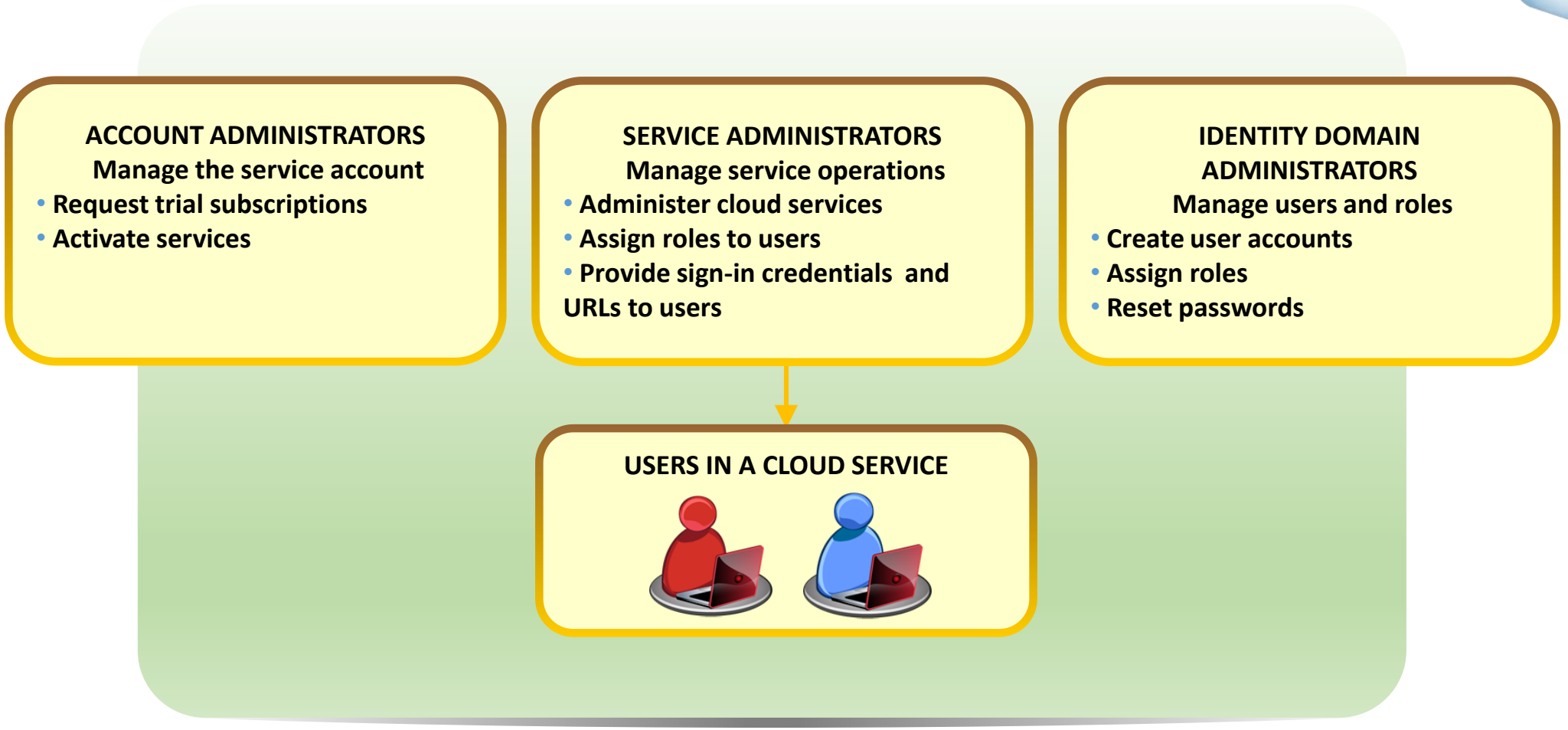
Administering User Security

Objectives

- After completing this lesson, you should be able to:
  - Create database users
  - Grant privileges to database users
  - Create and grant roles to users or other roles
  - Revoke privileges and roles from users and other roles
  - Create and assign profiles to users
  - Explain the various authentication options for users
  - Assign quota to users
  - Apply the principle of least privilege

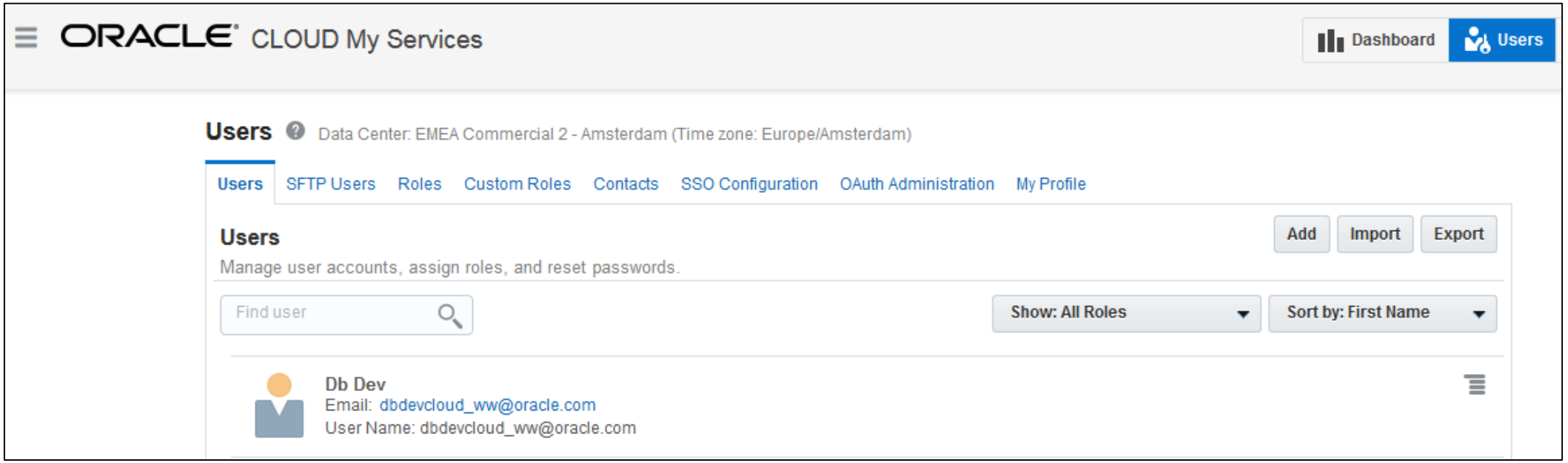
Oracle Cloud User Roles and Privileges



See Oracle Cloud User Roles and Privileges in *Getting Started with Oracle Cloud* for additional information.

Administering Oracle Cloud Users, Roles, and Privileges

- Administer Cloud Services users by accessing the Users page.
- Cloud Services users are different from Oracle Database users.



