Oracle® Database 2 Day DBA
11*g* Release 2 (11.2)
Part Number E10897-06

7 Administering User Accounts and Security

This chapter describes how to create and manage user accounts. It contains the following sections:

About User Accounts

For users to access your database, you must create user accounts and grant appropriate database access privileges to those accounts. A user account is identified by a user name and defines the attributes of the user, including the following:

- Authentication method
- Password for database authentication
- Default tablespaces for permanent and temporary data storage
- Tablespace quotas
- Account status (locked or unlocked)
- Password status (expired or not)

When you create a user account, you must not only assign a user name, a password, and default tablespaces for the account, but you must also do the following:

- Grant the appropriate system privileges, object privileges, and roles to the account.
- If the user will be creating database objects, then give the user account a space usage quota on each tablespace in which the objects will be created.

Oracle recommends that you grant each user just enough privileges to perform his job, and no more. For example, a database application developer needs privileges to create and modify tables, indexes, views, and stored procedures, but does not need (and should not be granted) privileges to drop (delete) tablespaces or recover the database. You can create user accounts for database administration, and grant only a subset of administrative privileges to those accounts.

In addition, you may want to create user accounts that are used by applications only. That is, nobody logs in with these accounts; instead, applications use these accounts to connect to the database, and users log in to the applications. This type of user account avoids giving application users the ability to log in to the database directly, where they could unintentionally cause damage. See "About User Privileges and Roles" for more information.

When you create a user account, you are also implicitly creating a schema for that user. A **schema** is a logical container for the database objects (such as tables, views, triggers, and so on) that the user creates. The schema name is the same as the user name, and can be used to unambiguously refer to objects owned by the user. For example, `hr.employees` refers to the table named `employees` in the `hr` schema. (The `employees` table is owned by `hr`.) The terms *database object* and *schema object* are used interchangeably.

When you delete a user, you must either simultaneously delete all schema objects of that user, or you must have previously deleted the schema objects in separate operations.

Predefined User Accounts

In addition to the user accounts that you create, the database includes several user accounts that are automatically created upon installation.

All databases include the administrative accounts `SYS`, `SYSTEM`, `SYSMAN`, and `DBSNMP`. **Administrative accounts** are highly privileged accounts, and are needed only by individuals authorized to perform administrative tasks such as starting and stopping the database, managing database memory and storage, creating and managing database users, and so on. You log in to Oracle Enterprise Manager Database Control (Database Control) with `SYS`, `SYSTEM`, or `SYSMAN`. The Management Agent of Database Control uses the `DBSNMP` account to monitor and manage the database. You assign the passwords for these accounts when you create the database with Oracle Database Configuration Assistant (DBCA). You must not delete these accounts.

All databases also include **internal accounts**, which are automatically created so that individual Oracle Database features or components such as Oracle Application Express can have their own schemas. To protect these accounts from unauthorized access, they are initially locked and their passwords are expired. (A **locked account** is an account for which login is disabled.) You must not delete internal accounts, and you must not use them to log in to the database.

Your database may also include **sample schemas**, which are a set of interlinked schemas that enable Oracle

Hide Navigation

Search
○ This Book ● Entire Library

Quick Lookup
Advanced Search

Main Categories
- Home
- System Implementor or Integrator Guides
- System Administrator Guides
- Customization or Extension Developer Guides
- Security Guides
- Deployment Guides
- Roadmaps and Concepts Guides
- Installation and Patching Guides
- Release Notes

This Page

documentation and Oracle instructional materials to illustrate common database tasks. These schemas also provide a way for you to experiment without endangering production data.

Each sample schema has a user account associated with it. For example, the hr user account owns the hr schema, which contains a set of simple tables for a human resources application. The sample schema accounts are also initially locked and have an expired password. As the database administrator, you are responsible for unlocking these accounts and assigning passwords to these accounts.

See Also:

- *Oracle Database 2 Day + Security Guide* for a list of predefined user accounts
- "Locking and Unlocking User Accounts"
- "About Administrative Accounts and Privileges"
- "Administering Database User Accounts"
- *Oracle Database Sample Schemas* for a description of the sample schemas
- *Oracle Database Concepts* for an overview of database security

About User Privileges and Roles

User privileges provide a basic level of database security. They are designed to control user access to data and to limit the kinds of SQL statements that users can execute. When creating a user, you grant privileges to enable the user to connect to the database, to run queries and make updates, to create schema objects, and more.

The main types of user privileges are as follows:

- **System privileges**—A system privilege gives a user the ability to perform a particular action, or to perform an action on any schema objects of a particular type. For example, the system privilege CREATE TABLE permits a user to create tables in the schema associated with that user, and the system privilege CREATE USER permits a user to create database users.
- **Object privileges**—An object privilege gives a user the ability to perform a particular action on a specific schema object. Different object privileges are available for different types of schema objects. The privilege to select rows from the EMPLOYEES table or to delete rows from the DEPARTMENTS table are examples of object privileges.

Managing privileges is made easier by using **roles**, which are named groups of related privileges. You create roles, grant system and object privileges to the roles, and then grant roles to users. You can also grant roles to other roles. Unlike schema objects, roles are not contained in any schema.

Table 7-1 lists three widely used roles that are predefined in Oracle Database. You can grant these roles when you create a user or at any time thereafter.

Table 7-1 Oracle Database Predefined Roles

Role Name	Description
CONNECT	Enables a user to connect to the database. Grant this role to any user or application that needs database access. If you create a user using Database Control, then this role is automatically granted to the user.
RESOURCE	Enables a user to create, modify, and delete certain types of schema objects in the schema associated with that user. Grant this role only to developers and to other users that must create schema objects. This role grants a subset of the create object system privileges. For example, it grants the CREATE TABLE system privilege, but does not grant the CREATE VIEW system privilege. It grants only the following privileges: CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, CREATE TYPE.
DBA	Enables a user to perform most administrative functions, including creating users and granting privileges; creating and granting roles; creating, modifying, and deleting schema objects in any schema; and more. It grants all system privileges, but does not include the privileges to start or shut down the database instance. It is by default granted to users SYS and SYSTEM.

See Also:

- "Administering Roles"
- "Administering Database User Accounts"
- Chapter 8, "Managing Schema Objects"
- *Oracle Database 2 Day + Security Guide* for more information about privileges and roles
- *Oracle Database SQL Language Reference* for tables of system privileges, object privileges, and predefined roles
- *Oracle Database Concepts* for an overview of database security

About Administrative Accounts and Privileges

Administrative accounts and privileges enable you to perform administrative functions such as managing users, managing database memory, and starting up and shutting down the database.

