## Lab 4.1: Creating a New CDB

### Objective:

To create a new Container Database (CDB) named `CDBDEV` using the `CREATE DATABASE` SQL command with specified characteristics.

### Pre-req

Start a NEW terminal shell and execute:

```
xhost +
```

Then create the directories needed (note that if you want a new DB - then replace CDBDEV with your new DB):
REMEMBER YOU ARE DOING THIS AS ROOT

```
mkdir -p /u01/app/oracle/oradata/CDBDEV
mkdir -p /u01/app/oracle/oradata/pdbseed
mkdir -p /u01/app/oracle/fast_recovery_area
mkdir -p /u01/app/oracle/admin/CDBDEV/adump

chown -R oracle:oinstall /u01/app/oracle/oradata
chown -R oracle:oinstall /u01/app/oracle/fast_recovery_area
chown -R oracle:oinstall /u01/app/oracle/admin

chmod -R 775 /u01/app/oracle/oradata
chmod -R 775 /u01/app/oracle/fast_recovery_area
chmod -R 775 /u01/app/oracle/admin
```

Then switch to oracle

```
su - oracle
```

It looks like the Oracle environment variables are not being set correctly due to the ORACLE_BASE environment variable not being set for the current user. Let's go through the steps to troubleshoot and resolve this issue:

1. **Check /etc/oratab**: Ensure that CDBDEV is not present in the `/etc/oratab` file. If it is, remove it.

   ```
   vi /etc/oratab
   ```

   Delete any line containing `CDBDEV` if it exists.

2. **Set Oracle Environment Variables Manually**: Since the ORACLE_BASE is not set automatically, you can set it manually in the terminal.

   ```
   export ORACLE_BASE=/u01/app/oracle
   export ORACLE_HOME=/u01/app/oracle/product/19.3.0/dbhome_1
   export ORACLE_SID=CDBDEV
   export PATH=$ORACLE_HOME/bin:$PATH
   ```

3. **Create Initialization Parameter File**: Follow the steps to create and edit the initialization parameter file.

   ```
   cp $ORACLE_HOME/dbs/init.ora $ORACLE_HOME/dbs/initCDBDEV.ora
   vi $ORACLE_HOME/dbs/initCDBDEV.ora
   ```

Set the following parameters in the `initCDBDEV.ora` file:

```
# Change <ORACLE_BASE> to point to the oracle base (the one you specify at
install time)

db_name='CDBDEV'
enable_pluggable_database=true
sga_target=512M
pga_aggregate_target=512M
processes=150
audit_file_dest='/u01/app/oracle/admin/CDBDEV/adump'
audit_trail='db'
db_block_size=8192
db_domain=''
db_create_file_dest='/u01/app/oracle/oradata'
db_recovery_file_dest='/u01/app/oracle/fast_recovery_area'
db_recovery_file_dest_size=2G
diagnostic_dest='/u01/app/oracle'
dispatchers='(PROTOCOL=TCP) (SERVICE=CDBDEVXDB)'
open_cursors=300
remote_login_passwordfile='EXCLUSIVE'
undo_tablespace=UNDOTBS1
control_files=(ora_control1, ora_control2)
compatible='19.0.0'
```

4. **Verify Required Directories**: Ensure that the required directories exist. Create them if they do not.

```
mkdir -p /u01/app/oracle/oradata
mkdir -p /u01/app/oracle/fast_recovery_area
mkdir -p /u01/app/oracle/admin/CDBDEV/adump
```

Also delete any files from a prior attempt and note that if the SQL fails - you'll need to delete these artifacts to run again:

```
rm /u01/app/oracle/product/19.3.0/dbhome_1/dbs/ora_control1
rm /u01/app/oracle/product/19.3.0/dbhome_1/dbs/ora_control2
rm /u01/app/oracle/oradata/CDBDEV/redo0*.log
```

