

Grant:

Oracle provides extensive security feature in oracle to safeguard information stored in its tables from unauthorized viewing damage. Depending on a user's status and responsibility appropriate right on oracle's resource can be assigned to the user by the DBA. The right that allow the use of some or all of oracle's resource on the server are called privileges.

Oracles that are created by a user owned and controlled by that user. If a user wishes to access any of the objects belonging to another user, the owner of the object will have to give permissions for such access. This is called granting of privileges.

The owner of the object can take privileges once given back. This is called revoking of privileges. Granting privileges using the grant statement.

Syntax:

```
GRANT <Object Privilege> ON < Object Name> to <User name>  
[With Grant Option]
```

- Object Privileges: ALTER, DELETE, INDEX, INSERT, SELECT, UPDATE
- User can do alter, delete, insert, select, and update operation on a table. As well as can set index to a particular column of a table.
- With grant option: Using this 'with grant option' allows the user to in turn grant object privileges to other users.

Example:

1. To give all privileges:
Grant all on client_master to abc;
2. To give selected privileges:
Grant select, insert on client_master to abc;
3. To further grant:
Grant select on client_master to abc with grant option;

Now abc user can see records:

```
Select * from xyz.client_master;
```

For further grant abc can write:

```
Grant select on xyz.client_master to meet;
```

REVOKE:

Privileges once given can be denied to a user using the revoke command. The object owner can revoke privileges grant to another user. A user of an object who is not owner, but has been granted the grant privileges has the power of revoke the privileges from a grantee. Revoking permission using the revoke statement.

Syntax: Revoke <Object privilege> ON <Object name> FROM <User name>

Example: Revoke delete ON emp_mstr From ABC;

Revoke select on xyz.client_master from meet;

Role:

A role is set of privileges that a user can grant to another user. Oracle has eight system-supplied role as listed in the following. The first three are most commonly used.

Connect	Alter, session, create cluster, create database link, create sequence, create session, create synonym, create table, create view
Resource	Create cluster, create procedure, create sequence, create table, create trigger, create type
DBA	All system privileges with admin option
Exp_full_database	Select any table, backup any table, insert, update, delete on sys.incxp, sys.incvld, sys.incfil
Imp_full_database	Become user
Delete_catalog_role	Delete on sys.aud\$
Execute_catalog_role	Execute on data dictionary packages
Select_catalog_role	Select on data dictionary views

Syntax:

Create role role_name [identified by password];

Ex:

Create role data;

Grant select on client_master to data;

Grant select on product master to data;

Grant select,update on student_master to data;

Now one can grant this single role create to any user. So that will grant these privileges on different tables to related user just by granting role.

Create User:

The oracle system comes with two users already created, system and sys. You log onto the system user to create other users, since system has that privileges. Create user create a user account that lets you log onto the database with a certain set of privileges and strong settings. If you specify a password you must supply that password to login.

When you first create user, the user has no privileges. You must use the grant command to grant role and privileges to user. You should usually grant create session as a minimal privilege.

Syntax:

Create user Username identified by password;

Ex: create user abc identified by abc;

What is transaction?

A series of one or more SQL statements that are logically related or a series of operations performed on oracle table data is termed as a transaction.

- Transaction is a group of events,
- Connecting to oracle
- Disconnecting from oracle
- Committing changes to the database table
- Rollback

Closing transaction: Commit and Rollback**Commit:**

A commit ends the current transaction and makes permanent any changes made during the transaction all transaction locks acquired on table are released.

Syntax : Commit [work];

Example : Insert into Employee Values('1001','ABC','Jetpur');
SQL> Commit;

Rollback:

A rollback does exactly the opposite of commit. It ends the transaction but undoes any changes made during the transaction. All transaction locks acquired on tables are released.

Syntax:

Rollback [work];

Example : SQL> Insert into Employee Values('1001','ABC','Jetpur');
SQL> Rollback;

- Ends the transaction
- Undoes all the changes in the current transaction.
- Erase all save point in that transaction
- Releases the transactional locks.

Savepoint:

Savepoint marks and saves the current point in the processing of a transaction. When a savepoint is used with rollback statement, parts of a transaction can be undo. An active savepoint is one that is specified since the last commit or rollback.

Syntax: Savepoint <savepoint name>

A predetermined portion of the transaction is rollback. Retains the savepoint rollback to but lose those created after the named savepoint.
