9

# Administering User Security

# Objectives

After completing this lesson, you should be able to:

- Create and manage database user accounts:
  - Authenticate users
  - Assign default storage areas (tablespaces)
- Grant and revoke privileges
- Create and manage roles
- Create and manage profiles:
  - Implement standard password security features
  - Control resource usage by users

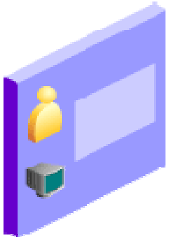
The following terms relate to administering database users and assist you in understanding the objectives:

- A *database user account* is a way to organize the ownership of and access to database objects.
- A *password* is an authentication by the Oracle database.
- A *privilege* is a right to execute a particular type of SQL statement or to access another user's object.
- A *role* is a named group of related privileges that are granted to users or to other roles.
- *Profiles* impose a named set of resource limits on database usage and instance resources, and manage account status and password management rules.
- A *quota* is a space allowance in a given tablespace. This is one of the ways by which you can control resource usage by users.

# Database User Accounts

Each database user account has:

- A unique username
- An authentication method
- A default tablespace
- A temporary tablespace
- A user profile
- An initial consumer group
- An account status



A schema:

- Is a collection of database objects that are owned by a database user
- Has the same name as the user account

To access the database, a user must specify a valid database user account and successfully authenticate as required by that user account. Each database user has a unique database account.

Oracle recommends this to avoid potential security holes and provide meaningful data for certain audit activities. However, users may sometimes share a common database account. In these rare cases, the operating system and applications must provide adequate security for the database. Each user account has:

- **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. Oracle Database supports password, global, and external authentication methods (such as biometric, certificate, and token authentication).
- **Default tablespace:** This is a place where a user creates objects if the user does not specify some other tablespace. Note that having a default tablespace does not imply that the user has the *privilege* of creating objects in that tablespace, nor does the user have a *quota* of space in that tablespace in which to create objects. Both of these are granted separately.

- **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.
- **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.”

**Schemas:** A *schema* is a collection of database objects that are owned by a database user. Schema objects are the logical structures that directly refer to the database’s data. Schema objects include such structures as tables, views, sequences, stored procedures, synonyms, indexes, clusters, and database links. In general, schema objects include everything that your application creates in the database.

**Note:** A database user is not necessarily a person. It is a common practice to create a user that owns the database objects of a particular application, such as HR. The database user can be a device, an application, or just a way to group database objects for security purposes. The personal identifying information of a person is not needed for a database user.

# Predefined Administrative Accounts

- SYS:
  - Owns the data dictionary and the Automatic Workload Repository (AWR)
  - Used for startup and shutdown of 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

The SYS and SYSTEM accounts have the database administrator (DBA) role granted to them by default. In addition, the SYS account has all privileges with ADMIN OPTION and owns the data dictionary. To connect to the SYS account, you must use the AS SYSDBA clause for a database instance and AS SYSASM for an Automatic Storage Management (ASM) instance. Any user that is granted the SYSDBA privilege can connect to the SYS account by using the AS SYSDBA clause. Only “privileged” users who are granted the SYSDBA, SYSOPER, or SYSASM privileges are allowed to start up and shut down instances. The SYSTEM account does not have the SYSDBA privilege. SYSTEM is also granted the AQ_ADMINISTRATOR_ROLE and MGMT_USER roles. The SYS and SYSTEM accounts are required accounts in the database. They cannot be dropped.

**Best practice:** Applying the principle of least privilege, these accounts are not used for routine operations. Users who need DBA privileges have separate accounts with the required privileges granted to them.

The SYSBACKUP, SYSDG, and SYSKM users are created to facilitate separation of duties for database administrators. Each of these provides a designated use for an administrative privilege by the same name. You should create a user and grant the appropriate administrative privilege to that user.

Privileges are discussed in detail later in the lesson.

# Administrative Privileges

| Privilege | Description |
|---|---|
| `SYSDBA` | Standard database operations, such as starting and shutting down the database instance, creating the server parameter file (SPFILE), and changing the ARCHIVELOG mode<br>Allows the grantee to view user data |
| `SYSOPER` | Standard database operations, such as starting and shutting down the database instance, creating the server parameter file (SPFILE), and changing the ARCHIVELOG mode |
| `SYSBACKUP` | Oracle Recovery Manager (RMAN) backup and recovery operations by using RMAN or SQL*Plus |
| `SYSDG` | Data Guard operations by using the Data Guard Broker or the DGMGRL command-line interface |
| `SYSKM` | Manage Transparent Data Encryption wallet operations |

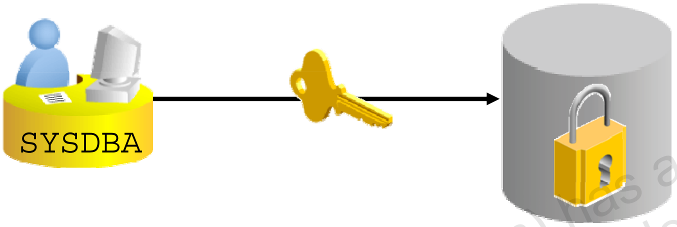
Oracle Database includes five administrative privileges that are provided to facilitate separation of duty. The `SYSDBA` and `SYSOPER` administrative privileges are used to perform a variety of standard database operations including starting up the database instance and shutting it down. Refer to the *Oracle Database Administrator's Guide* for a complete list of authorized operations for the `SYSDBA` and `SYSOPER` privileges.

`SYSBACKUP`, `SYSDG`, and `SYSKM` are new to Oracle Database 12*c* and are tailored for the specific administrative tasks of backup and recovery, Oracle Data Guard, and Transparent Data Encryption key management. In previous releases, the `SYSDBA` privilege was required for these tasks. These privileges enable you to connect to the database even if the database is not open. Refer to the *Oracle Database Security Guide* for a list of supported operations for the `SYSBACKUP`, `SYSDG`, and `SYSKM` privileges.

# Protecting Privileged Accounts

Privileged accounts can be protected by:

- Using a password file with case-sensitive passwords
- Enabling strong authentication for administrator roles



Users with SYSDBA or SYSOPER privileges must always be authenticated. When connecting locally, the user is authenticated by the local OS by being a member of a privileged OS group. If connecting remotely, a password file is used to authenticate privileged users. If the password file is configured, it will be checked first. In Oracle Database, these passwords are case-sensitive. Oracle Database provides other methods that make remote administrator authentication more secure and centralize the administration of these privileged users.

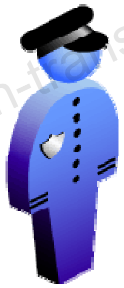
When a database is created using the Database Configuration Assistant, the password file is case-sensitive. If you upgrade from earlier database versions, be sure to make the password file case-sensitive for remote connections:

```
orapwd file=orapworcl entries=5 ignorecase=N
```

If your concern is that the password file might be vulnerable or that the maintenance of many password files is a burden, strong authentication can be implemented. The Advanced Security option is required if you want to use strong authentication methods. For more information about strong authentication, see the *Oracle Database Advanced Security Administrator's Guide*.

# Authenticating Users

- Password: User definition includes a password that must be supplied when the user attempts to log in to the database
- External: Authentication by a method outside the database (operating system, Kerberos, or Radius)
- Global: Users are identified by using an LDAP-based directory service



*Authentication* means verifying the identity of someone or something (a user, device, or other entity) that wants to use data, resources, or applications. Validating that identity establishes a trust relationship for further interactions. Authentication also enables accountability by making it possible to link access and actions to specific identities. After authentication, authorization processes can allow or limit the levels of access and action that are permitted to that entity.

When you create a user, you must decide on the authentication technique to use, which can be modified later.

**Password:** This 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. These passwords may also contain multibyte characters and are limited to 30 bytes. Each password created in a database that is upgraded to Oracle Database 12*c* remains non-case-sensitive until the password is changed.

Passwords are always automatically and transparently encrypted using the Advanced Encryption Standard (AES) algorithm during network (client/server and server/server) connections before sending them across the network.

**External:** This is authentication by a method outside the database (operating system, Kerberos, or Radius). The Advanced Security Option is required for Kerberos or Radius. Users can connect to the Oracle database without specifying a username or password. The Advanced Security Option (which is a strong authentication) allows users to be identified by using biometrics, x509 certificates, and token devices. With external authentication, your database relies on the underlying operating system, network authentication service, or external authentication service to restrict access to database accounts. A database password is not used for this type of login. If your operating system or network service permits, you can have it authenticate users. If you use operating system authentication, set the `OS_AUTHENT_PREFIX` initialization parameter and use this prefix in Oracle usernames. The `OS_AUTHENT_PREFIX` parameter defines a prefix that the Oracle database adds to the beginning of each user's operating system account name. The default value of this parameter is `OPS$` for backward compatibility with the previous versions of the Oracle software. The Oracle database compares the prefixed username with the Oracle usernames in the database when a user attempts to connect. For example, suppose that `OS_AUTHENT_PREFIX` is set as follows:

```
OS_AUTHENT_PREFIX=OPS$
```

If a user with an operating system account named `tsmith` needs to connect to an Oracle database and be authenticated by the operating system, the Oracle database checks whether there is a corresponding database user `OPS$tsmith` and, if so, allows the user to connect. All references to a user who is authenticated by the operating system must include the prefix, as seen in `OPS$tsmith`.

**Note:** The text of the `OS_AUTHENT_PREFIX` initialization parameter is case-sensitive on some operating systems. See the Oracle documentation that is specific to your operating system for more information about this initialization parameter.

**Global:** With the Oracle Advanced Security option, global authentication enables users to be identified by using an LDAP-based directory service.

For more information about advanced authentication methods, see the *Oracle Database Security* course.

# Administrator Authentication

Operating system security:

- DBAs must have the OS privileges to create and delete files.
- Typical database users should not have the OS privileges to create or delete database files.

Administrator security:

- For SYSDBA and SYSOPER connections:
  - DBA user by name is audited for password file and strong authentication methods
  - OS account name is audited for OS authentication
  - OS authentication takes precedence over password file authentication for privileged users
  - Password file uses case-sensitive passwords

**Operating system security:** In UNIX and Linux, DBAs by default belong to the `oinstall` OS group, which has the required privileges to create and delete database files.

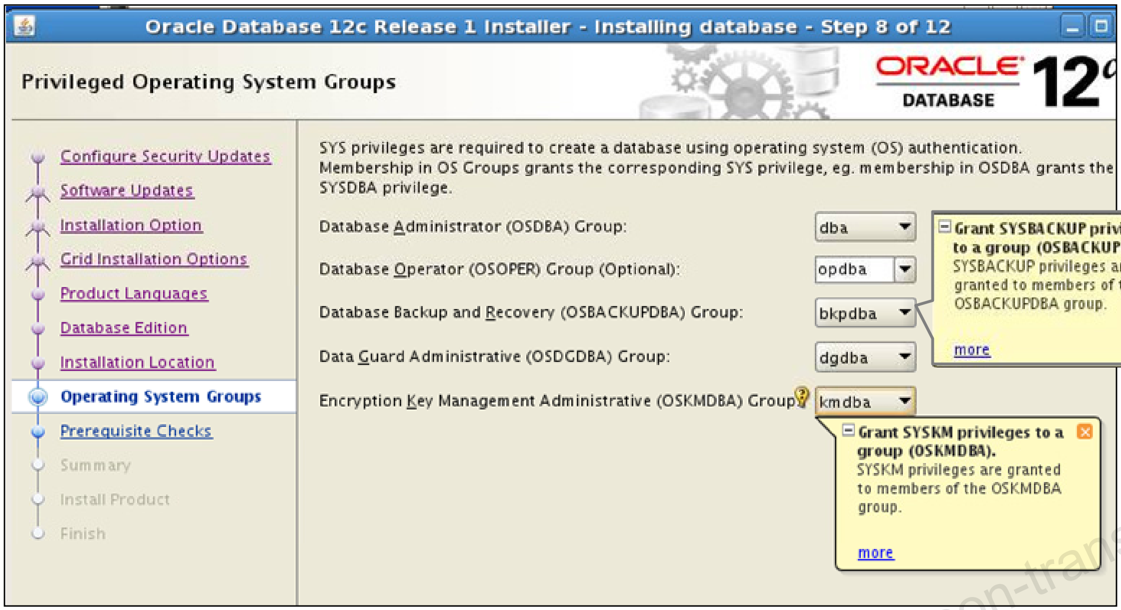
**Administrator security:** Connections for the privileged users `SYSDBA`, `SYSOPER`, and `SYSASM` are authorized only after verification with the password file or with the OS privileges and permissions. If OS authentication is used, the database does *not* use the supplied username and password. OS authentication is used if there is no password file, if the supplied username or password is not in that file, or if no username and password are supplied. The password file in Oracle Database 12*c* uses case-sensitive passwords by default.

However, if authentication succeeds by means of the password file, the connection is logged with the username. If authentication succeeds by means of the operating system, it is a `CONNECT /` connection that does not record the specific user.

**Note:** If you are a member of the `OSDBA` or `OSOPER` group for the operating system and you connect as `SYSDBA` or `SYSOPER`, you will be connected with the associated administrative privileges regardless of the username and password that you specify.

In Oracle Database 12*c*, a privileged user may use strong authentication methods: Kerberos, SSL, or directory authentication if the Advanced Security Option is licensed.