This section contains the following topics:

- SYS and SYSTEM Users
- SYSDBA and SYSOPER System Privileges

See Also:

- "About User Accounts"
- "About User Privileges and Roles"
- "Administering Database User Accounts"

SYS and SYSTEM Users

The following administrative user accounts are automatically created when you install Oracle Database. They are both created with the password that you supplied upon installation, and they are both automatically granted the `DBA` role.

- `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

  While this account can be used to perform day-to-day administrative tasks, Oracle strongly recommends creating named users account for administering the Oracle database to enable monitoring of database activity.

SYSDBA and SYSOPER System Privileges

`SYSDBA` and `SYSOPER` are administrative privileges required to perform high-level administrative operations such as creating, starting up, shutting down, backing up, or recovering the database. The `SYSDBA` system privilege is for fully empowered database administrators and the `SYSOPER` system privilege allows a user to perform basic operational tasks, but without the ability to look at user data.

The `SYSDBA` and `SYSOPER` system privileges allow access to a database instance even when the database is not open. Control of these privileges is therefore completely outside of the database itself. This control enables an administrator who is granted one of these privileges to connect to the database instance to start the database.

You can also think of the `SYSDBA` and `SYSOPER` privileges as types of connections that enable you to perform certain database operations for which privileges cannot be granted in any other way. For example, if you have the `SYSDBA` privilege, then you can connect to the database using `AS SYSDBA`.

The `SYS` user is automatically granted the `SYSDBA` privilege upon installation. When you log in as user `SYS`, you must connect to the database as `SYSDBA` or `SYSOPER`. Connecting as a `SYSDBA` user invokes the `SYSDBA` privilege; connecting as `SYSOPER` invokes the `SYSOPER` privilege. Oracle Enterprise Manager Database Control does not permit you to log in as user `SYS` without connecting as `SYSDBA` or `SYSOPER`.

When you connect with the `SYSDBA` or `SYSOPER` privilege, you connect with a default schema, not with the schema that is generally associated with your user name. For `SYSDBA` this schema is `SYS`; for `SYSOPER` the schema is `PUBLIC`.

Caution:
When you connect as user `SYS`, you have unlimited privileges on data dictionary tables. Be certain that you do not modify any data dictionary tables.

See Also:

- *Oracle Database Administrator's Guide* for the operations authorized with the `SYSDBA` and `SYSOPER` privileges

---

Administering Roles

Roles are named groups of related system and object privileges. You create roles and then assign them to users and to other roles.

This section contains the following topics:

- Viewing Roles

- Example: Creating a Role
- Example: Modifying a Role
- Deleting a Role

> **See Also:**
>
> - "About User Privileges and Roles"
> - *Oracle Database 2 Day + Security Guide* for more information about administering user security, roles, and privileges

Viewing Roles

You view roles on the Roles page of Oracle Enterprise Manager Database Control (Database Control).

To view roles:

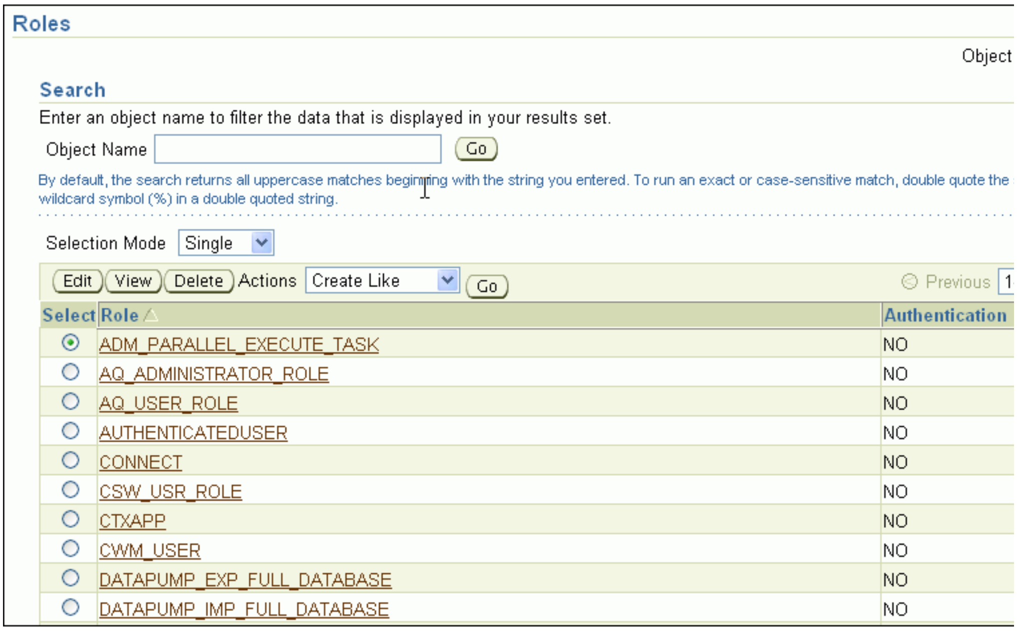
1. Go to the Database Home page, logging in with a user account that has privileges to manage roles. An example of such a user account is `SYSTEM`.

   See "Accessing the Database Home Page".

2. At the top of the page, click **Server** to view the Server subpage.

3. In the Security section, click **Roles**.

   The Roles page appears.



Roles

Object

Search

Enter an object name to filter the data that is displayed in your results set.

Object Name [ ] Go

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the wildcard symbol (%) in a double quoted string.

Selection Mode: Single

Select	Role	Authentication
●	ADM_PARALLEL_EXECUTE_TASK	NO
○	AQ_ADMINISTRATOR_ROLE	NO
○	AQ_USER_ROLE	NO
○	AUTHENTICATEDUSER	NO
○	CONNECT	NO
○	CSW_USR_ROLE	NO
○	CTXAPP	NO
○	CWM_USER	NO
○	DATAPUMP_EXP_FULL_DATABASE	NO
○	DATAPUMP_IMP_FULL_DATABASE	NO

Description of the illustration view_roles.gif

4. To view the details of a particular role, in the **Select** column, select the name of the role you want to view, and then click **View**.

   If you do not see the role, then it may be on another page. In this case, do one of the following:

   - Just above the list of roles, click **Next** to view the next page. Continue clicking **Next** until you see the desired role.
   - Use the Search area of the page to search for the desired role. In the **Object Name** field, enter the first few letters of the role, and then click **Go**.

     You can then select the role and click **View**.

   The View Role page appears. In this page, you can see all the privileges and roles granted to the selected role.

Example: Creating a Role

Suppose you want to create a role called `APPDEV` for application developers. Because application developers must be able to create, modify, and delete the schema objects that their applications use, you want the `APPDEV` role to include the system privileges shown in Table 7-2.

