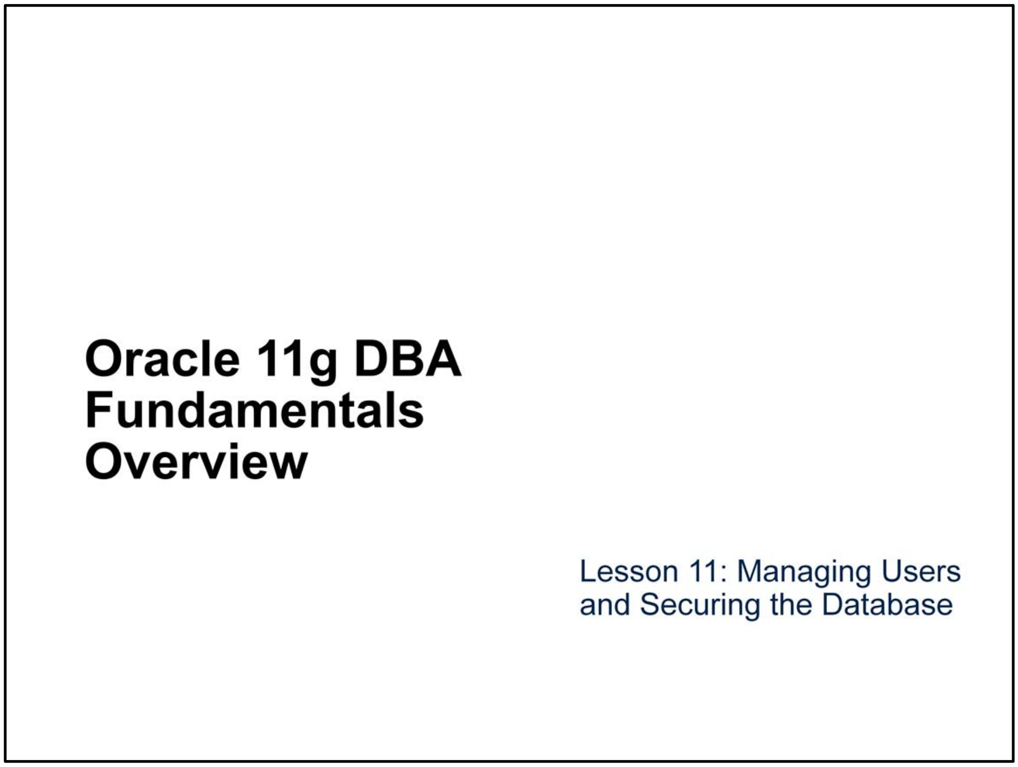
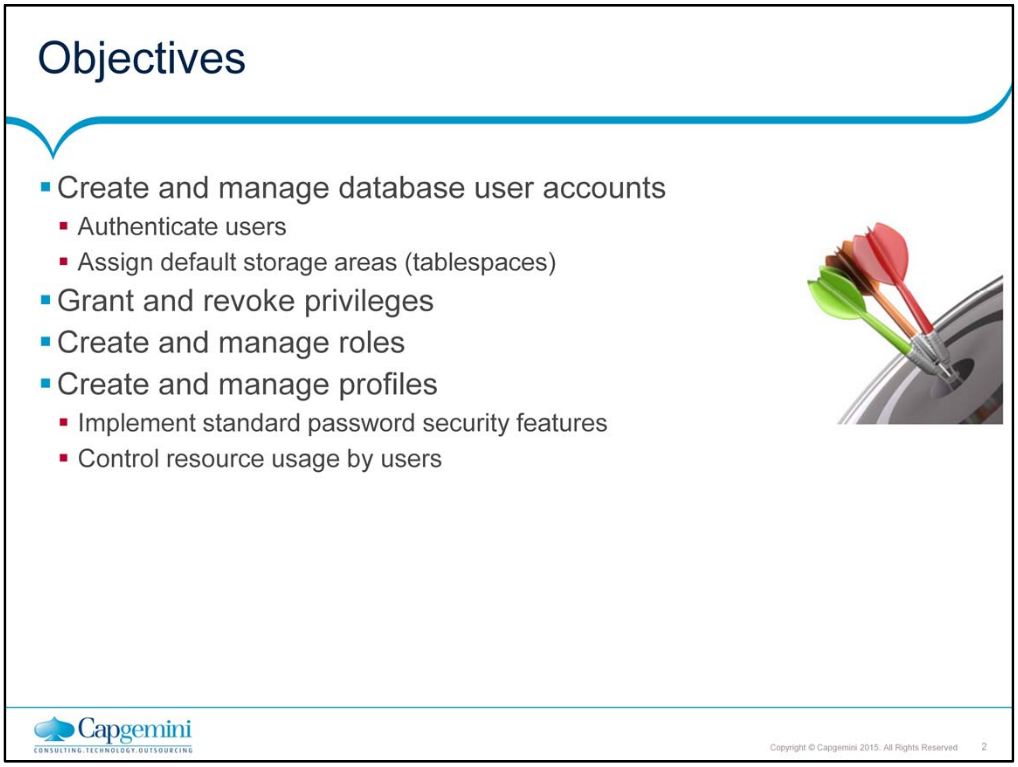


Oracle 11g DBA Fundamentals Overview

Lesson 11: Managing Users and Securing the Database



Objectives

- Create and manage database user accounts
  - Authenticate users
  - Assign default storage areas (tablespaces)
- Grant and revoke privileges
- Create and manage roles
- Create and manage profiles
  - Implement standard password security features
  - Control resource usage by users

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

2

Objectives

The following terms relate to administering database users and assist you in understanding the objectives:

A database user account is a means to organize the ownership of and access to database objects.

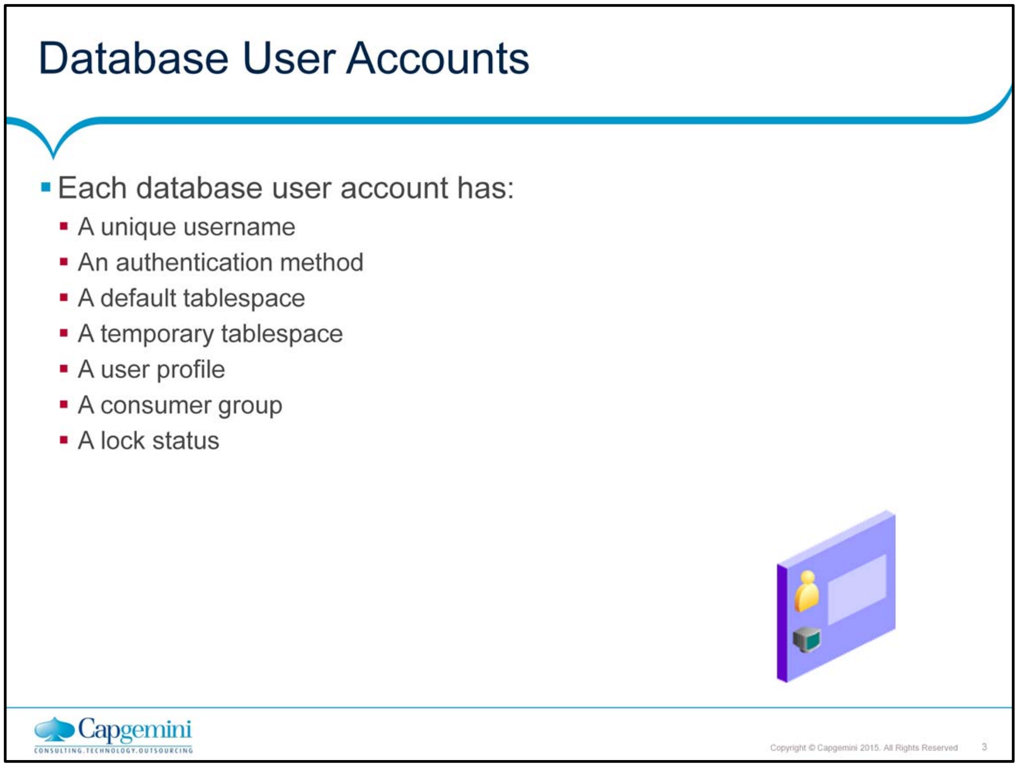
A password is an authentication by the Oracle database.

A privilege is a right to execute a particular type of SQL statement or to access another user’s object.

A role is a named group of related privileges that are granted to users or to other roles.

Profiles impose a named set of resource limits on database usage and instance resources.

Quota is a space allowance in a given tablespace. This is one of the ways by which you can control resource usage by users.



Database User Accounts

- Each database user account has:
  - A unique username
  - An authentication method
  - A default tablespace
  - A temporary tablespace
  - A user profile
  - A consumer group
  - A lock status

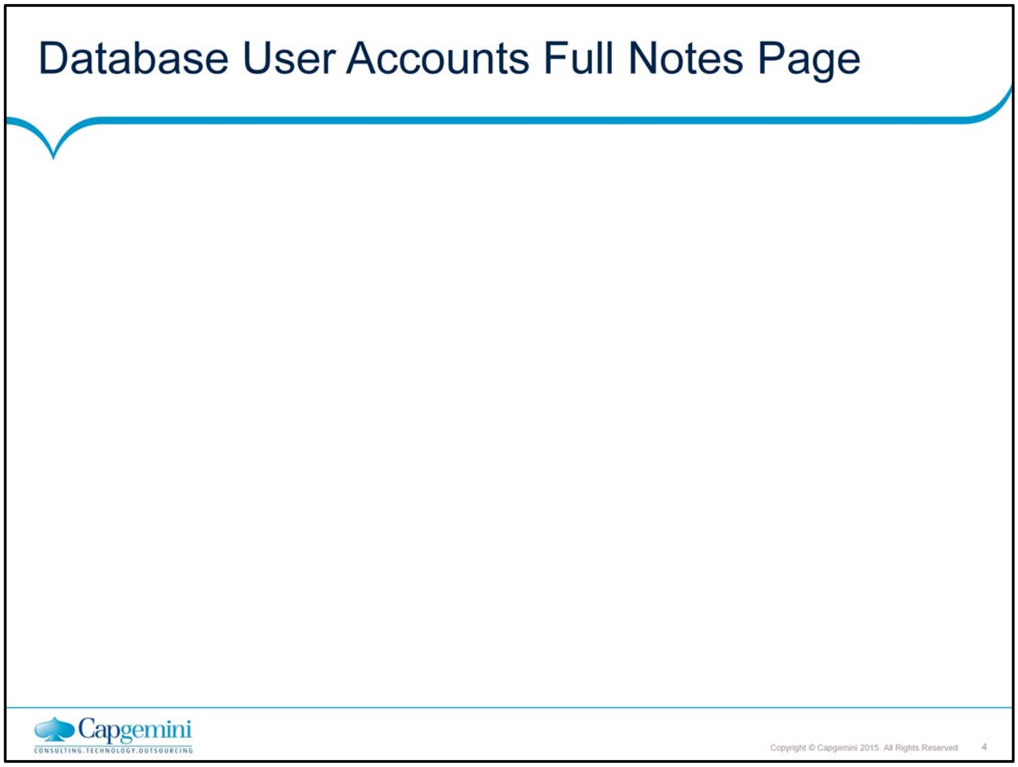
Database User Accounts

To access the database, a user must specify a valid database user account and successfully authenticate as required by that user account. Each database user has his or her own database account. This is Oracle’s best practice recommendation to avoid potential security holes and provide meaningful data for certain audit activities. However, in rare cases, users share a common database account. In this case, operating system and applications must provide adequate security for the database. Each user account has:

A unique username: Usernames cannot exceed 30 bytes, cannot contain special characters, and must start with a letter.

An authentication method: The most common authentication method is a password, but Oracle Database 11g supports several other authentication methods, including biometric, certificate, and token authentication.

A default tablespace: This is a place where the user creates objects if he or she does not specify some other tablespace. Note that having a default tablespace does not imply that the user has the privilege of creating objects in that tablespace, nor does the user have a quota of space within that tablespace in which to create objects. Both these are granted separately.



