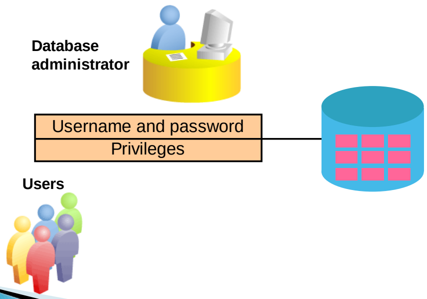
Controlling User Access

Objectives

- After completing this lesson, you should be able to do the following:
  - Differentiate system privileges from object privileges
  - Grant privileges on tables
  - Grant roles
  - Distinguish between privileges and roles

Controlling User Access



Database administrator

Username and password

Privileges

Users

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.
  - Create synonyms for database objects.

Database security

- 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 have on the objects.

Privileges

- Privileges are 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.
- System privileges: Gaining access to the database
- Object privileges: Manipulating the content of the database objects

System Privileges

- More than 100 privileges are available.
- The database administrator has high-level system privileges for tasks such as:
  - Creating new users
  - Removing users
  - Removing tables
  - Backing up tables

Creating Users

- The DBA creates a 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.

Creating Users

- The DBA creates users with the CREATE USER statement.

```
CREATE USER user
IDENTIFIED BY   password;
```

```
CREATE USER  USER1
IDENTIFIED BY   USER1;
CREATE USER succeeded.
```

  - In the syntax:

user	Is the name of the user to be created
Password	Specifies that the user must log in with this password

User System Privileges

- After a user is created, the DBA can grant specific system privileges to that user.

```
GRANT privilege [, privilege...]
TO user [, user| role, PUBLIC...];
```

  - An application developer, for example, may have the following system privileges:
    - CREATE SESSION
    - CREATE TABLE
    - CREATE SEQUENCE
    - CREATE VIEW
    - CREATE PROCEDURE

Granting System Privileges

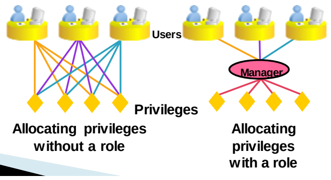
- The DBA uses the GRANT statement to allocate system privileges to the user. After the user has been granted the privileges, the user can immediately use those privileges.

```
GRANT  create session, create table,
       create sequence, create view
TO     scott;
GRANT CREATE succeeded.
```

- In the example in the slide, user Scott 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.


Users
Manager
Privileges
Allocating privileges without a role
Allocating privileges with a role

Creating and Granting Privileges to a Role

- First, the DBA must create the role. Then the DBA can assign privileges to the role and assign the role to users.
  - Create a role:

```
CREATE ROLE manager;
CREATE ROLE succeeded.
```

  - Grant privileges to a role:

```
GRANT create table, create view
TO manager;
GRANT succeeded.
```

  - Grant a role to users:

```
GRANT manager TO BELL, KOCHHAR;
GRANT succeeded.
```

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.

- Lists of 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.

Object Privileges

Object Privilege	Table	View	Sequence	Procedure
ALTER	√		√	
DELETE	√	√		
EXECUTE				√
INDEX	√			
INSERT	√	√		
REFERENCES	√			
SELECT	√	√	√	
UPDATE	√	√		

Object Privileges

- Object privileges vary from object to object.
- An owner has all the privileges on the object.
- An owner can give specific privileges on that owner's object.

```
GRANT          object_priv [(columns)]
ON             object
TO             {user|role|PUBLIC}
[WITH GRANT OPTION];
```

- 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

Granting Object Privileges

- Grant query privileges on the EMPLOYEES table:

```
GRANT  select
ON     employees
TO     sue, rich;
GRANT succeeded.
```

- In example grants users Sue and Rich the privilege to query your EMPLOYEES table.

- If Sue or Rich now want to use a SELECT statement to obtain data from the EMPLOYEES table, the syntax they must use is:

  - SELECT * FROM HR.employees;

- Grant privileges to update specific columns to users and roles:

```
GRANT  update (department_name, location_id)
ON     departments
TO     scott, manager;
GRANT  succeeded.
```

- In example grants UPDATE privileges on specific columns in the DEPARTMENTS table to Scott and the manager role.

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.