5. **Start the Database Instance in NOMOUNT Mode**:

```
sqlplus / as sysdba
```

Then call:

```
 STARTUP NOMOUNT PFILE=$ORACLE_HOME/dbs/initCDBDEV.ora;
```

6. **Create the CDB**: Execute the CREATE DATABASE command.

```
CREATE DATABASE CDBDEV
USER SYS IDENTIFIED BY fenago
USER SYSTEM IDENTIFIED BY fenago
LOGFILE GROUP 1 ('/u01/app/oracle/oradata/CDBDEV/redo01.log') SIZE 100M,
    GROUP 2 ('/u01/app/oracle/oradata/CDBDEV/redo02.log') SIZE 100M,
    GROUP 3 ('/u01/app/oracle/oradata/CDBDEV/redo03.log') SIZE 100M
MAXLOGFILES 5
MAXLOGMEMBERS 5
MAXLOGHISTORY 1
MAXDATAFILES 100
CHARACTER SET AL32UTF8
NATIONAL CHARACTER SET AL16UTF16
EXTENT MANAGEMENT LOCAL
DATAFILE '/u01/app/oracle/oradata/CDBDEV/system01.dbf' SIZE 700M REUSE
SYSAUX DATAFILE '/u01/app/oracle/oradata/CDBDEV/sysaux01.dbf' SIZE 550M REUSE
DEFAULT TABLESPACE users
   DATAFILE '/u01/app/oracle/oradata/CDBDEV/users01.dbf'
   SIZE 200M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED
DEFAULT TEMPORARY TABLESPACE temp
   TEMPFILE '/u01/app/oracle/oradata/CDBDEV/temp01.dbf'
   SIZE 20M REUSE
UNDO TABLESPACE undotbs1
   DATAFILE '/u01/app/oracle/oradata/CDBDEV/undotbs01.dbf'
   SIZE 200M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED
ENABLE PLUGGABLE DATABASE
   SEED
   FILE_NAME_CONVERT = ('/u01/app/oracle/oradata/CDBDEV/',
'/u01/app/oracle/oradata/pdbseed/')
   SYSTEM DATAFILES SIZE 125M AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED
   SYSAUX DATAFILES SIZE 100M;
```

7. *_Execute Catalog and Catproc Scripts from inside of SQL_Plus **:

   ```
   @$ORACLE_HOME/rdbms/admin/catalog.sql
   ```

   then

   ```
    @$ORACLE_HOME/rdbms/admin/catproc.sql
   ```

   **Purpose of Running catalog.sql and catproc.sql**
   After creating an Oracle database, you need to run two scripts: catalog.sql and catproc.sql. These scripts are essential for setting up the data dictionary and installing standard PL/SQL packages.

- catalog.sql: This script creates the data dictionary views and tables. The data dictionary is a collection of database tables and views containing reference information about the database, its structures, and its users.

- catproc.sql: This script installs the standard PL/SQL packages and procedures needed by Oracle. These packages provide various utility functions, including job scheduling, execution control, and data manipulation.

9. **Exit SQL*Plus**

   ```
   EXIT;
   ```

10. **Add Entry to /etc/oratab**

    Add the new entry to `/etc/oratab`:

```
echo "CDBDEV:/u01/app/oracle/product/19.3.0/dbhome_1:Y" | sudo tee -a
/etc/oratab
```

Verify the entry:

```
cat /etc/oratab
```

11. **Verify Database Characteristics**

    Verify that the specified tablespaces are created for the `CDB$ROOT`:

    ```
    sqlplus / as sysdba
    ```

    ```
    SELECT tablespace_name FROM dba_tablespaces;
    ```

    **Expected Output:** The output should include `SYSTEM`, `SYSAUX`, `UNDOTBS`, `TEMP`, and `USERS`.

Following these steps should help you successfully create the CDBDEV database. If you encounter any issues, please provide the specific error messages so I can assist further.