Database User Accounts Full Notes Page

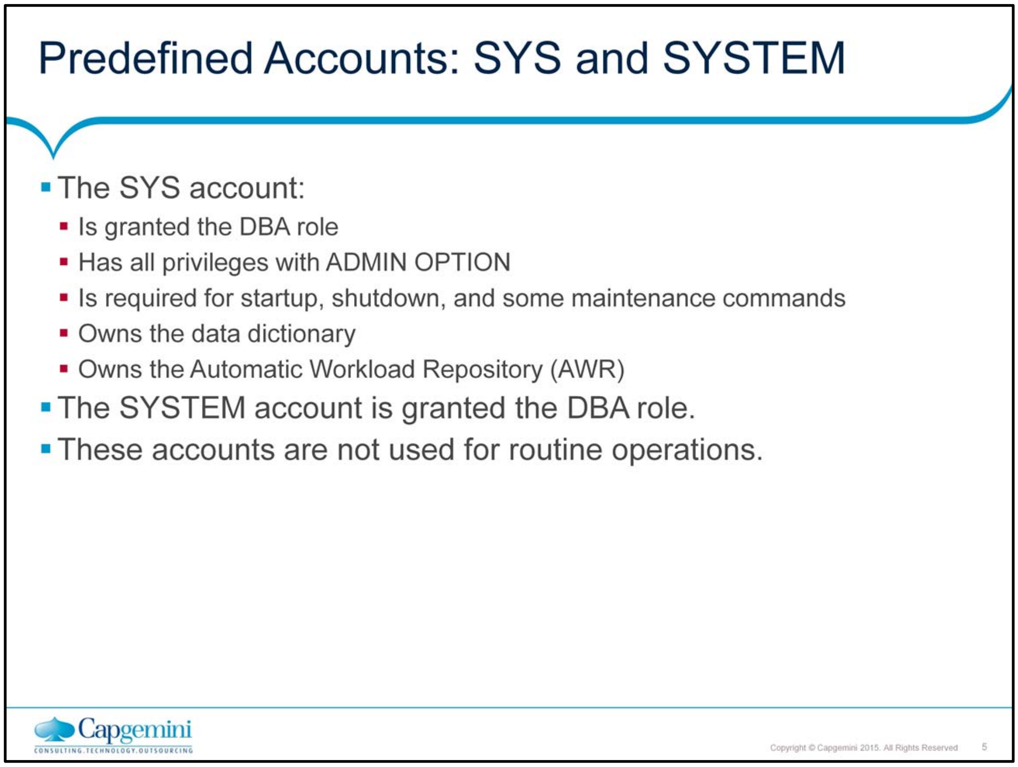
Database User Accounts (continued)

A temporary tablespace: This is a place where the user can create temporary objects, such as sorts and temporary tables.

A user profile: This is a set of resource and password restrictions assigned to the user.

A consumer group: This is used by the resource manager.

A lock status: Users can access only “unlocked” accounts.



Predefined Accounts: SYS and SYSTEM

- The SYS account:
  - Is granted the DBA role
  - Has all privileges with ADMIN OPTION
  - Is required for startup, shutdown, and some maintenance commands
  - Owns the data dictionary
  - Owns the Automatic Workload Repository (AWR)
- The SYSTEM account is granted the DBA role.
- These accounts are not used for routine operations.

Predefined Accounts: SYS and SYSTEM

The SYS and SYSTEM accounts have the database administrator (DBA) role granted to them by default.

The SYS account in addition has all privileges with ADMIN OPTION and owns the data dictionary. To connect to the SYS account, you must use the AS SYSDBA clause. Any user that is granted the SYSDBA privilege can connect to the SYS account by using the AS SYSDBA clause. Only “privileged” users, who are granted the SYSDBA or SYSOPER privilege, are allowed to start up and shut down the database instance.

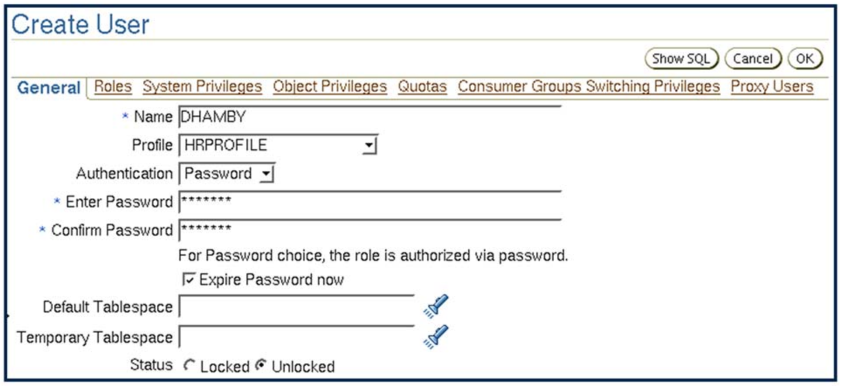
The SYSTEM account is granted the DBA role by default, but not the SYSDBA privilege.

Best practice tip: Applying the principle of least privilege, these accounts are not used for routine operations. Users who need DBA privileges have separate accounts with the required privileges granted to them. For example, Jim has a low privilege account called jim and a privileged account called jim_dba. This method allows the principle of least privilege to be applied, eliminates the need for account sharing, and allows individual actions to be audited.

The SYS and SYSTEM accounts are required accounts in the database. They cannot be dropped.

Creating a User

- Select Administration > Schema > Users & Privileges > Users, and then click the Create button.



Create User

* Name DHAMBY
Profile HRPROFILE
Authentication Password
* Enter Password *******
* Confirm Password *******
For Password choice, the role is authorized via password.
☑ Expire Password now
Default Tablespace
Temporary Tablespace
Status ○ Locked ● Unlocked

Creating a User

In Enterprise Manager, you can manage the list of database users, who are allowed to access the current database, by using the Users page. You can use this page to create, delete, and modify the settings of a user.

To create a database user, perform the following steps:

1. In Enterprise Manager Database Control, select Administration > Schema > Users & Privileges > Users.
2. Click the Create button.

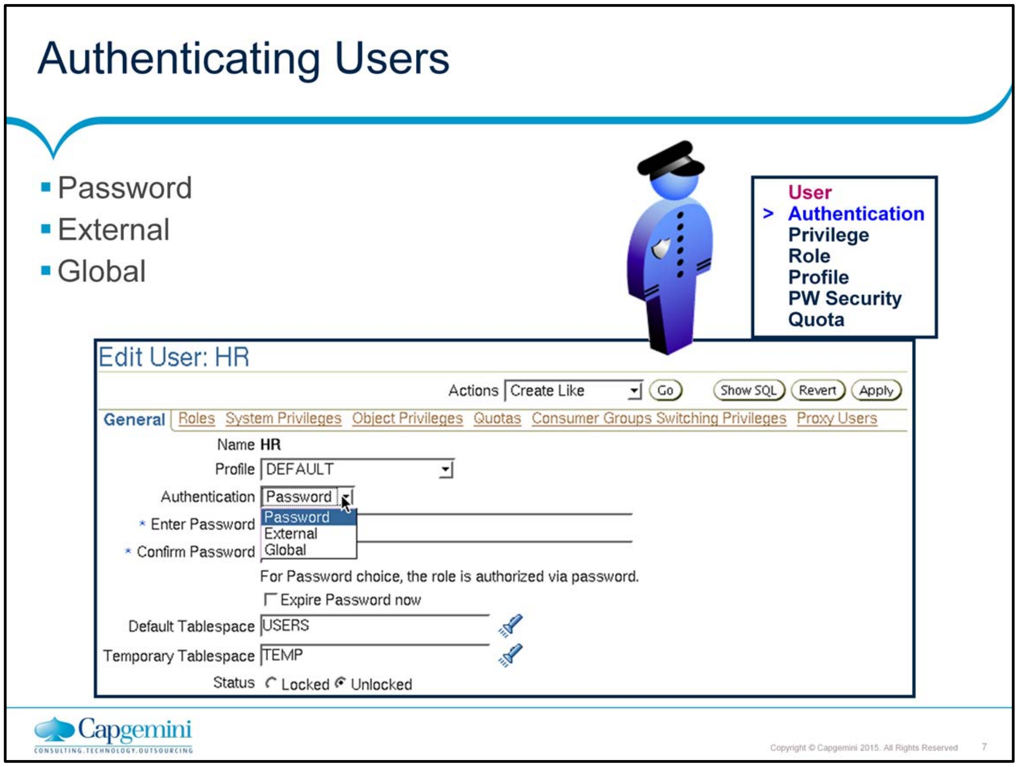
Provide the required information. Mandatory items, such as Name, are marked with a star.

The following pages give you more information about authentication.

Profiles are covered later in this lesson.

Assign a default tablespace and a temporary tablespace to each user. This allows you to control where their objects are created, if users do not specify a tablespace in the creation of an object.

If you do not choose a default tablespace, then the system-defined default permanent tablespace is used. Similarly for the temporary tablespace: if you do not specify one, then the system-defined temporary tablespace is used.



Authenticating Users

- Password
- External
- Global

User
> Authentication
Privilege
Role
Profile
PW Security
Quota

Edit User: HR

Actions Create Like Go
Show SQL Revert Apply
General Roles System Privileges Object Privileges Quotas Consumer Groups Switching Privileges Proxy Users

Name HR

Profile DEFAULT

Authentication Password

* Enter Password

* Confirm Password

For Password choice, the role is authorized via password.

□ Expire Password now

Default Tablespace USERS

Temporary Tablespace TEMP

Status ○ Locked ● Unlocked

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Authenticating Users

Authentication means verifying the identity of someone (a user, device, or other entity) who wants to use data, resources, or applications. Validating that identity establishes a trust relationship for further interactions. Authentication also enables accountability by making it possible to link access and actions to specific identities. After authentication, authorization processes can allow or limit the levels of access and action permitted to that entity.

When you create a user, you must decide on the authentication technique to use, which can be modified later.

Password: This is also referred to as authentication by the Oracle database. Create each user with an associated password that must be supplied when the user attempts to establish a connection. When setting up a password, you can expire the password immediately, which forces the user to change the password after first logging in. If you decide on expiring user passwords, make sure that users have the ability to change the password. Some applications do not have this functionality.

Passwords are always automatically and transparently encrypted during network (client/server and server/server) connections, by using a modified Data Encryption Standard (DES) algorithm, before sending them across the network.

Authenticating Users (continued)

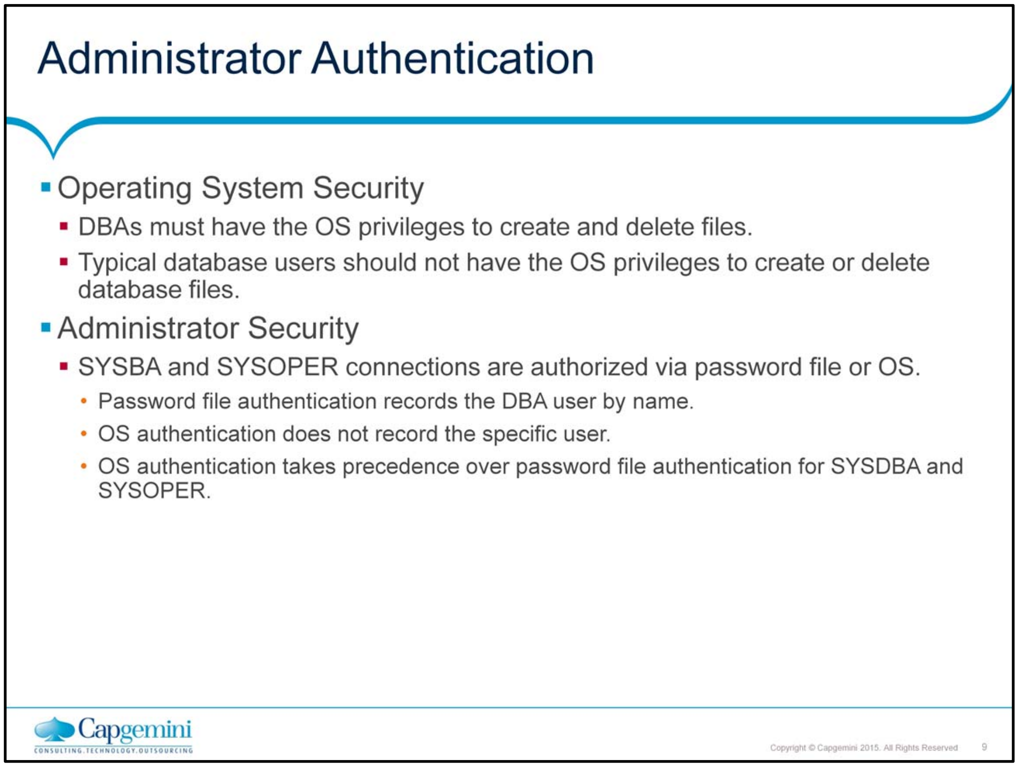
External: This is also referred to as authentication by the operating system. Users can connect to the Oracle database without specifying a username or password. With external authentication, your database relies on the underlying operating system or network authentication service to restrict access to database accounts. A database password is not used for this type of login. If your operating system or network service permits, you can have it authenticate users. If you do so, set the OS_AUTHENT_PREFIX initialization parameter and use this prefix in Oracle usernames. The OS_AUTHENT_PREFIX parameter defines a prefix that the Oracle database adds to the beginning of each user's operating system account name. The default value of this parameter is OPS$ for backward compatibility with the previous versions of the Oracle software. The Oracle database compares the prefixed username with the Oracle usernames in the database when a user attempts to connect. For example, assume that OS_AUTHENT_PREFIX is set as follows:

```
OS_AUTHENT_PREFIX=OPS$
```

If a user with an operating system account named tsmith needs to connect to an Oracle database and be authenticated by the operating system, then the Oracle database checks whether there is a corresponding database user OPS$tsmith and, if so, allows the user to connect. All references to a user who is authenticated by the operating system must include the prefix, as seen in OPS$tsmith.

