Cognizant | Academy

Business Focused. Learner Centric.

Oracle Version 12c

User Access

Enabling Objectives

After completing this chapter, in the next 90 minutes you will be able to :

- Create password & resource profiles.
- Create users.
- Assigning system privileges to users.
- Create Roles.

Key Topics

- Create password and resource profiles.
- Create and Alter Users.

Create Profile

Password Profile

- CREATE PROFILE is used to create password and resource profiles.

```
CREATE  PROFILE p_name LIMIT
FAILED_LOGIN_ATTEMPTS < n | UNLIMITED | DEFAULT>
PASSWORD_GRACE_TIME  < n | UNLIMITED | DEFAULT>
PASSWORD_LIFE_TIME < n | UNLIMITED | DEFAULT>
PASSWORD_LOCK_TIME <n | UNLIMITED | DEFAULT>
PASSWORD_REUSE_MAX <n | UNLIMITED | DEFAULT>
PASSWORD_REUSE_TIME <n | UNLIMITED | DEFAULT>
```

Password Profile

Parameter	Description
FAILED_LOGIN_ATTEMPTS	The number of failed attempts to log in to the use account before the account is locked
PASSWORD_GRACE_TIME	The number of days after the grace period begins during which a warning is issued and login is allowed.
PASSWORD_LIFE_TIME	The number of days the same password can be used for authentication.
PASSWORD_LOCK_TIME	The number of days an account will remain locked after the specified number of consecutive failed login attempts defined by FAILED_LOGIN_ATTEMPTS

Password Profile

Parameter	Description
PASSWORD_REUSE_MAX	The number of times a password can be reused
PASSWORD_REUSE_TIME	The number of days between reuses of a password

Example:

```
create profile pass_profile limit
        failed_login_attempts 3
        password_grace_time 10
        password_life_time 180
        password_lock_time 30
        password_reuse_max  0
        password_reuse_time  0;
```

Resource Profile

Syntax

CREATE PROFILE **p_name** LIMIT
CONNECT_TIME **<n | UNLIMITED | DEFAULT>**
CPU_PER_CALL **< n | UNLIMITED | DEFAULT>**
CPU_PER_SESSION **< n | UNLIMITED | DEFAULT>**
IDLE_TIME **< n | UNLIMITED | DEFAULT>**
LOGICAL_READS_PER_CALL **< n | UNLIMITED | DEFAULT>**
LOGICAL_READS_PER_SESSION **< n | UNLIMITED | DEFAULT>**
SESSIONS_PER_USER **< n | UNLIMITED | DEFAULT>**

Resource Profile

Parameter	Description
CONNECT_TIME	Allowable connect time per session in minutes
CPU_PER_CALL	Maximum CPU time per call (100th of a second)
CPU_PER_SESSION	Maximum CPU time per session (100th of a second)
IDLE_TIME	Allowed idle time before user is disconnected (minutes)
LOGICAL_READS_PER_CALL	Maximum number of database blocks read per call

Password Profile

Parameter	Description
LOGICAL_READS_PER SESSION	Maximum number of database blocks read per session
SESSIONS_PER_USER	Number of concurrent multiple sessions allowed per user

Example:

```
create profile res_profile limit
connect_time 600
cpu_per_session  unlimited
idle_time  20
logical_reads_per_session unlimited
sessions_per_user 1;
```

Create Users

Create Users

Syntax

```
CREATE USER user_name IDENTIFIED BY password
[ IDENTIFIED EXTERNALLY]
[ IDENTIFIED GLOBALLY AS external_name ]
[ PASSWORD EXPIRE]
[ DEFAULT TABLESPACE def_tablespace]
[TEMPORARY TABLESPACE temp_tablespace]
[ PROFILE profile_name]
[ ACCOUNT  { LOCK | UNLOCK} ]
```

Creating User

Parameter	Description
user_name	specifies name of the database user
password	specifies the password
Externally	creates an external user. Such a user must be identified (authorized) by an external service, such as OS or third party service
Globally	creates global users, authorized by enterprise directory service (Oracle Internet Dir)
def_tablespace	specifies the default tablespace where objects are stored
temp_tablespace	default tablespace where temporary objects are stored

Creating User

Parameter	Description
Password expire	if specified user is required to change the password as soon as he connects for the first time
Profile	Used to control resource usage and specify the password control mechanisms
Account Lock / Unlock	Can be used to lock or unlock users account (Unlock is the Default)

Example:

```
create user smith identified by pass
password expire
default tablespace user
temporary tablespace temp;
```

Create User Operating System Authentication

Example:

```
Create user Smith identified externally
default tablespace users
temporary tablespace temp;
```

Result:

The operating System user, Smith is authenticated by the OS.

