tài nguyyên profile một cách phổ thông.

Thiết lập giới hạn tài nguyên gây ra một sự suy giảm hiệu suất nhẹ khi các người dùng tạo các phiên làm việc, bởi vì cơ sở dữ liệu Oracle nạp tất cả giới hạn dữ liệu tài nguyên cho mỗi người dùng khi mỗi kết nối đến cơ sở dữ liệu.

**Đọc thêm:** *Oracle Database Administrator's Guide* for detailed information about managing resources

#### Types of SystemResources and Limits

Cơ sở dữ liệu Oracle có thể giới hạn việc sử dụng một số loại tài nguyên hệ thống, bao gồm CPU time và logical reads. Nhìn chung, bạn có thể điều khiểu các tài nguyên này ở tại cấp độ phiên làm viên, call level, hoặc cả hai, được giải thích ở các phần dưới đây:

* + [Limiting the User Session Level](#_bookmark119)
  + [Limiting Database Call Levels](#_bookmark121)
  + [Limiting CPU Time](#_bookmark123)
  + [Limiting Logical Reads](#_bookmark125)
  + [Limiting Other Resources](#_bookmark127)

###### Limiting the User Session Level

Mỗi lần một người dùng kết nối đến một cơ sở dữ liệu, một phiên làm việc được tạo ra. Mỗi phiên làm việc sử dụng CPU time và bộ nhớ trên máy tính cái mà chạy cơ sở dữ liệu Oracle. Bạn có thể thiết lập một vài giới hạn tài nguyên ở cấp độ phiên làm việc.

Nếu một người dùng vượt quá giới hạn tài nguyên ở cấp độ phiên làm việc, thì cơ sở dữ liệu Oracle ngắt (rolls back) câu lệnh hiện tại và trả về thông bán chỉ ra rằng giới hạn phiên làm việc đã vượt mức cho phép. Tại điểmn ày, Các câu lệnh trước đó trong giao tác hiện tại đều nguyên vẹn, và hành động duy nhất mà người dùng có thể thực hiện là COMMIT, ROLLBACK hoặc disconnect (Trong trường hợp này, giao tác hiện tại sẽ được commit). Các hành động khác sẽ sinh ra lỗi. Ngay cả khi giao tác được commit hoặc rolled back, người dùng khnogo thể hoàn thành bất kỳ công việc nào nữa trong phiên làm việc hiện tại.

###### Limiting Database Call Levels

Mỗi lần một người dùng chạy câu lệnh SQL, cơ sở dữ liệu Oracle thực hiện một vài bức để thực hiện câu lệnh. Trong quá trình xử lý, một vài lời gọi được thực hiện đến cơ sở dữ liệu như là một phần của các giai đoạn thực hiện khác nhau. Để tránh bất kỳ một lời gọi bằng cách sử dụng hệ thống quá nhiều, cơ sở dữ liẹu Oracle cho phép bạn thiết lập một vài các giới hạn tài nguyên ở tại cấp độ lời gọi.

Nếu một người dùng vượt quá giới hạn tài nguyên ở mức lời gọi, thì cơ sở dữ liệu Oracle tạm dừng quá trình xử lý lệnh, roll back lệnh, và trả về một lỗi. Tuy nhiên, Các câu lệnh trước đó của giao tác hiện tại vẫn còn nguyên vẹnn, và phiên làm việc người dùng vẫn còn được kết nối.

###### Limiting CPU Time

Khi các câu lệnh SQL và các kiểu khác của lời gọi được thực hiện trên cơ sở dữ liệu Oracle, một lượng CPU time là cần thiết để xử lý lời gọi. Các lời gọi trung bình yêu cầu một lượng nhỏ CPU time. Tuy nhiên, một câu lệnh SQL liên quan đến một lượng lớn dữ liệu hoặc câu truy vấn có thể sử dụng một lượng lớn CPU time, làm giảm thời gian CPU có sẵn để làm vivệc khác.

Để tránh việc không kiểm soát được sử dụng của CPU time, bạn có thể thiết lập các giới hạn cố định hoặc động trên CPU time cho mỗi lần gọi và tổng thời gian CPU time cho cơ sở dữ liệu Oracle trong suốt phiên làm việc. Các giới hạn được thiết lập và đo lường trong một phần ngàn giây được sử dụng bởi một lời gọi hoặc một phiên làm việc.

###### Limiting Logical Reads

Nhập/Xuất (I/O) là một trong những các hoạt động tốn kém nhất trong hệ thống cơ sở dữ liệu. Các câu lệnh SQL nghiêng về I/O có thể độc sử dụng quyền bộ nhớ và đĩa cứng và gây ra các hoặc động cơ sở dữ liệu khác để cạnh tranh với nhau những tài nguyên này.

Để tránh việc các nguồn đơn sử dụng quá mức I/O, bạn có thể thiết lập logical data block đọc được cho mỗi lời gọi và cho mỗi phiên làm việc. Logical data block đọc bao gồm data block đọc từ bộ nhớ hoặc đãi cứng. Các giới hạn này được thiết lập và đo lường bằng số lượng block được đọc thực hiện bởi một lời gọi hoặc trong suốt phiên làm việc.

###### Limiting Other Resources

Cơ sở dữ liệu Oracle cung cấp một vài giới hạn tài nguyên khác ở mức phiên làm việcl:

* + - **Bạn có thể giới hạn số lượng phiên làm việc đồng thời cho mỗi người dùng.** Mỗi người dùng chỉ có thể tạo ra lên đến một số đã định nghĩa trước số lượng phiên làm việc đồng thời.
    - **Bạn có thể giới hạn thời gian nhàn rỗi cho một phiên làm việc.** Nếu thời gian giữa các lần gọi trong một phiên làm việc đạt đến giới hạn thời gian nhàn rỗi, thì giao tác hiện tại sẽ được rolled back, phiên làm việc kết thúc, và các tài nguyên của phiên làm việc được trả về cho hệ thống. Lần gọi tiếp theo sẽ nhận được một lỗi chỉ ra rằng người dùng không còn kết nối vào thể hiện này. Giới hạn này được thiết lập với đơn vị tính là số phút đã trải qua.

**Lưu ý:**Ngay sau khi một phiên bị chấm dứt bởi vì nó đã vượt quá giới hạn thời gian không hoạt động, quá trình nền của quá trình giám sát (PMON) dọn dẹp sau khi phiên kết thúc. Cho đến khi PMON hoàn tất quá trình này, phiên chấm dứt vẫn được tính trong bất kỳ phiên hoặc giới hạn tài nguyên người dùng nào.

* + - **Bạn có thể giới hạn thời gian kết nối đã trôi qua cho mỗi phiên làm việc.**Nếu trong suốt phiên làm việc thời gian vượt quá thời gian giới hạn đã trôi qua, thì giao tác hiện tại lúc đó sẽ được rolled back, phiên làm việc sẽ bị hủy, và các tài nguyên của phiên là việc được trả về cho hệ thống. Giới hạn này được thiết lập như là số phút đã trôi qua.

**Lưu ý:** Cơ sở dữ liệu Oracle không liên tục theo dõi thời gian không hoạt động đã trôi qua hoặc thời gian kết nối đã trôi qua. Làm như vậy làm giảm hiệu năng hệ thống. Thay vào đó, nó sẽ kiểm tra sau vài phút. Do đó, một phiên có thể vượt quá giới hạn này một chút (ví dụ, 5 phút) trước khi Cơ sở dữ liệu Oracle thi hành giới hạn và chấm dứt phiên.

* + **Bạn có thể giới hạn dung lượng của khung vực hệ thống riêng từ toàn cầu (SGA) được sử dụng cho các khu sql riêng tư cho một phiên làm việc.** Giới hạn này chỉ quan trọng trong các hệ thống cái mà sử dụng máy chủ cấu hình chia sẻ. Ngược lại, khu vực hệ thống sql riêng tư được nằm trong khu vực chương trình toàn cục(PGA). Giới hạn này là một tập hợp các byes dữ liệu trong thể hiện của SGA. Sử dụng các kí tự K hoặc M để chỉ ra đó là kilobyes hay megabytes.

**Đọc thêm:** For instructions about enabling or disabling resource limits:

* + - ["Finding Information About Database Users and Profiles" on page 2-15](#_bookmark164)
    - ["Managing User Roles" on page 4-6](#_bookmark540)
    - *Oracle Database Administrator's Guide* for detailed information about managing resources

#### Determining Values for Resource Limits of Profiles

Trước khi tạo ra các profiles và thiết lập các giới hạn tài nguyên liên kết với chúng, bạn nên xác định các giá trị thích hợp cho mỗi giới hạn tài nguyên. Bạn có thể dựa trên những giá trị trên các loại hoạt động mà người dùng điển hình thực hiện. Ví dụ, nếu một lớp của người dùng không thường xuyên thực hiện việc đọc một số lượng lớn logical data block, thì sử dụng câu lệnh sql ALTER RESOURCE COST để thiết lập cài đặt LOGICAL\_READS\_PER\_SESSION một cách thận trọng.

Thông thường, cách tốt nhất để xác định các giá trị giới hạn tài nguyên thích hợp cho một profile người dùng là thu thập các thông tin lịch sử về mỗi loại sử dụng tài nguyên. Ví dụ, cơ sở dữ liệu hoặc nhà quản trị bảo mật có thể dụng mệnh đề AUDIT SESSION để thu thập thông tin về các giới hạn, CONNECT\_TIME, LOGICAL\_READS\_PER\_SESSION.

Bạn có thể thu thập cấc thống kê cho các giới hạn khác bằng cách sử dụng tính năng Monitor của Orracle Enterpise Mangager ( hoặc SQL\*Plus), cụ thể là theo dõi thống kê(Statistics monitor).

**See Also:**

* + - ["Using Data Dictionary Views to Find Information About Users and Profiles" on page 2-15](#_bookmark166)
    - [Chapter 9, "Verifying Security Access with Auditing"](#_bookmark1602)
    - *Oracle Database 2 Day DBA* for more information about Database Control
    - Enterprise Manager online Help for more information about the Monitor feature

#### Managing Resources with Profiles

Một **profile** là tên của một tập hợp các giới hạn về tài nguyên và các tham số về mật khẩu cái mà hạn chế việc sử dụng cơ sở dữ liệu và tài nguyên đối với người dùng. Bạn có thể gán một profile với mỗi một người dùng, và một profile mặc định đến với mọi người khác. Mỗi người dùng có thể chỉ có một profile, và việc tạo mới một cái sẽ thay thế cho phiên bản cũ hơn.

Bạn cần phải tạo và quản lý các profile người dùng chỉ khi các giới hạn tài nguyên là một yêu cầu của chính sách bảo mật cơ sở dữ liệu của bạn. Để sử dụng profiles, đầu tiên phân loại các người dùng có liên quan trong cơ sở dữ liệu. Cũng giống như vai trò được sử dụng để quản lý đặc quyền của người dùng có liên quan,profile được sử dụng để quản lý giới hạn tài nguyên của người dùng có liên quan. Xác định có bao nhiêu profile cần thiết để bao phủ tất cả các loại người dùng trong cơ sở dữ liệu và sau đó xác định các giới hạn tài nguyên thích hợp cho mỗi profile.

Nhìn chung, từ profile đề cập đến một tập hợp các thuộc tính áp cho một người dùng nào đó, cho phép một điểm tham chiếu duy nhất mà thông qua đó nhiều người dùng có cùng những thuộc tính này. Các profile trong Oracle Internet Directory chứa các thuộc tính cần thiết để sử dụng và xác thực cho mỗi người dùng. Tương tự, các profile trong Oracle Label Security chứa các thuộc tính hữu ích trong quản lý bảo mật theo nhãn và quản lý các hoạt động. Các thuộc tính trong profile có thể bảo gồm các hạn chế trên các giới hạn tài nguyên hệ thống. Bạn có thể sử dụng Database Resource Manager để thiết lập các loại giới hạn tài nguyên.

Các giới hạn tài nguyên profile được thực thi chỉ khi bạn bật các giới hạn tài nguyên liên kết với cơ sở dữ liệu. Việc cho phép giới hạn này có thể xảy ra trước khi database khởi động ( sử dụng các tham số khởi trạo trong RESOURCE\_LIMIT) hoặc trong khi database ở trạng thái mở ( sử dụng lệnh ALTER SYSTEM).

Mặc dù các tham số mật khẩu nằm bên trong các profile, chúng không bị ảnh hưởng bởi RESOURCE\_LIMIT hoặc ALTER SYSTEM và quản lý mật khẩu luôn được bật. Trong cơ sở dữ liệu Oracle, Database Resource Manager chủ yếu xử lý các phân bổ và hạn chế tài nguyên.

**Đọc thêm:**

* *Oracle Database Administrator's Guide* for detailed information on managing resources
* ["Finding Information About Database Users and Profiles" on page 2-15](#_bookmark164) for viewing resource information
* *Oracle Database SQL Language Reference* for information about

ALTER SYSTEM or RESOURCE\_LIMIT

###### Tạo ra các profile

Bất kỳ người dùng được ủy quyền đều óc thể tạo, gán cho người dùng, chỉnh sửa, và xóa một profile ở bất kỳ thời điểm nào ( sử dụng lệnh CREATE USER hoặc ALTER USER). Các profile có thể được gán cho những người dùng và không thể gán được cho các role (vai trò) và các profile khác. Việc gán profile không ảnh hưởng đến các phiên làm việc hiện tại, thay vào đó, nó sẽ có hiệu lực ở các phiên làm việc tiếp theo. Lưu ý rằng khi bạn gán một profile cho một người dùng bên ngoài (external user) hoặc người dùng toàn cục (global user), các tham số về mật khẩu không bị ảnh hưởng cho người dùng đó..

Để tìm thêm thông tin về các profile hiện có, truy vấn khung nhìn DBA\_PROFILES.

**Đọc thêm:**

* *Oracle Database SQL Language Reference* for more information about the SQL statements that are used to manage profiles, which are ALTER PROFILE, CREATE PROFILE, and DROP PROFILE
* *Oracle Database Administrator's Guide* for detailed information about managing resources
* ["Creating User Accounts" on page 2-1](#_bookmark59)
* ["Altering User Accounts" on page 2-7](#_bookmark97)

###### Dropping Profiles

Để xóa một profile, bạn phải có quyền hệ thống DROP PROFILE. Bạn có thể xóa một profile (hơn là default profile) thông qua câu lệnh sql DROP PROFILE. Để xóa một profile hiện đang gán cho một người dùng khác một cách thành cônhg sử dụng thêm tùy chọn CASCADE..

Câu lệnh sau đây xóa một profile tên clerk, mặc dù nó đã được gán cho người dùng:

DROP PROFILE clerk CASCADE;

Bất kỳ người dùng hiện tại lúc đó đã được gán vào một profile mà cái đó bị xóa thì nó sẽ tự động gán vào DEFAULT profile. DEFAULT profile không thể bị xóa. Khi một profile bị xóa, việc xóa profile đó không ảnh hưởng lập tức đến phiên làm việc hiện tại. Chỉ những phiên làm việc được tạo sau khi một profile bị xóa sẽ sử dụng profile đã được gán mới.

### Deleting User Accounts

Khi bạn xóa một tài khoản người dùng, cơ sở dữ liệu Oracle xóa tài khoản người dùng và các schema liên kết với người dùng đó từ data dictionary. nó cũng ngay lập tức xóa hết tất cả các đối tượng schema chứa trong schema người dùng, nếu có.

**Lưu ý:**

* + - Nếu bạn muốn schema người dùng liên kết với các đối tượng phải được giữ lại nhưng mà bạn muốn người dùng bị từ chối truy cập vào cơ sở dữ liệu, thì nên sử dụng quyền CREATE SESSION sẽ bị thu hồi lại..
    - Đừng cố thử xóa người dùng SYS hoặc SYSTEM. Làm như thế sẽ gây hại cho cơ sở dữ liệu của bạn.

Một người dùng hiện lúc đó kết nối vào một cơ sở dữ liệu thì không thể bị xóa. Để xóa một người dùng kết nối, bạn đầu tiên phải ngắt các phiên làm việc của người dùng đó thông qua câu lệnh SQL ALTER SYSTEM với mệnh đề KILL SESSION. Bạn có thể tìm thấy mã phiên làm viên (SID) bằng câu truy vấn với khung nhìn V$SESSION.

[Ví dụ 2–4](#_bookmark156) cho thấy cách để truy vấn V$SESSION và thể hiện ra mã phiên làm việc, số thứ tự, và tên người dùng đối với người dùng ANDY.

***Ví dụ 2–4 Truy vấn V$SESSION để lấy được mã phiên làm việc của người dùng***

SELECT SID, SERIAL#, USERNAME FROM V$SESSION;

SID SERIAL# USERNAME

------- --------------- ----------------------

127 55234 ANDY

...

[Ví dụ 2–5](#_bookmark158) cho thấy cách để dừng một phiên làm việc của người dùng andy.

***Ví dụ 2–5 Dừng phiên làm việc của người dùng***

ALTER SYSTEM KILL SESSION '127, 55234';

Bạn có thể xóa một người dùng khỏi cơ sở dữ liệu thông qua câu lệnh DROP USER. để xóa một người dùng và tất cả các đối tượng schema ( nếu có), bạn phải xóa (DROP USER) người dùng với quyền hết thống. Bởi vì DROP USER dưới quyền hệ thống rất mạnh, một nhà quản trị bảo mật thường điển hình là người có quyền này..

Nếu schema của người dứa chứa bất kỳ các đối tượng schema phụ thuộc, thì sử dụng tùy chọn CASCADE để xóa người dùng và tất cả các đối tượng liên kết và các khóa ngoại mà phụ thuộc vào các bảng của người dùng một cách thành công. Nếu bạn không chỉ ra tùy chọn CASCADE và schema người dùng chứa các đối tượng phụ thuộc, thì thông báo lỗi được trả về và người dùng không thể bị xóa.

Trước khi xóa amột người dùng mà các schema của người đó chứa các đối tượng, thì hãy xem xét thật cẩn thận các đối tượng nào được schema chứa và tác động của việc xóa đi chúng. Bạn có thể tìm các đối tượng được sở hữu bởi một người dùng nào đó bằng cách truy vấn khung nhìn DBA\_OBJECTS.

[Ví dụ 2–6](#_bookmark160) cho thấy cách để tìm các đối tượng sở hữu bởi người dùng andy.

***Ví dụ 2–6 Tìm kiếm các đối tượng bị sở hữu bởi một người dùng***

SELECT OWNER, OBJECT\_NAME FROM DBA\_OBJECTS WHERE OWNER LIKE 'ANDY';

(Nhập tên người dùng bằng các kí tự viết hoa). Hãy chú ý đến bật kỳ ảnh hưởng của hiệu ứng xếp tầng. Ví dụ, nếu bạn có ý để xóa một người dùng người mà sở hữu một bảng, thì hãy kiểm tra xem có bất kỳ khung nhìn hoặc thủ tục nào phụ thuộc vào bảng đó hay không.

[Ví dụ 2–7](#_bookmark163) dXóa người dùng andy và tất cả các đối tượng được liên kết và các khóa ngoại mà phụ thuọc vào các bảng sở hữu bởi andy.

***Ví dụ 2–7 Xóa một tài khoản người dùng***

DROP USER andy CASCADE;

**Đọc thêm:** *Oracle Database Administrator's Guide* for more information about terminating sessions

### Finding Information About Database Users and Profiles

This section contains:

* [Using Data Dictionary Views to Find Information About Users and Profiles](#_bookmark166)
* [Listing All Users and Associated Information](#_bookmark171)
* [Listing All Tablespace Quotas](#_bookmark173)
* [Listing All Profiles and Assigned Limits](#_bookmark175)
* [Viewing Memory Use for Each User Session](#_bookmark178)

#### Using Data Dictionary Views to Find Information About Users and Profiles

[Bảng 2–1](#_bookmark170) Liệt kê các khung nhìn từ điển dữ liệu cái mà chứa thông tin về các người dùng cơ sở dữ liệu và các profile. Để biết thêm chi tiết về những khung nhìn này, đọc Oracle Database Reference.

***Bảng 2–1 Các khung nhìn dữ liệu cái mà hiển thị thông về người dùng và profile***

**Khung nhìn Mô tả**

ALL\_OBJECTS Mô tả tất cả các đối tượng có thể truy cập từ người dùng hiện tại

ALL\_USERS Liệt kê tất cả các người dùng mà người dùng hiện tại có thể nhìn thấy, nhưng không có mô tả về họ

DBA\_PROFILES Hiển thị tất cả các profile và sự giới hạn của nó

DBA\_TS\_QUOTAS Mô tả các tablespace quota cho mọi người dùng

DBA\_OBJECTS Mô tả các đối tượng trong cơ sở dữ liệu

DBA\_USERS Mô tả tất cả người dùng trong cơ sở dữ liệu

***Table 2–1 (Cont.) Các khung nhìn dữ liệu cái mà hiển thị thông về người dùng và profile***

**Khung nhìn Mô tả**

DBA\_USERS\_WITH\_DEFPWD Liệt kê tất cả tài khoản người dùng dùng mật khẩu mặc định

PROXY\_USERS Mô tả các người dùng người mà có thể giả định giống như các người dùng khác

RESOURCE\_COST Liệt kê chi phí cho mỗi tài nguyên về CPU đối với mỗi phiên làm việc, về

việc đọc cho mỗi phiên, số lần kết nối và SGA

USER\_PASSWORD\_LIMITS Mô tả các tham số về mật khẩu profile được gán cho người dùng

USER\_RESOURCE\_LIMITS Hiển thị các giới hạn về tài nguyên đối với người dùng hiện tại

USER\_TS\_QUOTAS Mô tả các tablespace quota cho các người dùng

USER\_OBJECTS Mô tả mọi đường tượng được sở hữu bởi người dùng hiện tại

USER\_USERS Chỉ mô tả về người dùng hiện tại

V$SESSION Liệt kê thông tin về phiên làm việc với mỗi phiên làm việc hiện tại, bao gồm tên người dùng

V$SESSTAT Liệt kê các thông kê về phiên làm việc người dùng

V$STATNAME Hiển thị các tên được giải mã đsôi với các thống kê được thể hiện trong

khung nhìn V$SESSTAT

Trong các phần dưới đây thể hiện các ví dụng sử dụng những khung nhìn này. Các ví dụ này giả định rằng các lệnh đã được chạy:

CREATE PROFILE clerk LIMIT SESSIONS\_PER\_USER 1

IDLE\_TIME 30

CONNECT\_TIME 600;

CREATE USER jfee

IDENTIFIED BY *password* DEFAULT TABLESPACE users TEMPORARY TABLESPACE temp\_ts

QUOTA 500K ON users PROFILE clerk;

CREATE USER dcranney IDENTIFIED BY *password* DEFAULT TABLESPACE users

TEMPORARY TABLESPACE temp\_ts

QUOTA unlimited ON users;

CREATE USER userscott IDENTIFIED BY *password*;

#### Listing All Users and Associated Information

Để tìm mọi người dùng và các thông tin liên qua đã được định nghĩa trong cơ sở dữ liệu, truy vấn khung nhìn DBA\_USERS. Để biết thêm thông tin chi tiết trên khung nhìn DBA\_USERS, đọc thêm Oracle Database Reference.

Ví dụ:

SELECT USERNAME, PROFILE, ACCOUNT\_STATUS, AUTHENTICATION\_TYPE FROM DBA\_USERS;

USERNAME PROFILE ACCOUNT\_STATUS AUTHENTICATION\_TYPE

--------------- --------------- --------------- -------------------

|  |  |  |  |
| --- | --- | --- | --- |
| SYS | DEFAULT | OPEN | PASSWORD |
| SYSTEM | DEFAULT | OPEN | PASSWORD |
| USERSCOTT | DEFAULT | OPEN | PASSWORD |
| JFEE | CLERK | OPEN | GLOBAL |
| DCRANNEY | DEFAULT | OPEN | EXTERNAL |

#### Listing All Tablespace Quotas

Sử dụng khung nhìn DBA\_TS\_QUOTA để liệt kê mọi tablespace quota đã được gán cho từng người dùng một cách cụ thể. ( Để biết thêm thông tin chi tiết trên khung nhìn này, đọc Oralce Database Reference.) Ví dụ:

SELECT \* FROM DBA\_TS\_QUOTAS;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TABLESPACE | USERNAME | BYTES | MAX\_BYTES | BLOCKS | MAX\_BLOCKS |
| ---------- | --------- | -------- | ---------- | ------- | ---------- |
| USERS | JFEE | 0 | 512000 | 0 | 250 |
| USERS | DCRANNEY | 0 | -1 | 0 | -1 |

Khi quota cụ thể nào đó được gán, con số chính xác được chỉ định trong cột MAX\_BYTES. Con số này luôn là một bội số của database block size, vì thế nếu bạn đặt tablespace quota không phải là bội số của database block size, thì nó sẽ được làm tròn lên dựa vào con số đó. Quota không giới hạn được chỉ định bằng số -1.

#### Listing All Profiles and Assigned Limits

Khung nhìn DBA\_PROFILE liệt kê tất cả các profile trong cơ sở dữ liệu và liên kết các thiết lập về giới hạn cho từng profile. (Để biết thêm thông tin về khung nhìn này, đọc Oracle Database Reference. Ví dụ:

SELECT \* FROM DBA\_PROFILES ORDER BY PROFILE;

|  |  |  |  |
| --- | --- | --- | --- |
| PROFILE | RESOURCE\_NAME | RESOURCE\_TYPE | LIMIT |
| ----------------- | ------------------------- | ------------- | -------------- |
| CLERK | COMPOSITE\_LIMIT | KERNEL | DEFAULT |
| CLERK | FAILED\_LOGIN\_ATTEMPTS | PASSWORD | DEFAULT |
| CLERK | PASSWORD\_LIFE\_TIME | PASSWORD | DEFAULT |
| CLERK | PASSWORD\_REUSE\_TIME | PASSWORD | DEFAULT |
| CLERK | PASSWORD\_REUSE\_MAX | PASSWORD | DEFAULT |
| CLERK | PASSWORD\_VERIFY\_FUNCTION | PASSWORD | DEFAULT |
| CLERK | PASSWORD\_LOCK\_TIME | PASSWORD | DEFAULT |
| CLERK | PASSWORD\_GRACE\_TIME | PASSWORD | DEFAULT |
| CLERK | PRIVATE\_SGA | KERNEL | DEFAULT |
| CLERK | CONNECT\_TIME | KERNEL | 600 |
| CLERK | IDLE\_TIME | KERNEL | 30 |
| CLERK | LOGICAL\_READS\_PER\_CALL | KERNEL | DEFAULT |
| CLERK | LOGICAL\_READS\_PER\_SESSION | KERNEL | DEFAULT |
| CLERK | CPU\_PER\_CALL | KERNEL | DEFAULT |
| CLERK | CPU\_PER\_SESSION | KERNEL | DEFAULT |
| CLERK | SESSIONS\_PER\_USER | KERNEL | 1 |
| DEFAULT | COMPOSITE\_LIMIT | KERNEL | UNLIMITED |
| DEFAULT | PRIVATE\_SGA | KERNEL | UNLIMITED |
| DEFAULT | SESSIONS\_PER\_USER | KERNEL | UNLIMITED |
| DEFAULT | CPU\_PER\_CALL | KERNEL | UNLIMITED |
| DEFAULT | LOGICAL\_READS\_PER\_CALL | KERNEL | UNLIMITED |
| DEFAULT | CONNECT\_TIME | KERNEL | UNLIMITED |
| DEFAULT | IDLE\_TIME | KERNEL | UNLIMITED |
| DEFAULT | LOGICAL\_READS\_PER\_SESSION | KERNEL | UNLIMITED |
| DEFAULT | CPU\_PER\_SESSION | KERNEL | UNLIMITED |

|  |  |  |  |
| --- | --- | --- | --- |
| DEFAULT | FAILED\_LOGIN\_ATTEMPTS | PASSWORD | 10 |
| DEFAULT | PASSWORD\_LIFE\_TIME | PASSWORD | 180 |
| DEFAULT | PASSWORD\_REUSE\_MAX | PASSWORD | UNLIMITED |
| DEFAULT | PASSWORD\_LOCK\_TIME | PASSWORD | 1 |
| DEFAULT | PASSWORD\_GRACE\_TIME | PASSWORD | 7 |
| DEFAULT | PASSWORD\_VERIFY\_FUNCTION | PASSWORD | UNLIMITED |
| DEFAULT | PASSWORD\_REUSE\_TIME | PASSWORD | UNLIMITED |
| 32 rows selected. |  |  |  |

Để tìm các giá trị profile mặc định, chạy câu truy vấn dưới đây:

SELECT \* FROM DBA\_PROFILES WHERE PROFILE = 'DEFAULT';

|  |  |  |  |
| --- | --- | --- | --- |
| PROFILE | RESOURCE\_NAME | RESOURCE\_TYPE | LIMIT |
| ----------------- | ------------------------- | ------------- | -------------- |
| DEFAULT | COMPOSITE\_LIMIT | KERNEL | UNLIMITED |
| DEFAULT | SESSIONS\_PER\_USER | KERNEL | UNLIMITED |
| DEFAULT | CPU\_PER\_SESSION | KERNEL | UNLIMITED |
| DEFAULT | CPU\_PER\_CALL | KERNEL | UNLIMITED |
| DEFAULT | LOGICAL\_READS\_PER\_SESSION | KERNEL | UNLIMITED |
| DEFAULT | LOGICAL\_READS\_PER\_CALL | KERNEL | UNLIMITED |
| DEFAULT | IDLE\_TIME | KERNEL | UNLIMITED |
| DEFAULT | CONNECT\_TIME | KERNEL | UNLIMITED |
| DEFAULT | PRIVATE\_SGA | KERNEL | UNLIMITED |
| DEFAULT | FAILED\_LOGIN\_ATTEMPTS | PASSWORD | 10 |
| DEFAULT | PASSWORD\_LIFE\_TIME | PASSWORD | 180 |
| DEFAULT | PASSWORD\_REUSE\_TIME | PASSWORD | UNLIMITED |
| DEFAULT | PASSWORD\_REUSE\_MAX | PASSWORD | UNLIMITED |
| DEFAULT | PASSWORD\_VERIFY\_FUNCTION | PASSWORD | NULL |
| DEFAULT | PASSWORD\_LOCK\_TIME | PASSWORD | 1 |
| DEFAULT | PASSWORD\_GRACE\_TIME | PASSWORD | 7 |

16 rows selected.

#### Viewing Memory Use for Each User Session

Để tìm thông tin về bộ nhớ sử dụng cho mỗi phiên làm việc người dùng truy vấn V$SESSION. (Để biết thêm thông tin chi tiết đọc Oracle Database Reference). Câu truy vấn dưới đây liệt kê toàn bộ các phiên làm việc hiện tại lúc đó, các Oracle Database User và việc sử dụng bộ nhớ của User Global Area (UGA) đối với mỗi phiên làm việc:

SELECT USERNAME, VALUE || 'bytes' "Current UGA memory" FROM V$SESSION sess, V$SESSTAT stat, V$STATNAME name

WHERE sess.SID = stat.SID

AND stat.STATISTIC# = name.STATISTIC#

AND name.NAME = 'session uga memory';

USERNAME Current UGA memory

------------------------------ ---------------------------------------------

18636bytes 17464bytes 19180bytes 18364bytes 39384bytes 35292bytes 17696bytes 15868bytes

USERSCOTT 42244bytes

SYS 98196bytes

SYSTEM 30648bytes

11 rows selected.

Để thấy được bộ nhớ UGA lớn nhất được cấp phát cho mỗi phiên kể từ khi instance bắt đầu, thay thế 'sessionugamemory' trong câu truy vấn trước bằng 'sessionugamemory max'.

# 3

## Configuring Authentication

Nội dung chương 3:

* + [About Authentication](#_bookmark182)
  + [Configuring Password Protection](#_bookmark184)
  + [Authenticating Database Administrators](#_bookmark316)
  + [Using the Database to Authenticate Users](#_bookmark339)
  + [Using the Operating System to Authenticate Users](#_bookmark349)
  + [Using the Network to Authenticate Users](#_bookmark356)
  + [Configuring Global User Authentication and Authorization](#_bookmark374)
  + [Configuring an External Service to Authenticate Users and Passwords](#_bookmark393)
  + [Using Multitier Authentication and Authorization](#_bookmark407)
  + [Preserving User Identity in Multitiered Environments](#_bookmark413)
  + [Finding Information About User Authentication](#_bookmark488)

### About Authentication (xác thực)

Xác thực nghĩa là xác minh danh tính của một người nào đó (một người dùng, thiết bị, hoặc thực thể khác) người mà muốn sử dụng dữ liệu, các tài nguyên, hoặc các ứng dụng.Việc xác thực danh tính thiết lập một mối quan hệ tính tưởng cho các tương tác trong tương lai. Xác thực cũng cho phép theo dõi bằng cách. Xác thực cũng cho phép giải trình trách nhiệm bằng cách làm cho nó có thể liên kết truy cập và hành động với các danh tính cụ thể. Sau khi xác thực, Các quy trình phân quyền có thể cho phép hoặc giới hạn các cấp độ của truy cập và hành động được phép với thực thể đó.

Bạn có thể xác thực cả người dùng cơ sở dữ liệu và người không dùng cơ sở dữ liệu Oracle. Để đơn giản, phương thức xác thực thường được sử dụng cho tất cả người dùng cơ sở dữ liệu, nhưng Oracle Database cho phép một cá thể có thể sử dụng bất kỳ các phương thức. Cơ sở dữ liệu Oracle yêu cầu các thủ tục xác thực đặc biệt cho các nhà quản trị cơ sở dữ liệu, vì họ thực hiện các hành động đặt biệt trong cơ sở dữ liệu. Oracle cũng mã hóa mật khẩu trong quá trình truyền để đảm bảo tính bảo mật của xác thực mạng.

Sau khi xác thực, quy trình ủy quyền có thể cho phép giới hạn mức truy cập và cấp phát các các quyền truy cập đối với một thực thể nào đó. Ủy quyền được mô tả trong [Chapter 4, "Configuring](#_bookmark493) [Privilege and Role Authorization"](#_bookmark493).

### Configuring Password Protection (cấu hình bảo vệ mật khẩu)

Nội dung:

■ [Những gì Oracle Database xây dựng trong quá trình bảo vệ Password?](#_bookmark186)

■ [Yêu](#_bookmark196) cầu tối thiểu cho mật khẩu

■ [Sử](#_bookmark200) dụng chính sách quản lý mật khẩu

■ [Đảm bảo chống lại các mối đe dọa bằng cách sử dụng thuật toán bâm SHA-1](#_bookmark289)

■ [Quản lý cá phương thức để xác nhận mật khẩu cho việc tạo Password](#_bookmark297)

Xem thêm ["Guidelines for Securing Passwords"](#_bookmark2167) trên trang 10-7 để hiểu về cách bảo mật mật khẩu. Nếu bạn muốn cấu hình Oracle XML DB để xác thực người dùng bằng cách mã hóa mật khẩu của họ nhưng bạn không cần mã hóa dữ liệu khác (ví dụ, email Intranet), xem Hướng dẫn của nhà phát triển Oracle XML DB để biết thêm thông tin.

#### Những gì Oracle Database xây dựng trong quá trình bảo vệ Password?

Cơ sở dữ liệu Oracle cung cấp một bộ bảo vệ mật khẩu được thiết kế để bảo vệ mật khẩu của người dùng của bạn. Các bảo vệ mật khẩu này như sau:

■ **Mã hóa mật khẩu.** Cơ sở dữ liệu Oracle tự động mã hóa một cách minh bạch các mật khẩu trong các kết nối mạng (máy khách đến máy chủ và máy chủ đến máy chủ), sử dụng Chuẩn mã hóa nâng cao (AES) trước khi gửi chúng qua mạng.

■**Kiểm tra độ phức tạp của mật khẩu.** Trong cài đặt mặc định, Oracle Database cung cấp chức năng xác minh mật khẩu verify\_function\_11g để đảm bảo rằng mật khẩu mới hoặc mật khẩu đang thay đổi đủ phức tạp để ngăn chặn những kẻ xâm nhập cố gắng đột nhập vào hệ thống bằng cách đoán mật khẩu. Bạn phải bật thủ công kiểm tra độ phức tạp của mật khẩu. Bạn có thể tùy chỉnh thêm độ phức tạp của mật khẩu của người dùng. Xem ["Enforcing Password Complexity Verification"](#_bookmark268) trên trang 3-11 để biết thêm thông tin.

■ **Ngăn chặn tài khoản đăng nhập nhiều lần.**Nếu người dùng cố gắng đăng nhập vào cơ sở dữ liệu Oracle nhiều lần bằng mật khẩu không đúng, Cơ sở dữ liệu Oracle sẽ trì hoãn mỗi lần đăng nhập. Sự bảo vệ này áp dụng cho các tài khoản đang cố gắng đăng nhập nhiều lần hoặc nhiều người kết nối cùng một tài khoản. Sau đó, nó tăng dần thời gian trước khi người dùng có thể thử một mật khẩu khác, tối đa khoảng 10 giây. Nếu người dùng nhập đúng mật khẩu, họ có thể đăng nhập thành công mà không bị chậm trễ trong lần đăng nhập sau.

Tính năng này làm giảm đáng kể số lượng mật khẩu mà kẻ xâm nhập có thể thử trong khoảng thời gian ngắn khi đang cố gắng đăng nhập. Trì hoản đăng nhập sẽ tăng lên sau mỗi lần đăng nhập thất bại, tăng tổng thời gian cần thiết để thực hiện mật khẩu- đoán cuộc tấn công, bởi vì các cuộc tấn công như vậy thường đòi hỏi một số lượng rất lớn các mật khẩu khác nhau.

■ **Trường hợp bắt buộc đối với mật khẩu.** Mật khẩu phân biệt chữ hoa chữ thường. Ví dụ, mật khẩu hPP5620qr không thành công nếu nó được nhập dưới dạng hpp5620QR hoặc hPp5620Qr. Trong các bản phát hành trước, mật khẩu không phân biệt chữ hoa chữ thường. Xem ["Enabling or Disabling Password Case Sensitivity"](#_bookmark274) trên trang 3-13 để biết thông tin về cách hoạt động của độ nhạy trường hợp và cách nó ảnh hưởng đến các tệp mật khẩu và các liên kết cơ sở dữ liệu.

■ **Mật khẩu được băm bằng hàm băm mật mã bảo mật (SHA) SHA-1.** Cơ sở dữ liệu Oracle sử dụng trình xác minh SHA-1 là để xác thực mật khẩu người dùng và thiết lập phiên làm việc của người dùng. Ngoài ra, nó thực thi các trường hợp bắt buộc và hạn chế của mật khẩu đến 160 bit. Lợi thế của việc sử dụng trình xác minh SHA-1 là nó thường được các khách hàng của Oracle Database sử dụng và nó cung cấp bảo mật tốt hơn nhiều mà không cần nâng cấp mạng. Nó cũng tuân thủ các quy định về các trường hợp bắt buộc đối với mật khẩu

bằng thuật toán băm mật khẩu phù hợp. Xem ["Ensuring Against Password](#_bookmark289) [Security Threats by Using the SHA-1 Hashing Algorithm" trên trang 3-15](#_bookmark289) để biết thêm nhiều thông tin.

#### Yêu cầu tối thiểu cho mật khẩu

Mật khẩu không được vượt quá 30 ký tự hoặc 30 byte. Tuy nhiên, để bảo mật tốt hơn, hãy làm theo các nguyên tắc bổ sung được mô tả trong "Nguyên tắc về bảo mật mật khẩu" trên trang 10-7.

Để tạo mật khẩu cho người dùng, bạn có thể sử dụng câu lệnh SQL CREATE USER hoặc ALTER USER. Các câu lệnh SQL chấp nhận mệnh đề IDENTIFIED BY cũng cho phép bạn tạo mật khẩu. Ví dụ 3–1 cho thấy một số câu lệnh SQL tạo mật khẩu với mệnh đề IDENTIFIED BY.

***Ví dụ 3-1 Các câu lệnh SQL tạo mật khẩu***

CREATE USER psmith IDENTIFIED BY *password*;

GRANT CREATE SESSION TO psmith IDENTIFIED BY *password*;

ALTER USER psmith IDENTIFIED BY *password*;

CREATE DATABASE LINK AUTHENTICATED BY psmith IDENTIFIED BY *password*;

**Xem thêm:**

■ ["Enforcing Password Complexity Verification"](#_bookmark268) trên trang 3-11 cho bạn các cách có thể đảm bảo rằng mật khẩu đủ phức tạp cho trang web của bạn

■ ["Guidelines for Securing Passwords"](#_bookmark2167) trên trang 10-7 để biết thêm các cách để bảo mật mật khẩu

■ ["Securing Passwords in Application Design"](#_bookmark1013) trên trang 5-3 để biết hướng dẫn bảo vệ mật khẩu, nhà phát triển ứng dụng phải tuân theo

■ *Oracle Database SQL Language Reference* thêm thông tin về CREATE USER, ALTER USER, GRANT, and CREATE DATABASE LINK SQL statements

#### Sử dụng chính sách quản lý mật khẩu

Nội dung:

* [Về](#_bookmark202) quản lý mật khẩu
* [Tìm](#_bookmark206) tài khoản người dùng có mật khẩu mạc định
* [Cấu](#_bookmark209) hình mật khẩu trong cấu hình mạc định
* [Tắt](#_bookmark221) và bật cài đặt, bảo mật mật khẩu mạc định
* [Tự](#_bookmark227) động khóa tài khoản người dùng sau khi đăng nhập thất bại
* [Kiễm](#_bookmark249) xoát thời hạn của tài khoản
* [Thay](#_bookmark256) đổi mật khẩu theo chu kì
* [Cài đặt PASSWORD\_LIFE\_TIME thành](#_bookmark265) giá trị thấp
* [Kiếm](#_bookmark239) xoát khả năng người dùng sử dụng lại mật khẩu trước
* [Thực](#_bookmark268) thi xác minh độ phức tạp của mật khẩu
* [Bật](#_bookmark274) hoặc tắt Sensitivity của mật khẩu

**Xem thêm:**

* + ["Managing Resources with Profiles" Trên trang 2-12](#_bookmark140)
  + *Oracle Database SQL Language Reference* for syntax and specific information about SQL statements discussed in this section

###### Về quản lý mật khẩu

Các hệ thống bảo mật cơ sở dữ liệu phụ thuộc vào mật khẩu yêu cầu mật khẩu luôn được giữ bí mật. Bởi vì mật khẩu dễ bị trộm cắp và sử dụng sai, Oracle Database sử dụng chính sách quản lý mật khẩu. Quản trị viên cơ sở dữ liệu và nhân viên an ninh kiểm soát chính sách này thông qua hồ sơ người dùng, cho phép kiểm soát cơ sở dữ liệu tốt hơn.

Sử dụng câu lệnh CREATE PROFILE để tạo hồ sơ người dùng. Hồ sơ được gán cho người dùng với câu lệnh CREATE USER hoặc ALTER USER. Chi tiết về việc tạo và thay đổi hồ sơ người dùng cơ sở dữ liệu không được thảo luận trong phần này. Phần này mô tả các tham số mật khẩu có thể được chỉ định bằng cách sử dụng câu lệnh CREATE PROFILE (hoặc ALTER PROFILE).

###### Tìm tài khoản người dùng có mật khẩu mặc định

Khi bạn tạo một cơ sở dữ liệu trong Cơ sở dữ liệu Oracle 11g Release 2 (11.2), hầu hết các tài khoản mặc định của nó sẽ bị khóa với mật khẩu đã hết hạn. Nếu bạn đã nâng cấp từ phiên bản trước đó của Cơ sở dữ liệu Oracle, bạn có thể có tài khoản người dùng có mật khẩu mặc định. Đây là những tài khoản mặc định được tạo khi bạn tạo cơ sở dữ liệu, chẳng hạn như tài khoản HR, OE và SCOTT.

Để bảo mật tốt hơn, hãy thay đổi mật khẩu cho các tài khoản này. Sử dụng mật khẩu mặc định thường làm cho cơ sở dữ liệu của bạn dễ bị tấn công bởi những kẻ xâm nhập. Để tìm cả tài khoản bị khóa và mở khóa sử dụng mật khẩu mặc định, hãy đăng nhập vào SQL \* Plus bằng đặc quyền SYSDBA và sau đó truy vấn chế độ xem từ điển dữ liệu DBA\_USERS\_WITH\_DEFPWD.

Ví dụ: để tìm cả tên tài khoản có mật khẩu mặc định và trạng thái tài khoản:

SELECT d.username, u.account\_status

FROM DBA\_USERS\_WITH\_DEFPWD d, DBA\_USERS u

WHERE d.username = u.username ORDER BY 2,1;

USERNAME ACCOUNT\_STATUS

--------- --------------------------- SCOTT EXPIRED & LOCKED

Sau đó thay đổi mật khẩu cho bất kỳ tài khoản nào mà danh sách được liệt kê trong DBA\_USERS\_WITH\_DEFPWD. Oracle khuyến cáo rằng bạn không nên đặt lại mật khẩu củ của tài khoản đã có trong các bản phát hành trước của Cơ sở dữ liệu Oracle.

ALTER USER SCOTT ACCOUNT UNLOCK IDENTIFIED BY *password*;

Thay thế mật khẩu bằng mật khẩu an toàn. "Yêu cầu tối thiểu đối với mật khẩu" trên trang 3-3 mô tả các yêu cầu tối thiểu đối với mật khẩu.

###### Cấu hình cài đặt mật khẩu trong cấu hình mặc định

Hồ sơ là tập hợp các tham số được giới hạn về tài nguyên trong cơ sở dữ liệu. Nếu bạn chỉ định cấu hình cho người dùng, thì người dùng đó không thể vượt quá các giới hạn này. Bạn có thể cấu hình các thiết lập cơ sở dữ liệu như các phiên cho mỗi người dùng, ghi nhật ký và truy tìm

các tính năng, v.v. Hồ sơ cũng có thể kiểm soát mật khẩu người dùng. Để tìm thông tin về cài đặt mật khẩu hiện tại trong tiểu sử, bạn có thể truy vấn chế độ xem từ điển dữ liệu DBA\_PROFILES.

[Bản 3–1](#_bookmark213)  liệt kê các cài đặt tham số mật khẩu cụ thể trong cấu hình mặc định.

***Bản 3–1 Cài đặt dành riêng cho mật khẩu trong Cấu hình mặc định***

|  |  |  |
| --- | --- | --- |
| **Tham số** | **Mạc định** | **Mô tả** |
| FAILED\_LOGIN\_ATTEMPTS | 10 | Đặt thời gian tối đa mà người dùng cố gắng đăng nhập và không thành công trước khi khóa tài khoản. |
|  |  | **Chú ý:** |
|  |  | * Khi bạn đặt tham số này, hãy xem xét những người dùng có thể đăng nhập bằng đặc quyền CONNECT THROUGH. |
|  |  | * Bạn có thể đặt giới hạn về số lần người dùng không được phép (có thể là kẻ xâm nhập) cố gắng đăng nhập vào các ứng dụng Oracle Call Interface (OCI) bằng cách sử dụng tham số khởi tạo SEC\_MAX\_FAILED\_LOGIN\_ ATTEMPTS. Xem ["Configuring the Maximum Number of Authentication Attempts"](#_bookmark1109) trên trang 5-19 để biết thêm thông tin về tham số này. |
|  |  | Xem thêm ["Automatically Locking a User](#_bookmark227) [Account After a Failed Login"](#_bookmark227) [đến trang 3-6](#_bookmark227) để xem thông tin chi tiết. |
| PASSWORD\_GRACE\_TIME | 7 | Đặt số ngày mà người dùng phải thay đổi mật khẩu của mình trước khi hết hạn. |
|  |  | Đến ["Controlling Password Aging and](#_bookmark249) [Expiration"](#_bookmark249) trên trang 3-8 để biết thêm thông tin. |
| PASSWORD\_LIFE\_TIME | 180 | Đặt số ngày người dùng có thể sử dụng mật khẩu hiện tại của họ. |
|  |  | Xem ["Controlling Password Aging and](#_bookmark249) [Expiration"](#_bookmark249)  trên trang 3-8 để biết thêm thông tin. |
| PASSWORD\_LOCK\_TIME | 1 | Đặt số ngày mà tài khoản sẽ bị khóa sau số lần đăng nhập thất bại liên tiếp được chỉ định. Sau khi hết thời gian, tài khoản sẽ được mở khóa. Lưu thông số tiểu sử của người dùng này để giúp ngăn chặn các cuộc tấn công trên mật khẩu người dùng nhưng không tăng gánh nặng bảo trì cho quản trị viên. |
|  |  | Xem ["Automatically Locking a User Account](#_bookmark227) [After a Failed Login"](#_bookmark227)  trên trang 3-6 để biết thêm thông tin. |
| PASSWORD\_REUSE\_MAX | UNLIMITED | Đặt số lượng thay đổi mật khẩu tối đa được yêu cầu trước khi mật khẩu hiện tại có thể được sử dụng lại. |
|  |  | Xem ["Controlling User Ability to Reuse](#_bookmark239) [Previous Passwords"](#_bookmark239)  trên trang 3-7 để biết thêm thông tin. |

***Bản 3–1 (Cont.) Cài đặt dành riêng cho mật khẩu trong Cấu hình mặc định***

|  |  |  |
| --- | --- | --- |
| **Thông số** | **Mạc định** | **Mô tả** |
| PASSWORD\_REUSE\_TIME | UNLIMITED | Đặt số ngày trước khi sử dụng lại mật khẩu.  Xem ["Controlling User Ability to Reuse](#_bookmark239) [Previous Passwords"](#_bookmark239) trên trang 3-7 để biết thêm thông tin. |

Để bảo mật tốt hơn, hãy sử dụng các cài đặt mặc định được mô tả trong Bảng 3–1, dựa trên nhu cầu của bạn. Bạn có thể tạo hoặc sửa đổi các tham số riêng cho mật khẩu bằng cách sử dụng câu lệnh CREATE PROFILE hoặc ALTER PROFILE. Ví dụ:

ALTER PROFILE prof LIMIT FAILED\_LOGIN\_ATTEMPTS 9

PASSWORD\_LOCK\_TIME 10;

Xem *Oracle Database SQL Language Reference* xem thêm thông tin về CREATE PROFILE, ALTER PROFILE, và các tham số liên quan đến mật khẩu được mô tả trong phần này.

###### Tắt và bật cài đặt bảo mật mật khẩu mặc định

Nếu ứng dụng của bạn sử dụng cài đặt bảo mật mật khẩu mặc định từ Cơ sở dữ liệu Oracle 10g Release 2 (10.2), thì bạn có thể để nguyên các cài đặt này cho đến khi bạn sửa đổi các ứng dụng để sử dụng cài đặt bảo mật mật khẩu Release 11g. Để làm như vậy, hãy chạy tập lệnh undopwd.sql.

Sau khi bạn đã sửa đổi các ứng dụng của mình để phù hợp với cài đặt bảo mật mật khẩu Release 11g, bạn có thể cập nhật cơ sở dữ liệu theo cách thủ công để sử dụng cấu hình bảo mật mật khẩu phù hợp với nhu cầu của bạn, hoặc bạn có thể chạy tập lệnh secconf.sql để áp dụng mật khẩu mặc định của Release 11g cài đặt. Bạn có thể tùy chỉnh tập lệnh này để có các cài đặt bảo mật khác nhau nếu bạn muốn, nhưng hãy nhớ rằng các cài đặt được liệt kê trong tập lệnh ban đầu là các cài đặt được Oracle đề xuất.

Nếu bạn đã tạo cơ sở dữ liệu theo cách thủ công, thì bạn nên chạy lệnh secconf.sql để áp dụng các thiết lập mật khẩu mặc định Release 11g cho cơ sở dữ liệu. Cơ sở dữ liệu đã được tạo với cấu hình (DBCA) sẽ có các cài đặt này, nhưng cơ sở dữ liệu được tạo thủ công thì không.

Các lệnh undopwd.sql và secconf.sql nằm trong thư mục $ ORACLE\_HOME / rdbms / admin. Lệnh undopwd.sql chỉ ảnh hưởng đến cài đặt mật khẩu và Lệnh secconf.sql ảnh hưởng đến cả cài đặt mật khẩu và kiểm tra. Chúng không ảnh hưởng đến các cài đặt bảo mật khác.

###### Tự động khóa tài khoản người dùng sau khi đăng nhập thất bại

Cơ sở dữ liệu Oracle có thể khóa tài khoản của người dùng sau một số lần đăng nhập thất bại liên tiếp được chỉ định. Bạn có thể đặt thông số tiểu sử của người dùng PASSWORD\_LOCK\_TIME để cấu hình tự động mở khóa sau một khoảng thời gian cụ thể hoặc yêu cầu mở khóa sự can thiệp của quản trị viên cơ sở dữ liệu. Người quản trị cơ sở dữ liệu cũng có thể khóa tài khoản theo cách thủ công,và phải được mở khóa một cách rõ ràng và minh bạch bởi người quản trị.

Bạn có thể chỉ định số lần đăng nhập không được phép bằng cách sử dụng câu lệnh CREATE PROFILE. Bạn cũng có thể chỉ định số lượng thời gian bị khóa của tài khoản.

[Ví dụ 3–2](#_bookmark234)  đặt số lần đăng nhập thất bại tối đa cho johndoe của người dùng thành 10 (mặc định) và khoảng thời gian tài khoản bị khóa trong 30 ngày. Tài khoản sẽ tự động mở khóa sau 30 ngày.

***Ví dụ 3–2 Khóa một tài khoản với câu lệnh CREATE PROFILE***

CREATE PROFILE prof LIMIT FAILED\_LOGIN\_ATTEMPTS 10

PASSWORD\_LOCK\_TIME 30;

ALTER USER johndoe PROFILE prof;

Mỗi lần người dùng đăng nhập không thành công, Cơ sở dữ liệu Oracle sẽ tăng độ trễ theo cấp số nhân với mỗi lần đăng nhập thất bại.

Nếu bạn không chỉ định khoảng thời gian để mở khóa tài khoản, thì PASSWORD\_LOCK\_ TIME giả định giá trị được chỉ định trong cấu hình mặc định. (Giá trị được đề nghị là 1 ngày.) Nếu bạn chỉ định PASSWORD\_LOCK\_TIME là UNLIMITED, thì bạn phải mở khóa tài khoản một cách rõ ràng bằng cách sử dụng câu lệnh ALTER USER. Ví dụ: giả sử rằng PASSWORD\_LOCK\_TIME UNLIMITED được chỉ định cho johndoe, thì bạn sử dụng câu lệnh sau để mở khóa tài khoản johndoe:

ALTER USER johndoe ACCOUNT UNLOCK;

Sau khi người dùng đăng nhập thành công vào tài khoản, Cơ sở dữ liệu Oracle đặt lại số lần đăng nhập không thành công cho người dùng, nếu nó không khác thì thay đổi lại bằng không.

Nhân viên an ninh cũng có thể khóa tài khoản người dùng một cách rõ ràng. Khi điều này xảy ra, tài khoản không thể tự động mở khóa và chỉ có nhân viên bảo mật mới mở khóa tài khoản. Các câu lệnh CREATE USER hoặc ALTER USER được khóa hoặc mở khóa tài khoản người dùng một cách rõ ràng. Ví dụ, câu lệnh sau khóa tài khoản người dùng, susan:

ALTER USER susan ACCOUNT LOCK;

###### Kiểm soát khả năng sử dụng lại mật khẩu trước

Bạn có thể đảm bảo rằng người dùng không sử dụng lại mật khẩu trước đó của họ trong một khoảng thời gian cụ thể hoặc cho một số thay đổi mật khẩu được chỉ định. Để làm như vậy, hãy định cấu hình các quy tắc để sử dụng lại mật khẩu với các câu lệnh CREATE PROFILE hoặc ALTER PROFILE. Để biết cú pháp đầy đủ của các câu lệnh này, xem Tham chiếu ngôn ngữ SQL cơ sở dữ liệu Oracle.

[Bản 3–2](#_bookmark244)  liệt kê các tham số CREATE PROFILE và ALTER PROFILE để kiểm soát khả năng người dùng sử dụng lại mật khẩu trước đó.

***Bản 3–2 Tham số Kiểm soát việc sử dụng lại mật khẩu trước***

|  |  |
| --- | --- |
| **Tham số** | **Mô tả** |
| PASSWORD\_REUSE\_TIME | Yêu cầu một trong những điều sau đây:   * Một số xác định số ngày (hoặc một phần của một ngày) giữa việc sử dụng mật khẩu trước đó và lần sử dụng tiếp theo của nó * Từ UNLIMITED |
| PASSWORD\_REUSE\_MAX | Yêu cầu một trong những điều sau đây:   * Một số nguyên để xác định số lượng thay đổi mật khẩu cần thiết trước khi một mật khẩu có thể được tái sử dụng * Từ UNLIMITED |

Nếu bạn không chỉ định tham số, thì người dùng có thể sử dụng lại mật khẩu bất kỳ lúc nào, đây không phải là bảo mật tốt.

Nếu không có tham số nào là UNLIMITED, thì việc sử dụng lại mật khẩu sẽ được cho phép, nhưng chỉ sau khi đáp ứng cả hai điều kiện. Người dùng phải thay đổi mật khẩu với ngày được chỉ định và số ngày được chỉ định phải sau lần sử dụng mật khẩu trước đó.

Ví dụ, giả sử cấu hình của người dùng A có PASSWORD\_REUSE\_MAX được đặt thành 10 và PASSWORD\_REUSE\_TIME được đặt thành 30. Người dùng A không thể sử dụng lại mật khẩu cho đến khi người đó đặt lại mật khẩu 10 lần và cho đến 30 ngày kể từ khi mật khẩu được sử dụng lần cuối

Nếu tham số được chỉ định là UNLIMITED thì người dùng không bao giờ có thể sử dụng lại mật khẩu.

Nếu bạn đặt cả hai tham số thành UNLIMITED, thì Cơ sở dữ liệu Oracle sẽ bỏ qua cả hai và người dùng có thể sử dụng lại bất kỳ mật khẩu nào bất kỳ lúc nào.

**Chú ý:** Nếu bạn chỉ định DEFAULT cho một trong hai tham số, thì cơ sở dữ liệu Oracle sử dụng giá trị được định nghĩa trong cấu hình DEFAULT, nó đặt tất cả các tham số thành UNLIMITED. Cơ sở dữ liệu Oracle sử dụng UNLIMITED cho bất kỳ tham số nào được chỉ định là DEFAULT, trừ khi bạn thay đổi thiết lập cho tham số đó trong cấu hình DEFAULT.

###### Kiểm soát sự Aging và thời hạn của mật khẩu

Bạn có thể chỉ định thời hạn cho mật khẩu. Điều này có nghĩa là lần sau khi người dùng đăng nhập bằng mật khẩu hiện tại, đúng, họ được nhắc thay đổi mật khẩu. Theo mặc định, không có kiểm tra lịch sử thay đổi mật khẩu, do đó người dùng vẫn có thể sử dụng lại bất kỳ mật khẩu nào trước đó hoặc yếu. Bạn có thể kiểm soát các yếu tố này bằng cách đặt các tham số PASSWORD\_REUSE\_TIME, PASSWORD\_REUSE\_MAX và PASSWORD\_VERIFY\_FUNCTION. (Xem ["Controlling User Ability to Reuse](#_bookmark239) [Previous Passwords" đến trang 3-7](#_bookmark239) và ["Enforcing Password Complexity Verification"](#_bookmark268) [đến trang 3-11](#_bookmark268) để biết theme thông tin.)

Ngoài ra, bạn có thể đặt lại thời gian gia hạn mật khẩu, trong đó mỗi lần đăng nhập vào tài khoản cơ sở dữ liệu sẽ nhận được một thông báo cảnh báo để thay đổi mật khẩu. Nếu người dùng không thay đổi nó vào cuối khoảng thời gian đó, thì Cơ sở dữ liệu Oracle sẽ hết hạn tài khoản.

Là quản trị viên cơ sở dữ liệu, bạn có thể đặt trạng thái mật khẩu thành công theo cách thủ công, trạng thái này sẽ đặt trạng thái tài khoản thành HẾT HẠN. Người dùng sau đó phải làm theo hướng dẫn để thay đổi thời hạn mật khẩu trước khi tiến hành đăng nhập.

Ví dụ: trong SQL \* Plus, giả sử người dùng SCOTT cố đăng nhập bằng thông tin xác thực chính xác, nhưng mật khẩu của anh ấy đã hết hạn. Người dùng SCOTT sau đó sẽ thấy ORA-28001: Mật khẩu đã hết hạn và được nhắc thay đổi mật khẩu của mình, như sau:

Changing password for scott New password: *new\_password*

Retype new password: *new\_password*

Password changed.

Sử dụng câu lệnh CREATE PROFILE hoặc ALTER PROFILE để chỉ định tuổi thọ cho mật khẩu. Để hiểu vòng đời của mật khẩu, Xem ["Password Change Life](#_bookmark256) [Cycle" đến trang 3-9](#_bookmark256).

[Ví dụ 3–3](#_bookmark255)  trình bày cách tạo và gán một hồ sơ cho người dùng johndoe và tham số PASSWORD\_LIFE\_TIME chỉ định rằng johndoe có thể sử dụng cùng một mật khẩu trong 180 ngày trước khi nó hết hạn.

***Ví dụ 3–3 Đặt lại mật khẩu hết hạn với CREATE PROFILE***

CREATE PROFILE prof LIMIT FAILED\_LOGIN\_ATTEMPTS 4

PASSWORD\_LOCK\_TIME 30

PASSWORD\_LIFE\_TIME 180;

ALTER USER johndoe PROFILE prof;

Bạn có thể kiểm tra trạng thái của bất kỳ tài khoản nào, cho dù tài khoản đó đang mở, đang ẩn hay đã hết hạn, bằng cách chạy truy vấn sau:

SELECT ACCOUNT\_STATUS FROM DBA\_USERS WHERE USERNAME = '*username*';

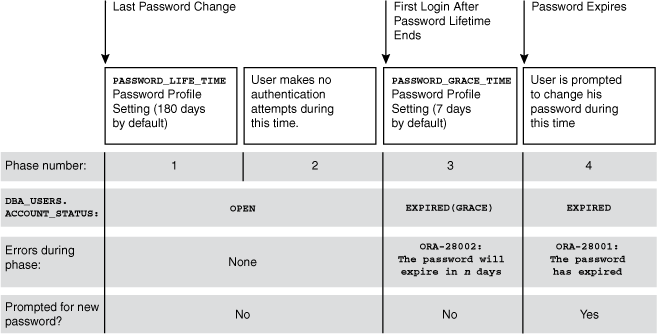
###### Vòng đời thay đổi mật khẩu

[Hình 3–1](#_bookmark258) cho thấy vòng đời của tuổi thọ mật khẩu và thời gian gia hạn.

* **Giai đoạn 1:** Sau khi tài khoản người dùng được tạo, hoặc mật khẩu của tài khoản hiện đang thay đổi, thời gian tồn tại của mật khẩu sẽ bắt đầu.
* **Giai đoạn 2:** Giai đoạn này đại diện cho khoảng thời gian sau khi thời hạn sử dụng mật khẩu kết thúc nhưng trước khi người dùng đăng nhập lại bằng mật khẩu chính xác. Các thông tin xác thực là cần thiết cho Oracle Database để cập nhật trạng thái tài khoản. Nếu không, trạng thái tài khoản sẽ không thay đổi. Cơ sở dữ liệu Oracle không có bất kỳ quá trình nào để cập nhật trạng thái tài khoản. Tất cả thay đổi đối với trạng thái tài khoản được điều khiển bởi quy trình máy chủ Cơ sở dữ liệu Oracle.
* **Giai đoạn 3:** Khi người dùng đăng nhập, thời gian gia hạn bắt đầu. Sau đó, Oracle Database cập nhật cột DBA\_USERS.EXPIRY\_DATE thành giá trị mới bằng cách sử dụng thời gian hiện tại cộng với giá trị của cài đặt PASSWORD\_GRACE\_TIME từ cấu hình mật khẩu của tài khoản. Tại thời điểm này, người dùng nhận được thông báo cảnh báo ORA-28002 về mật khẩu hết hạn trong tương lai gần (ví dụ: ORA-28002 Mật khẩu sẽ hết hạn trong vòng 7 ngày nếu PASSWORD\_GRACE\_TIME được đặt thành 7 ngày), nhưng người dùng vẫn có thể đăng nhập mà không thay đổi mật khẩu. Cột DBA\_ USERS.EXPIRY\_DATE hiển thị thời gian trong tương lai khi người dùng sẽ được nhắc thay đổi mật khẩu của họ.
* **Giai đoạn 4:** Sau giai đoạn gia hạn (giai đoạn 3) kết thúc, ORA-28001: Mật khẩu đã hết hạn xuất hiện, và người dùng được nhắc thay đổi mật khẩu sau khi nhập mật khẩu hiện tại đúng. Nếu người dùng có cấu hình Oracle Active Data Guard, nơi có cơ sở dữ liệu chính và cơ sở dữ liệu dự phòng, và xác thực được thực hiện trên cơ sở dữ liệu dự phòng (là cơ sở dữ liệu chỉ đọc). Người dùng nên đăng nhập vào cơ sở dữ liệu chính và thay đổi mật khẩu ở đó.

Trong bất kỳ giai đoạn nào trong bốn giai đoạn này, bạn có thể truy vấn chế độ xem dữ liệu DBA\_USERS để tìm trạng thái tài khoản của người dùng trong cột DBA\_USERS.ACCOUNT\_STATUS.

***Hình 3–1 Trình tự thời gian tồn tại của mật khẩu và thời gian gia hạn***



Trong ví dụ sau, Profile được gán cho johndoe bao gồm đặc tả của thời gian gia hạn: PASSWORD\_GRACE\_TIME = 3 (giá trị được đề nghị). Lần đầu tiên johndoe cố đăng nhập vào cơ sở dữ liệu sau 90 ngày (có thể là bất kỳ ngày nào sau ngày thứ 90, tức là ngày thứ 91, ngày thứ 100 hoặc ngày khác), anh ta nhận được thông báo cảnh báo rằng mật khẩu của anh ta sẽ hết hạn trong 3 ngày. Nếu 3 ngày trôi qua, và nếu anh ta không thay đổi thời hạn mật khẩu của mình thì mật khẩu sẽ hết hạn. Sau đó, anh ta nhận được lời nhắc thay đổi mật khẩu của mình trên mọi nỗ lực đăng nhập.

CREATE PROFILE prof LIMIT FAILED\_LOGIN\_ATTEMPTS 4

PASSWORD\_LIFE\_TIME 90

PASSWORD\_GRACE\_TIME 3;

ALTER USER johndoe PROFILE prof;

Người quản trị cơ sở dữ liệu hoặc người dùng có đặc quyền hệ thống ALTER USER rõ ràng có thể làm hết hạn một tào khoản bằng cách sử dụng câu lệnh CREATE USER và ALTER USER. Câu lệnh sau tạo một người dùng có mật khẩu đã hết hạn. Cài đặt này buộc người dùng phải thay đổi mật khẩu trước khi người dùng có thể đăng nhập vào cơ sở dữ liệu.

CREATE USER jbrown IDENTIFIED BY *password*

...

PASSWORD EXPIRE;

Không có mệnh đề "password unexpire" cho câu lệnh CREATE USER, nhưng một tài khoản có thể "chưa hết hạn" bằng cách thay đổi thời hạn mật khẩu trên tài khoản.

###### Đặt thông số profile PASSWORD\_LIFE\_TIME thành giá trị thấp

Hãy cẩn thận nếu bạn định đặt tham số PASSWORD\_LIFE\_TIME của CREATE PROFILE hoặc ALTER PROFILE thành một giá trị thấp (ví dụ, 1 ngày). Giới hạn PASSWORD\_LIFE\_TIME của profile được đo từ lần cuối cùng mật khẩu của tài khoản bị thay đổi hoặc thời gian tạo tài khoản nếu mật khẩu chưa bao giờ bị thay đổi. Những ngày này được ghi lại trong các cột PTIME (thời gian thay đổi mật khẩu) và CTIME (thời gian tạo tài khoản) của bảng hệ thống SYS.USER $. Giới hạn PASSWORD\_LIFE\_TIME không được đo bắt đầu từ dấu thời gian của thay đổi cuối cùng đến tham số hồ sơ PASSWORD\_LIFE\_TIME, như ban đầu có thể được nghĩ. Do đó, bất kỳ tài khoản nào bị ảnh hưởng bởi profile có thời gian thay đổi mật khẩu cuối cùng nhiều hơn PASSWORD\_LIFE\_TIME ngày trước hết hạn và nhập thời gian gia hạn vào kết nối tiếp theo của họ, cấp ORA-28002: Mật khẩu sẽ hết hạn trong vòng 3 ngày cảnh báo.

Là quản trị viên cơ sở dữ liệu, bạn có thể tìm thời gian thay đổi mật khẩu cuối cùng của tài khoản như sau:

ALTER SESSION SET NLS\_DATE\_FORMAT='DD-MON-YYYY HH24:MI:SS';

SELECT PTIME FROM SYS.USER$ WHERE NAME = '*user\_name*'; *-- Password change time*

Để tìm khi tài khoản được tạo và ngày hết hạn mật khẩu, hãy đưa ra truy vấn sau:

SELECT CREATED, EXPIRY\_DATE FROM DBA\_USERS WHERE USERNAME = '*user\_name*';

Nếu profile này hiện đang đăng nhập khi bạn đặt tham số PASSWORD\_LIFE\_TIME và vẫn đăng nhập, thì Cơ sở dữ liệu Oracle không thay đổi trạng thái tài khoản của người dùng từ OPEN thành EXPIRED (GRACE) khi ngày hết hạn được liệt kê hiện tại vượt qua. Thời gian bắt đầu chỉ khi người dùng đăng nhập vào cơ sở dữ liệu.

Khi thực hiện thay đổi đối với mật khẩu, quản trị viên cơ sở dữ liệu phải biết rằng nếu một số người dùng trong profile này hiện đang đăng nhập vào cơ sở dữ liệu Oracle trong khi hồ sơ mật khẩu của họ đang được quản trị viên cập nhật thì người dùng đó có khả năng vẫn đăng nhập vào hệ thống ngay cả ngoài ngày hết hạn của mật khẩu của họ. Bạn có thể tìm người dùng hiện đã đăng nhập bằng cách truy vấn cột USERNAME của chế độ xem V $ SESSION.

Điều này là do ngày hết hạn mật khẩu của người dùng dựa trên thời gian của thay đổi mật khẩu cuối cùng trên tài khoản của họ cộng với giá trị của thông số cấu hình mật khẩu PASSWORD\_LIFE\_TIME do quản trị viên đặt. Nó không dựa trên thời gian thay đổi cuối cùng đối với chính hồ sơ đó.

Lưu ý những điều dưới đây:

* Nếu người dùng không đăng nhập khi bạn đặt PASSWORD\_LIFE\_TIME thành giá trị thấp, thì trạng thái tài khoản của người dùng sẽ không thay đổi cho đến khi người dùng đăng nhập.
* Bạn có thể đặt tham số PASSWORD\_LIFE\_TIME thành UNLIMITED, nhưng điều này chỉ ảnh hưởng đến các tài khoản chưa nhập thời gian gia hạn. Sau khi thời gian gia hạn hết hạn, người dùng phải thay đổi mật khẩu.

###### Thực thi xác minh phức tạp mật khẩu

Xác minh phức tạp, kiểm tra xem mỗi mật khẩu có đủ phức tạp để cung cấp sự bảo vệ hợp lý chống lại những kẻ xâm nhập cố gắng đột nhập vào hệ thống bằng cách đoán mật khẩu hay không. Điều này buộc người dùng tạo mật khẩu mạnh, an toàn cho tài khoản người dùng cơ sở dữ liệu. Bạn cần đảm bảo rằng mật khẩu cho người dùng của bạn đủ phức tạp để cung cấp sự bảo vệ hợp lý chống lại những kẻ xâm nhập cố gắng đột nhập vào hệ thống bằng cách đoán mật khẩu.

Cơ sở dữ liệu Oracle kiểm tra tính phức tạp của mật khẩu như thế nào

Cơ sở dữ liệu Oracle cung cấp một hàm xác minh mật khẩu mẫu có tên là verify\_ function\_11G trong lệnh PL / SQL utlpwdmg.sql (nằm trong $ ORACLE\_ HOME / rdbms / admin), khi được kích hoạt, kiểm tra xem người dùng có đang tạo hoặc sửa đổi mật khẩu chính xác hay không. Tập lệnh utlpwdmg.sql cung cấp hai hàm xác minh mật khẩu: một cho các phiên bản trước của Cơ sở dữ liệu Oracle và một phiên bản cập nhật cho Cơ sở dữ liệu Oracle Release 11g.

Để bảo mật mật khẩu tốt hơn, Oracle khuyên bạn nên liên kết hàm verify\_ function\_11G với cấu hình mặc định. "Tùy chỉnh Xác minh Mật khẩu phức tạp" trên trang 3-12 cung cấp một ví dụ về cách thực hiện điều này.

Tập lệnh utlpwdmg.sql kiểm tra các yêu cầu sau khi người dùng tạo hoặc sửa đổi mật khẩu:

■ Mật khẩu không chứa ít hơn 8 ký tự và không vượt quá 30 ký tự.

■ Mật khẩu không giống như tên người dùng, cũng không phải là tên người dùng đánh vần ngược hoặc với số 1–100 được nối thêm.

■ Mật khẩu không giống với tên máy chủ hoặc tên máy chủ có số 1–100 được nối thêm.

■ Mật khẩu không quá đơn giản, ví dụ: welcome1, database1, account1, user1234, password1, oracle, oracle123, computer1, abcdefg1 hoặc change\_on\_ install.

■ Mật khẩu không phải là oracle or oracle với các số 1–100 được nối thêm.

■ Mật khẩu bao gồm ít nhất 1 chữ số và 1 ký tự chữ cái.

■ Mật khẩu khác với mật khẩu trước đó bằng ít nhất 3 chữ cái.

Ai có thể sử dụng chức năng mật khẩu phức tạp?

Trước khi bạn có thể sử dụng các hàm xác minh mật khẩu phức tạp trong câu lệnh CREATE PROFILE hoặc ALTER PROFILE, bạn phải được cấp đặc quyền EXECUTE trên chúng. Các hàm xác minh mật khẩu được đặt trong lược đồ SYS.

Tùy chỉnh xác minh phức tạp mật khẩu

Bạn có thể tạo hàm xác minh phức tạp của mật khẩu bằng cách sao lưu và tùy chỉnh hàm verify\_function\_11G trong tập lệnh utlpwdmg.sql. Trên thực tế, Oracle khuyên bạn nên làm như vậy để bảo mật hơn nữa mật khẩu trang web của bạn. Xem thêm Guideline [1](#_bookmark2169) in ["Guidelines for Securing Passwords"](#_bookmark2167) trên trang 10-7 để được tư vấn chung về cách tạo mật khẩu. Tuy nhiên, lưu ý rằng việc kiểm tra độ phức tạp của mật khẩu không được thực thi cho người dùng SYS.

Theo mặc định, xác minh phức tạp mật khẩu không được bật. Để bật xác minh phức tạp mật khẩu:

1. Đăng nhập vào SQL \* Plus với quyền quản trị. For ví dụ:

CONNECT SYS AS SYSDBA

Enter password: *password*

1. Chạy utlpwdmg.sql (hoặc phiên bản đã sửa đổi của tập lệnh này) để tạo các hàm phức tạp mật khẩu trong lược đồ SYS.

@$ORACLE\_HOME/RDBMS/ADMIN/utlpwdmg.sql

1. Cấp cho bất kỳ người dùng nào phải sử dụng hàm này đặc quyền EXECUTE trên đó.

Ví dụ:

GRANT pmsith EXECUTE ON verify\_function\_11G;

1. Trong cấu hình mặc định hoặc profile của người dùng, hãy đặt cài đặt PASSWORD\_VERIFY\_FUNCTION thành mật khẩu phức tạp trong tập lệnh utlpwdmg.sql hoặc chức năng tùy chỉnh của bạn. Sử dụng một trong các phương pháp sau:
   * Đăng nhập vào SQL \* Plus với quyền quản trị viên và sử dụng câu lệnh CREATE PROFILE hoặc ALTER PROFILE để kích hoạt chức năng. Ví dụ: để cập nhật hồ sơ mặc định để sử dụng hàm verify\_function\_11G:

ALTER PROFILE default LIMIT

PASSWORD\_VERIFY\_FUNCTION verify\_function\_11G;

* + - Trong Oracle Enterprise Manager, hãy vào trang Edit Profiles và sau đó là Complexity, chọn tên của hàm phức tạp mật khẩu từ danh sách hàm Complexity.

Sau khi bạn đã bật xác minh phức tạp mật khẩu, tính năng này sẽ có hiệu lực ngay lập tức.

**Chú ý:** Câu lệnh ALTER USER có mệnh đề REPLACE. Với mệnh đề này, người dùng có thể thay đổi mật khẩu chưa hết hạn của mình bằng cách cung cấp mật khẩu trước đó để tự xác thực.

Nếu mật khẩu đã hết hạn, thì người dùng không thể đăng nhập vào SQL để đưa ra lệnh ALTER USER. Thay vào đó, hàm OCIPasswordChange () phải được sử dụng, cũng yêu cầu mật khẩu trước đó.

Người quản trị cơ sở dữ liệu với đặc quyền ALTER ANY USER có thể thay đổi bất kỳ mật khẩu người dùng nào (buộc mật khẩu mới) mà không cung cấp mật khẩu cũ.

###### Bật hoặc tắt Sensitivity

Nội dung:

* [Giới](#_bookmark276) thiệu về bật và tắt sensitivity
* [Thủ](#_bookmark281) tục kích hoạt
* [Tìm](#_bookmark283) phiên bản mật khẩu và tài khoản của người dùng
* [Sensitivity](#_bookmark285) ảnh hưởng đến file mật khẩu như thế nào
* [Làm thế nào sensitivity ảnh hưởng đến tài khoản tạo cho kết nối cơ sở dữ liệu liên kết](#_bookmark288)

Giới thiệu về kích hoạt hoặc vô hiệu hóa sensitivity

Khi bạn tạo hoặc sửa đổi tài khoản người dùng, theo mặc định, mật khẩu phân biệt chữ hoa chữ thường. Để kiểm soát việc sử dụng Sensitivity của chữ trong mật khẩu, hãy đặt tham số khởi tạo SEC\_CASE\_SENSITIVE\_LOGON. Chỉ những người dùng có đặc quyền ALTER SYSTEM mới có thể đặt tham số SEC\_CASE\_SENSITIVE\_LOGON. Đặt nó thành TRUE để cho phép độ nhạy trường hợp hoặc FALSE để vô hiệu hóa độ nhạy trường hợp.

Để bảo mật tốt hơn, Oracle khuyến cáo bạn nên bật Sensitivity trong mật khẩu. Tuy nhiên, nếu bạn có vấn đề tương thích với các ứng dụng của mình, bạn có thể sử dụng tham số này để vô hiệu hóa Sensitivity của mật khẩu. Ví dụ về các vấn đề tương thích ứng dụng là mật khẩu cho các ứng dụng của bạn được mã hóa cứng để phân biệt chữ hoa chữ thường, hoặc các mô-đun ứng dụng khác nhau không nhất quán về độ nhạy trường hợp khi gửi thông tin đăng nhập để bắt đầu phiên cơ sở dữ liệu.

Không đặt tham số SEC\_CASE\_SENSITIVE\_LOGON thành FALSE khi chế độ exclusive được bật (tham số SQLNET.ALLOWED\_LOGON\_VERSION được đặt thành 11), vì trình xác minh an toàn hơn được sử dụng trong chế độ exclusive chỉ hỗ trợ kiểm tra mật khẩu phân biệt chữ hoa chữ thường. Vì lý do tương thích, Oracle Database không ngăn cản việc sử dụng cài đặt FALSE cho SEC\_CASE\_SENSITIVE\_LOGON khi tham số SQLNET.ALLOWED\_LOGON\_ VERSION được đặt thành 11. Điều này có thể khiến tài khoản không thể truy cập được nếu các cài đặt này có hiệu lực khi người dùng thay đổi mật khẩu hoặc bạn tạo tài khoản người dùng mới.

Quy trình cho phép kích hoạt Sensitivity mật khẩu

Để bật Sensitivity của chữ trong mật khẩu:

1. Nếu bạn đang sử dụng tệp mật khẩu, hãy đảm bảo rằng tệp được tạo bằng tham số IGNORECASE được đặt thành N.

Tham số IGNORECASE ghi đè tham số SEC\_CASE\_SENSITIVE\_LOGON. Theo mặc định, IGNORECASE được đặt thành Y, có nghĩa là mật khẩu được coi là không phân biệt chữ hoa chữ thường. Để biết thêm thông tin về tệp mật khẩu, hãy xem Hướng dẫn của quản trị viên cơ sở dữ liệu Oracle.

1. Nhập câu lệnh ALTER SYSTEM sau đây:

ALTER SYSTEM SET SEC\_CASE\_SENSITIVE\_LOGON = TRUE

Tìm phiên bản mật khẩu của tài khoản người dùng

Trong các bản phát hành trước của Cơ sở dữ liệu Oracle, mật khẩu không phân biệt chữ hoa chữ thường. Nếu bạn nhập tài khoản người dùng từ bản phát hành trước, ví dụ, Bản phát hành 10g, vào bản phát hành cơ sở dữ liệu hiện tại, mật khẩu vẫn không phân biệt chữ hoa chữ thường cho đến khi người dùng thay đổi mật khẩu của họ. Nếu tài khoản đã được cấp đặc quyền SYSDBA hoặc SYSOPER, nó sẽ được nhập vào tệp mật khẩu. (Xem "Làm thế nào trường hợp nhạy cảm ảnh hưởng đến tập tin mật khẩu" trên trang 3-14 để biết thêm thông tin.) Khi một mật khẩu từ một tài khoản người dùng từ bản phát hành trước đó được thay đổi, nó sau đó trở thành Sensitivity.

Bạn có thể tìm người dùng có mật khẩu phân biệt chữ hoa chữ thường hoặc chữ hoa chữ thường bằng cách truy vấn chế độ xem DBA\_USERS. Cột PASSWORD\_VERSIONS trong khung nhìn này cho biết bản phát hành đã tạo mật khẩu. Ví dụ:

SELECT USERNAME,PASSWORD\_VERSIONS FROM DBA\_USERS;

USERNAME PASSWORD\_VERSIONS

------------------------------ ----------------- JONES 10G 11G

ADAMS 10G 11G

CLARK 10G 11G

PRESTON 11G

BLAKE 10G

Mật khẩu cho các tài khoản jones, adams, và clark ban đầu được tạo ra trong Release 10g và sau đó thiết lập lại trong Release 11g. Mật khẩu của họ, giả định độ nhạy trường hợp đã được kích hoạt, bây giờ là trường hợp nhạy cảm, như là mật khẩu cho preston. Tuy nhiên, tài khoản cho blake vẫn đang sử dụng tiêu chuẩn Release 10g, vì vậy nó không phân biệt chữ hoa chữ thường. Yêu cầu anh ta đặt lại mật khẩu của mình để mật khẩu có phân biệt chữ hoa chữ thường và do đó bảo mật hơn.

Xem Tham khảo Oracle Database Reference để biết thêm thông tin về DBA\_USERS view.

Làm thế nào Sensitivity ảnh hưởng đến tập tin mật khẩu

Bạn có thể bật hoặc tắt tính sensitivity cho các tệp mật khẩu bằng cách sử dụng đối số ignorecase trong tiện ích bằng dòng lệnh ORAPWD. Giá trị mặc định cho ignorecase là n (không), thực thi sensitivity.

[Ví dụ 3–4](#_bookmark287)  cho thấy làm thế nào để cho sensitivity trong các tập tin mật khẩu.

***Ví dụ 3–4 Bật trường hợp mật khẩu Sensitivity***

orapwd file=orapw entries=100 ignorecase=n Enter password for SYS: *password*

Điều này tạo ra một tập tin mật khẩu được gọi là orapwd. Bởi vì ignorecase được đặt thành n (không), mật khẩu được nhập sẽ phân biệt chữ hoa chữ thường. Sau đó, nếu bạn kết nối bằng mật khẩu này, nó sẽ thành công — miễn là bạn nhập nó bằng cách sử dụng sensitivity chính xác mà nó được tạo ra. Nếu bạn nhập cùng một mật khẩu nhưng với sensitivity khác nhau, nó sẽ thất bại.

Nếu bạn đặt ignorecase thành y thì mật khẩu trong tệp mật khẩu không phân biệt chữ hoa chữ thường, có nghĩa là bạn có thể nhập mật khẩu bằng cách sử dụng bất kỳ cách viết hoa nào bạn muốn.

Nếu bạn đã nhập tài khoản người dùng từ bản phát hành trước và các tài khoản này được tạo bằng đặc quyền SYSDBA hoặc SYSOPER, thì chúng sẽ được đưa vào tệp mật khẩu. Mật khẩu cho các tài khoản này phân biệt chữ hoa chữ thường. Lần sau, những người dùng này thay đổi mật khẩu của họ và giả sử sensitivity được bật, mật khẩu sẽ phân biệt chữ hoa chữ thường. Để bảo mật hơn, hãy để những người dùng này thay đổi mật khẩu của họ.

Xem Hướng dẫn của Quản trị viên Cơ sở dữ liệu Oracle để biết thêm thông tin về các tệp mật khẩu..

Làm thế nào sensitivity ảnh hưởng đến tài khoản tạo cho kết nối cơ sở dữ liệu liên kết

Khi bạn tạo kết nối liên kết cơ sở dữ liệu, bạn phải xác định tên người dùng và mật khẩu cho kết nối. Khi bạn tạo kết nối liên kết cơ sở dữ liệu, mật khẩu phân biệt chữ hoa chữ thường. Cách người dùng này nhập mật khẩu của họ cho các kết nối phụ thuộc vào bản phát hàng mà họ sử dụng:

* + Người dùng có thể kết nối từ cơ sở dữ liệu 11g trước khi phát hành với cơ sở dữ liệu Release 11g. Nếu độ nhạy trường hợp bị tắt trong cơ sở dữ liệu Release 11g, thì người dùng có thể nhập mật khẩu bằng bất kỳ. Tuy nhiên, nếu sensitivity được bật, người dùng phải nhập mật khẩu bằng cách sử dụng trường hợp mật khẩu được tạo trong cơ sở dữ liệu Release 11g.
  + Nếu người dùng kết nối từ cơ sở dữ liệu Release 11g với cơ sở dữ liệu 11g trước khi phát hành, họ có thể nhập mật khẩu của mình bằng bất kỳ trường hợp nào, vì mật khẩu vẫn không phân biệt chữ hoa chữ thường.

Bạn có thể tìm tài khoản người dùng cho các liên kết cơ sở dữ liệu hiện có bằng cách chạy chế độ xem V$DBLINK. Ví dụ:

SELECT DB\_LINK, OWNER\_ID FROM V$DBLINK;

Xem Tham khảo cơ sở dữ liệu Oracle để biết thêm thông tin về khung nhìn V$DBLINK.

#### Chống lại các mối đe dọa bảo mật mật khẩu bằng cách sử dụng thuật toán băm SHA-1

Thuật toán băm mật mã SHA-1 bảo vệ chống lại các mối đe dọa bảo mật dựa trên mật khẩu bằng cách bao gồm hỗ trợ cho các ký tự chữ hoa, ký tự đặc biệt và các ký tự nhiều byte trong mật khẩu. Ngoài ra, thuật toán băm SHA-1 cung cấp bảo vệ bổ sung. Điều này cho phép người dùng của bạn tạo mật khẩu phức tạp hơn, và do đó, làm cho nó khó khăn hơn cho kẻ xâm nhập truy cập vào các mật khẩu này. Oracle khuyến cáo bạn sử dụng thuật toán băm SHA-1.

Các phiên bản mật khẩu (còn được gọi là giá trị băm mật khẩu) được coi là cực kỳ nhạy cảm, vì chúng được sử dụng như một "bí mật được chia sẻ" giữa máy chủ và người đang đăng nhập. Nếu kẻ xâm nhập biết bí mật này, thì hãy bảo vệ xác thực ngay lập tức và bị xâm phạm nghiêm trọng. Hãy nhớ rằng người dùng quản trị có quyền quản lý tài khoản, người dùng quản trị có đặc quyền hệ thống SYSDBA, hoặc thậm chí người dùng có vai trò EXP\_FULL\_ DATABASE có thể truy cập ngay lập tức các giá trị băm mật khẩu. Do đó, loại người dùng quản trị này phải đáng tin cậy nếu tính toàn vẹn của xác thực dựa trên mật khẩu cơ sở dữ liệu được bảo toàn. Nếu bạn không thể tin tưởng những

quản trị viên, thì tốt hơn là triển khai một máy chủ (như Oracle Database Enterprise User Security) để các phiên bản mật khẩu vẫn nằm trong thư mục Enterprise User Security và không bao giờ có thể truy cập được với bất kỳ ai ngoại trừ người quản trị Enterprise User Security.

Bạn có thể tùy chọn cấu hình Oracle Database để chạy trong chế độ đọc quyền cho Release 11 hoặc mới hơn. Khi bạn bật chế độ độc quyền, khi đó Cơ sở dữ liệu Oracle sử dụng thuật toán băm SHA-1. Chế độ độc quyền của Oracle Database 11g tương thích với Oracle Database 10g và các sản phẩm mới hơn sử dụng các trình điều khiển dựa trên OCI, bao gồm SQL \* Plus, ODBC, Oracle .NET, Oracle Forms và các bộ điều hợp Oracle Database của bên thứ ba. Tuy nhiên, lưu ý rằng chế độ độc quyền cho Release 11g không tương thích với phiên bản JDBC type-4 (mỏng) sớm hơn Oracle Database 11g hoặc Oracle Database Client interface (OCI) dựa trên trình điều khiển sớm hơn Oracle Database 10g. Sau khi bạn cấu hình chế độ độc quyền, Oracle khuyên bạn nên xóa các giá trị băm mật khẩu trước đó khỏi dữ liệu.

Làm theo các bước sau:

1. Bật chế độ độc quyền.
   1. Tạo một bản sao lưu của tệp tham số sqlnet.ora, theo mặc định nằm trong thư mục $ORACLE\_HOME/network / admin trên các hệ điều hành UNIX và thư mục% ORACLE\_HOME% \ network \ admin trên các hệ điều hành Microsoft Windows.
   2. Đảm bảo rằng tệp sqlnet.ora có dòng sau:

SQLNET.ALLOWED\_LOGON\_VERSION=12

Nếu bạn đã áp dụng CPU tháng 10 năm 2012 hoặc nếu bạn đang sử dụng Oracle Database Release 11.2.0.3, thì hãy đảm bảo rằng bạn đặt SQLNET.ALLOWED\_LOGON\_VERSION thành 12, chứ không phải 11.

* 1. Lưu và thoát tệp sqlnet.ora.

1. Xác minh rằng mật khẩu trong tập lệnh thử nghiệm hoặc công việc phù hợp trong việc sử dụng các trường hợp hỗn hợp và các ký tự đặc biệt.
2. Thay đổi tất cả mật khẩu để bao gồm cả trường hợp hỗn hợp và các ký tự đặc biệt.

Oracle khuyên bạn nên sử dụng mật khẩu ngẫu nhiên với độ dài ít nhất mười hai ký tự. See Guideline [1](#_bookmark2169) under ["Guidelines for Securing Passwords"](#_bookmark2167) trên trang 10-7 để biết thêm hướng dẫn về cách tạo mật khẩu và các kỹ thuật để tạo mật khẩu phức tạp nhưng dễ nhớ.

**Xem thêm:** ["Downloading Security Patches and Contacting Oracle](#_bookmark2122) [Regarding Vulnerabilities" đến trang 10-2](#_bookmark2122)

#### Quản lý Mật khẩu Bên ngoài Bảo mật cho Thông tin Xác thực Mật khẩu

Nội dung:

■ [Giới](#_bookmark299) thiệu về các loại bào mật bên ngoài

■ [Nó](#_bookmark304) hoạt động như thế nào

■ [Cấu](#_bookmark307) hình để sử dụng

■ [Quản](#_bookmark311) lý thông tin

###### Giới thiệu về bảo mật mật khẩu an toàn từ bên ngoài

Bạn có thể lưu trữ thông tin đăng nhập mật khẩu để kết nối với cơ sở dữ liệu bằng cách sử dụng phiên Oracle từ phía máy khách. Một phiên Oracle là một thùng chứa phần mềm bảo mật lưu trữ chứng thực và xác nhận thông tin đăng nhập.

Việc sử dụng phiên này có thể đơn giản hóa việc triển khai quy mô lớn dựa trên thông tin đăng nhập mật khẩu để kết nối với cơ sở dữ liệu. Khi tính năng này được cấu hình, tập lệnh không còn cần tài khoản người dùng và mật khẩu được nhúng nữa. Điều này làm giảm nguy cơ vì mật khẩu không còn tiếp xúc nữa và các chính sách quản lý mật khẩu được thực thi dễ dàng hơn mà không thay đổi mã ứng dụng bất cứ khi nào tên người dùng hoặc mật khẩu thay đổi.

**Xem thêm:**

■ ["Using Proxy Authentication with the Secure External Password Store" đến trang 3-40](#_bookmark440)

■ *Oracle Database Advanced Security Administrator's Guide* để biết thông tin chung về ví Oracle

**Chú ý:** Kho lưu trữ xác thực bên ngoài là tách biệt với khu vực mà thông tin đăng nhập cơ sở hạ tầng khóa công khai (PKI) được lưu

trữ.Do đó, bạn không thể sử dụng Oracle Wallet Manager để quản lý thông tin đăng nhập. Thay vào đó, hãy sử dụng tiện ích dòng lệnh mkstore để quản lý các thông tin đăng nhập này.

###### Xác thực Mật khẩu Bên ngoài hoạt động như thế nào?

Thông thường, người dùng (và như các ứng dụng, công việc và các lệnh) kết nối với cơ sở dữ liệu bằng cách sử dụng một câu lệnh CONNECT xác định một chuỗi kết nối cơ sở dữ liệu. Chuỗi này có thể bao gồm một tài khoản người dùng và mật khẩu, và một tên dịch vụ Oracle Net xác định cơ sở dữ liệu trên một mạng cơ sở dữ liệu Oracle. Nếu mật khẩu bị bỏ qua, kết nối sẽ nhắc người dùng nhập mật khẩu.

Ví dụ: tên dịch vụ có thể là URL xác định cơ sở dữ liệu đó hoặc một biệt danh TNS mà bạn đã nhập vào tệp tnsnames.ora trong cơ sở dữ liệu. Một khả năng khác là một host: port: sid string.

Các ví dụ sau là các câu lệnh CONNECT chuẩn có thể được sử dụng cho một máy khách không được cấu hình để sử dụng kho lưu trữ mật khẩu bên ngoài:

CONNECT salesapp@sales\_db.us.example.com Enter password: *password*

CONNECT salesapp@orasales Enter password: *password*

CONNECT salesapp@ourhost37:1527:DB17 Enter password: *password*

Trong các ví dụ này, salesapp là tên người dùng, với chuỗi kết nối duy nhất cho cơ sở dữ liệu được chỉ định theo ba cách khác nhau. Bạn có thể sử dụng URL của nó sales\_ db.us.example.com hoặc các biệt danh TNS của nó từ tệp tnsnames.ora hoặc máy chủ lưu trữ của nó: port: sid string.

Tuy nhiên, khi các máy khách được cấu hình để sử dụng xác nhận mật khẩu an toàn bên ngoài, các ứng dụng có thể kết nối tới cơ sở dữ liệu với cú pháp câu lệnh CONNECT sau đây, mà không chỉ rõ các thông tin đăng nhập cơ sở dữ liệu:

CONNECT /@*db\_connect\_string*

CONNECT /*@db\_connect\_string* AS SYSDBA CONNECT /*@db\_connect\_string* AS SYSOPER

Trong đặc tả này, db\_connect\_string là một chuỗi kết nối hợp lệ để truy cập vào

cơ sở dữ liệu dự định, chẳng hạn như tên dịch vụ, URL hoặc bí danh như được trình bày trong các ví dụ trước đó. Mỗi tài khoản người dùng phải có chuỗi kết nối duy nhất của riêng nó; bạn không thể tạo một chuỗi kết nối cho nhiều người dùng.

Trong trường hợp này, thông tin đăng nhập cơ sở dữ liệu, tên người dùng và mật khẩu, được lưu trữ an toàn trong phiên làm việc của Oracle được tạo cho mục đích này. Tính năng autologin của ví này được bật, vì vậy hệ thống không cần mật khẩu để mở một phiên. Từ một phiên, nó nhận được thông tin đăng nhập để truy cập cơ sở dữ liệu cho người dùng.

**Xem thêm:** *Hướng dẫn của quản trị viên bảo mật nâng cao cơ sở dữ liệu Oracle để biết thông tin về ví autologin*

###### Định cấu hình ứng dụng khách để sử dụng xác nhân mật khẩu từ bên ngoài

Nếu máy khách của bạn đã được định cấu hình để sử dụng xác thực bên ngoài, chẳng hạn như xác thực gốc Windows hoặc Lớp cổng bảo mật (SSL), thì Cơ sở dữ liệu Oracle sử dụng phương thức xác thực đó. Các thông tin tương tự được sử dụng cho loại xác thực này thường được sử dụng để đăng nhập vào cơ sở dữ liệu.

Đối với các máy khách không sử dụng các phương thức xác thực như vậy hoặc muốn ghi đè chúng để xác thực cơ sở dữ liệu, bạn có thể đặt tham số SQLNET.WALLET\_OVERRIDE trong sqlnet.ora thành TRUE. Giá trị mặc định cho SQLNET.WALLET\_OVERRIDE là FALSE, cho phép sử dụng thông tin xác thực chuẩn như trước đây.

Nếu bạn muốn một khách hàng sử dụng tính năng lưu trữ mật khẩu bên ngoài an toàn, hãy thực hiện tác vụ cấu hình sau:

1. Tạo một ví trên máy khách bằng cách sử dụng cú pháp sau tại dòng lệnh:

mkstore -wrl *wallet\_location* -create

Ví dụ:

mkstore -wrl c:\oracle\product\11.2.0\db\_1\wallets -create Enter password: *password*

wallet\_location là đường dẫn đến thư mục mà bạn muốn tạo và lưu trữ xác thực. Lệnh này tạo ra một phiên Oracle với tính năng autologin được kích hoạt tại vị trí bạn chỉ định. Tính năng autologin cho phép khách hàng truy cập vào nội dung ví mà không cần cung cấp mật khẩu. Xem Hướng dẫn của quản trị viên bảo mật nâng cao cơ sở dữ liệu Oracle để biết thông tin về phiên autologin.

Tùy chọn tiện ích mkstore -treate sử dụng xác minh phức tạp mật khẩu. Xem ["Enforcing Password Complexity Verification" trên trang 3-11](#_bookmark268) để biết thêm thông tin chi tiết.

1. Tạo thông tin đăng nhập kết nối cơ sở dữ liệu trong ví bằng cách sử dụng cú pháp sau tại dòng lệnh:

mkstore -wrl *wallet\_location* -createCredential *db\_connect\_string username*

Enter password: *password*

Ví dụ:

mkstore -wrl c:\oracle\product\11.2.0\db\_1\wallets -createCredential orcl system

Enter password: *password*

Trong kỹ thuật này:

* + *wallet\_location Là đường dẫn đến thư mục mà bạn đã tạo trong Bước 1.*
  + db\_connect\_string là bí danh TNS mà bạn sử dụng để chỉ định cơ sở dữ liệu trong tệp tnsnames.ora hoặc bất kỳ tên dịch vụ nào bạn sử dụng để nhận dạng cơ sở dữ liệu trên mạng Oracle. Theo mặc định, tnsnames.ora nằm trong thư mục $ ORACLE\_ HOME / network / admin trên các hệ thống UNIX và trong ORACLE\_ HOME \ network \ admin trên Windows.
  + username là thông tin xác thực đăng nhập cơ sở dữ liệu. Khi được nhắc, hãy nhập mật khẩu cho người dùng này.

Lặp lại bước này cho mỗi cơ sở dữ liệu bạn muốn truy cập bằng cách sử dụng CONNECT /@*db\_ connect\_string* syntax.

**Chú ý:** The *db\_connect\_string* sử dụng CONNECT /@*db\_connect\_ string* statement phải giống hệt với db\_connect\_string được chỉ định trong lệnh -createCredential.

1. trong client sqlnet.ora file, nhập tham số WALLET\_LOCATION v à đặt nó ở vị trí của thử mục tại Bước [1](#_bookmark309).

Ví dụ, nếu bạn tạo wallet trong $ ORACLE\_HOME / network / admin và nhà Oracle của bạn được đặt thành / private / ora11, thì bạn cần nhập thông tin sau vào tệp sqlnet.ora của ứng dụng khách:

WALLET\_LOCATION = (SOURCE =

(METHOD = FILE) (METHOD\_DATA =

(DIRECTORY = */private/ora11/network/admin*)

)

)

1. trong sqlnet.ora file, nhập tham số vào SQLNET.WALLET\_OVERRIDE và đặt nó thành True như sau:

SQLNET.WALLET\_OVERRIDE = TRUE

Cài đặt này là: tất cả các câu lệnh CONNECT / @ db\_connect\_string để sử dụng thông tin trong wallet tại vị trí được chỉ định để xác thực cơ sở dữ liệu.

Khi xác thực bên ngoài được sử dụng, người dùng được xác thực với wallet như vậy có thể sử dụng cú pháp CONNECT / @ db\_connect\_string để truy cập cơ sở dữ liệu đã chỉ định trước đó mà không cung cấp tên người dùng và mật khẩu. Tuy nhiên, nếu người dùng không xác thực được bên ngoài, thì các câu lệnh kết nối này cũng không thành công.

**Chú ý:** Nếu một ứng dụng sử dụng SSL để mã hóa, thì tham số sqlnet.ora, SQLNET.AUTHENTICATION\_SERVICES, chỉ định SSL và một wallet SSL được tạo ra. Nếu ứng dụng này muốn sử dụng thông tin đăng nhập bí mật để xác thực cơ sở dữ liệu (thay vì chứng chỉ SSL), thì các thông tin đăng nhập đó phải được lưu trữ trong ví SSL. Sau khi xác thực SSL, nếu SQLNET.WALLET\_OVERRIDE = TRUE, thì tên người dùng và mật khẩu từ ví được sử dụng để xác thực cơ sở dữ liệu. Nếu SQLNET.WALLET\_OVERRIDE = FALSE, thì chứng chỉ SSL được sử dụng.

[Ví dụ 3–5](#_bookmark310)  hiển thị tệp sqlnet.ora mẫu với WALLET\_LOCATION và Các tham số SQLNET.WALLET\_OVERRIDE được thiết lập như được mô tả trong bước 3 và 4.

***Ví dụ 3–5 Tệp SQLNET.ORA mẫu với wallet***

WALLET\_LOCATION = (SOURCE =

(METHOD = FILE) (METHOD\_DATA =

(DIRECTORY = */private/ora11/network/admin*)

)

)

SQLNET.WALLET\_OVERRIDE = TRUE SSL\_CLIENT\_AUTHENTICATION = FALSE SSL\_VERSION = 0

###### Quản lý thông tin xác thực mật khẩu bên ngoài

Phần này tóm tắt các tác vụ sau: bạn có thể thực hiện để quản lý thông tin xác thực trong kho lưu trữ mật khẩu từ bên ngoài bằng cách sử dụng tiện ích dòng lệnh mkstore:

■ [Liệt](#_bookmark312) kê các mật khẩu xác thực từ bên ngoài

■ [Thêm](#_bookmark313) thông tin xác thực từ kho lưu trữ mật khẩu

■ [Xữa](#_bookmark314) thông tin xác thực trong kho lưu trữ mật khẩu

■ [Xóa](#_bookmark315) thông tin xác thực từ kho lưu trữ mật khẩu

**Liệt kê mật khẩu bên ngoài** Định kỳ, bạn có thể muốn xem tất cả nội dung của một kho lưu trữ mật khẩu bên ngoài wallet của khách hàng hoặc bạn có thể cần phải kiểm tra thông tin đăng nhập cụ thể bằng cách xem chúng. Liệt kê nội dung lưu trữ mật khẩu bên ngoài cung cấp thông tin cho bạn có thể để quyết định thêm hoặc xóa thông tin đăng nhập từ kho lưu trữ này hay không.

Để liệt kê nội dung của kho lưu trữ mật khẩu bên ngoài, nhập lệnh sau tại dòng lệnh:

mkstore -wrl *wallet\_location* -listCredential

Ví dụ:

mkstore -wrl c:\oracle\product\11.2.0\db\_1\wallets -listCredential

wallet\_location chỉ định đường dẫn đến thư mục chứa wallet tiền có nội dung lưu trữ mật khẩu bên ngoài bạn muốn xem. Lệnh này liệt kê tất cả các tên dịch vụ cơ sở dữ liệu thông tin xác thực (bí danh) và tên người dùng (lược đồ) tương ứng cho cơ sở dữ liệu đó. Mật khẩu không được liệt kê.

**Thêm thông tin xác thực vào kho mật khẩu bên ngoài** Bạn có thể lưu trữ nhiều thông tin xác thực trong một ví khách hàng. Ví dụ, nếu một công việc của khách hàng kết nối với hr\_database và một lệnh kết nối với sales\_database, thì bạn có thể lưu trữ thông tin đăng nhập trong cùng một wallet máy user. Tuy nhiên, bạn không thể lưu trữ nhiều thông tin đăng nhập (đăng nhập vào nhiều tài khoản) cho cùng một cơ sở dữ liệu trong cùng một wallet. Nếu bạn có nhiều thông tin xác thực đăng nhập cho cùng một cơ sở dữ liệu, thì chúng phải được lưu trữ trong các wallet riêng biệt.

Để thêm thông tin xác thực đăng nhập cơ sở dữ liệu vào wallet máy khách hiện có, hãy nhập lệnh sau tại dòng lệnh:

mkstore -wrl *wallet\_location* -createCredential *db\_alias username*

Ví dụ:

mkstore -wrl c:\oracle\product\11.2.0\db\_1\wallets -createCredential orcl system Enter password: *password*

Trong đặt điểm kỹ thuật này:

■ *wallet\_location* is đường dẫn đến thư mục mà wallet của khách hàng mà bạn muốn thêm thông tin đăng nhập được lưu trữ.

■ *db\_alias* can là bí danh TNS bạn sử dụng để chỉ định cơ sở dữ liệu trong tnsnames.ora tệp hoặc bất kỳ tên dịch vụ để xác định cơ sở dữ liệu trên mạng Oracle.

■ *username* is thông tin đăng nhập cơ sở dữ liệu cho lược đồ mà ứng dụng của bạn kết nối. Khi được nhắc, hãy nhập mật khẩu cho người dùng này.

**Sửa đổi thông tin xác thực từ kho lưu trữ mật khẩu bên ngoài** Nếu các chuỗi kết nối cơ sở dữ liệu thay đổi, thì bạn có thể sửa đổi thông tin đăng nhập cơ sở dữ liệu được lưu trữ trong wallet.

Để sửa đổi thông tin đăng nhập cơ sở dữ liệu trong ví, hãy nhập lệnh sau tại dòng lệnh:

mkstore -wrl *wallet\_location* -modifyCredential *dbase\_alias username*

Ví dụ:

mkstore -wrl c:\oracle\product\11.2.0\db\_1\wallets -modifyCredential sales\_db Enter password: *password*

Trong đặc điểm kỹ thuật này:

■ *wallet\_location* is đường dẫn đến thư mục có wallet.

■ *db\_alias* is một bí danh mới hoặc khác biệt mà bạn muốn sử dụng để xác định cơ sở dữ liệu. Nó có thể là một bí danh TNS mà bạn sử dụng để chỉ định cơ sở dữ liệu trong tệp tnsnames.ora hoặc bất kỳ tên dịch vụ nào mà bạn sử dụng để xác định cơ sở dữ liệu trên một mạng Oracle.

■ *username* là thông tin xác thực đăng nhập cơ sở dữ liệu mới hoặc khác. Khi được nhắc, hãy nhập mật khẩu cho người dùng này.

**Xóa thông tin xác thực từ kho mật khẩu bên ngoài** Nếu cơ sở dữ liệu không còn tồn tại hoặc nếu bạn muốn vô hiệu hóa các kết nối đến một cơ sở dữ liệu cụ thể, thì bạn có thể xóa tất cả thông tin đăng nhập cho cơ sở dữ liệu đó khỏi wallet.

Để xóa thông tin xác thực đăng nhập cơ sở dữ liệu từ ví, hãy nhập lệnh sau tại dòng lệnh:

mkstore -wrl *wallet\_location* -deleteCredential *db\_alias*

Ví dụ:

mkstore -wrl c:\oracle\product\11.2.0\db\_1\wallets -deleteCredential orcl

Trong đặc điểm kỹ thuật này:

■ *wallet\_location* là đường dẫn đến thư mục có wallet.

■ *db\_alias* là bí danh TNS bạn sử dụng để chỉ định cơ sở dữ liệu trong tệp tnsnames.ora hoặc bất kỳ tên dịch vụ nào bạn sử dụng để xác định cơ sở dữ liệu trên mạng Cơ sở dữ liệu Oracle.

### Xác thực quản trị viên cơ sở dữ liệu

Người quản trị cơ sở dữ liệu thực hiện các hoạt động đặc biệt, chẳng hạn như tắt hoặc khởi động cơ sở dữ liệu, không nên được thực hiện bởi người dùng cơ sở dữ liệu không quản trị.

Cơ sở dữ liệu Oracle cung cấp các phương thức sau để bảo đảm xác thực các quản trị cơ sở dữ liệu có đặc quyền SYSDBA hoặc SYSOPER:

■ [Xác thực mạnh mẽ và quản lý tập trung cho quản trị viên cơ sở dữ liệu](#_bookmark318)

■ [Xác thực quản trị viên cơ sở dữ liệu bằng cách sử dụng hệ điều hành](#_bookmark329)

■ [Xác thực quản trị viên cơ sở dữ liệu bằng cách sử dụng mật khẩu của họ](#_bookmark332)

#### Xác thực mạnh mẽ và quản lý tập trung cho quản trị viên cơ sở dữ liệu

Xác thực mạnh cho phép bạn kiểm soát tập trung quyền truy cập SYSDBA và SYSOPER vào nhiều cơ sở dữ liệu. Xem xét sử dụng loại xác thực này để quản trị cơ sở dữ liệu cho các tình huống sau:

■ Bạn có thắc mắc về lỗ hổng tệp mật khẩu.

■ Trang web của bạn có yêu cầu bảo mật rất nghiêm ngặt.

■ Bạn muốn tách quản lý danh tính khỏi cơ sở dữ liệu của mình. Bằng cách sử dụng một máy chủ thư mục như Oracle Internet Directory (OID), ví dụ, bạn có thể duy trì, bảo mật và quản trị máy chủ đó một cách riêng biệt.

Để kích hoạt máy chủ Thư mục Internet của Oracle để cho phép các kết nối SYSDBA và SYSOPER, hãy sử dụng một trong các phương pháp sau, tùy thuộc vào môi trường của bạn:

■ [Cấu hình xác thực thư mục cho người quản trị](#_bookmark323)

■ [Cấu hình Xác thực Kerberos cho Người Quản trị](#_bookmark325)

■ [Cấu hình xác thực lớp cổng bảo mật cho người quản trị](#_bookmark327)

**Cấu hình xác thực thư mục cho người quản trị:** Để Cấu hình xác thực thư mục cho người quản trị:

1. Định cấu hình người dùng quản trị bằng cách sử dụng các quy trình tương tự bạn sẽ sử dụng để định cấu hình người dùng thông thường.

2. Trong Oracle Internet Directory, cấp đặc quyền SYSDBA hoặc SYSOPER cho người dùng cho cơ sở dữ liệu mà người dùng này sẽ quản lý.

Grant SYSDBA or SYSOPER chỉ cho người dùng đáng tin cậy. Xem ["Guidelines for Securing User](#_bookmark2131) [Accounts and Privileges" tại trang 10-2](#_bookmark2131) for advice on this topic.

3. Đặt tham số khởi tạo LDAP\_DIRECTORY\_SYSAUTH thành YES::

ALTER SYSTEM SET LDAP\_DIRECTORY\_SYSAUTH = YES;

Khi được đặt thành YES, tham số LDAP\_DIRECTORY\_SYSAUTH cho phép người dùng SYSDBA và SYSOPER xác thực với cơ sở dữ liệu bằng cách sử dụng phương thức xác thực mạnh.

Xem Tham khảo cơ sở dữ liệu Oracle để biết thêm thông tin về LDAP\_DIRECTORY\_ SYSAUTH.

1. Đặt tham số LDAP\_DIRECTORY\_ACCESS thành PASSWORD hoặc SSL. Ví dụ:

ALTER SYSTEM SET LDAP\_DIRECTORY\_ACCESS = PASSWORD;

Đảm bảo rằng tham số khởi tạo LDAP\_DIRECTORY\_ACCESS không được đặt thành KHÔNG. Đặt tham số này thành PASSWORD hoặc SSL đảm bảo rằng người dùng có thể được xác thực bằng các đặc quyền SYSDBA hoặc SYSOPER thông qua Oracle Internet Directory. Xem Tham khảo cơ sở dữ liệu Oracle để biết thêm thông tin về LDAP\_DIRECTORY\_ACCESS.

Sau đó, người dùng này có thể đăng nhập bằng cách bao gồm tên dịch vụ mạng trong câu lệnh CONNECT trong SQL \* Plus. Ví dụ, để đăng nhập với tên SYSDBA nếu tên dịch vụ mạng là orcl:

CONNECT SOMEUSER@ORCL AS SYSDBA

Enter password: *password*

Nếu cơ sở dữ liệu được cấu hình để sử dụng một tệp mật khẩu để xác thực từ xa, Oracle Database kiểm tra tệp mật khẩu trước tiên.

###### Định cấu hình Xác thực Kerberos cho Người dùng Quản trị

Để định cấu hình xác thực Kerberos cho người dùng quản trị:

1. Định cấu hình người dùng quản trị bằng cách sử dụng các quy trình tương tự bạn sẽ sử dụng để định cấu hình người dùng thông thường.

Xem Hướng dẫn của quản trị viên bảo mật nâng cao cơ sở dữ liệu Oracle để biết thêm thông tin.

1. Cấu hình Oracle Internet Directory để xác thực Kerberos.

Xem Hướng dẫn của Quản trị viên Bảo mật Người dùng Cơ sở dữ liệu Oracle để biết thêm thông tin.

1. Trong Oracle Internet Directory, cấp đặc quyền SYSDBA hoặc SYSOPER cho người dùng cho cơ sở dữ liệu mà người dùng này sẽ quản lý.

Chỉ cấp SYSDBA hoặc SYSOPER cho người dùng đáng tin cậy. Xem "Nguyên tắc bảo mật tài khoản người dùng và đặc quyền" trên trang 10-2 để được tư vấn về chủ đề này.

1. Đặt tham số khởi tạo LDAP\_DIRECTORY\_SYSAUTH thành YES:

ALTER SYSTEM SET LDAP\_DIRECTORY\_SYSAUTH = YES;

Khi được đặt thành YES, tham số LDAP\_DIRECTORY\_SYSAUTH cho phép người dùng SYSDBA và SYSOPER xác thực với cơ sở dữ liệu bằng cách sử dụng các phương thức xác thực mạnh. Xem Tham khảo cơ sở dữ liệu Oracle để biết thêm thông tin về LDAP\_ DIRECTORY\_SYSAUTH.

1. Đặt tham số LDAP\_DIRECTORY\_ACCESS thành PASSWORD hoặc SSL. Ví dụ:

ALTER SYSTEM SET LDAP\_DIRECTORY\_ACCESS = SSL;

Đảm bảo rằng tham số khởi tạo LDAP\_DIRECTORY\_ACCESS không được đặt thành KHÔNG. Đặt tham số này thành PASSWORD hoặc SSL đảm bảo rằng người dùng có thể được xác thực bằng SYSDBA hoặc SYSOPER thông qua Oracle Internet Directory. Xem Tham khảo cơ sở dữ liệu Oracle để biết thêm thông tin về LDAP\_DIRECTORY\_ACCESS.

Sau đó, người dùng này có thể đăng nhập bằng cách nhập câu lệnh CONNECT trong SQL \* Plus. Ví dụ, để đăng nhập với tên SYSDBA nếu tên dịch vụ mạng là orcl:

CONNECT /@orcl AS SYSDBA

###### Định cấu hình xác thực lớp cổng bảo mật cho người dùng quản trị

Để định cấu hình xác thực Lớp cổng bảo mật (SSL) cho người dùng quản trị:

1. Cấu hình máy khách để sử dụng SSL:
   1. Định cấu hình wallet máy khách và chứng chỉ người dùng. Cập nhật vị trí wallet trong tệp cấu hình sqlnet.ora.

Bạn có thể sử dụng Trình quản lý Wallet để định cấu hình ví tiền và chứng chỉ người dùng của ứng dụng khách. Xem Hướng dẫn của quản trị viên bảo mật nâng cao cơ sở dữ liệu Oracle để biết thêm thông tin.

* 1. Cấu hình tên dịch vụ mạng Oracle để bao gồm các máy chủ DN và sử dụng TCP / IP với SSL trong tnsnames.ora.
  2. Cấu hình TCP / IP với SSL trong listener.ora.
  3. Đặt bộ mã hóa SSL của khách hàng và phiên bản SSL bắt buộc, sau đó đặt SSL làm dịch vụ xác thực trong sqlnet.ora.

1. Cấu hình máy chủ để sử dụng SSL:
   1. Bật SSL cho trình nghe cơ sở dữ liệu của bạn trên TCPS và cung cấp tên TNS tương ứng. Bạn có thể sử dụng Net Configuration Assistant để cấu hình tên TNS.
   2. Lưu trữ thông tin đăng nhập cơ sở dữ liệu PKI trong ví cơ sở dữ liệu. Bạn có thể sử dụng Trình quản lý Wallet thực hiện việc này.
   3. Đặt tham số khởi tạo LDAP\_DIRECTORY\_ACCESS thành SSL:

ALTER SYSTEM SET LDAP\_DIRECTORY\_ACCESS = SSL;

Xem Tham khảo cơ sở dữ liệu Oracle để biết thêm thông tin về LDAP\_DIRECTORY\_ ACCESS.

1. Cấu hình Oracle Internet Directory để xác thực người dùng SSL.

Xem Hướng dẫn của quản trị viên bảo mật người dùng cơ sở dữ liệu Oracle để biết thông tin về định cấu hình xác thực SSL bảo mật người dùng doanh nghiệp.

1. Trong Oracle Internet Directory, cấp đặc quyền SYSDBA hoặc SYSOPER cho người dùng cho cơ sở dữ liệu mà người dùng sẽ quản lý.
2. Trên máy tính của máy chủ, đặt tham số khởi tạo LDAP\_DIRECTORY\_SYSAUTH thành YES.

ALTER SYSTEM SET LDAP\_DIRECTORY\_SYSAUTH = YES;

Khi được đặt thành YES, tham số LDAP\_DIRECTORY\_SYSAUTH cho phép người dùng SYSDBA và SYSOPER xác thực với cơ sở dữ liệu bằng cách sử dụng phương thức xác thực mạnh. Xem Tham khảo cơ sở dữ liệu Oracle để biết thêm thông tin về LDAP\_ DIRECTORY\_SYSAUTH.

Sau đó, người dùng này có thể đăng nhập bằng cách bao gồm tên dịch vụ mạng trong câu lệnh CONNECT trong SQL \* Plus. Ví dụ, để đăng nhập với tên SYSDBA nếu tên dịch vụ mạng là orcl:

CONNECT /@orcl AS SYSDBA

#### Xác thực quản trị viên cơ sở dữ liệu bằng cách sử dụng hệ điều hành

Xác thực hệ điều hành cho người quản trị cơ sở dữ liệu thường liên quan đến việc thiết lập một nhóm trên hệ điều hành, cấp đặc quyền DBA cho nhóm đó, và sau đó thêm tên của những người cần có những đặc quyền đó vào nhóm đó. (Trên các hệ thống UNIX, nhóm là nhóm dba.)

Trên các hệ thống Microsoft Windows, người dùng kết nối với đặc quyền SYSDBA có thể tận dụng xác thực gốc Windows. Nếu những người dùng này làm việc với Cơ sở dữ liệu Oracle bằng cách sử dụng tài khoản miền của họ, thì bạn phải cấp rõ ràng cho họ quyền quản trị cục bộ và tư cách thành viên ORA\_DBA.

**Xem thêm:** Tài liệu dành riêng cho hệ điều hành cơ sở dữ liệu Oracle của bạn để biết thông tin về cấu hình xác thực hệ điều hành của quản trị viên cơ sở dữ liệu

#### Xác thực quản trị viên cơ sở dữ liệu bằng cách sử dụng mật khẩu của họ

Cơ sở dữ liệu Oracle sử dụng các tệp mật khẩu dành riêng cho cơ sở dữ liệu để theo dõi các tên người dùng cơ sở dữ liệu đã được cấp các đặc quyền SYSDBA và SYSOPER. Các đặc quyền này cho phép các hoạt động sau:

* Đặc quyền hệ thống SYSOPER cho phép người quản trị cơ sở dữ liệu thực hiện các hoạt động STARTUP, SHUTDOWN, ALTER DATABASE OPEN / MOUNT, ALTER DATABASE BACKUP, ARCHIVE LOG và RECOVER. SYSOPER cũng bao gồm đặc quyền RESTRICTED SESSION.
* Đặc quyền hệ thống SYSDBA có tất cả các đặc quyền hệ thống với ADMIN OPTION, bao gồm đặc quyền hệ thống SYSOPER, và cho phép CREATE DATABASE và phục hồi theo thời gian.
* Một tệp mật khẩu có chứa người dùng có đặc quyền SYSDBA hoặc SYSOPER có thể được chia sẻ giữa các cơ sở dữ liệu khác nhau. Bạn có thể có tệp mật khẩu được chia sẻ có chứa người dùng ngoài người dùng SYS. Để chia sẻ tệp mật khẩu giữa các cơ sở dữ liệu khác nhau, hãy đặt tham số REMOTE\_LOGIN\_PASSWORDFILE trong tệp init.ora thành SHARED.

Nếu bạn đặt tham số khởi tạo REMOTE\_LOGIN\_PASSWORDFILE thành EXCLUSIVE hoặc SHARED từ KHÔNG, thì hãy đảm bảo rằng tệp mật khẩu được đồng bộ hóa với mật khẩu từ điển. Xem Hướng dẫn của quản trị viên cơ sở dữ liệu Oracle để biết thêm thông tin.

* Xác thực dựa trên tập tin mật khẩu được bật theo mặc định. Điều này có nghĩa là cơ sở dữ liệu sẵn sàng sử dụng tệp mật khẩu để xác thực người dùng có đặc quyền hệ thống SYSDBA hoặc SYSOPER. Xác thực dựa trên tệp mật khẩu được kích hoạt ngay sau khi bạn tạo tệp mật khẩu bằng tiện ích ORAPWD.
* Đặc quyền hệ thống SYSDBA có tất cả các đặc quyền hệ thống với ADMIN OPTION, bao gồm đặc quyền hệ thống SYSOPER, và cho phép CREATE DATABASE và phục hồi theo thời gian.
* Một tệp mật khẩu có chứa người dùng có đặc quyền SYSDBA hoặc SYSOPER có thể được chia sẻ giữa các cơ sở dữ liệu khác nhau. Bạn có thể có tệp mật khẩu được chia sẻ có chứa người dùng ngoài người dùng SYS. Để chia sẻ tệp mật khẩu giữa các cơ sở dữ liệu khác nhau, hãy đặt tham số REMOTE\_LOGIN\_PASSWORDFILE trong tệp init.ora thành SHARED.

Nếu bạn đặt tham số khởi tạo REMOTE\_LOGIN\_PASSWORDFILE thành EXCLUSIVE hoặc SHARED từ KHÔNG, thì hãy đảm bảo rằng tệp mật khẩu được đồng bộ hóa với mật khẩu từ điển. Xem Hướng dẫn của quản trị viên cơ sở dữ liệu Oracle để biết thêm thông tin.

* Xác thực dựa trên tập tin mật khẩu được bật theo mặc định. Điều này có nghĩa là cơ sở dữ liệu sẵn sàng sử dụng tệp mật khẩu để xác thực người dùng có đặc quyền hệ thống SYSDBA hoặc SYSOPER. Xác thực dựa trên tệp mật khẩu được kích hoạt ngay sau khi bạn tạo tệp mật khẩu bằng tiện ích ORAPWD.

Bất kỳ ai có EXECUTE đặc quyền và viết đặc quyền cho $ ORACLE\_HOME / dbs

thư mục có thể chạy tiện ích ORAPWD.

Tuy nhiên, lưu ý rằng việc sử dụng các tập tin mật khẩu có thể gây ra rủi ro bảo mật. Vì lý do này, hãy xem xét sử dụng các phương pháp xác thực được mô tả trong ["Strong Authentication and](#_bookmark318) [Centralized Management for Database Administrators"](#_bookmark318) trên trang 3-22. Ví dụ về rủi ro bảo mật mật khẩu như sau:

* Kẻ xâm nhập có thể lấy cắp hoặc tấn công tập tin mật khẩu.
* Nhiều người dùng không thay đổi mật khẩu mặc định.
* Mật khẩu có thể dễ dàng đoán được.
* Mật khẩu là dễ bị tổn thương nếu nó có thể được tìm thấy trong một từ điển.

■ Mật khẩu quá ngắn, có thể được chọn để dễ gõ, rất dễ bị tổn thương nếu kẻ xâm nhập có được mật mã băm mật khẩu.

**Chú ý:** Các kết nối được yêu cầu AS SYSDBA hoặc AS SYSOPER phải sử dụng các cụm từ này; không có chúng, kết nối không thành công. Tham số cơ sở dữ liệu Oracle O7\_DICTIONARY\_ACCESSIBILITY được đặt thành FALSE theo mặc định, để giới hạn truy cập từ điển dữ liệu nhạy cảm chỉ cho những người được ủy quyền. Tham số cũng thực thi cú pháp AS SYSDBA hoặc AS SYSOPER bắt buộc.

**Xem thêm:** *Oracle Database Administrator's Guide* để biết thông tin về cách tạo và duy trì tệp mật khẩu

### Sử dụng cơ sở dữ liệu để xác thực người dùng

Nội dung:

■ [Về xác thực cơ sở dữ liệu](#_bookmark341)

■ [Ưu điểm của xác thực cơ sở dữ liệu](#_bookmark345)

■ [Tạo một người dùng được xác thực bởi cơ sở dữ liệu](#_bookmark347)

#### Về xác thực cơ sở dữ liệu

Cơ sở dữ liệu Oracle có thể xác thực người dùng đang cố gắng kết nối với cơ sở dữ liệu bằng cách sử dụng thông tin được lưu trữ trong cơ sở dữ liệu đó. Để cấu hình Oracle Database để sử dụng xác thực cơ sở dữ liệu, bạn phải tạo mỗi người dùng với một mật khẩu liên quan. Tên người dùng có thể là multibyte, nhưng mỗi mật khẩu phải bao gồm các ký tự một byte, ngay cả khi cơ sở dữ liệu của bạn sử dụng một bộ ký tự nhiều byte. Người dùng phải cung cấp tên người dùng và mật khẩu này khi cố thiết lập kết nối. Cơ sở dữ liệu Oracle lưu trữ mật khẩu người dùng trong từ điển dữ liệu ở định dạng được mã hóa.

Để xác định các giao thức xác thực được cho phép bởi máy khách hoặc cơ sở dữ liệu, người quản trị cơ sở dữ liệu có thể thiết lập một cách rõ ràng tham số SQLNET.ALLOWED\_LOGON\_VERSION trong tệp sqlnet.ora của máy chủ. Mỗi lần thử kết nối được kiểm tra và nếu máy khách hoặc máy chủ không đáp ứng được phiên bản tối thiểu được chỉ định bởi đối tác của nó, xác thực không thành công với một lỗi giao thức xác thực không khớp ORA-28040. Tham số có thể lấy các giá trị 11, 10, 9, hoặc 8. Giá trị mặc định là 8. Các giá trị này đại diện cho các phiên bản máy chủ cơ sở dữ liệu. Oracle đề xuất giá trị 11 để bảo vệ mạnh nhất. Tuy nhiên, lưu ý rằng nếu bạn đặt SQLNET.ALLOWED\_LOGON\_VERSION thành 11, thì các ứng dụng khách trước của Oracle Database Release 11.1 hoặc các máy khách mỏng JDBC không thể xác thực với cơ sở dữ liệu Oracle bằng cách sử dụng xác thực dựa trên mật khẩu.

Để tăng cường bảo mật khi sử dụng xác thực cơ sở dữ liệu, Oracle khuyên bạn nên sử dụng quản lý mật khẩu, bao gồm khóa tài khoản, mật khẩu và thời gian hết hạn, lịch sử mật khẩu và xác minh phức tạp mật khẩu. Xem "Sử dụng Chính sách quản lý mật khẩu" trên trang 3-3 để biết thêm thông tin về quản lý mật khẩu.

#### Ưu điểm của xác thực cơ sở dữ liệu

Các ưu điểm của xác thực cơ sở dữ liệu như sau:

■ Tài khoản người dùng và tất cả xác thực được kiểm soát bởi cơ sở dữ liệu. Không có sự phụ thuộc vào bất cứ thứ gì bên ngoài cơ sở dữ liệu.

■ Cơ sở dữ liệu Oracle cung cấp các tính năng quản lý mật khẩu mạnh mẽ để tăng cường bảo mật khi sử dụng xác thực cơ sở dữ liệu.

■ Việc quản trị dễ dàng hơn khi có cộng đồng người dùng nhỏ.

#### Tạo một người dùng được xác thực bởi cơ sở dữ liệu

Câu lệnh SQL sau tạo ra một người dùng được xác định và xác thực bởi Cơ sở dữ liệu Oracle. Người dùng sebastian phải chỉ định mật khẩu được gán bất cứ khi nào anh ta kết nối với Cơ sở dữ liệu Oracle.

CREATE USER sebastian IDENTIFIED BY *password*;

### Sử dụng hệ điều hành để xác thực người dùng

Một số hệ điều hành cho phép Oracle Database sử dụng thông tin mà họ duy trì để xác thực người dùng. Điều này có những lợi ích sau:

■ Sau khi được xác thực bởi hệ điều hành, người dùng có thể kết nối với cơ sở dữ liệu Oracle thuận tiện hơn mà không cần chỉ định tên người dùng hoặc mật khẩu. Ví dụ, một người dùng được xác thực bởi hệ điều hành có thể gọi SQL \* Plus và bỏ qua các lời nhắc tên người dùng và mật khẩu bằng cách nhập lệnh sau tại dòng lệnh:

SQLPLUS /

Within SQL\*Plus, you enter:

CONNECT /

■ Với quyền kiểm soát xác thực người dùng tập trung trong hệ điều hành, Oracle Database không cần lưu trữ hoặc quản lý mật khẩu người dùng, mặc dù nó vẫn duy trì tên người dùng trong cơ sở dữ liệu.

■ Các đường mòn kiểm tra trong cơ sở dữ liệu và hệ điều hành có thể sử dụng cùng tên người dùng.

■ Bạn có thể xác thực cả người dùng hệ điều hành và người dùng hệ điều hành không trong cùng một hệ thống. Ví dụ:

**– Xác thực người dùng bằng hệ điều hành.** Bạn tạo tài khoản người dùng bằng mệnh đề IDENTIFIED EXTERNALLY của câu lệnh CREATE USER, và sau đó bạn đặt tham số khởi tạo OS\_AUTHENT\_PREFIX để chỉ định tiền tố mà Oracle Database sử dụng để xác thực người dùng đang cố kết nối với máy chủ.

**– Xác thực người dùng hệ điều hành không hoạt động.** Đây là những người dùng được chỉ định mật khẩu và được xác thực bởi cơ sở dữ liệu.

**– Xác thực người dùng Cơ sở dữ liệu Oracle Enterprise Enterprise Security.** Các tài khoản người dùng này được tạo bằng cách sử dụng mệnh đề GLOBALLY IDENTIFIED của câu lệnh CREATE USER, và sau đó được xác thực bởi Oracle Internet Directory (OID) hiện có trong cùng một cơ sở dữ liệu.

Tuy nhiên, bạn nên lưu ý những hạn chế sau đây khi sử dụng hệ điều hành để xác thực người dùng:

■ Người dùng phải có tài khoản hệ điều hành trên máy tính phải được truy cập. Không phải tất cả người dùng đều có tài khoản hệ điều hành, đặc biệtngười dùng không quản trị.

■ Nếu người dùng đã đăng nhập bằng cách sử dụng phương pháp này và bước ra khỏi thiết bị đầu cuối, người dùng khác có thể dễ dàng đăng nhập vì người dùng này không cần bất kỳ mật khẩu hoặc thông tin đăng nhập nào. Điều này có thể gây ra một vấn đề an ninh nghiêm trọng.

■ Khi một hệ điều hành được sử dụng để xác thực người dùng cơ sở dữ liệu, việc quản lý các môi trường cơ sở dữ liệu phân tán và các liên kết cơ sở dữ liệu đòi hỏi sự chăm sóc đặc biệt. Các liên kết cơ sở dữ liệu được xác thực bởi hệ điều hành có thể gây ra một điểm yếu bảo mật. Vì lý do này, Oracle khuyên bạn không nên sử dụng chúng.

**Xem thêm:**

■ *Oracle Database Administrator's Guide* để biết thêm thông tin về xác thực, hệ điều hành, các khái niệm cơ sở dữ liệu phân tán và quản lý dữ liệu phân tán

■ Operating system-specific documentation by Oracle Database để biết thêm thông tin về xác thực bằng cách sử dụng hệ điều hành của bạn

### Sử dụng mạng để xác thực người dùng

Bạn có thể xác thực người dùng qua mạng bằng cách sử dụng Lớp cổng bảo mật với các dịch vụ của bên thứ ba.

■ [Xác thực bằng cách sử dụng Lớp cổng bảo mật](#_bookmark357)

■ [Xác thực bằng cách sử dụng dịch vụ của bên thứ ba](#_bookmark361)

#### Xác thực bằng cách sử dụng Lớp cổng bảo mật

Giao thức Lớp cổng bảo mật (SSL) là giao thức lớp ứng dụng. Bạn có thể sử dụng nó để xác thực người dùng vào cơ sở dữ liệu và nó độc lập với quản lý người dùng toàn cầu trong Oracle Internet Directory. Tức là, người dùng có thể sử dụng SSL để xác thực cơ sở dữ liệu mà không cần máy chủ thư mục.

See *Oracle Database Advanced Security Administrator's Guide* để được hướng dẫn về cách định cấu hình SSL.