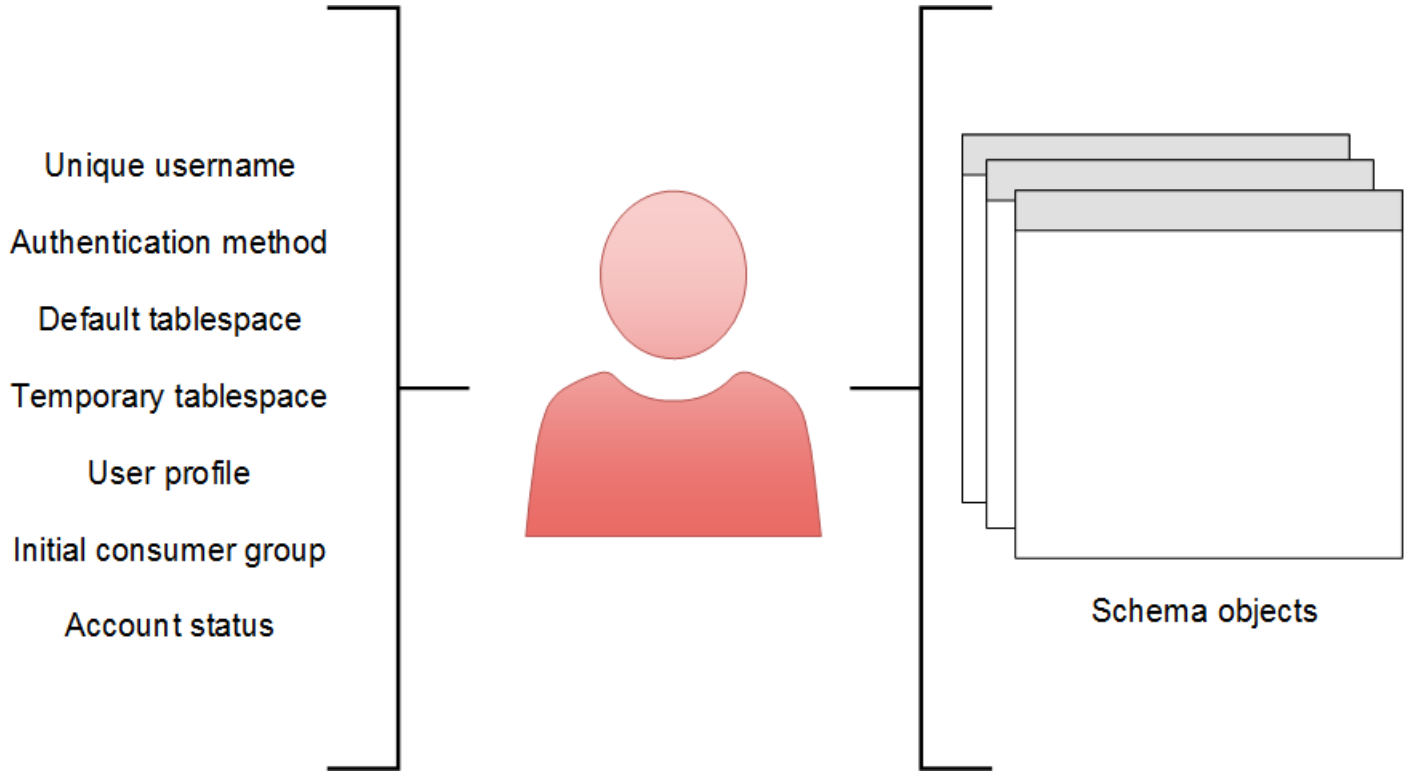
See Creating a User and Assigning a Role in *Getting Started with Oracle Cloud* for details.

Managing Oracle Cloud Compute Node Users

- When a database deployment is created, three Linux users are created.

OS User	Authorization
`opc`	Authorized to log in to the compute node Authorized to run `root` commands Can use `sudo -s`
`oracle`	Authorized to log in to the compute node Not authorized to run `root` commands
`root`	Not authorized to log in to the compute node

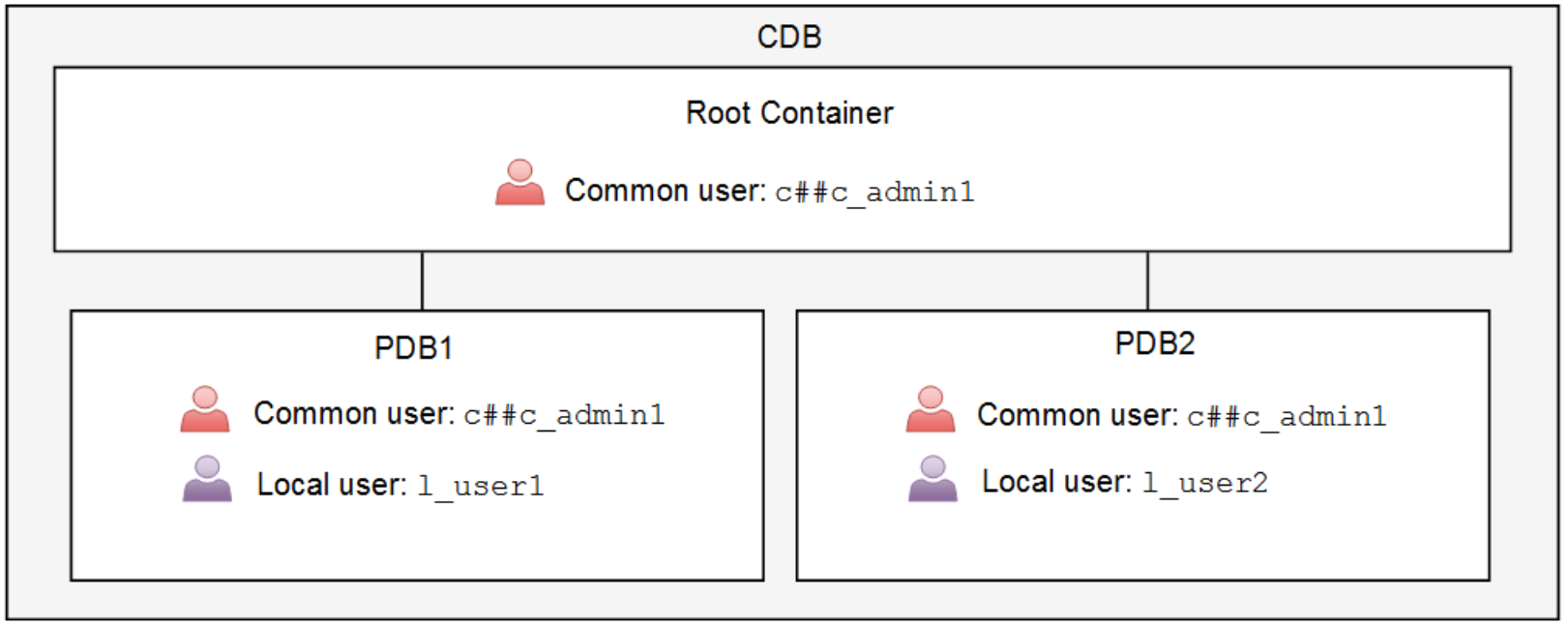
Database User Accounts



Oracle-Supplied Administrator Accounts

Account	Description
SYS	Super user. Owns the data dictionary and the Automatic Workload Repository (AWR). Used for starting up and shutting down the database instance.
SYSTEM	Owns additional administrative tables and views
SYSBACKUP	Facilitates Oracle Recovery Manager (RMAN) backup and recovery operations
SYSDG	Facilitates Oracle Data Guard operations
SYSKM	Facilitates Transparent Data Encryption wallet operations
SYSRAC	For Oracle Real Application Clusters (RAC) database administration tasks
SYSMAN	For Oracle Enterprise Manager database administration tasks
DBSNMP	Used by the Management Agent component of Oracle Enterprise Manager to monitor and manage the database

Creating Oracle Database Users in a Multitenant Environment



Schema-Only Account

Ensure that a user cannot log in to the instance:

- Enforce data access through the application.
- Secure schema objects.
  - Prevent objects from being dropped by the connected schema.
- Use the `NO AUTHENTICATION` clause.
  - Can be replaced by `IDENTIFIED BY VALUES`
- A schema-only account cannot be:
  - Granted system administrative privileges
  - Used in database links

```
DBA_USERS
   AUTHENTICATION_TYPE = NONE | PASSWORD
```

Authenticating Users

- Every user, including administrators, must be authenticated when connecting to a database instance.
- Authentication verifies that the user is a valid database user and establishes a trust relationship for further interactions.
- Authentication also enables accountability by making it possible to link access and actions to specific identities.
- The following authentication methods are possible:
  - Password (usually for database users)
  - Operating system (OS) authentication
  - Password file (for system administrative privileged users only)
  - Strong authentication with Kerberos, SSL, or directory authentication
- A system administrative privileged user must use OS authentication, password file authentication, or strong authentication. These methods can authenticate when the database is available or unavailable (not started).

