

Practice

Configuring Oracle Network Environment

Practice Target

In this practice you will manage the connections to Oracle databases and network configuration tools.

Specifically, you will perform the following:

- Use Easy Connect and Local Naming Methods
- Control the allowable connection naming methods
- Control the Listener using `lsnrctl`
- Install the sample application Swingbench
- Install Oracle instant client

Practice Assumptions

This practice assumes that the virtual machine `srv1` is up and running from its **CDB** snapshot.



Using Easy Connect and Local Naming Methods

In this section of the practice, you will examine the network configuration files in the server, test the Easy Connect configuration, and configure the Local Naming method.

Examining the Network Configuration Files

1. Open a Putty session to `srv1` as `oracle`
2. Verify the CDB database is up and running.

```
sqlplus / as sysdba
SELECT CDB FROM V$DATABASE;
```

3. Exit from SQL*Plus.
4. Check out the Oracle Net configuration files in the server.

- 4.1. Check if the `TNS_ADMIN` is configured.

```
echo $TNS_ADMIN
```

Because the `TNS_ADMIN` variable is not set, the Oracle Net Configuration files of the current Oracle database home are located in `$ORACLE_HOME/network/admin`. This applies in Linux and Windows platforms. If the variable is set, the configuration files will be located in the directory pointed by the `TNS_ADMIN` variable value.

- 4.2. Display the files located in `$ORACLE_HOME/network/admin`

You should see the Oracle Net configuration files: `listener.ora`, `sqlnet.ora`, and `tnsnames.ora`

```
ls -al $ORACLE_HOME/network/admin
```

- 4.3. Set the `TNS_ADMIN` variable in the bash profile file.

Although the network configuration files are located in the default directory, it is still a common practice to set the `$TNS_ADMIN` in the bash profile file, even if it points to the default location. This helps us to quickly change the current directory to the network configuration directory whenever we need to do so.

```
vi ~/.bash_profile
```

```
# add the following line to it:
export TNS_ADMIN=$ORACLE_HOME/network/admin
```

- 4.4. Source the bash profile file and verify that variable is set.

```
source ~/.bash_profile
echo $TNS_ADMIN
```

5. Display the contents of the Listener configuration file.

Observe that the file defines the network protocol, hostname, and port number. The port number is set to its default value, which is 1521.

```
cat $TNS_ADMIN/listener.ora
```

6. Display the contents of the `tnsnames.ora` file

The file contains one network service name to the root container. The name is the same as the database instance name (`ORADB`). This naming configuration was added to the file by the `dbca`.

The file does not contain a connection descriptor name for `pdb1`.

```
cat $TNS_ADMIN/tnsnames.ora
```

Using Easy Connect Method

After we examined the Oracle Net configuration files, let's try the Easy Connect method for connecting to the database from SQL*Plus.

7. Invoke the following command to list the database services registered in the Listener:

You should observe that among the services registered in the Listener are the following database services:

- `oradb.localdomain`, which represents the root container
- `pdb1.localdomain`, which represents the `PDB1` container

```
lsnrctl service
```

8. Invoke SQL*Plus and connect to `PDB1` as `HR` using Easy Connect method.

The service name that we should pass to the Easy Connect string must be exactly the same as the one displayed by the `lsnrctl service` command. If we omit the domain name from the full qualified service name, the connection attempt fails.

This connection example assumes that the Listener is listening to the default port number 1521.

```
sqlplus hr/ABcd##1234@//srv1/pdb1.localdomain
```

9. Let's examine connecting to the root container using Easy Connect method.

```
conn system/ABcd##1234@//srv1/oradb.localdomain
show con_name
```

Using Local Naming Method

In the following steps, you will configure and test the local naming method to connect to the Oracle database service. Naming method is the most common naming methods used to connect to Oracle databases.

