# Practices for Lesson 21: Configuring User Resource Limits

Practices for Lesson 21: Overview

Overview

In these practices, you will create and manage user profiles.

Practice 21-1: Using SQL\*Developer to Create a Local Profile

Overview

In this practice, the PDBADMIN user (local administrator for ORCLPDB1) creates a local profile called HRPROFILE in to limit the amount of idle time users can have in the PDB. If a user is idle or forgets to log out after 60 minutes, the user session is ended.

In addition, the profile will be configured to automatically lock a database user account if it did not log on after a specified number of days. This locking mechanism is implemented through the INACTIVE\_ACCOUNT\_TIME user resource profile limit.

Tip

A local profile is a profile that resides in a single PDB. Therefore, to create one, you must log in to the PDB.

Assumptions

You are currently logged in as the oracle OS user. The PDBADMIN user has been granted the DBA role.

Tasks

Log in to SQL\*Developer (ORCLPDB1)

Launch SQL\*Developer.

Click Menu View > DBA

In the DBA box, double click: **PDB1-pdbadmin.**

A worksheet for PDB1-pdbadmin will appear on the right.

View Privileges and Roles for PDBADMIN

In the DBA box, expand **PDB1-pdadmin** > **Security**> **Users**

On the Users tab, select **PDBADMIN**

The PDBADMIN tab is displayed. Click the **Roles** subtab. Notice that DBA is a ROLE.

In the DBA box, expand **Security** and **Roles** and select **DBA**.

In the DBA tab, click the **Sys Privs** subtab. Scroll down through the list of privileges.

Hover over the System Privilege header. Click the Filter icon. To look for specific privileges, for example, **CREATE PROFILE**, type **CREATE PROFILE** in the **Filter:System Privilege**field and press **Enter.**

Close the DBA Role tab.

Create a Local Profile

In the DBA box, expand **Security** and click **Profiles**.

In the Profiles tab, click **Actions** > **Create New …**

In the Create Profile box, enter:

Profile Name: **HRPROFILE**

IdleTime (minutes): **15**

Leave all other fields set to Default.

Click the **SQL** tab to review the SQL command for this task. When done, click **APPLY.**

In the Successful window, click **OK**.

Verify that HRPROFILE is in the list of profiles. Note, you may need to click **Refresh**  **Set the RESOURCE\_LIMIT Initialization Parameter**

In the DBA box, expand **Database Configuration** and select **Initialization Parameters**.

Hoover over the Parameter header and click the Filter icon. Type **resource\_limit** in the Filter: Parameter field and press **Enter**.

The RESOURCE\_LIMIT initialization parameter is listed in the table. Verify that the

RESOURCE LIMIT value is set to **TRUE**.

If RESOURCE\_LIMIT is not set to true, perform the following steps:

Double-click the Value field of the RESOURCE\_LIMIT row.

Select **TRUE**, and click somewhere outside the Value field.

The row number will have an asterisk, then you can press F11 or click the icon to commit the change,

In the Commit Strategy dialog box, check both **Memory** and **SPFile** boxes and then click **Apply**.

Close the SQL\*Developer window.

Modify the Profile so that Database User Accounts Will be Locked if Not Used in 10 Days

To lock database user accounts, modify the HRPROFILE profile to add the INACTIVE\_ACCOUNT\_TIME user resource profile limit. In this section, you use SQL\*Plus and learn by trial and error.

Open a new terminal window and source the oraenv script for the **orclcdb** database. .

Connect to ORCLPDB1 as the local DBA, PDBADMIN. Refer to *Course Practice Environment: Security Credentials* for the ***password*** value.

Issue the ALTER PROFILE command to set the INACTIVE\_ACCOUNT\_TIME limit in the profile to 10 days.

Question: Is INACTIVE\_ACCOUNT\_TIME a valid profile limit? To find out, query the

DBA\_PROFILES view and confirm that INACTIVE\_ACCOUNT\_TIME is listed in the table.