#### Xác thực bằng cách sử dụng dịch vụ của bên thứ ba

Bạn cần sử dụng các dịch vụ xác thực mạng của bên thứ ba nếu bạn muốn xác thực người dùng cơ sở dữ liệu Oracle qua mạng. Ví dụ nổi bật bao gồm Kerberos, PKI (cơ sở hạ tầng khóa công khai), RADIUS (Dịch vụ người dùng quay số xác thực từ xa) và các dịch vụ dựa trên thư mục, như được mô tả trong các phần sau.

Nếu các dịch vụ xác thực mạng có sẵn cho bạn, thì cơ sở dữ liệu Oracle có thể chấp nhận xác thực từ dịch vụ mạng. Nếu bạn sử dụng dịch vụ xác thực mạng, thì một số cân nhắc đặc biệt phát sinh đối với vai trò mạng và liên kết cơ sở dữ liệu.

**Chú ý:** Để sử dụng dịch vụ xác thực mạng với Oracle Database, bạn cần Oracle Database Enterprise Edition với tùy chọn Oracle Database Advanced Security.

**Xem thêm:** *Oracle Database Advanced Security Administrator's Guide* để biết thông tin về Oracle Enterprise Edition với tùy chọn Oracle Database Advanced Security

Xác thực bằng Kerberos

Kerberos là một hệ thống xác thực của bên thứ ba đáng tin cậy dựa trên các bí mật được chia sẻ. Nó giả định rằng bên thứ ba là an toàn, và cung cấp khả năng đăng nhập một lần, lưu trữ mật khẩu tập trung, xác thực liên kết cơ sở dữ liệu và bảo mật PC nâng cao.

Nó thực hiện điều này thông qua một máy chủ xác thực Kerberos hoặc thông qua Cybersafe Active Trust, một máy chủ xác thực dựa trên Kerberos thương mại.

**Xem thêm:** *Oracle Database Advanced Security Administrator's Guide*

để biết thêm thông tin về Kerberos

Xác thực bằng RADIUS

Cơ sở dữ liệu Oracle hỗ trợ xác thực từ xa của người dùng thông qua Dịch vụ người dùng quay số xác thực từ xa (RADIUS), một giao thức gọn nhẹ tiêu chuẩn được sử dụng để xác thực người dùng, ủy quyền và kế toán. Tính năng này cũng cho phép người dùng sử dụng RSA One-Time Password Specification (OTPS) để xác thực cơ sở dữ liệu Oracle.

**Xem thêm:**

■ *Oracle Database Advanced Security Administrator's Guide* để biết thông tin về cấu hình RADIUS

■ Tài liệu RSA về OTPS

Xác thực bằng cách sử dụng dịch vụ dựa trên thư mục

Sử dụng một thư mục trung tâm có thể làm cho xác thực và quản trị của nó hiệu quả. Các dịch vụ dựa trên thư mục bao gồm:

* **Thư mục Internet của Oracle**, sử dụng Giao thức truy cập thư mục hạng nhẹ (LDAP), sử dụng kho lưu trữ trung tâm để lưu trữ và quản lý thông tin về người dùng (được gọi là người dùng doanh nghiệp) có tài khoản được tạo trong môi trường phân tán. Mặc dù người dùng cơ sở dữ liệu phải được tạo (với mật khẩu) trong mỗi cơ sở dữ liệu mà họ cần truy cập, thông tin người dùng doanh nghiệp có thể truy cập tập trung trong Thư mục Internet của Oracle. Bạn cũng có thể tích hợp thư mục này với Microsoft Active Directory và SunOne.

Để biết thêm thông tin về Oracle Internet Directory, xem *Oracle Internet Directory* *Administrator's Guide*.

* **Trình quản lý bảo mật doanh nghiệp Oracle** cho phép bạn lưu trữ và truy xuất các vai trò từ Oracle Internet Directory, cung cấp khả năng quản lý đặc quyền tập trung để giúp quản trị dễ dàng hơn và tăng mức bảo mật. Để biết thêm thông tin về Oracle Enterprise Security Manager, hãy xem Cấu hình nâng cao của Trình quản lý doanh nghiệp Oracle.

Xác thực bằng cơ sở hạ tầng khóa công khai

Hệ thống xác thực dựa trên cơ sở hạ tầng khóa công khai (PKI) cấp chứng chỉ kỹ thuật số cho khách hàng người dùng, sử dụng chúng để xác thực trực tiếp với máy chủ trong doanh nghiệp mà không trực tiếp liên quan đến máy chủ xác thực. Cơ sở dữ liệu Oracle cung cấp PKI để sử dụng khóa và chứng chỉ công khai, bao gồm các thành phần sau:

* **Xác thực và quản lý khóa phiên bảo mật bằng SSL.** Xem ["Authentication Using Secure Sockets Layer"](#_bookmark357) trên trang 3-28 để biết thêm thông tin.
* **Chứng chỉ tin cậy.** Chúng được sử dụng để xác định các thực thể của bên thứ ba được tin cậy là người ký chứng chỉ người dùng khi danh tính đang được xác thực. Khi chứng chỉ người dùng đang được xác nhận, người ký sẽ được kiểm tra bằng cách sử dụng các điểm tin cậy hoặc chuỗi chứng chỉ đáng tin cậy của các tổ chức phát hành chứng chỉ được lưu trữ trong hệ thống xác nhận hợp lệ. Nếu có nhiều cấp độ chứng chỉ đáng tin cậy trong chuỗi này, thì chứng chỉ đáng tin cậy ở cấp thấp hơn chỉ đơn giản là đáng tin cậy mà không cần phải có tất cả cấp cao hơn

chứng chỉ được xác minh lại. Để biết thêm thông tin về chứng chỉ tin cậy, hãy xem *Oracle Database Advanced Security Administrator's Guide*.

■ **Cơ quan cấp chứng chỉ OracleAS.** Đây là một thành phần của cơ sở hạ tầng Quản lý danh tính Oracle, cung cấp giải pháp tích hợp để cung cấp chứng chỉ X.509 phiên bản 3 cho cá nhân, ứng dụng và máy chủ yêu cầu chứng chỉ cho các hoạt động dựa trên PKI như xác thực, SSL, S / MIME và Sớm. Để biết thêm thông tin về Cơ quan cấp chứng chỉ OracleAS, xem Hướng dẫn của quản trị viên cấp chứng chỉ ứng dụng Oracle.

■ **Oracle Wallet Manager.**  Một ví Oracle là một cấu trúc dữ liệu chứa khóa riêng của người dùng, chứng chỉ người dùng và tập hợp các điểm tin cậy của người dùng (cơ quan cấp chứng chỉ tin cậy). Xem Hướng dẫn của quản trị viên bảo mật nâng cao cơ sở dữ liệu Oracle để biết thông tin về quản lý các ví Oracle.

Bạn có thể sử dụng Oracle Wallet Manager để quản lý các ví Oracle. Đây là một ứng dụng Java độc lập được sử dụng để quản lý và chỉnh sửa thông tin xác thực bảo mật trong ví Oracle. Nó thực hiện các hoạt động sau:

* Tạo cặp khóa công khai và tạo yêu cầu chứng chỉ để gửi tới tổ chức phát hành chứng chỉ và tạo wallet
* Cài đặt chứng chỉ cho thực thể
* Quản lý chứng chỉ X.509 phiên bản 3 trên máy khách và máy chủ của Oracle Database
* Định cấu hình chứng chỉ đáng tin cậy cho thực thể
* Mở ví để cho phép truy cập vào các dịch vụ dựa trên PKI

■ **Chứng chỉ X.509 phiên bản 3 thu được từ (và được ký bởi) một thực thể đáng tin cậy, một cơ quan cấp chứng chỉ.** Bởi vì tổ chức phát hành chứng chỉ được tin cậy, các chứng chỉ này xác minh rằng thông tin của tổ chức yêu cầu là chính xác và khóa công khai trên chứng chỉ thuộc về pháp nhân được xác định. Chứng chỉ được nạp vào một ví Oracle để cho phép xác thực trong tương lai.

### Định cấu hình xác thực người dùng toàn cầu và ủy quyền

Bạn có thể sử dụng Oracle Advanced Security để tập trung quản lý thông tin liên quan đến người dùng, bao gồm các ủy quyền, trong một dịch vụ thư mục dựa trên LDAP. Điều này cho phép người dùng và quản trị viên được xác định trong cơ sở dữ liệu với tư cách là người dùng toàn cầu, nghĩa là họ được xác thực bằng SSL và việc quản lý những người dùng này được xử lý bên ngoài cơ sở dữ liệu bằng dịch vụ thư mục tập trung. Vai trò toàn cầu được định nghĩa trong cơ sở dữ liệu và chỉ được biết đến với cơ sở dữ liệu đó, nhưng dịch vụ thư mục xử lý ủy quyền cho vai trò toàn cầu.

**Chú ý:** Bạn cũng có thể có người dùng được chứng thực bởi SSL, người có thẩm quyền không được quản lý trong một thư mục, nghĩa là chúng chỉ có vai trò cơ sở dữ liệu cục bộ. Xem Hướng dẫn của quản trị viên bảo mật nâng cao cơ sở dữ liệu Oracle để biết chi tiết.

Quản lý tập trung này cho phép tạo ra người dùng doanh nghiệp và vai trò doanh nghiệp. Người dùng doanh nghiệp được xác định và quản lý trong thư mục. Chúng có các định danh duy nhất trên toàn doanh nghiệp và có thể được gán vai trò doanh nghiệp xác định đặc quyền truy cập của chúng trên nhiều cơ sở dữ liệu. Vai trò doanh nghiệp bao gồm một hoặc nhiều vai trò toàn cầu và có thể được coi là vùng chứa cho vai trò toàn cầu.

**Xem thêm:** ["Strong Authentication and Centralized Management for](#_bookmark318) [Database Administrators"](#_bookmark318) trên trang 3-22 nếu bạn muốn tập trung quản lý truy cập SYSDBA hoặc SYSOPER

#### Tạo một người dùng được ủy quyền bởi một dịch vụ thư mục

Bạn có các tùy chọn sau để chỉ định người dùng được ủy quyền bởi dịch vụ thư mục:

■ [Tạo một người dùng toàn cầu có một lược đồ riêng](#_bookmark383)

■ [Tạo nhiều người dùng doanh nghiệp chia sẻ lược đồ](#_bookmark387)

###### Tạo một người dùng toàn cầu có một lược đồ riêng

Câu lệnh sau đây cho thấy việc tạo một người dùng toàn cầu với một lược đồ riêng, được xác thực bởi SSL và được ủy quyền bởi dịch vụ thư mục doanh nghiệp:

CREATE USER psmith IDENTIFIED GLOBALLY AS 'CN=psmith,OU=division1,O=oracle,C=US';

Chuỗi được cung cấp trong mệnh đề AS cung cấp mã định danh (tên phân biệt hoặc

DN) có ý nghĩa đối với danh bạ doanh nghiệp.

Trong trường hợp này, psmith là một người dùng toàn cầu. Nhưng, bất lợi ở đây là người dùng psmith sau đó phải được tạo trong mọi cơ sở dữ liệu mà anh ta phải truy cập, cộng với thư mục.

###### Tạo nhiều người dùng Enterprise chia sẻ lược đồ

Nhiều người dùng doanh nghiệp có thể chia sẻ một lược đồ trong cơ sở dữ liệu. Những người dùng này được ủy quyền bởi dịch vụ thư mục doanh nghiệp nhưng không sở hữu các lược đồ riêng lẻ cá nhân trong cơ sở dữ liệu. Những người dùng này không được tạo riêng trong cơ sở dữ liệu. Chúng kết nối với một lược đồ được chia sẻ trong cơ sở dữ liệu.

Để tạo một người dùng không có lược đồ:

1. Tạo một lược đồ được chia sẻ trong cơ sở dữ liệu bằng cách sử dụng ví dụ sau:

CREATE USER appschema IDENTIFIED GLOBALLY AS '';

1. Trong thư mục, tạo nhiều người dùng doanh nghiệp và một đối tượng lập bản đồ.

Đối tượng ánh xạ cho cơ sở dữ liệu biết cách bạn muốn ánh xạ các DN cho người dùng đến giản đồ được chia sẻ. Bạn có thể tạo một ánh xạ DN đầy đủ (một mục nhập thư mục cho mỗi DN riêng biệt), hoặc bạn có thể ánh xạ, cho mỗi người dùng, nhiều thành phần DN vào một lược đồ. Ví dụ:

OU=division,O=Oracle,C=US

**Xem thêm:** *Oracle Database Enterprise User Security Administrator's Guide* for an explanation of these mappings

Hầu hết người dùng không cần lược đồ riêng của họ và triển khai người dùng độc lập với lược đồ phân tách người dùng khỏi cơ sở dữ liệu. Bạn tạo nhiều người dùng chia sẻ cùng một lược đồ trong cơ sở dữ liệu và với tư cách là người dùng doanh nghiệp, họ cũng có thể truy cập vào các lược đồ được chia sẻ trong các cơ sở dữ liệu khác.

#### Ưu điểm của Xác thực toàn cầu và Ủy quyền toàn cầu

Một số ưu điểm của việc xác thực và ủy quyền người dùng toàn cầu như sau:

■ Cung cấp xác thực mạnh bằng SSL, Kerberos hoặc xác thực gốc Windows.

■ Cho phép quản lý tập trung người dùng và đặc quyền trên toàn doanh nghiệp.

■ Dễ quản trị: Bạn không phải tạo lược đồ cho mọi người dùng trong mọi cơ sở dữ liệu trong doanh nghiệp.

■ Tạo điều kiện cho đăng nhập một lần: Người dùng cần đăng nhập một lần để chỉ truy cập nhiều cơ sở dữ liệu và dịch vụ. Hơn nữa, người dùng sử dụng mật khẩu có thể có một mật khẩu duy nhất để truy cập nhiều cơ sở dữ liệu chấp nhận người dùng doanh nghiệp đã được xác thực mật khẩu.

■ Vì xác thực và ủy quyền người dùng toàn cầu cung cấp quyền truy cập dựa trên mật khẩu, bạn có thể di chuyển người dùng cơ sở dữ liệu đã được xác thực mật khẩu đã xác định trước đó vào thư mục (sử dụng Tiện ích di chuyển người dùng) để được quản lý tập trung. Điều này làm cho xác thực và ủy quyền toàn cầu có sẵn cho các máy khách phát hành Cơ sở dữ liệu Oracle trước đó vẫn được hỗ trợ.

■ Các liên kết cơ sở dữ liệu CURRENT\_USER kết nối với tư cách là người dùng toàn cầu. Người dùng cục bộ có thể kết nối với tư cách là người dùng toàn cục trong ngữ cảnh của một thủ tục được lưu trữ, nghĩa là, không lưu trữ mật khẩu người dùng chung trong định nghĩa liên kết.

**Xem thêm:** Các hướng dẫn sau để biết thêm thông tin về xác thực và ủy quyền toàn cầu và người dùng và vai trò doanh nghiệp:

■ *Oracle Database Advanced Security Administrator's Guide*

■ *Oracle Database Enterprise User Security Administrator's Guide*

### Định cấu hình dịch vụ bên ngoài để xác thực người dùng và mật khẩu

Nội dung:

■ [Về xác thực bên ngoài](#_bookmark394)

■ [Ưu điểm của xác thực bên ngoài](#_bookmark397)

■ [Tạo người dùng được xác thực bên ngoài](#_bookmark399)

■ [Xác thực đăng nhập người dùng bằng hệ điều hành](#_bookmark401)

■ [Đăng nhập người dùng xác thực bằng cách sử dụng xác thực mạng](#_bookmark405)

#### Giới thiệu về xác thực bên ngoài

Khi bạn sử dụng xác thực bên ngoài cho tài khoản người dùng, Cơ sở dữ liệu Oracle duy trì tài khoản người dùng, nhưng một dịch vụ bên ngoài thực hiện quản trị mật khẩu và xác thực người dùng. Dịch vụ bên ngoài này có thể là hệ điều hành hoặc dịch vụ mạng, chẳng hạn như Oracle Net.

Với xác thực bên ngoài, cơ sở dữ liệu của bạn dựa vào hệ điều hành cơ bản hoặc dịch vụ xác thực mạng để hạn chế quyền truy cập vào tài khoản cơ sở dữ liệu. Mật khẩu cơ sở dữ liệu không được sử dụng cho loại đăng nhập này. Nếu hệ điều hành hoặc dịch vụ mạng của bạn cho phép, thì nó có thể xác thực người dùng trước khi họ có thể đăng nhập vào cơ sở dữ liệu. Để bật tính năng này, hãy đặt tham số khởi tạo OS\_AUTHENT\_PREFIX và sử dụng tiền tố này trong tên người dùng Cơ sở dữ liệu Oracle. Tham số OS\_AUTHENT\_PREFIX định nghĩa một tiền tố mà Oracle Database thêm vào đầu tên tài khoản hệ điều hành của mỗi người dùng. Cơ sở dữ liệu Oracle so sánh tên người dùng tiền tố với tên người dùng cơ sở dữ liệu Oracle trong cơ sở dữ liệu khi người dùng cố gắng kết nối.

Bạn nên đặt OS\_AUTHENT\_PREFIX thành chuỗi rỗng (một tập hợp dấu ngoặc kép trống: ""). Sử dụng một chuỗi rỗng loại bỏ việc bổ sung bất kỳ tiền tố nào vào tên tài khoản hệ điều hành, để tên người dùng cơ sở dữ liệu Oracle khớp chính xác với tên người dùng của hệ điều hành.

OS\_AUTHENT\_PREFIX=" "

Sau khi bạn đặt OS\_AUTHENT\_PREFIX, nó sẽ vẫn giữ nguyên cho cuộc sống của cơ sở dữ liệu. Nếu bạn thay đổi tiền tố, thì bất kỳ tên người dùng cơ sở dữ liệu nào bao gồm tiền tố cũ không thể được sử dụng để thiết lập kết nối, trừ khi bạn thay đổi tên người dùng để sử dụng nó để xác thực mật khẩu.

Giá trị mặc định của tham số này là OPS $ cho khả năng tương thích ngược với các phiên bản trước của Cơ sở dữ liệu Oracle. Ví dụ: giả sử rằng bạn đặt OS\_AUTHENT\_PREFIX như sau:

OS\_AUTHENT\_PREFIX=OPS$

**Chú ý:** Văn bản của tham số khởi tạo OS\_AUTHENT\_PREFIX phân biệt chữ hoa chữ thường trên một số hệ điều hành. Xem tài liệu Oracle Database dành riêng cho hệ điều hành của bạn để biết thêm thông tin về tham số khởi tạo này.

Nếu người dùng có tài khoản hệ điều hành có tên tsmith là kết nối với một cơ sở dữ liệu Oracle và được xác thực bởi hệ điều hành, thì Oracle Database kiểm tra rằng có một người dùng cơ sở dữ liệu tương ứng là OPS $ tsmith và, nếu có, cho phép người dùng kết nối. Tất cả các tham chiếu đến một người dùng được xác thực bởi hệ điều hành phải bao gồm tiền tố, OPS $, như đã thấy trong OPS $ tsmith.

#### Ưu điểm của xác thực bên ngoài

Ưu điểm của xác thực bên ngoài như sau:

■ Có nhiều lựa chọn hơn về cơ chế xác thực, chẳng hạn như thẻ thông minh, dấu vân tay, Kerberos hoặc hệ điều hành.

■ Nhiều dịch vụ xác thực mạng, chẳng hạn như Kerberos hỗ trợ đăng nhập một lần, cho phép người dùng có ít mật khẩu hơn để nhớ.

■ Nếu bạn đã sử dụng cơ chế bên ngoài để xác thực, chẳng hạn như một trong những cơ chế được liệt kê trước đó, thì có thể có ít chi phí quản trị hơn để sử dụng cơ chế đó với cơ sở dữ liệu.

#### Tạo người dùng được xác thực bên ngoài

Câu lệnh sau tạo một người dùng được xác định bởi Cơ sở dữ liệu Oracle và được xác thực bởi hệ điều hành hoặc dịch vụ mạng. Ví dụ này giả định rằng tham số OS\_AUTHENT\_PREFIX đã được đặt thành một khoảng trống ("").

CREATE USER psmith IDENTIFIED EXTERNALLY;

Sử dụng câu lệnh CREATE USER ... IDENTIFIED EXTERNALLY, bạn tạo các tài khoản cơ sở dữ liệu phải được xác thực bởi hệ điều hành hoặc dịch vụ mạng.

Cơ sở dữ liệu Oracle sau đó dựa vào xác thực đăng nhập bên ngoài này khi nó cung cấp cho người dùng hệ điều hành cụ thể có quyền truy cập vào tài nguyên cơ sở dữ liệu của một người dùng cụ thể.

**Xem thêm:** *Oracle Database Advanced Security Administrator's Guide*

để biết thêm thông tin về xác thực bên ngoài

#### Xác thực đăng nhập người dùng bằng cách sử dụng hệ điều hành

Theo mặc định, Cơ sở dữ liệu Oracle chỉ cho phép các thông tin đăng nhập được xác thực của hệ điều hành trên các kết nối an toàn, ngăn cản việc sử dụng Oracle Net và cấu hình máy chủ được chia sẻ. Hạn chế này ngăn cản người dùng từ xa mạo danh người dùng hệ điều hành khác qua kết nối mạng.

Đặt tham số REMOTE\_OS\_AUTHENT thành TRUE trong tệp tham số khởi tạo cơ sở dữ liệu sẽ buộc cơ sở dữ liệu chấp nhận tên người dùng hệ điều hành máy khách nhận được qua kết nối không an toàn và sử dụng nó để truy cập tài khoản. Bởi vì khách hàng, nói chung, chẳng hạn như máy tính, không đáng tin cậy để thực hiện xác thực hệ điều hành đúng cách, thực hành bảo mật rất kém để bật tính năng này.

Cài đặt mặc định, REMOTE\_OS\_AUTHENT = FALSE, tạo cấu hình an toàn hơn, thực thi xác thực dựa trên máy chủ, phù hợp của các máy khách kết nối với cơ sở dữ liệu Oracle.

Mọi thay đổi đối với tham số này sẽ có hiệu lực vào lần sau khi bạn khởi động cá thể và gắn kết cơ sở dữ liệu. Nói chung, xác thực người dùng thông qua hệ điều hành máy chủ cung cấp kết nối nhanh hơn và thuận tiện hơn cho Cơ sở dữ liệu Oracle mà không chỉ định tên người dùng hoặc mật khẩu cơ sở dữ liệu riêng biệt. Ngoài ra, các mục nhập của người dùng tương ứng trong cơ sở dữ liệu và các đường dẫn kiểm toán của hệ điều hành.

Lưu ý rằng tham số REMOTE\_OS\_AUTHENT không được chấp nhận trong Cơ sở dữ liệu Oracle 11g Phiên bản 1 (11.1) và chỉ được giữ lại cho khả năng tương thích ngược.

#### Đăng nhập người dùng xác thực bằng cách sử dụng xác thực mạng

Oracle Advanced Security thực hiện xác thực mạng, bạn có thể cấu hình để sử dụng dịch vụ của bên thứ ba như Kerberos. Nếu bạn đang sử dụng Oracle Advanced Security làm dịch vụ xác thực bên ngoài duy nhất của bạn, thì cài đặt tham số REMOTE\_OS\_AUTHENT không liên quan, bởi vì Oracle Advanced Security chỉ cho phép các kết nối an toàn.

### Sử dụng Xác thực và ủy quyền nhiều người dùng

Trong môi trường đa nhiệm, Oracle Database kiểm soát sự bảo mật của các ứng dụng tầng giữa bằng cách giới hạn các đặc quyền của chúng, bảo quản danh tính khách hàng thông qua tất cả các tầng và các hành động kiểm toán thay mặt cho các máy khách. Trong các ứng dụng sử dụng tầng giữa rất bận rộn, chẳng hạn như màn hình xử lý giao dịch, danh tính của các máy khách kết nối với tầng giữa phải được giữ nguyên. Một lợi thế của việc sử dụng một tầng giữa là kết nối tổng hợp, cho phép nhiều người dùng truy cập vào một máy chủ dữ liệu mà không cần mỗi người trong số họ cần một kết nối riêng biệt. Trong những môi trường như vậy, bạn cần có khả năng thiết lập và ngắt kết nối rất nhanh.

Đối với các môi trường này, bạn có thể sử dụng Giao diện cuộc gọi Oracle để tạo các phiên nhẹ, cho phép xác thực mật khẩu cơ sở dữ liệu cho từng người dùng. Phương pháp này bảo tồn danh tính của người dùng thực thông qua tầng giữa mà không phải trả phí kết nối cơ sở dữ liệu riêng cho từng người dùng.

Bạn có thể tạo các phiên nhẹ có hoặc không có mật khẩu. Tuy nhiên, nếu một tầng giữa ở bên ngoài hoặc trên tường lửa, thì bảo mật sẽ tốt hơn khi mỗi phiên hạng nhẹ có mật khẩu riêng của nó. Đối với máy chủ ứng dụng nội bộ, các phiên nhẹ không có mật khẩu có thể phù hợp.

#### Quản trị và bảo mật trong máy khách, máy chủ ứng dụng và máy chủ cơ sở dữ liệu

Trong môi trường đa nhiệm, một máy chủ ứng dụng cung cấp dữ liệu cho các máy khách và phục vụ như một giao diện từ chúng đến một hoặc nhiều máy chủ cơ sở dữ liệu. Máy chủ ứng dụng có thể xác thực thông tin đăng nhập của máy khách, chẳng hạn như trình duyệt Web và máy chủ cơ sở dữ liệu có thể kiểm tra các hoạt động được thực hiện bởi máy chủ ứng dụng. Các hoạt động kiểm tra này bao gồm các hành động được thực hiện bởi máy chủ ứng dụng thay mặt cho các máy khách, chẳng hạn như các yêu cầu thông tin được hiển thị trên máy khách. Yêu cầu kết nối với máy chủ cơ sở dữ liệu là một ví dụ về thao tác máy chủ ứng dụng không liên quan đến một máy khách cụ thể.

Việc xác thực trong môi trường đa nhiệm dựa trên các vùng tin cậy. Xác thực ứng dụng khách là miền của máy chủ ứng dụng. Bản thân máy chủ ứng dụng được xác thực bởi máy chủ cơ sở dữ liệu. Các hoạt động sau được thực hiện:

■ Người dùng cuối cung cấp bằng chứng xác thực cho máy chủ ứng dụng, thông thường, bằng cách sử dụng mật khẩu hoặc chứng chỉ X.509.

■ Máy chủ ứng dụng xác thực người dùng cuối và sau đó xác thực chính nó với máy chủ cơ sở dữ liệu.

■ Máy chủ cơ sở dữ liệu xác thực máy chủ ứng dụng, xác minh rằng người dùng cuối tồn tại và xác minh rằng máy chủ ứng dụng có đặc quyền để kết nối cho người dùng cuối.

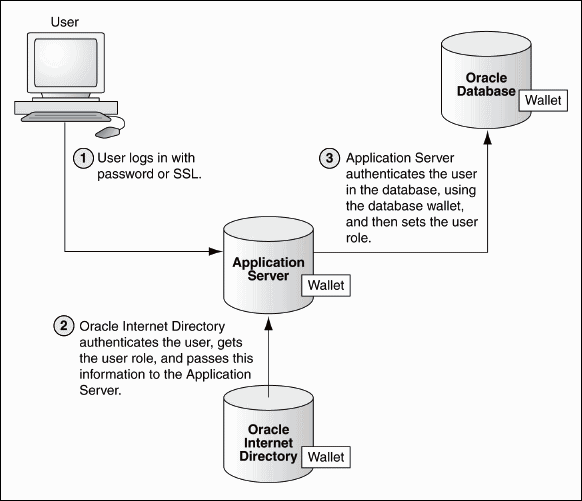
Máy chủ ứng dụng cũng có thể bật vai trò cho người dùng cuối thay mặt họ kết nối với ai. Máy chủ ứng dụng có thể có được các vai trò này từ một thư mục, phục vụ như một kho lưu trữ ủy quyền. Máy chủ ứng dụng chỉ có thể yêu cầu các vai trò này được kích hoạt. Cơ sở dữ liệu xác minh các yêu cầu sau:

■ Rằng khách hàng có các vai trò này bằng cách kiểm tra kho lưu trữ vai trò nội bộ của nó

■ Máy chủ ứng dụng có đặc quyền để kết nối thay mặt cho người dùng và do đó sử dụng các vai trò này khi người dùng có thể

[Hình 3–2](#_bookmark412)  cho thấy một ví dụ về xác thực đa điểm.

***Hình 3–2 Xác thực Multitier***



Các hành động sau đây diễn ra:

1. Người dùng đăng nhập bằng mật khẩu hoặc Lớp cổng bảo mật. Thông tin xác thực được chuyển qua Máy chủ ứng dụng Oracle.
2. Oracle Internet Directory xác thực người dùng, nhận vai trò liên kết với người dùng đó từ ví, và sau đó chuyển thông tin này về máy chủ ứng dụng Oracle.
3. Máy chủ ứng dụng Oracle kiểm tra danh tính của người dùng trong Cơ sở dữ liệu Oracle, trong đó có một ví lưu trữ thông tin này, và sau đó đặt vai trò cho người dùng đó.

Bảo mật cho các ứng dụng cấp trung bình phải giải quyết các vấn đề chính sau:

■**Trách nhiệm giải trình.** Máy chủ cơ sở dữ liệu phải có khả năng phân biệt giữa các hành động của ứng dụng và các hành động mà một ứng dụng thay mặt cho một ứng dụng khách. Nó phải có khả năng kiểm tra cả hai loại hành động.

■ **Đặc quyền tối thiểu.** Người dùng và tầng giữa nên được cấp các đặc quyền ít nhất cần thiết để thực hiện hành động của họ, để giảm nguy cơ vô tình hoặc các hoạt động trái phép nguy hiểm.

### Bảo tồn danh tính người dùng trong môi trường đa nhiệm

Nhiều tổ chức muốn biết ai là người dùng thông qua tất cả các tầng của một ứng dụng mà không phải hy sinh lợi ích của tầng giữa. Cơ sở dữ liệu Oracle hỗ trợ các cách sau đây để bảo vệ danh tính người dùng thông qua tầng giữa của một ứng dụng:

■ [Sử dụng Máy chủ Cấp Trung để Xác thực](#_bookmark415)

■ [Sử dụng mã định danh khách hàng để xác định người dùng ứng dụng không được biết đến cơ sở dữ liệu](#_bookmark466)

**Xem thêm:** ["Auditing SQL Statements and Privileges in a Multitier](#_bookmark1799) [Environment" trên trang 9-27](#_bookmark1799)

#### Sử dụng máy chủ cấp trung để xác thực proxy

Các phần sau giải thích cách sử dụng xác thực proxy:

[■ Giới thiệu về xác thực proxy](#_bookmark417)

[■ Ưu điểm của xác thực proxy](#_bookmark417)

[■ Ai có thể tạo tài khoản người dùng proxy?](#_bookmark417)

[■ Tạo tài khoản người dùng Proxy và cho phép người dùng kết nối thông qua họ](#_bookmark417)

[■ Sử dụng xác thực proxy với Cửa hàng mật khẩu an toàn bên ngoài](#_bookmark417)

[■ Truyền thông qua nhận dạng của người dùng thực bằng cách sử dụng xác thực proxy](#_bookmark417)

[■ Hạn chế đặc quyền của tầng giữa](#_bookmark417)

[■ Cấp quyền cho Cấp trung gian để Ủy quyền và Xác thực Người dùng](#_bookmark417)

[■ Ủy quyền một Cấp trung gian để ủy quyền cho một người dùng được xác thực bởi các phương tiện khác](#_bookmark417)

[■ Đánh giá lại người dùng thông qua tầng giữa tới cơ sở dữ liệu](#_bookmark417)

###### Giới thiệu về xác thực proxy

Cơ sở dữ liệu Oracle cung cấp xác thực proxy trong Oracle Call Interface (OCI), JDBC / OCI hoặc JDBC Thin Driver cho người dùng cơ sở dữ liệu hoặc người dùng doanh nghiệp. Doanh nghiệp

người dùng là những người được quản lý trong Oracle Internet Directory và người truy cập vào một lược đồ được chia sẻ trong cơ sở dữ liệu.

Bạn có thể thiết kế một máy chủ trung cấp để xác thực khách hàng một cách an toàn bằng cách sử dụng ba hình thức xác thực proxy sau:

■ Máy chủ tầng trung xác thực chính nó với máy chủ cơ sở dữ liệu và máy khách, trong trường hợp này là một người dùng ứng dụng hoặc một ứng dụng khác, tự xác thực với máy chủ tầng giữa. Danh tính khách hàng có thể được duy trì tất cả các cách thức thông qua cơ sở dữ liệu.

■ Khách hàng, trong trường hợp này là người dùng cơ sở dữ liệu, không được máy chủ trung cấp xác thực. Danh tính khách hàng và mật khẩu cơ sở dữ liệu được truyền qua máy chủ tầng giữa đến máy chủ cơ sở dữ liệu để xác thực.

■ Khách hàng, trong trường hợp này là một người dùng toàn cầu, được xác thực bởi máy chủ tầng giữa và chuyển một trong những điều sau thông qua tầng giữa để truy xuất tên người dùng của khách hàng.

**–** Distinguished name (DN)

**–** Certificate

**Chú ý:** Việc sử dụng chứng chỉ để xác thực proxy có thể không được hỗ trợ trong bản phát hành Cơ sở dữ liệu Oracle trong tương lai.

Trong mọi trường hợp, quản trị viên phải ủy quyền cho máy chủ tầng giữa hành động thay mặt cho khách hàng.

**Xem thêm:** *Oracle Call Interface Programmer's Guide* and *Oracle Database Advanced Application Developer's Guide* or details about designing a middle-tier server to proxy users

###### Ưu điểm của xác thực proxy

Trong môi trường đa nhiệm, xác thực proxy kiểm soát tính bảo mật của các ứng dụng tầng trung lưu bằng cách giữ lại danh tính và đặc quyền của khách hàng thông qua tất cả các tầng và bằng các hành động kiểm toán thay mặt cho khách hàng. Ví dụ, tính năng này cho phép nhận dạng người dùng bằng ứng dụng Web (hoạt động như một proxy) được chuyển qua ứng dụng tới máy chủ cơ sở dữ liệu.

Hệ thống ba cấp cung cấp những lợi ích sau đây cho các tổ chức:

■ Các tổ chức có thể tách biệt logic ứng dụng khỏi lưu trữ dữ liệu, phân vùng cũ trong các máy chủ ứng dụng và sau đó trong cơ sở dữ liệu.

■ Máy chủ ứng dụng và máy chủ Web cho phép người dùng truy cập dữ liệu được lưu trữ trong cơ sở dữ liệu.

■ Người dùng thích sử dụng giao diện trình duyệt quen thuộc, dễ sử dụng.

■ Các tổ chức cũng có thể hạ thấp chi phí tính toán của mình bằng cách thay thế nhiều máy khách dày với nhiều máy khách mỏng và một máy chủ ứng dụng.

Ngoài ra, xác thực proxy cơ sở dữ liệu Oracle cung cấp các lợi ích bảo mật sau:

■ Một mô hình tin cậy hạn chế, bằng cách kiểm soát người dùng trên các tầng giữa có thể kết nối và vai trò mà các tầng giữa có thể giả định cho người dùng

■ Khả năng mở rộng, bằng cách hỗ trợ các phiên người dùng thông qua trình điều khiển Mỏng OCI, JDBC / OCI hoặc JDBC và loại bỏ phí tổn của việc xác thực lại khách hàng

■ Trách nhiệm, bằng cách bảo quản danh tính của người dùng thực thông qua cơ sở dữ liệu và cho phép kiểm tra các hành động được thực hiện thay mặt cho người dùng thực

■ Tính linh hoạt, bằng cách hỗ trợ các môi trường mà người dùng được biết đến cơ sở dữ liệu và trong đó người dùng chỉ đơn thuần là người dùng ứng dụng mà cơ sở dữ liệu không có nhận thức

**Chú ý:** Cơ sở dữ liệu Oracle hỗ trợ chức năng xác thực proxy này chỉ trong ba tầng. Nó không hỗ trợ nó trên nhiều tầng giữa.

###### Ai có thể tạo tài khoản người dùng proxy?

Để tạo tài khoản người dùng proxy, người dùng phải có các đặc quyền tối thiểu sau:

■ Đặc quyền hệ thống CREATE USER để tạo một tài khoản người dùng cơ sở dữ liệu sẽ được sử dụng như một tài khoản người dùng proxy

■ Vai trò DV\_ACCTMGR nếu Oracle Database Vault được bật, để tạo tài khoản người dùng proxy

■ Khả năng cấp đặc quyền hệ thống CREATE SESSION cho tài khoản người dùng proxy

■ Đặc quyền hệ thống NGƯỜI SỬ DỤNG ALTER để kích hoạt tài khoản người dùng hiện có để kết nối với cơ sở dữ liệu thông qua tài khoản proxy

Làm theo các nguyên tắc này khi bạn tạo tài khoản người dùng proxy:

■ Để bảo mật tốt hơn và tuân thủ nguyên tắc đặc quyền tối thiểu, chỉ cấp cho tài khoản người dùng proxy đặc quyền CREATE SESSION. Không cấp cho người dùng này bất kỳ đặc quyền nào khác. Tài khoản người dùng proxy được thiết kế để chỉ cho phép người dùng khác kết nối bằng tài khoản proxy. Bất kỳ đặc quyền nào phải được thực hiện trong quá trình kết nối phải thuộc về người dùng kết nối, không phải là tài khoản proxy.

■ Như với tất cả các mật khẩu, hãy đảm bảo rằng mật khẩu bạn tạo cho người dùng proxy rất mạnh và không dễ đoán. Hãy nhớ rằng nhiều người dùng sẽ kết nối với tư cách là người dùng proxy, do đó, điều đặc biệt quan trọng là mật khẩu này phải mạnh mẽ. Xem ["Guidelines for Securing Passwords" trên trang 10-7](#_bookmark2167)  để được tư vấn về cách tạo mật khẩu mạnh.

■ Xem xét sử dụng các tính năng kết nối mạng tùy chọn bảo mật nâng cao, để ngăn chặn nghe trộm mạng.

■ Để tinh chỉnh thêm lượng điều khiển mà người dùng kết nối có, hãy xem xét hạn chế các vai trò được người dùng kết nối sử dụng khi họ được kết nối thông qua tài khoản proxy. Câu lệnh ALTER USER cho phép bạn định cấu hình người dùng kết nối bằng cách sử dụng các vai trò được chỉ định, bất kỳ vai trò nào ngoại trừ vai trò được chỉ định hoặc hoàn toàn không có vai trò nào.

###### Tạo tài khoản người dùng Proxy và ủy quyền người dùng kết nối thông qua họ

Câu lệnh CREATE USER cho phép bạn tạo các loại tài khoản người dùng sau, tất cả các tài khoản này có thể được sử dụng làm tài khoản proxy:

■ Tài khoản người dùng cơ sở dữ liệu được xác thực bằng mật khẩu

■ Tài khoản người dùng bên ngoài, được xác thực bởi các nguồn bên ngoài, chẳng hạn như Lớp cổng bảo mật (SSL) hoặc Kerberos

■ Tài khoản người dùng toàn cầu, được xác thực bởi dịch vụ thư mục doanh nghiệp (Oracle Internet Directory).

Để tạo tài khoản người dùng proxy và cho phép người dùng kết nối thông qua tài khoản:

* 1. Sử dụng câu lệnh CREATE USER để tạo tài khoản người dùng proxy. Ví dụ:

CREATE USER appuser IDENTIFIED BY *password*;

* 1. Sử dụng mệnh đề GRANT CONNECT THROUGH của câu lệnh ALTER USER để cho phép người dùng hiện có kết nối thông qua tài khoản người dùng proxy.

Ví dụ:

ALTER USER preston GRANT CONNECT THROUGH appuser;

Giả sử người dùng preston có một số lượng lớn vai trò, nhưng bạn chỉ muốn cô ấy sử dụng một vai trò (ví dụ, appuser\_role) khi cô ấy được kết nối với cơ sở dữ liệu thông qua tài khoản proxy appuser. Bạn có thể sử dụng câu lệnh ALTER USER sau đây:

ALTER USER preston GRANT CONNECT THROUGH appuser WITH ROLE appuser\_role;

Bất kỳ vai trò nào khác mà người dùng preston sẽ không có sẵn cho cô ấy miễn là cô ấy đang kết nối với proxy ứng dụng.

Sau khi bạn hoàn tất các bước này, người dùng preston có thể kết nối bằng cách sử dụng người dùng proxy appuser như sau:

CONNECT appuser[preston]

Enter password: *appuser\_password*

Lưu ý những điều dưới đây:

■ **Người dùng proxy chỉ có thể thực hiện các hoạt động mà người dùng preston có quyền thực hiện.** Hãy nhớ rằng chính người dùng proxy, appuser, chỉ có các đặc quyền tối thiểu (CREATE SESSION).

■ **Sử dụng vai trò với khách hàng ở tầng giữa.** Bạn cũng có thể chỉ định các vai trò mà tầng giữa được phép kích hoạt khi kết nối với tư cách máy khách. Các hoạt động được thực hiện thay mặt cho khách hàng bởi một máy chủ tầng trung có thể được kiểm toán.

■ **Tìm người dùng proxy.** Để tìm những người dùng hiện được ủy quyền kết nối thông qua một tầng giữa, hãy truy vấn chế độ xem từ điển dữ liệu PROXY\_USERS, ví dụ:

SELECT \* FROM PROXY\_USERS;

■ **Xóa kết nối proxy.** Sử dụng mệnh đề REVOKE CONNECT THROUGH của ALTER USER để không cho phép kết nối proxy. Ví dụ: để thu hồi người dùng preston từ kết nối thông qua ứng dụng người dùng proxy, hãy nhập tuyên bố sau:

ALTER USER preston REVOKE CONNECT THROUGH appuser

■ **Hết hạn mật khẩu và kết nối proxy.** Việc sử dụng hết hạn cấp mật khẩu ở cấp trung bình không áp dụng cho các tài khoản được xác thực thông qua proxy. Thay vào đó, hãy khóa tài khoản thay vì hết hạn mật khẩu.

**Xem thêm:**

* *Oracle Database SQL Language Reference* for detailed information about the CREATE USER statement
* *Oracle Database SQL Language Reference* for detailed information about the ALTER USER statement
* *Oracle Database Enterprise User Security Administrator's Guide* for information about managing proxy users in an enterprise user environment
* *Oracle Database Advanced Security Administrator's Guide* for more information about using network encryption and strong authentication
* ["Auditing SQL Statements and Privileges in a Multitier Environment"](#_bookmark1799) trên trang 9-27 để biết chi tiết về các hoạt động kiểm toán được thực hiện thay mặt cho người dùng bởi một tầng giữa

###### Sử dụng xác thực proxy với Cửa hàng mật khẩu an toàn bên ngoài

Nếu bạn lo ngại về mật khẩu được sử dụng trong xác thực proxy được người dùng độc hại thu thập, thì bạn có thể sử dụng kho lưu trữ mật khẩu bên ngoài an toàn với xác thực proxy để lưu trữ thông tin xác thực mật khẩu trong ví. Kết nối với cơ sở dữ liệu Oracle bằng cách sử dụng xác thực proxy và lưu trữ mật khẩu an toàn bên ngoài là lý tưởng cho các tình huống như chạy tập tin batch. Khi người dùng proxy kết nối với cơ sở dữ liệu và xác thực bằng mật khẩu bên ngoài bảo mật, mật khẩu không được hiển thị trong trường hợp người dùng độc hại cố lấy mật khẩu.

Để sử dụng xác thực proxy với cửa hàng mật khẩu an toàn bên ngoài:

1. Định cấu hình tài khoản xác thực proxy, như được hiển thị trong quy trình trong ["Creating Proxy User Accounts and Authorizing Users to Connect Through Them" trên trang 3-38](#_bookmark430).
2. Định cấu hình cửa hàng mật khẩu an toàn bên ngoài. See ["Configuring Clients to Use the External Password Store"](#_bookmark307) trên trang 3-18 để biết thêm thông tin.

Sau đó, người dùng có thể kết nối bằng proxy nhưng không phải chỉ định mật khẩu. Ví dụ:

sqlplus [preston]/@*db\_alias*

Khi bạn sử dụng cửa hàng mật khẩu an toàn bên ngoài, người dùng đăng nhập không cần phải cung cấp tên người dùng và mật khẩu. Chỉ giá trị SERVICE\_NAME (nghĩa là, db\_alias) từ tệp tnsnames.ora phải được chỉ định.

###### Thông qua nhận dạng của người dùng thực bằng cách sử dụng xác thực proxy

Đối với người dùng doanh nghiệp hoặc người dùng cơ sở dữ liệu, trình điều khiển Oracle Call Interface, JDBC / OCI hoặc Thin cho phép một tầng trung gian thiết lập một số phiên người dùng trong một kết nối cơ sở dữ liệu, mỗi người xác định duy nhất một người dùng được kết nối (kết nối tổng hợp). Các phiên làm giảm chi phí mạng của việc tạo các kết nối mạng riêng biệt từ tầng giữa tới cơ sở dữ liệu.

Nếu bạn muốn xác thực từ các máy khách thông qua một tầng giữa tới cơ sở dữ liệu, chuỗi xác thực đầy đủ từ máy khách đến tầng giữa tới cơ sở dữ liệu xảy ra như sau:

1. Khách hàng xác thực với tầng giữa, sử dụng bất kỳ hình thức xác thực nào mà tầng giữa sẽ chấp nhận. Ví dụ: khách hàng có thể xác thực với

middle tier by using a user name and password or an X.509 certificate by means of SSL.

1. The middle tier authenticates itself to the database by using whatever form of authentication the database accepts. This could be a password or an authentication mechanism supported by Oracle Advanced Security, such as a [**Kerberos ticket**](#_bookmark2344) or an X.509 certificate (SSL).
2. The middle tier then creates one or more sessions for users using OCI, JDBC/OCI, or Thin driver.
   * If the user is a database user, then the session must, as a minimum, include the database user name. If the database requires it, then the session can include a password (which the database verifies against the password store in the database). The session can also include a list of database roles for the user.
   * If the user is an enterprise user, then the session may provide different information depending on how the user is authenticated.

**Example 1:** If the user authenticates to the middle tier using SSL, then the middle tier can provide the DN from the X.509 certificate of the user, or the certificate itself in the session. The database uses the DN to look up the user in Oracle Internet Directory.

**Example 2:** If the user is a password-authenticated enterprise user, then the middle tier must provide, as a minimum, a globally unique name for the user. The database uses this name to look up the user in Oracle Internet Directory. If the session also provides a password for the user, then the database will verify the password against Oracle Internet Directory. User roles are automatically retrieved from Oracle Internet Directory after the session is established.

* + The middle tier may optionally provide a list of database roles for the client. These roles are enabled if the proxy is authorized to use the roles on behalf of the client.

1. The database verifies that the middle tier has the privilege to create sessions on behalf of the user.

The OCISessionBegin call fails if the application server cannot perform a proxy authentication on behalf of the client by the administrator, or if the application server is not allowed to activate the specified roles.

###### Limiting the Privilege of the Middle Tier

Least privilege is the principle that users should have the fewest privileges necessary to perform their duties and no more. As applied to middle tier applications, this means that the middle tier should not have more privileges than it needs. Oracle Database enables you to limit the middle tier such that it can connect only on behalf of certain database users, using only specific database roles. You can limit the privilege of the middle tier to connect on behalf of an enterprise user, stored in an LDAP directory, by granting to the middle tier the privilege to connect as the mapped database user. For instance, if the enterprise user is mapped to the APPUSER schema, then you must at least grant to the middle tier the ability to connect on behalf of APPUSER. Otherwise, attempts to create a session for the enterprise user will fail.

However, you cannot limit the ability of the middle tier to connect on behalf of enterprise users. For example, suppose that user Sarah wants to connect to the database through a middle tier, appsrv (which is also a database user). Sarah has multiple roles, but it is desirable to restrict the middle tier to use only the clerk role on her behalf.

An administrator could effectively grant permission for appsrv to initiate connections on behalf of Sarah using her clerk role only, using the following syntax:

ALTER USER sarah GRANT CONNECT THROUGH appsrv WITH ROLE clerk;

By default, the middle tier cannot create connections for any client. The permission must be granted for each user.

To allow appsrv to use all of the roles granted to the client Sarah, the following statement would be used:

ALTER USER sarah GRANT CONNECT THROUGH appsrv;

Each time a middle tier initiates an OCI, JDBC/OCI, or Thin driver session for another database user, the database verifies that the middle tier is authorized to connect for that user by using the role specified.

**Note:** Instead of using default roles, create your own roles and assign only necessary privileges to them. Creating your own roles enables you to control the privileges granted by them and protects you if Oracle Database changes or removes default roles. For example, the CONNECT role now has only the CREATE SESSION privilege, the one most directly needed when connecting to a database.

However, CONNECT formerly provided several additional privileges, often not needed or appropriate for most users. Extra privileges can endanger the security of your database and applications. These have now been removed from CONNECT.

See [Chapter 4, "Configuring Privilege and Role Authorization"](#_bookmark493) for more information about roles.

###### Authorizing a Middle Tier to Proxy and Authenticate a User

The following statement authorizes the middle-tier server appserve to connect as user bill. It uses the WITH ROLE clause to specify that appserve activate all roles associated with bill, except payroll.

ALTER USER bill

GRANT CONNECT THROUGH appserve WITH ROLE ALL EXCEPT payroll;

To revoke the middle-tier server (appserve) authorization to connect as user bill, the following statement is used:

ALTER USER bill REVOKE CONNECT THROUGH appserve;

###### Authorizing a Middle Tier to Proxy a User Authenticated by Other Means

Use the AUTHENTICATION REQURED clause of the ALTER USER ... GRANT CONNECT

THROUGH statement to authorize a user to be proxied, but not authenticated, by a middle tier. Currently, PASSWORD is the only means supported.

The following statement illustrates this form of authentication:

ALTER USER mary

GRANT CONNECT THROUGH midtier AUTHENTICATION REQUIRED;

In the preceding statement, middle-tier server midtier is authorized to connect as user mary, and midtier must also pass the user password to the database server for authorization.

###### Reauthenticating the User Through the Middle Tier to the Database

Administrators can specify that authentication is required by using the AUTHENTICATION REQUIRED proxy clause with the ALTER USER SQL statement. In this case, the middle tier must provide user authentication credentials.

For example, suppose that user Sarah wants to connect to the database through a middle tier, appsrv. An administrator could require that appsrv provides authentication credentials for Sarah by using the following syntax:

ALTER USER *sarah* GRANT CONNECT THROUGH *appsrv* AUTHENTICATION REQUIRED;

The AUTHENTICATION REQUIRED clause ensures that authentication credentials for the user must be presented when the user is authenticated through the specified proxy.

**Note:** For backward compatibility, if you use the AUTHENTICATED USING PASSWORD proxy clause, then Oracle Database transforms it to AUTHENTICATION REQUIRED.

**Using Password-Based Proxy Authentication** When you use password-based proxy authentication, Oracle Database passes the password of the client to the middle-tier server. The middle-tier server then passes the password as an attribute to the data server for verification. The main advantage to this is that the client computer does not have to have Oracle software installed on it to perform database operations.

To pass the password of the client, the middle-tier server calls the OCIAttrSet()

function as follows, passing OCI\_ATTR\_PASSWORD as the type of the attribute being set.

OCIAttrSet(

session\_handle, /\* Pointer to a handle whose attribute gets modified. \*/ OCI\_HTYPE\_SESSION, /\* Handle type: OCI user session handle. \*/ password\_ptr, /\* Pointer to the value of the password attribute. \*/ 0, /\* The size of the password attribute value is already

known by the OCI library. \*/ OCI\_ATTR\_PASSWORD, /\* The attribute type. \*/

error\_handle); /\* An error handle used to retrieve diagnostic

information in the event of an error. \*/

**Using Proxy Authentication with Enterprise Users** If the middle tier connects to the database as a client who is an enterprise user, then either the distinguished name, or the X.509 certificate containing the distinguished name is passed over instead of the database user name. If the user is a password-authenticated enterprise user, then the middle tier must provide, as a minimum, a globally unique name for the user. The database uses this name to look up the user in Oracle Internet Directory.

To pass over the distinguished name of the client, the application server would call the Oracle Call Interface method OCIAttrSet() with OCI\_ATTR\_DISTINGUISHED\_NAME as the attribute type, as follows:

OCIAttrSet(session\_handle,

OCI\_HTYPE\_SESSION,

distinguished\_name, 0,

OCI\_ATTR\_DISTINGUISHED\_NAME,

error\_handle);

To pass over the entire certificate, the middle tier would call OCIAttrSet() with OCI\_ ATTR\_CERTIFICATE as the attribute type, as follows.

OCIAttrSet(session\_handle,

OCI\_HTYPE\_SESSION,

certificate, certificate\_length, OCI\_ATTR\_CERTIFICATE,

error\_handle);

If the type is not specified, then the database uses its default certificate type of X.509.

**Note:**

* + OCI\_ATTR\_CERTIFICATE is Distinguished Encoding Rules (DER) encoded.
  + Certificate based proxy authentication using OCI\_ATTR\_ CERTIFICATE will not be supported in future Oracle Database releases. Use the OCI\_ATTR\_DISTINGUISHED\_NAME or OCI\_ATTR\_ USERNAME attribute instead

If you are using proxy authentication for password-authenticated enterprise users, then use the same OCI attributes as for database users authenticated by password (OCI\_ATTR\_USERNAME). Oracle Database first checks the user name against the database. If it finds no user, then the database checks the user name in the directory. This user name must be globally unique.

#### Using Client Identifiers to Identify Application Users Not Known tothe Database

The following sections explain how to use client identifiers:

■ [About Client Identifiers](#_bookmark468)

■ [How Client Identifiers Work in Middle Tier Systems](#_bookmark471)

■ [Using the CLIENT\_IDENTIFIER Attribute to Preserve User Identity](#_bookmark473)

■ [Using CLIENT\_IDENTIFIER Independent of Global Application Context](#_bookmark475)

■ [Using the DBMS\_SESSION PL/SQL Package to Set and Clear the Client Identifier](#_bookmark481)

###### About Client Identifiers

Oracle Database provides the CLIENT\_IDENTIFIER attribute of the built-in USERENV application context namespace for application users. These users are known to an application but unknown to the database. The CLIENT\_IDENTIFIER attribute can capture any value that the application uses for identification or access control, and passes it to the database. The CLIENT\_IDENTIFIER attribute is supported in OCI, JDBC/OCI, or Thin driver.

###### How Client Identifiers Work in Middle Tier Systems

Many applications use session pooling to set up several sessions to be reused by multiple application users. Users authenticate themselves to a middle-tier application, which uses a single identity to log in to the database and maintains all the user connections. In this model, application users are users who are authenticated to the middle tier of an application, but who are not known to the database. You can use a

CLIENT\_IDENTIFIER attribute, which acts like an application user proxy for these types of applications.

In this model, the middle tier passes a client identifier to the database upon the session establishment. The client identifier could actually be anything that represents a client connecting to the middle tier, for example, a cookie or an IP address. The client identifier, representing the application user, is available in user session information and can also be accessed with an application context (by using the USERENV naming context). In this way, applications can set up and reuse sessions, while still being able to keep track of the *application user* in the session. Applications can reset the client identifier and thus reuse the session for a different user, enabling high performance.

###### Using the CLIENT\_IDENTIFIER Attribute to Preserve User Identity

You can use the CLIENT\_IDENTIFIER predefined attribute of the built-in application context namespace, USERENV, to capture the application user name for use with global application context. You also can use the CLIENT\_IDENTIFIER attribute independently. When you use the CLIENT\_IDENTIFIER attribute independently from a global application context, you can set CLIENT\_IDENTIFIER with the DBMS\_SESSION interface. The ability to pass a CLIENT\_IDENTIFIER to the database is supported in Oracle Call Interface (OCI), JDBC/OCI, or Thin driver.

When you use the CLIENT\_IDENTIFIER attribute with global application context, it provides flexibility and high performance for building applications. For example, suppose a Web-based application that provides information to business partners has three types of users: gold partner, silver partner, and bronze partner, representing different levels of information available. Instead of each user having his or her own session set up with individual application contexts, the application could set up global application contexts for gold partners, silver partners, and bronze partners. Then, use the CLIENT\_IDENTIFIER to point the session at the correct context to retrieve the appropriate type of data. The application need only initialize the three global contexts once and use the CLIENT\_IDENTIFIER to access the correct application context to limit data access. This provides performance benefits through session reuse and through accessing global application contexts set up once, instead of having to initialize application contexts for each session individually.

**See Also:**

■ ["Using Global Application Contexts" on page 6-22](#_bookmark1238) for how to implement global application contexts

■ ["Tutorial: Creating a Global Application Context That Uses a Client Session ID" on page 6-35](#_bookmark1300)

###### Using CLIENT\_IDENTIFIER Independent of Global Application Context

Using the CLIENT\_IDENTIFIER attribute is especially useful for those applications in which the users are unknown to the database. In these situations, the application typically connects as a single database user and all actions are taken as that user.

Because all user sessions are created as the same user, this security model makes it difficult to achieve data separation for each user. These applications can use the CLIENT\_IDENTIFIER attribute to preserve the real application user identity through to the database.

With this approach, sessions can be reused by multiple users by changing the value of the CLIENT\_IDENTIFIER attribute, which captures the name of the real application user. This avoids the overhead of setting up a separate session and separate attributes for each user, and enables reuse of sessions by the application. When the CLIENT\_

IDENTIFIER attribute value changes, the change is added to the next OCI, JDBC/OCI, or Thin driver call for additional performance benefits.

For example, the user Daniel connects to a Web Expense application. Daniel is not a database user; he is a typical Web Expense application user. The application accesses the built-in application context namespace and sets DANIEL as the CLIENT\_IDENTIFIER attribute value. Daniel completes his Web Expense form and exits the application.

Then, Ajit connects to the Web Expense application. Instead of setting up a new session for Ajit, the application reuses the session that currently exists for Daniel, by changing the CLIENT\_IDENTIFIER to AJIT. This avoids the overhead of setting up a new connection to the database and the overhead of setting up a global application context. The CLIENT\_IDENTIFIER attribute can be set to any value on which the application bases access control. It does not have to be the application user name.

To set the CLIENT\_IDENTIFIER attribute with OCI, use the OCI\_ATTR\_CLIENT\_ IDENTIFIER attribute in the call to OCIAttrSet(). Then, on the next request to the server, the information is propagated and stored in the server sessions. For example:

OCIAttrSet (session, OCI\_HTYPE\_SESSION,

(dvoid \*) "appuser1", (ub4)strlen("appuser1"), OCI\_ATTR\_CLIENT\_IDENTIFIER,

\*error\_handle);

For applications that use JDBC, be aware that JDBC does not set the client identifier. To set the client identifier in a connection pooling environment, use Dynamic Monitoring Service (DMS) metrics. If DMS is not available, then use the connection.setClientInfo method. For example:

connection.setClientInfo("E2E\_CONTEXT.CLIENT\_IDENTIFIER", "appuser");

**See Also:**

* *Oracle Call Interface Programmer's Guide* about how the OCI\_ ATTR\_CLIENT\_IDENTIFIER user session handle attribute is used in middle-tier applications
* *Oracle Database JDBC Developer's Guide* for more information about configuring client connections using JDBC

###### Using the DBMS\_SESSION PL/SQL Package to Set and Clear the Client Identifier

To use the DBMS\_SESSION package to set and clear the CLIENT\_IDENTIFIER value on the middle tier, use the following interfaces:

■ SET\_IDENTIFIER

■ CLEAR\_IDENTIFIER

The middle tier uses SET\_IDENTIFIER to associate the database session with a particular user or group. Then, the CLIENT\_IDENTIFIER is an attribute of the session and can be viewed in session information.

If you plan to use the DBMS\_SESSION.SET\_IDENTIFIER procedure, be aware that the DBMS\_APPLICATION\_INFO.SET\_CLIENT\_INFO procedure can overwrite the value of the client identifier. Typically, these values should be the same, so if SET\_CLIENT\_INFO is set, its value can be automatically propagated to the value set by SET\_IDENTIFIER if the CLIENTID\_OVERWRITE event is set to ON.

To check the status of the CLIENTID\_OVERWRITE event, log in to SQL\*Plus and then enter the SHOW PARAMETER command. For example, assuming that CLIENTID\_OVERWRITE is enabled:

SHOW PARAMETER EVENT

NAME TYPE VALUE

------------------------------ ------------------ ------------------

event string clientid\_overwrite

To enable the CLIENTID\_OVERWRITE event system-wide, connect to SQL\*Plus as SYS

using the SYSDBA privilege, and then enter the following ALTER SYSTEM statement:

ALTER SYSTEM SET EVENTS 'CLIENTID\_OVERWRITE';

Or, enter the following line in your init.ora file:

event="clientid\_overwrite"

Then restart the database. To disable the CLIENTID\_OVERWRITE event, log in to SQL\*Plus as SYS with the SYSDBA privilege, and then run the following ALTER SYSTEM statement:

ALTER SYSTEM SET EVENTS 'CLIENTID\_OVERWRITE OFF';

If you prefer to change the CLIENTID\_OVERWRITE value for the session only, then use the ALTER SESSION statement.

Afterwards, if you set the client identifier using the DBMS\_APPLICATION\_INFO.SET\_ CLIENT\_INFO procedure, you must then run DBMS\_SESSION.SET\_IDENTIFIER so that the client identifier settings are the same.

**See Also:**

■ ["Using Global Application Contexts" on page 6-22](#_bookmark1238) for information about using client identifiers in a global application context

■ *Oracle Database PL/SQL Packages and Types Reference* for more information about the DBMS\_SESSION package

### Finding Information About User Authentication

[Table 3–3](#_bookmark492) lists data dictionary views that contain information about user authentication. For detailed information about these views, see *Oracle Database Reference*.

***Table 3–3 Data Dictionary Views That Describe User Authentication***

|  |  |
| --- | --- |
| **View** | **Description** |
| DBA\_PROFILES | Displays information about profiles, including their settings and limits. |
| DBA\_ROLES | Displays the kind of authentication used for a database role to log in to the database, such as NONE or GLOBAL (query the AUTHENTICATION\_TYPE column) |

***Table 3–3 (Cont.) Data Dictionary Views That Describe User Authentication***

|  |  |
| --- | --- |
| **View** | **Description** |
| DBA\_USERS | Among other user information, displays the following:   * The kind of authentication the user used to log in to the database, such as PASSWORD or EXTERNAL (AUTHENTICATION\_ TYPE column) * The release in which the user created his or her password (PASSWORD\_VERSIONS column) |
| DBA\_USERS\_WITH\_DEFPWD | Displays whether the user account password is a default password |
| PROXY\_USERS | Displays users who are currently authorized to connect through a middle tier |
| V$DBLINK | Displays user accounts for existing database links (DB\_LINK,  OWNER\_ID columns) |
| V$SESSION | Querying the USERNAME column displays the concurrently logged in users |

# 4

## Configuring Privilege and Role Authorization

This chapter contains:

* [About Privileges and Roles](#_bookmark496)
* [Who Should Be Granted Privileges?](#_bookmark500)
* [Granting the SYSDBA and SYSOPER Administrative Privileges to Users](#_bookmark504)
* [Managing System Privileges](#_bookmark506)
* [Managing User Roles](#_bookmark539)
* [Managing Object Privileges](#_bookmark712)
* [Granting a User Privileges and Roles](#_bookmark791)
* [Revoking Privileges and Roles from a User](#_bookmark822)
* [Granting to and Revoking from the PUBLIC Role](#_bookmark843)
* [Granting Roles Using the Operating System or Network](#_bookmark846)
* [When Do Grants and Revokes Take Effect?](#_bookmark862)
* [Managing Fine-Grained Access in PL/SQL Packages and Types](#_bookmark877)
* [Finding Information About User Privileges and Roles](#_bookmark960)

### About Privileges and Roles

Xác thực bao gồm 2 quá trình chính sau:

* Chỉ cho phép một số người dùng truy cập, xử lý hoặc thay đổi dữ liệu..
* Áp dụng các giới hạn khác nhau về quyền truy cập hoặc các hành đông của người dùng. Các giới hạn này có thể được ràng buộc hơặc gỡ bởi đối với các người dùng có thể áp dụng vào các đối tượng như là các schema, bảng, dòng hoặc các tài nguyên về thời gian như (thời gian cpu, kết nối, nghỉ)

User **Privilege** là quyền để chạy một câu lệnh SQL nào đso, hoặc quyền để truy xuất một đối tượng cái mà thuộc về một người dùng khác, chạy PL/SQL package, và còn hơn nữa. Các loại quyền này được định nghĩa trong cơ sở dữ liệu Oracle.

**Role** được tạo bởi các người dùng ( thường là các nhà quản trị) để nhóm cácprivilege hoặc các role khác. Đó là một cách để cấp nhiều quyền hoặc role cho các người dùng.

Phần này mô tả các thể loại chung sau đây:

* **Các quyền hệ thống.** Các quyền này cho phép ngược được cấp thực hiện các nhiệm vụ của người quản trị tiêu chuẩn trong cơ sở dữ liệu. Chỉ cho phép các hành động này được thực hiện bởi các người dùng đáng tin tưởng. ["Managing System Privileges" trên trang 4-2](#_bookmark506) mô tả các quyền hệ thống chi tiết.

■ **Các vai trò người dùng.** Một **vai trò** nhóm một vài quyền và các vai trò khác, vì vậy chúng có thể được cấp và thu hồi đến các người dùng một cách đồng thời. Bạn phải cho phép role đối với người dùng trước khi người dùng có thể sử dụng nó. Đọc ["Managing User Roles" trên trang 4-6](#_bookmark539) để biết thêm thông tin.

■ **Các quyền đối tượng.** Mỗi loại đối tượng có các quyền được liên kết với nó.. ["Managing Object Privileges"](#_bookmark712) [trên trang 4-23 mô tả cách để quản lý các quyền đối với các loại đối tượng khác nhau.](#_bookmark712)

### Who Should Be Granted Privileges?

Bạn có thể cấp những quyền đến các người dùng vì thế họ có thể hoàn thành các nhiệm vụ được yêu cầu đối với công việc của họ. Bạn có thể cấp chỉ một quyền đến một người dùng người mà yêu cầu quyền đó để hoàn thành các công việc cần thiết. Việc cấp các quyền một cách không cần thiết quá mức có thể làm tổn hại đến sự bảo mật. Ví dụ, Bạn không nên bao giờ cấp quyền quản trị SYSDBA hoặc SYSOPER đến các người dùng, người mà không thực hiện các công việc quản trị.

Một người dùng có thể được cấp quyền trong hai cách:

■ **Bạn có thể cấp các quyền đến các người dùng một cách tường minh.** Ví dụ, bạn có thể cấp quyền một cách tường minh đến psmith quyền để nhập các record vào bảng employees.

■ **Bạn có thể cấp các quyền đến một vai trò ( tên của một nhóm các quyền), và sau đó cấp vai trò đến cho một hoặc nhiều người dùng.** Ví dụ, Bạn có thể cấp các quyền truy vấn, chèn, và xóa các record từ bảng employees đến một vai trò tên là clerk, cái mà bạn có thể cấp cho các người dùng khác psmith and robert.

Bởi vì các vai trò cho phép quản lý dễ và tốt hơn các quyền, bạn nên thường xuyên cấp quyền đến các role và không cấp trực tiếp cho các người dùng.

**Đọc thêm:**

■ ["Guidelines for Securing User Accounts and Privileges" on page 10-2](#_bookmark2131) for best practices to follow when granting privileges

■ *Oracle Database SQL Language Reference* for the complete list of system privileges and their descriptions

### Granting the SYSDBA and SYSOPER Administrative Privileges to Users

Với việc sở tất cả các quyền mạnh, Chỉ có các nhà quả trị SYSDBA và SYSOPER mới có thể gán các quyền đó cho các người dùng mà họ thấy là tin tưởng. Tuy nhiên, có một lưu ý rằng có một sự hạn chế đối với các người dùng mà tên của họ không phải là các kí tự trong bản mã ASCII (ví dụ, âm sắc trong cái tên HÜBER). Bạn có thể cấp quyền quản trị đối với các người dùng này, nhưng nếu cơ sở dữu liệu Oracle instance đang "down" (tắt, không chạy), sự xác thực bằng cách sử dụng các quyền được cấp không được hỗ trợ nếu tên người dùng có các kí tự không nằm trong bảng mã ASCII. Nếu cơ sở dữ liệu hoạt động lại, thì cơ chế xác thực đó sẽ được hỗ trợ.

### Managing SystemPrivileges

Phần này chứa:

■ [About System Privileges](#_bookmark508)

■ [Why Is It Important to Restrict System Privileges?](#_bookmark511)

■ [Granting and Revoking System Privileges](#_bookmark528)

■ [Who Can Grant or Revoke System Privileges?](#_bookmark531)

■ [About ANY Privileges and the PUBLIC Role](#_bookmark533)

#### About SystemPrivileges

**System privilege** có quyền để thực hiện một hành động nào đó hoặc thực hiện hành động trên bất kỳ các đối tượng schema ở bất cứ loại nào. Ví dụ, các quyền dể tạo các tablespace và xóa dòng của bất kỳ bảng nào trong cơ sở dữ liệu đều là các quyền hệ thống..

Có hơn 100 các quyền hệ thống khác nhau. Mỗi quyền hệ thống cho phép một người dùng thực hiện một hành động cơ sở dữ liệu nào đó hoặc phân loại các hành động cơ sở dữ liệu. Hãy nhớ là các quyền hệ thống là rất mạnh. Chỉ cấp các quyền này khi cần thiết cho các role nào đó và các người dùng đáng tin trong cơ sở dữ liệu. Bạn có thể tìm thấy một danh sách hoàn chỉnh các quyền thống và các mô tả của nó trong Oracle Database SQL Language Reference. Để tìm các quyền hệ thống đã được gán cho một người dùng nào đó, bạn có thể truy vấn khung nhìn từ điển dữ liệu DBA\_SYS\_PRIVS.

#### Why Is It Important to Restrict SystemPrivileges?

Bởi vì các quyền hệ thống là cực kỳ mạnh, mặc định thì cơ sở dữ liệu được cấu hình để tránh việc các người dùng thông thường (không phải là quản trị) thực hiện bất kỳ các quyền hệ thống nào (như là quyền UPDATE ANY TABLE) trong từ điển dữ liệu.. Đọc ["Guidelines for Securing User](#_bookmark2131) [Accounts and Privileges" trên trang 10-2](#_bookmark2131) for additional guidelines about restricting system privileges.

■ [Restricting System Privileges by Securing the Data Dictionary](#_bookmark514)

■ [Allowing Access to Objects in the SYS Schema](#_bookmark521)

###### Restricting System Privileges by Securing the Data Dictionary

Để bảo mật từ điển dữ liệu, thiết lập tham số khởi tọa của O7\_DICTIONARY\_ACCESSIBILITY thành FALSE, như là một giá tị mặc định. Tính năng này được gọi là cơ chế bảo về từ điển.

Tham số khởi tạo O7\_DICTIONARY\_ACCESSIBILITY kiểm soát các hạn chế trên các quyền hệ thống khi bạn nâng cấp Oracle Database release 7 thành 8i và các phiên bản sau này. Nếu tham số này được thiết lập thành TRUE, thì các sự truy cập vào đối tượng trong SYS chema được cho phép (Oracle Database release 7 behavior). Bởi vì quyền ANY sẽ được áp vào cho từ điển dữ liệu, một người dùng độc hại với quyền ANY có thể truy cập hoặc chỉnh sửa các bảng trong từ điển dữ liệu.

Để thiết lập tham số khởi tạo O7\_DICTIONARY\_ACCESSIBILTY , chỉnh sửa nó trong file initSID.ora. Một cách thay thế, Bạn có thể đăng nhập vào SQL\*Plus với người dùng SYS with SYSDBA privilege và sau đó nhập câu lệnh ALTER SYSTEM, giả định rằng bạn đã bắt đầu chạy cơ sở dữ liệu và nạp một file server parameter file (SPFILE).

[Ví dụ 4–1](#_bookmark519) cho thấy cách để thiết lập tham số khởi tạo O7\_DICTIONARY\_ACCESSIBILTY thành FALSE bằng cách sử dụng câu lệnh ALTER SYSTEM trong SQL\*Plus.

***Ví dụ 4–1 Thiết lập O7\_DICTIONARY\_ACCESSIBILITY thành FALSE***

ALTER SYSTEM SET O7\_DICTIONARY\_ACCESSIBILITY=FALSE SCOPE=SPFILE;

Khi bạn thiết lập O7\_DICTIONARY\_ACCESSIBILITY thành FALSE, các quyền hệ thống cái quyền mà cho phép truy cập đến các đối tượng trong bất kỳ schema (ví dụ, các người dùng có quyền ANY, như là CREATE ANY PROCEDURE) không cho phép bạn truy cập vào schema SYS. Điều này có nghĩa là việc truy cập vào cấc đối tượng trong SYS schema (các đối tượng từ điển dữ liệu) bị giới hạn tới các người dùng người mà kết nối thông qua quyền SYSDBA. Hãy nhớ rằng người dùng SYS phải đăng nhập với một trong hai quyền SYSDBA hoặc SYSOPER; ngược lại, một lỗi ORA-28099: kết nối bằng SYS nên là SYSDBA hoặc là SYSOPER được tung ra. Nếu bạn thiết lập thành TRUE, thì bạn có thể đăng nhập vào cơ sở dữ liệu như là người dùng SYS mà không cần phải chỉ ra quyền SYSDBA hoặc SYSOPER..

Các quyền hệ thống cái mà cung cấp cách truy cập vào các đối tượng trong các schema khác không cho phép các người dùng khác truy cập vào các đối tượng trong SYS schema. Ví dụ quyền SELECT ANY TABLE cho phép người dùng truy cập các khung nhìn và bảng trong các schema khác, nhưng không cho phép chúng truy vấn bất kỳ đất tượng từ điển dữ liệu nào (các bảng cơ sở của các khung nhìn động, các khung nhìn thông thường, packages, và synoyms). Bạn có thể cấp các quyền cho những người dùng một cách tường mình để truy cập các đối tượng trong SYS schema..

See *Oracle Database Reference* for more information about the O7\_DICTIONARY\_ ACCESSIBILITY initialization parameter.

###### Allowing Access to Objects in the SYS Schema

Mọi người dùng với các quyền đối tượng được gán tường minh hoặc những người kết nối với quyền quản trị (SYSDBA) có thể truy cập các đối tượng trong schema SYS.

[Bảng 4–1](#_bookmark524) Liệt kê các vai trò mà bạn có thể cấp quyền cho người dùng, người mà cần truy cận đến các đối tượng trong schema SYS.

***Table 4–1 Các vai trò được phép truy cập đến các đối tượng trong schema SYS***

|  |  |
| --- | --- |
| **Vai trò** | **Mô tả** |
| SELECT\_CATALOG\_ROLE | Cấp vai trò này để cho phép các người dùng có các quyền truy vấn trên các khung nhìn từ điển dữ liệu. |
| EXECUTE\_CATALOG\_ROLE | Cấp vai trò này để cho phép người dùng có các quyền thực thi đối với các package và các thủ tục trong từ điển dữ liệu. |
| DELETE\_CATALOG\_ROLE | Cấp quyền này để cho phép các người dùng xóa các record khỏi các bảng hệ thống audit SYS.AUD$ và SYS.FGA\_LOG$. |

Ngoài ra, bạn có thể cấp quyền hệ thống SELECT ANY DICTIONARY đến các người dùng mà yêu cầu truy xuất các bảng đã được tạo ra trong schema SYS. Quyền hệ thống này cho phép truy vấn truy cập đến bất kỳ đối tượng nào trong schema SYS, bao gồm các bảng được tạo ra trong schema đó. Nó phải được cấp riêng lẻ đến từng người dùng yêu cầu quyền này.Nó không bao gồm các quyền GRANT ALL PRIVILEGES, nhưng nó có thể được cấp thông qua một vai trò..

**Lưu ý:** Bạn nên cấp những vai trò này và quyền hệ thống SELECT ANY DICTIONARY một cách cực kỳ cẩn thậ, bởi vì sư toàn vẹn của hệ thống của bạn có thể bị tác động bởi các người không biết sử dụng, hoặc sử dụng sai mục đích.

#### Granting and Revoking SystemPrivileges

Bạn có thể cấp hoặc thu hồi các quyền hệ thông trên các người dùng hoặc các role. Nếu bạn cấp quyền hệ thống cho các role, thì bạn có thể sử dụng các role đó để thực thi quyền hệ thống.Ví dụ, Vai trò cho phép các quyền được thực hiện một cách chọn lọc có sẵn. Đảo bảo rằng bạn tuân theo các hướng dẫn về việc phân chia nhiệm vụ được mô trong ["Guidelines for Securing Roles"](#_bookmark2160) [trên trang 10-6](#_bookmark2160).

Sử dụng hai phương pháp sau đây để cấp hoặc thu hồi các quyền hệ thống đối với các nugời dùng và các role:

■ GRANT aCác câu lệnh SQL GRANT và REVOKE

■ Oracle Enterprise Manager Database Control

**See Also:**

■ ["Granting a User Privileges and Roles" on page 4-37](#_bookmark791)

■ ["Revoking Privileges and Roles from a User" on page 4-41](#_bookmark822)

■ ["When Do Grants and Revokes Take Effect?" on page 4-48](#_bookmark862)

■ ["Finding Information About User Privileges and Roles" on page 4-71](#_bookmark960)

■ *Oracle Database 2 Day DBA* for more information about Database Control

#### Who Can Grant or Revoke SystemPrivileges?

Chỉ có hai loại người dùng có thể cấp quyền hệ thống cho các người dùng khác hoặc hặc thu hồi các quyền này khỏi họ:

* Các người dùng, người mà được cấp các quyền hệ thống nào đó với tùy chọn ADMIN OPTION
* Các người dùng có quyền hệ thống GRANT ANY PRIVILEGE

Với lý do này, chỉ nên cung cấp các quyền này đối với các người dùng đáng tin tưởng.

#### About ANY Privileges and the PUBLIC Role

Các quyền hệ thống sử dụng từ khóa ANY cho phép bạn thiết lập các quyền cho toàn bộ danh mục của các đối tượng trong cơ sở dữ liệu. Ví dụ, quyền hệ thống CREATE ANY PROCEDURE cho phép một người dùng tạo ra một thủ tục bất cứ đâu trong cơ sở dữ liệu. Hành vi của các đối tượng được được tạo bởi các người dùng với quyền ANY không bị gới hạn ở schema mà nó được tạo ra. Ví dụ, Nếu một người dùng JSMITH có quyền CREATE ANY PROCEDURE và tạo ra một thủ tục trong schema JONES, thì thủ tục đó sẽ chạy như là JONES. Tuy nhiên, JONES sẽ không nhận biết được là cái thủ tục mà JSMITH đã tạo đang được chạy dưới tên anh ta (JONES). nếu JONS có các quyền DBA, Để cho JSMITH chạy một thủ tục như là JONES có thể xảy ra một vi phạm bảo mật.

Vai trò PUBLIC là một vai trò đặc biệt mà mọi tài khoản cơ sở dữ liệu tự động có khi một tài khoản được tạo ra. Mặc định, Vai trò này không có quyền nào được gán, nhưng nó có nhiều nhiều quyền có thể dược cấp, chủ yếu là các đối tượng java. Bạn không thể xóa vai trò PUBLIC, và gán hoặc thu hồi vai trò này không có ý nghĩa gì hết, bởi vì mọi tài khoản người dùng đều sở hữu role này, nó sẽ không xuất hiện trong các khung nhìn từ diển dữ liệu DBA\_ROLES và SESSION\_ROLES.

Bạn có thể cấp các quyền vào vài trò PUBLIC, nhưng hãy nhớ rằng điều này làm cho các quyền đó khả dụng đối với mọi người dùng cơ sở dữ liệu Oracle. Vì lý do này, cẩn thận về việc cấp các quyền và các quyền hệ thống. Ví dụ, if JSMITH có quyền hệ thống CREATE PUBLIC SYNONUM, anh ta có thể định nghĩa lại một interface cái mà anh ta biết mọi ngoừi khác sẽ dùng, và sau đó point to it với PUBLIC SYNONYM mà anh ta đã tạo.

Thay vì truy cập vào đúng interface, các người dùng sẽ truy cập vào interface của JSMITH, cái mà có thể thực hiện các hoạt động trái phép như là ăn cấp thông tin đăng nhập của các người dùng.

Các kiểu quyền này là rất mạnh và có thể làm lộ một nguy cơ bảo mật nếu nó được trao nhầm người. Cẩn thận về việc cấp các quyền về ANY hoặc PUBLIC. Cũng như là tất cả các quyền, bạn nên biết có chế "least privilege" khi cấp quyền đến các người dùng.

Dể bảo vệ từ điển dữ liệu (nội dung của SYS schema) khỏi các nggười dùng, người mà có một hoặc nhiều quyền hệ thống ANY thiết lập tham số khởi tạo O7\_DICTIONARY\_ACCESSIBILITY thành FALSE. Bạn có thể thiết lập tham số này bằng cách sử dụng câu lệnh ALTER SYSTEM (Đọc [Example 4–1, "Setting O7\_DICTIONARY\_](#_bookmark519) [ACCESSIBILITY to FALSE" trên trang 4-3](#_bookmark519)) hoặc bằng cách sửa file initSID.ora. Đọc ["Guidelines for Securing a Database Installation and Configuration" trên trang 10-12](#_bookmark2220) for additional guidelines.

### Managing User Roles

Phần này chứa:

■ [About User Roles](#_bookmark542)

■ [Predefined Roles in an Oracle Database Installation](#_bookmark582)

■ [Creating a Role](#_bookmark643)

■ [Specifying the Type of Role Authorization](#_bookmark656)

■ [Dropping Roles](#_bookmark693)

■ [Restricting SQL\*Plus Users from Using Database Roles](#_bookmark697)

■ [Securing Role Privileges by Using Secure Application Roles](#_bookmark707)

#### About User Roles

Quản lý và kiểm soát các quyền thì dễ hơn khi bạn làm điều tương tự với các vai trò, cái mà được đặt tên để nhóm các quyền có liên quan mà bạn có thể cấp như là một nhóm các người dùng và các role khác. Bên trong một cơ sở dữ liệu, mỗi tên vai trò phải duy nhất, khác nhau với tất cả các tên người dùng và các tên vai trò khác. Không giống như các đối tượng schema, vai trò không được chứa bất kỳ schema nào. Do đó, một người dùng, người mà tạo một vai trò có thể bị xóa đi mà không bị ảnh hưởng trên vai trò đó.

Phần này chứa:

■ [The Functionality of Roles](#_bookmark545)

■ [Properties of Roles and Why They Are Advantageous](#_bookmark553)

■ [Common Uses of Roles](#_bookmark559)

■ [How Roles Affect the Scope of a User’s Privileges](#_bookmark566)

■ [How Roles Work in PL/SQL Blocks](#_bookmark569)

■ [How Roles Aid or Restrict DDL Usage](#_bookmark575)

■ [How Operating Systems Can Aid Roles](#_bookmark579)

■ [How Roles Work in a Distributed Environment](#_bookmark581)

###### The Functionality of Roles

Các vai trò là rất hữu dụng cho việc cấp quyền một cách nhanh chóng và dễ dàng đến các người dùng. Mặc dù bạn có thể sử dụng các vai trò đã được định nghĩa trong cơ sở dữ liệu Oracle. Bạn có nhiều quyền kiểm soát và tính liên tục nếu bạn tạo các vai trò của riêng bạn cái mà chỉ chứa các quyền liên quan đến các yêu cầu của bạn. Oracle có thể thay đổi hoặc xóa các quyền nằm trong các vai trò đã được Oracle định nghĩa. Như là với vai trò CONNECT, cái mà hiện giờ chỉ có quyền CREATE SESSION. Trước đó, vai trò CONNECT có tới tám queyèn khác nữa.

Các vai trò có các chức năng sau đây:

■ Một vai trò có thể được cấp các quyền hệ thống hoặc quyền đối tượng.

■ Bất kỳ vai trò nào cũng có thể được cấp đến bất kỳ cơ sở dữ liệu người dùng.

■ Mỗi vai trò cấp cho một người dùng,ở thời điểm đó, có thể được bật hoặc được tắt.Miền bảo mật của người dùng bao gồm các quyền của các vai trò hiện tại lúc đó được bật cho người dùng và không bao gồm các quyền của bất kỳ vai trò hiện đang được tắt đối với người dùng đó. Cơ sở dữ liệu Oracle cho phép các ứng dụng cơ sở dữ liệu và các người dùng bật hoặc tắt các vai trò này để cung cấp các quyềnn khả dụng mà có sự chọn lọc.

■ Một vai trò có thể được cấp cho các vai trò khác. Tuy nhiên, một vai trò không thể được cấp đến chính nó và không được cấp vòng (circularly). Ví dụ, vai trò role1 không thể được cấp cho vai trò role2 nếu vai trò role2 trước đó đã được cấp từ vai trò role1.

■ Nếu một vai trò không được xác thực bằng mật khẩu hoặc không phải là một vai trò bảo mật ứng dụng, thì bạn có thể cấp vai trò đó gián tiếp cho người dùng. Việc cấp vai trò gián tiếp là một vai trò được cấp cho người dùng thông qua một vai trò khác cái mà đã được gán cho người dùng này. Ví dụ, giả sử bạn cấp cho người dùng vai trò role1. Thì bạn cấp vai trò role2 và role3 vào role1. Các vai trò role2 và role3 hiện đang trong role1. Điều này có nghĩa psmith đã được cấp gián tiếp cho các vai trò role2 và role3. Ngoài việc cấp trực tiếp vai trò role1 , việc bật vai trò role1 cho người dùng psmith cho phép các vai trò không gián tiếp role2 và role3 cho người dùng này.

■ Tùy chọn, bạn có thể gán trực tiếp một vai trò cho một vai trò mặc định. Bạn bật hoặc tắt trạng thái của vai trò mặc định của một vai trò được gán trực tiếp bằng cách sử dụng mệnh đề DEAFULT ROLE của câu lệnh ALTER USER. Đảm bảo rằng mệnh đề DEFAULT ROLE chỉ đề cập đến các vai trò mà đã được gán trực tiếp cho người dùng. Để tìm các vai trò đã được gán cho người dùng, truy vấn khung nhìn từ điển dữ liệu DBA\_ROLE\_PRIVS. Khung nhìn này không bao gồm các vai trò được gán cho người dùng một cách gián tiếp. Để tìm các vai trò cái mà được gán cho các vai trò khác, truy vấn khung nhìn ROLE\_ROLE\_PRIVS.

■ Nếu vai trò được xác thực bằng mật khẩu hoặc một vai trò bảo mật ứng dụng, thì bạn không thể cấp nó một cách gián tiếp cho người dùng, cũng như bạn cũng không thể để nó là vai trò mặc định. Bạn chỉ có thể cấp kiểu vai trò này một cách trực tiếp cho người dùng. Thông thường, Bạn bật xác thực mật khẩu hoặc các vai trò bảo mật ứng dụng bằng cách sử dụng câu lệnh SET ROLE.

###### Properties of Roles and Why They Are Advantageous

[Bảng 4–2](#_bookmark556) mô tả các thuộc tính của các vai trò cái mà cho phép quản lý các quyền dễ hơn bên trong một cơ sở dữ liệu.

***Table 4–2 Các thuộc tính của các vai trò và mô tả của chúng***

|  |  |
| --- | --- |
| **Property** | **Description** |
| Reduced privilege administration | Thay vì cấp một tập hợp các quyền giống nhau một cách tường minh cho một vài người dùng, bạn có thể cấp các quyền này như là một nhóm các người dùng liên quan như là một vai trò, và sau đó chỉ có vai trò được cấp cho các thành viên trong một nhóm. |
| Dynamic privilege management | Nếu các quyền của một nhóm cần phải thay đổi. thì chỉ cần thay đổi các quyền của vai trò đó. Các miền bảo mật của tất cả người dùng được cấp vai trò của nhóm tự động phản ánh các thay đổi được thực hiện cho vai trò. |
| Selective availability of privileges | Bạn có thể bật hoặc tắt một cách tự chọn các vai trò được cấp cho một người dùng. Điều này cho phép kiểm soát cụ thể đặc quyền của người dùng trong bất kỳ tình huống cụ thể nào |
| Application awareness | Từ điển dữ liệu ghi chép các vai trò đã tồn tại, vì vậy bạn có thể thiết kế các ứng dụng để truy vấn từ điển và bật hoặc tắt một cách tự động các vai trò mà bạn có thể chọn lựa khi một người dùng cố để thực thi ứng dụng bằng cách đưa ra tên người dùng. |

***Table 4–2 (Cont.) Properties of Roles and Their Description***

|  |  |
| --- | --- |
| **Property** | **Description** |
| Application-specific security | Bạn có thể bảo vệ vai trò sử dụng bằng mật khẩu. Các ứng dụng có thể được tạo ra đặc biệt để kích hoạt một vai trò khi cung cấp mật khẩu chính xác. Người dùng không thể bật vai trò nếu họ không biết mật khẩu. |

Các nhà quản trị cơ sở dữ liệu thường tạo ra các vai trò cho các ứng dụng cơ sở dữ liệu. Bạn nên cấp một vai trò bảo mật ứng dụng các quyền cần thiết để chạy ứng dụng. Bạn sau đó có thể cấp các vai trò bảo mật ứng dụng này đến các vai trò hoặc các người dùng. Một ứng dụng có thể có các vai trò khác nhau, mỗi cái được cấp với tập hợp các quyền khác nhau cái mà cho phép nhiều hoặc ít việc truy cập dữ liệu trong quá tình sử dụng ứng dụng.

DBA có thể tạo một vai trò với mật khẩu để ngăn chặn việc sử dụng trái phép các đặc quyền được cấp cho vai trò đó. Thông thường, một ứng dụng được thiết kế vì thế khi nó hoạt động, nó bật các vai trò phù hợp. Kết quả là , một ứng dụng người dùng không cần biết mật khẩu cho vai trò ứng dụng đó.

**Đọc thêm:** ["How Roles Aid or Restrict DDL Usage" trên trang 4-9](#_bookmark575) for information about restrictions for procedures

###### Common Uses of Roles

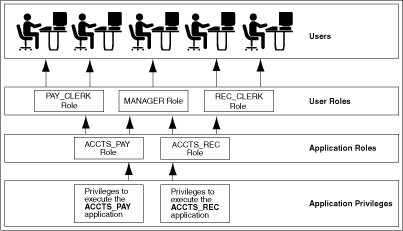
Nhìn nhung, bạn có thể tạo một vai trò để phục vụ một trong hai mục đích sau:

■ Để quản lý các quyền đối với cơ sở dữ liệu ứng dụng (see ["Common Uses of Application Roles" on page 4-8](#_bookmark562))

■ Để quản lý các quyền dối với một nhóm người dùng (see ["Common Uses of User Roles" on page 4-9](#_bookmark564))

[Hình 4–1](#_bookmark561) and the sections that follow describe the two uses of roles.

***Hình 4–1 Common Uses for Roles***



**Common Uses of Application Roles**Cấp một vai trò ứng dụng với mọi quyền cần thiết để chạy một cơ sở dữ liệu ứng dụng cho trước. Sau đó, Cấp một vai trò bảo mật ứng dụng cho các vai trò khác hoặc cho các người dùng cụ thể. Một ứng dụng có thể có vài vai trò khác nhau, với mỗi vai trò được gán một tập hợp các quyềnkhác nhau mà cho phép truy cập dữ liệu ít hay nhiều trong khi sử dụng ứng dụng.

**Common Uses of User Roles**tạo một vai trò ứng người dùng cho một nhóm các người dùng cơ sở dữ liệu với các quyền giống nhau về các yêu cầu. Bạn có thể quản lý các quyền người dùng bằng cách cấp các vai trò bảo mật và các quyền cho vai trò người dùng sau đó cấp vai trò người dùng đó cho các người dùng ứng dụng.

###### How Roles Affect the Scope of a User’s Privileges

Mỗi vai trò và người dùng đều có miền bảo mật duy nhất của riêng mình. Miền bảo mật của vai trò bao gồm các đặc quyền được cấp cho vai trò cộng với các đặc quyền được cấp cho bất kỳ vai trò nào được cấp cho vai trò đó.

Miền bảo mật của một người dùng bao gồm các quyền trên tất cả các đối tượng schema trong schema tương ứng của họ, các quyền được cấp cho người dùng, và các quyền của các vai trò cấp cho người dùng đó mà hiện tại đang được bật. ( Một vai trò có thể đông thời bật cho mội người dùng và tắt cho người khác). Miền này cũng bao gồm các quyền và các vai trò đã được cấp cho vai trò PUBLIC. Vai trò PUBLIC được cấp cho tất cả mọi người dùng trong cơ sở dữ liệu.

###### How Roles Work in PL/SQL Blocks

Việc sử dụng các vai trò trong khối lệnh PL/SQL phụ thuộc vào nó là khối lệnh vô danh hay là khối lệnh đặt tên ( thủ tục lưu trú, hàm, trigger), và nó có thực thi với quyền của người định nghĩa hay là quyền của người gọi khối lệnh đó.

**Roles Used in Named Blocks with Definer's Rights**Mọi vai trò đều bị vô hiệu hóa trong bất kỳ khối lệnh được đặt tên PL/SQL (stored proc,func,trigger) mà nó thực thi với quyền của người định nghĩa.Các vai trò không được sử dụng cho việc kiểm tra quyền và bạn không thể thiết lập các vai trò bên trong thủ tục quyền hạn của người định nghĩa.

Khung nhìn SESSION\_ROLES cho biết tất cả các vai trò hiện đang được cho phép. Nếu một khối lệnh được đặt tên PL/SQL thực thi truy vấn SESSION\_ROLES với quyền của người định nghĩa, thì câu truy vấn không trả về dòng nào cả.

**See Also:** *Oracle Database Reference*

**Roles Used in Named Blocks with Invoker's Rights and Anonymous PL/SQL Blocks**Các khối lệnh được đặt tên PL/SQL cái mà thực thi với quyền của người gọi khối lệnh đó và khối lệnh vô danh PL/SQL được thực thi dựa trên quyền được cấp thông qua các vai trò đang được cho phép. Các vai trò hiện tại được sử dụng để kiểm tra quyền nằm bên trong khối lệnh QL/SQL với quyền của người gọi khối lệnh đó.Bạn có thể sử dụng dynamic SQL để thiết lập một vai trò trong phiên làm việc.

**See Also:**

■ *Oracle Database PL/SQL Language Reference* for an explanation of how invoker's and definer's rights can be used for name resolution and privilege checking

■ *Oracle Database PL/SQL Language Reference* for information about dynamic SQL in PL/SQL

###### How Roles Aid or Restrict DDL Usage

Một người dùng yêu cầu một hoặc nhiều quyền để thực thi thành công một câu lệnh DDL, phụ thuộc vào câu lệnh. Ví dụ, để tạo bảng, người dùng phải có quyền hệ thống CREATE TABLE hoặc CREATE ANY TABLE. Để tạo khung nhìn của một bảng cái mà thuộc về người dùng khác, người tạo bảng yêu cầu quyền hệ thống CREATE VIEW hoặc CREATE ANY VIEW cũng như quyền đối tượng SELECT cho bảng hoặc quyền hệ thống SELECT ANY TABLE.

Cơ sở dữ liệu Oracle tránh sự phụ thuộc vào các đặc quyền nhận được bằng cách vai trò bằng cách hạn chế việc sử dụng các đặc quyền cụ thể trong một số câu lệnh DDL nhất định. Các quy tắc sau mô tả các hạn chế đặc quyền này liên quan đến các câu lệnh DDL:

■ Các quyền hêj thống và quyền đối tượng cho phép người dùng thực hiện thao tác DDL khi nhận được thông qua một vai trò. Ví dụ:

* **System privileges:** Các quyền CREATE TABLE, CREATE VIEW, and CREATE PROCEDURE
* **Object privileges:** Các quyền ALTER and INDEX đối với bảng

Bạn không thể sử dụng quyền đối tượng REFERENCES cho một bảng để địng nghĩa khóa ngoại của một bảng nếu quyền đó nhận được thông qua một vai trò.

■ Tất cả các quyền hệ thống và quyền đối tượng cho phép người dùng thực hiện các thao tác DML mà được yêu cầu để cung cấp câu lệnh DDL không sử dụng được khi nhận được thông qua một vai trò. Miền bảo mật không chứa các vai trò khi câu lệnh CREATE VIEW được sử dụng. Ví dụ, một người dùng, người mà dược cấp quyền hệ thống SELECT ANY TABLE hoặc quyền đối tượng SELECT cho bảng thông qua một vai trò không thể sử dụng cả hai quyền này để tạo một khung nhìn trên bảng mà nó thuộc về người khác. Đây là bởi vì các khung nhìn là quyền của người tạo ra nó, vì vậy khi tạo chúng bạn không thể sử dụng bất kỳ quyền nào ( quyền hệ thống hay quyền đối tượng) được cấp thong qua một vai trò. Nếu quyền này được cấp trực tiếp cho bạn, thì bạn có thể sử dụng chúng. Tuy nhiên, nếu các quyền này bị thu hồi ở thời điểm sau đó, thì định nghĩa khung nhìn trở nên không hợp lệ ("chứa lỗi") và phải biên dịch lại trước khi nó có thể sử dụng lại như cũ..

Ví dụ dưới dây làm rõ việc cho phép và cấm sử dụng các quyền thông qua các vai trò..

Giả định người dùng như sau:

■ Được cấp một vai trò cái mà có quyền hệ tohóng CREATE VIEW

■ Được gán trực tiếp một vai trò cái mà có quyền đối tượng SELECT đối với bảng employees

■ Được gán trực tiếp quyền dói tượng SELECT đối với bảng departments Given these directly and indirectly granted privileges:

■ Người dùng có thể chạy câu lệnh SELECT trên cả hai bảng employees và departments.

■ Mặc dù người dùng có cả hai quyền CREATE VIEW và SELECT đối với bảng employees thông qua một vai trò, người dùng không thể tạo một khung nhìn trên bảng employees, bởi vì quyền đối tượng SELECT đói với bảng employees được cấp thông qua một vai trò.

■ Người dùng có thể tạo khung nhìn trên bảng departments, bởi vì người dùng có quyền CREATE VIEW thông qua một vai trò và quyền SELECT đối với bảng departments được gán trưc tiếp.

###### How Operating Systems Can Aid Roles

Trong một số môi trường, bạn có thể quản trị bảo mật cơ sở dữ liệu thông qua hệ điều hành. Hệ điều hành có thể được sử dụng để cấp và thu hồi các vai trò cơ sở dữ liệu và quản lý xác thực mật khẩu. Khả năng này không khả dụng trên mọi hệ điều hành.

**See Also:** Your operating system-specific Oracle Database documentation for details about managing roles through the operating system

###### How Roles Work in a Distributed Environment

When you use roles in a distributed database environment, ensure that all needed roles are set as the default roles for a distributed (remote) session. These roles cannot be enabled when the user connects to a remote database from within a local database session. For example, the user cannot execute a remote procedure that attempts to enable a role at the remote site.

**See Also:** *Oracle Database Heterogeneous Connectivity User's Guide*

#### Predefined Roles in an Oracle Database Installation

Oracle Database provides a set of predefined roles to help in database administration. These roles, listed in [Table 4–3](#_bookmark586), are automatically defined for Oracle databases when you run the standard scripts that are part of database creation. If you install other options or products, then other predefined roles may be created.

***Table 4–3 Orac******le Database Predefined Roles***

|  |  |
| --- | --- |
| **Predefined Role** | **Description** |
| ADM\_PARALLEL\_EXECUTE\_TASK | Provides privileges to update table data in parallel by using the DBMS\_ PARALLEL\_EXECUTE PL/SQL package.  **See Also:** *Oracle Database PL/SQL Packages and Types Reference* for more information about the DBMS\_PARALLEL\_EXECUTE PL/SQL package. |
| AQ\_ADMINISTRATOR\_ROLE | Provides privileges to administer Advanced Queuing. Includes ENQUEUE ANY QUEUE, DEQUEUE ANY QUEUE, and MANAGE ANY QUEUE, SELECT privileges  on Advanced Queuing tables and EXECUTE privileges on Advanced Queuing packages. |
| AQ\_USER\_ROLE | Obsolete, but kept mainly for release 8.0 compatibility. Provides EXECUTE  privileges on the DBMS\_AQ and DBMS\_AQIN packages. |
| AUTHENTICATEDUSER | Used by the XDB protocols to define any user who has logged in to the system. |
| CAPI\_USER\_ROLE | Provides access to packages used for implementing Information Lifecycle Management (ILM) and hierarchical storage and other applications.  **See Also:***Oracle Database SecureFiles and Large Objects Developer's Guide* |
| CONNECT | Provides the CREATE SESSION system privilege.  This role is provided for compatibility with previous releases of Oracle Database. You can determine the privileges encompassed by this role by querying the DBA\_SYS\_PRIVS data dictionary view.  **Note:** Oracle recommends that you design your own roles for database security rather than relying on this role. This role may not be created automatically by future releases of Oracle Database.  **See Also:** *Oracle Database Reference* for a description of the DBA\_SYS\_PRIVS  view |
| CSW\_USR\_ROLE | Provides user privileges to manage the Catalog Services for the Web (CSW) component of Oracle Spatial.  **See Also:** *Oracle Spatial Developer's Guide* for more information |

***Table 4–3 (Cont.) Oracle Database Predefined Roles***

|  |  |
| --- | --- |
| **Predefined Role** | **Description** |
| CTXAPP | Provides privileges to create Oracle Text indexes and index preferences, and to use PL/SQL packages. This role should be granted to Oracle Text users.  **See Also:** *Oracle Text Application Developer's Guide* for more information |
| CWM\_USER | Provides privileges to manage Common Warehouse Metadata (CWM), which is a repository standard used by Oracle data warehousing and decision support.  **See Also:** *Oracle Database Data Warehousing Guide* for more information |
| DATAPUMP\_EXP\_FULL\_DATABASE | Provides privileges to export data from an Oracle database using Oracle Data Pump.  **Caution:** This is a very powerful role because it provides a user access to any data in any schema in the database. Use caution when granting this role to users.  **See Also:** *Oracle Database Utilities* for more information |
| DATAPUMP\_IMP\_FULL\_DATABASE | Provides privileges to import data into an Oracle database using Oracle Data Pump.  **Caution:** This is a very powerful role because it provides a user access to any data in any schema in the database. Use caution when granting this role to users.  **See Also:** *Oracle Database Utilities* for more information |
| DBA | Provides all system privileges that were created with the ADMIN option.  This role is provided for compatibility with previous releases of Oracle Database. You can determine the privileges encompassed by this role by querying the DBA\_SYS\_PRIVS data dictionary view.  **Note:** Oracle recommends that you design your own roles for database security rather than relying on this role. This role may not be created automatically by future releases of Oracle Database.  **See Also:** *Oracle Database Reference* for a description of the DBA\_SYS\_PRIVS  view |
| DBFS\_ROLE | Provides access to the DBFS (the Database Filesystem) packages and objects.  **See Also:** *Oracle Database SecureFiles and Large Objects Developer's Guide* |
| DELETE\_CATALOG\_ROLE | Provides the DELETE privilege on the system audit table (AUD$). |
| EJBCLIENT | Provides privileges to connect to EJBs from a Java stored procedure. |
| EXECUTE\_CATALOG\_ROLE | Provides EXECUTE privileges on objects in the data dictionary. |
| EXP\_FULL\_DATABASE | Provides the privileges required to perform full and incremental database exports using the Export utility (later replaced with Oracle Data Pump). It includes these privileges: SELECT ANY TABLE, BACKUP ANY TABLE, EXECUTE ANY PROCEDURE, EXECUTE ANY TYPE, ADMINISTER RESOURCE MANAGER, and INSERT, DELETE, and UPDATE on the tables SYS.INCVID, SYS.INCFIL, and  SYS.INCEXP. Also the following roles: EXECUTE\_CATALOG\_ROLE and SELECT\_ CATALOG\_ROLE.  This role is provided for convenience in using the export and import utilities.  **Caution:** This is a very powerful role because it provides a user access to any data in any schema in the database. Use caution when granting this role to users.  **See Also:** *Oracle Database Utilities* for more information |

***Table 4–3 (Cont.) Oracle Database Predefined Roles***

|  |  |
| --- | --- |
| **Predefined Role** | **Description** |
| GATHER\_SYSTEM\_STATISTICS | Provides privileges to update system statistics, which are collected using the  DBMS\_STATS.GATHER\_SYSTEM\_STATISTICS procedure  **See Also:** *Oracle Database Performance Tuning Guide* for more information about managing optimizer statistics |
| GLOBAL\_AQ\_USER\_ROLE | Provides privileges to establish a connection to an LDAP server, for use with Oracle Streams AQ.  **See Also:** *Oracle Streams Advanced Queuing User's Guide* for more information |
| HS\_ADMIN\_EXECUTE\_ROLE | Provides the EXECUTE privilege for users who want to use the Heterogeneous Services (HS) PL/SQL packages.  **See Also:** *Oracle Database Heterogeneous Connectivity User's Guide* for more information |
| HS\_ADMIN\_ROLE | Provides privileges to both use the Heterogeneous Services (HS) PL/SQL packages and query the HS-related data dictionary views.  **See Also:** *Oracle Database Heterogeneous Connectivity User's Guide* for more information |
| HS\_ADMIN\_SELECT\_ROLE | Provides privileges to query the Heterogeneous Services data dictionary views.  **See Also:** *Oracle Database Heterogeneous Connectivity User's Guide* for more information |
| IMP\_FULL\_DATABASE | Provides the privileges required to perform full database imports using the Import utility (later replaced with Oracle Data Pump). Includes an extensive list of system privileges (use view DBA\_SYS\_PRIVS to view privileges) and the following roles: EXECUTE\_CATALOG\_ROLE and SELECT\_CATALOG\_ROLE.  This role is provided for convenience in using the export and import utilities.  **Caution:** This is a very powerful role because it provides a user access to any data in any schema in the database. Use caution when granting this role to users.s.  **See Also:** *Oracle Database Utilities* for more information |
| JAVADEBUGPRIV | Provides privileges to run the Oracle Database Java applications debugger.  **See Also:** *Oracle Database Java Developer's Guide* for more information about managing security for Oracle Java applications |
| JAVAIDPRIV | Deprecated for this release. |
| JAVASYSPRIV | Provides major permissions to use Java2, including updating Oracle JVM-protected packages.  **See Also:** *Oracle Database Java Developer's Guide* for more information about managing security for Oracle Java applications |
| JAVAUSERPRIV | Provides limited permissions to use Java2.  **See Also:** *Oracle Database Java Developer's Guide* for more information about managing security for Oracle Java applications |
| JAVA\_ADMIN | Provides administrative permissions to update policy tables for Oracle Database Java applications.  **See Also:** *Oracle Database Java Developer's Guide* for more information about managing security for Oracle Java applications |

***Table 4–3 (Cont.) Oracle Database Predefined Roles***

|  |  |
| --- | --- |
| **Predefined** **Role** | **Description** |
| JAVA\_DEPLOY | Provides privileges to deploy ncomp DLLs into the javavm/admin directory using the ncomp and deployns utilities. Without this role, the javavm/deploy and javavm/admin directories can be accessible.  **See Also:** *Oracle Database Advanced Application Developer's Guide* for more information |
| JMXSERVER | Provides privileges to start and maintain a JMX agent in a database session.  **See Also:** *Oracle Database Java Developer's Guide* for more information about managing Oracle Java applications |
| LBAC\_DBA | Provides permissions to use the SA\_SYSDBA PL/SQL package.  **See Also:** *Oracle Label Security Administrator's Guide* for more information |
| LOGSTDBY\_ADMINISTRATOR | Provides administrative privileges to manage the SQL Apply (logical standby database) environment.  **See Also:** *Oracle Data Guard Concepts and Administration* for more information |
| MGMT\_USER | Grants the SELECT privilege on the different views used for the SYSMAN  schema. |
| OEM\_ADVISOR | Provides privileges to create, drop, select (read), load (write), and delete a SQL tuning set through the DBMS\_SQLTUNE PL/SQL package, and to access to the Advisor framework using the ADVISOR PL/SQL package.  **See Also:** *Oracle Database Performance Tuning Guide* for more information |
| OEM\_MONITOR | Provides privileges needed by the Management Agent component of Oracle Enterprise Manager to monitor and manage the database.  **See Also:** *Oracle Database Performance Tuning Guide* for more information |
| OLAP\_DBA | Provides administrative privileges to create dimensional objects in different schemas for Oracle OLAP.  **See Also:** *Oracle OLAP User's Guide* for more information |
| OLAP\_USER | Provides application developers privileges to create dimensional objects in their own schemas for Oracle OLAP.  **See Also:** *Oracle OLAP User's Guide* for more information |
| OLAP\_XS\_ADMIN | Provides privileges to administer security for Oracle OLAP.  **See Also:** *Oracle OLAP User's Guide* for more information |
| ORDADMIN | Provides privileges to administer Oracle Multimedia DICOM.  **See Also:** *Oracle Multimedia DICOM Developer's Guide* |
| OWB$CLIENT | Provides privileges to perform standard client-related tasks for Oracle Warehouse Builder, such as creating projects, modules, tables, views, maps, and so on. Warehouse Builder automatically grants this role to all workspace owners and users. (That is, you do not need to explicitly grant it to anyone who must use Warehouse Builder.) For security reasons, the OWB$CLIENT role is not a default role for Warehouse Builder users: Oracle Warehouse Builder enables this role only when it is needed.  **See Also:** *Oracle Warehouse Builder Installation and Administration Guide* for more information |

***Table 4–3 (Cont.) Oracle Database Predefined Roles***

|  |  |
| --- | --- |
| **Predefined Role** | **Description** |
| OWB\_DESIGNCENTER\_VIEW | Provides privileges from the database level for any registered Oracle Warehouse Builder user to query the Warehouse Builder public views, such as ALL\_IV\_PROJECTS. A Warehouse Builder administrator can use the ACCESS\_PUBLICVIEW\_BROWSER system privilege from the Warehouse Builder security level to control an Warehouse Builder user's access to those public views.  **See Also:** *Oracle Warehouse Builder Installation and Administration Guide* for more information |
| OWB\_USER | Provides privileges to create and own an Oracle Warehouse Builder workspace. When a workspace owner registers other database users to this workspace, Oracle Database grants this role to these users. Users with this role also have access to Warehouse Builder Control Center public views and other Control Center utilities. Oracle Warehouse Builder grants this role to all Warehouse Builder users.  **See Also:** *Oracle Warehouse Builder Installation and Administration Guide* for more information |
| RECOVERY\_CATALOG\_OWNER | Provides privileges for owner of the recovery catalog. Includes: CREATE SESSION, ALTER SESSION, CREATE SYNONYM, CREATE VIEW, CREATE DATABASE LINK, CREATE TABLE, CREATE CLUSTER, CREATE SEQUENCE, CREATE TRIGGER, and CREATE PROCEDURE |
| RESOURCE | Provides the following system privileges: CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, CREATE TYPE.  This role is provided for compatibility with previous releases of Oracle Database. You can determine the privileges encompassed by this role by querying the DBA\_SYS\_PRIVS data dictionary view.  **Note:** Oracle recommends that you design your own roles for database security rather than relying on this role. This role may not be created automatically by future releases of Oracle Database.  **See Also:** *Oracle Database Reference* for a description of the DBA\_SYS\_PRIVS  view |
| SCHEDULER\_ADMIN | Allows the grantee to execute the procedures of the DBMS\_SCHEDULER package. It includes all of the job scheduler system privileges and is included in the DBA role.  **See Also:** *Oracle Database Administrator's Guide* for more information about the DBMS\_SCHEDULER package |
| SELECT\_CATALOG\_ROLE | Provides SELECT privilege on objects in the data dictionary. |
| SNMPAGENT | Used by the Enterprise Manager Management Agent. |
| SPATIAL\_CSW\_ADMIN | Provides administrative privileges to manage the Catalog Services for the Web (CSW) component of Oracle Spatial.  **See Also:** *Oracle Spatial Developer's Guide* for more information |
| SPATIAL\_WFS\_ADMIN | Provides administrative privileges to manage the Web Feature Service (WFS) component of Oracle Spatial.  **See Also:** *Oracle Spatial Developer's Guide* for more information |
| WFS\_USR\_ROLE | Provides user privileges for the Web Feature Service (WFS) component of Oracle Spatial.  **See Also:** *Oracle Spatial Developer's Guide* for more information |

***Table 4–3 (Cont.) Oracle Database Predefined Roles***

|  |  |
| --- | --- |
| **Predefined** **Role** | **Description** |
| WM\_ADMIN\_ROLE | Provides administrative privileges for Oracle Workspace Manage. This enables users to run any DBMS\_WM procedures on all version enabled tables, workspaces, and savepoints regardless of their owner. It also enables the user to modify the system parameters specific to Workspace Manager.  **See Also:** *Oracle Database Workspace Manager Developer's Guide* for more information |
| XDBADMIN | Allows the grantee to register an XML schema globally, as opposed to registering it for use or access only by its owner. It also lets the grantee bypass access control list (ACL) checks when accessing Oracle XML DB Repository.  **See Also:** *Oracle XML DB Developer's Guide* for information about XML schemas and the XML DB Repository |
| XDB\_SET\_INVOKER | Allows the grantee to define invoker’s rights handlers and to create or update the resource configuration for XML repository triggers. By default, Oracle Database grants this role to the DBA role but not to the XDBADMIN role.  **See Also:** *Oracle XML DB Developer's Guide* for information about Oracle Database XML repository triggers |
| XDB\_WEBSERVICES | Allows the grantee to access Oracle Database Web services over HTTPS. However, it does not provide the user access to objects in the database that are public. To allow public access, you need to grant the user the XDB\_ WEBSERVICES\_WITH\_PUBLIC role. For a user to use these Web services, SYS must enable the Web service servlets.  **See Also:** *Oracle XML DB Developer's Guide* for information about Oracle Database Web services |
| XDB\_WEBSERVICES\_OVER\_HTTP | Allows the grantee to access Oracle Database Web services over HTTP. However, it does not provide the user access to objects in the database that are public. To allow public access, you need to grant the user the XDB\_ WEBSERVICES\_WITH\_PUBLIC role.  **See Also:** *Oracle XML DB Developer's Guide* for information about Oracle Database Web services |
| XDB\_WEBSERVICES\_WITH\_PUBLIC | Allows the grantee access to public objects through Oracle Database Web services.  **See Also:** *Oracle XML DB Developer's Guide* for information about Oracle Database Web services |

**Note:** Each installation should create its own roles and assign only those privileges that are needed, thus retaining detailed control of the privileges in use. This process also removes any need to adjust existing roles, privileges, or procedures whenever Oracle Database changes or removes roles that Oracle Database defines.

#### Creating a Role

You can create a role using the CREATE ROLE statement, but you must have the CREATE ROLE system privilege to do so. Typically, only security administrators have this system privilege.

After you create a role, the role has no privileges associated with it. Your next step is to grant either privileges or other roles to the new role.

You must give each role you create a unique name among existing user names and role names of the database. Roles are not contained in the schema of any user. In a database

that uses a multibyte character set, Oracle recommends that each role name contain at least one single-byte character. If a role name contains only multibyte characters, then the encrypted role name and password combination is considerably less secure. See Guideline [1](#_bookmark2169) in ["Guidelines for Securing Passwords" on page 10-7](#_bookmark2167) for password guidelines.

[Example 4–2](#_bookmark649) creates the clerk role.

***Example 4–2 Creating a User Role Authorized by a Password***

CREATE ROLE clerk IDENTIFIED BY *password*;

You can use the IDENTIFIED BY clause to authorize the role with a password. The IDENTIFIED BY clause of the CREATE ROLE statement specifies how the user must be authorized before the role can be enabled for use by a specific user to which it has been granted. If you do not specify this clause, or if you specify NOT IDENTIFIED, then no authorization is required when the role is enabled. Roles can be specified to be authorized by the following:

■ The database using a password

■ An application using a specified package

■ Externally by the operating system, network, or other external source

■ Globally by an enterprise directory service

These authorizations are discussed in the following sections.

As an alternative to creating password-protected roles, Oracle recommends that you use secure application roles instead. See ["Securing Role Privileges by Using Secure](#_bookmark707) [Application Roles" on page 4-22](#_bookmark707) for more information.

You can set or change the authorization method for a role using the ALTER ROLE statement. Remember that you can only directly grant secure application roles or password-authenticated roles to a user.

[Example 4–3](#_bookmark654) shows how to alter the clerk role to specify that the user must have been authorized by an external source before enabling the role.

***Example 4–3 Altering a Role to be Authorized by an External Source***

ALTER ROLE clerk IDENTIFIED EXTERNALLY;

To alter the authorization method for a role, you must have the ALTER ANY ROLE

system privilege or have been granted the role with ADMIN option.

**See Also:** *Oracle Database SQL Language Reference* for syntax, restrictions, and authorization information about the SQL statements used to manage roles and privileges

#### Specifying the Type of Role Authorization

The methods of authorizing roles are presented in this section. A role must be enabled for you to use it.

This section contains:

■ [Authorizing a Role by Using the Database](#_bookmark658)

■ [Authorizing a Role by Using an Application](#_bookmark663)

■ [Authorizing a Role by Using an External Source](#_bookmark666)

**See Also:** ["When Do Grants and Revokes Take Effect?" on](#_bookmark862) [page 4-48](#_bookmark862) for a discussion about enabling roles

###### Authorizing a Role by Using the Database

You can protect a role authorized by the database by assigning the role a password. If a user is granted a role protected by a password, then you can enable or disable the role by supplying the proper password for the role in the SET ROLE statement. You cannot authenticate a password-authenticated role on logon, even if you add it to the list of default roles. You must explicitly enable it with the SET ROLE statement using the required password.

[Example 4–4](#_bookmark661) shows how to set a password-authenticated role by using the SET ROLE

statement.

***Example 4–4 Using SET ROLE for a Password-Authenticated Role***

SET ROLE clerk IDENTIFIED BY *password*;

[Example 4–2, "Creating a User Role Authorized by a Password" on page 4-17](#_bookmark649) shows a CREATE ROLE statement that creates a role called clerk. When it is enabled, the password must be supplied.

**Note:**In a database that uses a multibyte character set, passwords for roles must include only single-byte characters. Multibyte characters are not accepted in passwords. See Guideline [1](#_bookmark2169) in ["Guidelines for Securing Passwords" on page 10-7](#_bookmark2167) for password guidelines.

###### Authorizing a Role by Using an Application

An application role (secure application role) can be enabled only by applications using an authorized PL/SQL package. Application developers do not need to secure a role by embedding passwords inside applications. Instead, they can create an application role and specify which PL/SQL package is authorized to enable the role.

To create a role enabled by an authorized PL/SQL package, use the IDENTIFIED USING

*pack**age\_name* clause in the CREATE ROLE SQL statement.

[Example 4–5](#_bookmark665) indicates that the role admin\_role is an application role and the role can only be enabled by any module defined inside the PL/SQL package hr.admin.

***Example 4–5 Creating a Role Authorized by a PL/SQL Package for an Application***

CREATE ROLE admin\_role IDENTIFIED USING hr.admin;

See the following for more information about secure application roles:

■ ["Securing Role Privileges by Using Secure Application Roles" on page 4-22](#_bookmark707)

■ ["Creating Secure Application Roles to Control Access to Applications" on page 5-12](#_bookmark1037)

■ *Oracle Database 2 Day + Security Guide*

###### Authorizing a Role by Using an External Source

You can define the external role locally in the database, but you cannot grant the external role to global users, to global roles, or to any other roles in the database. You can create roles that are authorized by the operating system or network clients.

[Example 4–6](#_bookmark670) creates a role named accts\_rec and requires that the user is authorized by an external source before it can be enabled:

***Example 4–6 Creating a Role Authorized by an External Source***

CREATE ROLE accts\_rec IDENTIFIED EXTERNALLY;

**Authorizing a Role by Using the Operating System** Role authentication through the operating system is useful only when the operating system is able to dynamically link operating system privileges with applications. When a user starts an application, the operating system grants an operating system privilege to the user. The granted operating system privilege corresponds to the role associated with the application. At this point, the application can enable the application role. When the application is terminated, the previously granted operating system privilege is revoked from the operating system account of the user.

If a role is authorized by the operating system, then you must configure information for each user at the operating system level. This operation is operating system dependent.

If roles are granted by the operating system, then you do not need to have the operating system authorize them also.

**See Also:** ["Granting Roles Using the Operating System or](#_bookmark846) [Network" on page 4-45](#_bookmark846) for more information about roles granted by the operating system

**Authorizing a Role by Using a Network Clien****t**If users connect to the database over Oracle Net, then by default, the operating system cannot authenticate their roles. This includes connections through a shared server configuration, as this connection requires Oracle Net. This restriction is the default because a remote user could impersonate another operating system user over a network connection. Oracle recommends that you set REMOTE\_OS\_ROLES to FALSE, which is the default.

If you are not concerned with this security risk and want to use operating system role authentication for network clients, then set the initialization parameter REMOTE\_OS\_ ROLES in the database initialization parameter file to TRUE. The change will take effect the next time you start the instance and mount the database.

###### Global Role Authorization by an Enterprise Directory Service

A role can be defined as a global role, where a (global) user can only be authorized to use the role by an enterprise directory service. You define the global role locally in the database by granting privileges and roles to it, but you cannot grant the global role itself to any user or other role in the database. When a global user attempts to connect to the database, the enterprise directory is queried to obtain any global roles associated with the user.

[Example 4–7](#_bookmark682) creates a global role.

***Example 4–7 Creating a Global Role***

CREATE ROLE supervisor IDENTIFIED GLOBALLY;

Global roles are one component of enterprise user security. A global role only applies to one database, but you can grant it to an enterprise role defined in the enterprise directory. An enterprise role is a directory structure that contains global roles on multiple databases and can be granted to enterprise users.

See ["Configuring Global User Authentication and Authorization" on page 3-30](#_bookmark375) for a general discussion of global authentication and authorization of users, and its role in enterprise user management.

**See Also:** *Oracle Database Enterprise User Security Administrator's Guide* for information about implementing enterprise user management

#### Granting and Revoking Roles

This section contains:

■ [About Granting and Revoking Roles](#_bookmark686)

■ [Who Can Grant or Revoke Roles?](#_bookmark690)

**See Also:**

■ ["Granting a User Privileges and Roles" on page 4-37](#_bookmark791)

■ ["Revoking Privileges and Roles from a User" on page 4-41](#_bookmark822)

■ ["When Do Grants and Revokes Take Effect?" on page 4-48](#_bookmark862)

■ ["Finding Information About User Privileges and Roles" on page 4-71](#_bookmark960)

■ *Oracle Database 2 Day DBA* for more information about Database Control

###### About Granting and Revoking Roles

You can grant system or object privileges to a role, and any role can be granted to any database user or to another role (but not to itself). However, a role cannot be granted circularly, that is, role X cannot be granted to role Y if role Y has previously been granted to role X.

To provide selective availability of privileges, Oracle Database permits applications and users to enable and disable roles. Each role granted to a user is, at any given time, either enabled or disabled. The security domain of a user includes the privileges of all roles currently enabled for the user and excludes the privileges of any roles currently disabled for the user.

A role granted to a role is called an indirectly granted role. You can explicitly enable or disable it for a user. However, whenever you enable a role that contains other roles, you implicitly enable all indirectly granted roles of the directly granted role.

You grant roles to (or revoke roles from) users or other roles by using either of the following methods:

■ Oracle Enterprise Manager Database Control

■ The GRANT and REVOKE SQL statements

Privileges are granted to and revoked from roles using the same options.

You cannot grant a secure role (that is, an IDENTIFIED BY role, IDENTIFIED USING role, or IDENTIFIED EXTERNALLY role) to a non-secure role. You can use the SET ROLE statement to enable the secure role for the session.

###### Who Can Grant or Revoke Roles?

Any user with the GRANT ANY ROLE system privilege can grant or revoke any role except a global role to or from other users or roles of the database. (A global role is managed

in a directory, such as Oracle Internet Directory, but its privileges are contained within a single database.) By default, the SYS or SYSTEM user has this privilege. You should grant this system privilege conservatively because it is very powerful.

Any user granted a role with the ADMIN OPTION can grant or revoke that role to or from other users or roles of the database. This option allows administrative powers for roles to be granted on a selective basis.

**See Also:** *Oracle Database Enterprise User Security Administrator's Guide* for information about global roles

#### Dropping Roles

In some cases, it may be appropriate to drop a role from the database. The security domains of all users and roles granted a dropped role are immediately changed to reflect the absence of the dropped role privileges. All indirectly granted roles of the dropped role are also removed from affected security domains. Dropping a role automatically removes the role from all user default role lists.

Because the existence of objects is not dependent on the privileges received through a role, tables and other objects are not dropped when a role is dropped.

You can drop a role using the SQL statement DROP ROLE. To drop a role, you must have the DROP ANY ROLE system privilege or have been granted the role with the ADMIN option.

The following statement drops the role CLERK:

DROP ROLE clerk;

#### Restricting SQL\*Plus Users fromUsing Database Roles

This section describes features that you can use to restrict SQL\*Plus users from using database roles and thus, prevent serious security problems.

■ [Potential Security Problems of Using Ad Hoc Tools](#_bookmark699)

■ [Limiting Roles Through the PRODUCT\_USER\_PROFILE Table](#_bookmark703)

■ [Using Stored Procedures to Encapsulate Business Logic](#_bookmark706)

###### Potential Security Problems of Using Ad Hoc Tools

Prebuilt database applications explicitly control the potential actions of a user, including the enabling and disabling of user roles while using the application. By contrast, ad hoc query tools such as SQL\*Plus, permit a user to submit any SQL statement (which may or may not succeed), including enabling and disabling a granted role.

Potentially, an application user can exercise the privileges attached to that application to issue destructive SQL statements against database tables by using an ad hoc tool.

For example, consider the following scenario:

■ The Vacation application has a corresponding vacation role.

■ The vacation role includes the privileges to issue SELECT, INSERT, UPDATE, and

DELETE statements against the emp\_tab table.

■ The Vacation application controls the use of privileges obtained through the

vacation role.

Now, consider a user who has been granted the vacation role. Suppose that, instead of using the Vacation application, the user executes SQL\*Plus. At this point, the user is restricted only by the privileges granted to him explicitly or through roles, including the vacation role. Because SQL\*Plus is an ad hoc query tool, the user is not restricted to a set of predefined actions, as with designed database applications. The user can query or modify data in the emp\_tab table as he or she chooses.

###### Limiting Roles Through the PRODUCT\_USER\_PROFILE Table

You can use the PRODUCT\_USER\_PROFILE table, which is in the SYSTEM schema, to disable certain SQL and SQL\*Plus commands in the SQL\*Plus environment for each user. SQL\*Plus, not the Oracle Database, enforces this security. You can even restrict access to the GRANT, REVOKE, and SET ROLE commands to control user ability to change their database privileges.

The PRODUCT\_USER\_PROFILE table enables you to list roles that you do not want users to activate with an application. You can also explicitly disable the use of various commands, such as SET ROLE.

For example, you could create an entry in the PRODUCT\_USER\_PROFILE table to:

■ Disallow the use of the clerk and manager roles with SQL\*Plus

■ Disallow the use of SET ROLE with SQL\*Plus

Suppose user Marla connects to the database using SQL\*Plus. Marla has the clerk, manager, and analyst roles. As a result of the preceding entry in PRODUCT\_USER\_ PROFILE, Marla is only able to exercise her analyst role with SQL\*Plus. Also, when Ginny attempts to issue a SET ROLE statement, she is explicitly prevented from doing so because of the entry in the PRODUCT\_USER\_PROFILE table prohibiting use of SET ROLE.

Be aware that the PRODUCT\_USER\_PROFILE table does not completely guarantee security, for multiple reasons. In the preceding example, while SET ROLE is disallowed with SQL\*Plus, if Marla had other privileges granted to her directly, then she could exercise these using SQL\*Plus.

**See Also:** *SQL\*Plus User's Guide and Reference* for more information about the PRODUCT\_USER\_PROFILE table

###### Using Stored Procedures to Encapsulate Business Logic

Stored procedures encapsulate the use of privileges with business logic so that privileges are only exercised in the context of a well-formed business transaction. For example, an application developer can create a procedure to update the employee name and address in the employees table, which enforces that the data can only be updated in normal business hours. Also, rather than grant a human resources clerk the UPDATE privilege on the employees table, a security administrator may grant the privilege on the procedure only. Then, the human resources clerk can exercise the privilege only in the context of the procedures, and cannot update the employees table directly.

#### Securing Role Privileges by Using Secure Application Roles

A secure application role is a role that can be enabled only by an authorized PL/SQL package (or procedure). The PL/SQL package itself reflects the security policies needed to control access to the application.

This method of role creation restricts the enabling of this type of role to the invoking application. For example, the application can perform authentication and customized authorization, such as checking whether the user has connected through a proxy.

This type of role strengthens security because passwords are not embedded in application source code or stored in a table. This way, the actions the database performs are based on the implementation of your security policies, and these definitions are stored in one place, the database, rather than in your applications. If you need to modify the policy, you do so in one place without having to modify your applications. No matter how users connect to the database, the result is always the same, because the policy is bound to the role.

To enable the secure application role, you must execute its underlying package by invoking it directly from the application when the user logs in, before the user exercises the privileges granted by the secure application role. You cannot use a logon trigger to enable a secure application role, nor can you have this type of role be a default role.

When you enable the secure application role, Oracle Database verifies that the authorized PL/SQL package is on the calling stack, that is, it verifies that the authorized PL/SQL package is issuing the command to enable the role.

You can use secure application roles to ensure the existence of a database connection. Because a secure application role is a role implemented by a package, the package can validate that users can connect to the database through a middle tier or from a specific IP address. In this way, the secure application role prevents users from accessing data outside an application. They are forced to work within the framework of the application privileges that they have been granted.

**See Also:**

■ ["Creating Secure Application Roles to Control Access to Applications" on page 5-12](#_bookmark1037)

■ *Oracle Database 2 Day + Security Guide*

### Managing Object Privileges

This section contains:

* [About Object Privileges](#_bookmark713)
* [Granting or Revoking Object Privileges](#_bookmark715)
* [Managing Object Privileges](#_bookmark719)
* [Managing Table Privileges](#_bookmark728)
* [Managing View Privileges](#_bookmark735)
* [Managing Procedure Privileges](#_bookmark745)
* [Managing Type Privileges](#_bookmark775)

#### About Object Privileges

An **object privilege** is a right that you grant to a user on a database object. Some examples of object privileges include the right to:

* Use an edition
* Update a table

■ Select rows from another user’s table

■ Execute a stored procedure of another user

**See Also:** *Oracle Database SQL Language Reference* for a list of object privileges and the operations they authorize

#### Granting or Revoking Object Privileges

Each type of object has different privileges associated with it.

You can specify ALL [PRIVILEGES] to grant or revoke all available object privileges for an object. ALL is not a privilege; rather, it is a shortcut, or a way of granting or revoking all object privileges with one GRANT and REVOKE statement. If all object privileges are granted using the ALL shortcut, then individual privileges can still be revoked.

Similarly, you can revoke all individually granted privileges by specifying ALL. However, if you REVOKE ALL, and revoking causes integrity constraints to be deleted (because they depend on a REFERENCES privilege that you are revoking), then you must include the CASCADE CONSTRAINTS option in the REVOKE statement.

[Example 4–8](#_bookmark718) revokes all privileges on the orders table in the HR schema using CASCADE CONSTRAINTS.

***Example 4–8 Revoking All Object Privileges Using CASCADE CONSTRAINTS***

REVOKE ALL

ON orders FROM hr CASCADE CONSTRAINTS;

#### Managing Object Privileges

An **object privilege** grants permission to perform a particular action on a specific schema object.

Different object privileges are available for different types of schema objects. The privilege to delete rows from the departments table is an example of an object privilege.

Some schema objects, such as clusters, indexes, triggers, and database links, do not have associated object privileges. Their use is controlled with system privileges. For example, to alter a cluster, a user must own the cluster or have the ALTER ANY CLUSTER system privilege.

The following sections discuss granting and revoking such privileges:

■ ["Granting and Revoking Object Privileges" on page 4-25](#_bookmark723)

■ ["Who Can Grant Object Privileges?" on page 4-25](#_bookmark725)

■ ["Using Object Privileges with Synonyms" on page 4-25](#_bookmark726)

The following sections discuss object privileges that apply to specific schema objects:

■ ["Managing Table Privileges" on page 4-26](#_bookmark728)

■ ["Managing View Privileges" on page 27](#_bookmark735)

■ Sequences (see *Oracle Database Administrator's Guide* for information about managing sequences)

■ ["Managing Procedure Privileges" on page 4-29](#_bookmark745)

■ Functions and Packages(*Oracle Database Administrator's Guide* for information about managing object dependencies)

■ ["Managing Type Privileges" on page 4-33](#_bookmark775)

###### Granting and Revoking Object Privileges

Object privileges can be granted to and revoked from users and roles. If you grant object privileges to roles, then you can make the privileges selectively available.

You can grant or revoke object privileges to or from users and roles using the following methods:

■ The GRANT and REVOKE SQL statements

■ Oracle Enterprise Manager Database Control

**See Also:** *Oracle Database 2 Day DBA* for more information about Database Control

###### Who Can Grant Object Privileges?

A user automatically has all object privileges for schema objects contained in his or her schema. A user with the GRANT ANY OBJECT PRIVILEGE can grant any specified object privilege to another user with or without the WITH GRANT OPTION clause of the GRANT statement. A user with the GRANT ANY OBJECT PRIVILEGE can also use that privilege to revoke any object privilege that was granted either by the object owner or by some other user with the GRANT ANY OBJECT PRIVILEGE privilege. Otherwise, the grantee can use the privilege, but cannot grant it to other users.

**See Also:** *Oracle Database SQL Language Reference* for information about GRANT and GRANT ANY OBJECT PRIVILEGE

###### Using Object Privileges with Synonyms

You can use the CREATE SYNONYM statement to create synonyms for tables, views, sequences, operators, procedures, stored functions, packages, materialized views, Java class schema objects, user-defined object types, or other synonyms. If you grant users the privilege to use the synonym, then the object privileges granted on the underlying objects apply whether the user references the base object by name or by using the synonym.

For example, suppose user OE creates the following synonym for the CUSTOMERS table:

CREATE SYNONYM customer\_syn FOR CUSTOMERS;

Then OE grants the SELECT privilege on the customer\_syn synonym to user HR.