Table 7-2 System Privileges Granted to the APPDEV Role

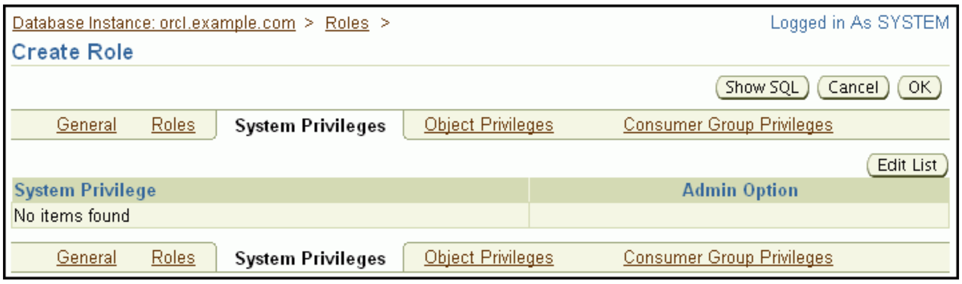
Privilege	Description
`CREATE TABLE`	Enables a user to create, modify, and delete tables in his schema.
`CREATE VIEW`	Enables a user to create, modify, and delete views in his schema.

`CREATE PROCEDURE`	Enables a user to create, modify, and delete procedures in his schema.
`CREATE TRIGGER`	Enables a user to create, modify, and delete triggers in his schema.
`CREATE SEQUENCE`	Enables a user to create, modify, and delete sequences in his schema.
`CREATE SYNONYM`	Enables a user to create, modify, and delete synonyms in his schema.

To create the APPDEV role:

1. Go to the Roles page, as described in "Viewing Roles".
2. Click **Create**.

   The Create Role page appears.
3. In the **Name** field, enter `APPDEV`.
4. Click **System Privileges** to go to the System Privileges subpage.



Database Instance: orcl.example.com > Roles >

Logged in As SYSTEM

Create Role

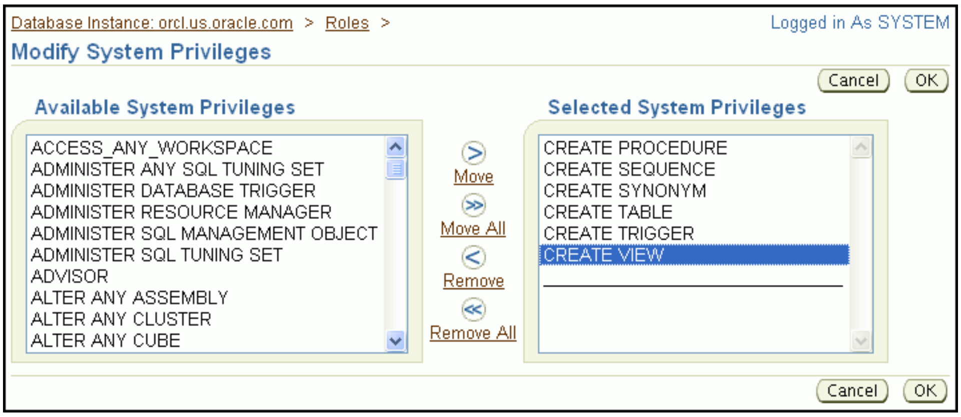
System Privilege	Admin Option
No items found	

Description of the illustration create_role.gif

   The table of system privileges for this role contains no rows yet.
5. Click **Edit List**.

   The Modify System Privileges page appears.
6. In the Available System Privileges list, double-click privileges to add them to the Selected System Privileges list.

   The privileges to add are listed in Table 7-2.



Database Instance: orcl.us.oracle.com > Roles >

Logged in As SYSTEM

Modify System Privileges

Available System Privileges

ACCESS_ANY_WORKSPACE
ADMINISTER ANY SQL TUNING SET
ADMINISTER DATABASE TRIGGER
ADMINISTER RESOURCE MANAGER
ADMINISTER SQL MANAGEMENT OBJECT
ADMINISTER SQL TUNING SET
ADVISOR
ALTER ANY ASSEMBLY
ALTER ANY CLUSTER
ALTER ANY CUBE

Selected System Privileges

CREATE PROCEDURE
CREATE SEQUENCE
CREATE SYNONYM
CREATE TABLE
CREATE TRIGGER
CREATE VIEW

Description of the illustration modify_system_privs.gif

> **Note:**
> Double-clicking a privilege is a shortcut. You can also select a privilege and then click the **Move** button. To select multiple privileges, hold down the Shift key while selecting a range of privileges, or press the Ctrl key and select individual privileges, then click **Move** after you have selected the privileges.

7. Click **OK**.

   The System Privileges subpage returns, showing the system privileges that you selected. At this point, you could click **Roles** to assign other roles to the `APPDEV` role, or click **Object Privileges** to assign object privileges to the `APPDEV` role.
8. Click **OK** to return to the Roles page.

   The `APPDEV` role now appears in the table of database roles.

Example: Modifying a Role

Suppose your applications make use of Oracle Streams Advanced Queuing, and you determine that developers must be granted the roles `AQ_ADMINISTRATOR_ROLE` and `AQ_USER_ROLE` to develop and test their applications. You must edit the `APPDEV` role to grant it these two Advanced Queuing roles.

To modify the APPDEV role:

1. Go to the Roles page, as described in "Viewing Roles".
2. In the Select column, click **APPDEV** role, and then click **Edit**.

   The Edit Role page appears.
3. Click **Roles** to navigate to the Roles subpage.
4. Click **Edit List**.

   The Modify Roles page appears.
5. In the Available Roles list, double-click the roles `AQ_ADMINISTRATOR_ROLE` and `AQ_USER_ROLE` to add them to the Selected Roles list.
6. Click **OK**.

   The Roles subpage returns, showing that the roles that you selected were granted to the `APPDEV` role.
7. Click **Apply** to save your changes.

   An update message appears, indicating that the role `APPDEV` was modified successfully.

Deleting a Role

Use caution when deleting a role, because Database Control deletes a role even if that role is currently granted to one or more users. Before deleting a role, you may want to determine if the role has any grantees. Dropping (deleting) a role automatically removes the privileges associated with that role from all users that had been granted the role.

To determine if a role has any grantees:

1. Go to the Roles page as described in "Viewing Roles".
2. In the Select column, click the desired role.

   If you do not see the desired role, then it may be on another page. In this case, do one of the following:

   - Just above the list of roles, click **Next** to view the next page. Continue clicking **Next** until you see the desired role.
   - Use the Search area of the page to search for the desired role. In the **Object Name** field, enter the first few letters of the role, and then click **Go**.

   You can then select the role.
3. In the Actions list, select **Show Grantees**, and then click **Go**.

   A report appears, listing the users that are granted the selected role.
4. Click **Cancel** to return to the Roles page.

