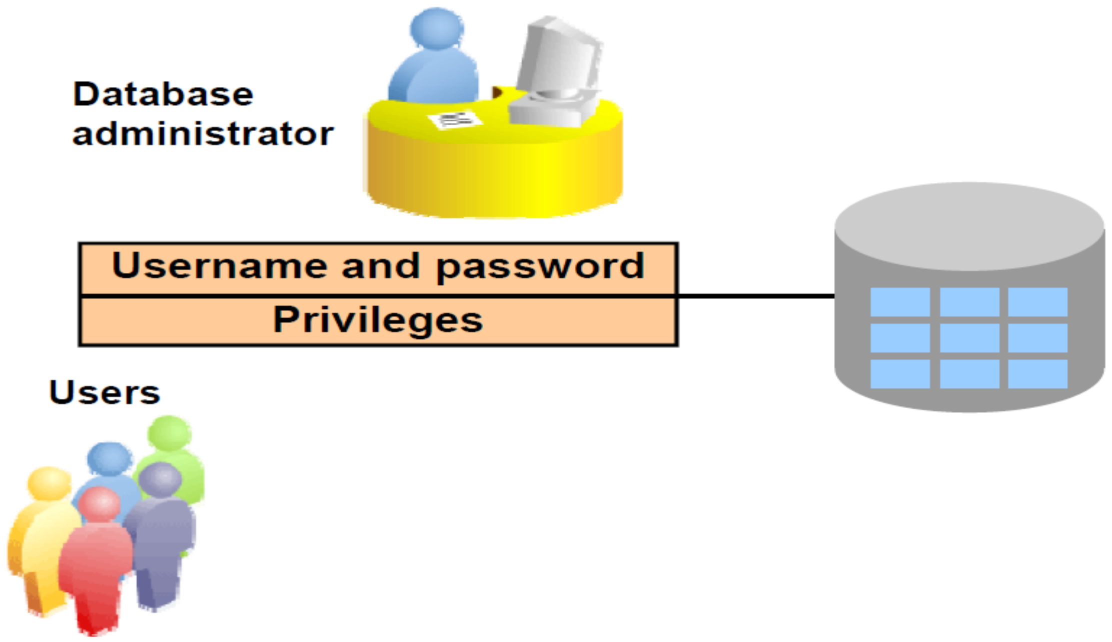
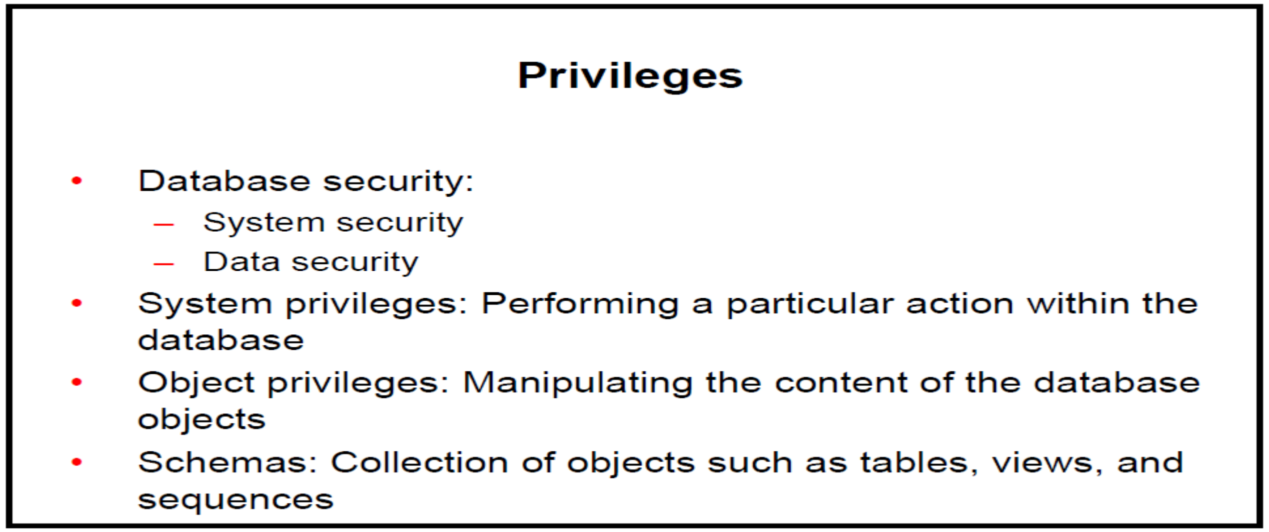
# Controlling User Access

