Practice 5-1: Start Oracle Database

Overview

In this practice, you will start oracle database which has been already created as follows:

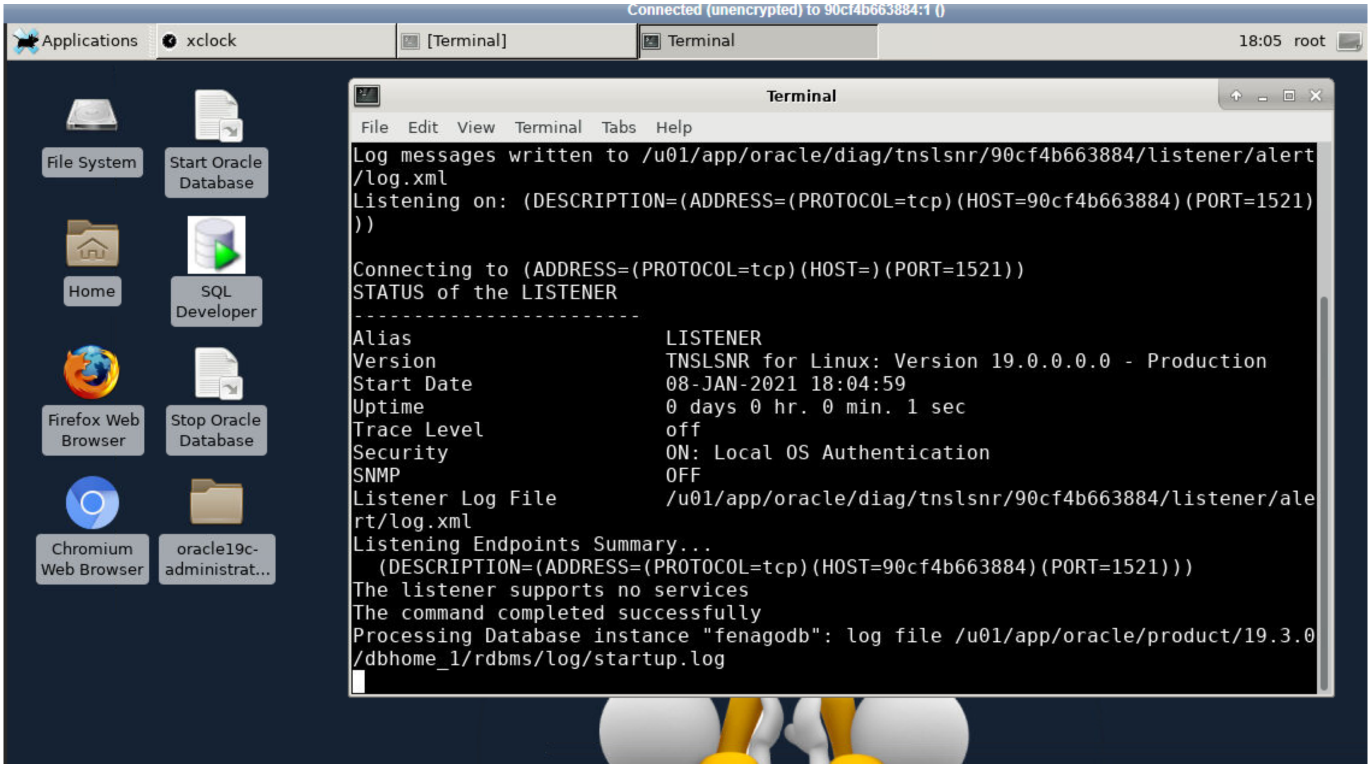
ORACLE_SID=fenagodb

ORACLE_PDB=fenagodb1

ORACLE_PWD=fenago

Tasks

1. Log in to your lab environment and double click “Start Oracle Database” shortcut.



```
Log messages written to /u01/app/oracle/diag/tnslsnr/90cf4b663884/listener/alert
/log.xml
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=90cf4b663884)(PORT=1521)
))

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 19.0.0.0.0 - Production
Start Date                08-JAN-2021 18:04:59
Uptime                    0 days 0 hr. 0 min. 1 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Log File         /u01/app/oracle/diag/tnslsnr/90cf4b663884/listener/ale
rt/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=90cf4b663884)(PORT=1521)))
The listener supports no services
The command completed successfully
Processing Database instance "fenagodb": log file /u01/app/oracle/product/19.3.0
/dbhome_1/rdbms/log/startup.log
```

