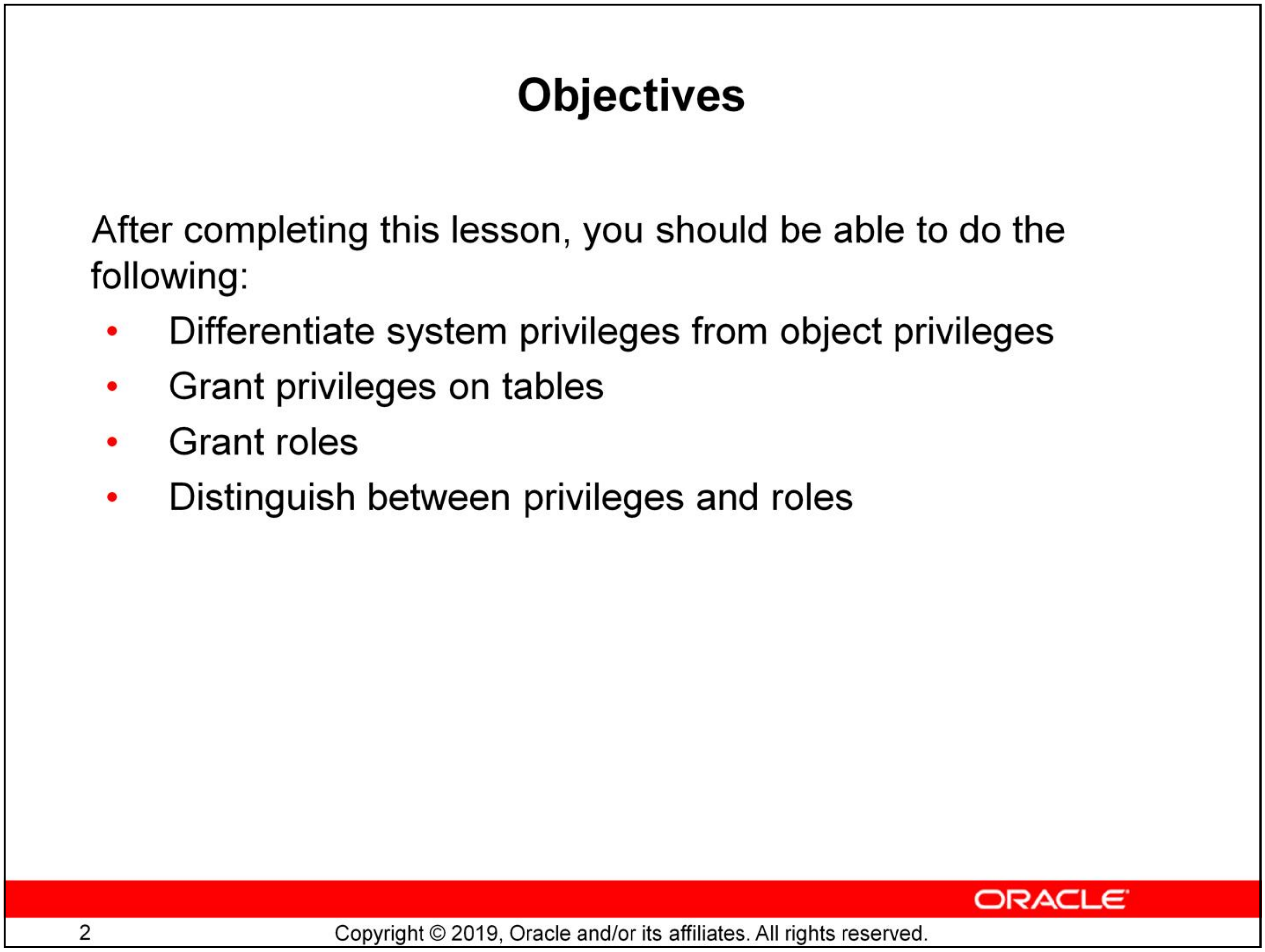
Controlling User Access



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

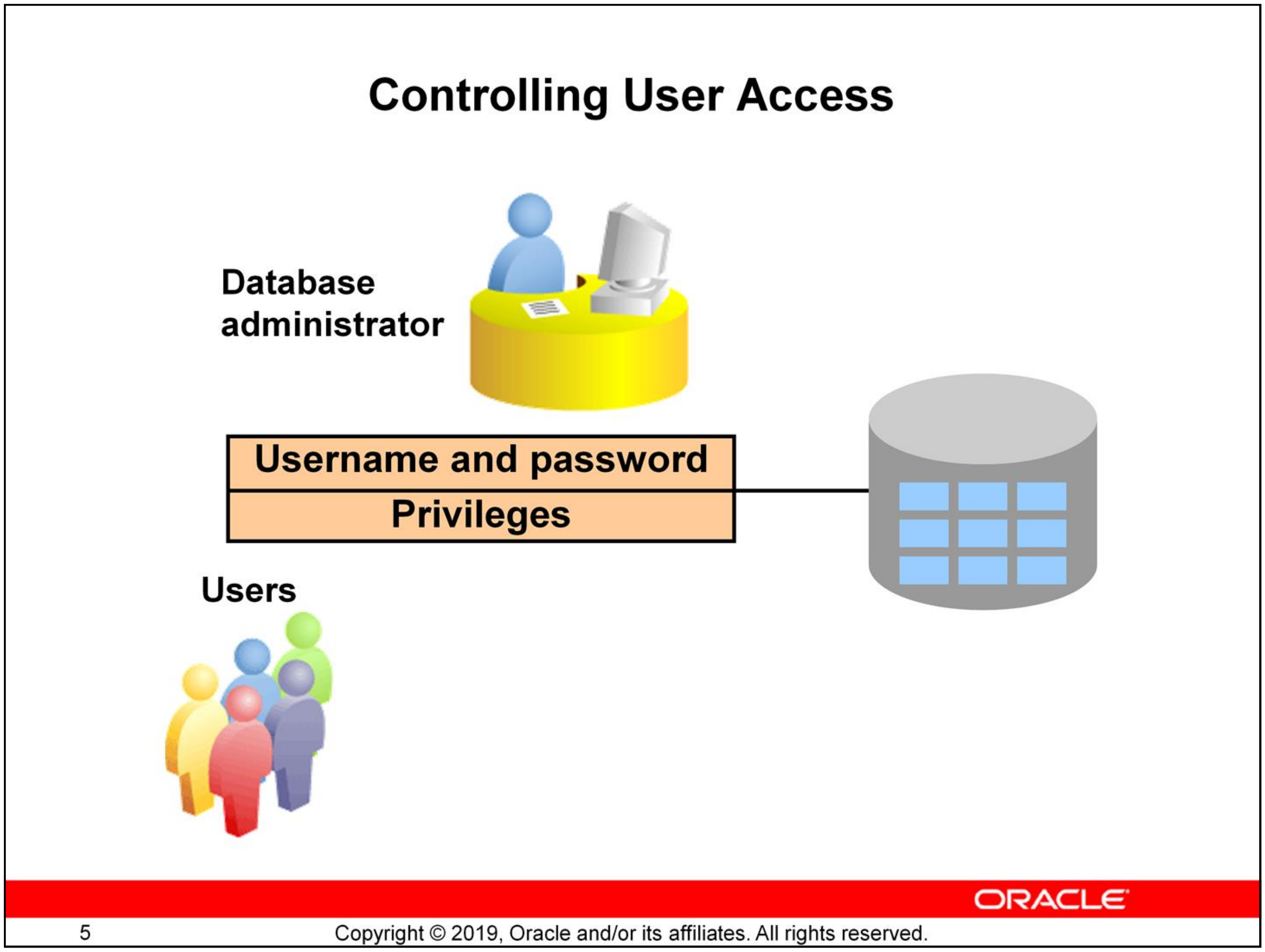
In this lesson, you learn how to control database access to specific objects and add new users with different levels of access privileges.

Lesson Agenda

- System privileges
- Creating a role
- Object privileges
- Revoking object privileges

Controlling User Access

- In a multiple-user environment, you want to maintain security of 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 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 actions that those users can perform on the objects.



