Database Security Issues

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.

The centralized and multi-user nature of a DBMS requires that some form of *security control* is in place, both to prevent unauthorized access and to limit access for authorized users. Security control can generally be divided into two areas, *user authorization* and *transaction authorization*.

User Authorization

User authorization helps to protect a database against unauthorized use, usually by requiring that a user enter a user name and a password to gain entry to the system. The password is usually known only to the user and the DBMS, and is protected by the DBMS at least as well as the data in the database. However, it should be noted this user name and password scheme can not *guarantee* the security of the database. It does not prevent you from choosing a password that is easy to guess (like the name of a spouse or pet) or from recording your password in an accessible location (like on the front of your computer!).

Transaction Authorization

Generally, not all users are given the same access rights to different databases or different parts of the same database. In some cases, sensitive data such as employee salaries should only be accessible to those users who need it. In other cases, some users may only require the ability to read some data items, where other users require the ability to both read and update the data.

A Point-Of-Sale (POS) system is a good example of the second case: clerks working in a store might need read access for the price of an item, but should not be able to change the price. Employees at the head office may need to read and update the data, in order to enter new prices for the item.

Transaction authorization helps to protect a database against an authorized user trying to access a data item they do not have permission to access (this may occur either intentionally or unintentionally). The DBMS usually keeps a record of what rights have been granted to users on all of the data objects in the database, and checks these rights every time a user transaction tries to access the database. If the user does not have the proper rights to a data item, the transaction will not be allowed. It is the responsibility of the Database Administrator to explicitly grant the rights assigned to each user.

Database Security

Multi-user database systems, such as Oracle, include security features that control how a database is accessed and used. For example, security mechanisms do the following:

- prevent unauthorized database access
- prevent unauthorized access to schema objects
- control disk usage
- control system resource usage (such as CPU time)
- audit user actions

Associated with each database user is a *schema* by the same name. A schema is a logical collection of objects (tables, views, sequences, synonyms, indexes, clusters, procedures, functions, packages, and database links). By default, each database user creates and has access to all objects in the corresponding schema.

Database security can be classified into two distinct categories: system security and data security.

System security includes the mechanisms that control the access and use of the database at the system level. For example, system security includes:

- valid username/password combinations
- the amount of disk space available to the objects of a user
- the resource limits for a user

System security mechanisms check:

- whether a user is authorized to connect to the database

- whether database auditing is active
- which system operations a user can perform

Data security includes the mechanisms that control the access and use of the database at the object level. For example, data security includes

- which users have access to a specific schema object and the specific types of actions allowed for each user on the object (for example, user SCOTT can issue SELECT and INSERT statements but not DELETE statements using the EMP table)
- the actions, if any, that are audited for each schema object

Security Mechanisms

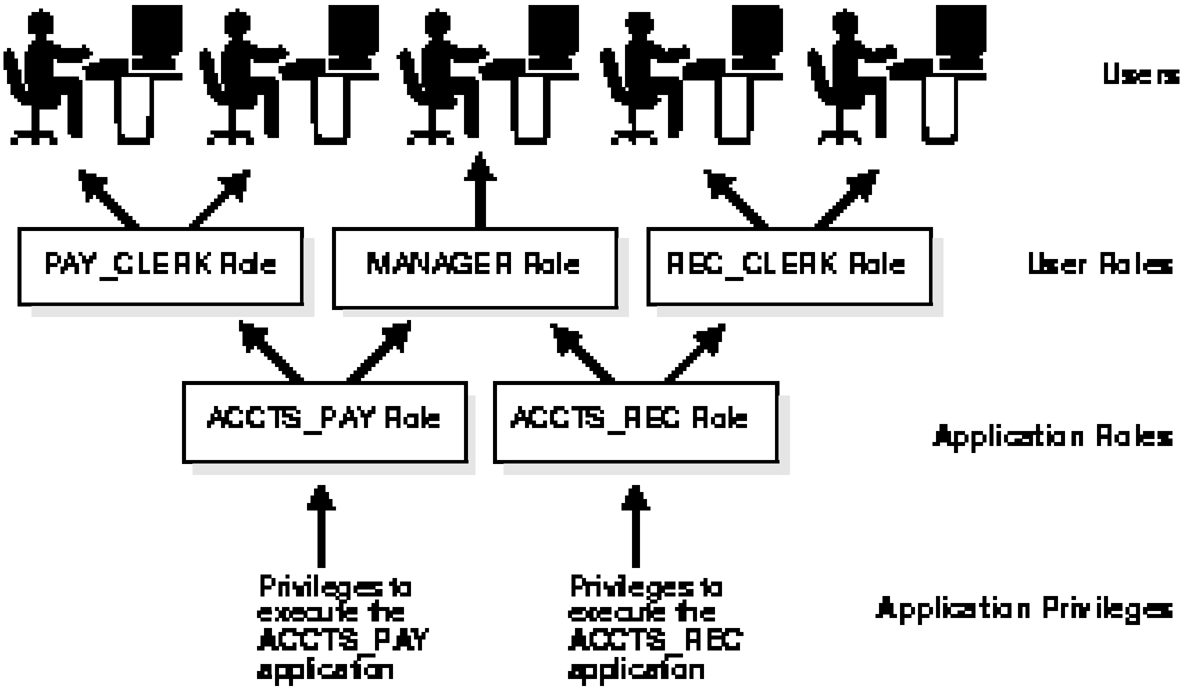
The Oracle Server provides *discretionary access control*, which is a means of restricting access to information based on privileges. The appropriate privilege must be assigned to a user in order for that user to access an object. Appropriately privileged users can grant other users privileges at their discretion; for this reason, this type of security is called "discretionary".