Note: The text of the OS_AUTHENT_PREFIX initialization parameter is case sensitive on some operating systems. See your operating system–specific Oracle documentation for more information about this initialization parameter.

Global: Using the Oracle Advanced Security option, global authentication (which is a strong authentication) allows users to be identified through the use of biometrics, x509 certificates, token devices, and Oracle Internet Directory. For more information about advanced authentication methods, refer to the *Oracle Enterprise Identity Management* course.



Administrator Authentication

- Operating System Security
  - DBAs must have the OS privileges to create and delete files.
  - Typical database users should not have the OS privileges to create or delete database files.
- Administrator Security
  - SYSBA and SYSOPER connections are authorized via password file or OS.
    - Password file authentication records the DBA user by name.
    - OS authentication does not record the specific user.
    - OS authentication takes precedence over password file authentication for SYSDBA and SYSOPER.

Administrator Authentication

Operating System Security: In UNIX and Linux, by default, DBAs belong to the install OS group, which has the required privileges to create and delete database files.

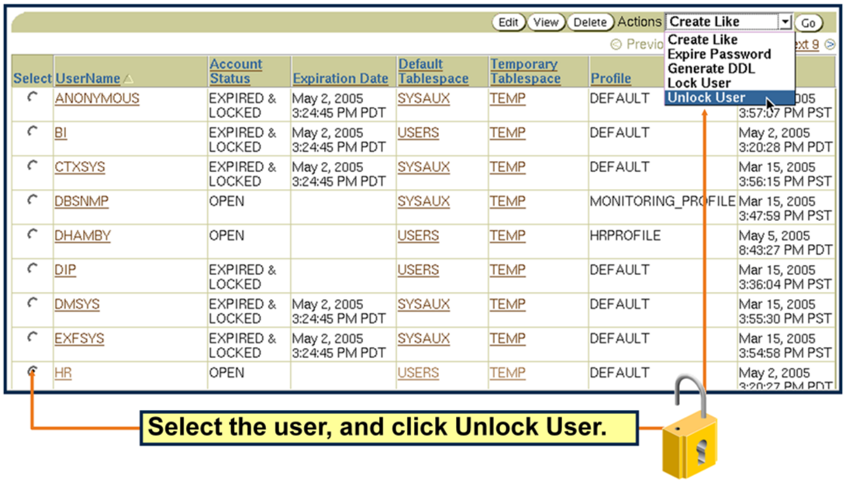
Administrator Security: SYSBA and SYSOPER connections are authorized only after verification with the password file or with the operating system privileges and permissions. If operating system authentication is used, then the database does not use the supplied username and password.

Operating system authentication is used if there is no password file, if the supplied username or password is not in that file, or if no username and password is supplied.

However, if authentication succeeds by means of the password file, then the connection is logged with the username. If authentication succeeds by means of the operating system, then it is a CONNECT / connection that does not record the specific user.

Note: OS authentication takes precedence over password file authentication. Specifically, if you are a member of the OSDBA or OSOPER group for the operating system, and you connect as SYSDBA or SYSOPER, you will be connected with the associated administrative privileges regardless of the username and password that you specify.

Unlocking a User Account and Resetting the Password



Select	UserName △	Account Status	Expiration Date	Default Tablespace	Temporary Tablespace	Profile	
○	ANONYMOUS	EXPIRED & LOCKED	May 2, 2005 3:24:45 PM PDT	SYSAUX	TEMP	DEFAULT	005 3:57:07 PM PST
○	BI	EXPIRED & LOCKED	May 2, 2005 3:24:45 PM PDT	USERS	TEMP	DEFAULT	May 2, 2005 3:20:28 PM PDT
○	CTXSYS	EXPIRED & LOCKED	May 2, 2005 3:24:45 PM PDT	SYSAUX	TEMP	DEFAULT	Mar 15, 2005 3:56:15 PM PST
○	DBSNMP	OPEN		SYSAUX	TEMP	MONITORING_PROFILE	Mar 15, 2005 3:47:59 PM PST
○	DHAMBY	OPEN		USERS	TEMP	HRPROFILE	May 5, 2005 8:43:27 PM PDT
○	DIP	EXPIRED & LOCKED		USERS	TEMP	DEFAULT	Mar 15, 2005 3:36:04 PM PST
○	DMSYS	EXPIRED & LOCKED	May 2, 2005 3:24:45 PM PDT	SYSAUX	TEMP	DEFAULT	Mar 15, 2005 3:55:30 PM PST
○	EXFSYS	EXPIRED & LOCKED	May 2, 2005 3:24:45 PM PDT	SYSAUX	TEMP	DEFAULT	Mar 15, 2005 3:54:58 PM PST
●	HR	OPEN		USERS	TEMP	DEFAULT	May 2, 2005 3:20:27 PM PDT

Select the user, and click Unlock User.

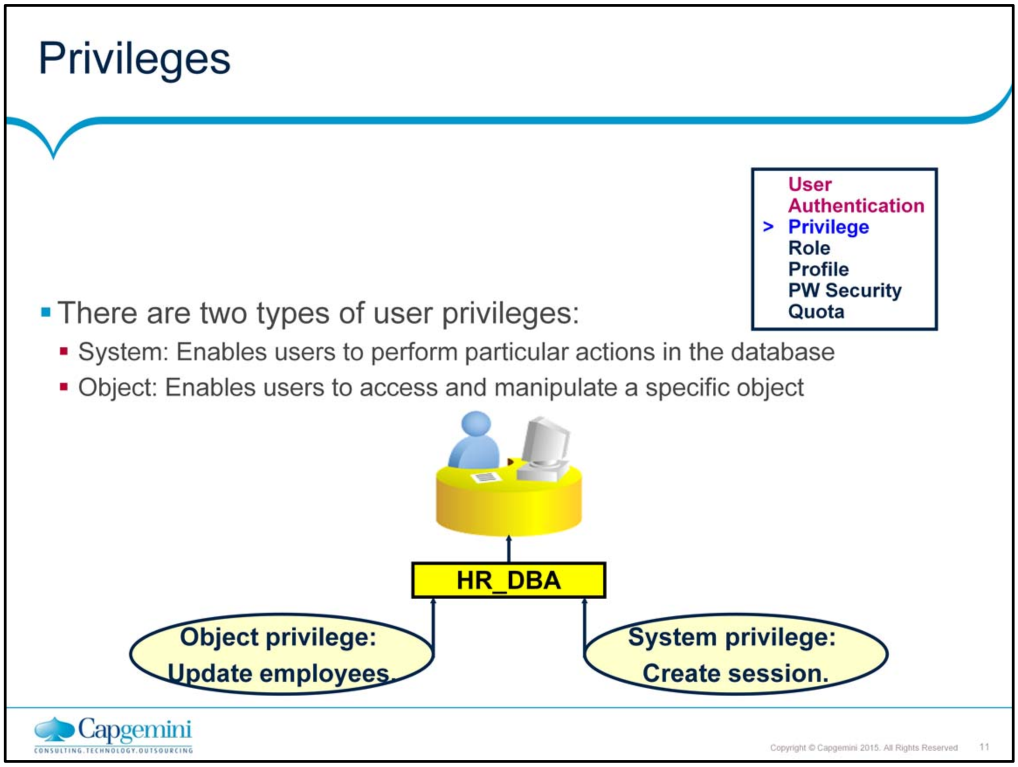
10

Unlocking a User Account and Resetting the Password

During installation and database creation, you can unlock and reset many of the Oracle-supplied database user accounts. If you have not chosen to unlock the user accounts at that time, you can unlock the users and reset the passwords by selecting the user on the Users page and clicking Unlock User.

Alternatively, if you are on the Edit Users page, perform the following steps:

1. Enter the new password in the Enter Password and Confirm Password fields.
2. Select the Unlocked check box.
3. Click Apply to reset the password and unlock the user account.



Privileges

User
Authentication
> Privilege
Role
Profile
PW Security
Quota

- There are two types of user privileges:
  - System: Enables users to perform particular actions in the database
  - Object: Enables users to access and manipulate a specific object

HR_DBA

Object privilege:
Update employees.

System privilege:
Create session.

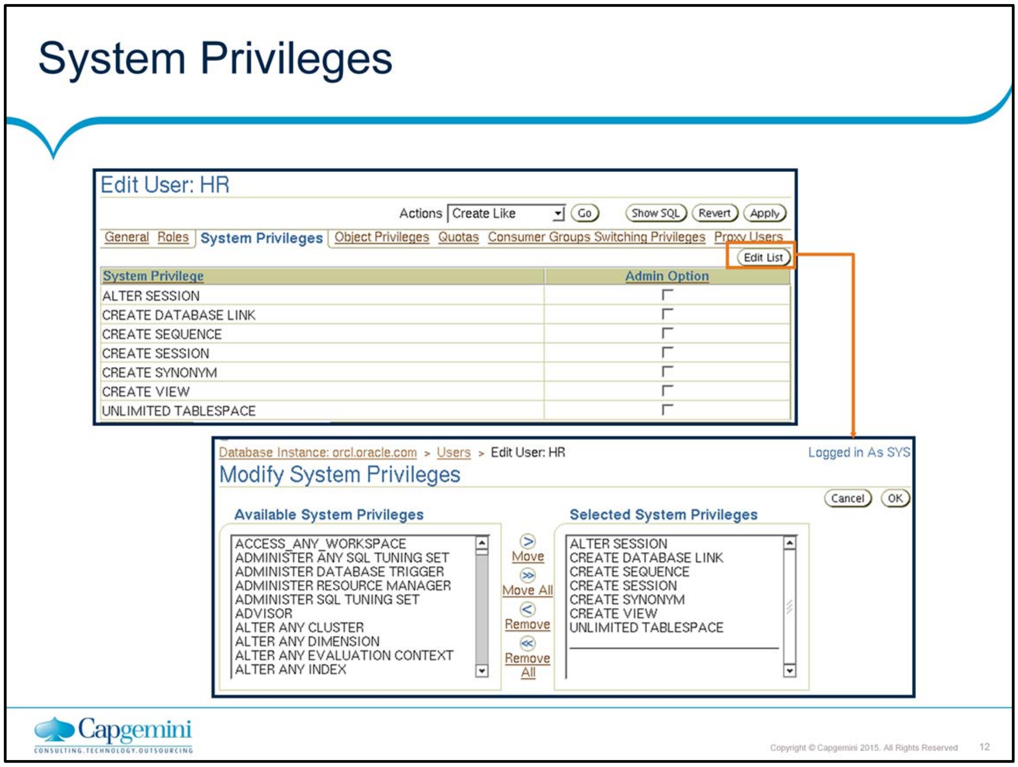
11

Privileges

A privilege is a right to execute a particular type of SQL statement or to access another user’s object. The Oracle database enables you to control what users can or cannot do within the database. Privileges are divided into two categories:

System privileges: Each system privilege allows a user to perform a particular database operation or class of database operations. For example, the privilege to create tablespaces is a system privilege. System privileges can be granted by the administrator or by someone who explicitly gives permission to administer the privilege. There are more than a hundred distinct system privileges. Many system privileges contain the ANY clause.

