Database Security(Level If Security)

Controlling Database Access

This lecture explains how to control access to database. It includes:

- Database Security
- Schemas, Database Users, and Security Domains
- User Authentication
- User Table space Settings and Quotas
- The User Group PUBLIC
- User Resource Limits and Profiles
- Licensing

Database Security

Database security entails allowing or disallowing user actions on the database and the objects within it. Oracle uses schemas and security domains to control access to data and to restrict the use of various database resources.

Oracle provides comprehensive discretionary access control. *Discretionary access control* regulates all user access to named objects through privileges. A privilege is permission to access a named object in a prescribed manner; for example, permission to query a table. Privileges are granted to users at the discretion of other users-hence the term "discretionary access control".

Schemas, Database Users, and Security Domains

A *user* (sometimes called a *username*) is a name defined in the database that can connect to and access objects. A *schema* is a named collection of objects, such as tables, views, clusters, procedures, and packages, associated with a particular user. Schemas and users help database administrators manage database security.

To access a database, a user must run a database application (such as an Oracle Forms form, SQL*Plus, or a precompiler program) and connect using a username defined in the database.

When a database user is created, a corresponding schema of the same name is created for the user. By default, once a user connects to a database, the user has access to all objects contained in the corresponding schema. A user is associated only with the schema of the same name; therefore, the terms user and schema are often used interchangeably.

The access rights of a user are controlled by the different settings of the user's security domain. When creating a new database user or altering an existing one, the security administrator must make several decisions concerning a user's security domain. These include

- whether user authentication information is maintained by the database, the operating system, or a network authentication service
- settings for the user's default and temporary tablespaces
- a list, if any, of tablespaces accessible to the user and the associated quotas for each listed tablespace
- the user's resource limit profile; that is, limits on the amount of system resources available to the user
- the privileges and roles that provide the user with appropriate access to objects needed to perform database operations

User Authentication

To prevent unauthorized use of a database username, Oracle provides user validation via three different methods for normal database users:

- authentication by the operating system
- authentication by a network service
- authentication by the associated Oracle database

For simplicity, one method is usually used to authenticate all users of a database. However, Oracle allows use of all methods within the same database instance.

Oracle also encrypts passwords during transmission to ensure the security of network authentication.

Oracle requires special authentication procedures for database administrators, because they perform special database operations.

Authentication by the Operating System

Some operating systems permit Oracle to use information maintained by the operating system to authenticate users. The benefits of operating system authentication are:

- Users can connect to Oracle more conveniently (without specifying a username or password). For example, a user can invoke SQL*Plus and skip the username and password prompts by entering
- SQLPLUS /
- Control over user authorization is centralized in the operating system; Oracle need not store or manage user passwords. However, Oracle still maintains usernames in the database.
- Username entries in the database and operating system audit trails correspond.

If the operating system is used to authenticate database users, some special considerations arise with respect to distributed database environments and database links.

Additional Information: For more information about authenticating via your operating system, see your Oracle operating system-specific documentation.

Authentication by the Network

If network authentication services are available to you (such as DCE, Kerberos, or SESAME), Oracle can accept authentication from the network service. To use a network

authentication service with Oracle, you must also have the Oracle Secure Network Services product.

Authentication by the Oracle Database

Oracle can authenticate users attempting to connect to a database by using information stored in that database. You *must* use this method when the operating system cannot be used for database user validation.

When Oracle uses database authentication, you create each user with an associated password. A user provides the correct password when establishing a connection to prevent unauthorized use of the database. Oracle stores a user's password in the data dictionary in an encrypted format. A user can change his or her password at any time.

Password Encryption while Connecting

To protect password confidentiality, Oracle allows you to encrypt passwords during network (client/server and server/server) connections. If you enable this functionality on the client and server machines, Oracle encrypts passwords using a modified DES (Data Encryption Standards) algorithm before sending them across the network.

Account Locking

Oracle can lock a user's account if the user fails to login to the system within a specified number of attempts. Depending on how the account is configured, it can be unlocked automatically after a specified time interval or it must be unlocked by the database administrator.

The CREATE PROFILE statement configures the number of failed logins a user can attempt and the amount of time the account remains locked before automatic unlock.

The database administrator can also lock accounts manually. When this occurs, the account cannot be unlocked automatically but must be unlocked explicitly by the database administrator.