Password Authentication

- Password authentication is also referred to as "authentication" by the Oracle Database server.
- Create each user with an associated password that must be supplied when the user attempts to establish a connection.
- When setting up a password, you can expire the password immediately, which forces the user to change the password after first logging in.
  - If you decide on expiring user passwords, make sure that users have the ability to change the password. Some applications do not have this functionality.
  - All passwords created in Oracle Database are case-sensitive by default.
  - Passwords may contain multibyte characters and are limited to 30 bytes.
- Passwords are always automatically and transparently encrypted by using the Advanced Encryption Standard (AES) algorithm during network (client/server and server/server) connections before sending them across the network.

Password File Authentication

- You can use password file authentication for an Oracle database instance and for an Oracle Automatic Storage Management (Oracle ASM) instance.
- If authentication succeeds, the connection is logged with the `SYS` user.
- A password file stores database usernames and case-sensitive passwords for administrator users (common and local administrators).
- DBCA creates a password file during installation.
- To prepare for password file authentication, you must:
  - Create the password file.
  - Set the `REMOTE_LOGIN_PASSWORDFILE` initialization parameter.
  - Grant system administrative privileges (for example, `GRANT SYSDBA TO mydba`).
- Use the `CONNECT` command in SQL*Plus to connect. For example:

```
SQL> CONNECT mydba AS SYSDBA
```

OS Authentication

- Oracle Universal Installer creates operating system groups, assigns them specific names, and maps each group to a specific system privilege.
  - Example: Members of the dba group are granted `SYSDBA`
- As a group member, you can be authenticated, enabled as an administrative user, and connected to a local database:

```
SQL> CONNECT / AS SYSDBA
 SQL> CONNECT / AS SYSOPER
 SQL> CONNECT / AS SYSBACKUP
 SQL> CONNECT / AS SYSDG
 SQL> CONNECT / AS SYSKM
 SQL> CONNECT / AS SYSRAC
```

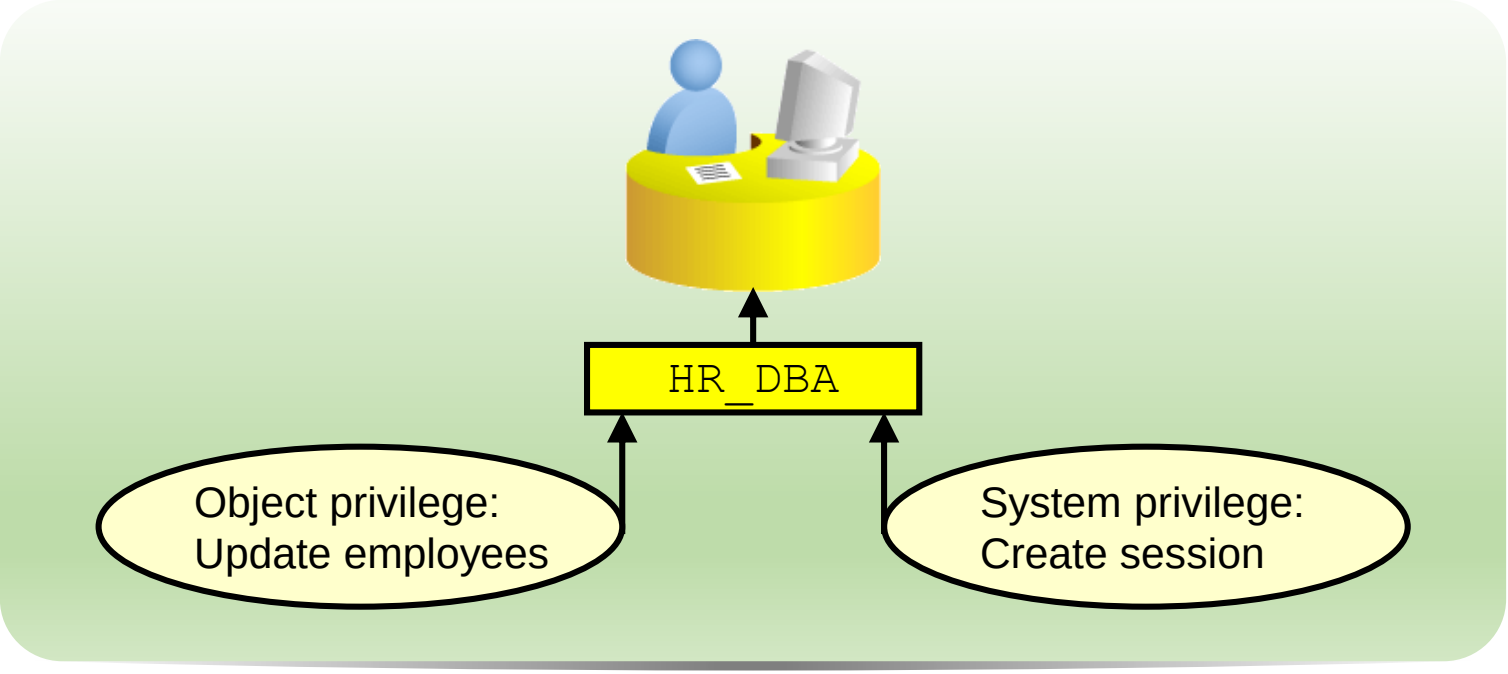
- If you are not a member of one of these OS groups, you will not be able to connect as an administrative user via OS authentication.

OS Authentication for Privileged Users

OS Group	UNIX or Linux User Group	Special System Privilege Granted to Members
Oracle Software Group (top level group)	`oinstall`	Allowed to create and delete database files on the OS. All database administrators belong to this group.
Database Administrator Group (`OSDBA`)	`dba`	`SYSDBA` (Connects you as the `SYS` user)
Database Operator Group (`OSOPER`) – optional	`oper`	`SYSOPER` (Connects you as the `PUBLIC` user)
Database Backup and Recovery Group (`OSBACKUPDBA`)	`backupdba`	`SYSBACKUP`
Data Guard Administrative Group (`OSDGDBA`)	`dgdba`	`SYSDG`
Encryption Key Management Administrative Group (`OSKMDBA`)	`kmdba`	`SYSKM`
Real Application Cluster Administrative Group (`OSRACDBA`)	`rac`	`SYSRAC`

Privileges

- There are two types of user privileges:
  - System: Enables users to perform particular actions in the database
  - Object: Enables users to access and manipulate a specific object



System Privileges

- Each system privilege allows a user to perform a particular database operation or class of database operations.
- Administrators have special system privileges.
- A system privilege with the `ANY` clause means the privilege applies to all schemas, not just your own.
- If you grant a system privilege with the `ADMIN OPTION` enabled, you enable the grantee to administer the system privilege and grant it to other users.