# OS Authentication and OS Groups



Oracle Database 12c Release 1 Installer - Installing database - Step 8 of 12

Privileged Operating System Groups

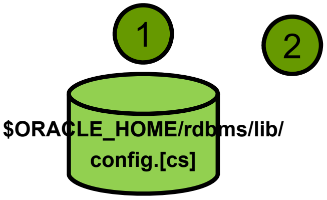
- Configure Security Updates
- Software Updates
- Installation Option
- Grid Installation Options
- Product Languages
- Database Edition
- Installation Location
- Operating System Groups
- Prerequisite Checks
- Summary
- Install Product
- Finish

SYS privileges are required to create a database using operating system (OS) authentication. Membership in OS Groups grants the corresponding SYS privilege, eg. membership in OSDBA grants the SYSDBA privilege.

|  |  |
|---|---|
| Database Administrator (OSDBA) Group: | dba |
| Database Operator (OSOPER) Group (Optional): | opdba |
| Database Backup and Recovery (OSBACKUPDBA) Group: | bkpdba |
| Data Guard Administrative (OSDGDBA) Group: | dgdba |
| Encryption Key Management Administrative (OSKMDBA) Group: | kmdba |

Grant SYSBACKUP privileges to a group (OSBACKUPDBA). SYSBACKUP privileges are granted to members of the OSBACKUPDBA group. more

Grant SYSKM privileges to a group (OSKMDBA). SYSKM privileges are granted to members of the OSKMDBA group. more



2

```
#define SS_DBA_GRP "dba"
#define SS_OPER_GRP "opdba"
#define SS_ASM_GRP ""
#define SS_BKP_GRP "bkpdba"
#define SS_DGD_GRP "dgdba"
#define SS_KMT_GRP "kmdba"
```

3

```
$ mv config.o config.o.orig
$ make -f ins_rdbms.mk ioracle
```

In OS authentication, the groups are created and assigned specific names as part of the database installation process, provided that OS UNIX or WINDOWS groups are created and accounts are assigned to the appropriate operating-system-defined groups.

Membership in a UNIX or WINDOWS group affects your connection to the database as follows:

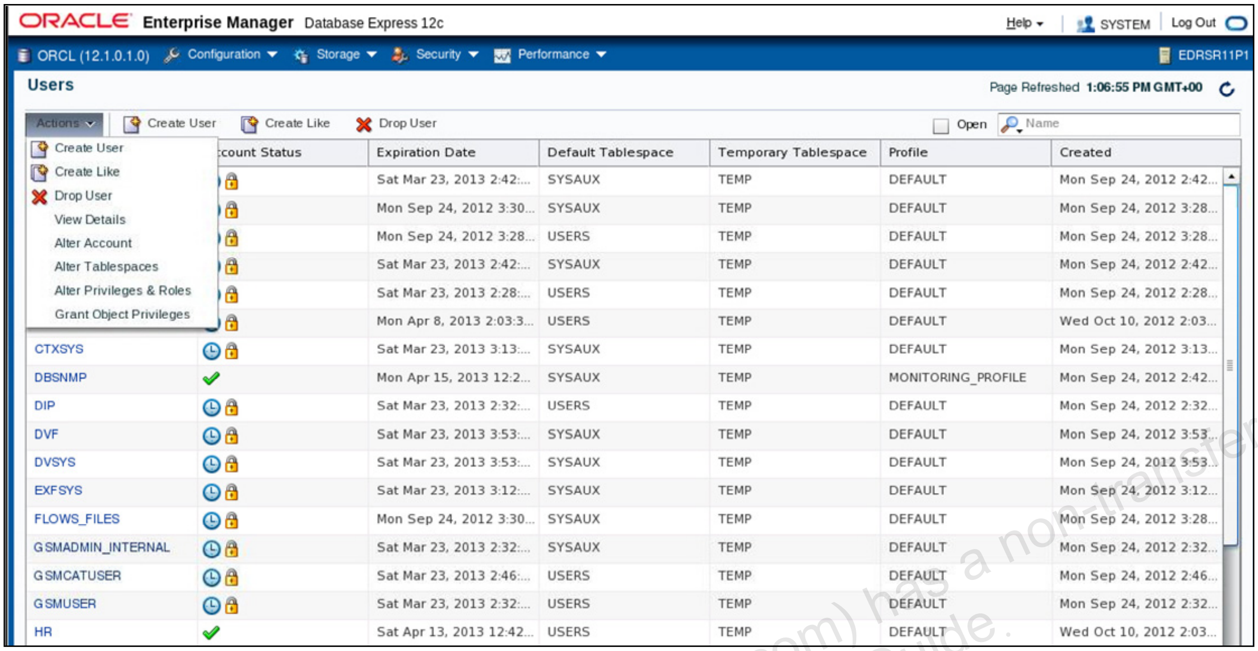
- If you are a member of the `OSBACKUP` group, and you specify `AS SYSBACKUP` when you connect to the database, you connect to the database with the `SYSBACKUP` administrative privilege under the `SYSBACKUP` user.
- If you are a member of the `OSDG` group, and you specify `AS SYSDG` when you connect to the database, you connect to the database with the `SYSDG` administrative privilege under the `SYSDG` user.
- If you are a member of the `OSKM` group, and you specify `AS SYSKM` when you connect to the database, you connect to the database with the `SYSKM` administrative privilege under the `SYSKM` user.

If they are not members of these OS groups, users will not be able to connect as administrative users with the OS authentication. That is, "CONNECT / AS SYSDBA" will fail. However, the users can still connect using other authentication mechanisms (for example, network, password, or directory-based authentication). To change the OS group names (as shown in step 1 and 2 in the slide), ensure that you are using the groups defined in the $ORACLE_HOME/rdbms/lib/config.[cs] file. Next, shut down all databases, and then relink the Oracle executable as shown in step 3 in the slide.

### Windows User Groups

The Windows user groups are ORA_%HOMENAME%_SYSBACKUP, ORA_%HOMENAME%_SYSDG, and ORA_%HOMENAME%_SYSKM. These user groups cannot be changed.

# Managing Users

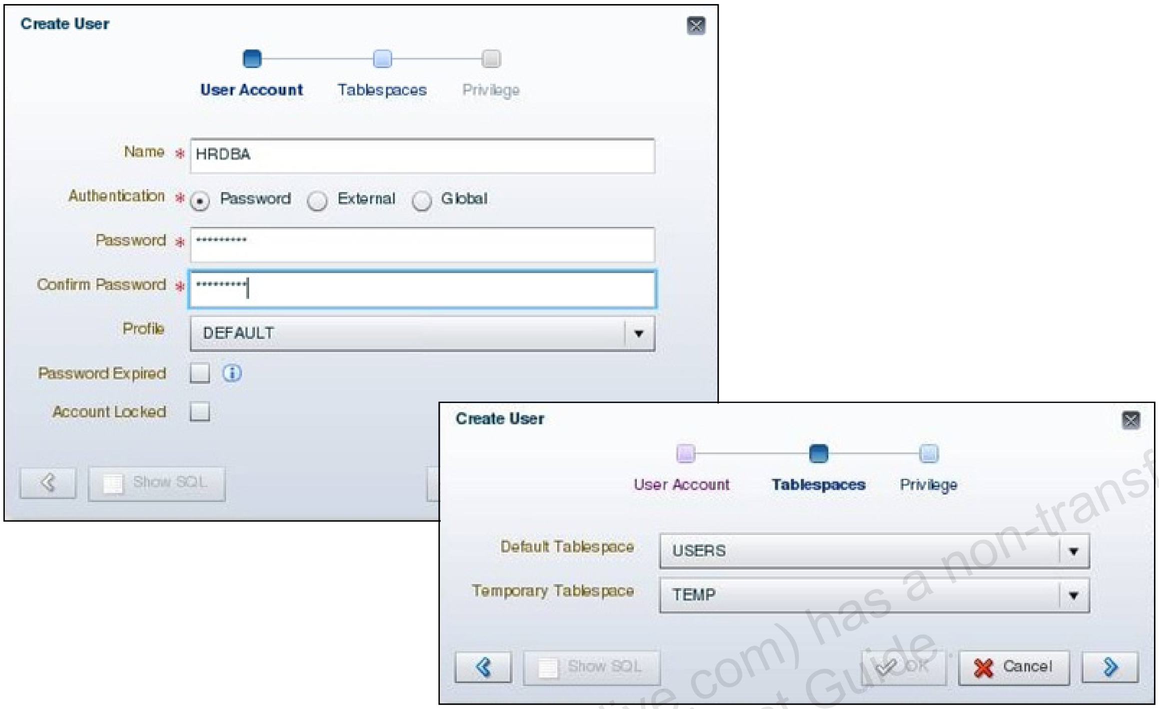


| | :count Status | Expiration Date | Default Tablespace | Temporary Tablespace | Profile | Created |
|---|---|---|---|---|---|---|
| | 🔒 | Sat Mar 23, 2013 2:42:... | SYSAUX | TEMP | DEFAULT | Mon Sep 24, 2012 2:42... |
| | 🔒 | Mon Sep 24, 2012 3:30... | SYSAUX | TEMP | DEFAULT | Mon Sep 24, 2012 3:28... |
| | 🔒 | Mon Sep 24, 2012 3:28... | USERS | TEMP | DEFAULT | Mon Sep 24, 2012 3:28... |
| | 🔒 | Sat Mar 23, 2013 2:42:... | SYSAUX | TEMP | DEFAULT | Mon Sep 24, 2012 2:42... |
| | 🔒 | Sat Mar 23, 2013 2:28:... | USERS | TEMP | DEFAULT | Mon Sep 24, 2012 2:28... |
| | 🔒 | Mon Apr 8, 2013 2:03:3... | USERS | TEMP | DEFAULT | Wed Oct 10, 2012 2:03... |
| CTXSYS | 🕒🔒 | Sat Mar 23, 2013 3:13:... | SYSAUX | TEMP | DEFAULT | Mon Sep 24, 2012 3:13... |
| DBSNMP | ✓ | Mon Apr 15, 2013 12:2... | SYSAUX | TEMP | MONITORING_PROFILE | Mon Sep 24, 2012 2:42... |
| DIP | 🕒🔒 | Sat Mar 23, 2013 2:32:... | USERS | TEMP | DEFAULT | Mon Sep 24, 2012 2:32... |
| DVF | 🕒🔒 | Sat Mar 23, 2013 3:53:... | SYSAUX | TEMP | DEFAULT | Mon Sep 24, 2012 3:53... |
| DVSYS | 🕒🔒 | Sat Mar 23, 2013 3:53:... | SYSAUX | TEMP | DEFAULT | Mon Sep 24, 2012 3:53... |
| EXFSYS | 🕒🔒 | Sat Mar 23, 2013 3:12:... | SYSAUX | TEMP | DEFAULT | Mon Sep 24, 2012 3:12... |
| FLOWS_FILES | 🕒🔒 | Mon Sep 24, 2012 3:30... | SYSAUX | TEMP | DEFAULT | Mon Sep 24, 2012 3:28... |
| GSMADMIN_INTERNAL | 🕒🔒 | Sat Mar 23, 2013 2:32:... | SYSAUX | TEMP | DEFAULT | Mon Sep 24, 2012 2:32... |
| GSMCATUSER | 🕒🔒 | Sat Mar 23, 2013 2:46:... | USERS | TEMP | DEFAULT | Mon Sep 24, 2012 2:46... |
| GSMUSER | 🕒🔒 | Sat Mar 23, 2013 2:32:... | USERS | TEMP | DEFAULT | Mon Sep 24, 2012 2:32... |
| HR | ✓ | Sat Apr 13, 2013 12:42... | USERS | TEMP | DEFAULT | Wed Oct 10, 2012 2:03... |

On the Users page of Enterprise Manager Database Express, you manage the database users who are allowed to access the current database. You use this page to create, delete, and modify the settings of a user.

To access the Users page, expand the Security menu and select Users.

# Creating a User



Create User

Name * HRDBA

Authentication * ● Password ○ External ○ Global

Password * •••••••••

Confirm Password * •••••••••

Profile DEFAULT

Password Expired □

Account Locked □

Create User

Default Tablespace USERS

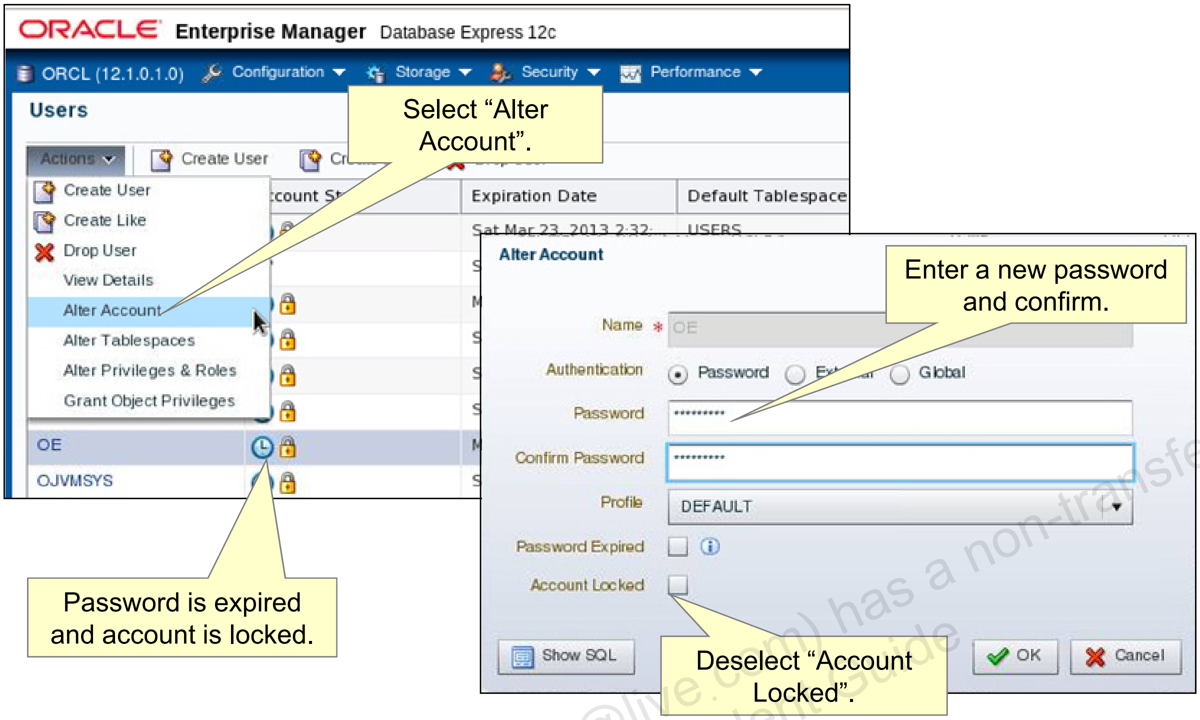
Temporary Tablespace TEMP

To create a database user by using Enterprise Manager Database Express, select Create User on the Users page.