Controlling User Access

Database administrator

Username and password

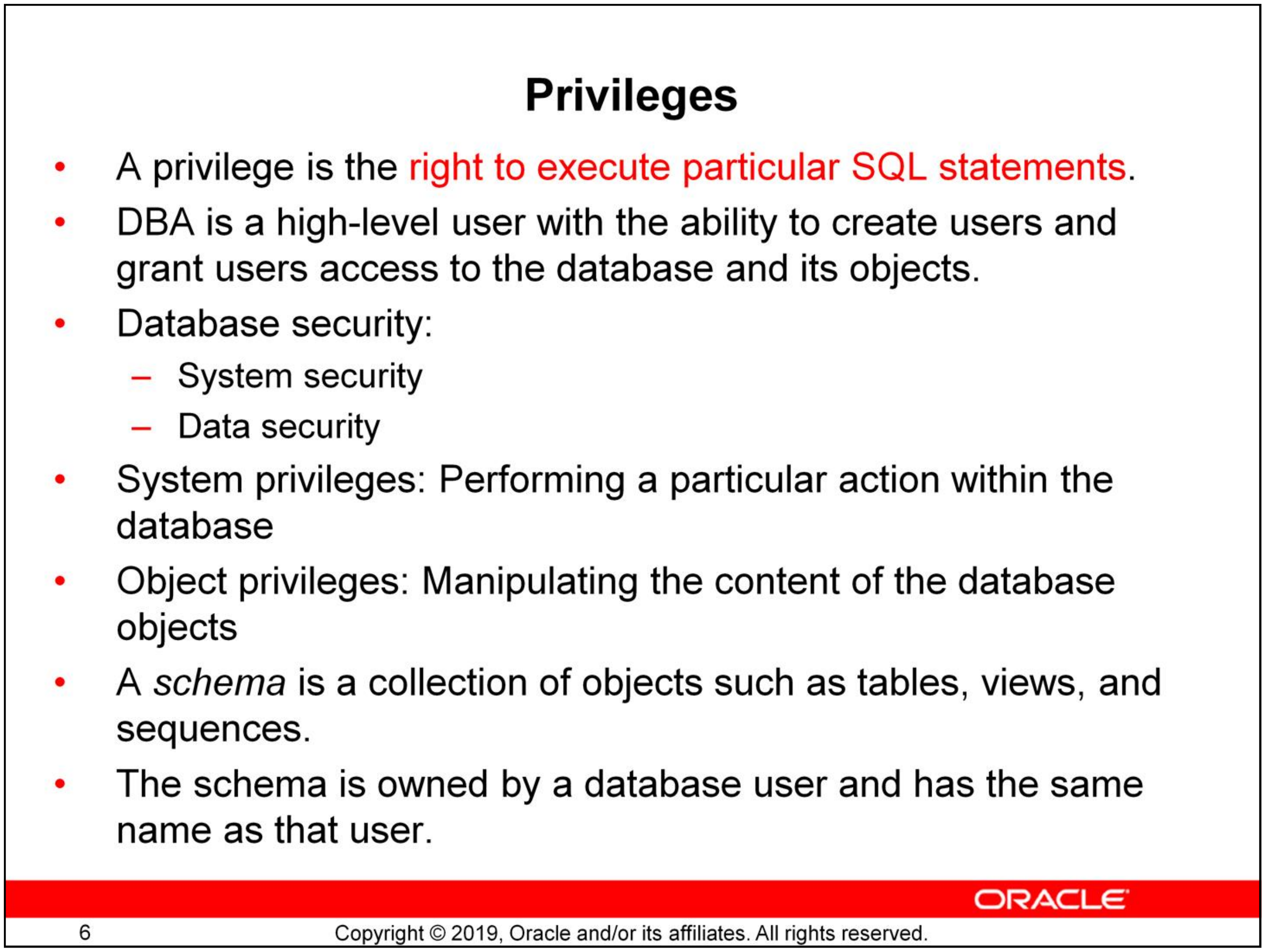
Privileges

Users

In a multiple-user environment, you want to maintain security of 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.
- DBA is a high-level user with the ability to create users and grant users access to the database and its objects.
- Database security:
  - System security
  - Data security
- System privileges: Performing a particular action within the database
- Object privileges: Manipulating the content of the database objects
- 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.

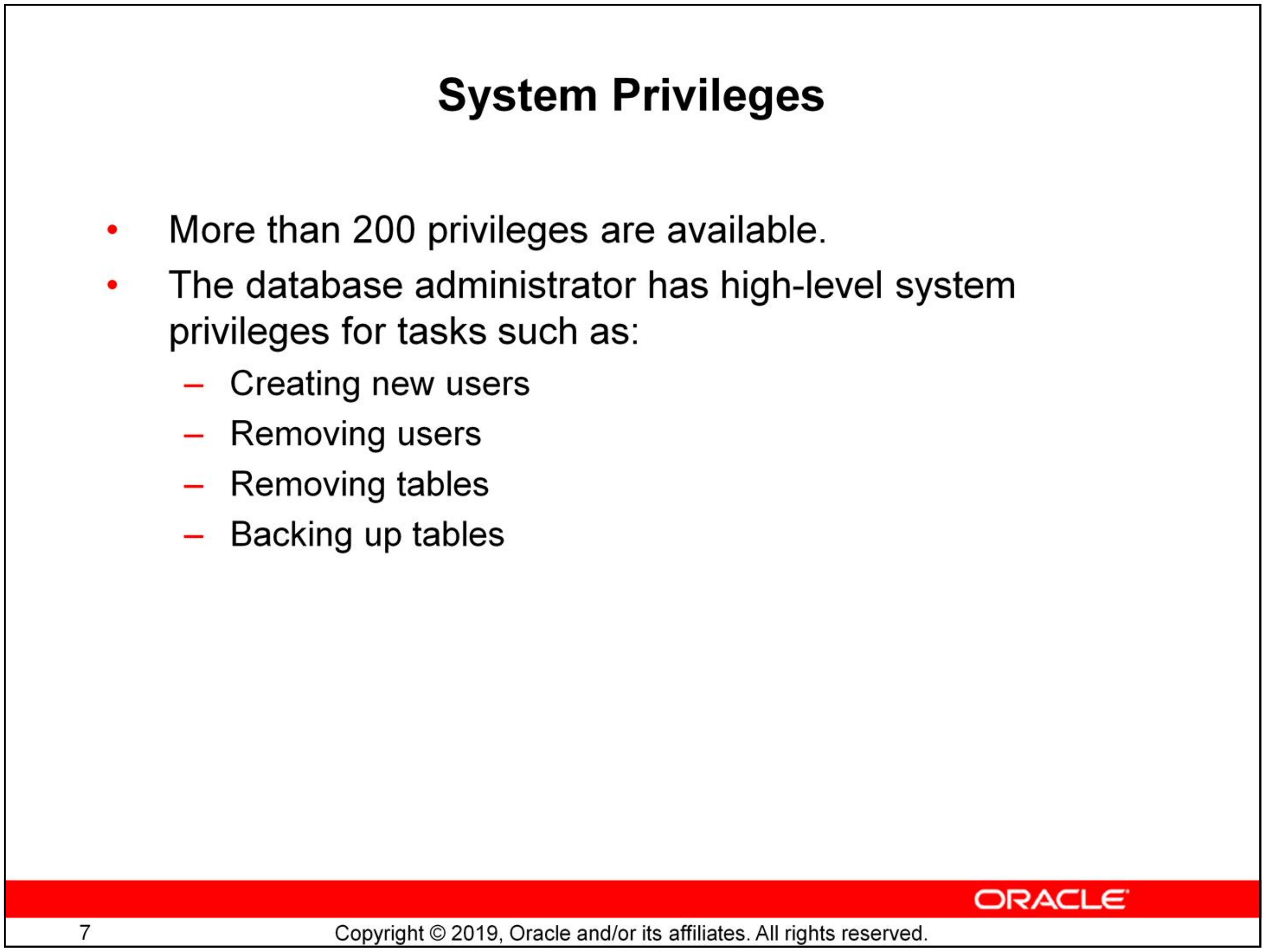
A privilege is the right to execute particular SQL statements. The database administrator (DBA) is a high-level user with the ability to create users and grant users access to the database and its objects. Users require *system privileges* to gain access to the database and *object privileges* to manipulate 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.

For more information, see the *Oracle Database 2 Day DBA* reference manual for Oracle Database19c.



System Privileges

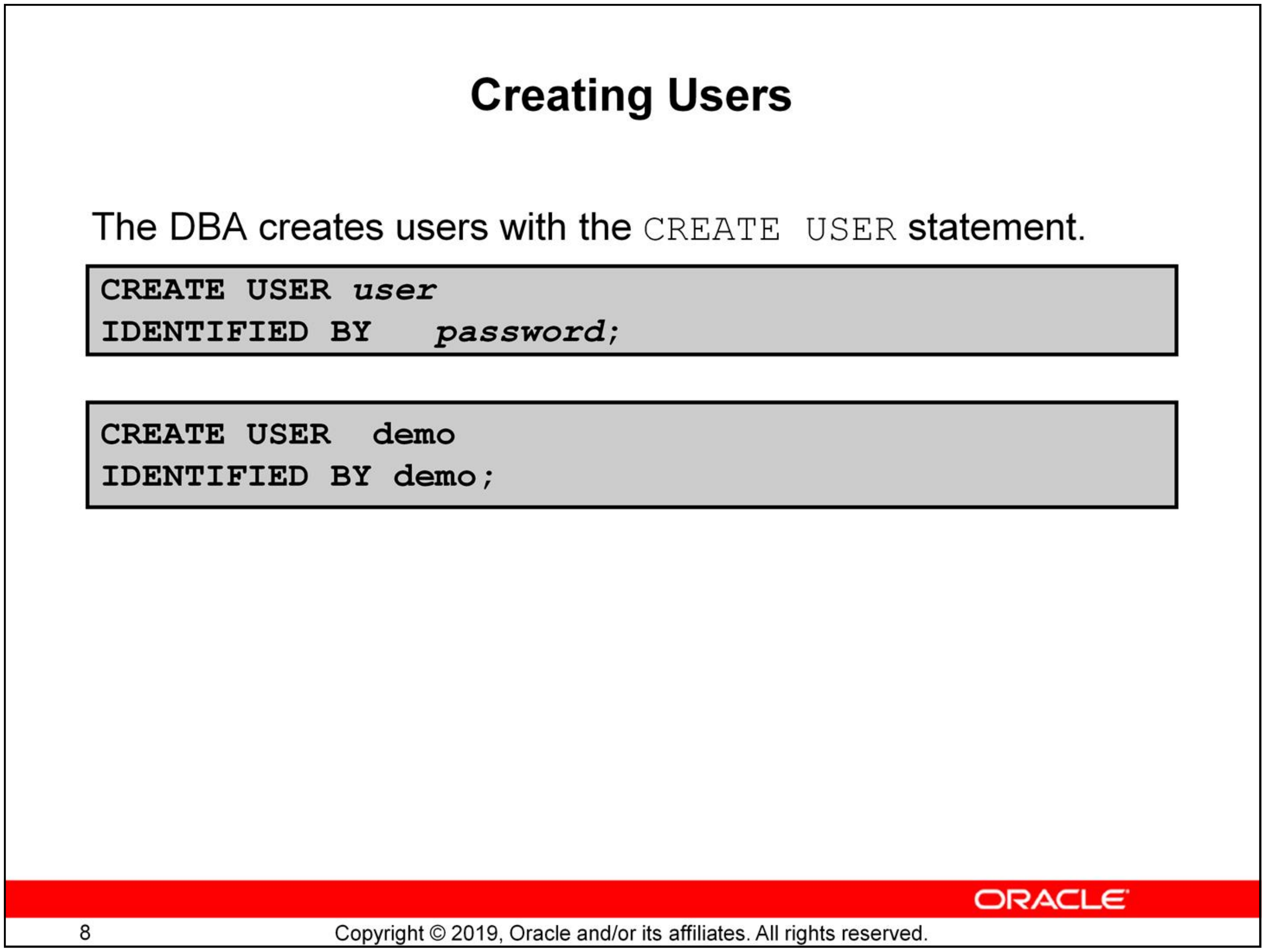
- More than 200 privileges are available.
- The database administrator has high-level system privileges for tasks such as:
  - Creating new users
  - Removing users
  - Removing tables
  - Backing up tables