System Privileges for Administrators

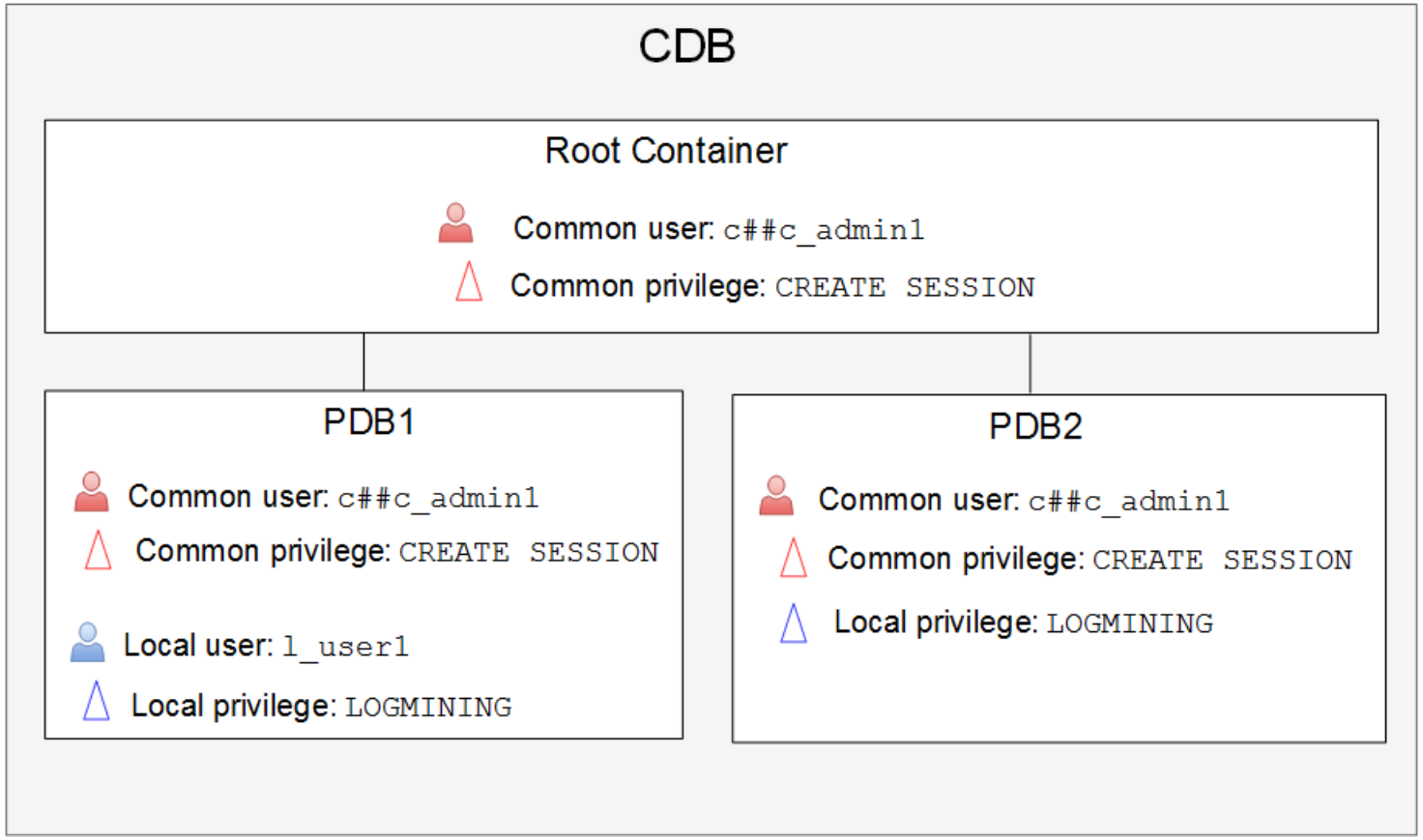
Privilege	Description
SYSDBA	Perform all administrative tasks in the database, including create and drop a database, open and mount a database, start up and shut down an Oracle database, create an SPFILE, put a database in or remove a database from ARCHIVELOG mode, perform incomplete recovery operations, patch, and migrate. This privilege enables you to connect as the SYS user.
SYSOPER	Perform similar administration tasks as the SYSDBA privilege, but without the ability to look at user data. For example, you can start up and shut down the database, create an SPFILE, and perform complete recovery operations (not incomplete recovery operations).
SYSASM	Start up, shut down, and administer an Automatic Storage Management instance.
SYSBACKUP	Perform backup and recovery operations by using RMAN or SQL*Plus.
SYSDG	Perform Data Guard operations by using the Data Guard Broker or the DGMGRL command-line interface.
SYSKM	Manage Transparent Data Encryption wallet operations.
SYSRAC	Perform day-to-day administration tasks on an Oracle Real Application Clusters (RAC) cluster.

Object Privileges

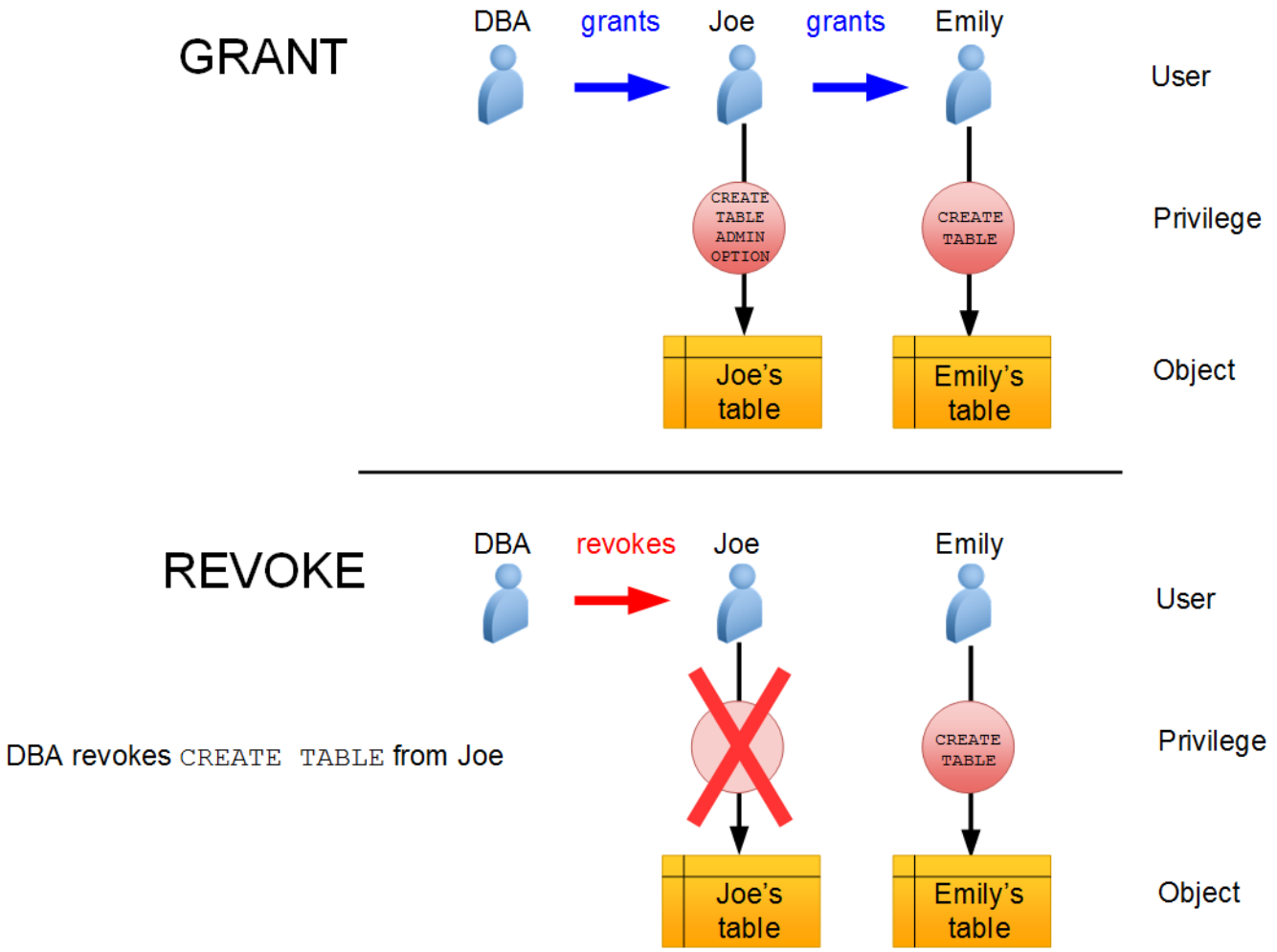
- Object privileges allow a user to perform a particular action on a specific object, such as a table, view, sequence, procedure, function, or package.
- Without specific permission, users can access only their own objects.
- Object privileges can be granted by the owner of an object, by the administrator, or by someone who has been explicitly given permission to grant privileges on the object.
- The SQL syntax for granting object privileges is:

```
GRANT <object_privilege> ON <object> TO <grantee clause>
[WITH GRANT OPTION]
```

