Transaction Commands

Prepared By: Dr. Vipul Vekariya
Controlling User access

- Database administrator
- Username and password
- Privileges
- Users
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
Creating Users

The DBA creates users by using the `CREATE USER` statement.

```
CREATE USER user
IDENTIFIED BY password;
```

```
CREATE USER scott
IDENTIFIED BY tiger;
User created.
```
User System Privileges

- Once a user is created, the DBA can grant specific system privileges to a user.

```sql
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
Granting System Privileges

The DBA can grant a user specific system privileges.

```
GRANT create session, create table,
create sequence, create view
TO scott;
Grant succeeded.
```
What is a Role?

Allocating privileges without a role

Allocating privileges with a role

Manager

Users

Privileges
Creating and Granting Privileges to a Role

• Create a role

CREATE ROLE manager;
Role created.

• Grant privileges to a role

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

• Grant a role to users

GRANT manager TO DEHAAN, KOCHHAR;
Grant succeeded.
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

<table>
<thead>
<tr>
<th>Object Privilege</th>
<th>Table</th>
<th>View</th>
<th>Sequence</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER</td>
<td>√</td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>DELETE</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXECUTE</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>INDEX</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INSERT</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REFERENCES</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SELECT</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>UPDATE</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
</tr>
</tbody>
</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];
```
Granting Object Privileges

- Grant query privileges on the **EMPLOYEES** table.

```
GRANT select
ON employees
TO sue, rich;
Grant succeeded.
```

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

```
GRANT update (department_name, location_id)
ON departments
TO scott, manager;
Grant succeeded.
```
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];
```
Revoking Object Privileges

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

```sql
REVOKE select, insert
ON departments
FROM scott;
Revoke succeeded.
```
Commit & Rollback

- Commit ends the current transaction and makes any permanent change made during transaction.
  
  syntax:  SQL>commit;
  
  commit complete.

Rollback: it is totally opposite of commit. It ends the transaction but undoes any changes made during transaction.

Syntax:  SQL>Rollback;
  
  rollback complete.
Savepoint

- Savepoint marks and save the current point in the processing of transaction. Savepoint is used with the rollback statement.

  syntax:   SQL>savepoint A;
           Savepoint created.

With rollback command.

  example : Rollback to A.