To delete a role:

1. If you are not there, then go to the Roles page, as described in "Viewing Roles".
2. In the Select column, click the desired role, and then click **Delete**.

   A confirmation page appears.
3. Click **Yes**.

   A confirmation message indicates that the role has been deleted successfully.

Administering Database User Accounts

This section provides instructions for creating and managing user accounts for the people and applications that use your database. It contains the following topics:

- Viewing User Accounts
- Example: Creating a User Account
- Creating a New User Account by Duplicating an Existing User Account
- Example: Granting Privileges and Roles to a User Account
- Example: Assigning a Tablespace Quota to a User Account
- Example: Modifying a User Account
- Locking and Unlocking User Accounts
- Expiring a User Password
- Example: Deleting a User Account

See Also:

- "About User Accounts"

Viewing User Accounts

You view user accounts on the Users page of Oracle Enterprise Manager Database Control (Database Control).

To view users:

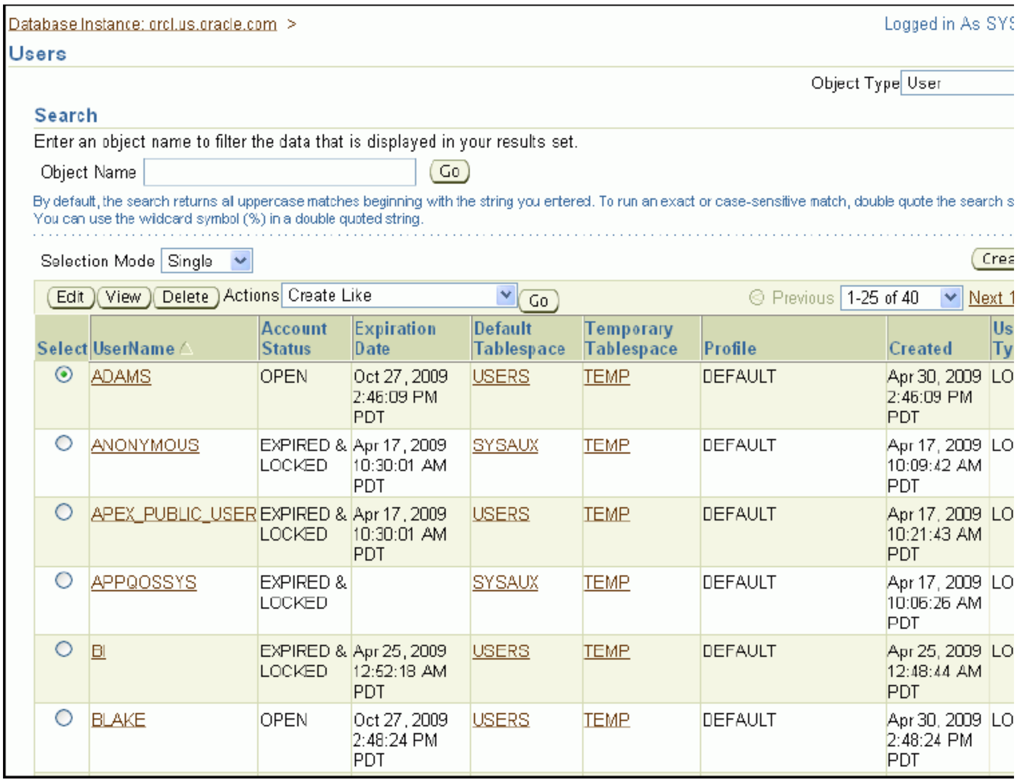
1. Go to the Database Home page, logging in with a user account that has privileges to manage users, for example, `SYSTEM`.

   See "Accessing the Database Home Page".

2. At the top of the page, click **Server** to view the Server subpage.

3. In the Security section, click **Users**.

   The Users page appears.



Database Instance: orcl.us.oracle.com > Logged in As SYS

Users

Object Type User

Search

Enter an object name to filter the data that is displayed in your results set.

Object Name [ ] Go

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search s

You can use the wildcard symbol (%) in a double quoted string.

Selection Mode Single

Edit | View | Delete | Actions Create Like | Go | Previous 1-25 of 40 Next

Select	UserName	Account Status	Expiration Date	Default Tablespace	Temporary Tablespace	Profile	Created	Us Ty
●	ADAMS	OPEN	Oct 27, 2009 2:46:09 PM PDT	USERS	TEMP	DEFAULT	Apr 30, 2009 2:46:09 PM PDT	LO
○	ANONYMOUS	EXPIRED & LOCKED	Apr 17, 2009 10:30:01 AM PDT	SYSAUX	TEMP	DEFAULT	Apr 17, 2009 10:09:42 AM PDT	LO
○	APEX_PUBLIC_USER	EXPIRED & LOCKED	Apr 17, 2009 10:30:01 AM PDT	USERS	TEMP	DEFAULT	Apr 17, 2009 10:21:43 AM PDT	LO
○	APPQOSSYS	EXPIRED & LOCKED		SYSAUX	TEMP	DEFAULT	Apr 17, 2009 10:06:26 AM PDT	LO
○	BI	EXPIRED & LOCKED	Apr 25, 2009 12:52:18 AM PDT	USERS	TEMP	DEFAULT	Apr 25, 2009 12:48:44 AM PDT	LO
○	BLAKE	OPEN	Oct 27, 2009 2:48:24 PM PDT	USERS	TEMP	DEFAULT	Apr 30, 2009 2:48:24 PM PDT	LO

Description of the illustration users_page.gif

4. To view the details of a particular user, in the **Select** column, click the user, and then click **View**.

   If you do not see the user, then it may be on another page. In this case, do one of the following:

   - Just above the list of users, click **Next** to view the next page. Continue clicking **Next** until you see the desired user.
   - Use the Search area of the page to search for the desired user. In the **Object Name** field, enter the first few letters of the user name, and then click **Go**.
   - Click a table column to change the sort order of the data in the table. For example, to list the users in reverse alphabetical order, click the UserName column heading.

   You can then select the user and click **View**.

   The View User page appears, and displays all user attributes.