10. Connect to the root container using the local naming method.

ORADB in the command below does not represent the database service name. It is the network service name defined in the `tnsnames.ora` file. The `dbca` utility sets its name the same as the CDB database service name but it can take any name.

```
conn system/ABcd##1234@oradb
```

11. Try connecting to the root container as `sys` using the `ORADB` network service name.

This connection type uses database username/password authentication to login to the database.

```
conn sys/ABcd##1234@oradb as sysdba
```

12. Try connecting to the root container as `sys` using the `ORADB` connection name but with passing a wrong password.

The connection is refused because the password is wrong. The OS authentication is not used because this is considered as a remote connection.

```
conn sys/xyz@oradb as sysdba
```

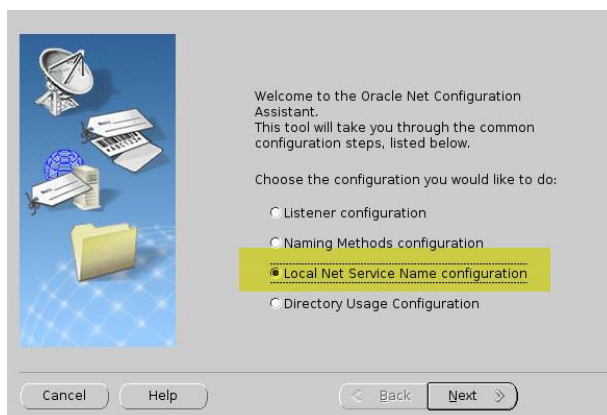
At the moment, we cannot connect to `PDB1` using the local naming method because there is no connection name configured in the `tnsnames.ora` file for this container service. Let's configure a network service name to `PDB1` in the `tnsnames.ora` file using the `netca` utility.


Note: We could manually add a connection descriptor to `PDB1` in `tnsnames.ora` file by simply editing the file using any editor. We are using `netca` over here just to gain some experience on using it.

13. In the VirtualBox window of `srv1`, login as `oracle`, open a terminal window then invoke the `netca` utility.

```
netca
```

14. Respond to the Wizard as follows:






To access an Oracle database, or other service, across the network you use a net service name. The Oracle Net Configuration Assistant allows you to work with net service names resolved using local naming.

Select what you want to do:

- ☒ Add
- ☐ Reconfigure
- ☐ Delete
- ☐ Rename
- ☐ Test


Cancel Help Back Next



Each Oracle database or service has a service name. An Oracle database's service name is normally its global database name. Enter the service name of the database or other service you want to access.

Service Name:

Cancel Help Back Next




To communicate with the database across a network, a network protocol is used. Select the protocol used for the database you want to access.

☒ TCP

☐ TCPS

☐ IPC

Cancel Help Back Next



To communicate with the database using the TCP/IP protocol, the database computer's host name is required. Enter the host name for the computer where the database is located.

Host name:

A TCP/IP port number is also required. In most cases the standard port number should be used.

☒ Use the standard port number of 1521

☐ Use another port number:

Cancel Help Back Next

The image displays four sequential screenshots of the Oracle Database Configuration Assistant (DBCA) interface, illustrating the process of configuring the network environment.

Screenshot 1: Connection Test

You can verify that an Oracle database can be reached, using the information provided, by performing a connection test.

Would you like to test that a connection can be made to the database?

☐ No, do not test

☒ Yes, perform a test

Buttons: Cancel, Help, Back, Next

Screenshot 2: Change Login Details

Change Login

Details:

Change Login

Buttons: Cancel, Help, Back, Next

Screenshot 3: Change Login Username and Password

Change Login

Details:

Change Login

Username: system

Password: *****

Buttons: OK, Cancel

Change Login

Buttons: Cancel, Help, Back, Next

Screenshot 4: Connection Test Successful

Change Login

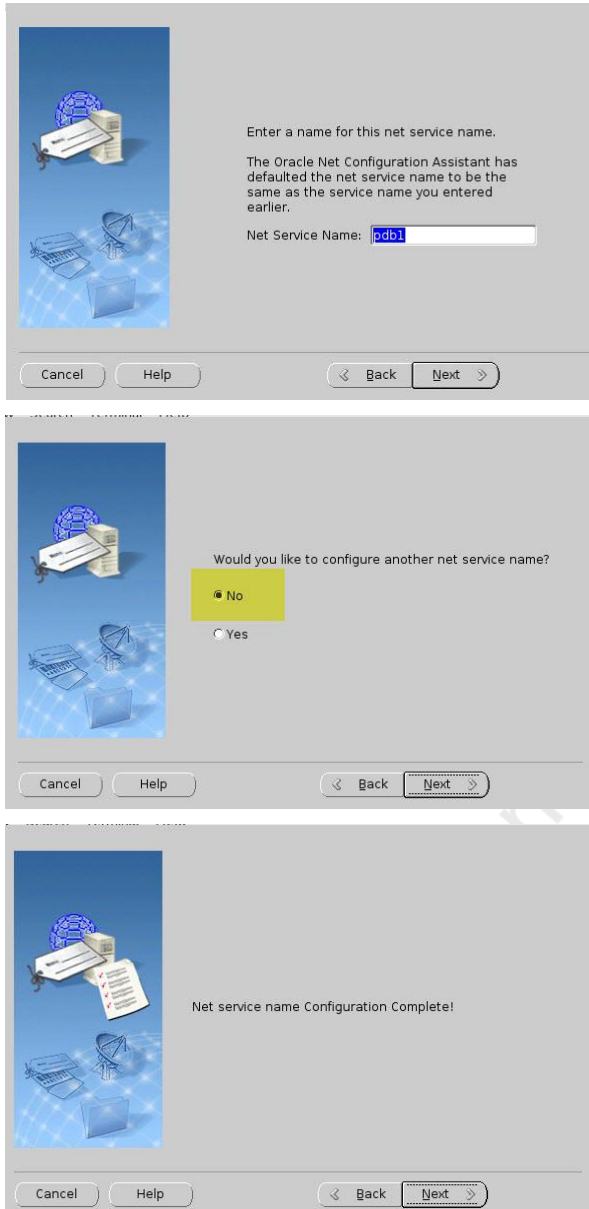
Details:

Connecting...Test successful.

Change Login

Buttons: Cancel, Help, Back, Next

Although we normally set the Net Service Name the same as the PDB name, technically we can assign any name to it. This is the name we use with the "@" symbol to connect to the database.



- 15.** In the Putty session, verify that the `PDB1` network service name is added to the `tnsnames.ora` file.

We should see a connection descriptor assigned to the name `PDB1`.

`host` is a SQL*Plus command which allows us to execute an OS command from within the SQL*Plus command prompt.

```
SQL> host cat $ORACLE_HOME/network/admin/tnsnames.ora
```

- 16.** Using `tnsping` utility, test if the machine can see the host defined in the added connection name.

The utility just pings the connection hostname but it does not make a connection attempt to the database and it does not verify that the database service is up and running. The ping is executed against the port number stated in the network service name.

```
host tnsping pdb1
```

- 17.** Try connecting to `PDB1` using its connection name in the `tnsnames.ora` file

The connection should be successful.

```
conn system/ABcd##1234@pdb1
```

Although `netca` provides an easy to use interface to adding a connection descriptor name to the `tnsnames.ora` file, DBAs normally use an editor to add a connection name by copying and pasting an existing connection name configuration and then changing the pasted connection name and its connection configuration parameter values in it. That is what you will do in the following steps.

- 18.** Open the `tnsnames.ora` file with the `vi` editor.

In a Windows platform, you could use any editor like the notepad.

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

- 19.** Highlight the connection configuration of `PDB1`.

- 20.** Go to the end of the file and paste the code from the clipboard.

- 21.** Change the connection name to `hrpdb`

- 22.** Save the changes and exit from the `vi` editor.

- 23.** Try connecting to `pdb1` using the new network service name.

The connection should be successful.

```
conn system/ABcd##1234@hrpdb
```

- 24.** In `tnsnames.ora` file, add a connection descriptor to the database in `winsrv`.

```
host vi $TNS_ADMIN/tnsnames.ora
```

```
ORAWINDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = winsrv)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orawindb)
    )
  )
```


25. If `winsrv` is running, test connecting to it from `srv1`.

```
sqlplus system/ABcd##1234@ORAWINDB
```

26. Exit from SQL*Plus.



Ahmed Baraka
Oracle Database Administrator

Controlling the Allowable Connection Naming Methods

In this section of the practice, you will disable the Easy Connect method to the database and test this configuration.