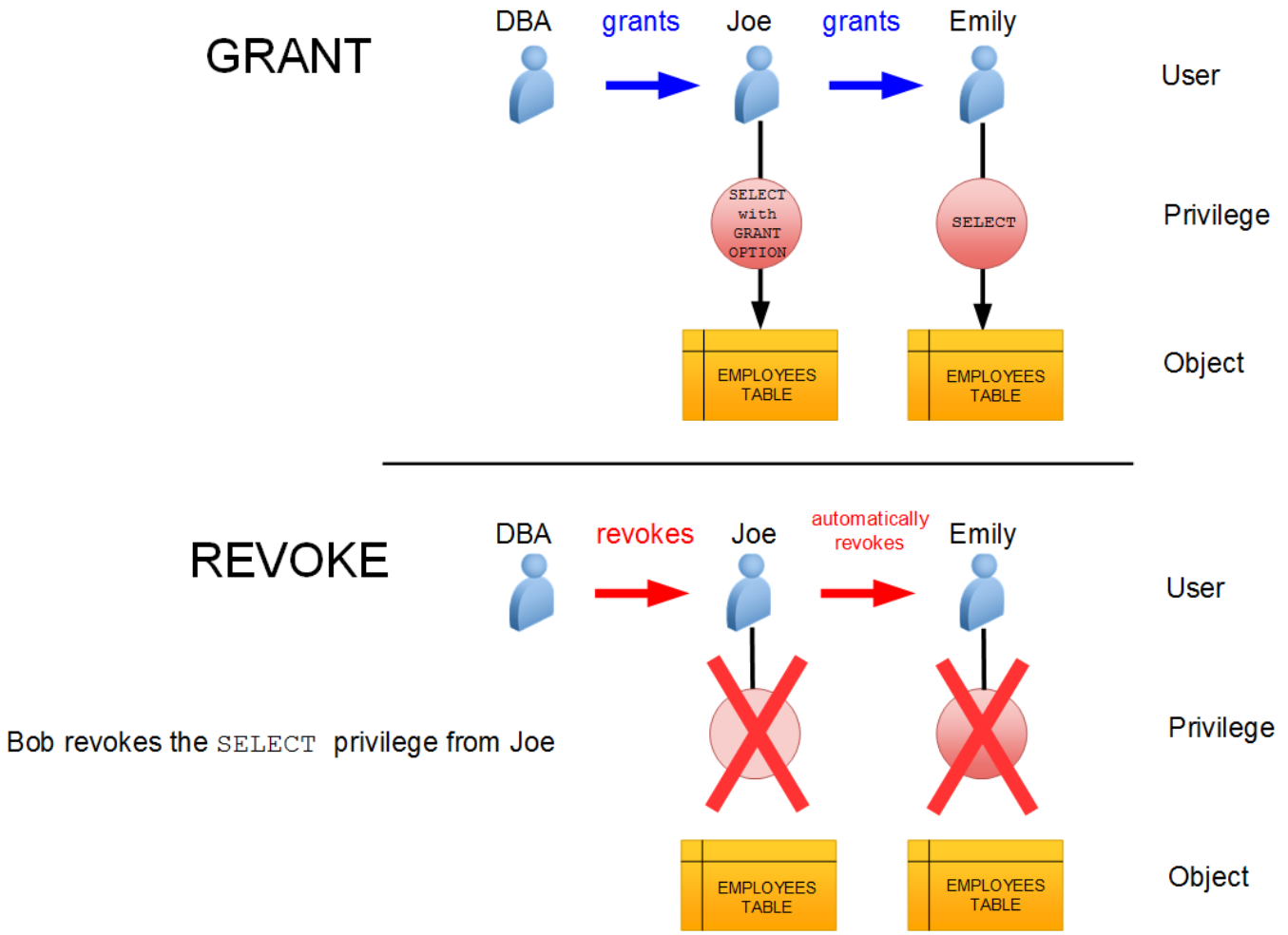
Granting Privileges in a Multitenant Environment