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;
```

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.

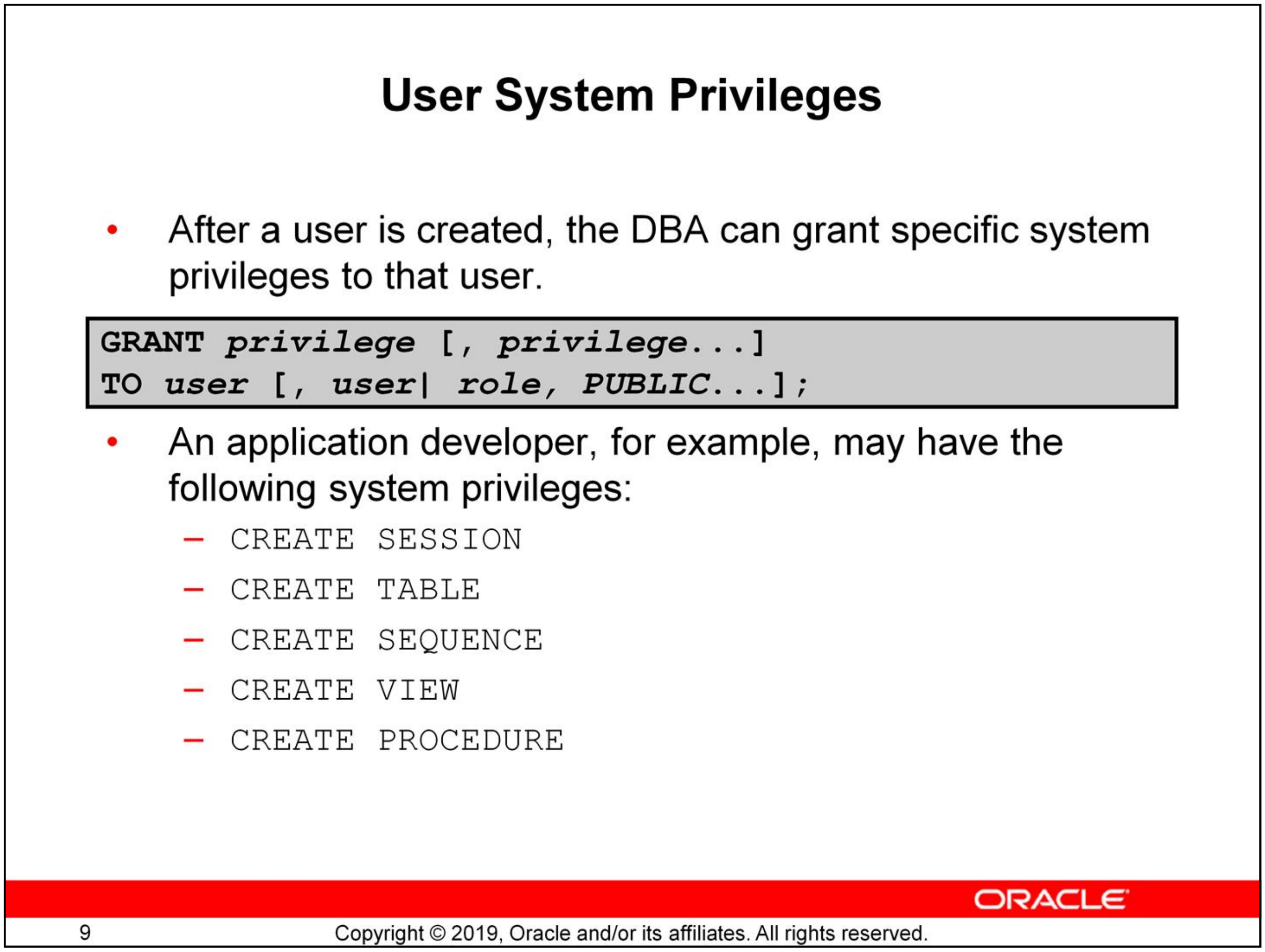
The slide gives the abridged syntax for creating a user.

In the syntax:

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

For more information, see the *Oracle Database SQL Language Reference* for Oracle Database19c.

Note: Starting with Oracle Database 11*g*, passwords are case-sensitive.



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`

Typical User Privileges

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

System Privilege	Operations Authorized
`CREATE SESSION`	Connect to the database.
`CREATE TABLE`	Create tables in the user’s schema.
`CREATE SEQUENCE`	Create a sequence in the user’s schema.
`CREATE VIEW`	Create a view in the user’s schema.
`CREATE PROCEDURE`	Create a stored procedure, function, or package in the user’s schema.

In the syntax:

privilege Is the system privilege to be granted

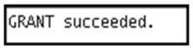
user |role|PUBLIC Is the name of the user, the name of the role, or PUBLIC (which designates that every user is granted the privilege)

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 can grant specific system privileges to a user.

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



```
GRANT succeeded.
```

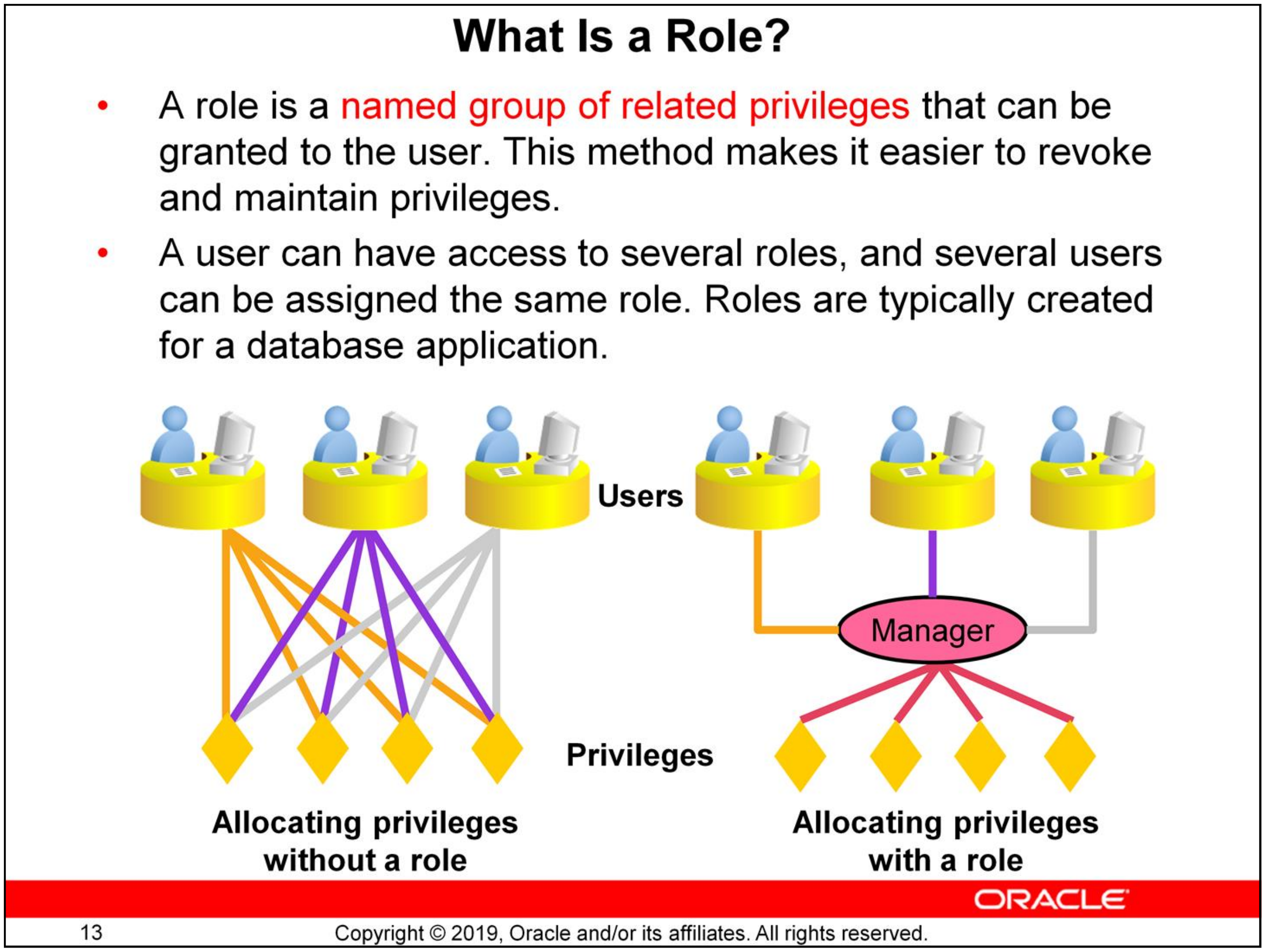
Current system privileges can be found in the `SESSION_PRIVS` dictionary view.

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.

Lesson Agenda

- System privileges
- Creating a role
- Object privileges
- Revoking object privileges



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

ORACLE

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.

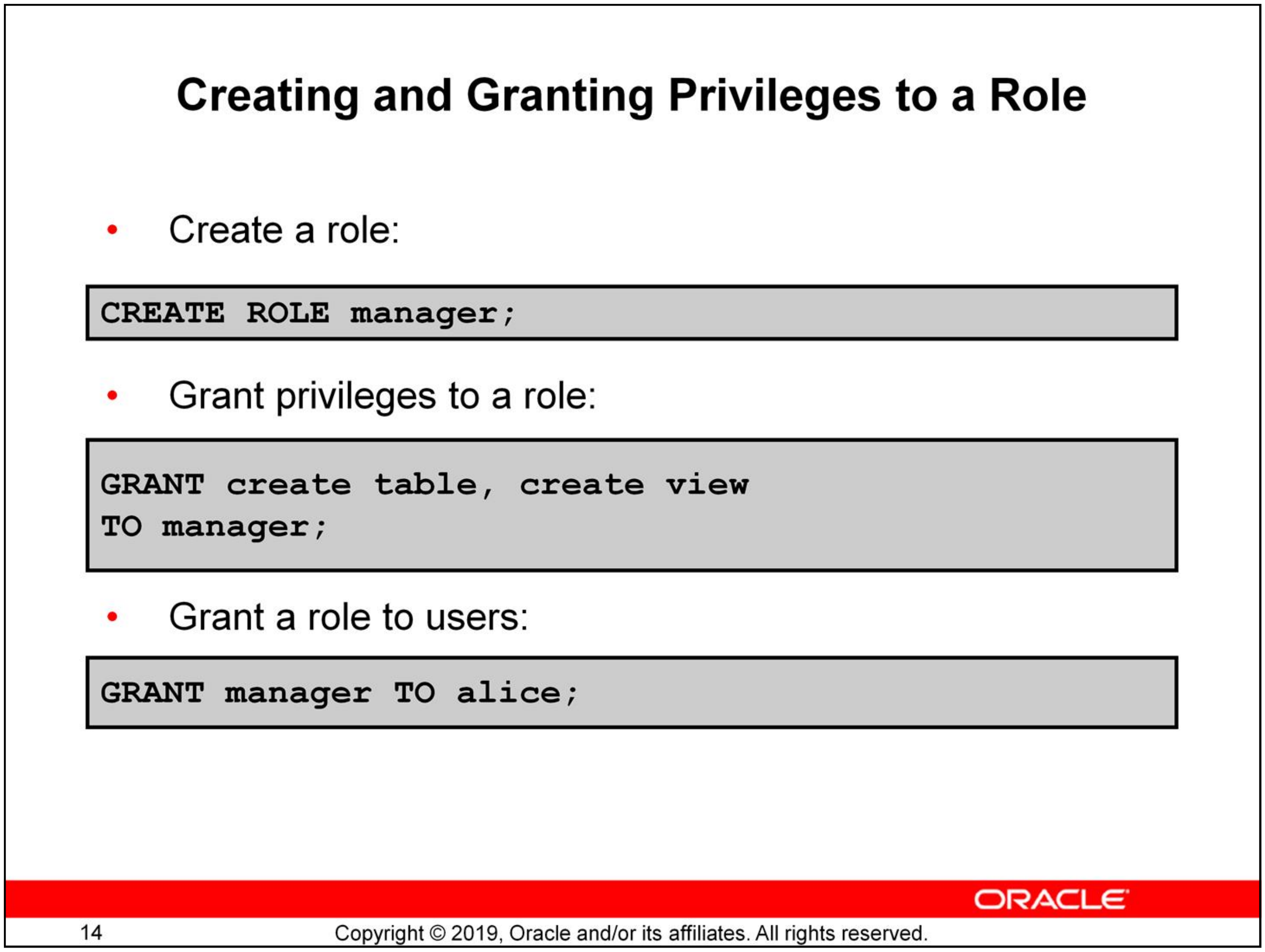
Syntax

