# Practices for Lesson 19: Creating and Managing User Accounts

Practices for Lesson 19: Overview

Overview

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

Practice 19-1: Creating Common and Local Users

Overview

In this practice, you log on to the database in SQL\*Plus as the SYS user and create two types of administrators:

CDB administrator named c##CDB\_ADMIN1: Create this user as a common user so that it exists in every container in the CDB. Grant this user the most powerful administrator privilege, the SYSDBA privilege, in all containers. This privilege enables c##CDB\_ADMIN1 to access containers whether they are open or not. Because most database operations don't require the SYSDBA privilege, also grant this user the DBA role and CREATE SESSION privilege in all containers so that the user can operate as a regular user too.

ORCLPDB1 administrator named PDB1\_ADMIN: Create this user as a local user in ORCLPDB1 and grant this user the DBA role and CREATE SESSION privilege. This grant will provide the necessary system and object privileges. All tasks required by this user must be performed on an open PDB.

Tip

It's good practice to create a user separate from SYS and SYSTEM to perform database administration tasks. Each DBA in your organization should have his or her own privileged account to aid in auditing. Keep in mind that when you connect with the SYSDBA privilege, the database shows you logged in as the SYS user, regardless of your actual username. Audit trails, however, will show your real username.

Organizations that need to implement the tightest security possible separate the database duties and create many accounts for each database administrator (DBA) distinctly named and use the security principle of least privileges. Only the minimum privileges needed to perform a job are given. If an administrator doesn't need access to data but still performs maintenance operations, you can grant that user the SYSOPER privilege instead. Also consider other administrative privileges, such as SYSDG, SYSKM, SYSBACKUP, and SYSRAC, as necessary.

Assumptions

You are logged in as the oracle user.

The ORCLPDB2 pdb exists and is pristine. ORCLPDB2 will be used to refresh

ORCLPDB1.

Tasks

**Reset** ORCLPDB1 and common users.

Open terminal and source the environment variables to ORCLCDB

Execute /home/oracle/labs/DBMod\_UsersSec/reset\_ORCLPDB1.sh.

**Note:** Error messages saying object does not exist can be ignored.

Reset common users.

**Note:** Error messages saying object does not exist can be ignored.

Create c##CDB\_ADMIN1

Start SQL\*Plus and connect to the root container as the SYS user with the SYSDBA

privilege. This method of connecting uses OS authentication.

Create a common user named c##CDB\_ADMIN1 by using the CREATE USER command. Set the USERS tablespace as the default and TEMP as the temporary tablespace. Also, unlock the account so that c##CDB\_ADMIN1 can log in right away.

**Important**! To create a common user, you must start the user name with c## or C##, and you must include the CONTAINER=ALL clause so that the user's identity and password are created in all the containers.

**Note**: The username in the Oracle database is NOT case sensitive, all user names are stored in uppercase. Refer to *Course Practice Environment: Security Credentials* for the ***password*** value.

Grant c##CDB\_ADMIN1 the DBA role, the CREATE SESSION privilege, and the SYSDBA

privilege in all containers. This is an example of granting privileges and a role commonly.

**Question:** Can you use the following statement to complete the same operation (granting privileges and a role commonly)?

**Answer:** No, because without the CONTAINER=ALL clause, the CREATE SESSION privilege and DBA role are granted locally (in the root container only) to c##CDB\_ADMIN1, and not to each c##CDB\_ADMIN1 user in each PDB.

List the common users by querying the DBA\_USERS view. Scroll down and verify that

c##CDB\_ADMIN1 is included.

Compare Exercising and Not Exercising the SYSDBA Privilege

This section compares logging in as the c##CDB\_ADMIN1 user with and without the SYSDBA

privilege.

Disconnect from the root container.

Show the current user by issuing the SHOW USER command. You are not connected as any user,.

Connect to the root container as *anyuser/anystring* and exercise the SYSDBA privilege. **Note:** If you are connected to the OS as a user that is a privileged database user, that is a member of the DBA group, you can enter anything as a user name and anything as a

password and be connected.

Show the current container name.

Show the current user. The current user is SYS, which means the c##CDB\_ADMIN1 user can now do anything that the SYS user can do.

**Note:** Audit trails will show the / user with the SYSDBA privilege, not SYS.

View the list of privileges for the c##CDB\_ADMIN1 user by querying the SESSION\_PRIVS

static data dictionary view. Scroll down to view the privileges listed.

Disconnect from the root container.

Connect to the root container as c##CDB\_ADMIN1 again, but this time, do not exercise the SYSDBA privilege. Refer to *Course Practice Environment: Security Credentials* for the ***password*** value.

Show the current user. You are connected as c##CDB\_ADMIN1. Because you included the CONTAINER=ALL clause when granting the CREATE SESSION privilege and DBA role, c##CDB\_ADMIN1 can connect as a regular user to any open PDB and perform system and object operations that the DBA role allows.

View the list of privileges for the c##CDB\_ADMIN1 user by querying the SESSION\_PRIVS static data dictionary view. Scroll through the list of privileges. Notice that there are fewer privileges listed than when c##CDB\_ADMIN1 was connected with the SYSDBA privilege.

Switch to ORCLPDB1 by issuing the ALTER SESSION command.

Show the current container. It is ORCLPDB1.

Create the PDB1\_ADMIN User

You just connected to ORCLPDB1 as c##CDB\_ADMIN1. You need to be logged into ORCLPDB1 to create a local administrator for ORCLPDB1. The c##CDB\_ADMIN1 user can create the PDB1\_ADMIN user.

