Chap (Database security & auditing)

Privileges and Roles

Privileges

A *privilege* is a right to execute a particular type of SQL statement or to access another user's object. Some examples of privileges include

- the right to connect to the database (create a session)
- the right to create a table
- the right to select rows from another user's table
- the right to execute another user's stored procedure

You grant privileges to users so these users can accomplish tasks required for their job. You should grant a privilege only to a user who absolutely requires the privilege to accomplish necessary work. Excessive granting of unnecessary privileges can lead to compromised security. A user can receive a privilege in two different ways:

- You can grant privileges to users explicitly. For example, you can explicitly grant the privilege to insert records into the EMP table to the user SCOTT.
- You can also grant privileges to a role (a named group of privileges), and then grant the role to one or more users. For example, you can grant the privileges to select, insert, update, and delete records from the EMP table to the role named CLERK, which in turn you can grant to the users SCOTT and BRIAN.

Because roles allow for easier and better management of privileges, you should normally grant privileges to roles and not to specific users.

There are two distinct categories of privileges:

- system privileges
- object privileges

System Privileges

A system privilege is the right to perform a particular action, or to perform a particular action on a particular type of object. For example, the privileges to create tablespaces and to delete the rows of any table in a database are system privileges. There are over 60 distinct system privileges.

Granting and Revoking System Privileges

You can grant or revoke system privileges to users and roles. If system privileges are granted to roles, the advantages of roles can be used to manage system privileges (for example, roles permit privileges to be made selectively available).

System privileges are granted to or revoked from users and roles using either of the following:

- the Users or the Roles folders of Server Manager
- the SQL commands GRANT and REVOKE

Who Can Grant or Revoke System Privileges?

Only users granted a specific system privilege with the ADMIN OPTION or users with the GRANT ANY PRIVILEGE system privilege (typically database or security administrators) can grant or revoke system privileges to other users.

Object Privileges

An object privilege is a privilege or right to perform a particular action on a *specific* table, view, sequence, procedure, function, or package. For example, the privilege to delete rows from the table DEPT is an object privilege. Depending on the type of object, there are different types of object privileges.

Object privileges granted for a table, view, sequence, procedure, function, or package apply whether referencing the base object by name or using a synonym. For example, assume there is a table JWARD.EMP with a synonym named JWARD.EMPLOYEE. JWARD issues the following statement:

```
GRANT SELECT ON emp TO swilliams;
```

The user SWILLIAMS can query JWARD.EMP by referencing the table by name or using the synonym JWARD.EMPLOYEE:

```
SELECT * FROM jward.emp;
SELECT * FROM jward.employee;
```

If you grant object privileges on a table, view, sequence, procedure, function, or package to a synonym for the object, the effect is the same as if no synonym were used For example, if JWARD wanted to grant the SELECT privilege for the EMP table to SWILLIAMS, JWARD could issue either of the following statements:

```
GRANT SELECT ON emp TO swilliams;
GRANT SELECT ON employee TO swilliams;
```

If a synonym is dropped, all grants for the underlying object remain in effect, even if the privileges were granted by specifying the dropped synonym.

Granting and Revoking Object Privileges

Object privileges can be granted to and revoked from users and roles. If you grant object privileges to roles, you can make the privileges selectively available. Object privileges can be granted to, or revoked from, users and roles using the SQL commands GRANT and REVOKE, respectively.

Who Can Grant Object Privileges?

A user automatically has all object privileges for the objects contained in the schema that corresponds to the user's name -- in other words, the schema the user owns. A user can grant any object privilege on any object he or she owns to any other user or role. If the grant includes the GRANT OPTION (of the GRANT command), the grantee can further grant the object privilege to other users; otherwise, the grantee can only use the privilege but not grant it to other users.

Roles

Oracle provides for easy and controlled privilege management through roles. ***Roles*** **are named groups of related privileges that you grant to users or other roles. Roles are designed to ease the administration of end-user system and object privileges.**

These properties of roles allow for easier privilege management within a database:

- *Reduced privilege administration* Rather than explicitly granting the same set of privileges to several users, you can grant the privileges for a group of related users to a role, and then only the role needs to be granted to each member of the group.

- *Dynamic privilege management* If 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* You can selectively enable or disable the roles granted to a user. This allows specific control of a user's privileges in any given situation.

- *Application awareness* Because the data dictionary records which roles exist, you can design database applications to query the dictionary and automatically enable (and disable) selective roles when a user attempts to execute the application via a given username.

- *Application-specific security* You can protect role use with a password. Applications can be created specifically to enable a role when supplied the correct password. Users cannot enable the role if they do not know the password..

Common Uses for Roles

In general, you create a role to serve one of two purposes: to manage the privileges for a database application or to manage the privileges for a user group.

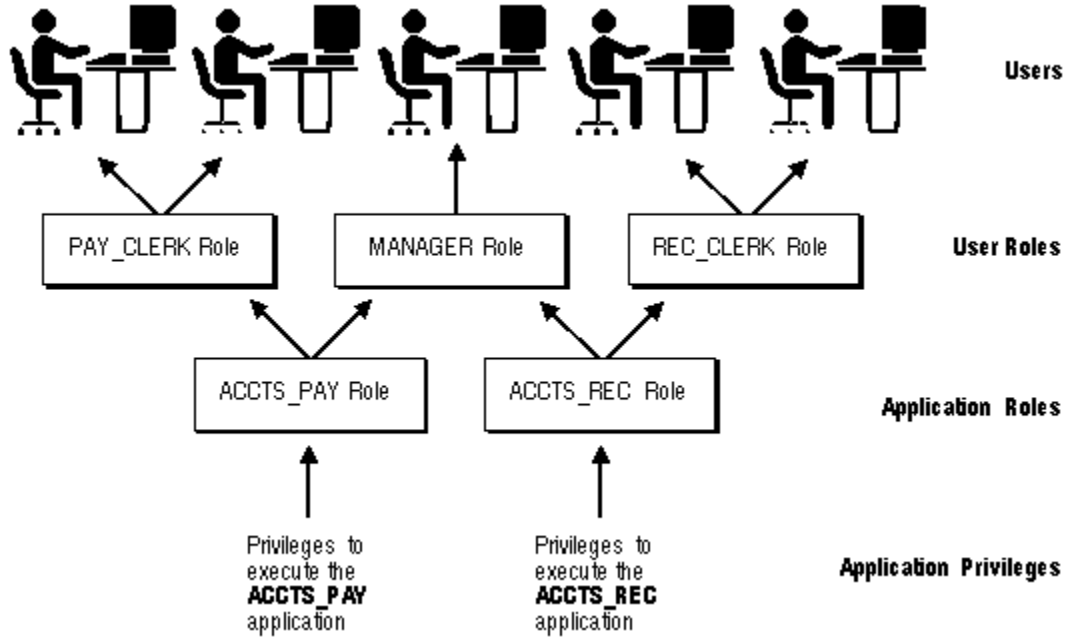

Figure 18 - 1. Common Uses for Roles