

UNIT 4

Managing Users, Roles and Privileges

(6 Hours) — Database Administration (CACS405)

*Note: These notes are prepared for study purposes and **may contain errors or omissions**. Syntax, file paths and screen names differ between Oracle versions and operating systems (examples here follow Oracle 19c on Linux). Please verify important points with your textbook, class notes and the official Oracle documentation, and test commands on a practice database, never on a production system.*

4.1 Database Users and Schemas

A **database user** is an account that can connect to the database. Each user owns a **schema**: the collection of objects (tables, views, indexes, etc.) that the user creates. Every user has an authentication method, a default tablespace, a temporary tablespace, a profile and (optionally) quotas and roles.

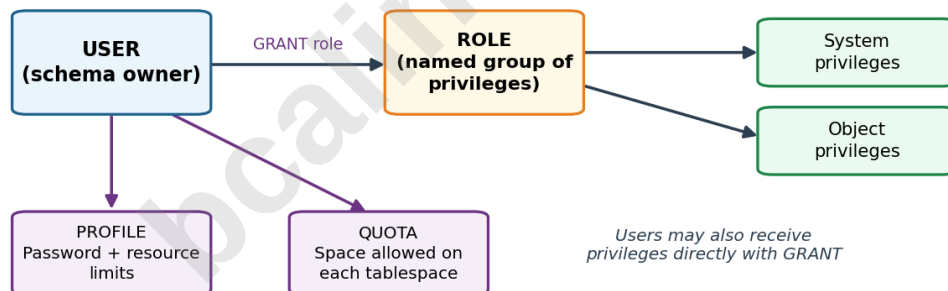


Figure 4.1: Relationship between users, roles, privileges, profiles and quotas

```
CREATE USER ram IDENTIFIED BY "Str0ng#Pass1"
  DEFAULT TABLESPACE users
  TEMPORARY TABLESPACE temp
  QUOTA 100M ON users
  PROFILE default
  PASSWORD EXPIRE -- must change the password at first login
  ACCOUNT UNLOCK;

GRANT CREATE SESSION TO ram; -- without this the user cannot log in

ALTER USER ram IDENTIFIED BY "N3w#Passw0rd";
ALTER USER ram ACCOUNT LOCK;
ALTER USER ram ACCOUNT UNLOCK;
DROP USER ram CASCADE; -- CASCADE also drops all objects owned by the user
```

Listing 4.1: Creating and altering a user

*Note: In a multitenant database, users created in the root container are **common users** and must have names starting with C## (see Unit 5). Users created inside a PDB are **local users**.*

4.2 Assigning Quotas to Users

A **quota** limits the amount of space a user's objects can occupy in a tablespace. A user with no quota on a tablespace cannot create segments there (error ORA-01950: no privileges on tablespace). Quota is not needed for temporary or undo tablespaces.

```
ALTER USER ram QUOTA 500M ON users;
ALTER USER ram QUOTA UNLIMITED ON app_data;
ALTER USER ram QUOTA 0 ON users;          -- removes quota (objects stay)

SELECT username, tablespace_name, bytes/1024/1024 used_mb,
       DECODE(max_bytes,-1,'UNLIMITED',max_bytes/1024/1024||' MB') quota
FROM   dba_ts_quotas WHERE username = 'RAM';
```

Listing 4.2: Setting and checking quotas

The system privilege `UNLIMITED TABLESPACE` gives unlimited quota on all tablespaces and should be granted rarely. It is better to assign specific quotas.

4.3 Privileges

A **privilege** is the right to perform a particular action. There are two kinds.

Kind	Meaning	Examples
System privilege	Right to perform an action on a type of object or a system-wide operation	CREATE SESSION, CREATE TABLE, CREATE USER, ALTER SYSTEM, SELECT ANY TABLE
Object privilege	Right to perform an action on one specific object	SELECT, INSERT, UPDATE, DELETE, EXECUTE, ALTER, REFERENCES on hr.employees

```

-- system privileges
GRANT CREATE SESSION, CREATE TABLE, CREATE VIEW TO ram;
GRANT CREATE USER TO dba_asst WITH ADMIN OPTION;    -- can pass it on

-- object privileges
GRANT SELECT, INSERT ON hr.employees TO ram;
GRANT UPDATE (salary) ON hr.employees TO ram;      -- column-level
GRANT SELECT ON hr.departments TO ram WITH GRANT OPTION;

REVOKE INSERT ON hr.employees FROM ram;
REVOKE CREATE TABLE FROM ram;

SELECT * FROM dba_sys_privs WHERE grantee = 'RAM';
SELECT * FROM dba_tab_privs WHERE grantee = 'RAM';
SELECT * FROM session_privs;                      -- privileges active now

```

Listing 4.3: Granting and revoking privileges

Option	Applies to	Effect of revoke
WITH ADMIN OPTION	System privileges and roles	Revoking from the first user does not revoke what that user already granted to others.
WITH GRANT OPTION	Object privileges	Revoking from the first user cascades to users they granted the privilege to.