### Lab Addendum: Open up the Database by Configuring the Listener

1. **Create/Edit the `listener.ora` File**:

   If the `listener.ora` file does not exist, create it in the `$ORACLE_HOME/network/admin` directory. If it exists, ensure it is configured correctly.

   ```
   vi $ORACLE_HOME/network/admin/listener.ora
   ```

   Add the following configuration:

   ```
   LISTENER =
     (DESCRIPTION_LIST =
       (DESCRIPTION =
         (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
       )
     )
   ```

2. **Configure the `tnsnames.ora` File**:

   Ensure that the `tnsnames.ora` file exists and is configured to connect to the `CDBDEV` service. This file is also located in `$ORACLE_HOME/network/admin`.

   ```
   vi $ORACLE_HOME/network/admin/tnsnames.ora
   ```

   Add the following entry:

   ```
   CDBDEV =
     (DESCRIPTION =
       (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
       (CONNECT_DATA =
         (SERVER = DEDICATED)
         (SERVICE_NAME = CDBDEV)
   ```

```
    )
  )
```

3. **Start/Reload the Listener**:

   After configuring the `listener.ora` file, restart or reload the listener.

   ```
   lsnrctl stop
   lsnrctl start
   ```

4. **Register the Database with the Listener**:

   Connect to the database as SYSDBA and register the database with the listener.

   ```
   sqlplus / as sysdba
   ALTER SYSTEM REGISTER;
   ```

5. **Verify the Listener Status**:

   Check the status of the listener again to ensure it is now supporting the `CDBDEV` service.

   ```
   lsnrctl status
   ```

### Example Commands

Here's a summary of the commands you'll need to run:

```
# Create/Edit listener.ora
vi $ORACLE_HOME/network/admin/listener.ora
# Add the following content:
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
    )
  )

# Create/Edit tnsnames.ora
vi $ORACLE_HOME/network/admin/tnsnames.ora
# Add the following content:
CDBDEV =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = CDBDEV)
    )
  )

# Restart the listener
lsnrctl stop
lsnrctl start

# Register the database with the listener
```

```
sqlplus / as sysdba
ALTER SYSTEM REGISTER;

# Verify listener status
lsnrctl status
```

### Verification

After performing these steps, retry connecting to the database using SQL Developer. You should now be able to connect successfully if the listener is properly configured and the database is registered.

### Summary

This addendum ensures that students can correctly set up the listener and connect to their Oracle database using SQL Developer. Proper configuration of `listener.ora` and `tnsnames.ora` is crucial for remote connections, which is an essential part of database administration.

### Lab Addendum: Connecting to the CDB Using SQL Developer

#### Objective:

To guide students on how to launch SQL Developer, connect to the newly created Container Database (CDB) `CDBDEV`, and provide a tour of SQL Developer from an administrative perspective.

### Steps to Follow

#### 1. Launch SQL Developer

1. **Open SQL Developer**:
   - Locate the SQL Developer launcher icon on your desktop.
   - Double-click the SQL Developer icon to open the application.

#### 2. Create a New Connection

1. **Open the New Connection Window**:

   - In SQL Developer, go to the menu bar and select `File` -> `New` -> `Database Connection...`.
   - Alternatively, you can click the green `+` button in the Connections pane.

2. **Enter Connection Details**:

   - **Connection Name**: `CDBDEV_Admin`
   - **Username**: `SYS`
   - **Password**: `fenago`
   - **Connection Type**: `Basic`
   - **Role**: `SYSDBA`
   - **Hostname**: `localhost` (or the appropriate hostname where the Oracle database is running)
   - **Port**: `1521`
   - **SID**: `CDBDEV`

   The connection details should look like this:

   ```
   Connection Name: CDBDEV_Admin
   Username: SYS
   Password: fenago
   ```

```
Connection Type: Basic
Role: SYSDBA
Hostname: localhost
Port: 1521
SID: CDBDEV
```

3. **Test the Connection**:
   - Click the `Test` button to verify the connection details. You should see a success message if everything is configured correctly.
4. **Save and Connect**:
   - Click the `Save` button to save the connection.
   - Click the `Connect` button to establish the connection to `CDBDEV`.