Password Lifetime and Expiration

Password lifetime and expiration options allow the database administrator to specify a lifetime for passwords, after which time they expire and must be changed before a login to the account can be completed. On first attempt to login to the database account after the password expires, the user's account enters the grace period, and a warning message is issued to the user every time the user tries to login until the grace period is over.

The user is expected to change the password within the grace period. If the password is not changed within the grace period, the account is locked and no further logins to that account are allowed without assistance by the database administrator.

The database administrator can also set the password state to expired. When this happens, the users account status is changed to expired, and when the user logs in, the account enters the grace period.

Password History

The password history option checks each newly specified password to ensure that a password is not reused for the specified amount of time or for the specified number of password changes. The database administrator can configure the rules for password reuse with CREATE PROFILE statements.

Password Complexity Verification

Complexity verification checks that each password is complex enough to provide reasonable protection against intruders who try to break into the system by guessing passwords.

The Oracle default password complexity verification routine requires that each password:

- be a minimum of four characters in length
- not equal the userid
- include at least one alpha, one numeric, and one punctuation mark

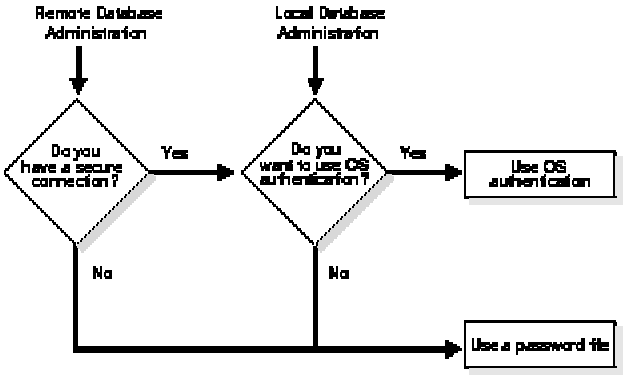
- not match any word on an internal list of simple words like welcome, account, database, user, and so on.
- differ from the previous password by at least three characters.

Database Administrator Authentication

Database administrators perform special operations (such as shutting down or starting up a database) that should not be performed by normal database users. Oracle provides a more secure authentication scheme for database administrator usernames.

You can choose between operating system authentication or password files to authenticate database administrators; Figure A-1 illustrates the choices you have for database administrator authentication schemes, depending on whether you administer your database locally (on the same machine on which the database resides) or if you administer many different database machines from a single remote client.

Figure A-1: Database Administrator Authentication Methods



On most operating systems, OS authentication for database administrators involves placing the OS username of the database administrator in a special group (on UNIX systems, this is the dba group) or giving that OS username a special process right.

Additional Information: For information about OS authentication of database administrators, see your Oracle operating system-specific documentation.

The database uses password files to keep track of database usernames who have been granted the SYSDBA and SYSOPER privileges. These privileges allow database administrators to perform the following actions:

SYSOPER	Permits you to perform STARTUP, SHUTDOWN, ALTER DATABASE OPEN/MOUNT, ALTER DATABASE BACKUP, ARCHIVE LOG, and RECOVER, and includes the RESTRICTED SESSION privilege.
SYSDBA	Contains all system privileges with ADMIN OPTION, and the SYSOPER system privilege; permits CREATE DATABASE and time-based recovery.

For information about password files, see the Oracle8 Server Administrator's Guide.

User Tablespace Settings and Quotas

As part of every user's security domain, the database administrator can set several options regarding tablespace usage:

- the user's default tablespace
- the user's temporary tablespace
- space usage quotas on tablespaces of the database for the user

Default Tablespace

When a user creates a schema object without specifying a tablespace to contain the object, Oracle places the object in the user's default tablespace. You set a user's default tablespace when the user is created; you can change it after the user has been created.

Temporary Tablespace

When a user executes a SQL statement that requires the creation of a temporary segment, Oracle allocates that segment in the user's temporary tablespace.

Tablespace Access and Quotas

You can assign to each user a *tablespace quota* for any tablespace of the database. Doing so can accomplish two things:

- You allow the user to use the specified tablespace to create objects, provided that the user has the appropriate privileges.
- You can limit the amount of space allocated for storage of a the's objects in the specified tablespace.

By default, each user has no quota on any tablespace in the database. Therefore, if the user has the privilege to create some type of schema object, he or she must also have been either assigned a tablespace quota in which to create the object or been given the privilege to create that object in the schema of another user who was assigned a sufficient tablespace quota.