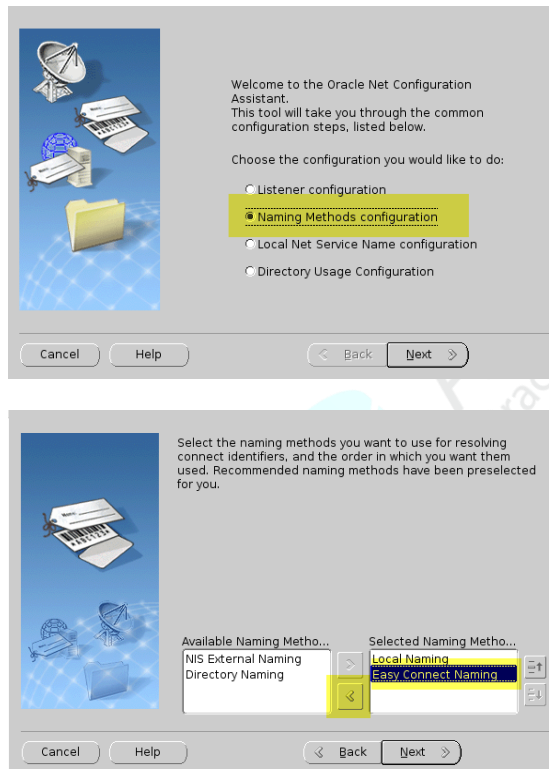
27. Display the contents of the `sqlnet.ora` file

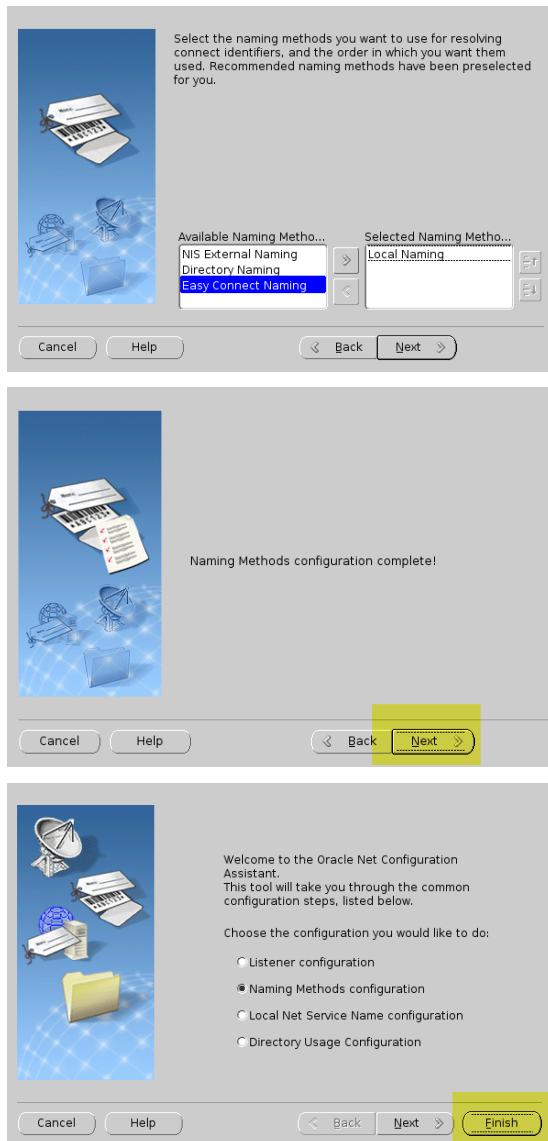
The file has one parameter, which is `NAMES.DIRECTORY_PATH`. This parameter defines the allowed naming methods to the database and their priorities. In its current value, the parameter value allows to connect to the database using local naming (`TNSNAMES`) and Easy Connect (`EZCONNECT`) methods.

