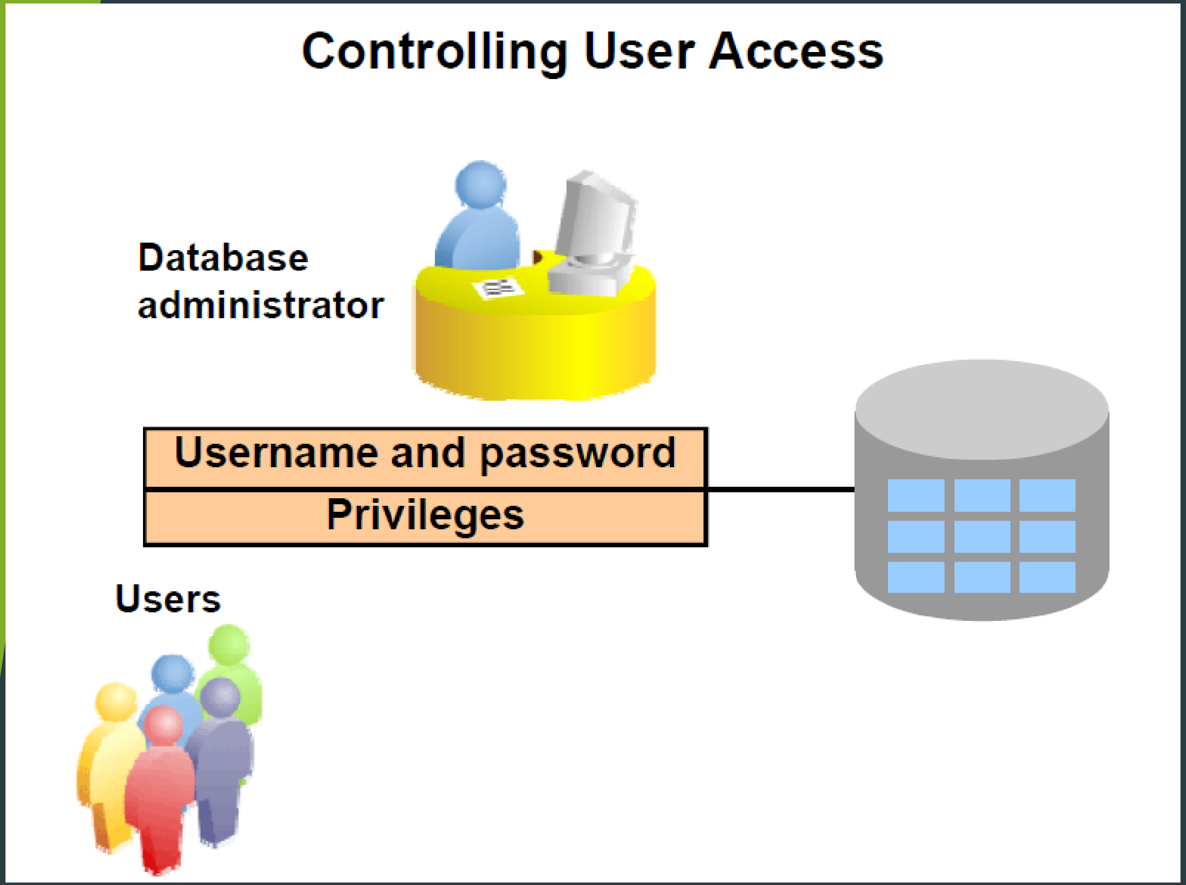
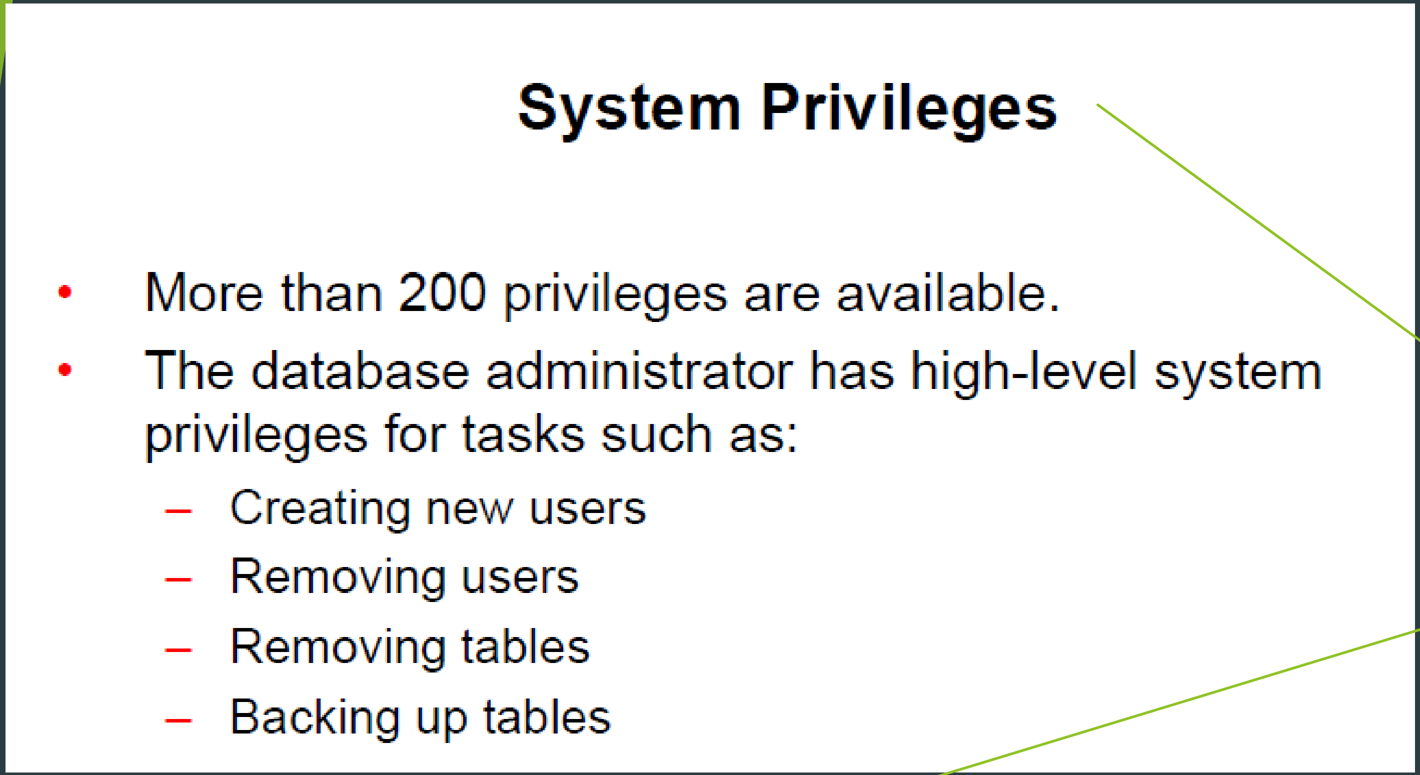
Administering User Security



- Database security:
  - System security
  - Data security
- System privileges: Performing a particular action within the database
- Object privileges: Manipulating the content of the database objects
- Schemas: Collection of objects such as tables, views, and sequences

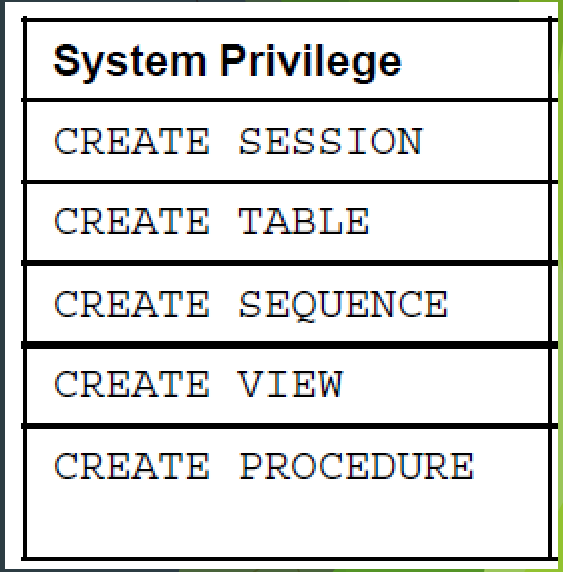
users and schemas are database users , but when the user has objects then we call it schema

Administering User Security

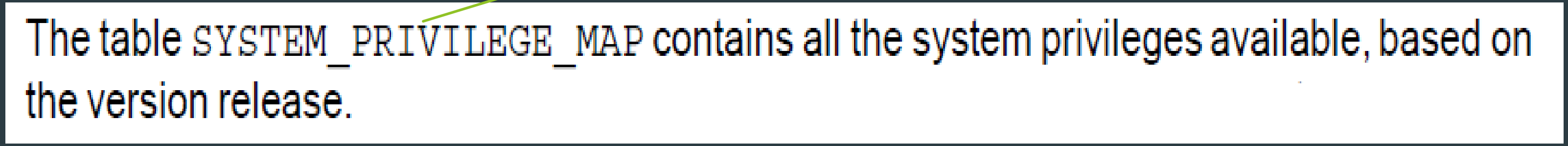


System Privileges

- More than 200 privileges are available.
- The database administrator has high-level system privileges for tasks such as:
  - Creating new users
  - Removing users
  - Removing tables
  - Backing up tables

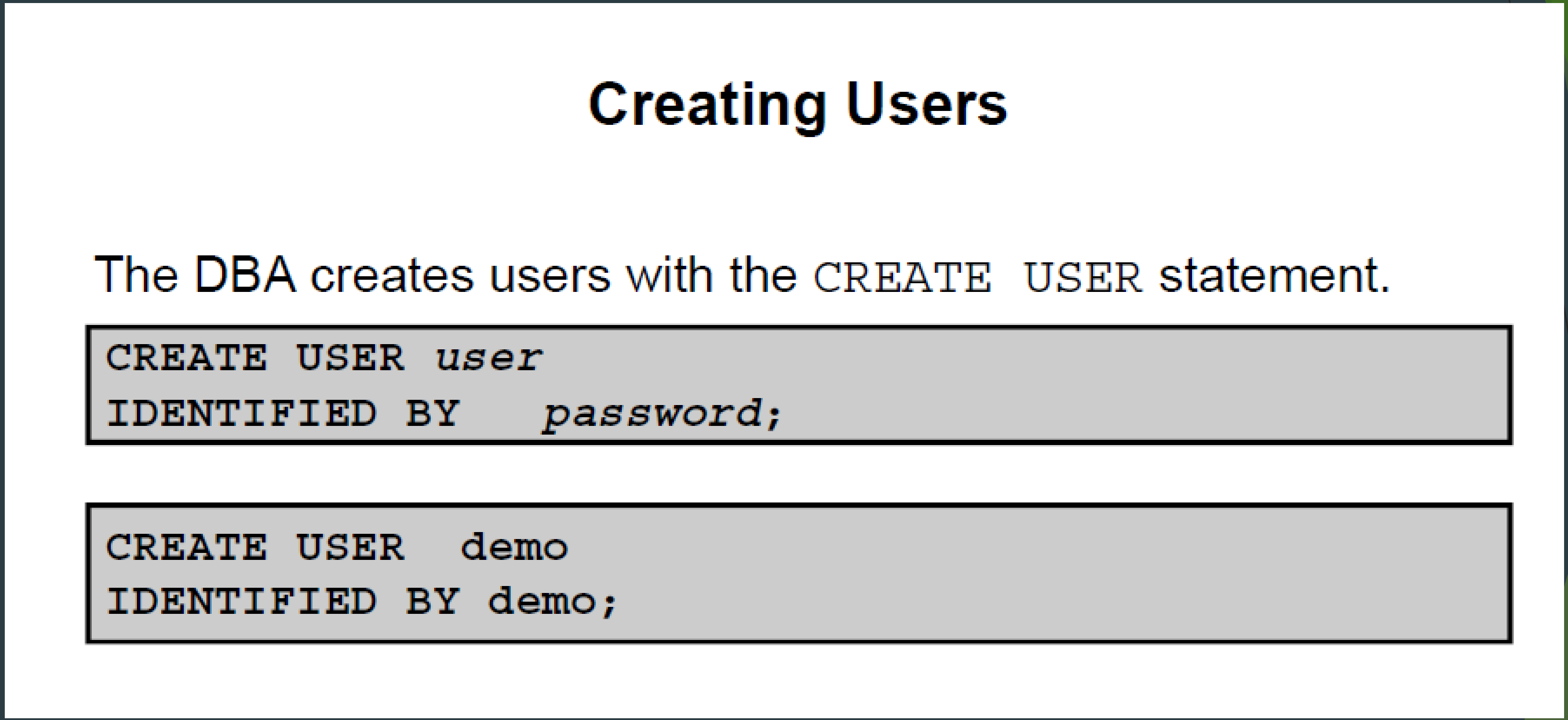


System Privilege
CREATE SESSION
CREATE TABLE
CREATE SEQUENCE
CREATE VIEW
CREATE PROCEDURE



The table SYSTEM_PRIVILEGE_MAP contains all the system privileges available, based on the version release.