```
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 and Granting Privileges to a Role

- Create a role:

```
CREATE ROLE manager;
```

- Grant privileges to a role:

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

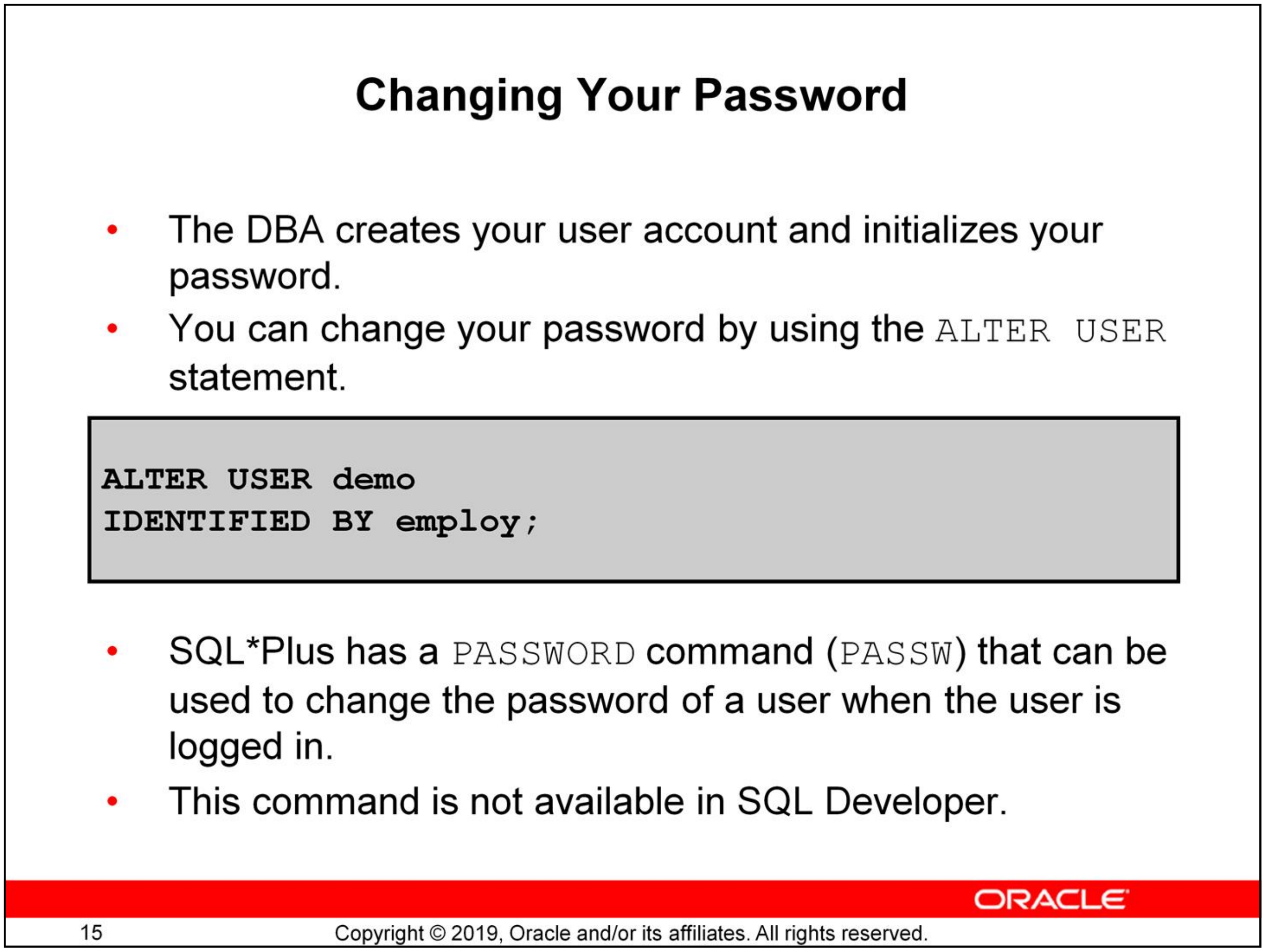
- Grant a role to users:

```
GRANT manager TO alice;
```

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

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

- 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.