Create a local user named PDB1\_ADMIN by using the CREATE USER command. Set the USERS tablespace as the default and TEMP as the temporary tablespace. Also, unlock the account so that PDB1\_ADMIN can log in right away. Because this is a local user and not a common user, do not include the CONTAINER=ALL clause. Refer to *Course Practice Environment: Security Credentials* for the ***password*** value.

Grant PDB1\_ADMIN the DBA role and the CREATE SESSION privilege in ORCLPDB1 only. This is an example of granting a privilege and role locally.

List the local user accounts for ORCLPDB1 by querying the DBA\_USERS view. The

PDB1\_ADMIN account is included in the list.

**Note:** The PDBADMIN user was created when ORCLPDB1 was created.

Disconnect c##CDB\_ADMIN1 from PDB1.

Connect to ORCLPDB1 as PDB1\_ADMIN. Refer to *Course Practice Environment: Security Credentials* for the ***password*** value.

Show the current user. You are connected as PDB1\_ADMIN1.

View the list of privileges for PDB1\_ADMIN by querying the SESSION\_PRIVS view. The results below are only some of the privileges returned from the query.

Try to connect to the root container as the PDB1\_ADMIN user. This user does not have the SET CONTAINER privilege and does not exist in the root container, and therefore, you get an error message stating that the user has insufficient privileges. The c##CDB\_ADMIN1 user has the DBA role and CREATE SESSION privileges in all containers, including the root container. PDB1\_ADMIN has the same role and privilege, but only in PDB1.

Exit SQL\*Plus.

Exit all terminals.

Practice 19-2: Creating a Local User for an Application

Overview

In this practice, you log in to ORCLPDB1 as the local administrator (PDB1\_ADMIN) and create a local user account called INVENTORY, which will own the new Inventory software application. INVENTORY is an example of a user account that does not represent a person.

Assumptions

You are logged in to the compute node as the oracle user.

Tasks

Create the INVENTORY User Account

Open a terminal and use oraenv to set the environment to ORCLCDB. Then start SQL\*Plus and connect to ORCLPDB1 as the PDB1\_ADMIN user. Refer to *Course Practice Environment: Security Credentials* for the ***password*** value.

Create a local user account named INVENTORY. Set the default tablespace to the USERS tablespace and grant unlimited quota on that tablespace. Refer to *Course Practice Environment: Security Credentials* for the ***password*** value.

Grant the CREATE SESSION privilege to INVENTORY.

List the local user accounts for ORCLPDB1 by querying the DBA\_USERS view. The

INVENTORY account is included in the list.

Connect as INVENTORY and Verify Privileges

Disconnect PDB1\_ADMIN from ORCLPDB1.

Verify that the INVENTORY user account can connect to ORCLPDB1. Refer to *Course Practice Environment: Security Credentials* for the ***password*** value.

List the privileges for INVENTORY by querying the SESSION\_PRIVS view. The results show that INVENTORY has the CREATE SESSION privilege.

Exit SQL\*Plus.

Close all terminals.

Practice 19-3: Exploring OS and Password File Authentication

Overview

In this practice, you explore the OS and password file authentication.

Assumptions

You are currently logged in to the compute node as the oracle user.

Tasks

Exploring OS Authentication

During the course practices, you have logged in to the Oracle database as the oracle user and were authenticated using OS authentication. This section explores the groups and users in the Linux OS and how they are linked to authentication.

Open a new terminal & list the file: /etc/group

Linux and Unix operating systems have groups of users, and those are stored in the text file

/etc/group. Use the cat command to view the group file on the compute node. The format of each line is group\_name:password:Group ID (GID):user\_list. The groups you select to be the Oracle software owner (oinstall by default) and dba group must be created in the OS before the software is installed. Notice that these groups are included in the list below. The dba group consists of the oracle user.

To find out the user that you are currently logged in as, execute whoami. The result shows that you are currently logged in to the OS as the oracle user.

Find out more about the oracle user. For example, verify that oracle is part of the dba

group.

The /etc/passwd file is a text file that lists user account information needed for logging in to the OS. Execute the following command to search for the oracle user. The format of the row is user:password:user ID:primary group ID:home directory:login\_shell. Passwords are stored in the /etc/shadow file, so an x is used here as a placeholder.

The information above tells you that oracle's primary group ID is 54321. To find the name of that group, search for it in the group file. The result shows that the oracle user's primary group is the oinstall group.

Investigate further. Search for oracle in the group file. The results tell you that oracle is a user in the dba group. The dba group is the Database Administrator Group, and any user in this group has the SYSDBA system privilege. So, if you log on to the Oracle database by using OS authentication and exercise the SYSDBA privilege, then the oracle user becomes the SYS user. If you recall, to log on using OS authentication, all you need to specify is CONNECT / AS SYSDBA. The / tells SQL\*Plus to look up the privileges for the OS user's group.

An alternate way to get all of this information for the current user in a single command is the command id.

Exploring Password Authentication

When you grant an administrative privilege to a user, for example, SYSDBA or SYSOPER, that user's name and privilege information are added to the database password file. The V$PWFILE\_USERS view contains information about users that have been granted administrative privileges.

Start SQL\*Plus and connect to the root container as the SYS user with the SYSDBA

privilege.

View the columns in the V$PWFILE\_USERS view by issuing the DESCRIBE command.

List the users in the password file by querying the V$PWFILE\_USERS view.

Find out the SYS user's account status and whether the SYS user has the SYSDBA privilege by querying the V$PWFILE\_USERS view. ACCOUNT\_STATUS shows if the administrative user is OPEN, LOCKED (the user can no longer connect), or EXPIRED (the user must change the password at the connection).

Exit SQL\*Plus and close the terminal window.

Close all terminals.