Object privileges: Object privileges allow a user to perform a particular action on a specific object, such as a table, view, sequence, procedure, function, or package. Without specific permission, users can access only their own objects. Object privileges can be granted by the owner of an object, by the administrator, or by someone who has been explicitly given permission to grant privileges on the object.



System Privileges

Edit User: HR

System Privilege	Admin Option
ALTER SESSION	☐
CREATE DATABASE LINK	☐
CREATE SEQUENCE	☐
CREATE SESSION	☐
CREATE SYNONYM	☐
CREATE VIEW	☐
UNLIMITED TABLESPACE	☐

Database Instance: orcl.oracle.com > Users > Edit User: HR

Logged in As SYS

Modify System Privileges

Available System Privileges

ACCESS_ANY_WORKSPACE
ADMINISTER ANY SQL TUNING SET
ADMINISTER DATABASE TRIGGER
ADMINISTER RESOURCE MANAGER
ADMINISTER SQL TUNING SET
ADVISOR
ALTER ANY CLUSTER
ALTER ANY DIMENSION
ALTER ANY EVALUATION CONTEXT
ALTER ANY INDEX

Selected System Privileges

ALTER SESSION
CREATE DATABASE LINK
CREATE SEQUENCE
CREATE SESSION
CREATE SYNONYM
CREATE VIEW
UNLIMITED TABLESPACE

System Privileges

To grant system privileges, click the Systems Privileges tab on the Edit User page. Select the appropriate privileges from the list of available privileges, and move them to the Selected System Privileges list by clicking the Move arrow.

Granting a privilege with the ANY clause means that the privilege crosses schema lines. For example, the CREATE TABLE privilege allows you to create a table but only within your own schema. The SELECT ANY TABLE privilege allows you to select from tables owned by other users.

Selecting the Admin Option check box enables you to administer the privilege and grant the system privilege to other users.

Carefully consider security requirements before granting system permissions. Some system privileges are usually granted only to administrators:

RESTRICTED SESSION: This privilege allows you to log in even if the database has been opened in restricted mode.

System Privileges (continued)

SYSDBA and SYSOPER: These privileges allow you to shut down, start up, and perform recovery and other administrative tasks in the database. SYSOPER allows a user to perform basic operational tasks, but without the ability to look at user data. It includes the following system privileges:

STARTUP and SHUTDOWN
CREATE SPFILE
ALTER DATABASE OPEN/MOUNT/BACKUP
ALTER DATABASE ARCHIVELOG
ALTER DATABASE RECOVER (Complete recovery only. Any form of incomplete recovery, such as UNTIL TIME|CHANGE|CANCEL|CONTROLFILE requires connecting as SYSDBA.)
RESTRICTED SESSION

The SYSDBA system privilege additionally authorizes incomplete recovery and the deletion of a database. Effectively, the SYSDBA system privilege allows a user to connect as the SYS user.

DROP ANY *object*: The DROP ANY privilege allows you to delete objects that other schema users own.

CREATE, MANAGE, DROP, and ALTER TABLESPACE: These privileges allow for tablespace administration including creating, dropping, and changing their attributes.

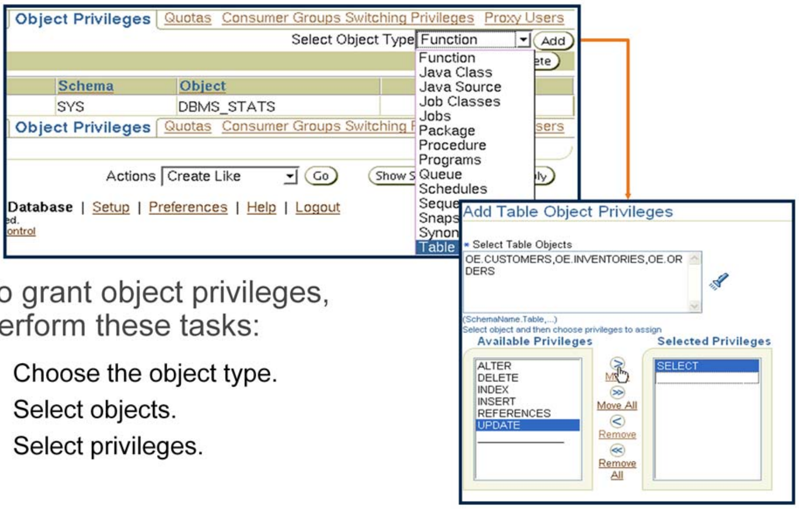
CREATE ANY DIRECTORY: The Oracle database allows developers to call external code (for example, a C library) from within PL/SQL. As a security measure, the operating system directory where the code resides must be linked to a virtual Oracle directory object. With the CREATE ANY DIRECTORY privilege, you can potentially call insecure code objects.

The CREATE ANY DIRECTORY privilege allows a user to create a directory object (with read and write access) to any directory that the Oracle software owner can access. This means that the user can access external procedures in those directories. The user can attempt to directly read and write any database file, such as data files, redo log, and audit logs. Ensure that your organization has a security strategy that prevents misuse of powerful privileges such as this one.

GRANT ANY OBJECT PRIVILEGE: This privilege allows you to grant object permissions on objects that you do not own.

ALTER DATABASE and ALTER SYSTEM: These very powerful privileges allow you to modify the database and the Oracle instance, such as renaming a data file or flushing the buffer cache.

Object Privileges



Select Object Type: Function

Schema	Object
SYS	DBMS_STATS

Actions: Create Like

Add Table Object Privileges

* Select Table Objects: OE.CUSTOMERS,OE.INVENTORIES,OE.ORDERS

(SchemaName.Table,...)

Select object and then choose privileges to assign

Available Privileges: ALTER, DELETE, INDEX, INSERT, REFERENCES, UPDATE

Selected Privileges: SELECT

- To grant object privileges, perform these tasks:

1. Choose the object type.
2. Select objects.
3. Select privileges.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

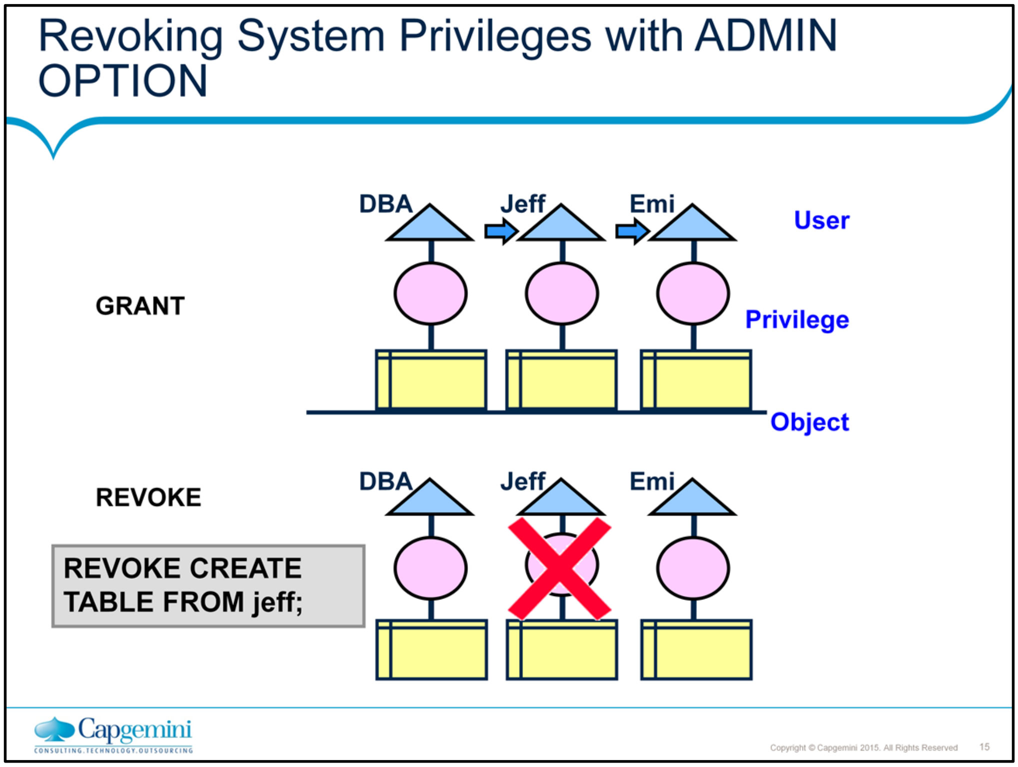
14

Object Privileges

To grant object privileges, click the Object Privileges tab on the Edit User page. Select the type of object you want to grant privileges on, and click the Add button. Choose the objects you want to grant privileges on by either entering <username.object name> or selecting them from the list.

Next, select the appropriate privileges from the Available Privileges list, and click the Move button. When you have finished selecting privileges, click OK.

Back on the Edit User page, select the Grant check box if this user is allowed to grant other users the same access.



Revoking System Privileges with ADMIN OPTION

```
REVOKE CREATE
TABLE FROM jeff;
```

Revoking System Privileges

System privileges, which have been granted directly with a GRANT command, can be revoked by using the REVOKE SQL statement. Users with ADMIN OPTION for a system privilege can revoke the privilege from any other database user. The revoker does not have to be the same user who originally granted the privilege.

There are no cascading effects when a system privilege is revoked, regardless of whether it is given the ADMIN OPTION.

Read through the following steps that illustrate this:

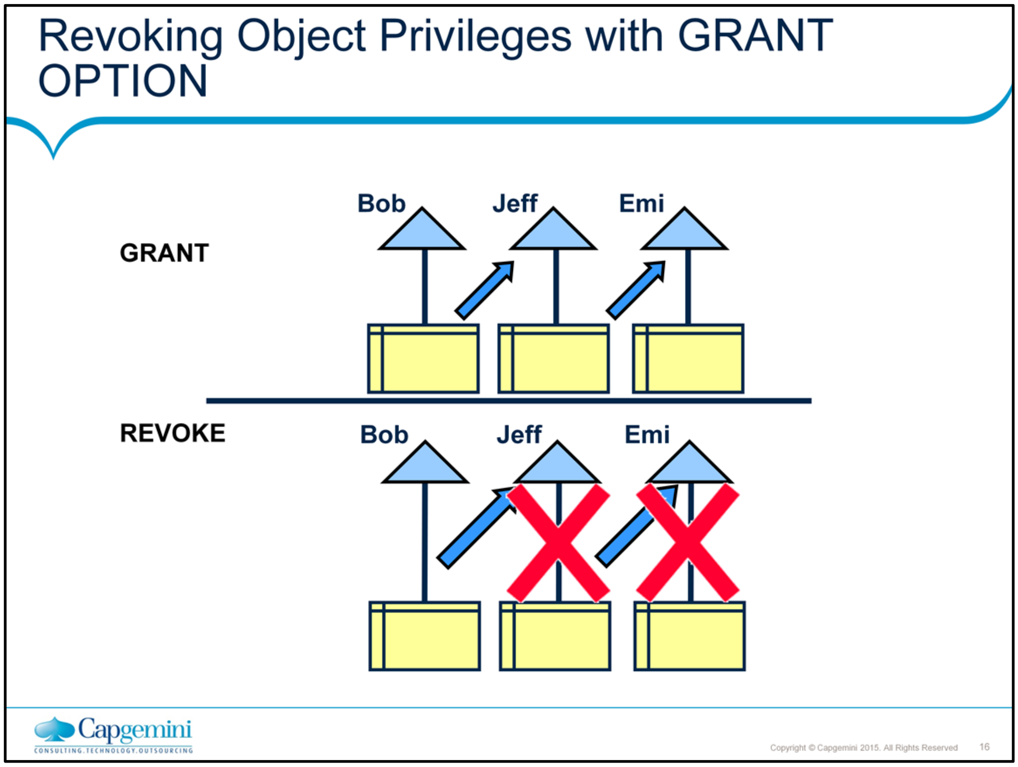
Scenario

1. The DBA grants the CREATE TABLE system privilege to Jeff with ADMIN OPTION.
2. Jeff creates a table.
3. Jeff grants the CREATE TABLE system privilege to Emi.
4. Emi creates a table.
5. The DBA revokes the CREATE TABLE system privilege from Jeff.

The result

Jeff’s table still exists, but no new tables can be created.

Emi’s table still exists, and she still has the CREATE TABLE system privilege.