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;
```

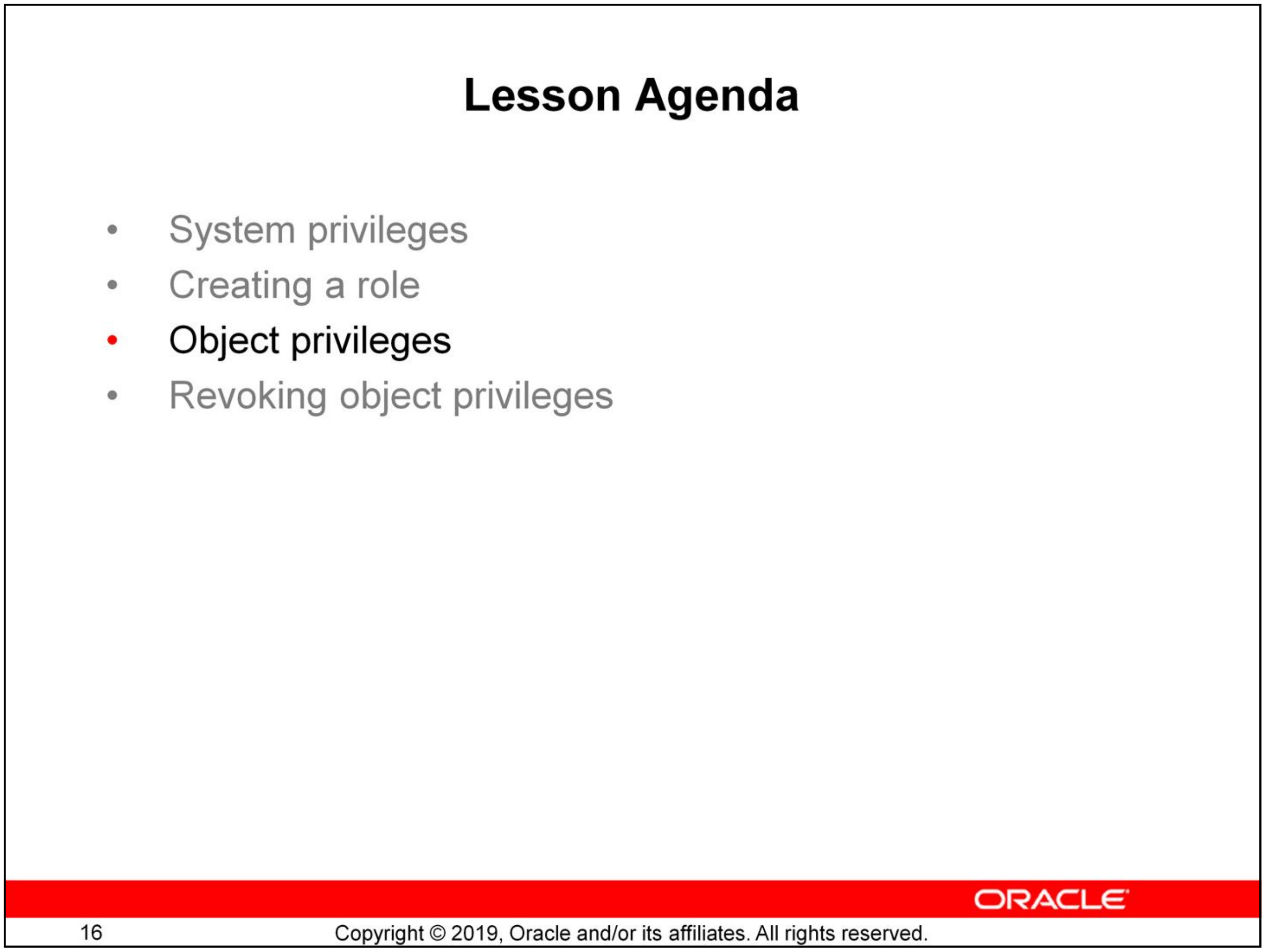
In the syntax:

`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.

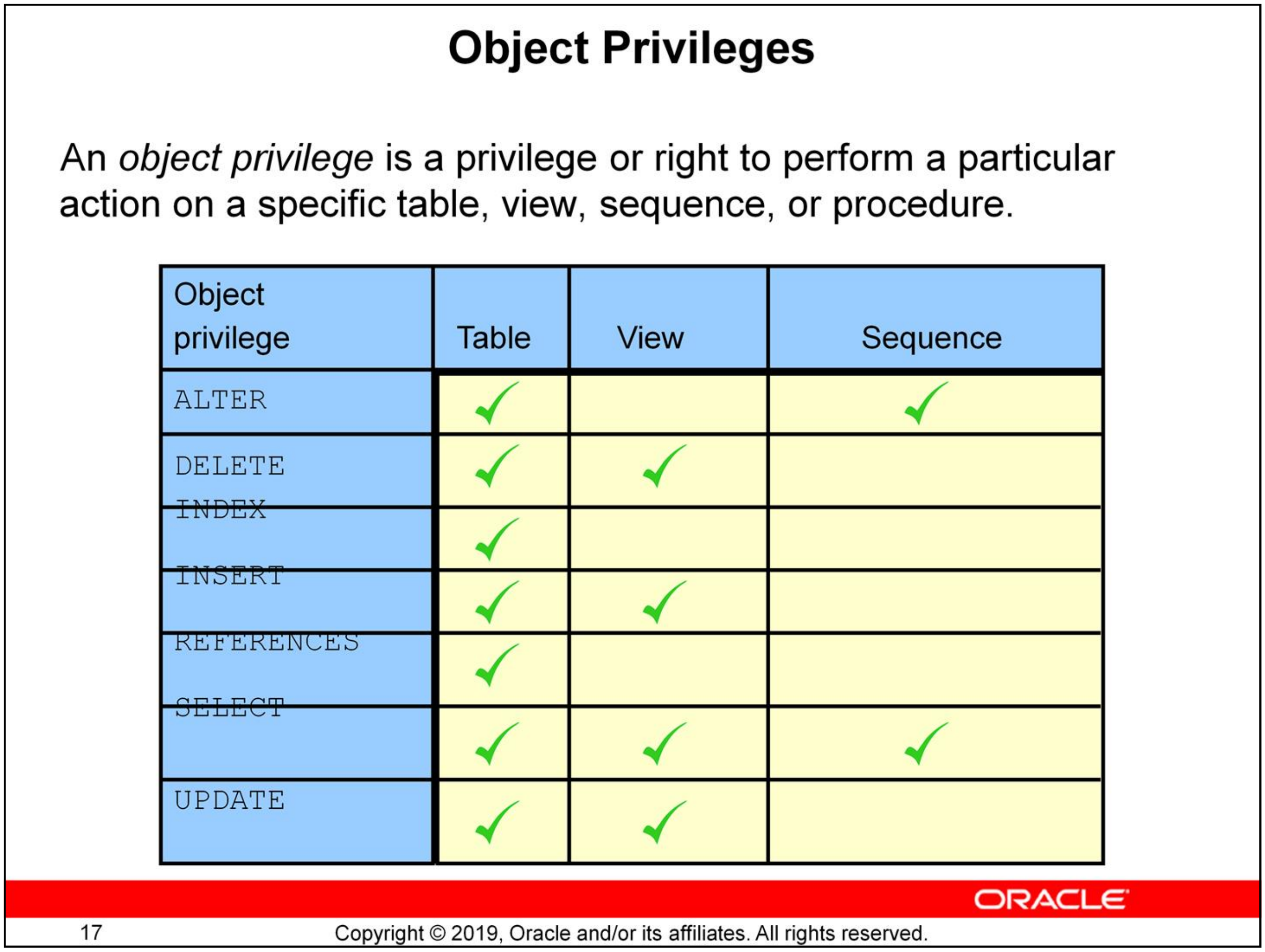
For more information, see the *Oracle Database SQL Language Reference* for Oracle Database 19c.

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.



Lesson Agenda

- System privileges
- Creating a role
- Object privileges
- Revoking object privileges



Object Privileges

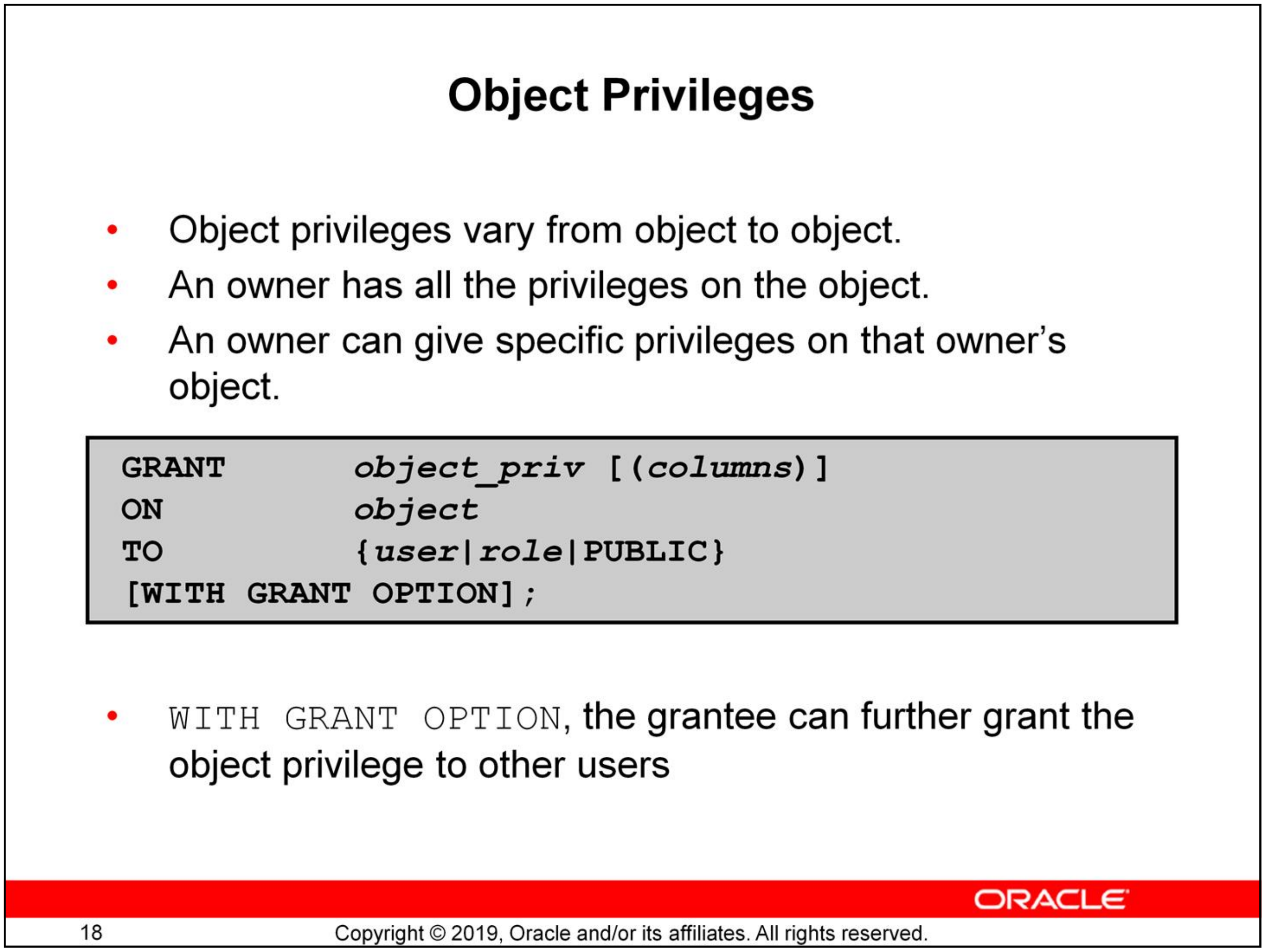
An *object privilege* is a privilege or right to perform a particular action on a specific table, view, sequence, or procedure.

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

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.



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];
```