Granting and Revoking System Privileges



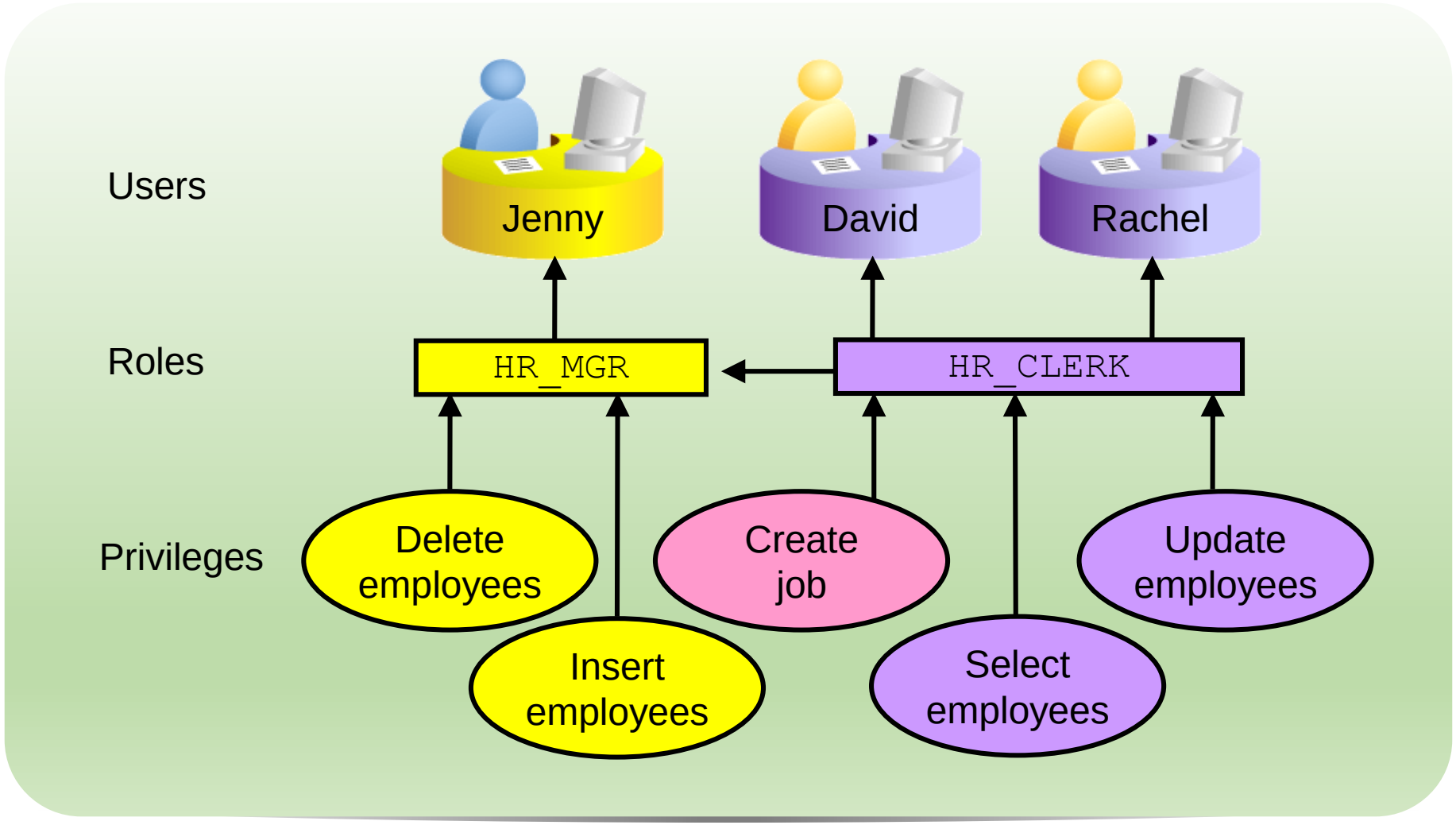
Granting and Revoking Object Privileges



Using Roles to Manage Privileges

- Roles:
  - Used to group together privileges and roles
  - Facilitate granting of multiple privileges or roles to users
- Benefits of roles:
  - Easier privilege management
  - Dynamic privilege management
  - Selective availability of privileges

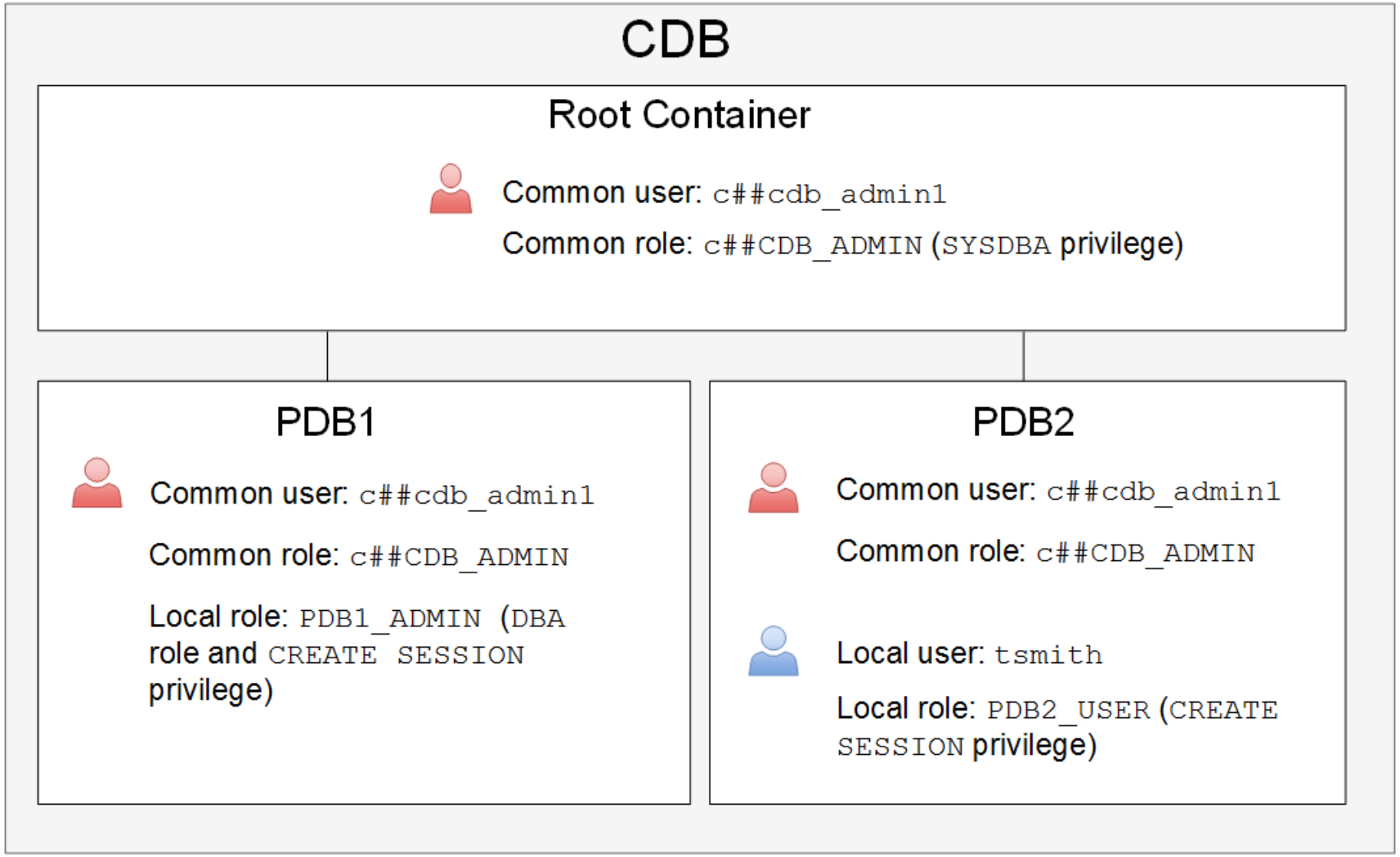
Assigning Privileges to Roles and Assigning Roles to Users



Oracle-Supplied Roles

Account	Description
DBA	Includes most system privileges and several other roles. Do not grant this role to nonadministrators. Users with this role can connect to the CDB or PDB only when it is open.
RESOURCE	CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, CREATE TYPE
SCHEDULER_ADMIN	CREATE ANY JOB, CREATE EXTERNAL JOB, CREATE JOB, EXECUTE ANY CLASS, EXECUTE ANY PROGRAM, MANAGE SCHEDULER
SELECT_CATALOG_ROLE	SELECT privileges on data dictionary objects

Creating and Granting Roles



Assigning Roles

- To assign (grant) a role to a user or another role by using SQL*Plus, use the `GRANT` command.
- There are two ways to grant a role in a multitenant architecture:
  - Commonly: Grant the role to the user (or role) in all containers.

```
SQL> CONNECT / AS SYSDBA
SQL> GRANT <common role> TO <common user or role> CONTAINER=ALL;
```

  - Locally: Grant the role to a user (or role) in one PDB only.

```
SQL> CONNECT SYS@PDB1 AS SYSDBA
SQL> GRANT <common or local role> TO <common or local user>;
```

Making Roles More Secure

- Roles are usually enabled by default, which means that if a role is granted to a user, then that user can exercise the privileges given to the role immediately.
- Default roles are assigned to the user at connect time.
- Use the following security measures to make roles more secure:
  - Make a role nondefault.
  - Use role authentication.
  - Create application roles.