SQL> col limit format a20

SQL> **select resource\_type, resource\_name, limit from dba\_profiles where profile='HRPROFILE' order by 1,2;**

RESOURCE RESOURCE\_NAME LIMIT

KERNEL COMPOSITE\_LIMIT DEFAULT

KERNEL CONNECT\_TIME DEFAULT

KERNEL CPU\_PER\_CALL DEFAULT

KERNEL CPU\_PER\_SESSION DEFAULT

KERNEL IDLE\_TIME 15

KERNEL LOGICAL\_READS\_PER\_CALL DEFAULT KERNEL LOGICAL\_READS\_PER\_SESSION DEFAULT KERNEL PRIVATE\_SGA DEFAULT

KERNEL SESSIONS\_PER\_USER DEFAULT

PASSWORD FAILED\_LOGIN\_ATTEMPTS DEFAULT

PASSWORD INACTIVE\_ACCOUNT\_TIME DEFAULT

PASSWORD PASSWORD\_GRACE\_TIME DEFAULT

PASSWORD PASSWORD\_LIFE\_TIME DEFAULT

PASSWORD PASSWORD\_LOCK\_TIME DEFAULT

PASSWORD PASSWORD\_REUSE\_MAX DEFAULT

PASSWORD PASSWORD\_REUSE\_TIME DEFAULT PASSWORD PASSWORD\_VERIFY\_FUNCTION DEFAULT

Answer: The results show a resource named INACTIVE\_ACCOUNT\_TIME, so INACTIVE\_ACCOUNT\_TIME is a valid profile limit. Therefore, the error must have something to do with the value that you are trying to set for the profile limit.

Investigate by displaying the full error message that you received in step 4. To do this, issue the oerr command for the error number ora 2377. Notice that the error states the limit cannot be less than 15 days.

Set an appropriate limit. Because 10 is too low, use the lowest valid number instead, which would be 15.

Query the DBA\_PROFILES view again to confirm that the limit is set.

Exit SQL\*Plus and close the terminal window.

Question: What will a DBA have to do if a database user account gets locked due to this new limit?

Answer: The DBA will have to unlock the database user account to make it available for use again by issuing the following command:

Practice 21-2: Using SQL\*Developer to Create Local Users

Overview

In this practice you use SQL\*Developer to connect to ORCLPDB1 as the PDBADMIN user and create local user accounts according to the following table. Assign the profile named HRPROFILE to the accounts as well as various privileges and roles that you've already created in previous practices. Afterward, test the accounts by logging in to SQL\*Plus as each user. Also, test the idle time setting in HRPROFILE.

Assumptions

You are currently logged in as the oracle user.

You created a local profile in ORCLPDB1 named HRPROFILE and two local roles (HRCLERK and HRMANAGER) in ORCLPDB1. You also assigned the DBA role to the PDBADMIN user, which is the local administrator for ORCLPDB1. If you haven't done so, complete the following practices before starting this one:

Practice 2-1 Granting the DBA Role to PDBADMIN

Practice 2-2 Creating Local Roles Using SQL\*Developer

Practice 3-1 Creating a Local Profile Using SQL\*Developer

Tasks

Connect to ORCLPDB1 as PDBADMIN Using SQL\*Developer

Launch SQL\*Developer from your Desktop. In the Connections box, double-click **PDB1-pdbadmin**.

Create a User Account for Jenny Goodman

In this section, you create a user account named JGOODMAN by using the SQL\*Developer interface.

In the DBA box, expand **PDB1-pdbadmin**.

Expand **Security** and select **Users.**

Click **Actions** > **Create New ..**.

The Create User dialog is displayed. On the User tab, enter or select the following values.

Name: Enter **JGOODMAN**

Enter the ***password*** and confirm. Refer to *Course Practice Environment: Security Credentials* for the password value.

Leave the check boxes for Password Expired, Operating System User, Account is Locked, and Edition Enabled deselected.

Default Tablespace: USERS

Temporary Tablespace: TEMP

On the Granted Roles tab, check the Granted box for **HRCLERK** role, and **HRMANAGER** role.

On the System Privileges tab, check the Granted box for **Create Session** privilege.

On the SQL tab, review the SQL statement, click **Apply**.

Note: The *password* value will be the one from *Course Practice Environment: Security Credentials*.

**Note:** In *Practice 3-3 Configuring a Default Role for a User*, you will assign the HRCLERK

role to be this user account's default role.

In the Successful dialog box, click **OK**.

Verify that JGOODMAN is listed in the Users table. From here, you can see that the

JGOODMAN account is unlocked and the password has not expired.

Create a User Account for David Hamby

In this section, you create another user account named DHAMBY by using the SQL\*Developer interface. While creating this user, you copy the SQL code to a text file so that in the next section, you can create more users by running a script.