GRANT SELECT ON customer\_syn TO HR;

User HR then tries either of the following queries:

SELECT COUNT(\*) FROM OE.customer\_syn; SELECT COUNT(\*) FROM OE.CUSTOMERS;

Both queries will yield the same result:

COUNT(\*)

----------

319

Be aware that when you grant the synonym to another user, the grant applies to the underlying object that the synonym represents, not to the synonym itself. For example, if user HR queries the ALL\_TAB\_PRIVS data dictionary view for his privileges, he will learn the following:

SELECT TABLE\_SCHEMA, TABLE\_NAME, PRIVILEGE FROM ALL\_TAB\_PRIVS

WHERE TABLE\_SCHEMA = 'OE';

|  |  |  |
| --- | --- | --- |
| TABLE\_SCHEMA | TABLE\_NAME | PRIVILEGE |
| ------------ | ---------- | ---------- |
| **OE** | **CUSTOMER** | **SELECT** |
| OE | ORDERS | UPDATE |

The results show that in addition to other privileges, he has the SELECT privilege for the underlying object of the customer\_syn synonym, which is the OE.CUSTOMER table.

At this point, if user OE then revokes the SELECT privilege on the customer\_syn

synonym from HR, here are the results if HR checks his privileges again:

|  |  |  |
| --- | --- | --- |
| TABLE\_SCHEMA | TABLE\_NAME | PRIVILEGE |
| ------------ | ---------- | --------- |
| OE | ORDERS | UPDATE |

User HR no longer has the SELECT privilege for the OE.CUSTOMER table. If he tries to query the OE.CUSTOMERS table, then the following error appears:

SELECT COUNT(\*) FROM OE.CUSTOMERS;

ERROR at line 1:

ORA-00942: table or view does not exist

#### Managing Table Privileges

Object privileges for tables enable table security at the DML (data manipulation language) or DDL (data definition language) level of operation.

The following sections discuss table privileges and DML and DDL operations:

* + [How Table Privileges Affect Data Manipulation Language Operations](#_bookmark730)
  + [How Table Privileges Affect Data Definition Language Operations](#_bookmark733)

###### How Table Privileges Affect Data Manipulation Language Operations

You can grant privileges to use the DELETE, INSERT, SELECT, and UPDATE DML operations on a table or view. Grant these privileges only to users and roles that need to query or manipulate data in a table.

You can restrict INSERT and UPDATE privileges for a table to specific columns of the table. With a selective INSERT privilege, a privileged user can insert a row with values for the selected columns. All other columns receive NULL or the default value of the column. With a selective UPDATE privilege, a user can update only specific column values of a row. You can use selective INSERT and UPDATE privileges to restrict user access to sensitive data.

For example, if you do not want data entry users to alter the salary column of the employees table, then selective INSERT or UPDATE privileges can be granted that exclude the salary column. Alternatively, a view that excludes the salary column could satisfy this need for additional security.

**See Also:** *Oracle Database SQL Language Reference* for more information about DML operations

###### How Table Privileges Affect Data Definition Language Operations

The ALTER, INDEX, and REFERENCES privileges allow DDL operations to be performed on a table. Because these privileges allow other users to alter or create dependencies on a table, you should grant these privileges conservatively.

A user attempting to perform a DDL operation on a table may need additional system or object privileges. For example, to create a trigger on a table, the user requires both the ALTER TABLE object privilege for the table and the CREATE TRIGGER system privilege.

As with the INSERT and UPDATE privileges, you can grant the REFERENCES privilege on specific columns of a table. The REFERENCES privilege enables the grantee to use the table on which the grant is made as a parent key to any foreign keys that the grantee wishes to create in his or her own tables. This action is controlled with a special privilege because the presence of foreign keys restricts the data manipulation and table alterations that can be done to the parent key. A column-specific REFERENCES privilege restricts the grantee to using the named columns (which, of course, must include at least one primary or unique key of the parent table).

**See Also:** "Data Integrity" in *Oracle Database Concepts* for more information about primary keys, unique keys, and integrity constraints

#### Managing View Privileges

This section contains:

* + - [About View Privileges](#_bookmark736)
    - [Privileges Required to Create Views](#_bookmark739)
    - [Increasing Table Security with Views](#_bookmark742)

###### About View Privileges

A **view** is a presentation of data selected from one or more tables, possibly including other views. A view shows the structure of the underlying tables. Its selected data can be thought of as the result of a stored query. A view contains no actual data but rather derives what it shows from the tables and views on which it is based. You can query a view, and change the data it represents. Data in a view can be updated or deleted, and new data inserted. These operations directly alter the tables on which the view is based, and are subject to the integrity constraints and triggers of the base tables.

You can apply DML object privileges to views, similar to tables. Object privileges for a view allow various DML operations, which as noted affect the base tables from which the view is derived.

###### Privileges Required to Create Views

To create a view, you must meet the following requirements:

* + - You must have been granted one of the following system privileges, either explicitly or through a role:
      * The CREATE VIEW system privilege (to create a view in your schema)
      * The CREATE ANY VIEW system privilege (to create a view in the schema of another user)
  + You must have been explicitly granted one of the following privileges:
* The SELECT, INSERT, UPDATE, or DELETE object privileges on all base objects underlying the view
* The SELECT ANY TABLE, INSERT ANY TABLE, UPDATE ANY TABLE, or DELETE ANY TABLE system privileges
  + In addition, before you can grant other users access to you view, you must have object privileges to the base objects with the GRANT OPTION clause or appropriate system privileges with the ADMIN OPTION clause. If you do not have these privileges, then you cannot to grant other users access to your view. If you try, an ORA-01720: grant option does not exist for *object\_name* error is raised, with *object\_name* referring to the view’s underlying object for which you do not have the sufficient privilege.

**See Also:** *Oracle Database SQL Language Reference*

###### Increasing Table Security with Views

To use a view, the user must have the appropriate privileges but only for the view itself, not its underlying objects. However, if access privileges for the underlying objects of the view are removed, then the user no longer has access. This behavior occurs because the security domain that is used when a user queries the view is that of the definer of the view. If the privileges on the underlying objects are revoked from the view’s definer, then the view becomes invalid, and no one can use the view. Therefore, even if a user has been granted access to the view, the user may not be able to use the view if the definer’s rights have been revoked from the view’s underlying objects.

For example, suppose User A creates a view. User A has definer’s rights on the underlying objects of the view. User A then grants the SELECT privilege on that view to User B so that User B can query the view. But if User A no longer has access to the underlying objects of that view, then User B no longer has access either.

Views add two more levels of security for tables, column-level security and value-based security, as follows:

* + **A view can provide access to selected columns of base tables.** For example, you can define a view on the employees table to show only the employee\_id, last\_ name, and manager\_id columns:

CREATE VIEW employees\_manager AS

SELECT last\_name, employee\_id, manager\_id FROM employees;

* + **A view can provide value-based security for the information in a table.** A WHERE clause in the definition of a view displays only selected rows of base tables. Consider the following two examples:

CREATE VIEW lowsal AS SELECT \* FROM employees WHERE salary < 10000;

The lowsal view allows access to all rows of the employees table that have a salary value less than 10000. Notice that all columns of the employees table are accessible in the lowsal view.

CREATE VIEW own\_salary AS SELECT last\_name, salary FROM employees

WHERE last\_name = USER;

In the own\_salary view, only the rows with an last\_name that matches the current user of the view are accessible. The own\_salary view uses the user pseudo column, whose values always refer to the current user. This view combines both column-level security and value-based security.

#### Managing Procedure Privileges

This section contains:

* + - [Using the EXECUTE Privilege for Procedure Privileges](#_bookmark746)
    - [Procedure Execution and Security Domains](#_bookmark752)
    - [How Procedure Privileges Affect Definer’s Rights](#_bookmark753)
    - [How Procedure Privileges Affect Invoker’s Rights](#_bookmark760)
    - [System Privileges Required to Create or Replace a Procedure](#_bookmark763)
    - [System Privileges Required to Compile a Procedure](#_bookmark766)
    - [How Procedure Privileges Affect Packages and Package Objects](#_bookmark770)

###### Using the EXECUTE Privilege for Procedure Privileges

The EXECUTE privilege is the only **object privilege** for procedures, including standalone procedures and functions, and for those within packages. Grant this privilege only to users who need to run a procedure or to compile another procedure that calls a desired procedure.

###### Procedure Execution and Security Domains

A user with the EXECUTE object privilege for a specific procedure can execute the procedure or compile a program unit that references the procedure. Oracle Database performs a run-time privilege check when any PL/SQL unit is called. A user with the EXECUTE ANY PROCEDURE system privilege can execute any procedure in the database. Privileges to run procedures can be granted to a user through roles.

**See Also:** *Oracle Database PL/SQL Language Reference* for more information about how Oracle Database checks privileges at run-time

###### How Procedure Privileges Affect Definer’s Rights

The owner of a procedure, called the *definer*, must have all the necessary object privileges for referenced objects. If the procedure owner grants to another user the right to use that procedure, then the privileges of the procedure owner (on the objects referenced by the procedure) apply to the grantee user’s exercise of the procedure. The privileges of the procedure’s definer must be granted directly to the user, not granted through roles. These are termed definer’s rights.

The user of a procedure who is not its owner is called the *invoker*. Additional privileges on referenced objects are required for invoker’s rights procedures, but not for definer's rights procedures.

**See Also:** ["How Roles Work in PL/SQL Blocks" on page 4-9](#_bookmark569)

A user of a definer’s rights procedure requires only the privilege to execute the procedure and no privileges on the underlying objects that the procedure accesses. This is because a definer's rights procedure operates under the security domain of the user who owns the procedure, regardless of who is executing it. The owner of the procedure must have all the necessary object privileges for referenced objects. Fewer

privileges have to be granted to users of a definer’s rights procedure. This results in stronger control of database access.

You can use definer’s rights procedures to control access to private database objects and add a level of database security. By writing a definer's rights procedure and granting only EXECUTE privilege to a user, the user can be forced to access the referenced objects only through the procedure.

At run time, Oracle Database checks whether the privileges of the owner of a definer’s rights stored procedure allow access to that procedure’s referenced objects, before the procedure is executed. If a necessary privilege on a referenced object was revoked from the owner of a definer's rights procedure, then the procedure cannot be run by the owner or any other user.

**Note:**Trigger processing follows the same patterns as definer's rights procedures. The user runs a SQL statement, which that user is privileged to run. As a result of the SQL statement, a trigger is fired. The statements within the triggered action temporarily execute under the security domain of the user that owns the trigger. For more information, see "Overview of Triggers" in *Oracle Database Concepts*.

###### How Procedure Privileges Affect Invoker’s Rights

An invoker’s rights procedure executes with all of the invoker’s privileges. Oracle Database enables the privileges that were granted to the invoker through any of the invoker’s enabled roles to take effect, unless a definer’s rights procedure calls the invoker's rights procedure directly or indirectly. A user of an invoker's rights procedure needs privileges (granted to the user either directly or through a role) on objects that the procedure accesses through external references that are resolved in the schema of the invoker.

The invoker needs privileges at run time to access program references embedded in DML statements or dynamic SQL statements, because they are effectively recompiled at run time.

For all other external references, such as direct PL/SQL function calls, Oracle Database checks the privileges of the owner at compile time, but does not perform a run-time check. Therefore, the user of an invoker's rights procedure does not need privileges on external references outside DML or dynamic SQL statements. Alternatively, the developer of an invoker’s rights procedure must only grant privileges on the procedure itself, not on all objects directly referenced by the invoker's rights procedure.

You can create a software bundle that consists of multiple program units, some with definer's rights and others with invoker's rights, and restrict the program entry points *(controlled step-in)*. A user who has the privilege to run an entry-point procedure can also execute internal program units indirectly, but cannot directly call the internal programs. For very precise control over query processing, you can create a PL/SQL package specification with explicit cursors.

**See Also:**

* ["Configuring an Oracle Virtual Private Database Policy" on page 7-5](#_bookmark1378)
* *Oracle Database PL/SQL Language Reference* for information about how Oracle Database handles name resolution and privilege checking at runtime using invoker’s and definer’s rights
* *Oracle Database PL/SQL Language Reference* for information about defining explicit cursors in the CREATE PACKAGE statement

###### System Privileges Required to Create or Replace a Procedure

To create or replace a procedure in your own schema, you must have the CREATE PROCEDURE system privilege. To create or replace a procedure in another user's schema, you must have the CREATE ANY PROCEDURE system privilege.

The user who owns the procedure also must have privileges for schema objects referenced in the procedure body. To create a procedure, you need to have been explicitly granted the necessary privileges (system or object) on all objects referenced by the procedure. You cannot obtain the required privileges through roles. This includes the EXECUTE privilege for any procedures that are called inside the procedure being created.

**Note:** Triggers require that privileges on referenced objects be granted directly to the owner of the trigger. Anonymous PL/SQL blocks can use any privilege, whether the privilege is granted explicitly or through a role.

###### System Privileges Required to Compile a Procedure

To compile a standalone procedure, run the ALTER PROCEDURE statement with the COMPILE clause. To compile a procedure that is part of a package, run the ALTER PACKAGE statement.

[Example 4–9](#_bookmark769) shows how to compile a standalone procedure.

***Example 4–9 Compiling a Procedure***

ALTER PROCEDURE psmith.remove\_emp COMPILE;

If the standalone or packaged procedure is in another user’s schema, you must have the ALTER ANY PROCEDURE privilege to recompile it. You can recompile procedures in your own schema without any privileges.

###### How Procedure Privileges Affect Packages and Package Objects

A user with the EXECUTE object privilege for a package can execute any public procedure or function in the package, and can access or modify the value of any public package variable. You cannot grant specific EXECUTE privileges for individual constructs in a package. Therefore, you may find it useful to consider two alternatives for establishing security when developing procedures, functions, and packages for a database application. The following examples describe these alternatives.

Procedure Privileges and Packages and Package Objects: Example 1

[Example 4–10](#_bookmark773) shows four procedures created in the bodies of two packages.

***Example 4–10 Package Objects Affected by Procedure Privileges***

CREATE PACKAGE BODY hire\_fire AS PROCEDURE hire(...) IS

BEGIN

INSERT INTO employees . . . END hire;

PROCEDURE fire(...) IS BEGIN

DELETE FROM employees . . . END fire;

END hire\_fire;

CREATE PACKAGE BODY raise\_bonus AS PROCEDURE give\_raise(...) IS

BEGIN

UPDATE employees SET salary = . . . END give\_raise;

PROCEDURE give\_bonus(...) IS BEGIN

UPDATE employees SET bonus = . . . END give\_bonus;

END raise\_bonus;

The following GRANT EXECUTE statements enable the big\_bosses and little\_bosses

roles to run the appropriate procedures:

GRANT EXECUTE ON hire\_fire TO big\_bosses; GRANT EXECUTE ON raise\_bonus TO little\_bosses;

**Note:** Granting EXECUTE privilege for a package provides uniform access to all package objects.

Procedure Privileges and Packages and Package Objects: Example 2

This example shows four procedure definitions within the body of a single package. Two additional standalone procedures and a package are created specifically to provide access to the procedures defined in the main package.

CREATE PACKAGE BODY employee\_changes AS PROCEDURE change\_salary(...) IS BEGIN ... END; PROCEDURE change\_bonus(...) IS BEGIN ... END;

PROCEDURE insert\_employee(...) IS BEGIN ... END; PROCEDURE delete\_employee(...) IS BEGIN ... END;

END employee\_changes;

CREATE PROCEDURE hire BEGIN

employee\_changes.insert\_employee(...) END hire;

CREATE PROCEDURE fire BEGIN

employee\_changes.delete\_employee(...) END fire;

PACKAGE raise\_bonus IS PROCEDURE give\_raise(...) AS

BEGIN

employee\_changes.change\_salary(...) END give\_raise;

PROCEDURE give\_bonus(...) BEGIN

employee\_changes.change\_bonus(...) END give\_bonus;

Using this method, the procedures that actually do the work (the procedures in the employee\_changes package) are defined in a single package and can share declared global variables, cursors, on so on. By declaring top-level procedures, hire and fire, and an additional package, raise\_bonus, you can grant selective EXECUTE privileges on procedures in the main package:

GRANT EXECUTE ON hire, fire TO big\_bosses; GRANT EXECUTE ON raise\_bonus TO little\_bosses;

#### Managing Type Privileges

The following sections describe the use of privileges for types, methods, and objects:

* + - [System Privileges for Named Types](#_bookmark777)
    - [Object Privileges](#_bookmark781)
    - [Method Execution Model](#_bookmark782)
    - [Privileges Required to Create Types and Tables Using Types](#_bookmark783)
    - [Example of Privileges for Creating Types and Tables Using Types](#_bookmark785)
    - [Privileges on Type Access and Object Access](#_bookmark787)
    - [Type Dependencies](#_bookmark790)

###### System Privileges for Named Types

[Table 4–4](#_bookmark779) lists system privileges for named types (object types, VARRAYs, and nested tables).

***Table 4–4 System Privileges for Named Types***

|  |  |
| --- | --- |
| **Privilege** | **Enables you to ...** |
| CREATE TYPE | Create named types in your own schemas |
| CREATE ANY TYPE | Create a named type in any schema |
| ALTER ANY TYPE | Alter a named type in any schema |
| DROP ANY TYPE | Drop a named type in any schema |
| EXECUTE ANY TYPE | Use and reference a named type in any schema |

The RESOURCE role includes the CREATE TYPE system privilege. The DBA role includes all of these privileges.

###### Object Privileges

The only object privilege that applies to named types is EXECUTE. If the EXECUTE

privilege exists on a named type, then a user can use the named type to:

* + Define a table
  + Define a column in a relational table
  + Declare a variable or parameter of the named type

The EXECUTE privilege permits a user to invoke the methods in the type, including the type constructor. This is similar to the EXECUTE privilege on a stored PL/SQL procedure.

###### Method Execution Model

Method execution is the same as any other stored PL/SQL procedure.

**See Also:** ["Managing Procedure Privileges" on page 4-29](#_bookmark745)

###### Privileges Required to Create Types and Tables Using Types

To create a type, you must meet the following requirements:

* + You must have the CREATE TYPE system privilege to create a type in your schema or the CREATE ANY TYPE system privilege to create a type in the schema of another user. These privileges can be acquired explicitly or through a role.
  + The owner of the type must be explicitly granted the EXECUTE object privileges to access all other types referenced within the definition of the type, or have been granted the EXECUTE ANY TYPE system privilege. The owner cannot obtain the required privileges through roles.
  + If the type owner intends to grant access to the type to other users, then the owner must receive the EXECUTE privileges to the referenced types with the GRANT OPTION or the EXECUTE ANY TYPE system privilege with the ADMIN OPTION. If not, then the type owner has insufficient privileges to grant access on the type to other users.

To create a table using types, you must meet the requirements for creating a table and the following additional requirements:

* + The owner of the table must have been directly granted the EXECUTE object privilege to access all types referenced by the table, or has been granted the EXECUTE ANY TYPE system privilege. The owner cannot exercise the required privileges if these privileges were granted through roles.
  + If the table owner intends to grant access to the table to other users, then the owner must have the EXECUTE privilege to the referenced types with the GRANT OPTION or the EXECUTE ANY TYPE system privilege with the ADMIN OPTION. If not, then the table owner has insufficient privileges to grant access on the table.

**See Also:** ["Managing Table Privileges" on page 4-26](#_bookmark728) for the requirements for creating a table

###### Example of Privileges for Creating Types and Tables Using Types

Assume that three users exist with the CONNECT and RESOURCE roles:

* + user1
  + user2
  + user3

The following DDL is run in the schema of user1:

CREATE TYPE type1 AS OBJECT (

attr1 NUMBER);

CREATE TYPE type2 AS OBJECT (

attr2 NUMBER);

GRANT EXECUTE ON type1 TO user2;

GRANT EXECUTE ON type2 TO user2 WITH GRANT OPTION;

The following DDL is performed in the schema of user2:

CREATE TABLE tab1 OF user1.type1; CREATE TYPE type3 AS OBJECT (

attr3 user1.type2); CREATE TABLE tab2 (

col1 user1.type2);

The following statements succeed because user2 has EXECUTE privilege on

user1.type2 with the GRANT OPTION:

GRANT EXECUTE ON type3 TO user3;

GRANT SELECT on tab2 TO user3;

However, the following grant fails because user2 does not have EXECUTE privilege on

user1.type1 with the GRANT OPTION:

GRANT SELECT ON tab1 TO user3;

The following statements can be successfully run by user3:

CREATE TYPE type4 AS OBJECT (

attr4 user2.type3); CREATE TABLE tab3 OF type4;

**Note:** Customers should discontinue using the CONNECT and RESOURCE roles. The CONNECT role presently retains only the CREATE SESSION privilege.

###### Privileges on Type Access and Object Access

Existing column-level and table-level privileges for DML statements apply to both column objects and row objects.

[Table 4–5](#_bookmark789) lists the privileges for object tables.

***Table 4–5 Privileges for Object Tables***

|  |  |
| --- | --- |
| **Privilege** | **Enables you to...** |
| SELECT | Access an object and its attributes from the table |
| UPDATE | Modify the attributes of the objects that make up the rows in the table |
| INSERT | Create new objects in the table |
| DELETE | Delete rows |

Similar table privileges and column privileges apply to column objects. Retrieving instances does not in itself reveal type information. However, clients must access named type information to interpret the type instance images. When a client requests type information, Oracle Database checks for the EXECUTE privilege on the type.

Consider the following schema:

CREATE TYPE emp\_type (

eno NUMBER, ename CHAR(31), eaddr addr\_t); CREATE TABLE emp OF emp\_t;

In addition, consider the following two queries:

SELECT VALUE(emp) FROM emp;

SELECT eno, ename FROM emp;

For either query, Oracle Database checks the SELECT privilege of the user for the emp table. For the first query, the user must obtain the emp\_type type information to interpret the data. When the query accesses the emp\_type type, Oracle Database checks the EXECUTE privilege of the user.

The second query, however, does not involve named types, so Oracle Database does not check type privileges.

In addition, by using the schema from the previous section, user3 can perform the following queries:

SELECT tab1.col1.attr2 FROM user2.tab1 tab1; SELECT attr4.attr3.attr2 FROM tab3;

Note that in both SELECT statements, user3 does not have explicit privileges on the underlying types, but the statement succeeds because the type and table owners have the necessary privileges with the GRANT OPTION.

Oracle Database checks privileges on the following events, and returns an error if the client does not have the privilege for the action:

* + Pinning an object in the object cache using its REF value causes Oracle Database to check for the SELECT privilege on the containing object table.
  + Modifying an existing object or flushing an object from the object cache causes Oracle Database to check for the UPDATE privilege on the destination object table.
  + Flushing a new object causes Oracle Database to check for the INSERT privilege on the destination object table.
  + Deleting an object causes Oracle Database to check for the DELETE privilege on the destination table.
  + Pinning an object of a named type causes Oracle Database to check EXECUTE

privilege on the object.

Modifying the attributes of an object in a client third-generation language application causes Oracle Database to update the entire object. Therefore, the user needs the UPDATE privilege on the object table. Having the UPDATE privilege on only certain columns of the object table is not sufficient, even if the application only modifies attributes corresponding to those columns. Therefore, Oracle Database does not support column-level privileges for object tables.

###### Type Dependencies

As with stored objects, such as procedures and tables, types being referenced by other objects are called dependencies. There are some special issues for types on which tables depend. Because a table contains data that relies on the type definition for access, any change to the type causes all stored data to become inaccessible. Changes that can cause this are when necessary privileges required to use the type are revoked, or the type or dependent types are dropped. If these actions occur, then the table becomes invalid and cannot be accessed.

A table that is invalid because of missing privileges can automatically become valid and accessible if the required privileges are granted again. A table that is invalid because a dependent type was dropped can never be accessed again, and the only permissible action is to drop the table.

Because of the severe effects that revoking a privilege on a type or dropping a type can cause, the SQL statements REVOKE and DROP TYPE, by default, implement restricted semantics. This means that if the named type in either statement has table or type dependents, then an error is received and the statement cancels. However, if the FORCE clause for either statement is used, then the statement always succeeds. If there are depended-upon tables, then they are invalidated.

**See Also:** *Oracle Database Reference* for details about using the

REVOKE, DROP TYPE, and FORCE clauses

### Granting a User Privileges and Roles

This section contains:

* + - [Granting System Privileges and Roles](#_bookmark793)
    - [Granting Object Privileges](#_bookmark809)
    - [Granting Privileges on Columns](#_bookmark819)

It is also possible to grant roles to a user connected through a middle tier or proxy. This is discussed in ["Using a Middle Tier Server for Proxy Authentication" on](#_bookmark416) [page 3-36](#_bookmark416).

#### Granting SystemPrivileges and Roles

You can use the GRANT SQL statement to grant system privileges and roles to users and roles. The following privileges are required:

* + - To grant a system privilege, a user must be granted the system privilege with the

ADMIN option or must be granted the GRANT ANY PRIVILEGE system privilege.

* + - To grant a role, a user must be granted the role with the ADMIN option or was granted the GRANT ANY ROLE system privilege.

[Example 4–11](#_bookmark797) grants the system privilege CREATE SESSION and the accts\_pay role to the user jward.

***Example 4–11 Granting a System Privilege and a Role to a User***

GRANT CREATE SESSION, accts\_pay TO jward;

[Example 4–11](#_bookmark797) grants the EXECUTE privilege on the exec\_dir directory object to the user

jward.

***Example 4–12 Granting the EXECUTE Privilege on a Directory Object***

GRANT EXECUTE ON DIRECTORY exec\_dir TO jward;

**Note:** Object privileges cannot be granted along with system privileges and roles in the same GRANT statement.

###### Granting the ADMIN Option

If you specify the WITH ADMIN OPTION clause when you grant a privilege or role to a user or role, then the privilege grant has the following expanded capabilities:

* + The grantee can grant or revoke the system privilege or role to or from any other user or role in the database. Users cannot revoke a role from themselves.
  + The grantee can grant the system privilege or role with the ADMIN option.
  + The grantee of a role can alter or drop the role.

[Example 4–13](#_bookmark804) grants the new\_dba role with the WITH ADMIN OPTION clause to user

michael.

***Example 4–13 Granting the ADMIN Option***

GRANT new\_dba TO michael WITH ADMIN OPTION;

User michael is able to not only use all of the privileges implicit in the new\_dba role, but he can also grant, revoke, and drop the new\_dba role as deemed necessary. Because of these powerful capabilities, use caution when granting system privileges or roles with the ADMIN option. These privileges are usually reserved for a security administrator, and are rarely granted to other administrators or users of the system.

**Note:** When a user creates a role, the role is automatically granted to the creator with the ADMIN option.

###### Creating a New User with the GRANT Statement

Oracle Database enables you to create a new user with the GRANT statement. If you specify a password using the IDENTIFIED BY clause, and the user name does not exist in the database, then a new user with that user name and password is created.

[Example 4–14](#_bookmark808) creates psmith as a new user while granting psmith the CREATE SESSION

system privilege.

***Example 4–14 Creating a New User with the GRANT Statement***

GRANT CREATE SESSION TO psmith IDENTIFIED BY *password*;

**See Also:**

* + - ["Creating User Accounts" on page 2-1](#_bookmark60)
    - ["Minimum Requirements for Passwords" on page 3-3](#_bookmark197)

#### Granting Object Privileges

You can use the GRANT statement to grant object privileges to roles and users. To grant an object privilege, you must fulfill one of the following conditions:

* + You own the object specified.
  + You have been granted the GRANT ANY OBJECT PRIVILEGE system privilege. This privilege enables you to grant and revoke privileges on behalf of the object owner.
  + The WITH GRANT OPTION clause was specified when you were granted the object privilege.

**Note:** System privileges and roles cannot be granted along with object privileges in the same GRANT statement.

[Example 4–15](#_bookmark812) grants the SELECT, INSERT, and DELETE object privileges for all columns of the emp table to the users jfee and tsmith.

***Example 4–15 Granting Object Privileges to Users***

GRANT SELECT, INSERT, DELETE ON emp TO jfee, tsmith;

To grant all object privileges on the salary view to user jfee, use the ALL keyword as shown in the following example:

GRANT ALL ON salary TO jfee;

**Note:** A grantee cannot regrant access to objects unless the original grant included the GRANT OPTION. Thus in the example just given, jfee cannot use the GRANT statement to grant object privileges to anyone else.

###### Specifying the GRANT OPTION Clause

Specify the WITH GRANT OPTION clause with the GRANT statement to enable the grantee to grant the object privileges to other users. The user whose schema contains an object is automatically granted all associated object privileges with the GRANT OPTION. This special privilege allows the grantee several expanded privileges:

* + - The grantee can grant the object privilege to any user in the database, with or without the GRANT OPTION, and to any role in the database.
    - If both of the following conditions are true, then the grantee can create views on the table, and grant the corresponding privileges on the views to any user or role in the database:
      * The grantee receives object privileges for the table with the GRANT OPTION.
      * The grantee has the CREATE VIEW or CREATE ANY VIEW system privilege.

**Note:**The GRANT OPTION is not valid when granting an object privilege to a role. Oracle Database prevents the propagation of object privileges through roles so that grantees of a role cannot propagate object privileges received by means of roles.

###### Granting Object Privileges on Behalf of the Object Owner

The GRANT ANY OBJECT PRIVILEGE system privilege enables users to grant and revoke any object privilege on behalf of the object owner. This privilege provides a convenient means for database and application administrators to grant access to objects in any schema without requiring that they connect to the schema. Login credentials do not need to be maintained for schema owners who have this privilege, which reduces the number of connections required during configuration.

This system privilege is part of the Oracle Database supplied DBA role and is thus granted (with the ADMIN option) to any user connecting AS SYSDBA (user SYS). As with other system privileges, the GRANT ANY OBJECT PRIVILEGE system privilege can only be granted by a user who possesses the ADMIN option.

The *recorded* grantor of access rights to an object is either the object owner or the person exercising the GRANT ANY OBJECT PRIVILEGE system privilege. If the grantor with GRANT ANY OBJECT PRIVILEGE does *not* have the object privilege with the GRANT OPTION, then the object owner is shown as the grantor. Otherwise, when that grantor has the object privilege with the GRANT OPTION, then that grantor is recorded as the grantor of the grant.

**Note:** The audit record generated by the GRANT statement always shows the actual user who performed the grant.

For example, consider the following scenario. User adams possesses the GRANT ANY OBJECT PRIVILEGE system privilege. He does not possess any other grant privileges. He issues the following statement:

GRANT SELECT ON HR.EMPLOYEES TO blake WITH GRANT OPTION;

If you examine the DBA\_TAB\_PRIVS view, then you will see that hr is shown as the grantor of the privilege:

SELECT GRANTEE, GRANTOR, PRIVILEGE, GRANTABLE FROM DBA\_TAB\_PRIVS

WHERE TABLE\_NAME = 'EMPLOYEES' and OWNER = 'HR';

GRANTEE GRANTOR PRIVILEGE GRANTABLE

-------- ------- ----------- ---------- BLAKE HR SELECT YES

Now assume that user blake also has the GRANT ANY OBJECT PRIVILEGE system. He issues the following statement:

GRANT SELECT ON HR.EMPLOYEES TO clark;

In this case, when you query the DBA\_TAB\_PRIVS view again, you see that blake is shown as being the grantor of the privilege:

GRANTEE GRANTOR PRIVILEGE GRANTABLE

-------- -------- --------- ---------- BLAKE HR SELECT YES

CLARK BLAKE SELECT NO

This occurs because blake already possesses the SELECT privilege on HR.EMPLOYEES

with the GRANT OPTION.

**See Also:** ["Revoking Object Privileges on Behalf of the Object](#_bookmark830) [Owner" on page 4-42](#_bookmark830)

###### Granting Privileges on Columns

You can grant INSERT, UPDATE, or REFERENCES privileges on individual columns in a table.

**Caution:** Before granting a column-specific INSERT privilege, determine if the table contains any columns on which NOT NULL constraints are defined. Granting selective insert capability without including the NOT NULL columns prevents the user from inserting any rows into the table. To avoid this situation, ensure that each NOT NULL column can either be inserted into or has a non-NULL default value. Otherwise, the grantee will not be able to insert rows into the table and will receive an error.

The following statement grants the INSERT privilege on the acct\_no column of the

accounts table to user psmith:

GRANT INSERT (acct\_no) ON accounts TO psmith;

In the following example, object privilege for the ename and job columns of the emp

table are granted to the users jfee and tsmith:

GRANT INSERT(ename, job) ON emp TO jfee, tsmith;

###### Row-Level Access Control

You can also provide access control at the row level, that is, within objects, using Virtual Private Database (VPD) or Oracle Label Security (OLS).

**See Also:**

* + - * + [Chapter 7, "Using Oracle Virtual Private Database to Control Data Access"](#_bookmark1341)
        + ["Adding Policies for Column-Level Oracle Virtual Private Database" on page 7-8](#_bookmark1408)
        + *Oracle Label Security Administrator's Guide*

### Revoking Privileges and Roles froma User

This section contains:

* + - [Revoking System Privileges and Roles](#_bookmark823)
    - [Revoking Object Privileges](#_bookmark827)
    - [Cascading Effects of Revoking Privileges](#_bookmark837)

#### Revoking SystemPrivileges and Roles

You can revoke system privileges and roles using the SQL statement REVOKE. Any user with the ADMIN option for a system privilege or role can revoke the privilege or role from any other database user or role. The revoker does not have to be the user that originally granted the privilege or role. Users with GRANT ANY ROLE can revoke *any* role.

The following statement revokes the CREATE TABLE system privilege and the accts\_ rec role from user psmith:

REVOKE CREATE TABLE, accts\_rec FROM psmith;

**Note:** The ADMIN option for a system privilege or role cannot be selectively revoked. Instead, revoke the privilege or role, and then grant the privilege or role again but without the ADMIN option.

#### Revoking Object Privileges

To revoke an object privilege, you must fulfill one of the following conditions:

* + You previously granted the object privilege to the user or role.
  + You possess the GRANT ANY OBJECT PRIVILEGE system privilege that enables you to grant and revoke privileges on behalf of the object owner.

You can only revoke the privileges that you, the person who granted the privilege, directly authorized. You cannot revoke grants that were made by other users to whom you granted the GRANT OPTION. However, there is a cascading effect. If the object privileges of the user who granted the privilege are revoked, then the object privilege grants that were propagated using the GRANT OPTION are revoked as well.

Assuming you are the original grantor of the privilege, the following statement revokes the SELECT and INSERT privileges on the emp table from users jfee and psmith:

REVOKE SELECT, INSERT ON emp FROM jfee, psmith;

The following statement revokes all object privileges for the dept table that you originally granted to the human\_resource role:

REVOKE ALL ON dept FROM human\_resources;

**Note:** The GRANT OPTION for an object privilege cannot be selectively revoked. Instead, revoke the object privilege and then grant it again but without the GRANT OPTION. Users cannot revoke object privileges from themselves.

###### Revoking Object Privileges on Behalf of the Object Owner

The GRANT ANY OBJECT PRIVILEGE system privilege enables you to revoke any specified object privilege where the object owner is the grantor. This occurs when the object privilege is granted by the object owner, or on behalf of the owner by any user holding the GRANT ANY OBJECT PRIVILEGE system privilege.

In a situation where the object privilege was granted by both the owner of the object and the user executing the REVOKE statement (who has both the specific object privilege and the GRANT ANY OBJECT PRIVILEGE system privilege), Oracle Database only revokes the object privilege granted by the user issuing the REVOKE statement. This can be illustrated by continuing the example started in ["Granting Object Privileges on](#_bookmark816) [Behalf of the Object Owner" on page 4-39](#_bookmark816).

At this point, user blake granted the SELECT privilege on HR.EMPLOYEES to clark. Even though blake possesses the GRANT ANY OBJECT PRIVILEGE system privilege, he also holds the specific object privilege, thus this grant is attributed to him. Assume that user HR also grants the SELECT privilege on HR.EMPLOYEES to user clark. A query of the DBA\_TAB\_PRIVS view shows that the following grants are in effect for the HR.EMPLOYEES table:

GRANTEE GRANTOR PRIVILEGE GRANTABLE

-------- ------- ----------- ---------- BLAKE HR SELECT YES

CLARK BLAKE SELECT NO

CLARK HR SELECT NO

User blake now issues the following REVOKE statement:

REVOKE SELECT ON HR.EMPLOYEES FROM clark;

Only the object privilege for user clark granted by user blake is removed. The grant by the object owner, HR, remains.

GRANTEE GRANTOR PRIVILEGE GRANTABLE

-------- ------- ----------- ---------- BLAKE HR SELECT YES

CLARK HR SELECT NO

If blake issues the REVOKE statement again, then this time the effect is to remove the object privilege granted by adams (on behalf of HR), using the GRANT ANY OBEJCT PRIVILEGE system privilege.

**See Also:** ["Granting Object Privileges on Behalf of the Object](#_bookmark816) [Owner" on page 4-39](#_bookmark816)

###### Revoking Column-Selective Object Privileges

Although users can grant column-specific INSERT, UPDATE, and REFERENCES privileges for tables and views, they cannot selectively revoke column-specific privileges with a similar REVOKE statement. Instead, the grantor must first revoke the object privilege for all columns of a table or view, and then selectively repeat the grant of the

column-specific privileges that the grantor intends to keep in effect.

For example, assume that role human\_resources was granted the UPDATE privilege on the deptno and dname columns of the table dept. To revoke the UPDATE privilege on just the deptno column, issue the following two statements:

REVOKE UPDATE ON dept FROM human\_resources; GRANT UPDATE (dname) ON dept TO human\_resources;

The REVOKE statement revokes the UPDATE privilege on all columns of the dept table from the role human\_resources. The GRANT statement then repeats, restores, or reissues the grant of the UPDATE privilege on the dname column to the role human\_resources.

###### Revoking the REFERENCES Object Privilege

If the grantee of the REFERENCES object privilege has used the privilege to create a foreign key constraint (that currently exists), then the grantor can revoke the privilege only by specifying the CASCADE CONSTRAINTS option in the REVOKE statement:

REVOKE REFERENCES ON dept FROM jward CASCADE CONSTRAINTS;

Any foreign key constraints currently defined that use the revoked REFERENCES

privilege are dropped when the CASCADE CONSTRAINTS clause is specified.

#### Cascading Effects of Revoking Privileges

Depending on the type of privilege, there may be cascading effects when a privilege is revoked. This is discussed in the following sections:

* + - [Cascading Effects When Revoking System Privileges](#_bookmark839)
    - [Cascading Effects When Revoking Object Privileges](#_bookmark841)

###### Cascading Effects When Revoking System Privileges

There are no cascading effects when revoking a system privilege related to DDL operations, regardless of whether the privilege was granted with or without the ADMIN option. For example, assume the following:

1. The security administrator grants the CREATE TABLE system privilege to user jfee

with the ADMIN option.

1. User jfee creates a table.
2. User jfee grants the CREATE TABLE system privilege to user tsmith.
3. User tsmith creates a table.
4. The security administrator revokes the CREATE TABLE system privilege from user

jfee.

1. The table created by user jfee continues to exist. User tsmith still has the table and the CREATE TABLE system privilege.

You can observe cascading effects when you revoke a system privilege related to a DML operation. If the SELECT ANY TABLE privilege is revoked from a user, then all procedures contained in the user’s schema relying on this privilege can no longer be executed successfully until the privilege is reauthorized.

###### Cascading Effects When Revoking Object Privileges

Revoking an object privilege can have cascading effects. Remember the following:

* + **Object definitions that depend on a DML object privilege can be affected if the DML object privilege is revoked.** For example, assume that the body of the test procedure includes a SQL statement that queries data from the emp table. If the SELECT privilege on the emp table is revoked from the owner of the test procedure, then the procedure can no longer be executed successfully.
  + **When a REFERENCES privilege for a table is revoked from a user, any foreign key integrity constraints that are defined by the user and require the dropped REFERENCES privilege are automatically dropped.** For example, assume that user jward is granted the REFERENCES privilege for the deptno column of the dept table. This user now creates a foreign key on the deptno column in the emp table that references the deptno column of the dept table. If the REFERENCES privilege on the deptno column of the dept table is revoked, then the foreign key constraint on the deptno column of the emp table is dropped in the same operation.
  + **The object privilege grants propagated using the GRANT OPTION are revoked if the object privilege of a grantor is revoked.** For example, assume that user1 is granted the SELECT object privilege on the emp table with the GRANT OPTION, and grants the SELECT privilege on emp to user2. Subsequently, the SELECT privilege is revoked from user1. This REVOKE statement is also cascaded to user2. Any objects that depend on the revoked SELECT privilege of user1 and user2 can also be affected, as described earlier.

Object definitions that require the ALTER and INDEX DDL object privileges are not affected if the ALTER or INDEX object privilege is revoked. For example, if the INDEX privilege is revoked from a user that created an index on a table that belongs to another user, then the index continues to exist after the privilege is revoked.

### Granting to and Revoking fromthe PUBLIC Role

You can grant and revoke privileges and roles from the role PUBLIC. Because PUBLIC is accessible to every database user, all privileges and roles granted to PUBLIC are accessible to every database user.

Security administrators and database users should grant a privilege or role to PUBLIC only if every database user requires the privilege or role. This recommendation reinforces the general rule that, at any given time, each database user should have only the privileges required to accomplish the current group tasks successfully.

Revoking a privilege from PUBLIC can cause significant cascading effects. If any privilege related to a DML operation is revoked from PUBLIC (for example, SELECT ANY TABLE or UPDATE ON emp), then all procedures in the database, including functions and packages, must be *reauthorized* before they can be used again. Therefore, be careful when you grant and revoke DML-related privileges to or from PUBLIC.

**See Also:**

* + - Managing Object Dependencies in *Oracle Database Administrator's Guide* for more information about object dependencies
    - ["Guidelines for Securing Data" on page 10-10](#_bookmark2195)

### Granting Roles Using the Operating Systemor Network

This section contains:

* [About Granting Roles Using the Operating System or Network](#_bookmark848)
* [Using Operating System Role Identification](#_bookmark849)
* [Using Operating System Role Management](#_bookmark852)
* [Granting and Revoking Roles When OS\_ROLES Is Set to TRUE](#_bookmark854)
* [Enabling and Disabling Roles When OS\_ROLES Is Set to TRUE](#_bookmark856)
* [Using Network Connections with Operating System Role Management](#_bookmark859)

#### About Granting Roles Using the Operating Systemor Network

Instead of a security administrator explicitly granting and revoking database roles to and from users using GRANT and REVOKE statements, the operating system on which Oracle Database runs can grant roles to users at connect time. Roles can be administered using the operating system and passed to Oracle Database when a user creates a session. As part of this mechanism, the default roles of a user and the roles granted to a user with the ADMIN option can be identified. If the operating system is used to authorize users for roles, then all roles must be created in the database and privileges assigned to the role with GRANT statements.

Roles can also be granted through a network service.

The advantage of using the operating system to identify the database roles of a user is that privilege management for an Oracle database can be externalized. The security facilities offered by the operating system control user privileges. This option may offer advantages of centralizing security for several system activities, such as the following situation:

* MVS Oracle administrators want RACF groups to identify database user roles.
  + UNIX Oracle administrators want UNIX groups to identify database user roles.
  + VMS Oracle administrators want to use rights identifiers to identify database user roles.

The main disadvantage of using the operating system to identify the database roles of a user is that privilege management can only be performed at the role level. Individual privileges cannot be granted using the operating system, but they can still be granted inside the database using GRANT statements.

A second disadvantage of using this feature is that, by default, users cannot connect to the database through the shared server or any other network connection if the operating system is managing roles. However, you can change this default as described in ["Using Network Connections with Operating System Role Management"](#_bookmark859) [on page 4-47](#_bookmark859).

**Note:** The features described in this section are available only on some operating systems. See your operating system-specific Oracle Database documentation to determine if you can use these features.

#### Using Operating SystemRole Identification

To cause a database to use the operating system to identify the database roles of each user when a session is created, set the initialization parameter OS\_ROLES to TRUE (and restart the instance, if it is currently running). When a user tries to create a session with the database, Oracle Database initializes the user security domain using the database roles identified by the operating system.

To identify database roles for a user, the operating system account for each Oracle Database user must have operating system identifiers (these may be called groups, rights identifiers, or other similar names) that indicate which database roles are to be available for the user. Role specification can also indicate which roles are the default roles of a user and which roles are available with the ADMIN option. No matter which operating system is used, the role specification at the operating system level follows the format:

ora\_*ID*\_*ROLE*[[\_d][\_a][\_da]]

In this specification:

* + ID has a definition that varies on different operating systems. For example, on VMS, ID is the instance identifier of the database; on VMS, it is the computer type; and on UNIX, it is the system ID.

**Note:** ID is case-sensitive to match your ORACLE\_SID. ROLE is not case-sensitive.

* + ROLE is the name of the database role.
  + d is an optional character that indicates this role is to be a default role of the database user.
  + a is an optional character that indicates this role is to be granted to the user with the ADMIN option. This allows the user to grant the role to other roles only. Roles cannot be granted to users if the operating system is used to manage roles.

**Note:** If either the d or a character is specified, then precede that character by an underscore (\_).

For example, an operating system account might have the following roles identified in its profile:

ora\_PAYROLL\_ROLE1

ora\_PAYROLL\_ROLE2\_a ora\_PAYROLL\_ROLE3\_d ora\_PAYROLL\_ROLE4\_da

When the corresponding user connects to the payroll instance of Oracle Database, role3 and role4 are defaults, while role2 and role4 are available with the ADMIN option.

#### Using Operating SystemRole Management

When you use operating system-managed roles, remember that database roles are being granted to an operating system user. Any database user to which the operating system user is able to connect will have the authorized database roles enabled. For this reason, you should consider defining all Oracle Database users as IDENTIFIED EXTERNALLY if you are using OS\_ROLES = TRUE, so that the database accounts are tied to the operating system account that was granted privileges.

#### Granting and Revoking Roles When OS\_ROLES Is Set to TRUE

If the OS\_ROLES parameter is set to TRUE, then the operating system completely manages the granting and revoking of roles to users. Any previous granting of roles to users using GRANT statements do not apply. However, they are still listed in the data dictionary. Only the role grants to users made at the operating system level apply.

Users can still grant privileges to roles and users.

**Note:** If the operating system grants a role to a user with the

ADMIN option, then the user can grant the role only to other roles.

#### Enabling and Disabling Roles When OS\_ROLES Is Set to TRUE

If the OS\_ROLES initialization parameter is set to TRUE, then any role granted by the operating system can be dynamically enabled using the SET ROLE statement. This still applies, even if the role was defined to require a password or operating system authorization. However, any role not identified in the operating system account of a user cannot be specified in a SET ROLE statement, even if a role was granted using a GRANT statement when OS\_ROLES = FALSE. (If you specify such a role, then Oracle Database ignores it.)

When OS\_ROLES is set to TRUE, then the user can enable up to 148 roles. Remember that this number includes other roles that may have been granted to the role.

#### Using Network Connections with Operating SystemRole Management

If you have the operating system manage roles, then, by default, users cannot connect to the database through the shared server. This restriction is the default because a remote user could impersonate another operating system user over an unsecure connection.

If you are not concerned with this security risk and want to use operating system role management with the shared server, or any other network connection, then set the initialization parameter REMOTE\_OS\_ROLES to TRUE. The change takes effect the next time you start the instance and mount the database. The default setting of this parameter is FALSE.

### When Do Grants and Revokes Take Effect?

Depending on what is granted or revoked, a grant or revoke takes effect at different times:

* + All grants and revokes of system and object privileges to anything (users, roles, and PUBLIC) take immediate effect.
  + All grants and revokes of roles to anything (users, other roles, PUBLIC) take effect only when a current user session issues a SET ROLE statement to reenable the role after the grant and revoke, or when a new user session is created after the grant or revoke.

You can see which roles are currently enabled by examining the SESSION\_ROLES data dictionary view.

#### Howthe SET ROLE Statement Affects Grants and Revokes

During the user session, the user or an application can use the SET ROLE statement any number of times to change the roles currently enabled for the session. The user must already be granted the roles that are named in the SET ROLE statement.

[Example 4–16](#_bookmark868) enables the role clerk, which you have already been granted, and specifies the password.

***Example 4–16 Using SET ROLE to Grant a Role and Specify a Password***

SET ROLE clerk IDENTIFIED BY *password*;

Replace *password* with a password that is secure. ["Minimum Requirements for](#_bookmark197) [Passwords" on page 3-3](#_bookmark197) describes the minimum requirements for passwords.

[Example 4–17](#_bookmark869) shows how to use SET ROLE to disable all roles.

***Example 4–17 Using SET ROLE to Disable All Roles***

SET ROLE NONE;

#### Specifying a Default Role

When a user logs on, Oracle Database enables all privileges granted explicitly to the user and all privileges in the default roles of the user.

You can set and alter a list of default roles for a user by using the ALTER USER SQL statement. The ALTER USER statement specifies roles that are to be enabled when a user connects to the database. The user must have been directly granted the roles with a GRANT statement, or the roles must have been created by the user with the CREATE ROLE privilege. For information about the restrictions of the DEFAULT ROLE clause of the ALTER USER statement, see *Oracle Database SQL Language Reference*.

[Example 4–18](#_bookmark873) sets the default roles payclerk and pettycash for user jane:

***Example 4–18 Using ALTER USER to Set Default Roles***

ALTER USER jane DEFAULT ROLE payclerk, pettycash;

You cannot set default roles for a user in the CREATE USER statement. When you first create a user, the default user role setting is ALL, which causes all roles subsequently granted to the user to be default roles. Use the ALTER USER statement to limit the default user roles.

**Note:** When you create a role (other than a global role or an application role), it is granted implicitly to you, and your set of default roles is updated to include the new role. Be aware that only 148 roles can be enabled for a user session. When aggregate roles, such as the DBA role, are granted to a user, the roles granted to the role are included in the number of roles the user has. For example, if a role has 20 roles granted to it and you grant that role to the user, then the user now has 21 additional roles. Therefore, when you grant new roles to a user, use the DEFAULT ROLE clause of the ALTER USER statement to ensure that not too many roles are specified as that user's default roles.

#### The MaximumNumber of Roles That a User Can Enable

A user can enable no more than 148 roles.You can grant a user as many roles as you want, but you should restrict the number of roles granted to a user to the minimum roles the user needs. See ["Guidelines for Securing Roles" on page 10-6](#_bookmark2160) for additional guidelines on granting roles to users.

### Managing Fine-Grained Access in PL/SQL Packages and Types

You can configure user access control to external network services and wallets through the UTL\_TCP, UTL\_SMTP, UTL\_MAIL, UTL\_HTTP, and UTL\_INADDR PL/SQL packages, the DBMS\_LDAP PL/SQL package, and the HttpUriType type.

* + - **Configuring fine-grained access control for users and** **roles that need to access external network services from the database.** This way, specific groups of users can connect to one or more host computers, based on privileges that you grant them. Typically, you use this feature to control access to applications that run on specific host addresses.
    - **Configuring fine-grained access control to Oracle wallets to make HTTP requests that require password or client-certificate authentication.** This feature enables you to grant privileges to users who are using passwords and client certificates stored in Oracle wallets to access external protected HTTP resources through the UTL\_HTTP package. For example, you can configure applications to use the credentials stored in the wallets instead of hard-coding the credentials in the applications. For more information about how you can use wallets to store passwords and credentials, see *Oracle Database Advanced Security Administrator's Guide*.

This section contains:

* + - [About Fine-Grained Access Control to External Network Services](#_bookmark881)
    - [About Access Control to Wallets](#_bookmark885)
  + [Upgrading Applications That Depend on Packages That Use External Network Services](#_bookmark888)
  + [Creating an Access Control List for External Network Services](#_bookmark891)
  + [Configuring Access Control to a Wallet](#_bookmark900)
  + [Examples of Creating Access Control Lists](#_bookmark914)
  + [Specifying a Group of Network Host Computers](#_bookmark927)
  + [Precedence Order for a Host Computer in Multiple Access Control List Assignments](#_bookmark929)
  + [Precedence Order for a Host in Access Control List Assignments with Port Ranges](#_bookmark931)
  + [Checking Privilege Assignments That Affect User Access to a Network Host](#_bookmark933)
  + [Setting the Precedence of Multiple Users and Roles in One Access Control List](#_bookmark948)
  + [Finding Information About Access Control Lists Configured for User Access](#_bookmark951)

#### About Fine-Grained Access Control to External Network Services

To configure fine-grained access control to external network services, you create an access control list (ACL), which is stored in Oracle XML DB. You can create the access control list by using Oracle XML DB itself, or by using the DBMS\_NETWORK\_ACL\_ADMIN and DBMS\_NETWORK\_ACL\_UTILITY PL/SQL packages. This guide explains how to use these packages to create and manage the access control list. To create an access control list by using Oracle XML DB and for general conceptual information about access control lists, see *Oracle XML DB Developer's Guide*.

This feature enhances security for network connections because it restricts the external network hosts that a database user can connect to using the PL/SQL network utility packages UTL\_TCP, UTL\_SMTP, UTL\_MAIL, UTL\_HTTP, and UTL\_INADDR, the DBMS\_LDAP

PL/SQL package, and the HttpUriType type. Otherwise, an intruder who gained access to the database could maliciously attack the network, because, by default, the PL/SQL utility packages are created with the EXECUTE privilege granted to PUBLIC users. These PL/SQL network utility packages, and the DBMS\_NETWORK\_ACL\_ADMIN and DBMS\_NETWORK\_ACL\_UTILITY packages, support both IP Version 4 (IPv4) and IP Version 6 (IPv6) addresses. This guide explains how to manage access control to both versions. For detailed information about how the IPv4 and IPv6 notation works with Oracle Database, see *Oracle Database Net Services Administrator's Guide*.

**See Also:** ["Tutorial: Adding an Email Alert to a Fine-Grained Audit](#_bookmark1943) [Policy" on page 9-44](#_bookmark1943) for an example of configuring access control to external network services for email alerts

#### About Access Control to Wallets

When a user accesses Web pages that are protected by a remote Web server, the user can authenticate himself or herself by supplying the passwords and client certificates that are stored in an Oracle wallet. The Oracle wallet provides secure storage of user passwords and client certificates.

To configure access control to a wallet, you need the following components:

* + **An Oracle wallet.** You can create the wallet using the Oracle Database mkstore utility or Oracle Wallet Manager. The HTTP request will use the external password store or the client certificate in the wallet to authenticate the user
    - **An access control list to grant privileges to the user to use the wallet.** To create the access control list, you use the DBMS\_NETWORK\_ACL\_ADMIN PL/SQL package.
    - **A way to associate the wallet with the access control list.** To do so, use the DBMS\_ NETWORK\_ACL\_ADMIN PL/SQL package.

The use of wallets is beneficial because it provides secure storage of passwords and client certificates necessary to access protected Web pages.

**See Also:** ["Configuring Access Control to a Wallet" on page 4-55](#_bookmark900)

#### Upgrading Applications That Depend on Packages That Use External Network Services

If you have upgraded from a release before Oracle Database 11*g* Release 1 (11.1), and your applications depend on PL/SQL network utility packages UTL\_TCP, UTL\_SMTP, UTL\_MAIL, UTL\_HTTP, and UTL\_INADDR, the DBMS\_LDAP PL/SQL package, or the HttpUriType type, then the following error may occur when you try to run the application:

ORA-24247: network access denied by access control list (ACL)

Use the procedures in this section to reconfigure the network access for the application. See also *Oracle Database Upgrade Guide* for compatibility issues for applications that depend on the PL/SQL network utility packages. For detailed information about the network utility packages, see *Oracle Database PL/SQL Packages and Types Reference*.

#### Creating an Access Control List for External Network Services

When you create access control lists for network connections, you should create one access control list dedicated to a group of common users, for example, users who need access to a particular application that resides on a specific host computer. For ease of administration and for good system performance, do not create too many access control lists. Network hosts accessible to the same group of users should share the same access control list.

To create the access control list by using the DBMS\_NETWORK\_ACL\_ADMIN package, follow these steps:

* + - [Step 1: Create the Access Control List and Its Privilege Definitions](#_bookmark894)
    - [Step 2: Assign the Access Control List to One or More Network Hosts](#_bookmark897)

###### Step 1: Create the Access Control List and Its Privilege Definitions

Use the DBMS\_NETWORK\_ACL\_ADMIN.CREATE\_ACL procedure to create the content of the access control list. It contains a name of the access control list, a brief description, and privilege settings for one user or role that you want to associate with the access control list. In an access control list, privileges for each user or role are grouped together as an access control entry (ACE). An access control list must have the privilege settings for at least one user or role.

**Note:** You cannot import or export the access control list settings by using the Oracle Database import or export utilities such as Oracle Data Pump.

for example:

BEGIN

DBMS\_NETWORK\_ACL\_ADMIN.CREATE\_ACL (

acl => '*file\_name*.xml', description => '*file description*', principal => '*user\_or\_role*', is\_grant => TRUE|FALSE, privilege => 'connect|resolve',

start\_date => null|*timestamp\_with\_time\_zone*, end\_date => null|*timestamp\_with\_time\_zone*);

END;

In this specification:

* + acl: Enter a name for the access control list XML file. Oracle Database creates this file relative to the /sys/acls directory in the XML DB Repository in the database. Include the .xml extension. For example:

acl => 'us-example-com-permissions.xml',

* + description: Enter a brief description of the purpose of this file. For example:

description => 'Network connection permission for ACCT\_MGR role',

* + principal: Enter the first user account or role being granted or denied permissions. For example:

principal => 'ACCT\_MGR',

Enter the name of the user account or role in case sensitive characters. For example, if the database stores the role name ACCT\_MGR in all capital letters, entering it in mixed or lower case will not work. You can find the user accounts and roles in the current database instance by querying the DBA\_USERS and DBA\_ ROLES data dictionary views. Typically, user names and roles are stored in upper-case letters.

If you want to enter multiple users or grant additional privileges to this user or role, use the DBMS\_NETWORK\_ACL.ADD\_PRIVILEGE procedure (described next) after you have created this access control list XML file.

* + is\_grant: Enter either TRUE or FALSE, to indicate whether the privilege is to be granted or denied. For example:

is\_grant => TRUE,

* + privilege: Enter either connect or resolve. This setting is case sensitive, so always enter it in lowercase. For example:

privilege => 'connect',

The connect privilege grants the user permission to connect to a network service at an external host. The resolve privilege grants the user permission to resolve a network host name or an IP address.

A database user needs the connect privilege to an external network host computer if he or she is connecting using the UTL\_TCP, UTL\_SMTP, UTL\_MAIL, UTL\_HTTP, the DBMS\_LDAP package, and the HttpUriType type. To resolve the host name that was given a host IP address, or the IP address that was given a host name, with the UTL\_INADDR package, grant the database user the resolve privilege instead.

You can use the data dictionary views described in ["Finding Information About](#_bookmark951) [Access Control Lists Configured for User Access" on page 4-71](#_bookmark951) to find more information about existing privileges and network connections.

* + - start\_date: (Optional) Enter the start date for the access control entry (ACE), in TIMESTAMP WITH TIME ZONE format (YYYY-MM-DD HH:MI:SS.FF TZR). When specified, the access control entry will be valid only on or after the specified date. The default is null. For example, to set a start date of February 28, 2008, at 6:30

a.m. in San Francisco, California, U.S., which is in the Pacific time zone:

start\_date => '2008-02-28 06:30:00.00 US/Pacific',

The NLS\_TIMESTAMP\_FORMAT initialization parameter sets the default timestamp format. See *Oracle Database Reference* for more information.

* + - end\_date: (Optional) Enter the end date for the access control entry (ACE), in TIMESTAMP WITH TIME ZONE format (YYYY-MM-DD HH:MI:SS.FF TZR). When specified, the access control entry expires after the specified date. The end\_date setting must be greater than or equal to the start\_date setting. The default is null.

For example, to set an end date of December 10, 2008, at 11:59 p.m. in San Francisco, California, U.S., which is in the Pacific time zone:

end\_date => '2008-12-10 23:59:00.00 US/Pacific');

To add more users or roles to the access control list, or grant additional privileges to one user or role, use the DBMS\_NETWORK\_ACL.ADD\_PRIVILEGE procedure. The syntax is as follows:

BEGIN DBMS\_NETWORK\_ACL\_ADMIN.ADD\_PRIVILEGE (

acl => '*file\_name*.xml', principal => '*user\_or\_role*', is\_grant => TRUE|FALSE, privilege => 'connect|resolve', position => null|*value*,

start\_date => null|*timestamp\_with\_time\_zone*, end\_date => null|*timestamp\_with\_time\_zone*);

END;

As you can see, the parameters to add the privilege are the similar to those in the CREATE\_ACL procedure, except that description is not included and the position parameter, which sets the order of precedence for multiple users or roles, was added. Because you now are adding more than one user or role, you may want to consider setting their precedence. ["Setting the Precedence of Multiple Users and Roles in One](#_bookmark948) [Access Control List" on page 4-69](#_bookmark948) provides more information.

Other DBMS\_NETWORK\_ACL\_ADMIN procedures that are available for this step are DELETE\_ PRIVILEGE and DROP\_ACL.

At this stage, you have created an access control list that defines the privileges needed to connect to a network host. However, the access control list has no effect until you complete [Step 2: Assign the Access Control List to One or More Network Hosts](#_bookmark897).

###### Step 2: Assign the Access Control List to One or More Network Hosts

After you create the access control list, then you are ready to assign it to one or more network host computers. You can use the DBMS\_NETWORK\_ACL\_ADMIN.ASSIGN\_ACL procedure to do so.

For example:

BEGIN DBMS\_NETWORK\_ACL\_ADMIN.ASSIGN\_ACL (

acl => '*file\_name*.xml',

host => '*network\_host*', lower\_port => null|*port\_number*, upper\_port => null|*port\_number*);

END;

In this specification:

* + acl: Enter the name of the access control list XML file (from [Step 1: Create the Access Control List and Its Privilege Definitions](#_bookmark894)) to assign to the network host. Oracle Database creates this file relative to the /sys/acls directory in the XML DB Repository in the database. Include the .xml extension. For example:

acl => 'us-example-com-permissions.xml',

* + host: Enter the network host to which this access control list will be assigned. This setting can be a name or IP address of the network host. Host names are case insensitive. For example:

host => 'us.example.com',

If you specify localhost, and if the host name has not been specified with the UTL\_INADDR and UTL\_HTTP PL/SQL packages in situations in which the local host is assumed, then these packages will search for and use the ACL that has been assigned localhost for the host setting.

See the following sections for more information about how network host computers in access control list assignments work:

* ["Specifying a Group of Network Host Computers" on page 4-64](#_bookmark927)
* ["Checking Privilege Assignments That Affect User Access to a Network Host" on page 4-66](#_bookmark933)
* ["Precedence Order for a Host Computer in Multiple Access Control List Assignments" on page 4-65](#_bookmark929)
* ["Precedence Order for a Host in Access Control List Assignments with Port Ranges" on page 4-66](#_bookmark931)
  + lower\_port: (Optional) For TCP connections, enter the lower boundary of the port range. Use this setting for the connect privilege only; omit it for the resolve privilege. The default is null, which means that there is no port restriction (that is, the ACL applies to all ports). The range of port numbers is between 1 and 65535.

For example:

lower\_port => 80,

* + upper\_port: (Optional) For TCP connections, enter the upper boundary of the port range. Use this setting for connect privileges only; omit it for resolve privileges. The default is null, which means that there is no port restriction (that is, the ACL applies to all ports). The range of port numbers is between 1 and 65535

For example:

upper\_port => 3999);

If you enter a value for the lower\_port and leave the upper\_port at null (or just omit it), Oracle Database assumes the upper\_port setting is the same as the lower\_ port. For example, if you set lower\_port to 80 and omit upper\_port, the upper\_ port setting is assumed to be 80.

The resolve privilege in the access control list takes no effect when a port range is specified in the access control list assignment.

Only one access control list can be assigned to any host computer, domain, or IP subnet, and if specified, the TCP port range. When you assign a new access control list to a network target, Oracle Database unassigns the previous access control list that was assigned to the same target. However, Oracle Database does not drop the access control list. You can drop the access control list by using the DROP\_ACL procedure. To remove an access control list assignment, use the UNASSIGN\_ACL procedure.

Depending on how you create and maintain the access control list, the two steps may overlap. For example, you can create an access control list that has privileges for five users in it, and then apply it to two host computers. Later on, you can modify this access control list to have different or additional users and privileges, and assign it to different or additional host computers.

All access control list changes, including the assignment to network hosts, are transactional. They do not take effect until the transaction is committed.

You can find information about existing privileges and network connections by using the data dictionary views described in [Table 4–6, " Data Dictionary Views That Display](#_bookmark955) [Information about Access Control Lists" on page 4-71](#_bookmark955).

For information about using the DBMS\_NETWORK\_ACL\_ADMIN package, see *Oracle Database PL/SQL Packages and Types Reference*.

#### Configuring Access Control toa Wallet

This method lets you grant access to the passwords and client certificates that are stored in an Oracle wallet to users to authenticate themselves to an external Web server. This enables the user to retrieve protected Web pages from the Web server.

This section contains:

* + - [Step 1: Create an Oracle Wallet](#_bookmark910)
    - [Step 2: Create an Access Control List that Grants the Wallet Privileges](#_bookmark911)
    - [Step 3: Assign the Access Control List to the Wallet](#_bookmark912)
    - [Step 4: Make the HTTP Request with the Passwords and Client Certificates](#_bookmark913)

###### Step 1: Create an Oracle Wallet

To create the wallet, you can use either the mkstore command-line utility or the Oracle Wallet Manager user interface. To store passwords in the wallet, you must use mkstore. You can use both standard and PKCS11 wallet types, and the wallet can be an auto-login wallet if you want. For detailed information about creating wallets, see *Oracle Database Advanced Security Administrator's Guide*.

When you create the wallet, do the following:

* + - Ensure that you have exported the wallet to a file.
    - Make a note of the directory in which you created the wallet. You will need this directory path when you complete the procedures in this section.

**See Also:**

* ["Example of an Access Control List for Using Passwords in a Non-Shared Wallet" on page 4-62](#_bookmark923)
* ["Example of an Access Control List for Wallets in a Shared Database Session" on page 4-64](#_bookmark925)

###### Step 2: Create an Access Control List that Grants the Wallet Privileges

After you have created the wallet, you are ready to create the access control list that will assign the password or client certificate privilege the user needs to use password credentials in the wallet for HTTP authentication.

For example:

BEGIN DBMS\_NETWORK\_ACL\_ADMIN.CREATE\_ACL (

acl => '*file\_name*.xml', description => '*description*', principal => '*user\_or\_role*', is\_grant => TRUE|FALSE, privilege => '*privilege*';

... END;

In this specification:

* + acl: Enter a name for the ACL, and make a note of this name. You will need this name in [Step 3: Assign the Access Control List to the Wallet](#_bookmark912), next. Oracle Database creates this file relative to the /sys/acls directory in the XML DB Repository in the database. Include the .xml extension. For example:

acl => 'hr\_access\_wallet\_acl.xml',

* + description: Enter a brief description of the purpose of this file. For example:

description => 'Wallet ACL for the hr\_access application',

* + principal: Enter the user account or role being granted or denied privileges. For example:

principal => 'HR\_CLERK',

Enter this name using case sensitive characters. For example, if the database stores the role name HR\_CLERK in all capital letters, entering it in mixed or lower-case letters will not work. You can find the user accounts and roles in the current database instance by querying the DBA\_USERS and DBA\_ROLES data dictionary views. Typically, user names and roles are stored in upper-case letters.

If you want to add multiple users, or if you want to grant this user an additional privilege, you can use the DBMS\_NETWORK\_ACL.ADD\_PRIVILEGE procedure after you have created this access control list XML file.

* + is\_grant: Enter either TRUE or FALSE, to indicate whether the privilege is to be granted or denied. For example:

is\_grant => TRUE,

* + privilege: Enter one of the following settings using lowercase letters and hyphens. Remember that the privilege name is case-sensitive.
* use-passwords to give the user permission to use passwords in the wallet

**–** use-client-certificates to authenticate the user with a client certificate in the wallet

For example:

privilege => 'use-client-certificates');

###### Step 3: Assign the Access Control List to the Wallet

In this step, you assign this access control list to the wallet you created earlier. Afterward, you can check your settings by querying the DBA\_WALLET\_ACLS data dictionary view.

For example:

BEGIN

...

DBMS\_NETWORK\_ACL\_ADMIN.ASSIGN\_WALLET\_ACL (

acl => '*file\_name*.xml',

wallet\_path => 'file:*path\_to\_directory\_containing\_wallet*'); END;

In this specification:

* + acl: Enter the name that you created for this wallet in [Step 2: Create an Access Control List that Grants the Wallet Privileges](#_bookmark911), in the previous section. For example:

acl => 'hr\_access\_wallet\_acl.xml',

* + wallet\_path: Enter the path to the directory that contains the wallet. When you specify the wallet path, you must use an absolute path and include file: before this directory path. Do not use environment variables, such as $ORACLE\_HOME, nor insert a space after file: and before the path name. For example:

wallet\_path => 'file:/oracle/wallets/hr\_access\_access'

###### Step 4: Make the HTTP Request with the Passwords and Client Certificates

In this step, you use the UTL\_HTTP PL/SQL package to create a request context object that is used privately with the HTTP request and its response. For detailed information about the UTL\_HTTP package, see *Oracle Database PL/SQL Packages and Types Reference*.

For example:

DECLARE

req\_context UTL\_HTTP.REQUEST\_CONTEXT\_KEY; req UTL\_HTTP.REQ;

BEGIN

req\_context := UTL\_HTTP.CREATE\_REQUEST\_CONTEXT (

wallet\_path => 'file:*path\_to\_directory\_containing\_wallet*', wallet\_password => '*wallet\_password*'|NULL);

req := UTL\_HTTP.BEGIN\_REQUEST(

url => '*URL\_to\_application*', request\_context => '*request\_context*'|NULL);

... END;

In this specification:

* + req\_context: Use the UTL\_HTTP.CREATE\_REQUEST\_CONTEXT\_KEY datatype to create the request context object. This object stores a randomly-generated numeric key that Oracle Database uses to identify the request context. The UTL\_HTTP.CREATE\_ REQUEST\_CONTEXT function creates the request context itself.
  + req: Use the UTL\_HTTP.REQ datatype to create the object that will be used to begin the HTTP request. You will refer to this object later on, when you set the user name and password from the wallet to access a password-protected Web page.
  + wallet\_path: Enter the path to the directory that contains the wallet. Ensure that this path is the same path you specified when you created access control list in [Step 3: Assign the Access Control List to the Wallet](#_bookmark912) in the previous section.You must include file: before the directory path. Do not use environment variables, such as $ORACLE\_HOME.

For example:

wallet\_path => 'file:/oracle/wallets/hr\_access\_access',

* + wallet\_password: Enter the password used to open the wallet. The default is NULL, which is used for auto-login wallets. For example:

wallet\_password => NULL);

* + url: Enter the URL to the application that uses the wallet. For example:

url => 'www.hr\_access.example.com',

* + request\_context: Enter the name of the request context object that you created earlier in this section. This object prevents the wallet from being shared with other applications in the same database session.

For example:

request\_context => req\_context);

Using a Request Context to Hold the Wallet When Sharing the Session with Other Applications

You should use a request context to hold the wallet when the database session is shared with other applications. If your application has exclusive use of the database session, you can hold the wallet in the database session by using the SET\_WALLET procedure instead.

For example:

DECLARE

req UTL\_HTTP.REQ; BEGIN UTL\_HTTP.SET\_WALLET(

path => 'file:*path\_to\_directory\_containing\_wallet*', password => '*wallet\_password*'|NULL);

req := UTL\_HTTP.BEGIN\_REQUEST(

url => '*URL\_to\_application*');

... END;

If the protected URL being requested requires the user name and password to authenticate, then use the SET\_AUTHENTICATION\_FROM\_WALLET procedure to set the user name and password from the wallet to authenticate.

Using Only a Client Certificate to Authenticate

If the protected URL being requested requires only the client certificate to authenticate, the BEGIN\_REQUEST function sends the necessary client certificate from the wallet. assuming the user has been granted the use-client-certificates privilege in the

ACL assigned to the wallet. The authentication should succeed at the remote Web server and the user can proceed to retrieve the HTTP response by using the GET\_ RESPONSE function.

Using the Password to Authenticate

If the protected URL being requested requires the username and password to authenticate, you should use the SET\_AUTHENTICATION\_FROM\_WALLET procedure to set the username and password from the wallet to authenticate.

For example:

DECLARE

req\_context UTL\_HTTP.REQUEST\_CONTEXT\_KEY; req UTL\_HTTP.REQ;

BEGIN

...

UTL\_HTTP.SET\_AUTHENTICATION\_FROM\_WALLET( r => *HTTP\_REQUEST*,

alias => '*alias\_to\_retrieve\_credentials\_stored\_in\_wallet*', scheme => 'AWS|Basic',

for\_proxy => TRUE|FALSE); END;

In this specification:

* + - r: Enter the HTTP request defined in the UTL\_HTTP.BEGIN\_REQUEST procedure that you created above, in the previous section. For example:

r => req,

* + - alias: Enter the alias used to identify and retrieve the user name and password credential stored in the Oracle wallet. For example, assuming the alias used to identify this user name and password credential is hr\_access.

alias => 'hr\_access',

* + - scheme: Enter one of the following:
      * AWS: Specifies the Amazon Simple Storage Service (S3) scheme. Use this scheme only if you are configuring access to the Amazon.com Web site. (Contact Amazon for more information about this setting.)
      * Basic: Specifies HTTP basic authentication. The default is Basic. For example:

scheme => 'Basic',

* + - for\_proxy: Specify whether the HTTP authentication information is for access to the HTTP proxy server instead of the Web server. The default is FALSE.

For example:

for\_proxy => TRUE);

The use of the user name and password in the wallet requires the use-passwords

privilege to be granted to the user in the ACL assigned to the wallet.

#### Examples of Creating Access Control Lists

The following examples demonstrate how to create access control lists.

* + - [Example of an Access Control List for a Single Role and Network Connection](#_bookmark917)
  + [Example of an Access Control List with Multiple Roles Assigned to Multiple Hosts](#_bookmark919)
  + [Example of an Access Control List for Using Passwords in a Non-Shared Wallet](#_bookmark923)
  + [Example of an Access Control List for Wallets in a Shared Database Session](#_bookmark925)

**See Also:** *Oracle Database Vault Administrator's Guide* for a tutorial that demonstrates how to use an access control list when an administrator must use the UTL\_MAIL PL/SQL package to configure an email alert

###### Example of an Access Control List for a Single Role and Network Connection

[Example 4–19](#_bookmark918) shows how you would create an access control list called

us-example-com-permissions.xml to grant users who have the ACCT\_MGR role access to network services that run on the host us.example.com.

***Example 4–19 Creating an Access Control List for a Single Role and Network Connection***

**-- 1. Create the access control list, which includes one role:**

BEGIN DBMS\_NETWORK\_ACL\_ADMIN.CREATE\_ACL (

acl => 'us-example-com-permissions.xml',

description => 'Network connection permission for ACCT\_MGR', principal => 'ACCT\_MGR', *-- Must be in upper case* is\_grant => TRUE,

privilege => 'connect'); END;

/

**-- 2. Assign the access control list a network host:**

BEGIN DBMS\_NETWORK\_ACL\_ADMIN.ASSIGN\_ACL (

acl => 'us-example-com-permissions.xml', host => 'www.us.example.com',

lower\_port => 80,

upper\_port => 80); END;

/

This example creates the us-example-com-permissions.xml file in the /sys/acls

directory, which is the default location. The XML file appears as follows:

<acl description="Network connection permission for ACCT\_MGR" xmlns=["http://xmlns.oracle.com/xdb/acl.xsd"](http://xmlns.oracle.com/xdb/acl.xsd) xmlns:plsql[="http://xmlns.oracle.com/plsql"](http://xmlns.oracle.com/plsql) xmlns:xsi="[http://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance) xsi:schemaLocation="<http://xmlns.oracle.com/xdb/acl.xsd>

[http://xmlns.oracle.com/xdb/acl.xsd">](http://xmlns.oracle.com/xdb/acl.xsd)

<security-class>plsql:network</security-class>

<ace>

<grant>true</grant>

<principal>ACCT\_MGR</principal>

<privilege><plsql:connect/></privilege>

</ace>

</acl>

The xmlns and xsi elements are fixed and should not be modified, for example, in a text editor.

You can check the contents of the access control list in SQL\*Plus. See *Oracle XML DB Developer's Guide* for examples.

###### Example of an Access Control List with Multiple Roles Assigned to Multiple Hosts

[Example 4–20](#_bookmark920) shows how to create a slightly more complex version of the

us-example-com-permissions.xml access control list. In this example, you specify multiple role privileges and their precedence position, and assigned to multiple host computers.

See["Specifying a Group of Network Host Computers" on page 4-64](#_bookmark927) and ["Precedence](#_bookmark929) [Order for a Host Computer in Multiple Access Control List Assignments" on page 4-65](#_bookmark929) for more information about host names. See also ["Setting the Precedence of Multiple](#_bookmark948) [Users and Roles in One Access Control List" on page 4-69](#_bookmark948) to determine the order of multiple ACE elements in the access control list XML file.

***Example 4–20 Creating an Access Control List for Multiple Roles and Network Connections***

**-- 1. Create the access control list:** BEGIN DBMS\_NETWORK\_ACL\_ADMIN.CREATE\_ACL (

acl => 'us-example-com-permissions.xml',

description => 'Network connection permission for ACCT\_MGR and ACCT\_CLERK', principal => 'ACCT\_MGR', *-- Must be in upper case*

is\_grant => TRUE, privilege => 'resolve');

DBMS\_NETWORK\_ACL\_ADMIN.ADD\_PRIVILEGE ( *-- Creates the second role privilege*

acl => 'us-example-com-permissions.xml', principal => 'ACCT\_CLERK',

is\_grant => TRUE, privilege => 'connect', position => null);

END;

/

**-- 2. Assign the access control list to hosts:**

BEGIN

DBMS\_NETWORK\_ACL\_ADMIN.ASSIGN\_ACL ( *-- Creates the first target host*

acl => 'us-example-com-permissions.xml',

host => '\*.us.example.com'); DBMS\_NETWORK\_ACL\_ADMIN.ASSIGN\_ACL ( *-- Creates the second target host* acl => 'us-example-com-permissions.xml',

host => '\*.uk.example.com', lower\_port => 80,

upper\_port => 99); END;

/

The us-example-com-permissions.xml appears as follows:

<acl description="Network connection permission for ACCT\_MGR and ACCT\_CLERK" xmlns=["http://xmlns.oracle.com/xdb/acl.xsd"](http://xmlns.oracle.com/xdb/acl.xsd) xmlns:plsql[="http://xmlns.oracle.com/plsql"](http://xmlns.oracle.com/plsql) xmlns:xsi="[http://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance) xsi:schemaLocation="<http://xmlns.oracle.com/xdb/acl.xsd>

[http://xmlns.oracle.com/xdb/acl.xsd"](http://xmlns.oracle.com/xdb/acl.xsd)>

<security-class>plsql:network</security-class>

<ace>

<grant>true</grant>

<principal>ACCT\_MGR</principal>

<privilege><plsql:resolve/></privilege>

</ace>

<ace>

<grant>true</grant>

<principal>ACCT\_CLERK</principal>

<privilege><plsql:connect/></privilege>

</ace>

</acl>

[Example 4–21](#_bookmark921) shows how the DBA\_NETWORK\_ACL\_PRIVILEGES data dictionary view displays the privilege granted in the previous access control list.

***Example 4–21 Using the DBA\_NETWORK\_ACL\_PRIVILEGES View to Show Granted Privileges***

ACL

ACLID PRINCIPAL PRIVILEGE IS\_GRANT INVERT START\_DATE END\_DATE

------------------------------------------

-------------------------------- ---------- ------- -------- -------

---------- ----------

/sys/acls/us-example-com-permissions.xml 2EF86135D0E29B2AE040578CE4043250 ACCT\_ MGR resolve true false

/sys/acls/us-example-com-permissions.xml 2EF86135D0E29B2AE040578CE4043250 ACCT\_ CLERK connect true false

[Example 4–22](#_bookmark922) shows how the DBA\_NETWORK\_ACLS data dictionary view displays the host assignment of the access control list.

***Example 4–22 Using the DBA\_NETWORK\_ACLS View to Show Host Assignments***

HOST LOWER\_PORT UPPER\_PORT ACL

ACLID

-------------------- ---------- ----------

------------------------------------------

--------------------------------

\*.us.example.com

/sys/acls/us-example-com-permissions.xml 2EF86135D0E29B2AE040578CE4043250

\*.uk.example.com 80 99

/sys/acls/us-example-com-permissions.xml 2EF86135D0E29B2AE040578CE4043250

In these examples, the ACCT\_MGR role has the resolve privilege to the first host, and the ACCT\_CLERK role has the connect privilege to the first and second target hosts. The ACCT\_MGR role does not have the resolve privilege to the second host because a port range is specified in the assignment to the second host.

To check the contents of the access control list in SQL\*Plus, see *Oracle XML DB Developer's Guide* for examples.

###### Example of an Access Control List for Using Passwords in a Non-Shared Wallet

[Example 4–23](#_bookmark924) configures wallet access for two Human Resources department roles, hr\_clerk and hr\_manager. These roles use the use-passwords privilege to access passwords stored in the wallet. In this example, the wallet will not be shared with other applications within the same database session.

***Example 4–23 Configuring ACL Access Using Passwords in a Non-Shared Wallet***

**/\* 1. At a command prompt, create the wallet. The following example uses the user name hr\_access as the alias to identify the user name and password stored in the wallet. You must use this alias name when you call the SET\_AUTHENTICATION\_FROM\_WALLET procedure later on. \*/**

$ mkstore -wrl $ORACLE\_HOME/wallets/hr\_access\_access -create Enter password: *password*

Enter password again: *password*

$ mkstore -wrl $ORACLE\_HOME/wallets/hr\_access\_access -createCredential hr\_access hr\_usr

Your secret/Password is missing in the command line Enter your secret/Password: *password*

Re-enter your secret/Password: *password*

Enter wallet password: *password*

**/\* 2. In SQL\*Plus, create an access control list to grant privileges for the wallet. The following example grants the use-passwords privilege to the hr\_clerk and hr\_manager roles, and then it assigns this ACL to the wallet.\*/**

BEGIN

DBMS\_NETWORK\_ACL\_ADMIN.CREATE\_ACL(

acl => 'hr\_access\_wallet\_acl.xml',

description => 'Wallet ACL for hr\_access application', principal => 'HR\_CLERK', *-- Must be in upper case* is\_grant => TRUE,

privilege => 'use-passwords');

DBMS\_NETWORK\_ACL\_ADMIN.ADD\_PRIVILEGE(

acl => 'hr\_access\_wallet\_acl.xml', principal => 'HR\_MANAGER',

is\_grant => TRUE,

privilege => 'use-passwords');

DBMS\_NETWORK\_ACL\_ADMIN.ASSIGN\_WALLET\_ACL(

acl => 'hr\_access\_wallet\_acl.xml',

wallet\_path => 'file:/oracle/wallets/hr\_access\_access'); END;

/ COMMIT;

**/\* 3. Create a request context and request object, and then set the authentication for the wallet. \*/**

DECLARE

req\_context UTL\_HTTP.REQUEST\_CONTEXT\_KEY; req UTL\_HTTP.REQ;

BEGIN

req\_context := UTL\_HTTP.CREATE\_REQUEST\_CONTEXT(

wallet\_path => 'file:/oracle/wallets/hr\_access\_access', wallet\_password => NULL,

enable\_cookies => TRUE,

max\_cookies => 300,

max\_cookies\_per\_site => 20); req := UTL\_HTTP.BEGIN\_REQUEST(

url => 'www.hr\_access.example.com', request\_context => req\_context);

UTL\_HTTP.SET\_AUTHENTICATION\_FROM\_WALLET(

r => req,

alias => 'hr\_access'),

scheme => 'Basic',

for\_proxy => FALSE);

END;

/

###### Example of an Access Control List for Wallets in a Shared Database Session

[Example 4–24](#_bookmark926) is almost the same as [Example 4–23](#_bookmark924), except that it configures the wallet to be used for a shared database session; that is, all applications within the current database session will have access to this wallet.

***Example 4–24 Configuring ACL Access for a Wallet in a Shared Database Session***

**/\* Follow these steps:**

* 1. **Use Oracle Wallet Manager to create the wallet and add the client certificate. See *Oracle Database Advanced Security Administrator's Guide* for detailed information about using Oracle Wallet Manager.**
  2. **In SQL\*Plus, create an access control list to grant privileges for the wallet. The following example grants the use-client-certificates privilege to the hr\_clerk and hr\_manager roles, then it assigns this ACL to the wallet. \*/**

BEGIN

DBMS\_NETWORK\_ACL\_ADMIN.CREATE\_ACL(

acl => 'hr\_access\_wallet\_acl.xml',

description => 'Wallet ACL for hr\_access application', principal => 'HR\_CLERK', *-- Must be in upper case* is\_grant => TRUE,

privilege => 'use-client-certificates');

DBMS\_NETWORK\_ACL\_ADMIN.ADD\_PRIVILEGE(

acl => 'hr\_access\_wallet\_acl.xml', principal => 'HR\_MANAGER',

is\_grant => TRUE,

privilege => 'use-client-certificates');

DBMS\_NETWORK\_ACL\_ADMIN.ASSIGN\_WALLET\_ACL(

acl => 'hr\_access\_wallet\_acl.xml',

wallet\_path => 'file:/oracle/wallets/hr\_access\_access');

END;

/ COMMIT;

**/\* 3. Create a request object to handle the HTTP authentication for the wallet.\*/**

DECLARE

req UTL\_HTTP.req;

BEGIN

UTL\_HTTP.SET\_WALLET(

path => 'file: $ORACLE\_HOME/wallets/hr\_access\_access', password => NULL);

req := UTL\_HTTP.BEGIN\_REQUEST(

url => 'www.hr\_access.example.com',

method => 'POST', http\_version => NULL, request\_context => NULL);

END;

/

#### Specifying a Group of Network Host Computers

If you want to assign an access control list to a group of network host computers, you can use the asterisk (\*) wildcard character. For example, enter \*.example.com for host

computers that belong to a domain or 192.0.2.\* for IPv4 addresses that belong to an IP subnet. The asterisk wildcard must be at the beginning, before a period (.) in a domain, or at the end, after a period (.), in an IP subnet. For example, \*.example.com is valid, but \*example.com and \*.example.\* are not. Be aware that the use of wildcard characters affects the order of precedence for multiple access control lists that are assigned to the same host computer. You cannot use wildcard characters for IPv6 addresses.

The Classless Inter-Domain Routing ([**CIDR**](#_bookmark2334)) notation defines how IPv4 and IPv6 addresses are categorized for routing IP packets on the internet. The DBMS\_NETWORK\_ ACL\_ADMIN package supports CIDR notation for both IPv4 and IPv6 addresses. This package considers an IPv4-mapped IPv6 address or subnet equivalent to the

IPv4-native address or subnet it represents. For example, ::ffff:192.0.2.1 is equivalent to 192.0.2.1, and ::ffff:192.0.2.1/120 is equivalent to 192.0.2.\*.

#### Precedence Order for a Host Computer in Multiple Access Control List Assignments

For multiple access control lists that are assigned to the host computer and its domains, the access control list that is assigned to the host computer takes precedence over those assigned to the domains. The access control list assigned to a domain has a lower precedence than those assigned to the subdomains.

For example, Oracle Database first selects the access control list assigned to the host server.us.example.com, ahead of other access control lists assigned to its domains. If additional access control lists were assigned to the sub domains, their order of precedence is as follows:

* + 1. server.us.example.com
    2. \*.us.example.com
    3. \*.example.com
    4. \*.com

**5.** \*

Similarly, for multiple access control lists that are assigned to the IP address (both IPv4 and IPv6) and the subnets it belongs to, the access control list that is assigned to the IP address takes precedence over those assigned to the subnets. The access control list assigned to a subnet has a lower precedence than those assigned to the smaller subnets it contains.

For example, Oracle Database first selects the access control list assigned to the IP address 192.0.2.3, ahead of other access control lists assigned to the subnets it belongs to. If additional access control lists were assigned to the subnets, their order of precedence is as follows:

**1.** 192.0.2.3 (or ::ffff:192.0.2.3)

**2.** 192.0.2.3/31 (or ::ffff:192.0.2.3/127)

**3.** 192.0.2.3/30 (or ::ffff:192.0.2.3/126)

**4.** 192.0.2.3/29 (or ::ffff:192.0.2.3/125)

**5.** ...

**6.** 192.0.2.3/24 (or ::ffff:192.0.2.3/120 or 192.0.2.\*)

**7.** ...

**8.** 192.0.2.3/16 (or ::ffff:192.0.2.3/112 or 192.0.\*)

**9.** ...

**10.** 192.0.2.3/8 (or ::ffff:192.0.2.3/104 or 192.\*)

**11.** ...

**12.** ::ffff:192.0.2.3/95

**13.** ::ffff:192.0.2.3/94

**14.** ...

**15.** \*

#### Precedence Order for a Host in Access Control List Assignments with Port Ranges

When an access control list is assigned to a host computer, a domain, or an IP subnet with a port range, it takes precedence over the access control list assigned to the same host, domain, or IP subnet without a port range.

For example, for TCP connections to any port between port 80 and 99 at server.us.example.com, Oracle Database first selects the access control list assigned to port 80 through 99 at server.us.example.com, ahead of the other access control list assigned to server.us.example.com that is without a port range.

#### Checking Privilege Assignments That Affect User Access toa Network Host

Database administrators can use the DBA\_NETWORK\_ACL\_PRIVILEGES data dictionary view to query network privileges that have been granted to or denied from database users and roles in the access control lists, and whether those privileges take effect during certain times only. Using the information provided by the view, you may need to combine the data to determine if a user is granted the privilege at the current time, the roles the user has, the order of the access control entries, and so on. To simplify this privilege evaluation, you can use the following DBMS\_NETWORK\_ACL\_ADMIN functions to check the privilege granted to a user in an access control list:

* + CHECK\_PRIVILEGE: Checks if the specified privilege is granted to or denied from the specified user in an access control list. This procedure identifies the access control list by its path in the XML DB Repository. Use CHECK\_PRIVILEGE if you want to evaluate a single access control list with a known path.
  + CHECK\_PRIVILEGE\_ACLID: Similar to the CHECK\_PRIVILEGE procedure, except that it enables you to specify the object ID of the access control list. Use CHECK\_ PRIVILEGE\_ACLID if you need to evaluate multiple access control lists, when you query the DBA\_NETWORK\_ACLS data dictionary view. For better performance, call CHECK\_PRIVILEGE\_ACLID on multiple access control lists rather than using CHECK\_ PRIVILEGE on each one individually.

Users without database administrator privileges do not have the privilege to access the access control lists or to invoke those DBMS\_NETWORK\_ACL\_ADMIN functions. However, they can query the USER\_NETWORK\_ACL\_PRIVILEGES data dictionary view to check their privileges instead.

Database administrators and users can use the following DBMS\_NETWORK\_ACL\_UTILITY functions to determine if two hosts, domains, or subnets are equivalent, or if a host, domain, or subnet is equal to or contained in another host, domain, or subnet:

* + EQUALS\_HOST: Returns a value to indicate if two hosts, domains, or subnets are equivalent
    - CONTAINS\_HOST: Returns a value to indicate if a host, domain, or subnet is equal to or contained in another host, domain, or subnet, and the relative order of precedence of the containing domain or subnet for its ACL assignments

If you do not use IPv6 addresses, database administrators and users can use the following DBMS\_NETWORK\_ACL\_UTILITY functions to generate the list of domains or IPv4 subnet a host belongs to and to sort the access control lists by their order of precedence according to their host assignments:

* + - DOMAINS: Returns a list of the domains or IP subnets whose access control lists may affect permissions to a specified network host, subdomain, or IP subnet
    - DOMAIN\_LEVEL: Returns the domain level of a given host

The following sections explain how database administrators and users can check permissions for the user to connect to a network host or to perform domain name resolutions:

* + - [How a DBA Can Check User Network Connection and Domain Privileges](#_bookmark936)
    - [How Users Can Check Their Network Connection and Domain Privileges](#_bookmark942)

###### How a DBA Can Check User Network Connection and Domain Privileges

A database administrator can query the DBA\_NETWORK\_ACLS view to determine which access control lists are present for a specified host computer. This view shows the access control lists that determine the access to the network connection or domain, and then determines if each access control list grants (GRANTED), denies (DENIED), or does not apply (NULL) to the access privilege of the user. Only the database administrator can query this view.

The following sections provide examples that demonstrate how the database administrator can check user privileges for network connections and domain name resolution.

* + - [Database Administrator Checking User Connection Privileges](#_bookmark938)
    - [Database Administrator Checking User Privileges for Domain Name Resolution](#_bookmark940)

Database Administrator Checking User Connection Privileges

[Example 4–25](#_bookmark939) shows how a database administrator can check the privileges for user preston to connect to [www.us.example.com.](http://www.us.example.com/) Remember that the user name you enter for the user parameter in the CHECK\_PRIVILEGE\_ACLID procedure is case sensitive. In this example, entering the user name preston is correct, but entering Preston or preston is incorrect.

You can find the users in the current database instance by querying the DBA\_USERS data dictionary view, for example:

SELECT USERNAME FROM DBA\_USERS;

***Example 4–25 Administrator Checking User Permissions for Network Host Connections***

SELECT HOST, LOWER\_PORT, UPPER\_PORT, ACL, DECODE(

DBMS\_NETWORK\_ACL\_ADMIN.CHECK\_PRIVILEGE\_ACLID(ACLID, 'PRESTON',

'connect'),

1, 'GRANTED', 0, 'DENIED', NULL) PRIVILEGE

FROM (SELECT HOST, LOWER\_PORT, UPPER\_PORT, ACL, ACLID,

DBMS\_NETWORK\_ACL\_UTILITY.CONTAINS\_HOST('www.us.example.com',

HOST) PRECEDENCE

FROM DBA\_NETWORK\_ACLS)

WHERE PRECEDENCE IS NOT NULL ORDER BY PRECEDENCE DESC,

LOWER\_PORT NULLS LAST, UPPER\_PORT NULLS LAST;

HOST LOWER\_PORT UPPER\_PORT ACL PRIVILEGE

-------------------- ---------- ---------- -------------------- ---------

|  |  |  |  |
| --- | --- | --- | --- |
| [www.us.example.com](http://www.us.example.com/) | 80 | 80 /sys/acls/www.xml | GRANTED |
| [www.us.example.com](http://www.us.example.com/) | 3000 | 3999 /sys/acls/www.xml | GRANTED |
| [www.us.example.com](http://www.us.example.com/) |  | /sys/acls/www.xml | GRANTED |
| \*.example.com |  | /sys/acls/all.xml |  |
| \* |  | /sys/acls/all.xml |  |

In this example, user preston was granted privileges for all the network host connections found for [www.us.example.com.](http://www.us.example.com/) However, suppose preston had been granted access to a host connection on port 80, but then denied access to the host connections on ports 3000–3999. In this case, you need to create one access control list for the host connection on port 80, and a separate access control list for the host connection on ports 3000–3999.

Database Administrator Checking User Privileges for Domain Name Resolution

[Example 4–26](#_bookmark941) shows how a database administrator can check the privileges of user preston to perform domain name resolution for the host [www.us.example.com.](http://www.us.example.com/) In this example, only the access control lists assigned to hosts without a port range because the resolve privilege has no effect to those with a port range. (Remember that the user name you enter for the user parameter in CHECK\_PRIVILEGE\_ACLID is case sensitive.)

***Example 4–26 Administrator Checking Permissions for Domain Name Resolution***

SELECT HOST, ACL,

DECODE(

DBMS\_NETWORK\_ACL\_ADMIN.CHECK\_PRIVILEGE\_ACLID(ACLID, 'PRESTON',

'resolve'),

1, 'GRANTED', 0, 'DENIED', NULL) PRIVILEGE

FROM (SELECT HOST, LOWER\_PORT, UPPER\_PORT, ACL, ACLID,

DBMS\_NETWORK\_ACL\_UTILITY.CONTAINS\_HOST('www.us.example.com',

HOST) PRECEDENCE

FROM DBA\_NETWORK\_ACLS

WHERE LOWER\_PORT IS NULL AND UPPER\_PORT IS NULL) WHERE PRECEDENCE IS NOT NULL

ORDER BY PRECEDENCE DESC;

HOST ACL PRIVILEGE

-------------------- -------------------- ---------

|  |  |  |
| --- | --- | --- |
| [www.us.example.com](http://www.us.example.com/) | /sys/acls/www.xml | GRANTED |
| \*.example.com | /sys/acls/all.xml |  |
| \* | /sys/acls/all.xml |  |

###### How Users Can Check Their Network Connection and Domain Privileges

Users can query the USER\_NETWORK\_ACL\_PRIVILEGES view to check their network and domain permissions. The USER\_NETWORK\_ACL\_PRIVILEGES view is PUBLIC, so all users can select from it.

This view hides the access control lists from the user. It evaluates the permission status for the user (GRANTED or DENIED) and filters out the NULL case because the user does not need to know when the access control lists do not apply to him or her. In other words, Oracle Database only shows the user on the network hosts that explicitly grant or deny

access to him or her. Therefore, the output does not display the \*.example.com and \* that appear in the output from the database administrator-specific DBA\_NETWORK\_ACLS view.

The following sections provide examples that demonstrate how a database administrator can check user permissions for network connections and domain name resolution.

* + - [User Checking His or Her Network Connection Privileges](#_bookmark944)
    - [User Checking Own Privileges for Domain Name Resolution](#_bookmark946)

User Checking His or Her Network Connection Privileges

[Example 4–27](#_bookmark945) shows how user preston can check her privileges to connect to

[www.us.example.com.](http://www.us.example.com/)

***Example 4–27 User Checking Permissions for Network Host Connections***

SELECT HOST, LOWER\_PORT, UPPER\_PORT, STATUS PRIVILEGE FROM (SELECT HOST, LOWER\_PORT, UPPER\_PORT, STATUS,

DBMS\_NETWORK\_ACL\_UTILITY.CONTAINS\_HOST('www.us.example.com',

HOST) PRECEDENCE

FROM USER\_NETWORK\_ACL\_PRIVILEGES WHERE PRIVILEGE = 'connect')

WHERE PRECEDENCE IS NOT NULL ORDER BY PRECEDENCE DESC,

LOWER\_PORT NULLS LAST, UPPER\_PORT NULLS LAST;

HOST LOWER\_PORT UPPER\_PORT ACL PRIVILEGE

-------------------- ---------- ---------- -------------------- ---------

|  |  |  |  |
| --- | --- | --- | --- |
| [www.us.example.com](http://www.us.example.com/) | 80 | 80 /sys/acls/www.xml | GRANTED |
| [www.us.example.com](http://www.us.example.com/) | 3000 | 3999 /sys/acls/www.xml | GRANTED |
| [www.us.example.com](http://www.us.example.com/) |  | /sys/acls/www.xml | GRANTED |

User Checking Own Privileges for Domain Name Resolution

[Example 4–26](#_bookmark941) shows how the user preston can check her privileges to perform domain name resolution for [www.us.example.com:](http://www.us.example.com/)

***Example 4–28 User Checking Privileges for Domain Name Resolution***

SELECT HOST, STATUS PRIVILEGE from (SELECT HOST, STATUS,

DBMS\_NETWORK\_ACL\_UTILITY.CONTAINS\_HOST('www.us.example.com',

HOST) PRECEDENCE

FROM USER\_NETWORK\_ACL\_PRIVILEGES WHERE PRIVILEGE = 'resolve' AND

LOWER\_PORT IS NULL AND UPPER\_PORT IS NULL) WHERE PRECEDENCE IS NOT NULL

ORDER BY PRECEDENCE DESC;

HOST PRIVILEGE

-------------------- ---------

[www.us.example.com](http://www.us.example.com/) GRANTED

#### Setting the Precedence of Multiple Users and Roles in One Access Control List

By default, Oracle Database grants or denies privileges to users and roles based on their physical position in the access control list. The first user or role listed is granted

or denied privileges first, followed the second user or role, and so on. For instance, suppose the code in [Example 4–20](#_bookmark920) defined one role, ACCT\_MGR, and two users, sebastian and preston, and the access control list XML file ordered these three as follows:

<acl ...>

...

<ace>

<principal>ACCT\_MGR</principal>

<grant>true</grant>

<privilege><plsql:connect/></privilege>

</ace>

<ace>

<principal>SEBASTIAN</principal>

<grant>false</grant>

<privilege><plsql:connect/></privilege>

</ace>

<ace>

<principal>PRESTON</principal>

<grant>false</grant>

<privilege><plsql:connect/></privilege>

</ace>

</acl>

ACCT\_MGR is granted permissions first, followed by permission denials for sebastian and then preston. However, if sebastian and preston have been granted the ACCT\_ MGR role, they still could log in, because the ACCT\_MGR role appears first in the list.

Even though these two users were granted the acct\_mgr role, their specific jobs do not require them to have access to the [www.example.com](http://www.example.com/) host. If the positions were reversed—the acct\_mgr role listed after sebastian and preston—they would be denied the privilege of connecting to the network. To set the order of precedence of the ACE elements irrespective of their physical location in the CREATE\_ACL and ADD\_ PRIVILEGE statements, you can use the position attribute.

For example, the following statements set the ACE elements in the resultant XML file in this order:

1. The ACE element for sebastian appears first.
2. The ACE element for preston appears second.
3. The acct\_mgr role appears last.

In this case, neither of these users will be able to connect, because their grant privileges, which are set to FALSE, are evaluated before the acct\_mgr role.

BEGIN DBMS\_NETWORK\_ACL\_ADMIN.CREATE\_ACL (

acl => 'us-example-com-permissions.xml',

description => 'Network connection permission for ACCT\_MGR and users', principal => 'ACCT\_MGR',

is\_grant => TRUE, privilege => 'connect');

DBMS\_NETWORK\_ACL\_ADMIN.ADD\_PRIVILEGE (

acl => 'us-example-com-permissions.xml', principal => 'SEBASTIAN',

is\_grant => FALSE, privilege => 'connect', position => 1);

DBMS\_NETWORK\_ACL\_ADMIN.ADD\_PRIVILEGE (

acl => 'us-example-com-permissions.xml',

principal => 'PRESTON', is\_grant => FALSE, privilege => 'connect', position => 2);

END;

/

#### Finding Information About Access Control Lists Configured for User Access

[Table 4–6](#_bookmark955) lists data dictionary views that you can use to find information about existing access control lists. See *Oracle Database Reference* for more information about these views.

***Table 4–6 Data Dictionary Views That Display Information about Access Control Lists***

|  |  |
| --- | --- |
| **View** | **Description** |
| DBA\_NETWORK\_ACLS | Shows the access control list assignments to the network hosts. The SELECT  privilege on this view is granted to the SELECT\_CATALOG\_ROLE role only. |
| DBA\_NETWORK\_ACL\_PRIVILEGES | Shows the network privileges defined in all access control lists that are currently assigned to network hosts. The SELECT privilege on this view is granted to the SELECT\_CATALOG\_ROLE role only. |
| DBA\_WALLET\_ACLS | Lists wallets that have been assigned access control lists. |
| USER\_NETWORK\_ACL\_PRIVILEGES | Shows the status of the network privileges for the current user to access network hosts. The SELECT privilege on the view is granted to PUBLIC. |

### Finding Information About User Privileges and Roles

[Table 4–7](#_bookmark965) lists data dictionary views that you can query to access information about grants of privileges and roles. See *Oracle Database Reference* for detailed information about these views.

***Table 4–7 Data Dictionary Views That Display Information about Privileges and Roles***

|  |  |
| --- | --- |
| **View** | **Description** |
| ALL\_COL\_PRIVS | Describes all column object grants for which the current user or  PUBLIC is the object owner, grantor, or grantee |
| ALL\_COL\_PRIVS\_MADE | Lists column object grants for which the current user is object owner or grantor. |
| ALL\_COL\_PRIVS\_RECD | Describes column object grants for which the current user or  PUBLIC is the grantee |
| ALL\_TAB\_PRIVS | Lists the grants on objects where the user or PUBLIC is the grantee |
| ALL\_TAB\_PRIVS\_MADE | Lists the all object grants made by the current user or made on the objects owned by the current user. |
| ALL\_TAB\_PRIVS\_RECD | Lists object grants for which the user or PUBLIC is the grantee |
| DBA\_COL\_PRIVS | Describes all column object grants in the database |
| DBA\_EPG\_DAD\_ AUTHORIZATION | Describes the database access descriptors (DAD) that are authorized to use a different user’s privileges. |
| DBA\_TAB\_PRIVS | Lists all grants on all objects in the database |
| DBA\_ROLES | Lists all roles that exist in the database, including secure application roles |

***Table 4–7 (Cont.) Data Dictionary Views That Display Information about Privileges and***

|  |  |
| --- | --- |
| **View** | **Description** |
| DBA\_ROLE\_PRIVS | Lists roles directly granted to users and roles. Note that it does not list the PUBLIC role. |
| DBA\_SYS\_PRIVS | Lists system privileges granted to users and roles |
| ROLE\_ROLE\_PRIVS | Lists roles granted to other roles. Information is provided only about roles to which the user has access. |
| ROLE\_SYS\_PRIVS | Lists system privileges granted to roles. Information is provided only about roles to which the user has access. |
| ROLE\_TAB\_PRIVS | Lists object privileges granted to roles. Information is provided only about roles to which the user has access. |
| SESSION\_PRIVS | Lists the privileges that are currently enabled for the user |
| SESSION\_ROLES | Lists all roles that are enabled for the current user. Note that it does not list the PUBLIC role. |
| USER\_COL\_PRIVS | Describes column object grants for which the current user is the object owner, grantor, or grantee |
| USER\_COL\_PRIVS\_MADE | Describes column object grants for which the current user is the object owner |
| USER\_COL\_PRIVS\_RECD | Describes column object grants for which the current user is the grantee |
| USER\_EPG\_DAD\_ AUTHORIZATION | Describes the database access descriptors (DAD) that are authorized to use a different user’s privileges. |
| USER\_ROLE\_PRIVS | Lists roles directly granted to the current user |
| USER\_TAB\_PRIVS | Lists grants on all objects where the current user is the grantee |
| USER\_SYS\_PRIVS | Lists system privileges granted to the current user |
| USER\_TAB\_PRIVS\_MADE | Lists grants on all objects owned by the current user |
| USER\_TAB\_PRIVS\_RECD | Lists object grants for which the current user is the grantee |

This section provides some examples of using these views. For these examples, assume the following statements were issued:

CREATE ROLE security\_admin IDENTIFIED BY *password*;

GRANT CREATE PROFILE, ALTER PROFILE, DROP PROFILE,

CREATE ROLE, DROP ANY ROLE, GRANT ANY ROLE, AUDIT ANY,

AUDIT SYSTEM, CREATE USER, BECOME USER, ALTER USER, DROP USER

TO security\_admin WITH ADMIN OPTION;

GRANT SELECT, DELETE ON SYS.AUD$ TO security\_admin; GRANT security\_admin, CREATE SESSION TO swilliams; GRANT security\_admin TO system\_administrator; GRANT CREATE SESSION TO jward;

GRANT SELECT, DELETE ON emp TO jward;

GRANT INSERT (ename, job) ON emp TO swilliams, jward;

**See Also:** *Oracle Database Reference* for a detailed description of these data dictionary views

#### Listing All SystemPrivilege Grants

The following query returns all system privilege grants made to roles and users:

SELECT \* FROM DBA\_SYS\_PRIVS;

GRANTEE PRIVILEGE ADM

-------------- --------------------------------- --- SECURITY\_ADMIN ALTER PROFILE YES

SECURITY\_ADMIN ALTER USER YES

SECURITY\_ADMIN AUDIT ANY YES

SECURITY\_ADMIN AUDIT SYSTEM YES

SECURITY\_ADMIN BECOME USER YES

SECURITY\_ADMIN CREATE PROFILE YES

SECURITY\_ADMIN CREATE ROLE YES

SECURITY\_ADMIN CREATE USER YES

SECURITY\_ADMIN DROP ANY ROLE YES

SECURITY\_ADMIN DROP PROFILE YES

SECURITY\_ADMIN DROP USER YES

SECURITY\_ADMIN GRANT ANY ROLE YES

SWILLIAMS CREATE SESSION NO

JWARD CREATE SESSION NO

See *Oracle Database Reference* for detailed information about the DBA\_SYS\_PRIVS view.

#### Listing All Role Grants

The following query returns all the roles granted to users and other roles:

SELECT \* FROM DBA\_ROLE\_PRIVS;

GRANTEE GRANTED\_ROLE ADM

------------------ ------------------------------------ --- SWILLIAMS SECURITY\_ADMIN NO

See *Oracle Database Reference* for detailed information about the DBA\_ROLE\_PRIVS view.

#### Listing Object Privileges Granted toa User

The following query returns all object privileges (not including column-specific privileges) granted to the specified user:

SELECT TABLE\_NAME, PRIVILEGE, GRANTABLE FROM DBA\_TAB\_PRIVS WHERE GRANTEE = 'jward';

TABLE\_NAME PRIVILEGE GRANTABLE

----------- ------------ ---------- EMP SELECT NO

EMP DELETE NO

To list all the column-specific privileges that have been granted, use the following query:

SELECT GRANTEE, TABLE\_NAME, COLUMN\_NAME, PRIVILEGE FROM DBA\_COL\_PRIVS;

GRANTEE TABLE\_NAME COLUMN\_NAME PRIVILEGE

|  |  |  |  |
| --- | --- | --- | --- |
| ----------- | ------------ | ------------- | -------------- |
| SWILLIAMS | EMP | ENAME | INSERT |
| SWILLIAMS | EMP | JOB | INSERT |
| JWARD | EMP | NAME | INSERT |
| JWARD | EMP | JOB | INSERT |

See *Oracle Database Reference* for detailed information about the DBA\_TAB\_PRIVS view.

#### Listing the Current Privilege Domain of Your Session

The following query lists all roles currently enabled for the issuer:

SELECT \* FROM SESSION\_ROLES;

If user swilliams has the security\_admin role enabled and issues the previous query, then Oracle Database returns the following information:

ROLE

------------------------------ SECURITY\_ADMIN

The following query lists all system privileges currently available in the security domain of the issuer, both from explicit privilege grants and from enabled roles:

SELECT \* FROM SESSION\_PRIVS;

If user swilliams has the security\_admin role enabled and issues the previous query, then Oracle Database returns the following results:

PRIVILEGE

---------------------------------------- AUDIT SYSTEM

CREATE SESSION CREATE USER BECOME USER ALTER USER DROP USER CREATE ROLE DROP ANY ROLE GRANT ANY ROLE AUDIT ANY CREATE PROFILE ALTER PROFILE DROP PROFILE

If the security\_admin role is disabled for user swilliams, then the first query would return no rows, while the second query would only return a row for the CREATE SESSION privilege grant.

See *Oracle Database Reference* for detailed information about the SESSION\_ROLES view.

#### Listing Roles of the Database

You can use the DBA\_ROLES data dictionary view to list all roles of a database and the authentication used for each role. For example, the following query lists all the roles in the database:

SELECT \* FROM DBA\_ROLES;

ROLE PASSWORD

---------------- --------

CONNECT NO

RESOURCE NO

DBA NO

SECURITY\_ADMIN YES

See *Oracle Database Reference* for detailed information about the DBA\_ROLES view.

#### Listing Information About the Privilege Domains of Roles

The ROLE\_ROLE\_PRIVS, ROLE\_SYS\_PRIVS, and ROLE\_TAB\_PRIVS data dictionary views contain information about the privilege domains of roles. For example, the following query lists all the roles granted to the system\_admin role:

SELECT GRANTED\_ROLE, ADMIN\_OPTION FROM ROLE\_ROLE\_PRIVS

WHERE ROLE = 'SYSTEM\_ADMIN';

GRANTED\_ROLE ADM

---------------- ----

SECURITY\_ADMIN NO

The following query lists all the system privileges granted to the security\_admin role:

SELECT \* FROM ROLE\_SYS\_PRIVS WHERE ROLE = 'SECURITY\_ADMIN';

ROLE PRIVILEGE ADM

----------------------- ----------------------------- ---

|  |  |  |
| --- | --- | --- |
| SECURITY\_ADMIN | ALTER PROFILE | YES |
| SECURITY\_ADMIN | ALTER USER | YES |
| SECURITY\_ADMIN | AUDIT ANY | YES |
| SECURITY\_ADMIN | AUDIT SYSTEM | YES |
| SECURITY\_ADMIN | BECOME USER | YES |
| SECURITY\_ADMIN | CREATE PROFILE | YES |
| SECURITY\_ADMIN | CREATE ROLE | YES |
| SECURITY\_ADMIN | CREATE USER | YES |
| SECURITY\_ADMIN | DROP ANY ROLE | YES |
| SECURITY\_ADMIN | DROP PROFILE | YES |
| SECURITY\_ADMIN | DROP USER | YES |
| SECURITY\_ADMIN | GRANT ANY ROLE | YES |

The following query lists all the object privileges granted to the security\_admin role:

SELECT TABLE\_NAME, PRIVILEGE FROM ROLE\_TAB\_PRIVS WHERE ROLE = 'SECURITY\_ADMIN';

TABLE\_NAME PRIVILEGE

--------------------------- ---------------- AUD$ DELETE

AUD$ SELECT

See *Oracle Database Reference* for detailed information about the ROLE\_ROLE\_PRIVS, ROLE\_SYS\_PRIVS, and ROLE\_TAB\_PRIVS views.

# 5

## Managing Security for Application Developers

This chapter contains:

* + [About Application Security Policies](#_bookmark992)
  + [Considerations for Using Application-Based Security](#_bookmark995)
  + [Securing Passwords in Application Design](#_bookmark1012)
  + [Managing Application Privileges](#_bookmark1031)
  + [Creating Secure Application Roles to Control Access to Applications](#_bookmark1036)
  + [Associating Privileges with User Database Roles](#_bookmark1050)
  + [Protecting Database Objects by Using Schemas](#_bookmark1058)
  + [Managing Object Privileges in an Application](#_bookmark1071)
  + [Parameters for Enhanced Security of Database Communication](#_bookmark1094)

### About Application Security Policies

Creating an application security policy is the first step to create a secure database application. An application security policy is a list of application security requirements and rules that regulate user access to database objects.

You should draft security policies for each database application. For example, each database application should have one or more database roles that provide different levels of security when executing the application. You then can grant the database roles to other roles or directly to specific users.

Applications that can potentially allow unrestricted SQL statement processing (through tools such as SQL\*Plus or SQL Developer) also need security policies that prevent malicious access to confidential or important schema objects. In particular, you must ensure that your applications handle passwords in a secure manner.

The following sections describe aspects of application security and the Oracle Database features that you can use to plan and develop secure database applications.

### Considerations for Using Application-Based Security

Two main questions to consider when you formulate and implement application security are covered in the following sections:

* + [Are Application Users Also Database Users?](#_bookmark997)
  + [Is Security Better Enforced in the Application or in the Database?](#_bookmark1007)

#### Are Application Users Also Database Users?

Where possible, you should build applications in which application users are database users. In this way, you can leverage the intrinsic security mechanisms of the database.

For many commercial packaged applications, application users are not database users. For these applications, multiple users authenticate themselves to the application, and the application then connects to the database as a single, highly-privileged user. This is called the *One Big Application User* model.

Applications built in this way generally cannot use many of the intrinsic security features of the database, because the identity of the user is not known to the database.

[Table 5–1](#_bookmark1001) describes how the One Big Application User model affects various Oracle Database security features:

***Table 5–1 Features Affected by the One Big Application User Model***

|  |  |
| --- | --- |
| **Oracle** **Database Feature** | **Limitations of One Big Application User Model** |
| Auditing | A basic principle of security is accountability through auditing. If One Big Application User performs all actions in the database, then database auditing cannot hold individual users accountable for their actions. The application must implement its own auditing mechanisms to capture individual user actions. |
| Oracle Advanced Security enhanced authentication | Strong forms of authentication supported by Oracle Advanced Security (such as client authentication over SSL, tokens, and so on) cannot be used if the client authenticating to the database is the application, rather than an individual user. |
| Roles | Roles are assigned to database users. Enterprise roles are assigned to enterprise users who, though not created in the database, are known to the database. If application users are not database users, then the usefulness of roles is diminished.  Applications must then craft their own mechanisms to distinguish between the privileges which various application users need to access data within the application. |
| Enterprise user management feature of Oracle Advanced Security | The Enterprise user management feature enables an Oracle database to use the Oracle Identity Management Infrastructure by securely storing and managing user information and authorizations in an LDAP-based directory such as Oracle Internet Directory. While enterprise users do not need to be created in the database, they do need to be known to the database. The One Big Application User model cannot take advantage of Oracle Identity Management. |

#### Is Security Better Enforced inthe Application or inthe Database?

Applications, whose users are also database users, can either build security into the application, or rely on intrinsic database security mechanisms such as granular privileges, virtual private databases (fine-grained access control with application context), roles, stored procedures, and auditing (including fine-grained auditing). Oracle recommends that applications use the security enforcement mechanisms of the database as much as possible.

When security is enforced in the database itself, rather than in the application, it cannot be bypassed. The main shortcoming of application-based security is that security is bypassed if the user bypasses the application to access data. For example, a user who has SQL\*Plus access to the database can execute queries without going through the Human Resources application. The user, therefore, bypasses all of the security measures in the application.

Applications that use the One Big Application User model must build security enforcement into the application rather than use database security mechanisms. Because it is the application, and not the database, that recognizes users; the application itself must enforce security measures for each user.

This approach means that each application that accesses data must reimplement security. Security becomes expensive, because organizations must implement the same security policies in multiple applications, and each new application requires an expensive reimplementation.

**See Also:** ["Potential Security Problems of Using Ad Hoc Tools"](#_bookmark700) [on page 4-21](#_bookmark700)

### Securing Passwords in Application Design

This section provides strategies for securely invoking password-protected services from a batch job, script, installation file, or application. In addition to password protection, most of these strategies can be applied to other sensitive data, such as cryptographic keys.

This section contains:

* + [General Guidelines for Securing Passwords in Applications](#_bookmark1015)
  + [Securing Passwords Using an External Password Store](#_bookmark1026)
  + [Securing Passwords Using the orapwd Utility](#_bookmark1027)
  + [Example of Reading Passwords in Java](#_bookmark1028)

**See Also:**

* + - ["Minimum Requirements for Passwords" on page 3-3](#_bookmark197)
    - [Chapter 10, "Keeping Your Oracle Database Secure"](#_bookmark2118) for general guidelines on securing an Oracle database

#### General Guidelines for Securing Passwords in Applications

These guidelines are in the following categories:

* + [Platform-Specific Security Threats](#_bookmark1016)
  + [Designing Applications to Handle Password Input](#_bookmark1018)
  + [Configuring Password Formats and Behavior](#_bookmark1021)
  + [Handling Passwords in SQL\*Plus and SQL Scripts](#_bookmark1022)

###### Platform-Specific Security Threats

Be aware of the following potential security threats, which may not be obvious:

* + **On UNIX and Linux platforms, command parameters are available for viewing by all operating system users on the same host computer.** As a result, passwords entered on the command line could be exposed to other users. However, do not assume that non-UNIX and Linux platforms are safe from this threat.
  + **On some UNIX platforms, such as HP Tru64 and IBM AIX, environment variables for all processes are available for viewing by all operating system users.** However, do not assume that non-UNIX and Linux platforms are safe from this threat.
  + **On Microsoft Windows, the** **command recall feature (the Up arrow) remembers user input across command invocations.** For example, if you use the CONNECT SYSTEM/*password* notation in SQL\*Plus, exit, and then press the Up arrow to repeat the CONNECT command, the command recall feature reveals the connect string and displays the password. In addition, do not assume that non-Microsoft Windows platforms are safe from this threat.

###### Designing Applications to Handle Password Input

Follow these guidelines:

* + **Design applications to interactively prompt for passwords.** For command-line utilities, do not force users to expose passwords at a command prompt.

Check the APIs for the programming language you use to design applications for the best way to handle passwords from users. For an example of Java code that handles this functionality, see ["Example of Reading Passwords in Java" on](#_bookmark1028)

[page 5-7](#_bookmark1028).

* + **Protect your database against SQL injection attacks.**A SQL injection attack occurs when SQL statements are appended or altered in a manner not intended by the PL/SQL application. For example, an intruder can bypass password authentication by setting a WHERE clause to TRUE.

To address the problem of SQL injection attacks, use bind variable arguments or create validation checks. If you cannot use bind variables, then consider using the DBMS\_ASSERT PL/SQL package to validate the properties of input values. *Oracle Database PL/SQL Packages and Types Reference* describes the DBMS\_ASSERT package in detail. You also should review any grants to roles such as PUBLIC.

See *Oracle Database PL/SQL Language Reference* for more information about preventing SQL injection.

* + If possible, design your applications to defer authentication. For example:
* Use certificates for logins.
* Authenticate users by using facilities provided by the operating system. For example, applications on Microsoft Windows can use domain authentication.
  + **Mask or encrypt passwords.** If you must store passwords, then mask or encrypt them. For example, you can mask passwords in log files and encrypt passwords in recovery files.
  + **Authenticate each connection.** For example, if schema A exists in database 1, then do not assume that schema A in database 2 is the same user. Similarly, the local operating system user psmith is not necessarily the same person as remote user psmith.
  + **Do not store clear text passwords in files or repositories.** Storing passwords in files increases the risk of an intruder accessing them.
  + **Use a single master password.** For example:
* You can grant a single database user proxy authentication to act as other database users. In this case, only a single database password is needed. See ["Creating Proxy User Accounts and Authorizing Users to Connect Through Them" on page 3-38](#_bookmark431) for more information.
* You can create a password wallet, which can be opened by the master password. The wallet then contains the other passwords. See *Oracle Database Advanced Security Administrator's Guide* for more information about Wallet Manager.

###### Configuring Password Formats and Behavior

Follow these guidelines:

* + **Limit the lifetime for passwords.** You can set a password lifetime, after which the password expires and must be changed before the user can log in to the account. See ["Controlling Password Aging and Expiration" on page 3-8](#_bookmark250) for parameters you can use to control the lifetime of a password.
  + **Limit the ability of users to reuse old passwords.** See ["Controlling User Ability to Reuse Previous Passwords" on page 3-7](#_bookmark240) for more information.
  + **Force users to create strong, secure passwords.** See ["Guidelines for Securing Passwords" on page 10-7](#_bookmark2167) for advice on creating strong passwords. ["Enforcing Password Complexity Verification" on page 3-11](#_bookmark269) explains how you can customize password requirements.
  + **Enable case sensitivity in passwords.** See ["Enabling or Disabling Password Case Sensitivity" on page 3-13](#_bookmark275) for more information.

**See Also:**

* + - ["Minimum Requirements for Passwords" on page 3-3](#_bookmark197)
    - ["Enforcing Password Complexity Verification" on page 3-11](#_bookmark269)
    - ["Guidelines for Securing Passwords" on page 10-7](#_bookmark2167)

###### Handling Passwords in SQL\*Plus and SQL Scripts

Follow these guidelines:

* + **Do not invoke SQL\*Plus with a password on the** **command line, either in programs or scripts.** If a password is required but omitted, SQL\*Plus prompts the user for it and then automatically disables the echo feature so that the password is not displayed.

The following examples are secure because passwords are not exposed on the command line. Oracle Database also automatically encrypts these passwords over the network.

$ sqlplus system

Enter password: *password*

SQL> connect system Enter password: *password*

The following example exposes the password to other operating system users:

sqlplus system/*password*

The next example poses two security risks. First, it exposes the password to other users who may be watching over your shoulder. Second, on some platforms, such as Microsoft Windows, it makes the password vulnerable to a command line recall attack.

$ sqlplus /nolog

SQL> connect system/*password*

* + **For SQL scripts that require passwords or secret keys, for example, to create an account or to log in as an account, do not use positional parameters, such as substitution variables &1, &2, and so on.** Instead, design the script to prompt the user for the value. You should also disable the echo feature, which displays output

from a script or if you are using spool mode. To disable the echo feature, use the following setting:

SET ECHO OFF

A good practice is to ensure that the script makes the purpose of the value clear. For example, it should be clear whether or not the value will establish a new value, such as an account or a certificate, or if the value will authenticate, such as logging in to an existing account.

The following example is secure because it prevents users from invoking the script in a manner that poses security risks: It does not echo the password; it does not record the password in a spool file.

SET VERIFY OFF

ACCEPT user CHAR PROMPT ‘Enter user to connect to: ‘

ACCEPT password CHAR PROMPT ‘Enter the password for that user: ' HIDE CONNECT &user/&password

In this example:

* SET VERIFY OFF: Prevents the password from being displayed. (SET VERIFY lists each line of the script before and after substitution.) Combining the SET VERIFY OFF command with the HIDE command (in **Line 3**) is a useful technique for hiding passwords and other sensitive input data.
* ACCEPT password**:** Includes the HIDE option for the ACCEPT password prompt, which prevents the input password from being echoed.

The next example, which uses positional parameters, poses security risks because a user may invoke the script by passing the password on the command line. If the user does not enter a password and instead is prompted, the danger lies in that whatever the user types is echoed to the screen and to a spool file if spooling is enabled.

CONNECT &1/&2

* + **Control the log in times for batch scripts.** For batch scripts that require passwords, configure the account so that the script can only log in during the time in which it is supposed to run. For example, suppose you have a batch script that runs for an hour each evening starting at 8 p.m. Set the account so that the script can only log in during this time. If an intruder manages to gain access, then he or she has less of a chance of exploiting any compromised accounts.
  + **Be careful when using DML or DDL SQL statements that prompt for passwords.** In this case, sensitive information is passed in clear text over the network. You can remedy this problem by using Oracle Advanced Security. See *Oracle Database Advanced Security Administrator's Guide* for more information.

The following example of altering a password is secure because the password is not exposed:

password psmith

Changing password for psmith New password: *password*

Retype new password: *password*

This example poses a security risk because the password is exposed both at the command line and on the network:

ALTER USER psmith IDENTIFIED BY *password*

#### Securing Passwords Using an External Password Store

You can store password credentials for connecting to a database by using a client-side Oracle wallet. An Oracle wallet is a secure software container that stores the authentication and signing credentials needed for a user to log in.

See ["Managing the Secure External Password Store for Password Credentials" on](#_bookmark298) [page 3-16](#_bookmark298) for more information about the secure external password store. See also *Oracle Database Advanced Security Administrator's Guide* for information about using Oracle Wallet Manager to configure Oracle wallets.

#### Securing Passwords Using the orapwd Utility

You can create a password file for users who need to connect to an application using the SYSDBA or SYSOPER privileges over a network. To create the password file, use the ORAPWD utility. See *Oracle Database Administrator's Guide* for more information about creating and maintaining a password file.

#### Example of Reading Passwords in Java

[Example 5–1](#_bookmark1030) demonstrates how to create a Java package that can be used to read passwords.

***Example 5–1 Java Code for Reading Passwords***

// Change the following line to a name for your version of this package package passwords.sysman.emSDK.util.signing;

import java.io.IOException; import java.io.PrintStream;

import java.io.PushbackInputStream; import java.util.Arrays;

/\*\*

* The static readPassword method in this class issues a password prompt
* on the console output and returns the char array password
* entered by the user on the console input.

\*/

public final class ReadPassword {

//----------------------------------

/\*\*

* + Test driver for readPassword method.
  + @param args the command line args

\*/

public static void main(String[] args) {

char[] pass = ReadPassword.readPassword("Enter password: "); System.out.println("The password just entered is \""

+ new String(pass) + "\"");

System.out.println("The password length is " + pass.length);

}

* + Issues a password prompt on the console output and returns
  + the char array password entered by the user on the console input.
  + The password is not displayed on the console (chars are not echoed).
  + As soon as the returned char array is not needed,
  + it should be erased for security reasons (Arrays.fill(charArr, ' '));
  + A password should never be stored as a java String.

\*

* + Note that Java 6 has a Console class with a readPassword method,
  + but there is no equivalent in Java 5 or Java 1.4.
  + The readPassword method here is based on Sun's suggestions at
* [http://java.sun.com/developer/technicalArticles/Security/pwordmask.](http://java.sun.com/developer/technicalArticles/Security/pwordmask)

\*

* @param prompt the password prompt to issue
* @return new char array containing the password
* @throws RuntimeException if some error occurs

\*/

public static final char[] readPassword(String prompt) throws RuntimeException {

try {

StreamMasker masker = new StreamMasker(System.out, prompt); Thread threadMasking = new Thread(masker);

int firstByte = -1; PushbackInputStream inStream = null; try {

threadMasking.start();

inStream = new PushbackInputStream(System.in); firstByte = inStream.read();

} finally { masker.stopMasking();

}

try {

threadMasking.join();

} catch (InterruptedException e) {

throw new RuntimeException("Interrupt occurred when reading password");

}

if (firstByte == -1) {

throw new RuntimeException("Console input ended unexpectedly");

}

if (System.out.checkError()) {

throw new RuntimeException("Console password prompt output error");

}

inStream.unread(firstByte); return readLineSecure(inStream);

}

catch (IOException e) {

throw new RuntimeException("I/O error occurred when reading password");

}

}

//----------------------------------

/\*\*

* Reads one line from an input stream into a char array in a secure way
* suitable for reading a password.
* The char array will never contain a '\n' or '\r'.

\*

* @param inStream the pushback input stream
* @return line as a char array, not including end-of-line-chars;
* never null, but may be zero length array
* @throws RuntimeException if some error occurs

\*/

private static final char[] readLineSecure(PushbackInputStream inStream) throws RuntimeException {

if (inStream == null) {

throw new RuntimeException("readLineSecure inStream is null");

}

try {

char[] buffer = null; try {

buffer = new char[128]; int offset = 0;

// EOL is '\n' (unix), '\r\n' (windows), '\r' (mac)

loop:

while (true) {

int c = inStream.read(); switch (c) {

case -1: case '\n':

break loop; case '\r':

int c2 = inStream.read();

if ((c2 != '\n') && (c2 != -1)) inStream.unread(c2);

break loop; default:

buffer = checkBuffer(buffer, offset); buffer[offset++] = (char) c;

break;

}

}

char[] result = new char[offset]; System.arraycopy(buffer, 0, result, 0, offset); return result;

}

finally {

if (buffer != null) Arrays.fill(buffer, ' ');

}

}

catch (IOException e) {

throw new RuntimeException("I/O error occurred when reading password");

}

}

//----------------------------------

/\*\*

* + This is a helper method for readLineSecure.

\*

* + @param buffer the current char buffer
  + @param offset the current position in the buffer
  + @return the current buffer if it is not yet full;
  + otherwise return a larger buffer initialized with a copy
  + of the current buffer and then erase the current buffer
  + @throws RuntimeException if some error occurs

\*/

private static final char[] checkBuffer(char[] buffer, int offset) throws RuntimeException

{

if (buffer == null)

throw new RuntimeException("checkBuffer buffer is null"); if (offset < 0)

throw new RuntimeException("checkBuffer offset is negative"); if (offset < buffer.length)

return buffer; else {

try {

char[] bufferNew = new char[offset + 128]; System.arraycopy(buffer, 0, bufferNew, 0, buffer.length); return bufferNew;

} finally { Arrays.fill(buffer, ' ');

}

}

}

//----------------------------------

/\*\*

* This private class prints a one line prompt
* and erases reply chars echoed to the console.

\*/

private static final class StreamMasker extends Thread {

private static final String BLANKS = StreamMasker.repeatChars(' ', 10); private String m\_promptOverwrite;

private String m\_setCursorToStart; private PrintStream m\_out;

private volatile boolean m\_doMasking;

//----------------------------------

/\*\*

* + Constructor.
  + @throws RuntimeException if some error occurs

\*/

public StreamMasker(PrintStream outPrint, String prompt) throws RuntimeException {

if (outPrint == null)

throw new RuntimeException("StreamMasker outPrint is null"); if (prompt == null)

throw new RuntimeException("StreamMasker prompt is null"); if (prompt.indexOf('\r') != -1)

throw new RuntimeException("StreamMasker prompt contains a CR"); if (prompt.indexOf('\n') != -1)

throw new RuntimeException("StreamMasker prompt contains a NL"); m\_out = outPrint;

m\_setCursorToStart = StreamMasker.repeatChars('\010', prompt.length() + BLANKS.length());

m\_promptOverwrite = m\_setCursorToStart + prompt + BLANKS

+ m\_setCursorToStart + prompt;

}

//----------------------------------

/\*\*

* + Begin masking until asked to stop.
  + @throws RuntimeException if some error occurs

\*/

public void run()

throws RuntimeException {

int priorityOriginal = Thread.currentThread().getPriority(); Thread.currentThread().setPriority(Thread.MAX\_PRIORITY);

try {

m\_doMasking = true; while (m\_doMasking) {

m\_out.print(m\_promptOverwrite); if (m\_out.checkError())

throw new RuntimeException("Console output error writing prompt"); try {

Thread.currentThread().sleep(1);

} catch (InterruptedException ie) { Thread.currentThread().interrupt(); return;

}

}

m\_out.print(m\_setCursorToStart);

} finally { Thread.currentThread().setPriority(priorityOriginal);

}

}

//----------------------------------

/\*\*

\* Instructs the thread to stop masking.

\*/

public void stopMasking() { m\_doMasking = false;

}

//----------------------------------

/\*\*

* Returns a repeated char string.

\*

* @param c the char to repeat
* @param length the number of times to repeat the char
* @throws RuntimeException if some error occurs

\*/

private static String repeatChars(char c, int length) throws RuntimeException {

if (length < 0)

throw new RuntimeException("repeatChars length is negative"); StringBuffer sb = new StringBuffer(length);

for (int i = 0; i < length; i++) sb.append(c);

return sb.toString();

}

}

}

### Managing Application Privileges

Most database applications involve different privileges on different schema objects. Keeping track of the privileges that are required for each application can be complex. In addition, authorizing users to run an application can involve many GRANT operations.

To simplify application privilege management, you can create a role for each application and grant that role all the privileges a user must run the application. In fact, an application can have several roles, each granted a specific subset of privileges that allow greater or lesser capabilities while running the application.

For example, suppose every administrative assistant uses the Vacation application to record the vacation taken by members of the department. To best manage this application, you should:

1. Create a VACATION role.
2. Grant all privileges required by the Vacation application to the VACATION role.
3. Grant the VACATION role to all administrative assistants. Better yet, create a role that defines the privileges the administrative assistants have, and then grant the VACATION role to that role.

Grouping application privileges in a role aids privilege management. Consider the following administrative options:

* + - You can grant the role, rather than many individual privileges, to those users who run the application. Then, as employees change jobs, you need to grant or revoke only one role, rather than many privileges.
  + You can change the privileges associated with an application by modifying only the privileges granted to the role, rather than the privileges held by all users of the application.
  + You can determine the privileges that are necessary to run a particular application by querying the ROLE\_TAB\_PRIVS and ROLE\_SYS\_PRIVS data dictionary views.
  + You can determine which users have privileges on which applications by querying the DBA\_ROLE\_PRIVS data dictionary view.