You can assign two types of tablespace quotas to a user: a quota for a specific amount of disk space in the tablespace (specified in bytes, kilobytes, or megabytes), or a quota for an unlimited amount of disk space in the tablespace. You should assign specific quotas to prevent a user's objects from consuming too much space in a tablespace.

Tablespace quotas and temporary segments have no effect on each other:

- Temporary segments do not consume any quota that a user might possess.
- Temporary segments can be created in a tablespace for which a user has no quota.

You can assign a tablespace quota to a user when you create that user, and you can change that quota or add a different quota later.

Revoke a user's tablespace access by altering the user's current quota to zero. With a quota of zero, the user's objects in the revoked tablespace remain, but the objects cannot be allocated any new space.

The User Group PUBLIC

Each database contains a user group called PUBLIC. The PUBLIC user group provides public access to specific schema objects (tables, views, and so on) and provides all users with specific system privileges. Every user automatically belongs to the PUBLIC user group.

As members of PUBLIC, users may see (select from) all data dictionary tables prefixed with USER and ALL. Additionally, a user can grant a privilege or a role to PUBLIC. All users can use the privileges granted to PUBLIC.

You can grant (or revoke) any system privilege, object privilege, or role to PUBLIC. See Chapter 25, "Privileges and Roles" for more information on privileges and roles. However, to maintain tight security over access rights, grant only privileges and roles of interest to **all** users to PUBLIC.

Granting and revoking some system and object privileges to and from PUBLIC can cause every view, procedure, function, package, and trigger in the database to be recompiled.

PUBLIC has the following restrictions:

- You cannot assign tablespace quotas to PUBLIC, although you can assign the UNLIMITED TABLESPACE system privilege to PUBLIC.
- You can create database links and synonyms as PUBLIC (using CREATE PUBLIC DATABASE LINK/SYNONYM), but no other object can be owned by PUBLIC. For example, the following statement is not legal:
- CREATE TABLE public.emp . . . ;

Note: Rollback segments can be created with the keyword PUBLIC, but these are not owned by the PUBLIC user group. All rollback segments are owned by SYS. See Chapter 2, "Data Blocks, Extents, and Segments", for more information about rollback segments.

User Resource Limits and Profiles

You can set limits on the amount of various system resources available to each user as part of a user's security domain. By doing so, you can prevent the uncontrolled consumption of valuable system resources such as CPU time.

This resource limit feature is very useful in large, multiuser systems, where system resources are very expensive. Excessive consumption of these resources by one or more users can detrimentally affect the other users of the database. In single-user or small-scale multiuser database systems, the system resource feature is not as important, because users' consumption of system resources is less likely to have detrimental impact.

You manage a user's resource limits with his or her profile-a named set of resource limits that you can assign to that user. Each Oracle database can have an unlimited number of profiles. Oracle allows the security administrator to enable or disable the enforcement of profile resource limits universally.

If you set resource limits, a slight degradation in performance occurs when users create sessions. This is because Oracle loads all resource limit data for the user when a user connects to a database.

Types of System Resources and Limits

Oracle can limit the use of several types of system resources, including CPU time and logical reads. In general, you can control each of these resources at the session level, the call level, or both:

Session Level	Each time a user connects to a database, a session is created. Each session consumes CPU time and memory on the computer that executes Oracle. You can set several resource limits at the session level. If a user exceeds a session-level resource limit, Oracle terminates (rolls back) the current statement and returns a message indicating the session limit has been reached. At this point, all previous statements in the current transaction

	are intact, and the only operations the user can perform are COMMIT, ROLLBACK, or disconnect (in this case, the current transaction is committed); all other operations produce an error. Even after the transaction is committed or rolled back, the user can accomplish no more work during the current session.
Call Level	Each time a SQL statement is executed, several steps are taken to process the statement. During this processing, several calls are made to the database as part of the different execution phases. To prevent any one call from using the system excessively, Oracle allows you to set several resource limits at the call level. If a user exceeds a call-level resource limit, Oracle halts the processing of the statement, rolls back the statement, and returns an error. However, all previous statements of the current transaction remain intact, and the user's session remains connected.

CPU Time

When SQL statements and other types of calls are made to Oracle, an amount of CPU time is necessary to process the call. Average calls require a small amount of CPU time. However, a SQL statement involving a large amount of data or a runaway query can potentially consume a large amount of CPU time, reducing CPU time available for other processing.