Example: Creating a User Account

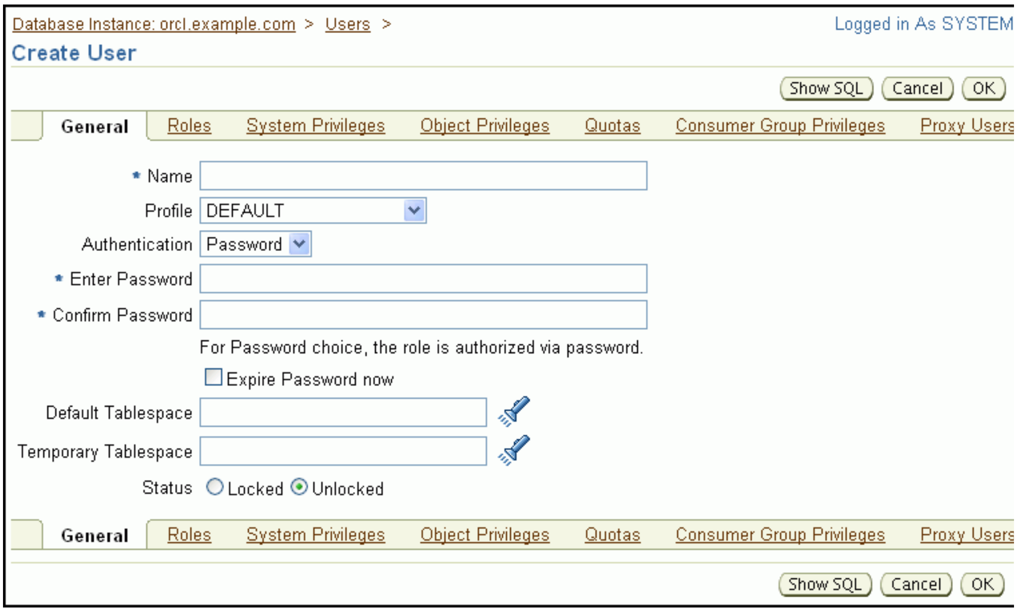
Suppose you want to create a user account for a database application developer named Nick. Because Nick is a developer, you want to grant him the database privileges and roles that he requires to build and test his applications. You also want to give Nick a 10 MB quota on his default tablespace so that he can create schema objects in that tablespace.

To create the user Nick:

1. Go to the Users page, as described in "Viewing User Accounts".

2. On the Users page, click **Create**.

   The Create User page appears, displaying the General subpage.



* Name

Profile DEFAULT

Authentication Password

* Enter Password

* Confirm Password

For Password choice, the role is authorized via password.

□ Expire Password now

Default Tablespace

Temporary Tablespace

Status ○ Locked ● Unlocked

Description of the illustration create_user.gif

3. In the **Name** field, enter `NICK`.
4. In the Profile list, accept the value `DEFAULT`.

   This setting assigns the default password policy to user Nick.

   See "Setting the Database Password Policy".
5. Accept the default value `Password` in the Authentication list.

   For information about other more advanced methods to authenticate users, see *Oracle Database Security Guide*.
6. In the **Enter Password** and **Confirm Password** fields, enter a password that is secure.

   Create a password that is secure. See *Oracle Database Security Guide* for more information.
7. Do not select **Expire password now**. If the account status is set to expired, then the user or the database administrator must change the password before the user can log in to the database.
8. (Optional) Next to the **Default Tablespace** field, click the flashlight icon, select the **USERS** tablespace, and then click **Select**.

   All schema objects that Nick creates will then be created in the `USERS` tablespace unless he specifies otherwise. If you leave the Default Tablespace field blank, then Nick is assigned the default tablespace for the database, which is `USERS` in a newly installed database. For more information about the `USERS` tablespace, see "About Tablespaces".
9. (Optional) Next to the **Temporary Tablespace** field, click the flashlight icon, select the **TEMP** tablespace, and then click **Select**.

   If you leave the Temporary Tablespace field blank, then Nick is assigned the default temporary tablespace for the database, which is `TEMP` in a newly installed database. For more information about the `TEMP` tablespace, see "About Tablespaces".
10. For the Status option, accept the default selection of **Unlocked**.

    You can later lock the user account to prevent users from logging in with it. To temporarily deny access to a user account, locking the user account is preferable to deleting it, because deleting it also deletes all schema objects owned by the user.
11. Grant roles, system privileges, and object privileges to the user, as described in "Example: Granting Privileges and Roles to a User Account".

    > **Note:**
    > Do not click OK in Step 13 of "Example: Granting Privileges and Roles to a User Account". Instead, skip that step and continue with Step 12 in this procedure.
12. Assign a 10 MB quota on the `USERS` tablespace, as described in "Example: Assigning a Tablespace Quota to a User Account".
13. If you did not click OK while assigning the tablespace quota (previous step), then click **OK** now to create the user.

> **See Also:**
>
> - "Creating Database Control Administrative Users"
> - *Oracle Database 2 Day + Security Guide*

Creating a New User Account by Duplicating an Existing User Account

To create a user account that is similar in attributes to an existing user account, you can duplicate the existing user account.

To create a new user account by duplicating an existing user account:

1. Go to the Users page, as described in "Viewing User Accounts".
2. In the **Select** column, click the user to duplicate.
3. In the Actions list, select **Create Like**, and then click **Go**.

   The Create User page appears. This page displays a new user with the same attributes as the duplicated user.
4. Enter a user name and password, modify the user attributes or privileges if desired, and then click **OK** to save the new user.

The Actions list also provides shortcuts for other actions, and provides a way to display the SQL command used to create a user.

Example: Granting Privileges and Roles to a User Account

Suppose you are creating or modifying a user account named Nick. Because Nick is a database application developer, you want to grant him the APPDEV role, which enables him to create database objects in his own schema. (You created the APPDEV role in "Example: Creating a Role".) Because you want Nick to be able to create tables and views in other schemas besides his own, you want to grant him the CREATE ANY TABLE and CREATE ANY VIEW system privileges. In addition, because he is developing a human resources application, you want him to be able to view the tables in the hr sample schema and use them as examples. You therefore want to grant him the SELECT object privilege on those tables. Finally, you want Nick to be able to log in to Database Control so that he can use the graphical user interface to create and manage his database objects. You therefore want to grant him the SELECT_CATALOG_ROLE role. The following table summarizes the privileges and roles to grant to Nick.

Grant Type	Privilege or Role Name
System privileges	CREATE ANY TABLE, CREATE ANY VIEW
Object privileges	SELECT on all tables in the hr schema
Roles	APPDEV, SELECT_CATALOG_ROLE

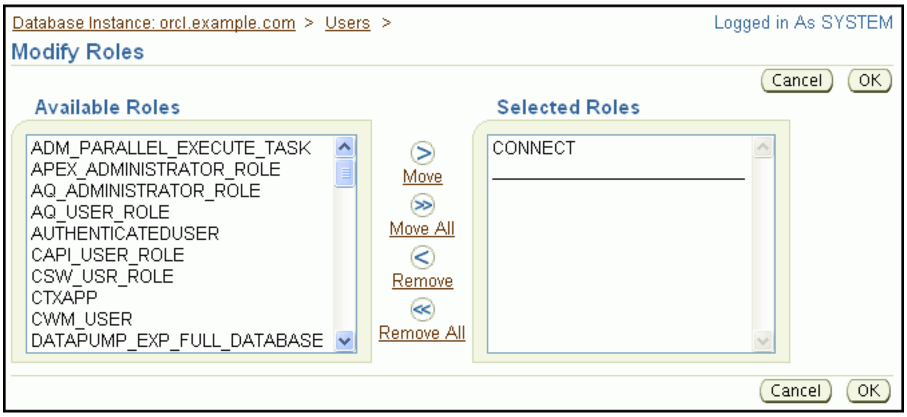
The following example assumes that you are in the process of creating the user account for Nick or editing the account. Either you have accessed the Create User page and have entered all required fields on the General subpage (see "Example: Creating a User Account"), or you have accessed the Edit User page for Nick (see "Example: Modifying a User Account"). The example also assumes that you have not yet granted any privileges or roles to Nick.