**See Also:**

* + - [Chapter 4, "Configuring Privilege and Role Authorization"](#_bookmark493) for a complete discussion of creating, enabling, and disabling roles, and granting and revoking privileges
    - ["Finding Information About User Privileges and Roles" on page 4-71](#_bookmark961) for more information about the security uses of the ROLE\_TAB\_PRIVS, ROLE\_SYS\_PRIVS, and DBA\_ROLE\_PRIVS data dictionary views

### Creating Secure Application Roles to Control Access to Applications

As explained in ["Securing Role Privileges by Using Secure Application Roles" on page 4-22](#_bookmark708), a secure application role is a role that is only enabled through its associated PL/SQL package or procedure. This package defines the policy needed to control access to an application.

This section contains:

* + [Step 1: Create the Secure Application Role](#_bookmark1038)
  + [Step 2: Create a PL/SQL Package to Define the Access Policy for the Application](#_bookmark1040)

**See Also:** *Oracle Database 2 Day + Security Guide* for a tutorial on creating a secure application role

#### Step 1: Create the Secure Application Role

You create a secure application role by using the SQL statement CREATE ROLE with the IDENTIFIED USING clause. You must have the CREATE ROLE system privilege to execute this statement.

For example, to create a secure application role called hr\_admin that is associated with the sec\_mgr.hr\_admin package, follow these steps:

1. Create the security application role as follows:

CREATE ROLE hr\_admin IDENTIFIED USING sec\_mgr.hr\_admin\_role\_check;

This statement indicates the following:

* + The role hr\_admin to be created is a secure application role.
  + The role can only be enabled by modules defined inside the PL/SQL procedure sec\_mgr.hr\_admin\_role\_check. At this stage, this procedure does not need to exist; ["Step 2: Create a PL/SQL Package to Define the Access Policy for the Application" on page 5-13](#_bookmark1040) explains how to create the package or procedure.

1. Grant the security application role the privileges you would normally associate with this role.

For example, to grant the hr\_admin role SELECT, INSERT, UPDATE, and DELETE

privileges on the HR.EMPLOYEES table, you enter the following statement:

GRANT SELECT, INSERT, UPDATE, DELETE ON HR.EMPLOYEES TO hr\_admin;

Do not grant the role directly to the user. The PL/SQL procedure or package does that for you, assuming the user passes its security policies.

#### Step 2: Create a PL/SQL Package to Define the Access Policy for the Application

To enable or disable the secure application role, you create the security policies of the role within a PL/SQL package. You also can create an individual procedure to do this, but a package lets you group a set of procedures together. This lets you group a set of policies that, used together, present a solid security strategy to protect your applications. For users (or potential intruders) who fail the security policies, you can add auditing checks to the package to record the failure. Typically, you create this package in the schema of the security administrator.

The package or procedure must accomplish the following:

* + **It must use invoker’s rights to enable the role.**To create the package using invoker’s rights, you must set the AUTHID property to CURRENT\_USER. You cannot create the package by using definer’s rights.

For more information about invoker’s rights and definer’s rights, see *Oracle Database PL/SQL Language Reference*.

* + **It must include one or** **more security checks to validate the user.** One way to validate users is to use the SYS\_CONTEXT SQL function. See *Oracle Database SQL Language Reference* for more information about SYS\_CONTEXT. To find session information for a user, you can use SYS\_CONTEXT with an application context. See [Chapter 6, "Using Application Contexts to Retrieve User Information"](#_bookmark1122) for details.
  + **It must issue a SET ROLE SQL statement or DBMS\_SESSION.SET\_ROLE procedure when the user passes the security checks.** Because you create the package using invoker’s rights, you must set the role by issuing the SET ROLE SQL statement or the DBMS\_SESSION.SET\_ROLE procedure. (However, you cannot use the SET ROLE ALL statement for this type of role enablement.) The PL/SQL embedded SQL syntax does not support the SET ROLE statement, but you can invoke SET ROLE by using dynamic SQL (for example, with EXECUTE IMMEDIATE).

For more information about EXECUTE IMMEDIATE, see *Oracle Database PL/SQL Language Reference*.

Because of the way that you must create this package or procedure, you cannot use a logon trigger to enable or disable a secure application role. Instead, invoke the package directly from the application when the user logs in, before the user must use the privileges granted by the secure application role.

For example, suppose you wanted to restrict anyone using the hr\_admin role to employees who are on site (that is, using certain terminals) and between the hours of 8

a.m. and 5 p.m. As the system or security administrator, follow these steps. (You can copy and paste this text by positioning the cursor at the start of CREATE OR REPLACE in the first line.)

1. Create the procedure as follows:

CREATE OR REPLACE PROCEDURE hr\_admin\_role\_check AUTHID CURRENT\_USER

AS BEGIN

IF (SYS\_CONTEXT ('userenv','ip\_address') BETWEEN '192.0.2.10' and '192.0.2.20' AND

TO\_CHAR (SYSDATE, 'HH24') BETWEEN 8 AND 17) THEN

EXECUTE IMMEDIATE 'SET ROLE hr\_admin'; END IF;

END;

/

In this example:

* AUTHID CURRENT\_USER sets the AUTHID property to CURRENT\_USER so that invoker’s rights can be used.
* IF (SYS\_CONTEXT ('userenv','ip\_address') validates the user by using the

SYS\_CONTEXT SQL function to retrieve the user session information.

* BETWEEN ... TO\_CHAR ceates a test to grant or deny access. The test restricts access to users who are on site (that is, using certain terminals) and working between the hours of 8:00 a.m. and 5:00 p.m. If the user passes this check, the hr\_admin role is granted.
* THEN... EXECUTE grants the role to the user by issuing the SET ROLE statement using the EXECUTE IMMEDIATE command.

1. Grant EXECUTE permissions for the hr\_admin\_role\_check procedure to any user who was assigned it.

For example:

GRANT EXECUTE ON hr\_admin\_role\_check TO psmith;

To test the secure application role, log in to SQL\*Plus as the user, try to enable the role, and then try to perform an action that requires the privileges the role grants.

For example:

CONNECT PSMITH

Enter password: *password*

EXECUTE sec\_admin.hr\_admin\_role\_check;

-- *Actions requiring privileges granted by the role*

### Associating Privileges with User Database Roles

Ensure that users have only the privileges associated with the current database role. This section contains:

* + [Why Users Should Only Have the Privileges of the Current Database Role](#_bookmark1052)
  + [Using the SET ROLE Statement to Automatically Enable or Disable Roles](#_bookmark1056)

#### Why Users Should Only Have the Privileges of the Current Database Role

A single user can use many applications and associated roles. However, you should ensure that the user has only the privileges associated with the current database role. Consider the following scenario:

* + The ORDER role (for an application called Order) contains the UPDATE privilege for the INVENTORY table.
    - The INVENTORY role (for an application called Inventory) contains the SELECT

privilege for the INVENTORY table.

* + - Several order entry clerks were granted both the ORDER and INVENTORY roles.

In this scenario, an order entry clerk who was granted both roles can use the privileges of the ORDER role when running the INVENTORY application to update the INVENTORY table. The problem is that updating the INVENTORY table is not an authorized action for the INVENTORY application. It is an authorized action for the ORDER application. To avoid this problem, use the SET ROLE statement as explained in the following section.

#### Using the SET ROLE Statement to Automatically Enable or Disable Roles

Use a SET ROLE statement at the beginning of each application to automatically enable its associated role and to disable all others. This way, each application dynamically enables particular privileges for a user only when required.

The SET ROLE statement simplifies privilege management. You control what information users can access and when they can access it. The SET ROLE statement also keeps users operating in a well-defined privilege domain. If a user obtains privileges only from roles, then the user cannot combine these privileges to perform unauthorized operations.

**See Also:**

* ["When Do Grants and Revokes Take Effect?" on page 4-48](#_bookmark863) for information about enabling and disabling roles
* ["How the SET ROLE Statement Affects Grants and Revokes" on page 4-48](#_bookmark866)

### Protecting Database Objects by Using Schemas

A *schema* is a security domain that can contain database objects. The privileges granted to each user or role control access to these database objects.

This section contains:

* + - [Protecting Database Objects in a Unique Schema](#_bookmark1060)
    - [Protecting Database Objects in a Shared Schema](#_bookmark1066)

#### Protecting Database Objects ina Unique Schema

You can think of most schemas as user names: the accounts that enable users to connect to a database and access the database objects. However, a *unique schema* does not allow connections to the database, but is used to contain a related set of objects. Schemas of this sort are created as typical users, and yet are not granted the CREATE SESSION system privilege (either explicitly or through a role). However, you must temporarily grant the CREATE SESSION and RESOURCE privilege to a unique schema if you want to use the CREATE SCHEMA statement to create multiple tables and views in a single transaction.

For example, a given schema might own the schema objects for a specific application. If application users have the privileges to do so, then they can connect to the database using typical database user names and use the application and the corresponding objects. However, no user can connect to the database using the schema set up for the application. This configuration prevents access to the associated objects through the schema, and provides another layer of protection for schema objects. In this case, the

application could issue an ALTER SESSION SET CURRENT\_SCHEMA statement to connect the user to the correct application schema.

#### Protecting Database Objects ina Shared Schema

For many applications, users do not need their own accounts or schemas in a database. These users only need to access an application schema. For example, users John, Firuzeh, and Jane are all users of the Payroll application, and they need access to the payroll schema on the finance database. None of them need to create their own objects in the database. They need to only access the payroll objects. To address this issue, Oracle Advanced Security provides the enterprise users, which are

schema-independent users.

Enterprise users, users managed in a directory service, do not need to be created as database users because they use a shared database schema. To reduce administration costs, you can create an enterprise user once in the directory, and point the user at a shared schema that many other enterprise users can also access.

For more information about managing enterprise users, see *Oracle Database Enterprise User Security Administrator's Guide*.

### Managing Object Privileges in an Application

As part of designing your application, you need to determine the types of users who will be working with the application and the level of access that they need to accomplish their designated tasks. You must categorize these users into role groups, and then determine the privileges that must be granted to each role.

This section contains:

* + [What Application Developers Need to Know About Object Privileges](#_bookmark1074)
  + [SQL Statements Permitted by Object Privileges](#_bookmark1080)

#### What Application Developers Need to Know About Object Privileges

End users are typically granted object privileges. An object privilege allows a user to perform a particular action on a specific table, view, sequence, procedure, function, or package.

[Table 5–2](#_bookmark1078) summarizes the object privileges available for each type of object.

***Table 5–2 How Privileges Relate to Schema Objects***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Object Privilege**  ALTER DELETE EXECUTE INDEX | **Applies to Table?**  Yes Yes No Yes2 | **Applies to View?**  No Yes No No |  | **Applies to Applies to Sequence? Procedure?1**  Yes No  No No  No Yes  No No |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| INSERT | Yes | Yes |  | No No |
| REFERENCES | Yes | No |  | No No |
| SELECT | Yes | Yes3 |  | Yes No |
| UPDATE | Yes | Yes |  | No No |

1 Standalone stored procedures, functions, and public package constructs

|  |
| --- |
| 2 Privilege that cannot be granted to a role |
| 3 Can also be granted for snapshots |

See also ["Auditing Schema Objects" on page 9-28](#_bookmark1808) for detailed information about how schema objects can be audited.

#### SQL Statements Permitted by Object Privileges

As you implement and test your application, you should create each necessary role. Test the usage scenario for each role to ensure that the users of your application will have proper access to the database. After completing your tests, coordinate with the administrator of the application to ensure that each user is assigned the proper roles.

[Table 5–3](#_bookmark1085) lists the SQL statements permitted by the object privileges shown in [Table 5–2](#_bookmark1078).

***Table 5–3 SQL Statements Permitted by Database Object Privileges***

|  |  |
| --- | --- |
| **Object Privilege** | **SQL Statements Permitted** |
| ALTER | ALTER object (table or sequence)  CREATE TRIGGER ON object (tables only) |
| DELETE | DELETE FROM object (table, view, or synonym) |
| EXECUTE | EXECUTE object (procedure or function)  References to public package variables |
| INDEX | CREATE INDEX ON object (table, view, or synonym) |
| INSERT | INSERT INTO object (table, view, or synonym) |
| REFERENCES | CREATE or ALTER TABLE statement defining a FOREIGN KEY  integrity constraint on object (tables only) |
| SELECT | SELECT...FROM object (table, view, synonym, or snapshot)  SQL statements using a sequence |

See ["About Privileges and Roles" on page 4-1](#_bookmark497) for a discussion of object privileges. See also ["Auditing SQL Statements" on page 9-23](#_bookmark1759) for detailed information about how SQL statements can be audited.

### Parameters for Enhanced Security of Database Communication

Database administrators can manage security for their applications by following the procedures in this section.

* + - [Reporting Bad Packets Received on the Database from Protocol Errors](#_bookmark1099)
    - [Terminating or Resuming Server Execution After Receiving a Bad Packet](#_bookmark1105)
    - [Configuring the Maximum Number of Authentication Attempts](#_bookmark1108)
    - [Controlling the Display of the Database Version Banner](#_bookmark1112)
    - [Configuring Banners for Unauthorized Access and Auditing User Actions](#_bookmark1115)

#### Reporting Bad Packets Received on the Database fromProtocol Errors

Networking communication utilities such as Oracle Call Interface (OCI) or Two-Task Common (TTC) can generate a large disk file containing the stack trace and heap

dump when the server receives a bad packet, out-of-sequence packet, or a private or an unused remote procedure call. Typically, this disk file can grow quite large. An intruder can potentially cripple a system by repeatedly sending bad packets to the server, which can result in disk flooding and denial of service. An unauthenticated client can also mount this type of attack.

You can prevent these attacks by setting the SEC\_PROTOCOL\_ERROR\_TRACE\_ACTION

initialization parameter to one of the following values:

* + None: Configures the server to ignore the bad packets and does not generate any trace files or log messages. Use this setting if the server availability is overwhelmingly more important than knowing that bad packets are being received.

For example:

SEC\_PROTOCOL\_ERROR\_TRACE\_ACTION = None

* + Trace (default setting): Creates the trace files, but it is useful for debugging purposes, for example, when a network client is sending bad packets as a result of a bug.

For example:

SEC\_PROTOCOL\_ERROR\_TRACE\_ACTION = Trace

* + Log: Writes a short, one-line message to the server trace file. This choice balances some level of auditing with system availability.

For example:

SEC\_PROTOCOL\_ERROR\_TRACE\_ACTION = Log

* + Alert: Sends an alert message to a database administrator or monitoring console. For example:

SEC\_PROTOCOL\_ERROR\_TRACE\_ACTION = Alert

#### Terminating or Resuming Server Execution After Receiving a Bad Packet

After Oracle Database detects a client or server protocol error, it must continue execution. However, this could subject the server to further bad packets, which could lead to disk flooding or denial-of-service attacks.

You can control the further execution of a server process when it is receiving bad packets from a potentially malicious client by setting the SEC\_PROTOCOL\_ERROR\_ FURTHER\_ACTION initialization parameter to one of the following values:

* + Continue (default setting): Continues the server execution. However, be aware that the server may be subject to further attacks.

For example:

SEC\_PROTOCOL\_ERROR\_FURTHER\_ACTION = Continue

* + Delay,*m*: Delays the client *m* seconds before the server can accept the next request from the same client connection. This setting prevents malicious clients from excessively using server resources while legitimate clients experience a degradation in performance but can continue to function.

For example:

SEC\_PROTOCOL\_ERROR\_FURTHER\_ACTION = Delay,3

* + - Drop,*n*: Forcefully terminates the client connection after *n* bad packets. This setting enables the server to protect itself at the expense of the client, for example, loss of a transaction. However, the client can still reconnect, and attempt the same operation again.

For example:

SEC\_PROTOCOL\_ERROR\_FURTHER\_ACTION = Drop,10

#### Configuring the MaximumNumber of Authentication Attempts

With Oracle Database, a server process is first started, and then the client authenticates with this server process. An intruder could start a server process first, and then issue an unlimited number of authenticated requests with different user names and passwords in an attempt to gain access to the database.

You can limit the number of failed login attempts for application connections by setting the SEC\_MAX\_FAILED\_LOGIN\_ATTEMPTS initialization parameter to restrict the number of authentication attempts on a connection. After the specified number of authentication attempts fail, the database process drops the connection. By default, SEC\_MAX\_FAILED\_LOGIN\_ATTEMPTS is set to 10.

Remember that the SEC\_MAX\_FAILED\_LOGIN\_ATTEMPTS initialization parameter is designed to prevent potential intruders from attacking your applications; it does not apply to valid users. The sqlnet.ora INBOUND\_CONNECT\_TIMEOUT parameter and the FAILED\_LOGIN\_ATTEMPTS initialization parameter also restrict failed logins, but the difference is that these two parameters only apply to valid user accounts.

For example, to limit the maximum attempts to 5, set SEC\_MAX\_FAILED\_LOGIN\_ ATTEMPTS as follows in the init*sid*.ora initialization parameter file:

SEC\_MAX\_FAILED\_LOGIN\_ATTEMPTS = 5

#### Controlling the Display of the Database Version Banner

Detailed product version information should not be accessible before a client connection (including an Oracle Call Interface client) is authenticated. An intruder could use the database version to find information about security vulnerabilities that may be present in the database software.

You can restrict the display of the database version banner to unauthenticated clients by setting the SEC\_RETURN\_SERVER\_RELEASE\_BANNER initialization parameter in the init*sid*.ora initialization parameter file to either TRUE or FALSE. By default, SEC\_ RETURN\_SERVER\_RELEASE\_BANNER is set to FALSE.

For example, if you set it to TRUE, the Oracle Database displays the full correct database version:

Oracle Database 11g Enterprise Edition Release 11.2.0.0 - Production

In the future, if you install Oracle Database 11.2.0.2, for example, it will display the following banner:

Oracle Database 11g Enterprise Edition Release 11.2.0.2 - Production

However, if in that same release, you set it to FALSE, then Oracle Database restricts the banner to display the following fixed text starting with Release 11.2:

Oracle Database 11g Release 11.2.0.0.0 - Production

#### Configuring Banners for Unauthorized Access and Auditing User Actions

You should create and configure banners to warn users against unauthorized access and possible auditing of user actions. The notices are available to the client application when it logs into the database.

To configure these banners to display, set the following sqlnet.ora parameters on the database server side to point to a text file that contains the banner information:

* + SEC\_USER\_UNAUTHORIZED\_ACCESS\_BANNER. For example:

SEC\_USER\_UNAUTHORIZED\_ACCESS\_BANNER = /opt/Oracle/11g/dbs/unauthaccess.txt

* + SEC\_USER\_AUDIT\_ACTION\_BANNER. For example:

SEC\_USER\_AUDIT\_ACTION\_BANNER = /opt/Oracle/11g/dbs/auditactions.txt

By default, these parameters are not set. In addition, be aware that there is a 512-byte limitation for the number of characters used for the banner text.

After you set these parameters, the Oracle Call Interface application must use the appropriate OCI APIs to retrieve these banners and present them to the end user.

# 6

## Using Application Contexts to Retrieve User Information

This chapter contains:

* + - [About Application Contexts](#_bookmark1124)
    - [Types of Application Contexts](#_bookmark1143)
    - [Using Database Session-Based Application Contexts](#_bookmark1147)
    - [Using Global Application Contexts](#_bookmark1237)
    - [Using Client Session-Based Application Contexts](#_bookmark1317)
    - [Finding Information About Application Contexts](#_bookmark1332)

### About Application Contexts

This section contains:

* + - [What Is an Application Context?](#_bookmark1125)
    - [Components of the Application Context](#_bookmark1127)
    - [Where Are the Application Context Values Stored?](#_bookmark1130)
    - [Benefits of Using Application Contexts](#_bookmark1138)
    - [How Editions Affects Application Context Values](#_bookmark1141)

#### What Is an Application Context?

An application context is a set of **name-value** pairs that Oracle Database stores in memory. The application context has a label called a **namespace** (for example, empno\_ ctx for an application context that retrieves employee IDs). Inside the context are the name-value pairs (an associative array): the **name** points to a location in memory that holds the **value**. An application can use the application context to access session information about a user, such as the user ID or other user-specific information, or a client ID, and then securely pass this data to the database. You can then use this information to either permit or prevent the user from accessing data through the application. You can use application contexts to authenticate both database and nondatabase users.

#### Components of the Application Context

The components of the name-value pair are as follows:

* + **Name.** Refers to the name of the attribute set that is associated with the value. For example, if the empno\_ctx application context retrieves an employee ID from the HR.EMPLOYEES table, it could have a name such as employee\_id.
  + **Value.** Refers to a value set by the attribute. For example, for the empno\_ctx application context, if you wanted to retrieve an employee ID from the HR.EMPLOYEES table, you could create a value called emp\_id that sets the value for this ID.

Think of an application context as a global variable that holds information that is accessed during a database session. To set the values for a secure application context, you must create a PL/SQL package procedure that uses the DBMS\_SESSION.SET\_ CONTEXT procedure. In fact, this is the only way that you can set application context values if the context is not marked INITIALIZED EXTERNALLY or INITIALIZED GLOBALLY. You can assign the values to the application context attributes at run time, not when you create the application context. Because the **trusted** procedure, and not the user, assigns the values, it is a called secure application context. For client-session based application contexts, another way to set the application context is to use Oracle Call Interface (OCI) calls.

#### Where Are the Application Context Values Stored?

Oracle Database stores the application context values in a secure data cache available in the User Global Area (UGA) or the System (sometimes called "Shared") Global Area (SGA). This way, the application context values are retrieved during the session.

Because the application context stores the values in this data cache, it increases performance for your applications. You can use an application context by itself, with Oracle Virtual Private Databases policies, or with other fine-grained access control policies. See ["Using Oracle Virtual Private Database with an Application Context" on](#_bookmark1365) [page 7-3](#_bookmark1365) if you are interested in using application contexts with Virtual Private Database policies.

#### Benefits of Using Application Contexts

Most applications contain the kind of information that can be used for application contexts. For example, in an order entry application that uses a table containing the columns ORDER\_NUMBER and CUSTOMER\_NUMBER, you can use the values in these columns as security attributes to restrict access by a customer to his or her own orders, based on the ID of that customer.

Application contexts are useful for the following purposes:

* + Enforcing fine-grained access control (for example, in Oracle Virtual Private Database polices)
  + Preserving user identity across multitier environments
  + Enforcing stronger security for your applications, because the application context is controlled by a trusted procedure, not the user
  + Increasing performance by serving as a secure data cache for attributes needed by an application for fine-grained auditing or for use in PL/SQL conditional statements or loops

This cache saves the repeated overhead of querying the database each time these attributes are needed. Because the application context stores session data in cache rather than forcing your applications to retrieve this data repeatedly from a table, it greatly improves the performance of your applications.

* + - Serving as a holding area for name-value pairs that an application can define, modify, and access

#### How Editions Affects Application Context Values

Oracle Database sets the application context in all editions that are affected by the application context package. The values the application context sets are visible in all editions the application context affects.

**See Also:** *Oracle Database Advanced Application Developer's Guide* for detailed information about editions

### Types of Application Contexts

There are three general categories of application contexts:

* + - **Database session-based application contexts.** This type retrieves data that is stored in the database user session (that is, the UGA) cache. There are three categories of database session-based application contexts:
      * **Initialized locally.** Initializes the application context locally, to the session of the user.
      * **Initialized externally.** Initializes the application context from an Oracle Call Interface (OCI) application, a job queue process, or a connected user database link.
      * **Initialized globally.** Uses attributes and values from a centralized location, such as an LDAP directory.

["Using Database Session-Based Application Contexts" on page 6-4](#_bookmark1147) describes this type of application context.

* + - **Global application contexts.** This type retrieves data that is stored in the System Global Area (SGA) so that it can be used for applications that use a sessionless model, such as middle-tier applications in a three-tiered architecture. A global application context is useful if the session context must be shared across sessions, for example, through connection pool implementations.

["Using Global Application Contexts" on page 6-22](#_bookmark1237) describes this type.

* + - **Client session-based application contexts.** This type uses Oracle Call Interface functions on the client side to set the user session data, and then to perform the necessary security checks to restrict user access.

["Using Client Session-Based Application Contexts" on page 6-42](#_bookmark1317) describes this type.

[Table 6–1](#_bookmark1146) summarizes the different types of application contexts.

***Table 6–1 Types of Application Contexts***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Application Context Type** | **Stored in UGA** | **Stored in SGA** | **Supports Connected User Database Links** | **Supports Centralized Storage of Users' Application Context** | **Supports Sessionless Multitier Applications** |
| Database session-based application context initialized locally | Yes | No | No | No | No |
| Database session-based application context initialized externally | Yes | No | Yes | No | No |
| Database session-based application context initialized globally | Yes | No | No | Yes | No |
| Global application context | No | Yes | No | No | Yes |
| Client session-based application context | Yes | No | Yes | No | Yes |

### Using Database Session-Based Application Contexts

This section contains:

* + [About Database Session-Based Application Contexts](#_bookmark1149)
  + [Creating a Database Session-Based Application Context](#_bookmark1155)
  + [Creating a PL/SQL Package to Set the Database Session-Based Application Context](#_bookmark1162)
  + [Creating a Logon Trigger to Run a Database Session Application Context Package](#_bookmark1193)
  + [Tutorial: Creating and Using a Database Session-Based Application Context](#_bookmark1206)
  + [Initializing Database Session-Based Application Contexts Externally](#_bookmark1218)
  + [Initializing Database Session-Based Application Contexts Globally](#_bookmark1227)
  + [Using Externalized Database Session-Based Application Contexts](#_bookmark1235)

#### About Database Session-Based Application Contexts

If you must retrieve session information for database users, then use a database session-based application context. This type of application context uses a PL/SQL procedure within Oracle Database to retrieve, set, and secure the data it manages.

**Note:** If your users are application users, that is, users who are not in your database, consider using a global application context instead. See ["Using Global Application Contexts" on page 6-22](#_bookmark1237) for more information.

The database session-based application context is managed entirely within Oracle Database. Oracle Database sets the values, and then when the user exits the session, automatically clears the application context values stored in cache. If the user connection ends abnormally, for example, during a power failure, then the PMON background process cleans up the application context data.You do not need to explicitly clear the application context from cache.

The advantage of having Oracle Database manage the application context is that you can centralize the application context management. Any application that accesses this database will need to use this application context to permit or prevent user access to

that application. This provides benefits both in improved performance and stronger security.

You use the following components to create and use a database session-based application context:

* + - **The application context.** You use the CREATE CONTEXT SQL statement to create an application context. This statement names the application context (namespace) and associates it with a PL/SQL procedure that is designed to retrieve session data and set the application context.
    - **A PL/SQL procedure to perform the data retrieval and set the context.** ["About the Package That Manages the Database Session-Based Application Context" on](#_bookmark1164)

[page 6-7](#_bookmark1164) describes the tasks this procedure must perform. Ideally, create this procedure within a package, so that you can include other procedures if you want (for example, to perform error checking tasks).

* + - **A way to set the application context when the user logs on.** Users who log on to applications that use the application context must run a PL/SQL package that sets the application context. You can achieve this with either a logon trigger that fires each time the user logs on, or you can embed this functionality in your applications.

["Tutorial: Creating and Using a Database Session-Based Application Context" on](#_bookmark1206) [page 6-12](#_bookmark1206) shows how to create and use a database session-based application context that is initialized locally.

You can also initialize session-based application contexts either externally or globally. Either method stores the context information in the user session.

* + - **External initialization.** This type can come from an OCI interface, a job queue process, or a connected user database link. See ["Initializing Database](#_bookmark1218)

[Session-Based Application Contexts Externally" on page 6-16](#_bookmark1218) for detailed information.

* + - **Global initialization.** This type uses attributes and values from a centralized location, such as an LDAP directory. ["Initializing Database Session-Based Application Contexts Globally" on page 6-18](#_bookmark1227) provides more information.

#### Creating a Database Session-Based Application Context

To create a database session-based application context, you use the CREATE CONTEXT SQL statement. Here, you create a namespace for the application context and then associate it with a PL/SQL package that manages the name-value pair that holds the session information of the user. You must have the CREATE ANY CONTEXT system privilege to run this statement, and the DROP ANY CONTEXT privilege to use the DROP CONTEXT statement if you drop the application context. In a database session-based application context, data is stored in the database user session (UGA) in a namespace that you create with the CREATE CONTEXT SQL statement.

Each application context must have a unique attribute and belong to a namespace. That is, context names must be unique within the database, not just within a schema.

The ownership of the application context is as follows: Even though a user who has been granted the CREATE ANY CONTEXT and DROP ANY CONTEXT privileges can create and drop the application context, it is owned by the SYS schema. Oracle Database associates the context with the schema account that created it, but if you drop this user, the context still exists in the SYS schema. As user SYS, you can drop the application context.

[Example 6–1](#_bookmark1159) shows how to use CREATE CONTEXT to create a database session-based application context:

***Example 6–1 Creating a Database Session-Based Application Context***

CREATE CONTEXT empno\_ctx USING set\_empno\_ctx\_pkg;

Here, empno\_ctx is the context namespace and set\_empno\_ctx\_pkg is the package that sets attributes for the empno\_ctx namespace. When you create the application context, the PL/SQL package does not need to exist, but it must exist at run time. ["Step 3:](#_bookmark1212)

[Create a Package to Retrieve Session Data and Set the Application Context" on](#_bookmark1212) [page 6-14](#_bookmark1212) shows an example of how to create a package that can be used with this application context.

Notice that when you create the context, you do not set its name-value attributes in the CREATE CONTEXT statement. Instead, you set these in the package that you associate with the application context. The reason you do this is to prevent a malicious user from changing the context attributes without proper attribute validation.

**Note:** You cannot create a context called CLIENTCONTEXT. This word is reserved for use with client session-based application contexts. See ["Using Client Session-Based Application Contexts" on page 6-42](#_bookmark1317) for more information about this type of application context.

For each application, you can create an application context that has its own attributes. Suppose, for example, you have three applications: General Ledger, Order Entry, and Human Resources. You can specify different attributes for each application:

* + For the order entry application context, you can specify the attribute CUSTOMER\_ NUMBER.
  + For the general ledger application context, you can specify the attributes SET\_OF\_ BOOKS and TITLE.
  + For the human resources application context, you can specify the attributes

ORGANIZATION\_ID, POSITION, and COUNTRY.

The data the attributes access is stored in the tables behind the applications. For example, the order entry application uses a table called OE.CUSTOMERS, which contains the CUSTOMER\_NUMBER column, which provides data for the CUSTOMER\_NUMBER attribute. In each case, you can adapt the application context to your precise security needs.

#### Creating a PL/SQL Package to Set the Database Session-Based Application Context

This section contains:

* + [About the Package That Manages the Database Session-Based Application Context](#_bookmark1164)
  + [Using SYS\_CONTEXT to Retrieve Session Information](#_bookmark1168)
  + [Using Dynamic SQL with SYS\_CONTEXT](#_bookmark1176)
  + [Using SYS\_CONTEXT in a Parallel Query](#_bookmark1179)
  + [Using SYS\_CONTEXT with Database Links](#_bookmark1183)
  + [Using DBMS\_SESSION.SET\_CONTEXT to Set Session Information](#_bookmark1186)

###### About the Package That Manages the Database Session-Based Application Context

The PL/SQL package, usually created in the schema of the security administrator, defines procedures that manage the session data represented by the application context. It must perform the following tasks:

* + - **Retrieve session in****formation.** To retrieve the user session information, you can use the SYS\_CONTEXT SQL function. The SYS\_CONTEXT function returns the value of the parameter associated with the context namespace. You can use this function in both SQL and PL/SQL statements. Typically, you will use the built-in USERENV namespace to retrieve the session information of a user. (For detailed information about the SYS\_CONTEXT function, see *Oracle Database SQL Language Reference*.)
    - **Set the name-value attributes of the** **application context you created with CREATE CONTEXT.** You can use the DBMS\_SESSION.SET\_CONTEXT procedure to set the name-value attributes of the application context. The name-value attributes can hold information such as the user ID, IP address, authentication mode, the name of the application, and so on. The values of the attributes you set remain either until you reset them, or until the user ends the session. Note the following:
      * If the value of the parameter in the namespace already has been set, then SET\_ CONTEXT overwrites this value.
      * Be aware that any changes in the context value are reflected immediately and subsequent calls to access the value through the SYS\_CONTEXT function will return the most recent value.
    - **Be executed by users.** After you create the package, the user will need to execute the package when he or she logs on. You can create a logon trigger to execute the package automatically when the user logs on, or you can embed this functionality in your applications. Remember that the application context session values are cleared automatically when the user ends the session, so you do not need to manually remove the session data.

It is important to remember that the procedure is a trusted procedure: It is designed to prevent the user from setting his or her own application context attribute values. The user runs the procedure, but the procedure sets the application context values, not the user.

["Tutorial: Creating and Using a Database Session-Based Application Context" on](#_bookmark1206) [page 6-12](#_bookmark1206) shows how to create a database session-based application context.

###### Using SYS\_CONTEXT to Retrieve Session Information

The syntax for the PL/SQL function SYS\_CONTEXT is as follows:

SYS\_CONTEXT ('*namespace*','*parameter*'[,*length*])

In this specification:

* + - *namespace*: The name of the application context. You can specify either a string or an expression that evaluates to a string. The SYS\_CONTEXT function returns the value of parameter associated with the context namespace at the current instant. If the value of the parameter in the namespace already has been set, then SET\_ CONTEXT overwrites this value.
    - *parameter*: A parameter within the *namespace* application context. This value can be a string or an expression.
    - *length*: Optional. The default maximum size of the return type is 256 bytes, but you can override the length by specifying a value up to 4000 bytes. Enter a value

that is a NUMBER data type, or a value that can be can be implicitly converted to

NUMBER. The data type of the SYS\_CONTEXT return type is a VARCHAR2.

The SYS\_CONTEXT function provides a default namespace, USERENV, which describes the current session of the user logged on. You can use SYS\_CONTEXT to retrieve different types of session-based information about a user, such as the user host computer ID, host IP address, operating system user name, and so on. Remember that you only use USERENV to *retrieve* session data, not *set* it. The predefined attributes are listed in the description for the PL/SQL function in the *Oracle Database SQL Language Reference*.

For example, to retrieve the name of the host computer to which a client is connected, you can use the HOST parameter of USERENV as follows:

SYS\_CONTEXT ('userenv','host')

You can check the SYS\_CONTEXT settings by issuing a SELECT SQL statement on the DUAL table. The DUAL table is a small table in the data dictionary that Oracle Database and user-written programs can reference to guarantee a known result. This table has one column called DUMMY and one row that contains the value X.

[Example 6–2](#_bookmark1174) demonstrates how to find the host computer on which you are logged, assuming that you are logged on to the SHOBEEN\_PC host computer under EMP\_USERS.

***Example 6–2 Finding SYS\_CONTEXT Values***

SELECT SYS\_CONTEXT ('USERENV', 'HOST') FROM DUAL;

SYS\_CONTEXT(USERENV,HOST)

------------------------- EMP\_USERS\SHOBEEEN\_PC

**Note:** The USERENV application context namespace replaces the

USERENV function provided in earlier Oracle Database releases.

###### Using Dynamic SQL with SYS\_CONTEXT

During a session in which you expect a change in policy between executions of a given query, the query must use dynamic SQL. You must use dynamic SQL because static SQL and dynamic SQL parse statements differently:

* + Static SQL statements are parsed at compile time. They are not parsed again at execution time for performance reasons.
  + Dynamic SQL statements are parsed every time they are executed.

Consider a situation in which Policy A is in force when you compile a SQL statement, and then you switch to Policy B and run the statement. With static SQL, Policy A remains in force. Oracle Database parses the statement at compile time, but does not parse it again upon execution. With dynamic SQL, Oracle Database parses the statement upon execution, then the switch to Policy B takes effect.

For example, consider the following policy:

EMPLOYEE\_NAME = SYS\_CONTEXT ('USERENV', 'SESSION\_USER')

The policy EMPLOYEE\_NAME matches the database user name. It is represented in the form of a SQL predicate in Oracle Virtual Private Database: the predicate is considered a policy. If the predicate changes, then the statement must be parsed again to produce the correct result.

**See Also:** ["Using Automatic Reparsing for Fine-Grained Access](#_bookmark1512) [Control Policy Functions" on page 7-35](#_bookmark1512)

###### Using SYS\_CONTEXT in a Parallel Query

If you use SYS\_CONTEXT inside a SQL function that is embedded in a parallel query, then the function includes the application context.

Consider a user-defined function within a SQL statement, which sets the user ID to 5:

CREATE FUNCTION set\_id RETURN NUMBER IS

BEGIN

IF SYS\_CONTEXT ('hr', 'id') = 5 THEN RETURN 1; ELSE RETURN 2;

END IF;

END;

Now consider the following statement:

SELECT \* FROM emp WHERE set\_id( ) = 1;

When this statement is run as a parallel query, the user session, which contains the application context information, is propagated to the parallel execution servers (query child processes).

###### Using SYS\_CONTEXT with Database Links

When SQL statements within a user session involve database links, then Oracle Database runs the SYS\_CONTEXT SQL function at the host computer of the database link, and then captures the context information there (at the host computer).

If remote PL/SQL procedure calls are run on a database link, then Oracle Database runs any SYS\_CONTEXT function inside such a procedure at the destination database of the link. In this case, only externally initialized application contexts are available at the database link destination site. For security reasons, Oracle Database propagates only the externally initialized application context information to the destination site from the initiating database link site.

###### Using DBMS\_SESSION.SET\_CONTEXT to Set Session Information

After you have used the SYS\_CONTEXT function to retrieve the session data of a user, you are ready to set the application context values from the session of this user. To do so, use the DBMS\_SESSION.SET\_CONTEXT procedure. (Ensure that you have the EXECUTE privilege for the DBMS\_SESSION PL/SQL package.)

Its syntax is as follows:

DBMS\_SESSION.SET\_CONTEXT (

namespace VARCHAR2, attribute VARCHAR2, value VARCHAR2, username VARCHAR2, client\_id VARCHAR2);

In this specification:

* + - namespace: The namespace of the application context to be set, limited to 30 bytes. For example, if you were using a namespace called custno\_ctx, you would specify it as follows:

namespace => 'custno\_ctx',

* + attribute: The attribute of the application context to be set, limited to 30 bytes. For example, to create the ctx\_attrib attribute for the custno\_ctx namespace:

attribute => 'ctx\_attrib',

* + value: The value of the application context to be set, limited to 4000 bytes. Typically, this is the value retrieved by the SYS\_CONTEXT function and stored in a variable. For example:

value => ctx\_value,

* + username: Optional. The database user name attribute of the application context. The default is NULL, which permits any user to access the session. For database session-based application contexts, omit this setting so that it uses the NULL default.

The username and client\_id parameters are used for globally accessed application contexts. See ["Setting the DBMS\_SESSION.SET\_CONTEXT username](#_bookmark1269) [and client\_id Parameters" on page 6-25](#_bookmark1269) for more information.

* + client\_id: Optional. The application-specific client\_id attribute of the application context (64-byte maximum). The default is NULL, which means that no client ID is specified. For database session-based application contexts, omit this setting so that it uses the NULL default.

See *Oracle Database PL/SQL Packages and Types Reference* for detailed information about the DBMS\_SESSION package.

For example, remember the application context created in [Example 6–1 on page 6-6](#_bookmark1159):

CREATE CONTEXT empno\_ctx USING set\_empno\_ctx\_proc;

[Example 6–3](#_bookmark1192) shows how to create a simple procedure that creates an attribute for the

empno\_ctx application context.

***Example 6–3 Simple Procedure to Create an Application Context Value***

CREATE OR REPLACE PROCEDURE set\_empno\_ctx\_proc( emp\_value IN VARCHAR2)

IS BEGIN

DBMS\_SESSION.SET\_CONTEXT('empno\_ctx', 'empno\_attrib', emp\_value); END;

/

In this example:

* + emp\_value IN VARCHAR2 takes emp\_value as the input parameter. This parameter specifies the value associated with the application context attribute empno\_attrib. Its limit is 4000 bytes.
  + DBMS\_SESSION.SET\_CONTEXT('empno\_ctx', 'empno\_attrib', emp\_value) sets the value of the application context by using the DBMS\_SESSION.SET\_CONTEXT procedure:
* 'empno\_ctx': Refers to the application context namespace. Enclose its name in single quotation marks.
* 'empno\_attrib': Creates the attribute associated with the application context namespace.
  + emp\_value: Specifies the value for the empno\_attrib attribute. Here, it refers to the emp\_value parameter defined in **Line 2**.

At this stage, you can run the set\_empno\_ctx\_proc procedure to set the application context:

EXECUTE set\_empno\_ctx\_proc ('42783');

(In a real world scenario, you would set the application context values in the procedure itself, so that it becomes a trusted procedure. This example is only used to show how data can be set for demonstration purposes.)

To check the application context setting, run the following SELECT statement:

SELECT SYS\_CONTEXT ('empno\_ctx', 'empno\_attrib') empno\_attrib FROM DUAL;

EMPNO\_ATTRIB

-------------- 42783

You can also query the SESSION\_CONTEXT data dictionary view to find all the application context settings in the current session of the database instance. For example:

SELECT \* FROM SESSION\_CONTEXT;

NAMESPACE ATTRIBUTE VALUE

-------------------------------------------------- EMPNO\_CTX EMP\_ID 42783

**See Also:**

* + - ["Tutorial: Creating and Using a Database Session-Based Application Context" on page 6-12](#_bookmark1206) for how to create a package that retrieves the user session information and then sets the application context based on this information
    - *Oracle Database PL/SQL Packages and Types Reference* for detailed information about the DBMS\_SESSION.SET\_CONTEXT procedure

#### Creating a Logon Trigger to Run a Database Session Application Context Package

After you create the application context and its associated package, the user must run the package procedure when he or she logs on. You can create a logon trigger that handles this automatically. You do not need to grant the user EXECUTE permissions to run the package.

[Example 6–4](#_bookmark1197) shows a simple logon trigger that executes a PL/SQL procedure.

***Example 6–4 Creating a Simple Logon Trigger***

CREATE OR REPLACE TRIGGER set\_empno\_ctx\_trig AFTER LOGON ON DATABASE BEGIN

sec\_mgr.set\_empno\_ctx\_proc; END;

[Example 6–5](#_bookmark1201) shows how to create a logon trigger that uses a WHEN OTHERS exception. Otherwise, if there is an error in the PL/SQL logic that creates an unhandled exception, then all connections to the database are blocked. This example shows a WHEN OTHERS exception that writes errors to a table in the security administrator’s schema. In

a production environment, this is safer than sending the output to the user session, where it could be vulnerable to security attacks.

***Example 6–5 Creating a Logon Trigger for a Production Environment***

CREATE OR REPLACE TRIGGER set\_empno\_ctx\_trig AFTER LOGON ON DATABASE BEGIN

sec\_mgr.set\_empno\_ctx\_proc; EXCEPTION

WHEN OTHERS THEN

v\_code := SQLCODE;

v\_errm := SUBSTR(SQLERRM, 1 , 64);

-- Invoke another procedure,

-- declared with PRAGMA AUTONOMOUS\_TRANSACTION,

-- to insert information about errors.

INSERT INTO sec\_mgr.errors VALUES (v\_code, v\_errm, SYSTIMESTAMP); END;

/

[Example 6–6](#_bookmark1203) shows how to create the same logon trigger for a development environment, in which you may want to output errors the user session for debugging purposes.

***Example 6–6 Creating a Logon Trigger for a Development Environment***

CREATE TRIGGER set\_empno\_ctx\_trig AFTER LOGON ON DATABASE

BEGIN

sysadmin\_ctx.set\_empno\_ctx\_pkg.set\_empno; EXCEPTION

WHEN OTHERS THEN RAISE\_APPLICATION\_ERROR(

-20000, 'Trigger sysadmin\_ctx.set\_empno\_ctx\_trig violation. Login denied.'); END;

/

Note the following:

* + **If the PL/SQL package procedure called by the logon trigger has any unhandled exceptions or raises any exceptions (because, for example, a security check failed), then the logon trigger fails.** When the logon trigger fails, the logon fails, that is, the user is denied permission to log in to the database.
  + **Logon triggers may affect performance.** In addition, test the logon trigger on a sample schema user first before creating it for the database. That way, if there is an error, you can easily correct it.
  + **Be aware of situations in which if you have a changing set of books, or if positions change constantly.** In these cases, the new attribute values may not be picked up right away, and you must force a cursor reparse to pick them up.

**Note:** A logon trigger can be used because the user context (information such as EMPNO, GROUP, MANAGER) should be set before the user accesses any data.

#### Tutorial: Creating and Using a Database Session-Based Application Context

This section contains:

* + - [About This Tutorial](#_bookmark1207)
    - [Step 1: Create User Accounts and Ensure the User SCOTT Is Active](#_bookmark1210)
    - [Step 2: Create the Database Session-Based Application Context](#_bookmark1211)
    - [Step 3: Create a Package to Retrieve Session Data and Set the Application Context](#_bookmark1212)
    - [Step 4: Create a Logon Trigger for the Package](#_bookmark1215)
    - [Step 5: Test the Application Context](#_bookmark1216)
    - [Step 6: Remove the Components for This Tutorial](#_bookmark1217)

###### About This Tutorial

This tutorial shows how to create an application context that checks the employee ID of any database user who tries to log in to the database.

###### Step 1: Create User Accounts and Ensure the User SCOTT Is Active

* 1. Log on as user SYS and connect using the AS SYSDBA privilege.

sqlplus sys as sysdba Enter password: *password*

* 1. Create the sysadmin\_ctx account, who will administer the database session-based application context.

GRANT CREATE SESSION, CREATE ANY CONTEXT, CREATE PROCEDURE, CREATE TRIGGER,

ADMINISTER DATABASE TRIGGER TO sysadmin\_ctx IDENTIFIED BY *password*; GRANT SELECT ON HR.EMPLOYEES TO sysadmin\_ctx;

GRANT EXECUTE ON DBMS\_SESSION TO sysadmin\_ctx;

Replace *password* with a password that is secure. See ["Minimum Requirements for](#_bookmark197) [Passwords" on page 3-3](#_bookmark197) for more information.

* 1. Create the following user account for Lisa Ozer, who is listed as having lozer for her email account in the HR.EMPLOYEES table.

GRANT CREATE SESSION TO LOZER IDENTIFIED BY *password*;

Replace *password* with a password that is secure. See ["Minimum Requirements for](#_bookmark197) [Passwords" on page 3-3](#_bookmark197) for more information.

* 1. The sample user SCOTT will also be used in this tutorial, so query the DBA\_USERS

data dictionary view to ensure that SCOTT is not locked or expired.

SELECT USERNAME, ACCOUNT\_STATUS FROM DBA\_USERS WHERE USERNAME = 'SCOTT';

If the DBA\_USERS view lists user SCOTT as locked and expired, then enter the following statement to unlock the SCOTT account and create a new password for him:

ALTER USER SCOTT ACCOUNT UNLOCK IDENTIFIED BY *password*;

Enter a password that is secure. For greater security, do **not** give the SCOTT account the same password from previous releases of Oracle Database. See ["Minimum](#_bookmark197) [Requirements for Passwords" on page 3-3](#_bookmark197) for the minimum requirements for creating passwords.

###### Step 2: Create the Database Session-Based Application Context

1. Log on to SQL\*Plus as sysadmin\_ctx.

CONNECT sysadmin\_ctx Enter password: *password*

1. Create the application context using the following statement:

CREATE CONTEXT empno\_ctx USING set\_empno\_ctx\_pkg;

Remember that even though user sysadmin\_ctx has created this application context, the SYS schema owns the context.

###### Step 3: Create a Package to Retrieve Session Data and Set the Application Context

[Example 6–7](#_bookmark1213) shows how to create the package you need to retrieve the session data and set the application context. Before creating the package, ensure that you are still logged on as user sysadmin\_ctx. (You can copy and paste this text by positioning the cursor at the start of CREATE OR REPLACE in the first line.)

***Example 6–7 Package to Retrieve Session Data and Set a Database Session Context***

CREATE OR REPLACE PACKAGE set\_empno\_ctx\_pkg IS PROCEDURE set\_empno;

END;

/

CREATE OR REPLACE PACKAGE BODY set\_empno\_ctx\_pkg IS PROCEDURE set\_empno

IS

emp\_id HR.EMPLOYEES.EMPLOYEE\_ID%TYPE; BEGIN

SELECT EMPLOYEE\_ID INTO emp\_id FROM HR.EMPLOYEES

WHERE email = SYS\_CONTEXT('USERENV', 'SESSION\_USER');

DBMS\_SESSION.SET\_CONTEXT('empno\_ctx', 'employee\_id', emp\_id); EXCEPTION

WHEN NO\_DATA\_FOUND THEN NULL; END;

END;

/

This package creates a procedure called set\_empno that performs the following actions:

* + emp\_id HR.EMPLOYEES.EMPLOYEE\_ID%TYPE declares a variable, emp\_id, to store the employee ID for the user who logs on. It uses the same data type as the EMPLOYEE\_ ID column in HR.EMPLOYEES.
  + SELECT EMPLOYEE\_ID INTO emp\_id FROM HR.EMPLOYEES performs a SELECT statement to copy the employee ID that is stored in the employee\_id column data from the HR.EMPLOYEES table into the emp\_id variable.
  + WHERE email = SYS\_CONTEXT('USERENV', 'SESSION\_USER') uses a WHERE clause to find all employee IDs that match the email account for the session user. The SYS\_CONTEXT function uses the predefined USERENV context to retrieve the user session ID, which is the same as the email column data. For example, the user ID and email address for Lisa Ozer are both the same: lozer.
  + DBMS\_SESSION.SET\_CONTEXT('empno\_ctx', 'employee\_id', emp\_id) uses the

DBMS\_SESSION.SET\_CONTEXT procedure to set the application context:

* 'empno\_ctx': Calls the application context empno\_ctx. Enclose empno\_ctx in single quotes.
  + 'employee\_id': Creates the attribute value of the empno\_ctx application context name-value pair, by naming it employee\_id. Enclose employee\_id in single quotes.
  + emp\_id: Sets the value for the employee\_id attribute to the value stored in the emp\_id variable. The emp\_id variable was created in **Line 8** and the employee ID was retrieved in **Lines 10–11**.

To summarize, the set\_empno\_ctx\_pkg.set\_empno procedure says, "Get the session ID of the user and then match it with the employee ID and email address of any user listed in the HR.EMPLOYEES table."

* EXCEPTION ... WHEN\_NO\_DATA\_FOUND adds a WHEN NO\_DATA\_FOUND system exception to catch any no data found errors that may result from the SELECT statement. Without this exception, the package and logon trigger will work fine and set the application context as needed, but then any non-system administrator users other than the users listed in the HR.EMPLOYEES table will not be able to log in to the database. Other users should be able to log in to the database, assuming they are valid database users. Once the application context information is set, then you can use this session information as a way to control user access to a particular application.

###### Step 4: Create a Logon Trigger for the Package

As user sysadmin\_ctx, create the following trigger:

CREATE TRIGGER set\_empno\_ctx\_trig AFTER LOGON ON DATABASE BEGIN

sysadmin\_ctx.set\_empno\_ctx\_pkg.set\_empno; END;

/

###### Step 5: Test the Application Context

* 1. Log on as user lozer.

CONNECT lozer

Enter password: *password*

When user lozer logs on, the empno\_ctx application context collects her employee ID. You can check it as follows:

SELECT SYS\_CONTEXT('empno\_ctx', 'employee\_id') emp\_id FROM DUAL;

The following output should appear:

EMP\_ID

-------------------------------------------------------- 168

* 1. Log on as user SCOTT.

CONNECT SCOTT

Enter password: *password*

User SCOTT is not listed as an employee in the HR.EMPLOYEES table, so the empno\_ ctx application context cannot collect an employee ID for him.

SELECT SYS\_CONTEXT('empno\_ctx', 'employee\_id') emp\_id FROM DUAL;

The following output should appear:

EMP\_ID

--------------------------------------------------------

From here, the application can use the user session information to determine how much access the user can have in the database. You can use Oracle Virtual Private Database to accomplish this. See [Chapter 7, "Using Oracle Virtual Private Database](#_bookmark1341) [to Control Data Access"](#_bookmark1341) for more information.

###### Step 6: Remove the Components for This Tutorial

1. Log on as SYS and connect using AS SYSDBA.

CONNECT SYS/AS SYSDBA

Enter password: *password*

1. Drop the users sysadmin\_ctx and lozer:

DROP USER sysadmin\_ctx CASCADE; DROP USER lozer;

1. Drop the application context.

DROP CONTEXT empno\_ctx;

Remember that even though sysadmin\_ctx created the application context, it is owned by the SYS schema.

1. If you want, lock and expire SCOTT, unless other users want to use this account:

ALTER USER SCOTT PASSWORD EXPIRE ACCOUNT LOCK;

#### Initializing Database Session-Based Application Contexts Externally

When you initialize a database session-based application context externally, you specify a special type of namespace that accepts the initialization of attribute values from external resources and then stores them in the local user session. Initializing an application context externally enhances performance because it is stored in the UGA and enables the automatic propagation of attributes from one session to another.

Connected user database links are supported only by application contexts initialized from OCI-based external sources.

This section contains:

* + [Obtaining Default Values from Users](#_bookmark1221)
  + [Obtaining Values from Other External Resources](#_bookmark1222)
  + [Initializing Application Context Values from a Middle-Tier Server](#_bookmark1224)

###### Obtaining Default Values from Users

Sometimes you need the default values from users. Initially, these default values may be hints or preferences, and then after validation, they become trusted contexts.

Similarly, it may be more convenient for clients to initialize some default values, and then rely on a login event trigger or applications to validate the values.

For job queues, the job submission routine records the context being set at the time the job is submitted, and restores it when executing the batched job. To maintain the integrity of the context, job queues cannot bypass the designated PL/SQL package to set the context. Rather, the externally initialized application context accepts initialization of context values from the job queue process.

Automatic propagation of context to a remote session may create security problems. Developers or administrators can effectively handle the context that takes default

values from resources other than the designated PL/SQL procedure by using logon triggers to reset the context when users log in.

###### Obtaining Values from Other External Resources

You can create an application context that accepts the initialization of attributes and values through external resources. Examples include an OCI interface, a job queue process, or a database link.

Externally initialized application contexts provide the following features:

* + - For remote sessions, automatic propagation of context values that are in the externally initialized application context namespace
    - For job queues, restoration of context values that are in the externally initialized application context namespace
    - For OCI interfaces, a mechanism to initialize context values that are in the externally initialized application context namespace

Although any client program that is using Oracle Call Interface can initialize this type of namespace, you can use login event triggers to verify the values. It is up to the application to interpret and trust the values of the attributes.

[Example 6–8](#_bookmark1223) shows how to create a database session-based application context that obtains values from an external source.

***Example 6–8 Creating an Externalized Database Session-based Application Context***

CREATE CONTEXT ext\_ctx USING ext\_ctx\_pkg INITIALIZED EXTERNALLY;

###### Initializing Application Context Values from a Middle-Tier Server

Middle-tier servers can initialize application context values on behalf of database users. Context attributes are propagated for the remote session at initialization time, and the remote database accepts the values if the namespace is externally initialized.

For example, a three-tier application creating lightweight user sessions through OCI or JDBC/OCI can access the PROXY\_USER attribute in USERENV. This attribute enables you to determine if the user session was created by a middle-tier application. You could allow a user to access data only for connections where the user is proxied. If users connect directly to the database, then they would not be able to access any data.

You can use the PROXY\_USER attribute from the USERENV namespace within Oracle Virtual Private Database to ensure that users only access data through a particular middle-tier application. For a different approach, you can develop a secure application role to enforce your policy that users access the database only through a specific proxy.

**See Also:**

* ["Preserving User Identity in Multitiered Environments" on page 3-36](#_bookmark414) for information about proxy authentication and about using the USERENV attribute CLIENT\_IDENTIFIER to preserve user identity across multiple tiers
* ["Using a Middle Tier Server for Proxy Authentication" on](#_bookmark416)

[page 3-36](#_bookmark416) for information about using a secure application role to enforce a policy through a specific proxy

* *Oracle Database JDBC Developer's Guide*
* *Oracle Call Interface Programmer's Guide*

#### Initializing Database Session-Based Application Contexts Globally

This section contains:

* + [About Initializing Database Session-Based Application Contexts Globally](#_bookmark1228)
  + [Using Database Session-Based Application Contexts with LDAP](#_bookmark1230)
  + [How Globally Initialized Database Session-Based Application Contexts Work](#_bookmark1232)
  + [Example of Initializing a Database Session-Based Application Context Globally](#_bookmark1234)

###### About Initializing Database Session-Based Application Contexts Globally

You can use a centralized location to store the database session-based application context of the user. This enables applications to set up a user context during initialization based upon user identity. In particular, this feature supports Oracle Label Security labels and privileges. Initializing an application context globally makes it easier to manage contexts for large numbers of users and databases.

For example, many organizations want to manage user information centrally, in an LDAP-based directory. Enterprise User Security, a feature of Oracle Advanced Security, supports centralized user and authorization management in Oracle Internet Directory. However, there may be additional attributes an application must retrieve from Lightweight Directory Access Protocol (LDAP) to use for Oracle Virtual Private Database enforcement, such as the user title, organization, or physical location.

Initializing an application context globally enables you to retrieve these types of attributes.

###### Using Database Session-Based Application Contexts with LDAP

An application context that is initialized globally uses LDAP, a standard, extensible, and efficient directory access protocol. The LDAP directory stores a list of users to which this application is assigned. Oracle Database uses a directory service, typically Oracle Internet Directory, to authenticate and authorize enterprise users.

**Note:**

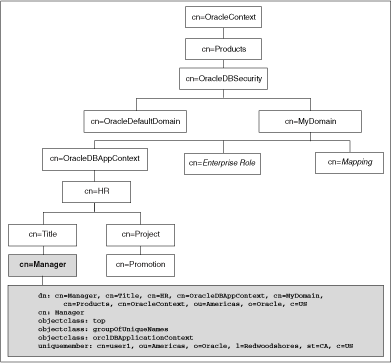
* + - Enterprise User Security requires Oracle Advanced Security.
    - You can use third-party directories such as Microsoft Active Directory and Sun Microsystems SunONE as the directory service.

The orclDBApplicationContext LDAP object (a subclass of groupOfUniqueNames) stores the application context values in the directory. The location of the application context object is described in [Figure 6–1](#_bookmark1231), which is based on the Human Resources example.

The LDAP object inetOrgPerson enables multiple entries to exist for some attributes. However, be aware that when these entries are loaded into the database and accessed with the SYS\_LDAP\_USER\_DEFAULT context namespace, only the first of these entries is returned. For example, the inetOrgPerson object for a user allows multiple entries for telephoneNumber (thus allowing a user to have multiple telephone numbers stored). When you use the SYS\_LDAP\_USER\_DEFAULT context namespace, only the first telephone number is retrieved.

On the LDAP side, an internal C function is required to retrieve the orclDBApplicationContext value, which returns a list of application context values to the database. In this example, HR is the namespace; Title and Project are the attributes; and Manager and Promotion are the values.

***Figure 6–1 Location of Application Context in LDAP Directory Information Tree***



###### How Globally Initialized Database Session-Based Application Contexts Work

To use a globally initialized secure application, you need to first configure Enterprise User Security, a feature of Oracle Advanced Security. Then, you set up the application context values for the user in the database and the directory.

When a global user (enterprise user) connects to the database, Enterprise User Security verifies the identity of the user connecting to the database. After authentication, the global user roles and application context are retrieved from the directory. When the user logs on to the database, the global roles and initial application context are already set.

**See Also:** *Oracle Database Enterprise User Security Administrator's Guide* for information about configuring Enterprise User Security

###### Example of Initializing a Database Session-Based Application Context Globally

You can configure and store the initial application context for a user, such as the department name and title, in the LDAP directory. The values are retrieved during user login so that the context is set properly. In addition, any information related to the user is retrieved and stored in the SYS\_USER\_DEFAULTS application context namespace. The following procedure shows how this is accomplished:

* 1. Create the application context in the database.

CREATE CONTEXT hr USING hrapps.hr\_manage\_pkg INITIALIZED GLOBALLY;

* 1. Create and add new entries in the LDAP directory.

An example of the entries added to the LDAP directory follows. These entries create an attribute named Title with the attribute value Manager for the application (namespace) HR, and assign user names user1 and user2. In the following, cn=example refers to the name of the domain.

dn: cn=OracleDBAppContext,cn=example,cn=OracleDBSecurity,cn=Products,cn=OracleConte xt,ou=Americas,o=oracle,c=US

changetype: add

cn: OracleDBAppContext objectclass: top objectclass: orclContainer

dn: cn=hr,cn=OracleDBAppContext,cn=example,cn=OracleDBSecurity,cn=Products,cn=Oracl eContext,ou=Americas,o=oracle,c=US

changetype: add cn: hr objectclass: top

objectclass: orclContainer

dn: cn=Title,cn=hr, cn=OracleDBAppContext,cn=example,cn=OracleDBSecurity,cn=Products,cn=OracleConte xt,ou=Americas,o=oracle,c=US

changetype: add cn: Title objectclass: top

objectclass: orclContainer

dn: cn=Manager,cn=Title,cn=hr, cn=OracleDBAppContext,cn=example,cn=OracleDBSecurity,cn=Products,cn=OracleConte xt,ou=Americas,o=oracle,c=US

cn: Manager objectclass: top

objectclass: groupofuniquenames objectclass: orclDBApplicationContext

uniquemember: CN=user1,OU=Americas,O=Oracle,L=Redwoodshores,ST=CA,C=US uniquemember: CN=user2,OU=Americas,O=Oracle,L=Redwoodshores,ST=CA,C=US

* 1. If an LDAP inetOrgPerson object entry exists for the user, then the connection retrieves the attributes from inetOrgPerson, and assigns them to the namespace SYS\_LDAP\_USER\_DEFAULT. The following is an example of an inetOrgPerson entry:

dn: cn=user1,ou=Americas,O=oracle,L=redwoodshores,ST=CA,C=US changetype: add

objectClass: top objectClass: person

objectClass: organizationalPerson objectClass: inetOrgPerson

cn: user1 sn: One

givenName: User initials: UO

title: manager, product development uid: uone

mail: [uone@us.example.com](mailto:uone@us.example.com)

telephoneNumber: +1 650 555 0105

employeeNumber: 00001 employeeType: full time

* 1. Connect to the database.

When user1 connects to a database that belongs to the example domain, user1 will have his Title set to Manager. Any information related to user1 will be retrieved from the LDAP directory. The value can be obtained using the following syntax:

SYS\_CONTEXT('*namespace*','*attribute name*')

For example:

DECLARE

tmpstr1 VARCHAR2(30); tmpstr2 VARCHAR2(30); BEGIN

tmpstr1 = SYS\_CONTEXT('HR','TITLE);

tmpstr2 = SYS\_CONTEXT('SYS\_LDAP\_USER\_DEFAULT','telephoneNumber'); DBMS\_OUTPUT.PUT\_LINE('Title is ' || tmpstr1); DBMS\_OUTPUT.PUT\_LINE('Telephone Number is ' || tmpstr2);

END;

The output of this example is:

Title is Manager

Telephone Number is +1 650 555 0105

#### Using Externalized Database Session-Based Application Contexts

Many applications store attributes used for fine-grained access control within a database metadata table. For example, an employees table could include cost center, title, signing authority, and other information useful for fine-grained access control. Organizations also centralize user information for user management and access control in LDAP-based directories, such as Oracle Internet Directory. Application context attributes can be stored in Oracle Internet Directory, and assigned to one or more enterprise users. They can also be retrieved automatically upon login for an enterprise user, and then used to initialize an application context.

**Note:** Enterprise User Security is a feature of Oracle Advanced Security.

**See Also:**

* + - ["Initializing Database Session-Based Application Contexts Externally" on page 6-16](#_bookmark1218) for information about initializing local application context through external resources such as an OCI interface, a job queue process, or a database link
    - ["Initializing Database Session-Based Application Contexts Globally" on page 6-18](#_bookmark1227) for information about initializing local application context through a centralized resource, such as Oracle Internet Directory
    - *Oracle Database Enterprise User Security Administrator's Guide* for information about enterprise users

### Using Global Application Contexts

This section contains:

* + [About Global Application Contexts](#_bookmark1239)
  + [Using Global Application Contexts in an Oracle Real Application Clusters Environment](#_bookmark1254)
  + [Creating a Global Application Context](#_bookmark1256)
  + [Creating a PL/SQL Package to Manage a Global Application Context](#_bookmark1260)
  + [Embedding Calls in Middle-Tier Applications to Manage the Client Session ID](#_bookmark1292)
  + [Tutorial: Creating a Global Application Context That Uses a Client Session ID](#_bookmark1299)
  + [Global Application Context Processes](#_bookmark1312)

#### About Global Application Contexts

A global application context enables application context values to be accessible across database sessions, including Oracle RAC instances.Oracle Database stores the global application context information in the System (sometimes called "Shared") Global Area (SGA) so that it can be used for applications that use a sessionless model, such as middle-tier applications in a three-tiered architecture. These applications cannot use a session-based application context because users authenticate to the application, and then it typically connects to the database as a single identity. Oracle Database initializes the global application context once, rather than for each user session. This improves performance, because connections are reused from a connection pool.

There are three general uses for global application contexts:

* + **You must share application values globally for all database users.** For example, you may need to disable access to an application based on a specific situation. In this case, the values the application context sets are not user-specific, nor are they based on the private data of a user. The application context defines a situation, for example, to indicate the version of application module that is running.
  + **You have database users who must move from one application to another.** In this case, the second application the user is moving to has different access requirements from the first application.
  + **You must authenticate no****ndat****abase** **users,** **that is, users who are not known to the database.** This type of user, who does not have a database account, typically connects through a Web application by using a connection pool. These types of applications connect users to the database as single user, using the One Big Application User authentication model. To authenticate this type of user, you use the client session ID of the user.

A global application context has the following components:

* + **The global application context.** You use the CREATE CONTEXT SQL statement to create the global application context, and include the ACCESSED GLOBALLY clause in the statement. This statement names the application context and associates it with a PL/SQL procedure that is designed to set the application data context data. The global application context is created and stored in the database schema of the security administrator who creates it.
  + **A PL/SQL package to set the attributes.** The package must contain a procedure that uses the DBMS\_SESSION.SET\_CONTEXT procedure to set the global application context. The SET\_CONTEXT procedure provides parameters that enable you to create

a global application context that fits any of the three user situations described in this section. You create, store, and run the PL/SQL package on the database server. Typically, it belongs in the schema of the security administrator who created it.

* + - **A middle-tier application to get and set the client session ID.** For nondatabase users, which require a client session ID to be authenticated, you can use the Oracle Call Interface (OCI) calls in the middle-tier application to retrieve and set their session data. You can also use the DBMS\_SESSION.SET\_IDENTIFIER procedure to set the client session ID. An advantage of creating a client session ID to store the nondatabase user’s name is that you can query the CLIENT\_ID column of DBA\_ AUDIT\_TRAIL, DBA\_FGA\_AUDIT\_TRAIL, and DBA\_COMMON\_AUDIT\_TRAIL data dictionary views to audit this user’s activity.

**Note:** Be aware that the DBMS\_APPLICATION\_INFO.SET\_CLIENT\_INFO setting can overwrite the value. See ["Using the DBMS\_SESSION](#_bookmark482) [PL/SQL Package to Set and Clear the Client Identifier" on page 3-46](#_bookmark482) for more information.

#### Using Global Application Contexts in an Oracle Real Application Clusters Environment

In an Oracle RAC environment, whenever a global application context is loaded or changed, it is visible only to the existing active instances. Be aware that setting a global application context value in an Oracle RAC environment has performance overhead of propagating the context value consistently to all Oracle RAC instances.

If you flush the global application context (using the ALTER SYSTEM FLUSH GLOBAL\_ CONTEXT SQL statement) in one Oracle RAC instance, then all the global application context is flushed in all other Oracle RAC instances as well.

#### Creating a Global Application Context

To create a global application context, use the CREATE CONTEXT SQL statement to create the application context and include the ACCESSED GLOBALLY clause in the statement.

You must have the CREATE ANY CONTEXT system privilege before you can use the CREATE CONTEXT statement, and the DROP ANY CONTEXT privilege before you can drop the context with the DROP CONTEXT statement. As with local application contexts, the global application context is created and stored in the database schema of a security administrator.

The ownership of the global application context is as follows: Even though a user who has been granted the CREATE ANY CONTEXT and DROP ANY CONTEXT privileges can create and drop the global application context, it is owned by the SYS schema. Oracle Database associates the context with the schema account that created it, but if you drop this user, the context still exists in the SYS schema. As user SYS, you can drop the application context.

[Example 6–9](#_bookmark1259) shows how to create the global application context global\_hr\_ctx, which is set by the hr\_ctx\_pkg package.

***Example 6–9 Creating a Global Application Context***

CREATE OR REPLACE CONTEXT global\_hr\_ctx USING hr\_ctx\_pkg ACCESSED GLOBALLY;

#### Creating a PL/SQL Package to Manage a Global Application Context

This section contains:

* + [About the Package That Manages the Global Application Context](#_bookmark1262)
  + [How Editions Affects the Results of a Global Application Context PL/SQL Package](#_bookmark1264)
  + [Setting the DBMS\_SESSION.SET\_CONTEXT username and client\_id Parameters](#_bookmark1269)
  + [Sharing Global Application Context Values for All Database Users](#_bookmark1274)
  + [Setting a Global Context for Database Users Who Move Between Applications](#_bookmark1278)
  + [Setting a Global Application Context for Nondatabase Users](#_bookmark1282)
  + [Clearing Session Data When the Session Closes](#_bookmark1290)

For detailed information about the DBMS\_SESSION package, see *Oracle Database PL/SQL Packages and Types Reference*.

###### About the Package That Manages the Global Application Context

The task of the PL/SQL package that you associate with a global application context is to use the DBMS\_SESSION package to set and clear the global application context values. You must have the EXECUTE privilege for the DBMS\_SESSION package before you use its procedures. Typically, you create and store this package in the database schema of a security administrator. The SYS schema owns the DBMS\_SESSION package.

Unlike PL/SQL packages used to set a local application context, you do not include a SYS\_CONTEXT function to get the user session data. You do not need to include this function because the owner of the session, recorded in the USERENV context, is the same for every user who is connecting.

You can run the procedures within the PL/SQL package for a global application context at any time. You do not need to create logon and logoff triggers to execute the package procedures associated with the global application context. A common practice is to run the package procedures from within the database application. Additionally, for nondatabase users, you use middle-tier applications to get and set client session IDs.

###### How Editions Affects the Results of a Global Application Context PL/SQL Package

You can control the behavior of a global application context package—and for packages used for Oracle Virtual Private Database and fine-grained audit policies, as well—across multiple editions, as follows:

* + **Have the PL/SQL package results be the same across all editions.** To do so, create the package in the schema of a user who has not been editions enabled. To find users who are not editions enabled, you can query the DBA\_USERS and USER\_USERS data dictionary views. Remember that SYS, SYSTEM, and other default Oracle Database administrative accounts that are listed in the DBA\_REGISTRY data dictionary view are not and cannot be editions enabled.
  + **Have the PL/SQL package results depend on the current state of the edition in which the package is run.** Here, the results may be different across all editions to which the package applies. In this case, create the package in the schema of a user who has been editions enabled. If the schema is editions enabled, then it is likely that there will be different actual copies of the package in different editions, where each copy has different behavior. This is useful for the following types of scenarios:
* The package must use a new application context.
* The package must encode input values using a different scheme.

**–** The package must apply different validation rules for users logging in to the database.

For PL/SQL packages that set a global application context, use a single getter function to wrap the primitive SYS\_CONTEXT calls that will read the key-value application context pairs. You can put this getter function in the same package as the application context setter procedure. This approach lets you tag the value for the application context key to reflect a relevant concept. For example, the tag can be the edition in which the setter function is actual. Or, it can be the current edition of the session that set the context, which you can find by using SYS\_ CONTEXT('USERENV', 'CURRENT\_EDITION\_NAME'). This tag can be any specific notion to which the setter function applies.

**See Also:** *Oracle Database Advanced Application Developer's Guide* for detailed information about editions

###### Setting the DBMS\_SESSION.SET\_CONTEXT username and client\_id Parameters

In addition to the namespace, attribute, and value parameters, the DBMS\_ SESSION.SYS\_CONTEXT procedure provides the client\_id and username parameters. Use these settings for global application contexts. [Table 6–2](#_bookmark1272) explains how the combination of these settings controls the type of global application context you can create.

***Table 6–2 Setting the DBMS\_SESSION.SET\_CONTEXT username and client\_id Parameters***

|  |  |
| --- | --- |
| **Combination Settings** | **Result** |
| username set to NULL client\_id set to NULL | This combination enables all users to access the application context. See ["Sharing](#_bookmark1274) [Global Application Context Values for All Database Users" on page 6-26](#_bookmark1274) for more information.  These settings are also used for database session-based application contexts. See ["Using Database Session-Based Application Contexts" on page 6-4](#_bookmark1147) for more information. |
| username set to a value  client\_id set to NULL | This combination enables an application context to be accessed by multiple sessions, as long as the username setting is the same throughout. Ensure that the user name specified is a valid database user. See ["Setting a Global Context for Database Users](#_bookmark1278) [Who Move Between Applications" on page 6-27](#_bookmark1278) for more information. |
| username set to NULL client\_id set to a value | This combination enables an application to be accessed by multiple user sessions, as long as the client\_id parameter is set to the same value throughout. This enables sessions of all users to see the application context values. |
| username set to a value  client\_id set to a value | This combination enables the following two scenarios:   * **Lightweight users.** If the user does not have a database account, the username specified is a connection pool owner. The client\_id setting is then associated with the nondatabase user who is logging in. * **Database users.** If the user is a database user, this combination can be used for stateless Web sessions.   Setting the username parameter in the SET\_CONTEXT procedure to USER calls the Oracle Database-supplied USER function. The USER function specifies the session owner from the application context retrieval process and ensures that only the user who set the application context can access the context. See *Oracle Database SQL Language Reference* for more information about the USER function.  See ["Setting a Global Application Context for Nondatabase Users" on page 6-28](#_bookmark1282) for more information. |

###### Sharing Global Application Context Values for All Database Users

To share global application values for all database users, set the namespace, attribute, and value parameters in the SET\_CONTEXT procedure. In this scenario, *all* users who have database accounts will potentially have access to data in the database.

[Example 6–10](#_bookmark1277) shows how to create a package that sets and clears this type of global application context.

***Example 6–10 Package to Manage Global Application Values for All Database Users***

CREATE OR REPLACE PACKAGE hr\_ctx\_pkg AS

PROCEDURE set\_hr\_ctx(sec\_level IN VARCHAR2); PROCEDURE clear\_hr\_context;

END;

/

CREATE OR REPLACE PACKAGE BODY hr\_ctx\_pkg AS

PROCEDURE set\_hr\_ctx(sec\_level IN VARCHAR2) AS

BEGIN DBMS\_SESSION.SET\_CONTEXT(

namespace => 'global\_hr\_ctx', attribute => 'job\_role', value => sec\_level);

END set\_hr\_ctx;

PROCEDURE clear\_hr\_context AS

BEGIN

DBMS\_SESSION.CLEAR\_CONTEXT('global\_hr\_ctx', 'job\_role'); END clear\_context;

END;

/

In this example:

* + DBMS\_SESSION.SET\_CONTEXT ... END set\_hr\_ctx uses the DBMS\_SESSION.SET\_ CONTEXT procedure to set values for the namespace, attribute, and value parameters. The sec\_level value is specified when the database application runs the hr\_ctx\_pkg.set\_hr\_ctx procedure.

The username and client\_id values are not set, hence, they are NULL. This enables all users (database users) to have access to the values, which is appropriate for server-wide settings.

* + namespace => 'global\_hr\_ctx' sets the namespace to global\_hr\_ctx.
  + attribute => 'job\_role' creates the job\_role attribute.
  + value => sec\_level sets the value for the job\_role attribute to sec\_level.
  + PROCEDURE clear\_hr\_context creates the clear\_hr\_context procedure to clear the context values. See ["Clearing Session Data When the Session Closes" on page 6-32](#_bookmark1290) for more information.

Typically, you execute this procedure within a database application. For example, if all users logging in are clerks, and you want to use "clerk" as a security level, you would embed a call within a database application similar to the following:

BEGIN

hr\_ctx\_pkg.set\_hr\_ctx('clerk');

END;

/

If the procedure successfully completes, you can check the application context setting as follows:

SELECT SYS\_CONTEXT('global\_hr\_ctx', 'job\_role') job\_role FROM DUAL;

JOB\_ROLE

-----------

clerk

To clear this application context, enter the following:

BEGIN

hr\_ctx\_pkg.clear\_hr\_context; END;

/

To check that it is really cleared, the following SELECT statement should return no values:

SELECT SYS\_CONTEXT('global\_hr\_ctx', 'job\_role') job\_role FROM DUAL;

JOB\_ROLE

-----------

**Note:** If Oracle Database returns error messages saying that you have insufficient privileges, ensure that you have correctly created the global application context. You should also query the DBA\_CONTEXT database view to ensure that your settings are correct, for example, that you are calling the procedure from the schema in which you created it.

If NULL is returned, then you may have inadvertently set a client identifier. To clear the client identifier, run the following procedure:

EXEC DBMS\_SESSION.CLEAR\_IDENTIFIER;

###### Setting a Global Context for Database Users Who Move Between Applications

To set a global application context for database users who move from one application to another, particularly when the applications have different access requirements, include the username parameter in the SET\_CONTEXT procedure. This parameter specifies that the same schema be used for all sessions.

Use the following SET\_CONTEXT parameters:

* + - namespace
    - attribute
    - value
    - username

Oracle Database matches the username value so that the other application can recognize the application context. This enables the user to move between applications.

By omitting the client\_id setting, its value is NULL, the default. This means that values can be seen by multiple sessions if the username setting is the same for a database user who maintains the same context in different applications. For example, you can have a

suite of applications that control user access with Oracle Virtual Private Database policies, with each user restricted to a job role.

[Example 6–11](#_bookmark1281) demonstrates how to set the username parameter so that a specific user can move between applications. This example is similar to the package that was created in [Example 6–10 on page 6-26](#_bookmark1277). The use of the username parameter is indicated in **bold** typeface.

***Example 6–11 Package to Manage Global Application Context Values for a User Moving Between Applications***

CREATE OR REPLACE PACKAGE hr\_ctx\_pkg AS

PROCEDURE set\_hr\_ctx(sec\_level IN VARCHAR2, **user\_name IN VARCHAR2);**

PROCEDURE clear\_hr\_context; END;

/

CREATE OR REPLACE PACKAGE BODY hr\_ctx\_pkg AS

PROCEDURE set\_hr\_ctx(sec\_level IN VARCHAR2, **user\_name IN VARCHAR2)**

AS

BEGIN DBMS\_SESSION.SET\_CONTEXT(

namespace => 'global\_hr\_ctx', attribute => 'job\_role', value => sec\_level, **username => user\_name);**

END set\_hr\_ctx;

PROCEDURE clear\_hr\_context AS

BEGIN

DBMS\_SESSION.CLEAR\_CONTEXT('global\_hr\_ctx'); END clear\_context;

END;

/

Typically, you execute this procedure within a database application by embedding a call similar to the following example. Ensure that the value for the user\_name parameter (scott in this case) is a valid database user name.

BEGIN

hr\_ctx\_pkg.set\_hr\_ctx('clerk', 'scott'); END;

A secure way to manage this type of global application context is within your applications, embed code to grant a secure application role to the user. This code should include EXECUTE permissions on the trusted PL/SQL package that sets the application context. In other words, the application, not the user, will set the context for the user.

###### Setting a Global Application Context for Nondatabase Users

When a nondatabase user, that is, a user who is not known to the database (such as a Web application user), starts a client session, the application server generates a client session ID. Once this ID is set on the application server, it must be passed to the database server side. You do this by using the DBMS\_SESSION.SET\_IDENTIFIER procedure to set the client session ID. To set the context, you set the client\_id parameter in the DBMS\_SESSION.SET\_CONTEXT procedure, in a PL/SQL procedure on

the server side. This enables you to manage the application context globally, yet each client sees only his or her assigned application context.

The client\_id value is the key here to getting and setting the correct attributes for the global application context. Remember that the client identifier is controlled by the middle-tier application, and once set, it remains open until it is cleared.

A typical way to manage this type of application context is to place the session\_id value (client\_identifier) in a cookie, and send it to the end user’s HTML page so that is returned on the next request. A lookup table in the application should also keep client identifiers so that they are prevented from being reused for other users and to implement an end-user session time out.

For nondatabase users, configure the following SET\_CONTEXT parameters:

* + - namespace
    - attribute
    - value
    - username
    - client\_id

[Example 6–12](#_bookmark1289) shows how to create a package that manages this type of global application context.

***Example 6–12 Package to Manage Global Application Context Values for Nondatabase Users***

CREATE OR REPLACE PACKAGE hr\_ctx\_pkg AS

PROCEDURE set\_session\_id(session\_id\_p IN NUMBER); PROCEDURE set\_hr\_ctx(sec\_level\_attr IN VARCHAR2,

sec\_level\_val IN VARCHAR2);

PROCEDURE clear\_hr\_session(session\_id\_p IN NUMBER); PROCEDURE clear\_hr\_context;

END;

/

CREATE OR REPLACE PACKAGE BODY hr\_ctx\_pkg AS

session\_id\_global NUMBER;

PROCEDURE set\_session\_id(session\_id\_p IN NUMBER) AS

BEGIN

session\_id\_global := session\_id\_p; DBMS\_SESSION.SET\_IDENTIFIER(session\_id\_p);

END set\_session\_id;

PROCEDURE set\_hr\_ctx(sec\_level\_attr IN VARCHAR2, sec\_level\_val IN VARCHAR2)

AS BEGIN

DBMS\_SESSION.SET\_CONTEXT(

namespace => 'global\_hr\_ctx', attribute => sec\_level\_attr, value => sec\_level\_val, username => USER,

client\_id => session\_id\_global); END set\_hr\_ctx;

PROCEDURE clear\_hr\_session(session\_id\_p IN NUMBER) AS

BEGIN

DBMS\_SESSION.SET\_IDENTIFIER(session\_id\_p); DBMS\_SESSION.CLEAR\_IDENTIFIER;

END clear\_hr\_session;

PROCEDURE clear\_hr\_context AS

BEGIN

DBMS\_SESSION.CLEAR\_CONTEXT('global\_hr\_ctx', session\_id\_global); END clear\_hr\_context;

END;

/

In this example:

* + session\_id\_global NUMBER creates the session\_id\_global variable, which will hold the client session ID. The session\_id\_global variable is referenced throughout the package definition, including the procedure that creates the global application context attributes and assigns them values. This means that the global application context values will always be associated with this particular session ID.
    - PROCEDURE set\_session\_id ... END set\_session\_id creates the set\_session\_ id procedure, which writes the client session ID to the session\_id\_global variable.
    - PROCEDURE set\_hr\_ctx ... END set\_hr\_ctx creates the set\_hr\_ctx procedure, which creates global application context attributes and enables you to assign values to these attributes. Within this procedure:
      * username => USER specifies the username value. This example sets it by calling the Oracle Database-supplied USER function, which adds the session owner from the context retrieval process. The USER function ensures that only the user who set the application context can access the context. See *Oracle Database SQL Language Reference* for more information about the USER function.

If you had specified NULL (the default for the username parameter), then any user can access the context.

Setting both the username and client\_id values enables two scenarios. For lightweight users, set the username parameter to a connection pool owner (for example, APPS\_USER), and then set client\_id to the client session ID. If you want to use a stateless Web session, set the user\_name parameter to the same database user who has logged in, and ensure that this user keeps the same client session ID. See ["Setting the DBMS\_SESSION.SET\_CONTEXT username](#_bookmark1269) [and client\_id Parameters" on page 6-25](#_bookmark1269) for an explanation of how different username and client\_id settings work.

* + - * client\_id => session\_id\_global specifies client\_id value. This example sets it to the session\_id\_global variable. This associates the context settings defined here with a specific client session ID, that is, the one that is set when you run the set\_session\_id procedure. If you specify the client\_id parameter default, NULL, then the global application context settings could be used by any session.
    - PROCEDURE clear\_hr\_session ... END clear\_hr\_session creates the clear\_hr\_ session procedure to clear the client session identifier. **Line 33** sets it to ensure that you are clearing the correct session ID, that is, the one stored in variable session\_id\_p defined in **Line 10**.
    - PROCEDURE clear\_hr\_context ... END clear\_hr\_context creates the clear\_hr\_ context procedure, so that you can clear the context settings for the current user session, which were defined by the global\_hr\_ctx variable. See ["Clearing Session Data When the Session Closes" on page 6-32](#_bookmark1290) for more information.

**See Also:**

* ["Tutorial: Creating a Global Application Context That Uses a Client Session ID" on page 6-35](#_bookmark1299) for a tutorial that demonstrates how a global application context used for client session IDs works
* ["Setting the Client Session ID Using a Middle-Tier Application" on page 6-33](#_bookmark1296)
* ["Using Client Identifiers to Identify Application Users Not Known to the Database" on page 3-44](#_bookmark467) for information about how client identifiers work on middle-tier systems

###### Clearing Session Data When the Session Closes

The application context exists entirely within memory. When the user exits a session, you need to clear the context for the client\_identifier value. This releases memory and prevents other users from accidentally using any left over values.

To clear session data when a user exits a session, use either of the following methods in the server-side PL/SQL package:

* + **Clearing the client identifier when a user exits a session.** Use the DBMS\_ SESSION.CLEAR\_IDENTIFIER procedure. For example:

DBMS\_SESSION.CLEAR\_IDENTIFIER;

* + **Continuing the session but still clearing the context.** If you want the session to continue, but you still need to clear the context, use the DBMS\_SESSION.CLEAR\_ CONTEXT or the DBMS\_SESSION.CLEAR\_ALL\_CONTEXT procedure. For example:

DBMS\_SESSION.CLEAR\_CONTEXT('my\_ctx', 'my\_attribute');

The CLEAR\_CONTEXT procedure clears the context for the current user. To clear the context values for all users, for example, when you need to shut down the application server, use the CLEAR\_ALL\_CONTEXT procedure.

Global application context values are available until they are cleared, so you should use CLEAR\_CONTEXT or CLEAR\_ALL\_CONTEXT to ensure that other sessions do not have access to these values. Be aware that any changes in the context value are reflected immediately and subsequent calls to access the value through the SYS\_ CONTEXT function will return the most recent value.

#### Embedding Calls in Middle-Tier Applications to Manage the Client Session ID

This section contains:

* + [About Managing Client Session IDs Using a Middle-Tier Application](#_bookmark1293)
  + [Retrieving the Client Session ID Using a Middle-Tier Application](#_bookmark1294)
  + [Setting the Client Session ID Using a Middle-Tier Application](#_bookmark1296)
  + [Clearing Session Data Using a Middle-Tier Application](#_bookmark1298)

###### About Managing Client Session IDs Using a Middle-Tier Application

The application server generates the client session ID. From a middle-tier application, you can get, set, and clear the client session IDs. To do so, embed either Oracle Call Interface (OCI) calls or DBMS\_SESSION PL/SQL package procedures into the middle-tier application code.

The application authenticates the user, sets the client identifier, and sets it in the current session. The PL/SQL package SET\_CONTEXT sets the client\_identifier value in the application context. See ["Setting a Global Application Context for Nondatabase](#_bookmark1282) [Users" on page 6-28](#_bookmark1282) for more information.

###### Retrieving the Client Session ID Using a Middle-Tier Application

When a user starts a client session, the application server generates a client session ID. To retrieve this client ID, you can use the OCIStmtExecute call with any of the following statements:

SELECT SYS\_CONTEXT('userenv', 'client\_identifier') FROM dual; SELECT CLIENT\_IDENTIFIER from V$SESSION;

SELECT value FROM session\_context WHERE attribute='CLIENT\_IDENTIFIER';

[Example 6–13](#_bookmark1295) shows how to use the OCIStmtExecute call to retrieve a client session ID value.

***Example 6–13 Using OCIStmtExecute to Retrieve a Client Session ID Value***

oratext clientid[31];

OCIDefine \*defnp1 = (OCIDefine \*) 0; OCIStmt \*statementhndle;

oratext \*selcid = (oratext \*)"SELECT SYS\_CONTEXT('userenv', 'client\_identifier') FROM DUAL";

OCIStmtPrepare(statementhndle, errhp, selcid,

(ub4) strlen((char \*) selcid), (ub4) OCI\_NTV\_SYNTAX, (ub4) OCI\_DEFAULT);

OCIDefineByPos(statementhndle, &defnp1, errhp, 1, (dvoid \*)clientid, 31, SQLT\_STR, (dvoid \*) 0, (ub2 \*) 0, (ub2 \*) 0, OCI\_DEFAULT);

OCIStmtExecute(servhndle, statementhndle, errhp, (ub4) 1, (ub4) 0, (CONST OCISnapshot \*) NULL, (OCISnapshot \*) NULL, OCI\_DEFAULT);

printf("CLIENT\_IDENTIFIER = %s \n", clientid);

In this example:

* + - oratext, OCIDefine, OCIStmt, and oratext create variables to store the client session ID, reference call for OCIDefine, the statement handle, and the SELECT statement to use.
    - OCIStmtPrepar prepares the statement selcid for execution.
    - OCIDefineByPos defines the output variable clientid for client session ID.
    - OCIStmtExecute executes the statement in the selcid variable.
    - printf prints the formatted output for the retrieved client session ID.

###### Setting the Client Session ID Using a Middle-Tier Application

After you use the OCIStmtExecute call to retrieve the client session ID, you are ready to set this ID. The DBMS\_SESSION.SET\_CONTEXT procedure in the server-side PL/SQL package then sets this session ID and optionally, overwrites the application context values.

Ensure that the middle-tier application code checks that the client session ID value (for example, the value written to user\_id in the previous examples) matches the client\_ id setting defined in the server-side DBMS\_SESSION.SET\_CONTEXT procedure. The sequence of calls on the application server side should be as follows:

1. **Get the current client session ID.** The session should already have this ID, but it is safer to ensure that it truly has the correct value.
2. **Clear the current client session ID.** This prepares the application to service a request from a different end user.
3. **Set the new client session ID or the client session ID that has been assigned to the end user.** This ensures that the session is using a different set of global application context values.

You can use the following methods to set the client session ID on the application server side:

* + **Oracle Call Interface.** Set the OCI\_ATTR\_CLIENT\_IDENTIFIER attribute in an OCIAttrSet OCI call. This attribute sets the client identifier in the session handle to track the end user identity.

The following example shows how to use OCIAttrSet with the ATTR\_CLIENT\_ IDENTIFIER parameter. The user\_id setting refers to a variable that stores the ID of the user who is logging on.

OCIAttrSet((void \*)session\_handle, (ub4) OCI\_HTYPE\_SESSION, (void \*) user\_id, (ub4)strlen(user\_id), OCI\_ATTR\_CLIENT\_IDENTIFIER, error\_handle);

* + **DBMS\_SESSION package.** Use the DBMS\_SESSION.SET\_IDENTIFIER procedure to set the client identifier for the global application context. For example, assuming you are storing the ID of the user logging on in a variable called user\_id, you would enter the following line into the middle-tier application code:

DBMS\_SESSION.SET\_IDENTIFIER(user\_id);

**Note:** When the application generates a session ID for use as a CLIENT\_IDENTIFIER, then the session ID must be suitably random and protected over the network by encryption. If the session ID is not random, then a malicious user could guess the session ID and access the data of another user. If the session ID is not encrypted over the network, then a malicious user could retrieve the session ID and access the connection.

You can encrypt the session ID by using Oracle Advanced Security. See *Oracle Database Advanced Security Administrator's Guide* for more information. To learn more about encrypting data over a network, see *Oracle Database 2 Day + Security Guide*.

For both OCIAttrSet and DBMS\_SESSION.SET\_IDENTIFIER, you can check the value of this identifier as follows:

SELECT SYS\_CONTEXT('userenv', 'client\_identifier') FROM dual;

Another way to check this value is to query the V$SESSION view:

SELECT CLIENT\_IDENTIFIER from V$SESSION;

###### Clearing Session Data Using a Middle-Tier Application

The application context exists entirely within memory. When the user exits a session, you need to clear the context for the client\_identifier value. This releases memory and prevents other users from accidentally using any left over values

To clear session data when a user exits a session, use either of the following methods in the middle-tier application code:

* + **Clearing the client identifier when a user exits a session.** Use the DBMS\_ SESSION.CLEAR\_IDENTIFIER procedure. For example:

DBMS\_SESSION.CLEAR\_IDENTIFIER;

* + - **Continuing the session but still clearing the context.** If you want the session to continue, but you still need to clear the context, use the DBMS\_SESSION.CLEAR\_ CONTEXT or the DBMS\_SESSION.CLEAR\_ALL\_CONTEXT procedure. For example:

DBMS\_SESSION.CLEAR\_CONTEXT(*namespace*, *client\_identifier*, *attribute*);

The CLEAR\_CONTEXT procedure clears the context for the current user. To clear the context values for all users, for example, when you need to shut down the application server, use the CLEAR\_ALL\_CONTEXT procedure.

Global application context values are available until they are cleared, so you should use CLEAR\_CONTEXT or CLEAR\_ALL\_CONTEXT to ensure that other sessions do not have access to these values.

#### Tutorial: Creating a Global Application Context That Uses a Client Session ID

This section contains:

* + - [About This Tutorial](#_bookmark1301)
    - [Step 1: Create User Accounts](#_bookmark1306)
    - [Step 2: Create the Global Application Context](#_bookmark1307)
    - [Step 3: Create a Package for the Global Application Context](#_bookmark1308)
    - [Step 4: Test the Global Application Context](#_bookmark1309)
    - [Step 5: Remove the Components for This Tutorial](#_bookmark1311)

###### About This Tutorial

This tutorial shows how to create a global application context that uses a client session ID for a lightweight user application. It demonstrates how to control nondatabase user access by using a connection pool.

###### Step 1: Create User Accounts

You must create two users for this example: a security administrator who will manage the application context and its package, and a user account that owns the connection pool.

In this tutorial:

1. Log on to SQL\*Plus as SYS and connect using AS SYSDBA.

sqlplus sys as sysdba Enter password: *password*

1. Create the sysadmin\_ctx account, who will administer the global application context.

GRANT CREATE SESSION, CREATE ANY CONTEXT, CREATE PROCEDURE TO sysadmin\_ctx

IDENTIFIED BY *password*;

GRANT EXECUTE ON DBMS\_SESSION TO sysadmin\_ctx;

Replace *password* with a password that is secure. See ["Minimum Requirements for](#_bookmark197) [Passwords" on page 3-3](#_bookmark197) for more information.

1. Create the database account apps\_user, who will own the connection pool.

GRANT CREATE SESSION TO apps\_user IDENTIFIED BY *password*;

Replace *password* with a password that is secure. See ["Minimum Requirements for](#_bookmark197) [Passwords" on page 3-3](#_bookmark197) for more information.

###### Step 2: Create the Global Application Context

1. Log on as the security administrator sysadmin\_ctx.

CONNECT sysadmin\_ctx Enter password: *password*

1. Create the cust\_ctx global application context.

CREATE CONTEXT global\_cust\_ctx USING cust\_ctx\_pkg ACCESSED GLOBALLY;

The cust\_ctx context is created and associated with the schema of the security administrator sysadmin\_ctx. However, the SYS schema owns the application context.

###### Step 3: Create a Package for the Global Application Context

1. As sysadmin\_ctx, create the following PL/SQL package:

CREATE OR REPLACE PACKAGE cust\_ctx\_pkg AS

PROCEDURE set\_session\_id(session\_id\_p IN NUMBER); PROCEDURE set\_cust\_ctx(sec\_level\_attr IN VARCHAR2,

sec\_level\_val IN VARCHAR2);

PROCEDURE clear\_hr\_session(session\_id\_p IN NUMBER); PROCEDURE clear\_hr\_context;

END;

/

CREATE OR REPLACE PACKAGE BODY cust\_ctx\_pkg AS

session\_id\_global NUMBER;

PROCEDURE set\_session\_id(session\_id\_p IN NUMBER) AS

BEGIN

session\_id\_global := session\_id\_p; DBMS\_SESSION.SET\_IDENTIFIER(session\_id\_p);

END set\_session\_id;

PROCEDURE set\_cust\_ctx(sec\_level\_attr IN VARCHAR2, sec\_level\_val IN VARCHAR2) AS

BEGIN DBMS\_SESSION.SET\_CONTEXT(

namespace => 'global\_cust\_ctx', attribute => sec\_level\_attr, value => sec\_level\_val,

username => USER, *-- Retrieves the session user, in this case, apps\_user*

client\_id => session\_id\_global); END set\_cust\_ctx;

PROCEDURE clear\_hr\_session(session\_id\_p IN NUMBER) AS

BEGIN

DBMS\_SESSION.SET\_IDENTIFIER(session\_id\_p); DBMS\_SESSION.CLEAR\_IDENTIFIER;

END clear\_hr\_session; PROCEDURE clear\_hr\_context

AS BEGIN

DBMS\_SESSION.CLEAR\_CONTEXT('global\_cust\_ctx', session\_id\_global); END clear\_hr\_context;

END;

/

For a detailed explanation of how this type of package works, see [Example 6–12](#_bookmark1289) [on page 6-30](#_bookmark1289).

1. Grant EXECUTE privileges on the cust\_ctx\_pkg package to the connection pool owner, apps\_user.

GRANT EXECUTE ON cust\_ctx\_pkg TO apps\_user;

###### Step 4: Test the Global Application Context

At this stage, you are ready to explore how this global application context and session ID settings work.

1. Log on to SQL\*Plus as the connection pool owner, user apps\_user.

CONNECT apps\_user

Enter password: *password*

1. When the connection pool user logs on, the application sets the client session identifier as follows:

BEGIN

sysadmin\_ctx.cust\_ctx\_pkg.set\_session\_id(34256); END;

/

You can test and check the value of the client session identifier as follows:

* 1. Connect to SQL\*Plus as the connection pool user apps\_user.
  2. Set the session ID:

EXEC sysadmin\_ctx.cust\_ctx\_pkg.set\_session\_id(34256);

* 1. Check the session ID:

SELECT SYS\_CONTEXT('userenv', 'client\_identifier') FROM dual;

The following output should appear:

SYS\_CONTEXT('USERENV','CLIENT\_IDENTIFIER')

-------------------------------------------------- 34256

1. As user apps\_user, set the global application context as follows:

EXEC sysadmin\_ctx.cust\_ctx\_pkg.set\_cust\_ctx('Category', 'Gold Partner'); EXEC sysadmin\_ctx.cust\_ctx\_pkg.set\_cust\_ctx('Benefit Level', 'Highest');

(In a real-world scenario, the middle-tier application would set the global application context values, similar to how the client session identifier was set in Step [2](#_bookmark1310).)

1. Enter the following SELECT SYS\_CONTEXT statement to check that the settings were successful:

col category format a13

col benefit\_level format a14

SELECT SYS\_CONTEXT('global\_cust\_ctx', 'Category') category, SYS\_ CONTEXT('global\_cust\_ctx', 'Benefit Level') benefit\_level FROM dual;

The following output should appear:

CATEGORY BENEFIT\_LEVEL

------------- --------------

Gold Partner Highest

What apps\_user has done here, within the client session 34256, is set a global application context on behalf of a nondatabase user. This context sets the Category and Benefit Level DBMS\_SESSION.SET\_CONTEXT attributes to be Gold Partner and Highest, respectively. The context exists only for user apps\_user with client ID 34256. When a nondatabase user logs in, behind the scenes, he or she is really logging on as the connection pool user apps\_user. Hence, the Gold Partner and Highest context values are available to the nondatabase user.

Suppose the user had been a database user and could log in without using the intended application. (For example, the user logs in using SQL\*Plus.) Because the user has not logged in through the connection pool user apps\_user, the global application context appears empty to our errant user. This is because the context was created and set under the apps\_user session. If the user runs the SELECT SYS\_CONTEXT statement, the following output appears:

CATEGORY BENEFIT\_LEVEL

------------- --------------

Next, try the following test:

1. As user apps\_user, clear the session ID.

EXEC sysadmin\_ctx.cust\_ctx\_pkg.clear\_hr\_session(34256);

1. Check the global application context settings again.

SELECT SYS\_CONTEXT('global\_cust\_ctx', 'Category') category, SYS\_ CONTEXT('global\_cust\_ctx', 'Benefit Level') benefit\_level FROM dual;

CATEGORY BENEFIT\_LEVEL

------------- --------------

Because apps\_user has cleared the session ID, the global application context settings are no longer available.

1. Restore the session ID to 34256, and then check the context values.

EXEC sysadmin\_ctx.cust\_ctx\_pkg.set\_session\_id(34256);

SELECT SYS\_CONTEXT('global\_cust\_ctx', 'Category') category, SYS\_ CONTEXT('global\_cust\_ctx', 'Benefit Level') benefit\_level FROM dual;

The following output should appear:

CATEGORY BENEFIT\_LEVEL

------------- --------------

Gold Partner Highest

As you can see, resetting the session ID to 34256 brings the application context values back again. To summarize, the global application context must be set only *once* for this user, but the client session ID must be set *each time* the user logs on.

1. Now try clearing and then checking the global application context values.

EXEC sysadmin\_ctx.cust\_ctx\_pkg.clear\_hr\_context;

SELECT SYS\_CONTEXT('global\_cust\_ctx', 'Category') category, SYS\_ CONTEXT('global\_cust\_ctx', 'Benefit Level') benefit\_level FROM dual;

The following output should appear:

CATEGORY BENEFIT\_LEVEL

------------- --------------

At this stage, the client session ID, 34256 is still in place, but the application context settings no longer exist. This enables you to continue the session for this user but without using the previously set application context values.

###### Step 5: Remove the Components for This Tutorial

* 1. Log on as SYS and connect using AS SYSDBA.

CONNECT sys/as sysdba Enter password: *password*

* 1. Drop the global application context.

DROP CONTEXT global\_cust\_ctx;

Remember that even though sysadmin\_ctx created the global application context, it is owned by the SYS schema.

* 1. Drop the two sample users.

DROP USER sysadmin\_ctx CASCADE; DROP USER apps\_user;

#### Global Application Context Processes

This section contains:

* [Simple Global Application Context Process](#_bookmark1313)
* [Global Application Context Process for Lightweight Users](#_bookmark1315)

###### Simple Global Application Context Process

Consider the application server, AppSvr, that has assigned the client identifier 12345 to client SCOTT. The AppSvr application uses the SCOTT user to create a session (that is, it is not a connection pool.) The value assigned to the context attribute can come from anywhere, for example, from running a SELECT statement on a table that holds the responsibility codes for users. When the application context is populated, it is stored in memory. As a result, any action that needs the responsibility code can access it quickly with SYS\_CONTEXT call, without the overhead of accessing a table. The only advantage of a global context over a local context in this case is if SCOTT were changing applications frequently and used the same context in each application.

The following steps show how the global application context process sets the client identifier for SCOTT:

1. The administrator creates a global context namespace by using the following statement:

CREATE OR REPLACE CONTEXT hr\_ctx USING hr.init ACCESSED GLOBALLY;

1. The administrator creates a PL/SQL package for the hr\_ctx application context to indicate that, for this client identifier, there is an application context called responsibility with a value of 13 in the HR namespace.:

CREATE OR REPLACE PROCEDURE hr.init AS

BEGIN DBMS\_SESSION.SET\_CONTEXT(

namespace => 'hr\_ctx', attribute => 'responsibility', value => '13',

username => 'SCOTT', client\_id => '12345' );

END;

/

This PL/SQL procedure is stored in the HR database schema, but typically it is stored in the schema of the security administrator.

1. The AppSvr application issues the following command to indicate the connecting client identity each time scott uses AppSvr to connect to the database:

EXEC DBMS\_SESSION.SET\_IDENTIFIER('12345');

1. When there is a SYS\_CONTEXT('hr\_ctx','responsibility') call within the database session, the database matches the client identifier, 12345, to the global context, and then returns the value 13.
2. When exiting this database session, AppSvr clears the client identifier by issuing the following procedure:

EXEC DBMS\_SESSION.CLEAR\_IDENTIFIER( );

1. To release the memory used by the application context, AppSvr issues the following procedure:

DBMS\_SESSION.CLEAR\_CONTEXT('hr\_ctx', '12345');

CLEAR\_CONTEXT is needed when the user session is no longer active, either on an explicit logout, timeout, or other conditions determined by the AppSvr application.

**Note:** After a client identifier in a session is cleared, it becomes a NULL value. This implies that subsequent SYS\_CONTEXT calls only retrieve application contexts with NULL client identifiers, until the client identifier is set again using the SET\_IDENTIFIER interface.

###### Global Application Context Process for Lightweight Users

The following steps show the global application context process for a lightweight user application. The lightweight user, robert, is not known to the database through the application.

1. The administrator creates the global context namespace by using the following statement:

CREATE CONTEXT hr\_ctx USING hr.init ACCESSED GLOBALLY;

1. The HR application server, AppSvr, starts and then establishes multiple connections to the HR database as the appsmgr user.
2. User robert logs in to the HR application server.
3. AppSvr authenticates robert to the application.
4. AppSvr assigns a temporary session ID (or uses the application user ID), 12345, for this connection.
5. The session ID is returned to the Web browser used by robert as part of a cookie or is maintained by AppSvr.
6. AppSvr initializes the application context for this client by calling the hr.init

package, which issues the following statements:

DBMS\_SESSION.SET\_CONTEXT( 'hr\_ctx', 'id', 'robert', 'APPSMGR', 12345 ); DBMS\_SESSION.SET\_CONTEXT( 'hr\_ctx', 'dept', 'sales', 'APPSMGR', 12345 );

1. AppSvr assigns a database connection to this session and initializes the session by issuing the following statement:

DBMS\_SESSION.SET\_IDENTIFIER( 12345 );

1. All SYS\_CONTEXT calls within this database session return application context values that belong only to the client session.

For example, SYS\_CONTEXT('hr','id') returns the value robert.

1. When finished with the session, AppSvr issues the following statement to clean up the client identity:

DBMS\_SESSION.CLEAR\_IDENTIFIER ( );

Even if another user logged in to the database, this user cannot access the global context set by AppSvr, because AppSvr specified that only the application with user APPSMGR logged in can see it. If AppSvr used the following, then any user session with client ID set to 12345 can see the global context:

DBMS\_SESSION.SET\_CONTEXT( 'hr\_ctx', 'id', 'robert', NULL , 12345 ); DBMS\_SESSION.SET\_CONTEXT( 'hr\_ctx', 'dept', 'sales', NULL , 12345 );

Setting USERNAME to NULL enables different users to share the same context.

**Note:** Be aware of the security implication of different settings of the global context. NULL in the user name means that any user can access the global context. A NULL client ID in the global context means that a session with an uninitialized client ID can access the global context. To ensure that only the user who has logged on can access the session, specify USER instead of NULL.

You can query the client identifier set in the session as follows:

SELECT SYS\_CONTEXT('USERENV','CLIENT\_IDENTIFIER') FROM dual;

The following output should appear:

SYS\_CONTEXT('USERENV','CLIENT\_IDENTIFIER')

------------------------------------------------- 12345

A security administrator can see which sessions have the client identifier set by querying the V$SESSION view for the CLIENT\_IDENTIFIER and USERNAME, for example:

COL client\_identifier format a18

SELECT CLIENT\_IDENTIFIER, USERNAME from V$SESSION;

The following output should appear:

CLIENT\_IDENTIFIER USERNAME

------------------ -------- 12345 APPSMGR

To check the amount of global context area (in bytes) being used, use the following query:

SELECT SYS\_CONTEXT('USERENV','GLOBAL\_CONTEXT\_MEMORY') FROM dual;

The following output should appear:

SYS\_CONTEXT('USERENV','GLOBAL\_CONTEXT\_MEMORY')

---------------------------------------------- 584

**See Also:** For more information about using the CLIENT\_ IDENTIFIER predefined attribute of the USERENV application context:

* + ["Using the CLIENT\_IDENTIFIER Attribute to Preserve User Identity" on page 3-45](#_bookmark474)
  + *Oracle Database SQL Language Reference*
  + *Oracle Call Interface Programmer's Guide*

### Using Client Session-Based Application Contexts

This section contains:

* + [About Client Session-Based Application Contexts](#_bookmark1318)
  + [Setting a Value in the CLIENTCONTEXT Namespace](#_bookmark1323)
  + [Retrieving the CLIENTCONTEXT Namespace](#_bookmark1325)
  + [Clearing a Setting in the CLIENTCONTEXT Namespace](#_bookmark1328)
  + [Clearing All Settings in the CLIENTCONTEXT Namespace](#_bookmark1330)

#### About Client Session-Based Application Contexts

In a client session-based application context, you use Oracle Call Interface (OCI) functions to set and clear user session information, which is then stored in the User Global Area (UGA).

The advantage of this type of application context is that an individual application can check for specific nondatabase user session data, rather than having the database perform this task. Another advantage is that the calls to set the application context value are included in the next call to the server, which improves performance.

However, be aware that application context security is compromised with a client session-based application context: any application user can set the client application context, and no check is performed in the database.

You configure the client session-based application context for the client application only. You do not configure any settings on the database server to which the client connects. Any application context settings in the database server do not affect the client session-based application context.

To configure a client session-based application context, use the OCIAppCtxSet OCI function. A client session-based application context uses the CLIENTCONTEXT

namespace, updatable by any OCI client or by the existing DBMS\_SESSION package for application context. Oracle Database performs no privilege or package security checks for this type.

The CLIENTCONTEXT namespace enables a single application transaction to both change the user context information and use the same user session handle to service the new user request. You can set or clear individual values for attributes in the CLIENTCONTEXT namespace, or clear all their values.

* + - An OCI client uses the OCIAppCtx function to set variable length data for the namespace, called OCISessionHandle. The OCI network single, round-trip transport sends all the information to the server in one round-trip. On the server side, you can query the application context information by using the SYS\_CONTEXT SQL function on the namespace. For example:
    - A JDBC client uses the oracle.jdbc.internal.OracleConnection function to achieve the same purposes.

Any user can set, clear, or collect the information in the CLIENTCONTEXT namespace, because it is not protected by package-based security.

**See Also:** *Oracle Call Interface Programmer's Guide* for more information about client application contexts

#### Setting a Value inthe CLIENTCONTEXT Namespace

For Oracle Call Interface, to set a value in the CLIENTCONTEXT namespace, use a command in the following syntax:

err = OCIAppCtxSet((void \*) *session\_handle*,(dvoid \*)"CLIENTCONTEXT",(ub4) 13, (dvoid \*)*attribute\_name*, *length\_of\_attribute\_name*

(dvoid \*)*attribute\_value*, *length\_of\_attribute\_value*, errhp, OCI\_DEFAULT);

In this specification:

* + - *session\_handle*: Represents the OCISessionHandle namespace.
    - *attribute\_name*: Name of attribute. For example, responsibility, with a length of 14.
    - *attribute\_value*: Value of attribute. For example, manager, with a length of 7.

**See Also:** "Managing Scalable Platforms" in *Oracle Call Interface Programmer's Guide* for details about the OCIAppCtx function

#### Retrieving the CLIENTCONTEXT Namespace

To retrieve the CLIENTCONTEXT namespace, you can use the Oracle Call Interface

OCIStmtExecute call with either of the following statements:

SELECT SYS\_CONTEXT('CLIENTCONTEXT', '*Attribute-1*') FROM dual;

SELECT VALUE FROM SESSION\_CONTEXT

WHERE NAMESPACE='CLIENTCONTEXT' AND ATTRIBUTE='*attribute-1*';

The *Attribute-1* value can be any attribute value that has already been set in the CLIENTCONTEXT namespace. Oracle Database only retrieves the set attribute; otherwise, it returns NULL. Typically, you set the attribute by using the OCIAppCtxSet call. In addition, you can embed a DBMS\_SESSION.SET\_CONTEXT call in the OCI code to set the attribute value.

[Example 6–13](#_bookmark1295) shows how to use the OCIStmtExecute call to retrieve a client session ID value.

***Example 6–14 Retrieving a Client Session ID Value for Client Session-Based Contexts***

oratext clientid[31];

OCIDefine \*defnp1 = (OCIDefine \*) 0; OCIStmt \*statementhndle;

oratext \*selcid = (oratext \*)"SELECT SYS\_CONTEXT('CLIENTCONTEXT', attribute) FROM DUAL";

OCIStmtPrepare(statementhndle, errhp, selcid, (ub4) strlen((char \*) selcid), (ub4) OCI\_NTV\_SYNTAX, (ub4) OCI\_DEFAULT);

OCIDefineByPos(statementhndle, &defnp1, errhp, 1, (dvoid \*)clientid, 31, SQLT\_STR, (dvoid \*) 0, (ub2 \*) 0, (ub2 \*) 0, OCI\_DEFAULT);

OCIStmtExecute(servhndle, statementhndle, errhp, (ub4) 1, (ub4) 0, (CONST OCISnapshot \*) NULL, (OCISnapshot \*) NULL, OCI\_DEFAULT);

printf("CLIENT\_IDENTIFIER = %s \n", clientid);

In this example:

* + oratext, OCIDefine, OCIStmt, and oratext create variables to store the client session ID, reference call for OCIDefine, the statement handle, and the SELECT statement to use.
  + OCIStmtPrepare prepares the statement selcid for execution.
  + OCIDefineByPos defines the output variable clientid for client session ID.
  + OCIStmtExecute executes the statement in the selcid variable.
  + printf prints the formatted output for the retrieved client session ID.

#### Clearing a Setting inthe CLIENTCONTEXT Namespace

For Oracle Call Interface, to clear a setting in CLIENTCONTEXT, set the value to NULL or to an empty string by using one of the following commands:

(void) OCIAppCtxSet((void \*) *session\_handle*, (dvoid \*)"CLIENTCONTEXT", 13, (dvoid \*)*attribute\_name*, *length\_of\_attribute\_name*, (dvoid \*)0, 0,errhp

OCI\_DEFAULT);

or

(void) OCIAppCtxSet((void \*) *session\_handle*, (dvoid \*)"CLIENTCONTEXT", 13 (dvoid \*)*attribute\_name*, *length\_of\_attribute\_name*, (dvoid \*)"", 0,errhp,

OCI\_DEFAULT);

#### Clearing All Settings inthe CLIENTCONTEXT Namespace

For Oracle Call Interface (OCI), use a command of the following form:

err = OCIAppCtxClearAll((void \*) *session\_handle*,

(dvoid \*)"CLIENTCONTEXT", 13,

errhp, OCI\_DEFAULT);

### Finding Information About Application Contexts

[Table 6–3](#_bookmark1336) lists data dictionary views that you can query to find information about application contexts. For detailed information about these views, see *Oracle Database Reference.*

***Table 6–3 Data Dictionary Views That Display Information about Application Contexts***

|  |  |
| --- | --- |
| **View** | **Description** |
| ALL\_CONTEXT | Describes all context namespaces in the current session for which attributes and values were specified using the DBMS\_SESSION.SET\_CONTEXT procedure. It lists the namespace and its associated schema and PL/SQL package. |
| ALL\_POLICY\_CONTEXTS | Describes the driving contexts defined for the synonyms, tables, and views accessible to the current user. (A driving context is a context used in a Virtual Private Database policy.) |
| DBA\_CONTEXT | Provides all context namespace information in the database. Its columns are the same as those in the ALL\_CONTEXT view, except that it includes the TYPE column. The TYPE column describes how the application context is accessed or initialized. |
| DBA\_POLICY\_CONTEXTS | Describes all driving contexts in the database that were added by the DBMS\_RLS.ADD\_ POLICY\_CONTEXT procedure. Its columns are the same as those in ALL\_POLICY\_ CONTEXTS. |
| SESSION\_CONTEXT | Describes the context attributes and their values set for the current session. |
| USER\_POLICY\_CONTEXTS | Describes the driving contexts defined for the synonyms, tables, and views owned by the current user. Its columns (except for OBJECT\_OWNER) are the same as those in ALL\_POLICY\_CONTEXTS. |
| V$CONTEXT | Lists set attributes in the current session. Users do not have access to this view unless you grant the user the SELECT privilege on it. |
| V$SESSION | Lists detailed information about each current session. Users do not have access to this view unless you grant the user the SELECT privilege on it. |

**Tip:** In addition to these views, check the database trace file if you find errors when running applications that use application contexts. See *Oracle Database Performance Tuning Guide* for more information about trace files. The USER\_DUMP\_DEST initialization parameter sets the directory location of the trace files. You can find the value of this parameter by issuing SHOW PARAMETER USER\_DUMP\_DEST in SQL\*Plus.

# 7

## Using Oracle Virtual Private Database

**to Control Data Access**

This chapter contains:

* + - [About Oracle Virtual Private Database](#_bookmark1343)
    - [Components of an Oracle Virtual Private Database Policy](#_bookmark1370)
    - [Configuring an Oracle Virtual Private Database Policy](#_bookmark1377)
    - [Tutorials: Creating Oracle Virtual Private Database Policies](#_bookmark1465)
    - [How Oracle Virtual Private Database Works with Other Oracle Features](#_bookmark1500)
    - [Finding Information About Oracle Virtual Private Database Policies](#_bookmark1543)

### About Oracle Virtual Private Database

This section contains:

* + - [What Is Oracle Virtual Private Database?](#_bookmark1344)
    - [Benefits of Using Oracle Virtual Private Database Policies](#_bookmark1348)
    - [Which Privileges Are Used to Run Oracle Virtual Private Database Policy Functions?](#_bookmark1357)
    - [Using Oracle Virtual Private Database with an Application Context](#_bookmark1364)

#### What Is Oracle Virtual Private Database?

Oracle Virtual Private Database (VPD) enables you to create security policies to control database access at the row and column level. Essentially, Oracle Virtual Private Database adds a dynamic WHERE clause to a SQL statement that is issued against the table, view, or synonym to which an Oracle Virtual Private Database security policy was applied.

Oracle Virtual Private Database enforces security, to a fine level of granularity, directly on database tables, views, or synonyms. Because you attach security policies directly to these database objects, and the policies are automatically applied whenever a user accesses data, there is no way to bypass security.

When a user directly or indirectly accesses a table, view, or synonym that is protected with an Oracle Virtual Private Database policy, Oracle Database dynamically modifies the SQL statement of the user. This modification creates a WHERE condition (called a predicate) returned by a function implementing the security policy. Oracle Database modifies the statement dynamically, transparently to the user, using any condition that

can be expressed in or returned by a function. You can apply Oracle Virtual Private Database policies to SELECT, INSERT, UPDATE, INDEX, and DELETE statements.

For example, suppose a user performs the following query:

SELECT \* FROM OE.ORDERS;

The Oracle Virtual Private Database policy dynamically appends the statement with a

WHERE clause. For example:

SELECT \* FROM OE.ORDERS WHERE SALES\_REP\_ID = 159;

In this example, the user can only view orders by Sales Representative 159.

If you want to filter the user based on the session information of that user, such as the ID of the user, then you can create the WHERE clause to use an application context. For example:

SELECT \* FROM OE.ORDERS

WHERE SALES\_REP\_ID = SYS\_CONTEXT('USERENV','SESSION\_USER');

**Note:** Oracle Virtual Private Database does not support filtering for DDLs, such as TRUNCATE or ALTER TABLE statements.

#### Benefits of Using Oracle Virtual Private Database Policies

Oracle Virtual Private Database policies provide the following benefits:

* + [Basing Security Policies on Database Objects Rather Than Applications](#_bookmark1350)
  + [Controlling How Oracle Database Evaluates Policy Functions](#_bookmark1353)

###### Basing Security Policies on Database Objects Rather Than Applications

Attaching Oracle Virtual Private Database security policies to database tables, views, or synonyms, rather than implementing access controls in all your applications, provides the following benefits:

* + **Security.** Associating a policy with a database table, view, or synonym can solve a potentially serious application security problem. Suppose a user is authorized to use an application, and then drawing on the privileges associated with that application, wrongfully modifies the database by using an ad hoc query tool, such as SQL\*Plus. By attaching security policies directly to tables, views, or synonyms, fine-grained access control ensures that the same security is in force, no matter how a user accesses the data.
  + **Simplicity.** You add the security policy to a table, view, or synonym only once, rather than repeatedly adding it to each of your table-based, view-based, or synonym-based applications.
  + **Flexibility.** You can have one security policy for SELECT statements, another for INSERT statements, and still others for UPDATE and DELETE statements. For example, you might want to enable Human Resources clerks to have SELECT privileges for all employee records in their division, but to update only salaries for those employees in their division whose last names begin with A through F. Furthermore, you can create multiple policies for each table, view, or synonym.

**Hiểu cách Oracle Database đánh giá các hàm policy**

Thực thi ràng buộc của các hàm nhiều lần có thể ảnh hưởng đến hiệu năng. Bạn có thể kiểm soát hiệu năng bằng cách cấu hình cách để Oracle Database lưu vào bộ nhớ đệm những điều kiện của Oracle Virtual Private Database. Những chọn lựa khả thi:

* + - Đánh giá hành vi một lần cho mỗi câu truy vấn (cơ chế tĩnh)
    - Đánh giá hành vi chỉ khi ngữ cảnh trong hàm thay đổi. ( cơ chế nhạy với ngữ cảnh)

Đánh giá hành vi mỗi khi thực thi ( cơ chế động)

Đọc "Optimizing Performance by Using Oracle Virtual Private Database Policy Types" trang 7-14 chi tiết cấu hình những cơ chế này.

#### Những đặc quyền được sử dụng để chạy các hàm đúng của CSDL riêng ảo?

Để có bảo mật tốt hơn, hàm policy Oracle VPD thực thi nếu được khai báo dưới quyền của người định nghĩa. Không khai báo dưới quyền của khách vì điều này có thể làm bối rối bản thân và người khác khi sự dụng code.

**See Also:** *Oracle Database PL/SQL Language Reference* for detailed information about definer’s rights

#### Sử dụng Oracle VPD với một ngữ cảnh ứng dụng

Bạn có thể sử dụng những ngữ cảnh ứng dụng với chính sách OVPD. Khi bạn tạo một ngữ cảnh ứng dụng, nó lưu vào bộ nhớ đệm thông tin người dùng một cách an toàn.Chỉ gói ứng dụng được chỉ định mới có thể cài đặt môi trường bộ nhớ đệm. Nó không thể bị thay đổi bởi người dùng hay gói bên ngoài. Ngoài ra, bởi vì dữ liệu được lưu trong cache, hiệu năng sẽ được gia tăng. Chương 6, "Sử dụng ngữ cảnh ứng dụng để truy xuất thông tin người dùng" mô tả ngũ cảnh ứng dụng chi tiết.

Ví dụ, giả sử bạn muốn truy cập cơ bản vào ORDERS\_TABtable cột customer ID number. Thay vì truy vấn ID khách hàng phải đăng nhập khi cần nó, bạn có thể lưu trên ngữ cảnh ứng dụng. Tadaa, số của khác hàng luôn sẵn sàng trong session khi bạn cần.

Ngữ cảnh ứng dụng đặc biệt hữu ích nêu chính sách bảo mật của bạn dựa trên nhiều thuộc tính bảo mật. Ví dụ, nếu một policy function dựa vào điều kiện WHERE với 4 thuộc tính ( như là employee number, cost center, position, spending limit), sau đó nhiều câu subquerie phải thực thi để truy xuất thông tin này. Thay vì, nếu dữ liệu này có sẵn trong ngũ cảnh ứng dụng, khi đó hiệu suất sẽ nhanh hơn.

Bạn có thể sử dụng một ngữ cảnh ứng dụng để trả về chính sách bảo mật đúng, bắt buộc thông qua điều kiện. Ví dụ, xem xét một đơn hàng truy cập vào ứng dụng phải theo các luật: khách hàng chỉ thấy đơn hàng của họ, và nhân viên thấy hết đơn hàng của mọi khách hàng. Có 2 chính sách khác biệt. Bạn có thể định nghĩa một ngữ cảnh ứng dụng với một thuộc tính vị trí, và thuộc tính này có thể được truy cập trong policy function để trả về điều kiện đúng, dựa trên giá trị của thuộc tính. Theo cách này, bạn có thể kích hoạt một người dùng trong vị trí nhân viên để truy xuất mọi hoá đơn, nhưng một người dùng trong vị trí khách hàng chỉ có thể thấy những bản ghi có liên quan đến người dùng đó thôi.

SELECT \* FROM orders\_tab

Kiểm soát truy cập chi tiết tự động chỉnh sửa câu truy vấn bao hàm điều kiện WHERE:

SELECT \* FROM orders\_tab

WHERE custno = SYS\_CONTEXT ('order\_entry', 'cust\_num');

Tiếp tục với ví dụ trước, giả sử bạn có 50,000 khách hàng, và bạn không muốn có một điều kiện khác trả về mỗi một khách hàng. Tất cả khách hàng có chung điều kiện WHERE, cái mà quy định họ chỉ có thể thấy đơn đặt hàng của họ. Chỉ đơn thuần số của khách hàng khác nhau mà thôi.

Sử dụng ngữ cảnh ứng dụng, bạn có thể trả về một điều kiện WHERE trong một policy function áp dụng cho 50,000 khách hàng. Kết quả là, một con trỏ sẽ thực thi theo từng khách hàng khác nhau, bởi vì số khách hàng được thay đổi trong thời gian thực thi. Giá trị này sẽ khác cho mỗi khách hàng. Tác dụng của ngữ cảnh ứng dụng trong trường hợp này cung cấp hiệu suất tốt nhất, và bảo mật mức row-level.

SYS\_CONTEXTfunction hoạt động như là một biến liên kết, chỉ những SYS\_CONTEXT mới là hằng số.

### Các thành phần của một chính sách CSDL riêng ảo Oracle

Để triển khai Oracle VPD, bạn phải tạo một phương thức để tạo mệnh đề WHERE tự động, và một chính sách để gắn phương thức này với đối tượng mà bạn muốn bảo vệ.

* + Tạo một hàm để tạo mệnh đề WHERE tự động
  + Tạo một chính sách để gắn phương thức với đối tượng bạn muốn bảo vệ

**See Also:**

* + - ["Which Privileges Are Used to Run Oracle Virtual Private Database Policy Functions?" on page 7-3](#_bookmark1357)
    - ["Tutorials: Creating Oracle Virtual Private Database Policies" on page 7-19](#_bookmark1465)

#### Tạo một phương thức để tạo mệnh đề WHERE tự động

Để tạo mệnh đề WHERE tự động (điều kiện), bạn phải tạo một phương thức (không phải là một thủ tục) để định nghĩa những hạn chế mà bạn muốn thực hiện. Thông thường, quản trị viên an ninh tạo phương thức này trên schema của họ. Đối với nhiều những hành động phức tạp, như là gọi những phương thức hay thêm những kiềm tra để theo dõi những nỗ lực đăng nhập thất bại, tạo phương thức này trong package.

Phương thức phải có các hành vi sau đây:

* + Nó phải có tham số là tên schema và tên một đối tượng (bảng, view, hay synonym) như thông số truyền vào. Định nghĩa những thông số truyền vào để giữ những thông tin, nhưng không chỉ định schema và tên đối tượng đó trong phương thức. Chính sách mà bạn tạo với DBMS\_RLSpackage ( miêu tả trong "Creating a Policy to Attach the Function to the Objects You Want to Protect" trong trang 7-5) cung cấp những cái tên của schema và đối tượng mà chính sách áp dụng. Bạn phải tạo tham số cho schema đầu tiên, sau đó là tham số cho đối tượng.
  + Nó phải cung cấp một giá trị trả về cho điều kiện của mệnh đề WHERE được tao ra. Giá trị trả về cho mệnh đề WHERE luôn luôn là dữ liệu kiểu VARCHAR2.
    - **Nó phải tạo ra một mệnh đề WHERE hợp lệ. Code nà có thể đơn giản như ví dụ trong "Tutorial: Creating a Simple Oracle Virtual Private Database Policy" trang 7-19, trong đó mệnh đề WHERE của mọi user đã đăng nhập là như nhau.**

Nhưng trong hầu hết trường hợp, bạn có lẽ muốn thiết kế mệnh đề WHERE khác nhau với mỗi user, mỗi nhóm người dùng, hay mỗi ứng dụng mà truy cập đối tượng bạn muốn bảo vệ. Ví dụ, nếu một quản lý đăng nhập, mệnh đề WHERE có thể chỉ định những quyền cụ thể của người quản lý đó. Bạn có thể làm điều này bằng cách kết hợp một ngữ cảnh ứng dụng, cái mà truy cập session thông tin người dùng, vào việc tạo câu lệnh WHERE. "Tutorial: Implementing a Policy with a Database Session-Based Application Context" on page 7-22 làm rõ cách tạo một chính sách Oracle VPD dùng trong ngữ cảnh ứng dụng.

Bạn có thể tạo phương thức OVPD không sử dụng một ngữ cảnh ứng dụng, nhưng một ngữ cảnh ứng dụng tạo một chính sách OVPD mạnh hơn, chắc chắn dựa vào truy cập của user để có các thuộc tính phiên của user đó, như là user ID. Chapter 6, "Using Application Contexts to Retrieve User Information" bàn về các loại ngữ cảnh ứng dụng một cách chi tiết.

Ngoài ra, bạn có thể nhúng C hoặc Java call để truy cập thông tin hệ thống hay để trả về mệnh đề WHERE từ file hệ thống hoặc nguồn khác

* + - Nó không được select từ một bản trong hàm ràng buộc liên quan.

Mặc dù bạn có thể định nghĩa một chính sách tránh một bảng, bạn không thể select bảng đó từ trong ràng buộc đã được định nghĩa tránh bảng đó.

**Note: Nếu bạn dự định chạy phương thức trong nhiều phiên bản khác nhau, bạn có thể kiểm soát kết quả của phương thức: liệu kết quả là thống nhất trên tất cả các phiên bản, hay cụ thể cho phiên bản mà trong đó phương thức được chạy. Đọc "How Editions Affects the Results of a Global Application Context PL/SQL Package" on page 6-24 để biết thêm thông tin.**

#### Tạo một ràng buộc để gắn phương thức với đối tượng mà bạn muốn bảo vệ

Sau khi bạn tạo phương thức, bạn cần tạo một ràng buộc Oracle VPD liên kết phương thức với table, view, hay synonym. Bạn tạo ràng buộc bằng cách sử dụng DBMS\_RLSpackage. Package này bao gồm những thủ tục cho phép bạn quản lý ràng buộc và thiết lập kiểm soát truy cập chi tiết. Ví dụ, để gắn ràng buộc với bảng, bạn sử dụng thủ tục DBMS\_RLS.ADD\_POLICY. Trong cài đặt này, bạn kiểm soát truy cập chi tiết, đặc biệt là ràng buộc để có ảnh hưởng khi một user cung cấp một câu SELECT hay UPDATE trên bảng hoặc view.

Sự phối hợp của việc tạo phương thức và sau đó áp dụng nó vào bảng hoặc view được khuyến khích hơn là tạo ràng buộc Oralce VPD.

["Tutorials: Creating Oracle Virtual Private Database Policies" on page 7-19](#_bookmark1465) provides examples of how to create Virtual Private Database policies. See ["Configuring an](#_bookmark1377) [Oracle Virtual Private Database Policy" on page 7-5](#_bookmark1377) for detailed information.

### Cấu hình một ràng buộc Oracle VPD

Phần này bao gồm:

* + - Về những ràng buộc của Oracle VPD
    - Gắn một ràng buộc vào table, view, hay synonym
  + Thực hiện ràng buộc trên những loại truy vấn cụ thể
  + Kiểm soát trình bày những cột dữ liệu với ràng buộc
  + Làm việc với những nhóm ràng buộc Oracle VPD
  + Tối ưu hiệu năng bằng cách sử dụng các loại ràng buộc Oracle VPD

#### 

#### Về những ràng buộc của Oracle VPD

Sau khi bạn tạo một phương thức để định nghĩa những hành động của mệnh đề WHERE OVPD, bạn cần liên kế phương thức này với bảng cái mà hành động VPD áp dụng. Bạn có thể làm điều này bằng cách cấu hình một ràng buộc VPD. Ràng buộc bản thân nó là một cơ chế để quản lý hàm VPD. Ràng buộc cũng cho phép bạn thêm kiểm soát truy cập chi tiết, đặc biệt là chỉ định những loại của câu lệnh SQL hay những cột trong bảng cụ thể mà ràng buộc ảnh hưởng. Khi một user cố gắng truy cập dư liệu trong CSDL này, ràng buộc sẽ có hiệu lực một cách tự động.

Phần này miêu tả những cách sử dụng thông thường của việc gắn những ràng buộc vào bảng, view, và synonym. Để quản lý một ràng buộc Oracle VPD, bạn sử dụng DBMS\_RLS package, cái mà miêu tả chi tiết trong Oracle Database PL/SQL Packages and Types Reference.

[Table 7–1](#_bookmark1384) lists the procedures in the DBMS\_RLS package.

***Table 7–1 DBMS\_RLS Procedures***

|  |  |
| --- | --- |
| **Procedure** | **Description** |
| **For Handling Individual Policies** |  |
| DBMS\_RLS.ADD\_POLICY | Thêm một ràng buộc vào table, view, synonym |
| DBMS\_RLS.ENABLE\_POLICY | Bật (hoặc vô hiệu) một chính sách mà bạn đã thêm vào bảng, view, or synonym |
| DBMS\_RLS.REFRESH\_POLICY | Vô hiệu hóa con trỏ được liên kết với các chính sách non static. |
| DBMS\_RLS.DROP\_POLICY | Bỏ một chính sách từ bảng, view, synonym |
| **For Handling Grouped Policies** |  |
| DBMS\_RLS.CREATE\_POLICY\_GROUP | Tạo một nhóm chính sách |
| DBMS\_RLS.DELETE\_POLICY\_GROUP | Xoá một nhóm chính sách |
| DBMS\_RLS.ADD\_GROUPED\_POLICY | Thêm một chính sách vào nhóm được chỉ định |
| DBMS\_RLS.ENABLE\_GROUPED\_POLICY | Kích hoạt một chính sách trong một nhóm |
| DBMS\_RLS.REFRESH\_GROUPED\_POLICY | Phân tích cú pháp một lần nữa các câu lệnh SQL được liên kết với chính sách được làm mới |
| DBMS\_RLS.DISABLE\_GROUPED\_POLICY | Vô hiệu một chính sách trong một group |
| DBMS\_RLS.DROP\_GROUPED\_POLICY | Xoá một chính sách trong một group chỉ định |
| **For Handling Application Contexts** |  |
| DBMS\_RLS.ADD\_POLICY\_CONTEXT | Thêm ngữ cảnh cho ứng dụng đang hoạt động |
| DBMS\_RLS.DROP\_POLICY\_CONTEXT | Xoá ngữ cảnh của ứng dụng |

**See Also:**

* + - ["Components of an Oracle Virtual Private Database Policy" on page 7-4](#_bookmark1370) for a description of the type of function that you need to create to control user access to a database table, view, or synonym
    - [Chapter 6, "Using Application Contexts](#_bookmark1122)

[to Retrieve User Information"](#_bookmark1122) if you plan to use application contexts in the Oracle Virtual Private Database policy (which in most cases, you would)

* + - ["Tutorials: Creating Oracle Virtual Private Database Policies" on page 7-19](#_bookmark1465) for examples of using application contexts in sample Oracle Virtual Private Database functions

#### 

#### Gắn một ràng buộc tới một table, view, hay synonym

Để gắn một ràng buộc tới table, view, hay synonym, bạn sử dụng thủ tục DBMS\_RLS.ADD\_POLICY. bạn cần xác định bảng, view, hay synonym mà cần thêm ràng buộc, và tên của ràng buộc đó. Bạn cũng có thể xác định thông tin khác, như là loại của câu truy vấn (SELECT, UPDATE, DELETE, CREATE INDEX, hay ALTERINDEX).