- `WITH GRANT OPTION`, the grantee can further grant the object privilege to other users

Granting Object Privileges

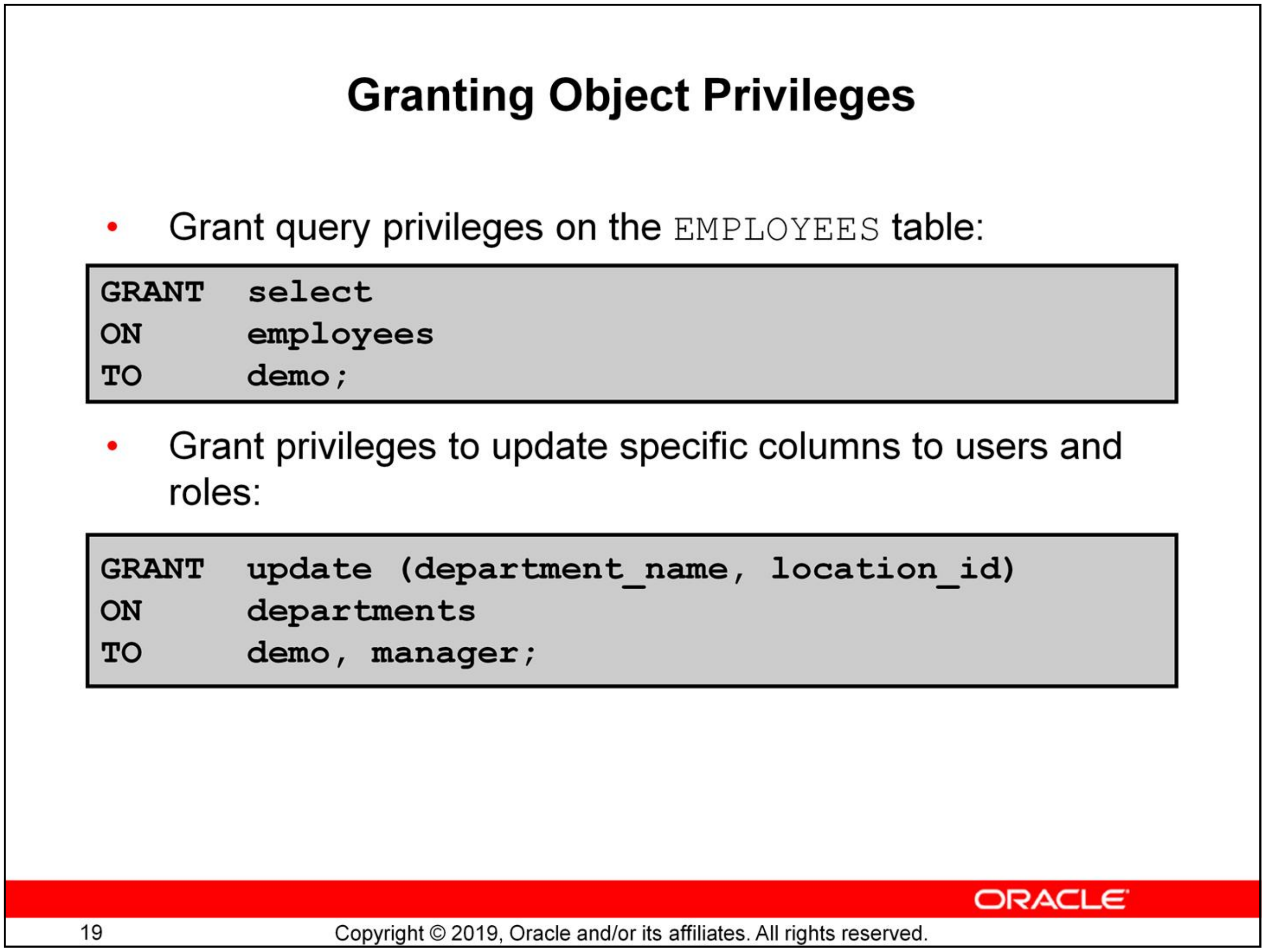
Different object privileges are available for different types of schema objects. A user automatically has all object privileges for schema objects contained in the user's schema. A user can grant any object privilege on any schema object that the user owns to any other user or role. If the grant includes `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.



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.

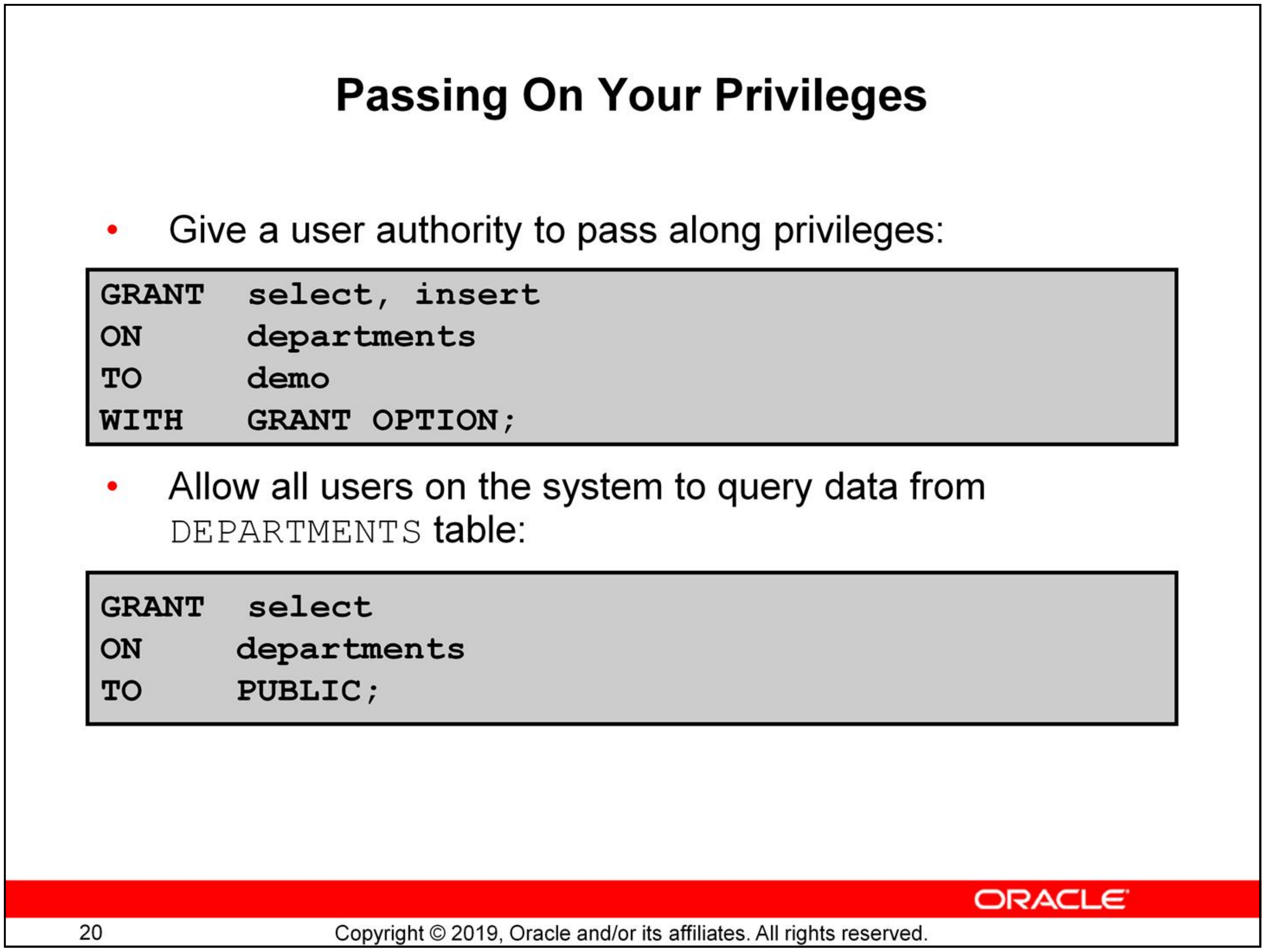
For example, if your schema is `ora`*`xx`*, and the `demo` user now wants to use a `SELECT` statement to obtain data from your `EMPLOYEES` table, the syntax he or she must use is:

```
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

- 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;
```

