**Controlling User Access**

In a multiple-user environment, you want to maintain security of the database access and use. With Oracle Server database security, you can do the following:

* + - Control database access.
    - Give access to specific objects in the database.
    - Confirm given and received privileges with the Oracle data dictionary.

Database security can be classified into two categories**: system security** and **data security**. **System security** covers access and use of the database at the system level, such as the username and password, the disk space allocated to users, and the system operations that users can perform. **Database security** covers access and use of the database objects and the actions that those users can perform on the objects.

**Privileges**

A privilege is the right to execute particular SQL statements. The database administrator (DBA) is a high-level user with the ability to create users and grant users access to the database and its objects. Users require ***system privileges*** to gain access to the database and ***object privileges*** to manipulate the content of the objects in the database. Users can also be given the privilege to grant additional privileges to other users or to *roles*, which are named groups of related privileges.

**Schemas**

A *schema* is a collection of objects such as tables, views, and sequences. The schema is owned by a database user and has the same name as that user.

A system privilege is the right to perform a particular action, or to perform an action on any schema objects of a particular type. An object privilege provides the user the ability to perform a particular action on a specific schema object.

**System Privileges**

More than 100 distinct system privileges are available for users and roles. Typically, system privileges are provided by the database administrator (DBA).

**Typical DBA Privileges**

**Creating Users**

The DBA creates the user by executing the CREATE USER statement. The user does not have any privileges at this point. The DBA can then grant privileges to that user. These privileges determine what the user can do at the database level.

The slide gives the abridged syntax for creating a user.

In the syntax:

*user* Is the name of the user to be created

*Password* Specifies that the user must log in with this password

**Typical User Privileges**

After the DBA creates a user, the DBA can assign privileges to that user.

In the syntax:

*privilege* Is the system privilege to be granted

*user* |role|PUBLIC Is the name of the user, the name of the role, or PUBLIC (which designates that every user is granted the privilege)

**Note:** Current system privileges can be found in the SESSION\_PRIVS dictionary view. Data dictionary is a collection of tables and views created and maintained by the Oracle Server. They contain information about the database.

**Granting System Privileges**

The DBA uses the GRANT statement to allocate system privileges to the user. After the user has been granted the privileges, the user can immediately use those privileges.

In the example in the slide, the demo user has been assigned the privileges to create sessions, tables, sequences, and views.

**What Is a Role?**

A role is a named group of related privileges that can be granted to the user. This method makes it easier to revoke and maintain privileges.

A user can have access to several roles, and several users can be assigned the same role. Roles are typically created for a database application.

**Creating and Assigning a Role**

First, the DBA must create the role. Then the DBA can assign privileges to the role and assign the role to users.

**Syntax**

CREATE ROLE *role*;

In the syntax:

*role* Is the name of the role to be created

After the role is created, the DBA can use the GRANT statement to assign the role to users as well as assign privileges to the role. A role is not a schema object, therefore any user can add privileges to a role.

**Creating a Role**

The example in the slide creates a manager role and then enables the manager to create tables and views. It then grants user alice the role of a manager. Now alice can create tables and views.

If users have multiple roles granted to them, they receive all the privileges associated with all the roles.

**Changing Your Password**

The DBA creates an account and initializes a password for every user. You can change your password by using the ALTER USER statement.

The slide example shows that the demo user changes the password by using the ALTER USER statement.

**Syntax**

ALTER USER user IDENTIFIED BY password;

In the syntax:

*user* Is the name of the user

*password* Specifies the new password

Although this statement can be used to change your password, there are many other options. You must have the ALTER USER privilege to change any other option.

**Note:**SQL\*Plus has a PASSWORD command (PASSW) that can be used to change the password of a user when the user is logged in. This command is not available in SQL Developer.

**Object Privileges**

An *object privilege* is a privilege or right to perform a particular action on a specific table, view, sequence, or procedure. Each object has a particular set of grantable privileges. The table in the slide lists the privileges for various objects. Note that the only privileges that apply to a sequence are SELECT and ALTER. UPDATE, REFERENCES, and INSERT can be restricted by specifying a subset of updatable columns.

A SELECT privilege can be restricted by creating a view with a subset of columns and granting the SELECT privilege only on the view. A privilege granted on a synonym is converted to a privilege on the base table referenced by the synonym.

**Note:** With the REFERENCES privilege, you can ensure that other users can create FOREIGN KEY constraints that reference your table.