### 3. Explore SQL Developer from an Admin Perspective

1. **Connections Pane**:
   - Once connected, you will see `CDBDEV_Admin` listed under the Connections pane.
   - Expand the `CDBDEV_Admin` connection to see the various database objects.
2. **Navigating the Database**:
   - **Tables**: Expand `CDBDEV_Admin` -> `Tables` to view and manage tables.
   - **Views**: Expand `CDBDEV_Admin` -> `Views` to see the database views.
   - **Indexes**: Expand `CDBDEV_Admin` -> `Indexes` to view the indexes.
   - **Users**: Expand `CDBDEV_Admin` -> `Security` -> `Users` to manage database users.
   - **Roles**: Expand `CDBDEV_Admin` -> `Security` -> `Roles` to manage database roles.
   - **Storage**: Expand `CDBDEV_Admin` -> `Storage` to manage tablespaces and other storage options.
3. **Running SQL Scripts**:
   - Click on the `SQL Worksheet` button (the pencil icon) or right-click on the `CDBDEV_Admin` connection and select `SQL Worksheet`.
   - In the worksheet, you can run SQL commands and scripts. For example, to view the tablespaces:

     ```sql
     SELECT tablespace_name FROM dba_tablespaces;
     ```

   - Execute the command by pressing `F5` or clicking the `Run Script` button.
4. **Viewing Database Sessions**:
   - Navigate to `CDBDEV_Admin` -> `Performance` -> `Sessions` to monitor active database sessions.
5. **Monitoring Performance**:
   - SQL Developer provides various performance monitoring tools under the `Performance` tab.
   - You can view active sessions, wait events, and other performance metrics.
6. **Creating and Managing Users**:
   - Go to `CDBDEV_Admin` -> `Security` -> `Users`.

- Right-click on `Users` and select `Create User` to create a new database user.
- Fill in the necessary details and click `OK` to create the user.

7. **Exporting and Importing Data**:

   - SQL Developer allows you to export and import data using the Data Pump utility.
   - Navigate to `CDBDEV_Admin` -> `Data Pump` to access these features.

8. **PL/SQL Development**:

   - Use the `PL/SQL` tab to develop and debug PL/SQL code.
   - SQL Developer provides a PL/SQL debugger and profiler to help with code development.

## Summary

By following these steps, you will be able to launch SQL Developer, connect to the `CDBDEV` database, and explore various administrative functionalities. This tour covers the essential aspects of database management using SQL Developer, enabling you to efficiently manage and monitor your Oracle database.

### Optional

No, you don't need to run the `. oraenv` command if you manually set the environment variables using `export`. Here's a recap of what you need to do:

1. **Set Oracle Environment Variables Manually**: Set the Oracle environment variables manually in the terminal:

   ```
   export ORACLE_BASE=/u01/app/oracle
   export ORACLE_HOME=/u01/app/oracle/product/19.3.0/dbhome_1
   export ORACLE_SID=CDBDEV
   ```

After setting these environment variables, you can proceed with the rest of the steps to create the CDBDEV database.

If you prefer to use `. oraenv`, follow these steps instead:

1. **Update /etc/oratab**: Ensure that `CDBDEV` is in the `/etc/oratab` file with the correct path.

   ```
   echo "CDBDEV:/u01/app/oracle/product/19.3.0/dbhome_1:Y" | sudo tee -a
   /etc/oratab
   ```

2. **Run . oraenv**: Execute the `oraenv` command and specify `CDBDEV`.

   ```
   . oraenv
   ```

   When prompted, enter `CDBDEV`.

By running `. oraenv`, it will source the necessary environment variables from the `/etc/oratab` file.

After setting the environment variables using either method, proceed with the remaining steps to create and configure the CDBDEV database.