Administering User Security



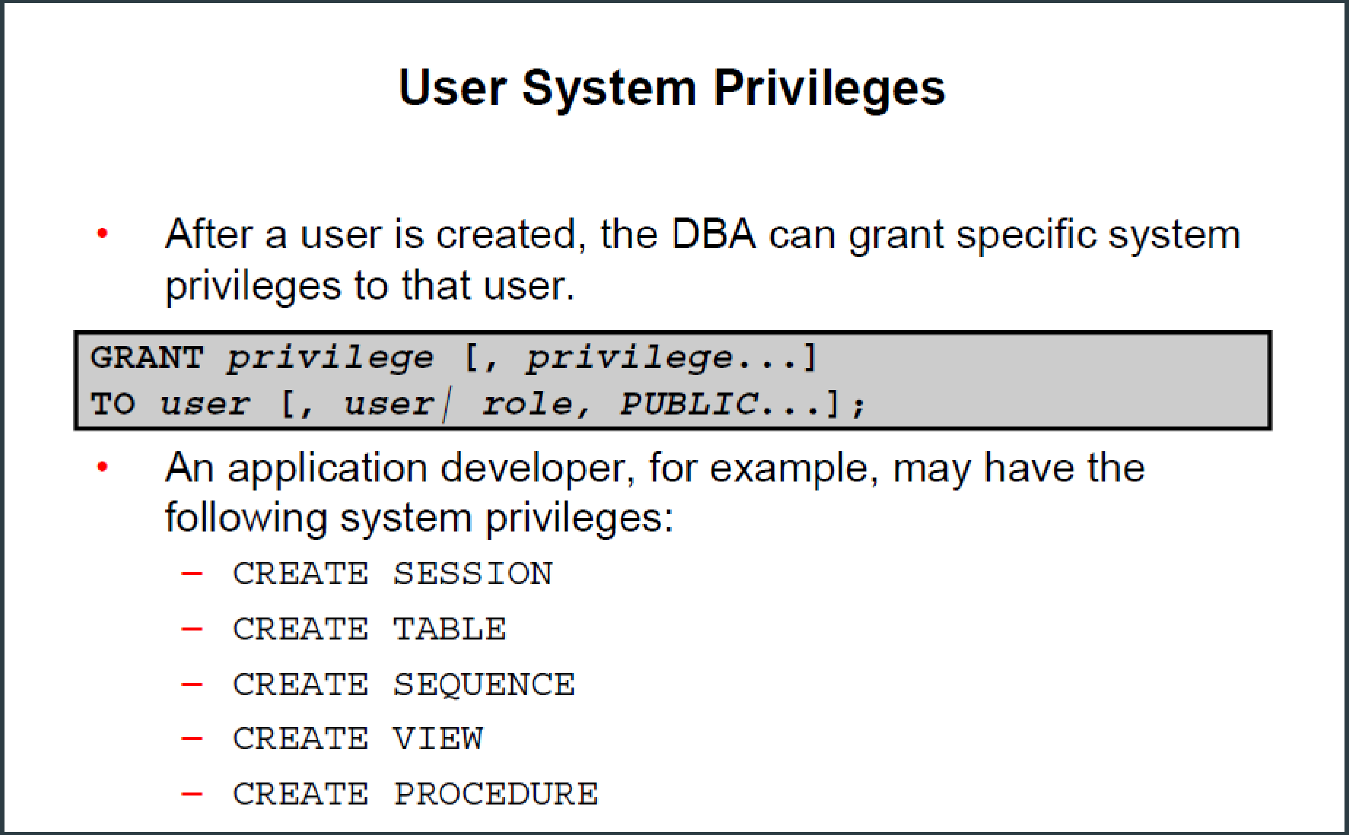
Creating Users

The DBA creates users with the `CREATE USER` statement.

```
CREATE USER user
IDENTIFIED BY   password;
```

```
CREATE USER  demo
IDENTIFIED BY demo;
```

Administering User Security

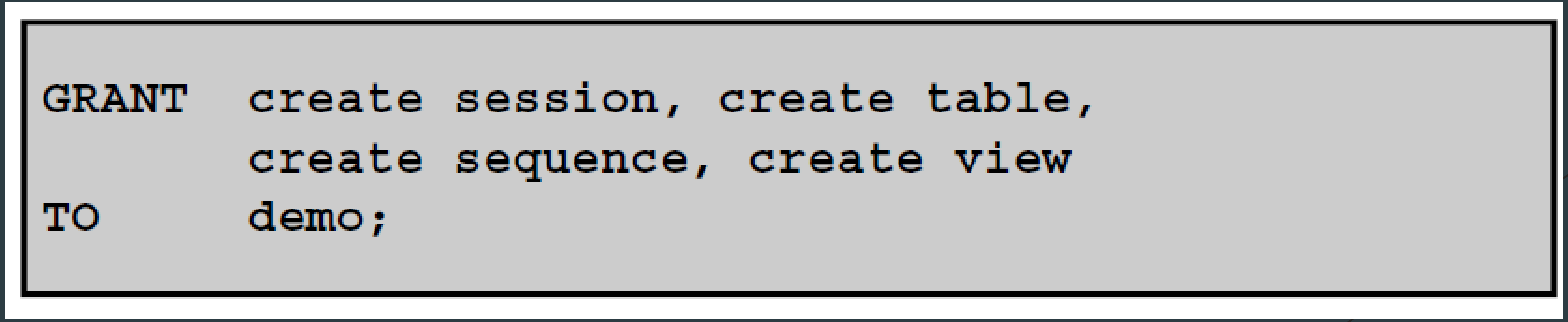


User System Privileges

- After a user is created, the DBA can grant specific system privileges to that user.

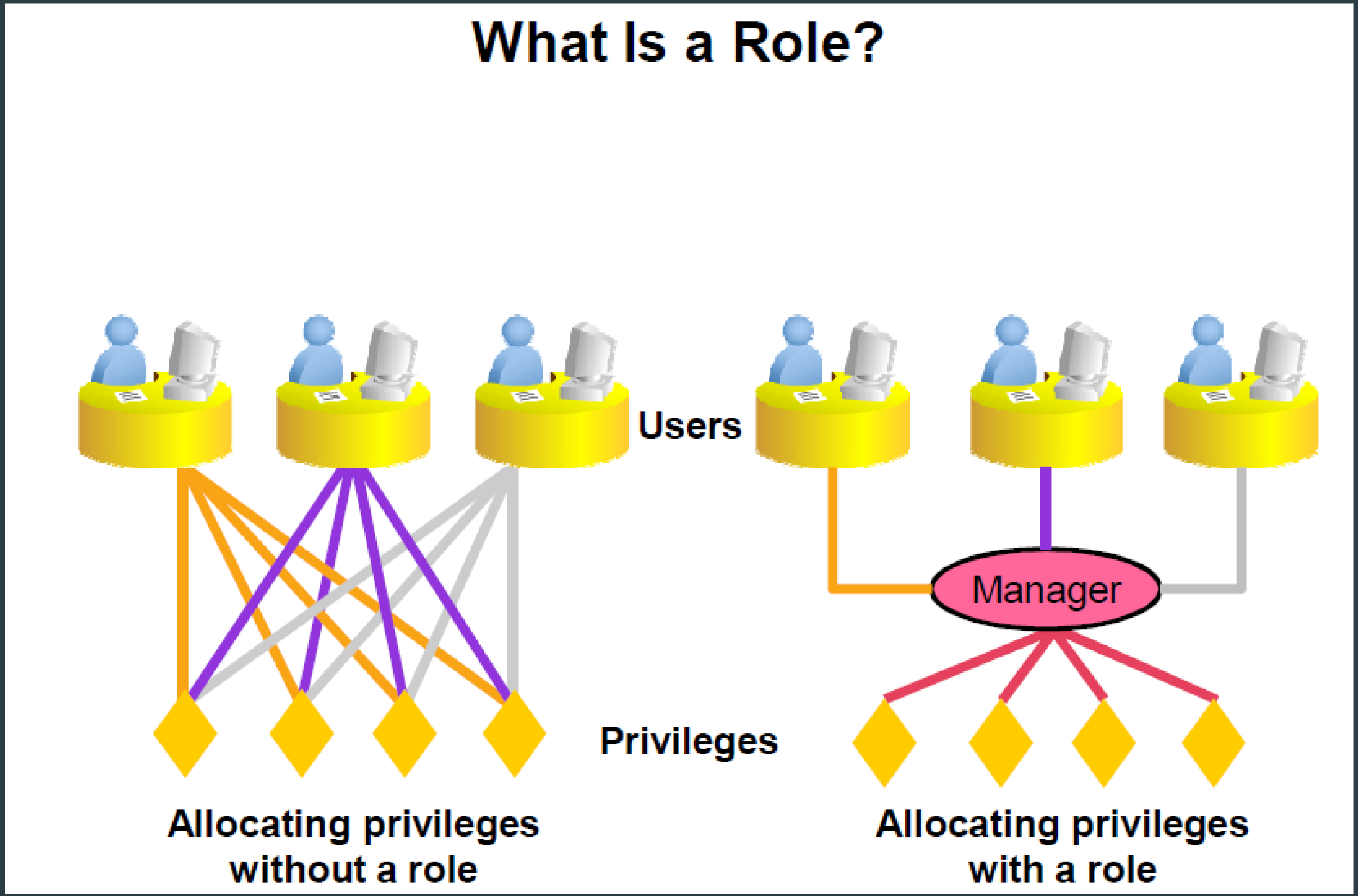
```
GRANT privilege [, privilege...]
TO user [, user| role, PUBLIC...];
```

- An application developer, for example, may have the following system privileges:
  - `CREATE SESSION`
  - `CREATE TABLE`
  - `CREATE SEQUENCE`
  - `CREATE VIEW`
  - `CREATE PROCEDURE`



```
GRANT  create session, create table,
       create sequence, create view
TO     demo;
```

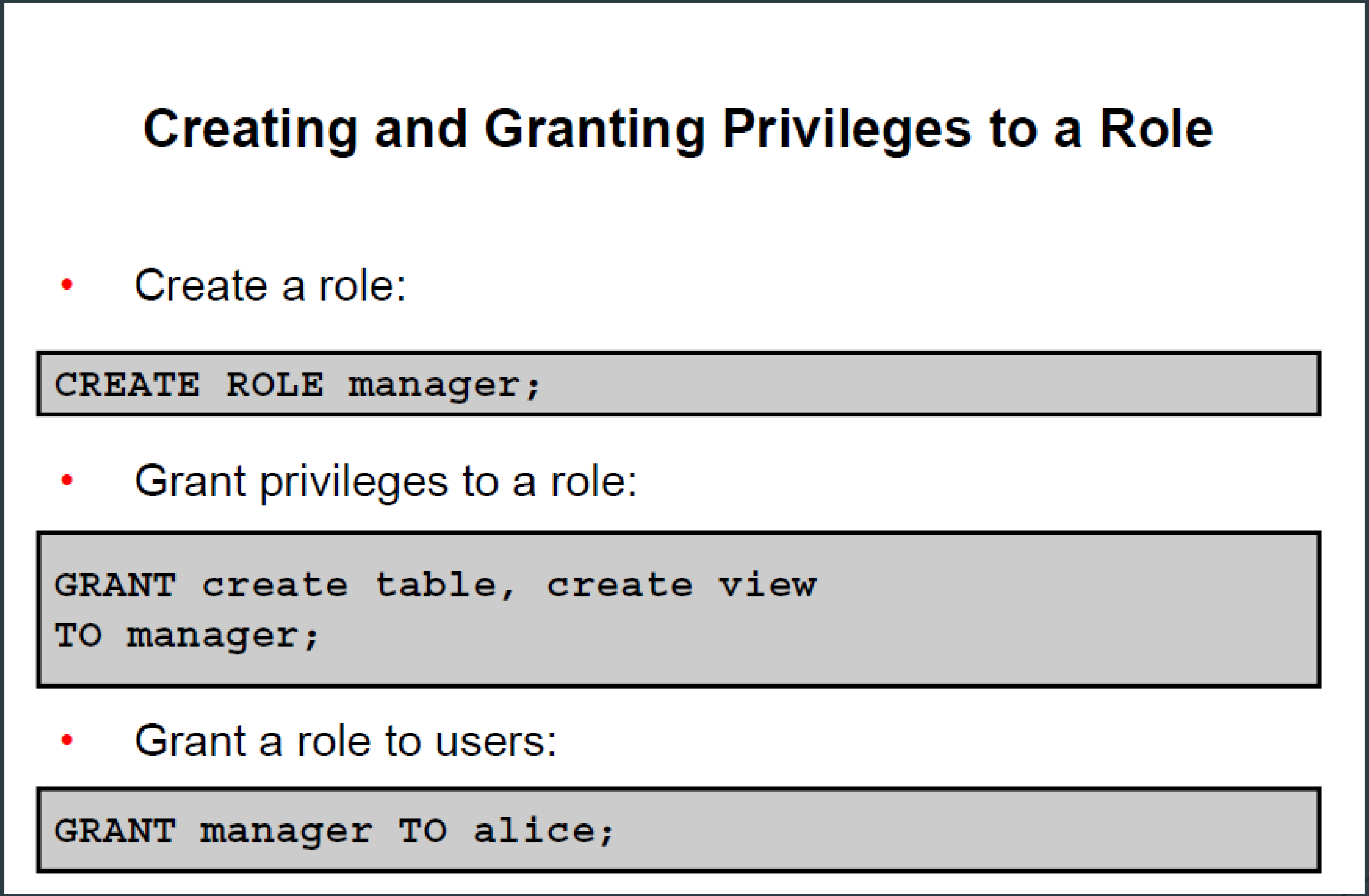
Administering User Security


What Is a Role?
Users
Manager
Privileges
Allocating privileges without a role
Allocating privileges with a role

A role is a named group of related privileges that can be granted to the user. This method makes it easier to revoke and maintain privileges.

A user can have access to several roles, and several users can be assigned the same role. Roles are typically created for a database application.

