Managing User Privileges

By Ahmed Baraka

Objectives

In this lecture, you will learn how to perform the following:

- Describe the difference between system and user privileges
- Manage system privileges
- Manage object privileges
- Understand the cascading effects in system and object privileges

About User Privileges

- **System privileges**: Enables users to perform particular actions in the database. Examples: connect to the database, create tablespace, create table, create views,… etc.
  - Can be granted by the DBA or by someone who has been given explicit permission to administer the privilege
- **Object privileges**: Enables users to access and manipulate a specific object. Examples: insert into **`EMPLOYEES`** table, select from **`EMPLOYEES`** table, execute **`GET_SALARY`** function,… etc.
  - Can be granted by the owner of an object, by the DBA, or by someone who has been explicitly given permission to grant privileges on the object.
- **Administrative privileges**: used for commonly performed specific administrative tasks, like backup and recovery, Data Guard, and others.

About Managing System Privileges

- Examples:

System Privilege	Description
`CREATE TABLE`	Enables a user to create a table owned by that user.
`CREATE ANY TABLE`	Enables a user to create a table owned by any user in the database.
`ALTER ANY TABLE`	Enables a user to alter any table in the database. **Note**: There is no **`ALTER TABEL`** privilege.
`CREATE PROCEDURE`	Enables a user to create a PL/SQL procedure, function or package owned by that user.

- To list all the system privileges (workaround):

```
SELECT PRIVILEGE, NAME FROM SYSTEM_PRIVILEGE_MAP;
```

- Oracle Database 19c SQL Language Reference: table 18-1

Who Can Grant or Revoke System Privileges?

- Users with the system privilege **`GRANT ANY PRIVILEGE`**
- Users who were granted a specific system privilege with the **`ADMIN OPTION`**

Granting System Privileges to Users

- To grant a system privilege:

```
GRANT <system_privilege> TO <grantee> [WITH ADMIN OPTION]
[CONTAINER = CURRENT | ALL ]
```

- An example for an application account:

```
GRANT CREATE SESSION, CREATE VIEW, ALTER SESSION, CREATE SEQUENCE,
CREATE SYNONYM, CREATE DATABASE LINK TO hr;
```

- Some privileges have the **ANY** option:

```
GRANT CREATE ANY PROCEDURE TO ogg;
GRANT SELECT ANY TABLE TO ogg;
```

- **SYS** and users with the DBA role are granted all of the **ANY** privileges

Granting `ALL PRIVILEGES` to Users

- Represents all the system privileges, except: **`SELECT ANY DICTIONARY`**, **`ALTER DATABASE LINK`**, **`ALTER PUBLIC DATABASE LINK`**, and **`ADMINISTER KEY MANAGEMENT`**
- Format:

```
GRANT ALL PRIVILEGES TO <username>
```

- Example:

```
GRANT ALL PRIVILEGES TO ourDBA;
```

- Avoid using it in production systems. Consider using roles instead.

Creating Users or Changing Their Passwords using `GRANT` Statement

- Format:

```
GRANT <system privilege> TO <grantee> [IDENTIFIED BY <password>]
```

  - If the password is provided:
    - If the user exists, its password is reset to the provided password
    - If the user doesn't exist, it will be created

- Example

```
GRANT ALL PRIVILEGES TO scott IDENTIFIED BY "ABcd##1234";
```

Revoking System Privileges

- To revoke a system privilege:

```
REVOKE <system_privilege> FROM <grantee> [CONTAINER = CURRENT | ALL ]
```

- The action take effect immediately including the current session.
- Example:

```
REVOKE CREATE DATABASE LINK, CREATE TABLE FROM hr;
```

About Managing Object Privileges

- A permission to perform a particular action on a specific schema object.
- Examples:
  - Update **`HR.JOBS`** table
  - Select rows from **`HR.EMPLOYEES`**
  - Execute the procedure **`HR.PAYROLL`**
- Some schema objects, such as indexes, triggers, and database links, do not have associated object privileges.
- Oracle Database 19c SQL Language Reference: table 18-2 "Object Privileges"

Who Can Grant or Revoke Object Privileges?

