User:

A Property of database to access it. In Oracle user can be divided into 2 parts. Generic user and schema.

Generic user usually used to connect with database and do necessary action and modifications.

Schema is a property that contains tables, views, indexes, sequences, triggers etc. Generic user creates any object of a database under a schema. Like UNIVERSITY.STUDENTS.

Create User:

When Oracle Database is installed, a default user SYS is created with DBA privilege. A user is always created with password to protect it. Beside password it has some like default and temporary tablespace, profile and account status.

In an Oracle database, tablespace is a logical property that consists physical datafiles of a database. If a tablespace is not specified, the system tablespace will be used by default.

```
CREATE TABLESPACE TBS
  DATAFILE '/test_db/oradb/oradata/TESTDB/TBS/TBS_01.dbf'
  SIZE 500M                            #may not assign more than 30 GB to a single datafile.
;
```

During SQL statement performance some temporary data generate and these data store in temporary tablespace TEMP with tempfiles. For large database, sometime we need self-configured temporary tablespace. We can add or remove any datafile in tablespaces.

```
ALTER TABLESPACE TBS ADD DATAFILE '/TEST_DB/ORADB/ORADATA/TESTDB/TBS/TBS_02. dbf' SIZE 10G;
ALTER TABLESPACE TBS DROP DATAFILE '/TEST_DB/ORADB/ORADATA/TESTDB/TBS/TBS_02.dbf';
```

We can align quota of tablespace size to individual user. We can also align unlimited quota.

```
ALTER USER USERNAME QUOTA 100M ON TBS;
```

Profile:

Profile is use to configure user password management policies.

```
CREATE PROFILE PROFILE1 LIMIT
  SESSIONS_PER_USER DEFAULT              #number of user for a single session
  CPU_PER_SESSION UNLIMITED              #CPU time limit for a session
  CPU_PER_CALL UNLIMITED                 #CPU time limit for a SQL statement execution
  CONNECT_TIME UNLIMITED                 #Total time limit for a session
  IDLE_TIME UNLIMITED                    #continuous inactive time during a session
  LOGICAL_READS_PER_SESSION UNLIMITED #number of data blocks read in a session
```

```
LOGICAL_READS_PER_CALL UNLIMITED      #number of data blocks read for a SQL
                                      #statement execution
COMPOSITE_LIMIT UNLIMITED             #total resource cost for a session
PRIVATE_SGA UNLIMITED                 #space from SGA
FAILED_LOGIN_ATTEMPTS 10              #number of attempts before lock password
PASSWORD_LIFE_TIME UNLIMITED          #expiry date of password
PASSWORD_REUSE_TIME UNLIMITED         #number of days to reuse same password
PASSWORD_REUSE_MAX UNLIMITED          #number of password to be use to reuse
                                      #same password
PASSWORD_LOCK_TIME 1                  #number days to wait before unlock
PASSWORD_GRACE_TIME 7                 #grace period after password expiry
PASSWORD_VERIFY_FUNCTION NULL
;
```

During crating a profile failed attempts, password durability, reuse durability, max number of days to wait to reuse same password, number of user for a session and etc. can be defined. For a user, only one profile can be assigned. If any parameter of a profile is default then, it will be same as DEFUALT profile parameter. We can find details of each profile from ***DBA_PROFILES*** view.

We can alter any parameter of any profile by using following command.

```
ALTER PROFILE PROFILE1 LIMIT SESSIONS_PER_USER 10;
```

Any profile can also be dropped.

```
DROP PROFILE PROFILE1 CASCADE; #cascade helps to drop the profile if,
                                it is not assigned to any user.
```

VERIFY FUNCTION:

VERIFY_FUNCTION is used to define some criteria to password for its' security. Some of criteria are below.

- A password must be different from the user name.
- A password length must be greater than or equal to 4.
- A password must not be expectable. The following words cannot be used: 'database', 'account', 'user', 'password' and 'abcd'.
- A password must contain at least one digit, one character, and one special character.

We can modify ***VERIFY_FUNCTION*** as per our requirement.

Account Lock:

Generally, a user can be lock due to prevent access database.