Revoking Roles and Privileges

- You can use the `REVOKE` statement to:
  - Revoke system privileges from users and roles
  - Revoke roles from users, roles, and program units
  - Revoke object privileges for a particular object from users and roles

Profiles and Users

- Users are assigned only one profile at a time.
- Profiles:
  - Control resource consumption
  - Manage account status and password expiration
- `RESOURCE_LIMIT` must be set to `TRUE` before profiles can impose resource limitations.

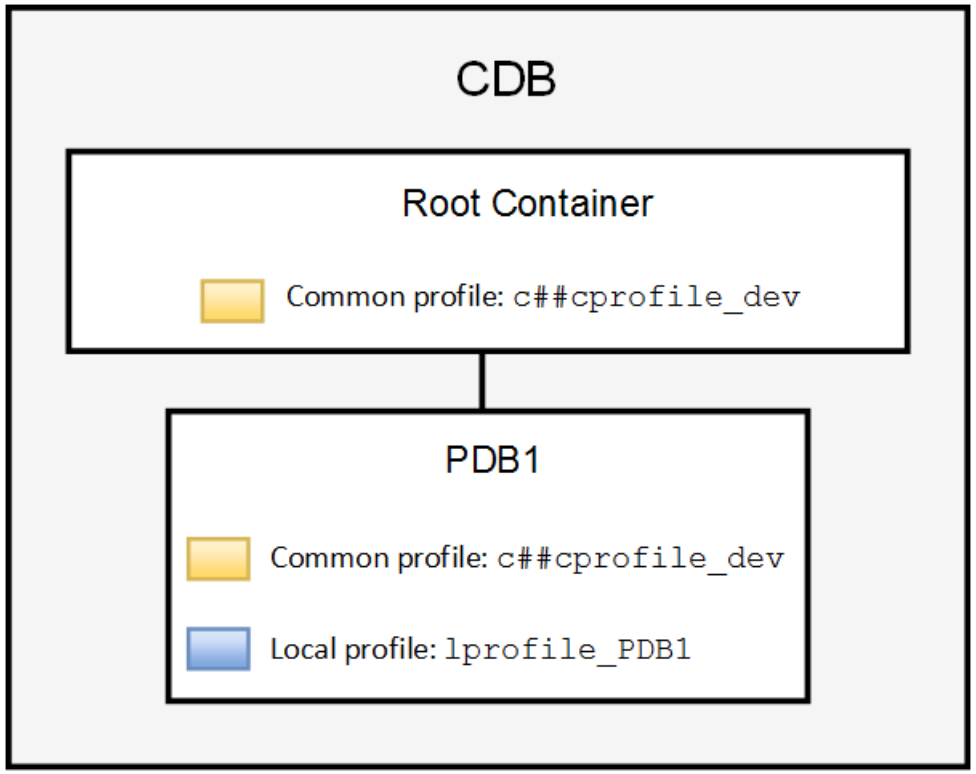
Creating Profiles in a Multitenant Architecture

- Common profile:

```
SQL> CREATE PROFILE c##cprofile_dev
  2  limit … CONTAINER=ALL;
```

- Local profile:

```
SQL> CREATE PROFILE lprofile_PDB1
  2  limit … ;
```



Profile Parameters: Resources

- In a profile, you can control:
  - CPU resources: May be limited to a per-session or per-call basis
  - Network and memory resources (Connect time, Idle time, Concurrent sessions, Private SGA)
- Disk I/O resources: Limit the amount of data a user can read at the per-session level or per-call level.
- Profiles cannot impose resource limitations on users unless the `RESOURCE_LIMIT` initialization parameter is set to `TRUE`. With `RESOURCE_LIMIT` at its default value of `FALSE`, profile resource limitations are ignored.
- Profiles also allow composite limits, which are based on weighted combinations of CPU/session, reads/session, connect time, and private SGA.

Profile Parameters: Locking and Passwords

- In a profile, specific parameters control account locking, password aging and expiration, and password history.
- Profile password settings are always enforced.
- Account locking enables automatic locking of accounts for a set duration when users fail to log in to the system in the specified number of attempts or when accounts sit inactive for a predefined number of days (users have not attempted to log in to their accounts).
- Password aging and expiration enables user passwords to have a lifetime, after which the passwords expire and must be changed.
- Password history checks the new password to ensure that the password is not reused for a specified amount of time or a specified number of password changes.
- Password complexity verification makes a complexity check on the password to verify that it meets certain rules.

Oracle-Supplied Password Verification Functions

- Complexity verification checks that each password is complex enough to provide reasonable protection against intruders who try to break into the system by guessing passwords.
- You can create your own password verification functions.
- Oracle Database provides the following functions that you can create by executing the `utlpwdmg.sql` script:
  - `ORA12c_VERIFY_FUNCTION`
  - `ORA12c_STRONG_VERIFY_FUNCTION`
  - `VERIFY_FUNCTION_11g`
- These functions must be owned by the `SYS` user.
- Password complexity checking is not enforced for the `SYS` user.

Assigning Profiles

- There are two ways to assign a profile:
  - Commonly: The profile assignment is replicated in all current and future containers.

```
SQL> CONNECT / AS SYSDBA
SQL> ALTER USER <common user> PROFILE <common
profile>                              CONTAINER=ALL;
```

  - Locally: The profile assignment occurs in one PDB (stand-alone or application container) only.

```
SQL> CONNECT SYS@PDB1 AS SYSDBA
SQL> ALTER USER <common or local user> PROFILE
<common or local          profile>;
```

Assigning Quotas

- A quota is a space allowance in a given tablespace.
- By default, a user has no quota on any of the tablespaces.
- Database accounts that need quota are those that own database objects (for example, accounts for applications).
- Only those activities that use space in a tablespace count against quota.
  - Oracle server checks quota when you create or extend a segment.
  - Activities that don't use space don't impact quota (example: `CREATE VIEW`).
  - You can be granted permission to use objects without needing any quota.
- Quota is not needed for assigned temporary tablespaces or undo tablespaces.
- A user's quota is replenished when he drops objects (with the `PURGE` clause) or purges his objects in the recycle bin.