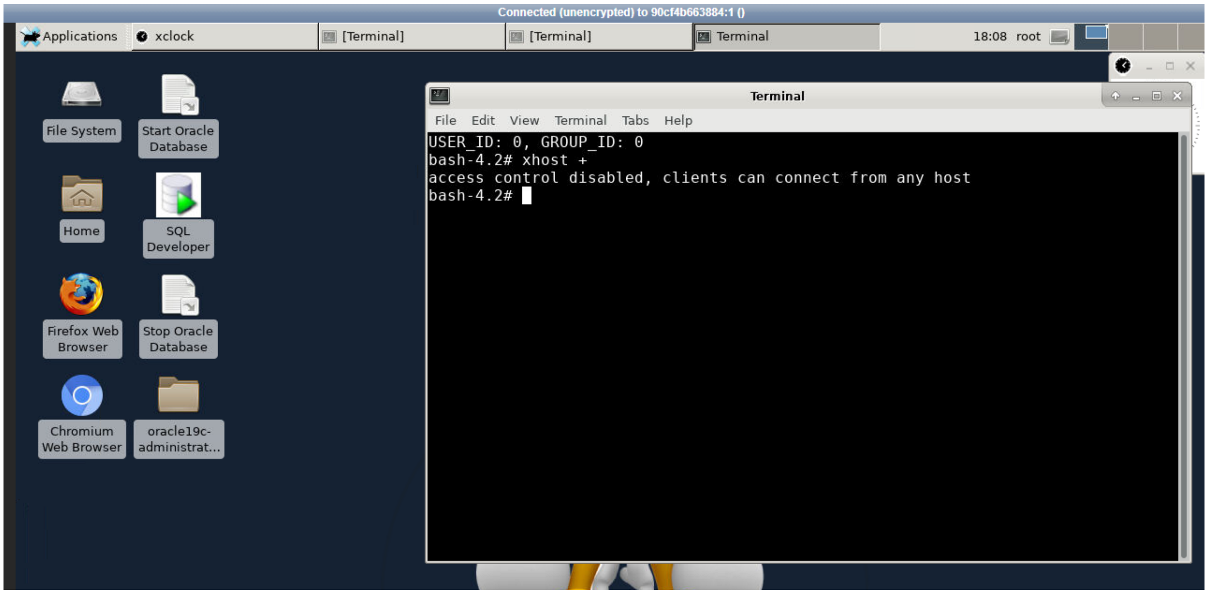
Practice 5-2: Switch to oracle user from terminal