Administering User Security



Creating and Granting Privileges to a Role

- Create a role:

```
CREATE ROLE manager;
```

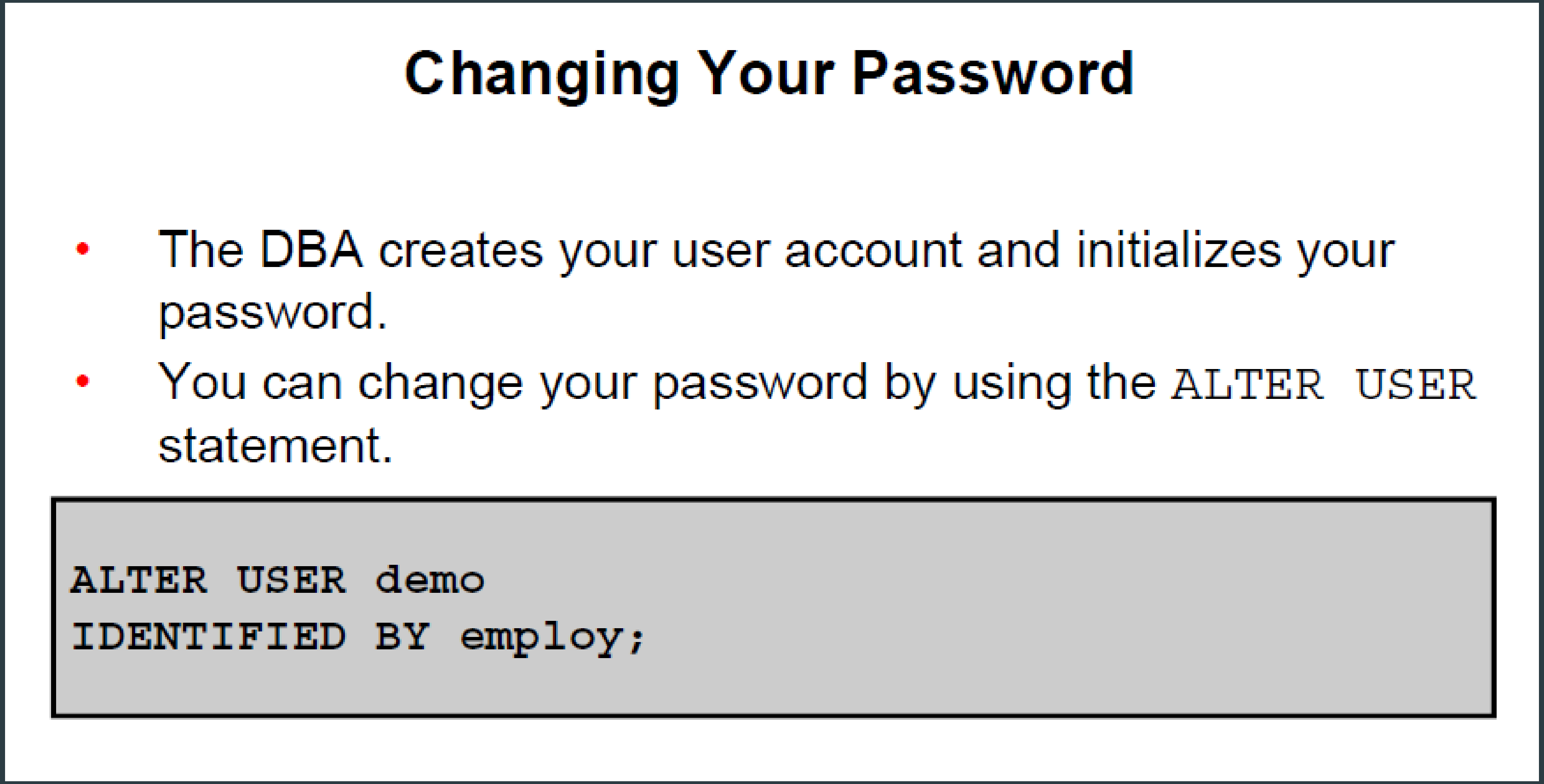
- Grant privileges to a role:

```
GRANT create table, create view
TO manager;
```

- Grant a role to users:

```
GRANT manager TO alice;
```

Administering User Security



Changing Your Password

- The DBA creates your user account and initializes your password.
- You can change your password by using the `ALTER USER` statement.

```
ALTER USER demo
IDENTIFIED BY employ;
```

Administering User Security

Object Privileges

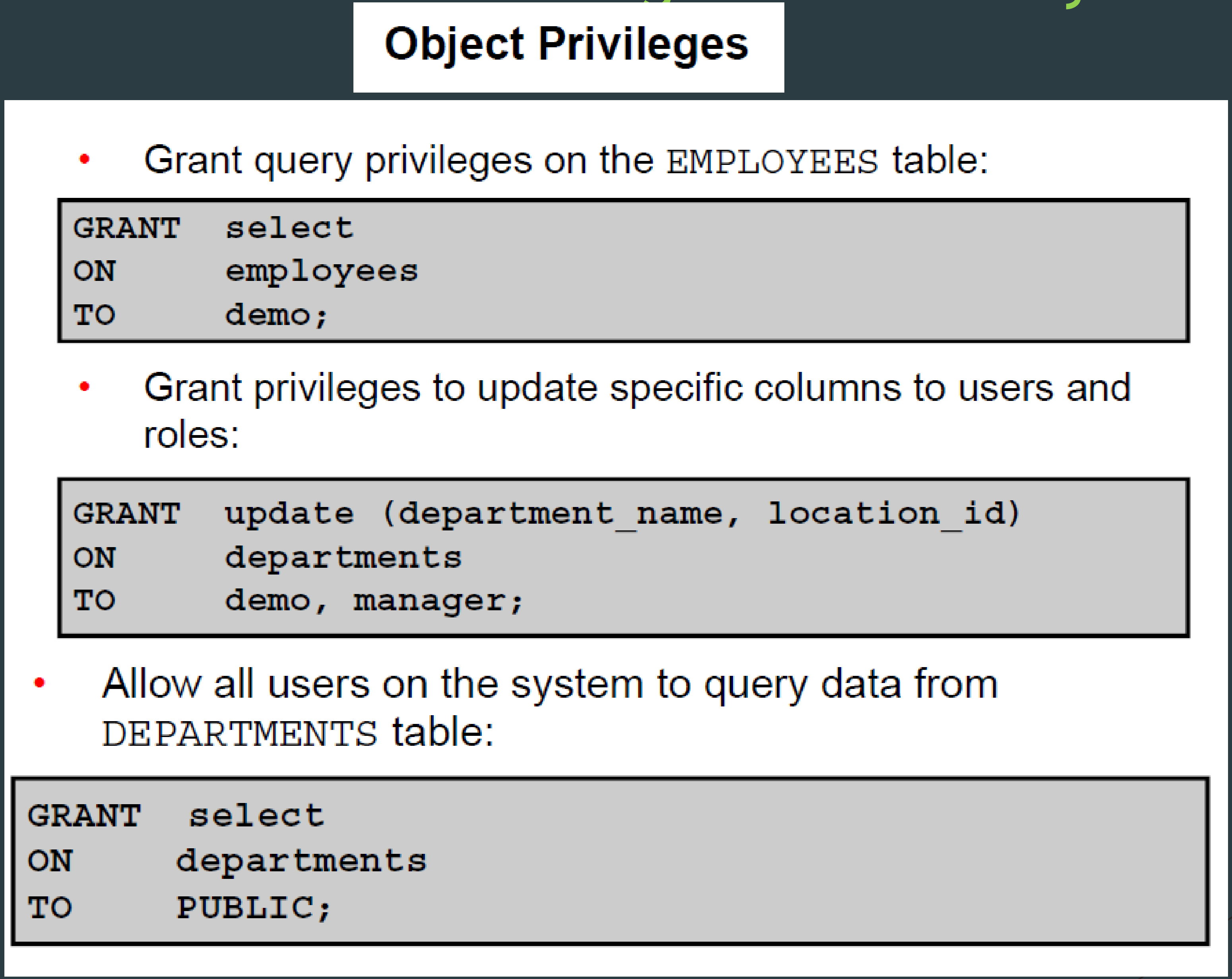
An *object privilege* is a privilege or right to perform a particular action on a specific table, view, sequence, or procedure.

- Object privileges vary from object to object.
- An owner has all the privileges on the object.
- An owner can give specific privileges on that owner's object.

```
GRANT       object_priv [(columns)]
ON          object
TO          {user|role|PUBLIC}
[WITH GRANT OPTION];
```

Enables the grantee to grant the object privileges to other users and roles

Administering User Security



Object Privileges

- Grant query privileges on the `EMPLOYEES` table:

```
GRANT   select
ON      employees
TO      demo;
```

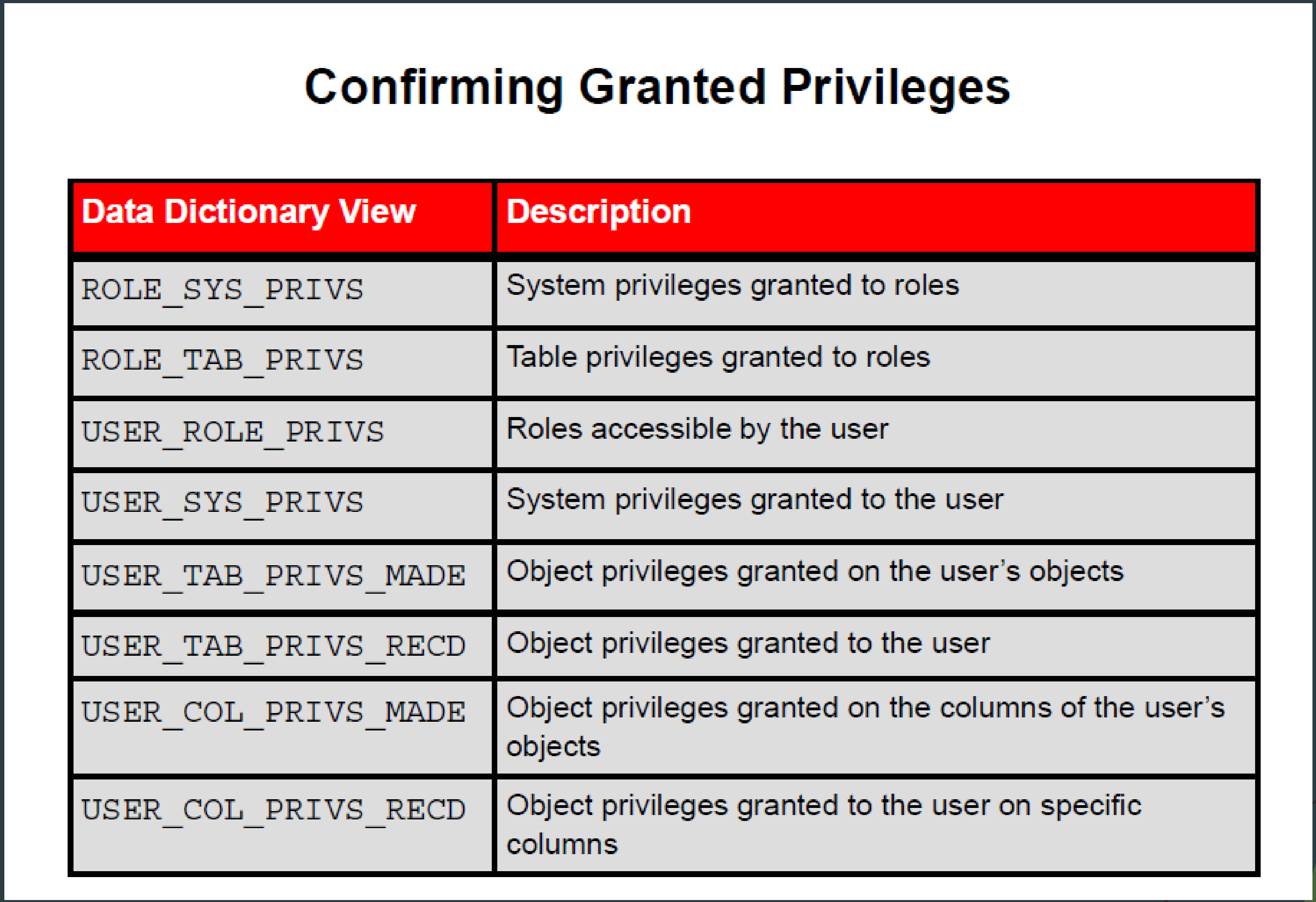
- Grant privileges to update specific columns to users and roles:

```
GRANT   update (department_name, location_id)
ON      departments
TO      demo, manager;
```

- Allow all users on the system to query data from `DEPARTMENTS` table:

```
GRANT   select
ON      departments
TO      PUBLIC;
```

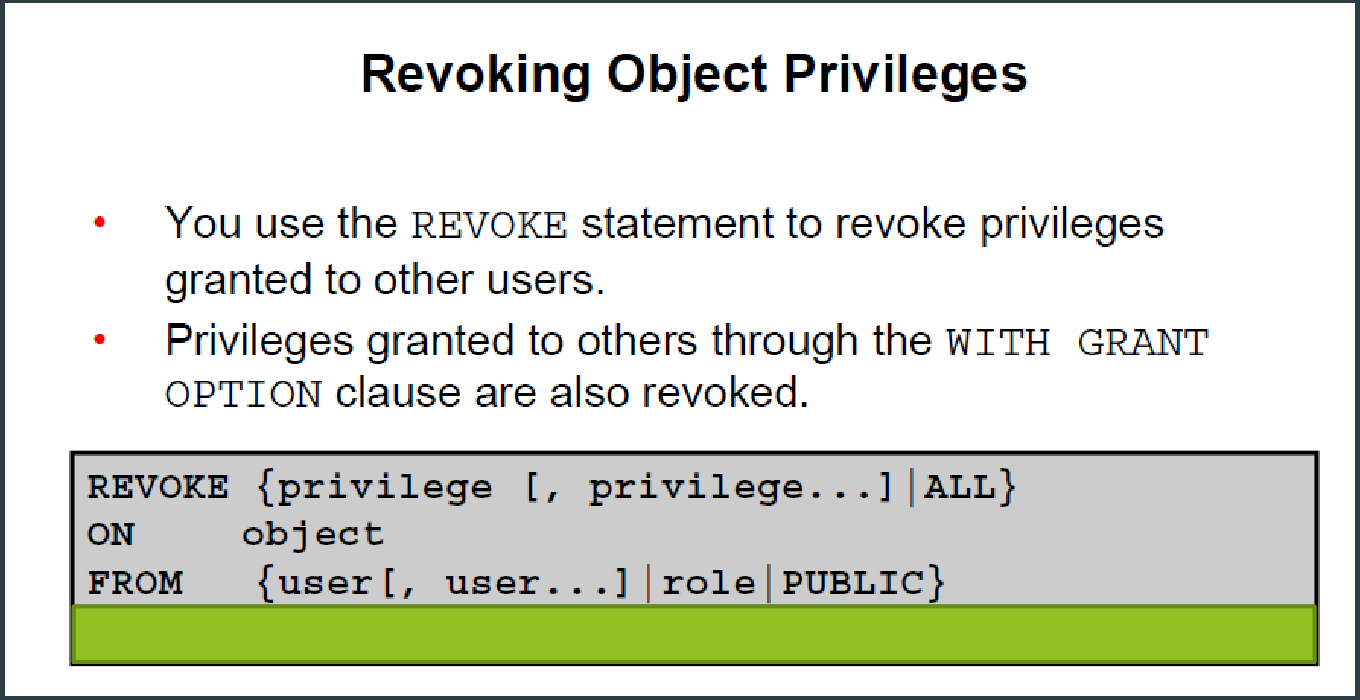
Administering User Security



Confirming Granted Privileges

Data Dictionary View	Description
`ROLE_SYS_PRIVS`	System privileges granted to roles
`ROLE_TAB_PRIVS`	Table privileges granted to roles
`USER_ROLE_PRIVS`	Roles accessible by the user
`USER_SYS_PRIVS`	System privileges granted to the user
`USER_TAB_PRIVS_MADE`	Object privileges granted on the user's objects
`USER_TAB_PRIVS_RECD`	Object privileges granted to the user
`USER_COL_PRIVS_MADE`	Object privileges granted on the columns of the user's objects
`USER_COL_PRIVS_RECD`	Object privileges granted to the user on specific columns