Example 4.1

Create a user `sita` who can log in, create tables in her own schema (in the `USERS` tablespace, limited to 50 MB) and read `hr.employees`.

Solution

```

CREATE USER sita IDENTIFIED BY "Sita#2026" DEFAULT TABLESPACE users
QUOTA 50M ON users;

GRANT CREATE SESSION, CREATE TABLE TO sita;

GRANT SELECT ON hr.employees TO sita;

```

4.4 Roles

A **role** is a named group of privileges (and other roles) that is granted to users as a unit. Roles make privilege management easier: change the role once and every user with that role is affected. Roles are not owned by any schema.

Predefined role	Description
CONNECT	In recent releases (12.2 and later) only CREATE SESSION. Older releases included more privileges.

Predefined role	Description
RESOURCE	Basic object-creation privileges (create tables, sequences, procedures, etc.) for application developers.
DBA	Very powerful: most system privileges with admin option. Grant only to real administrators.
SELECT_CATALOG_ROLE	SELECT on data dictionary views.
EXP_FULL_DATABASE, IMP_FULL_DATABASE	Full export / import with Data Pump.

```

CREATE ROLE hr_clerk;
GRANT CREATE SESSION TO hr_clerk;
GRANT SELECT, INSERT, UPDATE ON hr.employees TO hr_clerk;
GRANT SELECT ON hr.departments TO hr_clerk;

GRANT hr_clerk TO ram, sita;
ALTER USER ram DEFAULT ROLE ALL;           -- roles active at login
SET ROLE hr_clerk;                         -- enable in current session

CREATE ROLE payroll_role IDENTIFIED BY "Pay#Role1"; -- password-protected role
REVOKE hr_clerk FROM sita;
DROP ROLE hr_clerk;

SELECT * FROM dba_role_privs WHERE grantee = 'RAM';
SELECT * FROM role_sys_privs;

```

Listing 4.4: Creating and using roles

Note: Privileges received through a role do not work inside definer's-rights stored procedures; those need direct grants.

4.5 Applying the Principle of Least Privilege

The **principle of least privilege** says that every user and program should have only the privileges required to do its job, and no more. This limits damage from mistakes, malware and misuse of an account.

Practice	Explanation
Grant only what is needed	Give CREATE SESSION plus specific object privileges rather than DBA or ANY privileges.
Use roles	Group privileges by job function and grant roles, not individual privileges.
Avoid ANY privileges	SELECT ANY TABLE, DROP ANY TABLE, etc. work across all schemas and are dangerous.
Limit powerful accounts	Do not use SYS/SYSTEM for applications; separate application owner and application user accounts.

Practice	Explanation
Review PUBLIC	Privileges granted to PUBLIC are available to everyone; revoke unnecessary ones (for example on UTL_FILE, UTL_HTTP packages).
Lock unused accounts	Lock and expire default/sample accounts; remove sample schemas from production.
Separate duties	Use SYSBACKUP, SYSDG, SYSKM for backup, Data Guard and key management instead of always using SYSDBA.
Audit and review	Regularly review DBA_SYS_PRIVS, DBA_ROLE_PRIVS, DBA_TAB_PRIVS; use unified auditing.

4.6 Creating and Assigning Profiles

A **profile** is a named set of limits on **password management** and **system resource usage**. Every user has exactly one profile; if none is specified, the DEFAULT profile is used.

Password parameter	Meaning
FAILED_LOGIN_ATTEMPTS	Number of consecutive failed logins before the account is locked.
PASSWORD_LOCK_TIME	Days the account stays locked after too many failures.
PASSWORD_LIFE_TIME	Days a password is valid before it must be changed.
PASSWORD_GRACE_TIME	Days after expiry during which login is still allowed with a warning.
PASSWORD_REUSE_TIME / PASSWORD_REUSE_MAX	Days before a password may be reused / number of changes before reuse.
PASSWORD_VERIFY_FUNCTION	PL/SQL function that checks password complexity.

