**Controlling User Access and Data Base Security**

* Used to share the data, database objects with other users.
* Users can also cancel the permission.
* Use the GRANT and REVOKE statements to grant and revoke object privileges

**Privileges:**

Privileges are the right to execute particular SQL statements. The database administrator (DBA) is a high-level user with the ability to grant users access to the database and its objects. The 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.

**Grant:**

• The GRANT command is used to allow another schema access to a privilege.

• GRANT command can be issued not only on TABLE OBJECT, but also on VIEWS, SYNONYMS, SEQUENCES Etc.

• The users are GRANTED at time all 11 privilege

(i.e. INSERT, UPDATE, DELETE, SELECT) by using ALL.

Syntax:

Sql>GRANT <privilege1 [, privilege2...] ON <ObjectName>TO <user1>

[, user| role, PUBLIC...];

**Creating Users:**

Login as

User:-system/ sys as sysdba

Password:-12345678

**Syntax:**

Sql>Create User <User Name> Identified By <Password>;

Sql>Grant Connect, Resource To <User Name>;

**EX:**

Sql> Create User UserA Identified By userA;

Sql>Grant Connect,Resource To userA;

**Login as Scott**

Sql>Grant ALL

ON Emp

TO userA;

Sql>Grant INSERT,SELECT,DELETE

ON Dept

TO userA;

**Grant columns:**

Sql>Grant INSERT(Empno,Ename,Sal,Deptno)

ON Emp

TO userC;

**Note:**

* Only insert and update permission are allowed.

Sql>Grant Select ON Student TO userA, userB;

Sql>Grant Delete ON Supp TO Public;

**Public**

• Public Keyword represents all users in database

• All users can Delete on Supp table.

**Note:**

• We cannot give permission at time on more than one database object

**Steps To Be Performed:**

* Connect to the required user using the USER Name and Password.
* Execute the required SQL statement using the object hierarchy.

**Login as userA**

Sql>Select \*

From Scott.emp;

Sql>Insert Into Scott.Dept

values(50,'SW','Hyd');

Sql>Select \*

From scott.Student;

**Note:**

At time DML, DRL operation we have to mention owners name.

**REVOKE Command:**

• It used to remove the access allowed by GRANT.

• REVOKE privileges is assigned not only on TABLE OBJECT, but also on VIEWS,SYNONYMS,SEQUENCES Etc.

Sql>REVOKE <privilege1 [, privilege2...]

ON <ObjectName>

FROM <user1> [, user| role, PUBLIC...];

Sql>Revoke Select

ON Student

From userA,userB;

Sql> REVOKE ALL

ON Emp

From userA;

**Types of Privileges:**

System Privileges:

• They allow a user to perform certain action within the Database.

Object Privileges:

• An object privilege allow a user to perform certain actions on Database Objects.

**Checking the object(table) privileges, granted:**

* The schema object that stores that stores the information about the privileges granted is USER\_TAB\_PRIVS\_MADE.
* The columns of USER\_TAB\_PRIVS\_MADE.
* GRANTEE
* TABLE NAME
* GRANTOR
* PRIVILEGE
* GRANTABLE
* HIERARCHY

Sql>Select GRANTEE, TABLE\_NAME, GRANTOR, PRIVILEGE

From USER\_TAB\_PRIVS\_MADE

Where TABLE\_NAME = 'EMP';

**USER\_COL\_PRIVS**

This table query to view the object privileges granted to the user on specific columns.

Sql>Select Grantee,Grantor,Column\_Name

From USER\_COL\_PRIVS

Where Table\_Name = 'EMP';

Sql>Select GRANTEE

From All\_Tab\_Privs

Where Grantor='SCOTT' AND

Table\_Name = 'EMP';

**Checking Object Privileges Received:**

* The schema object that stores the information about the privileges that are received is USER\_TAB\_PRIVS\_RECD.
* The columns of USER\_TAB\_PRIVS\_RECD.
* OWNER
* GRANTEE
* TABLE NAME
* GRANTOR
* PRIVILEGE
* GRANTABLE
* HIERARCHY

Sql> Select OWNER NAME, GRANTOR, PRIVILEGE

From USER\_TAB\_PRIVS\_RECD;

**Roles In Oracle:**

A role is a named group of related privileges that can be granted to the user.

**Advantages:**

* Rather than assigning Privileges one at a time directly to s USER, we can CREATE a ROLE, assign PRIVILEGES to that ROLE, and then GRANT that ROLE to Multiple USERS and ROLES.
* When you add or delete a privilege from a role, all users and roles assigned that ROLE automatically receive or lose those privileges.
* We can assign multiple roles to a single USER or ROLE to another ROLE.
* A role can be assigned with a password.

**ROLE'S Creation:**

* To creation a role we should have the CREATE ROLE SYSTEM privilege
* The steps in implementing the roles
* Role creation
* Granting privileges to roles
* Granting ROLES TO USERS OR OBJECTS

**Syntax:**

Sql>CREATE ROLE<Role Name>

[IDENTIFIED BY <Password>];

Sql> CREATE ROLE Sales\_manager

Identified By SalesAudit;

**Note:** We can alter the ROLE for password and new password.

Sql> Alter ROLE Sales\_manager identified by salesAudit;

**Grant a ROLE to a USER:**

Sql>Grant Sales\_manager TO Scott;

**Granting Multiple ROLES to another ROLE**

Sql>Grant ROLE1,ROLE2,...

TO <Target\_role\_name>;

**Checking ROLES Granted To A user:**

* The schema object USER\_ROLE\_PRIVS specifies the ROLES granted to a USER.
* The column of USER\_ROLE\_PRIVS
* USERNAME
* GRANTED\_ROLE
* ADMIN\_OPTION
* DEFAULT\_ROLE
* OS\_GRANTED

Sql> Select USERNAME, GRANTED\_ROLE

From USER\_ROLE\_PRIVS;

**System privileges Granted to a ROLE:**

• The schema object ROLE\_SYS\_PRIVS specifies the SYSTEM PRIVILEGES granted to a ROLE.

• The columns of ROLE\_SYS\_PRIVS

* ROLE
* PRIVILEGE
* ADMIN\_OPTION

Object privileges Granted to a ROLE:

• The schema object R0LE\_TAB\_PRIVS specifies the SYSTEM PRIVILEGES granted to a ROLE.

• The columns of ROLE\_TAB\_ PRIVS

R0LE COLUMN\_NAME

OWNER PRIVILEGE

TABLE\_NAME GRANTABLE

Sql>Select ROLE, PRIVILEGE

From ROLE\_TAB\_PRIVS;

**Revoking a ROLE:**

Sql>Revoke Sales\_Manager From Scott;

Sql> Revoke ALL ON Emp From Sales\_Manager;

**Dropping A ROLE:**

Sql>Drop ROLE<Role name>;

Sql>DROP ROLE Sales\_manager;

**To change the Password of user**

**(WE MUST BE UNDER your Own Iogin.)**

Sql>Alter user krishna identified by wisdom;

Sql>Revoke connect,resource from krishna;

Sql>Drop user krishna;