- The owner
- A user with the **`GRANT ANY OBJECT PRIVILEGE`** system privilege
- A user granted the privilege using the **`GRANT`** statement with the **`WITH GRANT OPTION`** clause

Granting and Revoking Object Privileges

- For granting object privileges, use **GRANT ON TO** statement:

```
GRANT <object_priv> ON <object> TO <grantee> [WITH GRANT OPTION]
[CONTAINER = CURRENT | ALL ]
```

- For revoking object privileges, use **REVOKE ON FROM** statement:

```
REVOKE <object_priv> ON <object> FROM <grantee>
[CONTAINER = CURRENT | ALL ]
```

- Example:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON HR.EMPLOYEES TO scott;
```

```
GRANT UPDATE (FIRST_NAME, LAST_NAME) ON HR.EMPLOYEES TO scott;
```

SQL92_SECURITY and UPDATE and DELETE Privileges

- Applies on 12.2 and onwards, if **SQL92_SECURITY** is set to **TRUE** (default), the **UPDATE** and **DELETE** object privileges do not take effect unless the **SELECT** privilege on the same table is also granted.
- Example, the following grant is not enough to allow the grantee update **EMPLOYEES** table:

```
GRANT UPDATE ON HR.EMPLOYEES TO SCOTT;
```

- The following statement does not work either:

```
GRANT READ, UPDATE ON HR.EMPLOYEES TO SCOTT;
```

- The following statement works:

```
GRANT SELECT, UPDATE ON HR.EMPLOYEES TO SCOTT;
```

Using ALL Clause with GRANT or REVOKE

- **ALL [PRIVILEGES]** represents all available object privileges for an object:

```
GRANT ALL ON HR.EMPLOYEES TO scott;
```

```
GRANT ALL ON HR.EMPLOYEES TO scott;
REVOKE DELETE, INDEX ON HR.EMPLOYEES FROM scott;
```

```
REVOKE ALL ON HR.EMPLOYEES FROM scott;
```

- Revoking **REFERENCES** privilege may require **CASCADE CONSTRAINTS**:

```
REVOKE ALL ON HR.EMPLOYEES FROM scott CASCADE CONSTRAINTS;
```

Note: With object privileges the keyword **PRIVILEGES** is optional, with system privileges the keyword **PRIVILEGES** is mandatory.

READ and SELECT Object Privileges

- If you want the user to only query the segment:

```
GRANT READ ON HR.EMPLOYEES TO scott;
```

- If you want the user to query the segment and be able to perform:
  - **LOCK TABLE <table_name> IN EXCLUSIVE MODE**;
  - **SELECT ... FROM <table_name> FOR UPDATE**;

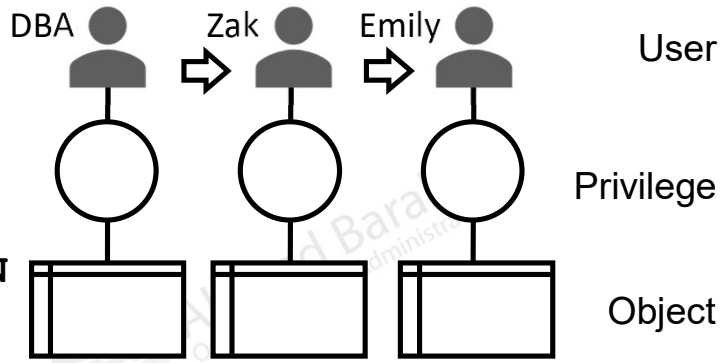
  … then, grant the user the **SELECT** privilege:

