**Oracle User Management**

To check all users inside the database

SQL>select username, account\_status, default\_tablespace from dba\_users;

To check the current user

SQL> show user;

1. Create a user

create user FOYEZ identified by FOYEZ

PROFILE DEFAULT

DEFAULT TABLESPACE USERS

TEMPORARY TABLESPACE TEMP;

The minimum privilege required to connect to a database is to create a session

grant create session to FOYEZ;

2. Change the password of a user

alter user FOYEZ identified by FOYEZ#91234;

3. Lock/unlock a user

alter user FOYEZ account lock;

alter user FOYEZ account unlock;

#### 4. Make a user password expiry

When we make a user id expiry, then when the user does log in, it will prompt him to set a new password.

alter user dev\_class account expire;

Change User Default Tablespace

alter user FOYEZ default tablespace TABLESPACENAME;

select username,default\_tablespace from dba\_users where username=FOYEZ';

**Tablespace Quota**

SQL> Alter user FOYEZ quota 100M on users;

SQL> ALTER USER FOYEZ QUOTA UNLIMITED ON users;

Or you can give unlimited quota to a user on all tablespace as follows.

GRANT UNLIMITED TABLESPACE TO FOYEZ;

**Tablespace used for a schema**

select distinct tablespace\_name from dba\_segments where owner='FOYEZ';

### PRIVILEGES

A privilege is permission to execute either a particular type of SQL statement or to perform a particular action on database objects.

**Two types of privilege**

1. SYSTEM PRIVILEGE

2. OBJECT PRIVILEGE

### **SYSTEM PRIVILEGE**

A system privilege is the right to perform a particular action or to perform an action on any object of a particular type.

**List of all system privileges**

SQL> select distinct privilege from dba\_sys\_privs;

#### Grant a system privilege to a user

Grant create any table, alter any table to FOYEZ;

select privilege, grantee from dba\_sys\_privs where grantee='FOYEZ';

**Revoke a system privilege from a user**

REVOKE create any table from FOYEZ;

### **OBJECT PRIVILEGE**

An object privilege is the right to perform a particular action on an object or to access another user’s object.

#### list of object privileges

SQL> select distinct privilege from DBA\_TAB\_PRIVS;

#### Grant object privilege

grant insert, update, delete on HR.EMPLOYEES to FOYEZ;

-- grant execute on a procedure

grant execute on HR.DAILYPROC to FOYEZ;

-- View the granted object privilege:

select grantee,owner,table\_name,privilege from dba\_tab\_privs where grantee='FOYEZ';

#### Revoke object privilege

revoke update, delete on HR.EMPLOYEES from FOYEZ;

## **Grant**SELECT**on all tables in a schema to a user**

CREATE PROCEDURE grant\_select(

username VARCHAR2,

grantee VARCHAR2)

AS

BEGIN

FOR r IN (

SELECT owner, table\_name

FROM all\_tables

WHERE owner = username

)

LOOP

EXECUTE IMMEDIATE

'GRANT SELECT ON '||r.owner||'.'||r.table\_name||' to ' || grantee;

END LOOP;

END;

This example grants the SELECT object privileges of all tables that belong to the user HR to the user FOYEZ:

EXEC grant\_select('HR', 'FOYEZ');

**ROLE**

A role is a collection of privileges. It allows easier management of privileges.

Create a role

create role DEV\_ROLE;

grant create session to dev\_role;

grant select any table to dev\_role;

-- any table that allows the grantee to **Query tables, views, or materialized views in any schema except SYS**

grant insert on HR.EMPLOYEES to dev\_role;

-- List of SYSTEM privileges granted to a ROLE

SQL> select role, privilege from role\_sys\_privs where role='DEV\_ROLE';

-- List of OBJECT privileges granted to ROLE;

SQL> select role,owner,table\_name,privilege from role\_tab\_privs where role='DEV\_ROLE';

Grant role to a User

grant dev\_role to FOYEZ

-- List of the user and granted role:

SQL> select grantee,GRANTED\_ROLE from dba\_role\_privs where granted\_role='DEV\_ROLE';

#### Drop a user

Dropping a user will drop all the objects it owns.

drop user FOYEZ cascade;

#### Drop a Role

Drop role DEV\_ROLE;

#### PROFILE:

A profile enforces a set of password security rules and resource usage limits.  
While creating a user if no profile is mentioned, then a DEFAULT profile will be assigned.

#### DEFAULT PROFILE SETTING:

SQL> select profile,resource\_name,RESOURCE\_TYPE,limit from dba\_profiles where profile='DEFAULT';

\*SESSION\_PER\_USER – No. of allowed concurrent sessions for a user

\*CPU\_PER\_SESSION – CPU time limit for a session, expressed in hundredth of seconds.

\*CPU\_PER\_CALL – Specify the CPU time limit for a call (a parse, execute, or fetch), expressed in hundredths of seconds.

\*CONNECT\_TIME – Specify the total elapsed time limit for a session, expressed in minutes.

\*IDLE\_TIME – Specify the permitted periods of continuous inactive time during a session, expressed in minutes.

\*LOGICAL\_READS\_PER\_SESSION – Specify the permitted number of data blocks read in a session, including blocks read from memory and disk

\*LOGICAL\_READS\_PER\_CALL –permitted number of data blocks read for a call to process a SQL statement (a parse, execute, or fetch).

\*PRIVATE\_SGA – SGA a session can allocate in the shared pool of the system global area (SGA), expressed in bytes.