To grant privileges and roles to the user Nick:

1. Toward the top of the Create User or Edit User page, click **Roles** to display the Roles subpage.

   The Roles subpage shows that the CONNECT role is assigned to Nick. Database Control automatically assigns this role to all users that you create. (The selected Default check box indicates that the CONNECT role is a **default role** for Nick, which means that it is automatically enabled whenever Nick logs in.)
2. Click **Edit List**.

   The Modify Roles page appears.



   Description of the illustration modify_roles.gif

3. In the Available Roles list, locate the APPDEV role, and double-click it to add it to the Selected Roles list. Do the same with the SELECT_CATALOG_ROLE role and then click **OK**.

   The Create User or Edit User page returns, showing that the CONNECT, APPDEV, and SELECT_CATALOG_ROLE roles are granted to Nick.

   Note:
   Double-clicking a role is a shortcut. You can also select the role and then click the **Move** button. To select multiple privileges, hold down the Shift key while selecting a range of privileges, or press the Ctrl key and select individual privileges.

4. Toward the top of the page, click **System Privileges** to select the System Privileges subpage.

5. Click **Edit List**.

   The Modify System Privileges page appears.

6. In the Available System Privileges list, scroll to locate the `CREATE ANY TABLE` and `CREATE ANY VIEW` privileges, double-click each to add them to the Selected System Privileges list, and then click **OK**.

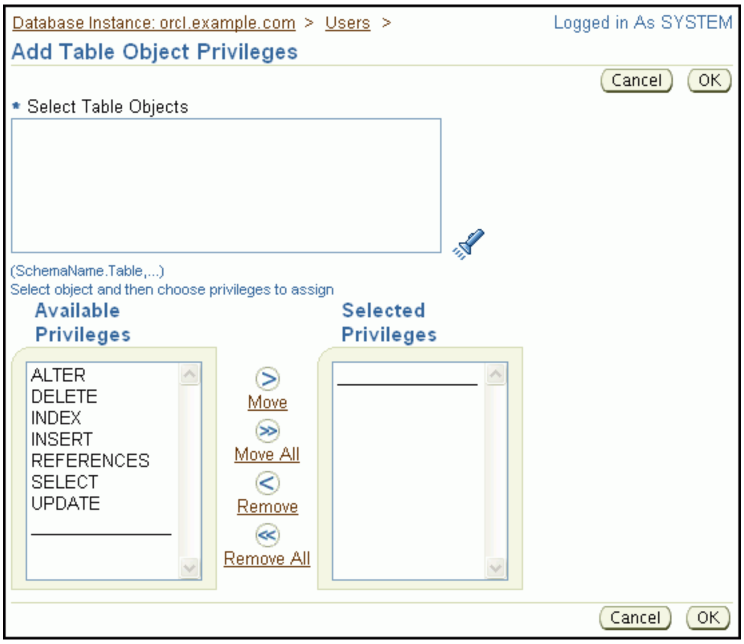
   The Create User or Edit User page returns, showing the newly added system privileges.

> **Note:**
> To revoke a role, double-click it in the Selected Roles list on the Modify Roles page. To revoke a system privilege, double-click it in the Selected System Privileges list on the Modify System Privileges page.

7. Toward the top of the page, click **Object Privileges** to select the Object Privileges subpage.

8. In the Select Object Type list, select **Table** and then click **Add**.

   The Add Table Object Privileges page appears.



Database Instance: orcl.example.com > Users >

Logged in As SYSTEM

Add Table Object Privileges

* Select Table Objects

(SchemaName.Table,...)
Select object and then choose privileges to assign

Available Privileges

ALTER
DELETE
INDEX
INSERT
REFERENCES
SELECT
UPDATE

Selected Privileges

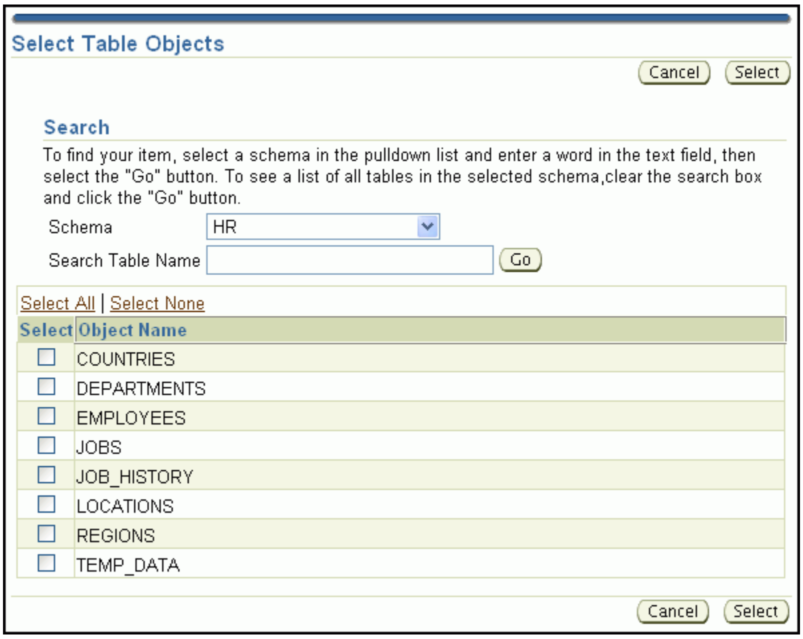
Description of the illustration add_object_privileges.gif

9. Click the flashlight icon next to the Select Table Objects list.

   The Select Table Objects dialog box appears.

10. In the Schema list, select **HR**, and then click **Go**.

    All tables in the `hr` schema are displayed.



Select Table Objects

Search

To find your item, select a schema in the pulldown list and enter a word in the text field, then select the "Go" button. To see a list of all tables in the selected schema,clear the search box and click the "Go" button.