This is not the only parameter that we may configure in the `sqlnet.ora` file. There are plenty of other parameters and they are used on-need basis.

```
cat $ORACLE_HOME/network/admin/sqlnet.ora
```

28. In the VirtualBox window of `srv1`, invoke the `netca` and perform the actions as follows:





29. Display the contents of the `sqlnet.ora` file

The `EZCONNECT` is removed from the `NAMES.DIRECTORY_PATH` parameter.

Note: We could use an editor to simply remove the `EZCONNECT` from the parameter list of values.

```
cat $ORACLE_HOME/network/admin/sqlnet.ora
```

30. Try connecting to the database using Easy Connect method.

The connection attempt fails because Easy Connect naming method is disabled. The returned error does not state that the Easy Connect method is disabled.

```
sqlplus system/ABcd##1234@//srv1/oradb.localdomain
```

- 31.** Open the `sqlnet.ora` file with the `vi` editor and add the `EZCONNECT` value back to the `NAMES.DIRECTORY_PATH` parameter.

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

```
NAMES.DIRECTORY_PATH= (TNSNAMES,EZCONNECT)
```

- 32.** Try connecting to the database using Easy Connect method.
The connection attempt should succeed.

```
sqlplus system/ABcd##1234@//srv1/oradb.localdomain
```

- 33.** Exit from SQL*Plus.

```
quit
```



Ahmed Baraka
Oracle Database Administrator

Controlling the Listener using lsnrctl

In this section of the practice, you will practice the common tasks involving in managing the listener using `lsnrctl` utility.

- 34.** Using the `lsnrctl` utility, check out the status of the Listener.

The command does not clearly specify that the Listener is running but if we see the "Listening Endpoints Summary" returned by the output, then this means the Listener is running.

```
lsnrctl status
```

- 35.** Using the OS `ps` command, verify that the Listener process is up and running.

This command could be a better way to check the status of the Listener. In some cases, the Listener is running from a different Oracle home or from a different OS user. In this case, `lsnrctl` does not return the status of the running Listener because the current OS user is not the owner of the Listener process or the Listener is running from an Oracle Home different than the current Oracle Home. This `ps` command can detect such cases.

You can tell from the output of this command from which Oracle home the Listener is running and by which OS user.

```
ps -ef | grep tns
```

- 36.** Stop the Listener.

Stopping the Listener could be needed when we do not want to allow the user applications to connect to the database (probably for maintenance purposes) and yet we want the administrator users be able to connect locally to the database. We also need to stop that listener and start it again when we want to apply changes made on the `listener.ora` file.

However, stopping the Listener does not disconnect existing sessions.

```
lsnrctl stop
```

- 37.** Check the status of the Listener.

The command tries to connect to the listening endpoints and displays the message "no listener" for each end point.

```
lsnrctl status
```

- 38.** Try connecting to a database service.

You should receive the following error:

```
ORA-12541: TNS:no listener
```

```
sqlplus system/ABcd##1234@pdb1
```

- 39.** Start the Listener then check out its status.

```
lsnrctl start  
lsnrctl status
```

40. Try connecting to a database service. The connection should be successful.

```
sqlplus system/ABcd##1234@pdb1
```

The Listener is listening to an endpoint named `EXTPROC*`. This is needed only when we need to call external procedures. As we do not need to call External Procedures in our environment, it is among the best security practices to remove this listening point from the Listener.

41. Open the `listener.ora` file and remove the `EXTPROC*` from it.

Tip: In `vi` editor, to delete the current link, double press on `[d]` key.

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

```
LISTENER =  
  (DESCRIPTION_LIST =  
    (DESCRIPTION =  
      (ADDRESS = (PROTOCOL = TCP)(HOST = srv1)(PORT = 1521))  
    )  
  )
```

42. Reload the Listener configuration and check out whether the `EXTPROC` end point is not listened by the Listener.

Unfortunately, the listening endpoint is still there.

```
lsnrctl reload  
lsnrctl status
```

43. Restart the Listener and then check out the status of the `EXTPROC` endpoint.

The start command displays the status of the Listener after starting the Listener.