If the Users tab is not open, expand **Security** and then select **Users**.

Click **Actions > Create New ..**.

The Create User dialog is displayed. On the User Account page, enter or select the following values.

Name: Enter **DHAMBY**

Set the *password* and confirm. See *Course Practice Environment: Security Credentials*

for password values.

Select the Password Expired check box to force the user to change his password at logon.

Default Tablespace: **USERS**

Temporary Tablespace: **TEMP**

On the Granted Roles tab, select **HRCLERK** role and check **Granted**.

On the System Privileges tab, select the **CREATE SESSION** privilege, and check **Granted**.

Click **SQL** tab.

Create a SQL script that contains the SQL statements displayed in the previous step. Turn the username and role values into substitution variables, rather than hard-coding them. This script will be used to create future users.

Tip: You can create substitution variables in SQL scripts by using single ampersands (&) and/or double ampersands (&&). A single ampersand indicates to SQL\*Plus to prompt you to enter a value each time the substitution variable occurs in the script. A double ampersand indicates to SQL\*Plus to prompt you to enter a value only once for a substitution variable and use that same value for all occurrences of the variable in the script.

Copy the SQL statements in the previous step to the clipboard.

Click **Apply** to create the DHAMBY user.

Click **OK** in the Successful box.

Click the **PDB1-pdadmin** tab to view the worksheet.

**Paste** the SQL statements in the worksheet. Change every occurrence as follows:

Verify that the code looks like the following code. Don't worry if your GRANT statements are in a different order. See *Course Practice Environment: Security Credentials* for password values.

Save this script. Click the Save file Icon or **ctrl-S**.

In the Save dialog, Browse to **/home/oracle/labs**. In the File Name box, enter

**CreateHRUser.sql**. Click **Save**. The file is saved and formatted.

Verify that **DHAMBY** is listed in the Users table. The account is unlocked and the password has expired.

Minimize, but don't close, SQL\*Developer.

Create a User Account for Rachel Pandya by Using a Script

In this section, you create another user account named RPANDYA. Rather than use the SQL\*Developer interface to create this user, you use the SQL script that you generated in the previous section.

Source the oraenv script for the orclcdb database.

Start SQL\*Plus and connect to ORCLPDB1 as the PDBADMIN user. See *Course Practice Environment: Security Credentials* for ***password*** values.

Execute the **CreateHRUser.sql** script. Enter **RPANDYA** when prompted for the username. Enter the password for the user from the *Course Practice Environment: Security* Credentials when prompted for the password. Enter **HRCLERK** when prompted for the role. The order of the GRANT statements does not matter.

Return to SQL\*Developer and click the **Refresh** button in the **Users** tab to refresh the data then scroll down the list and verify that the user RPANDYA has been created as expected.

Close the SQL\*Develop window.

Test DHAMBY's Access in SQL\*Plus

Connect to ORCLPDB1 as the DHAMBY user. Select the row with employee\_id=197 from the HR.EMPLOYEES table. Then attempt to delete it. You should get the “insufficient privileges” error. This happens because in *Practice 2-2 Using SQL\*Developer to Create Local Roles*, you granted DHAMBY the HRCLERK role, which has SELECT and UPDATE privileges on the HR.EMPLOYEES table, not INSERT and DELETE.

No need to test for RPANDYA as this user has the same role as DHAMBY.

Return to the terminal window.

Connect to ORCLPDB1 as DHAMBY. When prompted, enter the new ***password***. See *Course Practice Environment: Security Credentials* for the original and new ***password***. When you enter the new password, it is not displayed in the interface.

View the salary for employee 197 from the HR.EMPLOYEES table. The query returns a value of 3000 for SALARY.

Now attempt to delete the same row from the HR.EMPLOYEES table. DHAMBY is not allowed to perform DELETE operations on this table; therefore, the query returns an "insufficient privileges" error message.

Disconnect from ORCLPDB1.

Test JGOODMAN's Access in SQL\*Plus

Repeat the test that you just did with DHAMBY with the JGOODMAN user account. After deleting the row, issue a ROLLBACK, so that you still have the original 107 rows.

Connect to ORCLPDB1 as JGOODMAN. Refer to *Course Practice Environment: Security Credentials* for the ***password*** value. When creating this user, you did not expire the password, so you won't have to change the password here.