This option is generally used when user logs on directly to the machine where Oracle Server is running.

Alter User

- ALTER USER command is used to alter the parameters associated with the create user command :

```
ALTER USER user_name
[ IDENTIFIED BY new_password ]
[ IDENTIFIED EXTERNALLY ]
[ IDENTIFIED GLOBALLY AS external_name ]
[ DEFAULT TABLESPACE def_tablespace ]
[ TEMPORARY TABLESPACE temp_tablespace ]
[ PASSWORD EXPIRE]
```

System Privileges

System privileges

System Privilege	Allows you to
CREATE SESSION	Connects to a database
CREATE SEQUENCE	Create sequences in the grantee's own schema
CREATE TABLE	Create table in the grantee's own schema
CREATE ANY TABLE	Create a table in any schema
DROP TABLE	Drop table in the grantee's own schema
DROP ANY TABLE	Drop a table from any schema
CREATE PROCEDURE	Create new procedures, functions, and packages in the grantees own schema

System privileges

System Privilege	Allows you to
CREATE ANY PROCEDURE	Create new procedures, functions, and packages in any schema
EXECUTE ANY PROCEDURE	Execute a procedure in any schema
CREATE USER	Create a user
DROP USER	Drop a user
CREATE VIEW	Create a view
CREATE SYNONYM	Create a synonym

Grant system privileges

- By default a DBA has all the system privileges
- GRANT command - is used for granting privileges to a user
- WITH ADMIN OPTION – is enable a user to grant a privilege to another user
- PUBLIC - can be used to grant the privilege to all the users.

Example :

- connect system/manager
- Grant create session, create user, create table to smith;
- Grant execute any procedure to smith with admin option;
- Grant create table to public;

System privileges

- System privileges assigned to a user can be checked by querying : user_sys_privs;

Example:

- connect smith/password
- select * from user_sys_privs;

USERNAME	PRIVILEGE	ADMIN_OPTION
NO	PUBLIC	CREATE TABLE
NO	SMITH	CREATE SESSION
NO	SMITH	CREATE TABLE
NO	SMITH	CREATE USER
NO	SMITH	EXECUTE ANY PROCEDURE

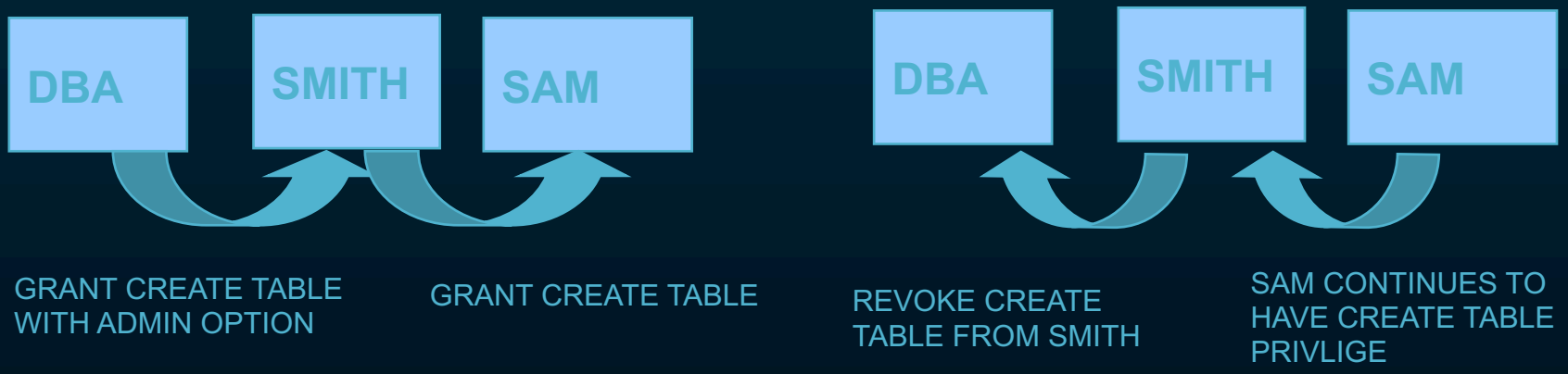
Revoke privileges

- REVOKE command takes back privileges from a user.