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.

```
GRANT  select, insert
ON     departments
TO     scott
WITH   GRANT OPTION;
GRANT succeeded.
```

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

Passing On Your Privileges

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

```
GRANT  select
ON     alice.departments
TO     PUBLIC;
GRANT succeeded.
```

  - In example allows all users on the system to query data from Alice's `DEPARTMENTS` table.

Confirming Privileges Granted

- 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," then 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
- You can access the data dictionary to view the privileges that you have.

Confirming Privileges Granted

- The chart describes various data dictionary views.

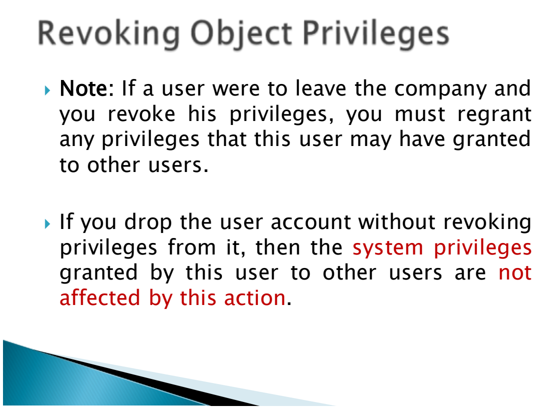
Data Dictionary View	Description
ROLE_SYS_PRIVS	System privileges granted to roles
ROLE_TAB_PRIVS	Table privileges granted to roles
USER_ROLE_PRIVS	Roles accessible by the user
USER_TAB_PRIVS_MADE	Object privileges granted on the user's objects
USER_TAB_PRIVS_RECD	Object privileges granted to the user
USER_COL_PRIVS_MADE	Object privileges granted on the columns of the user's objects
USER_COL_PRIVS_RECD	Object privileges granted to the user on specific columns
USER_SYS_PRIVS	System privileges granted to the user

Revoking Object Privileges

- You use the `REVOKE` statement to revoke privileges granted to other users.
- Privileges granted to others through the `WITH GRANT OPTION` clause are also revoked.

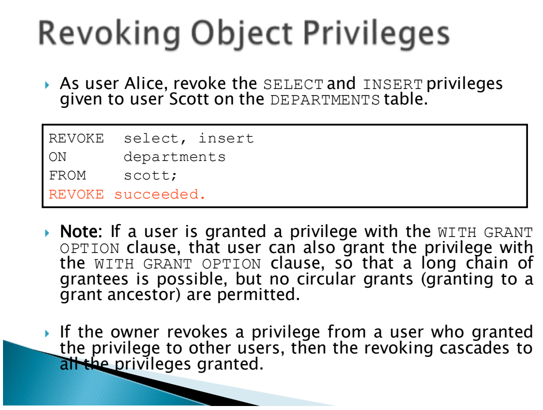
```
REVOKE {privilege [, privilege...]|ALL}
ON     object
FROM   {user[, user...]|role|PUBLIC}
[CASCADE CONSTRAINTS];
```

- Note:
- `CASCADE` is required to remove any referential integrity constraints made to the `CONSTRAINTS` object by means of the `REFERENCES` privilege



Revoking Object Privileges

- **Note**: If a user were to leave the company and you revoke his 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, then the system privileges granted by this user to other users are not affected by this action.

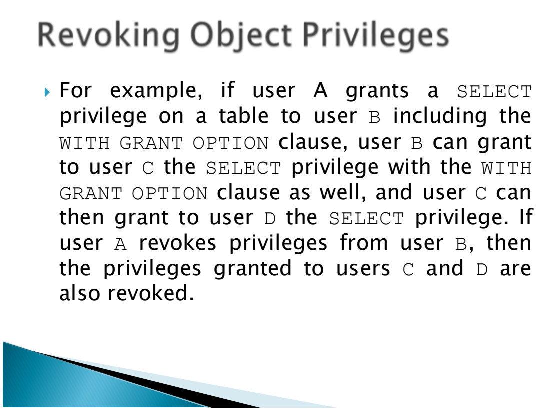


Revoking Object Privileges

- As user Alice, revoke the `SELECT` and `INSERT` privileges given to user Scott on the `DEPARTMENTS` table.

```
REVOKE  select, insert
ON      departments
FROM    scott;
REVOKE succeeded.
```

- **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, then the revoking cascades to all the privileges granted.



Revoking Object Privileges

- 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`, then the privileges granted to users `C` and `D` are also revoked.