Note: the command may display the following message:

```
"The listener supports no services"
```

This could happen because the database services need some time to register themselves in the Listener after restarting the Listener. Just wait for a few seconds then try again. If, for any reason, a database service is still not registered in the Listener, we can manually register that database service by logging in locally to the database and issuing the following statement:

```
ALTER SYSTEM REGISTER;
```

```
lsnrctl stop  
lsnrctl start  
lsnrctl status
```

Installing the Sample Application Swingbench

Swingbench is a free sample application developed by [Dominic Giles](#). We will use this application as a sample application in the course practices.

In this section of the practice, you will install Swingbench in your hosting PC and connect it to the database using Easy Connect method.

44. In the hosting PC, open a command prompt and issue the following command to check which Java version is installed in your PC.

You should have Java 1.8.x installed in your PC. In my environment, the output looked like the following:

```
java version "1.8.0_271"  
Java(TM) SE Runtime Environment (build 1.8.0_271-b09)  
Java HotSpot(TM) 64-Bit Server VM (build 25.271-b09, mixed mode)
```

```
java -version
```

45. If the required Java runtime is not installed, download Java installer from this [link](#). Manually download the file titled "Windows Offline (64-bit)". Do not use the online installed because it installs Java the 32-bit.

46. Download the Swingbench installation file from the lecture downloadable resources.

47. In your **hosting PC**, copy the software zip file to the disk drive where you want to install the software. In my case, I copied it to the c: drive.

48. Extract the zip file into the selected drive letter. This folder is referred to in this document as \$SWINGHOME

If you use the installation file attached to the course, the application files should be extracted into a folder named as 'swingbench2.6'.

```
<disk drive letter>:\swingbench2.6
```

49. In the Putty session to `srv1`, verify that the OMF is enabled.

In the next steps, you will create a new tablespace. We want to make sure the OMF is enabled so that we do not need to provide the full name path of the datafile for that tablespace.

```
sqlplus / as sysdba  
ALTER SESSION SET CONTAINER=pdb1;  
show parameter DB_CREATE_FILE_DEST
```