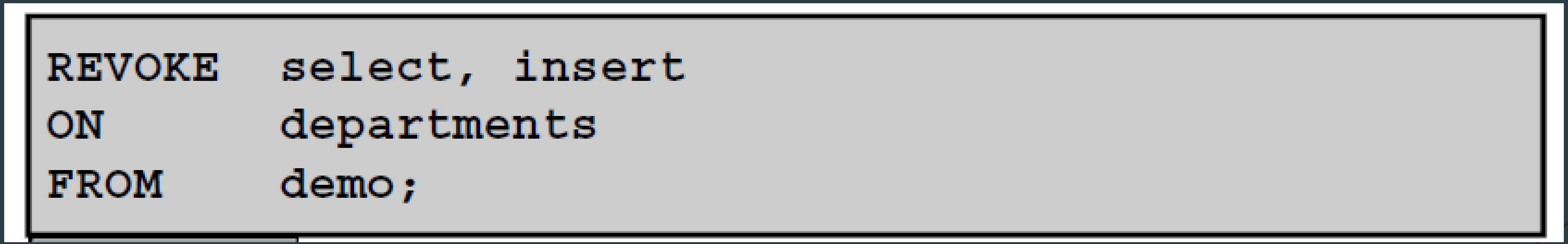
Administering User Security



Revoking Object Privileges

- You use the `REVOKE` statement to revoke privileges granted to other users.
- Privileges granted to others through the `WITH GRANT OPTION` clause are also revoked.

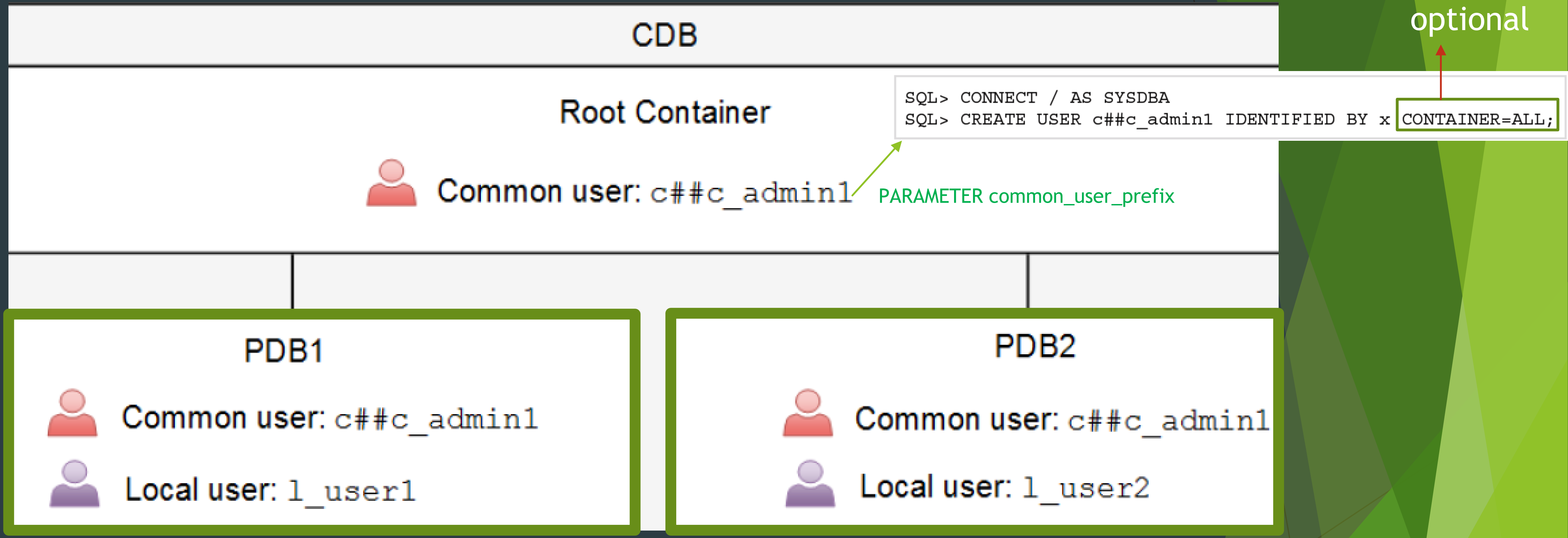
```
REVOKE {privilege [, privilege...]|ALL}
ON     object
FROM   {user[, user...]|role|PUBLIC}
```



```
REVOKE   select, insert
ON       departments
FROM     demo;
```

Administering User Security

Common users VS local users

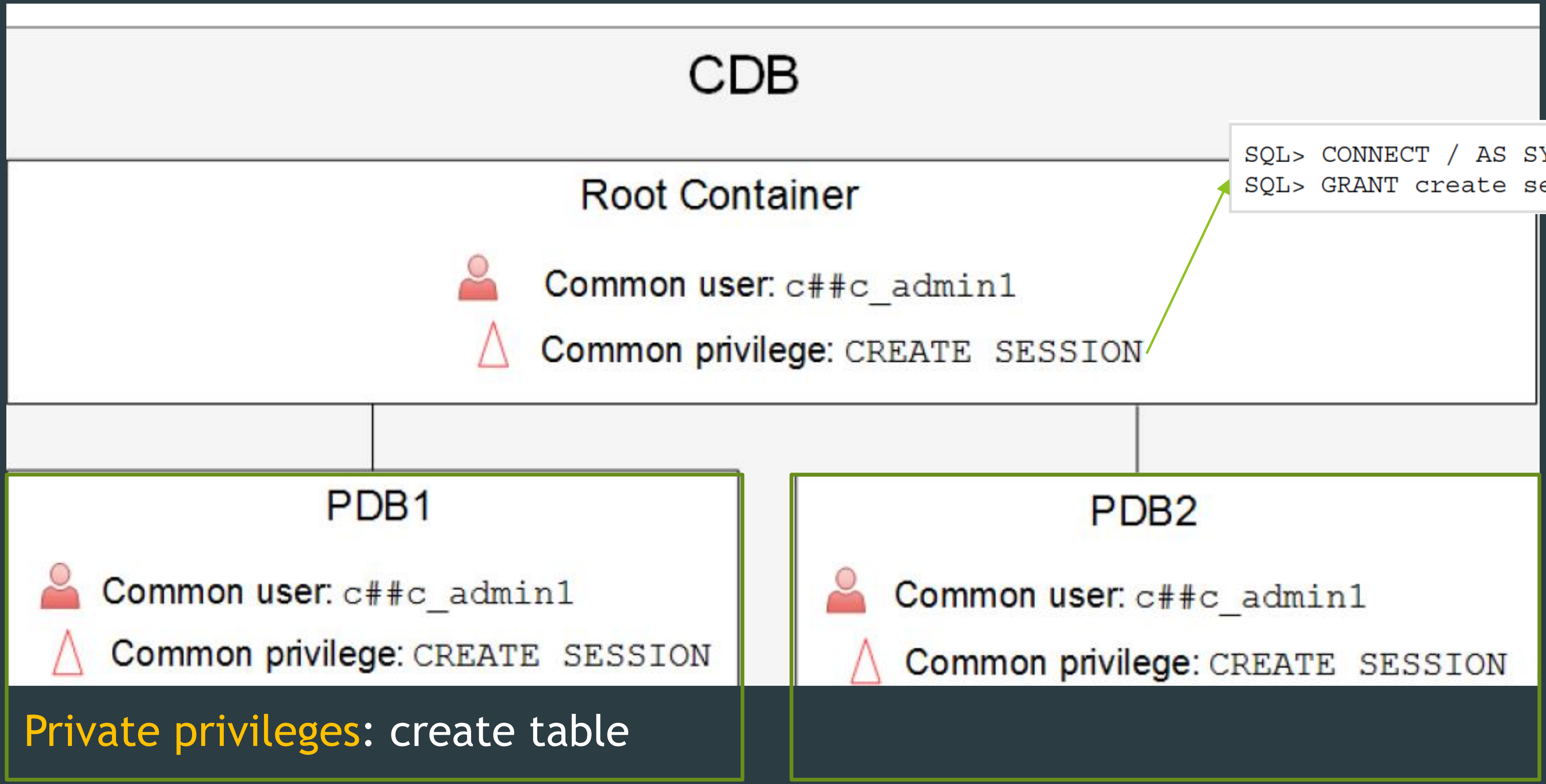


```
SQL> CONNECT / AS SYSDBA
SQL> CREATE USER c##c_admin1 IDENTIFIED BY x CONTAINER=ALL;
```

Refer to section : Helpful queries to explore Oracle DB architecture
lessons: Common users VS local users part 1
Common users VS local users part 2

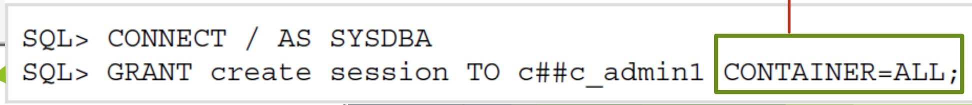
Administering User Security

Common privileges VS local privileges



Private privileges: create table

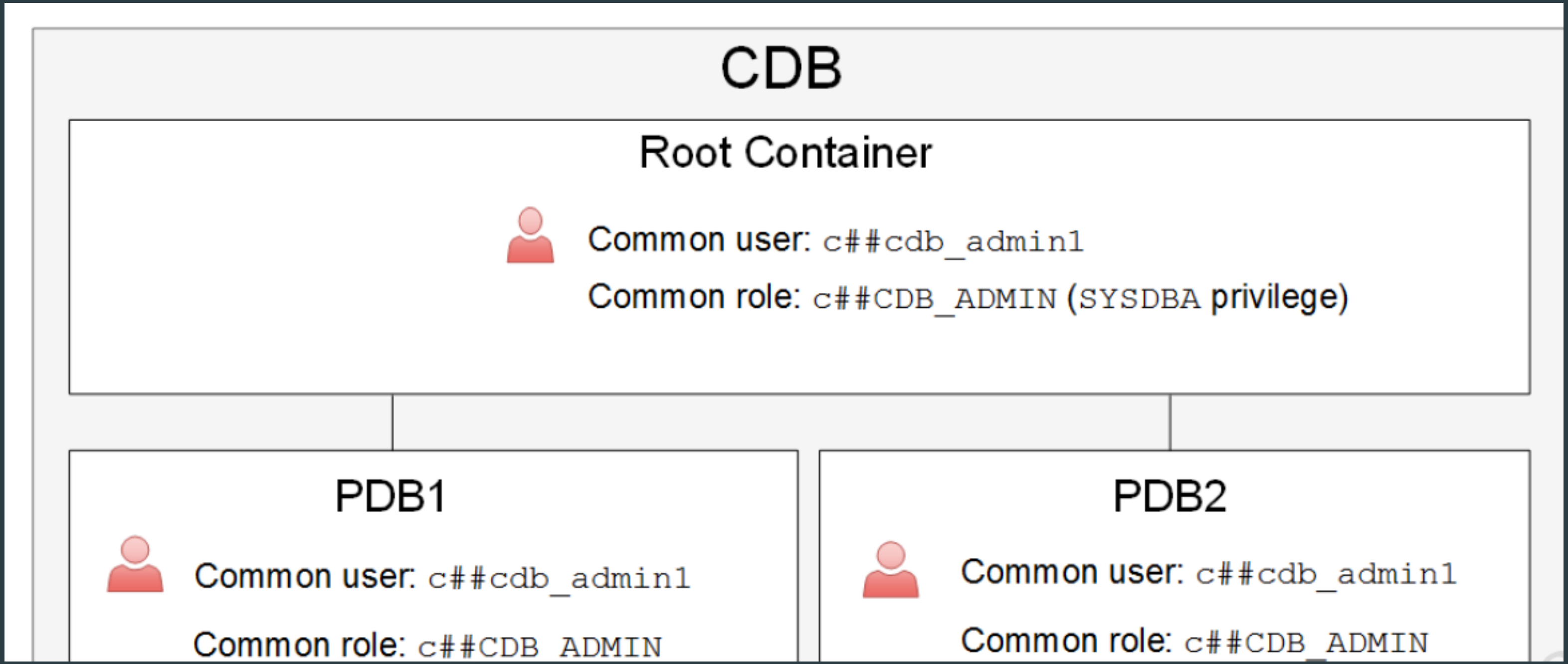
It must be in order to be common



```
SQL> CONNECT / AS SYSDBA
SQL> GRANT create session TO c##c_admin1 CONTAINER=ALL;
```