Revoking Object Privileges with GRANT OPTION

Revoking Object Privileges

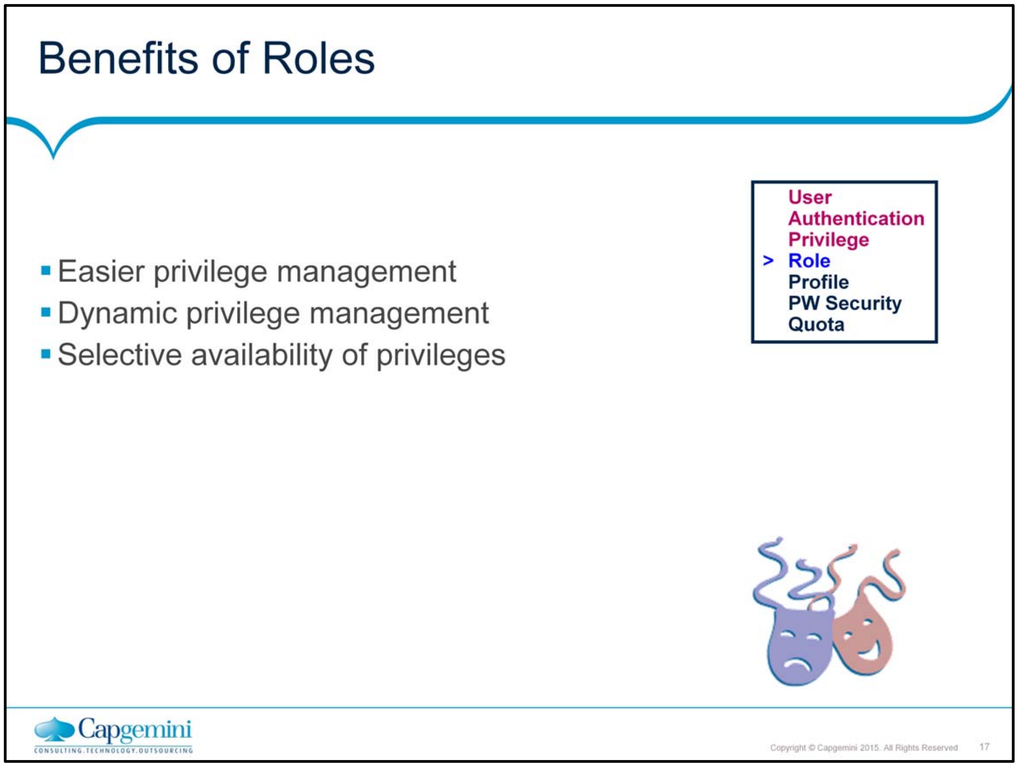
Cascading effects can be observed when revoking a system privilege that is related to a data manipulation language (DML) operation. For example, if the SELECT ANY TABLE privilege is granted to a user, and that user has created procedures that use the table, all procedures that are contained in the user’s schema must be recompiled before they can be used again.

Revoking object privileges also cascades when given WITH GRANT OPTION.

Read through the following steps that illustrate this:

Scenario

1. Jeff is granted the SELECT object privilege on EMPLOYEES with GRANT OPTION.
2. Jeff grants the SELECT privilege on EMPLOYEES to Emi.
3. Later, the SELECT privilege is revoked from Jeff. This revoke is cascaded to Emi as well.



Benefits of Roles

- Easier privilege management
- Dynamic privilege management
- Selective availability of privileges

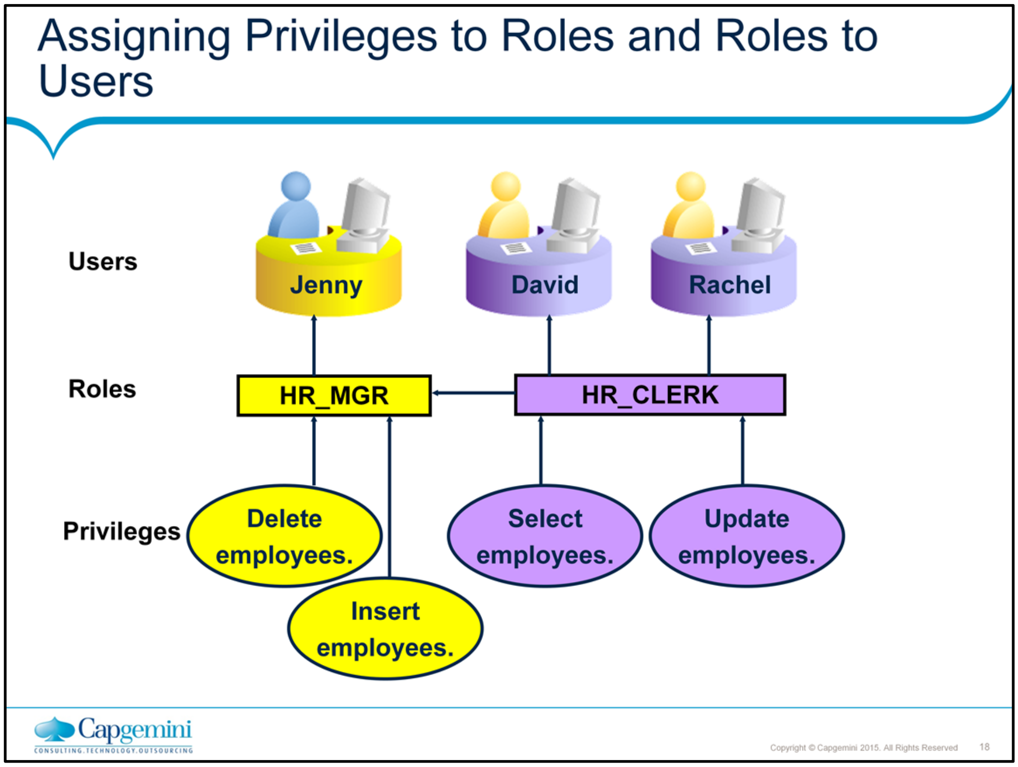
User
Authentication
Privilege
> Role
Profile
PW Security
Quota

Benefits of Roles

Easier privilege management: Use roles to simplify privilege management. Rather than granting the same set of privileges to several users, you can grant the privileges to a role, and then grant that role to each user.

Dynamic privilege management: If the privileges associated with a role are modified, all the users who are granted the role acquire the modified privileges automatically and immediately.

Selective availability of privileges: Roles can be enabled and disabled to turn privileges on and off temporarily. Enabling a role can also be used to verify that a user has been granted that role.



Assigning Privileges to Roles and Roles to Users

Users

Jenny

David

Rachel

Roles

HR_MGR

HR_CLERK

Privileges

Delete employees.

Insert employees.

Select employees.

Update employees.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Assigning Privileges to Roles and Roles to Users

In most systems, it is too time-consuming to grant necessary privileges to each user individually, and there is too great a chance of error. The Oracle software provides for easy and controlled privilege management through roles. Roles are named groups of related privileges that are granted to users or to other roles. Roles are designed to ease the administration of privileges in the database and, therefore, improve security.

Role characteristics

Privileges are granted to and revoked from roles as though the role were a user.

Roles can be granted to and revoked from users or other roles as though they were system privileges.

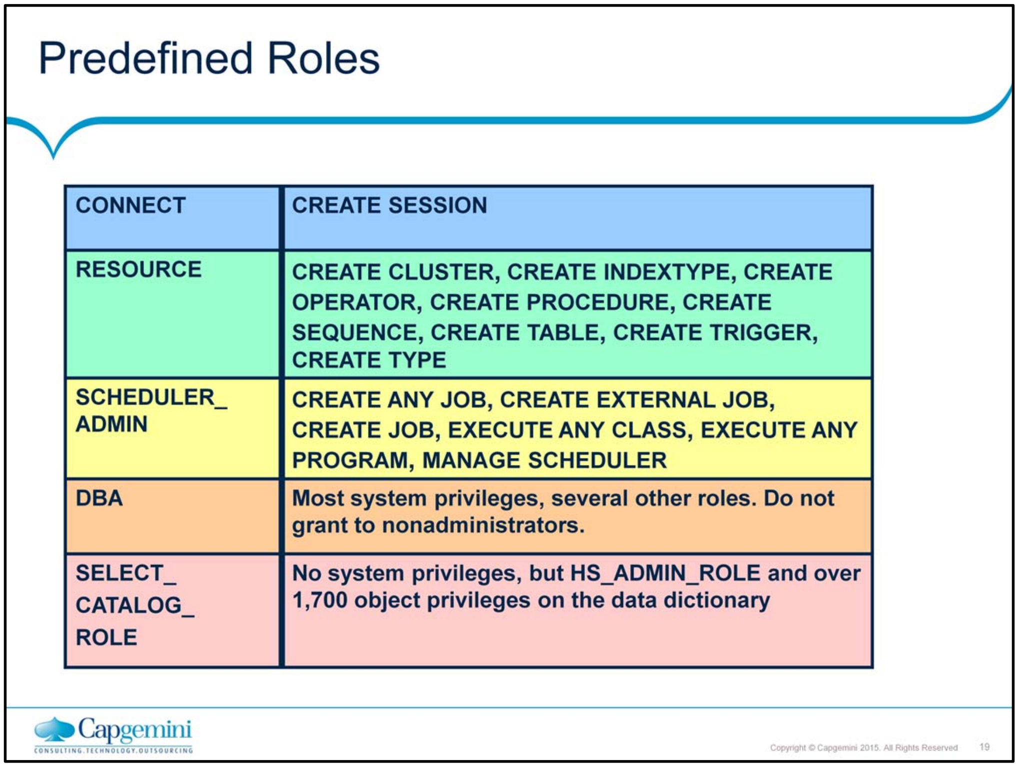
A role can consist of both system and object privileges.

A role can be enabled or disabled for each user who is granted the role.

A role can require a password to be enabled.

Roles are not owned by anyone; and they are not in any schema.

In the slide example, the HR_CLERK role is granted the SELECT and UPDATE privileges on the employees table. The HR_MGR role is granted the DELETE and INSERT privileges on the employees table and the HR_CLERK role. The manager is granted the HR_MGR role and can now select, delete, insert, and update the employees table.



Predefined Roles

CONNECT	CREATE SESSION
RESOURCE	CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, CREATE TYPE
SCHEDULER_ ADMIN	CREATE ANY JOB, CREATE EXTERNAL JOB, CREATE JOB, EXECUTE ANY CLASS, EXECUTE ANY PROGRAM, MANAGE SCHEDULER
DBA	Most system privileges, several other roles. Do not grant to nonadministrators.
SELECT_ CATALOG_ ROLE	No system privileges, but HS_ADMIN_ROLE and over 1,700 object privileges on the data dictionary

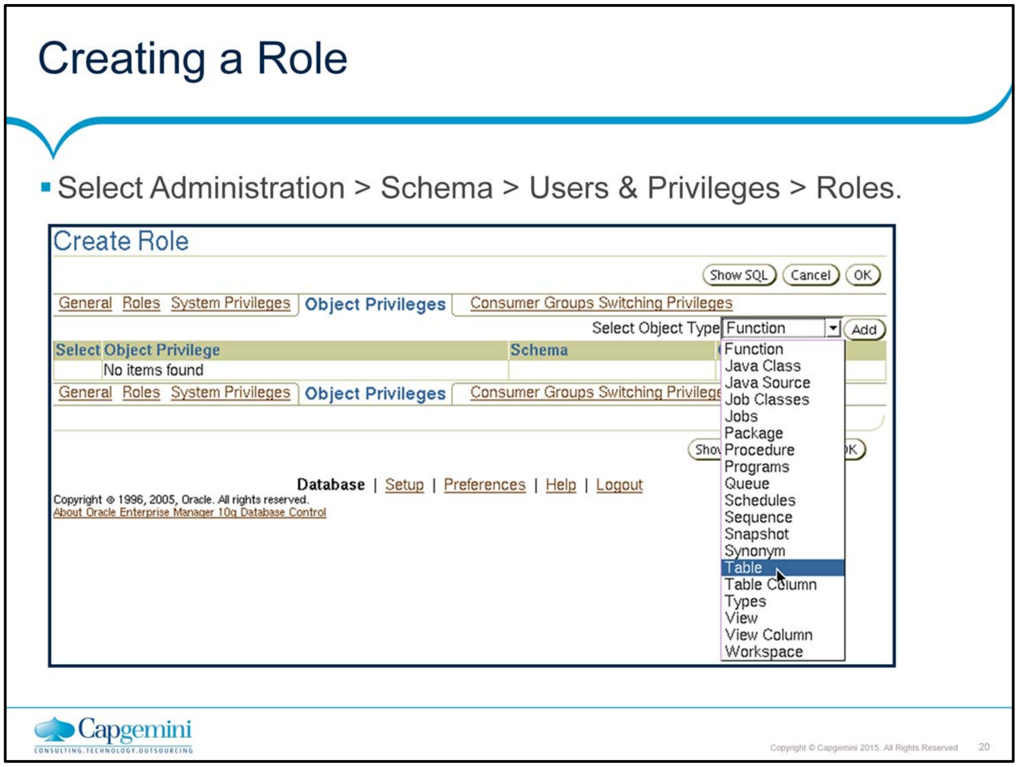
Predefined Roles

There are several roles that are defined automatically for Oracle databases when you run database creation scripts. CONNECT is granted automatically to any user created with Enterprise Manager. In earlier versions of the database (before Oracle Database 11g Release 2), the CONNECT role included more privileges, such as CREATE TABLE and CREATE DATABASE LINK, which have been removed for security reasons.