Provide the required information. Mandatory items (such as Name) are marked with an asterisk (*). The name specified must follow the same rules as those used for creating database objects. The pages that follow in this lesson give you more information about authentication. Profiles are covered later in this lesson.

Assign a default tablespace and a temporary tablespace to each user. If users do not specify a tablespace when creating an object, the object will be created in the default tablespace assigned to the object owner. This enables you to control where their objects are created. If you do not choose a default tablespace, the system-defined default permanent tablespace is used. The case is similar for the temporary tablespace: if you do not specify a tablespace, the system-defined temporary tablespace is used.

# Unlocking a User Account and Resetting the Password



Alter Account

Name * OE

Authentication ● Password ○ External ○ Global

Password *********

Confirm Password *********

Profile DEFAULT

Password Expired □

Account Locked □

Select “Alter Account”.

Enter a new password and confirm.

Password is expired and account is locked.

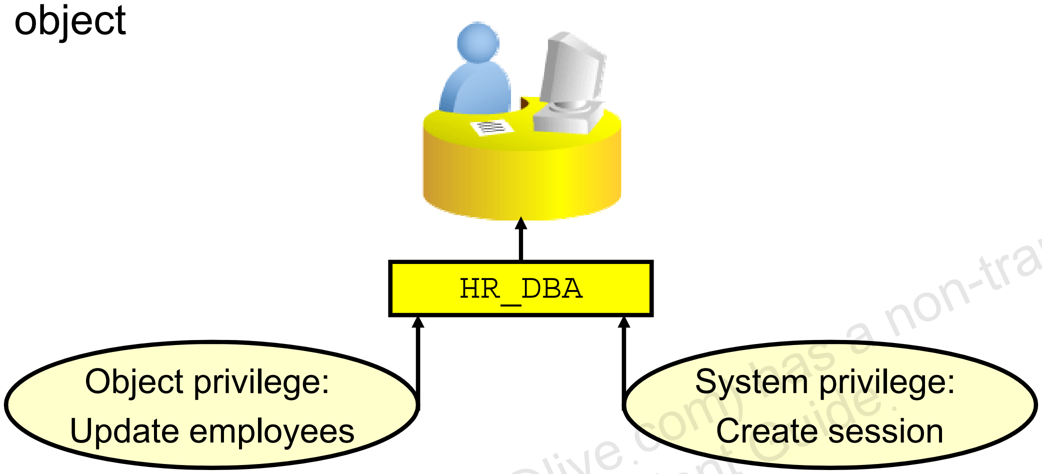
Deselect “Account Locked”.

During installation and database creation, you can unlock and reset many of the Oracle-supplied database user accounts. If you did not choose to unlock the user accounts at that time, you can unlock the user by selecting the user on the Users page, selecting Alter User from the Actions list, and deselecting Account Locked. If the password is expired, enter a new password in the Password and Confirm Password fields.

# 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



HR_DBA

Object privilege:
Update employees

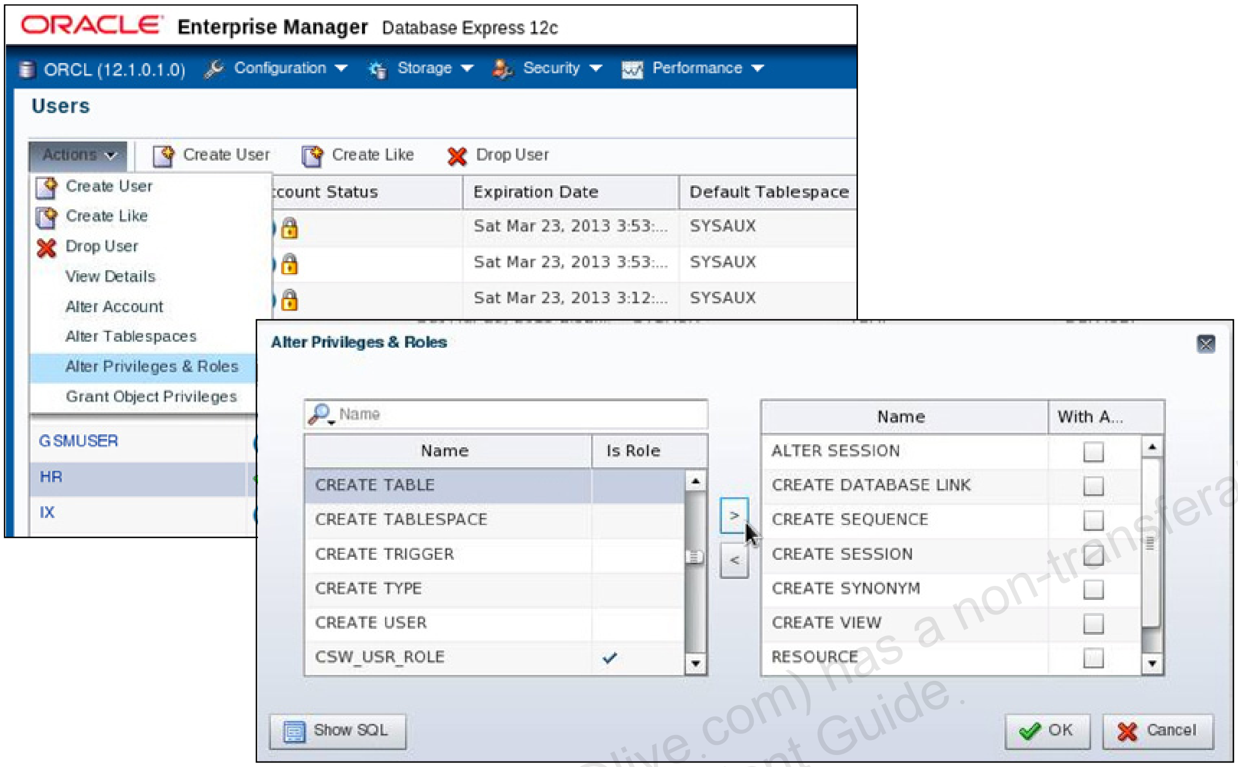
System privilege:
Create session

A *privilege* is a right to execute a particular type of SQL statement or to access another user's object.

Privileges are divided into two categories:

- **System privileges:** Each system privilege allows a user to perform a particular database operation or class of database operations. For example, the privilege to create tablespaces is a system privilege. System privileges can be granted by the administrator or by someone who has been given explicit permission to administer the privilege. There are more than 170 distinct system privileges. Many system privileges contain the `ANY` clause.
- **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.

# System Privileges



| :count Status | Expiration Date | Default Tablespace |
|---|---|---|
| 🔒 | Sat Mar 23, 2013 3:53:... | SYSAUX |
| 🔒 | Sat Mar 23, 2013 3:53:... | SYSAUX |
| 🔒 | Sat Mar 23, 2013 3:12:... | SYSAUX |

| Name | Is Role |
|---|---|
| CREATE TABLE | |
| CREATE TABLESPACE | |
| CREATE TRIGGER | |
| CREATE TYPE | |
| CREATE USER | |
| CSW_USR_ROLE | ✓ |

| Name | With A... |
|---|---|
| ALTER SESSION | □ |
| CREATE DATABASE LINK | □ |
| CREATE SEQUENCE | □ |
| CREATE SESSION | □ |
| CREATE SYNONYM | □ |
| CREATE VIEW | □ |
| RESOURCE | □ |

You can administer system privileges when you create a user or at a later time.

To grant or revoke system privileges, select the user and then select “Alter Privileges & Roles” on the Users page. Select the appropriate privileges from the list of privileges and move them by clicking the appropriate arrow.

Granting a privilege with the `ANY` clause means that the privilege crosses schema lines. For example, if you have the `CREATE TABLE` privilege, you can create a table—but only in your own schema. The `SELECT ANY TABLE` privilege allows you to select from tables owned by other users. The `SYS` user and users with the `DBA` role are granted all of the `ANY` privileges; they can therefore do anything to any data object. The scope of the `ANY` system privileges can be controlled using the Oracle Database Vault Option.

Select the With Admin check box to enable the user to administer the privilege and grant the system privilege to other users.

The SQL syntax for granting system privileges is:

```
GRANT <system_privilege> TO <grantee clause> [WITH ADMIN OPTION]
```

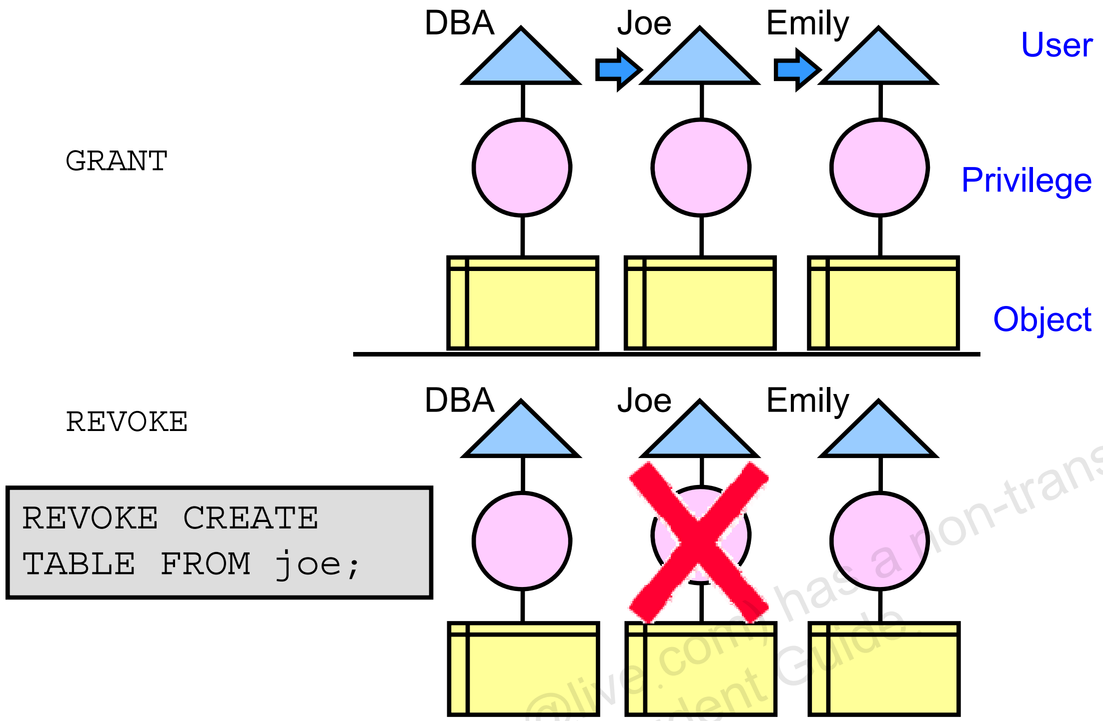
Carefully consider security requirements before granting system permissions. Some system privileges are usually granted only to administrators:

- **`RESTRICTED SESSION`:** This privilege allows you to log in even if the database has been opened in restricted mode.

- **SYSDBA and SYSOPER:** These privileges allow you to shut down, start up, and perform recovery and other administrative tasks in the database. SYSOPER allows a user to perform basic operational tasks, but without the ability to look at user data. It includes the following system privileges:
  - STARTUP and SHUTDOWN
  - CREATE SPFILE
  - ALTER DATABASE OPEN/MOUNT/BACKUP
  - 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.)
  - RESTRICTED SESSION

  The SYSDBA system privilege additionally authorizes incomplete recovery and the deletion of a database. Effectively, the SYSDBA system privilege allows a user to connect as the SYS user.
- **SYSASM:** This privilege allows you to start up, shut down, and administer an ASM instance.
- **DROP ANY *object*:** The DROP ANY privilege allows you to delete objects that other schema users own.
- **CREATE, MANAGE, DROP, and ALTER TABLESPACE:** These privileges allow for tablespace administration, including creating, dropping, and changing tablespace attributes.
- **CREATE LIBRARY:** The Oracle database allows developers to create and call external code (for example, a C library) from PL/SQL. The library must be named by a LIBRARY object in the database. The CREATE LIBRARY privilege allows a user to create an arbitrary code library that is executable from PL/SQL.
- **CREATE ANY DIRECTORY:** As a security measure, the operating system directory where the code resides must be linked to a virtual Oracle directory object. With the CREATE ANY DIRECTORY privilege, you can potentially call insecure code objects.
  The CREATE ANY DIRECTORY privilege allows a user to create a directory object (with read and write access) to any directory that the Oracle software owner can access. This means that the user can access external procedures in those directories. The user can attempt to directly read and write any database file, such as data files, redo log, and audit logs. Ensure that your organization has a security strategy that prevents misuse of powerful privileges such as this one.
- **GRANT ANY OBJECT PRIVILEGE:** This privilege allows you to grant object permissions on objects that you do not own.
- **ALTER DATABASE and ALTER SYSTEM:** These very powerful privileges allow you to modify the database and the Oracle instance (for example, renaming a data file or flushing the buffer cache).