Schema: HR

Search Table Name:

Select All | Select None

Select	Object Name
□	COUNTRIES
□	DEPARTMENTS
□	EMPLOYEES
□	JOBS
□	JOB_HISTORY
□	LOCATIONS
□	REGIONS
□	TEMP_DATA

Description of the illustration select_table_objects.gif

11. Click **Select All**, and then click the **Select** button.

    The Select Table Objects dialog box closes, and the names of all tables in the `hr` schema appear in the Select Table Objects field on the Add Table Object Privileges page.

12. In the Available Privileges list, double-click the **`SELECT`** privilege to move it to the Selected Privileges list, and then click **OK**.

    The Create User or Edit User page returns, showing that the `SELECT` object privilege for all `hr` tables is granted to user Nick.

    Note:
    To revoke an object privilege, select it on the Create User or Edit User page (Object Privileges subpage), and then click **Delete**.

13. Do one of the following to save the role and privilege grants:

    - If you are creating a user account, then click **OK** to save the new user account.
    - If you are modifying a user account, then click **Apply** to save the changes for the user account.

    See Also:

    - "About User Privileges and Roles"
    - *Oracle Database 2 Day + Security Guide*

Example: Assigning a Tablespace Quota to a User Account

Suppose you are creating or modifying a user account named Nick. You want to assign Nick a space usage quota of 10 MB on his default tablespace.

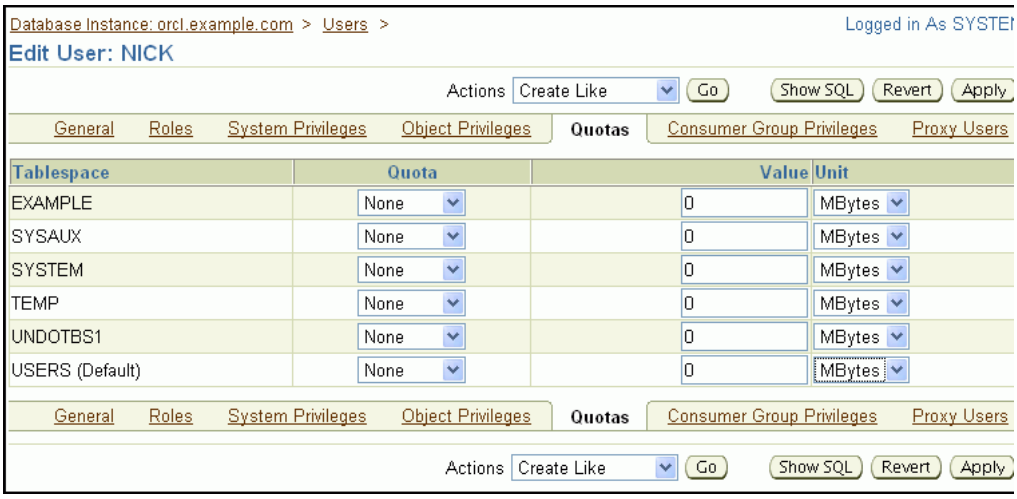
You must assign Nick a tablespace quota on his default tablespace before he can create objects in that tablespace. (This is also true for any other tablespace in which Nick wants to create objects.) After a quota is assigned to Nick for a particular tablespace, the total space used by all of his objects in that tablespace cannot exceed the quota. You can also assign a quota of `UNLIMITED`.

The following example assumes that you are in the process of creating the user account for Nick or editing the account. Either you have accessed the Create User page and have entered all required fields on the General subpage (see "Example: Creating a User Account"), or you have accessed the Edit User page for Nick (see "Example: Modifying a User Account"). The example also assumes that Nick has not yet been assigned a quota on any tablespaces.

To assign a tablespace quota to user Nick:

1. Toward the top of the Create User or Edit User page, select the **Quotas** subpage.

   The Quotas subpage appears, showing that user Nick does not have a quota assigned on any tablespace.



Database Instance: orcl.example.com > Users >
Logged in As SYSTE
Edit User: NICK
Actions Create Like Go
Show SQL Revert Apply
General Roles System Privileges Object Privileges Quotas Consumer Group Privileges Proxy Users

Tablespace	Quota	Value	Unit
EXAMPLE	None	0	MBytes
SYSAUX	None	0	MBytes
SYSTEM	None	0	MBytes
TEMP	None	0	MBytes
UNDOTBS1	None	0	MBytes
USERS (Default)	None	0	MBytes

General Roles System Privileges Object Privileges Quotas Consumer Group Privileges Proxy Users
Actions Create Like Go
Show SQL Revert Apply

Description of the illustration quotas.gif

2. In the **Quota** column for tablespace `USERS`, select **Value** from the list.
3. In the **Value** column for tablespace `USERS`, enter **`10`**.
4. Do one of the following to save the new quota assignment:

   - If you are creating a user account, then click **OK** to save the new user account.
   - If you are modifying a user account, then click **Apply** to save changes for the user account.

Example: Modifying a User Account

Suppose you want to remove the quota limitations for the user Nick on his default tablespace, USERS. To do so, you must modify his user account.

To modify the user Nick:

1. Go to the Users page, as described in "Viewing User Accounts".
2. In the **Select** column, select the user account Nick, and then click **Edit**.

   If you do not see user Nick, then he may be on another page. In this case, do one of the following:

   - Just above the list of user accounts, click **Next** to view the next page. Continue clicking **Next** until you see the user account for Nick.
   - Use the Search area of the page to search for his account. In the **Object Name** field, enter the letters **NI**, and then click **Go**.

   You can then select the user account for Nick and click **Edit**.

   The Edit User page appears, and displays the general attributes for Nick.
3. Toward the top of the page, select the **Quotas** subpage.
4. In the **Quota** column for tablespace USERS, select **Unlimited** from the list, and then click **Apply**.

   A message appears, indicating that user Nick was modified successfully.

Locking and Unlocking User Accounts

To temporarily deny access to the database for a particular user account, you can lock the user account. If the user then attempts to connect, then the database displays an error message and does not allow the connection. You can unlock the user account when you want to permit database access again for that user.

To lock or unlock a user account:

1. Go to the Users page, as described in "Viewing User Accounts".
2. In the **Select** column, click the desired user account.

   If you do not see the desired user account, then it may be on another page. In this case, use the **Next** button to view additional pages or use the Search area of the page to search for the desired user account.
3. Do one of the following:

   - To lock the account, select **Lock User** from the Actions list, and then click **Go**.
   - To unlock the account, select **Unlock User** from the Actions list, and then click **Go**.

   A confirmation message appears.
4. Click **Yes**.

Expiring a User Password

When you expire a user password, the user is prompted to change his or her password the next time that user logs in. Reasons to expire a password include the following:

- A user password becomes compromised.
- You have a security policy in place that requires users to change their passwords on a regular basis.

  Note:
  You can automate the automatic expiring of user passwords after a certain interval. See "Setting the Database Password Policy".