Overview

In this practice, you will switch to oracle user from terminal

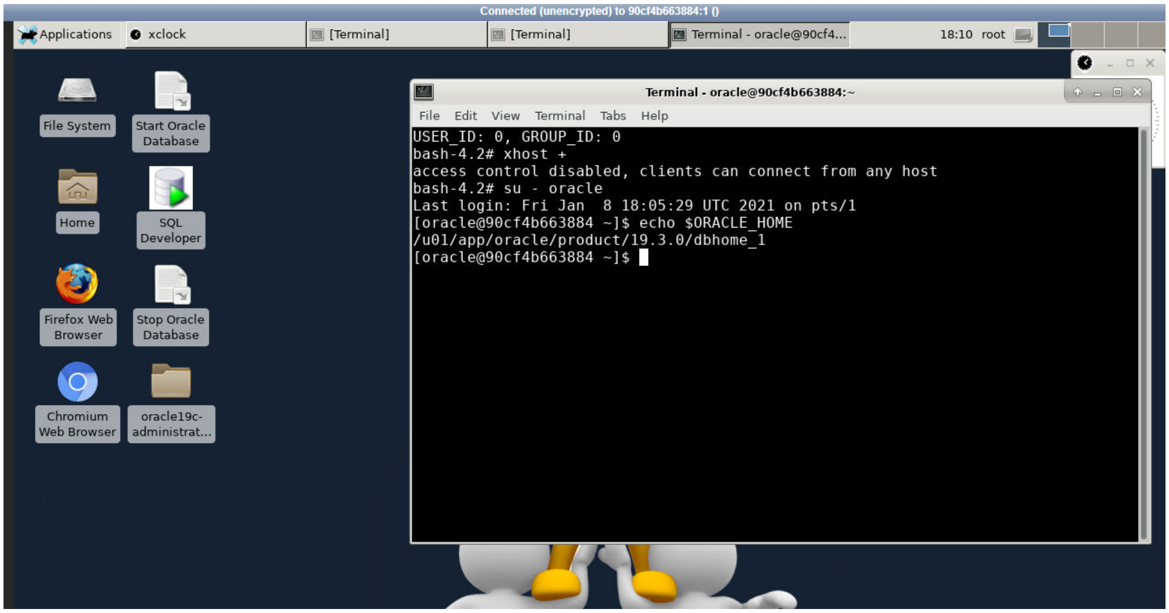
Tasks

a. Open terminal and run “xhost +” command as root user:



```
USER_ID: 0, GROUP_ID: 0
bash-4.2# xhost +
access control disabled, clients can connect from any host
bash-4.2#
```

b. Run and run “su - oracle” command in the terminal to switch to **oracle** user:



```
USER_ID: 0, GROUP_ID: 0
bash-4.2# xhost +
access control disabled, clients can connect from any host
bash-4.2# su - oracle
Last login: Fri Jan  8 18:05:29 UTC 2021 on pts/1
[oracle@90cf4b663884 ~]$ echo $ORACLE_HOME
/u01/app/oracle/product/19.3.0/dbhome_1
[oracle@90cf4b663884 ~]$
```

Practice 5-3: Exploring a CDB by Using SQL*Plus

Overview

In this practice, you will learn how to do the following things:

- Set the Oracle environment variables
- Connect to the root container by using SQL*Plus
- Query the data dictionary to view information about the containers, data files, users, instance, and services in a CDB
- List the services created automatically for each container

Some things to remember when you want to query the data dictionary for multiple PDBs or the whole CDB:

- Log in to the root container as a common user. A CDB common user is a database account created in the root container and is inherited by all PDBs in the CDB.
- Query container data objects, such as views whose names begin with V$ and CDB_.

For more information, refer to the following sections in *Oracle Database Administrator's Guide*:

- About Viewing Information When the Current Container is the CDB Root
- Viewing Information About the Containers in a CDB

In some of the steps below, you will format columns by using the `COLUMN` command. For example, applying the format `A55` specifies an alphabetic format of 55 characters wide. Format `999` is an example of a numeric format.

Commands in the practices are in uppercase and variables are in lower case. Any commands that you need to enter are bolded, for example:

```
SQL> SELECT regions FROM hr.departments;
```

Assumptions

You are connected to the compute node as the `oracle` user. See Practice 5-2 for detail.

Tasks

1. Set the Oracle environment variables. You need to set these each time you open a new terminal window.

   a. In the terminal window, list the search path that holds the `oraenv`script.

   ```
   [oracle@MYDBCS ~]$ which oraenv
   /u01/app/oracle/product/19.3.0/dbhome_1/bin/oraenv
   [oracle@MYDBCS ~]$
   ```

   b. Source the `oraenv` script. `oraenv` sets the required environment variables needed for you to connect to your database instance. The `oraenv` script sets the `ORACLE_SID` and `ORACLE_HOME` environment variables and includes the `$ORACLE_HOME/bin` directory in the `PATH` environment variable setting. Environment variables that this script sets will persist in the terminal window until you close it. For the ORACLE_SID value, enter ORCL.

```
[oracle@MYDBCS ~]$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base has been set to /u01/app/oracle
[oracle@MYDBCS ~]$
```