Administering User Security

Common roles


CDB
Root Container
Common user: c##cdb_admin1
Common role: c##CDB_ADMIN (SYSDBA privilege)
PDB1
Common user: c##cdb_admin1
Common role: c##CDB_ADMIN
PDB2
Common user: c##cdb_admin1
Common role: c##CDB_ADMIN

Two ways to grant a role:

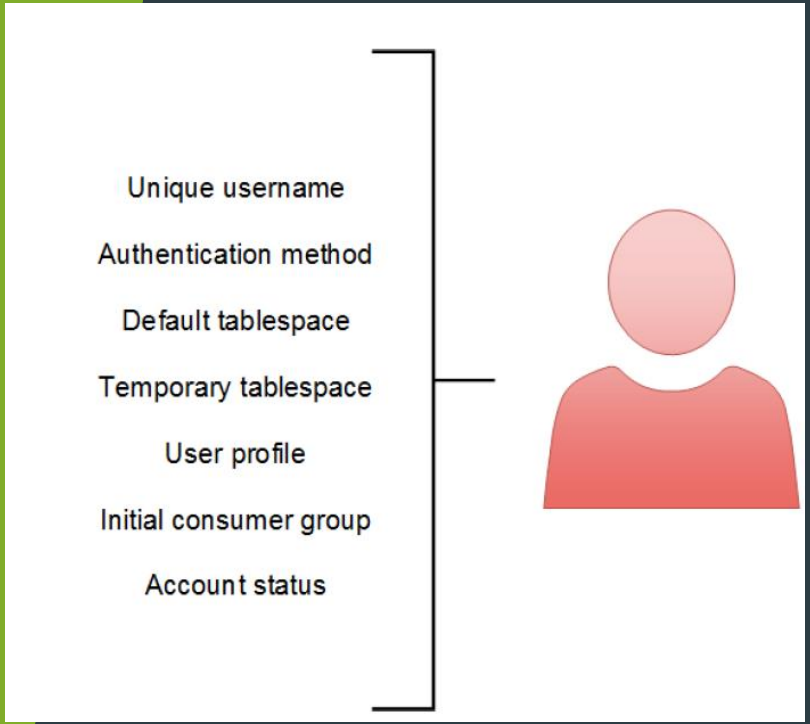
- Commonly: Grant the role to the user (or role) in all containers.

```
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>;
```

Administering User Security



More About Users accounts

- **Unique username:** Usernames cannot exceed 30 bytes, cannot contain special characters, and must start with a letter.

- **Authentication method:** The most common authentication method is a password.

- **Default tablespace:** This is a place where a user creates objects if the user does not specify some other tablespace.

- **Temporary tablespace:** This is a place where temporary objects, such as sorts and temporary tables, are created on behalf of the user by the instance. No quota is applied to temporary tablespaces. If an administrator does not define a temporary tablespace for a user, the system-defined temporary tablespace is used when the user creates objects.

- **User profile:** This is a set of resource and password restrictions assigned to the user.
- **Initial consumer group:** This is used by the Resource Manager.
- **Account status:** Users can access only “open” accounts. The account status may be “locked” and/or “expired.”

Administering User Security

Oracle- supplied administrator accounts

- ❑ SYS

This account can perform all administrative functions.
All base (underlying) tables and views for the database data dictionary are stored in the SYS schema.
These base tables and views are critical for the operation of Oracle Database.
To maintain the integrity of the data dictionary, tables in the SYS schema are manipulated
only by the database. They should never be modified by any user or database administrator.
You must not create any tables in the SYS schema.
The SYS user is granted the SYSDBA privilege,
which enables a user to perform high-level administrative tasks such as backup and recovery.

- ❑ SYSTEM

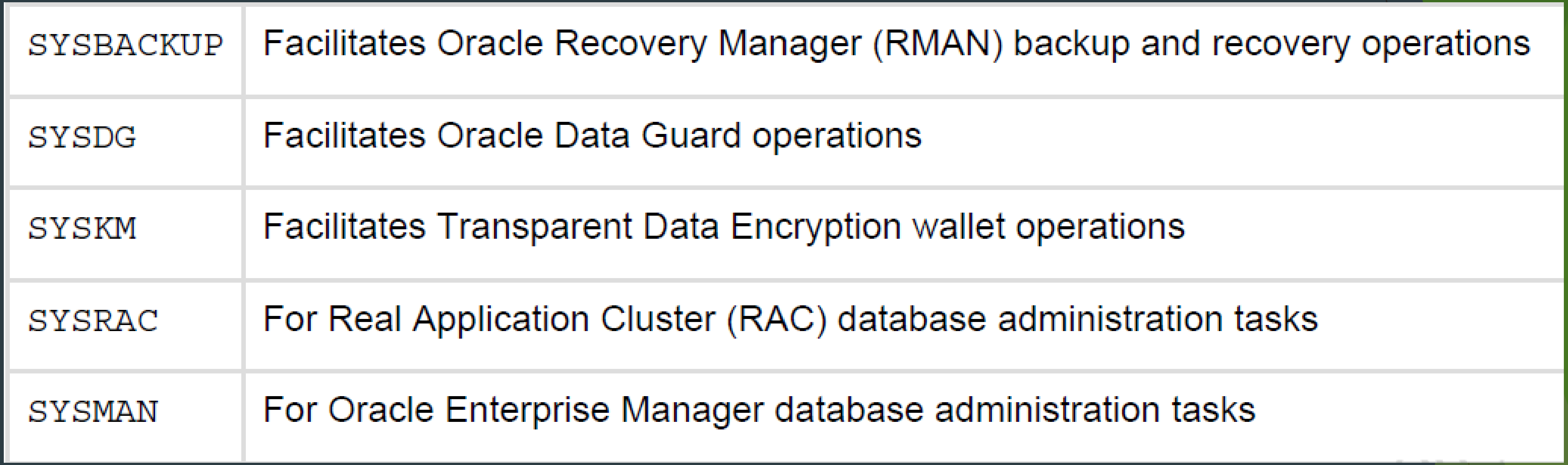
This account can perform all administrative functions except the following:

Backup and recovery

Database upgrade

Administering User Security

Oracle- supplied administrator accounts

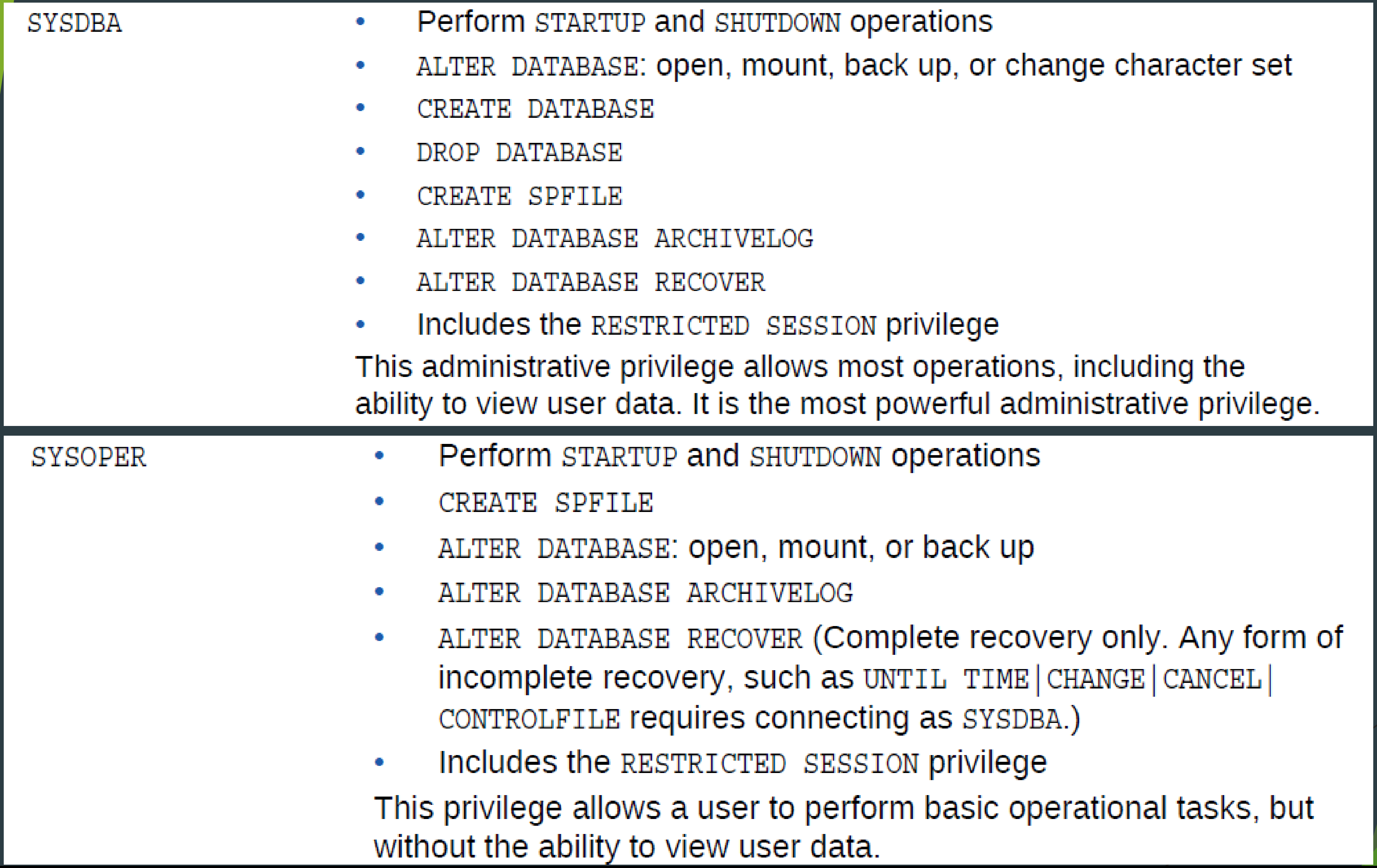


SYSBACKUP	Facilitates Oracle Recovery Manager (RMAN) backup and recovery operations
SYSDG	Facilitates Oracle Data Guard operations
SYSKM	Facilitates Transparent Data Encryption wallet operations
SYSRAC	For Real Application Cluster (RAC) database administration tasks
SYSMAN	For Oracle Enterprise Manager database administration tasks

Oracle Data Guard provides the management, monitoring, and automation software to create and maintain one or more standby databases to protect Oracle data from failures, disasters, human error, and data corruptions while providing high availability for mission critical applications. Data Guard is included with Oracle Database Enterprise Edition.
Transparent Data Encryption (TDE) enables you to encrypt sensitive data that you store in tables and tablespaces.

Administering User Security

Special system privileges for administrators



SYSDBA	• Perform STARTUP and SHUTDOWN operations • ALTER DATABASE: open, mount, back up, or change character set • CREATE DATABASE • DROP DATABASE • CREATE SPFILE • ALTER DATABASE ARCHIVELOG • ALTER DATABASE RECOVER • Includes the RESTRICTED SESSION privilege This administrative privilege allows most operations, including the ability to view user data. It is the most powerful administrative privilege.
SYSOPER	• Perform STARTUP and SHUTDOWN operations • CREATE SPFILE • ALTER DATABASE: open, mount, or back up • ALTER DATABASE ARCHIVELOG • ALTER DATABASE RECOVER (Complete recovery only. Any form of incomplete recovery, such as UNTIL TIME|CHANGE|CANCEL|CONTROLFILE requires connecting as SYSDBA.) • Includes the RESTRICTED SESSION privilege This privilege allows a user to perform basic operational tasks, but without the ability to view user data.