`WITH GRANT OPTION` Keyword

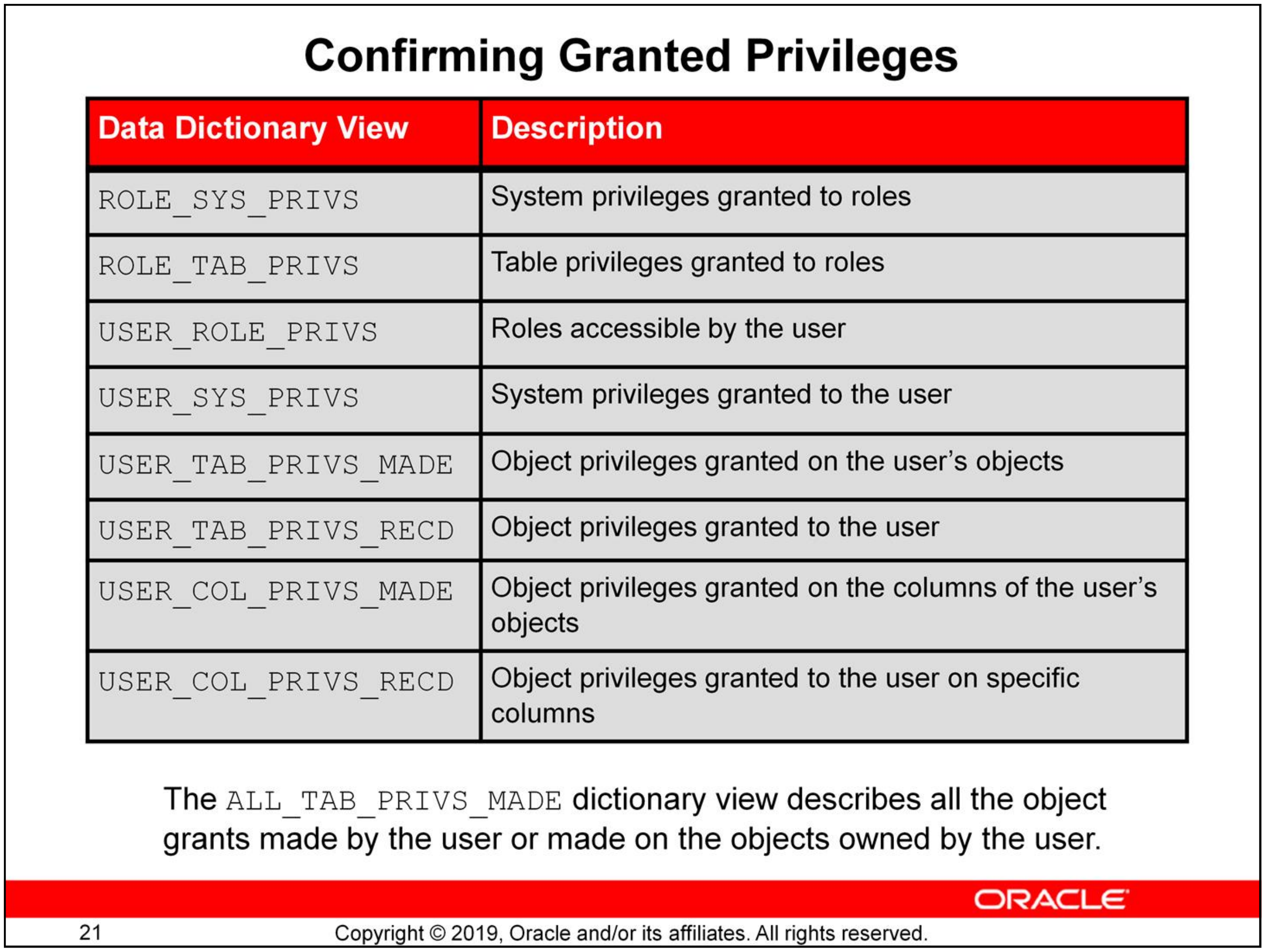
A privilege that is granted with the `WITH GRANT OPTION` clause can be passed on to other users and roles by the grantee. Object privileges granted with the `WITH GRANT OPTION` clause are revoked when the grantor's privilege is revoked.

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 `demo` 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 `DEPARTMENTS` table.



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

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.

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

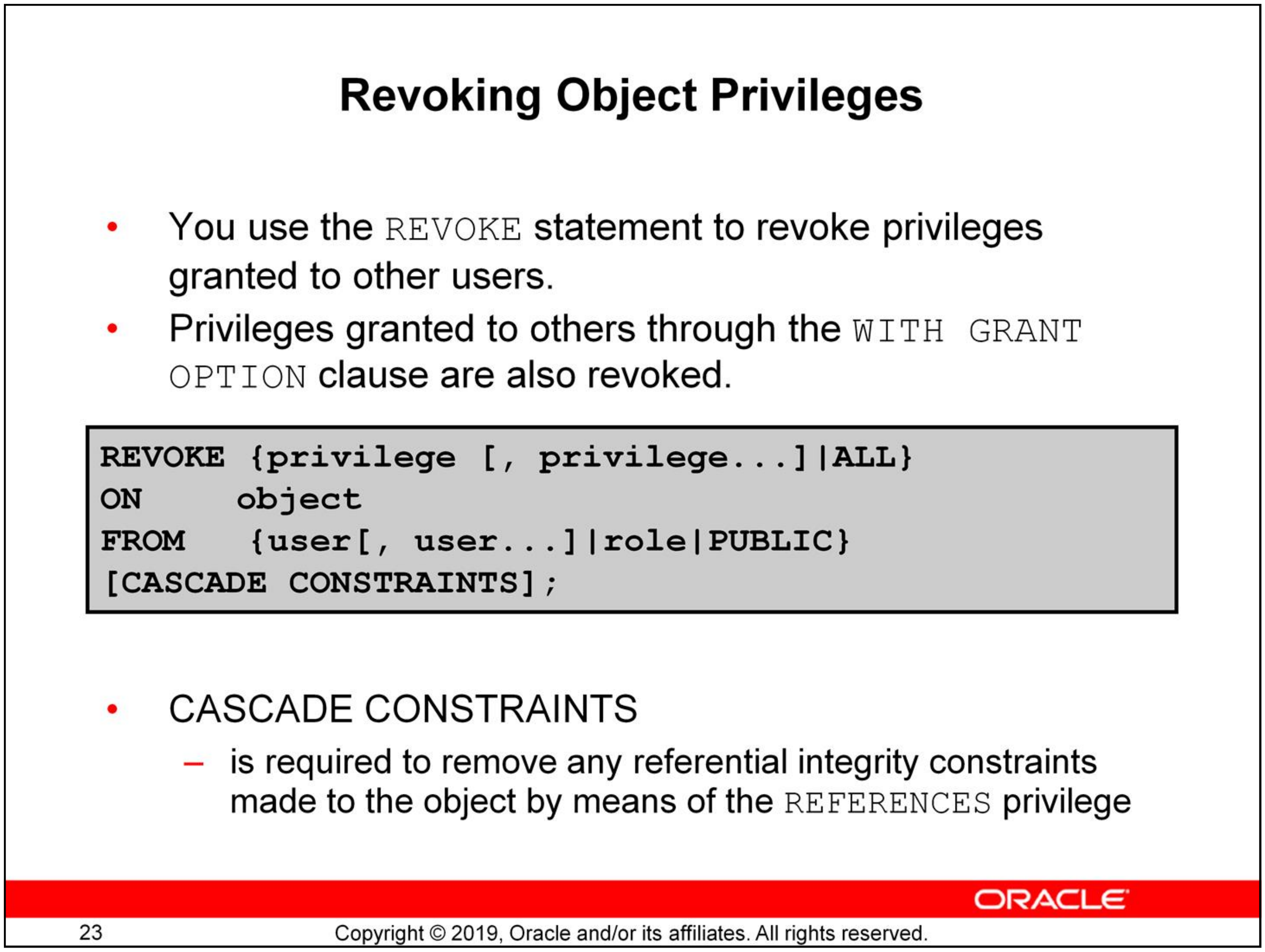
The data dictionary is organized in tables and views and contains information about the database. You can access the data dictionary to view the privileges that you have. The table in the slide describes various data dictionary views.

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.

Lesson Agenda

- System privileges
- Creating a role
- Object privileges
- Revoking object privileges