# Revoking System Privileges with ADMIN OPTION



```
REVOKE CREATE
TABLE FROM joe;
```

System privileges that have been granted directly with a `GRANT` command can be revoked by using the `REVOKE` SQL statement. Users with `ADMIN OPTION` for a system privilege can revoke the privilege from any other database user. The revoker does not have to be the same user who originally granted the privilege.

There are no cascading effects when a system privilege is revoked, regardless of whether it is given the `ADMIN OPTION`.

The SQL syntax for revoking system privileges is:

```
        REVOKE <system_privilege> FROM <grantee clause>
```

The slide illustrates the following situation.

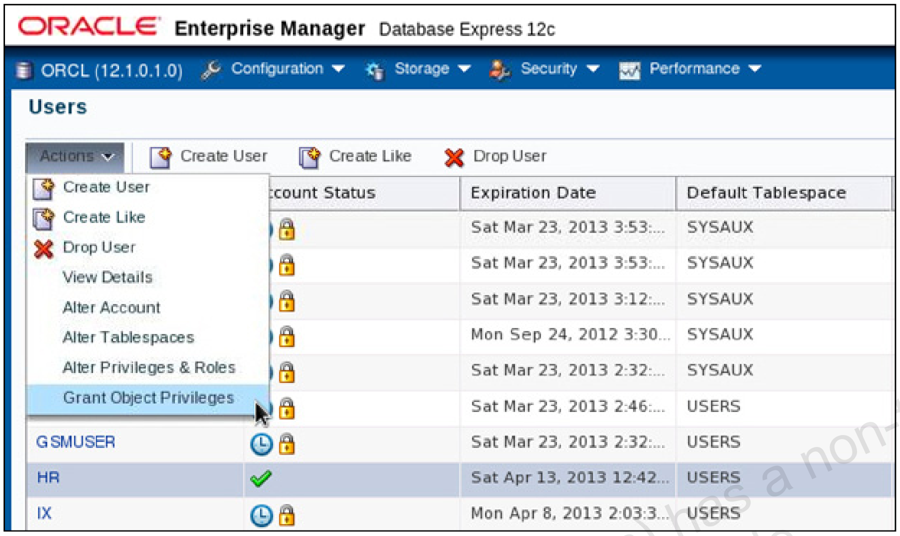
**Scenario**

1. The DBA grants the `CREATE TABLE` system privilege to Joe with `ADMIN OPTION`.
2. Joe creates a table.
3. Joe grants the `CREATE TABLE` system privilege to Emily.
4. Emily creates a table.
5. The DBA revokes the `CREATE TABLE` system privilege from Joe.

**Result**

Joe's table still exists, but Joe cannot create new tables. Emily's table still exists, and she still has the `CREATE TABLE` system privilege.

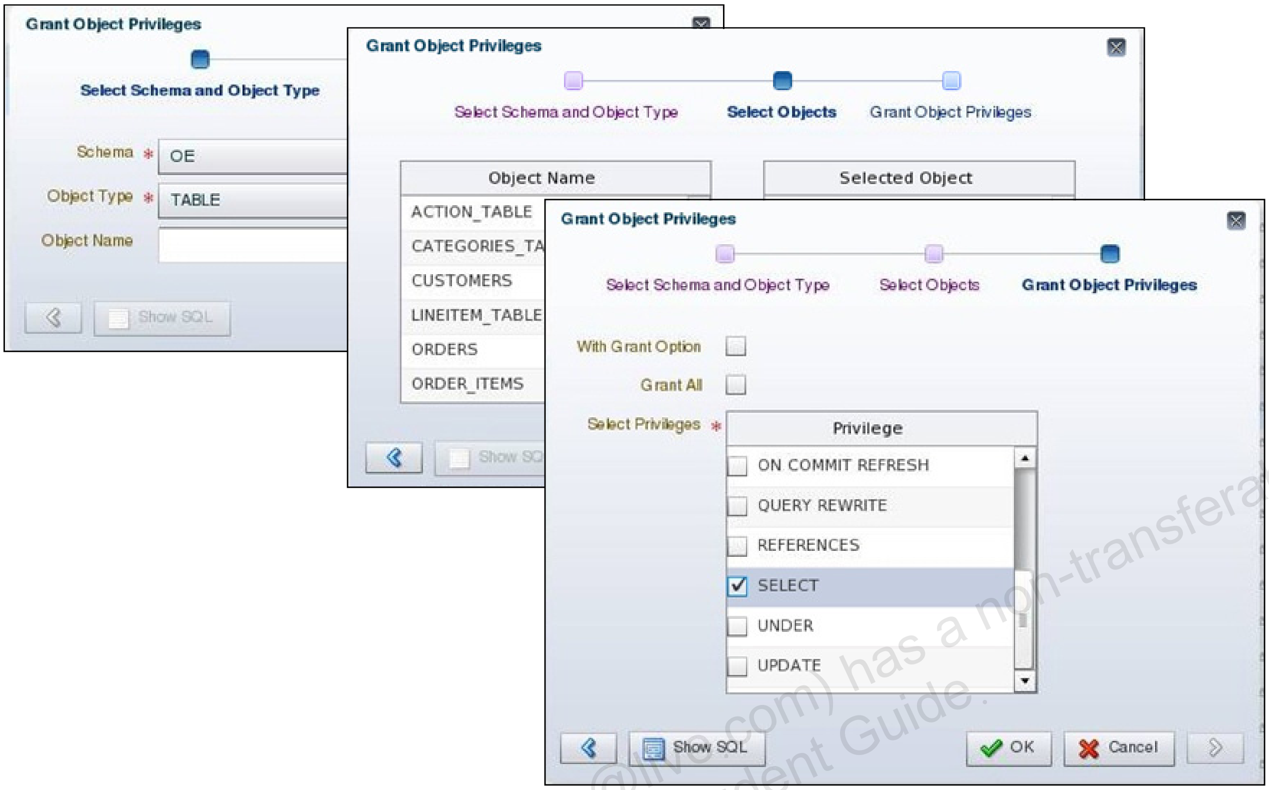
# Granting Object Privileges



| | :count Status | Expiration Date | Default Tablespace |
|---|---|---|---|
| | 🔒 | Sat Mar 23, 2013 3:53:... | SYSAUX |
| | 🔒 | Sat Mar 23, 2013 3:53:... | SYSAUX |
| | 🔒 | Sat Mar 23, 2013 3:12:... | SYSAUX |
| | 🔒 | Mon Sep 24, 2012 3:30... | SYSAUX |
| | 🔒 | Sat Mar 23, 2013 2:32:... | SYSAUX |
| | 🔒 | Sat Mar 23, 2013 2:46:... | USERS |
| GSMUSER | 🕒 🔒 | Sat Mar 23, 2013 2:32:... | USERS |
| HR | ✓ | Sat Apr 13, 2013 12:42... | USERS |
| IX | 🕒 🔒 | Mon Apr 8, 2013 2:03:3... | USERS |

To grant object privileges, select the user on the Users page. Select “Grant Object Privileges” in the Actions menu.

# Object Privileges



Schema * OE
Object Type * TABLE
Object Name

| Object Name | Selected Object |
|---|---|
| ACTION_TABLE | |
| CATEGORIES_TA | |
| CUSTOMERS | |
| LINEITEM_TABLE | |
| ORDERS | |
| ORDER_ITEMS | |

With Grant Option □
Grant All □
Select Privileges *

| Privilege |
|---|
| □ ON COMMIT REFRESH |
| □ QUERY REWRITE |
| □ REFERENCES |
| ☑ SELECT |
| □ UNDER |
| □ UPDATE |

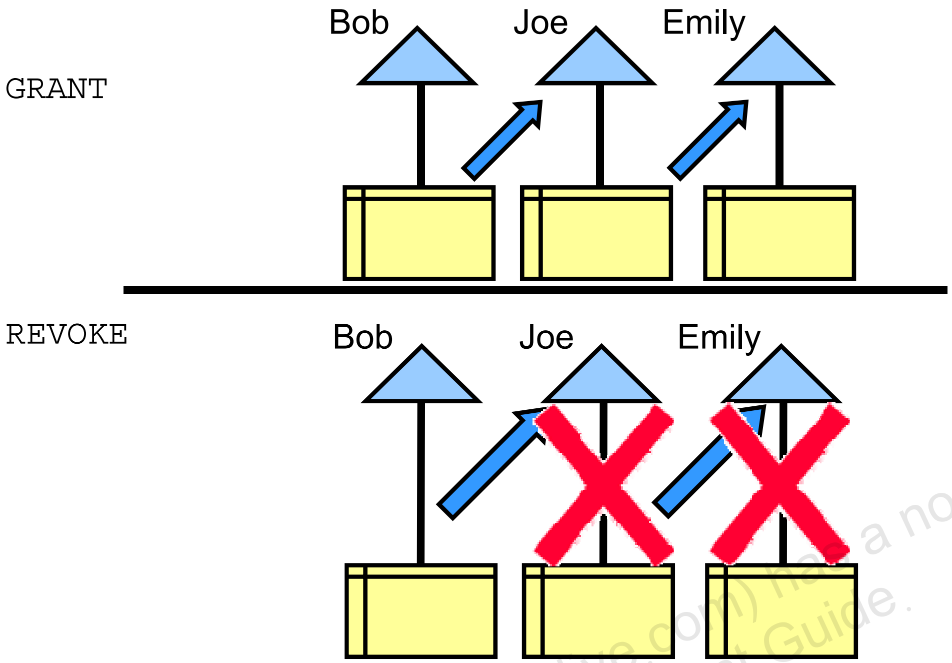
Perform the following steps to grant object privileges by using Enterprise Manager Database Express:

1. Specify the schema name for the object and select the type of object on which you want to grant privileges. Click the arrow to proceed.
2. Select the objects from the Object Name list and move them to the Selected Object list. When you have selected all objects, click the arrow to move to the next page.
3. Then select the appropriate privileges from the Privileges list. Select the “With Grant Option” check box if this user is allowed to grant other users the same access.

The SQL syntax for granting object privileges is:

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

# Revoking Object Privileges with GRANT OPTION



Cascading effects can be observed when revoking a system privilege that is related to a data manipulation language (DML) operation. For example, if the SELECT ANY TABLE privilege is granted to a user, and if that user has created procedures that use the table, all procedures that are contained in the user's schema must be recompiled before they can be used again.

Revoking object privileges also cascades when given with GRANT OPTION. As a user, you can revoke only those privileges that you have granted. For example, Bob cannot revoke the object privilege that Joe granted to Emily. Only the grantee or a user with the privilege called GRANT ANY OBJECT PRIVILEGE can revoke object privileges.

**Scenario**

1. Joe is granted the SELECT object privilege on EMPLOYEES with GRANT OPTION.
2. Joe grants the SELECT privilege on EMPLOYEES to Emily.
3. The SELECT privilege is revoked from Joe. This revoke is cascaded to Emily as well.

# 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

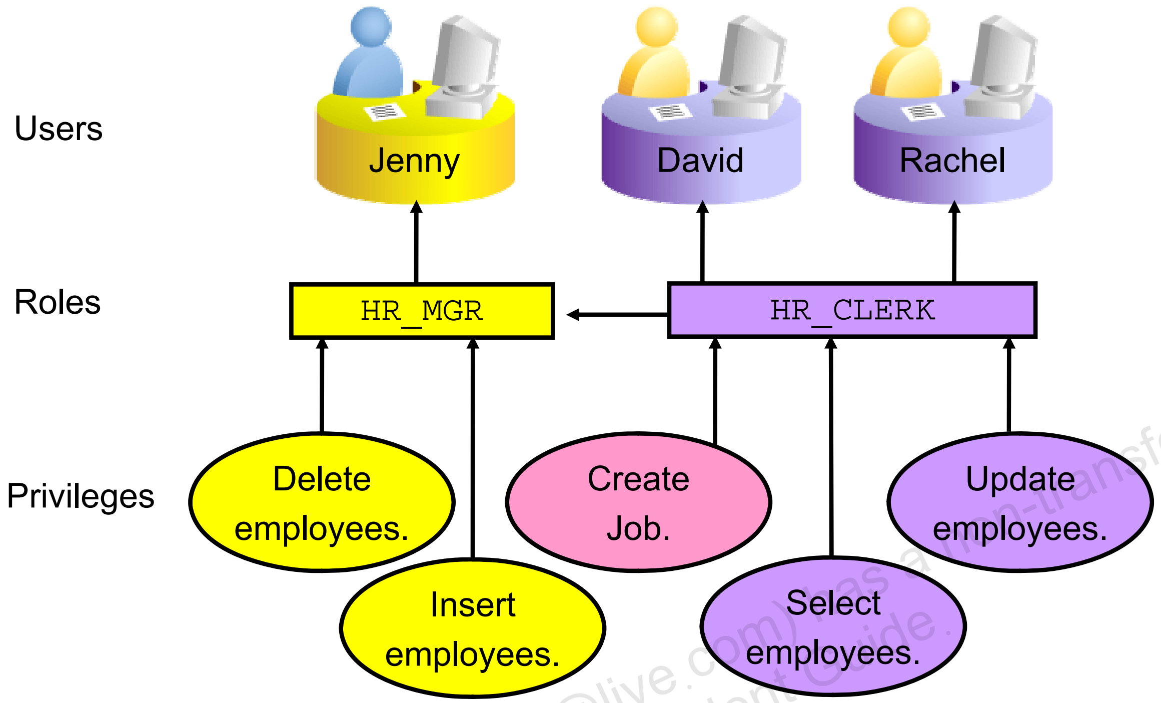
A role is a named group of related privileges that are granted to users or to other roles.