c. View the environment variables set by the oraenv command.

```
[oracle@MYDBCS ~]$ set | grep ORACLE
OLD_ORACLE_BASE=/u01/app/oracle
ORACLE_BASE=/u01/app/oracle
ORACLE_HOME=/u01/app/oracle/product/19.3.0/dbhome_1
ORACLE_HOSTNAME=MYDBCS.compute-588436052.oraclecloud.internal
ORACLE_SID=ORCL
ORACLE_UNQNAME=ORCL
[oracle@MYDBCS ~]$
```

Note: Remember that from this point on, each time you open a terminal window you will need to source the oraenv script to set the environment variables for your CDB.

2. Connect to the root container by using SQL*Plus.

   a. Start SQL*Plus and log in to the root container of your CDB as the SYS user with the SYSDBA privilege. You can connect to a database without a password when you have a local connection (on the same machine) and the current operating system user is a member of the privileged OSDBAgroup.

```
[oracle@MYDBCS ~]$ sqlplus / as sysdba
SQL*Plus: Release 18.0.0.0.0 Production on Tue May 29  20:18:18
2018
Version 19.3.0.0.0

Copyright (c) 1982,  2017, Oracle.  All rights reserved.


Connected to:
Oracle Database 19c EE High Perf Release 18.0.0.0.0 - Production
Version 19.3.0.0.0
SQL>
```

   b. Verify that you are logged in to the root container as the SYS user by using the SHOW USER command.

```
SQL> SHOW user
USER is "SYS"
SQL>
```

3. View information about the containers in your CDB.

a. Verify that you have a container database by querying the V$DATABASE view. The NAME column should contain ORCL, the CDB column should contain YES, and the ID should be 0 (zero). A value of zero is used for rows containing data that pertain to the entire CDB. This value is also used for rows in non-CDBs.

```
SQL> SELECT name, cdb, con_id FROM  v$database;

NAME      CDB     CON_ID
--------- --- ----------
ORCL      YES          0

SQL>
```

b. Show the current container name. Because you're currently connected to the root container, the name should be CDB$ROOT.

```
SQL> SHOW con_name

CON_NAME
------------------------------
CDB$ROOT

SQL>
```

c. Show the current container ID. Because you're currently connected to the root container, the ID should be 1.

```
SQL> SHOW con_id

CON_ID
------------------------------
1

SQL>
```

d. Determine the version of Oracle Database by querying the V$VERSION view. This view displays version numbers of core library components in Oracle Database.

```
SQL> SELECT banner FROM v$version;

BANNER
----------------------------------------------------------------
Oracle Database 19c EE High Perf Release 18.0.0.0.0 - Production

SQL>
```

e. List all the containers in your CDB by querying the V$CONTAINERS view. The results should show three containers—the root container (CDB$ROOT), the seed PDB (PDB$SEED), and PDB1.

```
SQL> COLUMN name FORMAT A8
SQL> SELECT name, con_id FROM v$containers ORDER BY  con_id;

NAME         CON_ID
-------- ----------
CDB$ROOT          1
PDB$SEED          2
PDB1              3

SQL>
```

f. List the PDBs in the CDB by using the `SHOW` command. The result should show two PDBs—the seed PDB (`PDB$SEED`) and `PDB1`. You can also list PDBs by querying the `V$PDBS` view. The `SHOW` command includes information about the open mode of each PDB and whether the PDB is restricted. The open mode for a PDB determines what type of activities a PDB will allow at that time. `PDB$SEED` is in `READ ONLY` mode and `PDB1` is in `READ WRITE` mode. The `RESTRICTED` column indicates whether only users possessing the `RESTRICTED SESSION` privilege can connect to the PDB.

```
SQL> SHOW pdbs

    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY  NO
         3 PDB1                           READ WRITE NO
SQL>
```

g. View the status of all PDBs in the CDB by querying the `CDB_PDBS` view. The status of a PDB describes the state of the PDB. For example, if the PDB is new, but never opened, the status is `NEW`. If it is available and ready for use, the status is `NORMAL`.

```
SQL> COLUMN pdb_name FORMAT A8
SQL> SELECT pdb_name, status FROM cdb_pdbs ORDER BY 1;

PDB_NAME STATUS
-------- ----------
PDB1     NORMAL
PDB$SEED NORMAL

SQL>
```