ALTER USER USERNAME ACCOUNT LOCK;

Beside these, expiring value of FAILED_LOGIN_ATTEMPTS property of ***Profile***, a use can be blocked.

ALTER USER USERNAME ACCOUNT UNLOCK;

Due to expired PASSWORD_LIFE_TIME user can be locked.

ALTER USER USERNAME IDENTIFIED BY NEW_PASSWORD;

SYS, SYSTEM and SYSMAN cannot be locked and dropped. But their password can be changed. We can find details of every user from ***DBA_USERS*** view.

User Privileges:

After creations ***USERS*** needs privileges to perform SQL statements and modify objects. New user has no privileges to connect database. To connect database, user need to be ***GRANTED*** by system privilege ***CREATE SESSION***. Privileges can be 2 categories. Schema privileges and System privileges.

Schema Privileges:

Schema privileges are user access the objects of schema like table, view, index and etc. A user can also grant schema privileges to another user to access its' own objects.

GRANT SELECT ON USER1.TABLE1 TO USER2;

Here some schema privileges and its' description

Privilege	**Description**
SELECT	Privilege to select rows in a table.
INSERT	Privilege to insert a row in a table.
UPDATE	Privilege to update rows in a table.
DELETE	Privilege to delete a row from a table.
ALTER	Privilege to modify a schema object.
INDEX	Privilege to create an index for a table.
REFERENCES	Privilege to create a reference constraint for a table.
TRUNCATE	Privilege to perform TRUNCATE on a table.

These privileges can be given to the whole table of specific columns also.

GRANT SELECT (COLUM1, COLUM2) ON USER1.TABLE1 TO USER2;

System Privileges:

System privileges are used to manage database. All system privileges are assign to ***SYS USER*** by default. System privileges list is given below.

- ***ALTER SYSTEM***
- ***CREATE SESSION***
- ***CREATE USER***
- ***ALTER USER***
- ***DROP USER***
- ***CREATE TABLESPACE***
- ***ALTER TABLESPACE***
- ***DROP TABLESPACE***
- ***SELECT ANY DICTIONARY***
- ***CREATE TABLE***
- ***CREATE ANY TABLE***
- ***ALTER ANY TABLE***
- ***DROP ANY TABLE***
- ***COMMENT ANY TABLE***
- ***SELECT ANY TABLE***
- ***INSERT ANY TABLE***
- ***UPDATE ANY TABLE***
- ***DELETE ANY TABLE***
- ***TRUNCATE ANY TABLE***
- ***CREATE ANY INDEX***
- ***ALTER ANY INDEX***
- ***DROP ANY INDEX***
- ***CREATE SYNONYM***
- ***CREATE ANY SYNONYM***
- ***DROP ANY SYNONYM***
- ***SYSDBA***
- ***CREATE PUBLIC SYNONYM***
- ***DROP PUBLIC SYNONYM***
- ***CREATE VIEW***
- ***CREATE ANY VIEW***
- ***DROP ANY VIEW***
- ***CREATE SEQUENCE***
- ***CREATE ANY SEQUENCE***
- ***ALTER ANY SEQUENCE***
- ***DROP ANY SEQUENCE***
- ***SELECT ANY SEQUENCE***
- ***CREATE ROLE***
- ***DROP ANY ROLE***
- ***GRANT ANY ROLE***
- ***ALTER ANY ROLE***
- ***ALTER DATABASE***
- ***CREATE PROCEDURE***
- ***CREATE ANY PROCEDURE***

- ***ALTER ANY PROCEDURE***
- ***DROP ANY PROCEDURE***
- ***EXECUTE ANY PROCEDURE***
- ***CREATE TRIGGER***
- ***CREATE ANY TRIGGER***
- ***ALTER ANY TRIGGER***
- ***DROP ANY TRIGGER***
- ***GRANT ANY OBJECT PRIVILEGE***
- ***GRANT ANY PRIVILEGE***

SYS user can grant any system privileges to user.

ADMIN OPTION:

ADMIN OPTION can be used to grant a user to share system privileges.

GRANT SELECT ANY TABLE TO USER1 WITH ADMIN OPTION;

USER1 can now ***GRANT SELECT*** privileges of ***ANY TABLE*** to any other user.