Note: Be aware that granting the RESOURCE role includes granting the UNLIMITED TABLESPACE privilege.

Functional Roles

Other roles that authorize you to administer special functions are created when that functionality is installed. For example, XDBADMIN contains the privileges required to administer the Extensible Markup Language (XML) database if that feature is installed. AQ_ADMINISTRATOR_ROLE provides privileges to administer advanced queuing. HS_ADMIN_ROLE includes the privileges needed to administer heterogeneous services. You must not alter the privileges granted to these functional roles without the assistance of Oracle support because you may inadvertently disable the needed functionality.



Creating a Role

- Select Administration > Schema > Users & Privileges > Roles.

Create Role

Select	Object Privilege	Schema
	No items found	

Copyright © 1996, 2005, Oracle. All rights reserved.
About Oracle Enterprise Manager 10g Database Control

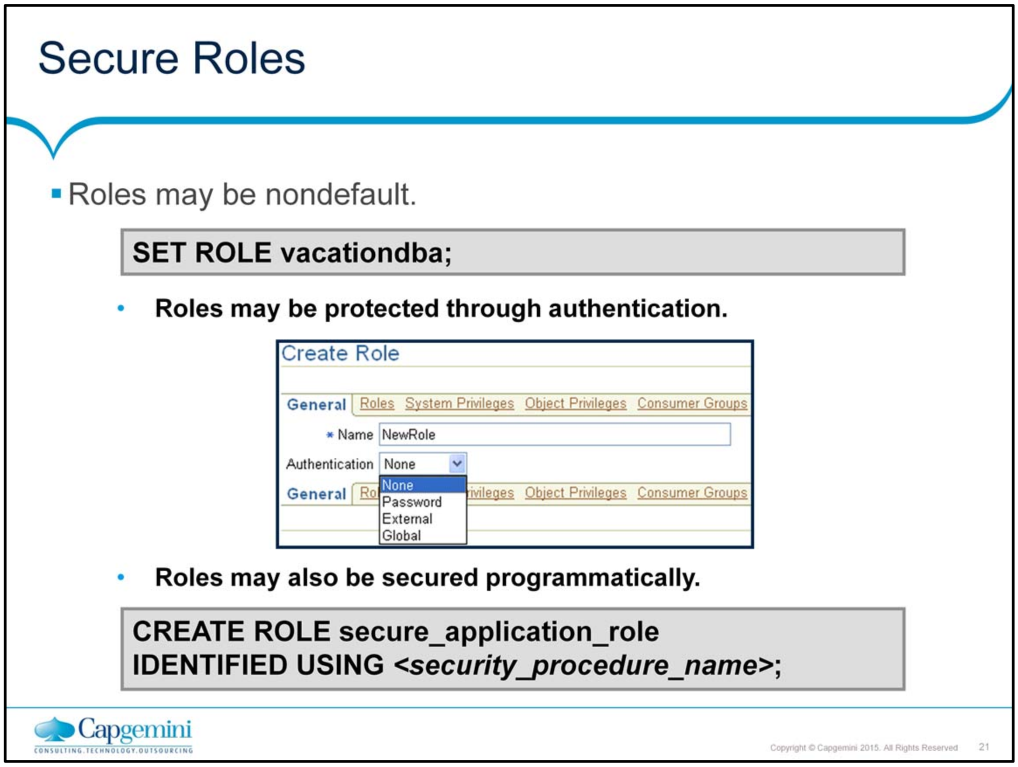
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Creating a Role

A role is a named group of related privileges that are granted to users or to other roles. A DBA manages privileges through roles.

To create a role, perform the following steps:

1. In Enterprise Manager Database Control, select Administration > Schema > Users & Privileges > Roles.
2. Click the Create button.



Secure Roles

- Roles may be nondefault.

```
SET ROLE vacationdba;
```

- **Roles may be protected through authentication.**

Create Role

* Name NewRole

Authentication None

- **Roles may also be secured programmatically.**

```
CREATE ROLE secure_application_role
IDENTIFIED USING <security_procedure_name>;
```

Secure Roles

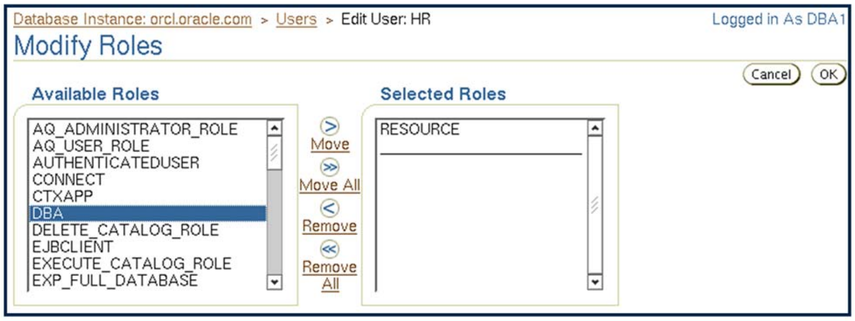
Roles are usually enabled by default, which means that if a role is granted to a user, that user can exercise the privileges given to that role. It is possible to:

Make a role nondefault. When the role is granted to a user, deselect the DEFAULT check box. The user must now explicitly enable the role before the role’s privileges can be exercised.

Have a role require additional authentication. The default authentication for a role is None, but it is possible to have the role require additional authentication before it can be set.

Create secure application roles that can be enabled only by executing a PL/SQL procedure successfully. The PL/SQL procedure can check things such as the user’s network address, which program the user is running, time of day, or other elements needed to properly secure a group of permissions.

Assigning Roles to Users



Database Instance: orcl.oracle.com > Users > Edit User: HR

Logged in As DBA1

Modify Roles

Available Roles:
AQ_ADMINISTRATOR_ROLE
AQ_USER_ROLE
AUTHENTICATEDUSER
CONNECT
CTXAPP
DBA
DELETE_CATALOG_ROLE
EJBCLIENT
EXECUTE_CATALOG_ROLE
EXP_FULL_DATABASE

Selected Roles:
RESOURCE

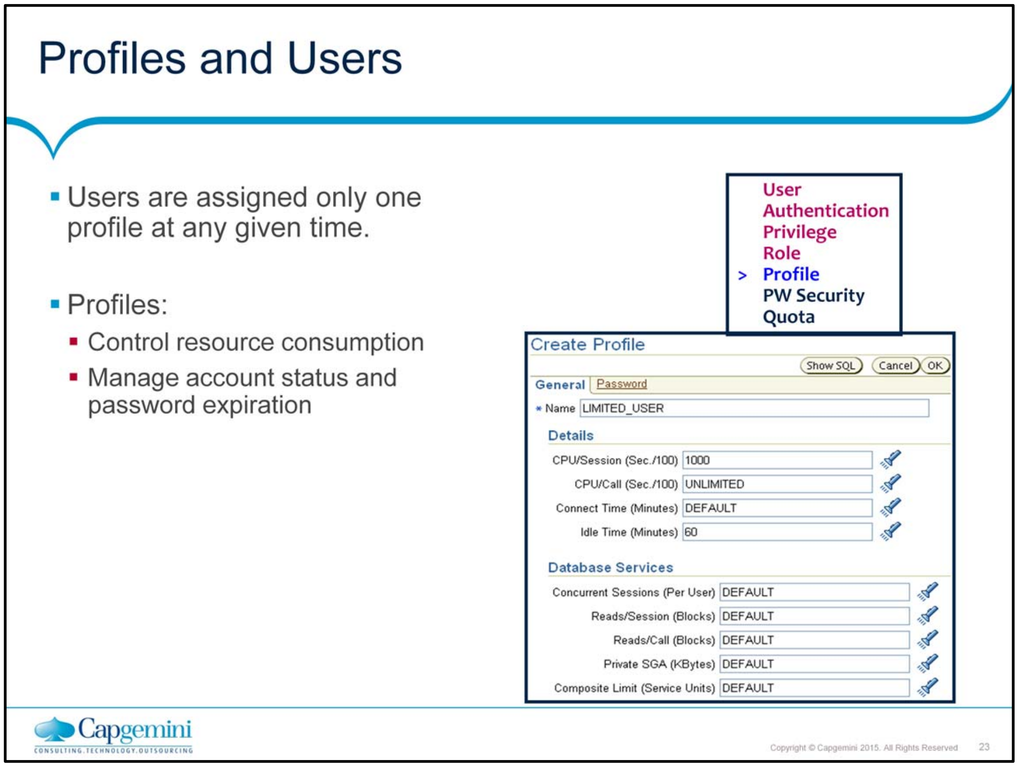
Assigning Roles to Users

A role is a set of privileges that can be granted to users or to other roles. You can use roles to administer database privileges. You can add privileges to a role and then grant the role to a user. The user can then enable the role and exercise the privileges granted by the role. A role contains all privileges granted to that role and all privileges of other roles granted to it.

By default, Enterprise Manager automatically grants the CONNECT role to new users. This allows users to connect to the database and create database objects in their own schemas.

To assign a role to a user, perform the following steps:

1. In Enterprise Manager Database Control, choose Administration > Schema > Users & Privileges > Users.
2. Select the user, and click the Edit button.
3. Click the Roles tab, and then click the Edit List button.
4. Select the desired role under Available Roles and move it under Selected Roles.
5. When you have assigned all appropriate roles, click the OK button.



Profiles and Users

- Users are assigned only one profile at any given time.
- Profiles:
  - Control resource consumption
  - Manage account status and password expiration

Create Profile

* Name	LIMITED_USER

Details

CPU/Session (Sec./100)	1000
CPU/Call (Sec./100)	UNLIMITED
Connect Time (Minutes)	DEFAULT
Idle Time (Minutes)	60

Database Services

Concurrent Sessions (Per User)	DEFAULT
Reads/Session (Blocks)	DEFAULT
Reads/Call (Blocks)	DEFAULT
Private SGA (KBytes)	DEFAULT
Composite Limit (Service Units)	DEFAULT

23

Profiles and Users

Profiles impose a named set of resource limits on database usage and instance resources. Profiles also manage the account status and place limitations on users' passwords (length, expiration time, and so on). Every user is assigned a profile and may belong to only one profile at any given time. If users have already logged in when you change their profile, the change does not take effect until their next login.

The default profile serves as the basis for all other profiles. As illustrated in the slide, limitations for a profile can be implicitly specified (as in CPU/Session), be unlimited (as in CPU/Call), or reference whatever setting is in the default profile (as in Connect Time).

Profiles cannot impose resource limitations on users unless the RESOURCE_LIMIT initialization parameter is set to TRUE. With RESOURCE_LIMIT at its default value of FALSE, profile limitations are ignored.

Profiles enable the administrator to control the following system resources:

CPU: CPU resources may be limited on a per-session or per-call basis. A CPU/Session limitation of 1,000 means that if any individual session that uses this profile consumes more than 10 seconds of CPU time (CPU time limitations are in hundredths of a second.), then that session receives an error and is logged off:

ORA-02392: exceeded session limit on CPU usage, you are being logged off

Profiles and Users (continued)

A per-call limitation does the same thing, but instead of limiting the user's overall session, it prevents any single command from consuming too much CPU. If CPU/Call is limited and the user exceeds the limitation, the command aborts, and the user gets an error message, such as the following:

ORA-02393: exceeded call limit on CPU usage

Network/Memory: Each database session consumes system memory resources and (if the session is from a user who is not local to the server) network resources. You can specify the following:

Connect Time: Indicates for how many minutes a user can be connected before being automatically logged off

Idle Time: Indicates for how many minutes a user's session can remain idle before being automatically logged off. Idle time is calculated for the server process only. It does not take into account application activity. The IDLE_TIME limit is not affected by long-running queries and other operations.