Select the salary for employee 197 from the HR.EMPLOYEES table. The query returns a value of 3000 for SALARY.

Delete the same row from the HR.EMPLOYEES table. JGOODMAN has the HRMANAGER role, and that role is granted SELECT, INSERT, UPDATE, and DELETE privileges on all tables in the HR schema. Therefore, the row is deleted.

Roll back the delete operation because this was just a test.

Confirm that you still have 107 rows in the HR.EMPLOYEES table.

Disconnect from ORCLPDB1.

Test the Idle Time Limit in HRPROFILE

If you recall, in *Practice 3-1 Using SQL\*Developer to Create a Local Profile*, you created a profile named HRPROFILE. In that profile, you configured the Idle Time limit to be 15 minutes. Assign this profile to all three users (JGOODMAN, DHAMBY, and RPANDYA). In this section, test that limit by connecting to ORCLPDB1 as RPANDYA and letting the session remain inactive for more than 15 minutes. After 15 minutes, verify that RPANDYA was automatically logged out by performing an operation; for example, try to select from the HR.EMPLOYEES table. While you're waiting, you can continue on to the next practice.

Connect to ORCLPDB1 as PDBADMIN and assign the profile named HRPROFILE to all three users (JGOODMAN, DHAMBY, and RPANDYA). Refer to *Course Practice Environment: Security Credentials* for the ***password*** value

Connect to ORCLPDB1 as RPANDYA. When prompted, enter the new password. Refer to *Course Practice Environment: Security Credentials* for the ***password*** value. When you enter the new password, it is not displayed in the interface.

Wait for 15 minutes. You can leave this terminal window open while waiting.

After 15 minutes, query the salary for employee 197 from the HR.EMPLOYEES table. The query returns the message "exceeded maximum idle time..." which indicates that HRPROFILE is working.

Exit SQL\*Plus

Exit all terminals.

Practice 21-3: Configuring a Default Role for a User

Overview

In this practice, PDBADMIN configures HRCLERK as the default role for JGOODMAN (user account for Jenny Goodman in ORCLPDB1). Jenny logs in to ORCLPDB1 and views the privileges that she gets from her default role. She requires more privileges to perform her management tasks, so she enables her non-default role, HRMANAGER, and views her new set of privileges.

Assumptions

You are currently logged in as the oracle user.

You created the user account called JGOODMAN and granted the HRMANAGER role to it, as well as the less-privileged HRCLERK role. To complete this practice, you must first complete the following practices:

Practice 2-1 Granting the DBA Role to PDBADMIN

Practice 2-2 Practice

Tasks

Configure a Default Role for JGOODMAN

Open a new terminal, use the oraenv command to source the orclcdb database, then start SQL\*Plus and connect to ORCLPDB1 as the PDBADMIN user. Refer to *Course Practice Environment: Security Credentials* for the ***password*** value.

View the current roles for JGOODMAN by querying the DBA\_ROLE\_PRIVS view. Also, show whether the roles are default roles. The results show that JGOODMAN is granted two roles, HRMANAGER and HRCLERK, and both are default roles (the DEF column = YES).

Set the default role for JGOODMAN to be HRCLERK only by using the ALTER USER command and DEFAULT ROLE clause.

View the current roles and default role settings for JGOODMAN again by querying the DBA\_ROLE\_PRIVS view. The results show that the default role is HRCLERK and the HRMANAGER role is no longer a default role. Jenny still has this role; however, she'll need to enable it to exercise its privileges.

Disconnect PDBADMIN from ORCLPDB1.

Enable a Non-Default Role

Connect to ORCLPDB1 as JGOODMAN. Refer to *Course Practice Environment: Security Credentials* for the ***password*** value.

View the roles for the current session. Notice that the default role, HRCLERK, is in effect.

Suppose JGOODMAN needs to operate as an HR Manager, and not an HR Clerk. Change the enabled role to HRMANAGER. Caution: If you use the SET ROLE command, any roles not included in the command will be disabled.

View the roles for the current session again. The HRMANAGER role is now enabled.

Suppose JGOODMAN needs both roles. Use the SET ROLE command to enable them both.

View the roles for the current session again. The HRMANAGER and HRCLERK roles are now in effect.

Exit SQL\*Plus.

Exit all terminals.