GRANT OPTION:

GRANT OPTION is used to share allowed privileges to another user. Like,

GRANT SELECT ON USER1.TABLE1 TO USER2 WITH GRANT OPTION;

This means, now USER2 can ***GRANT SELECT*** privileges of *USER1.TABLE1* **TABLE** to any other user.

Any privileges can be revoked from any user.

REVOKE SELECT ON USER1.TABLE1 FROM USER2;

ROLE:

Role is a group of privileges. It helps to ***GRANT*** a number of privileges to a number of users or single user. There some redefined roles are available.

Role	**Included Privileges**	**Description**
CONNECT	CREATE SESSION	Simple database access role. This role is required for all users.
RESOURCE	CREATE PROCEDURE	Role for creating basic schema objects within the user's own schema. This role is needed by application developers.
	CREATE SEQUENCE	

	CREATE TABLE	
	CREATE TRIGGER	
DBA		All system privileges are granted with the WITH ADMIN OPTION clause.

Beside predefine role we can ***CREATE*** any role.

CREATE ROLE ROLE1;

GRANT SELECT, INSERT, UPDATE ON USER1.TABLE1 TO ROLE1;

Like privileges, assigned role can also share with another user.

GRANT ROLE1 TO USER2 WITH ADMIN OPTION;

USER2 now can ***GRANT*** ROLE1 ***ROLE*** to another user.

Assigned ***ROLE*** can also be revoked.

REVOKE ROLE1 FROM USER2;

Activating and inactivating roles is very useful to control user privileges. To grant particular privileges to a user when executing an application, use the SET ROLE command to activate the role of the privileges at the start of the program and to inactivate the role before exiting the program.

SET ROLE RESOURCE; #Turns on the RESOURCE roles

SET ROLE ALL EXCEPT RESOURCE; # Turns on all roles except RESOURCE

SET ROLE ALL; #Turns on all roles

SET ROLE NONE; # Turns off all roles

All roles given to the user will be activated by default. We can define default ***ROLE*** of a user.

ALTER USER *USER2*DEFAULT ROLE *ROLE1*;

The following ***VIEW***s will help us find details of a user and its' objects.

View	**Description**
ALL_USERS	Basic information about all users within the database.
DBA_USERS	Detailed information about all users within the database.
USER_USERS	Information about the current user.
DBA_SYS_PRIVS	Information about the system privileges granted to all users.
USER_SYS_PRIVS	Information about the system privileges granted to the current user.
DBA_TBL_PRIVS	Information about all schema object privileges in the database.
USER_TBL_PRIVS	Information about all schema object privileges that the current user owns.
ALL_TBL_PRIVS	Information about all schema object privileges owned by the current user and of all schema object privileges that are owned by the public user.

DBA_COL_PRIVS	Information about object privileges of all columns in the database.
USER_COL_PRIVS	Information about object privileges of columns for which the current user is the object owner, granter, or grantee.
ALL_COL_PRIVS	Information about object privileges of columns for which the current user or public user is the object owner or granter.
USER_COL_PRIVS_MADE	Information about object privileges of columns for which the current user is the granter.
ALL_COL_PRIVS_MADE	Information about object privileges of columns for which the current user is the object owner or granter.
USER_COL_PRIVS_RECD	Information about object privileges of columns for which the current user is the grantee.
ALL_COL_PRIVS_RECD	Information about object privileges of columns for which the current user or public user is the grantee.
DBA_ROLES	Information about all roles.
DBA_ROLE_PRIVS	Information about all roles granted to users or other roles.
USER_ROLE_PRIVS	Information about the roles granted to the current user or the public user.
ROLE_SYS_PRIVS	Information about the system privileges granted to the roles that can be accessed by the current user.
ROLE_TAB_PRIVS	Information about the system object privileges granted to the roles that can be accessed by the current user.
ROLE_ROLE_PRIVS	Information about the other roles granted to the roles that can be accessed by the current user.
DBA_AUDIT_TRAIL	All audit trails that are saved in a database.
USER_AUDIT_TRAIL	Audit trails of the current user that are saved in a database.

We can remove unwanted users.

```
DROP USER USER1;
```