Concurrent Sessions: Indicates how many concurrent sessions can be created by using a database user account

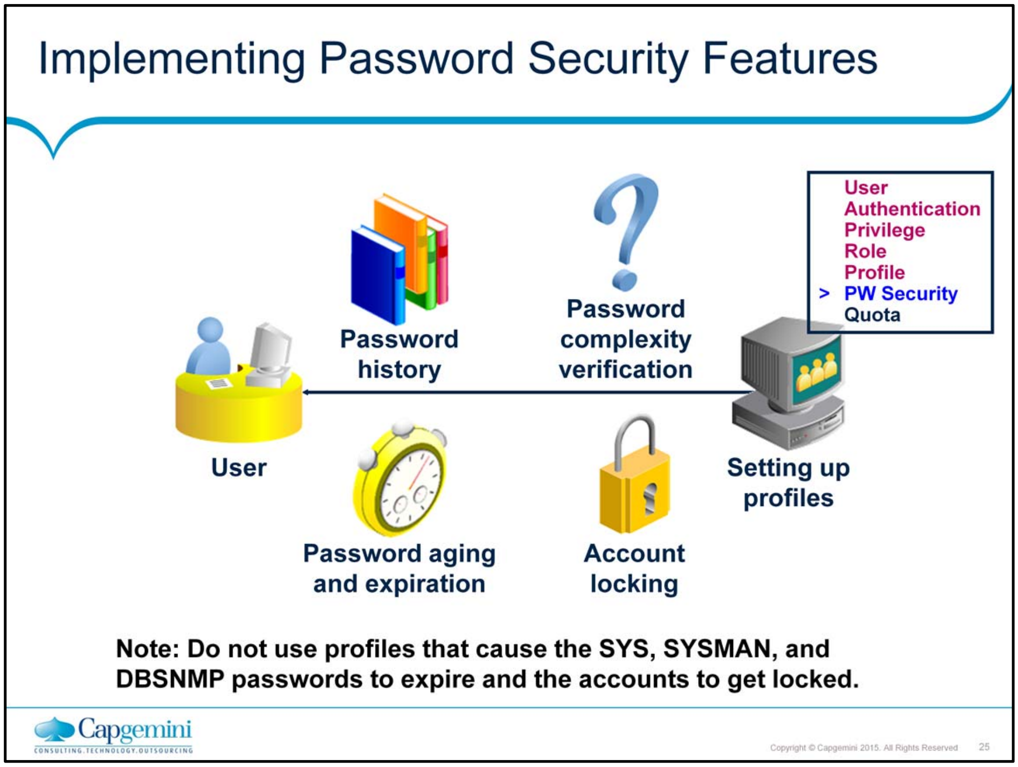
Private SGA: Limits the amount of space consumed within the System Global Area (SGA) for sorting, merging bitmaps, and so on. This restriction takes effect only if the session uses a shared server. (Shared servers are discussed in the lesson titled "Configuring the Oracle Network Environment").

Disk I/O: This limits the amount of data a user can read either at the per-session or per-call level. Reads/Session and Reads/Call place a limitation on the total number of reads from both memory and the disk. This can be done to ensure that no input/output (I/O)-intensive statements overuse memory and disks.

Profiles also allow a composite limit. Composite limits are based on a weighted combination of CPU/Session, Reads/Session, Connect Time, and Private SGA. Composite limits are discussed in more detail in the *Oracle Database Security Guide*.

To create a profile, select Administration > Schema > Users & Privileges > Profiles, and click the Create button.

Note: Resource Manager is an alternative to many of the profile settings. For more details about Resource Manager, see the *Oracle Database Administrator's Guide*.



Implementing Password Security Features

User
Authentication
Privilege
Role
Profile
> PW Security
Quota

Password history

Password complexity verification

User

Setting up profiles

Password aging and expiration

Account locking

Note: Do not use profiles that cause the SYS, SYSMAN, and DBSNMP passwords to expire and the accounts to get locked.

Implementing Password Security Features

Oracle password management is implemented with user profiles. Profiles can provide many standard security features including the following:

Account locking: Enables automatic locking of accounts for a set duration when users fail to log in to the system in the specified number of attempts.

The FAILED_LOGIN_ATTEMPTS parameter specifies the number of failed login attempts before the lockout of the account.

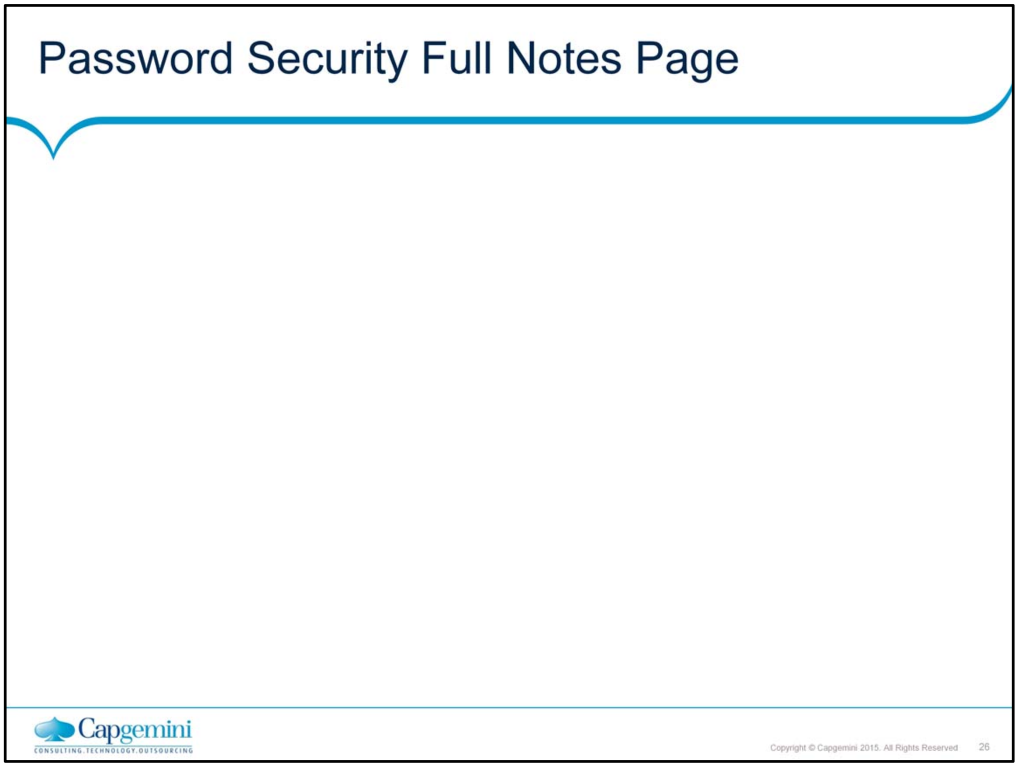
The PASSWORD_LOCK_TIME parameter specifies the number of days for which the account is locked after the specified number of failed login attempts.

Password aging and expiration: Enables user passwords to have a lifetime, after which the passwords expire and must be changed

The PASSWORD_LIFE_TIME parameter determines the lifetime of the password in days, after which the password expires.

The PASSWORD_GRACE_TIME parameter specifies a grace period in days for changing the password after the first successful login after the password has expired.

Note: Expiring passwords and locking the SYS, SYSMAN, and DBSNMP accounts prevent Enterprise Manager from functioning properly. The applications must catch the “password expired” warning message and handle the password change; otherwise, the grace period expires and the user is locked out without knowing the reason.



Password Security Full Notes Page

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

26

Implementing Password Security Features (continued)

Password history: Checks the new password to ensure that the password is not reused for a specified amount of time or a specified number of password changes. These checks can be implemented by using one of the following:

PASSWORD_REUSE_TIME: Specifies that a user cannot reuse a password for a given number of days

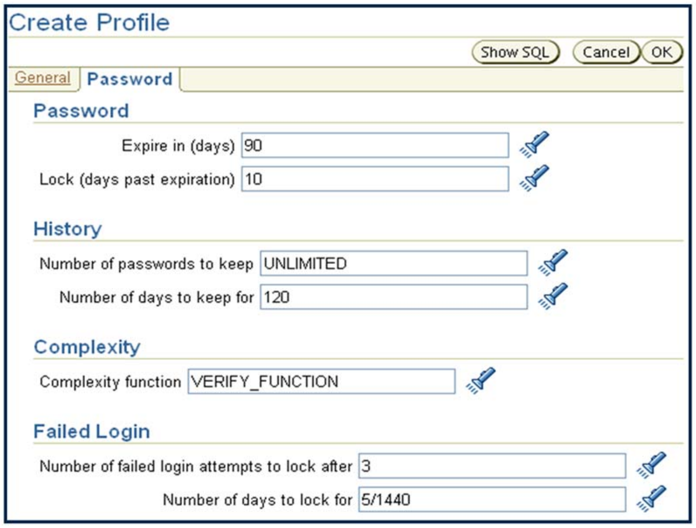
PASSWORD_REUSE_MAX: Specifies the number of password changes that are required before the current password can be reused

These two parameters are mutually exclusive, and so when one parameter is set to a value other than UNLIMITED (or DEFAULT, if the DEFAULT profile has the value set to UNLIMITED), the other parameter must be set to UNLIMITED.

Password complexity verification: Makes a complexity check on the password to verify that it meets certain rules. The check must ensure that the password is complex enough to provide protection against intruders who may try to break into the system by guessing the password.

The PASSWORD_VERIFY_FUNCTION parameter names a PL/SQL function that performs a password complexity check before a password is assigned. Password verification functions must be owned by the SYS user and must return a Boolean value (TRUE or FALSE).

Creating a Password Profile



Create Profile

Password

Expire in (days) 90

Lock (days past expiration) 10

History

Number of passwords to keep UNLIMITED

Number of days to keep for 120

Complexity

Complexity function VERIFY_FUNCTION

Failed Login

Number of failed login attempts to lock after 3

Number of days to lock for 5/1440

Creating a Password Profile

To create a password profile, select Administration > Schema > Users & Privileges > Profiles, and click the Create button.

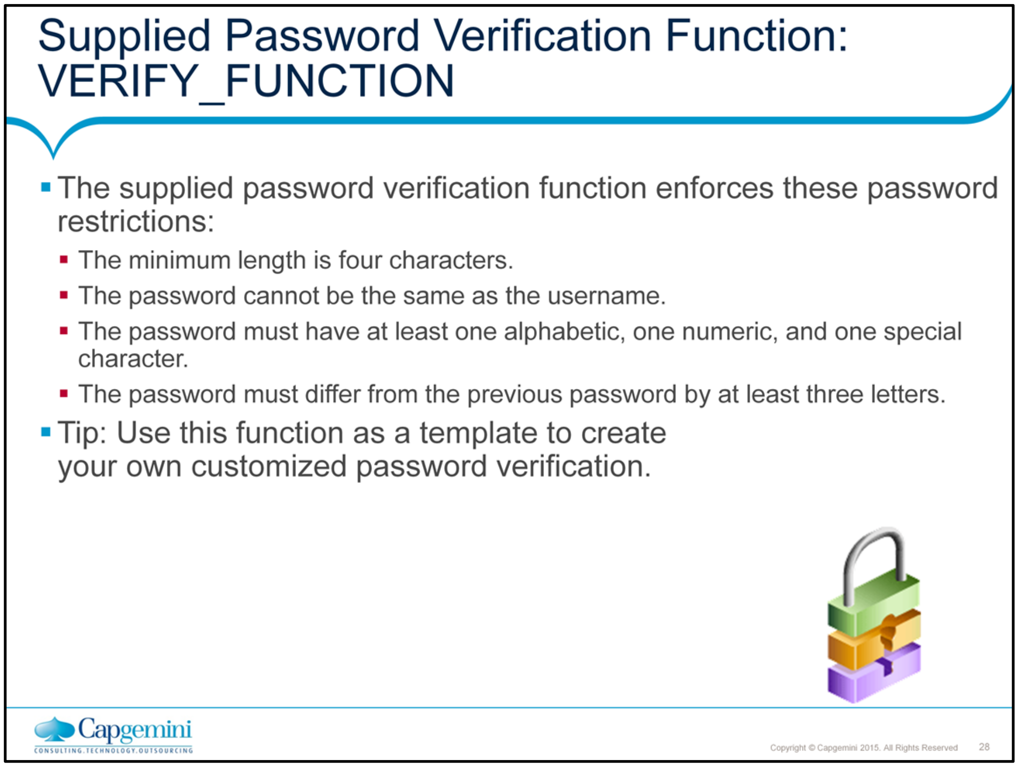
Common values for each of the settings can be chosen from a list of values (Click the flashlight icon to browse.), or you can enter a custom value.