4. View information about the data files in your CDB.

   a. List all the data files in the CDB (for the root container and all PDBs) by querying the `CDB_DATA_FILES` view. The order of your results may vary.

```
SQL> COLUMN file_name FORMAT A50
SQL> COLUMN tablespace_name FORMAT A10
SQL> SELECT file_name, tablespace_name FROM cdb_data_files;

FILE_NAME                                            TABLESPACE
--------------------------------------------------   ----------
/u02/app/oracle/oradata/ORCL/users01.dbf             USERS
/u02/app/oracle/oradata/ORCL/undotbs01.dbf           UNDOTBS1
/u02/app/oracle/oradata/ORCL/system01.dbf            SYSTEM
/u02/app/oracle/oradata/ORCL/sysaux01.dbf            SYSAUX
/u02/app/oracle/oradata/ORCL/PDB1/system01.dbf       SYSTEM
/u02/app/oracle/oradata/ORCL/PDB1/sysaux01.dbf       SYSAUX
/u02/app/oracle/oradata/ORCL/PDB1/undotbs01.dbf      UNDOTBS1
/u02/app/oracle/oradata/ORCL/PDB1/PDB1_users01.dbf   USERS

8 rows selected.

SQL>
```

b. List all the tablespaces in the CDB (for both the root container and all the PDBs) by querying the `V$DATAFILE` and `V$TABLESPACE` views.

```
SQL> COL name FORMAT A12
SQL> SELECT d.file#, ts.name, ts.ts#,  ts.con_id
  2  FROM v$datafile d, v$tablespace ts
  3  WHERE d.ts#=ts.ts# AND  d.con_id=ts.con_id
  4  ORDER BY 4;

     FILE# NAME                TS#     CON_ID
---------- ------------ ---------- ----------
         1 SYSTEM                0          1
         3 SYSAUX                1          1
         4 UNDOTBS1              2          1
         7 USERS                 4          1
         6 SYSAUX                1          2
        13 USERS                 5          2
         8 UNDOTBS1              2          2
         5 SYSTEM                0          2
         9 SYSTEM                0          3
        10 SYSAUX                1          3
        11 UNDOTBS1              2          3
        12 USERS                 5          3
12 rows selected.

SQL>
```

c. List all temp files in the CDB (for the root container and all PDBs) by querying the `CDB_TEMP_FILES` view.

```
SQL> SELECT file_name, tablespace_name FROM cdb_temp_files;

FILE_NAME                                             TABLESPACE
----------------------------------------------------  ----------
/u04/app/oracle/oradata/temp/temp01.dbf               TEMP
/u02/app/oracle/oradata/ORCL/PDB1/pdbseed_temp0120 TEMP
18-02-19_18-48-12-642-PM.dbf

SQL>
```

d. List all the redo log files in the CDB (for the root container and all PDBs) by querying the `V$LOGFILE` view.

```
SQL> COLUMN member FORMAT A42
SQL> SELECT group#, member, con_id FROM v$logfile;

    GROUP# MEMBER                                         CON_ID
---------- ---------------------------------------------- ----------
         3 /u04/app/oracle/redo/redo03.log                         0
         2 /u04/app/oracle/redo/redo02.log                         0
         1 /u04/app/oracle/redo/redo01.log                         0

SQL>
```

e. List the control files in the CDB by querying the `V$CONTROLFILE` view. There should be two—`control01.ctl` and `control02.ctl`.

```
SQL> COLUMN name FORMAT A55
SQL> SELECT name, con_id FROM  v$controlfile;

NAME                                                      CON_ID
------------------------------------------------------- --------
/u02/app/oracle/oradata/ORCL/control01.ctl                      0
/u03/app/oracle/fast_recovery_area/ORCL/control02.ctl           0

SQL>
```