1

2

Administering User Security

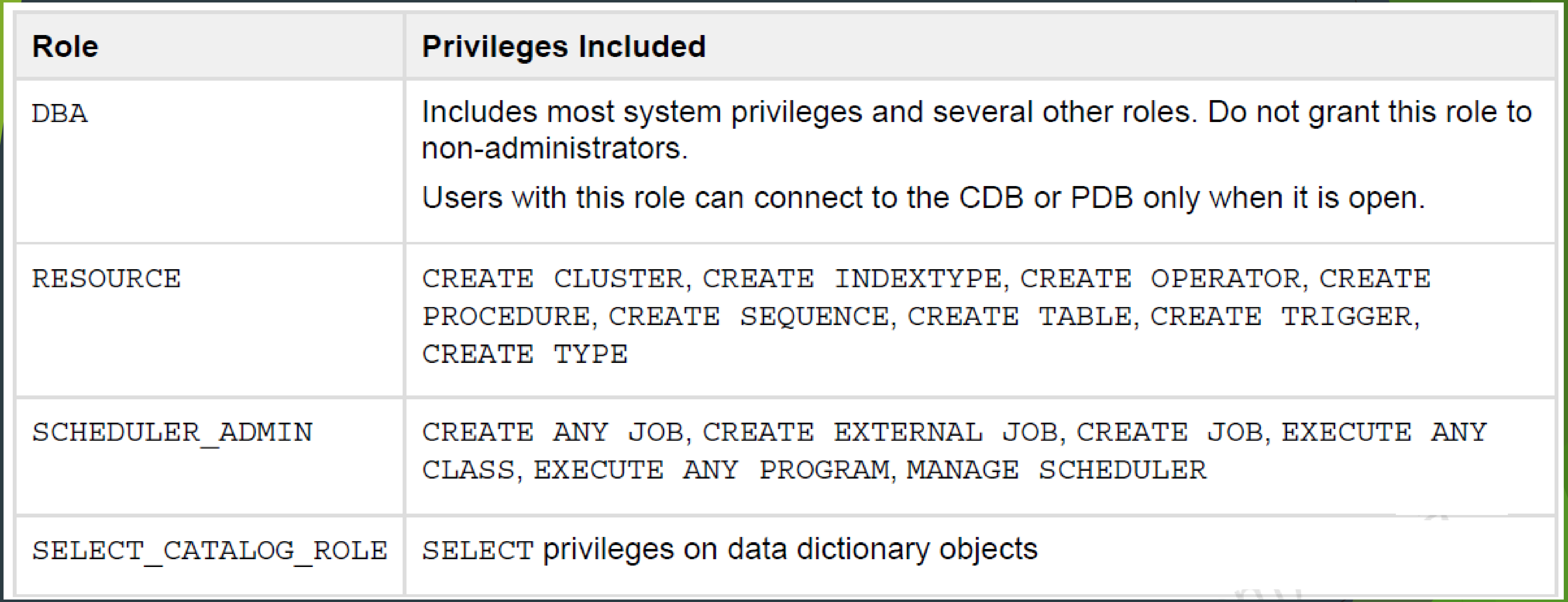
Special system privileges for administrators

SYSBACKUP	This privilege allows a user to perform backup and recovery operations either from Oracle Recovery Manager (RMAN) or SQL*Plus. See *Oracle Database Security Guide* for the full list of operations allowed by this administrative privilege.	3
SYSDG	This privilege allows a user to perform Data Guard operations. You can use this privilege with either Data Guard Broker or the `DGMGRL` command-line interface.	4
SYSKM	This privilege allows a user to perform Transparent Data Encryption keystore operations.	5
SYSRAC	This privilege allows the Oracle agent of Oracle Clusterware to perform Oracle Real Application Clusters (Oracle RAC) operations.	6

SYSASM is a system privilege that enables the separation of the SYSDBA database administration privilege from the Oracle ASM storage administration privilege 7

Administering User Security

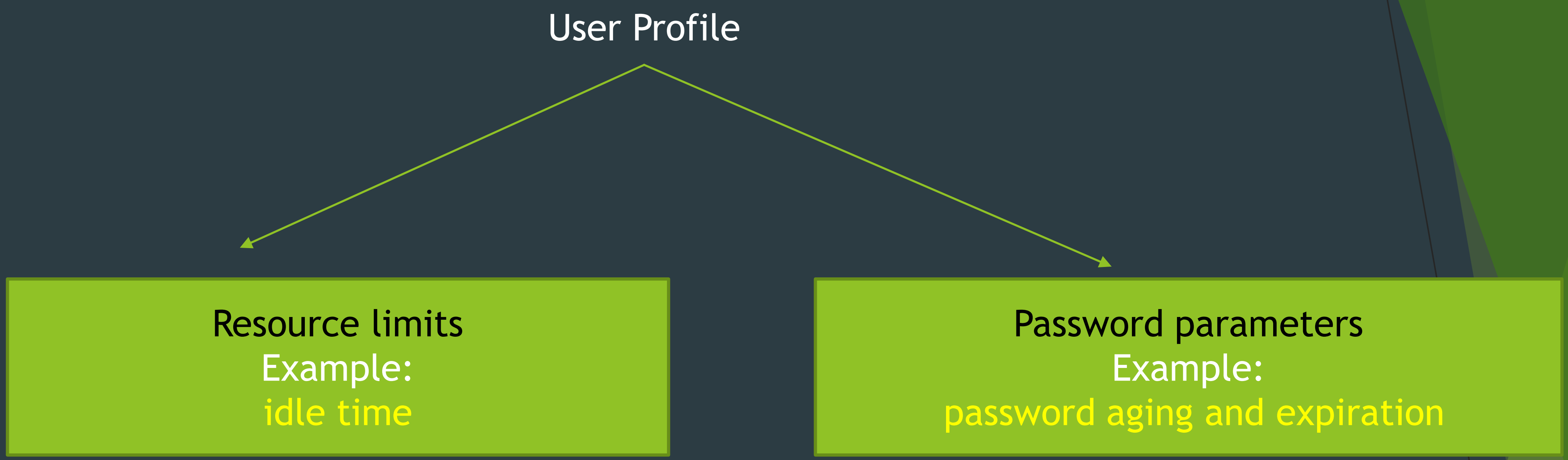
Oracle –supplied roles



Role	Privileges Included
DBA	Includes most system privileges and several other roles. Do not grant this role to non-administrators. 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

Note: SYS and SYSTEM users already have DBA role by default

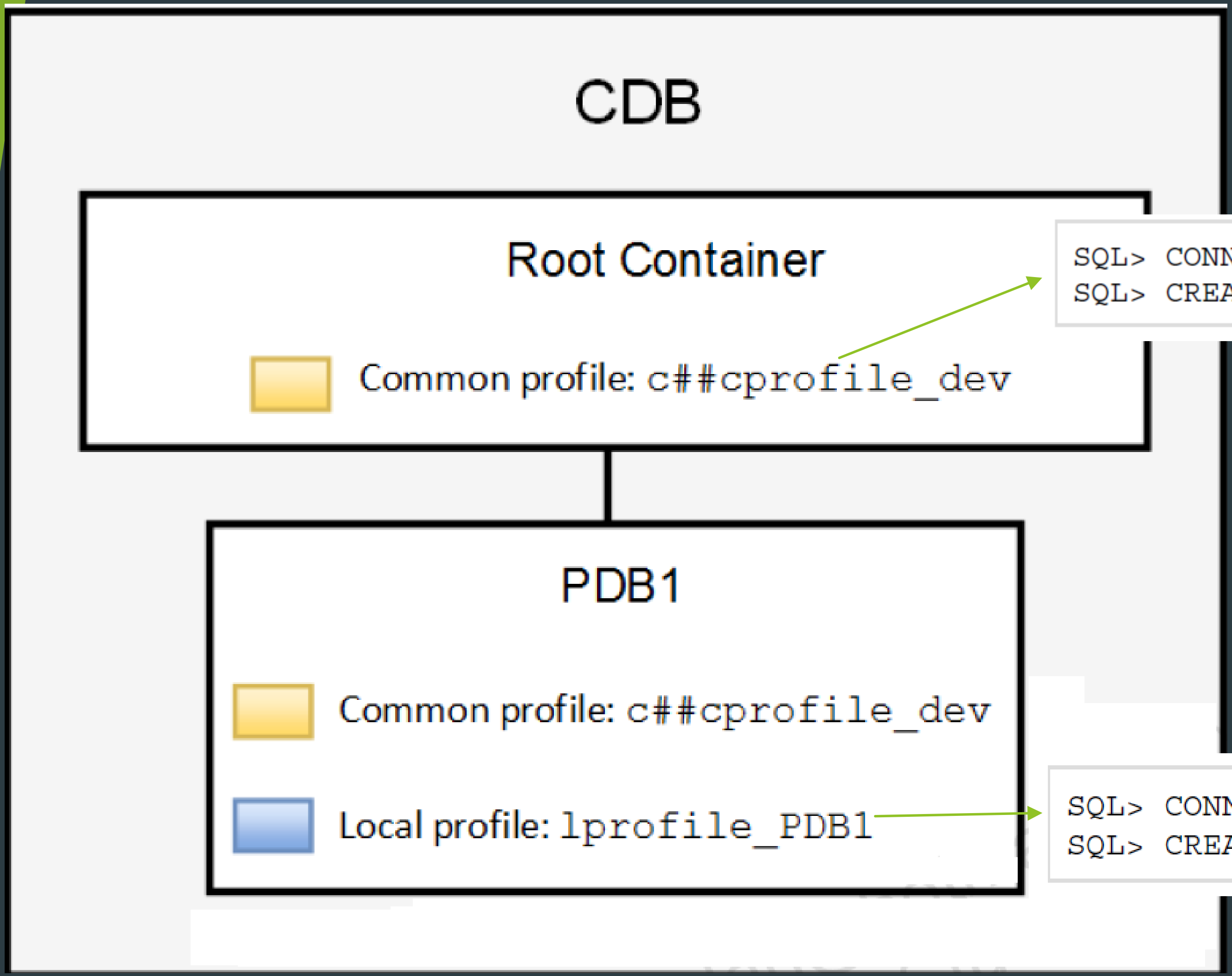
Administering User Security



- **a user profile** is a named set of resource limits and password parameters that restrict database usage and database instance resources for a user.
- If you assign a profile to a user, then that user can not exceed those limits.
- Every user , including the administrators is assigned to only one profile.
- By default when you crate a user, it will be assigned to default profile, unless you specified another profile

Administering User Security

User Profile

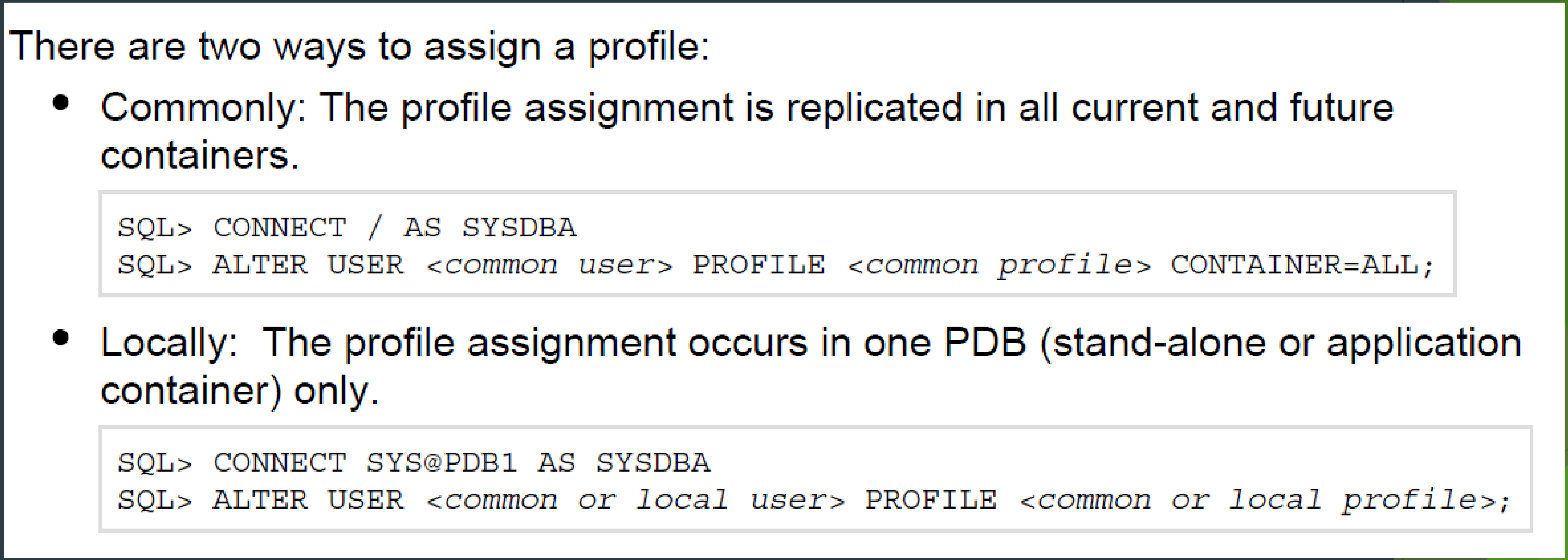

CDB
Root Container
Common profile: c##cprofile_dev
PDB1
Common profile: c##cprofile_dev
Local profile: lprofile_PDB1

```
SQL> CONNECT / AS SYSDBA
SQL> CREATE PROFILE c##cprofile_dev limit … CONTAINER=ALL;
```

```
SQL> CONNECT SYS@PDB1 AS SYSDBA
SQL> CREATE PROFILE lprofile_PDB1 limit … ;
```

Administering User Security

Assigning Profiles



There are two ways to assign a profile:

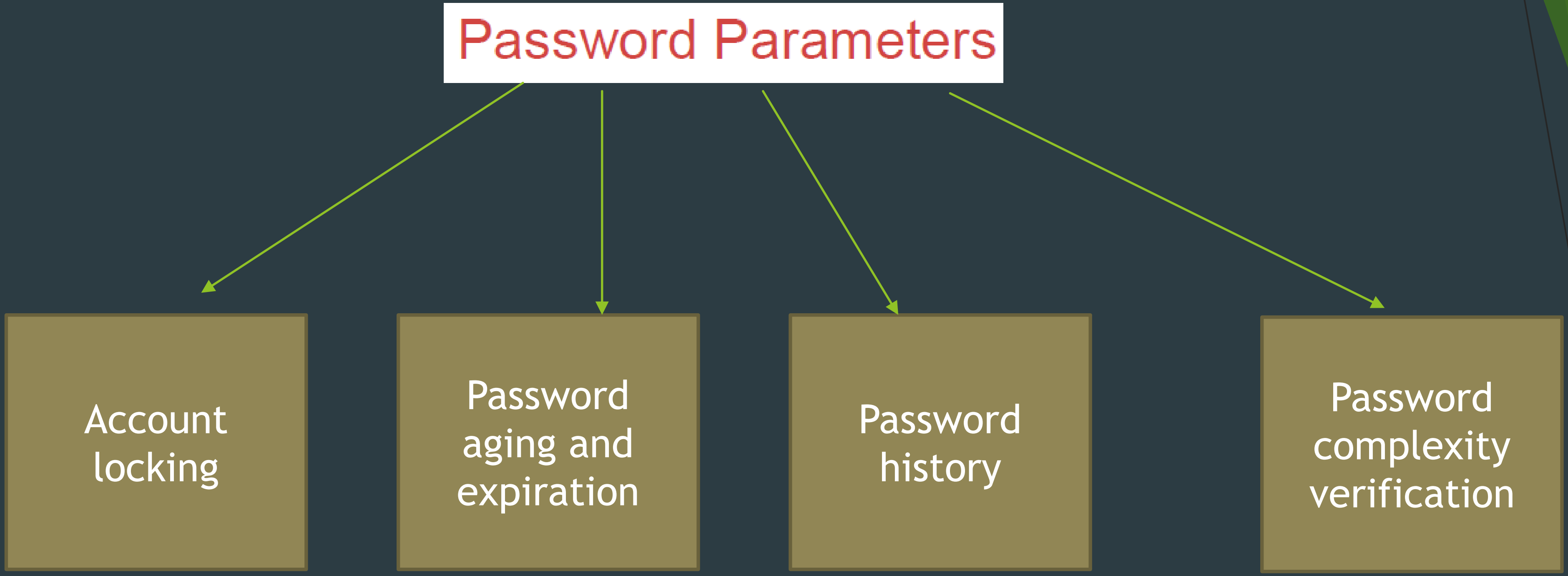
- Commonly: The profile assignment is replicated in all current and future containers.

```
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>;
```

Administering User Security


Password Parameters
Account locking
Password aging and expiration
Password history
Password complexity verification

Administering User Security

Password Parameters

1 **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 pre-defined number of days (meaning, users have not attempted to log in to their accounts).