To prevent uncontrolled use of CPU time, you can limit the CPU time per call and the total amount of CPU time used for Oracle calls during a session. The limits are set and measured in CPU one-hundredth seconds (0.01 seconds) used by a call or a session.

Logical Reads

Input/output (I/O) is one of the most expensive operations in a database system. I/O intensive statements can monopolize memory and disk use and cause other database operations to compete for these resources.

To prevent single sources of excessive I/O, Oracle let you limit the logical data block reads per call and per session. Logical data block reads include data block reads from both memory and disk. The limits are set and measured in number of block reads performed by a call or during a session.

Other Resources

Oracle also provides for the limitation of several other resources at the session level:

- You can limit the number of *concurrent sessions per user*. Each user can create only up to a predefined number of concurrent sessions.
- You can limit the *idle time* for a session. If the time between Oracle calls for a session reaches the idle time limit, the current transaction is rolled back, the session is aborted, and the resources of the session are returned to the system. The next call receives an error that indicates the user is no longer connected to the instance. This limit is set as a number of elapsed minutes.

Note: Shortly after a session is aborted because it has exceeded an idle time limit, the process monitor (PMON) background process cleans up after the aborted session. Until PMON completes this process, the aborted session is still counted in any session/user resource limit.

- You can limit the elapsed connect time per session. If a session's duration exceeds the elapsed time limit, the current transaction is rolled back, the session is dropped, and the resources of the session are returned to the system. This limit is set as a number of elapsed minutes.

Note: Oracle does not constantly monitor the elapsed idle time or elapsed connection time. Doing so would reduce system performance. Instead, it checks every few minutes.

Therefore, a session can exceed this limit slightly (for example, by five minutes) before Oracle enforces the limit and aborts the session.

- You can limit the amount of private SGA space (used for private SQL areas) for a session. This limit is only important in systems that use multithreaded server configuration; otherwise, private SQL areas are located in the PGA. This limit is set as a number of bytes of memory in an instance's SGA. Use the characters "K" or "M" to specify kilobytes or megabytes.

Instructions on enabling and disabling resource limits are included in the Oracle8 Server Administrator's Guide.

Profiles

A profile is a named set of specified resource limits that can be assigned to valid username of an Oracle database. Profiles provide for easy management of resource limits.

When to Use Profiles

You need to create and manage user profiles only if resource limits are a requirement of your database security policy. To use profiles, first categorize the related types of users in a database. Just as roles are used to manage the privileges of related users, profiles are used to manage the resource limits of related users. Determine how many profiles are needed to encompass all types of users in a database and then determine appropriate resource limits for each profile.

Determining Values for Resource Limits of a Profile

Before creating profiles and setting the resource limits associated with them, you should determine appropriate values for each resource limit. You can base these values on the type of operations a typical user performs. For example, if one class of user does not normally perform a high number of logical data block reads, then the

LOGICAL_READS_PER_SESSION and LOGICAL_READS_PER_CALL limits should be set conservatively.

Usually, the best way to determine the appropriate resource limit values for a given user profile is to gather historical information about each type of resource usage. For example, the database or security administrator can use the AUDIT SESSION option to gather information about the limits CONNECT_TIME, LOGICAL_READS_PER_SESSION, and LOGICAL_READS_PER_CALL. See Chapter 26, "Auditing", for more information.

You can gather statistics for other limits using the Monitor feature of Enterprise Manager, specifically the Statistics monitor.

Licensing

Oracle is usually licensed for use by a maximum number of named users or by a maximum number of concurrently connected users. The database administrator (DBA) is responsible for ensuring that the site complies with its license agreement. Oracle's licensing facility helps the DBA monitor system use by tracking and limiting the number of sessions concurrently connected to an instance or the number of users created in a database.

If the DBA discovers that more than the licensed number of sessions need to connect, or more than the licensed number of users need to be created, he or she can upgrade the Oracle license to raise the appropriate limit. (To upgrade an Oracle license, you must contact your Oracle representative.)

Note: When Oracle is embedded in an Oracle application (such as Oracle Office), run on some older operating systems, or purchased for use in some countries, it is not licensed for either a set number of sessions or a set group of users. In such cases only, the Oracle licensing mechanisms do not apply and should remain disabled.