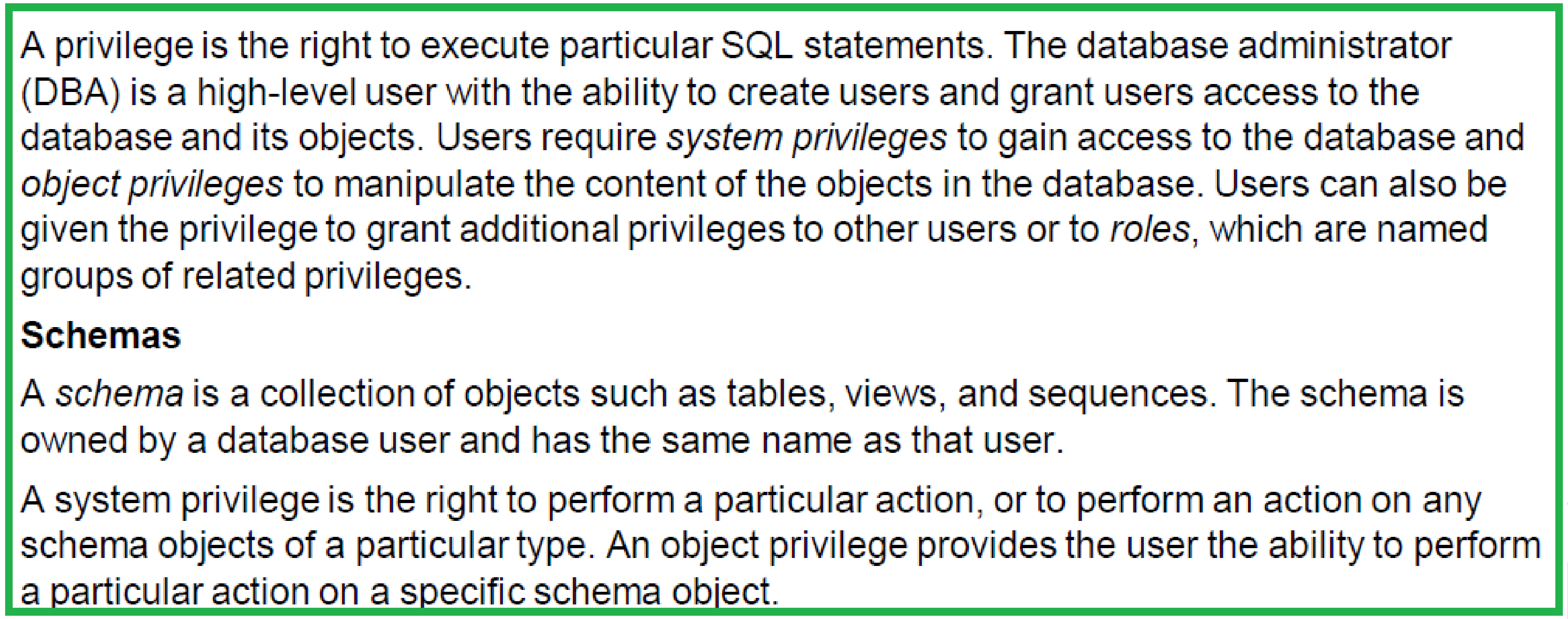
## Controlling User Access





## Privileges

- Database security:
  - System security
  - Data security
- System privileges: Performing a particular action within the database
- Object privileges: Manipulating the content of the database objects
- Schemas: Collection of objects such as tables, views, and sequences



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 200 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

More than 200 distinct system privileges are available for users and roles. Typically, system privileges are provided by the database administrator (DBA).

The table `SYSTEM_PRIVILEGE_MAP` contains all the system privileges available, based on the version release. This table is also used to map privilege type numbers to type names.

**Typical DBA Privileges**

| System Privilege | Operations Authorized |
|---|---|
| `CREATE USER` | Grantee can create other Oracle users. |
| `DROP USER` | Grantee can drop another user. |
| `DROP ANY TABLE` | Grantee can drop a table in any schema. |
| `BACKUP ANY TABLE` | Grantee can back up any table in any schema with the export utility. |
| `SELECT ANY TABLE` | Grantee can query tables, views, or materialized views in any schema. |
| `CREATE ANY TABLE` | Grantee can create tables in any schema. |

## Creating Users

The DBA creates users with the `CREATE USER` statement.