All time periods are expressed in days, but can be expressed as fractions also. There are 1,440 minutes in a day, and so 5/1440 is five minutes.

Enterprise Manager can also be used to edit existing password profiles.

Dropping a Password Profile

In Enterprise Manager, you cannot drop a profile that is used by users. However, if you drop a profile with the CASCADE option (for example, in SQL*Plus), then all users who have that profile are automatically assigned the DEFAULT profile.



Supplied Password Verification Function: VERIFY_FUNCTION

- The supplied password verification function enforces these password restrictions:
  - The minimum length is four characters.
  - The password cannot be the same as the username.
  - The password must have at least one alphabetic, one numeric, and one special character.
  - The password must differ from the previous password by at least three letters.
- Tip: Use this function as a template to create your own customized password verification.

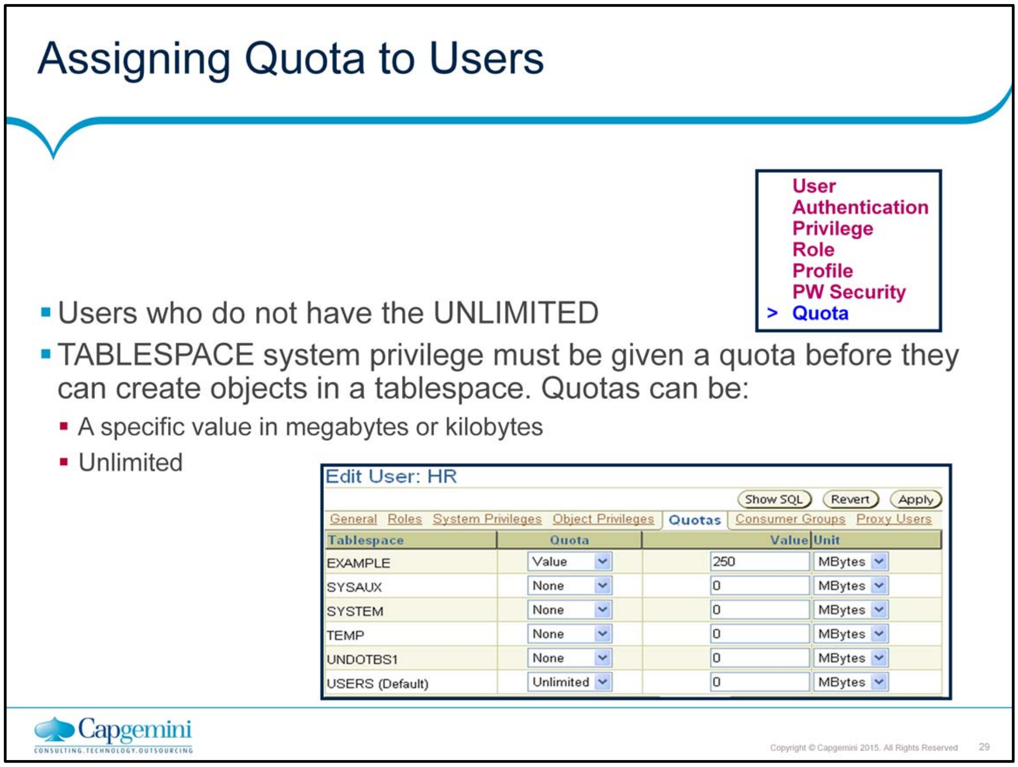
Supplied Password Verification Function: VERIFY_FUNCTION

The Oracle server provides a password complexity verification function named VERIFY_FUNCTION. This function is created with the <oracle_home>/rdbms/admin/utlpwdmg.sql script. The password complexity verification function must be created in the SYS schema. It can be used as a template for your customized password verification.

In addition to creating VERIFY_FUNCTION, the utlpwdmg script also changes the DEFAULT profile with the following ALTER PROFILE command:

```
ALTER PROFILE default LIMIT
PASSWORD_LIFE_TIME 60
PASSWORD_GRACE_TIME 10
PASSWORD_REUSE_TIME 1800
PASSWORD_REUSE_MAX UNLIMITED
FAILED_LOGIN_ATTEMPTS 3
PASSWORD_LOCK_TIME 1/1440
PASSWORD_VERIFY_FUNCTION
verify_function;
```

Remember that when users are created, they are assigned the DEFAULT profile, unless another profile is specified.



Assigning Quota to Users

User
Authentication
Privilege
Role
Profile
PW Security
> Quota

- Users who do not have the UNLIMITED
- TABLESPACE system privilege must be given a quota before they can create objects in a tablespace. Quotas can be:
  - A specific value in megabytes or kilobytes
  - Unlimited

Edit User: HR

Tablespace	Quota	Value	Unit
EXAMPLE	Value	250	MBytes
SYSAUX	None	0	MBytes
SYSTEM	None	0	MBytes
TEMP	None	0	MBytes
UNDOTBS1	None	0	MBytes
USERS (Default)	Unlimited	0	MBytes

29

Assigning Quota to Users

Quota is a space allowance in a given tablespace. By default, a user has no quota on any of the tablespaces. You have three options for providing a user quota on a tablespace.

Unlimited: This allows the user to use as much space as is available in the tablespace.

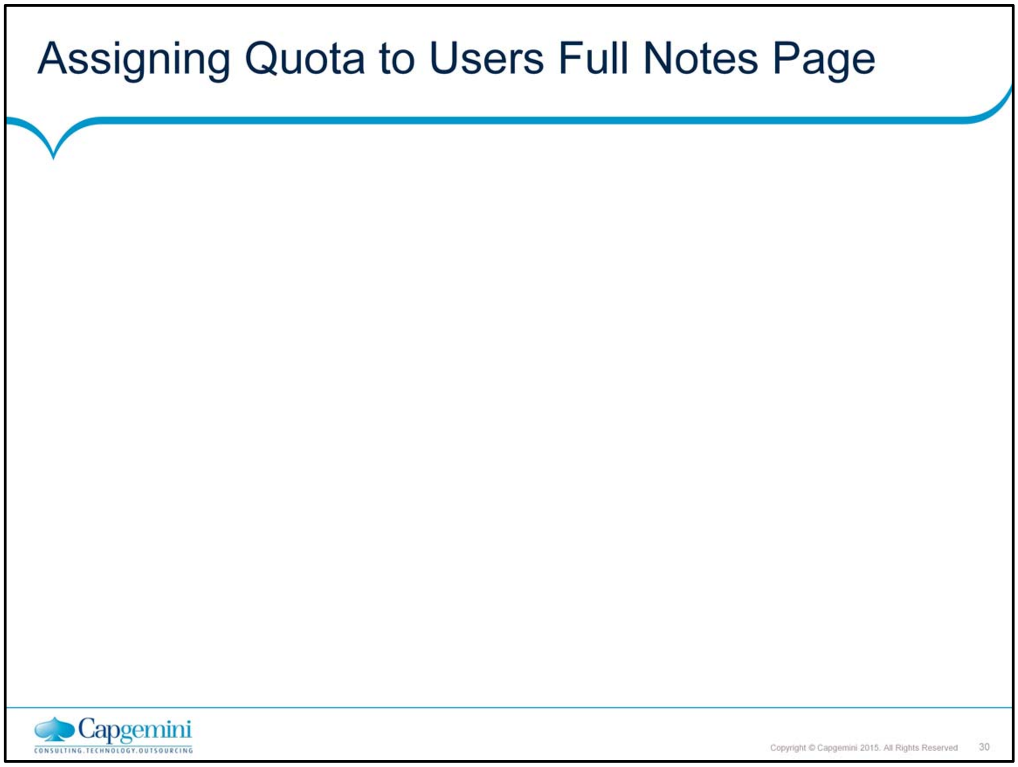
Value: This is a number of kilobytes or megabytes that the user can use. This does not guarantee that the space is set aside for the user. This value can be larger or smaller than the current space that is available in the tablespace.

UNLIMITED TABLESPACE system privilege: This system privilege overrides all individual tablespace quotas and gives the user unlimited quota on all tablespaces, including SYSTEM and SYSAUX. This privilege must be granted with caution.

Note: Be aware that granting the RESOURCE role includes granting this privilege.

You must not provide quota to users on the SYSTEM or SYSAUX tablespace. Typically, only the SYS and SYSTEM users must be able to create objects in the SYSTEM or SYSAUX tablespace.

You do not need quota on an assigned temporary tablespace or any undo tablespaces.



Assigning Quota to Users Full Notes Page

30

Assigning Quota to Users (continued)

When does the Oracle instance use quota?

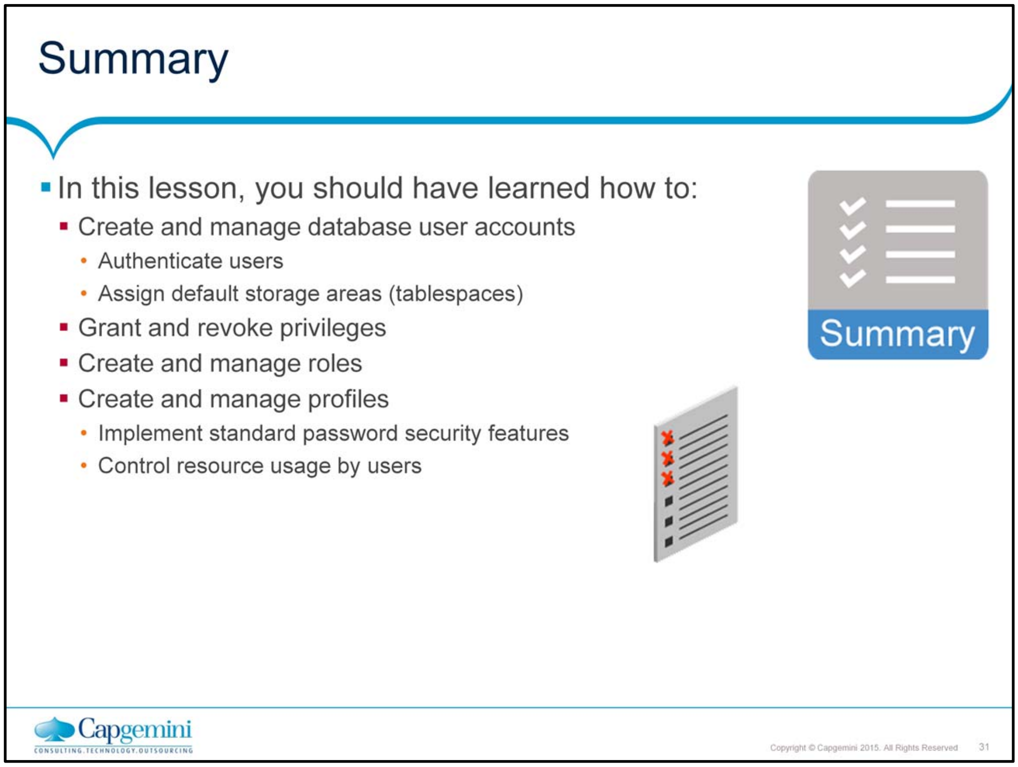
Quotas are used when a user creates or extends a segment.

Which activities do not count against the quota?

Activities that do not use space in the assigned tablespace do not affect the quota, such as creating views or using temporary tablespace.

When is the quota replenished?

The quota is replenished when objects owned by the user are dropped with the PURGE clause or the objects in the recycle bin are automatically purged.



Summary

- In this lesson, you should have learned how to:
  - Create and manage database user accounts
    - Authenticate users
    - Assign default storage areas (tablespaces)
  - Grant and revoke privileges
  - Create and manage roles
  - Create and manage profiles
    - Implement standard password security features
    - Control resource usage by users