The following sections explain the two major types of licensing available for Oracle.

Concurrent Usage Licensing

In *concurrent usage licensing*, the license specifies a number of *concurrent users*, which are sessions that can be connected concurrently to the database on the specified computer at any time. This number includes all batch processes and online users. If a single user has multiple concurrent sessions, each session counts separately in the total number of sessions. If multiplexing software (such as a TP monitor) is used to reduce the number of sessions directly connected to the database, the number of concurrent users is the number of distinct inputs to the multiplexing front end.

The concurrent usage licensing mechanism allows a DBA to:

- Set a limit on the number of concurrent sessions that can connect to an instance by setting the LICENSE_MAX_SESSIONS parameter. Once this limit is reached, only users who have the RESTRICTED SESSION system privilege can connect to the instance; this allows DBA to kill unneeded sessions, allowing other sessions to connect.
- Set a warning limit on the number of concurrent sessions that can connect to an instance by setting the LICENSE_SESSIONS_WARNING parameter. Once the warning limit is reached, Oracle allows additional sessions to connect (up to the maximum limit described above), but sends a warning message to any user who connects with RESTRICTED SESSION privilege and records a warning message in the database's ALERT file.

The DBA can set these limits in the database's parameter file so that they take effect when the instance starts and can change them while the instance is running (using the ALTER SYSTEM command). The latter is useful for databases that cannot be taken offline.

The session licensing mechanism allows a DBA to check the current number of connected sessions and the maximum number of concurrent sessions since the instance started. The V$LICENSE view shows the current settings for the license limits, the

current number of sessions, and the highest number of concurrent sessions since the instance started (the session "high water mark"). The DBA can use this information to evaluate the system's licensing needs and plan for system upgrades.

For instances running with the Parallel Server, each instance can have its own concurrent usage limit and warning limit. The sum of the instances' limits must not exceed the site's concurrent usage license. See the Oracle8 Server Administrator's Guide for more information.

The concurrent usage limits apply to all user sessions, including sessions created for incoming database links. They do not apply to sessions created by Oracle or to recursive sessions. Sessions that connect through external multiplexing software are not counted separately by the Oracle licensing mechanism, although each contributes individually to the Oracle license total. The DBA is responsible for taking these sessions into account.

Named User Licensing

In *named user licensing*, the license specifies a number of named users, where a *named user* is an individual who is authorized to use Oracle on the specified computer. No limit is set on the number of sessions each user can have concurrently, or on the number of concurrent sessions for the database.

Named user licensing allows a DBA to set a limit on the number of users that are defined in a database, including users connected via database links. Once this limit is reached, no one can create a new user. This mechanism assumes that each person accessing the database has a unique user name in the database and that no two (or more) people share a user name.

The DBA can set this limit in the database's parameter file so that it takes effect when the instance starts and can change it while the instance is running (using the ALTER SYSTEM command). The latter is useful for databases that cannot be taken offline.

Review Question

1. What are the different level of Security

Selected Bibliography

- [ARIES] C. Mohan, et al.: ARIES: A Transaction Recovery Method Supporting Fine-Granularity Locking and Partial Rollbacks Using Write-Ahead Logging., TODS 17(1): 94-162 (1992).
- [CACHE] C. Mohan: Caching Technologies for Web Applications, A Tutorial at the Conference on Very Large Databases (VLDB), Rome, Italy, 2001.
- [CODASYL] ACM: CODASYL Data Base Task Group April 71 Report, New York, 1971.
- [CODD] E. Codd: A Relational Model of Data for Large Shared Data Banks. ACM 13(6):377-387 (1970).
- [EBXML] http://www.ebxml.org.
- [FED] J. Melton, J. Michels, V. Josifovski, K. Kulkarni, P. Schwarz, K. Zeidenstein: SQL and Management of External Data', SIGMOD Record 30(1):70-77, 2001.
- [GRAY] Gray, et al.: Granularity of Locks and Degrees of Consistency in a Shared Database., IFIP Working Conference on Modelling of Database Management Systems, 1-29, AFIPS Press.
- [INFO] P. Lyman, H. Varian, A. Dunn, A. Strygin, K. Swearingen: How Much Information? at http://www.sims.berkeley.edu/research/projects/how-much-info/.
- [LIND] B. Lindsay, et. al: Notes on Distributed Database Systems. IBM Research Report RJ2571, (1979).