We can configure the following parameters

- `FAILED_LOGIN_ATTEMPTS` specifies the number of failed login attempts before the lockout of the account.
- `PASSWORD_LOCK_TIME` specifies the number of days for which the account is locked after the specified number of failed login attempts.
- `INACTIVE_ACCOUNT_TIME` specifies the number of days an account can be inactive before it is locked.

Administering User Security

Password Parameters

2 **Password aging and expiration** enables user passwords to have a lifetime, after which the passwords expire and must be changed.

We can configure the following parameters

- `PASSWORD_LIFE_TIME` determines the lifetime of the password in days, after which the password expires.
- `PASSWORD_GRACE_TIME` specifies a grace period in days for changing the password after the first successful login after the password has expired.

Administering User Security

Password Parameters

3 **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.

We can configure the following parameters

- `PASSWORD_REUSE_TIME` specifies that a user cannot reuse a password for a given number of days.
- `PASSWORD_REUSE_MAX` specifies the number of password changes that are required before the current password can be reused.
- `PASSWORD_VERIFY_FUNCTION` checks for password complexity for the `SYS` user.

Administering User Security

Password Parameters

4 **Password complexity verification** makes a complexity check on the password to verify that it meets certain rules.

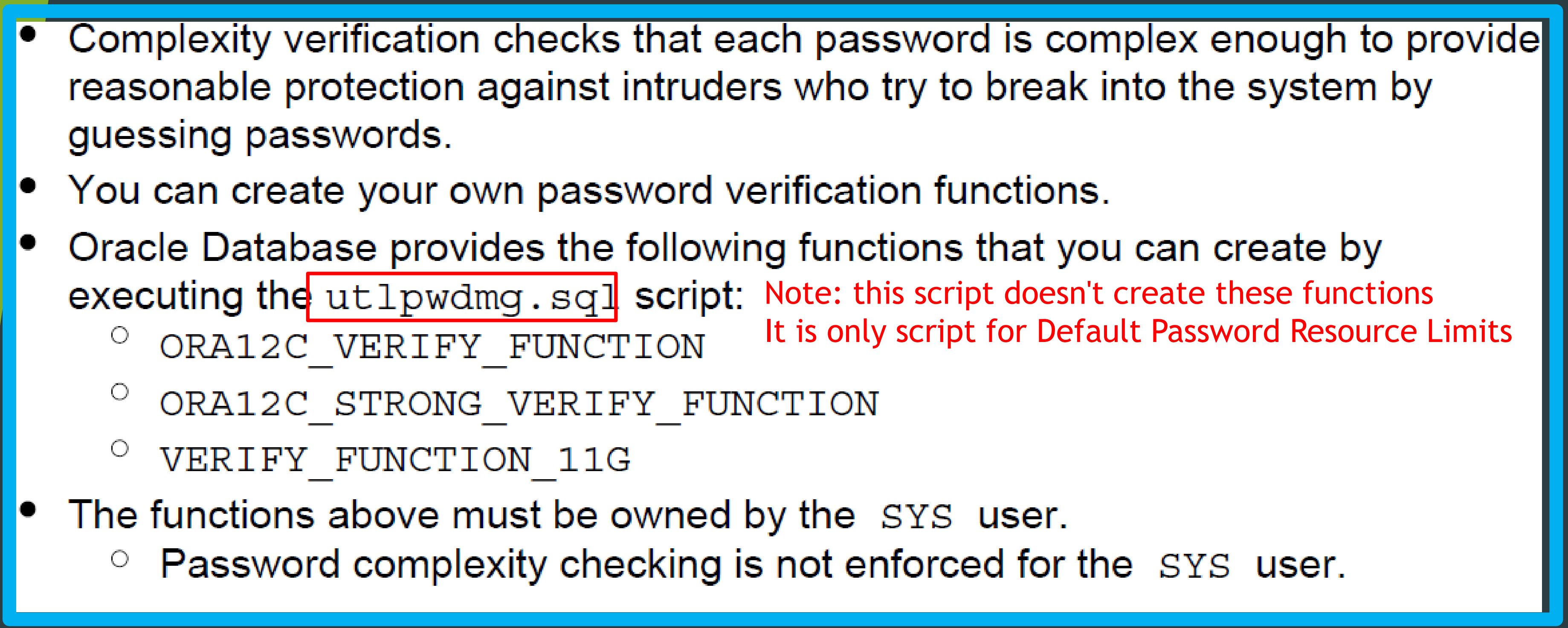
We control this by parameter :PASSWORD_VERIFY_FUNCTION

- it is PL/SQL function that perform password complexity check
- This function owned by user SYS
- It must return Boolean (true or false)
- A model verification function is provided in script called `utlpwdmg.sql`

$ORACLE_HOME/rdbms/admin

Administering User Security

Oracle-Supplied Password Verification Functions



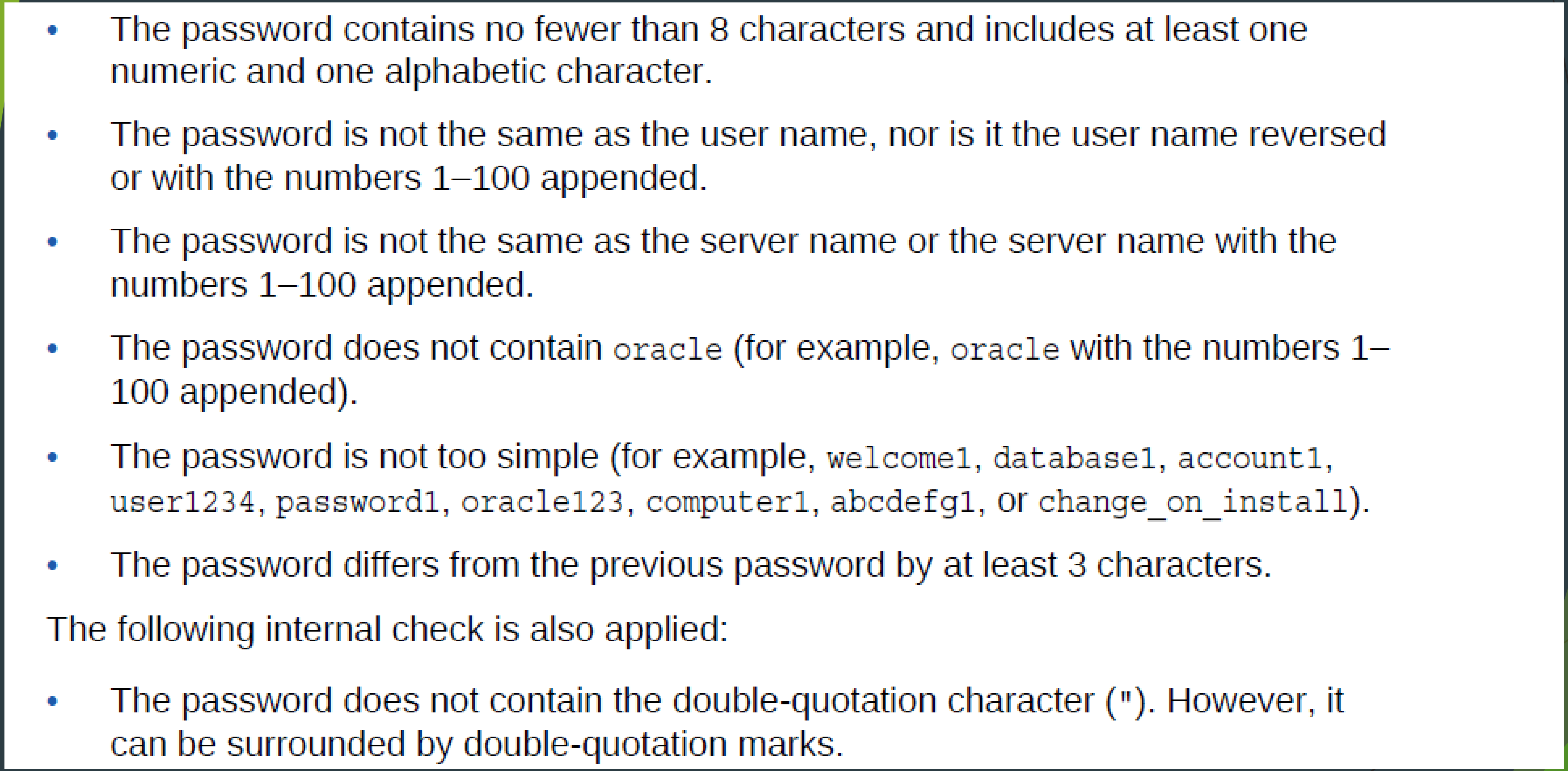
- Complexity verification checks that each password is complex enough to provide reasonable protection against intruders who try to break into the system by guessing passwords.
- You can create your own password verification functions.
- Oracle Database provides the following functions that you can create by executing the `utlpwdmg.sql` script: Note: this script doesn't create these functions
It is only script for Default Password Resource Limits
  - `ORA12C_VERIFY_FUNCTION`
  - `ORA12C_STRONG_VERIFY_FUNCTION`
  - `VERIFY_FUNCTION_11G`
- The functions above must be owned by the `SYS` user.
  - Password complexity checking is not enforced for the `SYS` user.

It is
catpvf.sql
Which create
these functions

Administering User Security

Oracle-Supplied Password Verification Functions

verify_function_11G Function Password Requirements



- The password contains no fewer than 8 characters and includes at least one numeric and one alphabetic character.
- The password is not the same as the user name, nor is it the user name reversed or with the numbers 1–100 appended.
- The password is not the same as the server name or the server name with the numbers 1–100 appended.
- The password does not contain `oracle` (for example, `oracle` with the numbers 1–100 appended).
- The password is not too simple (for example, `welcome1`, `database1`, `account1`, `user1234`, `password1`, `oracle123`, `computer1`, `abcdefg1`, or `change_on_install`).
- The password differs from the previous password by at least 3 characters.

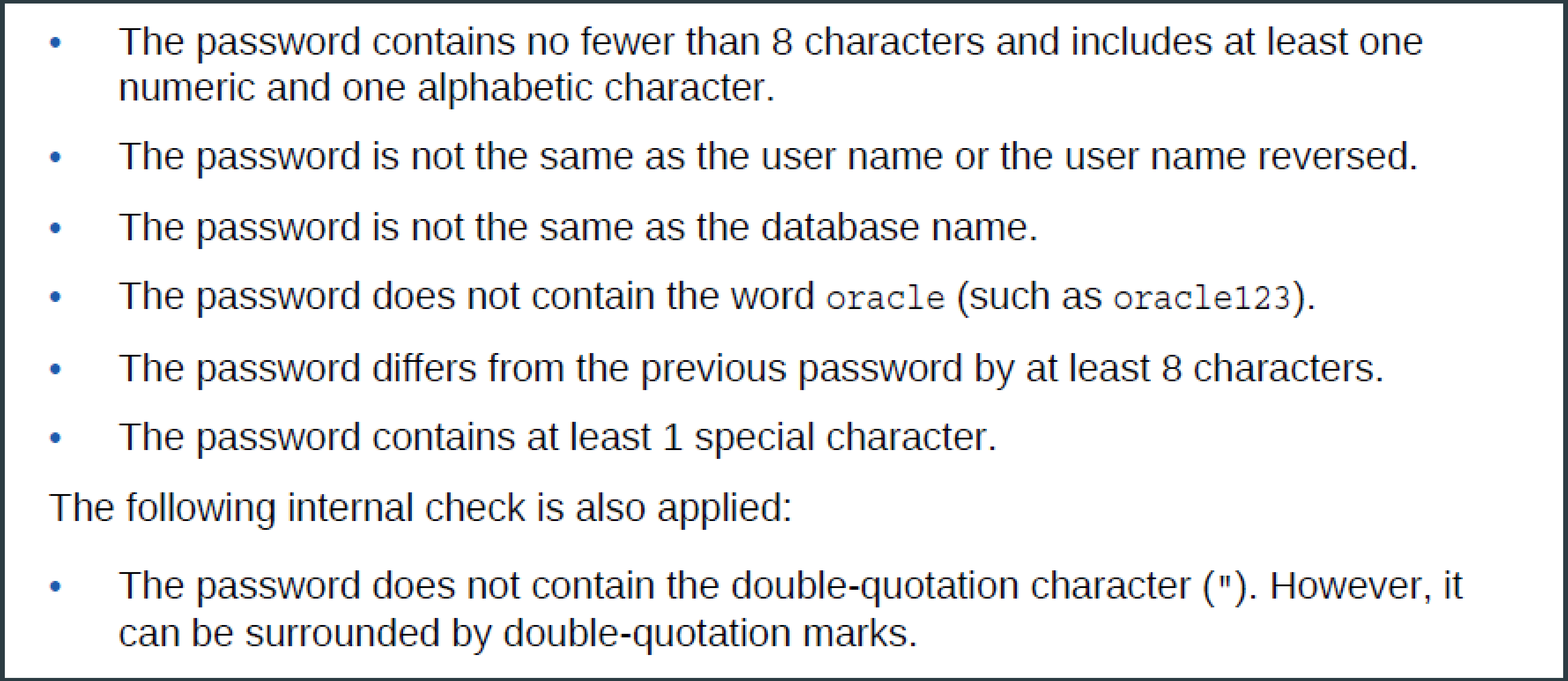
The following internal check is also applied:

- The password does not contain the double-quotation character ("). However, it can be surrounded by double-quotation marks.