[Example 7–1](#_bookmark1400) cho thấy cách sử dụng DBMS\_RLS.ADD\_POLICY để đính kèm chính sách Oracle VPD có tên là secure\_update vào bảng HR.EMPLOYEES. Hàm được đính kèm với chính sách là check\_updates.

***Example 7–1 Attaching a Simple Oracle Virtual Private Database Policy to a Table***

BEGIN DBMS\_RLS.ADD\_POLICY(

object\_schema => 'hr', object\_name => 'employees', policy\_name => 'secure\_update', policy\_function => 'check\_updates',

...

Nếu hàm được tạo bên trong một gói, hãy bao gồm tên gói. Ví dụ:

policy\_function => 'pkg.check\_updates',

...

**Note: Mặc dù bạn có thể định nghĩa một chính sách đối với một bảng, bạn không thể chọn bảng đó từ bên trong chính sách đã được định nghĩa với bảng.**

#### Thực thi các chính sách về các loại câu lệnh SQL cụ thể

Bạn có thể thực thi các chính sách Oracle Virtual Private Database cho các câu lệnh SELECT, INSERT, UPDATE, INDEX và DELETE. Nếu bạn không chỉ định một loại câu lệnh, theo mặc định, Cơ sở dữ liệu Oracle chỉ định SELECT, INSERT, UPDATE và DELETE, chứ không phải INDEX. Nhập bất kỳ kết hợp nào của các loại câu lệnh này bằng cách sử dụng tham số statement\_types trong thủ tục DBMS\_RLS.ADD\_POLICY. Bao danh sách trong một cặp dấu nháy đơn.

[Example 7–2](#_bookmark1404) shows an how to use the statement\_types parameter to specify the

SELECT and INDEX statements for a policy.

***Example 7–2 Specifying SQL Statement Types with DBMS\_RLS.ADD\_POLICY***

BEGIN DBMS\_RLS.ADD\_POLICY(

object\_schema => 'hr', object\_name => 'employees', policy\_name => 'secure\_update', policy\_function => 'check\_updates', **statement\_types => 'SELECT,INDEX'**);

END;

/

Khi bạn chỉ định tham số statement\_types, hãy lưu ý đến chức năng sau:

* + Code ứng dụng bị ảnh hưởng bởi chính sách VPD có thể bao gồm câu lệnh MERGE INTO. Tuy nhiên, trong chính sách PVD, bạn phải đảm bảo rằng tham số statement\_types bao gồm tất cả ba câu lệnh INSERT, UPDATE và DELETE cho chính sách để thành công. Ngoài ra, bạn có thể bỏ qua tham số statement\_types. (This functionality is available with Oracle Database 11*g* Release 2 (11.2.0.2).)
  + Lưu ý rằng người dùng có đặc quyền để duy trì chỉ mục có thể xem tất cả dữ liệu hàng, ngay cả khi người dùng không có quyền truy cập đầy đủ trong một truy vấn thông thường như SELECT. Ví dụ: người dùng có thể tạo chỉ mục dựa trên chức năng chứa hàm do người dùng xác định với các giá trị cột làm đối số của nó. Trong quá trình tạo chỉ mục, Cơ sở dữ liệu Oracle chuyển các giá trị cột của mỗi hàng vào hàm người dùng, làm cho dữ liệu hàng có sẵn cho người dùng tạo chỉ mục. Bạn có thể thực thi các chính sách Oracle Virtual Private Database trên các hoạt động duy trì chỉ mục bằng cách chỉ định INDEX với tham số statement\_types.

#### Kiểm soát hiển thị dữ liệu của cột với chính sách

Bạn có thể tạo các chính sách thực thi bảo mật mức hàng khi một cột liên quan đến bảo mật được tham chiếu trong một truy vấn.

Thêm chính sách cho cơ sở dữ liệu riêng ảo ở mức cột

* + Chỉ hiển thị các hàng cột có liên quan đến truy vấn
  + Sử dụng Column Masking để hiển thị các cột nhạy cảm các giá trị NULL

###### Thêm chính sách cho cơ sở dữ liệu riêng ảo ở mức cột

Chính sách cấp cột thực thi bảo mật cấp hàng khi truy vấn tham chiếu đến cột có liên quan đến bảo mật. Bạn có thể áp dụng chính sách Oracle VPD cho các bảng và view, nhưng không áp dụng cho các synonym.

Để áp dụng chính sách cho một cột, hãy chỉ định cột có liên quan đến bảo mật bằng cách sử dụng tham số SEC\_RELEVANT\_COLS của thủ tục DBMS\_RLS.ADD\_POLICY. Tham số này áp dụng chính sách bảo mật bất cứ khi nào cột được tham chiếu, rõ ràng hoặc ngầm, trong truy vấn.

Ví dụ: người dùng không thuộc bộ phận Nhân sự thường chỉ được phép xem số An sinh Xã hội của họ. Nhân viên bán hàng khởi tạo truy vấn sau:

SELECT fname, lname, ssn FROM emp;

Hàm thực thi chính sách bảo mật trả về điều kiện ssn = 'my\_ssn'. CSDL Oracle ghi lại truy vấn và thực hiện các thao tác sau:

SELECT fname, lname, ssn FROM emp WHERE ssn = 'my\_ssn';

[Example 7–3](#_bookmark1411)  hiển thị chính sách Oracle VPD trong đó người dùng bộ phận bán hàng không thể thấy mức lương của những người bên ngoài bộ phận (bộ phận số 30) của người dùng bộ phận bán hàng. Các cột có liên quan cho chính sách này là sal và comm. Đầu tiên, hàm chính sách Oracle VPD được tạo ra, và sau đó nó được thêm vào bằng cách sử dụng gói PLMS SQL DBMS\_RLS.

***Example 7–3 Creating a Column-Level Oracle Virtual Private Database Policy***

CREATE OR REPLACE FUNCTION hide\_sal\_comm ( v\_schema IN VARCHAR2,

v\_objname IN VARCHAR2)

RETURN VARCHAR2 AS con VARCHAR2 (200);

BEGIN

con := 'deptno=30'; RETURN (con);

END hide\_sal\_comm;

Sau đó bạn cấu hình chính sách với thủ tục DBMS\_RLS.ADD\_POLICY như sau:

BEGIN DBMS\_RLS.ADD\_POLICY (

object\_schema => 'scott', object\_name => 'emp',

policy\_name => 'hide\_sal\_policy', policy\_function => 'hide\_sal\_comm', **sec\_relevant\_cols => 'sal,comm');**

END;

###### Chỉ hiển thị các hàng cột có liên quan đến truy vấn

Hành vi mặc định cho Oracle VPD là hạn chế số lượng hàng được trả về cho một truy vấn tham chiếu đến các cột chứa thông tin nhạy cảm. Bạn chỉ định các cột có liên quan đến bảo mật này bằng cách sử dụng tham số SEC\_RELEVANT\_ COLUMNS của thủ tục DBMS\_RLS.ADD\_POLICY, như trong Ví dụ 7–3 trên trang 7-9.

Ví dụ, hãy xem xét người dùng bộ phận bán hàng với đặc quyền SELECT trên bảng emp, được bảo vệ bằng chính sách Orcale VPD được tạo trong ví dụ 7–3. Người dùng (ví dụ, người dùng SCOTT) chạy truy vấn sau:

SELECT ENAME, d.dname, JOB, SAL, COMM

FROM emp e, dept d

WHERE d.deptno = e.deptno;

The database returns the following rows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENAME | DNAME | JOB | SAL | COMM |
| ---------- -------------- --------- ---------- ---------- | | | | |
| ALLEN | SALES | SALESMAN | 1600 | 300 |
| WARD | SALES | SALESMAN | 1250 | 500 |
| MARTIN | SALES | SALESMAN | 1250 | 1400 |
| BLAKE | SALES | MANAGER | 2850 |  |
| TURNER | SALES | SALESMAN | 1500 | 0 |
| JAMES | SALES | CLERK | 950 |  |

6 rows selected.

Các hàng được hiển thị là những hàng mà người dùng có đặc quyền truy cập tất cả các cột trong hàng.

###### Sử dụng Column Masking để hiển thị các cột nhạy cảm các giá trị NULL

Nếu một truy vấn tham chiếu đến một cột nhạy cảm, thì hành động mặc định của Oracle VPD sẽ hạn chế số hàng được trả về. Với hành column-masking, tất cả các hàng đều hiển thị, ngay cả những hàng tham chiếu cột nhạy cảm. Tuy nhiên, các cột nhạy cảm hiển thị dưới dạng giá trị NULL. Để bật column-masking, hãy đặt tham số SEC\_ RELEVANT\_COLS\_opt của thủ tục DBMS\_RLS.ADD\_POLICY.

Ví dụ, xem xét kết quả của truy vấn nhân viên bán hàng, được mô tả trong ví dụ trước. Nếu column-masking được sử dụng, sau đó thay vì chỉ thấy hàng chứa các chi tiết và số an sinh xã hội của nhân viên bán hàng, nhân viên bán hàng sẽ thấy tất cả các hàng từ bảng emp, nhưng các giá trị cột ssn sẽ được trả về là NULL. Lưu ý rằng hành vi này về cơ bản là khác với tất cả các loại khác của chính sách Oracle VPD, chỉ trả lại một tập con các hàng.

Ngược lại với hành động mặc định của Oracle VPD, column-masking hiển thị tất cả các hàng, nhưng trả về các giá trị cột nhạy cảm giá trị NULL. Để bao gồm column-masking trong chính sách của bạn, hãy đặt tham số SEC\_RELEVANT\_COLS\_OPT của thủ tục DBMS\_RLS.ADD\_POLICY thành DBMS\_RLS.ALL\_ROWS.

[Example 7–4](#_bookmark1417) shows column-level Oracle Virtual Private Database column-masking. It uses the same VPD policy as [Example 7–3 on page 7-9](#_bookmark1411), but with sec\_relevant\_cols\_ opt specified as DBMS\_RLS.ALL\_ROWS.

***Example 7–4 Adding a Column Masking to an Oracle Virtual Private Database Policy***

BEGIN DBMS\_RLS.ADD\_POLICY(

object\_schema => 'scott',

object\_name => 'emp',

policy\_name => 'hide\_sal\_policy', policy\_function => 'hide\_sal\_comm', sec\_relevant\_cols =>' sal,comm', sec\_relevant\_cols\_opt => dbms\_rls.ALL\_ROWS);

END;

Giả sử rằng người dùng bộ phận bán hàng có đặc quyền SELECT trên bảng emp (chẳng hạn như người dùng SCOTT) chạy truy vấn sau:

SELECT ENAME, d.dname, job, sal, comm FROM emp e, dept d

WHERE d.deptno = e.deptno;

Cơ sở dữ liệu trả về tất cả các hàng được chỉ định trong truy vấn, nhưng với một số giá trị được che dấu vì chính sách Oracle VPD:

ENAME DNAME JOB SAL COMM

---------- -------------- --------- ---------- ---------- CLARK ACCOUNTING MANAGER

KING ACCOUNTING PRESIDENT MILLER ACCOUNTING CLERK JONES RESEARCH MANAGER

FORD RESEARCH ANALYST

ADAMS RESEARCH CLERK

SMITH RESEARCH CLERK

SCOTT RESEARCH ANALYST

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| WARD | SALES | SALESMAN | 1250 | 500 |
| TURNER | SALES | SALESMAN | 1500 | 0 |
| ALLEN | SALES | SALESMAN | 1600 | 300 |
| JAMES | SALES | CLERK | 950 |  |
| BLAKE | SALES | MANAGER | 2850 |  |
| MARTIN | SALES | SALESMAN | 1250 | 1400 |

14 rows selected.

Column-masking trả về tất cả các hàng được yêu cầu bởi truy vấn user bán hàng, nhưng đã thực hiện sal và comm cột NULL cho nhân viên bên ngoài bộ phận bán hàng.

Các cân nhắc sau đây áp dụng cho column-masking:

* Column-masking chỉ áp dụng cho SELECT.
* Các điều kiện column-masking được tạo bởi hàm chính sách phải là các biểu thức Boolean đơn giản, không giống như các điều kiện Oracle VPD thông thường.
* Đối với các ứng dụng thực hiện các phép tính, hoặc không mong đợi các giá trị NULL, hãy sử dụng Oracle VPD mức tiêu chuẩn, chỉ định SEC\_RELEVANT\_ COLS thay vì tùy chọn column-masking SEC\_RELEVANT\_COLS\_OPT.
* Không bao gồm các cột của kiểu dữ liệu đối tượng (bao gồm cả kiểu XML) trong cài đặt sec\_ relev\_cols. Loại cột này không được hỗ trợ cho cài đặt sec\_relevant\_ cols.
* Column-masking được sử dụng với UPDATE AS SELECT chỉ cập nhật các cột mà người dùng được phép xem.
* Đối với một số truy vấn, column-masking có thể ngăn một số hàng hiển thị. Ví dụ:

SELECT \* FROM emp WHERE sal = 10;

Vì tùy chọn column-masking đã được đặt, truy vấn này có thể không trả lại hàng nếu cột lương trả về giá trị NULL.

#### Làm việc với những nhóm chính sách Oracle VPD

Phần này bao gồm:

* [Nói](#_bookmark1420) về Oracle VPD
* [Tạo](#_bookmark1425) mới một nhóm Oracle VPD
* Chỉ định nhóm chính sách mặc định với nhóm chính sách SYS\_DEFAULT
* Thiết lập nhiều chính sách cho mỗi bảng, view hoặc synonym
* Xác thực ứng dụng được sử dụng để kết nối với cơ sở dữ liệu

**See Also:** ["Tutorial: Implementing an Oracle Virtual Private](#_bookmark1488) [Database Policy Group" on page 7-28](#_bookmark1488)

###### Nói về Oracle VPD

Bạn có thể nhóm nhiều chính sách bảo mật với nhau và áp dụng chúng cho một ứng dụng. Nhóm chính sách là tập hợp các chính sách bảo mật thuộc về một ứng dụng. Bạn có thể chỉ định ngữ cảnh ứng dụng để cho biết nhóm chính sách có hiệu lực. Sau đó, khi người dùng truy cập vào bảng, view hoặc cột synonum, Cơ sở dữ liệu Oracle tra cứu ngữ cãnh driving để xác định nhóm chính sách có hiệu lực. Nó thực thi tất cả các chính sách liên quan thuộc về nhóm chính sách.

Nhóm chính sách hữu ích cho các tình huống trong đó nhiều ứng dụng có nhiều chính sách bảo mật chia sẻ cùng một bảng, view hoặc synonym. Điều này cho phép bạn xác định các chính sách đó sẽ có hiệu lực khi bảng, view hoặc synonym được truy cập.

Ví dụ, trong môi trường lưu trữ, Công ty A có thể lưu trữ bảng BENEFIT cho Công ty B và Công ty C. Bảng này được truy cập bởi hai ứng dụng khác nhau, Nhân sự và Tài chính, với hai chính sách bảo mật khác nhau. Ứng dụng Nhân sự cho phép người dùng dựa trên xếp hạng trong công ty và ứng dụng Tài chính cho phép người dùng dựa trên bộ phận. Việc lồng ghép hai chính sách này vào bảng BENEFIT đòi hỏi sự phát triển chung của các chính sách giữa hai công ty, đó không phải là một lựa chọn khả thi. Bằng cách định nghĩa một ngữ cảnh ứng dụng để thúc đẩy việc thực thi một tập hợp các chính sách cụ thể cho các đối tượng cơ sở, mỗi ứng dụng có thể thực hiện một bộ chính sách bảo mật riêng tư.

Để làm điều này, bạn tổ chức các chính sách bảo mật thành các nhóm. Bằng cách tham khảo ngữ cảnh ứng dụng, Cơ sở dữ liệu Oracle xác định nhóm chính sách nào sẽ có hiệu lực vào thời gian chạy. Máy chủ thực thi tất cả các chính sách thuộc về nhóm chính sách đó.

###### Tạo mới một nhóm Oracle VPD

Để thêm chính sách vào bảng, view hay synonym, hãy sử dụng thủ tục DBMS\_RLS.ADD\_GROUPED\_POLICY để chỉ định nhóm mà chính sách thuộc về. Để chỉ định chính sách nào sẽ có hiệu lực, bạn có thể thêm ngữ cảnh driving bằng cách sử dụng thủ tục DBMS\_RLS.ADD\_POLICY\_ CONTEXT. Nếu ngữ cảnh driving trả về một nhóm chính sách không xác định, thì một lỗi được trả về.

Nếu driving context không được xác định, thì cơ sở dữ liệu Oracle sẽ chạy tất cả các chính sách. Tương tự như vậy, nếu driving context là NULL, thì các chính sách từ tất cả các nhóm chính sách được thực thi. Một ứng dụng truy cập dữ liệu không thể bỏ qua mô-đun thiết lập bảo mật (thiết lập ngữ cảnh ứng dụng) để tránh bất kỳ chính sách áp dụng nào.

Bạn có thể áp dụng nhiều driving context cho cùng một bảng, view hay synonym và mỗi ngữ cảnh sẽ được xử lý riêng lẻ. Điều này cho phép bạn định cấu hình nhiều bộ chính sách đang hoạt động được thực thi.

Hãy xem xét, ví dụ, một công ty lưu trữ lưu trữ các ứng dụng Lợi ích và Tài chính, trong đó chia sẻ một số đối tượng cơ sở dữ liệu. Cả hai ứng dụng đều có sọc để lưu trữ bằng cách sử dụng chính sách SUBSCRIBER trong nhóm chính sách SYS\_DEFAULT. Truy cập dữ liệu được phân đoạn đầu tiên theo ID người đăng ký, sau đó cho dù người dùng đang truy cập vào các lợi ích hoặc ứng dụng Tài chính (được xác định bởi driving context). Giả sử rằng Công ty A, sử dụng dịch vụ lưu trữ, muốn áp dụng chính sách tùy chỉnh chỉ liên quan đến quyền truy cập dữ liệu của chính nó. Bạn có thể thêm driving context bổ sung (chẳng hạn như CÔNG TY ĐẶC BIỆT) để đảm bảo rằng nhóm chính sách bổ sung, đặc biệt chỉ được áp dụng để truy cập dữ liệu cho Công ty A. Bạn sẽ không áp dụng điều này theo chính sách SUBSCRIBER, vì chính sách chỉ liên quan đến Công ty A và sẽ hiệu quả hơn để tách riêng chính sách lưu trữ cơ bản khỏi các chính sách khác.

###### Chỉ định nhóm chính sách mặc định với nhóm chính sách SYS\_DEFAULT

Trong một nhóm các chính sách bảo mật, bạn có thể chỉ định một chính sách bảo mật làm chính sách bảo mật mặc định. Điều này rất hữu ích trong các tình huống mà bạn phân vùng các chính sách bảo mật theo ứng dụng, do đó chúng sẽ luôn có hiệu lực. Các chính sách bảo mật mặc định cho phép các nhà phát triển thực thi bảo mật cơ bản trong mọi điều kiện, trong khi phân chia các chính sách bảo mật theo ứng dụng (sử dụng nhóm bảo mật) cho phép phân lớp bảo mật bổ sung, ứng dụng cụ thể trên các chính sách bảo mật mặc định. Để thực thi các chính sách bảo mật mặc định, bạn thêm chính sách vào nhóm chính sách SYS\_DEFAULT.

Các chính sách được xác định trong nhóm này cho một bảng, view hoặc synonym cụ thể được chạy cùng với nhóm chính sách được chỉ định bởi diving context. Như được mô tả trước đó,driving context là ngữ cảnh ứng dụng cho biết nhóm chính sách có hiệu lực. Nhóm chính sách SYS\_ DEFAULT có thể hoặc không thể chứa các chính sách. Bạn không thể bỏ nhóm chính sách SYS\_ DEFAULT. Nếu bạn làm như vậy, thì cơ sở dữ liệu Oracle sẽ hiển thị lỗi.

Nếu, với nhóm chính sách SYS\_DEFAULT, bạn thêm các chính sách liên kết với hai hoặc nhiều đối tượng, sau đó mỗi đối tượng sẽ có một nhóm chính sách SYS\_DEFAULT riêng biệt được liên kết với nó. Ví dụ, bảng emp trong lược đồ scott có một nhóm chính sách SYS\_DEFAULT và bảng dept trong lược đồ scott có một nhóm chính sách SYS\_DEFAULT khác được liên kết với nó. Hãy nghĩ về chúng như được tổ chức trong cấu trúc cây như sau:

SYS\_DEFAULT

* policy1 (scott/emp)
* policy3 (scott/emp) SYS\_DEFAULT
* policy2 (scott/dept)

Bạn có thể tạo các nhóm chính sách có tên giống nhau. Khi bạn chọn một nhóm chính sách cụ thể, lược đồ và tên đối tượng được liên kết của nó được hiển thị trong bảng thuộc tính ở bên phải của màn hình.

###### Thiết lập nhiều chính sách cho mỗi bảng, view hoặc synonym

Bạn có thể thiết lập một số chính sách cho cùng một bảng, view hoặc synonym. Giả sử, ví dụ, bạn có một ứng dụng cơ sở cho Entry Order, và mỗi bộ phận của công ty bạn có các quy tắc riêng để truy cập dữ liệu. Bạn có thể thêm một hàm chính sách phân chia cụ thể vào một bảng mà không cần phải viết lại hàm chính sách của ứng dụng cơ sở.

Tất cả các chính sách được áp dụng cho một bảng được thực thi bằng cú pháp AND. Nếu bạn có ba chính sách được áp dụng cho bảng CUSTOMERS, thì mỗi chính sách sẽ được áp dụng cho bảng. Bạn có thể sử dụng các nhóm chính sách và ngữ cảnh ứng dụng để phân vùng thực thi kiểm soát truy cập chi tiết để các chính sách khác nhau áp dụng, tùy thuộc vào ứng dụng nào đang truy cập dữ liệu. Điều này loại bỏ yêu cầu cho các nhóm phát triển để cộng tác trên các chính sách và đơn giản hóa việc phát triển ứng dụng. Bạn cũng có thể có nhóm chính sách mặc định luôn áp dụng (ví dụ: để thực thi dữ liệu được phân tách bởi người đăng ký trong môi trường lưu trữ).

###### Xác thực ứng dụng được sử dụng để kết nối với cơ sở dữ liệu

Gói triển driving context phải xác nhận hợp lệ ứng dụng đang được sử dụng để kết nối với cơ sở dữ liệu. Mặc dù Oracle Database kiểm tra hàng chờ để đảm bảo rằng gói triển khai driving context đặt thuộc tính ngữ cảnh, xác thực không đầy đủ vẫn có thể xảy ra trong gói.

Ví dụ, trong các ứng dụng mà người dùng cơ sở dữ liệu hoặc người dùng doanh nghiệp được biết đến cơ sở dữ liệu, người dùng cần đặc quyền EXECUTE trên gói để thiết lập driving context. Cân nhắc người dùng biết điều đó:

* Ứng dụng BENEFITS cho phép truy cập tự do hơn ứng dụng HR.
* Thủ tục setctx (đặt nhóm chính sách chính xác driving context) không thực hiện bất kỳ xác thực nào để xác định ứng dụng nào đang thực sự kết nối. Nghĩa là, thủ tục không kiểm tra địa chỉ IP của kết nối đến (đối với hệ thống ba tầng) hoặc thuộc tính proxy\_user của phiên người dùng.

Người dùng này có thể chuyển sang gói driving context một đối số thiết lập ngữ cảnh cho nhóm chính sáchBENEFITS tự do hơn, và thay vì truy cập vào ứng dụng HR. Do setctx không xác nhận thêm ứng dụng, người dùng này bỏ qua chính sách bảo mật HR hạn chế hơn.

Ngược lại, nếu bạn thực hiện xác thực proxy với Oracle Virtual Private Database, thì bạn có thể xác định danh tính của tầng giữa (và ứng dụng) đang kết nối với cơ sở dữ liệu thay mặt cho người dùng. Chính sách đúng sẽ được áp dụng cho mỗi ứng dụng để dàn xếp quyền truy cập dữ liệu.

Ví dụ, một nhà phát triển sử dụng tính năng xác thực proxy có thể xác định rằng ứng dụng (tầng giữa) kết nối với cơ sở dữ liệu là HRAPPSERVER. Do đó, gói thực hiện driving context có thể xác minh xem proxy\_user trong phiên người dùng có phải là HRAPPSERVER hay không. Nếu vậy, sau đó nó có thể thiết lập driving context để sử dụng nhóm chính sách nhân sự. Nếu proxy\_user không phải là HRAPPSERVER, thì nó có thể từ chối truy cập.

Trong trường hợp này, câu query sau đây sẽ thực thi:

SELECT \* FROM apps.benefit;

CSDLOracle chọn các chính sách từ nhóm chính sách mặc định (SYS\_DEFAULT) và kích hoạt namespace HR. Truy vấn được viết lại nội bộ như sau:

SELECT \* FROM apps.benefit

WHERE company = SYS\_CONTEXT('ID','MY\_COMPANY') and SYS\_CONTEXT('ID','TITLE') = 'MANAGER';

#### Tối ưu hóa hiệu suất bằng cách sử dụng các loại chính sách Oracle VPD

Phần này bao gồm:

* + Giới thiệu về các loại chính sách Oracle VPD
  + Sử dụng Kiểu chính sách động để tự động chạy lại các hàm chính sách
  + Sử dụng một chính sách tĩnh để ngăn chặn các hàm chính sách chạy lại mỗi câu query
  + Sử dụng một chính sách chung tĩnh để chia sẻ một chính sách với nhiều đối tượng
  + Khi nào sử dụng chế độ tĩnh và chia sẻ tĩnh các chính sách
  + Sử dụng chính sách ngữ cảnh nhạy cảm cho các điều kiện để không thay đổi sau khi phân tích cú pháp
  + Sử dụng chính sách có chung ngữ cảnh nhạy cảm để chia sẻ một chính sách với nhiều đối tượng
  + Thời điểm sử dụng Context-Sensitive và Shared Context-Sensitive Policies
  + Tổng kết về năm loại chính sách Oracle VPD

###### Giới thiệu về các loại chính sách Oracle VPD

Bạn có thể tối ưu hóa hiệu suất mỗi khi chính sách chạy bằng cách chỉ định loại chính sách cho chính sách của bạn. Các loại chính sách kiểm soát cách Oracle Database lưu trữ các vị từ chính sách cơ sở dữ liệu Oracle Virtual Private Database. Xem xét việc thiết lập một loại chính sách cho các chính sách của bạn, bởi vì việc thực hiện các hàm chính sách có thể sử dụng một lượng lớn tài nguyên hệ thống.Giảm thiểu số lần hàm chính sách có thể chạy tối ưu hóa hiệu suất csdl.

Bạn có thể chọn trong số năm loại chính sách: DYNAMIC, STATIC, SHARED\_STATIC, CONTEXT\_ SENSITIVE và SHARED\_CONTEXT\_SENSITIVE. Điều này cho phép bạn chỉ định chính xác tần suất một điều kiện chính sách sẽ thay đổi. Để chỉ định loại chính sách, hãy đặt tham số policy\_type của thủ tục DBMS\_RLS.ADD.

###### Sử dụng Kiểu chính sách động để tự động chạy lại các hàm chính sách

Kiểu chính sách DYNAMIC chạy hàm policy mỗi khi người dùng truy cập các đối tượng csdl được bảo vệ bởi VPD. Nếu bạn không chỉ định một loại chính sách trong thủ tục DBMS\_RLS.ADD\_POLICY, thì theo mặc định, chính sách của bạn sẽ là động. Bạn có thể định cấu hình cụ thể chính sách để động bằng cách đặt tham số policy\_type của thủ tục DBMS\_RLS.ADD\_POLICY thành DYNAMIC.

Loại chính sách này không tối ưu hóa hiệu suất csdl như các loại chính sách tĩnh và ngữ cảnh nhạy cảm. Tuy nhiên, Oracle khuyến cáo rằng trước khi bạn thiết lập các chính sách như là tĩnh hoặc ngữ cảnh nhạy cảm, trước tiên bạn nên kiểm tra chúng như là các loại chính sách DYNAMIC, lần nào cũng chạy. Kiểm tra các hàm policy như các chính sách DYNAMIC đầu tiên cho phép bạn quan sát cách hàm policy ảnh hưởng đến mỗi truy vấn, bởi vì không có gì được lưu trữ. Điều này đảm bảo rằng các phương thức hoạt động bình thường trước khi bạn kích hoạt chúng như là tĩnh hoặc các loại ngữ cảnh nhạy cảm để tối ưu hóa hiệu suất.

Bạn có thể sử dụng hàm DBMS\_UTILITY.GET\_TIME để đo thời gian bắt đầu và kết thúc cho một câu lệnh khi thực hiện. Ví dụ:

**-- 1. Thời gian bắt đầu:**

SELECT DBMS\_UTILITY.GET\_TIME FROM DUAL;

GET\_TIME

----------

2312721

**-- 2. Chạy câu lệnh:**

SELECT COUNT(\*) FROM HR.EMPLOYEES;

COUNT(\*)

----------

107

**-- 3. Thời gian kết thúc:**

SELECT DBMS\_UTILITY.GET\_TIME FROM DUAL;

GET\_TIME

----------

2314319

[Example 7–5](#_bookmark1442) shows how to create the DYNAMIC policy type.

***Example 7–5 Creating a DYNAMIC Policy with DBMS\_RLS.ADD\_POLICY***

BEGIN DBMS\_RLS.ADD\_POLICY(

object\_schema => 'hr', object\_name => 'employees', policy\_name => 'secure\_update', policy\_function => 'hide\_fin',

policy\_type => dbms\_rls.DYNAMIC); END;

/

**See Also:** ["About Auditing Functions, Procedures, Packages, and](#_bookmark1850) [Triggers" on page 9-32](#_bookmark1850) for information about how Oracle Database audits the underlying policy function for dynamic policies

###### Sử dụng một chính sách tĩnh để ngăn chặn các hàm chính sách chạy lại mỗi câu query

Kiểu chính sách tĩnh thực thi cùng một điều kiện cho tất cả người dùng. CSDL Oracle lưu trữ các điều kiện chính sách tĩnh trong SGA, vì vậy các hàm chính sách không chạy lại cho mỗi truy vấn. Điều này dẫn đến hiệu suất nhanh hơn.

Bạn có thể bật các chính sách tĩnh bằng cách đặt tham số policy\_type của thủ tục DBMS\_ RLS.ADD\_POLICY là STATIC hoặc SHARED\_STATIC, tùy thuộc vào việc bạn có muốn chính sách được chia sẻ trên nhiều đối tượng hay không.

[Example 7–6](#_bookmark1445) shows how to create the STATIC policy type.

***Example 7–6 Creating a STATIC Policy with DBMS\_RLS.ADD\_POLICY***

BEGIN DBMS\_RLS.ADD\_POLICY(

object\_schema => 'hr', object\_name => 'employees', policy\_name => 'secure\_update', policy\_function => 'hide\_fin',

policy\_type => dbms\_rls.STATIC); END;

/

Mỗi lần thực hiện cùng một con trỏ có thể tạo ra một tập hợp hàng khác nhau cho cùng một điều kiện, vì điều kiện có thể lọc dữ liệu khác nhau dựa trên các thuộc tính như SYS\_CONTEXT hoặc SYSDATE.

Ví dụ: giả sử bạn bật chính sách loại STATIC hoặc SHARED\_STATIC, nó sẽ thêm điều kiện sau vào tất cả các truy vấn dựa trên chính sách bảo vệ đối tượng csdl:

WHERE dept = SYS\_CONTEXT ('hr\_app','deptno')

Mặc dù điều kiện không thay đổi cho mỗi truy vấn, nó áp dụng cho truy vấn dựa trên thuộc tính phiên của SYS\_CONTEXT. Trong trường hợp của ví dụ trên, điều kiện chỉ trả về các hàng có số bộ phận khớp với thuộc tính deptno của SYS\_CONTEXT, là số bộ phận của người dùng đang truy vấn đối tượng được bảo vệ.

**Note:** Khi sử dụng các chính sách tĩnh chung, hãy đảm bảo rằng điều kiện không chứa các thuộc tính cụ thể cho một đối tượng cơ sở dữ liệu cụ thể, chẳng hạn như tên cột.

**See Also:** ["About Auditing Functions, Procedures, Packages, and](#_bookmark1850) [Triggers" on page 9-32](#_bookmark1850) for information about how Oracle Database audits the underlying policy function for static policies

###### Dùng một chính sách tĩnh chung để chia sẻ một chính sách cho nhiều đối tượng

Nếu, ví dụ, bạn muốn áp dụng chính sách trong Ví dụ 7–6 cho bảng thứ hai trong lược đồ HR có thể chứa dữ liệu tài chính mà bạn muốn sử dụng, bạn sẽ sử dụng cài đặt SHARED\_STATIC cho cả hai bảng.

[Example 7–7](#_bookmark1448) shows how to set the SHARED\_STATIC policy type for two tables that share the same policy.

***Example 7–7 Creating a SHARED\_STATIC Policy with DBMS\_RLS.ADD\_POLICY***

**-- 1. Create a policy for the first table, employees:**

BEGIN DBMS\_RLS.ADD\_POLICY(

object\_schema => 'hr', object\_name => 'employees', policy\_name => 'secure\_update', policy\_function => 'hide\_fin',

policy\_type => dbms\_rls.SHARED\_STATIC); END;

/

**-- 2. Create a policy for the second table, fin\_data:**

BEGIN DBMS\_RLS.ADD\_POLICY(

object\_schema => 'hr', object\_name => 'fin\_data', policy\_name => 'secure\_update', policy\_function => 'hide\_fin',

policy\_type => dbms\_rls.SHARED\_STATIC); END;

/

###### When to Use Static and Shared Static Policies

Các chính sách tĩnh là lý tưởng cho các môi trường mà mỗi truy vấn yêu cầu cùng một điều kiện và hiệu năng nhanh là rất cần thiết, chẳng hạn như môi trường lưu trữ. Đối với những tình huống này khi hàm chính sách nối cùng một điều kiện cho mọi truy vấn, hãy chạy lại hàm chính sách mỗi lần gây lãng phí không cần thiết vào hệ thống. Ví dụ: xem xét kho dữ liệu chứa dữ liệu nghiên cứu thị trường cho các tổ chức khách hàng là đối thủ cạnh tranh. Nhà kho phải thực thi chính sách rằng mỗi tổ chức chỉ có thể xem nghiên cứu thị trường của riêng họ, được thể hiện bằng điều kiện sau:

WHERE subscriber\_id = SYS\_CONTEXT('customer', 'cust\_num')

Sử dụng SYS\_CONTEXT cho ngữ cảnh ứng dụng cho phép cơ sở dữ liệu tự động thay đổi các hàng được trả về. Bạn không cần phải chạy lại hàm, vì vậy điều kiện có thể được lưu trữ trong SGA, do đó bảo tồn tài nguyên hệ thống và cải thiện hiệu suất.

###### Using a Context-Sensitive Policy for Predicates That Do Not Change After Parsing

Ngược lại với các chính sách tĩnh, các chính sách theo ngữ cảnh không phải lúc nào cũng lưu bộ nhớ cache các điều kiện. Với các chính sách ngữ cảnh nhạy cảm, csdl giả định rằng điều kiện sẽ thay đổi sau thời gian phân tích câu lệnh. Nhưng nếu không có sự thay đổi trong bối cảnh ứng dụng cục bộ, Csdl Oracle không chạy lại hàm chính sách trong phiên người dùng. Nếu có thay đổi về ngữ cảnh, thì cơ sở dữ liệu sẽ chạy lại hàm chính sách để đảm bảo rằng nó nắm bắt bất kỳ thay đổi nào đối với điều kiện kể từ khi phân tích cú pháp ban đầu.

Bạn có thể bật chính sách theo ngữ cảnh bằng cách đặt tham số policy\_type của thủ tục DBMS\_RLS.ADD\_POLICY thành CONTEXT\_SENSITIVE hoặc SHARED\_CONTEXT\_ SENSITIVE.

[Example 7–8](#_bookmark1454) shows how to create the CONTEXT\_SENSITIVE policy type.

***Example 7–8 Creating a CONTEXT\_SENSITIVE Policy with DBMS\_RLS.ADD\_POLICY***

BEGIN DBMS\_RLS.ADD\_POLICY(

object\_schema => 'hr',

object\_name => 'employees', policy\_name => 'secure\_update', policy\_function => 'hide\_fin',

policy\_type => dbms\_rls.CONTEXT\_SENSITIVE); END;

/

Các chính sách theo ngữ cảnh rất hữu ích khi các điều kiện khác nhau nên áp dụng tùy thuộc vào người dùng nào đang thực hiện truy vấn. Ví dụ, hãy xem xét trường hợp người quản lý nên có điều kiện WHERE được đặt thành người quản lý và nhân viên phải có điều kiện WHERE empno được đặt thành emp\_id.

Các chính sách chung thuộc ngữ cảnh nhạy cảm hoạt động như các chính sách ngữ cảnh nhạy cảm thông thường, ngoại trừ chúng có thể được chia sẻ trên nhiều đối tượng cơ sở dữ liệu. Đối với loại chính sách này, tất cả các đối tượng có thể chia sẻ chức năng chính sách từ UGA, trong đó điều kiện được lưu trữ cho đến khi ngữ cảnh phiên cục bộ thay đổi.

**Note:** Khi sử dụng chung chính sách ngữ cảnh nhạy cảm, đảm bảo rằng điều kiện chính sách không chứa các thuộc tính cụ thể cho một đối tượng cơ sở dữ liệu cụ thể, chẳng hạn như tên cột.

**See Also:** ["About Auditing Functions, Procedures, Packages, and](#_bookmark1850) [Triggers" on page 9-32](#_bookmark1850) for information about how Oracle Database audits the underlying policy function for dynamic policies

###### Using a Shared Context Sensitive Policy to Share a Policy with Multiple Objects

[Example 7–9](#_bookmark1457) Ishows how to create two shared context sensitive policies that share a policy with multiple tables.

***Example 7–9 Creating a SHARED\_CONTEXT\_SENSITIVE Policy with DBMS\_RLS.ADD\_ POLICY***

**-- 1. Create a policy for the first table, employees:**

BEGIN DBMS\_RLS.ADD\_POLICY(

object\_schema => 'hr', object\_name => 'employees', policy\_name => 'secure\_update', policy\_function => 'hide\_fin',

policy\_type => dbms\_rls.SHARED\_CONTEXT\_SENSITIVE); END;

/

**--2. Create a policy for the second table, fin\_data:**

BEGIN DBMS\_RLS.ADD\_POLICY(

object\_schema => 'hr', object\_name => 'fin\_data', policy\_name => 'secure\_update', policy\_function => 'hide\_fin',

policy\_type => dbms\_rls.SHARED\_CONTEXT\_SENSITIVE); END;

/

###### Khi nào dùng ngữ cảnh nhạy cảm và các chính sách của nó

Các chính sách theo ngữ cảnh nhạy cảm rất hữu ích khi một điều kiện không cần phải thay đổi cho một phiên người dùng, nhưng chính sách phải thực thi hai hoặc

nhiều điều kiện khác nhau người dùng hoặc nhóm. Ví dụ, hãy xem xét một bảng sales\_history với một chính sách duy nhất. Chính sách này quy định rằng các nhà phân tích chỉ có thể thấy các sản phẩm của riêng họ và nhân viên khu vực chỉ có thể nhìn thấy khu vực của họ. Trong trường hợp này, csdl phải chạy lại hàm chính sách mỗi lần loại người dùng thay đổi. Hiệu suất đạt được được thực hiện khi người dùng có thể đăng nhập và phát hành một số câu lệnh DML đối với đối tượng được bảo vệ mà không làm cho máy chủ chạy lại hàm chính sách.

**Note:** Đối với session pooling, trong đó nhiều máy khách chia sẻ một phiên cơ sở dữ liệu, tầng giữa phải đặt lại ngữ cảnh trong quá trình máy khách thay đổi.

###### Summary of the Five Oracle Virtual Private Database Policy Types

[Table 7–2](#_bookmark1464) summarizes the types of policy types available.

***Table 7–2 DBMS\_RLS.ADD\_POLICY Policy Types***

|  |  |  |  |
| --- | --- | --- | --- |
| **Policy** | **Types When the Policy Function Executes** | **Usage Example** | **Shared Across Multiple Objects?** |
| DYNAMI | C Policy function re-executes every time a policy-protected database object is accessed. | Applications where policy predicates must be generated for each query, such as time-dependent policies where users are denied access to database objects at certain times during the day | No |
| STATIC  SHARED STATIC  CONTEX SENSIT | Once, then the predicate is cached in the SGA1  \_ Same as STATIC  T\_ ■ At statement parse time  IVE   * At statement execution time   when the local application context changed since the last | View replacement  Hosting environments, such as data warehouses where the same predicate must be applied to multiple database objects  Three-tier, session pooling applications where policies enforce two or more predicates for different users or groups | No Yes  No |
|  |  |  |  |
|  | use of the cursor |  |  |
| SHARED CONTEX SENSIT | \_ First time the object is reference in a  T\_ database session.  IVE Predicates are cached in the private session memory UGA so policy functions can be shared among objects. | Same as CONTEXT\_SENSITIVE, but multiple objects can share the policy function from the session UGA | Yes |

1 Each execution of the same cursor could produce a different row set for the same predicate because the predicate may filter the data differently based on attributes such as SYS\_CONTEXT or SYSDATE.

### Tutorials: Creating Oracle Virtual Private Database Policies

This section contains:

* [Tutorial: Creating a Simple Oracle Virtual Private Database Policy](#_bookmark1467)
* [Tutorial: Implementing a Policy with a Database Session-Based Application Context](#_bookmark1475)
* [Tutorial: Implementing an Oracle Virtual Private Database Policy Group](#_bookmark1488)

#### Tutorial: Creating a Simple Oracle Virtual Private Database Policy

This section contains:

* [About This Tutorial](#_bookmark1468)
  + [Step 1: Ensure That the OE User Account Is Active](#_bookmark1470)
  + [Step 2: Create a Policy Function](#_bookmark1471)
  + [Step 3: Create the Oracle Virtual Private Database Policy](#_bookmark1472)
  + [Step 4: Test the Policy](#_bookmark1473)
  + [Step 5: Remove the Components for This Tutorial](#_bookmark1474)

###### Về hướng dẫn này

Giả sử bạn muốn tạo một chính sách Oracle VPD đơn giản giới hạn quyền truy cập vào tất cả các đơn đặt hàng trong bảng OE.ORDERS được tạo ra bởi Đại diện bán hàng 159. Về bản chất, chính sách dịch câu lệnh sau:

SELECT \* FROM OE.ORDERS;

Trở thành:

SELECT \* FROM OE.ORDERS WHERE SALES\_REP\_ID = 159;

###### Step 1: Đảm bảo rằng tài khoản người dùng OE đang hoạt động

1. Đăng nhập vào SQL\*Plus với user SYSTEM với ưu tiên SYSDBA.

sqlplus sys as sysdba Enter password: *password*

1. Chạy câu lệnh SELECT sau trên DBA\_USERS data dictionary view:

SELECT USERNAME, ACCOUNT\_STATUS FROM DBA\_USERS WHERE USERNAME = 'OE';

Nếu chế độ xem DBA\_USERS liệt kê OE của người dùng là bị khóa và đã hết hạn, hãy nhập câu lệnh sau để mở khóa tài khoản OE và tạo một mật khẩu mới::

ALTER USER OE ACCOUNT UNLOCK IDENTIFIED BY *password*;

Thay thế mật khẩu bằng mật khẩu an toàn. Để bảo mật tốt hơn, không sử dụng lại cùng một mật khẩu đã được sử dụng trong các bản phát hành trước của CSDL Oracle. See ["Minimum Requirements for Passwords" on page 3-3](#_bookmark197) for more information.

###### Step 2: Tạo hàm chính sách

Tạo hàm sau, nó sẽ thêm mệnh đề WHERE SALES\_REP\_ID = 159 vào bất kỳ câu lệnh SELECT nào trên bảng OE.ORDERS. (Bạn có thể sao chép và dán văn bản này bằng cách định vị con trỏ ở đầu CREATE OR REPLACE ở dòng đầu tiên.)

CREATE OR REPLACE FUNCTION auth\_orders( schema\_var IN VARCHAR2,

table\_var IN VARCHAR2

)

RETURN VARCHAR2 IS

return\_val VARCHAR2 (400); BEGIN

return\_val := 'SALES\_REP\_ID = 159'; RETURN return\_val;

END auth\_orders;

/

Trong ví dụ này:

* + schema\_var và table\_var tạo các tham số đầu vào để xác định để lưu trữ tên schema, OE và tên bảng, ORDERS. Đầu tiên, xác định tham số cho schema, và sau đó xác định tham số cho đối tượng, trong trường hợp này là một bảng. Luôn tạo chúng theo thứ tự này. Chính sách VPD mà bạn tạo sẽ cần các tham số này để chỉ định bảng OE.ORDERS.
  + RETURN VARCHAR2 trả về chuỗi sẽ được sử dụng cho mệnh đề điều kiện WHERE. Hãy nhớ rằng giá trị trả về luôn là kiểu dữ liệu VARCHAR2.
  + IS ... RETURN return\_val bao gồm việc tạo điều kiện WHERE SALES\_REP\_ID= 159.

.

###### Step 3: Tạo chính sách Oracle VPD

Tiếp theo, tạo chính sách sau bằng cách sử dụng thủ tục ADD\_POLICY trong gói DBMS\_RLS. (Bạn có thể sao chép và dán văn bản này bằng cách định vị con trỏ ở đầu BEGIN trong dòng đầu tiên.)

BEGIN

DBMS\_RLS.ADD\_POLICY (

object\_schema => 'oe', object\_name => 'orders', policy\_name => 'orders\_policy', function\_schema => 'sys', policy\_function => 'auth\_orders',

statement\_types => 'select, insert, update, delete'

); END;

/

Trong ví dụ này:

* + object\_schema => 'oe' chỉ định schema mà bạn muốn bảo vệ, đó là, OE.
  + object\_name => 'orders' chỉ định đối tượng trong schema để bảo vệ, đó là bảng ORDERS.
  + policy\_name => 'orders\_policy' làm rõ cho chính sách này orders\_policy.
  + function\_schema => 'sys' chỉ định schema trong đó hàm auth\_orders được tạo. Trong ví dụ này, auth\_orders được tạo trong schema SYS. Nhưng thông thường, nó sẽ được tạo trong schema của quản trị viên bảo mật.
  + policy\_function => 'auth\_orders' chỉ định một hàm để thực thi chính sách. Ở đây, bạn chỉ định hàm auth\_orders mà bạn đã tạo ở Bước 2: Tạo một hàm chính sách.
  + statement\_types => 'select' chỉ định các hoạt động mà chính sách áp dụng. Trong ví dụ này, chính sách áp dụng cho tất cả các câu lệnh SELECT, INSERT, UPDATE và DELETE mà người dùng có thể thực hiện.

###### Step 4: Chạy thử chính sách

Sau khi bạn tạo chính sách Oracle VPD, nó sẽ có hiệu lực ngay lập tức. Lần tiếp theo người dùng, bao gồm chủ sở hữu của schema, thực hiện lệnh SELECT trên OE.ORDERS, chỉ các đơn hàng của Đại diện bán hàng 159 mới được truy cập.

1. Đăng nhập với user OE.

CONNECT oe

Enter password: *password*

1. Nhập vào câu lệnh SELECT:

SELECT COUNT(\*) FROM ORDERS;

Output sau sẽ xuất hiện:

COUNT(\*)

---------

7

Chính sách có hiệu lực đối với người dùng OE: Như bạn thấy, chỉ có 7 trong số 105 hàng trong bảng đơn hàng được trả về.

Nhưng người dùng có đặc quyền quản trị vẫn có quyền truy cập vào tất cả các hàng trong bảng.

1. Đăng nhập lại với tư cách user SYS.

CONNECT sys/as sysdba Enter password: *password*

1. Nhập vào câu lệnh SELECT sau:

SELECT COUNT(\*) FROM OE.ORDERS;

Kết quả sẽ là:

COUNT(\*)

---------

105

###### Step 5: Xoá các thành phần cho hướng dẫn này

1. Nếu là user SYS, xoá các hàm và chính sách như sau:

DROP FUNCTION auth\_orders;

EXEC DBMS\_RLS.DROP\_POLICY('OE','ORDERS','ORDERS\_POLICY');

1. Nếu bạn cần khóa và làm hết hạn tài khoản OE, hãy nhập câu lệnh sau:

ALTER USER OE ACCOUNT LOCK PASSWORD EXPIRE;

#### Tutorial: Triển khai chính sách với ngữ cảnh ứng dụng CSDL theo phiên

Phần này bao gồm:

* + Tổng quát về hướng dẫn này
  + [Step 1: Tạo](#_bookmark1478) tài khoản người dùng và bảng mẫu
  + [Step 2: Tạo](#_bookmark1479) một ngữ cảnh ứng dụng CSDL theo phiên
  + [Step 3: Tạo](#_bookmark1480) một gói PL/SQL để cài đặt ngữ cảnh ứng dụng
  + [Step 4: Tạo trigger đăng nhập để chạy gói PL / SQL của ứng dụng ngữ cảnh](#_bookmark1481)
  + [Step 5: Tạo một hàm chính sách PL / SQL để hạn chế quyền truy cập của người dùng vào các đơn đặt hàng của họ](#_bookmark1482)
  + [Step 6: Tạo](#_bookmark1483) mới một chính sách bảo mật
  + [Step 7: Chạy](#_bookmark1484) thử chính sách mới
  + [Step 8: Xoá](#_bookmark1487) các chính sách

###### Tổng quát về tutorial này

Hướng dẫn này cho thấy làm thế nào bạn có thể sử dụng một ngữ ứng dụng CSDL dựa trên phiên để thực hiện một chính sách, trong đó khách hàng có thể chỉ thấy đơn đặt hàng riêng của họ. Bạn tạo các lớp bảo mật sau:

* 1. Khi người dùng đăng nhập, ngữ cảnh ứng dụng CSDL dựa trên phiên kiểm tra xem người dùng có phải là khách hàng hay không. Nếu người dùng không phải là khách hàng, người dùng vẫn có thể đăng nhập nhưng người dùng này không thể truy cập vào bảng mục nhập đơn đặt hàng mà bạn sẽ tạo cho ví dụ này.
  2. Nếu người dùng là khách hàng, họ có thể đăng nhập. Sau khi khách hàng đăng nhập, chính sách Oracle VPD hạn chế người dùng này chỉ xem các đơn đặt hàng của họ.
  3. Như một hạn chế hơn nữa, chính sách Oracle VPD ngăn người dùng thêm, sửa đổi hoặc xóa các đơn đặt hàng.

###### Step 1: Tạo các tài khoản và bảng mẫu

1. Mở SQL\*Plus và đăng nhập tài khoản quản trị.

sqlplus sys as sysdba Enter password: *password*

1. Tạo người dùng quản trị như sau, người sẽ quản trị chính sách Oracle VPD.

Các câu lệnh SQL sau tạo user này và sau đó cấp cho user các đặc quyền cần thiết để hoàn thành hướng dẫn này.

GRANT CREATE SESSION, CREATE ANY CONTEXT, CREATE PROCEDURE, CREATE TRIGGER,

ADMINISTER DATABASE TRIGGER TO sysadmin\_vpd IDENTIFIED BY *password*; GRANT EXECUTE ON DBMS\_SESSION TO sysadmin\_vpd;

GRANT EXECUTE ON DBMS\_RLS TO sysadmin\_vpd;

Thay *password*  bằng một password an toàn. See ["Minimum Requirements for](#_bookmark197) [Passwords" on page 3-3](#_bookmark197) for more information.

1. Tạo các tài khoản user sau:

GRANT CREATE SESSION TO tbrooke IDENTIFIED BY *password*; GRANT CREATE SESSION TO owoods IDENTIFIED BY *password*;

Thay *password*  bằng một password an toàn. See ["Minimum Requirements for](#_bookmark197) [Passwords" on page 3-3](#_bookmark197) for more information.

1. Kiểm tra trạng thái của người dùng mẫu SCOTT, cái mà bạn sẽ sử dụng cho hướng dẫn này:

SELECT USERNAME, ACCOUNT\_STATUS FROM DBA\_USERS WHERE USERNAME = 'SCOTT';

Nếu view DBA\_USERS liệt kê SCOTT đã khóa và đã hết hạn, hãy nhập câu lệnh sau để mở khóa tài khoản SCOTT và tạo mật khẩu mới cho anh ta:

ALTER USER SCOTT ACCOUNT UNLOCK IDENTIFIED BY *password*;

Thay *password*  bằng một password an toàn. For greater security, do not reuse the same password that was used in previous releases of Oracle Database. See ["Minimum Requirements for Passwords" on page 3-3](#_bookmark197) for more information.

1. Kết nối bằng user SCOTT, và sau đó tạo và điền bảng khách hàng.

CONNECT scott

Enter password: *password*

CREATE TABLE customers ( cust\_no NUMBER(4), cust\_email VARCHAR2(20), cust\_name VARCHAR2(20));

INSERT INTO customers VALUES (1234, 'TBROOKE', 'Thadeus Brooke'); INSERT INTO customers VALUES (5678, 'OWOODS', 'Oberon Woods');

Khi bạn nhập địa chỉ email của người dùng, hãy nhập chúng vào các chữ hoa. Sau này, khi bạn tạo ra gói ngữ cảnh ứng dụng PL / SQL , tham số SESSION\_USER của hàm SYS\_CONTEXT muốn tên người dùng viết hoa. Nếu không, bạn sẽ không thể đặt ngữ cảnh ứng dụng cho người dùng.

1. Người dùng sysadmin\_vpd sẽ cần các đặc quyền SELECT cho bảng khách hàng, như user SCOTT, cấp cho anh ta đặc quyền này

.

GRANT SELECT ON customers TO sysadmin\_vpd;

1. Tạo và điền bảng orders\_tab table.

CREATE TABLE orders\_tab ( cust\_no NUMBER(4), order\_no NUMBER(4));

INSERT INTO orders\_tab VALUES (1234, 9876);

INSERT INTO orders\_tab VALUES (5678, 5432);

INSERT INTO orders\_tab VALUES (5678, 4592);

1. Người dùng tbrooke và owoods cần truy vấn bảng orders\_tab, vì vậy hãy cấp cho họ quyền SELECT.

GRANT SELECT ON orders\_tab TO tbrooke; GRANT SELECT ON orders\_tab TO owoods;

Ở giai đoạn này, hai khách hàng mẫu, tbrooke và owoods, có hồ sơ mua hàng trong bảng đơn đặt hàng orders\_tab và nếu họ đã thử ngay bây giờ, họ có thể xem tất cả các đơn đặt hàng trong bảng này.

###### Step 2: Tạo một ngữ cảnh ứng dụng CSDL theo phiên

1. Kết nối bằng user sysadmin\_vpd.

CONNECT sysadmin\_vpd Enter password: *password*

1. Nhập câu lệnh sau:

CREATE OR REPLACE CONTEXT orders\_ctx USING orders\_ctx\_pkg;

Câu lệnh này tạo ra ngữ cảnh ứng dụng orders\_ctx. Hãy nhớ rằng mặc dù người dùng sysadmin\_vpd đã tạo ra ngữ cảnh này và nó được liên kết với lược đồ sysadmin\_vpd, lược đồ SYS sở hữu ngữ cảnh ứng dụng.

###### Step 3: Tạo một gói PL/SQL để cài đặt ngữ cảnh ứng dụng

As user sysadmin\_vpd, create the following PL/SQL package, which will set the database session-based application context when the customers tbrooke and owoods log onto their accounts. (You can copy and paste this text by positioning the cursor at the start of CREATE OR REPLACE in the first line.)

CREATE OR REPLACE PACKAGE orders\_ctx\_pkg IS

PROCEDURE set\_custnum; END;

/

CREATE OR REPLACE PACKAGE BODY orders\_ctx\_pkg IS PROCEDURE set\_custnum

AS

custnum NUMBER; BEGIN

SELECT cust\_no INTO custnum FROM SCOTT.CUSTOMERS

WHERE cust\_email = SYS\_CONTEXT('USERENV', 'SESSION\_USER');

DBMS\_SESSION.SET\_CONTEXT('orders\_ctx', 'cust\_no', custnum); EXCEPTION

WHEN NO\_DATA\_FOUND THEN NULL;

END set\_custnum;

END;

/

Trong ví dụ này:

* + custnum NUMBER tạo biến custnum, sẽ giữ ID khách hàng.
  + SELECT cust\_no INTO custnum thực hiện một câu lệnh SELECT để sao chép ID khách hàng được lưu trữ trong dữ liệu cột cust\_no từ bảng scott.customers vào biến custnum.
  + WHERE cust\_email = SYS\_CONTEXT ('USERENV', 'SESSION\_USER') sử dụng mệnh đề WHERE để tìm tất cả các ID khách hàng khớp với tên người dùng của người dùng đang đăng nhập.
  + DBMS\_SESSION.SET\_CONTEXT ('orders\_ctx', 'cust\_no', custnum) đặt giá trị ngữ cảnh ứng dụng orders\_ctx bằng cách tạo thuộc tính cust\_no và sau đó đặt nó thành giá trị được lưu trữ trong biến custnum.

EXCEPTION ... WHEN thêm một ngoại lệ hệ thống WHEN NO\_DATA\_FOUND để bắt ngoại lệ bất cứ không tìm thấy dữ liệu nào có thể phát sinh từ câu lệnh SELECT trong dòng 10-11.

Để tóm tắt, thủ tục sysadmin\_vpd.set\_cust\_num xác định người dùng phiên có phải là khách hàng đã đăng ký hay không bằng cách cố gắng chọn ID khách hàng của người dùng vào biến custnum. Nếu người dùng là khách hàng đã đăng ký, thì Cơ sở dữ liệu Oracle sẽ đặt giá trị ngữ cảnh của ứng dụng cho người dùng này. Như bạn sẽ thấy trong Bước 5: Tạo một hàm chính sách PL / SQL để giới hạn quyền truy cập của người dùng vào các đơn đặt hàng của họ, hàm chính sách sử dụng giá trị ngữ cảnh để kiểm soát quyền truy cập của người dùng có dữ liệu trong bảng orders\_tab.

###### Step 4: Tạo trigger đăng nhập để chạy gói PL / SQL của ứng dụng ngữ cảnh

Trigger đăng nhập chạy thủ tục trong gói PL / SQL mà bạn đã tạo trong Bước 3: Tạo một gói PL/SQL để cài đặt ngữ cảnh ứng dụng vào lần sau khi người dùng đăng nhập, để ngữ cảnh ứng dụng có thể được thiết lập.

Với user sysadmin\_vpd, tạo trigger như sau:

CREATE TRIGGER set\_custno\_ctx\_trig AFTER LOGON ON DATABASE BEGIN

sysadmin\_vpd.orders\_ctx\_pkg.set\_custnum; END;

/

**See Also:** ["Creating a Logon Trigger to Run a Database Session](#_bookmark1194) [Application Context Package" on page 6-11](#_bookmark1194)

Ở giai đoạn này, nếu bạn đăng nhập dưới dạng tbrooke hoặc owoods, trigger đăng nhập sẽ đặt ngữ cảnh ứng dụng cho người dùng khi nó kích hoạt thủ tục sysadmin\_vpd.orders\_ctx\_pkg.set\_ custnum. Bạn có thể kiểm tra nó như sau:

CONNECT tbrooke

Enter password: *password*

SELECT SYS\_CONTEXT('orders\_ctx', 'cust\_no') custnum FROM DUAL;

Kết quả như sau:

EMP\_ID

------------------------------------------------------------------- 1234

###### Step 5: Tạo một hàm chính sách PL / SQL để hạn chế quyền truy cập của người dùng vào các đơn đặt hàng của họ

Bước tiếp theo là tạo một hàm PL / SQL, khi người dùng đã đăng nhập thực hiện truy vấn SELECT \* FROM scott.orders\_tab, chỉ hiển thị các đơn đặt hàng của người dùng đó.

Với user sysadmin\_vpd, tạo hàm như sau:

CREATE OR REPLACE FUNCTION get\_user\_orders( schema\_p IN VARCHAR2,

table\_p IN VARCHAR2) RETURN VARCHAR2

AS

orders\_pred VARCHAR2 (400); BEGIN

orders\_pred := 'cust\_no = SYS\_CONTEXT(''orders\_ctx'', ''cust\_no'')'; RETURN orders\_pred;

END;

/

Hàm này tạo và trả về một biến điều kiện WHERE chuyển thành "nơi các đơn hàng được hiển thị thuộc về người dùng đã đăng nhập". Sau đó nó gắn thêm điều kiện WHERE này vào bất kỳ truy vấn nào mà người dùng này có thể chạy trên bảng scott.orders\_tab. Tiếp theo, bạn đã sẵn sàng để tạo ra một chính sách Oracle VPD áp dụng hàm này cho bảng orders\_tab.

###### Step 6: Tạo mới một chính sách bảo mật

Với user sysadmin\_vpd, tạo chính sách như sau:

BEGIN DBMS\_RLS.ADD\_POLICY (

object\_schema => 'scott', object\_name => 'orders\_tab', policy\_name => 'orders\_policy', function\_schema => 'sysadmin\_vpd', policy\_function => 'get\_user\_orders', statement\_types => 'select');

END;

/

Câu lệnh này tạo một chính sách có tên orders\_policy và áp dụng chính sách đó cho bảng orders\_tab, mà khách hàng sẽ truy vấn các đơn đặt hàng của họ, trong lược đồ SCOTT. Hàm get\_user\_ orders thực thi chính sách, được lưu trữ trong lược đồ sysadmin\_vpd.Chính sách tiếp tục hạn chế người dùng chỉ chạy được các câu lệnh SELECT.

###### Step 7: Chạy thử chính sách mới

1. Đăng nhập bằng user tbrooke.

CONNECT tbrooke

Enter password: *password*

Người dùng tbrooke có thể đăng nhập vì anh ta đã vượt qua các yêu cầu mà bạn đã xác định trong ngữ cảnh ứng dụng.

1. Với user tbrooke, truy cập mua hàng của bạn.

SELECT \* FROM scott.orders\_tab;

Kết quả sẽ như sau:

CUST\_NO ORDER\_NO

---------- ---------- 1234 9876

Người dùng tbrooke đã vượt qua bài kiểm tra thứ hai. Anh ta có thể truy cập các đơn đặt hàng của riêng mình trongbảng scott.orders\_tab.

1. Đăng nhập với user owoods, và truy cập mua hàng.

CONNECT owoods

Enter password: *passwords*

SELECT \* FROM scott.orders\_tab

Kết quả như sau:

CUST\_NO ORDER\_NO

---------- ---------- 5678 5432

5678 4592

Như với người dùng tbrooke, người dùng owoods có thể đăng nhập và xem danh sách các đơn đặt hàng của riêng mình.

Chú ý những điều sau:

* + Bạn có thể tạo một số điều kiện dựa trên vị trí của người dùng. Ví dụ: đại diện bán hàng chỉ có thể xem hồ sơ cho khách hàng của mình và nhân viên bán hàng có thể xem bất kỳ đơn đặt hàng nào của khách hàng. Bạn có thể mở rộng hàm custnum\_sec để trả về các điều kiện khác nhau dựa trên giá trị ngữ cảnh vị trí của người dùng.
  + Việc sử dụng ngữ cảnh ứng dụng trong gói kiểm soát truy cập chi tiết có hiệu quả sẽ cho bạn một biến liên kết trong một câu lệnh được phân tích cú pháp. Ví dụ:

SELECT \* FROM scott.orders\_tab

WHERE cust\_no = SYS\_CONTEXT('order\_entry', 'cust\_num');

Điều này được phân tích và tối ưu hóa đầy đủ, nhưng việc đánh giá giá trị thuộc tính cust\_num của người dùng cho bối cảnh order\_entry diễn ra vào thời gian chạy. Điều này có nghĩa là bạn nhận được lợi ích của một câu lệnh được tối ưu hóa thực hiện khác nhau cho mỗi người dùng phát hành câu lệnh.

**Note:** Bạn có thể cải thiện hiệu suất của hàm trong hướng dẫn này bằng cách lập chỉ mục cust\_no.

* + Bạn có thể đặt thuộc tính ngữ cảnh dựa trên dữ liệu từ bảng hoặc bảng cơ sở dữ liệu hoặc từ máy chủ thư mục sử dụng Giao thức truy cập thư mục hạng nhẹ (LDAP).

**See Also:** *Oracle Database PL/SQL Language Reference* for more information about triggers

Compare and contrast this tutorial, which uses an application context within the dynamically generated predicate, with ["About Oracle Virtual Private Database](#_bookmark1380) [Policies" on page 7-6](#_bookmark1380), which uses a subquery in the predicate.

###### Step 8: Xoá các chính sách trong hướng dẫn này

1. Kết nối bằng user OE và xóa bảng orders\_tab và bảng khách hàng.

CONNNECT SCOTT

Enter password: *password*

DROP TABLE orders\_tab; DROP TABLE customers;

1. Kết nối bằng user SYS, kết nối với AS SYSDBA.

CONNECT sys/as sysdba Enter password: *password*

1. Chạy những câu lệnh sau để xoá các chính sách trong hướng dẫn này:

DROP CONTEXT orders\_ctx;

DROP USER sysadmin\_vpd CASCADE; DROP USER tbrooke;

DROP USER owoods;

#### Tutorial: Triển khai một nhóm chính sách cơ sở dữ liệu riêng ảo của Oracle

Phần này bao gồm:

* + [Nói](#_bookmark1489) về hướng dẫn này
  + [Step 1: Tạo](#_bookmark1493) tài khoản người dùng và các thành phần khác cho bài học này
  + [Step 2: Tạo](#_bookmark1494) các nhóm hai chính sách
  + [Step 3: Tạo các hàm PL / SQL để kiểm soát các nhóm chính sách](#_bookmark1495)
  + [Step 4: Thêm](#_bookmark1496) các hàm PL / SQL vào các nhóm chính sách
  + [Step 5: Tạo](#_bookmark1497) ngữ cảnh ứng dụng điều khiển
  + [Step 6: Chạy](#_bookmark1498) thử các chính sách
  + [Step 7: Xoá](#_bookmark1499) các thành phần trong hướng dẫn này

###### Nói về hướng dẫn này

["Working with Oracle Virtual Private Database Policy Groups" on page 7-11](#_bookmark1419) mô tả cách bạn có thể nhóm một bộ chính sách để sử dụng trong một ứng dụng. Khi một người dùng không chính thức đăng nhập vào ứng dụng, Cơ sở dữ liệu Oracle cấp quyền truy cập cho người dùng dựa trên các chính sách được định nghĩa trong nhóm chính sách thích hợp.

Đối với kiểm soát truy cập cấp cột, mỗi cột hoặc tập hợp các cột ẩn được kiểm soát bởi một chính sách. Trong hướng dẫn này, bạn phải ẩn hai bộ cột. Vì vậy, bạn cần phải tạo hai chính sách, một cho mỗi tập hợp các cột mà bạn muốn ẩn. Bạn chỉ muốn có một chính sách cho mỗi người dùng; ngữ cảnh ứng dụng kiểm soát tách biệt các chính sách cho bạn.

###### Step 1: Tạo tài khoản người dùng,các thành phần khác cho bài học này

* 1. Đăng nhập với user SYS với quyền SYSDBA .

sqlplus sys as sysdba Enter password: *password*

* 1. Tạo các user sau:

GRANT CREATE SESSION TO apps\_user IDENTIFIED BY *password*;

GRANT CREATE SESSION, CREATE PROCEDURE, CREATE ANY CONTEXT TO sysadmin\_pg

IDENTIFIED BY *password*;

thay *password*  bằng một password an toàn. See ["Minimum Requirements for](#_bookmark197) [Passwords" on page 3-3](#_bookmark197) for more information.

* 1. Cấp đặc quyền bổ sung sau cho người dùng sysadmin\_pg:

GRANT EXECUTE ON DBMS\_RLS TO sysadmin\_pg;

* 1. Đăng nhập với user OE.

CONNECT OE

Enter password: *password*

Nếu tài khoản OE bị khóa và hết hạn, sau đó kết nối lại với tư cách người dùng SYS với đặc quyền SYSDBA và nhập câu lệnh sau để mở khóa tài khoản và cung cấp mật khẩu mới:

ALTER USER OE ACCOUNT UNLOCK IDENTIFIED BY *password*;

thay *password*  bằng một password an toàn. For greater security, do not reuse the same password that was used in previous releases of Oracle Database. See ["Minimum Requirements for Passwords" on page 3-3](#_bookmark197) for more information.

* 1. Tạo bảng product\_code\_names:

CREATE TABLE product\_code\_names( group\_a varchar2(32),

year\_a varchar2(32), group\_b varchar2(32), year\_b varchar2(32));

* 1. Thêm một vài giá trị vào product\_code\_names table:

INSERT INTO product\_code\_names values('Biffo','2008','Beffo','2004'); INSERT INTO product\_code\_names values('Hortensia','2008','Bunko','2008'); INSERT INTO product\_code\_names values('Boppo','2006','Hortensia','2003');

COMMIT;

* 1. Cấp cho người dùng apps\_user quyền SELECT bảng the product\_code\_names table.

GRANT SELECT ON product\_code\_names TO apps\_user;

###### Step 2: Tạo các nhóm hai chính sách

Tiếp theo, bạn phải tạo một nhóm chính sách từng người không có csdl, provider\_a và provaider\_b.

1. Kết nối bằng user sysadmin\_pg.

CONNECT sysadmin\_pg Enter password: *password*

1. Tạo nhóm provider\_a\_group polic, sử dụng bởi user provider\_a:

BEGIN DBMS\_RLS.CREATE\_POLICY\_GROUP(

object\_schema => 'oe',

object\_name => 'product\_code\_names', policy\_group => 'provider\_a\_group'); END;

/

1. Tạo nhóm provider\_b\_group policy, sử dụng bởi user provider\_b:

BEGIN DBMS\_RLS.CREATE\_POLICY\_GROUP(

object\_schema => 'oe',

object\_name => 'product\_code\_names', policy\_group => 'provider\_b\_group'); END;

/

###### Step 3: Tạo các hàm PL / SQL để kiểm soát các nhóm chính sách

Mỗi nhóm chính sách mà bạn đã tạo trong Bước 2: Tạo hai Nhóm chính sách phải có một hàm xác định cách ứng dụng có thể kiểm soát quyền truy cập dữ liệu cho người dùng provider\_a và provider\_b.

1. Tạo hàm vpd\_function\_provider\_a, giới hạn truy cập bởi provider\_a.

CREATE OR REPLACE FUNCTION vpd\_function\_provider\_a (schema in varchar2, tab in varchar2) return varchar2 as predicate varchar2(8) default NULL;

BEGIN

IF LOWER(SYS\_CONTEXT('USERENV','CLIENT\_IDENTIFIER')) = 'provider\_a'

THEN predicate := '1=2'; ELSE NULL;

END IF;

RETURN predicate; END;

/

Hàm này kiểm tra xem người dùng đăng nhập có thực sự là người dùng provider\_a hay không. Nếu điều này là đúng, thì chỉ dữ liệu trong cột bảng product\_code\_names group\_a và year\_a mới hiển thị với provider\_a. Dữ liệu trong cột group\_b và year\_b sẽ không xuất hiện cho provider\_a. Điều này hoạt động như sau: Đặt điều kiện: = '1 = 2' ẩn các cột có liên quan. Trong Bước 4: Thêm các hàm PL / SQL vào các nhóm chính sách, bạn chỉ rõ các cột này trong tham số SEC\_RELEVANT\_COLS.

See ["Creating a Function to Generate the Dynamic WHERE Clause" on page 7-4](#_bookmark1372) for detailed information on the components of this type of function.

1. Tạo hàm vpd\_function\_provider\_b, để giới hạn truy cập dữ liệu bởi user provider\_a.

CREATE OR REPLACE FUNCTION vpd\_function\_provider\_b (schema in varchar2, tab in varchar2) return varchar2 as predicate varchar2(8) default NULL;

BEGIN

IF LOWER(SYS\_CONTEXT('USERENV','CLIENT\_IDENTIFIER')) = 'provider\_b'

THEN predicate := '1=2'; ELSE NULL;

END IF;

RETURN predicate; END;

/

Similar to the vpd\_function\_provider\_a function, this function checks that the user logging in is really user provider\_b. If this is true, then only the data in the columns group\_b and year\_b will be visible to provider\_b, with data in the group\_a and year\_a not appearing for provider\_b. Similar to the vpd\_function\_ provider\_a function, predicate := '1=2' hides the relevant columns specified [Step 4: Add the PL/SQL Functions to the Policy Groups](#_bookmark1496) in the SEC\_RELEVANT\_COLS parameter.

###### Step 4: Thêm các hàm PL / SQL vào các nhóm chính sách

Bây giờ bạn đã tạo ra các hàm cần thiết, bạn đã sẵn sàng liên kết chúng với các nhóm chính sách thích hợp của chúng.

* 1. Thêm hàm vpd\_function\_provider\_a vào nhóm chính sách provider\_a\_group.

BEGIN DBMS\_RLS.ADD\_GROUPED\_POLICY(

object\_schema => 'oe',

object\_name => 'product\_code\_names',

policy\_group => 'provider\_a\_group',

policy\_name => 'filter\_provider\_a', function\_schema => 'sysadmin\_pg', policy\_function => 'vpd\_function\_provider\_a', statement\_types => 'select',

policy\_type => DBMS\_RLS.CONTEXT\_SENSITIVE, sec\_relevant\_cols => 'group\_b,year\_b', sec\_relevant\_cols\_opt => DBMS\_RLS.ALL\_ROWS);

END;

/

Các cột group\_b và year\_b được chỉ định trong tham số sec\_relevant\_cols bị ẩn khỏi provider\_a.

* 1. Thêm hàm vpd\_function\_provider\_b vào nhóm chính sách provider\_b\_group.

BEGIN DBMS\_RLS.ADD\_GROUPED\_POLICY(

object\_schema => 'oe',

object\_name => 'product\_code\_names',

policy\_group => 'provider\_b\_group',

policy\_name => 'filter\_provider\_b', function\_schema => 'sysadmin\_pg', policy\_function => 'vpd\_function\_provider\_b', statement\_types => 'select',

policy\_type => DBMS\_RLS.CONTEXT\_SENSITIVE, sec\_relevant\_cols => 'group\_a,year\_a', sec\_relevant\_cols\_opt => DBMS\_RLS.ALL\_ROWS);

END;

/

Cột group\_a và year\_a được chỉ định trong tham số sec\_relevant\_cols bị ẩn khỏi user provider\_b.

###### Step 5: Tạo ngữ cảnh ứng dụng điều khiển

Ngữ cảnh ứng dụng xác định chính sách nào mà người dùng không phải là người dùng đăng nhập nên sử dụng.

1. Nếu là user sysadmin\_pg, tạo ngữ cảnh ứng dụng kiểm soát như sau:

CREATE OR REPLACE CONTEXT provider\_ctx USING provider\_package;

1. Tạo gói provider\_package PL / SQL cho ngữ cảnh ứng dụng.

CREATE OR REPLACE PACKAGE provider\_package IS

PROCEDURE set\_provider\_context (policy\_group varchar2 default NULL); END;

/

CREATE OR REPLACE PACKAGE BODY provider\_package AS

PROCEDURE set\_provider\_context (policy\_group varchar2 default NULL) IS BEGIN

CASE LOWER(SYS\_CONTEXT('USERENV', 'CLIENT\_IDENTIFIER'))

WHEN 'provider\_a' THEN DBMS\_SESSION.SET\_CONTEXT('provider\_ctx','policy\_group','PROVIDER\_A\_GROUP'); WHEN 'provider\_b' THEN DBMS\_SESSION.SET\_CONTEXT('provider\_ctx','policy\_group','PROVIDER\_B\_GROUP'); END CASE;

END set\_provider\_context; END;

/

1. Liên kết ngữ cảnh ứng dụng provider\_ctx với bảng product\_code\_names, và sau đó cung cấp một tên

.

BEGIN DBMS\_RLS.ADD\_POLICY\_CONTEXT(

object\_schema =>'oe',

object\_name =>'product\_code\_names', namespace =>'provider\_ctx', attribute =>'policy\_group');

END;

/

1. Cấp tài khoản apps\_user đặc quyền EXECUTE cho gói provider\_package

.

GRANT EXECUTE ON provider\_package TO apps\_user;

###### Step 6: Chạy thử các nhóm chính sách

Bây giờ bạn đã sẵn sàng để kiểm tra hai nhóm chính sách.

1. Kết nối với tư cách người dùng apps\_user và sau đó nhập các câu lệnh sau để đảm bảo rằng kết quả bạn tạo sau này được định dạng đẹp.

CONNECT apps\_user

Enter password: *password*

col group\_a format a16 col group\_b format a16; col year\_a format a16; col year\_b format a16;

1. Đặt định danh session thành provider\_a.

EXEC DBMS\_SESSION.SET\_IDENTIFIER('provider\_a');

Ở đây, ứng dụng đặt mã định danh. Đặt mã định danh cho provider\_a đặt người dùng apps\_user thành người dùng chỉ nên xem các sản phẩm có sẵn cho các sản phẩm trong nhóm chính sách provider\_a\_group.

1. Chạy provider\_package để thiết lập nhóm chính sách dựa trên ngữ cảnh.

EXEC sysadmin\_pg.provider\_package.set\_provider\_context;

Ở đoạn này, bạn có thể kiểm tra ngữ cảnh ứng dụng đã được thiết lập, như sau:

SELECT SYS\_CONTEXT('USERENV', 'CLIENT\_IDENTIFIER') AS END\_USER FROM DUAL;

Kết quả như sau:

END\_USER

-----------------

provider\_a

1. Nhập câu lệnh SELECT sau đây:

SELECT \* FROM oe.product\_code\_names;

Kết quả như sau:

GROUP\_A YEAR\_A GROUP\_B YEAR\_B

---------------- ---------------- ---------------- ----------------

|  |  |
| --- | --- |
| Biffo | 2008 |
| Hortensia | 2008 |
| Boppo | 2006 |

1. Đặt mã nhận diện khách hàng thành provider\_b và sau đó nhập các câu lệnh sau:

EXEC DBMS\_SESSION.SET\_IDENTIFIER('provider\_b');

EXEC sysadmin\_pg.provider\_package.set\_provider\_context; SELECT \* FROM oe.product\_code\_names;

Kết quả như sau:

GROUP\_A YEAR\_A GROUP\_B YEAR\_B

---------------- ---------------- ---------------- ----------------

|  |  |
| --- | --- |
| Beffo | 2004 |
| Bunko | 2008 |
| Hortensia | 2003 |

###### Step 7: Xoá các thành phần trong hướng dẫn này

1. Kết nối bằng user OE and và xoá bảng product\_code\_names.

CONNECT OE

Enter password: *password*

DROP TABLE product\_code\_names;

1. Kết nối bằng user SYS and ngữ cảnh ứng dụng và user trong tutorial này.

CONNECT SYS/AS SYSDBA

Enter password: *password*

DROP CONTEXT provider\_ctx; DROP USER sysadmin\_pg cascade; DROP USER apps\_user;

### Oracle VPD hoạt động với các tính năng Oracle khác như thế nào

Phần này bao gồm:

* + [Sử](#_bookmark1501) dụng chính sách Oracle VPD với Editions
  + [Sử dụng SELECT FOR UPDATE trong truy vấn của user trong VPD-Protected Tables](#_bookmark1503)
  + [Oracle Virtual Private Database Policies ảnh hưởng đến Outer or ANSI Join như](#_bookmark1505) thế nào
  + [Oracle Virtual Private Database Security Policies hoạt động với](#_bookmark1508) ứng dụng như thế nào
  + [Sử dụng Automatic Reparsing cho hàm](#_bookmark1511) chính sách kiểm soát chi tiết
  + [Sử dụng Oracle Virtual Private Database Policies và Flashback Query](#_bookmark1514)
  + [Sử dụng Oracle Virtual Private Database và Oracle Label Security](#_bookmark1516)
  + [Xuất dữ liệu bằng tham số EXPDP Utility access\_method](#_bookmark1527)
  + [User Models và Oracle Virtual Private Database](#_bookmark1530)

#### Using Oracle Virtual Private Database Policies with Editions

If you are preparing an application for edition-based redefinition, and you cover each table that the application uses with an editioning view, then you must move the Virtual Private Database polices that protect these tables to the editioning view.

When an editioned object has a Virtual Private Database t policy, then it applies in all editions in which the object is visible. When an editioned object is actualized, any VPD policies that are attached to it are newly attached to the new actual occurrence. When you newly apply a VPD policy to an inherited editioned object, this action will actualize it.

**See Also:** *Oracle Database Advanced Application Developer's Guide* for detailed information about editions

#### Using SELECT FOR UPDATE in User Queries on VPD-Protected Tables

As a general rule, users should not include the FOR UPDATE clause when querying Virtual Private Database-protected tables. The Virtual Private Database technology depends on rewriting the user’s query against an inline view that includes the VPD predicate generated by the VPD policy function. Because of this, the same limitations on views also apply to VPD-protected tables. If a user’s query against a VPD-protected table includes the FOR UPDATE clause in a SELECT statement, in most cases, the query may not work. However, the user’s query may work in some situations if the inline view generated by VPD is sufficiently simple.

**See Also:** *Oracle Database SQL Language Reference* for more information about the restrictions of the FOR UPDATE clause in the SELECT statement

#### How Oracle Virtual Private Database Policies Affect Outer or ANSI Join Operations

Oracle Virtual Private Database rewrites SQL by using dynamic views. For SQL that contains outer join or ANSI operations, some views may not merge and some indexes may not be used. This problem is a known optimization limitation. To remedy this problem, rewrite the SQL to not use outer joins or ANSI operations.

#### How Oracle Virtual Private Database Security Policies Work with Applications

An Oracle Virtual Private Database security policy is applied within the database itself, rather than within an application. Hence, a user trying to access data by using a different application cannot bypass the Oracle Virtual Private Database security policy. Another advantage of creating the security policy in the database is that you maintain it in one central place, rather than maintaining individual security policies in multiple applications. Oracle Virtual Private Database provides stronger security than application-based security, at a lower cost of ownership.

You may want to enforce different security policies depending on the application that is accessing data. Consider a situation in which two applications, Order Entry and Inventory, both access the orders table. You may want to have the Inventory application use a policy that limits access based on type of product. At the same time, you may want to have the Order Entry application use a policy that limits access based on customer number.

In this case, you must partition the use of fine-grained access by application. Otherwise, both policies would be automatically concatenated together, which may not be the result that you want. You can specify two or more policy groups, and a driving application context that determines which policy group is in effect for a given transaction. You can also designate default policies that always apply to data access. In a hosted application, for example, data access should be limited by subscriber ID. See ["Tutorial: Implementing an Oracle Virtual Private Database Policy Group" on](#_bookmark1488)

[page 7-28](#_bookmark1488) for an example of how you can create policy groups that use an application context to determine which group should be used.

#### Using Automatic Reparsing for Fine-Grained Access Control Policy Functions

By default, queries against objects enabled with fine-grained access control run the policy function to ensure that the most current predicate is used for each policy. For example, in the case of a time-based policy function, in which queries are only allowed between 8:00 a.m. and 5:00 p.m., a cursor execution parsed at noon runs the policy function at that time, ensuring that the policy is consulted again for the query. Even if the curser was parsed at 9 a.m., when it runs later on (for example, at noon), then the Virtual Private Database policy function runs again to ensure that the execution of the cursor is still permitted at the current time (noon). This ensures that the security check it must perform is the most recent.

Automatic re-execution of the Virtual Private Database policy function does not occur when you set the DBMS\_RLS.ADD\_POLICY setting STATIC\_POLICY to TRUE while adding the policy. This setting causes the policy function to return the same predicate.

#### Using Oracle Virtual Private Database Policies and Flashback Query

By default, operations on the database use the most recently committed data available. The flashback query feature enables you to query the database at some point in the past. To write an application that uses flashback query, you can use the AS OF clause in SQL queries to specify either a time or a system change number (SCN), and then query against the committed data from the specified time. You can also use the DBMS\_ FLASHBACK PL/SQL package, which requires more code, but enables you to perform multiple operations, all of which refer to the same point in time.

However, if you use flashback query against a database object that is protected with Oracle Virtual Private Database policies, then the current policies are applied to the old data. Applying the current Oracle Virtual Private Database policies to flashback query data is more secure because it reflects the most current business policy.

**See Also:**

* + - *Oracle Database Advanced Application Developer's Guide* for more information about the flashback query feature and how to write applications that use it
    - *Oracle Database PL/SQL Packages and Types Reference* for more information about the DBMS\_FLASHBACK PL/SQL package

#### Using Oracle Virtual Private Database and Oracle Label Security

This section contains:

* + [Using Oracle Virtual Private Database to Enforce Oracle Label Security Policies](#_bookmark1517)
  + [Oracle Virtual Private Database and Oracle Label Security Exceptions](#_bookmark1520)

**See Also:** *Oracle Label Security Administrator's Guide*

###### Using Oracle Virtual Private Database to Enforce Oracle Label Security Policies

You can use Oracle Virtual Private Database policies to provide column or row-level access control based on Oracle Label Security user authorizations. In general, you need to perform the following steps:

1. When you create the Oracle Label Security policy, do not apply the policy to the table that you want to protect. (The Virtual Private Database policy that you create handles this for you.) In the SA\_SYSDBA.CREATE\_POLICY procedure, set the default\_options parameter to NO\_CONTROL.
2. Create the Oracle Label Security label components and authorize users as you normally would.
3. When you create the Oracle Virtual Private Database policy, do the following:
   * In the PL/SQL function that you create for the policy, use the Oracle Label Security DOMINATES function to compare the authorization of the user with the label that you created in Step [2](#_bookmark1519). (See *Oracle Label Security Administrator's Guide* for more information about the dominance functions.) The DOMINATES function determines if the user authorization is equal to, or if it is more sensitive than, the label used in the comparison. If the user authorization passes, then the user is granted access to the column. Otherwise, the user is denied access.
   * In the Virtual Private Database policy definition, apply this function to the table that you want to protect. In the DBMS\_RLS.ADD\_POLICY procedure, use the sensitive column (SEC\_RELEVANT\_COLS parameter) and column masking (SEC\_ RELEVANT\_COLS\_OPT parameter) functionality to show or hide columns based on Oracle Label Security user authorizations.

For an example of how to accomplish this, visit the following Oracle Technology Network site:

[http](http://www.oracle.com/technetwork/database/focus-areas/security/ols-cs1-099558.html)[://www.oracle.com/technetwork/database/focus-areas/security/ols-cs1](http://www.oracle.com/technetwork/database/focus-areas/security/ols-cs1-09)-09 9558.html

###### Oracle Virtual Private Database and Oracle Label Security Exceptions

Be aware of the following exceptions when you use Oracle Virtual Private Database and Oracle Label Security:

* + When you are exporting data, Oracle Virtual Private Database and Oracle Label Security policies are not enforced during a direct path export operation. In a

direct path export operation, Oracle Database reads data from disk into the buffer cache and transfers rows directly to the Export client. See *Oracle Database Utilities*for more information about direct path export operations.

* + - **You cannot apply Oracle Virtual Private Database policies and Oracle Label Security policies to objects in the SYS schema.** The SYS user and users making a DBA-privileged connection to the database (for example, CONNECT/AS SYSDBA) do not have Oracle Virtual Private Database or Oracle Label Security policies applied to their actions. The database user SYS is thus always exempt from Oracle Virtual Private Database or Oracle Label Security enforcement, regardless of the export mode, application, or utility used to extract data from the database.

However, you can audit SYSDBA actions by enabling auditing upon installation and specifying that this audit trail be stored in a secure location in the operating system. See ["Auditing SYS Administrative Users" on page 9-53](#_bookmark1969) for more information. You can also closely monitor the SYS user by using Oracle Database Vault.

* + - **Database users who were granted the EXEMPT ACCESS POLICY privilege, either directly or through a database role, are exempt from Oracle Virtual Private Database enforcements.** The system privilege EXEMPT ACCESS POLICY allows a user to be exempted from all fine-grained access control policies on any SELECT or DML operation (INSERT, UPDATE, and DELETE). This provides ease of use for administrative activities, such as installation and import and export of the database, through a non-SYS schema.

However, the following policy enforcement options remain in effect even when

EXEMPT ACCESS POLICY is granted:

* + - * INSERT\_CONTROL, UPDATE\_CONTROL, DELETE\_CONTROL, WRITE\_CONTROL, LABEL\_ UPDATE, and LABEL\_DEFAULT
      * If the Oracle Label Security policy specifies the ALL\_CONTROL option, then all enforcement controls are applied except READ\_CONTROL and CHECK\_CONTROL.

Because EXEMPT ACCESS POLICY negates the effect of fine-grained access control, you should only grant this privilege to users who have legitimate reasons for bypassing fine-grained access control enforcement. Do not grant this privilege using the WITH ADMIN OPTION. If you do, users could pass the EXEMPT ACCESS POLICY privilege to other users, and thus propagate the ability to bypass

fine-grained access control.

**Note:**

* + - * + The EXEMPT ACCESS POLICY privilege does not affect the enforcement of object privileges such as SELECT, INSERT, UPDATE, and DELETE. These privileges are enforced even if a user was granted the EXEMPT ACCESS POLICY privilege.
        + The SYS\_CONTEXT values that Oracle Virtual Private Database uses are not propagated to secondary databases for failover.

#### Exporting Data Using the EXPDP Utility access\_method Parameter

If you try to use the Oracle Data Pump Export (EXPDP) utility with the access\_method parameter set to direct\_path to export data from a schema which contains an object that has a Virtual Private Database policy defined on it, then the following error message may appear and the export operation will fail:

ORA-31696: unable to export/import TABLE\_DATA:"*schema.table*" using client specified DIRECT\_PATH method

This problem only occurs when you perform a schema-level export as a user who has not been granted the EXP\_FULL\_DATABASE role. It does not occur during a full database export, which requires the EXP\_FULL\_DATABASE role. The EXP\_FULL\_DATABASE role includes the EXEMPT ACCESS POLICY system privilege, which bypasses Virtual Private Database policies.

To find the underlying problem, try the EXPDP invocation again, but do not set the access\_method parameter to direct\_path. Instead, use either automatic or external\_ table. The underlying problem could be a permissions problem, for example:

ORA-39181: Only partial table data may be exported due to fine grain access control on "*schema\_name*"."*object\_name*"

**See Also:** *Oracle Database Utilities* for more information about using Data Pump Export.

#### User Models and Oracle Virtual Private Database

You can use Oracle Virtual Private Database in the following types of user models:

* + **Application users who are also database users.** Oracle Database enables applications to enforce fine-grained access control for each user, regardless of whether that user is a database user or an application user unknown to the database. When application users are also database users, Oracle Virtual Private Database enforcement works as follows: users connect to the database, and then the application sets up application contexts for each session. (You can use the default USERENV application context namespace, which provides many parameters for retrieve different types of user session data.) As each session is initiated under a different user name, it can enforce different fine-grained access control conditions for each user.
  + **Proxy authentication using OCI or JDBC/OCI.**Proxy authentication permits different fine-grained access control for each user, because each session (OCI or JDBC/OCI) is a distinct database session with its own application context.
  + **Proxy authentication integrated with Enterprise User Security.** If you have integrated proxy authentication by using Enterprise User Security, you can retrieve user roles and other attributes from Oracle Internet Directory to enforce Oracle Virtual Private Database policies. (In addition, globally initialized application context can also be retrieved from the directory.)
  + **Users connecting as One Big Application User.** Applications connecting to the database as a single user on behalf of all users can have fine-grained access control for each user. The user for that single session is often called *One Big Application User*. Within the context of that session, however, an application developer can create a global application context attribute to represent the individual application user (for example, REALUSER). Although all database sessions and audit records are created for One Big Application User, the attributes for each session can vary, depending on who the end user is. This model works best for applications with a limited number of users and no reuse of sessions. The scope of roles and database auditing is diminished because each session is created as the same database user. For more information about global application contexts, see ["Using Global Application Contexts" on page 6-22](#_bookmark1238).
  + **Web-based applications.** Web-based applications typically have hundreds of users. Even when there are persistent connections to the database, supporting data

retrieval for many user requests, these connections are not specific to particular Web-based users. Instead, Web-based applications typically set up and reuse connections, to provide scalability, rather than having different sessions for each user. For example, when Web users Jane and Ajit connect to a middle tier application, it may establish a single database session that it uses on behalf of both users. Typically, neither Jane nor Ajit is known to the database. The application is responsible for switching the user name on the connection, so that, at any given time, it is either Jane or Ajit using the session.

Oracle Virtual Private Database helps with connection pooling by allowing multiple connections to access more than one global application context. This ability makes it unnecessary to establish a separate application context for each distinct user session.

[Table 7–3](#_bookmark1542) summarizes how Oracle Virtual Private Database applies to user models.

***Table 7–3 Oracle Virtual Private Database in Different User Models***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Model Scenario** |  | **Individual Separate Database Application Context Connection per User** | **Single Database Connection** | **Application Must Switch User Name** |
| Application users are also database users |  | Yes Yes | No | No |
| Proxy authentication using OCI or JDBC/OCI |  | Yes Yes | No | No |
| Proxy authentication integrated with Enterprise User Security1  One Big Application User |  | No No  No No2 | Yes  No | Yes  Yes2 |
| Web-based applications |  | No No | Yes | Yes |
|  |  |  |  |  |

1 User roles and other attributes, including globally initialized application context, can be retrieved from Oracle Internet Directory to enforce Oracle Virtual Private Database.

2 Application developers can create a global application context attribute representing individual application users (for example,

REALUSER), which can then be used for controlling each session attributes, or for auditing.

### Finding Information About Oracle Virtual Private Database Policies

[Table 7–4](#_bookmark1546) lists data dictionary views that you can use to find information about Oracle Virtual Private Database policies. See *Oracle Database Reference* for more information about these views.

***Table 7–4 Data Dictionary Views That Display Information about VPD Policies***

|  |  |
| --- | --- |
| **View** | **Description** |
| ALL\_POLICIES | Describes all Oracle Virtual Private Database security policies for objects accessible to the current user. |
| ALL\_POLICY\_CONTEXTS | Describes the driving contexts defined for the synonyms, tables, and views accessible to the current user. A driving context is an application context used in an Oracle Virtual Private Database policy. |
| ALL\_POLICY\_GROUPS | Describes the Oracle Virtual Private Database policy groups defined for the synonyms, tables, and views accessible to the current user |
| ALL\_SEC\_RELEVANT\_COLS | Describes the security relevant columns of the security policies for the tables and views accessible to the current user |
| DBA\_POLICIES | Describes all Oracle Virtual Private Database security policies in the database. |

***Table 7–4 (Cont.) Data Dictionary Views That Display Information about VPD Policies***

|  |  |
| --- | --- |
| **View** | **Description** |
| DBA\_POLICY\_GROUPS | Describes all policy groups in the database. |
| DBA\_POLICY\_CONTEXTS | Describes all driving contexts in the database. Its columns are the same as those in ALL\_POLICY\_CONTEXTS. |
| DBA\_SEC\_RELEVANT\_COLS | Describes the security relevant columns of all security policies in the database |
| USER\_POLICIES | Describes all Oracle Virtual Private Database security policies associated with objects owned by the current user. This view does not display the OBJECT\_OWNER column. |
| USER\_POLICY\_CONTEXTS | Describes the driving contexts defined for the synonyms, tables, and views owned by the current user. Its columns (except for OBJECT\_OWNER) are the same as those in ALL\_POLICY\_CONTEXTS. |
| USER\_SEC\_RELEVANT\_COLS | Describes the security relevant columns of the security policies for the tables and views owned by the current user. Its columns (except for OBJECT\_OWNER) are the same as those in ALL\_SEC\_RELEVANT\_COLS. |
| USER\_POLICY\_GROUPS | Describes the policy groups defined for the synonyms, tables, and views owned by the current user. This view does not display the OBJECT\_OWNER column. |
| V$VPD\_POLICY | Displays all the fine-grained security policies and predicates associated with the cursors currently in the library cache. This view is useful for finding the policies that were applied to a SQL statement. |

**Tip:** In addition to these views, check the database trace file if you find errors in application that use Virtual Private Database policies. See *Oracle Database Performance Tuning Guide* for more information about trace files. The USER\_DUMP\_DEST initialization parameter specifies the current location of the trace files. You can find the value of this parameter by issuing SHOW PARAMETER USER\_DUMP\_DEST in SQL\*Plus.

# 8

## Developing Applications Using the Data Encryption API

This chapter contains:

* + - [Security Problems That Encryption Does Not Solve](#_bookmark1549)
    - [Data Encryption Challenges](#_bookmark1560)
    - [Storing Data Encryption by Using the DBMS\_CRYPTO Package](#_bookmark1583)
    - [Examples of Using the Data Encryption API](#_bookmark1591)
    - [Finding Information About Encrypted Data](#_bookmark1598)

**See Also:**

* *Oracle Database 2 Day + Security Guide* for an introduction to network encryption
* *Oracle Database Advanced Security Administrator's Guide* for information about using transparent data encryption and tablespace encryption

### Security Problems That Encryption Does Not Solve

While there are many good reasons to encrypt data, there are many reasons not to encrypt data. Encryption does not solve all security problems, and may make some problems worse. The following sections describe some misconceptions about encryption of stored data:

* + - [Principle 1: Encryption Does Not Solve Access Control Problems](#_bookmark1552)
    - [Principle 2: Encryption Does Not Protect Against a Malicious Database Administrator](#_bookmark1554)
    - [Principle 3: Encrypting Everything Does Not Make Data Secure](#_bookmark1558)

#### Principle 1: Encryption Does Not Solve Access Control Problems

Most organizations need to limit data access to users who need to see this data. For example, a human resources system may limit employees to viewing only their own employment records, while allowing managers of employees to see the employment records of subordinates. Human resource specialists may also need to see employee records for multiple employees.

Typically, you can use access control mechanisms to address security policies that limit data access to those with a need to see it. Oracle Database has provided strong,

independently evaluated access control mechanisms for many years. It enables access control enforcement to a fine level of granularity through Virtual Private Database.

Because human resource records are considered sensitive information, it is tempting to think that all information should be encrypted for better security. However, encryption cannot enforce granular access control, and it may hinder data access. For example, an employee, his manager, and a human resources clerk may all need to access an employee record. If all employee data is encrypted, then all three must be able to access the data in unencrypted form. Therefore, the employee, the manager and the human resources clerk would have to share the same encryption key to decrypt the data. Encryption would, therefore, not provide any additional security in the sense of better access control, and the encryption might hinder the proper or efficient functioning of the application. An additional issue is that it is difficult to securely transmit and share encryption keys among multiple users of a system.

A basic principle behind encrypting stored data is that it must not interfere with access control. For example, a user who has the SELECT privilege on emp should not be limited by the encryption mechanism from seeing all the data he is otherwise allowed to see. Similarly, there is little benefit to encrypting part of a table with one key and part of a table with another key if users need to see all encrypted data in the table. In this case, encryption adds to the overhead of decrypting the data before users can read it. If access controls are implemented well, then encryption adds little additional security within the database itself. A user who has privileges to access data within the database has no more nor any less privileges as a result of encryption. Therefore, you should never use encryption to solve access control problems.

#### Principle 2: Encryption Does Not Protect Against a Malicious Database Administrator

Some organizations, concerned that a malicious user might gain elevated (database administrator) privileges by guessing a password, like the idea of encrypting stored data to protect against this threat. However, the correct solution to this problem is to protect the database administrator account, and to change default passwords for other privileged accounts. The easiest way to break into a database is by using a default password for a privileged account that an administrator allowed to remain unchanged. One example is SYS/CHANGE\_ON\_INSTALL.

While there are many destructive things a malicious user can do to a database after gaining the DBA privilege, encryption will not protect against many of them. Examples include corrupting or deleting data, exporting user data to the file system to email the data back to himself to run a password cracker on it, and so on.

Some organizations are concerned that database administrators, typically having all privileges, are able to see all data in the database. These organizations feel that the database administrators should administer the database, but should not be able to see the data that the database contains. Some organizations are also concerned about concentrating so much privilege in one person, and would prefer to partition the DBA function, or enforce two-person access rules.

It is tempting to think that encrypting all data (or significant amounts of data) will solve these problems, but there are better ways to protect against these threats. For example, Oracle Database supports limited partitioning of DBA privileges. Oracle Database provides native support for SYSDBA and SYSOPER users. SYSDBA has all privileges, but SYSOPER has a limited privilege set (such as startup and shutdown of the database).

Furthermore, you can create smaller roles encompassing several system privileges. A jr\_dba role might not include all system privileges, but only those appropriate to a junior database administrator (such as CREATE TABLE, CREATE USER, and so on).

Oracle Database also enables auditing the actions taken by SYS (or SYS-privileged users) and storing that audit trail in a secure operating system location. Using this model, a separate auditor who has root privileges on the operating system can audit all actions by SYS, enabling the auditor to hold all database administrators accountable for their actions.

See ["Auditing SYS Administrative Users" on page 9-53](#_bookmark1969) for information about ways to audit database administrators.

You can also fine-tune the access and control that database administrators have by using Oracle Database Vault. See *Oracle Database Vault Administrator's Guide* for more information.

The database administrator function is a trusted position. Even organizations with the most sensitive data, such as intelligence agencies, do not typically partition the database administrator function. Instead, they manage their database administrators strongly, because it is a position of trust. Periodic auditing can help to uncover inappropriate activities.

Encryption of stored data must not interfere with the administration of the database, because otherwise, larger security issues can result. For example, if by encrypting data you corrupt the data, then you create a security problem, the data itself cannot be interpreted, and it may not be recoverable.

You can use encryption to limit the ability of a database administrator or other privileged user to see data in the database. However, it is not a substitute for managing the database administrator privileges properly, or for controlling the use of powerful system privileges. If untrustworthy users have significant privileges, then they can pose multiple threats to an organization, some of them far more significant than viewing unencrypted credit card numbers.

#### Principle 3: Encrypting Everything Does Not Make Data Secure

A common error is to think that if encrypting some data strengthens security, then encrypting everything makes all data secure.

As the discussion of the previous two principles illustrates, encryption does not address access control issues well, and it is important that encryption not interfere with normal access controls. Furthermore, encrypting an entire production database means that all data must be decrypted to be read, updated, or deleted. Encryption is inherently a performance-intensive operation; encrypting all data will significantly affect performance.

Availability is a key aspect of security. If encrypting data makes data unavailable, or adversely affects availability by reducing performance, then encrypting everything will create a new security problem. Availability is also adversely affected by the database being inaccessible when encryption keys are changed, as good security practices require on a regular basis. When the keys are to be changed, the database is inaccessible while data is decrypted and reencrypted with a new key or keys.

There may be advantages to encrypting data stored off-line. For example, an organization may store backups for a period of 6 months to a year off-line, in a remote location. Of course, the first line of protection is to secure the facility storing the data, by establishing physical access controls. Encrypting this data before it is stored may provide additional benefits. Because it is not being accessed on-line, performance need not be a consideration. While an Oracle database does not provide this capability, there are vendors who provide encryption services. Before embarking on large-scale encryption of backup data, organizations considering this approach should thoroughly

test the process. It is essential to verify that data encrypted before off-line storage can be decrypted and re-imported successfully.

### Data Encryption Challenges

In cases where encryption can provide additional security, there are some associated technical challenges, as described in the following sections:

* + [Encrypting Indexed Data](#_bookmark1562)
  + [Generating Encryption Keys](#_bookmark1564)
  + [Transmitting Encryption Keys](#_bookmark1566)
  + [Storing Encryption Keys](#_bookmark1568)
  + [Changing Encryption Keys](#_bookmark1578)
  + [Encrypting Binary Large Objects](#_bookmark1580)

#### Encrypting Indexed Data

Special difficulties arise when encrypted data is indexed. For example, suppose a company uses a national identity number, such as the U.S. Social Security number (SSN), as the employee number for its employees. The company considers employee numbers to be sensitive data, and, therefore, wants to encrypt data in the employee\_ number column of the employees table. Because employee\_number contains unique values, the database designers want to have an index on it for better performance.

However, if DBMS\_CRYPTO or the DBMS\_OBFUSCATION\_TOOLKIT (or another mechanism) is used to encrypt data in a column, then an index on that column will also contain encrypted values. Although an index can be used for equality checking (for example, SELECT \* FROM emp WHERE employee\_number = '987654321'), if the index on that column contains encrypted values, then the index is essentially unusable for any other purpose. You should not encrypt indexed data.

Oracle recommends that you do not use national identity numbers as unique IDs. Instead, use the CREATE SEQUENCE statement to generate unique identity numbers. Reasons to avoid using national identity numbers are as follows:

* + There are privacy issues associated with overuse of national identity numbers (for example, identity theft).
  + Sometimes national identity numbers can have duplicates, as with U.S. Social Security numbers.

#### Generating Encryption Keys

Encrypted data is only as secure as the key used for encrypting it. An encryption key must be securely generated using secure cryptographic key generation. Oracle Database provides support for secure random number generation, with the RANDOMBYTES function of DBMS\_CRYPTO. (This function replaces the capabilities provided by the GetKey procedure of the earlier DBMS\_OBFUSCATION\_TOOLKIT.) DBMS\_ CRYPTO calls the secure random number generator (RNG) previously certified by RSA Security.

**Note:** Do not use the DBMS\_RANDOM package. The DBMS\_RANDOM package generates pseudo-random numbers, which, as Randomness Recommendations for Security (RFC-1750) states that using

pseudo-random processes to generate secret quantities can result in pseudo-security.

Be sure to provide the correct number of bytes when you encrypt a key value. For example, you must provide a 16-byte key for the ENCRYPT\_AES128 encryption algorithm.

#### Transmitting Encryption Keys

If the encryption key is to be passed by the application to the database, then you must encrypt it. Otherwise, an intruder could get access to the key as it is being transmitted. Network encryption, such as that provided by Oracle Advanced Security, protects all data in transit from modification or interception, including cryptographic keys.

#### Storing Encryption Keys

Storing encryption keys is one of the most important, yet difficult, aspects of encryption. To recover data encrypted with a symmetric key, the key must be accessible to an authorized application or user seeking to decrypt the data. At the same time, the key must be inaccessible to someone who is maliciously trying to access encrypted data that he is not supposed to see.

The options available to a developer are:

* + - [Storing the Encryption Keys in the Database](#_bookmark1570)
    - [Storing the Encryption Keys in the Operating System](#_bookmark1571)
    - [Users Managing Their Own Encryption Keys](#_bookmark1573)
    - [Using Transparent Database Encryption and Tablespace Encryption](#_bookmark1575)

###### Storing the Encryption Keys in the Database

Storing the keys in the database cannot always provide infallible security if you are trying to protect against the database administrator accessing encrypted data. An all-privileged database administrator could still access tables containing encryption keys. However, it can often provide good security against the casual curious user or against someone compromising the database file on the operating system.

As a trivial example, suppose you create a table (EMP) that contains employee data. You want to encrypt the employee Social Security number (SSN) stored in one of the columns. You could encrypt employee SSN using a key that is stored in a separate column. However, anyone with SELECT access on the entire table could retrieve the encryption key and decrypt the matching SSN.

While this encryption scheme seems easily defeated, with a little more effort you can create a solution that is much harder to break. For example, you could encrypt the SSN using a technique that performs some additional data transformation on the employee\_number before using it to encrypt the SSN. This technique might be as simple as using an XOR operation on the employee\_number and the birth date of the employee to determine the validity of the values.

As additional protection, PL/SQL source code performing encryption can be wrapped, (using the WRAP utility) which obfuscates (scrambles) the code. The WRAP utility

processes an input SQL file and obfuscates the PL/SQL units in it. For example, the following command uses the keymanage.sql file as the input:

wrap iname=/mydir/keymanage.sql

A developer can subsequently have a function in the package call the DBMS\_ OBFUSCATION\_TOOLKIT with the key contained in the wrapped package.

Oracle Database enables you to obfuscate dynamically generated PL/SQL code. The DBMS\_DDL package contains two subprograms that allow you to obfuscate dynamically generated PL/SQL program units. For example, the following block uses the DBMS\_ DDL.CREATE\_WRAPPED procedure to wrap dynamically generated PL/SQL code.

BEGIN

......

SYS.DBMS\_DDL.CREATE\_WRAPPED(*function\_returning\_PLSQL\_code*());

...... END;

While wrapping is not unbreakable, it makes it harder for an intruder to get access to the encryption key. Even in cases where a different key is supplied for each encrypted data value, you should not embed the key value within a package. Instead, wrap the package that performs the key management (that is, data transformation or padding).

**See Also:** *Oracle Database PL/SQL Language Reference* for additional information about the WRAP command line utility and the DBMS\_DDL subprograms for dynamic wrapping

An alternative to wrapping the data is to have a separate table in which to store the encryption key and to envelope the call to the keys table with a procedure. The key table can be joined to the data table using a primary key to foreign key relationship. For example, employee\_number is the primary key in the employees table that stores employee information and the encrypted SSN. The employee\_number column is a foreign key to the ssn\_keys table that stores the encryption keys for the employee SSN. The key stored in the ssn\_keys table can also be transformed before use (by using an XOR operation), so the key itself is not stored unencrypted. If you wrap the procedure, then that can hide the way in which the keys are transformed before use.

The strengths of this approach are:

* + Users who have direct table access cannot see the sensitive data unencrypted, nor can they retrieve the keys to decrypt the data.
  + Access to decrypted data can be controlled through a procedure that selects the encrypted data, retrieves the decryption key from the key table, and transforms it before it can be used to decrypt the data.
  + The data transformation algorithm is hidden from casual snooping by wrapping the procedure, which obfuscates the procedure code.
  + SELECT access to both the data table and the keys table does not guarantee that the user with this access can decrypt the data, because the key is transformed before use.

The weakness to this approach is that a user who has SELECT access to both the key table and the data table, and who can derive the key transformation algorithm, can break the encryption scheme.

The preceding approach is not infallible, but it is adequate to protect against easy retrieval of sensitive information stored in clear text.

###### Storing the Encryption Keys in the Operating System

Storing keys in a flat file in the operating system is another option. Oracle Database enables you to make callouts from PL/SQL, which you could use to retrieve encryption keys. However, if you store keys in the operating system and make callouts to it, then your data is only as secure as the protection on the operating system. If your primary security concern is that the database can be broken into from the operating system, then storing the keys in the operating system makes it easier for an intruder to retrieve encrypted data than storing the keys in the database itself.

###### Users Managing Their Own Encryption Keys

Having the user supply the key assumes the user will be responsible with the key. Considering that 40 percent of help desk calls are from users who have forgotten their passwords, you can see the risks of having users manage encryption keys. In all likelihood, users will either forget an encryption key, or write the key down, which then creates a security weakness. If a user forgets an encryption key or leaves the company, then your data is not recoverable.

If you do decide to have user-supplied or user-managed keys, then you need to ensure you are using network encryption so that the key is not passed from the client to the server in the clear. You also must develop key archive mechanisms, which is also a difficult security problem. Key archives and backdoors create the security weaknesses that encryption is attempting to solve.

###### Using Transparent Database Encryption and Tablespace Encryption

Transparent database encryption and tablespace encryption provide secure encryption with automatic key management for the encrypted tables and tablespaces. If the application requires protection of sensitive column data stored on the media, then these two types of encryption are a simple and fast way of achieving this.

**See Also:** *Oracle Database Advanced Security Administrator's Guide* for more information about transparent data encryption

#### Changing Encryption Keys

Prudent security practice dictates that you periodically change encryption keys. For stored data, this requires periodically unencrypting the data, and reencrypting it with another well-chosen key. You would most likely change the encryption key while the data is not being accessed, which creates another challenge. This is especially true for a Web-based application encrypting credit card numbers, because you do not want to shut down the entire application while you switch encryption keys.

#### Encrypting Binary Large Objects

Certain data types require more work to encrypt. For example, Oracle Database supports storage of binary large objects (BLOBs), which stores very large objects (for example, multiple gigabytes) in the database. A BLOB can be either stored internally as a column, or stored in an external file.

For an example of using DBMS\_CRYPTO on BLOB data, see ["Example of Encryption and](#_bookmark1596) [Decryption Procedures for BLOB Data" on page 8-12](#_bookmark1596).

### Storing Data Encryption by Using the DBMS\_CRYPTO Package

The DBMS\_CRYPTO package provides several ways to address the security issues that were discussed. (For backward compatibility, DBMS\_OBFUSCATION\_TOOLKIT is also provided.)

While encryption is not the ideal solution for addressing several security threats, it is clear that selectively encrypting sensitive data before storage in the database does improve security. Examples of such data could include:

* + Credit card numbers
  + National identity numbers

Oracle Database provides the PL/SQL package DBMS\_CRYPTO to encrypt and decrypt stored data. This package supports several industry-standard encryption and hashing algorithms, including the Advanced Encryption Standard (AES) encryption algorithm. AES was approved by the National Institute of Standards and Technology (NIST) to replace the Data Encryption Standard (DES).

The DBMS\_CRYPTO package enables encryption and decryption for common Oracle Database data types, including RAW and large objects (LOBs), such as images and sound. Specifically, it supports BLOBs and CLOBs. In addition, it provides Globalization Support for encrypting data across different database character sets.

The following cryptographic algorithms are supported:

* + Data Encryption Standard (DES), Triple DES (3DES, 2-key)
  + Advanced Encryption Standard (AES)
  + SHA-1 Cryptographic Hash
  + SHA-1 Message Authentication Code (MAC)

Block cipher modifiers are also provided with DBMS\_CRYPTO. You can choose from several padding options, including Public Key Cryptographic Standard (PKCS) #5, and from four block cipher chaining modes, including Cipher Block Chaining (CBC). Padding must be done in multiples of eight bytes.

**Note:**

* + - DES is no longer recommended by the National Institute of Standards and Technology (NIST).
    - Usage of SHA-1 is more secure than MD5.
    - Keyed MD5 is not vulnerable.

[Table 8–1](#_bookmark1589) compares the DBMS\_CRYPTO package features to the other PL/SQL encryption package, the DBMS\_OBFUSCATION\_TOOLKIT.

***Table 8–1 DBMS\_CRYPTO and DBMS\_OBFUSCATION\_TOOLKIT Feature Comparison***

|  |  |  |
| --- | --- | --- |
| **Package Feature** | **DBMS\_CRYPTO** | **DBMS\_OBFUSCATION\_TOOLKIT** |
| Cryptographic algorithms | DES, 3DES, AES, RC4, 3DES\_2KEY | DES, 3DES |
| Padding forms | PKCS5, zeroes | None supported |
| Block cipher chaining modes | CBC, CFB, ECB, OFB | CBC |
| Cryptographic hash algorithms | SHA-1, MD4, MD5 | MD5 |

***Table 8–1 (Cont.) DBMS\_CRYPTO and DBMS\_OBFUSCATION\_TOOLKIT Feature Comparison***

|  |  |  |
| --- | --- | --- |
| **Package Feature** | **DBMS\_CRYPTO** | **DBMS\_OBFUSCATION\_TOOLKIT** |
| Keyed hash (MAC) algorithms | HMAC\_MD5, HMAC\_SH1 | None supported |
| Cryptographic pseudo-random number generator | RAW, NUMBER, BINARY\_INTEGER | RAW, VARCHAR2 |
| Database types | RAW, CLOB, BLOB | RAW, VARCHAR2 |

DBMS\_CRYPTO is intended to replace the OBFUSCATION\_TOOLKIT package, because it is easier to use and supports a range of algorithms that accommodate both new and existing systems. Although 3DES\_2KEY and MD4 are provided for backward compatibility, you achieve better security using 3DES, AES, or SHA-1. Therefore, 3DES\_2KEY is not recommended.

The DBMS\_CRYPTO package includes cryptographic checksum capabilities (MD5), which are useful for comparisons, and the ability to generate a secure random number (the RANDOMBYTES function). Secure random number generation is an important part of cryptography; predictable keys are easily guessed keys; and easily guessed keys may lead to easy decryption of data. Most cryptanalysis is done by finding weak keys or poorly stored keys, rather than through brute force analysis (cycling through all possible keys).

**Note:** Do not use DBMS\_RANDOM, because it is unsuitable for cryptographic key generation.

Key management is programmatic. That is, the application (or caller of the function) must supply the encryption key. This means that the application developer must find a way of storing and retrieving keys securely. The relative strengths and weaknesses of various key management techniques are discussed in the sections that follow. The DBMS\_OBFUSCATION\_TOOLKIT package, which can handle both string and raw data, requires the submission of a 64-bit key. The DES algorithm itself has an effective key length of 56-bits.

**Note:** The DBMS\_OBFUSCATION\_TOOLKIT is granted to PUBLIC by default. Oracle recommends that you revoke this grant.

While the DBMS\_OBFUSCATION\_TOOLKIT package can take either VARCHAR2 or RAW data types, it is preferable to use the RAW data type for keys and encrypted data. Storing encrypted data as VARCHAR2 can cause problems if it passes through Globalization Support routines. For example, when transferring a database to another database that uses another character set.

To convert between VARCHAR2 and RAW data types, use the CAST\_TO\_ RAW and CAST\_TO\_VARCHAR2 functions of the UTL\_RAW package.

**See Also:**

* + - *Oracle Database PL/SQL Packages and Types Reference* for detailed information about the DBMS\_CRYPTO package
    - *Oracle Database PL/SQL Packages and Types Reference* for detailed information about the OBFUSCATION\_TOOLKIT package
    - *Oracle Database PL/SQL Packages and Types Reference* for detailed information about the UTL\_RAW package

### Examples of Using the Data Encryption API

This section contains:

* + [Example of a Data Encryption Procedure](#_bookmark1592)
  + [Example of AES 256-Bit Data Encryption and Decryption Procedures](#_bookmark1594)
  + [Example of Encryption and Decryption Procedures for BLOB Data](#_bookmark1596)

#### Example of a Data Encryption Procedure

The following sample PL/SQL program (dbms\_crypto.sql) shows how to encrypt data. This example code performs the following actions:

* + Encrypts a string (VARCHAR2 type) using DES after first converting it into the RAW

data type.

This step is necessary because encrypt and decrypt functions and procedures in DBMS\_CRYPTO package work on the RAW data type only, unlike functions and packages in the DBMS\_OBFUSCATION\_TOOLKIT package.

* + Shows how to create a 160-bit hash using SHA-1 algorithm.
  + Demonstrates how MAC, a key-dependent one-way hash, can be computed using the MD5 algorithm.

The dbms\_crypto.sql procedure follows:

DECLARE

input\_string VARCHAR2(16) := 'tigertigertigert'; raw\_input RAW(128) :=

UTL\_RAW.CAST\_TO\_RAW(CONVERT(input\_string,'AL32UTF8','US7ASCII'));

key\_string VARCHAR2(8) := 'scottsco'; raw\_key RAW(128) :=

UTL\_RAW.CAST\_TO\_RAW(CONVERT(key\_string,'AL32UTF8','US7ASCII'));

encrypted\_raw RAW(2048); encrypted\_string VARCHAR2(2048); decrypted\_raw RAW(2048); decrypted\_string VARCHAR2(2048);

**-- Begin testing Encryption:**

BEGIN

dbms\_output.put\_line('> Input String : ' || CONVERT(UTL\_RAW.CAST\_TO\_VARCHAR2(raw\_input),'US7ASCII','AL32UTF8'));

dbms\_output.put\_line('> ========= BEGIN TEST Encrypt ========='); encrypted\_raw := dbms\_crypto.Encrypt(

src => raw\_input,

typ => DBMS\_CRYPTO.DES\_CBC\_PKCS5,

key => raw\_key);

dbms\_output.put\_line('> Encrypted hex value : ' || rawtohex(UTL\_RAW.CAST\_TO\_RAW(encrypted\_raw)));

decrypted\_raw := dbms\_crypto.Decrypt(

src => encrypted\_raw,

typ => DBMS\_CRYPTO.DES\_CBC\_PKCS5,

key => raw\_key); decrypted\_string :=

CONVERT(UTL\_RAW.CAST\_TO\_VARCHAR2(decrypted\_raw),'US7ASCII','AL32UTF8');

dbms\_output.put\_line('> Decrypted string output : ' || decrypted\_string);

if input\_string = decrypted\_string THEN

dbms\_output.put\_line('> String DES Encyption and Decryption successful'); END if;

dbms\_output.put\_line('');

dbms\_output.put\_line('> ========= BEGIN TEST Hash ========='); encrypted\_raw := dbms\_crypto.Hash(

src => raw\_input,

typ => DBMS\_CRYPTO.HASH\_SH1);

dbms\_output.put\_line('> Hash value of input string : ' || rawtohex(UTL\_RAW.CAST\_TO\_RAW(encrypted\_raw)));

dbms\_output.put\_line('> ========= BEGIN TEST Mac ========='); encrypted\_raw := dbms\_crypto.Mac(

src => raw\_input,

typ => DBMS\_CRYPTO.HMAC\_MD5,

key => raw\_key);

dbms\_output.put\_line('> Message Authentication Code : ' || rawtohex(UTL\_RAW.CAST\_TO\_RAW(encrypted\_raw)));

dbms\_output.put\_line('');

dbms\_output.put\_line('> End of DBMS\_CRYPTO tests '); END;

/

#### Example of AES 256-Bit Data Encryption and Decryption Procedures

The following PL/SQL block shows how to encrypt and decrypt a predefined variable named input\_string using the AES 256-bit algorithm with Cipher Block Chaining and PKCS #5 padding.

declare

input\_string VARCHAR2 (200) := 'Secret Message'; output\_string VARCHAR2 (200);

encrypted\_raw RAW (2000); *-- stores encrypted binary text* decrypted\_raw RAW (2000); *-- stores decrypted binary text* num\_key\_bytes NUMBER := 256/8; *-- key length 256 bits (32 bytes)* key\_bytes\_raw RAW (32); *-- stores 256-bit encryption key* encryption\_type PLS\_INTEGER := *-- total encryption type*

DBMS\_CRYPTO.ENCRYPT\_AES256

+ DBMS\_CRYPTO.CHAIN\_CBC

+ DBMS\_CRYPTO.PAD\_PKCS5;

begin

DBMS\_OUTPUT.PUT\_LINE ('Original string: ' || input\_string); key\_bytes\_raw := DBMS\_CRYPTO.RANDOMBYTES (num\_key\_bytes); encrypted\_raw := DBMS\_CRYPTO.ENCRYPT

(

src => UTL\_I18N.STRING\_TO\_RAW (input\_string, 'AL32UTF8'), typ => encryption\_type,

key => key\_bytes\_raw

);

**-- The encrypted value in the encrypted\_raw variable can be used here:**

decrypted\_raw := DBMS\_CRYPTO.DECRYPT (

src => encrypted\_raw, typ => encryption\_type,

key => key\_bytes\_raw

);

output\_string := UTL\_I18N.RAW\_TO\_CHAR (decrypted\_raw, 'AL32UTF8'); DBMS\_OUTPUT.PUT\_LINE ('Decrypted string: ' || output\_string);

end;

#### Example of Encryption and Decryption Procedures for BLOB Data

The following sample PL/SQL program (blob\_test.sql) shows how to encrypt and decrypt BLOB data. This example code does the following, and prints out its progress (or problems) at each step:

* + Creates a table for the BLOB column
  + Inserts the raw values into that table
  + Encrypts the raw data
  + Decrypts the encrypted data

The blob\_test.sql procedure follows:

**-- 1. Create a table for BLOB column:**

create table table\_lob (id number, loc blob);

**-- 2. Insert 3 empty lobs for src/enc/dec:** insert into table\_lob values (1, EMPTY\_BLOB()); insert into table\_lob values (2, EMPTY\_BLOB()); insert into table\_lob values (3, EMPTY\_BLOB());

set echo on

set serveroutput on

declare

srcdata RAW(1000); srcblob BLOB; encrypblob BLOB; encrypraw RAW(1000);

encrawlen BINARY\_INTEGER; decrypblob BLOB;

decrypraw RAW(1000); decrawlen BINARY\_INTEGER;

leng INTEGER; begin

-- RAW input data 16 bytes

srcdata := hextoraw('6D6D6D6D6D6D6D6D6D6D6D6D6D6D6D6D');

dbms\_output.put\_line('---'); dbms\_output.put\_line('input is ' || srcdata); dbms\_output.put\_line('---');

-- select empty lob locators for src/enc/dec

select loc into srcblob from table\_lob where id = 1; select loc into encrypblob from table\_lob where id = 2; select loc into decrypblob from table\_lob where id = 3;

dbms\_output.put\_line('Created Empty LOBS'); dbms\_output.put\_line('---');

leng := DBMS\_LOB.GETLENGTH(srcblob);

IF leng IS NULL THEN

dbms\_output.put\_line('Source BLOB Len NULL ');

ELSE

dbms\_output.put\_line('Source BLOB Len ' || leng);

END IF;

leng := DBMS\_LOB.GETLENGTH(encrypblob);

IF leng IS NULL THEN

dbms\_output.put\_line('Encrypt BLOB Len NULL ');

ELSE

dbms\_output.put\_line('Encrypt BLOB Len ' || leng);

END IF;

leng := DBMS\_LOB.GETLENGTH(decrypblob);

IF leng IS NULL THEN

dbms\_output.put\_line('Decrypt BLOB Len NULL ');

ELSE

dbms\_output.put\_line('Decrypt BLOB Len ' || leng);

END IF;

**-- 3. Write source raw data into blob:** DBMS\_LOB.OPEN (srcblob, DBMS\_LOB.lob\_readwrite); DBMS\_LOB.WRITEAPPEND (srcblob, 16, srcdata); DBMS\_LOB.CLOSE (srcblob);

dbms\_output.put\_line('Source raw data written to source blob'); dbms\_output.put\_line('---');

leng := DBMS\_LOB.GETLENGTH(srcblob);

IF leng IS NULL THEN

dbms\_output.put\_line('source BLOB Len NULL ');

ELSE

dbms\_output.put\_line('Source BLOB Len ' || leng);

END IF;

/\*

* Procedure Encrypt
* Arguments: srcblob -> Source BLOB
* encrypblob -> Output BLOB for encrypted data
* DBMS\_CRYPTO.AES\_CBC\_PKCS5 -> Algo : AES
* Chaining : CBC
* Padding : PKCS5
* 256 bit key for AES passed as RAW

\* ->

hextoraw('000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F')

* IV (Initialization Vector) for AES algo passed as RAW

\* -> hextoraw('00000000000000000000000000000000')

\*/

DBMS\_CRYPTO.Encrypt(encrypblob, srcblob, DBMS\_CRYPTO.AES\_CBC\_PKCS5,

hextoraw ('000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F'),

hextoraw('00000000000000000000000000000000'));

dbms\_output.put\_line('Encryption Done'); dbms\_output.put\_line('---');

leng := DBMS\_LOB.GETLENGTH(encrypblob);

IF leng IS NULL THEN

dbms\_output.put\_line('Encrypt BLOB Len NULL');

ELSE

dbms\_output.put\_line('Encrypt BLOB Len ' || leng);

END IF;

**-- 4. Read encrypblob to a raw:**

encrawlen := 999;

DBMS\_LOB.OPEN (encrypblob, DBMS\_LOB.lob\_readwrite); DBMS\_LOB.READ (encrypblob, encrawlen, 1, encrypraw); DBMS\_LOB.CLOSE (encrypblob);

dbms\_output.put\_line('Read encrypt blob to a raw'); dbms\_output.put\_line('---');

dbms\_output.put\_line('Encrypted data is (256 bit key) ' || encrypraw); dbms\_output.put\_line('---');

/\*

* Procedure Decrypt
* Arguments: encrypblob -> Encrypted BLOB to decrypt
* decrypblob -> Output BLOB for decrypted data in RAW
* DBMS\_CRYPTO.AES\_CBC\_PKCS5 -> Algo : AES
* Chaining : CBC
* Padding : PKCS5
* 256 bit key for AES passed as RAW (same as used during Encrypt)

\* ->

hextoraw('000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F')

* IV (Initialization Vector) for AES algo passed as RAW (same as used during Encrypt)

\* -> hextoraw('00000000000000000000000000000000')

\*/

DBMS\_CRYPTO.Decrypt(decrypblob,

encrypblob, DBMS\_CRYPTO.AES\_CBC\_PKCS5,

hextoraw ('000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F'),

hextoraw('00000000000000000000000000000000'));

leng := DBMS\_LOB.GETLENGTH(decrypblob);

IF leng IS NULL THEN

dbms\_output.put\_line('Decrypt BLOB Len NULL');

ELSE

dbms\_output.put\_line('Decrypt BLOB Len ' || leng);

END IF;

-- Read decrypblob to a raw decrawlen := 999;

DBMS\_LOB.OPEN (decrypblob, DBMS\_LOB.lob\_readwrite); DBMS\_LOB.READ (decrypblob, decrawlen, 1, decrypraw); DBMS\_LOB.CLOSE (decrypblob);

dbms\_output.put\_line('Decrypted data is (256 bit key) ' || decrypraw); dbms\_output.put\_line('---');

DBMS\_LOB.OPEN (srcblob, DBMS\_LOB.lob\_readwrite); DBMS\_LOB.TRIM (srcblob, 0);

DBMS\_LOB.CLOSE (srcblob);

DBMS\_LOB.OPEN (encrypblob, DBMS\_LOB.lob\_readwrite); DBMS\_LOB.TRIM (encrypblob, 0);

DBMS\_LOB.CLOSE (encrypblob);

DBMS\_LOB.OPEN (decrypblob, DBMS\_LOB.lob\_readwrite); DBMS\_LOB.TRIM (decrypblob, 0);

DBMS\_LOB.CLOSE (decrypblob);

end;

/

truncate table table\_lob; drop table table\_lob;

### Finding Information About Encrypted Data

[Table 8–2](#_bookmark1601) lists data dictionary views that you can query to access information about encrypted data. See *Oracle Database Reference* for detailed information about these views.

***Table 8–2 Data Dictionary Views That Display Information about Encrypted Data***

|  |  |
| --- | --- |
| **View** | **Description** |
| ALL\_ENCRYPTED\_COLUMNS | Describes encryption algorithm information for all encrypted columns in all tables accessible to the user |
| DBA\_ENCRYPTED\_COLUMNS | Describes encryption algorithm information for all encrypted columns in the database |
| USER\_ENCRYPTED\_COLUMNS | Describes encryption algorithm information for all encrypted columns in all tables in the schema of the user |
| V$ENCRYPTED\_TABLESPACES | Displays information about the tablespaces that are encrypted |
| V$ENCRYPTION\_WALLET | Displays information on the status of the wallet and the wallet location for transparent data encryption |
| V$RMAN\_ENCRYPTION\_ALGORITHMS | Displays supported encryption algorithms. |

# 9

## Verifying Security Access with Auditing

This chapter contains:

* + [About Auditing](#_bookmark1604)
  + [Selecting an Auditing Type](#_bookmark1631)
  + [Auditing General Activities with Standard Auditing](#_bookmark1639)
  + [Using Default Auditing for Security-Relevant SQL Statements and Privileges](#_bookmark1880)
  + [Auditing Specific Activities with Fine-Grained Auditing](#_bookmark1893)
  + [Auditing SYS Administrative Users](#_bookmark1968)
  + [Using Triggers to Write Audit Data to a Separate Table](#_bookmark1986)
  + [Managing Audit Trail Records](#_bookmark1995)
  + [Purging Audit Trail Records](#_bookmark2043)
  + [Finding Information About Audited Activities](#_bookmark2095)

**See Also:** ["Guidelines for Auditing" on page 10-18](#_bookmark2285) for general guidelines to follow for auditing your system

### About Auditing

This section contains:

* + [What Is Auditing?](#_bookmark1605)
  + [Why Is Auditing Used?](#_bookmark1608)
  + [Protecting the Database Audit Trail](#_bookmark1609)
  + [Activities That Are Always Written to the Standard and Fine-Grained Audit Records](#_bookmark1614)
  + [Activities That Are Always Audited for All Platforms](#_bookmark1619)
  + [Auditing in a Distributed Database](#_bookmark1627)
  + [Best Practices for Auditing](#_bookmark1629)

**See Also:** *Oracle Audit Vault and Database Firewall Administrator's Guide* for more information about Oracle Audit Vault and Database Firewall, which provides advanced auditing features

#### What Is Auditing?

**Auditing** is the monitoring and recording of selected user database actions, from both database users and nondatabase users1. You can base auditing on individual actions, such as the type of SQL statement executed, or on combinations of data that can include the user name, application, time, and so on. You can audit both successful and failed activities. To use auditing, you enable it, and then configure what must be audited. The actions that you audit are recorded in either data dictionary tables or in operating system files.

Oracle recommends that you enable and configure auditing. Auditing is an effective method of enforcing strong internal controls so that your site can meet its regulatory compliance requirements, as defined in the Sarbanes-Oxley Act. This enables you to monitor business operations, and find any activities that may deviate from company policy. Doing so translates into tightly controlled access to your database and the application software, ensuring that patches are applied on schedule and preventing ad hoc changes. By enabling auditing by default, you can generate an audit record for audit and compliance personnel. Be selective with auditing and ensure that it meets your business compliance needs.

#### Why Is Auditing Used?

You typically use auditing to perform the following activities:

* + **Enable accountability for actions.** These include actions taken in a particular schema, table, or row, or affecting specific content.
  + Deter users (or others, such as intruders) from inappropriate actions based on their accountability.
  + **Investigate suspicious activity.** For example, if a user is deleting data from tables, then a security administrator might decide to audit all connections to the database and all successful and unsuccessful deletions of rows from all tables in the database.
  + **Notify an auditor of the actions of an unauthorized user.** For example, an unauthorized user could be changing or deleting data, or the user has more privileges than expected, which can lead to reassessing user authorizations.
  + **Monitor and gather data about specific database activities.** For example, the database administrator can gather statistics about which tables are being updated, how many logical I/Os are performed, or how many concurrent users connect at peak times.
  + **Detect problems with an authorization or access control implementation.** For example, you can create audit policies that you expect will never generate an audit record because the data is protected in other ways. However, if these policies generate audit records, then you will know the other security controls are not properly implemented.
  + **Address auditing requirements for compliance.** Regulations such as the following have common auditing-related requirements:
* Sarbanes-Oxley Act
* Health Insurance Portability and Accountability Act (HIPAA)

1 "Nondatabase users" refers to application users who are recognized in the database using the CLIENT\_IDENTIFIER attribute. To audit this type of user, you can use a fine-grained audit policy. See ["Auditing Specific Activities with Fine-Grained Auditing" on page 9-36](#_bookmark1893) for more information.

