# Practices for Lesson 20: Configuring Privilege and Role Authorization

Practices for Lesson 20: Overview

Overview

In these practices, you will create roles. You will grant privileges to roles.

Practice 20-1: Granting a Local Role (DBA) to PDBADMIN

Overview

In this practice, you examine the default privileges and roles granted to the PDBADMIN user. PDBADMIN was created when the CDB and ORCLPDB1 were created. This user is intended to operate as the local PDB administrator.

After exploring, you grant PDBADMIN more power with the DBA role so that in later practices

PDBADMIN is able to create profiles, roles, and users.

Assumptions

You are logged in to the host machine as the oracle user.

It is assumed that the database and listener are running. You can use the pgrep -lf smon command to verify that the database is started and the pgrep -lf tns command to verify that the listener has started. If you need to restart the database and listener, use the dbstart.sh script.

Tasks

Explore the Privileges and Roles Granted to PDBADMIN

Open a new Terminal, set the environment to ORCLCDB, start SQL\*Plus, and connect as the SYS user with the SYSDBA privilege.

Note: PDBADMIN does not have the required privileges to view data from the

DBA\_SYS\_PRIVS view in ORCLPDB1, which you will do in the next step.

List the system privileges granted to the PDBADMIN user by querying the DBA\_SYS\_PRIVS view. This view describes system privileges granted to users and roles. The results show that PDBADMIN has not been granted any privileges directly. However, there may be privileges granted through roles.

List the roles granted to the PDB1\_ADMIN user by querying the CDB\_ROLE\_PRIVS view. This view describes the roles granted to all users and roles in the database. The results show that PDBADMIN is granted the PDB\_DBA role. Also, the ADMIN OPTION is enabled (ADM=YES), which means that PDBADMIN can grant the PDB\_DBA role to other users.

List the system privileges granted to the PDB\_DBA role by querying the ROLE\_SYS\_PRIVS

view.

Query the ROLE\_SYS\_PRIVS view. This view describes system privileges granted to roles. Information is provided only about roles to which the user has access. Because you're connected to ORCLPDB1 as the SYS user, you have access to all role information. The results show that the PDB\_DBA role consists of two system privileges: CREATE SESSION and CREATE PLUGGABLE DATABASE.

List the roles that are granted to the PDB\_DBA role by querying the DBA\_ROLE\_PRIVS view. The results show that the PDB\_DBA role is granted the CONNECT role.

List the privileges granted to the CONNECT role by querying the ROLE\_SYS\_PRIVS view. The results show that the CONNECT role consists of the SET CONTAINER and CREATE SESSION privileges.

Let's summarize our findings: From these queries, you learned that the PDBADMIN user is granted the PDB\_DBA role by default, and that role consists of the CONNECT role and the CREATE PLUGGABLE DATABASE system privilege. The CONNECT role contains the SET CONTAINER and CREATE SESSION system privileges.

Grant the DBA Role to PDBADMIN

Grant the DBA role locally to PDBADMIN.

List the roles that are granted to PDBADMIN by querying the DBA\_ROLE\_PRIVS view. The results show that PDBADMIN is now granted the DBA and PDB\_DBA roles.

Exit SQL\*Plus and close the terminal window.

Close all terminals

Practice 20-2: Using SQL\*Developer to Create Local Roles

Overview

In this practice, the PDBAMIN user uses SQL\*Developer to create the following local roles in

ORCLPDB1:

HRCLERK: Grant this role the SELECT and UPDATE object privileges on the EMPLOYEES

table in the HR schema.

HRMANAGER: Grant this role the SELECT, UPDATE, INSERT, and DELETE object privileges on the entire HR schema.

You will assign these roles to local users in Practice 3-2 Using SQL Developer to Create Local Users.

Assumptions

You are currently logged in as the oracle OS user.

You completed Practice 2-1 Granting the DBA Role to PDBADMIN.

Tasks

Log in to SQL\*Developer (PDB1)

Create a connection to ORCLPDB1 as the PDBADMIN.

Start SQL\*Developer. The SQL\*Developer icon is on the desktop.

In the Connections pane on the left, click the Name: **PDB1-pdbadmin** (a connection for ORCLPDB1 as PDBADMIN)

In the DBA connection box, click the connection: **PDB1-pdbadmin**

Create the HRCLERK Role

Expand the PDB1-pdbadmin connection in the DBA box.

Expand **Security** and then select **Roles**.

In the Roles tab, select **Actions > Create New …**

In the Create Role box,

On the **Roles** tab, enter Role Name: **HRCLERK**.

Click the **SQL** tab to view the SQL statement.

Click **Apply**.

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

In the Create Role box, click **Close**.

Verify that the **HRCLERK** role is listed in the table. Note you may have to click on refresh

Add Object Privileges to HRCLERK role

In the Connections box, expand **PDB1-pdbadmin> Other USERS**> **HR** > **Tables,**

then click **Employees.**

In the **EMPLOYEES** tab on the right, click the **Grants** subtab.

Click Actions > Privileges > Grant.

The Perform Grant Action box is displayed.

In the Users/Roles drop-down list, select **HRCLERK**.

In the left hand list, click **SELECT** and move it to right hand list.

In the left hand list, click **UPDATE** and move it to right hand list.

Click the SQL tab to view the SQL statement.

Click **Apply.**

In the Successful box click **OK.**

Refresh on the Grants subtab and verify SELECT and UPDATE on HR.EMPLOYEES

have been granted to HRCLERK.

Create the HRMANAGER Role

The steps in this section are similar to the ones in the previous section.

Expand the PDB1-pdbadmin connection in the DBA box.

Expand **Security** and then select **Roles**.

In the Roles tab, select **Actions > Create New …**

In the Create Role box,

On the Roles tab, enter Role Name: **HRMANAGER**.

Click the SQL tab to view the SQL statement.

Click **Apply**.

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

In the Create Role box, click **Close**.

Verify that the **HRMANAGER** role is listed in the table.

Add Object Privileges to HRMANAGER role

In the Connections box, expand **Other USERS**> **HR> Tables,** and click **COUNTRIES.**

Click the **Grants** subtab.

Click Actions > Privileges > Grant.

The Perform Grant Action box is displayed.

In the Users/Roles drop-down list, select **HRMANAGER**.

In the left hand list, click **DELETE** and move it to right hand list.

Repeat for **INSERT**,**SELECT**, and **UPDATE**.

Click the SQL tab to view the SQL statement.

Highlight the SQL statement, and copy it (use the middle mouse button or ctrl-V).

Cancel the Perform Grant Action box, click **Cancel**.

Click the **PDB1-pdbadmin** tab for the Worksheet. Note if you closed the tab, you can launch a new Worksheet using ALT-F10 or select from the SQL Developer menu **Tools** > **SQLWorksheet**

Paste the SQL Statement into the worksheet (use the middle mouse button or ctrl-V).

Press **Crtl-Enter** or the Green right arrow to execute the statement.

Change **COUNTRIES** in the statement to **EMPLOYEES**, and press the Green right arrow in the menu to execute. Note: since the table name is in quotes, upper case is required.

Repeat changing the table name to **EMPLOYEES, DEPARTMENTS, JOBS, JOB\_HISTORY, LOCATIONS, REGIONS .** Execute all the statements by clicking run script button. Note the script output window:

Verify the HRMANAGER role and required privileges.

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

Double-click the **HRMANAGER** role in the view on the right.

In the **HRMANAGER** tab, click the **Object Privs** subtab.

Verify the privileges granted on the HR tables are shown.

Exit SQL\*Developer by clicking **File** > **Exit**