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];
```

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

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.

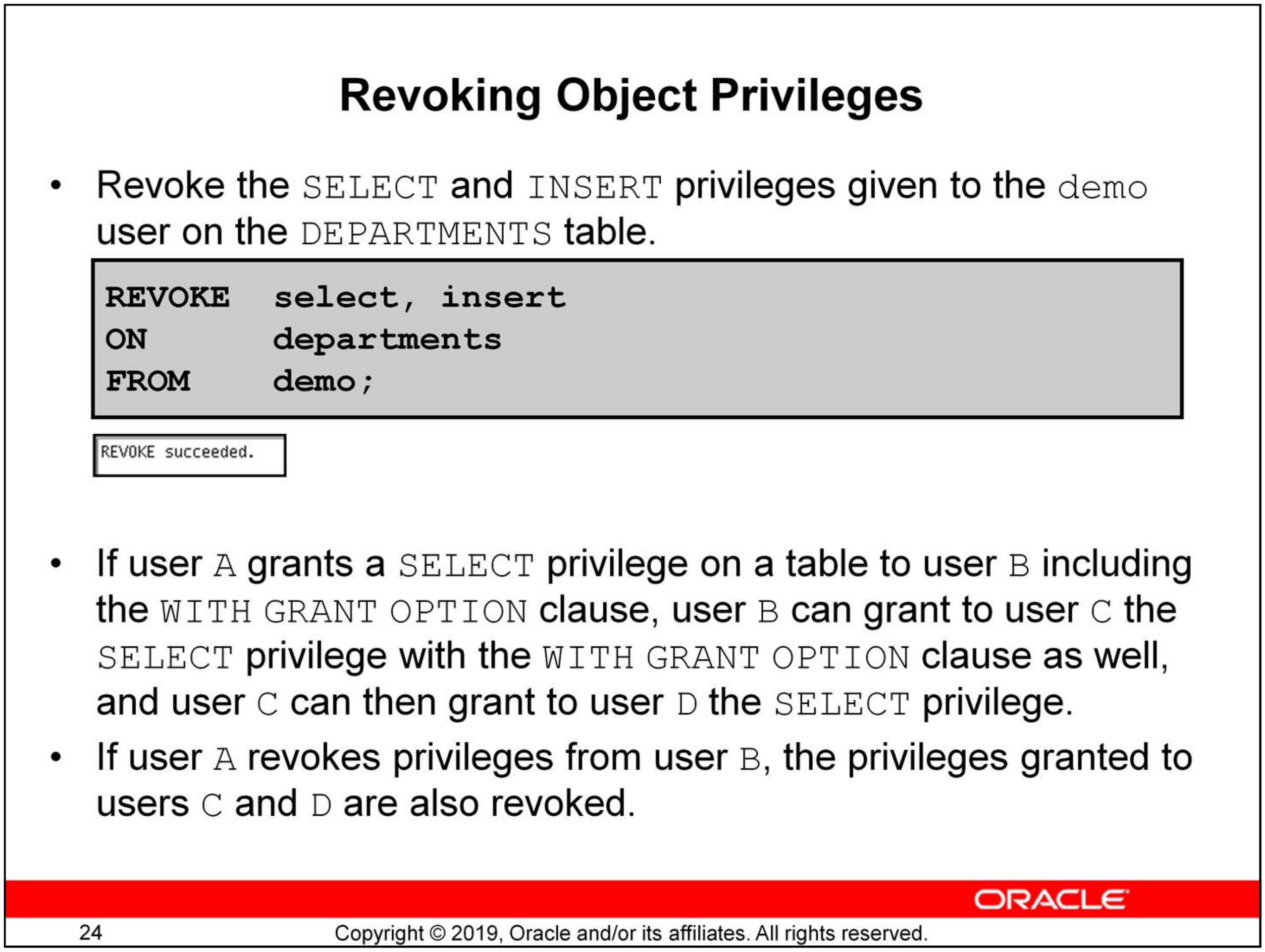
In the syntax:

`CASCADE CONSTRAINTS`

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

For more information, see the *Oracle Database SQL Language Reference* for Oracle Database19c.

Note: If a user leaves the company and you revoke his or her privileges, you must regrant any privileges that this user 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

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

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

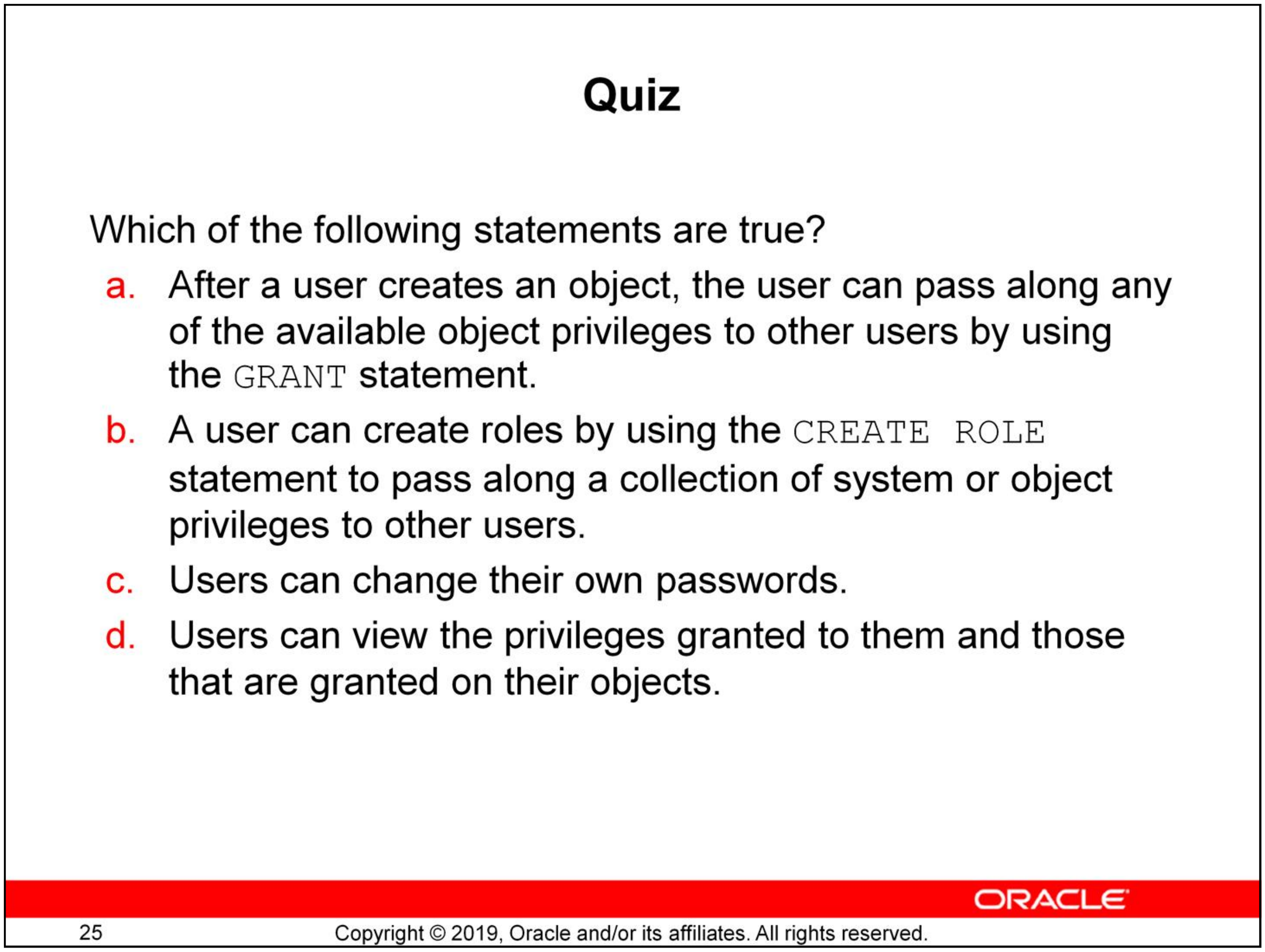
```
REVOKE succeeded.
```

- 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.

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.



Quiz

Which of the following statements are true?

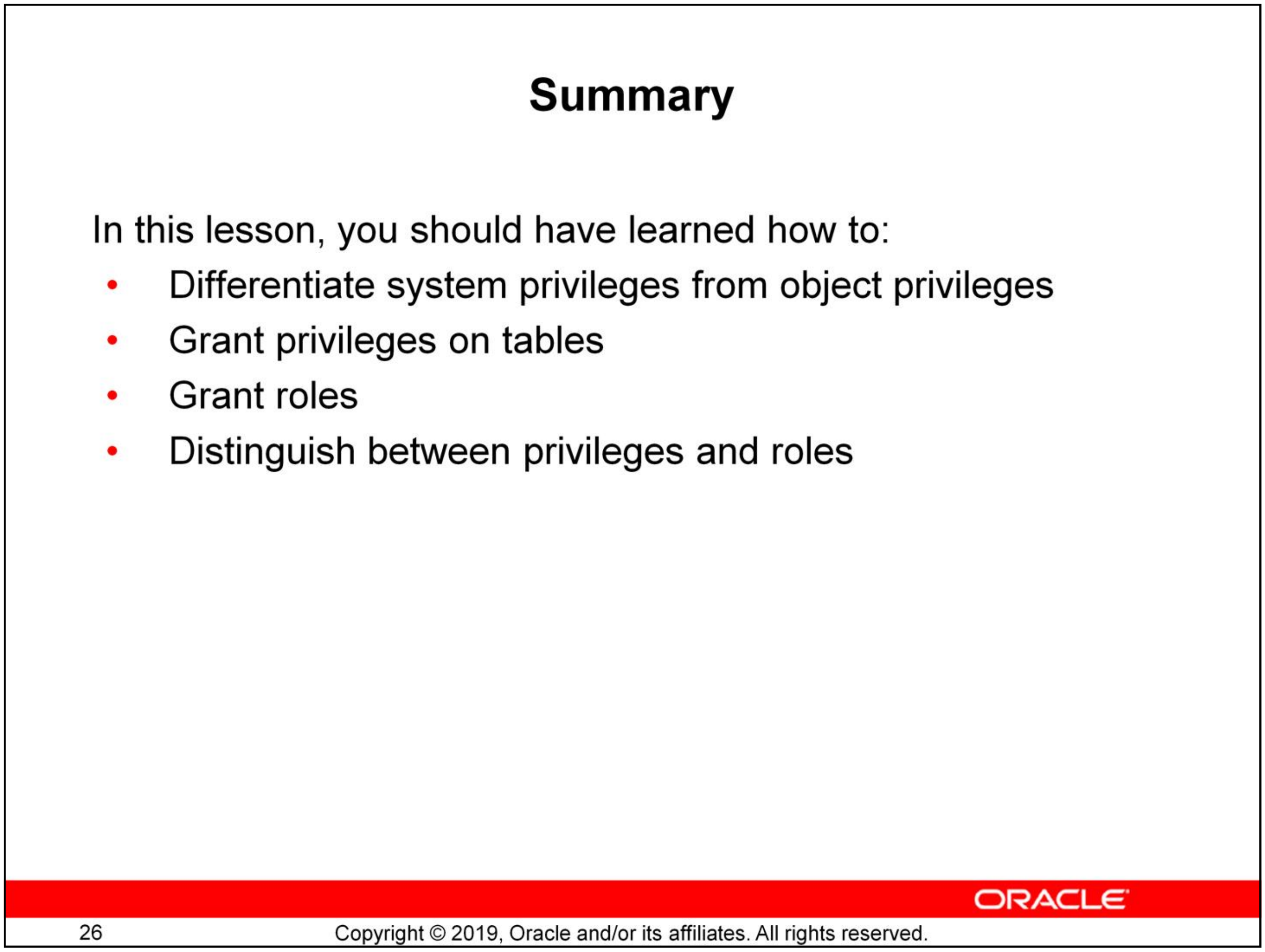
a. After a user creates an object, the user can pass along any of the available object privileges to other users by using the `GRANT` statement.

b. A user can create roles by using the `CREATE ROLE` statement to pass along a collection of system or object privileges to other users.

c. Users can change their own passwords.

d. Users can view the privileges granted to them and those that are granted on their objects.

Answer: a, c, d



Summary

In this lesson, you should have learned how to:

- Differentiate system privileges from object privileges
- Grant privileges on tables
- Grant roles
- Distinguish between privileges and roles

DBAs establish initial database security for users by assigning privileges to the users.

- The DBA creates users who must have a password. The DBA is also responsible for establishing the initial system privileges for a user.
- After the user has created an object, the user can pass along any of the available object privileges to other users or to all users by using the `GRANT` statement.
- A DBA can create roles by using the `CREATE ROLE` statement to pass along a collection of system or object privileges to multiple users. Roles make granting and revoking privileges easier to maintain.
- Users can change their passwords by using the `ALTER USER` statement.
- You can remove privileges from users by using the `REVOKE` statement.
- With data dictionary views, users can view the privileges granted to them and those that are granted on their objects.