\*FAILED\_LOGIN\_ATTEMPTS – No. of failed attempts to log in to the user account before the account is locked

\*PASSWORD\_LIFE\_TIME: No. of days the account will be open. after that it will expiry.

\*PASSWORD\_REUSE\_TIME: number of days before which a password cannot be reused

\*PASSWORD\_REUSE\_MAX: number of days before which a password can be reused

\*PASSWORD\_LOCK\_TIME: Number of days the user account remains locked after failed login

\*PASSWORD\_GRACE\_TIME: Number of grace days for user to change password

\*PASSWORD\_VERIFY\_FUNCTION: PL/SQL that can be used for password verification

CREATE PROFILE "APP\_PROFILE"

LIMIT

COMPOSITE\_LIMIT UNLIMITED

SESSIONS\_PER\_USER UNLIMITED

CPU\_PER\_SESSION UNLIMITED

CPU\_PER\_CALL UNLIMITED

LOGICAL\_READS\_PER\_SESSION UNLIMITED

LOGICAL\_READS\_PER\_CALL UNLIMITED

IDLE\_TIME 90

CONNECT\_TIME UNLIMITED

PRIVATE\_SGA UNLIMITED

FAILED\_LOGIN\_ATTEMPTS 10

PASSWORD\_LIFE\_TIME 180

PASSWORD\_REUSE\_TIME UNLIMITED

PASSWORD\_REUSE\_MAX UNLIMITED

PASSWORD\_VERIFY\_FUNCTION NULL

PASSWORD\_LOCK\_TIME UNLIMITED

PASSWORD\_GRACE\_TIME UNLIMITED;

***Note:*** *password lock time by default is for 1 day. You can specify it in minutes (n/1440) or even in seconds (n/86400)*

#### Alter a profile

ALTER PROFILE APP\_PROFILE LIMIT FAILED\_LOGIN\_ATTEMPS UNLIMITED;

#### Change the profile of a user

SQL> select username, profile from dba\_users where username='FOYEZ';

SQL> ALTER HR PROFILE APP\_PROFILE;

#### How to make a user non-expiry

#### Usually, application users need to set non-expiry. I.e it will never expire. To set it, we need to either create a profile with PASSWORD\_LIFE\_TIME UNLIMITED or alter the profile of that user.

SQL> select username,profile,EXPIRY\_DATE

from dba\_users

where username='FOYEZ';

ALTER PROFILE APP\_PROFILE LIMIT PASSWORD\_LIFE\_TIME UNLIMITED;

Find which user gonna expire and alter them:

SELECT ' alter user '

|| u.NAME

|| ' identified by values '''

|| u.PASSWORD

|| ''';'

status

FROM SYS.user$ u, dba\_users d

WHERE u.user# = d.user\_id

AND account\_status NOT LIKE '%LOCKED%'

AND TRUNC (d.expiry\_date) BETWEEN TRUNC (SYSDATE - 30)

AND TRUNC (SYSDATE + 30);

## Find User Permissions

To check system privileges granted to a user

select privilege from dba\_sys\_privs where grantee='SCOTT';

To check object level privileges granted to a user or role

SQL> select owner, table\_name, privilege from dba\_tab\_privs where grantee='SALES\_CLERK';

To check roles assigned to a user

select granted\_role from dba\_role\_privs where grantee='SCOTT';

To check permissions assigned to role

select privilege from role\_sys\_privs where role='SALES\_CLERK';

select owner, table\_name, privilege from role\_tab\_privs where role='SALES\_CLERK';

To check roles granted to another role

SQL> select granted\_role from role\_role\_privs where role='SALES\_CLERK';

CREATE PROFILE "DEFAULT" LIMIT

SESSIONS\_PER\_USER UNLIMITED

CPU\_PER\_SESSION UNLIMITED

CPU\_PER\_CALL UNLIMITED

CONNECT\_TIME UNLIMITED

IDLE\_TIME UNLIMITED

LOGICAL\_READS\_PER\_SESSION UNLIMITED

LOGICAL\_READS\_PER\_CALL UNLIMITED

COMPOSITE\_LIMIT UNLIMITED

PRIVATE\_SGA UNLIMITED

FAILED\_LOGIN\_ATTEMPTS 10

PASSWORD\_LIFE\_TIME 180

PASSWORD\_REUSE\_TIME UNLIMITED

PASSWORD\_REUSE\_MAX UNLIMITED

PASSWORD\_LOCK\_TIME 1

PASSWORD\_GRACE\_TIME 7

PASSWORD\_VERIFY\_FUNCTION NULL;

CREATE PROFILE ORA\_STIG\_PROFILE LIMIT

SESSIONS\_PER\_USER DEFAULT

CPU\_PER\_SESSION DEFAULT

CPU\_PER\_CALL DEFAULT

CONNECT\_TIME DEFAULT

IDLE\_TIME 15

LOGICAL\_READS\_PER\_SESSION DEFAULT

LOGICAL\_READS\_PER\_CALL DEFAULT

COMPOSITE\_LIMIT DEFAULT

PRIVATE\_SGA DEFAULT

FAILED\_LOGIN\_ATTEMPTS 3

PASSWORD\_LIFE\_TIME 60

PASSWORD\_REUSE\_TIME 365

PASSWORD\_REUSE\_MAX 10

PASSWORD\_LOCK\_TIME UNLIMITED

PASSWORD\_GRACE\_TIME 5

PASSWORD\_VERIFY\_FUNCTION ORA12C\_STIG\_VERIFY\_FUNCTION;

https://dbaclass.com/article/user-management-in-oracle/

https://www.support.dbagenesis.com/post/users-roles-profiles-in-oracle