5. View information about the pre-created users in your CDB.

a. List only the common users in the CDB by querying the `CDB_USERS` view.

```
SQL> SELECT DISTINCT username FROM cdb_users
  2  WHERE common ='YES' ORDER BY 1;

USERNAME
------------------------------------------------------------------
```

```
ANONYMOUS
APPQOSSYS
AUDSYS
C##DBAAS_BACKUP
…
SYSTEM
WMSYS
XDB
XS$NULL

38 rows selected.

SQL>
```

b. List all the users in every PDB in the CDB by querying the `CDB_USERS` view. In the results, notice that the `SYS`, `SYSTEM`, and `PDBADMIN` user accounts are listed for `PDB1`. The root container's id is `1` and PDB1's id is `3`.

```
SQL> COLUMN username FORMAT A25
SQL> SELECT con_id, username FROM cdb_users
  2  ORDER BY username, con_id;

    CON_ID USERNAME
---------- -------------------------
         1 ANONYMOUS
         3  ANONYMOUS
         3 APEX_050100
         3 APEX_INSTANCE_ADMIN_USER
…
         1 OUTLN
         3 OUTLN
         3 PDBADMIN
         1  REMOTE_SCHEDULER_AGENT
         3  REMOTE_SCHEDULER_AGENT
…
         1 SYS
         3 SYS
…
         1 SYSRAC
         3 SYSRAC
         1 SYSTEM
         3 SYSTEM
…
84 rows selected.
```

```
SQL>
```

6. View information about the database instance and the services.

   a. View the database instance name, its status, and which container database it is associated with by querying the `V$INSTANCE` view. The instance's status is `OPEN`, which means users can access the CDB and PDB.

```
SQL> SELECT instance_name, status, con_id FROM  v$instance;

INSTANCE_NAME    STATUS           CON_ID
---------------- ------------ ----------
ORCL             OPEN                  0

SQL>
```

   b. List the services for all the containers in the CDB by querying the `V$SERVICES` view. The query returns five services. The `PDB$SEED` service is not listed because no one should connect to it and no operation should be performed with it. It is reserved as a template to create other PDBs.

```
SQL> SELECT con_id, name FROM v$services ORDER BY  1;

    CON_ID NAME
----------  ----------------------------------------------------
         1 SYS$BACKGROUND
         1 ORCL.588436052.oraclecloud.internal
         1 ORCL.588436052.oraclecloud.internalXDB
         1 SYS$USERS
         3 pdb1

SQL>
```

7. Exit SQL*Plus.

```
SQL > exit
Disconnected from Oracle Database 19c EE High Perf  Release
18.0.0.0.0 - Production
Version 19.3.0.0.0
[oracle@MYDBCS  ~]$
```

Practice 5-4: Exploring a PDB by Using SQL*Plus

Overview

In this practice, you will learn how to do the following things:

- Connect to a PDB indirectly through a CDB
- Query the data dictionary to view information about data files, temp files, and users in a PDB
- Connect to a PDB directly by using the Easy Connect syntax

To find data dictionary information specific to a root container or a PDB:

- Query `DBA_` views to return container-specific information.
- When you are logged in to a PDB, queries against the data dictionary return information about that PDB only, regardless of the view you query.
- When queried from a PDB, the `DBA_PDBS` view returns the information related to the PDB to which you are connected. When queried from the root container, the `DBA_PDBS` view provides information on all PDBs belonging to a given CDB.

Assumptions

You have a connection to the compute node through PuTTY or SSH and are logged in as the `oracle` user.

Tasks

1. Connect to `PDB1` indirectly through the root container.
   a. Start SQL*Plus and connect to the root container as the `SYS` user with the `SYSDBA` privilege. Oracle allows any DBA group user at the operating system level to log into SQL*Plus without any authentication.