Example

- revoke create user from smith

- select * from user_sys_privs;



Object privileges

- Allows a user to perform certain actions on database objects.

Object Privilege	Table	View	Sequence	Procedure
ALTER	X	-----	X	-----
DELETE	X	X	-----	-----
EXECUTE	-----	-----	-----	X
INDEX	X	-----	-----	-----
INSERT	X	X	-----	-----
REFERENCES	X	X	-----	-----
SELECT	X	X	X	-----
UPDATE	X	X	-----	-----

Granting Object Privileges to a User

- GRANT : is used for granting privileges to a user
- WITH GRANT : enables a user to grant a privilege to another user
- PUBLIC : can be used to grant the privilege to all the users

Example

- connect hr/hr;
- Grant update(dept_id, m_id) on hr.employee to smith;
- Grant select on hr.employee to smith with grant option

Object Privileges checking

- Table Object privileges one has assigned to other user can be checked by querying : user_tab_privs.

Example :

```
select grantee, grantor, privilege, grantable from
user_tab_privs    where table_name='EMPLOYEE;
```

GRANTEE	GRANTOR	PRIVILEGE	GRANTABLE
SMITH	HR	UPDATE	NO
SMITH	HR	SELECT	YES

Object Privileges checking

- Column Object privileges one has assigned to other user can be checked by querying : user_col_privs.

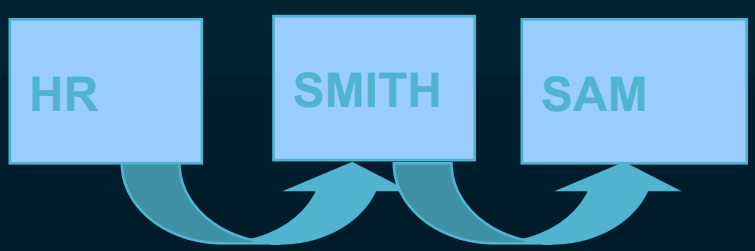
Example :

```
select grantee, grantor, privilege, column_name, grantable
from user_col_privs where    table_name='EMPLOYEE;
```

GRANTEE	GRANTOR	PRIVILEGE	COLUMN_NAME	GRANTABLE
SMITH	HR	UPDATE	DEPT_ID	NO
SMITH	HR	UPDATE	M_ID	NO

Revoke Object Privileges

- REVOKE command takes back privileges from a user.
- Example
  - revoke select on employee from smith



GRANT SELECT ON EMPLOYEE WITH GRANT OPTION

GRANT SELECT ON EMPLOYEE

REVOKE SELECT ON EMPLOYEE FROM SMITH

SELECT PRIVILEGE ON EMPLOYEE REVOKED FROM SMITH AND SAM

Create Roles

Roles

- Is a group of privileges.
- Helps in managing privileges.
- Can assign password to a role.
- When privileges are added or deleted from a role, all users and roles assigned that role automatically receive or lose that privilege.
- CREATE ROLE command is used for creating roles:
- Example.
- create role rmg; - Creates a role named rmg.
- create role manager identified by manager_pass; - Creates a role named manager with password manager_pass

Grant Privilege

- Add privileges to the role with GRANT command.

  Example :

```
grant select, insert, update on employee to rmg;
grant create user, create session, create table to manager;
grant manager to sam;
grant rmg to smith;
```

- Object privileges are granted to the role, rmg
- System privileges are granted to the role, manager
- The roles are then granted to sam and smith

Checking Roles

- Roles that have been granted to a user can be queried from : user_role_privs;
- System privileges granted to roles can be queried from : role_sys_privs;
- Object Privileges granted to a role can queried from : role_tab_privs
- Default Role: By default a role granted to a user is enabled.

Checking Roles

- For security purposes a role can remain disabled by default.
- A user will need to enable it to as and when required.

Example:

```
alter user smith default role all except rmg;
```

Result : When you login as smith, you will need to enable rmg using SET ROLE command as stated below.

- set role rmg;

Practice Check

1.Which of the following will check the system privileges ?

a. user_sys_privs
b. user_role_privs
c. user_col_privs

2. _____ command takes back privileges from a user.

a. Remove
b. Cancel
c. Grant
d. Revoke

RECAP

In this chapter we have learnt how to:

- Create password & resource profiles.
- Create users.
- Assigning system privileges to users.
- Create Roles.

You have successfully completed -

Controlling user access.