```
GRANT SELECT ON HR.EMPLOYEES TO scott;
```

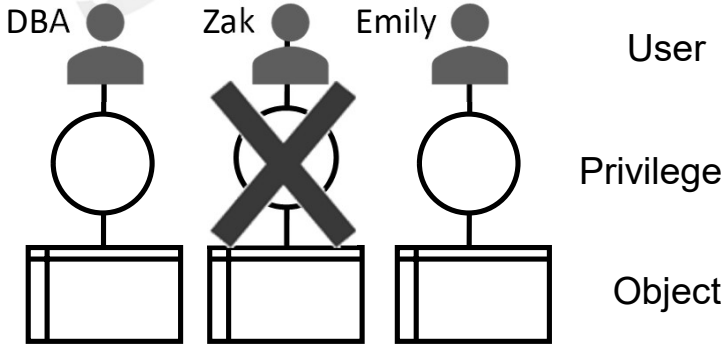
- The corresponding system privileges are: **READ ANY TABLE** and **SELECT ANY TABLE**
- Applicable segments: tables, views, materialized views, or synonyms

Revoking System Privileges with ADMIN OPTION

```
GRANT CREATE
TABLE TO ZAK
WITH ADMIN OPTION
```




```
REVOKE
```

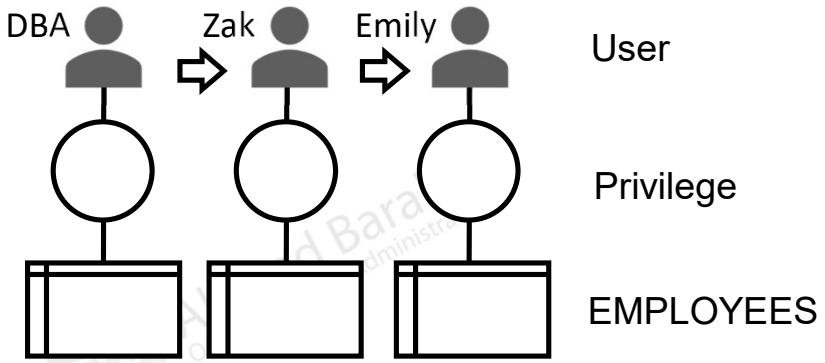


Revoking System Privileges with ADMIN OPTION

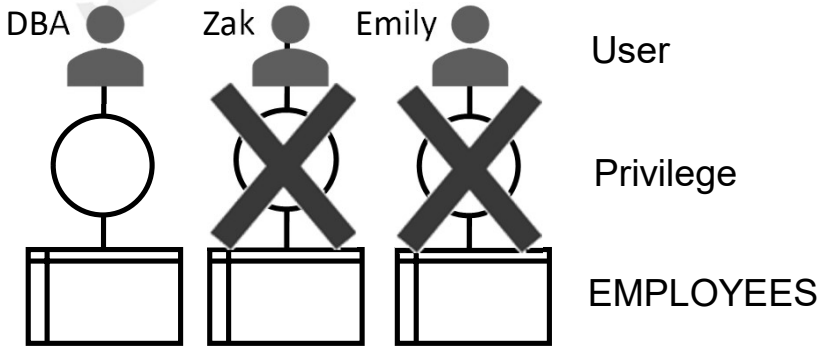
- Users with **ADMIN OPTION** for a system privilege can revoke the privilege from any other database user.
- There are no cascading effects when a system privilege is revoked, regardless of whether it is given by the **ADMIN OPTION**.

Revoking Object Privileges with GRANT OPTION

```
GRANT SELECT
ON EMPLOYEES
TO ZAK
WITH GRANT OPTION
```




```
REVOKE
```



Revoking Object Privileges with GRANT OPTION

- Cascading effects are observed when an object privilege is revoked, regardless of whether it is given by the **`GRANT OPTION`**.
- As a user, you can revoke only those object privileges that you have granted (or a user with the privilege **`GRANT ANY OBJECT PRIVILEGE`**)

User Privilege Dictionary Views

View	Description
DBA_SYS_PRIVS	system privileges granted to users (and roles)
USER_TAB_PRIVS	describes the object grants for which the current user is the object owner, grantor, or grantee
DBA_TAB_PRIVS	describes all object grants in the database
USER_TAB_PRIVS_MADE	retrieves all the object grants made by the current user
USER_TAB_PRIVS_RECD	retrieves all the object grants for which the current user is the grantee

Summary

In this lecture, you should have learnt how to perform the following:

- Describe the difference between system and user privileges
- Manage system privileges
- Manage object privileges
- Understand the cascading effects in system and object privileges