   ```
   [oracle@MYDBCS ~]$ sqlplus / as sysdba

   …

   SQL>
   ```

   b. Verify that `PDB1` is open. After DBCA creates a PDB, it opens it automatically. The results below indicate that the open mode is `READ WRITE`, which means `PDB1` is open. PDB users with the `SYSDBA`, `SYSOPER`, `SYSBACKUP`, `SYSDG`, `SYSKM`, or `SYSRAC` privilege can connect to a closed PDB; however, all other PDB users can connect only when the PDB is open.

   ```
   SQL> COLUMN con_id FORMAT 999
   SQL> COLUMN name FORMAT A10
   SQL> SELECT con_id, name, open_mode FROM v$pdbs;

   CON_ID NAME       OPEN_MODE
   ```

```
------ ---------- ----------
     2 PDB$SEED   READ ONLY
     3 PDB1       READ WRITE

SQL>
```

c. If PDB1 is closed for some reason and its open mode was MOUNTED in the previous step, open it by using the ALTER PLUGGABLE DATABASE command.

```
SQL> ALTER PLUGGABLE DATABASE PDB1  OPEN;

Pluggable database altered.

SQL>
```

d. Switch to PDB1. When logged in to a CDB as an appropriately privileged user, you can use the ALTER SESSION command to switch between containers within the CDB. From this point on, your queries against the data dictionary will retrieve information for PDB1 only.

```
SQL> ALTER SESSION SET CONTAINER = PDB1;

Session altered.

SQL>
```

e. Verify that the container name is PDB1.

```
SQL> SHOW con_name

CON_NAME
------------------------------
PDB1
SQL>
```

2. Query the data dictionary to list the data files and temp files for PDB1.

   a. List the data files for PDB1 and the tablespaces to which they belong by querying the DBA_DATA_FILES view.

```
SQL> col file_name format a60
SQL> col tablespace_name format a10
SQL> SELECT file_name, tablespace_name FROM  dba_data_files;

FILE_NAME                                                    TABLESPACE
------------------------------------------------------------ ----------
/u02/app/oracle/oradata/ORCL/PDB1/system01.dbf               SYSTEM
/u02/app/oracle/oradata/ORCL/PDB1/sysaux01.dbf               SYSAUX
/u02/app/oracle/oradata/ORCL/PDB1/undotbs01.dbf              UNDOTBS1
```

```
/u02/app/oracle/oradata/ORCL/PDB1/PDB1_users01.dbf    USERS

SQL>
```

b. List the temp files for `PDB1` and the tablespaces to which they belong by querying the `DBA_TEMP_FILES` view.

```
SQL> SELECT file_name, tablespace_name FROM  dba_temp_files;

FILE_NAME                                                    TABLESPACE
---------------------------------------------------          ----------
/u02/app/oracle/oradata/ORCL/PDB1/pdbseed_temp012018-02-19_1 TEMP 8-
48-12-642-PM.dbf
SQL>
```

c. List the local users for `PDB1` by querying the `DBA_USERS` view.

```
SQL> SELECT DISTINCT username FROM dba_users WHERE common='NO';

USERNAME
--------------------------------------------------------------
PDBADMIN
APEX_LISTENER
APEX_PUBLIC_USER
APEX_REST_PUBLIC_USER
FLOWS_FILES
APEX_050100
APEX_INSTANCE_ADMIN_USER
SCOTT

8 rows selected.

SQL>
```

3. Make a direct connection to PDB1 by using the Easy Connect syntax. The Easy Connect syntax enables you to connect to the PDB without 1) requiring a connection to the root container and 2) having to set up a net service name for the PDB.