* + International Convergence of Capital Measurement and Capital Standards: a Revised Framework (Basel II)
  + Japan Privacy Law
  + European Union Directive on Privacy and Electronic Communications

#### Protecting the Database Audit Trail

When auditing for suspicious database activity, you should protect the integrity of the audit trail records to guarantee the accuracy and completeness of the auditing information.

Oracle Database writes the database audit trail to the SYS.AUD$ and SYS.FGA\_LOG$ tables. Audit records generated as a result of object audit options set for the SYS.AUD$ and SYS.FGA\_LOG$ tables can only be deleted from the audit trail by someone who has connected with administrator privileges. Remember that administrators are also audited for unauthorized use. See ["Auditing SYS Administrative Users" on page 9-53](#_bookmark1968) for more information.

Other ways to protect the database audit trail are as follows:

* **Set the O7\_DICTIONARY\_ACCESSIBILITY initialization parameter to FALSE (the default).** This way, only users who have the SYSDBA privilege can perform DML actions on the audit data in the SYS.AUD$ and SYS.FGA\_LOG$ tables. In a default installation, O7\_DICTIONARY\_ACCESSIBILITY is set to FALSE.
* **If you have Oracle Data****base Vault installed, create a realm around the SYSTEM.AUD$ table.** By default, the AUD$ table is in the SYSTEM schema. (The synonym SYS.AUD$ refers to the SYSTEM.AUD$ table.) See *Oracle Database Vault Administrator's Guide* for more information about realms in Oracle Database Vault.

**See Also:**

* + ["Auditing General Activities with Standard Auditing" on page 9-6](#_bookmark1639)
  + ["Auditing Specific Activities with Fine-Grained Auditing" on page 9-36](#_bookmark1893)

#### Activities That Are Always Written tothe Standard and Fine-Grained Audit Records

When standard auditing is enabled (that is, you set AUDIT\_TRAIL to DB or DB,EXTENDED), Oracle Database audits all data manipulation language (DML) operations, such as INSERT, UPDATE, MERGE, and DELETE on the SYS.AUD$ and SYS.FGA\_ LOG$ tables by non-SYS users. (It performs this audit even if you have not set audit options for the AUD$ and FGA\_LOGS$ tables.) Typically, non-SYS users do not have access to these tables, except if they have been explicitly granted access. If a non-SYS user tampers with the data in the SYS.FGA\_LOG$ and SYS.AUD$ tables, then Oracle Database writes an audit record for each action.

Oracle Database audits SYS user’s DELETE, INSERT, UPDATE, and MERGE operations on the SYS.FGA\_LOG$ and SYS.AUD$ tables if you have set the AUDIT\_SYS\_OPERATIONS initialization parameter to TRUE. In this case the audit records of all SYS operations are written to whatever directory the AUDIT\_FILE\_DEST initialization parameter points to.

If AUDIT\_FILE\_DEST is not set, then it writes the records to an operating system-dependent location.

**See Also:**

* ["Auditing General Activities with Standard Auditing" on page 9-6](#_bookmark1639)
* ["Auditing Specific Activities with Fine-Grained Auditing" on page 9-36](#_bookmark1893)

#### Activities That Are Always Audited for All Platforms

Oracle Database always audits certain database-related operations and writes them to the operating system audit files. It includes the actions of any user who is logged in with the SYSDBA or SYSOPER privilege. This is called [**mandatory auditing**](#_bookmark2347). Even if you have enabled the database audit trail (that is, setting the AUDIT\_TRAIL parameter to DB), Oracle Database still writes mandatory records to operating system files.

By default, the operating system files are in the $ORACLE\_BASE/admin/$ORACLE\_ SID/adump directory for both UNIX and Windows systems. On Windows systems, Oracle Database also writes this information to the Windows Event Viewer. You can change the location of this directory by setting the AUDIT\_FILE\_DEST initialization parameter, which is described in ["Specifying a Directory for the Operating System](#_bookmark1713) [Audit Trail" on page 9-17](#_bookmark1713).

Mandatory auditing includes the following operations:

* + **Database startup.** An audit record is generated that lists the operating system user starting the instance, the user terminal identifier, and the date and time stamp. This data is stored in the operating system audit trail because the database audit trail is not available until after the startup has successfully completed.
  + **SYSDBA and SYSOPER logins.** Oracle Database records all SYSDBA and SYSOPER

connections.

* + **Database shutdown.** An audit record is generated that lists the operating system user shutting down the instance, the user terminal identifier, and the date and time stamp.

**Note:** If you set the AUDIT\_SYSLOG\_LEVEL initialization parameter, mandatory actions are written the to the UNIX syslog. See ["Using the](#_bookmark1719) [Syslog Audit Trail on UNIX Systems" on page 9-18](#_bookmark1719) for more information about the syslog audit trail. See also your operating system-specific Oracle Database documentation for more information about the operating system and syslog audit trail.

#### Auditing ina Distributed Database

Auditing is site autonomous. An instance audits only the statements issued by directly connected users. A local Oracle Database node cannot audit actions that take place in a remote database.

#### Best Practices for Auditing

Follow these best practices guidelines:

* + As a general rule, design your auditing strategy to collect the amount of information that you need to meet compliance requirements, but being sure to focus on activities that cause the greatest security concerns. For example, auditing every table in the database is not practical, but auditing table columns that contain sensitive data, such as salaries, is. With both standard and fine-grained auditing,

there are mechanisms you can use to design audit policies that focus on specific activities to audit.

* + - Periodically archive and purge the audit trail data. See ["Purging Audit Trail Records" on page 9-65](#_bookmark2043) for more information.

**See Also:** ["Guidelines for Auditing" on page 10-18](#_bookmark2285) for general guidelines to follow for auditing your system

### Selecting an Auditing Type

You must perform a specific set of steps depending on the type of auditing that you want to perform: general activities (such as SQL statement actions), commonly used auditing activities, or fine-grained auditing.

In addition to these types of auditing, remember that Oracle Database mandatorily audits some activities. See Activities That Are Mandatorily Audited for more information.

#### Auditing SQL Statements, Privileges, and Other General Activities

Oracle Database provides a set of default audit settings that you can enable for commonly used security-relevant SQL statements and privileges.

**Location of audit records:** Oracle Database writes these audit records to the location based on the AUDIT\_TRAIL initialization parameter. See also ["About Audit Records" on](#_bookmark1996) [page 9-57](#_bookmark1996).

General steps:

1. Follow the instructions in ["Using Default Auditing for Security-Relevant SQL Statements and Privileges" on page 9-35](#_bookmark1880) to enable default auditing.

To understand more about the database audit trail, see ["Managing Audit Trail](#_bookmark1995) [Records" on page 9-57](#_bookmark1995).

1. To monitor audit activities, periodically query the database audit trail data dictionary views. See ["Finding Information About Audited Activities" on page 9-80](#_bookmark2095).
2. Perform maintenance on the database audit trail. See ["Managing the Database Audit Trail" on page 9-58](#_bookmark1999).
3. Periodically archive and purge the contents of the audit trail. See ["Purging Audit Trail Records" on page 9-65](#_bookmark2043).

#### Auditing Commonly Used Security-Relevant Activities

You can audit at the most granular level, data access, and actions based on content, using Boolean measures, such as value > 7800 or the IP address from which an action occurred.

**Location of audit records:** You can write the audit records to either the database audit trail or an operating system audit trail in XML format. See also ["About Audit Records"](#_bookmark1996) [on page 9-57](#_bookmark1996).

General steps:

1. See ["Auditing Specific Activities with Fine-Grained Auditing" on page 9-36](#_bookmark1893) to understand more about auditing specific activities.
2. Decide whether you want to write audit records to the database audit trail or to an operating system file. See ["Managing the Database Audit Trail" on page 9-58](#_bookmark1999).
3. Use the DBMS\_FGA PL/SQL package to configure fine-grained auditing policies. The DBMS\_FGA.ADD\_POLICY procedure provides the audit\_trail parameter, which you use to select the audit trail type. You can choose between a database audit trail or an operating system audit trail using XML files. See the following sections:

["Creating an Audit Trail for Fine-Grained Audit Records" on page 9-39](#_bookmark1914)

["Using the DBMS\_FGA Package to Manage Fine-Grained Audit Policies" on](#_bookmark1918) [page 9-39](#_bookmark1918)

1. To monitor audit activities, periodically check the operating system records you configured, or query the audit trail data dictionary views. See ["Finding Information About Audited Activities" on page 9-80](#_bookmark2095).
2. Perform maintenance on the audit trail. See ["Managing Audit Trail Records" on page 9-57](#_bookmark1995).
3. Periodically archive and purge the contents of the audit trail. See ["Purging Audit Trail Records" on page 9-65](#_bookmark2043).

#### Auditing Specific, Fine-Grained Activities

You can audit the top-level SQL statements issued by users who have connected using the SYSDBA or SYSOPER privilege. (Top-level refers to statements directly issued by a user. Statements run from a PL/SQL procedure or function are not considered

top-level.)

**Location of audit records:** Oracle Database writes these audit records to an operating system audit trail only. On Windows, Oracle Database writes the SYS audit records to the Windows Event log by default. For UNIX systems, you can write records to a syslog file. See also ["About Audit Records" on page 9-57](#_bookmark1996).

General steps:

1. See ["Auditing SYS Administrative Users" on page 9-53](#_bookmark1968) to configure administrative auditing.
2. To understand more about the operating system audit trail, see [Managing the Operating System Audit Trail on page 9-62](#_bookmark2029).
3. To monitor audit activities, periodically check the operating system or syslog records you configured. If you are writing to an XML file, you can query the V$XML\_AUDIT\_TRAIL and DBA\_COMMON\_AUDIT\_TRAIL views. See ["Finding Information About Audited Activities" on page 9-80](#_bookmark2095).
4. Perform maintenance on the audit trail. See ["Managing Audit Trail Records" on page 9-57](#_bookmark1995)
5. Periodically archive and purge the contents of the audit trail. See ["Purging Audit Trail Records" on page 9-65](#_bookmark2043).

### Auditing General Activities with Standard Auditing

This section contains:

* + [About Standard Auditing](#_bookmark1640)
  + [Configuring Standard Auditing with the AUDIT\_TRAIL Initialization Parameter](#_bookmark1654)
  + [What Do the Operating System and Database Audit Trails Have in Common?](#_bookmark1686)
    - [Using the Operating System Audit Trail](#_bookmark1695)
    - [Using the Syslog Audit Trail on UNIX Systems](#_bookmark1719)
    - [How the AUDIT and NOAUDIT SQL Statements Work](#_bookmark1732)
    - [Auditing SQL Statements](#_bookmark1758)
    - [Auditing Privileges](#_bookmark1784)
    - [Auditing SQL Statements and Privileges in a Multitier Environment](#_bookmark1798)
    - [Auditing Schema Objects](#_bookmark1807)
    - [Auditing Directory Objects](#_bookmark1839)
    - [Auditing Functions, Procedures, Packages, and Triggers](#_bookmark1848)
    - [Auditing Network Activity](#_bookmark1867)

**See Also:**

* ["Auditing SYS Administrative Users" on page 9-53](#_bookmark1968) to learn how to use standard auditing to audit SYS users
* *Oracle Database 2 Day + Security Guide* for a tutorial on creating a standard audit trail

#### About Standard Auditing

This section contains:

* + - [What Is Standard Auditing?](#_bookmark1641)
    - [Who Can Perform Standard Auditing?](#_bookmark1645)
    - [When Are Standard Audit Records Created?](#_bookmark1648)

###### What Is Standard Auditing?

In standard auditing, you audit SQL statements, privileges, schema objects, and network activity. You configure standard auditing by using the AUDIT SQL statement and NOAUDIT to remove this configuration. You can write the audit records to either the database audit trail or to operating system audit files.

###### Who Can Perform Standard Auditing?

Any user can configure auditing for the objects in his or her own schema, by using the AUDIT statement. To undo the audit configuration for this object, the user can use the NOAUDIT statement. No additional privileges are needed to perform this task. Users can run AUDIT statements to set auditing options regardless of the AUDIT\_TRAIL parameter setting. If auditing has been disabled, the next time it is enabled, Oracle Database will record the auditing activities set by the AUDIT statements. ["Enabling or Disabling the](#_bookmark1656) [Standard Audit Trail" on page 9-8](#_bookmark1656) explains how to enable standard auditing.

Note the following:

* + - To audit objects in another schema, the user must have the AUDIT ANY system privilege.
    - To audit system privileges, the user must have the AUDIT SYSTEM privilege.
    - If the O7\_DICTIONARY\_ACCESSIBILITY initialization parameter has been set to FALSE (the default), then only users who have the SYSDBA privilege can perform DML actions on the audit data in the SYS.AUD$ and SYS.FGA\_LOG$ tables. For

greater security, set the O7\_DICTIONARY\_ACCESSIBILITY parameter to FALSE so that non-SYSDBA users cannot audit SYS objects.

**See Also:**

* GRANT in *Oracle Database SQL Language Reference* for a listing of available system and object privileges
* AUDIT in *Oracle Database SQL Language Reference* for a full listing of audit options

###### When Are Standard Audit Records Created?

You, as the security administrator, enable or disable standard auditing for the entire database. If it is disabled, then no audit records are created. Configuring audit options is described in the previous section, ["Who Can Perform Standard Auditing?"](#_bookmark1645)

When auditing is enabled in the database and an action configured to be audited occurs, Oracle Database generates an audit record during or after the execution phase of the SQL statement. Oracle Database individually audits SQL statements inside PL/SQL program units, as necessary, when the program unit is run.

The generation and insertion of an audit trail record is independent of a user transaction being committed. That is, even if a user transaction is rolled back, the audit trail record remains committed.

Statement and privilege audit options in effect at the time a database user connects to the database remain in effect for the duration of the session. When the session is already active, setting or changing statement or privilege audit options does not take effect in that session. The modified statement or privilege audit options take effect only when the current session ends and a new session is created.

In contrast, changes to schema object audit options become immediately effective for current sessions.

**See Also:** *Oracle Database Concepts* for information about the different phases of SQL statement processing and shared SQL

#### Configuring Standard Auditing with the AUDIT\_TRAIL Initialization Parameter

This section contains:

* + [Enabling or Disabling the Standard Audit Trail](#_bookmark1656)
  + [Settings for the AUDIT\_TRAIL Initialization Parameter](#_bookmark1666)

###### Enabling or Disabling the Standard Audit Trail

You enable the standard audit trail by setting the AUDIT\_TRAIL initialization parameter. This setting determines whether to create the audit trail in the database audit trail, write the audit activities to an operating system file, or to disable auditing.

To enable or disable the standard audit trail, log in to SQL\*Plus with administrative privileges, and use the ALTER SYSTEM statement. Afterwards, you need to restart the database instance.

To check the current value of the AUDIT\_TRAIL parameter, use the SHOW PARAMETER

command in SQL\*Plus.

[Example 9–1](#_bookmark1664) shows how to check the AUDIT\_TRAIL parameter setting.

***Example 9–1 Checking the Current Value of the AUDIT\_TRAIL Initialization Parameter***

SHOW PARAMETER AUDIT\_TRAIL

NAME TYPE VALUE

------------------------------------ ----------- -------

audit\_trail string DB

[Example 9–2](#_bookmark1665) shows how to log onto SQL\*Plus, enable the standard audit trail, and then restart the database instance.

***Example 9–2 Enabling the Standard Audit Trail***

CONNECT SYSTEM

Enter password: *password*

ALTER SYSTEM SET AUDIT\_TRAIL=DB SCOPE=SPFILE;

System altered.

CONNECT SYS/AS SYSOPER

Enter password: *password*

SHUTDOWN

Database closed.

Database dismounted.

ORACLE instance shut down.

STARTUP

ORACLE instance started.

This example uses the SCOPE clause because the database instance had been started using a server parameter file (SPFILE). Starting the database with a server parameter file is the preferred way of starting a database instance. See *Oracle Database Administrator's Guide* for information about creating configuring server parameter files.

###### Settings for the AUDIT\_TRAIL Initialization Parameter

[Table 9–1](#_bookmark1670) lists the settings you can use for the AUDIT\_TRAIL initialization parameter.

***Table 9–1 AUDIT\_TRAIL Initialization Parameter Settings***

|  |  |
| --- | --- |
| **AUDIT\_TRAIL Value** | **Description** |
| DB | Directs audit records to the database audit trail (the SYS.AUD$ table), except for mandatory and SYS audit records, which are always written to the operating system audit trail. (["Selecting an Auditing Type" on page 9-5](#_bookmark1631) describes the location of the audit records for each type of auditing.) Use this setting for a general database for manageability. DB is the default setting for the AUDIT\_TRAIL parameter.  If the database was started in read-only mode with AUDIT\_TRAIL set to DB, then Oracle Database internally sets AUDIT\_TRAIL to OS. Check the alert log for details.  See also ["Managing the Database Audit Trail" on page 9-58](#_bookmark1999). |
| DB,EXTENDED | Behaves the same as AUDIT\_TRAIL=DB, but also populates the SQL bind and SQL text CLOB-type columns of the SYS.AUD$ table, when available.  DB,EXTENDED enables you to capture the SQL statement used in the action that was audited. You can capture both the SQL statement that caused the audit, and any associated bind variables. However, be aware that you only can capture data from the following column datatypes: CHAR, NCHAR, VARCHAR, VARCHAR2, NVARCHAR2, NUMBER, FLOAT, BINARY\_FLOAT, BINARY\_DOUBLE, LONG, ROWID, DATE, TIMESTAMP, and TIMESTAMP WITH TIMEZONE. Also be  aware that DB, EXTENDED can capture sensitive data, such as credit card information. See also ["Auditing Sensitive Information" on page 10-18](#_bookmark2288).  If the database was started in read-only mode with AUDIT\_TRAIL set to DB, EXTENDED, then Oracle Database internally sets AUDIT\_TRAIL to OS. Check the alert log for details.  You can specify DB,EXTENDED in any of the following ways:  ALTER SYSTEM SET AUDIT\_TRAIL=DB,EXTENDED SCOPE=SPFILE; ALTER SYSTEM SET AUDIT\_TRAIL=DB, EXTENDED SCOPE=SPFILE; ALTER SYSTEM SET AUDIT\_TRAIL='DB','EXTENDED' SCOPE=SPFILE; ALTER SYSTEM SET AUDIT\_TRAIL=EXTENDED,DB SCOPE=SPFILE; ALTER SYSTEM SET AUDIT\_TRAIL=EXTENDED, DB SCOPE=SPFILE;  However, do not enclose DB, EXTENDED in quotes, for example:  ALTER SYSTEM SET AUDIT\_TRAIL='DB, EXTENDED' SCOPE=SPFILE;  In previous releases, the setting was DB\_EXTENDED. This setting has been retained for backward compatibility but may not be available in future releases. |
| OS | Directs all audit records to an operating system file.  Oracle recommends that you use the OS setting, particularly if you are using an ultra-secure database configuration. See ["Advantages of the Operating System Audit Trail" on page 9-16](#_bookmark1707) for more information. See also [Example 9–3, "Text File Operating System Audit Trail" on](#_bookmark1703) [page 9-13](#_bookmark1703).  If you set AUDIT\_TRAIL to OS, then set the following additional initialization parameters:   * AUDIT\_FILE\_DEST, which specifies the location of the operating system audit record file. On UNIX systems, the default location is $ORACLE\_BASE/admin/$ORACLE\_ SID/adump. For better performance on UNIX systems, set the AUDIT\_FILE\_DEST parameter to a directory on a disk that is locally attached to the host running the Oracle Database instance. On Windows, the OS setting writes the audit trail to the   Application area of the Windows Event Viewer.   * AUDIT\_SYS\_OPERATIONS, if you want to audit the top-level SQL statements directly issued by users who have connected with the SYSDBA or SYSOPER privilege. To enable this auditing, set AUDIT\_SYS\_OPERATIONS to TRUE.   If you set AUDIT\_SYS\_OPERATIONS to TRUE and AUDIT\_TRAIL to XML or XML,EXTENDED,  then Oracle Database writes SYS audit records operating system files in XML format.   * AUDIT\_SYSLOG\_LEVEL, which writes SYS and standard OS audit records to the system audit log using the SYSLOG utility. This option only applies to UNIX environments. See ["Configuring Syslog Auditing" on page 9-19](#_bookmark1729) for more information.   See also ["Managing the Operating System Audit Trail" on page 9-62](#_bookmark2029). |

***Table 9–1 (Cont.) AUDIT\_TRAIL Initialization Parameter Settings***

|  |  |
| --- | --- |
| **AUDIT\_TRAIL Val** | **e Description** |
| XML | Writes to the operating system audit record file in XML format. Records all elements of the AuditRecord node given by the XML schema in <http://xmlns.oracle.com/oracleas/schema/dbserver_audittrail-11_2.xsd>except Sql\_ Text and Sql\_Bind to operating system XML audit files. (This .xsd file represents the schema definition of the XML audit file. An XML schema is a document written in the XML Schema language.)  See also ["Advantages of the Operating System Audit Trail" on page 9-16](#_bookmark1707) and [Example 9–4,](#_bookmark1706) ["XML File Operating System Audit Trail" on page 9-15](#_bookmark1706).  If you set the XML value, then also set the AUDIT\_FILE\_DEST parameter. For all platforms, including Windows, the default location for XML audit trail records is $ORACLE\_ BASE/admin/$ORACLE\_SID/adump.  The XML AUDIT\_TRAIL value does not affect syslog audit file. In other words, if you have set the AUDIT\_TRAIL parameter to XML, then the syslog audit records will still be in text format, not XML file format.  You can control the output for SYS and mandatory audit records as follows:   * **To write SYS and mandatory audit files to operating system files in XML format:** Set AUDIT\_TRAIL to XML or XML,EXTENDED, set AUDIT\_SYS\_OPERATIONS to TRUE, but do not set the AUDIT\_SYSLOG\_LEVEL parameter. * **To write SYS and mandatory audit records to syslog audit files and standard audit records to XML audit files:** Set AUDIT\_TRAIL to XML or XML,EXTENDED, set AUDIT\_SYS\_ OPERATIONS to TRUE, and set the AUDIT\_SYSLOG\_LEVEL parameter. |
| XML, EXTENDED | Behaves the same as AUDIT\_TRAIL=XML, but also includes SQL text and SQL bind information in the operating system XML audit files.  You can specify XML,EXTENDED in either of the following ways:  ALTER SYSTEM SET AUDIT\_TRAIL=XML, EXTENDED SCOPE=SPFILE; ALTER SYSTEM SET AUDIT\_TRAIL='XML','EXTENDED' SCOPE=SPFILE;  However, do not enclose XML, EXTENDED in quotes, for example:  ALTER SYSTEM SET AUDIT\_TRAIL='XML, EXTENDED' SCOPE=SPFILE;  See also the following sections:   * ["Advantages of the Operating System Audit Trail" on page 9-16](#_bookmark1707) * ["Auditing Sensitive Information" on page 10-18](#_bookmark2288) |
| NONE | Disables standard auditing. |

**u**

Note the following:

* + - **You do not** **need** **to rest****art the database afte****r you run the AUDIT or NOAUDIT statements.** You only need to restart the database if you made a universal change, such as changing the AUDIT\_TRAIL initialization parameter.
    - **You do not need to set AUDIT\_TRAIL to enable either fine-grained auditing or SYS auditing.** For fine-grained auditing, you add and remove fine-grained audit policies as necessary, applying them to the specific operations or objects you want to monitor. To enable SYS auditing, set the AUDIT\_SYS\_OPERATIONS parameter to TRUE.

#### What Do the Operating Systemand Database Audit Trails Have in Common?

The operating system and database audit trails both capture many of the same types of actions. [Table 9–2](#_bookmark1692) lists the operating system audit trail records. Most map to equivalent columns in the DBA\_AUDIT\_TRAIL view. For a description of these columns, see *Oracle Database Reference*.

***Table 9–2 Common Audited Actions in the Operating System and Database Audit Trails***

|  |  |
| --- | --- |
| **Operating Syste** | **m Audit Record Equivalent DBA\_AUDIT\_TRAIL View Column** |
| SESSIONID | SESSIONID |
| ENTRYID | ENTRYID |
| STATEMENT | STATEMENTID |
| USERID | USERNAME |
| USERHOST | USERHOST |
| TERMINAL | TERMINAL |
| ACTION | ACTION |
| SYS$OPTIONS  RETURNCODE OBJ$CREATOR OBJ$NAME OBJ$PRIVILEGES AUTH$GRANTEE NEW$OWNER NEW$NAME SES$ACTIONS  LOGOFF$PREAD | Indicates what audit option was set with AUDIT or  NOAUDIT, or what privilege was granted or revoked.1  RETURNCODE OWNER OBJ\_NAME OBJ\_PRIVILEGE GRANTEE NEW\_OWNER NEW\_NAME SES\_ACTIONS  LOGOFF\_PREAD |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| LOGOFF$LWRITE | LOGOFF\_LWRITE |
| COMMENT$TEXT | COMMENT\_TEXT |
| OS$USERID | OS\_USERNAME |
| PRIV$USED | PRIV\_USED |
| SES$LABEL | CLIENT\_ID |
| SES$TID | Does not have an equivalent in the DBA\_AUDIT\_TRAIL  view, but it does appear in the SYS.AUD$ table |
| SPARE2 | Does not have an equivalent in the DBA\_AUDIT\_TRAIL  view, but it does appear in the SYS.AUD$ table |

1 For example, if the ACTION value is 104 (for AUDIT) or 105 (for NOAUDIT), then the SYS$OPTIONS number represents an audit option listed in the STMT\_AUDIT\_OPTION\_MAP table. If the ACTION value is 108 (for GRANT) or 109 (for REVOKE), then the number represents a privilege listed in the SYSTEM\_PRIVILEGE\_MAP table.

#### Using the Operating SystemAudit Trail

This section contains:

* + [About the Operating System Trail](#_bookmark1696)
  + [What Do Operating System Audit Trail Records Look Like?](#_bookmark1700)
  + [Advantages of the Operating System Audit Trail](#_bookmark1707)
  + [How the Operating System Audit Trail Works](#_bookmark1711)
  + [Specifying a Directory for the Operating System Audit Trail](#_bookmark1713)

###### About the Operating System Trail

As an alternative to creating standard audit records in the DBA\_AUDIT\_TRAIL (SYS.AUD$ table), you can create standard audit records in operating system files. The operating system file that contains the audit trail can include any of the following data:

* + - Database audit trail records
    - Mandatory audit records (that is, database actions that are always audited)
    - Audit records for administrative users (SYS)

You can write the operating system audit records to either a text file or an XML file.

###### What Do Operating System Audit Trail Records Look Like?

The operating system audit trail files are in either text or XML file format. Be aware that the contents of the text and XML operating system files have some differences, and that the formats may change across different releases. With each release of Oracle Database, new enhancements, such as the audit type, have been made to the XML file, but not the text file. The text operating system file has a different presentation for the timestamp, for example:

Wed May 6 00:57:36 2009 -07:00

However, this timestamp does not appear in the event log or syslog, which have their own format for timestamps. The timestamp string only appears in the text operating system audit files.

[Example 9–3](#_bookmark1703) shows a typical text operating system audit trail for a logon operation on an Oracle database that is installed on Microsoft Windows. (The text in the actual record wraps around, but for this manual, each item is separated onto its own line for easier readability.)

***Example 9–3 Text File Operating System Audit Trail***

Audit trail:

LENGTH: "349"

SESSIONID:[5] "43464"

ENTRYID:[1] "1"

STATEMENT:[1] "1" USERID:[6] "DBSNMP" USERHOST:[7] "SHOBEEN" TERMINAL:[3] "MAU" ACTION:[3] "100"

RETURNCODE:[1] "0"

COMMENT$TEXT:[97] "Authenticated by: DATABASE; Client address: (ADDRESS=(PROTOCOL=tcp)(HOST=192.0.2.4)(PORT=2955))" OS$USERID:[19] "NT AUTHORITY\SYSTEM"

DBID:[10] "1212547373"

PRIV$USED:[1] "5"

In this example:

* + - LENGTH refers to the total number of bytes used in this audit record. This number includes the trailing newline bytes (\n), if any, at the end of the audit record.
    - [] brackets indicate the length of each value for each audit entry. For example, the

USERID entry, DBSNMP, is 6 bytes long.

* + - SESSIONID indicates the audit session ID number. You can also find the session ID by querying the AUDSID column in the V$SESSION data dictionary view.
  + ENTRYID indicates the current audit entry number, assigned to each audit trail record. The audit ENTRYID sequence number is shared between fine-grained audit records and regular audit records.
  + STATEMENT is a numeric ID assigned to the statement the user runs. It appears for each statement issued during the user session, because a statement can result in multiple audit records.
  + ACTION is a numeric value representing the action the user performed. The corresponding name of the action type is in the AUDIT\_ACTIONS table. For example, action 100 refers to LOGON.
  + RETURNCODE indicates if the audited action was successful. 0 indicates success. If the action fails, the return code lists the Oracle Database error number. For example, if you try to drop a non-existent table, the error number is ORA-00903 invalid table name, which in turn translates to 903 in the RETURNCODE setting.
  + COMMENT$TEXT indicates additional comments about the audit record. For example, for LOGON audit records, it can indicate the authentication method.It corresponds to the COMENT\_TEXT column of the DBA\_COMMON\_AUDIT\_TRAIL data dictionary view.
  + DBID is a database identifier calculated when the database is created. It corresponds to the DBID column of the V$DATABASE data dictionary view.
  + ECONTEXT\_ID indicates the application execution context identifier.
  + PRIVS$USED refers to the privilege that was used to perform an action. To find the privilege, query the SYSTEM\_PRIVILEGE\_MAP table. For example, privilege 5 refers to -5 in this table, which means CREATE SESSION. PRIVS$USED corresponds to the PRIV\_USED column in the DBA\_COMMON\_AUDIT\_TRAIL, which lists the privilege by name.

Other possible values are as follows:

* + SCN (for example, SCN:8934328925) indicates the System Change Number (SCN). Use this value if you want to perform a flashback query to find the value of a setting (for example, a column) at a time in the past. For example, to find the value of the ORDER\_TOTAL column of the OE.ORDERS table based on the SCN number, use the following SELECT statement:

SELECT ORDER\_TOTAL FROM OE.ORDERS

AS OF SCN = 8934328925 WHERE ORDER\_TOTAL = 86;

* + SES\_ACTIONS indicates the actions that took place during the session. This field is present only if the event was audited with the BY SESSION clause. Because this field does not explain in detail the actions that occurred during the session, you should configure the audit event with the BY ACCESS clause.

The SES\_ACTIONS field contains 16 characters. Positions 14, 15, and 16 are reserved for future use. In the first 12 characters, each position indicates the result of an action. They are: ALTER, AUDIT, COMMENT, DELETE, GRANT, INDEX, INSERT, LOCK,

RENAME, SELECT, UPDATE, and FLASHBACK. For example, if the user had successfully run the ALTER statement, the SES\_ACTIONS setting is as follows:

S---------------

The S, in the first position (for ALTER), indicates success. Had the ALTER statement failed, the letter F would have appeared in its place. If the action resulted in both a success and failure, then the letter is B.

* + - SES$TID indicates the ID of the object affected by the audited action.
    - SPARE2 indicates whether the user modified SYS.AUD$ table. 0 means the user modified SYS.AUD$; otherwise, the value is NULL.

Similarly, [Example 9–4](#_bookmark1706) shows how an XML audit trail record appears. The text wraps around in the actual record, but for this manual, each element appears on its own line for easier readability. To find all the tags that appear in the XML audit file, you can view its schema in a Web browser at

<http://www.oracle.com/technology/oracleas/schema/dbserver_audittrail-11_> 2.xsd

***Example 9–4 XML File Operating System Audit Trail***

<?xml version="1.0" encoding="UTF-8"?>

<Audit xmln[s="http://xmlns.oracle.com/oracleas/schema/dbserver\_audittrail-11\_](http://xmlns.oracle.com/oracleas/schema/dbserver_audittrail-11_) 2.xsd"

xmlns:xsi=["http://www.w3.org/2001/XMLSchema-instance](http://www.w3.org/2001/XMLSchema-instance)" xsi:schemaLocation=["http://xmlns.oracle.com/oracleas/schema/dbserver\_](http://xmlns.oracle.com/oracleas/schema/dbserver_)

audittrail-11\_2.xsd">

<Version>11.2</Version>

<AuditRecord>

<Audit\_Type>1</Audit\_Type>

<Session\_Id>43535</Session\_Id>

<StatementId>1</StatementId>

<EntryId>1</EntryId>

<Extended\_Timestamp>2009-04-29T18:32:26.062000Z</Extended\_Timestamp>

<DB\_User>SYSMAN</DB\_User>

<OS\_User>SYSTEM</OS\_User>

<Userhost>shobeen</Userhost>

<OS\_Process>3164:3648</OS\_Process>

<Terminal>mau</Terminal>

<Instance\_Number>0</Instance\_Number>

<Action>100</Action>

<TransactionId>0000000000000000</TransactionId>

<Returncode>0</Returncode>

<Comment\_Text>Authenticated by: DATABASE; Client address: (ADDRESS=(PROTOCOL=tcp)(HOST=192.0.2.4)(PORT=3536))</Comment\_Text>

<Priv\_Used>5</Priv\_Used>

</AuditRecord>

</Audit>

In this example:

* + - AuditRecord element contains the entire audit record. (See [Example 9–3](#_bookmark1703) for more information about the elements within the Audit\_Record element.)
    - Audit\_Type indicates the type of audit trail. Possible values are as follows:
      * 1: Standard audit record
      * 2: Fine-grained audit record
      * 4: SYS audit record
      * 8: Mandatory audit record

This field only appears in the XML audit files, not the OS text audit files.

* + - Extended\_Timestamp indicates the time of the audited operation (timestamp of user login for entries created by AUDIT SESSION), in Coordinated Universal Time

(UTC) or Greenwich Mean Time (GMT). This field only appears in the XML audit files, not the OS text audit files.

* + Instance\_Number indicates the instance number to which the user is connected, for an Oracle Real Application Clusters environment. In this example, the number is 0, which is used for single-instance database installations. The INSTANCE\_NUMBER initialization parameter specifies this number.

The following values can appear if you set the AUDIT\_TRAIL parameter to XML, EXTENDED. Both are listed in the DBA\_COMMON\_AUDIT\_TRAIL data dictionary view.

* + Sql\_Bind (for example, <Sql\_Bind>#1(5):89</Sql\_Bind>) shows the value of the bind variable. The syntax is as follows:

*VariablePosition*(*LengthOfVariableValue*):*ValueofBindVariable*

The example #1(5):89 indicates that there is 1 bind variable; its value is 5 characters long; and the value of the bind variable is 89.

* + Sql\_Text (for example, <Sql\_Text>begin procedure\_one(:num); end; </Sql\_ Text>) appears if you have set the AUDIT\_TRAIL parameter to XML, EXTENDED. It shows the SQL text that the user entered.

###### Advantages of the Operating System Audit Trail

Using the operating system audit trail offers these advantages:

* + It reduces the likelihood of a denial-of-service (DoS) attack.
  + It makes it easier to secure the audit trail. If the auditor is distinct from the database administrator, then you *must* use the OS, XML, or XML, EXTENDED setting. Otherwise, a database administrator can view and modify any auditing information that is stored in the database.
  + Because you are writing the audit trail to a specific location that you can restrict to specific users, the operating system audit trail enforces separation of duty concepts.
  + Writing the audit trail to an operating system file results in the least amount of overhead on the database. For this reason, it is excellent for very large databases.
  + Audit records stored in operating system files can be more secure than database-stored audit records because access can require file permissions that

database administrators do not have. Greater availability is another advantage to operating system storage for audit records, because they remain available even if the database is temporarily inaccessible.

* + If the AUDIT\_TRAIL initialization parameter is set to XML (or XML, EXTENDED), then Oracle Database writes audit records to the operating system as XML files. You can use the V$XML\_AUDIT\_TRAIL view to make XML audit records available to database administrators through a SQL query, providing enhanced usability.
  + The DBA\_COMMON\_AUDIT\_TRAIL view includes the standard and fine grained audit trails written to database tables, XML-format audit trail records, and the contents of the V$XML\_AUDIT\_TRAIL dynamic view (standard, fine grained, SYS and mandatory).
  + Using your operating system audit trail can enable you to consolidate audit records from multiple sources, including Oracle Database and other applications. Examining system activity can be more efficient with all audit records in one place. If you use XML audit records, then you can use of any standard XML editing tool to review or extract information from those records.

###### How the Operating System Audit Trail Works

The operating system audit trail writes the audit data to an operating system file. You can enable this feature by setting the AUDIT\_TRAIL initialization parameter to one of the following values:

* + - OS: Writes the audit trail records to a text operating system file on UNIX systems and to the applications Event Viewer on Microsoft Windows.
    - XML: Writes the audit trail records to an XML file.
    - XML, EXTENDED: Writes the audit trail records to an XML file and includes SQL text and SQL bind information in the operating system XML audit files.

The AUDIT\_FILE\_DEST initialization parameter sets the location of the operating system audit file. If you want to audit top-level statements issued by users who log in to the database with the SYSDBA or SYSOPER privilege, then set the AUDIT\_SYS\_OPERATIONS parameter to TRUE. See [Table 9–1, " AUDIT\_TRAIL Initialization Parameter Settings"](#_bookmark1670) [on page 9-10](#_bookmark1670) for more information about these settings.

The records that are written to an operating system file are not recorded to the SYS.AUD$ and SYS.FGA\_LOG$ tables. You can still view the contents of the XML operating system audit files by querying the DBA\_COMMON\_AUDIT\_TRAIL data dictionary views. Querying this view parses all XML files (all files with an .xml extension) in the AUDIT\_FILE\_DEST directory, and then presents them in relational table format. Because XML is a standard document format, many utilities are available to parse and analyze XML data. Consult the operating system-specific Oracle Database documentation to find if this feature has been implemented on your operating system.

###### Specifying a Directory for the Operating System Audit Trail

Use the AUDIT\_FILE\_DEST initialization parameter to specify an operating system directory into which the audit trail is written, when the AUDIT\_TRAIL initialization parameter is set to OS, XML, or XML, EXTENDED. You must set AUDIT\_FILE\_DEST to a valid directory with permissions restricted to the owner of the Oracle software and the DBA group. Mandatory auditing information also goes into that directory, as do audit records for user SYS if the AUDIT\_SYS\_OPERATIONS initialization parameter is specified. You can change the AUDIT\_FILE\_DEST parameter by using the following ALTER SYSTEM statement, which enables the new destination to be effective for all subsequent sessions.

ALTER SYSTEM SET AUDIT\_FILE\_DEST = *directory\_path* DEFERRED;

To find the current setting of the AUDIT\_FILE\_DEST parameter, issue the following command:

SHOW PARAMETER AUDIT\_FILE\_DEST

The location of the operating system files depends on the following:

* + - If the database is not running and you have not set the AUDIT\_FILE\_DEST

parameter, then the operating system files are placed in the first default location

$ORACLE\_BASE/admin/$ORACLE\_SID/adump directory.

* + - If the database is not running and the first default location, the $ORACLE\_ BASE/admin/$ORACLE\_SID/adump directory, is inaccessible or cannot be written to, or the Oracle process cannot identify the environment variables, then the second default location, $ORACLE\_HOME/rdbms/audit is used.
    - When the database is open and Oracle Database reads the initialization file (init*SID*.ora) for the database instance, the value of AUDIT\_FILE\_DEST parameter is used as the operating system audit file directory.
  + For UNIX and Solaris systems, all operating system files are written to a directory in the operating system. For Windows, the operating system text records are available from the Windows Event Viewer, but operating system XML files are available from an operating system directory, as explained in the preceding bulleted items.

**Notes:** For platforms other than UNIX, Solaris, and Windows, check the platform documentation to learn the correct target directory for operating system files.

#### Using the Syslog Audit Trail on UNIX Systems

On UNIX systems, you can audit the activities of users, including privileged users, and record these activities in a syslog file by creating a syslog audit trail.

This section contains:

* + [About the Syslog Audit Trail](#_bookmark1721)
  + [Format of the Information Stored in the Syslog Audit Trail](#_bookmark1724)
  + [What Does the Syslog Audit Trail Look Like?](#_bookmark1726)
  + [Configuring Syslog Auditing](#_bookmark1729)

###### About the Syslog Audit Trail

You can use a syslog audit trail to configure centralized audit facility provided by the operating system Syslog facility. Syslog is a standard protocol on UNIX-based systems for logging information from different components of a network. Applications call the syslog() function to log information to the syslog daemon, which then determines where to log the information. Be aware that when you configure the use of syslog files, the messages are sent to the syslog daemon process. The syslog daemon process does not return an acknowledgement to Oracle Database indicating a committed write to the syslog files.

Because applications, such as an Oracle process, use the syslog() function to log information to the syslog daemon, a privileged user would not have permissions to the file system where syslog messages are logged. For this reason, audit records stored using a syslog audit trail can be more secure than audit records stored using an operating system audit trail. In addition to restricting permissions to a file system for a privileged user, for a syslog audit trail to be secure, neither privileged users nor the Oracle process should have root access to the system where the audit records are written.

**Caution:** You should have a strong understanding of how to work with syslog before enabling syslog auditing. See the following references for more information about syslog:

* + - *Oracle Database Reference* for information about the AUDIT\_SYSLOG\_ LEVEL initialization parameter
    - The UNIX man page for the syslogd utility for more information about the *facility.priority* settings and their directory paths

###### Format of the Information Stored in the Syslog Audit Trail

Similar to the operating system audit trail records, Oracle Database encodes the syslog records to ensure greater security. If you have Oracle Audit Vault installed, you can use its Syslog Collector to extract and transfer syslog audit records to centralized Oracle Audit Vault server.

###### What Does the Syslog Audit Trail Look Like?

[Example 9–5](#_bookmark1728) shows how the syslog audit trail can appear. (For this example, the text has been reformatted for easier readability. In reality, the text is all on one line.) As with other Oracle Database audit trails, the brackets indicate the length of the value that was audited. For syslog audit trails, the text from (and including) LENGTH: is Oracle Database audit record. The prepended text (the date and Oracle Audit [10085] line) is added by the syslog utility.

***Example 9–5 Syslog Audit Trail for SYS User***

May 14 23:40:15 shobeen Oracle Audit[10085]: LENGTH : '171'

ACTION :[18] 'select \* from aud$' DATABASE USER:[1] '/'

PRIVILEGE :[6] 'SYSDBA'

CLIENT USER:[7] 'laurelh' CLIENT TERMINAL:[6] 'pts/12' STATUS:[1] '0'

DBID:[9] '562317007'

###### Configuring Syslog Auditing

To enable syslog auditing, follow these steps:

* 1. Assign the value of OS to the AUDIT\_TRAIL initialization parameter, as described in ["Enabling or Disabling the Standard Audit Trail" on page 9-8](#_bookmark1656).

For example:

ALTER SYSTEM SET AUDIT\_TRAIL=OS SCOPE=SPFILE;

* 1. Manually set the AUDIT\_SYSLOG\_LEVEL parameter to the initialization parameter file, init*sid*.ora.

Set the AUDIT\_SYSLOG\_LEVEL parameter to specify a facility and priority in the format AUDIT\_SYSLOG\_LEVEL=*facility.priority*.

* + - *facility*: Describes the part of the operating system that is logging the message. Accepted values are user, local0–local7, syslog, daemon, kern, mail, auth, lpr, news, uucp, and cron.

The local0–local7 values are predefined tags that enable you to sort the syslog message into categories. These categories can be log files or other destinations that the syslog utility can access. To find more information about these types of tags, refer to the syslog utility MAN page.

* + - *priority*: Defines the severity of the message. Accepted values are notice, info, debug, warning, err, crit, alert, and emerg.

The syslog daemon compares the value assigned to the facility argument of the AUDIT\_SYSLOG\_LEVEL parameter with the syslog.conf file to determine where to log information.

For example, the following statement identifies the facility as local1 with a priority level of warning:

AUDIT\_SYSLOG\_LEVEL=local1.warning

See *Oracle Database Reference* for more information about AUDIT\_SYSLOG\_LEVEL.

* 1. Log in to the computer that contains the syslog configuration file,

/etc/syslog.conf, with the superuser (root) privilege.

* 1. Add the audit file destination to the syslog configuration file syslog.conf.

For example, assuming you had set the AUDIT\_SYSLOG\_LEVEL to local1.warning, enter the following:

local1.warning /var/log/audit.log

This setting logs all warning messages to the /var/log/audit.log file.

* 1. Restart the syslog logger:

$/etc/rc.d/init.d/syslog restart

Now, all audit records will be captured in the file /var/log/audit.log through the syslog daemon.

* 1. Restart the database instance:

CONNECT SYS / AS SYSOPER

Enter password: *password*

SHUTDOWN IMMEDIATE STARTUP

#### Howthe AUDIT and NOAUDIT SQL Statements Work

This section contains:

* + [Enabling Standard Auditing with the AUDIT SQL Statement](#_bookmark1734)
  + [Auditing Statement Executions: Successful, Unsuccessful, or Both](#_bookmark1739)
  + [How Standard Audit Records Are Generated](#_bookmark1742)
  + [How Do Cursors Affect Standard Auditing?](#_bookmark1745)
  + [Benefits of Using the BY ACCESS Clause in the AUDIT Statement](#_bookmark1747)
  + [Auditing Actions Performed by Specific Users](#_bookmark1750)
  + [Removing the Audit Option with the NOAUDIT SQL Statement](#_bookmark1753)

**See Also:** *Oracle Database SQL Language Reference* for a description of the AUDIT statement syntax

###### Enabling Standard Auditing with the AUDIT SQL Statement

To configure the standard auditing option, use the AUDIT SQL statement. [Table 9–3](#_bookmark1738) lists the categories in which you can use the AUDIT statement.

***Table 9–3 Standard Auditing Levels and Their Effects***

|  |  |
| --- | --- |
| **Level** | **Effect** |
| Statement | Audits specific SQL statements or groups of statements that affect a particular type of database object. For example, AUDIT TABLE audits the CREATE TABLE, TRUNCATE TABLE, COMMENT ON  TABLE, and DELETE [FROM] TABLE statements. |
| Privilege | Audits SQL statements that are authorized by the specified system privilege. For example, AUDIT CREATE ANY TRIGGER audits statements issued using the CREATE ANY TRIGGER system privilege. |
| Object | Audits specific statements on specific objects, such as ALTER TABLE on the HR.EMPLOYEES table. |
| Network | Audits unexpected errors in network protocol or internal errors in the network layer. |

###### Auditing Statement Executions: Successful, Unsuccessful, or Both

For statement, privilege, and schema object auditing, Oracle Database permits the selective auditing of successful executions of statements, unsuccessful attempts to execute statements, or both. This enables you to monitor actions even if the audited statements do not complete successfully. Monitoring unsuccessful SQL statement can expose users who are snooping or acting maliciously, though most unsuccessful SQL statements are neither.

This method of auditing is also useful in that it reduces the audit trail, helping you to focus on specific actions. This can aid in maintaining good database performance.

The options are as follows:

* + - **WHENEVER SUCCESSFUL clause:**This clause audits only successful executions of the audited statement.
    - **WHENEVER NOT SUCCESSFUL clause:** This clause audits only unsuccessful executions of the audited statement.

Auditing an unsuccessful statement execution generates an audit report only if a valid SQL statement is issued but fails, because it lacks proper authorization or references a nonexistent schema object. Statements that fail to execute because they were not valid cannot be audited.

For example, an enabled privilege auditing option set to audit unsuccessful statement executions audits statements that use the target system privilege but failed for other reasons. One example is when a CREATE TABLE auditing condition is set, but some CREATE TABLE statements fail due to insufficient quota for the specified tablespace.

* + - **Omitting WHENEVER SUCCESSFUL or WHENEVER NOT SUCCESSFUL:** If you omit these clauses, then Oracle Database audits both successful and unsuccessful executions of the audited statement.

For example:

AUDIT CREATE TABLE BY ACCESS WHENEVER NOT SUCCESSFUL;

###### How Standard Audit Records Are Generated

Oracle Database generates an audit record for each execution of an audited statement or operation, as follows:

* + Each time the SQL statement for which auditing was configured is executed. This also includes the execution of the statements within PL/SQL procedures.
  + Each time the privilege for which auditing was configured is used
  + Each time the object for which auditing was configured is operated upon

###### How Do Cursors Affect Standard Auditing?

For each execution of an auditable operation within a cursor, Oracle Database inserts one audit record into the audit trail. Events that cause cursors to be reused include the following:

* + An application, such as Oracle Forms, holding a cursor open for reuse
  + Subsequent execution of a cursor using new bind variables
  + Statements executed within PL/SQL loops where the PL/SQL engine optimizes the statements to reuse a single cursor

Auditing is *not* affected by whether or not a cursor is shared. Each user creates her or his own audit trail records on first execution of the cursor.

###### Benefits of Using the BY ACCESS Clause in the AUDIT Statement

By default, Oracle Database writes a new audit record for every audited event, using the BY ACCESS clause functionality. To use this functionality, either include BY ACCESS in the AUDIT statement, or if you want, you can omit it because it is the default. (As of Oracle Database 11*g* Release 2 (11.2.0.2), the BY ACCESS clause is the default setting.)

Oracle recommends that you audit BY ACCESS and not BY SESSION in your AUDIT statements. The benefits of using the BY ACCESS clause in the AUDIT statement are as follows:

* + The audit records generated through the BY ACCESS audit option have more information, such as execution status (return code), date and time of execution, the privileges used, the objects accessed, the SQL text itself and its bind values. In addition, the BY ACCESS audit option captures the SCN for each execution and this can help flashback queries.
  + Oracle Database records separately each execution of a SQL statement, the use of a privilege, and access to the audited object. Given that the values for the return code, timestamp, SQL text recorded are accurate for each execution, this can help you find how many times the action was performed.
  + The BY ACCESS audit records have separate LOGON and LOGOFF entries, each with fine-grained timestamps.

For example:

AUDIT SELECT TABLE BY ACCESS;

In this scenario:

* + The user jward connects to the database and issues five SELECT statements against the table named departments and then disconnects from the database.
  + The user swilliams connects to the database and issues three SELECT statements against the departments table and then disconnects from the database.

The audit trail contains eight records, one recorded for each SELECT statement.

###### Auditing Actions Performed by Specific Users

Statement and privilege audit options can audit statements issued by any user or statements issued by a specific list of users. By focusing on specific users, you can minimize the number of audit records generated.

[Example 9–6](#_bookmark1752) shows how to audit statements by users scott and blake when they query or update a table or view.

***Example 9–6 Using AUDIT to Audit User Actions***

AUDIT SELECT TABLE, UPDATE TABLE BY scott, blake BY ACCESS;

See *Oracle Database SQL Language Reference* for additional information about auditing by user.

###### Removing the Audit Option with the NOAUDIT SQL Statement

The NOAUDIT statement removes the audit option. Use it to reset statement and privilege audit options, and object audit options. A NOAUDIT statement that sets statement and privilege audit options can include the BY *user* clause to specify a list of users to limit the scope of the statement and privilege audit options.

You can use the NOAUDIT statement to disable an audit option selectively using the WHENEVER clause. If the clause is not specified, then the auditing option is disabled entirely, for both successful and unsuccessful cases.

The NOAUDIT statement does not support the BY ACCESS clause. You can remove audit options, no matter how they were turned on, by using an appropriate NOAUDIT statement.

**See Also:** *Oracle Database SQL Language Reference* for a description of the NOAUDIT statement syntax

#### Auditing SQL Statements

This section contains:

* + - [About SQL Statement Auditing](#_bookmark1760)
    - [Types of SQL Statements That Are Audited](#_bookmark1764)
    - [Configuring SQL Statement Auditing](#_bookmark1767)
    - [Removing SQL Statement Auditing](#_bookmark1778)

###### About SQL Statement Auditing

SQL statement auditing is the selective auditing of related groups of SQL statements regarding a particular type of database structure or schema object, but not a specifically named structure or schema object.

###### Types of SQL Statements That Are Audited

The statements that you can audit are in the following categories:

* + - **DDL statements.** For example, AUDIT TABLE audits all CREATE and DROP TABLE

statements

* + - **DML statements.** For example, AUDIT SELECT TABLE audits all SELECT ... FROM TABLE/VIEW statements, regardless of the table or view

Statement auditing can be broad or focused, for example, by auditing the activities of all database users or of only a select list of activities.

###### Configuring SQL Statement Auditing

Use the AUDIT statement to configure SQL statement auditing. You must have the AUDIT SYSTEM system privilege before you can enable auditing. Typically, only the security administrator is granted this system privilege.

[Example 9–7](#_bookmark1770) shows how to audit the SELECT TABLE SQL statement.

***Example 9–7 Using AUDIT to Enable SQL Statement Auditing***

AUDIT SELECT TABLE BY ACCESS;

[Example 9–8](#_bookmark1771) shows how to audit all unsuccessful SELECT, INSERT, and DELETE

statements on all tables by all database users, and by individual audited statement.

***Example 9–8 Auditing Unsuccessful Statements***

AUDIT SELECT TABLE, INSERT TABLE, DELETE TABLE BY ACCESS

WHENEVER NOT SUCCESSFUL;

If you plan to audit all SQL statements, individual user connections, or references to non-existent objects, follow these guidelines:

* + **Auditing all SQL statements for individual users.** You can use the ALL STATEMENTS clause to audit *only* the top-level SQL statements. The behavior of this audit option is different from other statement audit options. If the SQL statement is issued from inside a PL/SQL procedure, then the ALL STATEMENTS audit option does not audit it. This audit option does not affect any other AUDIT options that you may have already set.

For example, to audit all successful statements issued by users jward and jsmith, enter the following:

AUDIT ALL STATEMENTS BY jward, jsmith BY ACCESS WHENEVER SUCCESSFUL;

* + **Audi****ting all the SQL statement shortcut activities performed by individual users.** You can use the ALL clause to audit all the SQL statement shortcuts listed in Table 13-1 and Table 13-2 in *Oracle Database SQL Language Reference*.

For example:

AUDIT ALL BY jward BY ACCESS;

* + **Auditing all SQL statements for the current session, regardless of user.**You can use the IN SESSION CURRENT clause for ALL STATEMENTS audit option to audit

top-level SQL statements in the lifetime of the user session. You cannot use the IN SESSION CURRENT clause for a specific user. You cannot use the NOAUDIT statement to cancel it, but the auditing lasts only as long as the user session lasts. When the user ends the session, the auditing ends.

For example, to audit all unsuccessful statements in any current user session:

AUDIT ALL STATEMENTS IN SESSION CURRENT BY ACCESS WHENEVER NOT SUCCESSFUL;

You can use the AUDIT ALL STATEMENTS audit option with the IN SESSION CURRENT clause in a database logon trigger. The database logon trigger can use SYS\_CONTEXT function to configure this auditing only under certain conditions,

such as the time a user logs in between 6:30 p.m. to 9:00 a.m. This would enable you to capture SQL statements performed by users who log in to the database during non-work hours.

This type of auditing is useful to increase the collection of audit activity when you suspect this connection may not be secure or could pose an internal threat. For example, by using a database logon trigger, you can query contents of the connection context using the SYS\_CONTEXT function.

The logon trigger functionality can establish that this connection should be audited more fully. Issue the following SQL command:

AUDIT ALL STATEMENTS IN SESSION CURRENT;

This type of auditing remains in effect until this session is terminated.

* + - **Auditing login and logoff connections and disconnections.** The AUDIT SESSION statement generates an independent audit record for every login and logoff event. This enables you to audit all successful and unsuccessful connections to and disconnections from the database, regardless of user.

For example:

AUDIT SESSION BY ACCESS;

You can set this option selectively for individual users also, as in the following example:

AUDIT SESSION BY jward, jsmith BY ACCESS;

* + - **Auditing statements that fail because an object does not exist.** The NOT EXISTS option of the AUDIT statement specifies auditing of all SQL statements that fail because the target object does not exist.

For example:

AUDIT NOT EXISTS;

See *Oracle Database SQL Language Reference* for detailed information about the AUDIT

SQL statement.

###### Removing SQL Statement Auditing

To remove SQL statement auditing, use the use the NOAUDIT SQL statement. (Privilege auditing will still be enabled.) You must have the AUDIT SYSTEM system privilege before you can remove SQL statement auditing. If you have configured the AUDIT ALL STATEMENTS option, then issuing the NOAUDIT AUDIT STATEMENTS statement does not affect other audit options you may have set. If you included the IN SESSION CURRENT clause in the AUDIT statement, you cannot remove this AUDIT statement using the NOAUDIT statement. (The audit setting discontinues when the user’s session ends.)

[Example 9–9](#_bookmark1782) shows examples of using the NOAUDIT statement to remove auditing.

***Example 9–9 Using NOAUDIT to Remove Session and SQL Statement Auditing***

NOAUDIT session;

NOAUDIT session BY preston, sebastian;

NOAUDIT SELECT TABLE, INSERT TABLE, DELETE TABLE, EXECUTE PROCEDURE;

[Example 9–10](#_bookmark1783) shows how to remove all statement auditing by using the NOAUDIT

statement.

***Example 9–10 Using NOAUDIT to Remove ALL STATEMENTS Auditing***

NOAUDIT ALL STATEMENTS;

See *Oracle Database SQL Language Reference* for detailed information about the NOAUDIT

statement.

#### Auditing Privileges

This section contains:

* + [About Privilege Auditing](#_bookmark1786)
  + [Types of Privileges That Can Be Audited](#_bookmark1789)
  + [Configuring Privilege Auditing](#_bookmark1791)
  + [Removing Privilege Auditing](#_bookmark1795)

###### About Privilege Auditing

Privilege auditing audits statements that use a system privilege, such as SELECT ANY TABLE. In this kind of auditing, SQL statements that require the audited privilege to succeed are recorded.

###### Types of Privileges That Can Be Audited

You can audit the use of any system privilege. Similar to statement auditing, privilege auditing audits the activities of all database users or only a specified list.

If you set similar audit options for both statement and privilege auditing, then only a single audit record is generated. For example, if the statement clause TABLE and the system privilege CREATE TABLE are both audited, then only a single audit record is generated each time a table is created.

Privilege auditing does not occur if the action is already permitted by the existing owner and object privileges. Privilege auditing is triggered only if the privileges are insufficient, that is, only if what makes the action possible is a system privilege. For example, suppose that user SCOTT has been granted the SELECT ANY TABLE privilege and the SELECT ANY TABLE is being audited. If SCOTT selects his own table (for example, SCOTT.EMP), then the SELECT ANY TABLE privilege is not used. Because he performed the SELECT statement within his own schema, no audit record is generated. On the other hand, if SCOTT selects from another schema (for example, the HR.EMPLOYEES table), then an audit record *is* generated. Because SCOTT selected a table outside his own schema, he needed to use the SELECT ANY TABLE privilege.

Privilege auditing is more focused than statement auditing, because each privilege auditing option audits only specific types of statements, not a related list of statements. For example, the statement auditing clause, TABLE, audits CREATE TABLE, ALTER TABLE, and DROP TABLE statements. However, the privilege auditing option, CREATE TABLE, audits only CREATE TABLE statements, because only the CREATE TABLE statement requires the CREATE TABLE privilege.

See the listing of system privileges in the GRANT SQL statement section of *Oracle Database SQL Language Reference*.

###### Configuring Privilege Auditing

Privilege audit options are the same as their corresponding system privileges. For example, the option to audit use of the DELETE ANY TABLE privilege is DELETE ANY TABLE.

[Example 9–11](#_bookmark1794) shows how to audit the DELETE ANY TABLE privilege.

***Example 9–11 Using AUDIT to Configure Privilege Auditing***

AUDIT DELETE ANY TABLE BY ACCESS;

To audit all successful and unsuccessful uses of the DELETE ANY TABLE system privilege, enter the following statement:

AUDIT DELETE ANY TABLE BY ACCESS;

###### Removing Privilege Auditing

The following statement removes all privilege audit options:

NOAUDIT ALL PRIVILEGES;

This example disables the audit settings from [Example 9–11](#_bookmark1794):

NOAUDIT DELETE ANY TABLE;

To disable privilege auditing options, you must have the AUDIT SYSTEM system privilege. Usually, only the security administrator is granted this system privilege.

#### Auditing SQL Statements and Privileges ina Multitier Environment

You can use the AUDIT statement to audit the activities of a client in a multitier environment. In a multitier environment, Oracle Database preserves the identity of a client through all tiers. Thus, you can audit actions taken on behalf of the client by a middle-tier application, by using the BY *user* clause in your AUDIT statement. The audit applies to all user sessions, including proxy sessions.

The middle tier can also set the user client identity in a database session, enabling the auditing of end-user actions through the middle-tier application. The end-user client identity then shows up in the audit trail.

[Example 9–12](#_bookmark1802) shows how to audit SELECT TABLE statements issued by the user

jackson.

***Example 9–12 Using AUDIT to Audit a SQL Statement for a User***

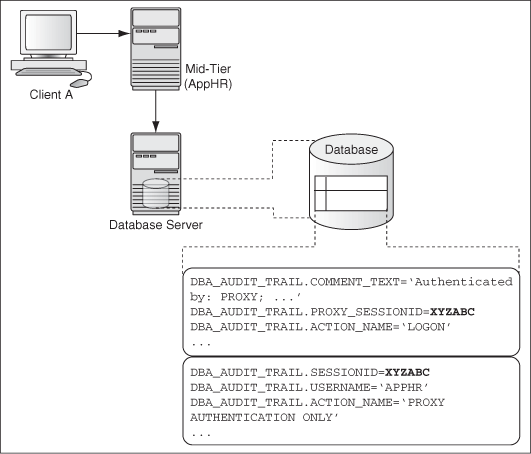
AUDIT SELECT TABLE BY jackson;

You can audit user activity in a multitier environment. Once audited, you can verify these activities by querying the DBA\_AUDIT\_TRAIL data dictionary view.

[Figure 9–1](#_bookmark1804) illustrates how you can audit proxy users by querying the COMMENT\_TEXT, PROXY\_SESSIONID, ACTION\_NAME, and SESSION\_ID columns of the DBA\_AUDIT\_TRAIL

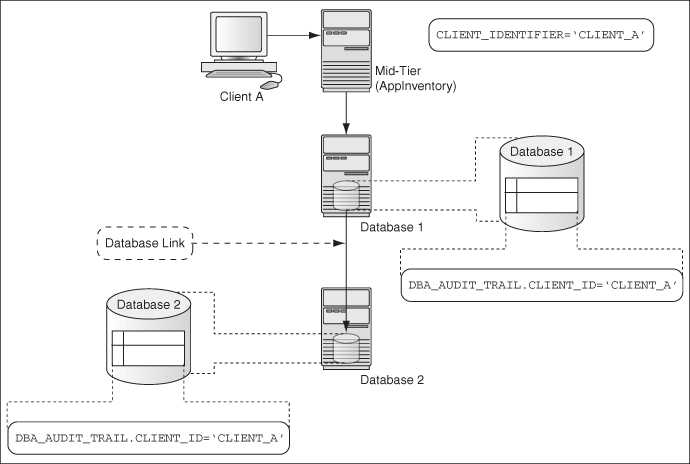
view. In this scenario, both the database user and proxy user accounts are known to the database. Session pooling can be used.

***Figure 9–1 Auditing Proxy Users***



[Figure 9–2](#_bookmark1806) illustrates how you can audit client identifier information across multiple database sessions by querying the CLIENT\_ID column of the DBA\_AUDIT\_TRAIL data dictionary view. In this scenario, the client identifier has been set to CLIENT\_A. As with the proxy user-database user scenario described in [Figure 9–1](#_bookmark1804), session pooling can be used.

***Figure 9–2 Auditing Client Identifier Information Across Sessions***



**See Also:** ["Preserving User Identity in Multitiered Environments" on](#_bookmark414) [page 3-36](#_bookmark414) for more information about user authentication in a multitiered environment

#### Auditing Schema Objects

This section contains:

* + - [About Schema Object Auditing](#_bookmark1809)
    - [Types of Schema Objects That Can Be Audited](#_bookmark1814)
    - [Using Standard Auditing with Editioned Objects](#_bookmark1819)
    - [Schema Object Audit Options for Views, Procedures, and Other Elements](#_bookmark1821)
    - [Configuring Schema Object Auditing](#_bookmark1823)
    - [Removing Object Auditing](#_bookmark1833)
    - [Setting Audit Options for Objects That May Be Created in the Future](#_bookmark1837)

###### About Schema Object Auditing

Schema object auditing monitors actions performed on the audited schema objects, such as tables or views. Object auditing applies to all users but is limited to the audited object only. Users can use the AUDIT and NOAUDIT statements on objects in their own schemas.

###### Types of Schema Objects That Can Be Audited

You can audit statements that refer to tables, views, sequences, standalone stored procedures or functions, and packages, but not individual procedures within packages. (See ["Auditing Functions, Procedures, Packages, and Triggers" on page 9-32](#_bookmark1848) for more information about auditing these types of objects.)

You cannot directly audit statements that reference clusters, database links, indexes, or synonyms. However, you can indirectly audit access to these schema objects, by auditing the operations that affect the base table.

When you audit a schema object, the auditing applies to all users of the database. You cannot set these options for a specific list of users. You can set default schema object audit options for all auditable schema objects.

**See Also:** *Oracle Database SQL Language Reference* for information about auditable schema objects

###### Using Standard Auditing with Editioned Objects

When an editioned object has an audit policy, then it applies in all editions in which the object is visible. When an editioned object is actualized, any audit policies that are attached to it are newly attached to the new actual occurrence. When you newly apply an audit policy to an inherited editioned object, this action will actualize it.

You can find the editions in which audited objects appear by querying the OBJECT\_ NAME and OBJ\_EDITION\_NAME columns in the DBA\_COMMON\_AUDIT\_TRAIL and V$XML\_ AUDIT\_TRAIL data dictionary views.

**See Also:** *Oracle Database Advanced Application Developer's Guide* for detailed information about editions

###### Schema Object Audit Options for Views, Procedures, and Other Elements

The definitions for views and procedures (including stored functions, packages, and triggers) reference underlying schema objects. Because of this dependency, some unique characteristics apply to auditing views and procedures, such as the likelihood of generating multiple audit records.

Views and procedures are subject to the enabled audit options on the base schema objects, including the default audit options. These options also apply to the resulting SQL statements.

Consider the following series of SQL statements:

AUDIT SELECT ON HR.EMPLOYEES BY ACCESS;

CREATE VIEW employees\_departments AS

SELECT employee\_id, last\_name, department\_id FROM employees, departments

WHERE employees.department\_id = departments.department\_id; AUDIT SELECT ON employees\_departments BY ACCESS;

SELECT \* FROM employees\_departments;

As a result of the query on the employees\_departments view, two audit records are generated: one for the query on the employees\_departments view and one for the query on the base table employees (indirectly through the employees\_departments view). The query on the base table departments does not generate an audit record because the SELECT audit option for this table is not enabled. All audit records pertain to the user that queried the employees\_departments view.

In the given example, if the AUDIT SELECT ON HR.EMPLOYEES; statement is omitted, then using the employees\_departments view does not generate an audit record for the EMPLOYEES table.

###### Configuring Schema Object Auditing

You can use the AUDIT statement to configure object auditing in the current edition. *Oracle Database SQL Language Reference* lists valid object audit options for AUDIT and the schema object types for which each option is available.

A user can set any object audit option for the objects contained in his or her schema. The AUDIT ANY system privilege is required to set an object audit option for an object contained in another user schema or to set the default object auditing option. Usually, only the security administrator is granted the AUDIT ANY privilege.

[Figure 9–13](#_bookmark1826) shows how to audit all successful and unsuccessful DELETE statements on the laurel.emp table.

***Example 9–13 Configuring Auditing for a Schema Table***

AUDIT DELETE ON laurel.emp BY ACCESS;

[Example 9–14](#_bookmark1827) shows how to audit all successful SELECT, INSERT, and DELETE

statements on the dept table owned by user jward.

***Example 9–14 Auditing Successful Statements on a Schema Table***

AUDIT SELECT, INSERT, DELETE

ON jward.dept BY ACCESS

WHENEVER SUCCESSFUL;

[Example 9–15](#_bookmark1830) shows how you can use the ON DEFAULT clause to apply to any new objects (tables, views, and sequences) that are created after you set the AUDIT statement. In this example, new objects that are created in the future will be audited for all unsuccessful SELECT statements:

***Example 9–15 Configuring Auditing for Any New Objects Using the DEFAULT Clause***

AUDIT SELECT

ON DEFAULT

BY ACCESS

WHENEVER NOT SUCCESSFUL;

[Example 9–16](#_bookmark1832) shows how to audit the execution of PL/SQL procedure or function.

***Example 9–16 Auditing the Execution of a Procedure or Function***

AUDIT EXECUTE ON sec\_mgr.auth\_orders BY ACCESS;

###### Removing Object Auditing

Use the NOAUDIT statement to remove object auditing. The following statements turn off the corresponding auditing options:

NOAUDIT DELETE

ON emp;

NOAUDIT SELECT, INSERT, DELETE

ON jward.dept;

To remove all object audit options on the emp table, enter the following statement:

NOAUDIT ALL ON emp;

To remove all default object audit options, enter the following statement:

NOAUDIT ALL ON DEFAULT;

All schema objects that are created before this NOAUDIT statement is issued continue to use the default object audit options in effect at the time of their creation, unless overridden by an explicit NOAUDIT statement after their creation.

To remove object audit options for a specific object, you must be the owner of the schema object. To remove the object audit options of an object in the schema of another user or to remove default object audit options, you must have the AUDIT ANY system privilege. A user with privileges to remove object audit options of an object can override the options set by any user.

###### Setting Audit Options for Objects That May Be Created in the Future

You can create audit settings for objects that do not exist yet, such as the insertion and deletion of tables to be created. There are two approaches that you can take. One approach is to use the statement audit options in the AUDIT statement. For example, to audit all inserts on future tables, enter the following SQL statement:

AUDIT INSERT TABLE BY ACCESS;

The second approach is to invoke the AUDIT statement using the ON DEFAULT clause. For example:

AUDIT ALL ON DEFAULT BY ACCESS;

This statement audits by default all subsequent object creation statements. The ON keyword specifies object auditing. The ON DEFAULT clause configures auditing for subsequently created objects that are affected by the following statements:

|  |  |  |  |
| --- | --- | --- | --- |
| **Statements A-D** | **Statements E-I** | **Statements I-R** | **Statements S-Y** |
| ALTER | EXECUTE | INSERT | SELECT |
| AUDIT | GRANT | LOCK | UPDATE |
| COMMENT | FLASHBACK | READ | - |

|  |  |  |  |
| --- | --- | --- | --- |
| **Statements A** | **-D Statements** | **E-I Statements I-** | **R Statements S-Y** |
| DELETE | INDEX | RENAME | - |

To restrict ON DEFAULT to a specific action, enter a statement similar to the following:

AUDIT ALTER, DELETE ON DEFAULT BY ACCESS;

For more information about the audit options and the ON DEFAULT clause of the AUDIT SQL statement, see *Oracle Database SQL Language Reference*. To find objects audited by default, query the ALL\_DEF\_AUDIT\_OPTS data dictionary view.

#### Auditing Directory Objects

This section contains:

* + [About Directory Object Auditing](#_bookmark1841)
  + [Configuring Directory Object Auditing](#_bookmark1843)
  + [Removing Directory Object Auditing](#_bookmark1846)

###### About Directory Object Auditing

You can audit directory objects. For example, suppose you create a directory object that contains a preprocessor program that the ORACLE\_LOADER access driver will use. You can audit anyone who runs this program within this directory object.

###### Configuring Directory Object Auditing

Use the AUDIT statement to enable object auditing. [Example 9–17](#_bookmark1845) shows how to audit the EXECUTE privilege on the directory object my\_exec.

***Example 9–17 Auditing a Directory Object***

AUDIT EXECUTE ON DIRECTORY my\_exec BY ACCESS;

###### Removing Directory Object Auditing

Use the NOAUDIT statement to disable directory object auditing. For example:

NOAUDIT EXECUTE ON DIRECTORY my\_exec;

#### Auditing Functions, Procedures, Packages, and Triggers

This section contains:

* + [About Auditing Functions, Procedures, Packages, and Triggers](#_bookmark1849)
  + [Configuring the Auditing of Functions, Procedures, Packages, and Triggers](#_bookmark1860)
  + [Removing the Auditing of Functions, Procedures, Packages, and Triggers](#_bookmark1866)

###### About Auditing Functions, Procedures, Packages, and Triggers

You can audit functions, procedures, PL/SQL packages, and triggers. The areas that you can audit are as follows:

* + You can individually audit standalone functions, standalone procedures, and PL/SQL packages.
  + If you audit a PL/SQL package, Oracle Database audits all functions and procedures within the package.
    - If you enable auditing for all executions, Oracle Database audits all triggers in the database, as well as all the functions and procedures within PL/SQL packages.
    - You cannot audit individual functions or procedures within a PL/SQL package.

If you want to audit functions that are associated with Oracle Virtual Private database policies, note the following:

* + - **Dynamic policies:** Oracle Database evaluates the policy function twice, once during SQL statement parsing and again during execution. As a result, two audit records are generated for each evaluation.
    - **Static policies:** Oracle Database evaluates the policy function once and then caches it in the SGA. As a result, only one audit record is generated.
    - **Context-sensitive policies:** Oracle Database executes the policy function once, during statement parsing. As a result, only one audit record is generated.

###### Configuring the Auditing of Functions, Procedures, Packages, and Triggers

[Example 9–18](#_bookmark1863) shows how to audit the execution of any function, procedure, package, or trigger, by any user in the database.

***Example 9–18 Auditing All Functions, Procedures, Packages, and Triggers***

AUDIT EXECUTE PROCEDURE BY ACCESS;

[Example 9–19](#_bookmark1864) shows how to audit user psmith’s successful and unsuccessful executions of functions, procedures, packages, and triggers.

***Example 9–19 Auditing a User’s Execution of Functions, Procedures, Packages, and Triggers***

AUDIT EXECUTE PROCEDURE BY psmith BY ACCESS;

[Example 9–20](#_bookmark1865) shows how to audit a standalone procedure entitled check\_work that is in the sales\_data schema. This idea applies to standalone functions as well.

***Example 9–20 Auditing the Execution of a Procedure or Function within a Schema***

AUDIT EXECUTE ON sales\_data.check\_work BY ACCESS WHENEVER SUCCESSFUL;

###### Removing the Auditing of Functions, Procedures, Packages, and Triggers

Use the NOAUDIT statement to remove the auditing of functions, procedures, and triggers. For example:

NOAUDIT EXECUTE PROCEDURE;

NOAUDIT EXECUTE PROCEDURE BY psmith;

NOAUDIT EXECUTE ON sales\_data.checkwork;

#### Auditing Network Activity

This section contains:

* + - [About Network Auditing](#_bookmark1868)
    - [Configuring Network Auditing](#_bookmark1874)
    - [Removing Network Auditing](#_bookmark1876)

###### About Network Auditing

You can use the AUDIT statement to audit unexpected errors in network protocol or internal errors in the network layer. Network auditing captures errors that occur during communication with the client on the network. These are errors thrown by the SQL\*Net driver. There can be several causes of network errors. For example, an internal event set by a database engineer for testing purposes can cause a network error. Other causes include conflicting configuration settings for encryption, such as the network not finding the information required to create or process expected encryption. The errors that network auditing uncovers (such as ACTION 122 Network Error) are not connection failures: network auditing is only concerned with data as it travels across the network.

The audit record for network auditing lists the authentication type and SQL\*Net address of the client (if available) in the COMMENT\_TEXT field of the audit record, using the following format:

Authenticated by: *authentication\_type*; Client Address: *SQLNetAddress\_of\_client*

The Client Address: *SQLNetAddress\_of\_client* portion only appears if this information is available.

[Table 9–4](#_bookmark1873) shows how to remedy four common error conditions.

***Table 9–4 Auditable Network Error Conditions***

|  |  |  |
| --- | --- | --- |
| **Error** | **Cause** | **Action** |
| TNS-02507  Encryption algorithm not installed | After picking an algorithm, the server was unable to find an index for it in its table of algorithms. This should be impossible because the algorithm was chosen (indirectly) from that list. | Turn on tracing for further details, and then rerun the operation. (Note that this error is not normally visible to the user.) If the error persists, then contact Oracle Support Services. |
| TNS-12648  Encryption or data integrity algorithm list empty | An Oracle Advanced Security list-of-algorithms parameter was empty. | Change the list to contain the name of at least one installed algorithm, or remove the list entirely if every installed algorithm is not acceptable. |
| TNS-12649  Unknown encryption or data integrity algorithm | An Oracle Advanced Security  list-of-algorithms parameter included an algorithm name that was not recognized. | Remove that algorithm name, correct it if it was misspelled, or install the driver for the missing algorithm. |
| TNS-12650  No common encryption or data integrity algorithm | The client and server have no algorithm in common for either encryption or data integrity or both. | Choose sets of algorithms that overlap. In other words, add one of the client algorithm choices to the server list, or add one of the server list choices to the client algorithm. |

###### Configuring Network Auditing

To configure network auditing, use the AUDIT statement. For example:

AUDIT NETWORK BY ACCESS;

###### Removing Network Auditing

To remove network auditing:

NOAUDIT NETWORK;

### Using Default Auditing for Security-Relevant SQL Statements and Privileges

This section contains:

* + - [About the Default Auditing Settings](#_bookmark1882)
    - [Privileges That Oracle Database Audits by Default](#_bookmark1887)
    - [Disabling and Enabling Default Audit Settings](#_bookmark1890)

#### About the Default Auditing Settings

When you use Database Configuration Assistant (DBCA) to create a new database, Oracle Database configures the database to audit the most commonly used security-relevant SQL statements and privileges. It also sets the AUDIT\_TRAIL initialization parameter to DB. If you decide to use a different audit trail type (for

example, OS if you want to write the audit trail records to operating system files), then you can do that: Oracle Database continues to audit the privileges that are audited by default. If you disable auditing by setting the AUDIT\_TRAIL parameter to NONE, then no auditing takes place.

If you manually create a database, then you should run the secconf.sql script to apply the default audit settings to your database. See ["Disabling and Enabling Default](#_bookmark1890) [Audit Settings" on page 9-36](#_bookmark1890) for more information.

To individually control the auditing of SQL statements and privileges, use the AUDIT and NOAUDIT statements. For more information, see ["Auditing SQL Statements" on](#_bookmark1758) [page 9-23](#_bookmark1758) and ["Auditing Privileges" on page 9-26](#_bookmark1784).

#### Privileges That Oracle Database Audits by Default

Oracle Database audits the following privileges by default:

|  |  |  |
| --- | --- | --- |
| **Privileges A-C** | **Privileges C-D** | **Privileges D-G** |
| ALTER ANY PROCEDURE | CREATE ANY LIBRARY | DROP ANY TABLE |
| ALTER ANY TABLE | CREATE ANY PROCEDURE | DROP PROFILE |
| ALTER DATABASE | CREATE ANY TABLE | DROP USER |
| ALTER PROFILE | CREATE EXTERNAL JOB | EXEMPT ACCESS POLICY |
| ALTER SYSTEM | CREATE PUBLIC DATABASE LINK | GRANT ANY OBJECT PRIVILEGE |
| ALTER USER | CREATE SESSION | GRANT ANY PRIVILEGE |
| AUDIT SYSTEM | CREATE USER | GRANT ANY ROLE |
| CREATE ANY JOB | DROP ANY PROCEDURE | - |

Oracle Database audits the following SQL shortcuts by default:

|  |  |  |
| --- | --- | --- |
| **Shortcuts D-P** | **Shortcuts P-R** | **Shortcuts S** |
| DATABASE LINK | PUBLIC SYNONYM | SYSTEM AUDIT |
| PROFILE | ROLE | SYSTEM GRANT |

**See Also:**

* *Oracle Database SQL Language Reference* for detailed information about the SQL statements described in this section

sql\_statement\_shortcut in *Oracle Database SQL Language Reference* for a list of accepted SQL shortcuts you can use with the AUDIT statement

#### Disabling and Enabling Default Audit Settings

If your applications use the default audit settings from Oracle Database 10*g* Release 2 (10.2), then you can revert to these audit settings until you modify the applications to use the Release 11*g* audit settings. To do so, run the undoaud.sql script.

After you have modified your applications to conform to the Release 11*g* audit settings, then you can manually update your database to use the audit configuration that suits your business needs, or you can run the secconf.sql script to apply the Release 11*g* default audit settings. You can customize this script to have different security settings if you like, but remember that the settings listed in the original script are Oracle-recommended settings.

If you created your database manually, then you should run the secconf.sql script to apply the Release 11*g* default audit settings to the database. Databases that have been created with Database Configuration Assistant will have these settings, but manually created databases do not.

The undoaud.sql and secconf.sql scripts are in the $ORACLE\_HOME/rdbms/admin directory. The undoaud.sql script affects audit settings only, and the secconf.sql script affects both audit and password settings. They have no effect on other security settings.

### Auditing Specific Activities with Fine-Grained Auditing

This section contains:

* + [About Fine-Grained Auditing](#_bookmark1895)
  + [Where Are Fine-Grained Audit Records Stored?](#_bookmark1902)
  + [Advantages of Fine-Grained Auditing](#_bookmark1904)
  + [What Permissions Are Needed to Create a Fine-Grained Audit Policy?](#_bookmark1907)
  + [Activities That Are Always Audited in Fine-Grained Auditing](#_bookmark1909)
  + [Using Fine-Grained Audit Policies with Editions](#_bookmark1912)
  + [Creating an Audit Trail for Fine-Grained Audit Records](#_bookmark1914)
  + [How the Fine-Grained Audit Trail Generates Records](#_bookmark1916)
  + [Using the DBMS\_FGA Package to Manage Fine-Grained Audit Policies](#_bookmark1918)
  + [Tutorial: Adding an Email Alert to a Fine-Grained Audit Policy](#_bookmark1942)
  + [Tutorial: Auditing Nondatabase Users](#_bookmark1960)

#### About Fine-Grained Auditing

Fine-grained auditing enables you to create policies that define specific conditions that must take place for the audit to occur. This enables you to monitor data access based on content. It provides granular auditing of queries, and INSERT, UPDATE, and DELETE operations. For example, a central tax authority must track access to tax returns to guard against employee snooping, with enough detail to determine what data was accessed. It is not enough to know that SELECT privilege was used by a specific user on a particular table. Fine-grained auditing provides this deeper functionality.

In general, fine-grained audit policies are based on simple, user-defined SQL predicates on table objects as conditions for selective auditing. During fetching, whenever policy conditions are met for a row, the query is audited.

You can use fine-grained auditing to audit the following types of actions:

* + - Accessing a table between 9 p.m. and 6 a.m. or on Saturday and Sunday
    - Using an IP address from outside the corporate network
    - Selecting or updating a table column
    - Modifying a value in a table column

**Note:**

* Fine-grained auditing is supported only with cost-based optimization. For queries using rule-based optimization, fine-grained auditing checks before applying row filtering, which could result in an unnecessary audit event trigger.
* Policies currently in force on an object involved in a flashback query are applied to the data returned from the specified flashback snapshot (based on time or system change number (SCN).
* If you want to use fine-grained auditing to audit data that is being directly loaded (for example, using Oracle Warehouse Builder to execute DML statements), then Oracle Database transparently makes all direct loads that are performed in the database instance into conventional loads. If you want to preserve the direct loading of data, consider using standard auditing instead.

#### Where Are Fine-Grained Audit Records Stored?

Fine-grained audit records are stored in the SYS.FGA\_LOG$ table. To find the records have been generated for the audit policies that are in effect, you can query the DBA\_ FGA\_AUDIT\_TRAIL data dictionary view. The DBA\_COMMON\_AUDIT\_TRAIL data dictionary view combines both standard and fine-grained audit log records. In addition, you can query the V$XML\_AUDIT\_TRAIL view to find fine-grained audit records that were written in XML formatted files. For detailed information about these views, see *Oracle Database Reference*.

Oracle Database always audits DELETE, INSERT, UPDATE, and MERGE operations on the SYS.FGA\_LOG$ (and SYS.AUD$) tables to the SYS.AUD$ table. It does not allow the audit records to be deleted, unless user SYS performs these operations. If you have set the AUDIT\_SYS\_OPERATIONS initialization parameter to TRUE, then user SYS's operations are audited. In this case the audit records of all SYS operations are written to whatever

directory the AUDIT\_FILE\_DEST initialization parameter points to. If AUDIT\_FILE\_DEST

is not set, then it writes the records to an operating system-dependent location.

#### Advantages of Fine-Grained Auditing

Fine-grained auditing creates a more meaningful audit trail, one that includes only very specific actions that you want to audit. It excludes unnecessary information that occurs if each table access was recorded. Fine-grained auditing has the following advantages over standard auditing:

* + **It performs a Boolean condition check.** If the Boolean condition you specify is met, for example, a table being accessed on a Saturday, then the audit takes place.
  + **It captures the SQL that triggered the audit.** You can capture both the SQL statement that caused the audit, and any associated bind variables. Be aware that you can only capture data from scalar column types. You cannot capture data from object columns, LOBs, or user-defined column types. For example, suppose you have the following query:

SELECT NAME FROM EMPLOYEE WHERE SSN = :1

If :1 is of integer type and the value for SSN is 987654321, then the audit trail can capture this information. However, the audit trail cannot capture this information if :1 is a BLOB, CLOB, object, or user-defined type.

This feature is available if you create the fine grained audit policy with the audit\_ trail parameter of the DBMS\_FGA.ADD\_POLICY PL/SQL procedure to DB+EXTENDED or XML+EXTENDED.

* + **It adds extra protection to sensitive columns.** You can audit specific relevant columns that may hold sensitive information, such as salaries or Social Security numbers.
  + **It provides an event handler feature.** For example, you can write a function that sends an email alert to a security administrator when an audited column that should not be changed at midnight is updated.
  + **You do not need to set initialization parameters to enable fine-grained auditing.** Instead of setting initialization parameters such as AUDIT\_TRAIL, you use the DBMS\_ FGA PL/SQL package to add and remove fine-grained auditing policies as necessary applying them to the specific operations or objects you want to monitor.

#### What Permissions Are Needed to Create a Fine-Grained Audit Policy?

To create a fine-grained audit policy, you must have EXECUTE privileges on the DBMS\_ FGA PL/SQL package. The package is owned by the SYS user.

#### Activities That Are Always Audited in Fine-Grained Auditing

The SYS.AUD$ table records all data manipulation language (DML) statements, such as INSERT, UPDATE, MERGE, and DELETE, that are performed on the SYS.FGA\_LOG$ table by non-SYS users. Oracle Database performs the audit even if auditing has not been configured for the SYS.FGA\_LOG$ table, which is the table in which these activities occur. You can check these activities by querying the DBA\_FGA\_AUDIT\_TRAIL and DBA\_ COMMON\_AUDIT\_TRAIL views. See also ["Activities That Are Always Written to the](#_bookmark1614) [Standard and Fine-Grained Audit Records" on page 9-3](#_bookmark1614).

#### Sử dụng Fine-Grained Audit Policies với Editions

Nếu bạn đang chuẩn bị 1 ứng dụng để xác định lại edition-base, và bao gồm mỗi bảng mà ứng dụng sử dụng với chế độ xem ấn bản, sau đó bản phải chuyển chế độ fine-grained audit để bảo về những bảng này dưới chế độ xem ấn bản

#### Tạo một Audit Trail cho Fine-Grained Audit Records

Bạn chỉ định 1 audit trail để định dạng cho fine-grained auditing bằng cách đặt tham số audit\_trail cho chính sách DBMS\_FGA.ADD\_POLICY (không được bối rối bởi việc khởi tạo tham số AUDIT\_TRAIL) khi bạn tạo chính sách kiểm toán. Đặt tham số này thành XML hoặc XML+ EXTENDED ghi các bản ghi vào các tệp hệ điều hành định dạng XML. Nếu bạn thích ghi các bản ghi fine-grained audit vào bảng SYS.FGA\_LOG$ hơn, thì đặt tham số audit\_trail cho DBMS\_FGA.ADD\_POLICY thành DB or DB+EXTENDED. có nhiều lí lo giải thích vì sao viết audit records vào file hệ thống lại lợi ích hơn, xem them ["Advantages of the Operating System Audit Trail" tại trang 9-16](#_bookmark1707).

Bạn có thể sử dụng V$XML\_AUDIT\_TRAIL chế độ xem từ điển dữ liệu để tạo Audit Records từ tệp XML có sẵn cho DBAs thông qua câu truy cấn SQL, tăng cường khả năng sử dụng. truy vấn view này làm cho tất cả tệp XML (tất các các tệp đuôi .xmlextendsion) trong thư mục AUDIT\_FILE\_DEST được phân tích và trình bày trong quan hệ các bảng.

View DBA\_COMMON\_AUDIT\_TRAIL bao gồm nội dung của bảng động tiêu chuẩn V$XML\_AUDIT\_TRAIL và fine\_grained audit records

Vì tệp kiểm toán XML được chứa trong nhiều tệp đuôi .xmlextendsion trên tất cả các nền tảng, view động cho chúng ta ta thông tin kiểm toán tương tự trên tất cả nền tảng. Xem thêm Oracle Database Reference để biết thêm thông tin chi tiết về view V$XML\_AUDIT\_TRAIL.

**Chú ý:** Nếu bạn kiểm toán các bảng có chưa dữ liệu nhạy cảm, nhớ cài đặt DB+EXTEND và XML+EXTENDED cho tham số DBMS\_FGA.ADD\_POLICY audit\_trail để bắt được dữ liệu này. Xem thêm ["Auditing Sensitive](#_bookmark2288)” trang 10-18 để biết thêm những cách xử lí khác.

#### Cách Fine-Grained Audit khởi tạo bảng ghi

The fine-grained audit trail bắt dữ liệu kiểm toán từ 1 bản hoặc 1 view trong câu 1 câu lệnh SQL. Ví dụ, nếu bạn chạy 1 câu lệnh UNION thể hiện bảng HR.EMPLOYEE 2 lần, thì câu lệnh kiểm toán sẽ khởi tạo 2 bảng ghi kiểm toán, mỗi cái đề chó thể truy cập bảng HR.EMPLOYEE

#### Sử dụng gói DBMS\_FGA để quản lí Fine-Grained Audit Policies

Phần này chứa:

* + - [thông in về gói DBMS\_FGA PL/SQL](#_bookmark1919)
    - [tạo 1 Fine-Grained Audit Policy](#_bookmark1921)
    - [kích hoạt và dừng 1 Fine-Grained Audit Policy](#_bookmark1934)
    - [xóa 1 Fine-Grained Audit Policy](#_bookmark1939)

###### thông tin về gói DBMS\_FGA PL/SQL

Để quản lí 1 fine-grained audit policy, bạn dử dụng gói DBMS\_FGA PL/SQL. Gói này cho phép bạn thêm tất cả sự kết hợp của câu lệnh SELECT, INSERT, UPDATE, và DELETE. bạn cũng có thể gộp các câu lệnh lại bằng cách kiểm toán các hành động nền tảng của INSERT và UPDATE. Để gộp các câu lệnh, cấu hình fine-grained truy cập trên câu lệnh INSERT và UPDATE. Chỉ 1 bảng ghi được tạo ra mỗi khi hoạt động gộp thành công. Để quản lí fine-grained audit policies, bạn phải có quyền thực thi trên gói DBMS\_FGA.

Chính sách kiểm toán bị ràng buộc vào bảng bạn đã tạo ra. Việc này đơn giản hóa việc quản lí các chính sách kiểm toán bởi vì chính sách chỉ phải được thay đổi 1 lần trong database, không phải trong ứng dụng. khi thêm vào, không cần biết người dùng kết nối với database như thế nào- từ ứng dụng, web, hoặc thông qua SQL\*plus hoặc Oracle SQL Developer- Oracle Database ghi lại bất kì hành động nào ảnh hưởng đến chính sách.

Nếu bất cứ dòng nào được trả về từ 1 câu truy vấn trùng khớp với điều kiện kiểm toán mà bạn định nghĩa, thì Oracle Database thêm 1 đường đẫn kiểm toán vào cái fine-grained audit trail. Đường dẫn này không bao gồm tất cả thông tin được báo cáo ở đường mòn kiểm toán thường xuyên. Nói cách khác, chỉ 1 dòng của thông tin kiểm toán được thêm vào audit trail cho tất cả fine-grained audit policy được đánh giá đúng.

Thông tin chi tiết về cú pháp của gói DBMS\_FGA, xem thêm *Oracle Database PL/SQL Packages and Types Reference*. Xem luôn *Oracle Database Advanced Application Developer's Guide*.

**Chú ý:** Nếu bạn định sử dụng gói DBMS\_FGA trên các phiên bản khác nhau, thì bạn có thể sử dụng kết quả của chính sách: liệu tất cả kết quả có đồng bộ trên tất cả phiên bản hay không, hoặc cụ thể trên 1 phiên bả mà chính sách đã được sử dụng. Xem thêm ["How Editions Affects the Results of a](#_bookmark1265) [Global Application Context PL/SQL Package" trên trang 6-24](#_bookmark1265) để biết thêm thông tin.

###### Tạo 1 Fine-Grained Audit Policy

Để tạo 1 fine-grained audit policy, sử dụng thủ tục DBMS\_FGA.ADD\_POLICY. Thủ tục này tạo 1 chính sách kiểm toán sử dụng vị ngữ được cung cấp như là điều kiện kiểm toán. Oracle Database thực thi chính sách định vị dưới nền tảng của người dùng người đã tạo ra chính sách. Con số lớn nhất của fine-grained policies trên bất kì 1 bảng nào hoặc bảng xem nào là 256. Oracle Database chưa chính sách trong bảng thư viện dữ liệu, nhưng ban có thể tạo ra chính sách trên bất kì bảng nào hoặc view không nằm trong SYSchema.

Sau khi bạn tạo fine-grained audit policy, nó không cứ trụ tại bất kì lược đồ cụ thể nào, mặc dù đích đến của chính sách là được chứa trong SYS.FGA$ bảng thư viện dữ liệu.

Bạn không thể chỉnh sửa 1 fine-grained audit policy sau khi bạn đã tạo ra nó. Nếu bạn cần chỉnh sửa chính sách, xóa nó và tạo lại cái mới.

Hãy cẩn thận nếu 1 cột bảng có fine-grained audit policy, bạn không thể mã hóa hoặc giải mã cột này (bằng việc sử dụng câu lệnh UPDATE). Để làm như vậy hãy đặt ra 1 ORA-28133:fulltableaccessisrestrictedbyfine-grainedsecurityerror. Nếu bạn muốn cập nhật cột đó, tạm thời tắt fine-grained audit policy sau đó mã hóa hoặc giải mã cột đó. Sau cùng, tái kích hoạt fine-grained audit policy. Xem thêm ["Disabling and Enabling a Fine-Grained Audit Policy" trên trang 9-43](#_bookmark1934) để biết thêm thông tin.

Cú pháp tạo thủ tục ADD\_POLICY là:

DBMS\_FGA.ADD\_POLICY(

object\_schema VARCHAR2,

object\_name VARCHAR2,

policy\_name VARCHAR2, audit\_condition VARCHAR2,

audit\_column VARCHAR2, handler\_schema VARCHAR2, handler\_module VARCHAR2, enable BOOLEAN, statement\_types VARCHAR2,

audit\_trail BINARY\_INTEGER IN DEFAULT, audit\_column\_opts BINARY\_INTEGER IN DEFAULT);

Cụ thể :

* + - object\_schema: Chỉ định cụ thể lược đồcủa đối tượng được kiểm toán (nếu Null, lược đồ đăng nhập hiện tại của người dùng được giả định)
    - object\_name: Chỉ định củ thể tên của đối tượng được kiểm toán.
    - policy\_name: Chỉ định cụ thể tên của chính sách được tạo ra. Đảm bảo tên này được duy nhất.
    - audit\_condition: Chỉ định cụ thể 1 điều kiện Boolean trong dòng. Null được cho phép và được hiểu như "TRUE". Xem thêm ["Auditing Specific Columns and Rows" trên trang 9-43](#_bookmark1932) để biết thêm thông tin. Nếu bạn chỉ định Null hoặc không có điều kiện kiểm toán nào, thì bất kì hành động nào trên bảng mà chính sách tạo 1 audit record, có hay không hàng đều được trả về.

Thực hiện theo các nguyên tắc sau:

* + - * Không bao gồm các hàm xử lí câu lệnh có thể kiểm toán trên cùng 1 bảng, trong cài đặt audit\_condition. Ví dụ, giả sử bạn tạo 1 hàm xử lí câu lệnh INSERT trên bảng HR.EMPLOYEE. Điều kiện kiểm toán của chính sách chứa hàm này và cái này dành cho câu lệnh INSERT (được đặt bởi loại câu lệnh). Khi chính sách được sử dụng, hàm thực thi đệ qui cho tới khi hẹ thống chạy hết bộ nhớ. Việc này có thể tạo ra lỗi ORA-1000:maximumopencursorexceeded hoặc ORA-0036:maximumnumberofrecursiveSQLLevels(50)exceeded.
    - audit\_column: 1 hoặc nhiều cột cụ thể kiểm toán, bao gồm cả những cột ẩn. Nếu đặt giá trị NULL hoặc bỏ qua, tất cả các cột đều được kiểm toán. Những cột này có thể bảo gồm Oracle Label Security các cột ẩn hoặc các cột loại đối tượng. Mặc định, NULL, nếu bất kì cột nào bị truy cập hoặc bị ảnh hưởng đều gây ra kiểm toán.
    - handler\_schema: Nếu 1 cảnh báo được sủ dụng làm cò để đáp lại khi chính sách bị vi phạm, cụ thể là tên của lược đồ có chứa tên của quản lí sự kiện. Mặc định, Null, sử dụng lược đồ hiện tại. Xem thêm ["Tutorial: Adding an Email Alert to a Fine-Grained Audit Policy" trang 9-44](#_bookmark1942).
    - handler\_module: Tên cụ thể của quản lí sự kiện. Bao gồm cả gói sự kiện bên trong nó. Hàm này chỉ được gọi sau khi hàng đầu tiên trùng khớp với điều kiện kiểm toán trong câu truy cấn được xử lí.

Làm theo các chỉ dẫn:

* + - * Không được tạo xử lí kiểm toán đệ qui fine-grained. Ví dụ, giả sử bạn tạo 1 xử lí thực thi câu lệnh INSERT trên bản HR.EMPLOYEES. Chính sách được liên kết với xử lí này là cho câu lệnh INSERT (được đặt bởi tham số statement-types). Khi chính sách được sử dụng, cái xử lí này thực thi đệ qui cho tới khi hệ thông chạy hết bộ nhớ. Việc này có thể dẫn đến lỗi ORA-100:maximumopencursorsexceeded hoặc ORA-00036:maximumnumberofrecursiveSQLlevels(50)exceeded.

**–** Không được ý kiến câu lệnh DBMS\_FGA.ENABLE\_POLICY hoặc DBMS\_FGA.DISABLE\_POLICY từ 1 chính sách xử lí. Thực hiện này có thể dẫn đến lỗi ORA-28144: Failed to execute fine-grained audit handler error.

* + enable: Cho phép hoặc vô hiệu chính sách đang sử dụng true hoặc false. Nếu bỏ qua, chính sách được cho chép. Mặc định là TRUE.
  + statement\_types: Câu lệnh SQL cụ thể sẽ được kiểm toán: INSERT, UPDATE, DELETE, hoặc SELECT. mặc định là SELECT.
  + audit\_trail: Đích đến cụ thể của bảng ghi kiểm toán fine-grained (DB or XML). Cũng cụ thể nếu chưa LSQLTEXT và LSQLBIND in FGA\_LOG$. Tuy nhiên, cẩn thận với dự liệu nhạy cảm, ví dụ như là thông tin tín dụng, có thể được ghi lại dưới dạng văn bản. Xem thêm ["Auditing Sensitive Information" trang 10-18](#_bookmark2288) để biết cách xử lí.

Nếu bạn đặt tham số audit\_trail sang XML, thì tệp XML sẽ được viết vào đường dẫn cụ thể bởi tham số khởi tạo AUDIT\_FILE\_DES.

Đối với dữ liệu chỉ được đọc, Orcle Databases viết fine-grained audit trail vào tệp XML.

* + audit\_column\_opts: Nếu bạn chỉ định hơn 1 cột trong tham số audit\_column, thì tham số sẽ xác định kiểm toán tất cả hoặc 1 cột cụ thể nào đó. Xem thêm ["Auditing Specific Columns and Rows" trang 9-43](#_bookmark1932) để biết thêm thông tin.

Xem thêm *Oracle Database PL/SQL Packages and Types Reference* để biết thêm chi tiết về cú pháp ADD\_POLICY

[Ví dụ 9–21](#_bookmark1931)  cho bạn thấy kiểm toán câu lệnh INSERT, UPDATE, DELETE và SELCET trên bảng HR.EMPLOYEES. Chú ý ví dụ này bỏ sót tham số audit\_column\_opts, vì nó không phải là tham số bắt buộc.

***Example 9–21 Using DBMS\_FGA.ADD\_POLICY to Create a Fine-Grained Audit Policy***

BEGIN

DBMS\_FGA.ADD\_POLICY(

object\_schema => 'HR',

object\_name => 'EMPLOYEES', policy\_name => 'chk\_hr\_employees', enable => TRUE,

statement\_types => 'INSERT, UPDATE, SELECT, DELETE', audit\_trail => DBMS\_FGA.DB);

END;

/

Ở đây, nếu bạn truy vấn bảng xem DBA\_AUDIT\_POLICIES, bạn sẽ tìm thấy 1 danh sách chính sách mới.

SELECT POLICY\_NAME FROM DBA\_AUDIT\_POLICIES;

POLICY\_NAME

------------------------------- CHK\_HR\_EMPLOYEES

Sau đó, any of the following SQL statements log an audit event record.

SELECT COUNT(\*) FROM HR.EMPLOYEES WHERE COMMISSION\_PCT = 20 AND SALARY > 4500; SELECT SALARY FROM HR.EMPLOYEES WHERE DEPARTMENT\_ID = 50;

DELETE FROM HR.EMPLOYEES WHERE SALARY > 1000000;

Kiểm tra các cột và hàng cụ thể

Bạn có thể tinh chỉnh hành vi kiểm toán bằng cách nhắm mục tiêu một cột cụ thể, được gọi là cột có liên quan, để được kiểm tra nếu một điều kiện được đáp ứng. Để thực hiện điều này, bạn sử dụng tham số audit\_column để chỉ định một hoặc nhiều cột nhạy cảm. Ngoài ra, bạn có thể kiểm tra dữ liệu trong các hàng cụ thể bằng cách sử dụng tham số audit\_condition để xác định điều kiện Boolean.

[Ví dụ 9–21](#_bookmark1931) [trên trang 9-42](#_bookmark1931) thực hiện kiểm toán nếu có ai trong Department 50 cố gắng truy cập cột salary và commission\_pcts.

audit\_condition => 'DEPARTMENT\_ID = 50', audit\_column => 'SALARY,COMMISSION\_PCT,'

Như bạn có thể thấy, tính năng này rất có lợi. Nó không chỉ cho phép bạn xác định các loại dữ liệu đặc biệt quan trọng để kiểm toán mà còn cung cấp khả năng bảo vệ tăng cường cho các cột chứa dữ liệu nhạy cảm, chẳng hạn như số an sinh xã hội, tiền lương, chẩn đoán bệnh nhân, v.v..

Nếu audit\_column sắp xếp hơn 1 cột, ban có thể sử dụng tham số audit\_column\_opts để các tham chiếu xác định liệu một câu lệnh được kiểm tra khi truy vấn tham chiếu đến bất kỳ cột nào được chỉ định trong tham số audit\_column hay chỉ khi tất cả các cột được tham chiếu. Ví dụ:

audit\_column\_opts => DBMS\_FGA.ANY\_COLUMNS, audit\_column\_opts => DBMS\_FGA.ALL\_COLUMNS,

Nếu bạn không chỉ định 1 cột liên quan thì việc kiểm tra sẽ áp dụng cho tất cả các cột

Để biết thêm thông tìn về tham số audit\_condition, audit\_column, and audit\_column\_ opts trong thủ tục DBMS\_FGA.ADD\_POLICY , xem thêm *Oracle Database PL/SQL Packages and Types Reference*. Xem thêm ghi chú sử dụng cho thủ tục ADD\_POLICY trong phần đó.

###### Vô hiệu hoát và Kích hoạt 1 Fine-Grained Audit Policy

Bạn có thể tắt chính sách kiểm tra chi tiết bằng cách sử dụng DBMS\_FGA.DISABLE\_POLICY  
thủ tục. Cú pháp cho DISABLE\_POLICY là:

DBMS\_FGA.DISABLE\_POLICY(

object\_schema VARCHAR2, object\_name VARCHAR2, policy\_name VARCHAR2 );

[Ví dụ 9–22](#_bookmark1936) cho thấy cách vô hiệu hóa chính sách kiểm toán fine-grained được tạo ra ở [ví dụ 9–21 trang 9-42](#_bookmark1931).

***Example 9–22 Disabling a Fine-Grained Audit Policy***

DBMS\_FGA.DISABLE\_POLICY(

object\_schema => 'HR',

object\_name => 'EMPLOYEES',

policy\_name => 'chk\_hr\_employees');

/

Để biết thêm thông tin cú pháp DISABLE\_POLICY, xem thêm *Oracle Database PL/SQL Packages and Types Reference*.

[Ví dụ 9–23](#_bookmark1938)  hiển thị cách kích hoạt lại chính sách chk\_hr\_emp bằng cách sử dụng thủ tục DBMS\_ FGA.ENABLE\_POLICY:

***Example 9–23 Enabling a Fine-Grained Audit Policy***

DBMS\_FGA.ENABLE\_POLICY(

object\_schema => 'HR',

object\_name => 'EMPLOYEES',

policy\_name => 'chk\_hr\_employees',

enable => TRUE);

/

Để biết thêm thông tin cú pháp ENABLE\_POLICY , cú pháp *Oracle Database PL/SQL Packages and Types Reference*.

###### Xáo 1 Fine-Grained Audit Policy

Cơ sở dữ liệu Oracle tự động giảm chính sách kiểm tra nếu bạn loại bỏ đối tượng được chỉ định trong tham số object\_name của thủ tục DBMS\_FGA.ADD\_POLICY hoặc nếu bạn xóa người dùng đã tạo chính sách kiểm tra.

[Ví dụ 9–24](#_bookmark1941) cho bạn thấy cách xóa 1 fine-grained audit policy bằng cách sử dụng thủ tục

DBMS\_FGA.DROP\_POLICY .

***Example 9–24 Dropping a Fine-Grained Audit Policy***

DBMS\_FGA.DROP\_POLICY(

object\_schema => 'HR',

object\_name => 'EMPLOYEES', policy\_name => 'chk\_hr\_employees');

See *Oracle Database PL/SQL Packages and Types Reference* for detailed information about the DROP\_POLICY syntax.

#### Hướng dẫn: thêm 1 email cảnh báo Fine-Grained Audit Policy

Phần này chứa:

* + Về phần hướng dẫn
  + [Bước 1: Cài đặt và cấu hình gói UTL\_MAIL PL/SQL](#_bookmark1950)
  + [Bước 2: Tạo](#_bookmark1952) tài khoản người dùng
  + [Bước 3: Cấu](#_bookmark1953) hình tệp danh sách truy cập cho dịch vụ mạng
  + [Bước 4: Tạo](#_bookmark1954) thủ tục PL/SQL cảnh báo bảo mật email
  + [Bước 5: Tạo và kiểm thử cài đặt Fine-Grained Audit Policy](#_bookmark1955)
  + [Bước 6:](#_bookmark1958) Kiểm thử cảnh báo
  + [Bước 7: Remove the Components for This Tutorial](#_bookmark1959)

###### About This Tutorial

Bạn có thể thêm cảnh báo email vào chính sách kiểm tra chi tiết có hiệu lực khi người dùng (hoặc kẻ xâm nhập) vi phạm chính sách. Để thực hiện điều này, trước tiên bạn phải tạo một thủ tục tạo cảnh báo và sau đó sử dụng các tham số DBMS\_FGA.ADD\_POLICY sau đây để gọi hàm này khi ai đó vi phạm chính sách này:

* + handler\_schema: Giản đồ trong đó sự kiện trình xử lý được lưu trữ
    - handler\_module: tên của xử lí sự kiện

Cảnh báo có thể có bất kỳ hình thức nào phù hợp nhất với môi trường của bạn: email hoặc thông báo máy nhắn tin, cập nhật cho một tệp hoặc bảng cụ thể, v.v. Tạo cảnh báo cũng giúp đáp ứng các quy định tuân thủ nhất định, chẳng hạn như Dự luật Thượng viện California 1386. Trong hướng dẫn này, bạn sẽ tạo cảnh báo qua email.

Trong hướng dẫn này, bạn tạo một cảnh báo email thông báo cho một quản trị viên bảo mật rằng một đại diện Nhân sự đang cố gắng chọn hoặc sửa đổi thông tin về lương trong bảng HR.EMPLOYEES. Người đại diện được phép thực hiện các thay đổi đối với bảng này, nhưng để đáp ứng các quy định tuân thủ, chúng tôi muốn tạo một bản ghi về tất cả các lựa chọn lương và các sửa đổi đối với bảng.

###### Bước 1: Cài đặt và cấu hình gói UTL\_MAIL PL/SQL

1. đăng nhập người dùng SYS dưới quyền SYSDBA.

sqlplus sys as sysdba Enter password: *password*

1. cài đặt gói UTL\_MAIL .

@$ORACLE\_HOME/rdbms/admin/utlmail.sql

@$ORACLE\_HOME/rdbms/admin/prvtmail.plb

Gói UTL\_MAIL cho phép bạn quản lí email. Xem thêm *Oracle Database PL/SQL Packages and Types Reference* để biết thêm thông tin về UTL\_MAIL.

Lưu ý rằng hiện tại, gói PLL SQL của UTL\_MAIL không hỗ trợ các máy chủ SSL.

1. Kiểm tra giá trị hiện tại của tham số khởi tạo SMTP\_OUT\_SERVER và ghi chú giá trị này để bạn có thể khôi phục lại khi hoàn tất hướng dẫn này..

Ví dụ:

SHOW PARAMETER SMTP\_OUT\_SERVER

Nếu tham số SMTP\_OUT\_SERVER đã được thiết lập, thì đầu ra tương tự như sau xuất hiện:

NAME TYPE VALUE

----------------------- ----------------- ----------------------------------

SMTP\_OUT\_SERVER string some\_imap\_server.example.com

1. Phát hành câu lệnh ALTER SYSTEM sau:

ALTER SYSTEM SET SMTP\_OUT\_SERVER="*imap\_mail\_server.example.com*";

Thay thế imap\_mail\_server bằng tên của máy chủ SMTP của bạn, mà bạn có thể tìm thấy trong cài đặt tài khoản trong công cụ email của mình. Kèm theo các cài đặt này trong dấu ngoặc kép. Ví dụ:

ALTER SYSTEM SET SMTP\_OUT\_SERVER="my\_imap\_server.example.com"

1. Kết nối với SYS bằng đặc quyền SYSOPER và sau đó khởi động lại cơ sở dữ liệu.

CONNECT SYS/AS SYSOPER

Enter password: *password*

SHUTDOWN IMMEDIATE STARTUP

1. Đảm bảo rằng cài đặt thông số SMTP\_OUT\_SERVER là chính xác..

CONNECT SYS/AS SYSDBA

Enter password: *password*

SHOW PARAMETER SMTP\_OUT\_SERVER

Output similar to the following appears:

NAME TYPE VALUE

----------------------- ----------------- ----------------------------------

SMTP\_OUT\_SERVER string my\_imap\_server.example.com

###### Bước 2: tạo tài khoản người dùng

1. Đảm bảo rằng bạn được kết nối như SYS bằng đặc quyền SYSDBA, và sau đó tạo tài khoản sysadmin\_fga, người sẽ tạo chính sách kiểm tra chi tiết.

Ví dụ:

CONNECT SYS/AS SYSDBA

Enter password: *password*

GRANT CREATE SESSION, DBA TO sysadmin\_fga IDENTIFIED BY *password*; GRANT EXECUTE ON DBMS\_FGA TO sysadmin\_fga;

GRANT CREATE PROCEDURE, DROP ANY PROCEDURE TO sysadmin\_fga;

GRANT EXECUTE ON UTL\_TCP TO sysadmin\_fga; GRANT EXECUTE ON UTL\_SMTP TO sysadmin\_fga; GRANT EXECUTE ON UTL\_MAIL TO sysadmin\_fga;

GRANT EXECUTE ON DBMS\_NETWORK\_ACL\_ADMIN TO sysadmin\_fga;

Thay thế *password* bằng 1 mật mã bảo mật hơn. Xem thêm ["Minimum Requirements for](#_bookmark197) [Passwords" trang 3-3](#_bookmark197) for để biết thêm thông tin.

UTL\_TCP, UTL\_SMTP, UTL\_MAIL và DBMS\_NETWORK\_ACL\_ADMIN PL / SQL  
các gói được sử dụng bởi cảnh báo bảo mật email mà bạn tạo.

1. Kết nối như người dùng SYSTEM.

CONNECT SYSTEM

Enter password: *password*

1. Đảm bảo rằng tài khoản lược đồ HR được mở khóa và có mật khẩu. Nếu cần, hãy mở khóa nhân sự và cấp cho người dùng này mật khẩu.

SELECT USERNAME, ACCOUNT\_STATUS FROM DBA\_USERS WHERE USERNAME = 'HR';

Nếu chế độ xem DBA\_USERS liệt kê HR của người dùng là bị khóa và hết hạn, sau đó nhập câu lệnh sau để mở khóa tài khoản HR và tạo mật khẩu mới:

ALTER USER HR ACCOUNT UNLOCK IDENTIFIED BY *password*;

Nhập mật khẩu an toàn. Để bảo mật hơn, không cung cấp cho tài khoản HR cùng một mật khẩu từ các bản phát hành trước của Cơ sở dữ liệu Oracle. ["Minimum](#_bookmark197) [Requirements for Passwords" trang 3-3](#_bookmark197)  cho các yêu cầu tối thiểu để tạo mật khẩu.

1. Tạo một tài khoản người dùng cho Susan Mavris, một đại diện nhân sự, và sau đó cấp quyền truy cập cho người dùng này vào bảng HR.EMPLOYEES.

GRANT CREATE SESSION TO smavris IDENTIFIED BY *password*;

GRANT SELECT, INSERT, UPDATE, DELETE ON HR.EMPLOYEES TO SMAVRIS;

###### Bước 3: Định cấu hình tệp danh sách kiểm soát truy cập cho dịch vụ mạng

Trước khi bạn có thể sử dụng các gói tiện ích mạng PL / SQL như UTL\_MAIL, bạn phải cấu hình tệp danh sách điều khiển truy cập (ACL) cho phép truy cập chi tiết vào các dịch vụ mạng bên ngoài. Để biết thông tin chi tiết về chủ đề này, hãy xem ["Managing](#_bookmark878)

[Fine-Grained Access in PL/SQL Packages and Types"trang 4-49](#_bookmark878). Để định cấu hình danh sách kiểm soát truy cập cho cảnh báo email:

* 1. Kết nối tới SQL\*Plus dưới người dùng sysadmin\_fga.

CONNECT sysadmin\_fga Enter password: *password*

* 1. Tạo danh sách điều khiển truy cập sau đây và các định nghĩa đặc quyền của nó.

BEGIN DBMS\_NETWORK\_ACL\_ADMIN.CREATE\_ACL (

acl => 'email\_server\_permissions.xml',

description => 'Enables network permissions for the email server', principal => 'SYSADMIN\_FGA',

is\_grant => TRUE, privilege => 'connect');

END;

/

Đảm bảo rằng bạn nhập tên người dùng chính xác cho cài đặt chính, trong các chữ cái viết hoa. Đối với hướng dẫn này, nhập SYSADMIN\_FGA cho tên của hiệu trưởng.

* 1. Gán danh sách điều khiển truy cập vào máy chủ mạng SMTP gửi đi cho máy chủ email của bạn.

BEGIN DBMS\_NETWORK\_ACL\_ADMIN.ASSIGN\_ACL (

acl => 'email\_server\_permissions.xml', host => '*SMTP\_OUT\_SERVER\_setting*', lower\_port => *port*);

END;

/

Trong ví dụ này:

* + - *SMTP\_OUT\_SERVER\_setting*: Nhập cài đặt SMTP\_OUT\_SERVER mà bạn đã đặt cho thông số SMTP\_OUT\_SERVER trong "Bước 1: Cài đặt và định cấu hình gói PLL SQL MAL MAL" trên trang 9-45. Cài đặt này phải khớp chính xác với cài đặt mà công cụ email của bạn chỉ định cho máy chủ thư đi
    - *port*: Nhập số cổng mà công cụ email của bạn chỉ định cho máy chủ gửi đi của nó. Thông thường, cài đặt này là 25. Nhập giá trị này cho cài đặt lower\_port. (Hiện tại, gói UTL\_MAIL không hỗ trợ SSL. Nếu máy chủ email của bạn là máy chủ SSL, sau đó nhập 25 cho số cổng, ngay cả khi máy chủ email sử dụng số cổng khác.)

###### Bước 4: Tạo quy trình PL / SQL cảnh báo bảo mật email

Là người dùng sysadmin\_fga, hãy tạo quy trình sau. (Bạn có thể sao chép và dán văn bản này bằng cách định vị con trỏ ở đầu CREATE OR REPLACE ở dòng đầu tiên.)

CREATE OR REPLACE PROCEDURE email\_alert (sch varchar2, tab varchar2, pol varchar2) AS

msg varchar2(20000) := 'HR.EMPLOYEES table violation. The time is: '; BEGIN

msg := msg||TO\_CHAR(SYSDATE, 'Day DD MON, YYYY HH24:MI:SS');

UTL\_MAIL.SEND (

sender => '*youremail@example.com*', recipients => '*recipientemail@example.com*',

subject => 'Table modification on HR.EMPLOYEES', message => msg);

END email\_alert;

/

Trong ví dụ này:

* + CREATE OR REPLACE PROCEDURE ...AS: Bạn phải bao gồm chữ ký mô tả tên lược đồ (sch), tên bảng (tab) và tên của quy trình kiểm tra (pol) mà bạn sẽ xác định trong chính sách kiểm tra trong bước tiếp theo.
  + sender and recipients: Thay thế youremail@example.com bằng địa chỉ email của bạn và recipientemail@example.com bằng địa chỉ email của người mà bạn muốn nhận thông báo.

###### Bước 5: Tạo và kiểm tra cài đặt the Fine-Grained Audit Policy

1. Là người dùng sysadmin\_fga, hãy tạo chính sách kiểm tra chi tiết chính sách chk\_hr\_emp như sau.

BEGIN DBMS\_FGA.ADD\_POLICY (

|  |  |  |
| --- | --- | --- |
| object\_schema  object\_name policy\_name | =>  =>  => | 'HR',  'EMPLOYEES', 'CHK\_HR\_EMP', |
| audit\_column | => | 'SALARY', |
| handler\_schema | => | 'SYSADMIN\_FGA', |
| handler\_module | => | 'EMAIL\_ALERT', |
| enable | => | TRUE, |
| statement\_types | => | 'SELECT, UPDATE', |
| audit\_trail | => | DBMS\_FGA.DB + DBMS\_FGA.EXTENDED); |

END;

/

1. Cam kết những thay đổi bạn đã thực hiện cho cơ sở dữ liệu.

COMMIT;

1. Kiểm tra các cài đặt bạn đã tạo cho đến thời điểm này.

EXEC email\_alert ('hr', 'employees', 'chk\_hr\_emp');

SQL \* Plus sẽ hiển thị một thủ tục PL / SQL thành công hoàn thành tin nhắn, và trong một thời điểm, tùy thuộc vào tốc độ của máy chủ email của bạn, bạn sẽ nhận được cảnh báo email.

Nếu bạn nhận được 1 lỗi ORA-24247: network access denied by access control list (ACL) e theo sau là ORA-06512: at *string*line *string* errors, thì kiểm tra các thiết lập trong danh sách điều khiển truy cập

###### Bước 6: Kiểm tra và cành báo

1. Kết nối tới SQL\*Plus dưới người dùng smavris, kiểm tra lương của bạn, và cho bản thân 1 con số thích hợp .

CONNECT smavris

Enter password: *password*

SELECT SALARY FROM HR.EMPLOYEES WHERE LAST\_NAME = 'Mavris';

SALARY

----------- 6500

UPDATE HR.EMPLOYEES SET SALARY = 13000 WHERE LAST\_NAME = 'Mavris';

1. Now select from a column other than SALARY in the HR.EMPLOYEES table.

SELECT FIRST\_NAME, LAST\_NAME FROM HR.EMPLOYEES WHERE LAST\_NAME = 'Raphaely';

The following output should appear:

FIRST\_NAME LAST\_NAME

-------------------- --------------------

Den Raphaely

By now, depending on the speed of you email server, you (or your recipient) should have received an email with the subject header Table modification on HR.EMPLOYEES notifying you of the tampering of the HR.EMPLOYEES table.

1. Khi người dùng sysadmin\_fga, truy vấn chế độ xem từ điển dữ liệu DBA\_FGA\_AUDIT\_TRAIL, chứa các hoạt động được kiểm tra của Susan Mavris.

CONNECT sysadmin\_fga Enter password: *password*

col dbuid format a10 col lsqltext format a66

col ntimestamp# format a15

SELECT DBUID, LSQLTEXT, NTIMESTAMP# FROM SYS.FGA\_LOG$ WHERE POLICYNAME='CHK\_HR\_ EMP';

Output similar to the following appears:

DBUID LSQLTEXT

---------- ------------------------------------------------------------------ NTIMESTAMP#

-------------------------------------------------------------------------- SMAVRIS SELECT SALARY FROM HR.EMPLOYEES WHERE LAST\_NAME = 'Mavris'

23-JUN-09 03.48.59.111000 PM

SMAVRIS UPDATE HR.EMPLOYEES SET SALARY = 13000 WHERE LAST\_NAME = 'Mavris' 23-JUN-09 03.49.07.330000 PM

Đường mòn kiểm tra nắm bắt hai câu lệnh SQL mà Susan Mavris chạy đã ảnh hưởng đến cột SALARY trong bảng HR.EMPLOYEES. Tuyên bố thứ ba mà cô ấy điều hành, trong đó cô ấy hỏi về Den Raphaely, không được ghi lại vì nó không bị ảnh hưởng bởi chính sách kiểm toán. Điều này là do Cơ sở dữ liệu Oracle thực hiện chức năng kiểm tra như một giao dịch tự quản, chỉ cam kết các hành động của thiết lập handler\_module và không phải bất kỳ giao dịch người dùng nào. Hàm này không ảnh hưởng đến bất kỳ giao dịch SQL người dùng nào.

###### Bước 7: Xóa các thành phần cho hướng dẫn này

Kết nối với SQL \* Plus với tư cách là đặc quyền SYSTEM SYSTEM, sau đó thả người dùng sysadmin\_fga (bao gồm các đối tượng trong lược đồ sysadmin\_fga) và smavris

CONNECT SYSTEM

Enter password: *password*

DROP USER sysadmin\_fga CASCADE;

DROP USER smavris;

1. Kết nối với tư cách là người dùng Nhân sự và xóa bỏ mức lương bổng của lương của Susan Mavris.

CONNECT HR

Enter password: *password*

UPDATE HR.EMPLOYEES SET SALARY = 6500 WHERE LAST\_NAME = 'Mavris';

1. Nếu bạn muốn, khóa và hết hạn HR, trừ khi những người dùng khác muốn sử dụng tài khoản này:

ALTER USER HR PASSWORD EXPIRE ACCOUNT LOCK;

1. Kết nối như người dùng SYS với đặc quyền SYSDBA, và thả danh sách điều khiển truy cập email\_server\_ permissions.xml.

BEGIN

DBMS\_NETWORK\_ACL\_ADMIN.DROP\_ACL(

acl => 'email\_server\_permissions.xml');

END;

/

Danh sách điều khiển truy cập nằm trong lược đồ SYS, không phải lược đồ của người dùng đã tạo chúng.

1. Đưa ra câu lệnh ALTER SYSTEM sau đây để khôi phục tham số SMTP\_OUT\_SERVER về giá trị trước đó, từ Bước 4 dưới ["Step 1: Install and Configure the UTL\_MAIL PL/SQL Package" trang 9-45](#_bookmark1950):

ALTER SYSTEM SET SMTP\_OUT\_SERVER="*previous\_value*";

Enclose this setting in quotation marks. For example:

ALTER SYSTEM SET SMTP\_OUT\_SERVER="some\_imap\_server.example.com"

1. Khởi động lại cá thể cơ sở dữ liệu.

SHUTDOWN STARTUP

#### Hướng dẫn: Kiểm tra người dùng Nondatabase

Phần này chứa:

* + [hướng](#_bookmark1961) dẫn
  + [Bước 1: Tạo](#_bookmark1963) tài khoản người dùng và đảm bảo tài khoản đang hoạt động
  + [Bước 2:Tạo](#_bookmark1964) chính sách kiểm toán Fine-grained
  + [Bước 3: Kiểm](#_bookmark1966) tra chính sách
  + [Bước 4: Gỡ](#_bookmark1967) các thành phần cho hướng dẫn

###### Về phần hướng dẫn

Hướng dẫn này cho thấy cách tạo chính sách kiểm tra chi tiết để kiểm tra hành động của người dùng không quen thuộc, dựa trên bộ nhận dạng trong mã định danh ứng dụng khách.

###### Bước 1: tạo tài khoản người dùng và đảm bảo nó hoạt động

1. Đăng nhập người dùng SYS dưới quyền SYSDBA.

sqlplus SYS AS SYSDBA Enter password: *password*

1. Tạo tài khoản sysadmin\_fga, người sẽ tạo chính sách kiểm tra chi tiết.

GRANT CREATE SESSION, DBA TO sysadmin\_fga IDENTIFIED BY *password*; GRANT SELECT ON OE.ORDERS TO sysadmin\_fga;

GRANT EXECUTE ON DBMS\_FGA TO sysadmin\_fga; GRANT SELECT ON SYS.FGA\_LOG$ TO sysadmin\_fga;

Thay thế *password* bằng mật mã bảo mật. Xem thêm ["Minimum Requirements for](#_bookmark197) [Passwords"](#_bookmark197) trang 3-3 để biết thêm thông tin.

1. Người dùng mẫu OE cũng sẽ được sử dụng trong hướng dẫn này, do đó truy vấn chế độ xem từ điển dữ liệu DBA\_USERS để đảm bảo rằng OE không bị khóa hoặc hết hạn.

SELECT USERNAME, ACCOUNT\_STATUS FROM DBA\_USERS WHERE USERNAME = 'OE';

Nếu khung nhìn DBA\_USERS liệt kê người dùng OE là bị khóa và hết hạn, hãy đăng nhập với tư cách người dùng SYSTEM và sau đó nhập câu lệnh sau để mở khóa tài khoản OE và tạo một mật khẩu mới:

ALTER USER OE ACCOUNT UNLOCK IDENTIFIED BY *password*;

Nhập mật khẩu bảo mật. để bảo mật hơn, không cung cấp cho tài khoản OE cùng 1 mật khẩu với các bản phát hành trước của Oracle Database. Xem thêm ["Minimum](#_bookmark197) [Requirements for Passwords" trang 3-3](#_bookmark197)  để biết thêm các yêu cầu tối thiểu để tạo mật khẩu

###### Bước 2: Tạo chính sách kiểm toán Fine-grained

1. Kết nối tới SQL\*Plus dưới người dùng sysadmin\_fga.

CONNECT sysadmin\_fga Enter password: *password*

1. Tạo chính sách sau:

BEGIN

DBMS\_FGA.ADD\_POLICY(OBJECT\_SCHEMA => 'OE', OBJECT\_NAME => 'ORDERS',

POLICY\_NAME => 'ORDERS\_FGA\_POL',

AUDIT\_CONDITION => 'SYS\_CONTEXT(''USERENV'', ''CLIENT\_ IDENTIFIER'') = ''Robert''',

HANDLER\_SCHEMA => NULL,

HANDLER\_MODULE => NULL,

ENABLE => True,

STATEMENT\_TYPES => 'INSERT,UPDATE,DELETE,SELECT',

AUDIT\_TRAIL => DBMS\_FGA.DB + DBMS\_FGA.EXTENDED,

AUDIT\_COLUMN\_OPTS => DBMS\_FGA.ANY\_COLUMNS); END;

/

Trong ví dụ này, tham số AUDIT\_CONDITION giả định rằng người dùng không phải là không quen được đặt tên là Robert. Chính sách sẽ theo dõi bất kỳ câu lệnh INSERT, UPDATE, DELETE và SELECT nào mà Robert sẽ cố gắng.

###### Bước 3: kiểm tra chính sách

**1.** kết nối dưới người dùng OE và chọn từ bảng OE.ORDERS.

CONNECT OE

Enter password: *password*

SELECT COUNT(\*) FROM ORDERS;

The following output appears:

COUNT(\*)

----------

105

1. Connect as user sysadmin\_fga and then check if any audit records were generated.

CONNECT sysadmin\_fga Enter password: *password*

SELECT DBUID, LSQLTEXT FROM SYS.FGA\_LOG$ WHERE POLICYNAME='ORDERS\_FGA\_POL';

The following output appears:

no rows selected

Because no nondatabase users were logged in to query the OE.ORDERS table, the audit trail is empty.

1. Reconnect as user OE, set the client identifier to Robert, and then reselect from the

OE.ORDERS table.

CONNECT OE

Enter password: *password*

EXEC DBMS\_SESSION.SET\_IDENTIFIER('Robert'); SELECT COUNT(\*) FROM ORDERS;

The following output should appear:

COUNT(\*)

----------

105

**4.** Reconnect as user sysadmin\_fga and then check the audit trail again.

CONNECT sysadmin\_fga Enter password: *password*

SELECT DBUID, LSQLTEXT FROM SYS.FGA\_LOG$ WHERE POLICYNAME='ORDERS\_FGA\_POL';

This time, because Robert has made his appearance and queried the OE.ORDERS

table, the audit trail captures his actions:

DBUID LSQLTEXT

---------------- ---------------------------- OE SELECT COUNT(\*) FROM ORDERS;

###### Step 4: Remove the Components for This Tutorial

1. Connect to SQL\*Plus as user SYSTEM, and then drop user sysadmin\_fga (including the objects in the sysadmin\_fga schema).

CONNECT SYSTEM

Enter password: *password*

DROP USER sysadmin\_fga CASCADE;

1. If you want, lock and expire OE, unless other users want to use this account:

ALTER USER OE PASSWORD EXPIRE ACCOUNT LOCK;

### Auditing SYS Administrative Users

This section contains:

* [Auditing User SYSTEM](#_bookmark1970)
* [Auditing User SYS and Users Who Connect as SYSDBA and SYSOPER](#_bookmark1973)

#### Auditing User SYSTEM

You can audit the SYSTEM user by using all the standard and fine-grained audit features. Insofar as auditing is concerned, user SYSTEM is a typical database user (such as HR or OE) and requires no special configuration to be audited.

[Example 9–25](#_bookmark1972) shows how to audit any table insert operations issued by user SYSTEM.

***Example 9–25 Auditing Table Insert Operations by User SYSTEM***

AUDIT INSERT ANY TABLE BY SYSTEM BY ACCESS;

#### Auditing User SYS and Users Who Connect as SYSDBA and SYSOPER

You can fully audit sessions for users who connect as SYS, including all users connecting using the SYSDBA or SYSOPER privileges. This enables you to write the actions of administrative users to an operating system file, even if the AUDIT\_TRAIL parameter is set to NONE, DB, or DB, EXTENDED. Writing the actions of administrator users to an operating system audit file is safer than writing to the SYS.AUD$ table, because administrative users can remove rows from this table that indicate their bad behavior.

To configure audit settings for SYSDBA and SYSOPER users:

1. Set the AUDIT\_SYS\_OPERATIONS initialization parameter to TRUE.

ALTER SYSTEM SET AUDIT\_SYS\_OPERATIONS=TRUE SCOPE=SPFILE;

This setting records the top-level operations directly issued by users who have connected to the database using the SYSDBA or SYSOPER privilege. It writes the audit records to the operation system audit trail. The SQL text of every statement is written to the ACTION field in the operating system audit trail record.

1. If you want to write system administrator activities to XML files, then set the

AUDIT\_TRAIL initialization parameter to either XML or XML, EXTENDED. For example:

ALTER SYSTEM SET AUDIT\_TRAIL=XML, EXTENDED SCOPE=SPFILE;

In all operating systems, if you set AUDIT\_TRAIL to either XML or XML,EXTENDED, then audit records are written as XML files in the directory specified by the AUDIT\_ FILE\_DEST initialization parameter. By default, Oracle Database writes the audit records to operating system files.

See [Table 9–1, " AUDIT\_TRAIL Initialization Parameter Settings" on page 9-10](#_bookmark1670) for more information about these settings. See also ["Enabling or Disabling the](#_bookmark1656) [Standard Audit Trail" on page 9-8](#_bookmark1656).

1. Restart the database.

After you restart the database, Oracle Database audits all successful actions performed by SYSDBA and SYSOPER users, and writes these audit records to the operating system audit trail, and not to the SYS.AUD$ table.

In Windows, if you have set the AUDIT\_TRAIL initialization parameter OS, then Oracle Database writes audit records as events to the Event Viewer log file.

**Note:**The $ORACLE\_BASE/admin/$ORACLE\_SID/adump directory is the first default location used if the AUDIT\_FILE\_DEST initialization parameter is not set or does not point to a valid directory. If writing to that first default location fails or the database is closed, then Oracle Database uses the *$ORACLE\_HOME/rdbms/audit* directory as the backup default location. If that attempt fails, then the audited operation fails and a message is written to the alert log.

When AUDIT\_TRAIL is set to OS, audit file names continue to be in the following form:

*$ORACLE\_SID\_short\_form\_process\_name\_processid\_sequence\_number*.aud

The sequence number starts from number 1.

For example, the short process name ora is used for dedicated server processes, and the names s001, s002, and so on are used for shared server processes.

When AUDIT\_TRAIL is set to XML or XML, EXTENDED, the same audit file names have the extension xml instead of aud.

