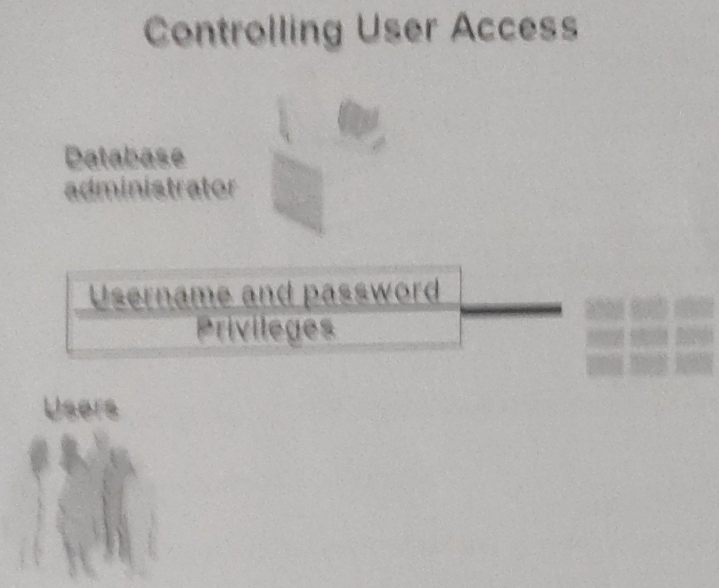
Ex.No.: 16	CONTROLLING USER ACCESS
Date: 19/11/25	

Objectives

After the completion of this exercise, the students will be able to do the following:

- Create users
- Create roles to ease setup and maintenance of the security model
- Use the GRANT and REVOKE statements to grant and revoke object privileges
- Create and access database links



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

Privileges

- Database security:
  - System security
  - Data security

• System privileges: Gaining access to the database
• Object privileges: Manipulating the content of the database objects
• Schemas: Collections of objects, such as tables, views, and sequences

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

Typical DBA Privileges

System Privilege	Operations Authorized
CREATE USER	Grantee can create other Oracle users (a privilege required for a DBA role)
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 snapshots in any schema
CREATE ANY TABLE	Grantee can create tables in any schema

Creating Users

The DBA creates users by using the CREATE USER statement.

EXAMPLE:

CREATE USER scott IDENTIFIED BY tiger;

User System Privileges

• Once a user is created, the DBA can grant specific system privileges to a user.
• An application developer, for example, may have the following system privileges:
– CREATE SESSION
– CREATE TABLE
– CREATE SEQUENCE
– CREATE VIEW
– CREATE PROCEDURE

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

Typical User Privileges

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 designates that every user is granted the privilege

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

Granting System Privileges

The DBA can grant a user specific system privileges.

GRANT create session, create table, create sequence, create view TO scott;

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 users to the role.

Syntax
CREATE ROLE *role*;
In the syntax:
role is the name of the role to be created
Now that the role is created, the DBA can use the GRANT statement to assign users to the role as well as
assign privileges to the role.

Creating and Granting Privileges to a Role

CREATE ROLE manager;
Role created.

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

GRANT manager TO DEHAAN, KOCHHAR;
Grant succeeded.

- Create a role
- Grant privileges to a role
- Grant a role to users

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 scott
IDENTIFIED BY lion;
User altered.

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 allows the grantee to grant the object privileges to other users and roles

Granting Object Privileges

- Grant query privileges on the EMPLOYEES table.

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

GRANT select
ON employees
TO sue, rich;

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

Using the WITH GRANT OPTION and PUBLIC Keywords

- Give a user authority to pass along privileges.

- Allow all users on the system to query data from Alice's DEPARTMENTS table.

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

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

How to Revoke 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];
```

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

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

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

3. Team1:

INSERT INTO DEPARTMENTS VALUES (500, 'Education');

Team 2:

INSERT INTO Department values (510, 'Human Resources');

Find the Solution for the following:

1. What privilege should a user be given to log on to the Oracle Server? Is this a system or an object privilege?

The CREATE SESSION privilege is needed to log on to the Oracle Server. This is a system privilege.

2. What privilege should a user be given to create tables?

The CREATE TABLE system privilege should be granted to allow Table Creation

3. If you create a table, who can pass along privileges to other users on your table?

Only the table owner or a user with GRANT option can pass privileges on their table.

4. You are the DBA. You are creating many users who require the same system privileges. What should you use to make your job easier?

use oracle roles to easily manage and grant the same set of privileges to many users.

5. What command do you use to change your password?

ALTER USER username IDENTIFIED BY newpassword;

6. Grant another user access to your DEPARTMENTS table. Have the user grant you query access to his or her DEPARTMENTS table.

GRANT SELECT DEPARTMENTS TO other_user;

7. Query all the rows in your DEPARTMENTS table.

SELECT * FROM DEPARTMENTS

8. Add a new row to your DEPARTMENTS table. Team 1 should add Education as department number 500. Team 2 should add Human Resources department number 510. Query the other team's table.

9. Query the USER_TABLES data dictionary to see information about the tables that you own.

SELECT * FROM USER_TABLES;

10. Revoke the SELECT privilege on your table from the other team.

REVOKE SELECT ON DEPARTMENTS FROM other_user;

11. Remove the row you inserted into the DEPARTMENTS table in step 8 and save the changes.

DELETE FROM DEPARTMENTS WHERE DEPARTMENT_ID=500;

COMMIT;

RESULT:

Thus all Privilege, DML and metadata tasks for the oracle experiment was performed successfully with correct SQL statements.