- A user has forgotten his or her password.

  In this third case, you modify the user account, assign a new temporary password, and expire the password. The user then logs in with the temporary password and is prompted to choose a new password.

To expire a user password:

1. Go to the Users page, as described in "Viewing User Accounts".
2. In the **Select** column, click the desired user account.

   If you do not see the desired user account, then it may be on another page. In this case, do one of the following:

   - Just above the list of user accounts, click **Next** to view the next page. Continue clicking **Next** until you see the desired user account.
   - Use the Search area of the page to search for the desired user account. In the **Object Name** field, enter the first few letters of the user account name, and then click **Go**.

   You can then select the user account.
3. To expire the passwords for all users, select the **Multiple** option, then click **Select All**.
4. Select **Expire Password** from the Actions list, and then click **Go**.

A confirmation message appears.

5. Click **Yes** to complete the task.

Example: Deleting a User Account

Suppose user Nick has moved to another department. Because it is no longer necessary for him to have access to the database, you want to delete his user account.

You must use caution when deciding to deleting a user account, because this action also deletes all schema objects owned by the user. To prevent a user from logging in to the database while keeping the schema objects intact, lock the user account instead. See "Locking and Unlocking User Accounts".

To delete user Nick:

1. Go to the Users page, as described in "Viewing User Accounts".
2. In the **Select** column, select the user account Nick, and then click **Delete**.

   If you do not see the user account Nick, then it may be on another page. In this case, do one of the following:

   - Just above the list of user accounts, click **Next** to view the next page. Continue clicking **Next** until you see the user account for Nick.
   - Use the Search area of the page to search for the user account. In the **Object Name** field, enter the letters **NI**, and then click **Go**.

   You can then select the user account for Nick and click **Delete**.

   A confirmation page appears.

3. Click **Yes** to confirm the deletion of the user account.

Setting the Database Password Policy

This section provides background information and instructions for setting the password policy for all user accounts in the database. It contains the following topics:

- About Password Policies
- Modifying the Default Password Policy

See Also:

- "Administering Database User Accounts"
- *Oracle Database 2 Day + Security Guide*

About Password Policies

When you create a user account, a default password policy is assigned to that user account. The default password policy for a newly installed database includes these directives:

- The password for the user account expires automatically in 180 days.
- The user account is locked 7 days after password expiration.
- The user account is locked for 1 day after 10 failed login attempts.

The default password policy is assigned to user accounts through a database object called a *profile*. Each user account is assigned a profile, and the profile has several attributes that describe a password policy. The database comes with a default profile (named `DEFAULT`), and unless you specify otherwise when you create a user account, the default profile is assigned to the user account.

For better database security, you may want to impose a more strict password policy. For example, you may want passwords to expire every 70 days, and you may want to lock user accounts after three failed login attempts. (A failed login attempt for a user account occurs when a user enters an incorrect password for the account.) You may also want to require that passwords be complex enough to provide reasonable protection against intruders who try to break into the system by guessing passwords. For example, you might specify that passwords must contain at least one number and one punctuation mark.

You change the password policy for every user account in the database by modifying the password-related attributes of the `DEFAULT` profile.

Note:
It is possible to have different password policies for different user accounts. You accomplish this by creating multiple profiles, setting password-related attributes differently for each profile, and assigning different profiles to different user accounts. This scenario is not addressed in this section.

Modifying the Default Password Policy

You modify the default password policy for every database user account by modifying the password-related attributes of the profile named `DEFAULT`.

To modify the default password policy:

1. Go to the Database Home page.

   See "Accessing the Database Home Page".

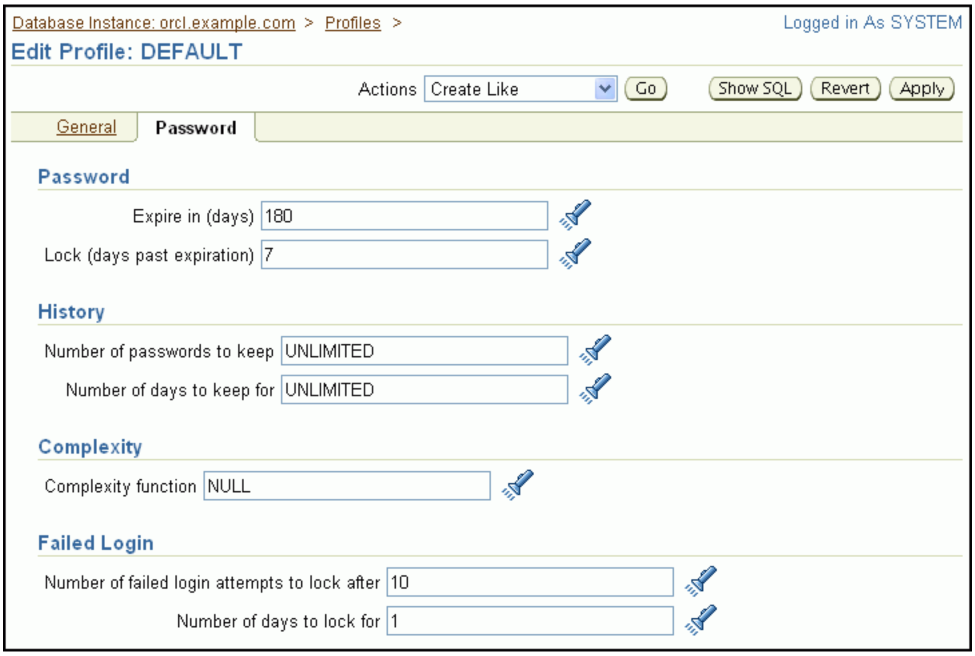
2. At the top of the page, click **Server** to view the Server subpage.
3. In the Security section, click **Profiles**.

   The Profiles page appears.

4. In the **Select** column, select the profile named **`DEFAULT`**, and then click **Edit**.

   The Edit Profile page appears.

5. Toward the top of the page, select the **Password** subpage.



Password

Expire in (days) 180

Lock (days past expiration) 7

History

Number of passwords to keep UNLIMITED

Number of days to keep for UNLIMITED

Complexity

Complexity function NULL

Failed Login

Number of failed login attempts to lock after 10

Number of days to lock for 1

Description of the illustration edit_profile.gif

6. Change field values as required. Click the flashlight icon next to each field to view a list of choices. (Click **Help** on this page for a description of the fields.)
7. Click **Apply** to save your changes.

See Also:

- "About Password Policies"
- *Oracle Database 2 Day + Security Guide*

Users: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this chapter and includes annotated screenshots.

To view the Users OBE, in your browser, enter the following URL:

`http://www.oracle.com/technology/obe/11gr2_2day_dba/users/users.htm`