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

```
CREATE USER  demo
IDENTIFIED BY demo;
```

## 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`

**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.

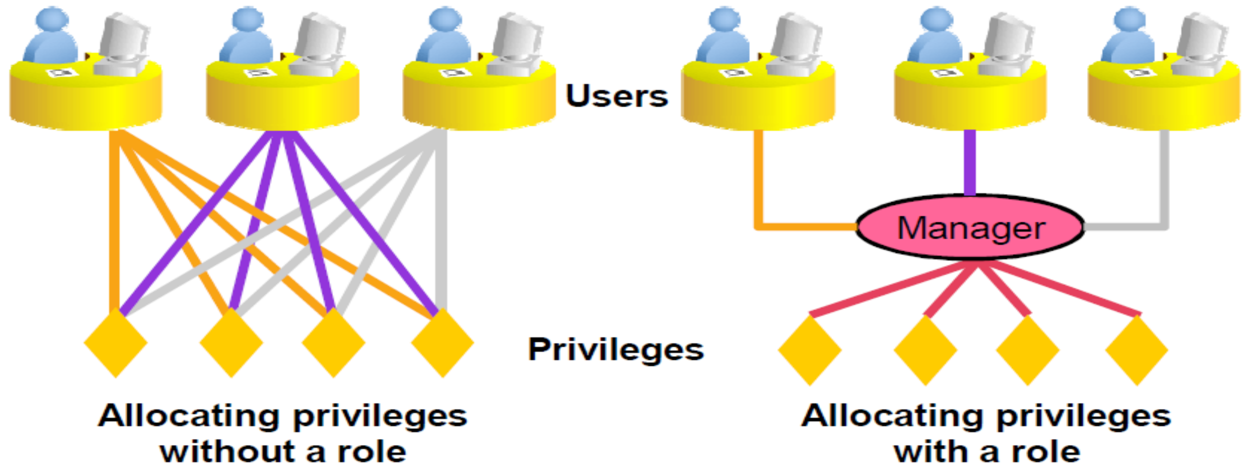
## Granting System Privileges

The DBA can grant specific system privileges to a user.

```
GRANT  create session, create table,
       create sequence, create view
TO     demo;
```

```
GRANT succeeded.
```

## What Is a Role?


Users
Manager
Privileges
Allocating privileges without a role
Allocating privileges with 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 Granting Privileges to a Role

- Create a role:

```
CREATE ROLE manager;
```

- Grant privileges to a role:

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

- Grant a role to users:

```
GRANT manager TO alice;
```

## Changing Your Password

- The DBA creates your user account and initializes your password.
- You can change your password by using the `ALTER USER` statement.

```
ALTER USER demo
IDENTIFIED BY employ;
```

## Object Privileges

| Object privilege | Table | View | Sequence |
|---|---|---|---|
| ALTER | ✓ | | ✓ |
| DELETE | ✓ | ✓ | |
| INDEX | ✓ | | |
| INSERT | ✓ | ✓ | |
| REFERENCES | ✓ | | |
| SELECT | ✓ | ✓ | ✓ |
| UPDATE | ✓ | ✓ | |

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

## 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];
```

**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 |

## Granting Object Privileges

- Grant query privileges on the EMPLOYEES table:

```
GRANT   select
ON      employees
TO      demo;
```

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

```
GRANT   update (department_name, location_id)
ON      departments
TO      demo, manager;
```

**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 oraxx, 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

- Give a user authority to pass along privileges:

```
GRANT  select, insert
ON     departments
TO     demo
WITH   GRANT OPTION;
```

- Allow all users on the system to query data from `DEPARTMENTS` table:

```
GRANT  select
ON    departments
TO    PUBLIC;
```

## Confirming Granted Privileges

| 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_SYS_PRIVS | System privileges granted to 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 |

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 about data dictionary views in the lesson titled "Introduction to 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 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];
```

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 |

## Revoking Object Privileges

Revoke the SELECT and INSERT privileges given to the demo user on the DEPARTMENTS table.

```
REVOKE   select, insert
ON       departments
FROM     demo;
```

```
REVOKE succeeded.
```

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.

## Thank You