If you do not specify the AUDIT\_FILE\_DEST initialization parameter, then the default location is $ORACLE\_BASE/admin/$ORACLE\_SID/adump in Linux and Solaris, and

%ORACLE\_BASE%\admin\%ORACLE\_SID%\adump for Microsoft Windows. For other operating systems, refer to their audit trail documentation.

Oracle Database audits all SYS-issued SQL statements indiscriminately and regardless of the setting of the AUDIT\_TRAIL initialization parameter.

Consider the following SYS session:

CONNECT SYS AS SYSDBA;

Enter password: *password*

ALTER SYSTEM FLUSH SHARED\_POOL;

UPDATE salary SET base=1000 WHERE name='laurel';

When SYS auditing is enabled, both the ALTER SYSTEM and UPDATE statements are displayed in the operating system audit file, similar to the following output. (Be aware that this format may change in different Oracle Database releases.)

Tue May 5 04:53:37 2009 -07:00

LENGTH : '159'

ACTION :[7] 'CONNECT' DATABASE USER:[1] '/' PRIVILEGE :[6] 'SYSDBA'

CLIENT USER:[7] 'laurelh' CLIENT TERMINAL:[5] 'pts/0' STATUS:[1] '0'

DBID:[9] '561542328'

Tue May 5 04:53:40 2009 -07:00

LENGTH : '183'

ACTION :[30] 'ALTER SYSTEM FLUSH SHARED\_POOL' DATABASE USER:[1] '/'

PRIVILEGE :[6] 'SYSDBA'

CLIENT USER:[7] 'laurelh' CLIENT TERMINAL:[5] 'pts/0' STATUS:[1] '0'

DBID:[9] '561542328'

Tue May 5 04:53:49 2009 -07:00

LENGTH : '200'

ACTION :[47] 'UPDATE salary SET base=1000 WHERE name='laurel'' DATABASE USER:[1] '/'

PRIVILEGE :[6] 'SYSDBA'

CLIENT USER:[7] 'laurelh' CLIENT TERMINAL:[5] 'pts/0' STATUS:[1] '0'

DBID:[9] '561542328'

The brackets indicate the length of the value. For example, PRIVILEGE is set to SYSDBA, which uses 6 characters. In addition, the values are in single quotes for SYS and mandatory audit records.

Because of the superuser privileges available to users who connect as SYSDBA, Oracle recommends that database administrators rarely use this connection and only when necessary. Database administrators can usually perform normal day-to-day maintenance activity. These database administrators are typical database users with the DBA role, or have been granted privileges that are the equivalent of a DBA role (for example, mydba or jr\_dba) that your organization customizes.

### Using Triggers to Write Audit Data toa Separate Table

You can use triggers to supplement the built-in auditing features of Oracle Database. The trigger that you create records user actions to a separate database table. When an activity fires the trigger, the trigger records the action in this table. Triggers are useful when you want to record customized information such as before-and-after changes to a table. For detailed information about creating triggers, see *Oracle Database PL/SQL Language Reference*.

You do not need to have auditing enabled for the trigger to work, nor does it matter what type of auditing you do have enabled. The trigger works outside of the database audit functionality.

Follow these guidelines if you want to create audit triggers:

* **Never write the trigger so that it writes data to the SYS.AUD$ table.** In fact, you should never modify the SYS.AUD$ table contents. If you try to write values to SYS.AUD$ and the trigger does not work as expected, then it could adversely affect standard auditing. The SYS.AUD$ table is an Oracle Database-owned table, and only Oracle Database should write to it.
* **If possible, create the trigger as an AFTER trigger.** The triggering statement is subjected to any applicable constraints. If no records are found, then the AFTER trigger does not fire, and audit processing is not carried out unnecessarily.
* **Create the trigger as either an AFTER row or AFTER statement trigger**. Choosing between AFTER row and AFTER statement triggers depends on the information being audited. For example, row triggers provide value-based auditing for each table row. Triggers can also require you to supply a reason code for issuing the

audited SQL statement, which can be useful in both row and statement-level auditing situations.

[Table 9–5](#_bookmark1993) provides a comparison of trigger-based auditing and the built-in database auditing features.

***Table 9–5 Comparison of Built-in Auditing and Trigger-Based Auditing***

|  |  |
| --- | --- |
| **Audit Feature** | **Description** |
| DML and DDL auditing | Standard auditing options permit auditing of DML and DDL statements regarding all types of schema objects and structures. Comparatively, triggers permit auditing of DML statements entered against tables, and DDL auditing at SCHEMA or DATABASE level. |
| Centralized audit trail | All database audit information is recorded centrally and automatically using the auditing features of the database. |
| Declarative method | Auditing features enabled using the standard database features are easier to declare and maintain, and less prone to errors, when compared to auditing functions defined by triggers. |
| Auditing options can be audited | Any changes to existing auditing options can also be audited to guard against malicious database activity. |
| Session and execution time auditing | Using the database auditing features, records are generated once every time an audited statement is entered. With triggers, an audit record is generated each time a trigger-audited table is referenced. |
| Auditing of unsuccessful data access | Database auditing can be set to audit when unsuccessful data access occurs. However, unless autonomous transactions are used, any audit information generated by a trigger is rolled back if the triggering statement is rolled back. For more information about autonomous transactions, see *Oracle Database Concepts*. |
| Sessions can be audited | Connections, disconnections, and session activity (physical I/Os, logical I/Os, deadlocks, and so on) can be recorded using standard database auditing. |

In [Example 9–26](#_bookmark1994), a trigger audits modifications to the emp\_tab table for specific rows. The trigger writes the old and new values to the emp\_audit\_tab table, including the user who performed the update and the time the update took place.

***Example 9–26 Audit Trigger to Record Before and After Changes to a Table***

**/\* 1. Create the following table: \*/**

CREATE TABLE emp\_tab (

empno NUMBER(4),

ename VARCHAR2(10),

job VARCHAR2(9),

mgr NUMBER(4),

hiredate DATE,

sal NUMBER(8,2),

deptno NUMBER(2));

**/\* 2. Create a table to capture the audit data. \*/**

CREATE TABLE emp\_audit\_tab ( oldname VARCHAR2(10),

oldjob VARCHAR2(9),

oldsal NUMBER (8,2),

newname VARCHAR2(10),

newjob VARCHAR2(9),

newsal NUMBER(8,2),

user1 varchar2(10),

systemdate TIMESTAMP);

**/\* 3. Create a trigger to record the old and new values, the author of the change, and when the change took place. \*/**

CREATE OR REPLACE TRIGGER emp\_audit\_trig AFTER INSERT OR DELETE OR UPDATE ON emp\_tab FOR EACH ROW

BEGIN

INSERT INTO emp\_audit\_tab ( oldname, oldjob, oldsal, newname, newjob, newsal, user1, systemdate

)

VALUES (

:OLD.ename, :OLD.job, :OLD.sal,

:NEW.ename, :NEW.job, :NEW.sal, user, sysdate

); END;

/

To test this trigger, add a row to the emp\_tab table, and then change the value the ename, job, or sal column in the emp\_tab table. Then query the emp\_audit\_tab table to find the audit data.

### Managing Audit Trail Records

This section contains:

* [About Audit Records](#_bookmark1996)
* [Managing the Database Audit Trail](#_bookmark1999)
* [Managing the Operating System Audit Trail](#_bookmark2029)

#### About Audit Records

Audit records include information about the operation that was audited, the user who performed the operation2, and the date and time of the operation. Depending on the type of auditing you choose, you can write audit records to data dictionary tables, called the **database audit trail**, or in operating system files, called the **operating system audit trail**.

If you choose to write audit records to the database audit trail, Oracle Database writes the audit records to the SYS.AUD$ table for default and standard auditing, and to the SYS.FGA\_LOG$ table for fine-grained auditing. Both of these tables reside in the SYSTEM tablespace and are owned by the SYS schema. You can check the contents of these tables by querying the following data dictionary views:

* DBA\_AUDIT\_TRAIL for the SYS.AUD$ contents
* DBA\_FGA\_AUDIT\_TRAIL for the SYS.FGA\_LOG$ contents
* DBA\_COMMON\_AUDIT\_TRAIL for both SYS.AUD$ and SYS.FGA\_LOG$ contents

2 Oracle Database records the actions of both database and nondatabase users in the SYS.AUD$ and SYS.FGA\_LOG$ tables. The CLIENTID column in these tables records the name of the nondatabase user. The USERID column in the SYS.AUD$ table and the DBUID column of the SYS.FGA\_LOG$ store the database user account. For nondatabase users, the USERID and DBUID columns store the database user account that was created to enable the nondatabase user access to the database. The DBA\_AUDIT\_TRAIL, DBA\_FGA\_AUDIT\_TRAIL, and DBA\_COMMON\_AUDIT\_ TRAIL views store this information in the CLIENT\_ID, USERNAME, and DB\_USER columns.

["Finding Information About Audited Activities" on page 9-80](#_bookmark2095) describes more data dictionary views that you can use to view to contents of the SYS.AUD$ and SYS.FGA\_ LOG$ tables.

If you choose to write audit records to an operating system file, you can write them to either a text file or to an XML file. You can check the contents of the audit XML files by querying the V$XML\_AUDIT\_TRAIL data dictionary view.

#### Managing the Database Audit Trail

This section contains:

* + [Database Audit Trail Contents](#_bookmark2000)
  + [Controlling the Size of the Database Audit Trail](#_bookmark2011)
  + [Moving the Database Audit Trail to a Different Tablespace](#_bookmark2015)
  + [Protecting the Database Audit Trail](#_bookmark1609)
  + [Auditing the Database Audit Trail](#_bookmark2018)
  + [Archiving the Database Audit Trail](#_bookmark2021)

**See Also:** ["Purging Audit Trail Records" on page 9-65](#_bookmark2043)

###### Database Audit Trail Contents

The database audit trail is a pair of tables, AUD$ (for standard auditing) and FGA\_LOG$ (for fine-grained auditing), in the SYS schema of each Oracle Database data dictionary. It records both standard and fine-grained audit activities. Several data dictionary views can help you use the information in this table. ["Finding Information About](#_bookmark2095) [Audited Activities" on page 9-80](#_bookmark2095) lists all the auditing-related views.

The database audit trail record contains different types of information, depending on the events audited and the auditing options set. For example, if you have set the AUDIT\_TRAIL initialization parameter to DB, EXTENDED or XML, EXTENDED, then the SQL\_BIND and SQL\_TEXT columns show any SQL bind variables used for a SQL statement and SQL text that triggered the audit, respectively. For full details about the contents of these views, refer to *Oracle Database Reference*. However, be aware that the format and columns of the DBA\_AUDIT\_TRAIL view may change across Oracle Database releases.

**Note:** If the AUDIT\_TRAIL initialization parameter is set to XML or XML, EXTENDED, then Oracle Database sends standard audit records to operating system files in XML format. Because XML is a standard document format, many utilities are available to parse and analyze XML data.

If the database destination for audit records becomes full or unavailable, and, therefore, unable to accept new records, then an audited action cannot complete. Instead, Oracle Database generates an error message and does not audit the action. You can control the size of the audit trail to make it more manageable. (In fact, Oracle strongly recommends that you do so.) See ["Controlling the Size of the Database Audit](#_bookmark2011) [Trail" on page 9-59](#_bookmark2011) for more information. See also ["Keeping Audited Information](#_bookmark2292) [Manageable" on page 10-18](#_bookmark2292).

The audit trail does not store information about any data values that might be involved in the audited statement. For example, old and new data values of updated

rows are not stored when an UPDATE statement is audited. However, you can perform this specialized type of auditing by using fine-grained auditing methods.

You can use the Flashback Query feature to show the old and new values of the updated rows, subject to any auditing policy presently in force. The current policies are enforced even if the flashback is to an old query that was originally subject to a different policy. Current business access rules always apply.

**See Also:**

* + - ["Auditing Specific Activities with Fine-Grained Auditing" on page 9-36](#_bookmark1893) for more information about methods of fine-grained auditing
    - *Oracle Database Administrator's Guide* for information about auditing table changes by using Flashback Transaction Query
    - Flashback entries in the table of system privileges listed in the GRANT SQL statement section of *Oracle Database SQL Language Reference*

**Note:** You can find information about the log history by querying the V$LOGMNR\_CONTENTS data dictionary view. The CLIENT\_ID column of this view records changes to the session client identifier. To query this view, you must have the SELECT ANY TRANSACTION system privilege.

###### Controlling the Size of the Database Audit Trail

If the database audit trail is full and no more audit records can be inserted, then underlying statement cannot complete successfully until you purge the audit trail. Oracle Database issues errors to all users who issue statements that cause the audit. Therefore, you must control the growth and size of the audit trail.

When auditing is enabled and audit records are being generated, the audit trail increases according to two factors:

* The number of audit options turned on
* The frequency of execution of audited statements

To control the growth of the audit trail, you can use the following methods:

* **Enable and disable database auditing.** If it is enabled, then audit records are generated and stored in the audit trail. If it is disabled, then audit records are not generated. (Remember that some activities are always audited.)
* **Be selective about the audit options that are turned on.** If more selective auditing is performed, then useless or unnecessary audit information is not generated and stored in the audit trail. You can use fine-grained auditing to selectively audit only certain conditions.
* **Tightly control the ability to perform object auditing.** You can accomplish this in the following ways:
  + A security administrator owns all objects and never grants the AUDIT ANY system privilege to any other user. Alternatively, all schema objects can belong to a schema for which the corresponding user does not have CREATE SESSION privilege.
  + All objects are contained in schemas that do not correspond to real database users (that is, the CREATE SESSION privilege is not granted to the

corresponding user). The security administrator is the only user granted the

AUDIT ANY system privilege.

In both scenarios, a security administrator controls entirely object auditing.

The maximum size of the database audit trail tables (AUD$ and FGA\_LOG$) is determined by the default storage parameters of the SYSTEM tablespace, in which it is stored by default. If you are concerned that a too-large database audit trail will affect the SYSTEM table performance, then consider moving the database audit trail tables to a different tablespace.

**See Also:** Operating system-specific Oracle Database documentation for more information about managing the operating system audit trail when directing audit records to that location

###### Moving the Database Audit Trail to a Different Tablespace

By default, the SYSTEM tablespace stores the database audit trail SYS.AUD$ and SYS.FGA\_LOG$ tables. You can change this default location to another tablespace, such as the SYSAUX tablespace or a user-created tablespace. You may want to move the database audit trail tables to a different tablespace if the SYSTEM tablespace is too busy. Another reason for moving these audit trail tables to a different tablespace is if you plan to purge them by using the DBMS\_AUDIT\_MGMT PL/SQL package procedures.

Be aware that moving the database audit trail tables to a different tablespace can take a long time, depending on the amount of audit data in the audit tables, so you may want to do this during a time when database activity is slow.

To move the database audit trail from SYSTEM to a different tablespace:

1. Log in to SQL\*Plus as an administrator who has the EXECUTE privilege on the

DBMS\_AUDIT\_MGMT PL/SQL package.

For more information about the DBMS\_AUDIT\_MGMT PL/SQL package, see *Oracle Database PL/SQL Packages and Types Reference*.

1. Check the tablespace to which you want to move the database audit trail tables.

You may need to optimize and allocate more space to this tablespace, including the SYSAUX auxiliary tablespace. For more information, see *Oracle Database Performance Tuning Guide*.

1. Run the DBMS\_AUDIT\_MGMT.SET\_AUDIT\_TRAIL\_LOCATION PL/SQL procedure to specify the name of the destination tablespace.

For example:

BEGIN DBMS\_AUDIT\_MGMT.SET\_AUDIT\_TRAIL\_LOCATION(

AUDIT\_TRAIL\_TYPE => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD, AUDIT\_TRAIL\_LOCATION\_VALUE => 'AUD\_AUX');

END;

In this example:

* + AUDIT\_TRAIL\_TYPE: Refers to the database audit trail type. Enter one of the following values:
    - DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD: Standard audit trail table, AUD$.
    - DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_FGA\_STD: Fine-grained audit trail table,

FGA\_LOG$.

**–** DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_DB\_STD: Both standard and fine-grained audit trail tables.

* + - * AUDIT\_TRAIL\_LOCATION\_VALUE: Specifies the destination tablespace. This example specifies a tablespace named AUD\_AUX.

###### Auditing the Database Audit Trail

At times an application must give the SYS.AUD$ system table access to regular users (non-SYSDBA users). For example, an audit report generator needs access to AUD$ table to generate daily reports on possible violations. Also, many installations have a distinct auditor role to achieve separation of duty.

In this case, be aware that DML statements such as INSERT, UPDATE, MERGE, and DELETE are always audited and recorded in the SYS.AUD$ table. You can check these activities by querying the DBA\_AUDIT\_TRAIL and DBA\_COMMON\_AUDIT\_TRAIL views.

If a user has SELECT, UPDATE, INSERT, and DELETE privileges on SYS.AUD$ and executes a SELECT operation, then the audit trail will have a record of that operation. That is, SYS.AUD$ will have a row identifying the SELECT action on itself, as for example row 1.

If a user later tries to delete this row from SYS.AUD$, then the DELETE operation succeeds, because the user has the privilege to perform this action. However, this DELETE action on SYS.AUD$ is also recorded in the audit trail. Setting up this type of auditing acts as a safety feature, potentially revealing unusual or unauthorized actions.

**Note:** DELETE, INSERT, UPDATE, and MERGE operations on the SYS.AUD$ and SYS.FGA\_LOG$ tables are always audited. These audit records are not allowed to be deleted.

**See Also:** ["Auditing Sensitive Information" on page 10-18](#_bookmark2288)

###### Archiving the Database Audit Trail

You should periodically archive and then purge the audit trail to prevent it from growing too large. Archiving and purging both frees audit trail space and facilitates the purging of the database audit trail. See ["Purging Audit Trail Records" on page 9-65](#_bookmark2043) for different ways of purging the audit trail records.

You can create an archive of the database audit trail by using one of the following methods:

* **Oracle Audit Vault.** You install Oracle Audit Vault separately from Oracle Database. For more information, see *Oracle Audit Vault Administrator's Guide*.
* **Oracle Data Warehouse.** Oracle Data Warehouse is automatically installed with Oracle Database. For more information, see *Oracle Warehouse Builder Installation and Administration Guide*.

After you complete the archive, you can purge the database audit trail contents. See ["Purging Audit Trail Records" on page 9-65](#_bookmark2043) for more information.

To archive standard and fine-grained audit records, you can copy the relevant records to a normal database table. For example:

INSERT INTO *table* SELECT ... FROM SYS.AUD$ ...; INSERT INTO *table* SELECT ... FROM SYS.FGA\_LOG$ ...;

**See Also:** The following sections for information about different ways of purging the database audit trail

* + ["Scheduling an Automatic Purge Job for the Audit Trail" on page 9-67](#_bookmark2051)
  + ["Manually Purging the Audit Trail" on page 9-72](#_bookmark2069)
  + ["Purging a Subset of Records from the Database Audit Trail" on page 9-74](#_bookmark2072)

#### Managing the Operating SystemAudit Trail

This section contains:

* + [If the Operating System Audit Trail Becomes Full](#_bookmark2031)
  + [Setting the Size of the Operating System Audit Trail](#_bookmark2033)
  + [Setting the Age of the Operating System Audit Trail](#_bookmark2036)
  + [Archiving the Operating System Audit Trail](#_bookmark2039)

**See Also:**

* + - ["Purging Audit Trail Records" on page 9-65](#_bookmark2043)
    - ["Using the Syslog Audit Trail on UNIX Systems" on page 9-18](#_bookmark1719)
    - ["Activities That Are Always Audited for All Platforms"](#_bookmark1619)

###### If the Operating System Audit Trail Becomes Full

Be aware that an operating system audit trail or file system, including the Windows Event Log, can become full, and therefore, unable to accept new records, including audit records from Oracle Database. In this case, Oracle Database cancels and rolls back the operation being performed, including operations that normally are always audited. (See ["Activities That Are Always Audited for All Platforms" on page 9-4](#_bookmark1619).) If the operating system audit trail becomes full, then set the AUDIT\_TRAIL parameter to use database audit trail (such as DB or DB, EXTENDED). This prevents the audited actions from completing if their audit records cannot be stored. You should periodically archive and purge the operating system audit file to prevent these types of failures.

If you plan to use operating system auditing, then ensure that the operating system audit trail or the file system does not fill completely. Most operating systems provide administrators with sufficient information and warning to ensure this does not occur. If you configure auditing to use the database audit trail, you can prevent this potential loss of audit information. Oracle Database prevents audited events from occurring if the audit trail is unable to accept the database audit record for the statement.

Periodically archive and then purge the operating system audit trail. See ["Archiving](#_bookmark2039) [the Operating System Audit Trail" on page 9-65](#_bookmark2039) and ["Purging Audit Trail Records" on](#_bookmark2043) [page 9-65](#_bookmark2043)for more information.

###### Setting the Size of the Operating System Audit Trail

To control the size of the operating system audit trail, set the DBMS\_AUDIT\_MGMT.OS\_ FILE\_MAX\_SIZE property by using the DBMS\_AUDIT\_MGMT.SET\_AUDIT\_TRAIL\_PROPERTY PL/SQL procedure. Remember that you must have the EXECUTE privilege for the DBMS\_ AUDIT\_MGMT PL/SQL package before you can use it. When the operating system file meets the size limitation you set, Oracle Database stops adding records to the current

file and then creates a new operating system file for the subsequent records. For more information about the DBMS\_AUDIT\_MGMT PL/SQL package, see *Oracle Database PL/SQL Packages and Types Reference*.

If you set both the DBMS\_AUDIT\_MGMT.OS\_FILE\_MAX\_SIZE and the DBMS\_AUDIT\_MGMT.OS\_ FILE\_MAX\_AGE (described in ["Setting the Age of the Operating System Audit Trail" on](#_bookmark2036) [page 9-64](#_bookmark2036)) properties, then Oracle Database performs the action based the property value limit that is met first.

For example:

BEGIN

DBMS\_AUDIT\_MGMT.SET\_AUDIT\_TRAIL\_PROPERTY(

AUDIT\_TRAIL\_TYPE => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_OS, AUDIT\_TRAIL\_PROPERTY => DBMS\_AUDIT\_MGMT.OS\_FILE\_MAX\_SIZE, AUDIT\_TRAIL\_PROPERTY\_VALUE => 10240);

END;

/

In this example:

* AUDIT\_TRAIL\_TYPE: Specifies the operating system audit trail. Enter one of the following values:
  + DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_OS: Operating system audit trail files with the

.aud extension. (This setting does not apply to Windows Event Log entries. Nor does it apply to syslog audit records, when the AUDIT\_SYSLOG\_LEVEL initialization parameter is set.)

* + DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_XML: XML audit trail files.
  + DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_FILES: Both operating system and XML audit trail files.
* AUDIT\_TRAIL\_PROPERTY: Specifies the DBMS\_AUDIT\_MGMT.OS\_FILE\_MAX\_SIZE property, which sets the maximum size. To find the status of the current property settings, query the PARAMETER\_NAME and PARAMETER\_VALUE columns of the DBA\_ AUDIT\_MGMT\_CONFIG\_PARAMS data dictionary view.
* AUDIT\_TRAIL\_PROPERTY\_VALUE: Sets the maximum size to 10240 kilobytes, that is, 10 megabytes. The default setting is 10,000 kilobytes (approximately 10 megabytes). Do not exceed 2 gigabytes.

Clearing the DBMS\_AUDIT\_MGMT.OS\_FILE\_MAX\_SIZE Setting

To clear the maximum file size setting, use the DBMS\_AUDIT\_MGMT.CLEAR\_AUDIT\_TRAIL\_ PROPERTY procedure.

For example:

BEGIN

DBMS\_AUDIT\_MGMT.CLEAR\_AUDIT\_TRAIL\_PROPERTY(

AUDIT\_TRAIL\_TYPE => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_OS, AUDIT\_TRAIL\_PROPERTY => DBMS\_AUDIT\_MGMT.OS\_FILE\_MAX\_SIZE, USE\_DEFAULT\_VALUES => TRUE );

END;

/

In this example:

* AUDIT\_TRAIL\_TYPE: Specifies the operating system audit trail. Enter one of the AUDIT\_TRAIL\_TYPE values described in ["Setting the Size of the Operating System Audit Trail" on page 9-62](#_bookmark2033).
  + AUDIT\_TRAIL\_PROPERTY: Specifies the DBMS\_AUDIT\_MGMT.OS\_FILE\_MAX\_SIZE property. You can query the DBA\_AUDIT\_MGMT\_CONFIG\_PARAMS data dictionary view to find the current status of this property.
  + USE\_DEFAULT\_VALUES: Enter one of the following values:
* TRUE: Clears the current value and uses the default value, 10,000 kilobytes, instead.
* FALSE: Oracle Database does not use a default maximum size for the operating system or XML file growth. The files will continue to grow without limitation unless you configure the DBMS\_AUDIT\_MGMT.OS\_FILE\_MAX\_AGE property. The default setting is FALSE.

###### Setting the Age of the Operating System Audit Trail

To control the age of the operating system audit trail, use the DBMS\_AUDIT\_MGMT.SET\_ AUDIT\_TRAIL\_PROPERTY PL/SQL procedure. Remember that you must have the EXECUTE privilege for the DBMS\_AUDIT\_MGMT PL/SQL package before you can use it. When the operating system file meets the age limitation you set, Oracle Database stops adding records to the current file and then creates a new operating system file for the subsequent records. For more information about the DBMS\_AUDIT\_MGMT PL/SQL package, see *Oracle Database PL/SQL Packages and Types Reference*.

If you set both the DBMS\_AUDIT\_MGMT.OS\_FILE\_MAX\_AGE and the DBMS\_AUDIT\_MGMT.OS\_ FILE\_MAX\_SIZE (described in ["Setting the Size of the Operating System Audit Trail" on](#_bookmark2033) [page 9-62](#_bookmark2033)) properties, then Oracle Database controls the growth of the Audit file based on the property value limit that is met first.

For example:

BEGIN

DBMS\_AUDIT\_MGMT.SET\_AUDIT\_TRAIL\_PROPERTY(

AUDIT\_TRAIL\_TYPE => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_OS, AUDIT\_TRAIL\_PROPERTY => DBMS\_AUDIT\_MGMT.OS\_FILE\_MAX\_AGE, AUDIT\_TRAIL\_PROPERTY\_VALUE => 10 );

END;

/

In this example:

* + AUDIT\_TRAIL\_TYPE: Specifies the operating system audit trail. Enter one of the following values:
* DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_OS: Operating system audit trail files with the

.aud extension. (This setting does not apply to Windows Event Log entries. Nor does it apply to syslog audit records, when the AUDIT\_SYSLOG\_LEVEL initialization parameter is set.)

* DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_XML: XML audit trail files.
* DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_FILES: Both operating system and XML audit trail files.
  + AUDIT\_TRAIL\_PROPERTY: Specifies the DBMS\_AUDIT\_MGMT.OS\_FILE\_MAX\_AGE property to set the maximum age. To find the status of the current property setting, query the DBA\_AUDIT\_MGMT\_CONFIG\_PARAMS data dictionary view.
  + AUDIT\_TRAIL\_PROPERTY\_VALUE: Sets the maximum age to 10 days. Enter a value between 1 and 495. The default age is 5 days.

Clearing the DBMS\_AUDIT\_MGMT.OS\_FILE\_MAX\_AGE Setting

To clear the maximum file age setting, use the DBMS\_AUDIT\_MGMT.CLEAR\_AUDIT\_TRAIL\_ PROPERTY procedure.

For example:

BEGIN

DBMS\_AUDIT\_MGMT.CLEAR\_AUDIT\_TRAIL\_PROPERTY(

AUDIT\_TRAIL\_TYPE => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_OS, AUDIT\_TRAIL\_PROPERTY => DBMS\_AUDIT\_MGMT.OS\_FILE\_MAX\_AGE, USE\_DEFAULT\_VALUES => TRUE );

END;

/

In this example:

* + - AUDIT\_TRAIL\_TYPE: Specifies operating system audit trail. Enter one of the AUDIT\_ TRAIL\_TYPE values listed in ["Setting the Age of the Operating System Audit Trail" on page 9-64](#_bookmark2036).
    - AUDIT\_TRAIL\_PROPERTY: Specifies the DBMS\_AUDIT\_MGMT.OS\_FILE\_MAX\_AGE property. Query the PARAMETER\_NAME and PARAMETER\_VALUE columns of the DBA\_ AUDIT\_MGMT\_CONFIG\_PARAMS data dictionary view to find the current status of this property.
    - USE\_DEFAULT\_VALUES: Specify one of the following values:
      * TRUE: Clears the current value and uses the default value, 5 days, instead.
      * FALSE: Oracle Database does not use a default maximum age for the operating system or XML file growth. In this case, the files will continue to age without limitation unless you configure the DBMS\_AUDIT\_MGMT.OS\_FILE\_MAX\_SIZE property. The default setting is FALSE.

###### Archiving the Operating System Audit Trail

You should periodically archive the operating system audit trail. Use your

platform-specific operating system tools to create an archive of the operating system audit files.

Use the following methods to archive the operating system audit files:

* + - **Use Oracle Audit Vault.** You install Oracle Audit Vault separately from Oracle Database. For more information, see *Oracle Audit Vault Administrator's Guide*.
    - **Create tape or disc backups.** You can create a compressed file of the audit files, and then store it on tapes or discs. Consult your operating system documentation for more information.

Afterwards, you should purge (delete) the operating system audit records both to free audit trail space and to facilitate audit trail management. See ["Purging Audit Trail](#_bookmark2043) [Records" on page 9-65](#_bookmark2043) for different ways that you can use to purge the operating system audit trail records.

### Purging Audit Trail Records

This section contains:

* + - [About Purging Audit Trail Records](#_bookmark2045)
    - [Selecting an Audit Trail Purge Method](#_bookmark2047)
    - [Scheduling an Automatic Purge Job for the Audit Trail](#_bookmark2051)
  + [Manually Purging the Audit Trail](#_bookmark2069)
  + [Purging a Subset of Records from the Database Audit Trail](#_bookmark2072)
  + [Other Audit Trail Purge Operations](#_bookmark2075)
  + [Example: Directly Calling a Database Audit Trail Purge Operation](#_bookmark2092)

#### About Purging Audit Trail Records

You should periodically archive and then delete (purge) audit trail records, because the audit trail cannot accept new records if it grows too large. This section describes a variety of ways that you can use to purge both the database and operating system audit trail records. You can purge a subset of database audit trail records. For both database and operating system audit trail types, you can manually purge the records or create a purge job that performs at a specified time interval. In that case, the purge operation either purges the audit trail records that were created before the archive timestamp, or it purges all audit trail records.

To perform the audit trail purge tasks, in most cases, you use the DBMS\_AUDIT\_MGMT PL/SQL package. You must have the EXECUTE privilege for DBMS\_AUDIT\_MGMT before you can use it.

If you have Oracle Audit Vault installed, the audit trail purge process differs from the procedures described in this manual. For example, Oracle Audit Vault archives the audit trail for you. See *Oracle Audit Vault Administrator's Guide*.

**Note:** Oracle Database audits all deletions from the audit trail, without exception. See ["Auditing the Database Audit Trail" on](#_bookmark2018) [page 9-61](#_bookmark2018) and ["Auditing SYS Administrative Users" on page 9-53](#_bookmark1968).

**See Also:**

* + - *Oracle Database PL/SQL Packages and Types Reference* for more information about the DBMS\_AUDIT\_MGMT PL/SQL package
    - *Oracle Database Reference* for detailed information about the DBA\_ AUDIT\_MGMT-related views

#### Selecting an Audit Trail Purge Method

[Table 9–6](#_bookmark2050) provides a roadmap for selecting an audit trail purge method.

***Table 9–6 Selecting an Audit Trail Purge Method***

|  |  |
| --- | --- |
| **What Do You Want to Purge?** | **About This Type of Purge Method** |
| All audit records, or audit records created before a specified timestamp, on a regularly scheduled basis | You can schedule a purge operation to occur an specific times. For example, you can schedule it for every Saturday at 2 a.m.  **General steps:**   1. If necessary, tune online and archive redo log sizes to accommodate the additional records generated during the audit table purge process. 2. Plan a timestamp and archive strategy. 3. Initialize the audit trail cleanup operation. 4. Set an archive timestamp for the audit records. 5. Create and schedule the purge job. 6. Optionally, configure the audit trail to be deleted in batches.   See ["Scheduling an Automatic Purge Job for the Audit Trail" on page 9-67](#_bookmark2051) for more information. |
| All audit records, or records that were created before a specified timestamp, when you want | You can manually purge the audit records right away in a one-time operation, rather than creating a purge schedule.  **General steps:**   1. If necessary, tune online and archive redo log sizes to accommodate the additional records generated during the audit table purge process. 2. Plan a timestamp and archive strategy. 3. Initialize the audit trail cleanup operation. 4. Set an archive timestamp for the audit records. 5. Optionally, configure the audit trail to be deleted in batches. 6. Run the purge operation.   See ["Manually Purging the Audit Trail" on page 9-72](#_bookmark2069) for more information. |
| Just a subset of the audit records from the database audit trail | You can manually purge just a subset of the audit records. For example, you can delete all audit records that were created between May 14, 2010 and June 14, 2010.  **General steps:**   1. If necessary, tune online and archive redo log sizes to accommodate the additional records generated during the audit table purge process. 2. Archive the audit records you want to purge. 3. As a user with administrative privileges, delete from the SYS.AUD$   table.  See ["Purging a Subset of Records from the Database Audit Trail" on](#_bookmark2072) [page 9-74](#_bookmark2072) for more information. |

#### Scheduling an Automatic Purge Job for the Audit Trail

You can purge the entire audit trail, or only a portion of the audit trail that was created before a timestamp. For the database audit trail, the individual audit records created before the timestamp can be purged. For the operating system audit trail, you purge audit files that were created before the timestamp.

Be aware that purging the audit trail, particularly a large one, can take a while to complete. Consider scheduling the purge job so that it runs during a time when the database is not busy.

You can create multiple purge jobs for different audit trail types, so long as they do not conflict. For example, you can create a purge job for the standard audit trail table and then the fine-grained audit trail table. However, you cannot then create a purge job for

both or all types, that is, by using the DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_DB\_STD or DBMS\_ AUDIT\_MGMT.AUDIT\_TRAIL\_ALL property.

To create and schedule an automatic purge job:

* + [Step 1: If Necessary, Tune Online and Archive Redo Log Sizes](#_bookmark2054)
  + [Step 2: Plan a Timestamp and Archive Strategy](#_bookmark2055)
  + [Step 3: Initialize the Audit Trail Cleanup Operation](#_bookmark2056)
  + [Step 4: Optionally, Set an Archive Timestamp for Audit Records](#_bookmark2058)
  + [Step 5: Create and Schedule the Purge Job](#_bookmark2063)
  + [Step 6: Optionally, Configure the Audit Trail Records to be Deleted in Batches](#_bookmark2065)

###### Step 1: If Necessary, Tune Online and Archive Redo Log Sizes

The purge process may generate additional redo logs. Before you run this process, you may need to tune online and archive redo log sizes to accommodate the additional records generated during the audit table purge process. For more information about tuning log files, see *Oracle Database Performance Tuning Guide* and *Oracle Database Administrator's Guide*.

###### Step 2: Plan a Timestamp and Archive Strategy

You must record the timestamp of the database and operating system audit records before you can archive them. You can check the timestamp date by querying the DBA\_ AUDIT\_MGMT\_LAST\_ARCH\_TS data dictionary view. Later on, when the purge takes place, Oracle Database purges only the audit trail records that were created before the date of this timestamp. See ["Step 4: Optionally, Set an Archive Timestamp for Audit Records"](#_bookmark2058) [on page 9-69](#_bookmark2058) for more information.

After you have timestamped the records, you are ready to archive them. See the following sections for more information:

* + ["Archiving the Database Audit Trail" on page 9-61](#_bookmark2021)
  + ["Archiving the Operating System Audit Trail" on page 9-65](#_bookmark2039)

###### Step 3: Initialize the Audit Trail Cleanup Operation

Before you can purge the audit trail by using the DBMS\_AUDIT\_MGMT.CLEAN\_AUDIT\_ TRAIL PL/SQL procedure, you must initialize the audit trail for the cleanup operation. For the database audit trail, if you have not moved the database audit trail tables (SYS.AUD$ and SYS.FGA\_LOG$) from the SYSTEM tablespace to another tablespace, this process moves these tables to the SYSAUX tablespace or to the tablespace that you specified in ["Moving the Database Audit Trail to a Different Tablespace" on page 9-60](#_bookmark2015). Be aware that moving these tables takes a while, so you may want to schedule the initialization process during time when the database is not busy.

To initialize the audit trail cleanup operation:

1. Log in to SQL\*Plus as an administrative user who has the EXECUTE privilege on the

DBMS\_AUDIT\_MGMT PL/SQL package.

1. If you have not done so already, initialize the audit trail cleanup operation by running the DBMS\_AUDIT\_MGMT.INIT\_CLEANUP procedure. (You only need to perform this step once.

You can check if the audit trail has been initialized for cleanup by running the DBMS\_AUDIT\_MGMT.IS\_CLEANUP\_INITIALIZED function. See ["Verifying That the](#_bookmark2076) [Audit Trail Is Initialized for Cleanup" on page 9-75](#_bookmark2076).)

For example:

BEGIN DBMS\_AUDIT\_MGMT.INIT\_CLEANUP(

AUDIT\_TRAIL\_TYPE => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD, DEFAULT\_CLEANUP\_INTERVAL => 12 );

END;

/

In this specification:

* + AUDIT\_TRAIL\_TYPE: Enter one of the following values:
    - DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD: Standard audit trail table, AUD$.
    - DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_FGA\_STD: Fine-grained audit trail table,

FGA\_LOG$.

* + - DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_DB\_STD: Both standard and fine-grained audit trail tables.
    - DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_OS: Operating system audit trail files with the .aud extension. (This setting does not apply to Windows Event Log entries.)
    - DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_XML: XML Operating system audit trail files.
    - DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_FILES: Both operating system and XML audit trail files.
    - DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_ALL: All audit trail records, that is, both database audit trail and operating system audit trail types.
  + DEFAULT\_CLEANUP\_INTERVAL: Specify the desired default hourly purge interval (for example, 12 for every 12 hours). The DBMS\_AUDIT\_MGMT procedures use this value to determine how to purge audit records. The timing begins when you run the DBMS\_AUDIT\_MGMT.INIT\_CLEANUP procedure. To update this value later, set the DBMS\_AUDIT\_MGMT.CLEAN\_UP\_INTERVAL property of the DBMS\_AUDIT\_ MGMT.SET\_AUDIT\_TRAIL\_PROPERTY procedure.

The DEFAULT\_CLEANUP\_INTERVAL setting must indicate the frequency in which DBMS\_AUDIT\_MGMT.CLEAN\_AUDIT\_TRAIL is called. If you are uncertain about the frequency, set it to an approximate value. You can change this value later on by using the DBMS\_AUDIT\_MGMT.SET\_AUDIT\_TRAIL\_PROPERTY procedure.

###### Step 4: Optionally, Set an Archive Timestamp for Audit Records

If you want to delete all of the audit trail, you can bypass this step.

You can set a timestamp when the last audit record was archived. Setting an archive timestamp provides a hint to the cleanup infrastructure that the cleanup operation will be invoked every 6 hours.

For the database audit trail, you must set the timestamp after you have initialized the audit trail cleanup operation. To find the last archive timestamps for the audit trail, you can query the DBA\_AUDIT\_MGMT\_LAST\_ARCH\_TS data dictionary view. After you set the timestamp, all audit records in the audit trail that indicate a time earlier than that timestamp are purged when you run the DBMS\_AUDIT\_MGMT.CLEAN\_AUDIT\_TRAIL PL/SQL procedure. If you want to clear the archive timestamp setting, see ["Clearing](#_bookmark2088) [the Archive Timestamp Setting" on page 9-78](#_bookmark2088).

For the operating system audit trail, remember that you cannot delete individual audit records in the operating system (including XML) audit files. Instead, Oracle Database removes the entire file that contains the timestamped records.

If you are using Oracle Real Application Clusters (Oracle RAC), then use Network Time Protocol (NTP) to synchronize the time on each computer where you have installed an Oracle Database instance. For example, suppose you set the time for one Oracle RAC instance node at 11:00:00 a.m. and then set the next Oracle RAC instance node at 11:00:05. As a result, the two nodes have inconsistent times. You can use Network Time Protocol (NTP) to synchronize the times for these Oracle RAC instance nodes.

To set the timestamp, use the DBMS\_AUDIT\_MGMT.SET\_LAST\_ARCHIVE\_TIMESTAMP

PL/SQL procedure.

For example:

BEGIN

DBMS\_AUDIT\_MGMT.SET\_LAST\_ARCHIVE\_TIMESTAMP(

AUDIT\_TRAIL\_TYPE => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD, LAST\_ARCHIVE\_TIME => '2009-05-28 06:30:00.00' RAC\_INSTANCE\_NUMBER => 0 );

END;

/

In this example:

* + AUDIT\_TRAIL\_TYPE: Enter one of the following settings:
* DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD: Specified the standard audit trail table, AUD$.
* DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_FGA\_STD: Specifies the fine-grained audit trail table, FGA\_LOG$.
* DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_OS: Operating system audit trail files with the

.aud extension. (This setting does not apply to Windows Event Log entries.)

* DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_XML: Specifies XML audit trail files.
  + LAST\_ARCHIVE\_TIME: Enter the timestamp in YYYY-MM-DD HH:MI:SS.FF UTC (Coordinated Universal Time) format for AUDIT\_TRAIL\_DB\_AUD and AUDIT\_TRAIL\_ FGA\_STD (standard and fine-grained audit trails), and in the Local Time Zone for AUDIT\_TRAIL\_OS and AUDIT\_TRAIL\_XML (operating system and XML audit trails).
  + RAC\_INSTANCE\_NUMBER: Specifies the instance number for an Oracle RAC installation. If you specified the DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD or DBMS\_ AUDIT\_MGMT.AUDIT\_TRAIL\_FGA\_STD audit trail types, you can omit the RAC\_ INSTANCE\_NUMBER argument. This is because there is only one AUD$ and FGA\_LOG$ table, even for an Oracle RAC installation. The default is 0, which is used for single-instance database installations.

Typically, after you set the timestamp, you can use the DBMS\_AUDIT\_MGMT.CLEAN\_ AUDIT\_TRAIL PL/SQL procedure to remove the audit records that were created before the timestamp date.

###### Step 5: Create and Schedule the Purge Job

Create and schedule the purge job by running the DBMS\_AUDIT\_MGMT.CREATE\_PURGE\_ JOB PL/SQL procedure.

For example:

BEGIN

DBMS\_AUDIT\_MGMT.CREATE\_PURGE\_JOB (

AUDIT\_TRAIL\_TYPE => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD, AUDIT\_TRAIL\_PURGE\_INTERVAL => 12,

AUDIT\_TRAIL\_PURGE\_NAME => 'Standard\_Audit\_Trail\_PJ', USE\_LAST\_ARCH\_TIMESTAMP => TRUE );

END;

/

In this example:

* + - AUDIT\_TRAIL\_TYPE: Enter one of the following values:
      * DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD: Standard audit trail table, AUD$
      * DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_FGA\_STD: Fine-grained audit trail table, FGA\_ LOG$
      * DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_DB\_STD: Both standard and fine-grained audit trail tables
      * DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_OS: Operating system audit trail files with the

.aud extension. (This setting does not apply to Windows Event Log entries.)

* + - * DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_XML: XML audit trail files
      * DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_FILES: Both operating system and XML audit trail files
      * DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_ALL: All audit trail records, that is, both database audit trail and operating system audit trail types
    - AUDIT\_TRAIL\_PURGE\_INTERVAL: Specify the hourly interval for this purge job to run. The timing begins when you run the DBMS\_AUDIT\_MGMT.CREATE\_PURGE\_JOB procedure, in this case, 12 hours after you run this procedure. Later on, if you want to update this value, run the DBMS\_AUDIT\_MGMT.SET\_PURGE\_JOB\_INTERVAL procedure.
    - USE\_LAST\_ARCH\_TIMESTAMP: Enter either of the following settings:
      * TRUE: Deletes audit records created before the last archive timestamp. To check the last recorded timestamp, query the LAST\_ARCHIVE\_TS column of the DBA\_ AUDIT\_MGMT\_LAST\_ARCH\_TS data dictionary view. The default value is TRUE. Oracle recommends that you set USE\_LAST\_ARCH\_TIMESTAMP to TRUE.
      * FALSE: Deletes all audit records without considering last archive timestamp. Be careful about using this setting, in case you inadvertently delete audit records that should not have been deleted.

###### Step 6: Optionally, Configure the Audit Trail Records to be Deleted in Batches

By default, the DBMS\_AUDIT\_MGMT package procedures delete the database and operating system audit trail records in batches of 10000 database audit records, or 1000 operating system audit files. You can set this batch size to a different value if you want. Later on, when Oracle Database runs the purge job, it deletes each batch, rather than all records together. If the audit trail is very large (and they can grow quite large), deleting the records in batches facilitates the purge operation.

To find the current batch setting, you can query the PARAMETER\_NAME and PARAMETER\_ VALUE columns of the DBA\_AUDIT\_MGMT\_CONFIG\_PARAMS data dictionary view. To set the batch size, use the DBMS\_AUDIT\_MGMT.SET\_AUDIT\_TRAIL\_PROPERTY procedure. If you later want to clear this setting, see ["Clearing the Database Audit Trail Batch Size" on](#_bookmark2090) [page 9-78](#_bookmark2090).

For example:

BEGIN DBMS\_AUDIT\_MGMT.SET\_AUDIT\_TRAIL\_PROPERTY(

AUDIT\_TRAIL\_TYPE => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD, AUDIT\_TRAIL\_PROPERTY => DBMS\_AUDIT\_MGMT.DB\_DELETE\_BATCH\_SIZE, AUDIT\_TRAIL\_PROPERTY\_VALUE => 100000);

END;

/

In this example:

* + AUDIT\_TRAIL\_TYPE: Specifies the audit trail type, which in this case is the database system audit trail. Enter one of the following values:
* DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD: Standard audit trail table, AUD$.
* DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_FGA\_STD: Fine-grained audit trail table, FGA\_ LOG$.
* DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_OS: Operating system audit files.
* DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_XML: XML audit files.
  + AUDIT\_TRAIL\_PROPERTY: Uses the DBMS\_AUDIT\_MGMT.DB\_DELETE\_BATCH\_SIZE property to indicate the database audit trail batch size setting. If you want to batch the operating system audit trail, then use the FILE\_DELETE\_BATCH\_SIZE property.
  + AUDIT\_TRAIL\_PROPERTY\_VALUE: Sets the number of audit record files to be 100,000 for each batch. Enter a value between 100 and 1000000. To determine this number, consider the total number of records being purged, and the time interval in which the purge operation is performed. The default is 10000 for the database audit trail and 1000 for the operating system audit trail records.

#### Manually Purging the Audit Trail

You can manually purge the audit trail right away, without scheduling a purge job. Similar to a purge job, you can purge audit trail records that were created before an archive timestamp date or all the records in the audit trail.

Note the following about the DBMS\_AUDIT\_MGMT.CLEAN\_AUDIT\_TRAIL PL/SQL procedure:

* + Only the current audit directory is cleaned up when you run this procedure.
  + On Microsoft Windows, because the DBMS\_AUDIT\_MGMT package does not support cleanup of Windows Event Viewer, setting the AUDIT\_TRAIL\_TYPE property to DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_OS has no effect. This is because operating system audit records on Windows are written to Windows Event Viewer. The DBMS\_ AUDIT\_MGMT package does not support this type of cleanup operation.
  + On UNIX platforms, if you set the AUDIT\_SYSLOG\_LEVEL initialization parameter to a valid value as listed in *Oracle Database Reference*, then Oracle Database writes the operating system log files to syslog files. If you set the AUDIT\_TRAIL\_TYPE property to DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_OS, then the procedure only removes .aud files under audit directory (This directory is specified by the AUDIT\_FILE\_DEST initialization parameter).
  + When the AUDIT\_TRAIL\_TYPE parameter is set to DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_ XML, this procedure only cleans up XML audit files (.xml) in the current audit directory. Oracle Database maintains an index file, called adx\_*$ORACLE\_SID*.txt,

which lists the XML files that were generated by the XML auditing. The cleanup procedure does not remove this file.

For database audit trails, you must initialize the cleanup infrastructure by running the DBMS\_AUDIT\_MGMT.INIT\_CLEANUP procedure, and then purging the database audit trail by using the method described in ["Purging a Subset of Records from the Database](#_bookmark2072) [Audit Trail" on page 9-74](#_bookmark2072).

To manually purge the audit trail:

1. Follow these steps under ["Scheduling an Automatic Purge Job for the Audit Trail" on page 9-67](#_bookmark2051):
   * [Step 1: If Necessary, Tune Online and Archive Redo Log Sizes](#_bookmark2054)
   * [Step 2: Plan a Timestamp and Archive Strategy](#_bookmark2055)
   * [Step 3: Initialize the Audit Trail Cleanup Operation](#_bookmark2056)
   * [Step 4: Optionally, Set an Archive Timestamp for Audit Records](#_bookmark2058)
   * [Step 5: Create and Schedule the Purge Job](#_bookmark2063)
   * [Step 6: Optionally, Configure the Audit Trail Records to be Deleted in Batches](#_bookmark2065)
2. Purge the audit trail records by running the DBMS\_AUDIT\_MGMT.CLEAN\_AUDIT\_TRAIL

PL/SQL procedure.

For example:

BEGIN

DBMS\_AUDIT\_MGMT.CLEAN\_AUDIT\_TRAIL(

AUDIT\_TRAIL\_TYPE => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD, USE\_LAST\_ARCH\_TIMESTAMP => TRUE );

END;

/

In this example:

* + AUDIT\_TRAIL\_TYPE: Enter one of the following values:
    - DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD: Standard audit trail table, AUD$
    - DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_FGA\_STD: Fine-grained audit trail table,

FGA\_LOG$

* + - DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_DB\_STD: Both standard and fine-grained audit trail tables
    - DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_OS: Operating system audit trail files with the .aud extension. (This setting does not apply to Windows Event Log entries.)
    - DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_XML: XML audit trail files
    - DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_FILES: Both operating system and XML audit trail files
    - DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_ALL: All audit trail records, that is, both database audit trail and operating system audit trail types
  + USE\_LAST\_ARCH\_TIMESTAMP: Enter either of the following settings:
    - TRUE: Deletes audit records created before the last archive timestamp. To set the archive timestamp, see ["Step 4: Optionally, Set an Archive Timestamp for Audit Records" on page 9-69](#_bookmark2058). The default (and

recommended) value is TRUE. Oracle recommends that you set USE\_LAST\_ ARCH\_TIMESTAMP to TRUE.

**–** FALSE: Deletes all audit records without considering last archive timestamp. Be careful about using this setting, in case you inadvertently delete audit records that should have been deleted.

#### Purging a Subset of Records fromthe Database Audit Trail

You can manually remove records from the database audit trail tables. This method can be useful if you want to remove a specific subset of records. You can use this method if the database audit trail table is in any tablespace, including the SYSTEM tablespace.

For example, to delete audit records that were created later than the evening of February 28, 2009 but before March 28, 2009, enter the following statement:

DELETE FROM SYS.AUD$

WHERE NTIMESTAMP# > TO\_TIMESTAMP ('28-FEB-09 09.07.59.907000 PM') AND NTIMESTAMP# < TO\_TIMESTAMP ('28-MAR-09 09.07.59.907000 PM');

Alternatively, to delete *all* audit records from the audit trail, enter the following statement:

DELETE FROM SYS.AUD$;

Only the user SYS or a user to whom SYS granted the DELETE privilege on SYS.AUD$ can delete records from the database audit trail.

**Note:** If the audit trail is full and connections are being audited (that is, if the AUDIT SESSION statement is set), then typical users cannot connect to the database because the associated audit record for the connection cannot be inserted into the audit trail. In this case, connect as SYS with the SYSDBA privilege, and make space available in the audit trail. Remember that operations by SYS are not recorded in the standard audit trail, but they are audited if you set the AUDIT\_SYS\_OPERATIONS parameter to TRUE.

After you delete the rows from the database audit trail table, the freed space is available for reuse by that table. (The SYS.AUD$ table is allocated only as many extents as are necessary to maintain current audit trail records.) You do not need to do anything to make this space available to the table for reuse. If you want to use this space for another table, then follow these steps:

1. Move the AUD$ table to an auto segment space managed tablespace. For example:

BEGIN

DBMS\_AUDIT\_MGMT.SET\_AUDIT\_TRAIL\_LOCATION

(audit\_trail\_type => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_DB\_STD, audit\_trail\_location\_value => 'USERS');

END;

/

1. Run the following statements:

ALTER TABLE SYSTEM.AUD$ ENABLE ROW MOVEMENT; ALTER TABLE SYSTEM.AUD$ SHRINK SPACE CASCADE;

1. If you must move the AUD$ table back to the SYSTEM tablespace, then run the following statement:

BEGIN DBMS\_AUDIT\_MGMT.SET\_AUDIT\_TRAIL\_LOCATION

(audit\_trail\_type => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_DB\_STD, audit\_trail\_location\_value => 'SYSTEM');

END;

/

If you want to both delete all the rows from the database audit trail table and free the used space for other tablespace objects, use the TRUNCATE TABLE statement. For example:

TRUNCATE TABLE SYS.AUD$;

**Note:** SYS.AUD$ and SYS.FGA\_LOG$ are the only SYS objects that can ever be directly modified.

#### Other Audit Trail Purge Operations

This section contains:

* [Verifying That the Audit Trail Is Initialized for Cleanup](#_bookmark2076)
* [Setting the Default Audit Trail Purge Interval for Any Audit Trail Type](#_bookmark2078)
* [Cancelling the Initialization Cleanup Settings](#_bookmark2080)
* [Enabling or Disabling an Audit Trail Purge Job](#_bookmark2082)
* [Setting the Default Audit Trail Purge Job Interval for a Specified Purge Job](#_bookmark2084)
* [Deleting an Audit Trail Purge Job](#_bookmark2086)
* [Clearing the Archive Timestamp Setting](#_bookmark2088)
* [Clearing the Database Audit Trail Batch Size](#_bookmark2090)

###### Verifying That the Audit Trail Is Initialized for Cleanup

You can check if the audit trail has been initialized for cleanup by running the DBMS\_ AUDIT\_MGMT.IS\_CLEANUP\_INITIALIZED function. If the audit trail has been initialized, then this function returns TRUE. If it is not, it returns FALSE.

For example:

SET SERVEROUTPUT ON BEGIN

IF

DBMS\_AUDIT\_MGMT.IS\_CLEANUP\_INITIALIZED(DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD) THEN

DBMS\_OUTPUT.PUT\_LINE('AUD$ is initialized for cleanup'); ELSE

DBMS\_OUTPUT.PUT\_LINE('AUD$ is not initialized for cleanup.'); END IF;

END;

/

This example verifies that the database standard audit trail has been initialized and returns a message indicating its status. To select a setting for a different audit trail, choose from the AUDIT\_TRAIL\_TYPE settings described in ["Step 3: Initialize the Audit](#_bookmark2056)

[Trail Cleanup Operation" on page 9-68](#_bookmark2056).

###### Setting the Default Audit Trail Purge Interval for Any Audit Trail Type

You can set a default purge operation interval, in hours, that must pass before the next purge operation takes place for a specified audit trail type.

For example:

BEGIN DBMS\_AUDIT\_MGMT.SET\_AUDIT\_TRAIL\_PROPERTY(

AUDIT\_TRAIL\_TYPE => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD, AUDIT\_TRAIL\_PROPERTY => DBMS\_AUDIT\_MGMT.CLEAN\_UP\_INTERVAL, AUDIT\_TRAIL\_PROPERTY\_VALUE => 24 );

END;

/

In this example:

* + AUDIT\_TRAIL\_TYPE: Specifies the audit trail type, which in this case is the database standard audit trail. Choose from the following settings:
* DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD: Standard audit trail table, AUD$
* DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_FGA\_STD: Fine-grained audit trail table, FGA\_ LOG$
* DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_DB\_STD: Both standard and fine-grained audit trail tables
* DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_OS: Operating system audit trail files with the

.aud extension. (This setting does not apply to Windows Event Log entries.)

* DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_XML: XML Operating system audit trail files
* DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_FILES: Both operating system and XML audit trail files
* DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_ALL: All audit trail records, that is, both database audit trail and operating system audit trail types

You can set a default interval for multiple audit trail types, so long as they do not conflict. For example, you can set individual intervals for the DBMS\_AUDIT\_ MGMT.AUDIT\_TRAIL\_AUD\_STD and DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_FGA\_STD

properties, but not for the DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_DB\_STD property.

* + AUDIT\_TRAIL\_PROPERTY: Sets the DBMS\_AUDIT\_MGMT.CLEAN\_UP\_INTERVAL property to indicate the purge operation interval setting. To find the current property settings, query the PARAMETER\_NAME and PARAMETER\_VALUE columns of the DBA\_ AUDIT\_MGMT\_CONFIG\_PARAMS data dictionary view. The timing begins when you set the DBMS\_AUDIT\_MGMT.CLEAN\_UP\_INTERVAL property.
  + AUDIT\_TRAIL\_PROPERTY\_VALUE: Updates the default hourly interval set by the

DBMS\_AUDIT\_MGMT.INIT\_CLEANUP procedure. Enter a value between 1 and 999.

###### Cancelling the Initialization Cleanup Settings

You can cancel the DBMS\_AUDIT\_MGMT.INIT\_CLEANUP settings, that is, the default cleanup interval, by invoking the DBMS\_AUDIT\_MGMT.DEINIT\_CLEANUP procedure.

For example, to cancel all purge settings for the standard audit trail:

BEGIN DBMS\_AUDIT\_MGMT.DEINIT\_CLEANUP(

AUDIT\_TRAIL\_TYPE => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD);

END;

/

In this example:

* + - AUDIT\_TRAIL\_TYPE: Enter one of the AUDIT\_TRAIL\_TYPE settings listed in ["Step 3: Initialize the Audit Trail Cleanup Operation" on page 9-68](#_bookmark2056).

###### Enabling or Disabling an Audit Trail Purge Job

To enable or disable an audit trail purge job, use the DBMS\_AUDIT\_MGMT.SET\_PURGE\_ JOB\_STATUS PL/SQL procedure.

For example:

BEGIN DBMS\_AUDIT\_MGMT.SET\_PURGE\_JOB\_STATUS(

AUDIT\_TRAIL\_PURGE\_NAME => 'OS\_Audit\_Trail\_PJ', AUDIT\_TRAIL\_STATUS\_VALUE => DBMS\_AUDIT\_MGMT.PURGE\_JOB\_ENABLE);

END;

/

In this example:

* + - AUDIT\_TRAIL\_PURGE\_NAME: Specifies a purge job called OS\_Audit\_Trail\_PJ. To find existing purge jobs, query the JOB\_NAME and JOB\_STATUS columns of the DBA\_ AUDIT\_MGMT\_CLEANUP\_JOBS data dictionary view.
    - AUDIT\_TRAIL\_STATUS\_VALUE: Enter one of the following properties:
      * DBMS\_AUDIT\_MGMT.PURGE\_JOB\_ENABLE: Enables the specified purge job.
      * DBMS\_AUDIT\_MGMT.PURGE\_JOB\_DISABLE: Disables the specified purge job.

###### Setting the Default Audit Trail Purge Job Interval for a Specified Purge Job

You can set a default purge operation interval, in hours, that must pass before the next purge job operation takes place. The interval setting that is used in the DBMS\_AUDIT\_ MGMT.CREATE\_PURGE\_JOB procedure takes precedence over this setting.

For example:

BEGIN DBMS\_AUDIT\_MGMT.SET\_PURGE\_JOB\_INTERVAL(

AUDIT\_TRAIL\_PURGE\_NAME => 'OS\_Audit\_Trail\_PJ', AUDIT\_TRAIL\_INTERVAL\_VALUE => 24 );

END;

/

In this example:

* + - AUDIT\_TRAIL\_PURGE\_NAME: Specifies the name of the audit trail purge job. To find a list of existing purge jobs, query the JOB\_NAME and JOB\_STATUS columns of the DBA\_ AUDIT\_MGMT\_CLEANUP\_JOBS data dictionary view.
    - AUDIT\_TRAIL\_INTERVAL\_VALUE: Updates the default hourly interval set by the DBMS\_AUDIT\_MGMT.CREATE\_PURGE\_JOB procedure. Enter a value between 1 and 999. The timing begins when you run the purge job.

###### Deleting an Audit Trail Purge Job

To delete an audit trail purge job, use the DBMS\_AUDIT\_MGMT.DROP\_PURGE\_JOB PL/SQL procedure. To find existing purge jobs, query the JOB\_NAME and JOB\_STATUS columns of the DBA\_AUDIT\_MGMT\_CLEANUP\_JOBS data dictionary view.

For example:

BEGIN DBMS\_AUDIT\_MGMT.DROP\_PURGE\_JOB(

AUDIT\_TRAIL\_PURGE\_NAME => 'FGA\_Audit\_Trail\_PJ'); END;

/

In this example:

* + AUDIT\_TRAIL\_PURGE\_NAME: Specifies a purge job called FGA\_Audit\_Trail\_PJ.

###### Clearing the Archive Timestamp Setting

To clear the archive timestamp setting, use the DBMS\_AUDIT\_MGMT.CLEAR\_LAST\_ ARCHIVE\_TIMESTAMP PL/SQL procedure.

For example:

BEGIN

DBMS\_AUDIT\_MGMT.CLEAR\_LAST\_ARCHIVE\_TIMESTAMP( AUDIT\_TRAIL\_TYPE => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_XML', RAC\_INSTANCE\_NUMBER => 1 );

END;

/

In this example:

* + RAC\_INSTANCE\_NUMBER: If the AUDIT\_TRAIL\_TYPE property is set to DBMS\_AUDIT\_ MGMT.AUDIT\_TRAIL\_OS or DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_XML, then you cannot set RAC\_INSTANCE\_NUMBER to 0. You can omit this setting or specify 1 to indicate an instance number.

You can omit the RAC\_INSTANCE\_NUMBER setting when AUDIT\_TRAIL\_TYPE is DBMS\_ AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD or DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_FGA\_STD, or

if the database is not an Oracle RAC database. Otherwise, specify the correct instance number. You can find the instance number by issuing the SHOW PARAMETER INSTANCE\_NUMBER command in SQL\*Plus.

###### Clearing the Database Audit Trail Batch Size

To clear the batch size setting, use the DBMS\_AUDIT\_MGMT.CLEAR\_AUDIT\_TRAIL\_ PROPERTY procedure.

For example:

BEGIN

DBMS\_AUDIT\_MGMT.CLEAR\_AUDIT\_TRAIL\_PROPERTY(

AUDIT\_TRAIL\_TYPE => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD, AUDIT\_TRAIL\_PROPERTY => DBMS\_AUDIT\_MGMT.DB\_DELETE\_BATCH\_SIZE, USE\_DEFAULT\_VALUES => TRUE );

END;

/

In this example:

* + AUDIT\_TRAIL\_TYPE: Specifies the audit trail type, which in this case is the database system audit trail. Enter one of the AUDIT\_TRAIL\_TYPE values listed in ["Step 6:](#_bookmark2065)

[Optionally, Configure the Audit Trail Records to be Deleted in Batches" on](#_bookmark2065) [page 9-71](#_bookmark2065).

* + - AUDIT\_TRAIL\_PROPERTY: Specifies the DB\_DELETE\_BATCH\_SIZE property. Query the DBA\_AUDIT\_MGMT\_CONFIG\_PARAMS data dictionary view to find the current status of this property.
    - USE\_DEFAULT\_VALUES: Is set to TRUE, which clears the current audit record batch size and uses the default value, 10000, instead.

#### Example: Directly Calling a Database Audit Trail Purge Operation

The pseudo code in [Example 9–27](#_bookmark2094) creates a database audit trail purge operation that the user calls by invoking the DBMS\_ADUIT.CLEAN\_AUDIT\_TRAIL procedure. The purge operation deletes records that were created before the last archived timestamp by using a loop. The loop archives the audit records, calculates which audit records were archived and uses the SetCleanUpAuditTrail call to set the last archive timestamp, and then calls the CLEAN\_AUDIT\_TRAIL procedure. It deletes the database audit trail records in batches of 100,000 records each. In this example, major steps are in **bold** typeface.

***Example 9–27 Directly Calling a Database Audit Trail Purge Operation***

**-- 1. Initialize the AUD$ table for cleanup:**

PROCEDURE CleanUpAuditTrailMain() BEGIN

-- Connect to the database using appropriate login. CALL ConnectToDatabase();

-- The login used must have privileges to modify Audit settings.

-- Currently, the DBA will be the authorized user

DBMS\_AUDIT\_MGMT.INIT\_CLEANUP(

AUDIT\_TRAIL\_TYPE => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD, DEFAULT\_CLEANUP\_INTERVAL => 12 );

END; /\*PROCEDURE \*/

/

**-- 2. Optionally, set the batch size:**

BEGIN

DBMS\_AUDIT\_MGMT.SET\_AUDIT\_TRAIL\_PROPERTY(

AUDIT\_TRAIL\_TYPE => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD, AUDIT\_TRAIL\_PROPERTY => DBMS\_AUDIT\_MGMT.DB\_DELETE\_BATCH\_SIZE,

AUDIT\_TRAIL\_PROPERTY\_VALUE => 100000 /\* delete batch size \*/); END; /\*PROCEDURE \*/

/

**-- 3. Set the last archive timestamp:** PROCEDURE SetCleanUpAuditTrail() BEGIN

CALL FindLastArchivedTimestamp(AUD$); DBMS\_AUDIT\_MGMT.SET\_LAST\_ARCHIVE\_TIMESTAMP(

AUDIT\_TRAIL\_TYPE => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD, LAST\_ARCHIVE\_TIME => '20-AUG-2009 00:00:00');

END /\* PROCEDURE \*/

/

**-- 4. Run a customized archive procedure to purge the audit trail records:**

BEGIN

CALL MakeAuditSettings(); LOOP (/\* How long to loop\*/)

-- Invoke function for audit record archival CALL DoAuditRecordArchival(AUD$);

CALL SetCleanUpAuditTrail();

IF(/\* Clean up is needed immediately \*/) DBMS\_AUDIT\_MGMT.CLEAN\_AUDIT\_TRAIL(

AUDIT\_TRAIL\_TYPE => DBMS\_AUDIT\_MGMT.AUDIT\_TRAIL\_AUD\_STD, USE\_LAST\_ARCH\_TIMESTAMP => TRUE);

END IF

END LOOP /\*LOOP\*/ END; /\* PROCEDURE \*/

/

### Finding Information About Audited Activities

This section contains:

* + [Using Data Dictionary Views to Find Information About the Audit Trail](#_bookmark2097)
  + [Using Audit Trail Views to Investigate Suspicious Activities](#_bookmark2102)
  + [Deleting the Audit Trail Views](#_bookmark2115)

**Tip:** To find error information about audit policies, check the trace files. The USER\_DUMP\_DEST initialization parameter sets the location of the trace files.

#### Using Data Dictionary Views to Find Information About the Audit Trail

[Table 9–7](#_bookmark2101) lists data dictionary views that provide auditing information. For detailed information about these views, see *Oracle Database Reference.*

***Table 9–7 Data Dictionary Views That Display Information about the Database Audit Trail***

|  |  |
| --- | --- |
| **View** | **Description** |
| ALL\_AUDIT\_POLICIES | Describes the fine-grained auditing policies on the tables and views accessible to the current user |
| ALL\_AUDIT\_POLICY\_COLUMNS | Describes the fine-grained auditing policy columns on the tables and views accessible to the current user. |
| ALL\_DEF\_AUDIT\_OPTS | Lists default object-auditing options that are to be applied when objects are created |
| AUDIT\_ACTIONS | Describes audit trail action type codes |
| DBA\_AUDIT\_EXISTS | Lists audit trail entries produced BY AUDIT NOT EXISTS |
| DBA\_AUDIT\_MGMT\_CLEAN\_EVENTS | Displays the history of purge events. Periodically, as user SYS connected with the SYSDBA privilege, you should delete the contents of this view so that it does not grow too large. For example:  DELETE FROM DBA\_AUDIT\_MGMT\_CLEAN\_EVENTS; |
| DBA\_AUDIT\_MGMT\_CLEANUP\_JOBS | Displays the currently configured audit trail purge jobs |
| DBA\_AUDIT\_MGMT\_CONFIG\_PARAMS | Displays the currently configured audit trail properties that are used by the DBMS\_AUDIT\_MGMT PL/SQL package |
| DBA\_AUDIT\_MGMT\_LAST\_ARCH\_TS | Displays the last archive timestamps that have set for audit trail purges. |
| DBA\_AUDIT\_OBJECT | Lists audit trail records for all objects in the system |
| DBA\_AUDIT\_POLICIES | Lists all the fine-grained auditing policies on the system |
| DBA\_AUDIT\_SESSION | Lists all audit trail records concerning CONNECT and DISCONNECT |
| DBA\_AUDIT\_POLICY\_COLUMNS | Describes the fine-grained auditing policy columns on the tables and views throughout the database. |

***Table 9–7 (Cont.) Data Dictionary Views That Display Information about the Database Audit Trail***

|  |  |
| --- | --- |
| **View** | **Description** |
| DBA\_AUDIT\_STATEMENT | Lists audit trail records concerning GRANT, REVOKE, AUDIT, NOAUDIT, and  ALTER SYSTEM statements throughout the database |
| DBA\_AUDIT\_TRAIL | Lists all standard audit trail entries in the AUD$ table |
| DBA\_COMMON\_AUDIT\_TRAIL | Combines standard and fine-grained audit log records, and includes  SYS and mandatory audit records written in XML format |
| DBA\_FGA\_AUDIT\_TRAIL | Lists audit trail records for fine-grained auditing. |
| DBA\_OBJ\_AUDIT\_OPTS | Displays the objects on which auditing options have been enabled |
| DBA\_PRIV\_AUDIT\_OPTS | Describes current system privileges being audited across the system and by user |
| DBA\_STMT\_AUDIT\_OPTS | Describes current statement auditing options across the system and by user |
| USER\_AUDIT\_OBJECT | Lists audit trail records for statements concerning objects that are accessible to the current user |
| USER\_AUDIT\_POLICIES | Describes the fine-grained auditing policy columns on the tables and views accessible to the current user. |
| USER\_AUDIT\_SESSION | Lists all audit trail records concerning connections and disconnections for the current user |
| USER\_AUDIT\_STATEMENT | Lists audit trail records concerning GRANT, REVOKE, AUDIT, NOAUDIT, and  ALTER SYSTEM statements issued by the user |
| USER\_AUDIT\_TRAIL | Lists all standard audit trail entries in the AUD$ table relating to the current user |
| USER\_OBJ\_AUDIT\_OPTS | Describes auditing options on all objects owned by the current user |
| V$LOGMNR\_CONTENTS | Contains log history information. To query this view, you must have the  SELECT ANY TRANSACTION privilege. |
| V$XML\_AUDIT\_TRAIL | Shows standard, fine-grained, SYS, and mandatory audit records written in XML format files. |

#### Using Audit Trail Views to Investigate Suspicious Activities

This section provides examples that demonstrate how to examine and interpret the information in the audit trail. Suppose you want to audit the database for the following suspicious activities:

* + - Passwords, tablespace settings, and quotas for some database users are altered without authorization.
    - A high number of deadlocks occur, most likely because of users acquiring exclusive table locks.
    - Rows are arbitrarily deleted from the emp table in laurel's schema.

You suspect the users jward and swilliams of several of these detrimental actions. To investigate, you issue the following statements (in the order specified):

AUDIT ALTER, INDEX, RENAME ON DEFAULT;

CREATE VIEW laurel.employee AS SELECT \* FROM laurel.emp; AUDIT SESSION BY jward, swilliams;

AUDIT ALTER USER; AUDIT LOCK TABLE BY ACCESS

WHENEVER SUCCESSFUL;

AUDIT DELETE ON laurel.emp BY ACCESS

WHENEVER SUCCESSFUL;

The following statements are subsequently issued by the user jward:

ALTER USER tsmith QUOTA 0 ON users; DROP USER djones;

The following statements are subsequently issued by the user swilliams:

LOCK TABLE laurel.emp IN EXCLUSIVE MODE;

DELETE FROM laurel.emp WHERE mgr = 7698;

ALTER TABLE laurel.emp ALLOCATE EXTENT (SIZE 100K);

CREATE INDEX laurel.ename\_index ON laurel.emp (ename); CREATE PROCEDURE laurel.fire\_employee (empid NUMBER) AS

BEGIN

DELETE FROM laurel.emp WHERE empno = empid; END;

/

EXECUTE laurel.fire\_employee(7902);

The following sections display the information relevant to your investigation that can be viewed using the audit trail views in the data dictionary:

* + [Listing Active Statement Audit Options](#_bookmark2104)
  + [Listing Active Privilege Audit Options](#_bookmark2107)
  + [Listing Active Object Audit Options for Specific Objects](#_bookmark2109)
  + [Listing Default Object Audit Options](#_bookmark2111)
  + [Listing Audit Records](#_bookmark2113)
  + [Listing Audit Records for the AUDIT SESSION Option](#_bookmark2114)

###### Listing Active Statement Audit Options

The following query returns all the statement audit options that are set:

SELECT \* FROM DBA\_STMT\_AUDIT\_OPTS;

Output similar to the following appears:

|  |  |  |  |
| --- | --- | --- | --- |
| USER\_NAME | AUDIT\_OPTION | SUCCESS | FAILURE |
| -------------------- | ------------------- | ---------- | --------- |
| JWARD | DROP ANY CLUSTER | BY ACCESS | BY ACCESS |
| SWILLIAMS | DEBUG PROCEDURE | BY ACCESS | BY ACCESS |
| MSEDLAK | ALTER RESOURCE COST | BY ACCESS | BY ACCESS |

###### Listing Active Privilege Audit Options

The following query returns all the privilege audit options that are set:

SELECT \* FROM DBA\_PRIV\_AUDIT\_OPTS;

Output similar to the following appears:

USER\_NAME PRIVILEGE SUCCESS FAILURE

------------------- -------------------- --------- ---------- PSMITH BY ACCESS BY ACCESS

###### Listing Active Object Audit Options for Specific Objects

The following query returns all audit options set for any objects with names that start with the characters emp and that are contained in the schema of laurel:

SELECT \* FROM DBA\_OBJ\_AUDIT\_OPTS

WHERE OWNER = 'LAUREL' AND OBJECT\_NAME LIKE 'EMP%';

Output similar to the following appears:

OWNER OBJECT\_NAME OBJECT\_TY ALT AUD COM DEL GRA IND INS LOC ...

----- ----------- --------- --- --- --- --- --- --- --- --- ... LAUREL EMP TABLE S/S -/- -/- A/- -/- S/S -/- -/- ...

LAUREL EMPLOYEE VIEW -/- -/- -/- A/- -/- S/S -/- -/- ...

The view returns information about all the audit options for the specified object. The information in the view is interpreted as follows:

* + - A dash (-) indicates that the audit option is not set.
    - The S character indicates that the audit option is set BY SESSION.
    - The A character indicates that the audit option is set BY ACCESS.
    - Each audit option has two possible settings, WHENEVER SUCCESSFUL and WHENEVER NOT SUCCESSFUL, separated by a slash (/). For example, the DELETE audit option for laurel.emp is set BY ACCESS for successful DELETE statements and not set at all for unsuccessful DELETE statements.

###### Listing Default Object Audit Options

The following query returns all default object audit options:

SELECT \* FROM ALL\_DEF\_AUDIT\_OPTS;

Output similar to the following appears:

ALT AUD COM DEL GRA IND INS LOC REN SEL UPD REF EXE FBK REA

--- --- --- --- --- --- --- --- --- --- --- --- --- --- ---

S/S -/- -/- -/- -/- S/S -/- -/- S/S -/- -/- -/- -/- /- -/-

Notice that the view returns information similar to the USER\_OBJ\_AUDIT\_OPTS and DBA\_ OBJ\_AUDIT\_OPTS views (refer to previous example).

###### Listing Audit Records

The following query lists audit records generated for all objects in the database:

SELECT \* FROM DBA\_AUDIT\_OBJECT;

###### Listing Audit Records for the AUDIT SESSION Option

The following query lists audit information corresponding to the AUDIT SESSION

statement audit option:

SELECT USERNAME, LOGOFF\_TIME, LOGOFF\_LREAD, LOGOFF\_PREAD, LOGOFF\_LWRITE, LOGOFF\_DLOCK

FROM DBA\_AUDIT\_SESSION;

Output similar to the following appears:

USERNAME LOGOFF\_TI LOGOFF\_LRE LOGOFF\_PRE LOGOFF\_LWR LOGOFF\_DLO

---------- --------- ---------- ---------- ---------- ----------

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| JWARD 02-AUG-91 | 53 | 2 | 24 | 0 |
| SWILLIAMS 02-AUG-91 | 3337 | 256 | 630 | 0 |

#### Deleting the Audit Trail Views

If you disable auditing and no longer need the audit trail views, then delete them by connecting to the database as SYS and run the script file CATNOAUD.SQL. The location of the CATNOAUD.SQL script is operating system-dependent.

# 10

## Keeping Your Oracle Database Secure

This chapter contains:

* + - [About the Security Guidelines in This Chapter](#_bookmark2120)
    - [Downloading Security Patches and Contacting Oracle Regarding Vulnerabilities](#_bookmark2121)
    - [Guidelines for Securing User Accounts and Privileges](#_bookmark2130)
    - [Guidelines for Securing Roles](#_bookmark2159)
    - [Guidelines for Securing Passwords](#_bookmark2166)
    - [Guidelines for Securing Data](#_bookmark2194)
    - [Guidelines for Securing the ORACLE\_LOADER Access Driver](#_bookmark2215)
    - [Guidelines for Securing a Database Installation and Configuration](#_bookmark2219)
    - [Guidelines for Securing the Network](#_bookmark2226)
    - [Guidelines for Auditing](#_bookmark2284)
    - [Addressing the CONNECT Role Change](#_bookmark2304)

### About the Security Guidelines in This Chapter

This chapter provides a set of guidelines to keep your Oracle database secure. Information security, and privacy and protection of corporate assets and data are critical in any business. Oracle Database comprehensively addresses the need for information security by providing cutting-edge security features such as deep data protection, auditing, scalable security, secure hosting, and data exchange.

Oracle Database leads the industry in security. To maximize the security features offered by Oracle Database in any business environment, it is imperative that the database itself be well protected.

Security guidelines provide advice about how to configure Oracle Database to be secure by adhering to and recommending industry-standard and advisable security practices for operational database deployments. Many of the guidelines described in this section address common regulatory requirements such as those described in the Sarbanes-Oxley Act. For more information about how Oracle Database addresses regulatory compliance, protection of personally identifiable information, and internal threats, visit:

<http://www.oracle.com/technetwork/topics/security/whatsnew/index.html>

### Downloading Security Patches and Contacting Oracle Regarding Vulnerabilities

This section contains:

* + [Applying Security Patches and Workaround Solutions](#_bookmark2123)
  + [Contacting Oracle Security Regarding Vulnerabilities in Oracle Database](#_bookmark2128)

#### Applying Security Patches and Workaround Solutions

Always apply all relevant security patches for both the operating system on which Oracle Database resides and Oracle Database itself, and for all installed Oracle Database options and components.

Periodically check the security site on Oracle Technology Network for details about security alerts released by Oracle at

<http://www.oracle.com/technetwork/topics/security/alerts-086861.html>

Also check the Oracle Worldwide Support Service site, My Oracle Support, for details about available and upcoming security-related patches at

[http](https://support.oracle.com/)s://support.oracle.com

#### Contacting Oracle Security Regarding Vulnerabilities in Oracle Database

If you are an Oracle customer or an Oracle partner, use My Oracle Support to submit a Service Request on any potential Oracle product security vulnerability. Otherwise, send an email to [secalert\_us@oracle.com](mailto:secalert_us@oracle.com) with a complete description of the problem, including product version and platform, together with any scripts and examples.

Oracle encourages those who want to contact Oracle Security to employ email encryption, using our encryption key.

### Guidelines for Securing User Accounts and Privileges

Follow these guidelines to secure user accounts and privileges:

1. Practice the principle of least privilege.

Oracle recommends the following guidelines:

* 1. Grant necessary privileges only.

Do not provide database users or roles more privileges than are necessary. (If possible, grant privileges to roles, not users.) In other words, the *principle of least privilege* is that users be given only those privileges that are actually required to efficiently perform their jobs.

To implement this principle, restrict the following as much as possible:

* + - The number of SYSTEM and OBJECT privileges granted to database users.
    - The number of people who are allowed to make SYS-privileged connections to the database.
    - The number of users who are granted the ANY privileges, such as the DROP ANY TABLE privilege. For example, there is generally no need to grant CREATE ANY TABLE privileges to a non-DBA-privileged user.

**–** The number of users who are allowed to perform actions that create, modify, or drop database objects, such as the TRUNCATE TABLE, DELETE TABLE, DROP TABLE statements, and so on.

* 1. Limit granting the CREATE ANY EDITION and DROP ANY EDITION privileges.

To maintain additional versions of objects, editions can increase resource and disk space consumption in the database. Only grant the CREATE ANY EDITION and DROP ANY EDITION privileges to trusted users who are responsible for performing upgrades.

* 1. Restrict the CREATE ANY JOB, BECOME USER, EXP\_FULL\_DATABASE, and IMP\_FULL\_DATABASE privileges.

These are powerful security-related privileges. Only grant these privileges to users who need them.

* 1. Restrict library-related privileges to trusted users only.

The CREATE LIBRARY, CREATE ANY LIBRARY, ALTER ANY LIBRARY, and EXECUTE

ANY LIBRARY privileges, and grants of EXECUTE ON *library\_name* convey a great deal of power to users. If you plan to create PL/SQL interfaces to libraries, only grant the EXECUTE privilege to the PL/SQL interface. Do not grant EXECUTE on the underlying library. You must have the EXECUTE privilege on a library to create the PL/SQL interface to it. However, users have this privilege implicitly on libraries that they create in their own schemas. Explicit grants of EXECUTE ON *library\_name* are rarely required. Only make an explicit grant of these privileges to trusted users, and never to the PUBLIC role.

* 1. Restrict synonym-related privileges to trusted users only.

The CREATE PUBLIC SYNONYM and DROP PUBLIC SYNONYM system privileges convey a great deal of power to these users. Do not grant these privileges to users, unless they are trusted.

* 1. Do not allow non-administrative users access to objects owned by the SYS schema.

Do not allow users to alter table rows or schema objects in the SYS schema, because doing so can compromise data integrity. Limit the use of statements such as DROP TABLE, TRUNCATE TABLE, DELETE, INSERT, or similar

object-modification statements on SYS objects only to highly privileged administrative users.

The SYS schema owns the data dictionary. You can protect the data dictionary by setting the O7\_DICTIONARY\_ACCESSIBILITY parameter to FALSE. See Guideline [1](#_bookmark2198) under ["Guidelines for Securing Data" on page 10-10](#_bookmark2194) for more information.

* 1. Only grant the EXECUTE privilege on the DBMS\_RANDOM PL/SQL package to trusted users.

The EXECUTE privilege on the DBMS\_RANDOM package could permit users who normally should have only minimal access to execute the functions associated with this package.

* 1. Restrict permissions on run-time facilities.

Many Oracle Database products use run-time facilities, such as Oracle Java Virtual Machine (OJVM). Do not assign all permissions to a database run-time facility. Instead, grant specific permissions to the explicit document root file paths for facilities that might run files and packages outside the database.

Here is an example of a vulnerable run-time call, which individual files are specified:

call dbms\_java.grant\_permission('wsmith', 'SYS:java.io.FilePermission','<<ALL FILES>>','read');

Here is an example of a better (more secure) run-time call, which specifies a directory path instead:

call dbms\_java.grant\_permission('wsmith', 'SYS:java.io.FilePermission','<<actual directory path>>','read');

1. Lock and expire default (predefined) user accounts.

Oracle Database installs with several default database user accounts. Upon successful installation of the database, the Database Configuration Assistant automatically locks and expires most default database user accounts.

If you perform a manual (without using Database Configuration Assistant) installation of Oracle Database, then no default database users are locked upon successful installation of the database server. Or, if you have upgraded from a previous release of Oracle Database, you may have default accounts from earlier releases. Left open in their default states, these user accounts can be exploited, to gain unauthorized access to data or disrupt database operations.

You should *lock* and *expire* all default database user accounts. Oracle Database provides SQL statements to perform these operations. For example:

ALTER USER ANONYMOUS PASSWORD EXPIRE ACCOUNT LOCK;

See *Oracle Database SQL Language Reference* for more information about the ALTER USER statement.

Installing additional products and components after the initial installation also results in creating more default database accounts. Database Configuration Assistant automatically locks and expires all additionally created database user accounts. Unlock only those accounts that need to be accessed on a regular basis and assign a strong, meaningful password to each of these unlocked accounts. Oracle provides SQL and password management to perform these operations.

If any default database user account other than the ones left open is required for any reason, then a database administrator (DBA) must unlock and activate that account with a new, secure password.

See *Oracle Database 2 Day + Security Guide* for a description of the predefined user accounts that are created when you install Oracle Database.

If a default database user account, other than the ones left open, is required for any reason, then a database administrator (DBA) can unlock and activate that account with a new, secure password.

Oracle Enterprise Manager Accounts

If you install Oracle Enterprise Manager, the SYSMAN and DBSNMP accounts are open, unless you configure Oracle Enterprise Manager for central administration. In this case, the SYSMAN account (if present) will be locked.

If you do not install Oracle Enterprise Manager, then only the SYS and SYSTEM accounts are open. Database Configuration Assistant locks and expires all other accounts (including SYSMAN and DBSNMP).

1. Use the following data dictionary views to find information about user access to the database.

* DBA\_\*
* DBA\_ROLES
* DBA\_SYS\_PRIVS
* DBA\_ROLE\_PRIVS
* DBA\_TAB\_PRIVS
* DBA\_AUDIT\_TRAIL (if standard auditing is enabled)
* DBA\_FGA\_AUDIT\_TRAIL (if fine-grained auditing is enabled)

1. Monitor the granting of the following privileges only to users and roles who need these privileges.

By default, Oracle Database audits the following privileges:

* ALTER SYSTEM
* AUDIT SYSTEM
* CREATE EXTERNAL JOB

Oracle recommends that you also audit the following privileges:

* ALL PRIVILEGES (which includes privileges such as BECOME USER, CREATE LIBRARY, and CREATE PROCEDURE)
* DBMS\_BACKUP\_RESTORE package
* EXECUTE to DBMS\_SYS\_SQL
* SELECT ANY TABLE
* SELECT on PERFSTAT.STATS$SQLTEXT
* SELECT on PERFSTAT.STATS$SQL\_SUMMARY
* SELECT on SYS.SOURCE$
* Privileges that have the WITH ADMIN clause
* Privileges that have the WITH GRANT clause
* Privileges that have the CREATE keyword

1. Revoke access to the following:

* The SYS.USER\_HISTORY$ table from all users except SYS and DBA accounts
* The RESOURCE role from typical application accounts
* The CONNECT role from typical application accounts
* The DBA role from users who do not need this role

1. Grant privileges only to roles.

Granting privileges to roles and not individual users makes the management and tracking of privileges much easier.

1. Limit the proxy account (for proxy authorization) privileges to CREATE SESSION only.
2. **Use secure application roles to protect roles that are enabled by application code.**

Secure application roles allow you to define a set of conditions, within a PL/SQL package, that determine whether or not a user can log on to an application. Users do not need to use a password with secure application roles.

Another approach to protecting roles from being enabled or disabled in an application is the use of role passwords. This approach prevents a user from directly accessing the database in SQL (rather than the application) to enable the privileges associated with the role. However, Oracle recommends that you use secure application roles instead, to avoid having to manage another set of passwords.

1. Discourage users from using the NOLOGGING clause in SQL statements.

In some SQL statements, the user has the option of specifying the NOLOGGING clause, which indicates that the database operation is not logged in the online redo log file. Even though the user specifies the clause, a redo record is still written to the online redo log file. However, there is no data associated with this record.

Because of this, using NOLOGGING has the potential for malicious code to be entered can be accomplished without an audit trail.

### Guidelines for Securing Roles

Follow these guidelines when managing roles:

1. Grant a role to users only if they need all privileges of the role.

Roles (groups of privileges) are useful for quickly and easily granting permissions to users. Although you can use Oracle-defined roles, you have more control and continuity if you create your own roles containing only the privileges pertaining to your requirements. Oracle may change or remove the privileges in an Oracle Database-defined role, as it has with the CONNECT role, which now has only the CREATE SESSION privilege. Formerly, this role had eight other privileges.

Ensure that the roles you define contain only the privileges that reflect job responsibility. If your application users do not need all the privileges encompassed by an existing role, then apply a different set of roles that supply just the correct privileges. Alternatively, create and assign a more restricted role.

For example, it is imperative to strictly limit the privileges of user SCOTT, because this is a well known account that may be vulnerable to intruders. Because the CREATE DBLINK privilege allows access from one database to another, drop its privilege for SCOTT. Then, drop the entire role for the user, because privileges acquired by means of a role cannot be dropped individually. Re-create your own role with only the privileges needed, and grant that new role to that user.

Similarly, for better security, drop the CREATE DBLINK privilege from all users who do not require it.

1. Do not grant user roles to application developers.

Roles are not meant to be used by application developers, because the privileges to access schema objects within stored programmatic constructs need to be granted directly. Remember that roles are not enabled within stored procedures except for invoker’s right procedures. See ["How Roles Work in PL/SQL Blocks" on page 4-9](#_bookmark570) for information about this topic.

1. Create and assign roles specific to each Oracle Database installation.

This principle enables the organization to retain detailed control of its roles and privileges. This also avoids the necessity to adjust if Oracle Database changes or

removes Oracle Database-defined roles, as it has with CONNECT, which now has only the CREATE SESSION privilege. Formerly, it also had eight other privileges.

1. For enterprise users, create global roles.

Global roles are managed by an enterprise directory service, such as Oracle Internet Directory. See the following sections for more information about global roles:

* + ["Configuring Global User Authentication and Authorization" on page 3-30](#_bookmark375)
  + ["Global Role Authorization by an Enterprise Directory Service" on page 4-19](#_bookmark678)
  + *Oracle Database Enterprise User Security Administrator's Guide*

### Guidelines for Securing Passwords

When you create a user account, Oracle Database assigns a default password policy for that user. The password policy defines rules for how the password should be created, such as a minimum number of characters, when it expires, and so on. You can strengthen passwords by using password policies. See also ["Configuring Password](#_bookmark185) [Protection" on page 3-1](#_bookmark185) for additional ways to protect passwords.

Follow these guidelines to further strengthen passwords:

1. Choose passwords carefully.

["Minimum Requirements for Passwords" on page 3-3](#_bookmark197) describes the minimum requirements for passwords. Follow these additional guidelines when you create or change passwords:

* + Make the password between 12 and 30 characters and numbers.
  + Have the password contain at least one digit, one upper-case character, and one lower-case character.
  + Use mixed case letters and special characters in the password. (See ["Ensuring Against Password Security Threats by Using the SHA-1 Hashing Algorithm" on page 3-15](#_bookmark290) for more information.)
  + You can include multibyte characters in the password.
  + Use the database character set for the password’s characters, which can include the underscore (\_), dollar ($), and number sign (#) characters.
  + You must enclose the following passwords in double-quotation marks:
    - Passwords containing multibyte characters.
    - Passwords starting with numbers or special characters and containing alphabetical characters. For example:

"123abc" "#abc" "123dc$"

* + - Passwords containing any character other than alphabetical characters,

numbers, and special characters. For example:

"abc>" "abc@", " "

* You do not need to specify the following passwords in double-quotation marks.
  + Passwords starting with an alphabet character (a–z, A–Z) and containing numbers(0–9) or special characters ($, #, \_). For example:

abc123 ab23a ab$#\_

* + Passwords containing only numbers.
  + Passwords containing only alphabetical characters.
* Do not use an actual word for the entire password.

1. To create a longer, more complex password from a shorter, easier to remember password, follow these techniques:

* Create passwords from the first letters of the words of an easy-to-remember sentence. For example, "I usually work until 6 almost every day of the week" can be Iuwu6aedotw.
* Combine two weaker passwords, such as welcome1 and binky into

WelBinkyCome1.

* Repeat a character at the beginning or end of the password.
* Add a string, another password, or part the same password to the beginning or end of the password that you want to create. For example, ways that you can modify the password fussy2all are as follows:
  + fussy2all34hj2
  + WelBinkyCome1fussy2all
  + fusfussy2all
* Double some or all of the letters. For example, welcome13 can become

wwellCcooMmee13.

1. Ensure that the password is sufficiently complex.

Oracle Database provides a password verification routine, the PL/SQL script UTLPWDMG.SQL, that you can run to check whether or not passwords are sufficiently complex. Ideally, edit the UTLPWDMG.SQL script to provide stronger password protections. See also ["Enforcing Password Complexity Verification" on page 3-11](#_bookmark269) for a sample routine that you can use to check passwords.

1. Associate a password function with the user profile or the default profile.

The PASSWORD\_VERIFY\_FUNCTION clause of the CREATE PROFILE and ALTER PROFILE

statements associates a password function with a user profile or the default profile. Password functions ensure that users create strong passwords using guidelines that are specific to your site. Having a password function also requires a user changing his or her own password (without the ALTER USER system privilege) to provide both the old and new passwords. You can create your own password functions or use the password functions that Oracle Database provides.

See ["Customizing Password Complexity Verification" on page 3-12](#_bookmark273) for more information.

1. Change default user passwords.

Oracle Database installs with a set of predefined, default user accounts. Security is most easily broken when a default database user account still has a default password *even after installation*. This is particularly true for the user account SCOTT, which is a well known account that may be vulnerable to intruders. In Oracle Database 11*g* Release 2 (11.2), default accounts are installed locked with the passwords expired, but if you have upgraded from a previous release, you may still have accounts that use default passwords.

To find user accounts that have default passwords, query the DBA\_USERS\_WITH\_ DEFPWD data dictionary view. See ["Finding User Accounts That Have Default](#_bookmark207) [Passwords" on page 3-4](#_bookmark207) for more information.

1. Change default passwords of administrative users.

You can use the same or different passwords for the SYS, SYSTEM, SYSMAN, and DBSNMP administrative accounts. Oracle recommends that you use different passwords for each. In any Oracle environment (production or test), assign strong, secure, and distinct passwords to these administrative accounts. If you use Database Configuration Assistant to create a new database, then it requires you to enter passwords for the SYS and SYSTEM accounts, disallowing the default passwords CHANGE\_ON\_INSTALL and MANAGER.

Similarly, for production environments, do not use default passwords for administrative accounts, including SYSMAN and DBSNMP.

See *Oracle Database 2 Day + Security Guide* for information about changing a default password.

1. Enforce password management.

Apply basic password management rules (such as password length, history, , and so forth) to all user passwords. Oracle Database has password policies enabled for the default profile. Guideline [1](#_bookmark2170) in this section lists these password policies. *Oracle Database 2 Day + Security Guide* lists initialization parameters that you can use to further secure user passwords.

You can find information about user accounts by querying the DBA\_USERS view. The PASSWORD column of the DBA\_USERS view indicates whether the password is global, external, or null. The DBA\_USERS view provides useful information such as the user account status, whether the account is locked, and password versions.

Oracle also recommends, if possible, using Oracle Advanced Security (an option to Oracle Database Enterprise Edition) with network authentication services (such as Kerberos), token cards, smart cards, or X.509 certificates. These services provide strong authentication of users, and provide protection against unauthorized access to Oracle Database.

1. Do not store user passwords in clear text in Oracle tables.

For better security, do not store passwords in clear text (that is, human readable) in Oracle tables. You can correct this problem by using a secure external password store to encrypt the table column that contains the password. See ["Managing the](#_bookmark298) [Secure External Password Store for Password Credentials" on page 3-16](#_bookmark298) for information.

When you create or modify a password for a user account, Oracle Database automatically creates a cryptographic hash or digest of the password. If you query the DBA\_USERS view to find information about a user account, the data in the PASSWORD column indicates if the user password is global, external, or null.

### Guidelines for Securing Data

Follow these guidelines to secure data on your system:

1. Enable data dictionary protection.

Oracle recommends that you protect the data dictionary to prevent users that have the ANY system privilege from using those privileges on the data dictionary.

Altering or manipulating the data in data dictionary tables can permanently and detrimentally affect the operation of a database.

To enable data dictionary protection, set the following initialization parameter to

FALSE (which is the default) in the init*sid*.ora control file:

O7\_DICTIONARY\_ACCESSIBILITY = FALSE

You can set the O7\_DICTIONARY\_ACCESSIBILITY parameter in a server parameter file. For more information about server parameter files, see *Oracle Database Administrator's Guide*.

After you set O7\_DICTIONARY\_ACCESSIBILTY to FALSE, only users who have the

SELECT ANY DICTIONARY privilege and those authorized users making

DBA-privileged (for example CONNECT / AS SYSDBA) connections can use the ANY system privilege on the data dictionary. If O7\_DICTIONARY\_ACCESSIBILITY parameter is not set to FALSE, then any user with the DROP ANY TABLE (for example) system privilege will be able to drop parts of the data dictionary.

However, if a user *needs* view access to the data dictionary, then you can grant that user the SELECT ANY DICTIONARY system privilege.

**Note:**

* + In a default installation, the O7\_DICTIONARY\_ACCESSIBILITY parameter is set to FALSE. However, in Oracle8*i*, this parameter is set to TRUE by default, and must be changed to FALSE to enable this security feature.
  + The SELECT ANY DICTIONARY privilege is not included in the GRANT ALL PRIVILEGES statement, but you can grant it through a role. [Chapter 4, "Configuring Privilege and Role Authorization"](#_bookmark493) describes roles in detail.

1. Restrict operating system access.

Follow these guidelines:

* + Limit the number of operating system users.
  + Limit the privileges of the operating system accounts (administrative,

root-privileged, or DBA) on the Oracle Database host computer to the least privileges required for a user to perform necessary tasks.

* + Restrict the ability to modify the default file and directory permissions for the Oracle Database home (installation) directory or its contents. Even privileged operating system users and the Oracle owner should not modify these permissions, unless instructed otherwise by Oracle.
  + Restrict symbolic links. Ensure that when you provide a path or file to the database, neither the file nor any part of the path is modifiable by an untrusted user. The file and all components of the path should be owned by the database administrator or trusted account, such as *root*.

This recommendation applies to all types of files: data files, log files, trace files, external tables, BFILE data types, and so on.

1. Encrypt sensitive data and all backup media that contains database files.

According to common regulatory compliance requirements, you must encrypt sensitive data such as credit card numbers and passwords. When you delete sensitive data from the database, encrypted data does not linger in data blocks, operating system files, or sectors on disk.

In most cases, you may want to use transparent data encryption to encrypt your sensitive data. See *Oracle Database Advanced Security Administrator's Guide* for more information. See also ["Security Problems That Encryption Does Not Solve" on](#_bookmark1550) [page 8-1](#_bookmark1550) for when you should not encrypt data.

1. For Oracle Automatic Storage Management (Oracle ASM) environments on Linux and UNIX systems, use Oracle ASM File Access Control to restrict access to the Oracle ASM disk groups.

If you use different operating system users and groups for Oracle Database installations, then you can configure Oracle ASM File Access Control to restrict the access to files in Oracle ASM disk groups to only authorized users. For example, a database administrator would only be able to access the data files for the databases that he or she manages. This administrator would not be able to see or overwrite the data files belonging (or used by) other databases.

For more information about managing Oracle ASM File Access Control for disk groups, see *Oracle Automatic Storage Management Administrator's Guide*. For information about the various privileges required for multiple software owners on Linux systems, see also *Oracle Automatic Storage Management Administrator's Guide*.

### Guidelines for Securing the ORACLE\_LOADER Access Driver

Follow these guidelines to secure the ORACLE\_LOADER access driver:

1. **Create a separate operating system directory to store the access driver preprocessors.** You (or the operating system manager) may need to create multiple directories if different Oracle Database users will run different preprocessors. If you want to prevent one set of users from using one preprocessor while allowing those users access to another preprocessor, then place the preprocessors in separate directories. If all the users need equal access, then you can place the preprocessors together in one directory. After you create these operating system directories, in SQL\*Plus, you can create a directory object for each directory.
2. **Grant the operating system user ORACLE the correct operating system privileges to run the access driver preprocessor.** In addition, protect the preprocessor program from WRITE access by operating system users other than the user responsible for managing the preprocessor program.
3. **Grant the EXECUTE privilege to each user who will run the preprocessor program in the directory object.** Do not grant this user the WRITE privilege on the directory object. Never grant users both the EXECUTE and WRITE privilege for directory objects.
4. **Grant the WRITE privilege sparingly to anyone who will manage directory objects that contain preprocessors.** This prevents database users from accidentally or maliciously overwriting the preprocessor program.
5. **Create a separate operating system directory and directory object for any data files that are required for external tables.** Ensure that these are separate from the directory and directory object used by the access directory preprocessor.

Work with the operating system manager to ensure that only the appropriate operating system users have access to this directory. Grant the ORACLE operating system user READ access to any directory that has a directory object with READ privileges granted to database users. Similarly, grant the ORACLE operating system user WRITE access to any directory that has the WRITE privilege granted to database users.

1. **Create a separate operating system directory and directory object for any files that the access driver generates.** This includes log files, bad files, and discarded files. You and the operating system manager must ensure that this directory and directory object have the proper protections, similar to those described in Guideline [5](#_bookmark2218). The database user may need to access these files when resolving problems in data files, so you and the operating system manager must determine a way for this user to read those files.
2. **Grant the CREATE ANY DIRECTORY and DROP ANY DIRECTORY privileges sparingly.** Users who have these privileges and users who have been granted the DBA role have full access to all directory objects.
3. **Consider auditing the DROP ANY DIRECTORY privilege.** See ["Auditing Privileges" on page 9-26](#_bookmark1785) for more information about auditing privileges.
4. **Consider auditing the directory object.** See ["Auditing Directory Objects" on page 9-32](#_bookmark1840) for more information.

**See Also:** Oracle Database Utilities for more information about the

ORACLE\_DATAPUMP access driver

### Guidelines for Securing a Database Installation and Configuration

For this release, changes were made to the default configuration of Oracle Database to make it more secure. The recommendations in this section augment the new, secure default configuration.

Follow these guidelines to secure the database installation and configuration:

1. Before you begin an Oracle Database installation on UNIX systems, ensure that the umask value is 022 for the Oracle owner account.

See *Oracle Database Administrator's Reference for Linux and UNIX-Based Operating Systems* for more information about managing Oracle Database on Linux and UNIX systems.

1. Install only what is required.

**Options and Products**: The Oracle Database CD pack contains products and options in addition to the database. Install additional products and options only as necessary. Use the Custom Installation feature to avoid installing unnecessary products, or perform a typical installation, and then deinstall options and products that are not required. There is no need to maintain additional products and options if they are not being used. They can always be properly installed, as required.

**Sample Schemas**: Oracle Database provides sample schemas to provide a common platform for examples. If your database will be used in a production environment, then do not install the sample schema. If you have installed the sample schema on a test database, then before going to production, remove or

relock the sample schema accounts. See *Oracle Database Sample Schemas* for more information about the sample schemas.

1. During installation, when you are prompted for a password, create a secure password.

Follow Guidelines [1](#_bookmark2170), [5](#_bookmark2174), and [6](#_bookmark2178) in ["Guidelines for Securing Passwords" on page 10-7](#_bookmark2166).

1. Immediately after installation, lock and expire default user accounts.

See Guideline [2](#_bookmark2152) in ["Guidelines for Securing User Accounts and Privileges" on](#_bookmark2130) [page 10-2](#_bookmark2130).

### Guidelines for Securing the Network

Security for network communications is improved by using client, listener, and network guidelines to ensure thorough protection. Using SSL is an essential element in these lists, enabling top security for authentication and communications.

These guidelines are as follows:

* [Securing the Client Connection](#_bookmark2228)
* [Securing the Network Connection](#_bookmark2239)
* [Securing a Secure Sockets Layer Connection](#_bookmark2269)

#### Securing the Client Connection

Because authenticating client computers is problematic, typically, user authentication is performed instead. This approach avoids client system issues that include falsified IP addresses, hacked operating systems or applications, and falsified or stolen client system identities. Nevertheless, the following guidelines improve the security of client connections:

1. Enforce access controls effectively and authenticate clients stringently.

By default, Oracle allows operating system-authenticated logins only over secure connections, which precludes using Oracle Net and a shared server configuration. This default restriction prevents a remote user from impersonating another operating system user over a network connection.

Setting the initialization parameter REMOTE\_OS\_AUTHENT to TRUE forces the database to accept the client operating system user name received over an unsecure connection and use it for account access. Because clients, such as PCs, are not trusted to perform operating system authentication properly, it is poor security practice to use this feature.

The default setting, REMOTE\_OS\_AUTHENT = FALSE, creates a more secure configuration that enforces proper, server-based authentication of clients connecting to an Oracle database. Be aware that the REMOTE\_OS\_AUTHENT was deprecated in Oracle Database Release 11*g* (11.1) and is retained only for backward compatibility.

You should not alter the default setting of the REMOTE\_OS\_AUTHENT initialization parameter, which is FALSE.

Setting this parameter to FALSE does not mean that users cannot connect remotely. It means that the database will not trust that the client has already authenticated, and will therefore apply its standard authentication processes.

Be aware that the REMOTE\_OS\_AUTHENT parameter was deprecated in Oracle Database 11*g* Release 1 (11.1), and is retained only for backward compatibility.

1. Configure the connection to use encryption.

Oracle network encryption makes eavesdropping difficult. To learn how to configure encryption, see *Oracle Database Advanced Security Administrator's Guide*.

1. Set up strong authentication.

See *Oracle Database Advanced Security Administrator's Guide* for more information about using Kerberos and public key infrastructure (PKI).

#### Securing the Network Connection

Protecting the network and its traffic from inappropriate access or modification is the essence of network security. You should consider all paths the data travels, and assess the threats on each path and node. Then, take steps to lessen or eliminate those threats and the consequences of a security breach. In addition, monitor and audit to detect either increased threat levels or penetration attempts.

To manage network connections, you can use Oracle Net Manager. For an introduction to using Oracle Net Manager, see *Oracle Database 2 Day DBA*. See also *Oracle Database Net Services Administrator's Guide*.

The following practices improve network security:

1. Use Secure Sockets Layer (SSL) when administering the listener.

See ["Securing a Secure Sockets Layer Connection" on page 10-16](#_bookmark2269) for more information.

1. Monitor listener activity.

You can monitor listener activity by using Enterprise Manager Database Control. In the Database Control home page, under General, click the link for your listener. The Listener page appears. This page provides detailed information, such as the category of alert generated, alert messages, when the alert was triggered, and so on. This page provides other information as well, such as performance statistics for the listener.

1. Prevent online administration by requiring the administrator to have the write privilege on the listener password and on the listener.ora file on the server.
   1. Add or alter this line in the listener.ora file:

ADMIN\_RESTRICTIONS\_LISTENER=ON

* 1. Use RELOAD to reload the configuration.
  2. Use SSL when administering the listener by making the TCPS protocol the first entry in the address list, as follows:

LISTENER= (DESCRIPTION=

(ADDRESS\_LIST= (ADDRESS=

(PROTOCOL=tcps)

(HOST = sales.us.example.com) (PORT = 8281)))

To administer the listener remotely, you define the listener in the listener.ora file on the client computer. For example, to access listener USER281 remotely, use the following configuration:

user281 = (DESCRIPTION =

(ADDRESS = (PROTOCOL = tcps)

(HOST = sales.us.example.com) (PORT = 8281))

)

)

For more information about the parameters in listener.ora, see *Oracle Database Net Services Reference*.

1. Do not set the listener password.

Ensure that the password has not been set in the listener.ora file. The local operating system authentication will secure the listener administration. The remote listener administration is disabled when the password has not been set. This prevents brute force attacks of the listener password.

The listener password has been deprecated in this release. It will not be supported in the next release of Oracle Database.

1. When a host computer has multiple IP addresses associated with multiple network interface controller (NIC) cards, configure the listener to the specific IP address.

This allows the listener to listen on all the IP addresses. You can restrict the listener to listen on a specific IP address. Oracle recommends that you specify the specific IP addresses on these types of computers, rather than allowing the listener to listen on all IP addresses. Restricting the listener to specific IP addresses helps to prevent an intruder from stealing a TCP end point from under the listener process.

1. Restrict the privileges of the listener, so that it cannot read or write files in the database or the Oracle server address space.

This restriction prevents external procedure agents spawned by the listener (or procedures executed by an agent) from inheriting the ability to perform read or write operations. The owner of this separate listener process should not be the owner that installed Oracle Database or executes the Oracle Database instance (such as ORACLE, the default owner).

For more information about configuring external procedures in the listener, see

*Oracle Database Net Services Administrator's Guide*.

1. Use encryption to secure the data in flight.

See *Oracle Database 2 Day + Security Guide* and *Oracle Database Advanced Security Administrator's Guide* for more information about network data encryption.

1. Use a firewall.

Appropriately placed and configured firewalls can prevent outside access to your databases.

* Keep the database server behind a firewall. Oracle Database network infrastructure, Oracle Net (formerly known as Net8 and SQL\*Net), provides support for a variety of firewalls from various vendors. Supported

proxy-enabled firewalls include Gauntlet from Network Associates and Raptor from Axent. Supported packet-filtering firewalls include PIX Firewall from Cisco, and supported stateful inspection firewalls (more sophisticated packet-filtered firewalls) include Firewall-1 from CheckPoint.

* Ensure that the firewall is placed outside the network to be protected.
* Configure the firewall to accept only those protocols, applications, or client/server sources that you know are safe.
* Use a product such as Oracle Connection Manager to manage multiplex multiple client network sessions through a single network connection to the database. It can filter on source, destination, and host name. This product enables you to ensure that connections are accepted only from physically secure terminals or from application Web servers with known IP addresses. (Filtering on IP address alone is not enough for authentication, because it can be falsified.)

1. Prevent unauthorized administration of the Oracle listener.

For more information about the listener, see *Oracle Database Net Services* *Administrator's Guide*.

1. Check network IP addresses.

Use the Oracle Net *valid node checking* security feature to allow or deny access to Oracle server processes from network clients with specified IP addresses. To use this feature, set the following sqlnet.ora configuration file parameters:

tcp.validnode\_checking = YES tcp.excluded\_nodes = {list of IP addresses} tcp.invited\_nodes = {list of IP addresses}

The tcp.validnode\_checking parameter enables the feature. The tcp.excluded\_

nodes and tcp.invited\_nodes parameters deny and enable specific client IP addresses from making connections to the Oracle listener. This helps to prevent potential Denial of Service attacks.

You can use Oracle Net Manager to configure these parameters. See *Oracle* *Database Net Services Administrator's Guide* for more information.

1. Encrypt network traffic.

If possible, use Oracle Advanced Security to encrypt network traffic among clients, databases, and application servers. *Oracle Database 2 Day + Security Guide* provides an introduction to network encryption. For detailed information about network encryption, see *Oracle Database Advanced Security Administrator's Guide*.

1. Secure the host operating system (the system on which Oracle Database is installed).

Secure the host operating system by disabling all unnecessary operating system services. Both UNIX and Windows provide a variety of operating system services, most of which are not necessary for typical deployments. These services include FTP, TFTP, TELNET, and so forth. Be sure to close both the UDP and TCP ports for each service that is being disabled. Disabling one type of port and not the other does not make the operating system more secure.

#### Securing a Secure Sockets Layer Connection

Secure Sockets Layer (SSL) is the Internet standard protocol for secure communication, providing mechanisms for data integrity and data encryption. These mechanisms can protect the messages sent and received by you or by applications and servers, supporting secure authentication, authorization, and messaging through certificates and, if necessary, encryption. Good security practices maximize protection and minimize gaps or disclosures that threaten security. The following guidelines show the

cautious attention to detail necessary for the successful use of SSL. For detailed information about Oracle SSL configuration, see *Oracle Database Advanced Security Administra**tor's Guide*.

1. Ensure that configuration files (for example, for clients and listeners) use the correct port for SSL, which is the port configured upon installation.

You can run HTTPS on any port, but the standards specify port 443, where any HTTPS-compliant browser looks by default. The port can also be specified in the URL, for example:

https://secure.example.com:4445/

If a firewall is in use, then it too must use the same ports for secure (SSL) communication.

1. Ensure that TCPS is specified as the PROTOCOL in the ADDRESS parameter in the tnsnames.ora file (typically on the client or in the LDAP directory).

An identical specification must appear in the listener.ora file (typically in the

$ORACLE\_HOME/network/admin directory).

1. Ensure that the SSL mode is consistent for both ends of every communication. For example, the database (on one side) and the user or application (on the other) must have the same SSL mode.

The mode can specify either client or server authentication (one-way), both client and server authentication (two-way), or no authentication.

1. Ensure that the server supports the client cipher suites and the certificate key algorithm in use.
2. **Enable DN matching for both the server and client, to** **prevent the server from falsifying its identity to the client during connections.**

This setting ensures that the server identity is correct by matching its global database name against the DN from the server certificate.

You can enable DN matching in the tnsnames.ora file. For example:

set:SSL\_SERVER\_CERT\_DN="cn=finance,cn=OracleContext,c=us,o=example"

Otherwise, a client application would not check the server certificate, which could allow the server to falsify its identity.

1. Do not remove the encryption from your RSA private key inside your server.key file, which requires that you enter your pass phrase to read and parse this file.

**Note:** A server without SSL does not require a pass phrase.

If you decide your server is secure enough, you could remove the encryption from the RSA private key while preserving the original file. This enables system boot scripts to start the database server, because no pass phrase is needed. Ideally, restrict permissions to the root user only, and have the Web server start as root, but then log on as another user. Otherwise, anyone who gets this key can impersonate you on the Internet, or decrypt the data that was sent to the server.

**See Also:**

* + *Oracle Database Advanced Security Administrator's Guide* for general SSL information, including configuration
  + *Oracle Database Net Services Reference* for TCP-related parameters in sqlnet.ora

### Guidelines for Auditing

This section contains:

* + [Auditing Sensitive Information](#_bookmark2287)
  + [Keeping Audited Information Manageable](#_bookmark2291)
  + [Auditing Typical Database Activity](#_bookmark2294)
  + [Auditing Suspicious Database Activity](#_bookmark2298)
  + [Recommended Audit Settings](#_bookmark2300)

#### Auditing Sensitive Information

Be aware that sensitive data, such as credit card numbers, appear in the fine-grained audit trail if you collect SQL text. For standard auditing, setting the AUDIT\_TRAIL initialization parameter to DB, EXTENDED or XML, EXTENDED enables the collection of SQL text. For fine-grained auditing, you would set the audit\_trail parameter of the DBMS\_FGA PL/SQL package to DBMS\_FGA.DB + DBMS\_FGA.EXTENDED or DBMS\_FGA.XML + DBMS\_FGA.EXTENDED.

If you have sensitive data that is being audited, consider using either of the following solutions:

* + **Move the audit trail out of the SYSTEM tablespace and into SYSAUX or another tablespace.** See ["Moving the Database Audit Trail to a Different Tablespace" on page 9-60](#_bookmark2016). Afterwards, encrypt this tablespace. For more information about tablespace encryption, see *Oracle Database Advanced Security Administrator's Guide*.
  + **Do not enable the collection of SQL text in the audit trail.** Use the following settings instead:
* **Standard auditing:** Set the AUDIT\_TRAIL initialization parameter to DB, OS, or XML. See ["Configuring Standard Auditing with the AUDIT\_TRAIL Initialization Parameter" on page 9-8](#_bookmark1655).
* **Fine-grained auditing:** Set the DBMS\_FGA.ADD\_POLICY audit\_trail parameter to DBMS\_FGA.DB or DBMS\_FGA.XML. See ["Creating a Fine-Grained Audit Policy" on page 9-40](#_bookmark1922).

#### Keeping Audited Information Manageable

Although auditing is relatively inexpensive, limit the number of audited events as much as possible. This minimizes the performance impact on the execution of audited statements and the size of the audit trail, making it easier to analyze and understand.

Follow these guidelines when devising an auditing strategy:

1. Evaluate your reason for auditing.

After you have a clear understanding of the reasons for auditing, you can devise an appropriate auditing strategy and avoid unnecessary auditing.

For example, suppose you are auditing to investigate suspicious database activity. This information by itself is not specific enough. What types of suspicious database activity do you suspect or have you noticed? A more focused auditing strategy might be to audit unauthorized deletions from arbitrary tables in the database. This purpose narrows the type of action being audited and the type of object being affected by the suspicious activity.

1. Audit knowledgeably.

Audit the minimum number of statements, users, or objects required to get the targeted information. This prevents unnecessary audit information from cluttering the meaningful information and using valuable space in the SYSTEM tablespace.

Balance your need to gather sufficient security information with your ability to store and process it.

For example, if you are auditing to gather information about database activity, then determine exactly what types of activities you want to track, audit only the activities of interest, and audit only for the amount of time necessary to gather the information that you want. As another example, do not audit *objects* if you are only interested in logical I/O information for each session.

1. Before you implement an auditing strategy, consult your legal department.

You should have the legal department of your organization review your audit strategy. Because your auditing will monitor other users in your organization, you must ensure that you are correctly following the compliance and corporate policy of your site.

#### Auditing Typical Database Activity

When your purpose for auditing is to gather historical information about particular database activities, use the following guidelines:

1. Audit only pertinent actions.

At a minimum, audit user access, the use of system privileges, and changes to the database schema structure. To avoid cluttering meaningful information with useless audit records and reduce the amount of audit trail administration, only audit the targeted database activities. Remember also that auditing too much can affect database performance.

For example, auditing changes to all tables in a database produces far too many audit trail records and can slow down database performance. However, auditing changes to critical tables, such as salaries in a Human Resources table, is useful.

You can audit specific actions by using fine-grained auditing, which is described in ["Auditing Specific Activities with Fine-Grained Auditing" on page 9-36](#_bookmark1894).

1. Archive audit records and purge the audit trail.

After you collect the required information, archive the audit records of interest and then purge the audit trail of this information. See the following sections:

* + ["Archiving the Database Audit Trail" on page 9-61](#_bookmark2022)
  + ["Archiving the Operating System Audit Trail" on page 9-65](#_bookmark2040)
  + ["Purging Audit Trail Records" on page 9-65](#_bookmark2044)
  + ["Controlling the Size of the Database Audit Trail" on page 9-59](#_bookmark2012)
  + ["Setting the Size of the Operating System Audit Trail" on page 9-62](#_bookmark2034)

1. Remember your company’s privacy considerations.

Privacy regulations often lead to additional business privacy policies. Most privacy laws require businesses to monitor access to personally identifiable information (PII), and monitoring is implemented by auditing. A business-level privacy policy should address all relevant aspects of data access and user accountability, including technical, legal, and company policy concerns.

1. Check the Oracle Database log files for additional audit information

The log files generated by Oracle Database contain useful information that you can use when auditing a database. For example, an Oracle database creates an alert file to record STARTUP and SHUTDOWN operations, and structural changes such as adding data files to the database.

For example, if you want to audit committed or rolled back transactions, you can use the redo log files.

#### Auditing Suspicious Database Activity

When you audit to monitor suspicious database activity, use the following guidelines:

1. First audit generally, and then specifically.

When you start to audit for suspicious database activity, often not much information is available to target specific users or schema objects. Therefore, set audit options more generally at first, that is, by using the standard audit options described in [Chapter 9, "Verifying Security Access with Auditing"](#_bookmark1602) explains how you can use the standard audit options to audit SQL statements, schema objects, privileges, and so on.

After you have recorded and analyzed the preliminary audit information, disable general auditing, and then audit specific actions. You can use fine-grained auditing, which is described in ["Auditing Specific Activities with Fine-Grained](#_bookmark1894) [Auditing" on page 9-36](#_bookmark1894), to audit specific actions. Continue this process until you have gathered enough evidence to draw conclusions about the origin of the suspicious database activity.

1. Audit common suspicious activities.

Common suspicious activities are as follows:

* + Users who access the database during unusual hours
  + Multiple failed user login attempts
  + Login attempts by non-existent users

In addition, monitor users who share accounts or multiple users who are logging in from the same IP address. You can query the DBA\_AUDIT\_SESSION data dictionary view to find this kind of activity. For a very granular approach, create fine-grained audit policies.

1. Protect the audit trail.

When auditing for suspicious database activity, protect the audit trail so that audit information cannot be added, changed, or deleted without being audited. You can audit the standard audit trail by using the AUDIT SQL statement.

For example:

AUDIT SELECT ON SYS.AUD$ BY ACCESS;

See also ["Auditing the Database Audit Trail" on page 9-61](#_bookmark2019).

To audit the fine-grained audit trail, as user SYS, you would enter the following statement:

AUDIT SELECT ON SYS.FGA$ BY ACCESS;

If you have Oracle Database Vault enabled, you can further protect the SYS.AUDIT$, SYSTEM.AUD$, SYS.FGA$, and SYS.FGA\_LOG$ tables by enclosing them in a realm. (In an Oracle Database Vault environment, the AUD$ table is moved to the SYSTEM schema when Oracle Label Security is enabled. SYS.AUD$ becomes a synonym for the SYSTEM.AUD$ table.) See *Oracle Database Vault Administrator's Guide* for more information.

#### Recommended Audit Settings

**Database schema** **or structure changes.** Use the following AUDIT statement settings:

* + AUDIT ALTER ANY PROCEDURE BY ACCESS;
  + AUDIT ALTER ANY TABLE BY ACCESS;
  + AUDIT ALTER DATABASE BY ACCESS;
  + AUDIT ALTER SYSTEM BY ACCESS;
  + AUDIT CREATE ANY EDITION;
  + AUDIT CREATE ANY JOB BY ACCESS;
  + AUDIT CREATE ANY LIBRARY BY ACCESS;
  + AUDIT CREATE ANY PROCEDURE BY ACCESS;
  + AUDIT CREATE ANY TABLE BY ACCESS;
  + AUDIT CREATE EXTERNAL JOB BY ACCESS;
  + AUDIT DROP ANY EDITION;
  + AUDIT DROP ANY PROCEDURE BY ACCESS;
  + AUDIT DROP ANY TABLE BY ACCESS;

**Database access and privileges.** Use these AUDIT statement settings:

* + AUDIT ALTER PROFILE BY ACCESS;
  + AUDIT ALTER USER BY ACCESS;
  + AUDIT AUDIT SYSTEM BY ACCESS;
  + AUDIT CREATE PUBLIC DATABASE LINK BY ACCESS;
  + AUDIT CREATE SESSION BY ACCESS;
  + AUDIT CREATE USER BY ACCESS;
  + AUDIT DROP PROFILE BY ACCESS;
  + AUDIT DROP USER BY ACCESS;
  + AUDIT EXEMPT ACCESS POLICY BY ACCESS;
  + AUDIT GRANT ANY OBJECT PRIVILEGE BY ACCESS;
  + AUDIT GRANT ANY PRIVILEGE BY ACCESS;
  + AUDIT GRANT ANY ROLE BY ACCESS;
  + AUDIT ROLE BY ACCESS;

### Addressing the CONNECT Role Change

The CONNECT role was introduced with Oracle Database version 7, which added new and robust support for database roles. The CONNECT role is used in sample code, applications, documentation, and technical papers. In Oracle Database 10*g* Release 2 (10.2), the CONNECT role was changed. If you are upgrading from a release earlier than Oracle Database 10.2 to the current release, then read this section.

This section contains:

* + [Why Was the CONNECT Role Changed?](#_bookmark2306)
  + [How the CONNNECT Role Change Affects Applications](#_bookmark2308)
  + [How the CONNECT Role Change Affects Users](#_bookmark2316)
  + [Approaches to Addressing the CONNECT Role Change](#_bookmark2324)

#### Why Was the CONNECT Role Changed?

The CONNECT role was originally established with the following privileges:

|  |  |
| --- | --- |
| **Privileges A-C** | **Privileges C** |
| ALTER SESSION | CREATE SESSION |
| CREATE CLUSTER | CREATE SYNONYM |
| CREATE DATABASE LINK | CREATE TABLE |
| CREATE SEQUENCE | CREATE VIEW |

Beginning in Oracle Database 10*g* Release 2, the CONNECT role has only the CREATE SESSION privilege, all other privileges are removed.

Although the CONNECT role was frequently used to provision new accounts in Oracle Database, connecting to the database does not require all those privileges. Making this change enables you to enforce good security practices more easily.

Each user should have only the privileges needed to perform his or her tasks, an idea called the principle of least privilege. Least privilege mitigates risk by limiting privileges, so that it remains easy to do what is needed while concurrently reducing the ability to do inappropriate things, either inadvertently or maliciously.

#### Howthe CONNNECT Role Change Affects Applications

The effects of the changes to the CONNECT role can be seen in database upgrades, account provisioning, and installation of applications using new databases.

###### How the CONNECT Role Change Affects Database Upgrades

Upgrading your existing Oracle database to Oracle Database 10*g* Release 2 (10.2) automatically changes the CONNECT role to have only the CREATE SESSION privilege. Most applications are not affected because the applications objects already exist: no new tables, views, sequences, synonyms, clusters, or database links need to be created.

Applications that create tables, views, sequences, synonyms, clusters, or database links, or that use the ALTER SESSION command dynamically, may fail due to insufficient privileges.

###### How the CONNECT Role Change Affects Account Provisioning

If your application or DBA grants the CONNECT role as part of the account provisioning process, then only CREATE SESSION privileges are included. Any additional privileges must be granted either directly or through another role.

This issue can be addressed by creating a new customized database role.

**See Also:** [Approaches to Addressing the CONNECT Role Change](#_bookmark2324) [on page 10-24](#_bookmark2324)

###### How the CONNECT Role Change Affects Applications Using New Databases

New databases created using the Oracle Database 10*g* Release 2 (10.2) Utility (DBCA), or using database creation templates generated from DBCA, define the CONNECT role with only the CREATE SESSION privilege. Installing an application to use a new database may fail if the database schema used for the application is granted privileges solely through the CONNECT role.

#### Howthe CONNECT Role Change Affects Users

The change to the CONNECT role affects three classes of users differently: general users, application developers, and client/server applications.

###### How the CONNECT Role Change Affects General Users

The new CONNECT role supplies only the CREATE SESSION privilege. Users who connect to the database to use an application are not affected, because the CONNECT role still has the CREATE SESSION privilege.

However, appropriate privileges will not be present for a certain set of users if they are provisioned solely with the CONNECT role. These are users who create tables, views, sequences, synonyms, clusters, or database links, or use the ALTER SESSION command. The privileges they need are no longer provided with the CONNECT role. To authorize the additional privileges needed, the database administrator must create and apply additional roles for the appropriate privileges, or grant them directly to the users who need them.

Note that the ALTER SESSION privilege is required for setting events. Few database users should require the ALTER SESSION privilege.

ALTER SESSION SET EVENTS ........

The alter session privilege is *not* required for other alter session commands.

ALTER SESSION SET NLS\_TERRITORY = FRANCE;

###### How the CONNECT Role Change Affects Application Developers

Application developers provisioned solely with the CONNECT role do not have appropriate privileges to create tables, views, sequences, synonyms, clusters, or database links, nor to use the ALTER SESSION statement. The database administrator must either create and apply additional roles for the appropriate privileges, or grant them directly to the application developers who need them.

###### How the CONNECT Role Change Affects Client Server Applications

Most client/server applications that use dedicated user accounts will not be affected by this change. However, applications that create private synonyms or temporary tables using dynamic SQL in the user schema during account provisioning or run-time

operations will be affected. They will require additional roles or grants to acquire the system privileges appropriate to their activities.

#### Approaches to Addressing the CONNECT Role Change

Oracle recommends the following three approaches to address the impact of this change.

###### Approach 1: Create a New Database Role

The privileges removed from the CONNECT role can be managed by creating a new database role.

First, connect to the upgraded Oracle database and create a new database role. The following example uses a role called my\_app\_developer.

CREATE ROLE my\_app\_developer;

GRANT CREATE TABLE, CREATE VIEW, CREATE SEQUENCE, CREATE SYNONYM, CREATE CLUSTER,

CREATE DATABASE LINK, ALTER SESSION TO my\_app\_developer;

Second, determine which users or database roles have the CONNECT role, and grant the new role to these users or roles.

SELECT USER$.NAME, ADMIN\_OPTION, DEFAULT\_ROLE FROM USER$, SYSAUTH$, DBA\_ROLE\_PRIVS

WHERE PRIVILEGE# =

(SELECT USER# FROM USER$ WHERE NAME = 'CONNECT') AND USER$.USER# = GRANTEE#

AND GRANTEE = USER$.NAME

AND GRANTED\_ROLE = 'CONNECT';

NAME ADMIN\_OPTI DEF

------------------------------ ---------- --- R1 YES YES

R2 NO YES

GRANT my\_app\_developer TO R1 WITH ADMIN OPTION; GRANT my\_app\_developer TO R2;

You can determine the privileges that users require by using Oracle Auditing. The audit information can then be analyzed and used to create additional database roles with finer granularity.

Privileges not used can then be revoked for specific users. Note that before auditing, the database initialization parameter AUDIT\_TRAIL must be initialized and the database restarted.

AUDIT CREATE TABLE, CREATE SEQUENCE, CREATE SYNONYM, CREATE DATABASE LINK, CREATE CLUSTER, CREATE VIEW, ALTER SESSION;

Database privilege usage can now be monitored periodically.

SELECT USERID, NAME FROM AUD$, SYSTEM\_PRIVILEGE\_MAP WHERE - PRIV$USED = PRIVILEGE;

USERID NAME

------------------------------ ---------------- ACME CREATE TABLE

ACME CREATE SEQUENCE

ACME CREATE TABLE

ACME ALTER SESSION

APPS CREATE TABLE

APPS CREATE TABLE

APPS CREATE TABLE

APPS CREATE TABLE

8 rows selected.

###### Approach 2: Restore CONNECT Privileges

Starting with Oracle Database 10*g* Release 2 (10.2), Oracle provided a script called rstrconn.sql in the $ORACLE\_HOME/rdbms/admin directory. After a database upgrade or new database creation, this script can be used to grant the privileges that were removed from the CONNECT role in Oracle Database 10*g* Release 2 (10.2).

If this approach is used, then privileges that are not used should be revoked from users who do not need them. To identify such privileges and users, the database must be restarted with the database initialization parameter AUDIT\_TRAIL initialized, for example, AUDIT\_TRAIL=DB. Oracle Database auditing should then be turned on to monitor what privileges are used, as follows:

AUDIT CREATE TABLE, CREATE SEQUENCE, CREATE SYNONYM, CREATE DATABASE LINK, CREATE CLUSTER, CREATE VIEW, ALTER SESSION;

Database privilege usage can also be monitored periodically.

SELECT USERID, NAME FROM AUD$, SYSTEM\_PRIVILEGE\_MAP WHERE - PRIV$USED = PRIVILEGE;

USERID NAME

------------------------------ ---------------- ACME CREATE TABLE

ACME CREATE SEQUENCE

ACME CREATE TABLE

ACME ALTER SESSION

APPS CREATE TABLE

APPS CREATE TABLE

APPS CREATE TABLE

APPS CREATE TABLE

8 rows selected.

**New View Showing CONNECT Grantees** A new view enables administrators who continue using the old CONNECT role to see quickly which users have that role.

[Table 10–1](#_bookmark2328) shows the columns in the new DBA\_CONNECT\_ROLE\_GRANTEES view.

***Table 10–1 Columns and Contents for DBA\_CONNECT\_ROLE\_GRANTEES***

|  |  |
| --- | --- |
| **Column Name** | **Contents** |
| Grantee | User granted the CONNECT role |
| Path\_of\_connect\_role\_grant | Role (or nested roles) by which the user is granted CONNECT |
| Admin\_opt | VARCHAR2(3), YES if user has the ADMIN option on CONNECT; otherwise, NO |

###### Approach 3: Conduct Least Privilege Analysis

Oracle partners and application providers should use this approach to deliver more secure products to the Oracle customer base. The principle of least privilege mitigates risk by limiting privileges to the minimum set required to perform a given function.

For each class of users that the analysis shows need the same set of privileges, create a role with only those privileges. Remove all other privileges from those users, and assign that role to those users. As needs change, you can grant additional privileges, either directly or through these new roles, or create new roles to meet new needs. This approach helps to ensure that inappropriate privileges have been limited, thereby reducing the risk of inadvertent or malicious harm.

## Glossary

application context

A name-value pair that enables an application to access session information about a user, such as the user ID or other user-specific information, and then securely pass this data to the database.

See also [**global application context**](#_bookmark2340).

application role

A database role that is granted to application users and that is secured by embedding passwords inside the application.

See also [**secure application role**](#_bookmark2351).

certificate

An ITU x.509 v3 standard data structure that securely binds an identify to a public key.

A certificate is created when an entity's public key is signed by a trusted identity, a certificate authority. The certificate ensures that the entity's information is correct, and that the public key belongs to that entity.

A certificate contains the entity's name, identifying information, and public key. It is also likely to contain a serial number, expiration date, and information about the rights, uses, and privileges associated with the certificate. Finally, it contains information about the certificate authority that issued it.

certificate revocation list (CRL)

See [**CRL**](#_bookmark2336).

Classless Inter-Domain Routing

See [**CIDR**](#_bookmark2335).

cleartext

Unencrypted plain text.

CIDR

The standard notation used for IP addresses. In CIDR notation, an IPv6 subnet is denoted by the subnet prefix and the size in bits of the prefix (in decimal), separated by the slash (/) character. For example, fe80:0000:0217:f2ff::/64 denotes a subnet with addresses fe80:0000:0217:f2ff:0000:0000:0000:0000 through fe80:0000:0217:f2ff:ffff:ffff:ffff:ffff. The CIDR notation includes support for IPv4 addresses. For example, 192.0.2.1/24 denotes the subnet with addresses

192.0.2.1 through 192.0.2.255.

CRL

A set of signed data structures that contain a list of revoked [**certificate**](#_bookmark2333)**s**. The authenticity and integrity of the CRL is provided by a digital signature appended to it. Usually, the CRL signer is the same entity that signed the issued certificate.

definer’s rights procedure

A procedure (or program unit) that executes with the privileges of its owner, not its current user. Definer's rights subprograms are bound to the schema in which they are located.

For example, assume that user blake and user scott each have a table called dept in their respective user schemas. If user blake calls a definer's rights procedure, which is owned by user scott, to update the dept table, then this procedure will update the dept table in the scott schema. This is because the procedure executes with the privileges of the user who owns (defined) the procedure (that is, scott).