Oracle manages database security using several different facilities:

- database users and schemas
- privileges
- roles
- storage settings and quotas
- resource limits
- auditing

Figure 1-4 illustrates the relationships of the different Oracle security facilities, and the following sections provide an overview of users, privileges, and roles.

Figure 1-4: Oracle Security Features



Database Users and Schemas

Each Oracle database has a list of usernames. To access a database, a user must use a database application and attempt a connection with a valid username of the database. Each username has an associated password to prevent unauthorized use.

Security Domain

Each user has a *security domain* - a set of properties that determine such things as the:

- actions (privileges and roles) available to the user
- tablespace quotas (available disk space) for the user
- system resource limits (for example, CPU processing time) for the user

Each property that contributes to a user's security domain is discussed in the following sections.

Privileges

A *privilege* is a right to execute a particular type of SQL statement. Some examples of privileges include the

- right to connect to the database (create a session)
- right to create a table in your schema
- right to select rows from someone else's table
- right to execute someone else's stored procedure

The privileges of an Oracle database can be divided into two distinct categories: system privileges and object privileges.

System Privileges

System privileges allow users to perform a particular systemwide action or a particular action on a particular type of object. For example, the privileges to create a tablespace or to delete the rows of any table in the database are system privileges. Many system privileges are available only to administrators and application developers because the privileges are very powerful.

Object Privileges

Object privileges allow users to perform a particular action on a specific schema object. For example, the privilege to delete rows of a specific table is an object privilege. Object privileges are granted (assigned) to end-users so that they can use a database application to accomplish specific tasks.

Granting Privileges

Privileges are granted to users so that users can access and modify data in the database. A user can receive a privilege two different ways:

- Privileges can be granted to users explicitly. For example, the privilege to insert records into the EMP table can be explicitly granted to the user SCOTT.

- Privileges can be granted to *roles* (a named group of privileges), and then the role can be granted to one or more users. For example, the privilege to insert records into the EMP table can be granted to the role named CLERK, which in turn can be granted to the users SCOTT and BRIAN.

Because roles allow for easier and better management of privileges, privileges are normally granted to roles and not to specific users. The following section explains more about roles and their use.

Roles

Oracle provides for easy and controlled privilege management through roles. *Roles* are named groups of related privileges that are granted to users or other roles. The following properties of roles allow for easier privilege management:

- *reduced granting of privileges* - Rather than explicitly granting the same set of privileges to many users, a database administrator can grant the privileges for a group of related users granted to a role. And then the database administrator can grant the role to each member of the group.
- *dynamic privilege management* - When the privileges of a group must change, only the privileges of the role need to be modified. The security domains of all users granted the group's role automatically reflect the changes made to the role.
- *selective availability of privileges* - The roles granted to a user can be selectively enabled (available for use) or disabled (not available for use). This allows specific control of a user's privileges in any given situation.
- *application awareness* - A database application can be designed to enable and disable selective roles automatically when a user attempts to use the application.

Database administrators often create roles for a database application. The DBA grants an application role all privileges necessary to run the application. The DBA then grants the application role to other roles or users. An application can have several different roles, each granted a different set of privileges that allow for more or less data access while using the application.

The DBA can create a role with a password to prevent unauthorized use of the privileges granted to the role. Typically, an application is designed so that when it starts, it enables the proper role. As a result, an application user does not need to know the password for an application's role.

Storage Settings and Quotas

Oracle provides means for directing and limiting the use of disk space allocated to the database on a per user basis, including default and temporary tablespaces and tablespace quotas.

Default Tablespace

Each user is associated with a *default tablespace*. When a user creates a table, index, or cluster and no tablespace is specified to physically contain the object, the user's default tablespace is used if the user has the privilege to create the object and a quota in the specified default tablespace. The default tablespace feature provides Oracle with information to direct space usage in situations where object location is not specified.

Temporary Tablespace

Each user has a *temporary tablespace*. When a user executes a SQL statement that requires the creation of temporary segments (such as the creation of an index), the user's temporary tablespace is used. By directing all users' temporary segments to a separate tablespace, the temporary tablespace feature can reduce I/O contention among temporary segments and other types of segments.

Tablespace Quotas

Oracle can limit the collective amount of disk space available to the objects in a schema. *Quotas* (space limits) can be set for each tablespace available to a user. The tablespace quota security feature permits selective control over the amount of disk space that can be consumed by the objects of specific schemas.

Profiles and Resource Limits

Each user is assigned a *profile* that specifies limitations on several system resources available to the user, including the

- number of concurrent sessions the user can establish
- CPU processing time
    - available to the user's session
    - available to a single call to Oracle made by a SQL statement
- amount of logical I/O
    - available to the user's session
    - available to a single call to Oracle made by a SQL statement
- amount of idle time for the user's session allowed
- amount of connect time for the user's session allowed
- password restrictions
    - account locking after multiple unsuccessful login attemts
    - password expiration and grace period
    - password reuse and complexity restrictions

Different profiles can be created and assigned individually to each user of the database. A default profile is present for all users not explicitly assigned a profile. The resource limit feature prevents excessive consumption of global database system resources.

Review Question

1. Write a short note on Data 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).