Administering User Security

Oracle-Supplied Password Verification Functions

ora12c_verify_function Function Password Requirements



- The password contains no fewer than 8 characters and includes at least one numeric and one alphabetic character.
- The password is not the same as the user name or the user name reversed.
- The password is not the same as the database name.
- The password does not contain the word `oracle` (such as `oracle123`).
- The password differs from the previous password by at least 8 characters.
- The password contains at least 1 special character.

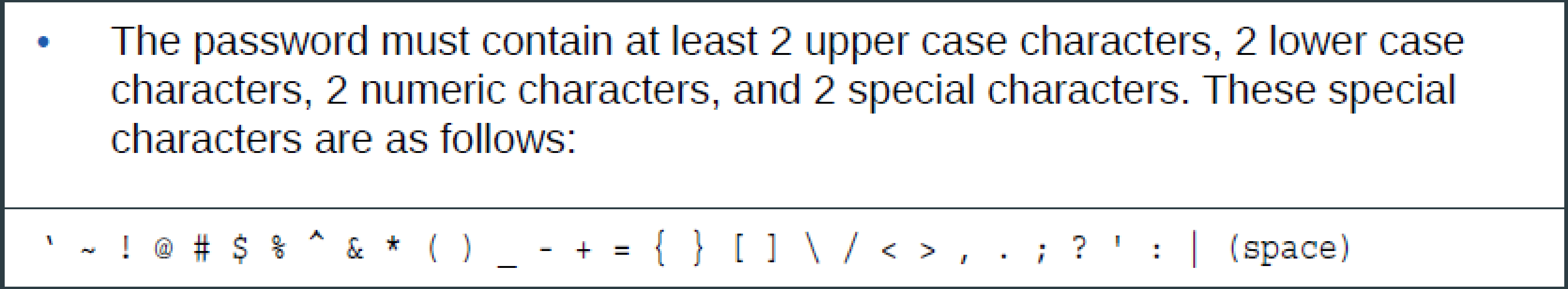
The following internal check is also applied:

- The password does not contain the double-quotation character (`"`). However, it can be surrounded by double-quotation marks.

Administering User Security

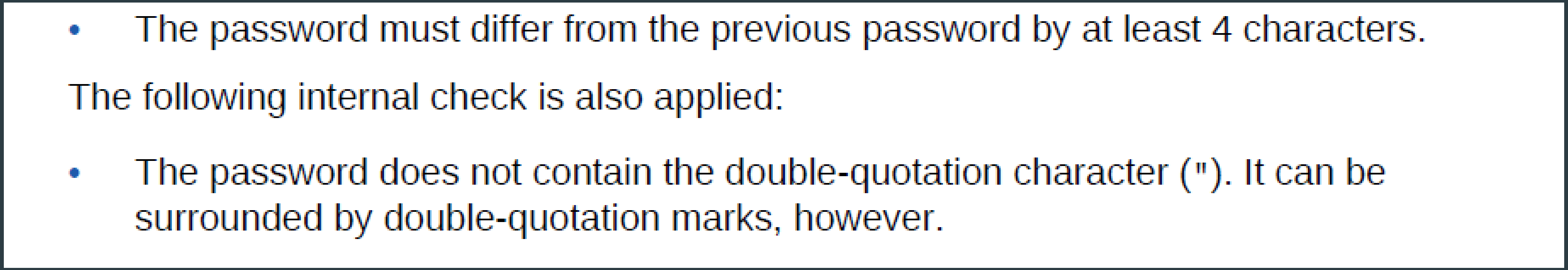
Oracle-Supplied Password Verification Functions

ora12c_strong_verify_function **Function** Password Requirements



- The password must contain at least 2 upper case characters, 2 lower case characters, 2 numeric characters, and 2 special characters. These special characters are as follows:

```
` ~ ! @ # $ % ^ & * ( ) _ - + = { } [ ] \ / < > , . ; ? ' : | (space)
```



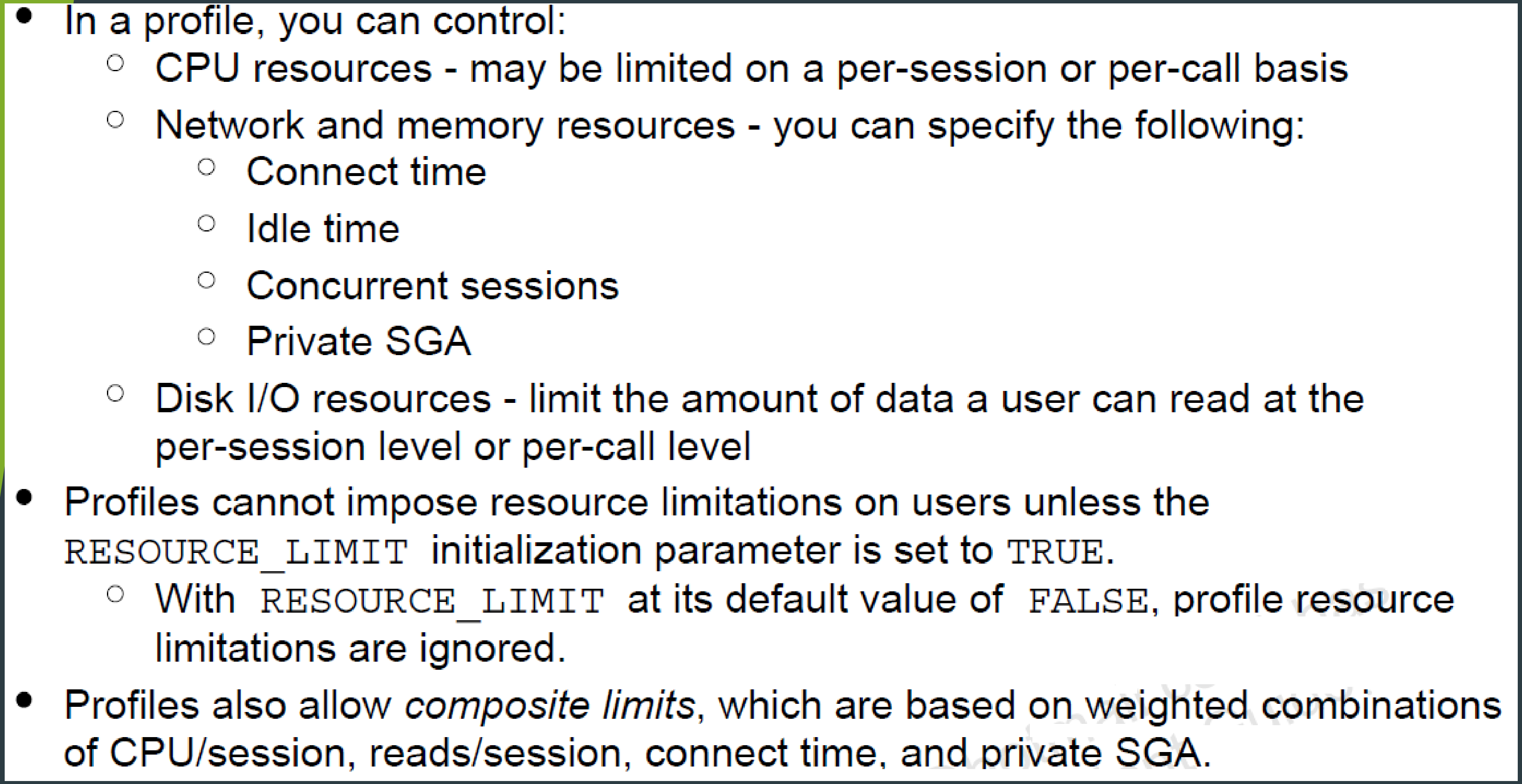
- The password must differ from the previous password by at least 4 characters.

The following internal check is also applied:

- The password does not contain the double-quotation character ("). It can be surrounded by double-quotation marks, however.

Administering User Security

Resource Parameters

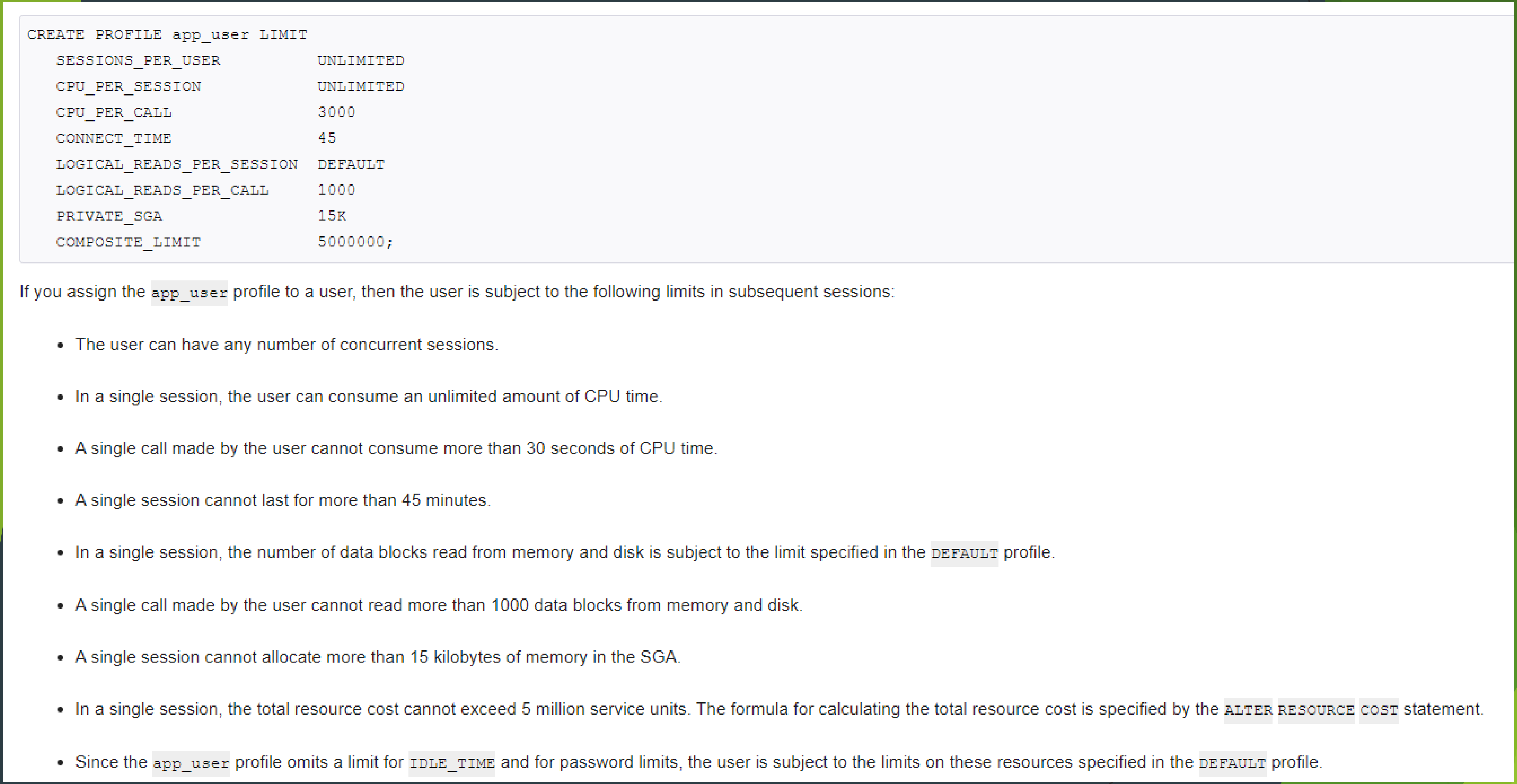


- In a profile, you can control:
  - CPU resources - may be limited on a per-session or per-call basis
  - Network and memory resources - you can specify the following:
    - Connect time
    - Idle time
    - Concurrent sessions
    - Private SGA
  - Disk I/O resources - limit the amount of data a user 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.

CPU_PER_SESSION/ CPU_PER_CALL

Example:
CPU_PER_CALL 3000
A single call made by the user cannot consume more than 30 seconds of CPU time.

Administering User Security



```
CREATE PROFILE app_user LIMIT
   SESSIONS_PER_USER          UNLIMITED
   CPU_PER_SESSION            UNLIMITED
   CPU_PER_CALL               3000
   CONNECT_TIME               45
   LOGICAL_READS_PER_SESSION  DEFAULT
   LOGICAL_READS_PER_CALL     1000
   PRIVATE_SGA                15K
   COMPOSITE_LIMIT            5000000;
```

If you assign the `app_user` profile to a user, then the user is subject to the following limits in subsequent sessions:

- The user can have any number of concurrent sessions.
- In a single session, the user can consume an unlimited amount of CPU time.
- A single call made by the user cannot consume more than 30 seconds of CPU time.
- A single session cannot last for more than 45 minutes.
- In a single session, the number of data blocks read from memory and disk is subject to the limit specified in the `DEFAULT` profile.
- A single call made by the user cannot read more than 1000 data blocks from memory and disk.
- A single session cannot allocate more than 15 kilobytes of memory in the SGA.
- In a single session, the total resource cost cannot exceed 5 million service units. The formula for calculating the total resource cost is specified by the `ALTER RESOURCE COST` statement.
- Since the `app_user` profile omits a limit for `IDLE_TIME` and for password limits, the user is subject to the limits on these resources specified in the `DEFAULT` profile.