See also [**invoker’s rights procedure**](#_bookmark2342).

decryption

Decoding an encyrpted message so that it is readable.

denial-of-service (DoS) attack

An attack that renders a Web site inaccessible or unusable. The denial-of-service attack can occur in many different ways but frequently includes attacks that cause the site to crash, reject connections, or perform too slowly to be usable. DoS attacks come in two forms:

* + Basic denial-of-service attacks, which require only one or a few computers
  + Distributed denial-of-service (DDoS) attacks, which require many computers to execute

directly granted role

A [**role**](#_bookmark2349) that has been granted directly to the user, as opposed to an [**indirectly granted**](#_bookmark2341)[**role**](#_bookmark2341).

encryption

Disguising a message, rendering it unreadable to all but the intended recipient.

forced cleanup

The ability to forcibly cleanup (that is, remove) all audit records from the database. To accomplish this, you set the USE\_LAST\_ARCH\_TIMESTAMP argument of the DBMS\_AUDIT\_ MGMT.CLEAN\_AUDIT\_TRAIL procedure to FALSE.

See also [**purge job**](#_bookmark2348).

Forwardable Ticket Granting Ticket

A special Kerberos ticket that can be forwarded to proxies, permitting the proxy to obtain additional Kerberos tickets on behalf of the client for proxy authentication.

See also [**Kerberos ticket**](#_bookmark2345).

global application context

A name-value pair that enables application context values to be accessible across database sessions.

See also [**application context**](#_bookmark2331).

indirectly granted role

A [**role**](#_bookmark2349) granted to a user through another role that has already been granted to this user. Then you grant the role2 and role3 roles to the role1 role. Roles role2 and role3 are now under role1. This means psmith has been indirectly granted the roles role2 and role3, in addition to the direct grant of role1. Enabling the direct role1 for psmith enables the indirect roles role2 and role3 for this user as well.

integrity

A guarantee that the contents of a message received were not altered from the contents of the original message sent.

invoker’s rights procedure

A procedure (or program unit) that executes with the privileges of the current user, that is, the user who invokes the procedure. These procedures are not bound to a particular schema. They can be run by a variety of users and allow multiple users to manage their own data by using centralized application logic. Invoker's rights procedures are created with the AUTHID clause in the declaration section of the procedure code.

For example, assume that user blake and user scott each have a table called dept in their respective user schemas. If user blake calls an invoker’s rights procedure, which is owned by user scott, to update the dept table, then this procedure will update the dept table in the blake schema. This is because the procedure executes with the privileges of the user who invoked the procedure (that is, blake.).

See also [**definer’s rights procedure**](#_bookmark2337).

KDC

A computer that issues Kerberos tickets. See also [**Kerberos ticket**](#_bookmark2345).

Kerberos ticket

A temporary set of electronic credentials that verify the identity of a client for a particular service. Also referred to as a service ticket.

Key Distribution Center (KDC)

See [**KDC**](#_bookmark2343).

last archive timestamp

A timestamp that indicates the timestamp of the last archived audit record. For the database audit trail, this timestamp indicates the last audit record archived. For operating system audit files, it indicates the highest last modified timestamp property of the audit file that was archived. To set this timestamp, you use the DBMS\_AUDIT\_ MGMT.SET\_LAST\_ARCHIVE\_TIMESTAMP PL/SQL procedure.

See also [**purge job**](#_bookmark2348).

lightweight user session

A user session that contains only information pertinent to the application that the user is logging onto. The lightweight user session does not hold its own database resources, such as transactions and cursors; hence it is considered "lightweight." Lightweight user sessions consume far less system resources than traditional database session.

Because lightweight user sessions consume much fewer server resources, a lightweight user session can be dedicated to each end user and can persist for as long as the application deems necessary.

mandatory auditing

Activities that are audited by default, regardless of whether or not auditing was enabled. These activities include connections to the instance with administrator privileges, database startups, and database shutdowns. Oracle Database writes these activities to the operating system audit trail.

namespace

In Oracle Database security, the name of an application context. You create this name in a CREATE CONTEXT statement.

Oracle Virtual Private Database

A set of features that enables you to create security policies to control database access at the row and column level. Essentially, Oracle Virtual Private Database adds a dynamic WHERE clause to a SQL statement that is issued against the table, view, or synonym to which an Oracle Virtual Private Database security policy was applied.

PUBLIC role

A special role that every database account automatically has. By default, it has no privileges assigned to it, but it does have grants to many Java objects. You cannot drop the PUBLIC role, and a manual grant or revoke of this role has no meaning, because the user account will always assume this role. Because all database user accounts assume the PUBLIC role, it does not appear in the DBA\_ROLES and SESSION\_ROLES data dictionary views.

purge job

A database job created by the DBMS\_AUDIT\_MGMT.CREATE\_PURGE\_JOB procedure, which manages the deletion of the audit trail. A database administrator schedules, enables, and disables the purge job. When the purge job becomes active, it deletes audit records from the database audit tables, or it deletes Oracle Database operating system audit files.

See also [**forced cleanup**](#_bookmark2339), [**last archive timestamp**](#_bookmark2346).

role

A named group of related privileges that you grant as a group to users or other roles. See also [**indirectly granted role**](#_bookmark2341).

salt

In cryptography, a way to strengthen the security of encrypted data. Salt is a random string that is added to the data before it is encrypted, making it more difficult for attackers to steal the data by matching patterns of ciphertext to known ciphertext samples. Salt is often also added to passwords, before the passwords are encrypted, to avoid dictionary attacks, a method that unethical hackers (attackers) use to steal passwords. The encrypted salted values make it difficult for attackers to match the hash value of encrypted passwords (sometimes called verifiers) with their dictionary lists of common password hash values.

secure application role

A database role that is granted to application users, but secured by using an invoker's right stored procedure to retrieve the role password from a database table. A secure application role password is not embedded in the application.

See also [**application role**](#_bookmark2332).

separation of duty

Restricting activities only to those users who must perform them. For example, you should not grant the SYSDBA privilege to any user. Only grant this privilege to administrative users. Separation of duty is required by many compliance policies. See ["Guidelines for Securing User Accounts and Privileges" on page 10-2](#_bookmark2131) for guidelines on granting privileges to the correct users.

service ticket

See [**Kerberos ticket**](#_bookmark2345).

wallet

A data structure used to store and manage security credentials for an individual entity.

## Index

##### A

access control

[encryption, problems not solved by, 8-1](#_bookmark1553) [enforcing, 10-13](#_bookmark2232)

[object privileges, 4-23](#_bookmark714)

[password encryption, 3-2](#_bookmark190) access control list (ACL)

examples

[external network connection for email alert, 9-44](#_bookmark1945)

[external network connections, 4-59](#_bookmark915) [wallet access, 4-59](#_bookmark916)

external network services [about, 4-50](#_bookmark882)

[adding more users or privileges, 4-53](#_bookmark896) [advantages, 4-49](#_bookmark879)

[affect of upgrade from earlier release, 4-51](#_bookmark889) [creating ACL, 4-51](#_bookmark892) [DBMS\_NETWORK\_ACL\_ADMIN package,](#_bookmark893)

[general process, 4-51](#_bookmark893)

[email alert for audit violation tutorial, 9-44](#_bookmark1946) [finding information about, 4-71](#_bookmark952)

[hosts, assigning, 4-53](#_bookmark898)

[network hosts, using wildcards to specify, 4-64](#_bookmark928)

[ORA-24247 errors, 4-51](#_bookmark889)

[order of precedence, hosts, 4-65](#_bookmark930) [port ranges, 4-66](#_bookmark932)

[privilege assignments, about, 4-66](#_bookmark934)

[privilege assignments, database administrators checking, 4-67](#_bookmark937)

[privilege assignments, users checking, 4-68](#_bookmark943) [setting precedence, multiple roles, 4-69](#_bookmark949) [setting precedence, multiple users, 4-69](#_bookmark950) [syntax for creating, 4-51](#_bookmark895)

hosts

[local host, 4-54](#_bookmark899)

[localhost setting, 4-54](#_bookmark899) wallet access

[about, 4-50](#_bookmark886)

[advantages, 4-51](#_bookmark887)

[client certificate credentials, using, 4-55](#_bookmark906) [finding information about, 4-71](#_bookmark953)

[non-shared wallets, 4-55](#_bookmark901)

[password credentials, 4-55](#_bookmark902)

[password credentials, using, 4-55](#_bookmark907) [shared database session, 4-55](#_bookmark901)

[wallets with sensitive information, 4-55](#_bookmark903) [wallets without sensitive information, 4-55](#_bookmark908)

account locking [example, 3-7](#_bookmark232)

[explicit, 3-7](#_bookmark237)

[password management, 3-6](#_bookmark228) [PASSWORD\_LOCK\_TIME profile parameter, 3-7](#_bookmark235)

ad hoc tools

[database access, security problems of, 4-21](#_bookmark701) ADM\_PARALLEL\_EXECUTE\_TASK role

[about, 4-11](#_bookmark587) ADMIN OPTION

[about, 4-38](#_bookmark801)

[revoking privileges, 4-41](#_bookmark824)

[revoking roles, 4-41](#_bookmark824)

[roles, 4-21](#_bookmark692)

[system privileges, 4-5](#_bookmark532) administrative privileges

[granting to users, 4-2](#_bookmark505) administrative user passwords

[default, importance of changing, 10-9](#_bookmark2179) administrator privileges

[access, 10-14](#_bookmark2242)

[operating system authentication, 3-25](#_bookmark330) [passwords, 3-25,](#_bookmark333) [10-9](#_bookmark2180)

[SYSDBA and SYSOPER access, centrally controlling, 3-22](#_bookmark320)

[write, on listener.ora file, 10-14](#_bookmark2242) [adump audit files directory, 9-54](#_bookmark1981)

[alerts, used in fine-grained audit policy, 9-44](#_bookmark1947) ["all permissions", 10-3](#_bookmark2147)

ALTER ANY LIBRARY statement [security guidelines, 10-3](#_bookmark2139)

ALTER privilege statement

[SQL statements permitted, 5-17](#_bookmark1086) ALTER PROCEDURE statement

[used for compiling procedures, 4-31](#_bookmark768) ALTER PROFILE statement

[password management, 3-4](#_bookmark204)

[ALTER RESOURCE COST statement, 2-14](#_bookmark149)

ALTER ROLE statement

[changing authorization method, 4-17](#_bookmark652) ALTER SESSION statement

[schema, setting current, 5-16](#_bookmark1065)

**Index-1**

[ALTER USER privilege, 2-7](#_bookmark99) ALTER USER statement

[default roles, 4-48](#_bookmark872)

[explicit account unlocking, 3-7](#_bookmark237)

[GRANT CONNECT THROUGH clause, 3-39](#_bookmark433)

[passwords, changing, 2-8](#_bookmark105)

[passwords, expiring, 3-10](#_bookmark263)

[profiles, changing, 3-10](#_bookmark262)

[REVOKE CONNECT THROUGH clause, 3-39](#_bookmark436)

[user profile, 3-4](#_bookmark205)

[altering users, 2-8](#_bookmark101) ANSI operations

[Oracle Virtual Private Database affect on, 7-34](#_bookmark1506) ANY system privilege

[guidelines for security, 10-10](#_bookmark2199) application contexts

[about, 6-1](#_bookmark1126)

[as secure data cache, 6-2](#_bookmark1140) [benefits of using, 6-2](#_bookmark1139) [bind variables, 7-4](#_bookmark1368)

[components, 6-1](#_bookmark1128)

[DBMS\_SESSION.SET\_CONTEXT procedure, 6-9](#_bookmark1187)

[driving context, 6-45](#_bookmark1337) [editions, affect on, 6-3](#_bookmark1142)

[finding errors by checking trace files, 6-45](#_bookmark1338) [finding information about, 6-45](#_bookmark1333)

global application contexts [authenticating user for multiple](#_bookmark1279)

[applications, 6-27](#_bookmark1279)

[creating, 6-23](#_bookmark1257)

[logon trigger, creating, 6-11](#_bookmark1195)

[Oracle Virtual Private Database, used with, 7-3](#_bookmark1366) [performance, 7-27](#_bookmark1485)

[policy groups, used in, 7-11](#_bookmark1422) [returning predicate, 7-3](#_bookmark1367)

[session information, retrieving, 6-7](#_bookmark1169) [support for database links, 6-16](#_bookmark1220) [types, 6-3](#_bookmark1144)

[users, nondatabase connections, 6-22,](#_bookmark1247) [6-28](#_bookmark1283) [where values are stored, 6-2](#_bookmark1131)

[*See also*](#_bookmark1242) client session-based application contexts, database session-based application contexts, global application contexts

application developers [CONNECT role change, 10-23](#_bookmark2321)

application security

[restricting wallet access to current application, 4-55](#_bookmark904)

[sharing wallet with other applications, 4-55](#_bookmark909) [specifying attributes, 6-6](#_bookmark1161)

application users who are database users

[Oracle Virtual Private Database, how it works with, 7-38](#_bookmark1532)

applications

[about security policies for, 5-1](#_bookmark993) [database users, 5-2](#_bookmark998)

[enhancing security with, 4-8](#_bookmark557) [object privileges, 5-16](#_bookmark1076)

[object privileges permitting SQL statements, 5-17](#_bookmark1081) One Big Application User authentication

[security considerations, 5-3](#_bookmark1011) [security risks of, 5-2](#_bookmark999)

[Oracle Virtual Private Database, how it works with, 7-35](#_bookmark1509)

[password handling, guidelines, 5-4](#_bookmark1019) [password protection strategies, 5-3](#_bookmark1014) [privileges, managing, 5-11](#_bookmark1032)

roles

[multiple, 4-8](#_bookmark563)

[privileges, associating with database roles, 5-14](#_bookmark1051)

[security, 4-21,](#_bookmark698) [5-2](#_bookmark1008)

[security considerations for use, 5-1](#_bookmark996) [security limitations, 7-35](#_bookmark1510)

[security policies, 7-12](#_bookmark1429)

[validating with security policies, 7-13](#_bookmark1433) AQ\_ADMINISTRATOR\_ROLE role

[about, 4-11](#_bookmark588) AQ\_USER\_ROLE role

[about, 4-11](#_bookmark589) archiving

[operating system audit files, 9-65](#_bookmark2041) [standard audit trail, 9-61](#_bookmark2023) [timestamping audit trail, 9-69](#_bookmark2060)

attacks

[*See*](#_bookmark2263) security attacks

[AUDIT EXECUTE PROCEDURE statement, 9-33](#_bookmark1861)

audit files

[activities always written to, 9-4](#_bookmark1620) [directory, 9-54](#_bookmark1982)

[file names, form of, 9-54](#_bookmark1983) operating system audit trail

[archiving, setting timestamp, 9-69](#_bookmark2061)

[audited actions in common with database audit trail, 9-11](#_bookmark1687)

operating system file [advantages of using, 9-16](#_bookmark1708) [appearance of text file, 9-13](#_bookmark1701)

[appearance of XML file, 9-15](#_bookmark1704) [archiving, 9-65](#_bookmark2041)

[contents, 9-13](#_bookmark1698)

[directory location, 9-17](#_bookmark1714) [how it works, 9-17](#_bookmark1712)

[if becomes too full, 9-62](#_bookmark2032) standard audit trail

[archiving, setting timestamp, 9-69](#_bookmark2060) [audited actions in common with operating](#_bookmark1688)

[system audit trail, 9-11](#_bookmark1688) [records, archiving, 9-61](#_bookmark2023)

[where written to, 9-54](#_bookmark1979) AUDIT statement

[about, 9-7](#_bookmark1644)

[schema objects, 9-30](#_bookmark1825) audit trail

[about, 9-57](#_bookmark1997)

[archiving, 9-61](#_bookmark2024)

[deleting views, 9-84](#_bookmark2116)

[finding information about, 9-80](#_bookmark2098) [interpreting, 9-81](#_bookmark2103)

[types of, 9-57](#_bookmark1997)

[*See also*](#_bookmark2001) standard audit trail, SYS.AUD$ table, SYS.FGA\_LOG$ table

AUDIT\_FILE\_DEST initialization parameter [about, 9-17](#_bookmark1716)

[setting for OS auditing, 9-17](#_bookmark1715) AUDIT\_SYS\_OPERATIONS initialization parameter

[auditing SYS, 9-53](#_bookmark1977)

AUDIT\_SYSLOG\_LEVEL initialization parameter [how it affects mandatory audit records, 9-4](#_bookmark1624)

AUDIT\_TRAIL initialization parameter [about, 9-8](#_bookmark1657)

[auditing SYS, 9-54](#_bookmark1985)

[database, starting in read-only mode, 9-10](#_bookmark1672) [DB (database) setting, 9-10](#_bookmark1671)

[DB, EXTENDED setting, 9-10](#_bookmark1674) [disabling, 9-11](#_bookmark1682)

[OS (operating system) setting, 9-10](#_bookmark1677) [setting, 9-8](#_bookmark1661)

[values, 9-9](#_bookmark1667)

[XML setting, 9-11](#_bookmark1680)

[XML, EXTENDED setting, 9-11](#_bookmark1683)

auditing

administrators

[*See*](#_bookmark1974) standard auditing [audit options, 9-5](#_bookmark1632)

[audit records, 9-57](#_bookmark1997)

[audit trail, sensitive data in, 10-18](#_bookmark2289) [audit trails, 9-57](#_bookmark1997)

[before-and-after changes, recording with triggers, 9-55](#_bookmark1987)

[committed data, 9-21,](#_bookmark1741) [10-20](#_bookmark2296) [database user names, 3-27](#_bookmark353) [default auditing, enabling, 9-35](#_bookmark1883) [distributed databases and, 9-4](#_bookmark1628) [finding information about, 9-80](#_bookmark2098) fine-grained

[*See*](#_bookmark1897) fine-grained auditing [functions, 9-32](#_bookmark1851)

[functions, Oracle Virtual Private Database, 9-33](#_bookmark1857) [general steps for, 9-5](#_bookmark1633)

[guidelines for security, 10-18](#_bookmark2286) [historical information, 10-19](#_bookmark2295)

[keeping information manageable, 10-18](#_bookmark2293) LOBs, auditing

[user-defined columns, 9-38](#_bookmark1906) [logon and logoff events, 9-25](#_bookmark1777) multitier environments

[*See*](#_bookmark1800) standard auditing network

[*See*](#_bookmark1869) standard auditing [object columns, 9-38](#_bookmark1906) objects

[*See*](#_bookmark1810) standard auditing

[One Big Application User authentication, compromised by, 5-2](#_bookmark1002)

operating system files [appearance, 9-13](#_bookmark1702)

[configuring, 9-10](#_bookmark1678)

[managing, 9-62](#_bookmark2030)

[operating-system user names, 3-27](#_bookmark353)

[Oracle Virtual Private Database policy functions, 9-33](#_bookmark1857)

[packages, 9-32](#_bookmark1852)

[performance, 9-4](#_bookmark1630)

[PL/SQL packages, 9-32](#_bookmark1853) privileges

[*See*](#_bookmark1787) standard auditing [procedures, 9-32](#_bookmark1854)

[range of focus, 9-5](#_bookmark1632) [recommended settings, 10-21](#_bookmark2301) Sarbanes-Oxley Act

[auditing, meeting compliance through, 9-2](#_bookmark1607) schema objects

[*See*](#_bookmark1811) standard auditing

[schema objects created in the future, 9-30](#_bookmark1828) SQL statements

[*See*](#_bookmark1761) standard auditing standard

[*See*](#_bookmark1642) standard audit trail, standard auditing statements

[*See*](#_bookmark1762) standard auditing [STMT\_AUDIT\_OPTION\_MAP table, 9-12](#_bookmark1693)

[suspicious activity, 10-20](#_bookmark2299)

[SYS user, 9-53](#_bookmark1975)

[SYS.FGA\_LOG$ table, 9-38](#_bookmark1911)

[SYSTEM user, 9-53](#_bookmark1971)

[SYSTEM\_PRIVILEGE\_MAP table, 9-12](#_bookmark1694)

[triggers, 9-32](#_bookmark1855)

[triggers used for, 9-55](#_bookmark1990) [UNIX syslog, 9-4](#_bookmark1625) views

[active object options, 9-83](#_bookmark2110) [active privilege options, 9-82](#_bookmark2108) [active statement options, 9-82](#_bookmark2105) [default object options, 9-83](#_bookmark2112)

[when audit options take effect, 9-8](#_bookmark1653) XML files

[appearance, 9-15](#_bookmark1705)

[configuring, 9-11](#_bookmark1684)

[*See also*](#_bookmark1634) SYS.AUD$ table, SYS.FGA\_LOG$ table, standard auditing, standard audit trail, fine-grained auditing

auditing, purging records [about, 9-66](#_bookmark2046)

[cancelling archive timestamp, 9-78](#_bookmark2089)

[clearing database audit trail batch size, 9-78](#_bookmark2091) creating audit trail

[purge job, 9-67](#_bookmark2053)

[creating the purge job, 9-70](#_bookmark2064) database audit trail

[purging subset of records, 9-74](#_bookmark2073) [deleting a purge job, 9-78](#_bookmark2087) [disabling purge jobs, 9-77](#_bookmark2083) [enabling purge jobs, 9-77](#_bookmark2083) [example, 9-79](#_bookmark2093)

[general steps for, 9-66](#_bookmark2048) initializing

[cancelling, 9-76](#_bookmark2081)

[initializing cleanup operation, 9-68](#_bookmark2057) [initializing, checking if done, 9-75](#_bookmark2077)

[purging audit trail manually, 9-72](#_bookmark2071) [purging records in batched groups, 9-71](#_bookmark2067) [roadmap, 9-66](#_bookmark2048)

[scheduling the purge job, 9-70](#_bookmark2064) [setting archive timestamp, 9-69](#_bookmark2061) [time interval for all purge jobs, 9-76](#_bookmark2079)

[time interval for named purge job, 9-77](#_bookmark2085) [AUTHENTICATEDUSER role, 4-11](#_bookmark590)

authentication [about, 3-1](#_bookmark183) administrators

[operating system, 3-25](#_bookmark330)

[passwords, 3-25](#_bookmark333)

[SYSDBA and SYSOPER access, centrally controlling, 3-22](#_bookmark321)

[by database, 3-26](#_bookmark340)

[by SSL, 3-31](#_bookmark386)

[client, 10-13](#_bookmark2233)

[client-to-middle tier process, 3-40](#_bookmark450) [database administrators, 3-22](#_bookmark317) databases, using

[about, 3-26](#_bookmark342)

[advantages, 3-26](#_bookmark346)

[procedure, 3-27](#_bookmark348)

[directory service, 3-31](#_bookmark382)

[directory-based services, 3-29](#_bookmark366) external authentication

[about, 3-32](#_bookmark395)

[advantages, 3-33](#_bookmark398)

[operating system authentication, 3-34](#_bookmark402) [user creation, 3-33](#_bookmark400)

global authentication [about, 3-30](#_bookmark376)

[advantages, 3-31](#_bookmark391)

[user creation for private schemas, 3-31](#_bookmark384) [user creation for shared schemas, 3-31](#_bookmark388)

middle-tier authentication [proxies, example, 3-42](#_bookmark456)

[multitier, 3-34](#_bookmark408) network authentication

[Secure Sockets Layer, 3-28](#_bookmark358) [third-party services, 3-28](#_bookmark362)

[One Big Application User, compromised by, 5-2](#_bookmark1003) operating system authentication

[about, 3-27](#_bookmark350)

[advantages, 3-27](#_bookmark351)

[disadvantages, 3-27](#_bookmark354) proxy user authentication

[about, 3-36](#_bookmark418)

[expired passwords, 3-39](#_bookmark437) [public key infrastructure, 3-29](#_bookmark369) [RADIUS, 3-29](#_bookmark365)

[remote, 10-13](#_bookmark2234)

[specifying when creating a user, 2-3](#_bookmark69) [strong, 10-9](#_bookmark2188)

[SYSDBA on Windows systems, 3-25](#_bookmark331) [Windows native authentication, 3-25](#_bookmark331) [*See also*](#_bookmark187) passwords, proxy authentication

AUTHID DEFINER clause

[used with Oracle Virtual Private Database](#_bookmark1358)

[functions, 7-3](#_bookmark1358) authorization

[about, 4-1](#_bookmark495)

[changing for roles, 4-17](#_bookmark651) global

[about, 3-30](#_bookmark376)

[advantages, 3-31](#_bookmark391)

[multitier, 3-34](#_bookmark408) [omitting for roles, 4-17](#_bookmark650) [operating system, 4-19](#_bookmark671)

[roles, about, 4-17](#_bookmark657) automatic reparse

[Oracle Virtual Private Database, how it works with, 7-35](#_bookmark1513)

##### B

banners

[auditing user actions, configuring, 5-20](#_bookmark1116) [unauthorized access, configuring, 5-20](#_bookmark1117)

[batch jobs, authenticating users in, 3-17](#_bookmark301) BFILEs

[guidelines for security, 10-11](#_bookmark2210) bind variables

[application contexts, used with, 7-4](#_bookmark1368) [information captured in audit trail, 9-10](#_bookmark1676)

BLOBS

[encrypting, 8-7](#_bookmark1581) BY ACCESS clause

[about, 9-22](#_bookmark1749)

[benefits of using, 9-22](#_bookmark1749)

[finding statement audit options, 9-82](#_bookmark2106) [NOAUDIT statement non-support of, 9-23](#_bookmark1756) [using, 9-22](#_bookmark1749)

##### C

[CAPI\_USER\_ROLE role, 4-11](#_bookmark591)

[cascading revokes, 4-43](#_bookmark838) CATNOAUD.SQL script

[about, 9-84](#_bookmark2116)

[audit trail views, deleting with, 9-84](#_bookmark2117) certificate key algorithm

[Secure Sockets Layer, 10-17](#_bookmark2280) [change\_on\_install default password, 10-9](#_bookmark2181) character sets

[role names, multibyte characters in, 4-16](#_bookmark646) [role passwords, multibyte characters in, 4-18](#_bookmark662)

cipher suites

[Secure Sockets Layer, 10-17](#_bookmark2280) client connections

[guidelines for security, 10-13](#_bookmark2229)

[secure external password store, 3-18](#_bookmark308) [securing, 10-13](#_bookmark2230)

client identifier

[setting for applications that use JDBC, 3-46](#_bookmark480) client identifiers

[about, 3-44](#_bookmark469)

[auditing users, 9-28](#_bookmark1805)

[consistency between DBMS\_SESSION.SET](#_bookmark484)\_

[IDENTIFIER and DBMS\_APPLICATION\_ INFO.SET\_CLIENT\_INFO, 3-46](#_bookmark484)

[global application context, independent of, 3-45](#_bookmark476) [setting with DBMS\_SESSION.SET\_IDENTIFIER](#_bookmark1253)

[procedure, 6-23](#_bookmark1253)

[*See also*](#_bookmark1248) nondatabase users

client session-based application contexts [about, 6-42](#_bookmark1319)

[CLIENTCONTEXT namespace, clearing value from, 6-44](#_bookmark1329)

[CLIENTCONTEXT namespace, setting value in, 6-43](#_bookmark1324)

[retrieving CLIENTCONTEXT namespace, 6-43](#_bookmark1326)

[*See also*](#_bookmark1320) application contexts CLIENT\_IDENTIFIER USERENV attribute

[setting and clearing with DBMS\_SESSION package, 3-46](#_bookmark483)

[setting with OCI user session handle attribute, 3-46](#_bookmark479)

[*See also*](#_bookmark477) USERENV namespace [CLIENTID\_OVERWRITE event, 3-46](#_bookmark486)

[column masking behavior, 7-10](#_bookmark1415) [column specification, 7-10](#_bookmark1416)

[restrictions, 7-11](#_bookmark1418) columns

[granting privileges for selected, 4-40](#_bookmark820) [granting privileges on, 4-40](#_bookmark820)

[INSERT privilege and, 4-40](#_bookmark820) [listing users granted to, 4-73](#_bookmark976) [privileges, 4-40](#_bookmark820)

pseudo columns [USER, 4-29](#_bookmark744)

[revoking privileges on, 4-43](#_bookmark834) [command line recall attacks, 5-4,](#_bookmark1017) [5-5](#_bookmark1024) committed data

[auditing, 9-21,](#_bookmark1741) [10-20](#_bookmark2296) configuration

[guidelines for security, 10-12](#_bookmark2221) configuration files

[listener.ora, 10-14](#_bookmark2246)

[sample listener.ora file, 10-14](#_bookmark2247) [server.key encryption file, 10-17](#_bookmark2282) [tsnames.ora, 10-17](#_bookmark2276)

[typical directory, 10-17](#_bookmark2276) CONNECT role

[about, 10-22](#_bookmark2305) applications

[account provisioning, 10-23](#_bookmark2313)

[affects of, 10-22](#_bookmark2309)

[database upgrades, 10-22](#_bookmark2311)

[installation of, 10-23](#_bookmark2315) [script to create, 4-11](#_bookmark592) users

[application developers, impact, 10-23](#_bookmark2321) [client-server applications, impact, 10-23](#_bookmark2323) [general users, impact, 10-23](#_bookmark2319)

[how affects, 10-23](#_bookmark2317)

[why changed, 10-22](#_bookmark2307)

[CONNECT role, privilege available to, 4-6](#_bookmark546) connection pooling

[about, 3-34](#_bookmark409)

[global application contexts, 6-22](#_bookmark1247) [nondatabase users, 6-28](#_bookmark1283)

[proxy authentication, 3-40](#_bookmark448) connections

[SYS privilege, 10-2](#_bookmark2137) [CPU time limit, 2-11](#_bookmark124)

CREATE ANY LIBRARY statement [security guidelines, 10-3](#_bookmark2140)

[CREATE ANY PROCEDURE system privilege, 4-31](#_bookmark764) CREATE ANY TABLE statement

[non-administrative users, 10-2](#_bookmark2138) CREATE CONTEXT statement

[about, 6-5](#_bookmark1156)

[example, 6-5](#_bookmark1157)

CREATE LIBRARY statement [security guidelines, 10-3](#_bookmark2141)

[CREATE PROCEDURE system privilege, 4-31](#_bookmark764) CREATE PROFILE statement

[account locking period, 3-6](#_bookmark231) [failed login attempts, 3-6](#_bookmark231)

[password aging and expiration, 3-8](#_bookmark251) [password management, 3-4](#_bookmark204)

[passwords, example, 3-10](#_bookmark259) CREATE ROLE statement

[IDENTIFIED EXTERNALLY option, 4-18](#_bookmark667)

CREATE SCHEMA statement [securing, 5-15](#_bookmark1062)

[CREATE SESSION statement, 4-6](#_bookmark547) [CONNECT role privilege, 10-7](#_bookmark2165) [securing, 5-15](#_bookmark1063)

CREATE USER statement [explicit account locking, 3-7](#_bookmark237) [IDENTIFIED BY option, 2-3](#_bookmark69)

[IDENTIFIED EXTERNALLY option, 2-3](#_bookmark69)

[passwords, expiring, 3-10](#_bookmark263)

[user profile, 3-4](#_bookmark205)

[CSW\_USR\_ROLE role, 4-11](#_bookmark593)

[CTXAPP role, 4-12](#_bookmark594)

cursors

[reparsing, for application contexts, 6-12](#_bookmark1204) [shared, used with Virtual Private Database, 7-4](#_bookmark1369)

[custom installation, 10-12](#_bookmark2222)

[CWM\_USER role, 4-12](#_bookmark595)

##### D

data definition language (DDL) [roles and privileges, 4-9](#_bookmark576) [standard auditing, 9-23](#_bookmark1765)

data dictionary [protecting, 10-10](#_bookmark2197)

[securing with O7\_DICTIONARY\_ ACCESSIBILITY, 4-3](#_bookmark515)

data dictionary views

[*See*](#_bookmark962) views [data files, 10-11](#_bookmark2210)

[guidelines for security, 10-10](#_bookmark2196) data manipulation language (DML) [privileges controlling, 4-26](#_bookmark731)

[standard auditing, 9-23](#_bookmark1766) data security

[encryption, problems not solved by, 8-3](#_bookmark1559) database administrators (DBAs)

[access, controlling, 8-2](#_bookmark1555)

[authentication, 3-22](#_bookmark317)

[malicious, encryption not solved by, 8-2](#_bookmark1555) database audit trail

[audited actions in common with operating system audit trail, 9-11](#_bookmark1689)

[batch size for records during purging, 9-71](#_bookmark2068) [protecting, 9-3](#_bookmark1610)

[tablespace, moving to one other than SYSTEM, 9-60](#_bookmark2017)

Database Configuration Assistant (DBCA) [default passwords, changing, 10-9](#_bookmark2182)

[user accounts, automatically locking and expiring, 10-4](#_bookmark2155)

database links

[application context support, 6-16](#_bookmark1220) [application contexts, 6-9](#_bookmark1184)

[auditing, 9-29](#_bookmark1818)

[authenticating with Kerberos, 3-28](#_bookmark364) [authenticating with third-party services, 3-28](#_bookmark363) [global user authentication, 3-32](#_bookmark392)

[object privileges, 4-24](#_bookmark722)

[operating system accounts, care needed, 3-28](#_bookmark355) [session-based application contexts, accessing, 6-9](#_bookmark1185)

database session-based application contexts [about, 6-4](#_bookmark1150)

[cleaning up after user exits, 6-4](#_bookmark1152) [components, 6-5](#_bookmark1154)

[creating, 6-5](#_bookmark1156)

[database links, 6-9](#_bookmark1184)

[dynamic SQL, 6-8](#_bookmark1177)

[externalized, using, 6-21](#_bookmark1236) [how to use, 6-4](#_bookmark1148)

[initializing externally, 6-16](#_bookmark1219)

[initializing globally, 6-18](#_bookmark1229)

[ownership, 6-5](#_bookmark1158)

[parallel queries, 6-9](#_bookmark1181)

[PL/SQL package creation, 6-6](#_bookmark1163) [session information, setting, 6-9](#_bookmark1189) [SYS\_CONTEXT function, 6-7](#_bookmark1170)

[trusted procedure, 6-2](#_bookmark1129)

[tutorial, 6-13](#_bookmark1208)

[*See also*](#_bookmark1151) application contexts

[database upgrades and CONNECT role, 10-22](#_bookmark2311) databases

access control

[password encryption, 3-2](#_bookmark190) [additional security resources, 1-2](#_bookmark54) [authentication, 3-26](#_bookmark340)

[database user and application user, 5-2](#_bookmark998) default audit settings

[about, 9-35](#_bookmark1884)

[DBCA-created databases, 9-35](#_bookmark1885)

[manually-created databases, 9-35](#_bookmark1886) [default password security settings, 3-6](#_bookmark222)

[DBCA-created databases, 3-6](#_bookmark226)

[manually-created databases, 3-6](#_bookmark224) [default security features, summary, 1-1](#_bookmark52) [granting privileges, 4-37](#_bookmark792)

[granting roles, 4-37](#_bookmark792) [limitations on usage, 2-10](#_bookmark115)

[read-only mode, starting in, 9-10](#_bookmark1673) [security and schemas, 5-15](#_bookmark1059)

[security embedded, advantages of, 5-2](#_bookmark1009) [security policies based on, 7-2](#_bookmark1351)

[DATAPUMP\_EXP\_FULL\_DATABASE role, 4-12](#_bookmark596)

[DATAPUMP\_IMP\_FULL\_DATABASE role, 4-12](#_bookmark597)

DBA role

[about, 4-12](#_bookmark598)

[DBA\_NETWORK\_ACL\_PRIVILEGES view, 4-66](#_bookmark935)

DBA\_ROLE\_PRIVS view

[application privileges, finding, 5-12](#_bookmark1035) DBA\_ROLES data dictionary view

[PUBLIC role, 4-5](#_bookmark535)

[DBFS\_ROLE role, 4-12](#_bookmark599) DBMS\_APPLICATION.SET\_CLIENT\_INFO

procedure [DBMS\_SESSION.SET\_IDENTIFIER value,](#_bookmark487)

[overwriting, 3-46](#_bookmark487) DBMS\_CRYPTO package

[about, 8-8](#_bookmark1584)

[encryption algorithms supported, 8-8](#_bookmark1587) [examples, 8-10](#_bookmark1593)

DBMS\_FGA package [about, 9-39](#_bookmark1920)

[ADD\_POLICY procedure, 9-40](#_bookmark1923)

[DISABLE\_POLICY procedure, 9-43](#_bookmark1935)

[DROP\_POLICY procedure, 9-44](#_bookmark1940)

[ENABLE\_POLICY procedure, 9-44](#_bookmark1937) DBMS\_OBFUSCATION\_TOOLKIT package

[backward compatibility, 8-8](#_bookmark1585)

[*See also*](#_bookmark1586) DBMS\_CRYPTO package DBMS\_RLS package

[about, 7-6](#_bookmark1382)

[DBMS\_RLS.ADD\_CONTEXT procedure, 7-6](#_bookmark1396) [DBMS\_RLS.ADD\_GROUPED\_POLICY](#_bookmark1391)

[procedure, 7-6](#_bookmark1391) DBMS\_RLS.ADD\_POLICY

[sec\_relevant\_cols parameter, 7-8](#_bookmark1409)

[sec\_relevant\_cols\_opt parameter, 7-10](#_bookmark1416) DBMS\_RLS.ADD\_POLICY procedure

[about, 7-6](#_bookmark1385) [DBMS\_RLS.CREATE\_POLICY\_GROUP](#_bookmark1389)

[procedure, 7-6](#_bookmark1389) [DBMS\_RLS.DELETE\_POLICY\_GROUPS](#_bookmark1390)

[procedure, 7-6](#_bookmark1390) [DBMS\_RLS.DISABLE\_GROUPED\_POLICY](#_bookmark1394)

[procedure, 7-6](#_bookmark1394)

[DBMS\_RLS.DROP\_CONTEXT procedure, 7-6](#_bookmark1397) [DBMS\_RLS.DROP\_GROUPED\_POLICY](#_bookmark1395)

[procedure, 7-6](#_bookmark1395)

[DBMS\_RLS.DROP\_POLICY procedure, 7-6](#_bookmark1388) [DBMS\_RLS.ENABLE\_GROUPED\_POLICY](#_bookmark1392)

[procedure, 7-6](#_bookmark1392)

[DBMS\_RLS.ENABLE\_POLICY procedure, 7-6](#_bookmark1386) [DBMS\_RLS.REFRESH\_GROUPED\_POLICY](#_bookmark1393)

[procedure, 7-6](#_bookmark1393)

[DBMS\_RLS.REFRESH\_POLICY procedure, 7-6](#_bookmark1387) DBMS\_SESSION package

[client identifiers, using, 3-46](#_bookmark483)

[global application context, used in, 6-23](#_bookmark1261) SET\_CONTEXT procedure

[about, 6-9](#_bookmark1188)

[application context name-value pair, setting, 6-7](#_bookmark1167)

DBMS\_SESSION.SET\_CONTEXT procedure [about, 6-9](#_bookmark1187)

[syntax, 6-9](#_bookmark1190)

[username and client\_id settings, 6-25](#_bookmark1270) DBMS\_SESSION.SET\_IDENTIFIER procedure

[client session ID, setting, 6-23](#_bookmark1253) [DBMS\_APPLICATION.SET\_CLIENT\_INFO value,](#_bookmark485)

[overwritten by, 3-46](#_bookmark485) DBSNMP user account

[password usage, 10-9](#_bookmark2183) DDL

[*See*](#_bookmark577) data definition language [default passwords, 10-8,](#_bookmark2173) [10-9](#_bookmark2175)

[change\_on\_install or manager passwords, 10-9](#_bookmark2181) [changing, importance of, 3-4](#_bookmark208)

[finding, 3-4](#_bookmark208)

[default permissions, 10-10](#_bookmark2207) default profiles

[about, 3-4](#_bookmark211) default roles

[setting for user, 2-7](#_bookmark96) [specifying, 4-48](#_bookmark871)

default users [accounts, 10-4](#_bookmark2153)

[Enterprise Manager accounts, 10-4](#_bookmark2158) [passwords, 10-9](#_bookmark2177)

defaults

[tablespace quota, 2-5](#_bookmark77)

[user tablespaces, 2-4](#_bookmark72) definer’s rights

[about, 4-29](#_bookmark754)

[procedure privileges, used with, 4-29](#_bookmark755) [procedure security, 4-29](#_bookmark757)

[secure application roles, 5-13](#_bookmark1045)

[used with Oracle Virtual Private Database functions, 7-3](#_bookmark1359)

DELETE privilege

[SQL statements permitted, 5-17](#_bookmark1087) DELETE\_CATALOG\_ROLE role

[about, 4-12](#_bookmark600)

[SYS schema objects, enabling access to, 4-4](#_bookmark527) denial-of-service (DoS) attacks

[bad packets, preventing, 5-18](#_bookmark1101) [networks, securing, 10-16](#_bookmark2264)

denial-of-service attacks [about, Glossary-2](#_bookmark2338)

denial-of-service(DoS) attacks

[audit trail, writing to operating system file, 9-16](#_bookmark1710) [dictionary protection mechanism, 4-3](#_bookmark517)

direct path load

[fine-grained auditing effects on, 9-37](#_bookmark1901)

[directory authentication, configuring for SYSDBA or SYSOPER access, 3-22](#_bookmark324)

directory object auditing [configuring, 9-32](#_bookmark1844)

[removing, 9-32](#_bookmark1847) directory objects

[auditing, 9-32](#_bookmark1842)

[granting EXECUTE privilege on, 4-37](#_bookmark798) [directory-based services authentication, 3-29](#_bookmark366) disabling unnecessary services

[FTP, TFTP, TELNET, 10-16](#_bookmark2268)

dispatcher processes (D*nnn*)

[limiting SGA space for each session, 2-12](#_bookmark134) distributed databases

[auditing and, 9-4](#_bookmark1628) DML

[*See*](#_bookmark732) data manipulation language [driving context, 6-45](#_bookmark1337)

DROP PROFILE statement [example, 2-14](#_bookmark149)

DROP ROLE statement [example, 4-21](#_bookmark696)

[security domain, affected, 4-21](#_bookmark694) DROP USER statement

[about, 2-14](#_bookmark159)

[schema objects of dropped user, 2-15](#_bookmark161) DUAL table

[about, 6-8](#_bookmark1173)

[dynamic Oracle Virtual Private Database policy types, 7-15](#_bookmark1440)

[DYNAMIC policy type, 7-15](#_bookmark1439)

##### E

editions

[application contexts, how affects, 6-3](#_bookmark1142)

[fine-grained auditing packages, results in, 6-24](#_bookmark1267) [global application contexts, how affects, 6-24](#_bookmark1266) [Oracle Virtual Private Database packages, results](#_bookmark1268)

[in, 6-24](#_bookmark1268)

[EJBCLIENT role, 4-12](#_bookmark601)

[email alert example, 9-44](#_bookmark1946) encryption

[access control, 8-1](#_bookmark1553)

[BLOBS, 8-7](#_bookmark1581)

[challenges, 8-4](#_bookmark1561)

[data security, problems not solved by, 8-3](#_bookmark1559) [data transfer, 10-15](#_bookmark2251)

[DBMS\_CRYPTO package, 8-8](#_bookmark1584) [deleted encrypted data, 10-11](#_bookmark2214) [examples, 8-10](#_bookmark1593)

[finding information about, 8-15](#_bookmark1599)

[fine-grained audit policies on encrypted columns, 9-40](#_bookmark1925)

[indexed data, 8-4](#_bookmark1563)

[key generation, 8-4](#_bookmark1565)

[key storage, 8-5](#_bookmark1569)

[key transmission, 8-5](#_bookmark1567)

[keys, changing, 8-7](#_bookmark1579)

[malicious database administrators, 8-2](#_bookmark1555)

[network traffic, 10-16](#_bookmark2266) [problems not solved by, 8-1](#_bookmark1551)

[transparent data encryption, 8-7](#_bookmark1576) [transparent tablespace encryption, 8-7](#_bookmark1577)

[enterprise directory service, 4-19](#_bookmark680) [Enterprise Edition, 10-9](#_bookmark2189) Enterprise Manager

[granting roles, 4-20](#_bookmark689)

[statistics monitor, 2-12](#_bookmark138)

[enterprise roles, 3-30,](#_bookmark379) [4-19](#_bookmark683) [enterprise user management, 5-2](#_bookmark1005) Enterprise User Security

[application context, globally initialized, 6-19](#_bookmark1233) proxy authentication

[Oracle Virtual Private Database, how it works with, 7-38](#_bookmark1533)

enterprise users

[centralized management, 3-30](#_bookmark380) [global role, creating, 4-19](#_bookmark684)

[One Big Application User authentication, compromised by, 5-2](#_bookmark1006)

[proxy authentication, 3-36](#_bookmark419)

[shared schemas, protecting users, 5-16](#_bookmark1068) errors

[OPW-00005, 2-9](#_bookmark111)

[ORA-00036, 9-41](#_bookmark1928)

[ORA-01720, 4-28](#_bookmark741)

[ORA-06512, 9-48](#_bookmark1957)

[ORA-1000, 9-41](#_bookmark1929)

[ORA-24247, 4-51,](#_bookmark890) [9-48](#_bookmark1956)

[ORA-28009, 4-3](#_bookmark520)

[ORA-28040, 3-26](#_bookmark344) ORA-28046

[ORA-28046 error, 2-9](#_bookmark110)

[ORA-28133, 9-40](#_bookmark1927)

[ORA-28144, 9-42](#_bookmark1930)

examples

access control lists

[external network connections, 4-59](#_bookmark915) [wallet access, 4-59](#_bookmark916)

[account locking, 3-7](#_bookmark232) [audit trail, purging, 9-79](#_bookmark2093)

[audit trigger to record before-and-after values, 9-55](#_bookmark1988)

data encryption

[encrypting and decrypting BLOB data, 8-12](#_bookmark1597) [encrypting and decrypting procedure with AES](#_bookmark1595)

[256-Bit, 8-11](#_bookmark1595)

[directory objects, granting EXECUTE privilege on, 4-37](#_bookmark798)

[encrypting procedure, 8-10](#_bookmark1593)

[Java code to read passwords, 5-7](#_bookmark1029)

[locking an account with CREATE PROFILE, 3-7](#_bookmark233) [login attempt grace period, 3-10](#_bookmark260)

[nondatabase user authentication, 6-28](#_bookmark1284) [O7\_DICTIONARY\_ACCESSIBILITY initialization](#_bookmark518)

[parameter, setting, 4-3](#_bookmark518) passwords

[aging and expiration, 3-8](#_bookmark253) [changing, 2-8](#_bookmark106)

[creating for user, 2-3](#_bookmark70) privileges

[granting ADMIN OPTION, 4-38](#_bookmark803) [views, 4-72](#_bookmark967)

[procedure privileges affecting packages, 4-32](#_bookmark772) [profiles, assigning to user, 2-7](#_bookmark94)

roles

[altering for external authorization, 4-17](#_bookmark653) [creating for application authorization, 4-18](#_bookmark664) [creating for external authorization, 4-19](#_bookmark669) [creating for password authorization, 4-17](#_bookmark648) [default, setting, 4-48](#_bookmark872)

[using SET ROLE for password-authenticated roles, 4-18](#_bookmark660)

[views, 4-72](#_bookmark967)

[secure external password store, 3-17](#_bookmark306) session ID of user

[finding, 2-14](#_bookmark155)

[terminating, 2-14](#_bookmark157)

[system privilege and role, granting, 4-37](#_bookmark796) tablespaces

[assigning default to user, 2-4](#_bookmark74) [quota, assigning to user, 2-5](#_bookmark78) [temporary, 2-6](#_bookmark90)

[type creation, 4-34](#_bookmark786) users

[account creation, 2-2](#_bookmark62)

[creating with GRANT statement, 4-38](#_bookmark806) [dropping, 2-15](#_bookmark162)

[middle-tier server proxying a client, 3-39](#_bookmark432) [naming, 2-3](#_bookmark67)

[object privileges granted to, 4-39](#_bookmark811) [proxy user, connecting as, 3-39](#_bookmark432)

[*See also*](#_bookmark1466) tutorials exceptions

[WHEN NO DATA FOUND, used in application context package, 6-15](#_bookmark1214)

WHEN OTHERS, used in triggers [development environment (debugging)](#_bookmark1202)

[example, 6-12](#_bookmark1202)

[production environment example, 6-11](#_bookmark1198) exclusive mode

[SHA-1 password hashing algorithm, enabling, 3-16](#_bookmark295)

EXECUTE ANY LIBRARY statement [security guidelines, 10-3](#_bookmark2142)

EXECUTE privilege

[SQL statements permitted, 5-17](#_bookmark1089) EXECUTE\_CATALOG\_ROLE role

[about, 4-12](#_bookmark602)

[SYS schema objects, enabling access to, 4-4](#_bookmark526) [execution time for statements, measuring, 7-15](#_bookmark1441) EXEMPT ACCESS POLICY privilege

[Oracle Virtual Private Database enforcements, exemption, 7-37](#_bookmark1526)

EXP\_FULL\_DATABASE role

[about, 4-12](#_bookmark603) expiring a password

[explicitly, 3-10](#_bookmark264) exporting data

[direct path export impact on Oracle Virtual Private Database, 7-36](#_bookmark1522)

[policy enforcement, 7-36](#_bookmark1524) external authentication

[about, 3-32](#_bookmark395)

[advantages, 3-33](#_bookmark398)

[network, 3-34](#_bookmark406)

[operating system, 3-34](#_bookmark403)

[user creation, 3-33](#_bookmark400)

external network services, fine-grained access to

[*See*](#_bookmark883) access control list (ACL) [external tables, 10-11](#_bookmark2210)

##### F

failed login attempts [account locking, 3-6](#_bookmark229)

[password management, 3-6](#_bookmark229)

[resetting, 3-7](#_bookmark238) features, new security

[*See*](#_bookmark6) new features, security files

BFILEs

[operating system access, restricting, 10-11](#_bookmark2210) [BLOB, 8-7](#_bookmark1582)

data

[operating system access, restricting, 10-11](#_bookmark2211) external tables

[operating system access, restricting, 10-11](#_bookmark2212) [keys, 8-7](#_bookmark1572)

listener.ora file

[guidelines for security, 10-14,](#_bookmark2246) [10-17](#_bookmark2277) log

[audit file location for Windows, 9-54](#_bookmark1980) [audit file locations, 9-17](#_bookmark1718)

[operating system access, restricting, 10-11](#_bookmark2211) [restrict listener access, 10-15](#_bookmark2248)

[server.key encryption file, 10-17](#_bookmark2282) [symbolic links, restricting, 10-10](#_bookmark2209) [tnsnames.ora, 10-17](#_bookmark2276)

trace

[operating system access, restricting, 10-11](#_bookmark2212) fine-grained access control

[*See*](#_bookmark1354) Oracle Virtual Private Database (VPD) fine-grained auditing

[about, 9-37](#_bookmark1898)

[activities always recorded, 9-38](#_bookmark1910) [advantages, 9-38](#_bookmark1905)

[alerts, adding to policy, 9-44](#_bookmark1948) [archiving audit trail, 9-61](#_bookmark2025) [columns, specific, 9-43](#_bookmark1933) [creating audit trail for, 9-39](#_bookmark1915) [DBMS\_FGA package, 9-39](#_bookmark1920) [direct loads of data, 9-37](#_bookmark1900)

[edition-based redefinitions, 9-39](#_bookmark1913) [editions, results in, 6-24](#_bookmark1267) [encrypted table columns, 9-40](#_bookmark1926)

[finding errors by checking trace files, 9-80](#_bookmark2096) [how audit records are generated, 9-39](#_bookmark1917) [how to use, 9-37](#_bookmark1899)

[location of audit records, 9-37](#_bookmark1903) [non-SYS activities audited, 9-3](#_bookmark1615) policies

[adding, 9-40](#_bookmark1923)

[disabling, 9-43](#_bookmark1935)

[dropping, 9-44](#_bookmark1940)

[enabling, 9-44](#_bookmark1937)

[modifying, 9-40](#_bookmark1923)

[where created, 9-40](#_bookmark1924)

[privileges needed, 9-38](#_bookmark1908) records

[archiving, 9-61](#_bookmark2027)

[*See also*](#_bookmark1896) SYS.FGA\_LOG$ table firewalls

[advice about using, 10-15](#_bookmark2252) [database server location, 10-15](#_bookmark2253) [ports, 10-17](#_bookmark2274)

[supported types, 10-15](#_bookmark2254) flashback query

[auditing, used with, 9-59](#_bookmark2009)

[Oracle Virtual Private Database, how it works with, 7-35](#_bookmark1515)

foreign keys

[privilege to use parent key, 4-27](#_bookmark734) [FTP service, 10-16](#_bookmark2268)

functions

[auditing, 9-32](#_bookmark1851)

Oracle Virtual Private Database [components of, 7-4](#_bookmark1373) [privileges used to run, 7-3](#_bookmark1360)

[privileges for, 4-29](#_bookmark747)

[roles, 4-9](#_bookmark571)

##### G

[GATHER\_SYSTEM\_STATISTICS role, 4-13](#_bookmark604)

global application contexts [about, 6-22](#_bookmark1240)

[authenticating nondatabase users, 6-28](#_bookmark1285) [components, 6-22](#_bookmark1249)

[editions, affect on, 6-24](#_bookmark1266)

[example of authenticating nondatabase users, 6-29](#_bookmark1287)

[example of authenticating user moving to different application, 6-27](#_bookmark1280)

[example of setting values for all users, 6-26](#_bookmark1276) [Oracle RAC environment, 6-23](#_bookmark1255)

[Oracle RAC instances, 6-22](#_bookmark1243) [ownership, 6-23](#_bookmark1258)

[PL/SQL package creation, 6-24](#_bookmark1263) [process, lightweight users, 6-40](#_bookmark1316) [process, standard, 6-39](#_bookmark1314)

[sharing values globally for all users, 6-26](#_bookmark1275) [system global area, 6-22](#_bookmark1244)

[tutorial for client session IDs, 6-35](#_bookmark1302)

[used for One Big Application User scenarios, 7-38](#_bookmark1537) [user name retrieval with USER function, 6-25](#_bookmark1273) [uses for, 7-38](#_bookmark1538)

[*See also*](#_bookmark1245) application contexts global authentication

[about, 3-30](#_bookmark376)

[advantages, 3-31](#_bookmark391)

[user creation for private schemas, 3-31](#_bookmark384) [user creation for shared schemas, 3-31](#_bookmark388)

global authorization [about, 3-30](#_bookmark376)

[advantages, 3-31](#_bookmark391)

[role creation, 4-19](#_bookmark681)

[roles, 3-30](#_bookmark377) global roles

[about, 4-19](#_bookmark679)

[global users, 3-30](#_bookmark378)

[GLOBAL\_AQ\_USER\_ROLE role, 4-13](#_bookmark605)

grace period for login attempts [example, 3-10](#_bookmark260)

[grace period for password expiration, 3-9](#_bookmark257) GRANT ALL PRIVILEGES statement

[SELECT ANY DICTIONARY privilege, exclusion of, 10-10](#_bookmark2204)

[GRANT ANY OBJECT PRIVILEGE system](#_bookmark817)

[privilege, 4-39,](#_bookmark817) [4-42](#_bookmark831)

[GRANT ANY PRIVILEGE system privilege, 4-5](#_bookmark532) GRANT CONNECT THROUGH clause

[consideration when setting FAILED\_LOGIN\_ ATTEMPTS parameter, 3-5](#_bookmark215)

[for proxy authorization, 3-39](#_bookmark434) [GRANT statement, 4-37](#_bookmark795)

[ADMIN OPTION, 4-38](#_bookmark801)

[creating a new user, 4-38](#_bookmark807) [object privileges, 4-38,](#_bookmark810) [5-16](#_bookmark1075)

[system privileges and roles, 4-37](#_bookmark792) [when takes effect, 4-48](#_bookmark864)

[WITH GRANT OPTION, 4-39](#_bookmark814)

granting privileges and roles [about, 4-4](#_bookmark529)

[finding information about, 4-71](#_bookmark963) [specifying ALL, 4-24](#_bookmark717)

guidelines for security [auditing, 10-18](#_bookmark2286)

[custom installation, 10-12](#_bookmark2222)

[data files and directories, 10-10](#_bookmark2196) [encrypting sensitive data, 10-11](#_bookmark2214) [installation and configuration, 10-12](#_bookmark2221) [networking security, 10-13](#_bookmark2227) [operating system accounts, limiting](#_bookmark2206)

[privileges, 10-10](#_bookmark2206)

[operating system users, limiting number of, 10-10](#_bookmark2205) [Oracle home default permissions, disallowing](#_bookmark2208)

[modification, 10-10](#_bookmark2208) [ORACLE\_DATAPUMP access driver, 10-11](#_bookmark2217) [passwords, 10-7](#_bookmark2168)

Secure Sockets Layer [mode, 10-17](#_bookmark2279)

[TCPS protocol, 10-17](#_bookmark2275) [symbolic links, restricting, 10-10](#_bookmark2209)

[user accounts and privileges, 10-2](#_bookmark2132)

##### H

hackers

[*See*](#_bookmark193) security attacks HS\_ADMIN\_EXECUTE\_ROLE role

[about, 4-13](#_bookmark606) HS\_ADMIN\_ROLE role

[about, 4-13](#_bookmark607) HS\_ADMIN\_SELECT\_ROLE role

[about, 4-13](#_bookmark608) HTTP authentication

[*See*](#_bookmark905) access control lists (ACL), wallet access HTTPS

[port, correct running on, 10-17](#_bookmark2273)

##### I

IMP\_FULL\_DATABASE role

[about, 4-13](#_bookmark609) INDEX privilege

[SQL statements permitted, 5-17](#_bookmark1090) indexed data

[encryption, 8-4](#_bookmark1563) [indirectly granted roles, 4-7](#_bookmark552) initialization parameters

[application protection, 5-17 to](#_bookmark1096) [5-20](#_bookmark1120)

[AUDIT\_FILE\_DEST, 9-4,](#_bookmark1622) [9-54](#_bookmark1984)

[AUDIT\_SYS\_OPERATIONS, 9-53](#_bookmark1978)

[AUDIT\_SYSLOG\_LEVEL, 9-19](#_bookmark1731) AUDIT\_TRAIL

[about, 9-8](#_bookmark1662)

[using, 9-9](#_bookmark1668)

[current value, checking, 9-8](#_bookmark1663) [MAX\_ENABLED\_ROLES, 4-49](#_bookmark875)

[O7\_DICTIONARY\_ACCESSIBILITY, 4-3](#_bookmark516)

[OS\_AUTHENT\_PREFIX, 3-32](#_bookmark396)

[OS\_ROLES, 4-19](#_bookmark672)

[REMOTE\_OS\_AUTHENT, 10-13](#_bookmark2235)

[RESOURCE\_LIMIT, 2-13](#_bookmark143)

[SEC\_CASE\_SENSITIVE\_LOGON, 3-13](#_bookmark278)

[SEC\_MAX\_FAILED\_LOGIN\_ATTEMPTS, 5-19](#_bookmark1111) [SEC\_PROTOCOL\_ERROR\_FURTHER\_](#_bookmark1107)

[ACTION, 5-18](#_bookmark1107) [SEC\_PROTOCOL\_ERROR\_TRACE\_](#_bookmark1104)

[ACTION, 5-18](#_bookmark1104) [SEC\_RETURN\_SERVER\_RELEASE\_](#_bookmark1114)

[BANNER, 5-19](#_bookmark1114)

[SEC\_USER\_AUDIT\_ACTION\_BANNER, 5-20](#_bookmark1119) [SEC\_USER\_UNAUTHORIZED\_ACCESS\_](#_bookmark1118)

[BANNER, 5-20](#_bookmark1118)

INSERT privilege [granting, 4-40](#_bookmark820)

[revoking, 4-43](#_bookmark834)

[SQL statements permitted, 5-17](#_bookmark1091) installation

[guidelines for security, 10-12](#_bookmark2221) intruders

[*See*](#_bookmark194) security attacks invoker’s rights

[about, 4-30](#_bookmark761)

[procedure privileges, used with, 4-29](#_bookmark756) [procedure security, 4-30](#_bookmark762)

[secure application roles, 5-13](#_bookmark1046)

[secure application roles, requirement for enabling, 5-13](#_bookmark1044)

IP addresses [falsifying, 10-16](#_bookmark2258)

##### J

[JAVA\_ADMIN role, 4-13](#_bookmark614)

[JAVA\_DEPLOY role, 4-14](#_bookmark615)

[JAVADEBUGPRIV role, 4-13](#_bookmark610)

[JAVAIDPRIV role, 4-13](#_bookmark611)

[JAVASYSPRIV role, 4-13](#_bookmark612)

[JAVAUSERPRIV role, 4-13](#_bookmark613)

JDBC connections

JDBC Thin Driver proxy authentication [configuring, 3-36](#_bookmark420)

[with real user, 3-40](#_bookmark449)

[JDBC/OCI proxy authentication, 3-36](#_bookmark420) [multiple user sessions, 3-40](#_bookmark449)

[Oracle Virtual Private Database, 7-38](#_bookmark1534) [JMXSERVER role, 4-14](#_bookmark616)

##### K

[Kerberos authentication, 3-28](#_bookmark364) [configuring for SYSDBA or SYSOPER](#_bookmark326)

[access, 3-23](#_bookmark326)

[password management, 10-9](#_bookmark2190) key generation

[encryption, 8-4](#_bookmark1565) key storage

[encryption, 8-5](#_bookmark1569) key transmission

[encryption, 8-5](#_bookmark1567)

##### L

[LBAC\_DBA role, 4-14](#_bookmark617)

[least privilege principle, 10-2](#_bookmark2134) [about, 10-2](#_bookmark2135)

[granting user privileges, 10-2](#_bookmark2133) [middle-tier privileges, 3-41](#_bookmark454)

libraries

[security guidelines, 10-3](#_bookmark2141) lightweight users

[example using a global application context, 6-35](#_bookmark1303) [Lightweight Directory Access Protocol](#_bookmark1486)

[(LDAP), 7-28](#_bookmark1486)

listener

[not an Oracle owner, 10-15](#_bookmark2249)

[preventing online administration, 10-14](#_bookmark2242) [restrict privileges, 10-15](#_bookmark2250)

[secure administration, 10-16](#_bookmark2260) listener.ora file

[administering remotely, 10-14](#_bookmark2246)

[default location, 10-17](#_bookmark2278)

[online administration, preventing, 10-14](#_bookmark2243) [TCPS, securing, 10-17](#_bookmark2277)

LOBS

[auditing, 9-38](#_bookmark1906) lock and expire

[default accounts, 10-4](#_bookmark2157) [predefined user accounts, 10-4](#_bookmark2154)

log files

[auditing, default location, 9-17](#_bookmark1718) [owned by trusted user, 10-11](#_bookmark2210) [Windows Event Viewer, 9-54](#_bookmark1980)

[logical reads limit, 2-11](#_bookmark126) logon triggers

[auditing current session, 9-25](#_bookmark1775) [examples, 6-11](#_bookmark1196)

[externally initialized application contexts, 6-12](#_bookmark1205) [secure application roles, 4-23](#_bookmark710)

[LOGSTDBY\_ADMINISTRATOR role, 4-14](#_bookmark618)

##### M

malicious database administrators

[*See also*](#_bookmark1556) security attacks [manager default password, 10-9](#_bookmark2181) mandatory auditing

[about, 9-4](#_bookmark1621)

[syslog, written to, 9-4](#_bookmark1626) memory

[users, viewing, 2-18](#_bookmark179)

[MERGE INTO statement, affected by DBMS\_ RLS.ADD\_POLICY statement\_types parameter, 7-8](#_bookmark1405)

methods

[privileges on, 4-33](#_bookmark776)

[MGMT\_USER role, 4-14](#_bookmark619)

middle-tier systems

[client identifiers, 3-44](#_bookmark472)

[enterprise user connections, 3-43](#_bookmark465)

[password-based proxy authentication, 3-43](#_bookmark464) [privileges, limiting, 3-41](#_bookmark454)

[proxies authenticating users, 3-42](#_bookmark456) [proxying but not authenticating users, 3-42](#_bookmark459) [reauthenticating user to database, 3-43](#_bookmark462)

[USERENV namespace attributes, accessing, 6-17](#_bookmark1225) monitoring user actions

[*See also*](#_bookmark1606) auditing, standard auditing, fine-grained auditing

[multiplex multiple-client network sessions, 10-16](#_bookmark2259) My Oracle Support

[security patches, downloading, 10-2](#_bookmark2127)

##### N

Net8

[*See*](#_bookmark2255) Oracle Net network auditing

[about, 9-34](#_bookmark1870)

[removing, 9-35](#_bookmark1877) network authentication

[external authentication, 3-34](#_bookmark406) [guidelines for securing, 10-9](#_bookmark2191) [roles, granting using, 4-45](#_bookmark847) [Secure Sockets Layer, 3-28](#_bookmark358) [smart cards, 10-9](#_bookmark2191)

[third-party services, 3-28](#_bookmark362)

[token cards, 10-9](#_bookmark2191)

[X.509 certificates, 10-9](#_bookmark2191) network connections

[denial-of-service (DoS) attacks, addressing, 10-16](#_bookmark2264) [guidelines for security, 10-13,](#_bookmark2227) [10-14](#_bookmark2240)

[securing, 10-14](#_bookmark2241) network IP addresses

[guidelines for security, 10-16](#_bookmark2261) [new features, security, 3-xxv](#_bookmark6) NOAUDIT statement

[audit options, removing, 9-23](#_bookmark1754)

[default object audit options, disabling, 9-31](#_bookmark1836) [network auditing, removing, 9-35](#_bookmark1878)

[object auditing, removing, 9-31](#_bookmark1835) [privilege auditing, removing, 9-27](#_bookmark1796) [statement auditing, removing, 9-25](#_bookmark1779)

nondatabase users [about, 6-22](#_bookmark1250)

[audit record information, 9-57](#_bookmark1998) [auditing, 9-50](#_bookmark1962)

[clearing session data, 6-32](#_bookmark1291)

[creating client session-based application](#_bookmark1321) [contexts, 6-42 to](#_bookmark1321) [6-44](#_bookmark1331)

global application contexts [package example, 6-29](#_bookmark1288) [reason for using, 6-22](#_bookmark1251) [setting, 6-28](#_bookmark1286)

[tutorial, 6-35](#_bookmark1304)

One Big Application User authentication [about, 7-38](#_bookmark1536)

[features compromised by, 5-2](#_bookmark1002) [security risks, 5-2](#_bookmark999)

Oracle Virtual Private Database [how it works with, 7-38](#_bookmark1539)

[tutorial for creating a policy group, 7-28](#_bookmark1490)

[*See also*](#_bookmark1252) application contexts, client identifiers

##### O

O7\_DICTIONARY\_ACCESSIBILITY initialization parameter

[about, 4-3](#_bookmark515)

[auditing privileges on SYS objects, 9-3,](#_bookmark1612) [9-7](#_bookmark1647) [data dictionary protection, 10-10](#_bookmark2200)

[default setting, 10-10](#_bookmark2203)

[securing data dictionary with, 4-3](#_bookmark515) object columns

[auditing, 9-38](#_bookmark1906)

[object privileges, 10-2](#_bookmark2136)

[about, 4-24](#_bookmark720)

[granting on behalf of the owner, 4-39](#_bookmark818) [managing, 5-16](#_bookmark1072)

[revoking, 4-42](#_bookmark828)

[revoking on behalf of owner, 4-42](#_bookmark832) [schema object privileges, 4-24](#_bookmark720)

[*See also*](#_bookmark720) schema object privileges [synonyms, 4-25](#_bookmark727)

objects

[applications, managing privileges in, 5-16](#_bookmark1072) [granting privileges, 5-17](#_bookmark1081)

privileges

[applications, 5-16](#_bookmark1076)

[managing, 4-33](#_bookmark776)

[protecting in shared schemas, 5-16](#_bookmark1067) [protecting in unique schemas, 5-15](#_bookmark1061) [SYS schema, access to, 4-4](#_bookmark522)

[OEM\_ADVISOR role, 4-14](#_bookmark620)

[OEM\_MONITOR role, 4-14](#_bookmark621)

[OLAP\_DBA role, 4-14](#_bookmark622)

[OLAP\_USER role, 4-14](#_bookmark623)

[OLAP\_XS\_ADMIN role, 4-14](#_bookmark624)

One Big Application User authentication

[*See*](#_bookmark1241) nondatabase users operating system audit trail [age, controlling, 9-64](#_bookmark2038)

[audited actions in common with database audit trail, 9-11](#_bookmark1690)

[size, controlling, 9-62](#_bookmark2035) operating systems

[accounts, 4-46](#_bookmark851) authentication

[about, 3-27](#_bookmark350)

[advantages, 3-27](#_bookmark351)

[disadvantages, 3-27](#_bookmark354)

[roles, using, 4-45](#_bookmark847)

[authentication, external, 3-34](#_bookmark403)

[default permissions, 10-10](#_bookmark2207) [enabling and disabling roles, 4-47](#_bookmark857) [operating system account privileges,](#_bookmark2206)

[limiting, 10-10](#_bookmark2206)

[role identification, 4-46](#_bookmark850)

[roles and, 4-10](#_bookmark580)

[roles, granting using, 4-45](#_bookmark847) [users, limiting number of, 10-10](#_bookmark2205)

[OPW-00005 error, 2-9](#_bookmark111)

[ORA-01720 error, 4-28](#_bookmark741)

[ORA-06512 error, 9-48](#_bookmark1957)

[ORA-1536 error, 2-5](#_bookmark83)

[ORA-24247 error, 4-51,](#_bookmark890) [9-48](#_bookmark1956)

[ORA-28009 error, 4-3](#_bookmark520)

[ORA-28040 error, 3-26](#_bookmark344)

Oracle Advanced Security

[network authentication services, 10-9](#_bookmark2187) [network traffic encryption, 10-16](#_bookmark2267)

[user access to application schemas, 5-16](#_bookmark1069) Oracle Call Interface (OCI)

[application contexts, client session-based, 6-42](#_bookmark1322) [proxy authentication, 3-36](#_bookmark421)

[Oracle Virtual Private Database, how it works with, 7-38](#_bookmark1535)

[proxy authentication with real user, 3-40](#_bookmark446) [security-related initialization](#_bookmark1097)

[parameters, 5-17 to](#_bookmark1097) [5-20](#_bookmark1121) Oracle Connection Manager

[securing client networks with, 10-16](#_bookmark2257) Oracle Data Pump

[exported data from VPD policies, 7-37](#_bookmark1528) Oracle Database Enterprise User Security

[password security threats, 3-16](#_bookmark294) Oracle Enterprise Security Manager

[role management with, 3-29](#_bookmark368) Oracle home

[default permissions, disallowing modification, 10-10](#_bookmark2208)

Oracle Internet Directory (OID)

[authenticating with directory-based service, 3-29](#_bookmark367) [SYSDBA and SYSOPER access, controlling, 3-22](#_bookmark322)

Oracle Java Virtual Machine (OJVM) [permissions, restricting, 10-3](#_bookmark2148)

Oracle Label Security (OLS)

[Oracle Virtual Private Database, using with, 7-36](#_bookmark1518) Oracle Net

[firewall support, 10-15](#_bookmark2254) Oracle Real Application Clusters

[archive timestamp for audit records, 9-70](#_bookmark2062) [global application contexts, 6-23](#_bookmark1255)

[global contexts, 6-22](#_bookmark1246) Oracle Technology Network

[security alerts, 10-2](#_bookmark2125)

Oracle Virtual Private Database (VPD) [about, 7-1](#_bookmark1345)

[ANSI operations, 7-34](#_bookmark1507) application contexts

[tutorial, 7-23](#_bookmark1477)

[used with, 7-3](#_bookmark1366) applications

[how it works with, 7-35](#_bookmark1509)

[users who are database users, how it works with, 7-38](#_bookmark1532)

[applications using for security, 5-2](#_bookmark1010) [automatic reparsing, how it works with, 7-35](#_bookmark1513) [benefits, 7-2](#_bookmark1349)

[column level, 7-8](#_bookmark1409) column masking behavior

[enabling, 7-10](#_bookmark1415)

[restrictions, 7-11](#_bookmark1418)

[column-level display, 7-8](#_bookmark1410)

[components, 7-4](#_bookmark1371)

[configuring, 7-5 to](#_bookmark1379) [7-19](#_bookmark1460)

[cursors, shared, 7-4](#_bookmark1369)

[edition-based redefinitions, 7-34](#_bookmark1502) [editions, results in, 6-24](#_bookmark1268)

[Enterprise User Security proxy authentication, how it works with, 7-38](#_bookmark1533)

[exporting data, 7-36](#_bookmark1523)

[exporting data using Data Pump Export, 7-37](#_bookmark1529) [finding information about, 7-39](#_bookmark1544)

[flashback query, how it works with, 7-35](#_bookmark1515) function

[auditing, 9-29](#_bookmark1815)

[components, 7-4](#_bookmark1374)

[how it is executed, 7-3](#_bookmark1361)

[JDBC proxy authentication, how it works with, 7-38](#_bookmark1534)

[nondatabase user applications, how works with, 7-38](#_bookmark1539)

[OCI proxy authentication, how it works with, 7-38](#_bookmark1535)

Oracle Label Security [exceptions in behavior, 7-36](#_bookmark1521)

[using with, 7-36](#_bookmark1518)

[outer join operations, 7-34](#_bookmark1507) [performance benefit, 7-3](#_bookmark1355)

policies, Oracle Virtual Private Database [about, 7-6](#_bookmark1381)

[applications, validating, 7-13](#_bookmark1434) [attaching to database object, 7-7](#_bookmark1399) [column display, 7-8](#_bookmark1410)

[column-level display, default, 7-9](#_bookmark1413) [dynamic, 7-15](#_bookmark1440)

[multiple, 7-13](#_bookmark1431)

[optimizing performance, 7-14](#_bookmark1437) [privileges used to run, 7-3](#_bookmark1362)

[SQL statements, specifying, 7-7](#_bookmark1403) policy groups

[about, 7-11](#_bookmark1423)

[benefits, 7-12](#_bookmark1424)

[creating, 7-12](#_bookmark1426)

[default, 7-12](#_bookmark1428)

[tutorial, implementation, 7-28](#_bookmark1491) policy types

[context sensitive, about, 7-17](#_bookmark1453) [context sensitive, when to use, 7-18](#_bookmark1459) [context-sensitive, audited, 9-33](#_bookmark1859)

[DYNAMIC, 7-15](#_bookmark1439)

[dynamic, audited, 9-33](#_bookmark1856)

[shared context sensitive, about, 7-18](#_bookmark1456) [shared context sensitive, when to use, 7-18](#_bookmark1459) [shared static, about, 7-16](#_bookmark1447)

[shared static, when to use, 7-17](#_bookmark1450) [static, about, 7-16](#_bookmark1444)

[static, audited, 9-33](#_bookmark1858) [static, when to use, 7-17](#_bookmark1450)

[summary of features, 7-19](#_bookmark1462) [SELECT FOR UPDATE statements in](#_bookmark1504)

[policies, 7-34](#_bookmark1504)

[tutorial, simple, 7-20](#_bookmark1469)

[user models, 7-38](#_bookmark1531)

[Web-based applications, how it works with, 7-38](#_bookmark1540) Oracle Wallet Manager

[X.509 Version 3 certificates, 3-30](#_bookmark371) Oracle wallets

[authentication method, 3-30](#_bookmark372) Oracle Warehouse Builder

[roles, predefined, 4-14](#_bookmark626) ORACLE\_DATAPUMP access driver

[guidelines for security, 10-11](#_bookmark2217) Oracle*MetaLink*

[*See*](#_bookmark2129) My Oracle Support ORAPWD password utility

[case sensitivity in passwords, 3-14](#_bookmark286) [password file authentication, 3-25](#_bookmark335) [permissions to run, 3-25](#_bookmark336)

ORAPWD utility

[changing SYS password with, 2-9](#_bookmark109) [ORDADMIN role, 4-14](#_bookmark625)

OS\_ROLES initialization parameter [operating system role grants, 4-47](#_bookmark857)

[operating-system authorization and, 4-19](#_bookmark671) [REMOTE\_OS\_ROLES and, 4-47](#_bookmark860)

[using, 4-46](#_bookmark850) outer join operations

[Oracle Virtual Private Database affect on, 7-34](#_bookmark1506) [OWB\_DESIGNCENTER\_VIEW role, 4-15](#_bookmark627)

[OWB\_USER role, 4-15](#_bookmark628)

[OWB$CLIENT role, 4-14](#_bookmark626)

##### P

packages

[auditing, 9-32](#_bookmark1852)

[examples, 4-32](#_bookmark774)

[examples of privilege use, 4-32](#_bookmark772) privileges

[divided by construct, 4-31](#_bookmark771) [executing, 4-29,](#_bookmark748) [4-31](#_bookmark771)

[parallel execution servers, 6-9](#_bookmark1182)

[parallel query, and SYS\_CONTEXT, 6-9](#_bookmark1180) pass phrase

[read and parse server.key file, 10-17](#_bookmark2283) password files

[case sensitivity, effect on SEC\_CASE\_SENSITIVE\_ LOGON parameter, 3-14](#_bookmark282)

[how used to authenticate administrators, 3-25](#_bookmark334) PASSWORD statement

[about, 2-8](#_bookmark104)

[PASSWORD\_LIFE\_TIME profile parameter, 3-8](#_bookmark251) [PASSWORD\_LOCK\_TIME profile parameter, 3-7](#_bookmark235) [PASSWORD\_REUSE\_MAX profile parameter, 3-7](#_bookmark247) [PASSWORD\_REUSE\_TIME profile parameter, 3-7](#_bookmark245) passwords

[about managing, 3-4](#_bookmark203)

[account locking, 3-6](#_bookmark230) administrator

[authenticating with, 3-25](#_bookmark334) [guidelines for securing, 10-9](#_bookmark2180)

[aging and expiration, 3-8](#_bookmark251) [ALTER PROFILE statement, 3-4](#_bookmark204) [altering, 2-8](#_bookmark106)

[application design guidelines, 5-4](#_bookmark1019) [applications, strategies for protecting](#_bookmark1014)

[passwords, 5-3](#_bookmark1014) [brute force attacks, 3-2](#_bookmark191)

[case sensitivity setting, SEC\_CASE\_SENSITIVE\_ LOGIN, 3-13](#_bookmark278)

[case sensitivity, configuring, 3-13](#_bookmark277) [changing for roles, 4-17](#_bookmark652) complexity verification

[about, 3-11](#_bookmark270)

[guidelines for security, 10-8](#_bookmark2172) [connecting without, 3-27](#_bookmark350) [CREATE PROFILE statement, 3-4](#_bookmark204)

[danger in storing as clear text, 10-9](#_bookmark2193) [database user authentication, 3-26](#_bookmark342) default profile settings

[about, 3-4](#_bookmark211)

[default user account, 10-9](#_bookmark2175) [default, finding, 3-4](#_bookmark208)

[delays for incorrect passwords, 3-2](#_bookmark192) [duration, 10-9](#_bookmark2185)

[encrypting, 3-2,](#_bookmark190) [10-9](#_bookmark2193) [examples of creating, 3-3](#_bookmark198) expiring

[explicitly, 3-10](#_bookmark264)

[procedure for, 3-8](#_bookmark252)

[proxy account passwords, 3-39](#_bookmark438) [with grace period, 3-9](#_bookmark257)

[failed logins, resetting, 3-7](#_bookmark238) [finding version of, 3-14](#_bookmark284) [grace period, example, 3-10](#_bookmark260) [guidelines for security, 10-7](#_bookmark2168) [history, 3-7](#_bookmark241)

[Java code example to read passwords, 5-7](#_bookmark1029) [life time set too low, 3-10](#_bookmark266)

[lifetime for, 3-8](#_bookmark251)

[lock time, 3-7](#_bookmark235)

[management rules, 10-9](#_bookmark2186)

[managing, 3-3](#_bookmark201)

[maximum reuse time, 3-7](#_bookmark247) [ORAPWD password utility, 3-14](#_bookmark286)

[password complexity verification, 3-11](#_bookmark271) [password file risks, 3-25](#_bookmark337)

[PASSWORD\_LOCK\_TIME profile parameter, 3-7](#_bookmark235) [PASSWORD\_REUSE\_MAX profile](#_bookmark247)

[parameter, 3-7](#_bookmark247) [PASSWORD\_REUSE\_TIME profile](#_bookmark245)

[parameter, 3-7](#_bookmark245)

[policies, 3-3](#_bookmark201)

[privileges for changing for roles, 4-17](#_bookmark655) [privileges to alter, 2-7](#_bookmark99)

[protections, built-in, 3-2](#_bookmark188)

[proxy authentication, 3-43](#_bookmark464) requirements

[additional, 10-7](#_bookmark2171)

[minimum, 3-3](#_bookmark198)

[reusing, 3-7,](#_bookmark245) [10-9](#_bookmark2185)

[reusing passwords, 3-7](#_bookmark242)

[roles authenticated by passwords, 4-17](#_bookmark647) [roles enabled by SET ROLE statement, 4-18](#_bookmark659) [secure external password store, 3-17](#_bookmark300) [security risks, 3-25](#_bookmark338)

[SYS account, 2-9](#_bookmark108)

[SYS and SYSTEM, 10-9](#_bookmark2184)

[used in roles, 4-8](#_bookmark558) UTLPWDMG.SQL password script

[password management, 3-11](#_bookmark271)

[verified using SHA-1 hashing algorithm, 3-15,](#_bookmark291) [3-16](#_bookmark296)

[*See also*](#_bookmark189) authentication, and access control list (ACL), wallet access

performance

[application contexts, 6-2](#_bookmark1132)

[auditing, 9-4](#_bookmark1630)

[database audit trail, moving to different tablespace, 9-60](#_bookmark2017)

[Oracle Virtual Private Database policies, 7-3](#_bookmark1356) [Oracle Virtual Private Database policy types, 7-14](#_bookmark1437) [resource limits and, 2-10](#_bookmark116)

permissions [default, 10-10](#_bookmark2207)

[run-time facilities, 10-3](#_bookmark2144) PKI

[*See*](#_bookmark370) public key infrastructure (PKI)

PL/SQL

[auditing of statements within, 9-8](#_bookmark1651) [roles in procedures, 4-9](#_bookmark571)

PL/SQL functions [auditing, 9-33](#_bookmark1862) PL/SQL packages

[auditing, 9-32,](#_bookmark1853) [9-33](#_bookmark1862) PL/SQL procedures

[auditing, 9-33](#_bookmark1862)

[setting application context, 6-7](#_bookmark1165) PMON background process

[application contexts, cleaning up, 6-4](#_bookmark1153) positional parameters

[security risks, 5-6](#_bookmark1025)

[principle of least privilege, 10-2](#_bookmark2134) [about, 10-2](#_bookmark2135)

[granting user privileges, 10-2](#_bookmark2133) [middle-tier privileges, 3-41](#_bookmark454)

privileges

[about, 4-1](#_bookmark498)

[access control lists, checking for external network services, 4-66](#_bookmark934)

altering

[passwords, 2-8](#_bookmark106)

[users, 2-7](#_bookmark99)

[altering role authentication method, 4-17](#_bookmark655) [applications, managing, 5-11](#_bookmark1032)

[audited when default auditing is enabled, 9-35](#_bookmark1888) [auditing use of, 9-26](#_bookmark1788)

[auditing, recommended settings for, 10-21](#_bookmark2303) [cascading revokes, 4-43](#_bookmark838)

[column, 4-40](#_bookmark820)

[compiling procedures, 4-31](#_bookmark767)

[creating or replacing procedures, 4-31](#_bookmark764) [creating users, 2-2](#_bookmark63)

[dropping profiles, 2-14](#_bookmark150)

[finding information about, 4-71](#_bookmark963) granting

[about, 4-4,](#_bookmark530) [4-37](#_bookmark792)

[examples, 4-32](#_bookmark772)

[object privileges, 4-25,](#_bookmark724) [4-38](#_bookmark810)

[system, 4-37](#_bookmark795)

[system privileges, 4-37](#_bookmark792)

[grants, listing, 4-73](#_bookmark969) [grouping with roles, 4-6](#_bookmark541) [managing, 5-16](#_bookmark1073)

[middle tier, 3-41](#_bookmark454)

[object, 4-23,](#_bookmark714) [4-24,](#_bookmark716) [5-17](#_bookmark1081)

[granting and revoking, 4-25](#_bookmark724) [on selected columns, 4-43](#_bookmark834) [procedures, 4-29](#_bookmark749)

[creating and replacing, 4-31](#_bookmark765) [executing, 4-29](#_bookmark750)

[in packages, 4-31](#_bookmark771) [reasons to grant, 4-2](#_bookmark501) revoking privileges

[about, 4-4](#_bookmark530)

[object, 4-42](#_bookmark828)

[object privileges, cascading effect, 4-44](#_bookmark842) [object privileges, requirements for, 4-42](#_bookmark829) [schema object, 4-25](#_bookmark724)

[revoking system privileges, 4-41](#_bookmark825) roles

[creating, 4-16](#_bookmark644)

[dropping, 4-21](#_bookmark695)

[restrictions on, 4-10](#_bookmark578)

[roles, why better to grant, 4-2](#_bookmark503) [schema object, 4-24](#_bookmark720)

[DML and DDL operations, 4-26](#_bookmark729) [packages, 4-31](#_bookmark771)

[procedures, 4-29](#_bookmark749)

[SQL statements permitted, 5-17](#_bookmark1082) [synonyms and underlying objects, 4-25](#_bookmark727) system

[granting and revoking, 4-4](#_bookmark530)

[SELECT ANY DICTIONARY, 10-10](#_bookmark2202) [SYSTEM and OBJECT, 10-2](#_bookmark2136)

system privileges [about, 4-3](#_bookmark509)

[trigger privileges, 4-30](#_bookmark759)

[used for Oracle Virtual Private Database policy functions, 7-3](#_bookmark1363)

view privileges

[creating a view, 4-27](#_bookmark740) [using a view, 4-28](#_bookmark743)

[views, 4-27](#_bookmark738)

[*See also*](#_bookmark507) access control list (ACL) and system privileges.

procedures

[auditing, 9-29,](#_bookmark1822) [9-32](#_bookmark1854)

[compiling, 4-31](#_bookmark767) definer’s rights

[about, 4-29](#_bookmark757)

[roles disabled, 4-9](#_bookmark572)

[examples of, 4-32](#_bookmark774)

[examples of privilege use, 4-32](#_bookmark772) invoker’s rights

[about, 4-30](#_bookmark762)

[roles used, 4-9](#_bookmark574) privileges for procedures

[create or replace, 4-31](#_bookmark765) [executing, 4-29](#_bookmark751)

[executing in packages, 4-31](#_bookmark771) [privileges required for, 4-31](#_bookmark764) [security enhanced by, 4-30](#_bookmark758)

process monitor process (PMON) [cleans up timed-out sessions, 2-11](#_bookmark131)

[PRODUCT\_USER\_PROFILE table, 4-22](#_bookmark705)

[SQL commands, disabling with, 4-22](#_bookmark704) products and options

[install only as necessary, 10-12](#_bookmark2223) profile parameters

[FAILED\_LOGIN\_ATTEMPTS, 3-5](#_bookmark214)

[PASSWORD\_GRACE\_TIME, 3-5,](#_bookmark216) [3-10](#_bookmark261)

[PASSWORD\_LIFE\_TIME, 3-5,](#_bookmark217) [3-8,](#_bookmark254) [3-10](#_bookmark267)

[PASSWORD\_LOCK\_TIME, 3-5,](#_bookmark218) [3-7](#_bookmark236)

[PASSWORD\_REUSE\_MAX, 3-5,](#_bookmark219) [3-7](#_bookmark248)

[PASSWORD\_REUSE\_TIME, 3-6,](#_bookmark220) [3-7](#_bookmark246)

[profiles, 2-12](#_bookmark141)

[about, 2-12](#_bookmark142)

[creating, 2-13](#_bookmark145)

[dropping, 2-14](#_bookmark150)

[finding information about, 2-15](#_bookmark167) [finding settings for default profile, 2-18](#_bookmark177) [managing, 2-12](#_bookmark141)

[password management, 3-4](#_bookmark205) [privileges for dropping, 2-14](#_bookmark150) [specifying for user, 2-7](#_bookmark93) [viewing, 2-17](#_bookmark176)

proxy authentication [about, 3-36,](#_bookmark418) [3-37](#_bookmark422)

[advantages, 3-37](#_bookmark424)

[auditing operations, 3-35](#_bookmark411)

[auditing users, 9-27](#_bookmark1803)

[client-to-middle tier sequence, 3-40](#_bookmark451) [creating proxy user accounts, 3-38](#_bookmark427) middle-tier

[authorizing but not authenticating users, 3-42](#_bookmark460) [authorizing to proxy and authenticate](#_bookmark457)

[users, 3-42](#_bookmark457)

[limiting privileges, 3-41](#_bookmark453)

[reauthenticating users, 3-43](#_bookmark463)

[passwords, expired, 3-39](#_bookmark439)

[privileges required for creating users, 3-38](#_bookmark428) [secure external password store, used with, 3-40](#_bookmark441) [security benefits, 3-37](#_bookmark425)

[users, passing real identity of, 3-40](#_bookmark447) proxy user accounts

[privileges required for creation, 3-38](#_bookmark428) [PROXY\_USER attribute, 6-17](#_bookmark1226)

[PROXY\_USERS view, 3-39](#_bookmark435)

pseudo columns [USER, 4-29](#_bookmark744)

PUBLIC

[procedures and, 4-45](#_bookmark845)

[role, 4-45](#_bookmark845)

public key infrastructure (PKI) [about, 3-29](#_bookmark369)

PUBLIC role [about, 4-5](#_bookmark534)

[granting and revoking privileges to, 4-45](#_bookmark844) [security domain of users, 4-9](#_bookmark568)

[security risk in privileges granted to, 4-5](#_bookmark536) PUBLIC\_DEFAULT profile

[profiles, dropping, 2-14](#_bookmark151)

##### Q

quotas

[tablespace, 2-5](#_bookmark76)

[temporary segments and, 2-5](#_bookmark79) [unlimited, 2-5](#_bookmark86)

[viewing, 2-17](#_bookmark174)

##### R

[RADIUS authentication, 3-29](#_bookmark365)

[read-only mode, affect on AUDIT\_TRAIL parameter, 9-10](#_bookmark1673)

reads

[limits on data blocks, 2-11](#_bookmark126) RECOVERY\_CATALOG\_OWNER role

[about, 4-15](#_bookmark629) redo log files

[auditing committed and rolled back transactions, 10-20](#_bookmark2297)

REFERENCES privilege

[CASCADE CONSTRAINTS option, 4-43](#_bookmark836)

[revoking, 4-43](#_bookmark834)

[SQL statements permitted, 5-17](#_bookmark1092) [remote authentication, 10-13](#_bookmark2234)

REMOTE\_OS\_AUTHENT initialization parameter [guideline for securing, 10-13](#_bookmark2237)

[setting, 3-34](#_bookmark404)

[remote\_os\_authentication, 10-13](#_bookmark2236) REMOTE\_OS\_ROLES initialization parameter

[OS role management risk on network, 4-48](#_bookmark861) [setting, 4-19](#_bookmark673)

resource limits [about, 2-10](#_bookmark115)

[call level, limiting, 2-11](#_bookmark122)

[connection time for each session, 2-11](#_bookmark132) [CPU time, limiting, 2-11](#_bookmark124)

[determining values for, 2-12](#_bookmark137) [idle time in each session, 2-11](#_bookmark129) [logical reads, limiting, 2-11](#_bookmark126)

[private SGA space for each session, 2-12](#_bookmark134) [profiles, 2-12](#_bookmark141)

[session level, limiting, 2-10](#_bookmark120) sessions

[concurrent for user, 2-11](#_bookmark128) [elapsed connection time, 2-11](#_bookmark133) [idle time, 2-11](#_bookmark130)

[SGA space, 2-12](#_bookmark135)

[types, 2-10](#_bookmark118) RESOURCE privilege

[CREATE SCHEMA statement, needed for, 5-15](#_bookmark1064) [RESOURCE role, 4-33](#_bookmark780)

[about, 4-15](#_bookmark630)

REVOKE CONNECT THROUGH clause

[revoking proxy authorization, 3-39](#_bookmark436) REVOKE statement

[system privileges and roles, 4-41](#_bookmark825) [when takes effect, 4-48](#_bookmark864)

revoking privileges and roles [cascading effects, 4-43](#_bookmark838)

[on selected columns, 4-43](#_bookmark834) [REVOKE statement, 4-41](#_bookmark825)

[specifying ALL, 4-24](#_bookmark717)

[when using operating-system roles, 4-47](#_bookmark855) role identification

[operating system accounts, 4-46](#_bookmark851) ROLE\_SYS\_PRIVS view

[application privileges, 5-12](#_bookmark1035) ROLE\_TAB\_PRIVS view

[application privileges, finding, 5-12](#_bookmark1035) roles

[about, 4-1,](#_bookmark499) [4-6](#_bookmark543)

[ADM\_PARALLEL\_EXECUTE\_TASK role, 4-11](#_bookmark587) [ADMIN OPTION and, 4-38](#_bookmark801)

[advantages in application use, 5-11](#_bookmark1034) [application, 4-8,](#_bookmark563) [4-21,](#_bookmark698) [5-14,](#_bookmark1051) [5-16](#_bookmark1073)

[application privileges, 5-11](#_bookmark1033) [applications, for user, 5-14](#_bookmark1051)

[AQ\_ADMINISTRATOR\_ROLE role, 4-11](#_bookmark588)

[AQ\_USER\_ROLE role, 4-11](#_bookmark589)

[AUTHENTICATEDUSER role, 4-11](#_bookmark590)

[authorization, 4-17](#_bookmark657)

[authorized by enterprise directory service, 4-19](#_bookmark680) [CAPI\_USER\_ROLE role, 4-11](#_bookmark591)

[changing authorization for, 4-17](#_bookmark651) [changing passwords, 4-17](#_bookmark652)

[CONNECT, 4-6](#_bookmark546)

CONNECT role [about, 4-11](#_bookmark592)

[create your own, 10-6](#_bookmark2162) [CSW\_USR\_ROLE role, 4-11](#_bookmark593)

[CTXAPP role, 4-12](#_bookmark594)

[CWM\_USER role, 4-12](#_bookmark595)

[database role, users, 5-14](#_bookmark1053) [DATAPUMP\_EXP\_FULL\_DATABASE role, 4-12](#_bookmark596)

[DATAPUMP\_IMP\_FULL\_DATABASE role, 4-12](#_bookmark597)

[DBA role, 4-12](#_bookmark598)

[DBFS\_ROLE role, 4-12](#_bookmark599)

[DDL statements and, 4-9](#_bookmark576) [default, 4-48](#_bookmark871)

[default, setting for user, 2-7](#_bookmark96)

[definer’s rights procedures disable, 4-9](#_bookmark572) [DELETE\_CATALOG\_ROLE role, 4-12](#_bookmark600)

[dependency management in, 4-10](#_bookmark578) [disabling, 4-48](#_bookmark867)

[dropping, 4-21](#_bookmark694)

[EJBCLIENT role, 4-12](#_bookmark601)

[enabled or disabled, 4-7,](#_bookmark550) [4-20](#_bookmark687)

[enabling, 4-48,](#_bookmark867) [5-14](#_bookmark1051)

[enterprise, 3-30,](#_bookmark379) [4-19](#_bookmark683)

[EXECUTE\_CATALOG\_ROLE role, 4-12](#_bookmark602)

[EXP\_FULL\_DATABASE role, 4-12](#_bookmark603)

[finding information about, 4-71](#_bookmark963) [functionality, 4-2,](#_bookmark502) [4-6](#_bookmark548)

[functionality of, 4-6](#_bookmark549)

[GATHER\_SYSTEM\_STATISTICS role, 4-13](#_bookmark604)

[global authorization, 4-19](#_bookmark680)

[about, 4-19](#_bookmark679) global roles

[about, 3-30](#_bookmark377)

[creating, 4-19](#_bookmark681)

[external sources, and, 4-18](#_bookmark668) [GLOBAL\_AQ\_USER\_ROLE role, 4-13](#_bookmark605)

[GRANT statement, 4-47](#_bookmark858) [granted to other roles, 4-7](#_bookmark551) granting roles

[about, 4-37](#_bookmark792)

[methods for, 4-20](#_bookmark688)

[system, 4-37](#_bookmark795)

[system privileges, 4-4](#_bookmark529) [guidelines for security, 10-6](#_bookmark2161)

[HS\_ADMIN\_EXECUTE\_ROLE role, 4-13](#_bookmark606)

[HS\_ADMIN\_ROLE role, 4-13](#_bookmark607)

[HS\_ADMIN\_SELECT\_ROLE role, 4-13](#_bookmark608)

[IMP\_FULL\_DATABASE role, 4-13](#_bookmark609)

[in applications, 4-8](#_bookmark557)

[indirectly granted, 4-7](#_bookmark552)

[invoker’s rights procedures use, 4-9](#_bookmark574) [JAVA\_ADMIN role, 4-13](#_bookmark614)

[JAVA\_DEPLOY role, 4-14](#_bookmark615)

[JAVADEBUGPRIV role, 4-13](#_bookmark610)

[JAVAIDPRIV role, 4-13](#_bookmark611)

[JAVASYSPRIV role, 4-13](#_bookmark612)

[JAVAUSERPRIV role, 4-13](#_bookmark613)

[JMXSERVER role, 4-14](#_bookmark616)

[job responsibility privileges only, 10-6](#_bookmark2163) [LBAC\_DBA role, 4-14](#_bookmark617)

[listing grants, 4-73](#_bookmark972)

[listing privileges and roles in, 4-75](#_bookmark986) [listing roles, 4-74](#_bookmark983)

[LOGSTDBY\_ADMINISTRATOR role, 4-14](#_bookmark618)

[management using the operating system, 4-45](#_bookmark847) managing roles

[about, 4-6](#_bookmark541)

[categorizing users, 5-16](#_bookmark1073)

[managing through operating system, 4-10](#_bookmark580) [maximum number a user can enable, 4-49](#_bookmark876) [MGMT\_USER role, 4-14](#_bookmark619)

[multibyte characters in names, 4-16](#_bookmark646) [multibyte characters in passwords, 4-18](#_bookmark662) [naming, 4-6](#_bookmark544)

[network authorization, 4-19](#_bookmark674) [network client authorization, 4-19](#_bookmark675) [OEM\_ADVISOR role, 4-14](#_bookmark620)

[OEM\_MONITOR role, 4-14](#_bookmark621)

[OLAP\_DBA role, 4-14](#_bookmark622)

[OLAP\_USER role, 4-14](#_bookmark623)

[OLAP\_XS\_ADMIN role, 4-14](#_bookmark624)

[One Big Application User, compromised by, 5-2](#_bookmark1004) [operating system, 4-46](#_bookmark851)

[operating system authorization, 4-19](#_bookmark671) [operating system granting of, 4-47](#_bookmark857) [operating system identification of, 4-46](#_bookmark850)

[operating system management and the shared server, 4-47](#_bookmark860)

[operating system-managed, 4-47](#_bookmark853)

[operating-system authorization, 4-18](#_bookmark667)

[ORDADMIN role, 4-14](#_bookmark625)

[OWB\_DESIGNCENTER\_VIEW role, 4-15](#_bookmark627)

[OWB\_USER role, 4-15](#_bookmark628)

[OWB$CLIENT role, 4-14](#_bookmark626)

[predefined, 4-11](#_bookmark583)

[privileges for creating, 4-16](#_bookmark644) [privileges for dropping, 4-21](#_bookmark695)

[privileges, changing authorization method for, 4-17](#_bookmark655)

[privileges, changing passwords, 4-17](#_bookmark655) [RECOVERY\_CATALOG\_OWNER role, 4-15](#_bookmark629)

[RESOURCE role, 4-15](#_bookmark630)

[restricting from tool users, 4-21](#_bookmark698) [restrictions on privileges of, 4-10](#_bookmark578)

[REVOKE statement, 4-47](#_bookmark858)

[revoking, 4-20,](#_bookmark688) [4-41](#_bookmark825)

[revoking ADMIN option, 4-42](#_bookmark826) [SCHEDULER\_ADMIN role, 4-15](#_bookmark631)

[schemas do not contain, 4-6](#_bookmark544) [security domains of, 4-9](#_bookmark567)

[SELECT\_CATALOG\_ROLE role, 4-15](#_bookmark632)

[SET ROLE statement, 4-47](#_bookmark857) [setting in PL/SQL blocks, 4-9](#_bookmark574) [SNMPAGENT role, 4-15](#_bookmark633)

[SPATIAL\_CSW\_ADMIN role, 4-15](#_bookmark634)

[SPATIAL\_WFS\_ADMIN role, 4-15](#_bookmark635)

[unique names for, 4-16](#_bookmark645) [use of passwords with, 4-8](#_bookmark558) [user, 4-9,](#_bookmark565) [5-16](#_bookmark1073)

[users capable of granting, 4-20](#_bookmark691) [uses of, 4-6,](#_bookmark548) [4-8](#_bookmark560)

[WFS\_USR\_ROLE role, 4-15](#_bookmark636)

[WITH GRANT OPTION and, 4-39](#_bookmark815)

[without authorization, 4-17](#_bookmark650)

[WM\_ADMIN\_ROLE role, 4-16](#_bookmark637)

[XDB\_SET\_INVOKER roles, 4-16](#_bookmark639)

[XDB\_WEBSERVICES role, 4-16](#_bookmark640)

[XDB\_WEBSERVICES\_OVER\_HTTP role, 4-16](#_bookmark641)

[XDB\_WEBSERVICES\_WITH\_PUBLIC role, 4-16](#_bookmark642)

[XDBADMIN role, 4-16](#_bookmark638)

[*See also*](#_bookmark1041) secure application roles root file paths

[for files and packages outside the database, 10-3](#_bookmark2149) row-level security

[*See*](#_bookmark1346) fine-grained access control, Oracle Virtual Private Database (VPD)

[RSA private key, 10-17](#_bookmark2282) [run-time facilities, 10-3](#_bookmark2145)

[restriction permissions, 10-3](#_bookmark2144)

##### S

Sample Schemas

[remove or relock for production, 10-12](#_bookmark2224) [test database, 10-12](#_bookmark2224)

[sample schemas, 10-12](#_bookmark2225) Sarbanes-Oxley Act

[auditing to meet compliance, 9-2](#_bookmark1607) SCHEDULER\_ADMIN role

[about, 4-15](#_bookmark631) schema object auditing

[enabling, 9-30](#_bookmark1824)

[removing, 9-31](#_bookmark1834)

[schema object privileges, 4-24](#_bookmark720) schema objects

[audit options, removing, 9-31](#_bookmark1835) [auditing, 9-29](#_bookmark1812)

[auditing procedures or functions, 9-31](#_bookmark1831) [cascading effects on revoking, 4-44](#_bookmark842) [default audit options, 9-30](#_bookmark1829)

[default tablespace for, 2-4](#_bookmark73) [dropped users, owned by, 2-14](#_bookmark153) [enabling audit options on, 9-30](#_bookmark1825) [granting privileges, 4-38](#_bookmark810)

privileges

[DML and DDL operations, 4-26](#_bookmark729) [granting and revoking, 4-25](#_bookmark724) [view privileges, 4-27](#_bookmark738)

[privileges on, 4-24](#_bookmark720) [privileges to access, 4-24](#_bookmark716) [privileges with, 4-24](#_bookmark716)

[removing audit options, 9-27](#_bookmark1796) [revoking privileges, 4-42](#_bookmark828)

[schema-independent users, 5-16](#_bookmark1070) schemas

[auditing, recommended settings for, 10-21](#_bookmark2302) [private, 3-31](#_bookmark385)

[shared among enterprise users, 3-31](#_bookmark389) [shared, protecting objects in, 5-16](#_bookmark1067) [unique, 5-15](#_bookmark1059)

[unique, protecting objects in, 5-15](#_bookmark1061) SCOTT user account

[restricting privileges of, 10-6](#_bookmark2164) script files

[audit trail views, removing, 9-84](#_bookmark2117) [CATNOAUD.SQL, 9-84](#_bookmark2117)

[scripts, authenticating users in, 3-17](#_bookmark302) SEC\_CASE\_SENSITIVE\_LOGON initialization

parameter [about, 3-13](#_bookmark278)

[conflict with SQLNET.ALLOWED\_LOGON\_ VERSION setting, 3-13](#_bookmark279)

[SEC\_MAX\_FAILED\_LOGIN\_ATTEMPTS](#_bookmark1111)

[initialization parameter, 5-19](#_bookmark1111) [SEC\_PROTOCOL\_ERROR\_FURTHER\_ACTION](#_bookmark1107)

[initialization parameter, 5-18](#_bookmark1107) [SEC\_PROTOCOL\_ERROR\_TRACE\_ACTION](#_bookmark1104)

[initialization parameter, 5-18](#_bookmark1104)

[sec\_relevant\_cols\_opt parameter, 7-10](#_bookmark1416) [SEC\_RETURN\_SERVER\_RELEASE\_BANNER](#_bookmark1114)

[initialization parameter, 5-19](#_bookmark1114) [SEC\_USER\_AUDIT\_ACTION\_BANNER initialization](#_bookmark1119)

[parameter, 5-20](#_bookmark1119) [SEC\_USER\_UNAUTHORIZED\_ACCESS\_BANNER](#_bookmark1118)

[initialization parameter, 5-20](#_bookmark1118) secconf.sql script

[audit settings, 9-36](#_bookmark1892)

[password settings, 3-6](#_bookmark225) secure application roles

[about, 4-22](#_bookmark709)

[creating, 5-12](#_bookmark1039)

[creating PL/SQL package, 5-13](#_bookmark1042) [finding with DBA\_ROLES view, 4-71](#_bookmark966) [invoker’s rights, 5-13](#_bookmark1046)

[invoker’s rights requirement, 5-13](#_bookmark1044) [package for, 5-13](#_bookmark1043)

[SET ROLE statement, 5-14](#_bookmark1049)

[user environment information from SYS\_](#_bookmark1047) [CONTEXT SQL function, 5-13,](#_bookmark1047) [5-14](#_bookmark1048)

[using to ensure database connection, 4-23](#_bookmark711) secure external password store

[about, 3-17](#_bookmark300)

[client configuration, 3-18](#_bookmark308)

[examples, 3-17](#_bookmark306)

[how it works, 3-17](#_bookmark305)

[proxy authentication, used with, 3-40](#_bookmark441) Secure Sockets Layer (SSL)

[about, 3-28](#_bookmark359)

[certificate key algorithm, 10-17](#_bookmark2280) [cipher suites, 10-17](#_bookmark2280)

[configuration files, securing, 10-17](#_bookmark2272) [configuring for SYSDBA or SYSOPER](#_bookmark328)

[access, 3-24](#_bookmark328)

[global users with private schemas, 3-31](#_bookmark386) [guidelines for security, 10-16](#_bookmark2270)

[listener, administering, 10-14](#_bookmark2244)

[mode, 10-17](#_bookmark2279)

[pass phrase, 10-17](#_bookmark2283) [RSA private key, 10-17](#_bookmark2282)

[securing SSL connection, 10-16](#_bookmark2271) [server.key file, 10-17](#_bookmark2282)

[TCPS, 10-17](#_bookmark2275)

security

[application enforcement of, 4-8](#_bookmark557) default user accounts

[locked and expired automatically, 10-4](#_bookmark2156) [locking and expiring, 10-4](#_bookmark2153)

[domains, enabled roles and, 4-20](#_bookmark687) [enforcement in application, 5-2](#_bookmark1008) [enforcement in database, 5-2](#_bookmark1008)

[multibyte characters in role names, 4-16](#_bookmark646) [multibyte characters in role passwords, 4-18](#_bookmark662) [passwords, 3-26](#_bookmark343)

policies

[applications, 5-1](#_bookmark994)

[SQL\*Plus users, restricting, 4-21](#_bookmark698) [tables or views, 7-2](#_bookmark1352)

[procedures enhance, 4-30](#_bookmark758)

[resources, additional, 1-2](#_bookmark54)

[roles, advantages in application use, 5-11](#_bookmark1034)

[*See also*](#_bookmark2146) security risks [security alerts, 10-2](#_bookmark2126) security attacks

[access to server after protocol errors, preventing, 5-18](#_bookmark1106)

[application context values, attempts to change, 6-6](#_bookmark1160)

[application design to prevent attacks, 5-3](#_bookmark1014) [command line recall attacks, 5-4,](#_bookmark1017) [5-5](#_bookmark1024) [denial of service, 10-16](#_bookmark2264)

denial-of-service

[bad packets, addressing, 5-18](#_bookmark1101)

[denial-of-service attacks through listener, 10-16](#_bookmark2265) [disk flooding, preventing, 5-17](#_bookmark1100)

[eavesdropping, 10-14](#_bookmark2238)

[encryption, problems not solved by, 8-2](#_bookmark1557) [falsified IP addresses, 10-13](#_bookmark2231)

[falsified or stolen client system identities, 10-13](#_bookmark2231) [hacked operating systems or applications, 10-13](#_bookmark2231) [intruders, 8-2](#_bookmark1557)

[non-SYS activities audited, 9-3](#_bookmark1616) [password cracking, 3-2](#_bookmark195)

[password protections against, 3-2](#_bookmark191)

[preventing malicious attacks from clients, 5-17](#_bookmark1098)

[preventing password theft with proxy authentication and secure external password store, 3-40](#_bookmark442)

[session ID, need for encryption, 6-34](#_bookmark1297) [shoulder surfing, 5-5](#_bookmark1024)

[SQL injection attacks, 5-4](#_bookmark1020) [unlimited authenticated requests,](#_bookmark1110)

[preventing, 5-19](#_bookmark1110)

[user session output, hiding from intruders, 6-12](#_bookmark1200)

[*See also*](#_bookmark443) security risks security domains

[enabled roles and, 4-7](#_bookmark550) security patches

[about, 10-2](#_bookmark2124)

[downloading, 10-2](#_bookmark2127) security policies

[*See*](#_bookmark1421) Oracle Virtual Private Database, policies security risks

[ad hoc tools, 4-21](#_bookmark701)

[application users not being database users, 5-2](#_bookmark999) [applications enforcing rather than database, 5-2](#_bookmark1009) [audit records being tampered with, 9-18](#_bookmark1722)

[bad packets to server, 5-18](#_bookmark1102) [database audit trail, protecting, 9-3](#_bookmark1611) [database version displaying, 5-19](#_bookmark1113)

[encryption keys, users managing, 8-7](#_bookmark1574) [password files, 3-25](#_bookmark337)

[passwords exposed in large deployments, 3-17](#_bookmark303) [passwords, exposing in programs or scripts, 5-5](#_bookmark1023) [positional parameters in SQL scripts, 5-6](#_bookmark1025) [privileges carelessly granted, 4-5](#_bookmark538)

[privileges granted to PUBLIC role, 4-5](#_bookmark536) [remote user impersonating another user, 4-19](#_bookmark676) [sensitive data in audit trail, 10-18](#_bookmark2290)

[server falsifying identities, 10-17](#_bookmark2281) [users with multiple roles, 5-15](#_bookmark1054) [*See also*](#_bookmark444) security attacks

security settings scripts audit settings

[secconf.sql, 9-36](#_bookmark1892) password settings

[secconf.sql, 3-6](#_bookmark225)

[undoaud.sql, 9-36](#_bookmark1891)

[undopwd.sql, 3-6](#_bookmark223)

SELECT ANY DICTIONARY privilege [data dictionary, accessing, 10-10](#_bookmark2201)

[exclusion from GRANT ALL PRIVILEGES privilege, 10-10](#_bookmark2204)

[SELECT FOR UPDATE statement in Virtual Private Database policies, 7-34](#_bookmark1504)

SELECT privilege

[SQL statements permitted, 5-17](#_bookmark1093) SELECT\_CATALOG\_ROLE role

[about, 4-15](#_bookmark632)

[SYS schema objects, enabling access to, 4-4](#_bookmark525) [separation of duty concepts, Glossary-5](#_bookmark2352) sequences

[auditing, 9-29](#_bookmark1816) server.key file

[pass phrase to read and parse, 10-17](#_bookmark2283)

SESSION\_ROLES data dictionary view [PUBLIC role, 4-5](#_bookmark537)

SESSION\_ROLES view

[queried from PL/SQL block, 4-9](#_bookmark573) sessions

[listing privilege domain of, 4-74](#_bookmark979) [memory use, viewing, 2-18](#_bookmark179) [time limits on, 2-11](#_bookmark129)

[when auditing options take effect, 9-8](#_bookmark1653) SET ROLE statement

[application code, including in, 5-15](#_bookmark1057) [associating privileges with role, 5-15](#_bookmark1055) [disabling roles with, 4-48](#_bookmark867)

[enabling roles with, 4-48](#_bookmark867) [secure application roles, 5-14](#_bookmark1049)

[when using operating-system roles, 4-47](#_bookmark858) SGA

[*See*](#_bookmark1133) System Global Area (SGA)

SHA-1 hashing algorithm [about, 3-15](#_bookmark292)

[enabling exclusive mode, 3-16](#_bookmark296)

[how it increases password safety, 3-15](#_bookmark291) [recommended by Oracle, 3-15](#_bookmark293)

Shared Global Area (SGA)

[*See*](#_bookmark1134) System Global Area (SGA) shared server

[limiting private SQL areas, 2-12](#_bookmark134) [operating system role management](#_bookmark860)

[restrictions, 4-47](#_bookmark860)

[shoulder surfing, 5-5](#_bookmark1024)

[SHOW PARAMETER command, 9-8](#_bookmark1663)

smart cards

[guidelines for security, 10-9](#_bookmark2192) SNMPAGENT role

[about, 4-15](#_bookmark633)

[SPATIAL\_CSW\_ADMIN role, 4-15](#_bookmark634)

[SPATIAL\_WFS\_ADMIN role, 4-15](#_bookmark635)

[SQL injection attacks, 5-4](#_bookmark1020) SQL statements

[audited when default auditing is enabled, 9-35](#_bookmark1889) auditing

[about, 9-23](#_bookmark1763)

[configuring, 9-24](#_bookmark1768)

[removing, 9-25](#_bookmark1780)

[when records generated, 9-8](#_bookmark1650) [dynamic, 6-8](#_bookmark1178)

[object privileges permitting in applications, 5-17](#_bookmark1081) [privileges required for, 4-24,](#_bookmark721) [5-17](#_bookmark1083)

[resource limits and, 2-11](#_bookmark122) [restricting ad hoc use, 4-21](#_bookmark702)

SQL\*Net

[*See*](#_bookmark2256) Oracle Net SQL\*Plus

[connecting with, 3-27](#_bookmark352) [restricting ad hoc use, 4-21](#_bookmark702) [statistics monitor, 2-12](#_bookmark138)

SQLNET.ALLOWED\_LOGON\_VERSION parameter

[conflict with SEC\_CASE\_SENSITIVE\_LOGON FALSE setting, 3-13](#_bookmark280)

SSL

[*See*](#_bookmark360) Secure Sockets Layer standard audit trail

[activities always recorded, 9-4](#_bookmark1623) [AUDIT SQL statement, 9-20](#_bookmark1735) [auditing standard audit trail, 9-61](#_bookmark2020) [controlling size of, 9-59](#_bookmark2013)

[disabling, 9-8](#_bookmark1658)

[enabling, 9-8](#_bookmark1659) [maximum size of, 9-60](#_bookmark2014)

[NOAUDIT SQL statement, 9-23](#_bookmark1755) [records, purging, 9-65](#_bookmark2042)

[size, reducing, 9-74](#_bookmark2074)

[transaction independence, 9-8](#_bookmark1652)

[when created, 9-8](#_bookmark1649) standard auditing

[about, 9-7](#_bookmark1643)

[administrative users on all platforms, 9-53](#_bookmark1976) [affected by editions, 9-29](#_bookmark1820)

[archiving audit trail, 9-61](#_bookmark2026) [audit option levels, 9-20](#_bookmark1736) audit trails

[database, 9-58](#_bookmark2002) auditing

[default auditing, enabling, 9-35](#_bookmark1883) [cursors, affect on auditing, 9-22](#_bookmark1746) [database audit trail records, 9-58](#_bookmark2002) [DDL statement auditing, 9-23](#_bookmark1765) [default options, 9-30](#_bookmark1829)

[default options, disabling, 9-31](#_bookmark1836) directory object auditing

[about, 9-32](#_bookmark1842)

[configuring, 9-32](#_bookmark1844)

[removing, 9-32](#_bookmark1847)

[disabling options versus auditing, 9-23](#_bookmark1757) [DML statements, 9-23](#_bookmark1766)

[information stored in operating system file, 9-13](#_bookmark1699) [mandatory auditing, 9-4](#_bookmark1621)

network auditing [about, 9-34](#_bookmark1870)

[configuring, 9-34](#_bookmark1875)

[error types recorded, 9-34](#_bookmark1871) [removing, 9-35](#_bookmark1879)

[non-SYS activities audited, 9-3](#_bookmark1615) object auditing

[*See*](#_bookmark1813) standard auditing, schema object [operating system audit trail, 9-13](#_bookmark1697)

[file location, 9-17](#_bookmark1717)

[operating system audit trail using, 9-16](#_bookmark1709) privilege auditing

[about, 9-26](#_bookmark1788)

[configuring, 9-26](#_bookmark1792)

[multitier environment, 9-27](#_bookmark1801)

[options, 9-26](#_bookmark1793)

[removing, 9-27](#_bookmark1797)

[types, 9-26](#_bookmark1790)

[privileges needed, 9-7](#_bookmark1646) [procedures or functions, 9-31](#_bookmark1831) [range of focus, 9-20](#_bookmark1733)

records

[archiving, 9-61](#_bookmark2027)

[removing, 9-23](#_bookmark1754) schema object

[objects created in the future, 9-31](#_bookmark1838) schema object auditing

[about, 9-29](#_bookmark1812)

[enabling, 9-30](#_bookmark1824)

[example, 9-30](#_bookmark1825)

[options, 9-29](#_bookmark1822)

[removing, 9-31](#_bookmark1835)

[types, 9-29](#_bookmark1817) SQL statement

[*See*](#_bookmark1781) standard auditing, statement auditing statement auditing

[about, 9-23](#_bookmark1763)

[all statements for individual users, 9-24](#_bookmark1772) [all statements for the current session, 9-24](#_bookmark1774) [configuring, 9-24](#_bookmark1768)

[multitier environment, 9-27](#_bookmark1801)

[removing, 9-25](#_bookmark1780)

[SQL statement shortcuts by individual users, 9-24](#_bookmark1773)

[statement level, 9-24](#_bookmark1769) [types you can audit, 9-23](#_bookmark1765)

[statement executions, number of, 9-21](#_bookmark1744) [successful or unsuccessful, 9-21](#_bookmark1740)

[setting, 9-21](#_bookmark1740)

[SYS users, 9-53](#_bookmark1976)

[syslog audit trail on UNIX systems, 9-18](#_bookmark1720) [user, 9-23](#_bookmark1751)

[*See also*](#_bookmark1660) auditing, standard audit trail, SYS.AUD$ table

[statement\_types parameter of DBMS\_RLS.ADD\_ POLICY procedure, 7-7](#_bookmark1403)

[STMT\_AUDIT\_OPTION\_MAP table, 9-12](#_bookmark1693)

storage

[quotas and, 2-5](#_bookmark80)

[unlimited quotas, 2-5](#_bookmark86) stored procedures

[using privileges granted to PUBLIC, 4-45](#_bookmark845) strong authentication

[centrally controlling SYSDBA and SYSOPER access to multiple databases, 3-22](#_bookmark320)

[guideline, 10-9](#_bookmark2188) symbolic links

[restricting, 10-10](#_bookmark2209) synonyms

[object privileges, 4-25](#_bookmark727) [privileges, guidelines on, 10-3](#_bookmark2143)

SYS account

[changing password, 2-9](#_bookmark108)

[policy enforcement, 7-37](#_bookmark1525) SYS and SYSTEM

[passwords, 10-9](#_bookmark2184) SYS schema

[objects, access to, 4-4](#_bookmark522) SYS\_CONTEXT function

[about, 6-7](#_bookmark1166)

[auditing current session, 9-25](#_bookmark1776) [auditing nondatabase users with, 9-51](#_bookmark1965) [database links, 6-9](#_bookmark1185)

[dynamic SQL statements, 6-8](#_bookmark1178) [example, 6-10](#_bookmark1191)

[parallel query, 6-9](#_bookmark1180)

[STATIC policies, 7-17](#_bookmark1451)

[syntax, 6-7](#_bookmark1171)

[validating users, 5-13](#_bookmark1047)

[SYS\_DEFAULT Oracle Virtual Private Database policy group, 7-12](#_bookmark1428)

SYS.AUD$ table [about, 9-58](#_bookmark2003)

[archiving, 9-61](#_bookmark2028)

[audit records, writing to, 9-10](#_bookmark1671) [contents, 9-58](#_bookmark2005)

[data values in audited statement, 9-58](#_bookmark2007) [location in Oracle Database Vault](#_bookmark1613)

[environment, 9-3](#_bookmark1613)

[modifying manually, dangers of, 9-55](#_bookmark1991) [non-SYS actions audited, 9-3](#_bookmark1617) [purging, 9-61](#_bookmark2028)

[too full or unavailable, 9-58](#_bookmark2008)

[*See also*](#_bookmark2004) standard auditing SYSAUX tablespace

[moving database audit trail tables to, 9-60](#_bookmark2017) SYS.FGA\_LOG$

[fine-grained auditing, 9-38](#_bookmark1911) SYS.FGA\_LOG$ table

[about, 9-58](#_bookmark2003)

[archiving, 9-61](#_bookmark2028)

[contents, 9-58](#_bookmark2005)

[data values in audited statement, 9-58](#_bookmark2007) [non-SYS actions audited, 9-3](#_bookmark1618)

[purging, 9-61](#_bookmark2028)

[too full or unavailable, 9-58](#_bookmark2008) SYS.FGA\_LOGS$ table

[*See also*](#_bookmark2006) fine-grained auditing syslog audit trail

[about, 9-18](#_bookmark1723)

[appearance, 9-19](#_bookmark1727)

[configuring, 9-19](#_bookmark1730)

[format, 9-19](#_bookmark1725)

[format when AUDIT\_TRAIL is set to XML, 9-11](#_bookmark1685) [mandatory audit records written to, 9-4](#_bookmark1624)

[SYSMAN user account, 10-9](#_bookmark2183)

[SYS-privileged connections, 10-2](#_bookmark2137) System Global Area (SGA)

[application contexts, storing in, 6-2](#_bookmark1135) [global application context information](#_bookmark1244)

[location, 6-22](#_bookmark1244)

[limiting private SQL areas, 2-12](#_bookmark134) [system privileges, 10-2](#_bookmark2136)

[about, 4-3](#_bookmark509)

[ADMIN OPTION, 4-5](#_bookmark532) ANY

[guidelines for security, 10-10](#_bookmark2199) [ANY system privileges, 4-3](#_bookmark513)

[GRANT ANY OBJECT PRIVILEGE, 4-39,](#_bookmark817) [4-42](#_bookmark831) [GRANT ANY PRIVILEGE, 4-5](#_bookmark532)

[granting, 4-37](#_bookmark795)

[granting and revoking, 4-4](#_bookmark529) [power of, 4-3](#_bookmark510)

[restriction needs, 4-3](#_bookmark512)

[revoking, cascading effect of, 4-44](#_bookmark840) [SELECT ANY DICTIONARY, 10-10](#_bookmark2202)

[SYSTEM\_PRIVILEGE\_MAP table, 9-12](#_bookmark1694)

##### T

tables

[auditing, 9-29](#_bookmark1816)

[privileges on, 4-26](#_bookmark729) tablespaces

[assigning defaults for users, 2-4](#_bookmark72) [default quota, 2-5](#_bookmark77)

[quotas for users, 2-5](#_bookmark76) [quotas, viewing, 2-17](#_bookmark174) temporary

[assigning to users, 2-6](#_bookmark91) [unlimited quotas, 2-5](#_bookmark86)

TCPS protocol

[Secure Sockets Layer, used with, 10-14](#_bookmark2245) [tnsnames.ora file, used in, 10-17](#_bookmark2275)

[TELNET service, 10-16](#_bookmark2268)

[TFTP service, 10-16](#_bookmark2268)

[time measurement for statement execution, 7-15](#_bookmark1441) [token cards, 10-9](#_bookmark2192)

[top-level SQL statements, 9-6](#_bookmark1638) trace files

[access to, importance of restricting, 10-11](#_bookmark2213) [bad packets, 5-18](#_bookmark1103)

[location of, finding, 6-45](#_bookmark1339) [transparent data encryption, 8-7](#_bookmark1576) [transparent tablespace encryption, 8-7](#_bookmark1577) triggers

[audit data, recording, 9-55](#_bookmark1989) [auditing, 9-29,](#_bookmark1822) [9-32,](#_bookmark1855) [9-33](#_bookmark1862) [CREATE TRIGGER ON, 5-17](#_bookmark1088)

logon

[examples, 6-11](#_bookmark1196)

[externally initialized application contexts, 6-12](#_bookmark1205) [privileges for executing, 4-30](#_bookmark759)

[roles, 4-9](#_bookmark571)

[WHEN OTHERS exception, 6-11](#_bookmark1199) troubleshooting

[finding errors by checking trace files, 6-45](#_bookmark1340) trusted procedure

[database session-based application contexts, 6-2](#_bookmark1129) [tsnames.ora configuration file, 10-17](#_bookmark2276)

tutorials

[application context, database session-based, 6-13](#_bookmark1208) auditing

[creating policy to audit nondatabase users, 9-50](#_bookmark1962)

[creating policy using email alert, 9-44](#_bookmark1947) [external network services, using email alert, 9-44](#_bookmark1949) [global application context with client session](#_bookmark1305)

[ID, 6-35](#_bookmark1305)

nondatabase users

[creating Oracle Virtual Private Database policy group, 7-28](#_bookmark1492)

[global application context, 6-35](#_bookmark1304)

Oracle Virtual Private Database [policy groups, 7-28](#_bookmark1491)

[policy implementing, 7-23](#_bookmark1477)

[simple example, 7-20](#_bookmark1469)

[*See also*](#_bookmark1209) examples types

[creating, 4-34](#_bookmark786)

[privileges on, 4-33](#_bookmark776) user defined

[creation requirements, 4-34](#_bookmark784)

##### U

UDP and TCP ports

[close for ALL disabled services, 10-16](#_bookmark2268) UGA

[*See*](#_bookmark1136) User Global Area (UGA) [undoaud.sql script, 9-36](#_bookmark1891)

[undopwd.sql script, 3-6](#_bookmark223) UNIX systems

[audit data written to syslog, 9-4](#_bookmark1625) [UNIX systems, auditing users on, 9-18](#_bookmark1720)

[UNLIMITED TABLESPACE privilege, 2-5,](#_bookmark87) [2-6](#_bookmark88)

UPDATE privilege [revoking, 4-43](#_bookmark834)

user accounts

[administrative user passwords, 10-9](#_bookmark2179) [default user account, 10-9](#_bookmark2176)

[password guidelines, 10-7](#_bookmark2168)

[passwords, encrypted, 10-9](#_bookmark2193)

[proxy users, 3-38](#_bookmark429) USER function

[global application contexts, 6-25](#_bookmark1273) User Global Area (UGA)

[application contexts, storing in, 6-2](#_bookmark1137) user names

[schemas, 5-15](#_bookmark1059)

[USER pseudo column, 4-29](#_bookmark744)

[user sessions, multiple within single database connection, 3-40](#_bookmark448)

user-defined columns [auditing, 9-38](#_bookmark1906)

[USERENV function, 6-8,](#_bookmark1175) [8-9](#_bookmark1590) USERENV namespace

[about, 6-8](#_bookmark1172)

[client identifiers, 3-44](#_bookmark470)

[*See also*](#_bookmark478) CLIENT\_IDENTIFIER USERENV attribute users

[administrative option (ADMIN OPTION), 4-38](#_bookmark802) [altering, 2-8](#_bookmark101)

[application users not known to database, 3-44](#_bookmark469) [assigning unlimited quotas for, 2-5](#_bookmark86)

[auditing, 9-23](#_bookmark1751)

[database role, current, 5-14](#_bookmark1053) [default roles, changing, 2-7](#_bookmark96) [default tablespaces, 2-4](#_bookmark72)

[dropping, 2-14](#_bookmark153)

[dropping profiles and, 2-14](#_bookmark151) [dropping roles and, 4-21](#_bookmark694) [enabling roles for, 5-14](#_bookmark1051)

[enterprise, 3-30,](#_bookmark380) [4-19](#_bookmark684)

[enterprise, shared schema protection, 5-16](#_bookmark1068) external authentication

[about, 3-32](#_bookmark395)

[advantages, 3-33](#_bookmark398)

[assigning profiles, 2-13](#_bookmark146)

[operating system, 3-34](#_bookmark402)

[user creation, 3-33](#_bookmark400)

[finding information about, 2-15](#_bookmark167)

[finding information about authentication, 3-47](#_bookmark489) [global, 3-30](#_bookmark378)

[assigning profiles, 2-13](#_bookmark147) hosts, connecting to multiple

[*See*](#_bookmark880) external network services, fine-grained access to

[information about, viewing, 2-16](#_bookmark172) [listing roles granted to, 4-73](#_bookmark972) [memory use, viewing, 2-18](#_bookmark179)

[network authentication, external, 3-34](#_bookmark406) [nondatabase, 6-22,](#_bookmark1247) [6-28](#_bookmark1283)

[objects after dropping, 2-14](#_bookmark154)

[operating system external authentication, 3-34](#_bookmark403) [password encryption, 3-2](#_bookmark190)

privileges

[for changing passwords, 2-7](#_bookmark99) [for creating, 2-2](#_bookmark63)

[granted to, listing, 4-73](#_bookmark969)

[of current database role, 5-14](#_bookmark1053) profiles

[creating, 2-13](#_bookmark145)

[specifying, 2-7](#_bookmark93)

[proxy authentication, 3-36](#_bookmark418) [proxy users, connecting as, 3-37](#_bookmark422) [PUBLIC role, 4-9,](#_bookmark568) [4-45](#_bookmark844)

[quota limits for tablespace, 2-5](#_bookmark82) [restricting application roles, 4-21](#_bookmark698) [roles and, 4-7](#_bookmark554)

[for types of users, 4-9](#_bookmark565) [schema-independent, 5-16](#_bookmark1070)

[schemas, private, 3-31](#_bookmark385) [security domains of, 4-9](#_bookmark567) [security, about, 2-1](#_bookmark58)

[tablespace quotas, 2-5](#_bookmark76) [tablespace quotas, viewing, 2-17](#_bookmark174) [user accounts, creating, 2-2](#_bookmark63)

[user models and Oracle Virtual Private Database, 7-38](#_bookmark1531)

[user name, specifying with CREATE USER statement, 2-2](#_bookmark66)

[views for finding information about, 2-15](#_bookmark165) UTLPWDMG.SQL

[about, 3-11](#_bookmark271)

[guidelines for security, 10-8](#_bookmark2172)

##### V

[V$LOGMNR\_CONTENTS data dictionary view, 9-59](#_bookmark2010)

[valid node checking, 10-16](#_bookmark2262) views

[about, 4-27](#_bookmark737)

access control list data

[external network services, 4-71](#_bookmark952) [wallet access, 4-71](#_bookmark953)

[application contexts, 6-45](#_bookmark1334)

[audit trail, 9-80](#_bookmark2099)

[auditing, 9-29](#_bookmark1816)

[authentication, 3-47](#_bookmark490)

[DBA\_COL\_PRIVS, 4-73](#_bookmark977)

[DBA\_NETWORK\_ACL\_PRIVILEGES, 4-66,](#_bookmark935) [4-71](#_bookmark957)

[DBA\_NETWORK\_ACLS, 4-71](#_bookmark956)

[DBA\_ROLE\_PRIVS, 4-73](#_bookmark973)

[DBA\_ROLES, 4-74](#_bookmark984)

[DBA\_SYS\_PRIVS, 4-73](#_bookmark970)

[DBA\_TAB\_PRIVS, 4-73](#_bookmark975)

[DBA\_USERS\_WITH\_DEFPWD, 3-4](#_bookmark208)

[DBA\_WALLET\_ACLS, 4-71](#_bookmark958)

[encrypted data, 8-15](#_bookmark1599)

[Oracle Virtual Private Database policies, 7-39](#_bookmark1544) [privileges, 4-27,](#_bookmark737) [4-71](#_bookmark963)

[profiles, 2-15](#_bookmark168)

[ROLE\_ROLE\_PRIVS, 4-75](#_bookmark987)

[ROLE\_SYS\_PRIVS, 4-75](#_bookmark988)

[ROLE\_TAB\_PRIVS, 4-75](#_bookmark989)

[roles, 4-71](#_bookmark963)

[security applications of, 4-28](#_bookmark743) [SESSION\_PRIVS, 4-74](#_bookmark981)

[SESSION\_ROLES, 4-74](#_bookmark980)

[USER\_NETWORK\_ACL\_PRIVILEGES, 4-71](#_bookmark959)

[users, 2-15](#_bookmark168)

Virtual Private Database

[*See*](#_bookmark1347) Oracle Virtual Private Database VPD

[*See*](#_bookmark1347) Oracle Virtual Private Database [vulnerable run-time call, 10-4](#_bookmark2150)

[made more secure, 10-4](#_bookmark2151)

##### W

Wallet Manager

[*See*](#_bookmark373) Oracle Wallet Manager wallets

[authentication method, 3-30](#_bookmark372)

[*See also*](#_bookmark884) access control lists (ACL), wallet access Web applications

[user connections, 6-22,](#_bookmark1247) [6-28](#_bookmark1283) Web-based applications

[Oracle Virtual Private Database, how it works with, 7-38](#_bookmark1540)

[WFS\_USR\_ROLE role, 4-15](#_bookmark636)

WHEN OTHERS exceptions [logon triggers, used in, 6-11](#_bookmark1199)

[WHERE clause, dynamic SQL, 7-5](#_bookmark1375) [Windows native authentication, 3-25](#_bookmark331) [WM\_ADMIN\_ROLE role, 4-16](#_bookmark637)

##### X

X.509 certificates

[guidelines for security, 10-9](#_bookmark2192)

[XDB\_SET\_INVOKER role, 4-16](#_bookmark639)

[XDB\_WEBSERVICES role, 4-16](#_bookmark640) XDB\_WEBSERVICES\_OVER\_HTTP role

[about, 4-16](#_bookmark641)

[XDB\_WEBSERVICES\_WITH\_PUBLIC role, 4-16](#_bookmark642)

[XDBADMIN role, 4-16](#_bookmark638) XML

[AUDIT\_TRAIL XML setting, 9-11](#_bookmark1679) [AUDIT\_TRAIL XML, EXTENDED setting, 9-11](#_bookmark1681)

XML, EXTENDED AUDIT\_TRAIL setting

[used with DB in AUDIT\_TRAIL, 9-10](#_bookmark1675) [used with XML in AUDIT\_TRAIL, 9-11](#_bookmark1681)