You can use roles to administer database privileges. You can add privileges to a role and grant the role to a user. The user can then enable the role and exercise the privileges granted by the role. A role contains all privileges that are granted to that role and all privileges of other roles that are granted to it.

Roles provide the following benefits with respect to managing privileges:

- **Easier privilege management:** Use roles to simplify privilege management. Rather than granting the same set of privileges to several users, you can grant the privileges to a role and then grant that role to each user.
- **Dynamic privilege management:** If the privileges associated with a role are modified, all users who are granted the role acquire the modified privileges automatically and immediately.
- **Selective availability of privileges:** Roles can be enabled and disabled to turn privileges on and off temporarily. This allows the privileges of the user to be controlled in a given situation.

# Assigning Privileges to Roles and Assigning Roles to Users

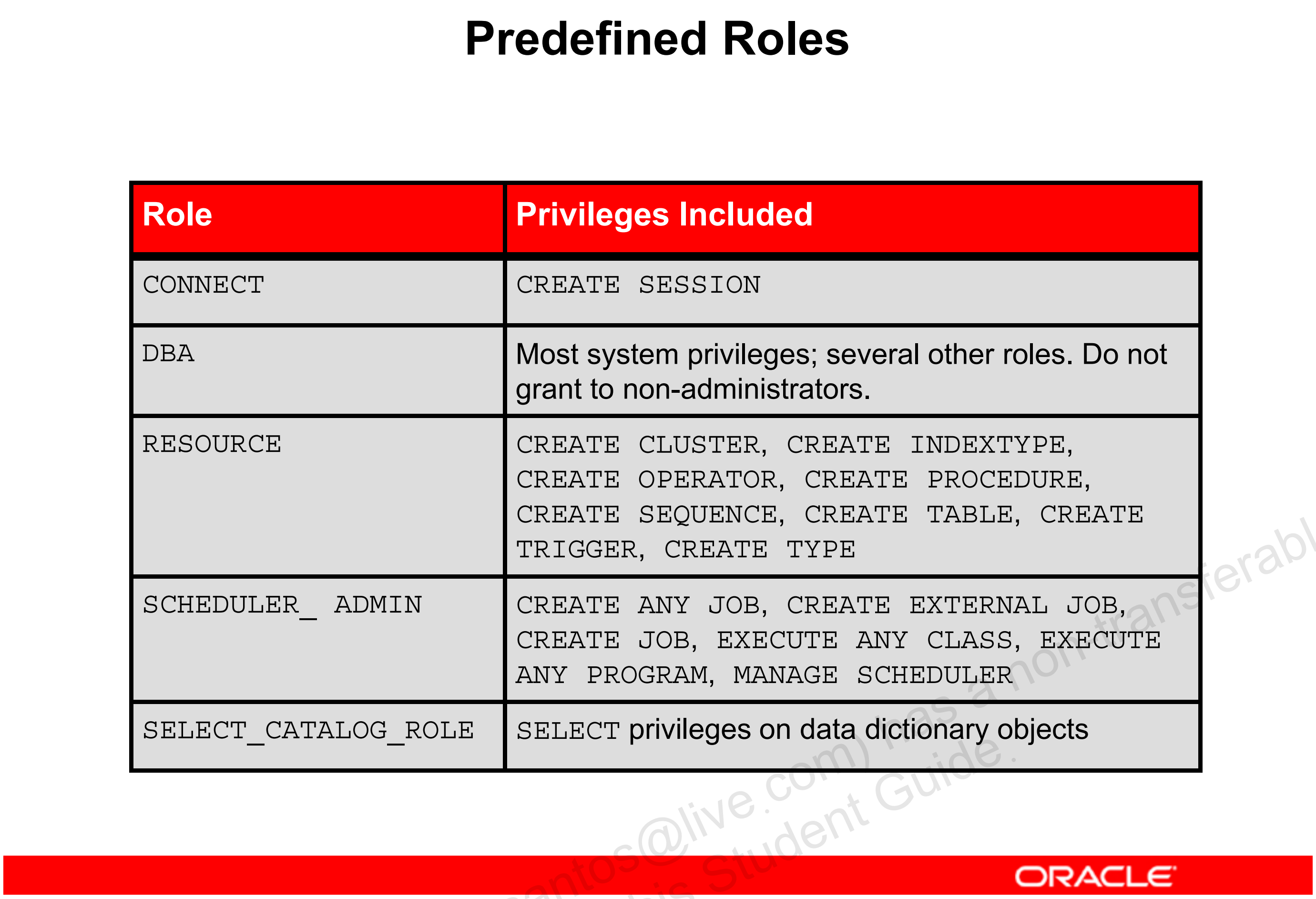


In most systems, it is time-consuming and error-prone to grant necessary privileges to each user individually. Oracle software provides for easy and controlled privilege management through roles. Roles are named groups of related privileges that are granted to users or to other roles. Roles are designed to ease the administration of privileges in the database and, therefore, improve security.

**Role characteristics**

- Privileges are granted to and revoked from roles as though the role were a user.
- Roles are granted to and revoked from users or other roles as though they were system privileges.
- A role can consist of both system and object privileges.
- A role can be enabled or disabled for each user who is granted the role.
- A role can require a password to be enabled.
- Roles are not owned by anyone, and they are not in any schema.

In the slide example, the `SELECT` and `UPDATE` privileges on the `employees` table *and* the `CREATE JOB` system privilege are granted to the `HR_CLERK` role. `DELETE` and `INSERT` privileges on the `employees` table *and* the `HR_CLERK` role are granted to the `HR_MGR` role. The manager is granted the `HR_MGR` role and can now select, delete, insert, and update the `employees` table.



# Predefined Roles

| Role | Privileges Included |
|---|---|
| CONNECT | CREATE SESSION |
| DBA | Most system privileges; several other roles. Do not grant to non-administrators. |
| 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 |

There are several roles that are defined automatically for Oracle databases when you run database creation scripts. CONNECT is granted automatically to any user that is created with Enterprise Manager.

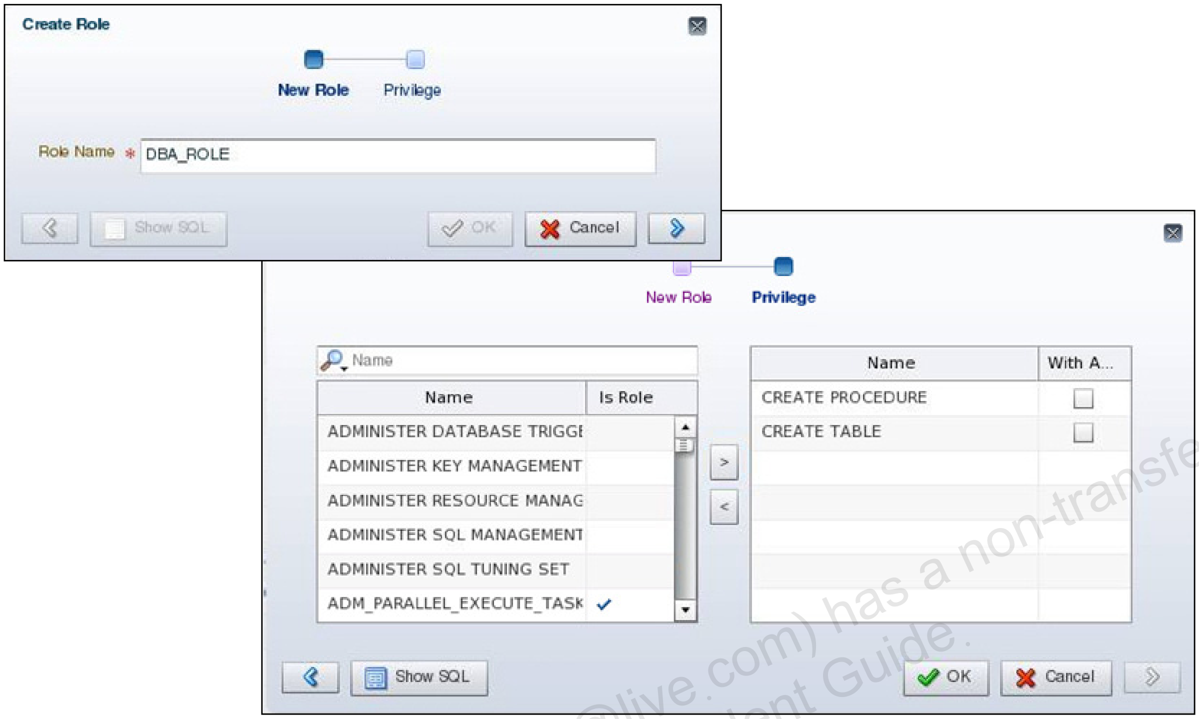
The SELECT ANY DICTIONARY system privilege does not permit access to sensitive data dictionary tables, which are owned by the SYS schema.

Refer to the *Oracle Database Security Guide* for a complete list of predefined roles.

Other roles that authorize you to administer special functions are created when that functionality is installed. For example, XDBADMIN contains the privileges required to administer the Extensible Markup Language (XML) database if that feature is installed. AQ_ADMINISTRATOR_ROLE provides privileges to administer advanced queuing. HS_ADMIN_ROLE includes the privileges needed to administer heterogeneous services.

You must not alter the privileges granted to these functional roles without the assistance of Oracle Support because you may inadvertently disable the needed functionality.

# Creating a Role



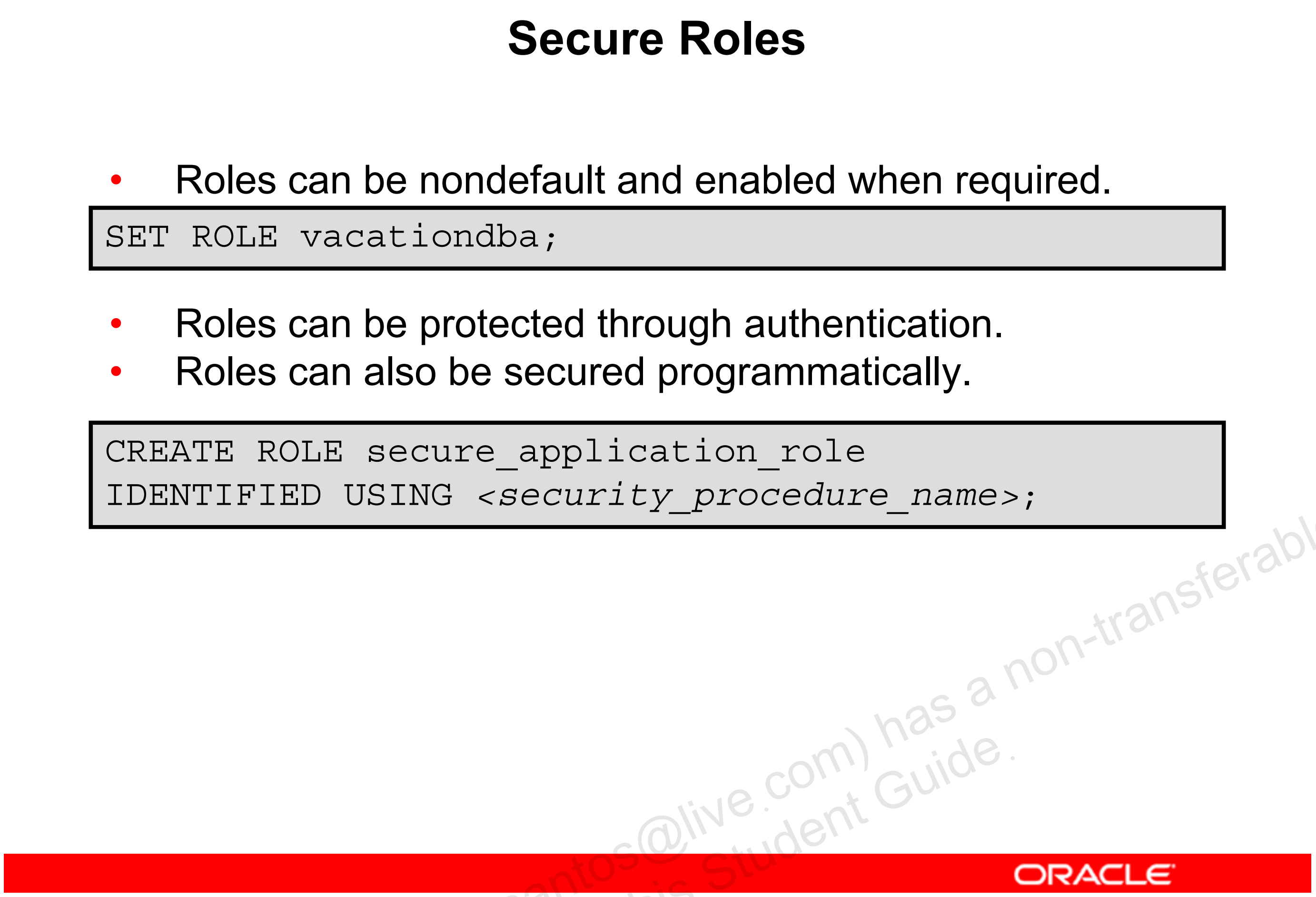
Role Name * DBA_ROLE

| Name | Is Role |
|---|---|
| ADMINISTER DATABASE TRIGGE | |
| ADMINISTER KEY MANAGEMENT | |
| ADMINISTER RESOURCE MANAG | |
| ADMINISTER SQL MANAGEMENT | |
| ADMINISTER SQL TUNING SET | |
| ADM_PARALLEL_EXECUTE_TASK | ✓ |

| Name | With A... |
|---|---|
| CREATE PROCEDURE | □ |
| CREATE TABLE | □ |

To create a role by using Enterprise Manager Database Express, perform the following steps:

1. Navigate to the Roles page and click Create Role.
2. Enter a name for the role.
3. Optionally add the system privileges and other roles as required. The role can be edited at a later time to modify these settings if necessary. You can also add object privileges to a role after it is created.
4. Click OK to create the role.