a. Disconnect from the PDB.

```
SQL > DISCONNECT
Disconnected from Oracle Database 19c EE High Perf  Release
18.0.0.0.0 - Production
Version 19.3.0.0.0
SQL>
```

b. Verify that you aren't connected as any user. The `SHOW user` command returns " " indicating that you are not connected.

```
SQL> SHOW user
```

```
USER is ""
SQL>
```

c. Connect to `PDB1` directly as the `SYSTEM` user by using the Easy Connect syntax. See *Course Practice Environment: Security Credentials* for the `SYSTEM` user password. In Practice 5-3, step 6b, you queried `V$SERVICES`. Append the value in the query results following `ORCL` to `pdb1` to create the service name as shown in this example.

```
SQL> CONNECT
system/password@localhost:1521/pdb1.588436052.oraclecloud.intern
al
Connected.
SQL>
```

d. Verify that you are now connected as the `SYSTEM` user by using the `SHOW USER` command again.

```
SQL> SHOW user
SQL> USER is "SYSTEM"
SQL>
```

4. Exit SQL*Plus.

```
SQL> EXIT
...
[oracle@MYDBCS ~]$
```

Practice 5-5: Installing the HR Sample Schema

Overview

In this practice, you will manually install the `HR` sample schema.

Assumptions

You have a connection to the compute node through PuTTY or SSH and are logged in as the `oracle` user.

Tasks

1. In your terminal window, navigate to the `$ORACLE_HOME/demo/schema/human_resources` directory.

```
[oracle@MYDBCS ~]$ cd  $ORACLE_HOME/demo/schema/human_resources
[oracle@MYDBCS human_resources]$
```

2. Use the `ls` command to view the contents of the `human_resources` directory. In a later step, you will execute the `hr_main.sql` to create the `HR` user, objects and load data into the `HR` tables.

```
[oracle@MYDBCS  human_resour ces]$ ls
hr_analz.sql   hr_comnt.sql   hr_drop_new.sql   hr_idx.sql
hr_main.sql
hr_code.sql    hr_cre.sql     hr_drop.sql       hr_main_new.sql
hr_popul.sql
[oracle@MYDBCS human_resources]$
```

3. Start SQL*Plus and connect to the root container as the `SYS` user with the `SYSDBA` privilege.

```
[oracle@MYDBCS human_resources]$ sqlplus / as sysdba
...
SQL>
```

4. Switch to `PDB1`.

```
SQL> ALTER SESSION SET CONTAINER = PDB1;
Session altered.

SQL>
```

5. Execute the **`hr_main.sql`** script and respond to the prompts as follows.

   a. Enter the password for the `HR` user as specified in the *Course Practice Environment: Security Credentials*.

   b. Enter **`USERS`** as the default tablespace for the `HR` user.

   c. Enter **`TEMP`** as the temporary tablespace for the `HR` user.

   d. Enter **`$ORACLE_HOME/demo/schema/log/`** for the log directory.

```
SQL> @hr_main

specify password for HR as parameter 1:
Enter value for 1: password

specify default tablespeace for HR as parameter 2:
Enter value for 2: USERS

specify temporary tablespace for HR as parameter 3:
Enter value for 3: TEMP

specify log path as parameter 4:
Enter value for 4: $ORACLE_HOME/demo/schema/log/

PL/SQL procedure successfully completed.

User created.

User altered.

Grant succeeded.
…

Comment created.

Commit complete.

PL/SQL procedure successfully completed.

SQL>
```

e. Exit from SQL*Plus.

```
SQL> exit
…
[oracle@MYDBCS human_resources]$
```

6. Query the `USER_TABLES` view as the `HR` user to verify that the user and tables were created.

   a. Connect as the `HR` user. Be sure to provide the correct service name for your PDB as you did in Practice 5-4, step 3c.

```
[oracle@MYDBCS human_resources]$ sqlplus
hr/password@localhost:1521/PDB1.588436052.oraclecloud.internal
…
SQL>
```

b. Query USER_TABLES.

```
SQL> SELECT table_name FROM user_tables;

TABLE_NAME
----------------------------------------------------------------
REGIONS
COUNTRIES
LOCATIONS
DEPARTMENTS
JOBS
EMPLOYEES
JOB_HISTORY

7 rows selected.

SQL>
```

7. Exit from SQL*Plus and close the connection to the compute node.

```
SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition  Release
18.0.0.0.0 - Production
Version 19.3.0.0.0
[oracle@MYDBCS human_resources]$ exit
```