**Granting Object Privileges**

Different object privileges are available for different types of schema objects. A user automatically has all object privileges for schema objects contained in the user’s schema. A user can grant any object privilege on any schema object that the user owns to any other user or role. If the grant includes WITH GRANT OPTION, the grantee can further grant the object privilege to other users; otherwise, the grantee can use the privilege but cannot grant it to other users.

In the syntax:

*object\_priv*  Is an object privilege to be granted

ALL Specifies all object privileges

*columns* Specifies the column from a table or view on which  
 privileges are granted

ON *object* Is the object on which the privileges are granted

TO Identifies to whom the privilege is granted

PUBLIC Grants object privileges to all users

WITH GRANT OPTION Enables the grantee to grant the object privileges to other users and roles

**Note:** In the syntax, *schema* is the same as the owner’s name.

**Guidelines**

* + - To grant privileges on an object, the object must be in your own schema, or you must have been granted the object privileges WITH GRANT OPTION.
    - An object owner can grant any object privilege on the object to any other user or role of the database.
    - The owner of an object automatically acquires all object privileges on that object.

The first example in the slide grants the demo user the privilege to query your EMPLOYEES table. The second example grants UPDATE privileges on specific columns in the DEPARTMENTS table to demo and to the manager role.

For example, if your schema is ora*xx*, and the demo user now wants to use a SELECT statement to obtain data from your EMPLOYEES table, the syntax he or she must use is:

SELECT \* FROM oraxx.employees;

Alternatively, the demo user can create a synonym for the table and issue a SELECT statement from the synonym:

CREATE SYNONYM emp FOR oraxx.employees;

SELECT \* FROM emp;

**Note:** DBAs generally allocate system privileges; any user who owns an object can grant object privileges.

**Passing On Your Privileges**

**WITH GRANT OPTION Keyword**

A privilege that is granted with the WITH GRANT OPTION clause can be passed on to other users and roles by the grantee. Object privileges granted with the WITH GRANT OPTION clause are revoked when the grantor’s privilege is revoked.

The example in the slide gives the demo user access to your DEPARTMENTS table with the privileges to query the table and add rows to the table. The example also shows that user1 can give others these privileges.

**PUBLIC Keyword**

An owner of a table can grant access to all users by using the PUBLIC keyword.

The second example allows all users on the system to query data from Alice’s DEPARTMENTS table.

**Confirming Granted Privileges**

If you attempt to perform an unauthorized operation, such as deleting a row from a table for which you do not have the DELETE privilege, the Oracle server does not permit the operation to take place.

If you receive the Oracle server error message “Table or view does not exist,” you have done either of the following:

* + - Named a table or view that does not exist
    - Attempted to perform an operation on a table or view for which you do not have the appropriate privilege

The data dictionary is organized in tables and views and contains information about the database. You can access the data dictionary to view the privileges that you have. The table in the slide describes various data dictionary views.

You learn more about data dictionary views in the lesson titled “Managing Objects with Data Dictionary Views.”

**Note:** The ALL\_TAB\_PRIVS\_MADE dictionary view describes all the object grants made by the user or made on the objects owned by the user.

**Revoking Object Privileges**

You can remove privileges granted to other users by using the REVOKE statement. When you use the REVOKE statement, the privileges that you specify are revoked from the users you name and from any other users to whom those privileges were granted by the revoked user.

In the syntax:

CASCADE Is required to remove any referential integrity constraints made to the CONSTRAINTS object by means of the REFERENCES privilege

For more information, see the *Oracle Database11g SQL Reference*.

**Note:** If a user were to leave the company and you revoke his or her privileges, you must regrant any privileges that this user may have granted to other users. If you drop the user account without revoking privileges from it, the system privileges granted by this user to other users are not affected by this action.

**Revoking Object Privileges (continued)**

The example in the slide revokes SELECT and INSERT privileges given to the demo user on the DEPARTMENTS table.

**Note:** If a user is granted a privilege with the WITH GRANT OPTION clause, that user can also grant the privilege with the WITH GRANT OPTION clause, so that a long chain of grantees is possible, but no circular grants (granting to a grant ancestor) are permitted. If the owner revokes a privilege from a user who granted the privilege to other users, the revoking cascades to all the privileges granted.

For example, if user A grants a SELECT privilege on a table to user B including the WITH GRANT OPTION clause, user B can grant to user C the SELECT privilege with the WITH GRANT OPTION clause as well, and user C can then grant to user D the SELECT privilege. If user A revokes privileges from user B, the privileges granted to users C and D are also revoked.