Resource parameter	Meaning
SESSIONS_PER_USER	Maximum concurrent sessions for the user.
CPU_PER_SESSION / CPU_PER_CALL	CPU time limit (in hundredths of a second) per session / per call.
CONNECT_TIME	Maximum session length in minutes.
IDLE_TIME	Minutes of inactivity before the session is ended.
LOGICAL_READS_PER_SESSION / _PER_CALL	Limit on data blocks read.
PRIVATE_SGA, COMPOSITE_LIMIT	Private SGA memory (shared server) / weighted combined limit.

Password limits are always enforced. Resource limits are enforced only when the initialization parameter `RESOURCE_LIMIT` is `TRUE` (the default in 12c and later; in earlier releases it was `FALSE`).

```
CREATE PROFILE clerk_profile LIMIT
  FAILED_LOGIN_ATTEMPTS 5
  PASSWORD_LOCK_TIME 1
  PASSWORD_LIFE_TIME 90
  PASSWORD_GRACE_TIME 7
  PASSWORD_REUSE_MAX 5
  SESSIONS_PER_USER 2
  IDLE_TIME 30
  CONNECT_TIME 480;

ALTER USER ram PROFILE clerk_profile;
ALTER PROFILE clerk_profile LIMIT IDLE_TIME 15;
DROP PROFILE clerk_profile CASCADE;          -- users revert to DEFAULT

SELECT profile, resource_name, limit FROM dba_profiles
WHERE profile = 'CLERK_PROFILE';
```

Listing 4.5: Profile creation and assignment

4.7 Administering User Authentication Methods

Authentication verifies the identity of a user before allowing access. Oracle supports several methods.

Method	How it works	Example
Password (database)	Password stored (as a hash) in the data dictionary; checked by the database.	CREATE USER ram IDENTIFIED BY pwd;
OS (external)	Database trusts the operating system login; no password is sent.	CREATE USER ops\$ali IDENTIFIED EXTERNALLY;
Network / strong	Kerberos, SSL/TLS certificates or RADIUS.	Configured in <code>sqlnet.ora</code>
Global (directory)	User identity is stored in a central LDAP directory (enterprise users).	IDENTIFIED GLOBALLY AS '...'
Proxy	A middle-tier user connects on behalf of the end user.	ALTER USER ram GRANT CONNECT THROUGH app_user;
Password file (SYSDBA)	Administrators connecting remotely are checked against the password file created with <code>orapwd</code> .	REMOTE_LOGIN_PASSWORDFILE = EXCLUSIVE

4.7.1 Administrative Privileges

Privilege	Typical use
SYSDBA	Full control: startup/shutdown, create/recover database. Connects as SYS.
SYSOPER	Startup, shutdown, backup, archive; cannot view user data.
SYSDG / SYSKM	Backup and recovery tasks (RMAN) only.
SYSDG / SYSKM	Data Guard administration / encryption key management.

```
$ sqlplus / as sysdba -- OS authentication (dba group)
$ orapwd file=$ORACLE_HOME/dbs/orapworcl password=Adm#1n entries=10
GRANT SYSBACKUP TO backup_admin;

ALTER SYSTEM KILL SESSION '45,1234' IMMEDIATE; -- SID,SERIAL# from V$SESSION
SELECT username, account_status, lock_date, expiry_date, profile FROM dba_users;
```

Listing 4.6: Administrative access and user monitoring

Example 4.2

Users complain that they are locked out after typing a wrong password three times. Which setting controls this and how do you unlock an account?

Solution

FAILED_LOGIN_ATTEMPTS in the user's profile controls it (PASSWORD_LOCK_TIME sets how long the lock lasts).

Unlock manually: ALTER USER ram ACCOUNT UNLOCK;

Key Takeaways

- ✓ A **user** owns a **schema**; needs CREATE SESSION to log in and a **quota** to store data in a tablespace.
- ✓ **System privileges** allow actions; **object privileges** apply to specific objects. WITH ADMIN OPTION and WITH GRANT OPTION allow re-granting.
- ✓ **Roles** bundle privileges and simplify administration; avoid giving DBA or ANY privileges casually.
- ✓ **Least privilege**: give the minimum needed, use roles, restrict PUBLIC, lock unused accounts, audit regularly.
- ✓ **Profiles** set password rules (lockout, expiry, reuse) and resource limits (sessions, idle time, CPU).
- ✓ Authentication can be by password, OS, network/strong methods, directory (global), proxy, or password file for SYSDBA.

Reminder: these notes may contain errors — cross-check with the official syllabus, your textbook and Oracle documentation.

— End of Unit 4 —

bcalink.com