## Secure Roles

- Roles can be nondefault and enabled when required.

```
SET ROLE vacationdba;
```

- Roles can be protected through authentication.
- Roles can also be secured programmatically.

```
CREATE ROLE secure_application_role
IDENTIFIED USING <security_procedure_name>;
```

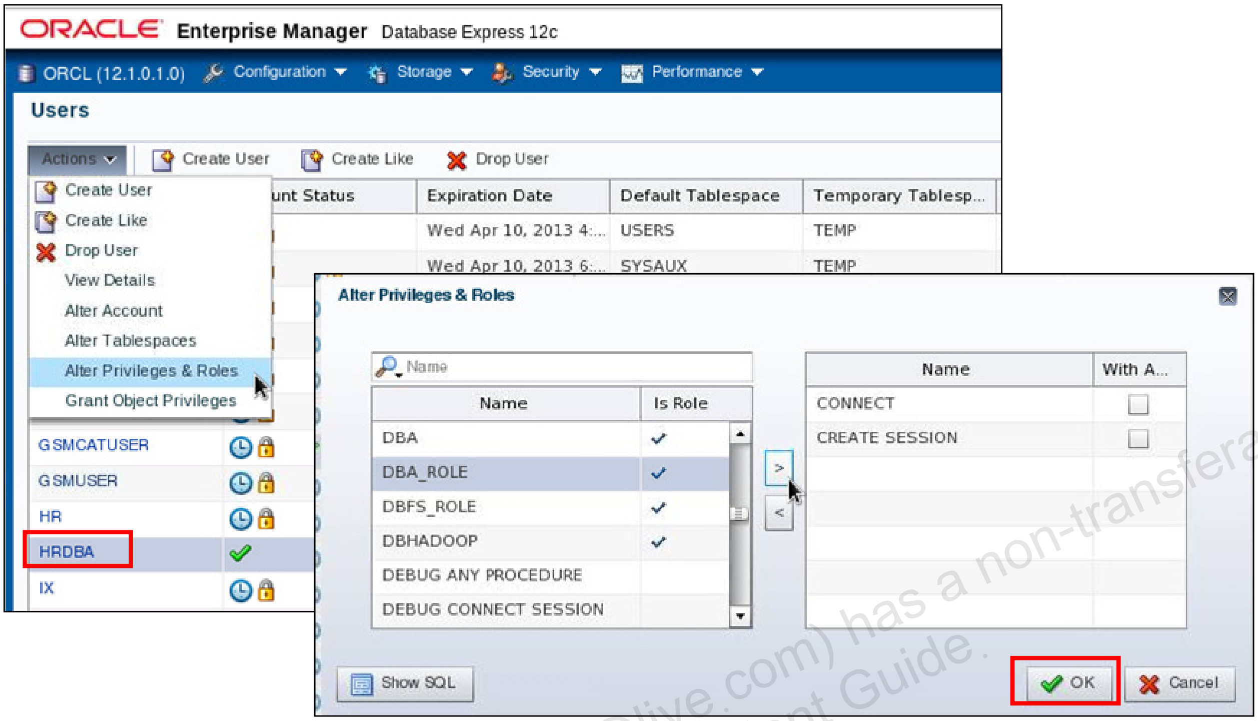
Roles are usually enabled by default, which means that if a role is granted to a user, then that user can exercise the privileges given to the role. Default roles are assigned to the user at connect time.

It is possible to:

- Make a role nondefault. The user must now explicitly enable the role before the role's privileges can be exercised.
- Have a role require additional authentication by using the `IDENTIFIED` clause to indicate that a user must be authorized by a specified method before the role is enabled with the `SET ROLE` statement. The default authentication for a role is None.
- Create secure application roles that can be enabled only by executing a PL/SQL procedure successfully. The PL/SQL procedure can check things such as the user's network address, the program that the user is running, the time of day, and other elements needed to properly secure a group of permissions.
- Administer roles easily using the Oracle Database Vault option. Secure application roles are simplified, and traditional roles can be further restricted.

**Note:** Role authentication can be defined by using Enterprise Manager Cloud Control, but not in Enterprise Manager Database Express.

# Assigning Roles to Users



| ...unt Status | Expiration Date | Default Tablespace | Temporary Tablesp... |
|---|---|---|---|
| | Wed Apr 10, 2013 4:... | USERS | TEMP |
| | Wed Apr 10, 2013 6:... | SYSAUX | TEMP |

| Name | Is Role |
|---|---|
| DBA | ✓ |
| DBA_ROLE | ✓ |
| DBFS_ROLE | ✓ |
| DBHADOOP | ✓ |
| DEBUG ANY PROCEDURE | |
| DEBUG CONNECT SESSION | |

| Name | With A... |
|---|---|
| CONNECT | □ |
| CREATE SESSION | □ |

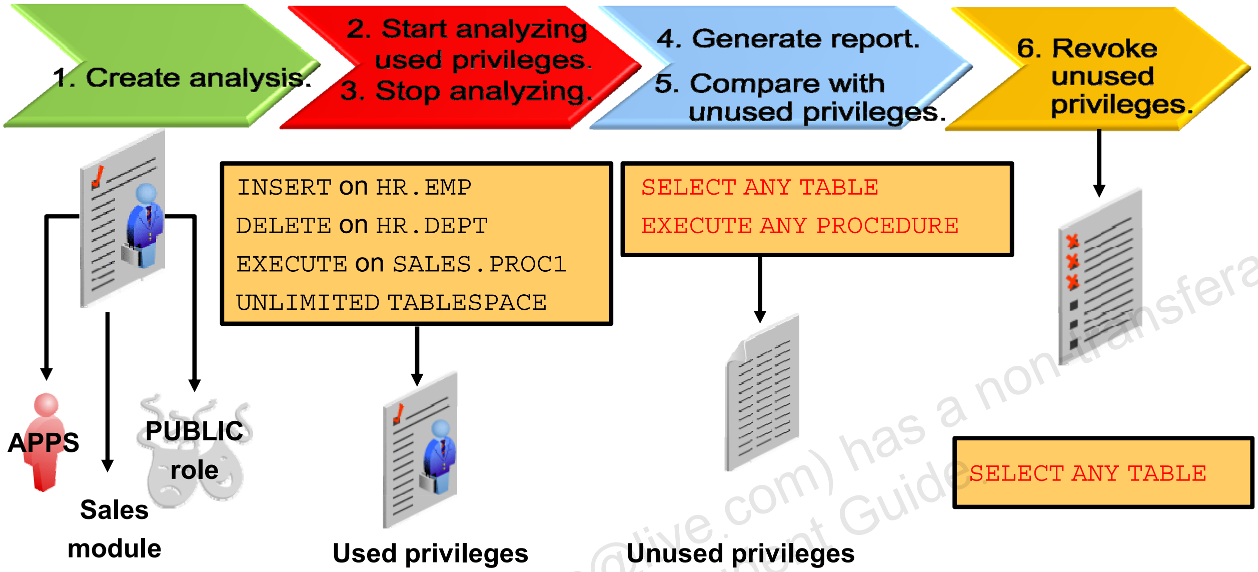
By default, Enterprise Manager Database Express automatically grants the `CONNECT` role to new users. This allows users to connect to the database and create database objects in their own schemas.

To assign a role to a user by using Enterprise Manager Database Express:

1. Navigate to the Users page.
2. Select the user and select “Alter Privileges & Roles” in the Actions menu.
3. Select the desired role in the list on the left and move it by using the arrow.
4. When you have assigned all appropriate roles, click OK.

# Privilege Analysis

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



INSERT on HR.EMP
DELETE on HR.DEPT
EXECUTE on SALES.PROC1
UNLIMITED TABLESPACE

SELECT ANY TABLE
EXECUTE ANY PROCEDURE

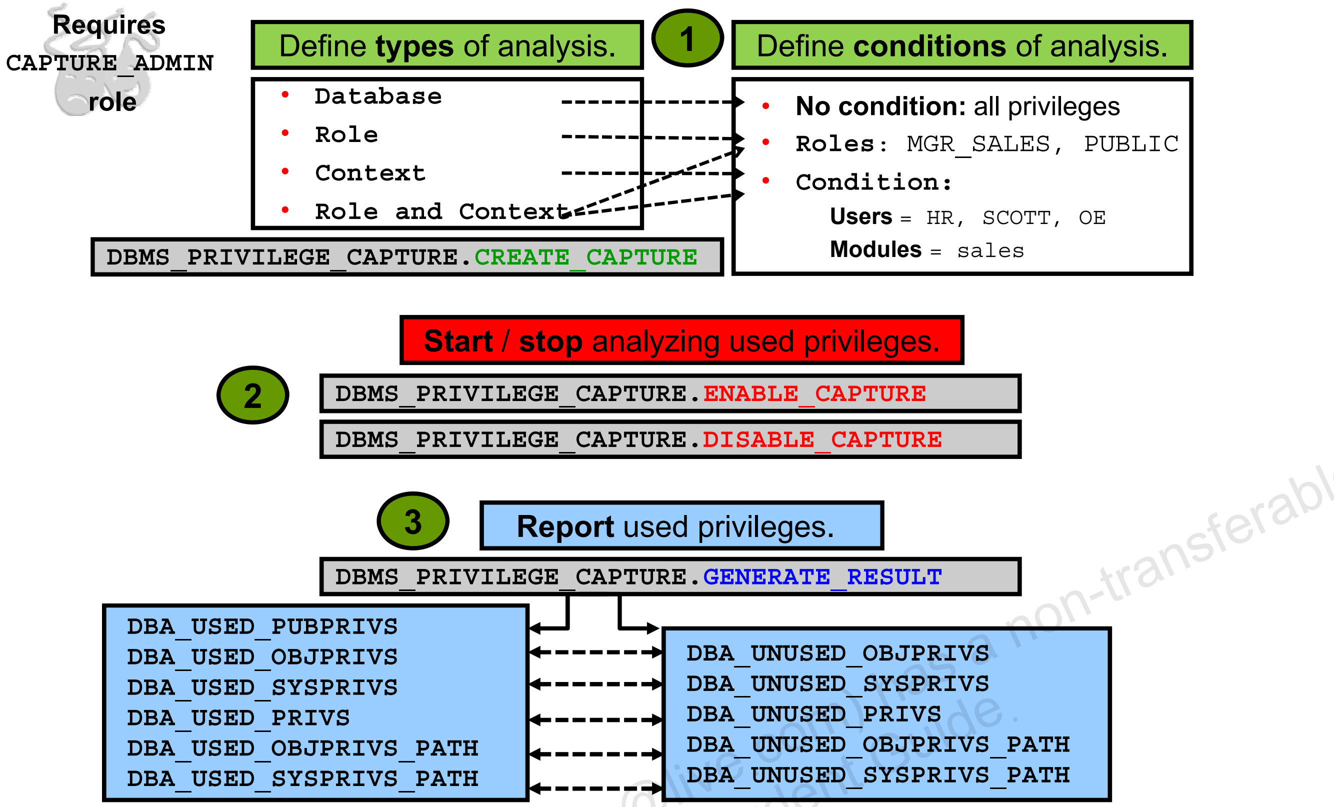
SELECT ANY TABLE

A major concern in many databases is that existing database and application users have excessive privileges. Excessive privileges violate the principle of least privilege. To achieve the least privilege principle, unused privileges need to be identified.

Oracle Database 12*c* provides a package named DBMS_PRIVILEGE_CAPTURE to analyze used privileges.

You can use a privilege analysis policy to identify object and system privileges used to run an application module or to execute certain SQL statements or privileges used by defined roles. You can generate reports of used and unused privileges during the analysis period. The report helps the security officer revoke unnecessary privileges by comparing the used and unused granted privilege lists.

# Privilege Analysis Flow



Requires CAPTURE_ADMIN role

1 Define **types** of analysis.

- Database
- Role
- Context
- Role and Context

DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE

Define **conditions** of analysis.

- **No condition:** all privileges
- Roles: MGR_SALES, PUBLIC
- Condition:
  **Users** = HR, SCOTT, OE
  **Modules** = sales

**Start** / **stop** analyzing used privileges.

2 DBMS_PRIVILEGE_CAPTURE.ENABLE_CAPTURE
DBMS_PRIVILEGE_CAPTURE.DISABLE_CAPTURE

3 **Report** used privileges.

DBMS_PRIVILEGE_CAPTURE.GENERATE_RESULT

DBA_USED_PUBPRIVS
DBA_USED_OBJPRIVS
DBA_USED_SYSPRIVS
DBA_USED_PRIVS
DBA_USED_OBJPRIVS_PATH
DBA_USED_SYSPRIVS_PATH

DBA_UNUSED_OBJPRIVS
DBA_UNUSED_SYSPRIVS
DBA_UNUSED_PRIVS
DBA_UNUSED_OBJPRIVS_PATH
DBA_UNUSED_SYSPRIVS_PATH

When creating an analysis, you first define the targeted objects to be analyzed in used privileges. You do that by setting the type of analysis:

- **Database analysis:** If no condition is given, it analyzes the used privileges (except privileges used by administrative users) within the whole database.
- **Role analysis:** If roles are defined, it analyzes the privileges exercised through any given role. For example, if you create a privilege analysis policy to analyze on PUBLIC, the privileges that are directly and indirectly granted to PUBLIC are analyzed when they are used.
- **Context-specific analysis:** If the contexts are defined, it analyzes the privileges that are used through a given application module or specified contexts.

Different conditions can be combined with “AND” and/or “OR” Boolean operators.