Assigning Quotas

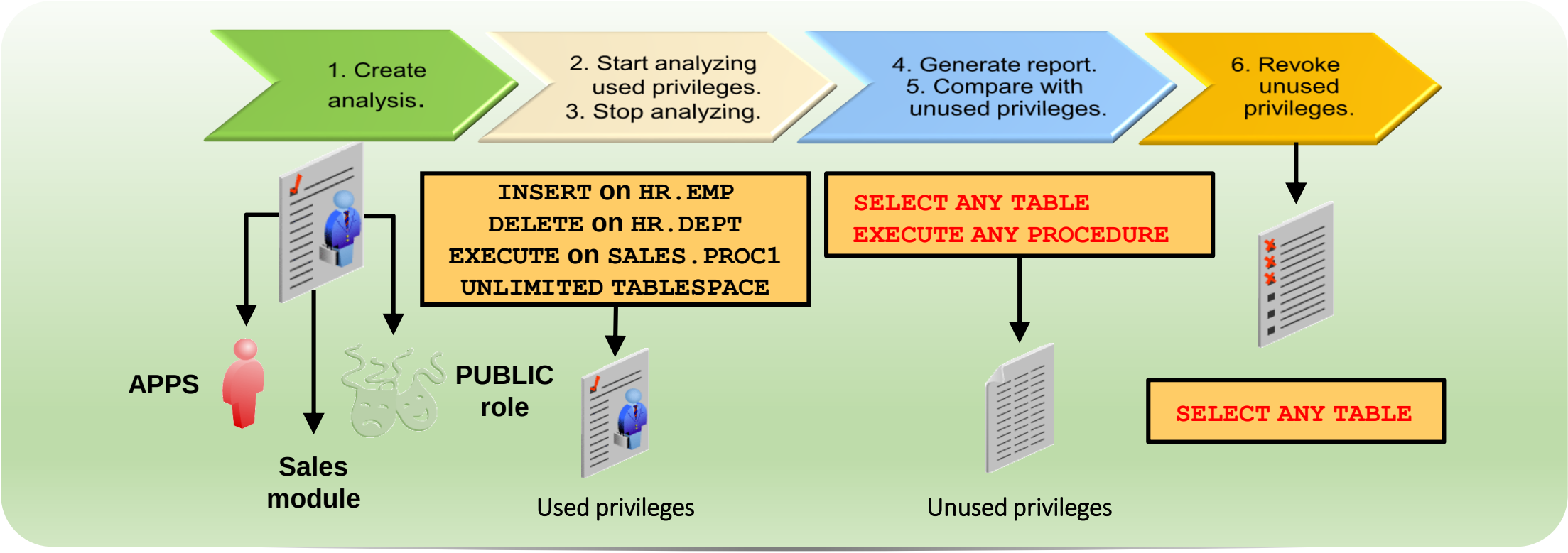
- You have three options for providing quota for a user on a tablespace:
  - `UNLIMITED`
  - Value
  - `UNLIMITED TABLESPACE` system privilege

Applying the Principle of Least Privilege

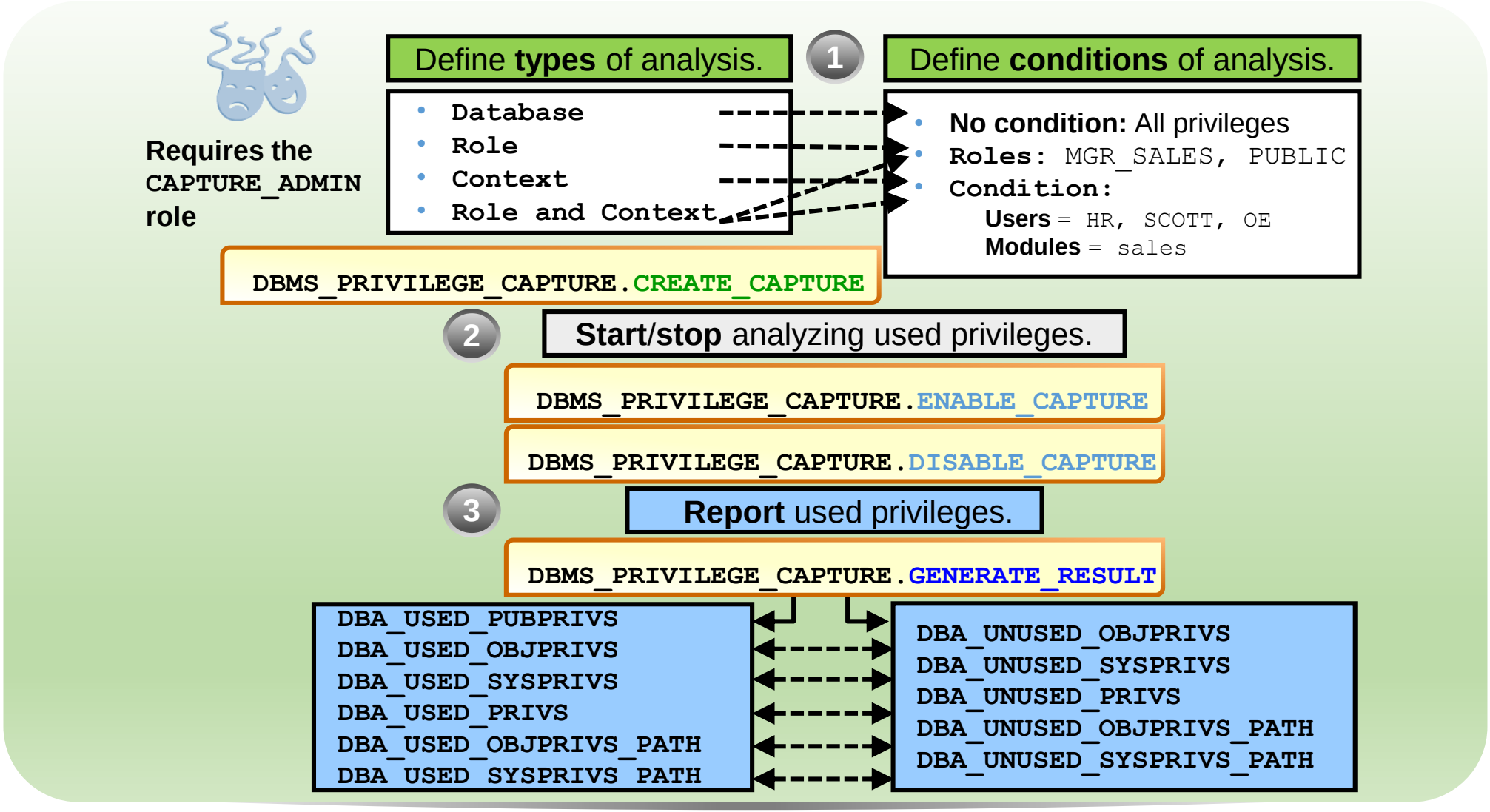
- The principle of least privilege means that a user must be given only those privileges that are required to efficiently complete a task.
- This reduces the chances of users modifying or viewing data (either accidentally or maliciously) that they do not have the privilege to modify or view.
- Ways to apply the principle of least privilege:
  - Protect the data dictionary
  - Revoke unnecessary privileges from `PUBLIC`
  - Use access control lists (ACLs) to control network access
  - Restrict access to OS directories
  - Limit users with administrative privileges
  - Restrict remote database authentication
  - Enable unified auditing

Privilege Analysis

- Analyze used privileges to revoke unnecessary privileges.
- Use the `DBMS_PRIVILEGE_CAPTURE` package.



Privilege Analysis Flow



Summary

- In this lesson, you should have learned how to:
  - Create database users
  - Grant privileges to database users
  - Create and grant roles to users or other roles
  - Revoke privileges and roles from users and other roles
  - Create and assign profiles to users
  - Explain the various authentication options for users
  - Assign quota to users
  - Apply the principle of least privilege

Practice 9: Overview

- 9-1: Creating Common and Local Users
- 9-2: Creating a Local User for an Application
- 9-3: Granting a Local Role (DBA) to PDBADMIN
- 9-4: Using EM Express to Create a Local Profile
- 9-5: Using EM Express to Create Local Roles
- 9-6: Using EM Express to Create Local Users
- 9-7: Configuring a Default Role for a User
- 9-8: Exploring OS and Password File Authentication