Lab 8.1: Configuring the Network to Access an Oracle Database

Objective:

To configure the network environment to connect to another Oracle database using local naming and create a new network service name. Use `CDBLAB` and the PDBs that were created earlier. Additionally, create a net service name for a PDB.

Part A: Configuring the Network to Access an Oracle Database

Use Case:

You need to configure the network to connect to `CDBTEST` from `CDBLAB`. This configuration ensures seamless communication between different databases in a multi-database environment, allowing for distributed database operations and easy management.

Steps:

1. **Open a Terminal and Set Environment to `CDBLAB`**

   Open a terminal and set your environment to your database SID to `CDBLAB`:

   ```
   . oraenv
   ORACLE_SID = [orclcdb] ? CDBLAB
   The Oracle base remains unchanged with value /u01/app/oracle
   ```

2. **Verify the Databases `CDBTEST` and `orclcdb` are in `/etc/oratab`**

   Verify the presence of `CDBTEST` and `orclcdb` in the `/etc/oratab` file:

   ```
   more /etc/oratab
   ```

   Expected Output:

   ```
   orclcdb:/u01/app/oracle/product/19.3.0/dbhome_1:N
   CDBTEST:/u01/app/oracle/product/19.3.0/dbhome_1:N
   CDBLAB:/u01/app/oracle/product/19.3.0/dbhome_1:N
   ```

3. **Make a Copy of Your `tnsnames.ora` File**

   Change the directory to `$ORACLE_HOME/network/admin` and make a copy of `tnsnames.ora`:

   ```
   cd /u01/app/oracle/product/19.3.0/dbhome_1/network/admin
   pwd
   cp tnsnames.ora tnsnames.old
   ls -l
   ```

   Expected Output:

   ```
   -rw-r----- 1 oracle oinstall 1870 Oct 16 05:06 tnsnames.ora
   -rw-r----- 1 oracle oinstall 1870 Oct 16 22:05 tnsnames.old
   ```

4. **Determine the Fully Qualified Host Name**

   Determine the fully qualified host name:

```
hostname -f
```

Expected Output:

```
your_fully_qualified_hostname
```

5. **Invoke Oracle Net Manager to Create the `testorcl` Net Service**

   Start Oracle Net Manager:

   ```
   netmgr
   ```

   In Oracle Net Manager:

   - Expand `Local` and select `Service Naming`.
   - Click the green plus sign to add a new service.
   - In the `Service Name` field, enter `testorcl` and click `Next`.
   - Select `TCP/IP` and click `Next`.
   - In the `Host Name` field, enter the fully qualified host name obtained in step 4.
   - In the `Port Number` field, enter `1521` and click `Next`.
   - In the `Service` field, enter `CDBTEST`.
   - Under `Connection type`, select `Dedicated Server` and click `Next`.
   - Click `Test`.
   - In the `Connection test` dialog box, click `Change Login` because the test will fail.
   - Enter username `system` and the corresponding password, then click `OK`.
   - Click `Test` again.
   - When the "The connection test was successful" message appears, click `Close` and then `Finish`.
   - Click `File > Save Network Configuration`.
   - Exit Oracle Net Manager.

6. **Test the Network Configuration Using SQL*Plus**

   Ensure your environment is set for the `CDBLAB` database:

   ```
   . oraenv
   ORACLE_SID = [oracle] ? CDBLAB
   The Oracle base remains unchanged with value /u01/app/oracle
   ```

   Invoke SQL*Plus and connect using the `testorcl` service name:

   ```
   sqlplus system@testorcl
   ```

   Expected Output:

   ```
   Enter password: password
   SQL>
   ```

7. **Verify the Connection to the Correct Database**

   Verify that you are connected to the correct database by selecting the `INSTANCE_NAME` and `HOST_NAME` columns from the `V$INSTANCE` view:

```
column host_name format a50
SELECT instance_name, host_name FROM v$instance;
```

Expected Output:

```
INSTANCE_NAME HOST_NAME
------------- --------------------------------------------------
CDBTEST       your_fully_qualified_hostname
```

8. **Exit SQL*Plus**

   Exit SQL*Plus:

   ```
   exit
   ```

Part B: Creating a Net Service Name for a PDB

Overview:

In this practice, you create a net service name called `MyPDB1` to access a PDB (`PDB1`) by using Oracle Net Manager.

Steps:

1. **Invoke Oracle Net Manager to Create the `MyPDB1` Net Service**

   Start Oracle Net Manager:

   ```
   netmgr
   ```

   In Oracle Net Manager:

   - Expand `Local` and select `Service Naming` .
   - Click the green plus sign to add a new service.
   - In the `Service Name` field, enter `MyPDB1` and click `Next` .
   - Select `TCP/IP` and click `Next` .
   - In the `Host Name` field, enter the fully qualified host name obtained in Part A, Step 4.
   - In the `Port Number` field, enter `1521` and click `Next` .
   - In the `Service` field, enter `PDB1` .
   - Under `Connection type` , select `Dedicated Server` and click `Next` .
   - Click `Test` .
   - In the `Connection test` dialog box, click `Change Login` because the test will fail.
   - Enter username `system` and the corresponding password, then click `OK` .
   - Click `Test` again.
   - When the "The connection test was successful" message appears, click `Close` and then `Finish` .
   - Click `File > Save Network Configuration` .
   - Exit Oracle Net Manager.

2. **Test the Network Configuration Using SQL*Plus**

   Ensure your environment is set for the `CDBLAB` database:

```
. oraenv
ORACLE_SID = [oracle] ? CDBLAB
The Oracle base remains unchanged with value /u01/app/oracle
```

Invoke SQL*Plus and connect using the `MyPDB1` service name:

```
sqlplus system@MyPDB1
```

Expected Output:

```
Enter password: password
SQL>
```

3. **Verify the Connection to the Correct PDB**

   Verify that you are connected to the correct PDB by selecting the `INSTANCE_NAME` and `HOST_NAME` columns from the `V$INSTANCE` view:

```
column host_name format a50
SELECT instance_name, host_name FROM v$instance;
```

   Expected Output:

```
INSTANCE_NAME HOST_NAME
------------- --------------------------------------------------
PDB1          your_fully_qualified_hostname
```

4. **Exit SQL*Plus**

   Exit SQL*Plus:

```
exit
```

Summary

In this lab, you configured the network environment to connect to `CDBTEST` using local naming and created a new network service name `testorcl`. You then created a net service name `MyPDB1` to access a PDB (`PDB1`). These configurations ensure seamless communication between different databases and pluggable databases, facilitating distributed database operations and easy management.