Because the created policy is not enabled by default, your next step is to enable the policy to start analyzing used privileges. After a certain time, you stop analyzing.

Your third step is to generate a report. Reporting includes two types of results:

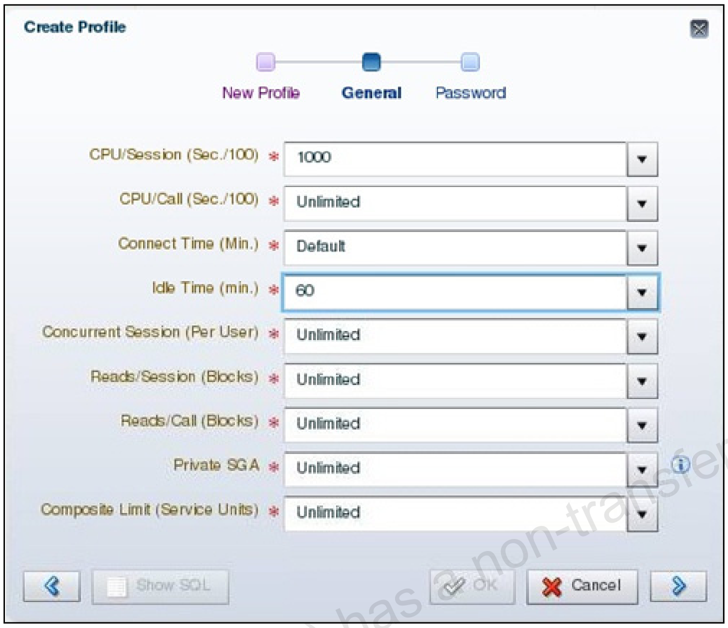
- Used privileges visible in DBA_USED_*xxx* and DBA_USED_*xxx*_PATH views
- Unused privileges visible in DBA_UNUSED_*xxx* and DBA_UNUSED_*xxx*_PATH views

# Profiles and Users

Users are assigned only one profile at a time.

Profiles:

- Control resource consumption
- Manage account status and password expiration



CPU/Session (Sec./100) * 1000

CPU/Call (Sec./100) * Unlimited

Connect Time (Min.) * Default

Idle Time (min.) * 60

Concurrent Session (Per User) * Unlimited

Reads/Session (Blocks) * Unlimited

Reads/Call (Blocks) * Unlimited

Private SGA * Unlimited

Composite Limit (Service Units) * Unlimited

**Note:** RESOURCE_LIMIT must be set to TRUE before profiles can impose resource limitations.

Profiles impose a named set of resource limits on database usage and instance resources. Profiles also manage the account status and place limitations on users' passwords (length, expiration time, and so on). Every user is assigned a profile and may belong to only one profile at any given time. If users have already logged in when you change their profile, the change does not take effect until their next login.

The DEFAULT profile serves as the basis for all other profiles. As illustrated in the slide, limitations for a profile can be implicitly specified (as in CPU/Session), can be unlimited (as in CPU/Call), or can reference whatever setting is in the DEFAULT profile (as in Connect Time).

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. Profile password settings are always enforced.

Profiles enable an administrator to control the following system resources:

- **CPU:** CPU resources may be limited on a per-session or per-call basis. A CPU/Session limitation of 1,000 means that if any individual session that uses this profile consumes more than 10 seconds of CPU time (CPU time limitations are in hundredths of a second), that session receives an error and is logged off:

```
ORA-02392: exceeded session limit on CPU usage, you are being
logged off
```

A per-call limitation does the same thing, but instead of limiting the user's overall session, it prevents any single command from consuming too much CPU. If CPU/Call is limited and the user exceeds the limitation, the command aborts. The user receives an error message such as the following:

```
ORA-02393: exceeded call limit on CPU usage
```

- **Network/Memory:** Each database session consumes system memory resources and (if the session is from a user who is not local to the server) network resources. You can specify the following:
  - **Connect Time:** Indicates for how many minutes a user can be connected before being automatically logged off
  - **Idle Time:** Indicates for how many minutes a user's session can remain idle before being automatically logged off. Idle time is calculated for the server process only. It does not take into account application activity. The `IDLE_TIME` limit is not affected by long-running queries and other operations.
  - **Concurrent Sessions:** Indicates how many concurrent sessions can be created by using a database user account
  - **Private SGA:** Limits the amount of space consumed in the System Global Area (SGA) for sorting, merging bitmaps, and so on. This restriction takes effect only if the session uses a shared server. (Shared servers are covered in the lesson titled "Configuring the Oracle Network Environment.")
- **Disk I/O:** This limits the amount of data a user can read at the per-session level or per-call level. Reads/Session and Reads/Call place a limitation on the total number of reads from both memory and the disk. This can be done to ensure that no I/O-intensive statements overuse memory and disks.

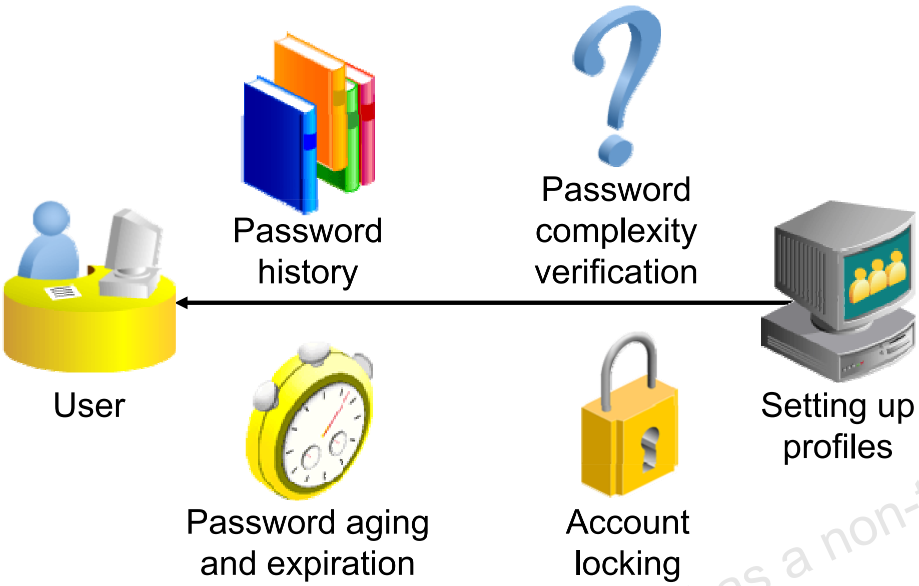
Profiles also allow a composite limit. Composite limits are based on a weighted combination of CPU/Session, Reads/Session, Connect Time, and Private SGA. Composite limits are discussed in more detail in the *Oracle Database Security Guide*.

To create a profile by using Enterprise Manager Database Express:

1. Select Profiles in the Security menu.
2. Select Create Profile.
3. Enter the profile name and then click the arrow to continue.
4. On the General page, enter the values for the resources you want to define. Click the arrow to the continue.
5. On the Password page, enter the appropriate values.
6. Click OK to create the profile.

**Note:** Resource Manager is an alternative to many of the profile settings. For more details about Resource Manager, see the *Oracle Database Administrator's Guide*.

# Implementing Password Security Features


Password history
Password complexity verification
User
Setting up profiles
Password aging and expiration
Account locking

**Note:** Do not use profiles that cause the SYS, SYSMAN, and DBSNMP passwords to expire and the accounts to be locked.

Oracle password management is implemented with user profiles. Profiles can provide many standard security features.

**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

- **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

**Password aging and expiration:** Enables user passwords to have a lifetime, after which the passwords expire and must be changed

- **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

**Note:** Expiring passwords and locking the SYS, SYSMAN, and DBSNMP accounts prevent Enterprise Manager from functioning properly. The applications must catch the "password expired" warning message and handle the password change; otherwise, the grace period expires and the user is locked out without knowing the reason.

**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. These checks can be implemented by using one of the following:

- **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

Recall that the values of the profile parameters are either set or inherited from the DEFAULT profile.

If both password history parameters have a value of UNLIMITED, Oracle Database ignores both. The user can reuse any password at any time, which is not a good security practice.

If both parameters are set, password reuse is allowed—but only after meeting both conditions. The user must have changed the password the specified number of times, and the specified number of days must have passed since the old password was last used.

For example, the profile of user ALFRED has PASSWORD_REUSE_MAX set to 10 and PASSWORD_REUSE_TIME set to 30. User ALFRED cannot reuse a password until he has reset the password 10 times and until 30 days have passed since the password was last used.

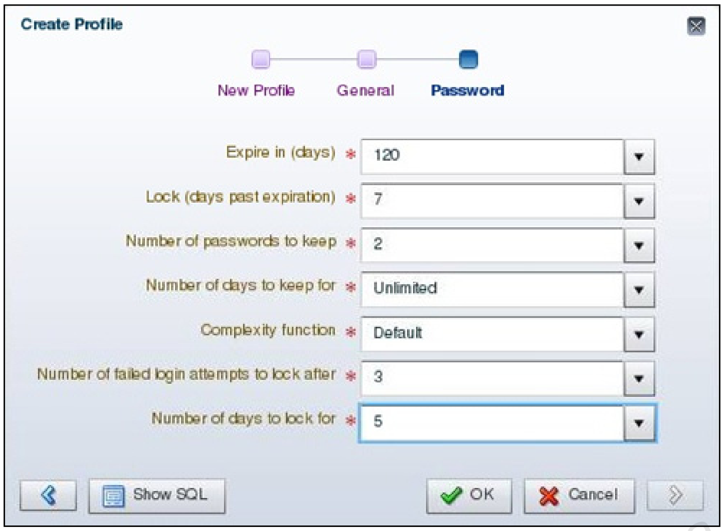
If one parameter is set to a number and the other parameter is specified as UNLIMITED, then the user can never reuse a password.

**Password complexity verification:** Makes a complexity check on the password to verify that it meets certain rules. The check must ensure that the password is complex enough to provide protection against intruders who may try to break into the system by guessing the password.

The PASSWORD_VERIFY_FUNCTION parameter names a PL/SQL function that performs a password complexity check before a password is assigned. Password verification functions must be owned by the SYS user and must return a Boolean value (TRUE or FALSE). A model password verification function is provided in the utlpwdmg.sql script found in the following directories:

- UNIX and Linux platforms: $ORACLE_HOME/rdbms/admin
- Windows platforms: %ORACLE_HOME%\rdbms\admin

# Creating a Password Profile



Create Profile

Expire in (days) * 120

Lock (days past expiration) * 7

Number of passwords to keep * 2

Number of days to keep for * Unlimited

Complexity function * Default

Number of failed login attempts to lock after * 3

Number of days to lock for * 5

You can choose common values for each of the settings from a list of values (click the flashlight icon to browse), or you can enter a custom value.

All time periods are expressed in days but can also be expressed as fractions. There are 1,440 minutes in a day; 5/1,440 is therefore five minutes.

Enterprise Manager can also be used to edit existing password profiles.

If the `utlpwdmg.sql` script has been executed, the `VERIFY_FUNCTION_11g`, `ORA12C_VERIFY_FUNCTION`, and `ORA12C_STRONG_VERIFY_FUNCTION` functions are available. Additional information about these functions is provided later in the lesson. If you have created your own complexity function, the name of that function may be entered. The function name does not appear in the Select list. If the function produces runtime errors, the user is unable to change the password.

**Dropping a Password Profile**

In Enterprise Manager, you cannot drop a profile that is used by users. However, if you drop a profile with the `CASCADE` option (for example, in SQL*Plus), all users who have that profile are automatically assigned the `DEFAULT` profile.

# Supplied Password Verification Functions

- The following functions are created by the `$ORACLE_HOME/rdbms/admin/utlpwdmg.sql` script:
  - `VERIFY_FUNCTION_11`*`g`*
  - `ORA12C_VERIFY_FUNCTION`
  - `ORA12C_STRONG_VERIFY_FUNCTION`
- The functions require the following of passwords:
  - Have a minimum number of characters
  - Not be the username, username with a number, or username reversed
  - Not be the database name or the database name with a number
  - Have at least one alphabetic and one numeric character
  - Differ from the previous password by at least three letters

You can create the following password verification functions by executing the `utlpwdmg.sql` script:

- `ORA12C_VERIFY_FUNCTION`: This function makes the minimum complexity checks such as checking for a minimum password length and that the password is not the same as the username.
- `ORA12C_STRONG_VERIFY_FUNCTION`: This function provides a stronger password complexity function that takes into consideration recommendations from the US Department of Defense Database Security Technical Implementation Guide.
- `VERIFY_FUNCTION_11`*`g`*: This function makes minimum complexity checks such as checking for a minimum password length and that the password is not the same as the username. This function was provided with Oracle Database 11*g*.

**Note:** The functions must be owned by the `SYS` user. Password complexity checking is not enforced for the `SYS` user.

Refer to the *Oracle Database Security Guide* for a detailed description of each of the functions.

# Modifications to the Default Profile

The `utlpwdmg.sql` script also modifies the `DEFAULT` profile as follows:

```
ALTER PROFILE DEFAULT LIMIT
PASSWORD_LIFE_TIME 180
PASSWORD_GRACE_TIME 7
PASSWORD_REUSE_TIME UNLIMITED
PASSWORD_REUSE_MAX  UNLIMITED
FAILED_LOGIN_ATTEMPTS 10
PASSWORD_LOCK_TIME 1
PASSWORD_VERIFY_FUNCTION
  ora12c_verify_function;
```

In addition to creating the functions as shown on the previous page, the `utlpwdmg` script also changes the `DEFAULT` profile with the following `ALTER PROFILE` command:

```
ALTER PROFILE default LIMIT
PASSWORD_LIFE_TIME 180
PASSWORD_GRACE_TIME 7
PASSWORD_REUSE_TIME UNLIMITED
PASSWORD_REUSE_MAX UNLIMITED
FAILED_LOGIN_ATTEMPTS 10
PASSWORD_LOCK_TIME 1
PASSWORD_VERIFY_FUNCTION ora12c_verify_function;
```

Remember that when users are created, they are assigned the `DEFAULT` profile unless another profile is specified.

# Assigning Quotas to Users

- Users who do not have the UNLIMITED TABLESPACE system privilege must be given a quota before they can create objects in a tablespace.
- Quotas can be:
  - A specific value in megabytes or kilobytes
  - Unlimited

A *quota* is a space allowance in a given tablespace. By default, a user has no quota on any of the tablespaces. You have three options for providing a quota for a user on a tablespace.

- **Unlimited:** Allows the user to use as much space as is available in the tablespace
- **Value:** Number of kilobytes or megabytes that the user can use. This does not guarantee that the space is set aside for the user. This value can be larger or smaller than the current space that is available in the tablespace.
- **UNLIMITED TABLESPACE system privilege:** Overrides all individual tablespace quotas and gives the user unlimited quota on all tablespaces, including SYSTEM and SYSAUX. This privilege must be granted with caution.

You must not provide a quota to users on the SYSTEM or SYSAUX tablespaces. Typically, only the SYS and SYSTEM users are able to create objects in the SYSTEM or SYSAUX tablespaces.

You do not need a quota on an assigned temporary tablespace or any undo tablespaces. You do not need to have a quota to insert, update, and delete data in an Oracle database. The only users that need quota are the accounts that own the database objects. It is typical when installing application code that the installer creates database accounts to own the objects. Only these accounts need quotas. Other database users can be granted permission to use these objects without a quota.

- The Oracle server checks the quota when a user creates or extends a segment.
- For activities that are assigned to a user schema, only those activities that use space in a tablespace count against the quota. Activities that do not use space in the assigned tablespace do not affect the quota (such as creating views or using temporary tablespaces).
- The quota is replenished when objects owned by the user are dropped with the `PURGE` clause or when the objects owned by the user in the recycle bin are purged.

# Applying the Principle of Least Privilege

- Protect the data dictionary:

```
O7_DICTIONARY_ACCESSIBILITY=FALSE
```

- Revoke unnecessary privileges from PUBLIC.
- Use access control lists (ACL) to control network access.
- Restrict the directories accessible by users.
- Limit users with administrative privileges.
- Restrict remote database authentication:

```
REMOTE_OS_AUTHENT=FALSE
```

ORACLE

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.

**Protect the data dictionary:** The O7_DICTIONARY_ACCESSIBILITY parameter is set by default to FALSE. You must not allow this to be changed without a very good reason because it prevents users with the ANY TABLE system privileges from accessing the data dictionary base tables. It also ensures that the SYS user can log in only as SYSDBA.

**Revoke unnecessary privileges from PUBLIC:** Several packages are extremely useful to applications that need them, but require proper configuration to be used securely. PUBLIC is granted execute privilege on the following packages: UTL_SMTP, UTL_TCP, UTL_HTTP, and UTL_FILE.

**Control network access:** In Oracle Database, network access is controlled by an access control list (ACL) that may be configured to allow certain users access to specific network services. Network access is denied by default. An ACL must be created to allow network access. File access through UTL_FILE is controlled at two levels:

- At the OS level with permissions on files and directories
- In the database by DIRECTORY objects that allow access to specific file system directories. The DIRECTORY object may be granted to a user for read or for read and write. Execute privileges on other PL/SQL packages should be carefully controlled.

The more powerful packages that may potentially be misused include:

- **UTL_SMTP:** Permits arbitrary email messages to be sent by using the database as a Simple Mail Transfer Protocol (SMTP) mail server. Use the ACL to control which machines may be accessed by which users.
- **UTL_TCP:** Permits outgoing network connections to be established by the database server to any receiving or waiting network service. Thus, arbitrary data can be sent between the database server and any waiting network service. Use the ACL to control access.
- **UTL_HTTP:** Allows the database server to request and retrieve data via HTTP. Granting this package to a user may permit data to be sent via HTML forms to a malicious website. Limit access by using the ACL.
- **UTL_FILE:** If configured improperly, allows text-level access to any file on the host operating system. When properly configured, this package limits user access to specific directory locations.

**Restrict access to OS directories:** The DIRECTORY object inside the database enables DBAs to map directories to OS paths and to grant privileges on those directories to individual users.

**Limit users with administrative privileges:** Do not provide database users more privileges than necessary. Non-administrators must not be granted the DBA role. To implement least privilege, restrict the following types of privileges:

- Grants of system and object privileges
- SYS-privileged connections to the database, such as SYSDBA and SYSOPER
- Other DBA-type privileges, such as DROP ANY TABLE

**Restrict remote database authentication:** The REMOTE_OS_AUTHENT parameter is set to FALSE by default. It must not be changed unless all clients can be trusted to authenticate users appropriately.

In the remote authentication process:

- The database user is authenticated externally
- The remote system authenticates the user
- The user logs in to the database without further authentication

**Note:** Always test your applications thoroughly if you have revoked privileges.

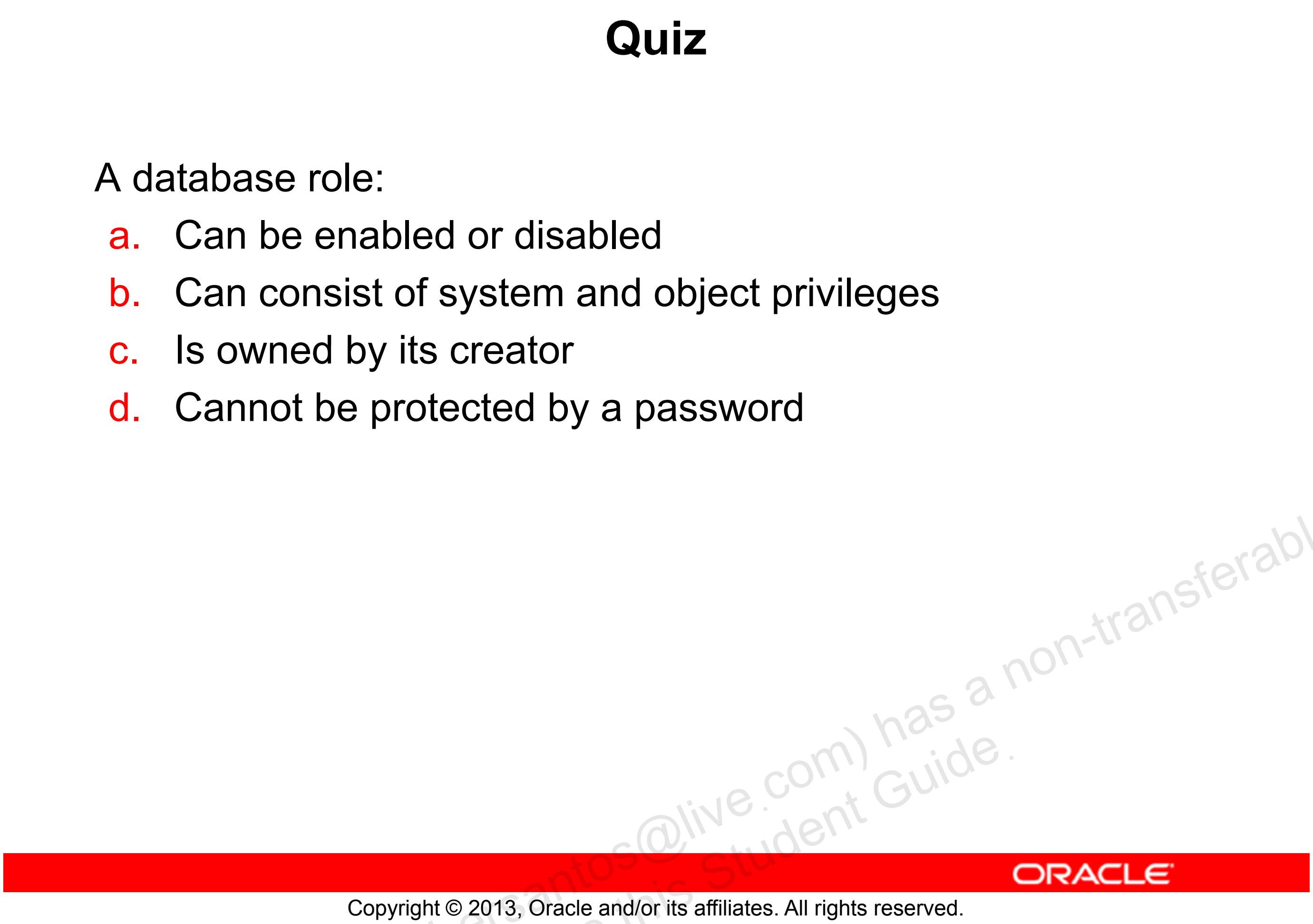
## Quiz

All passwords created in Oracle Database are not case-sensitive by default.

a. True

b. False

**Answer: b**



# Quiz

A database role:

a. Can be enabled or disabled

b. Can consist of system and object privileges

c. Is owned by its creator

d. Cannot be protected by a password

**Answer: a, b**

# Quiz

With `RESOURCE_LIMIT` set at its default value of `FALSE`, profile password limitations are ignored.

a. True

b. False

**Answer: b**

# Quiz

Applying the principle of least privilege is not enough to harden the Oracle database.

a. True

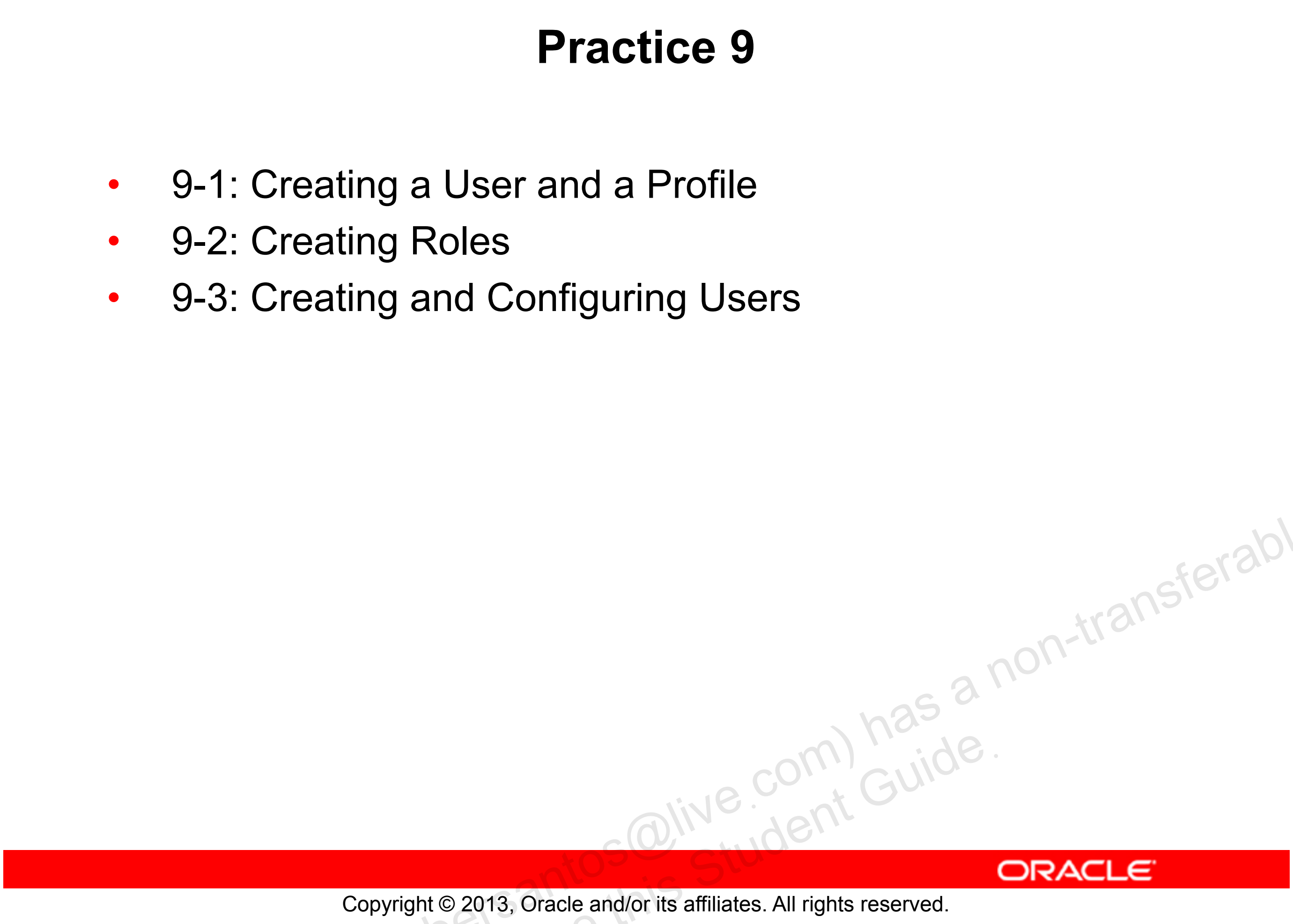
b. False

**Answer: a**

## Summary

In this lesson, you should have learned how to:

- Create and manage database user accounts:
  - Authenticate users
  - Assign default storage areas (tablespaces)
- Grant and revoke privileges
- Create and manage roles
- Create and manage profiles:
  - Implement standard password security features
  - Control resource usage by users



# Practice 9

- 9-1: Creating a User and a Profile
- 9-2: Creating Roles
- 9-3: Creating and Configuring Users