50. In the **hosting PC**, open a new command prompt window and change the directory to \$SWINGHOME\winbin

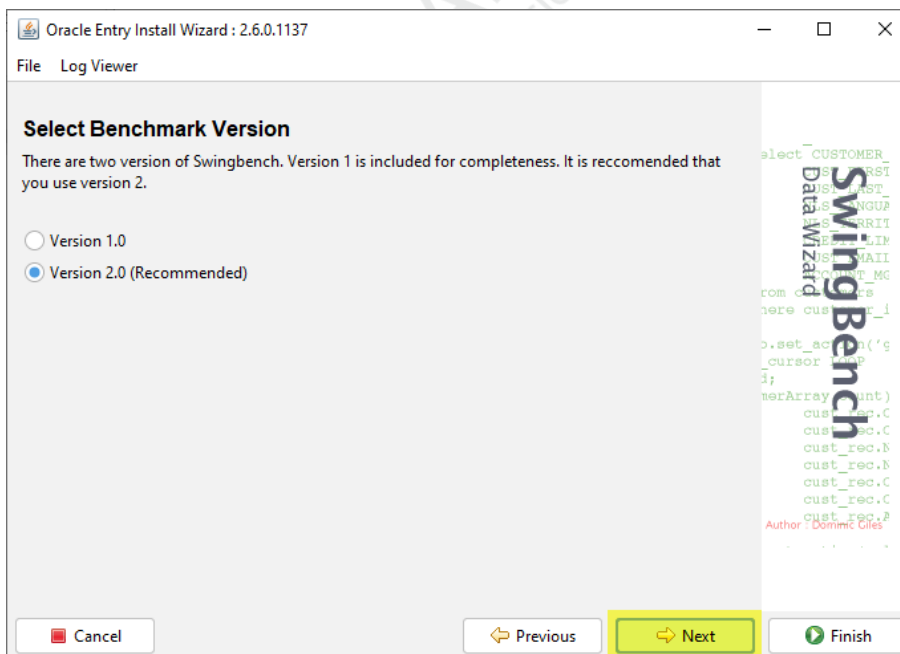
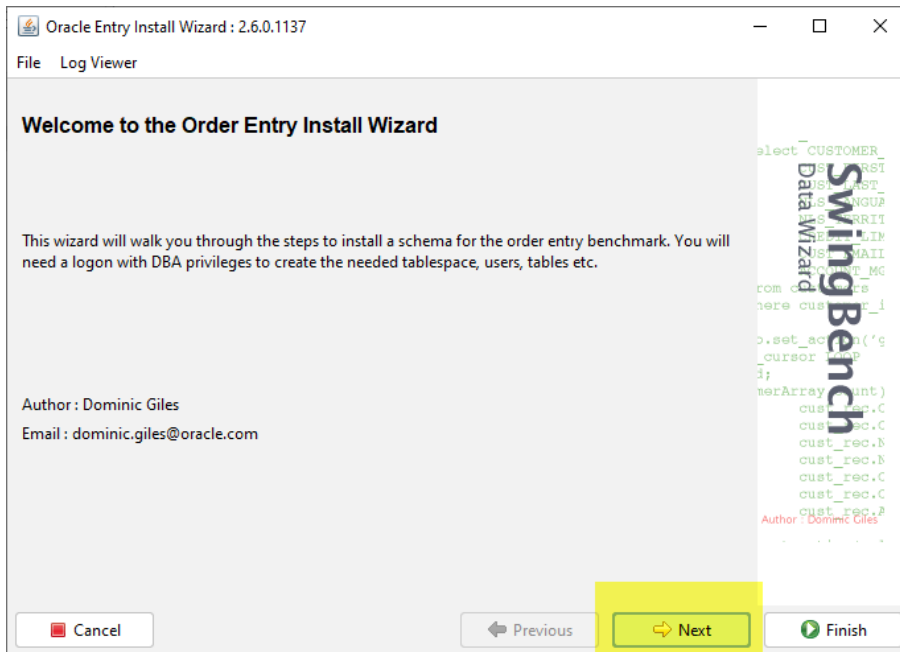
```
cd C:\swingbench2.6\winbin
```

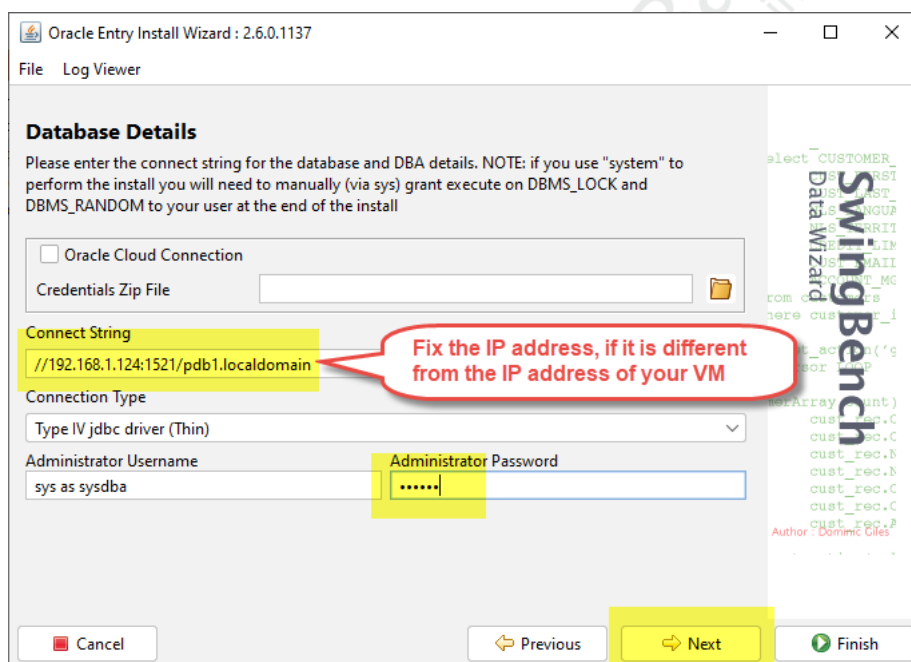
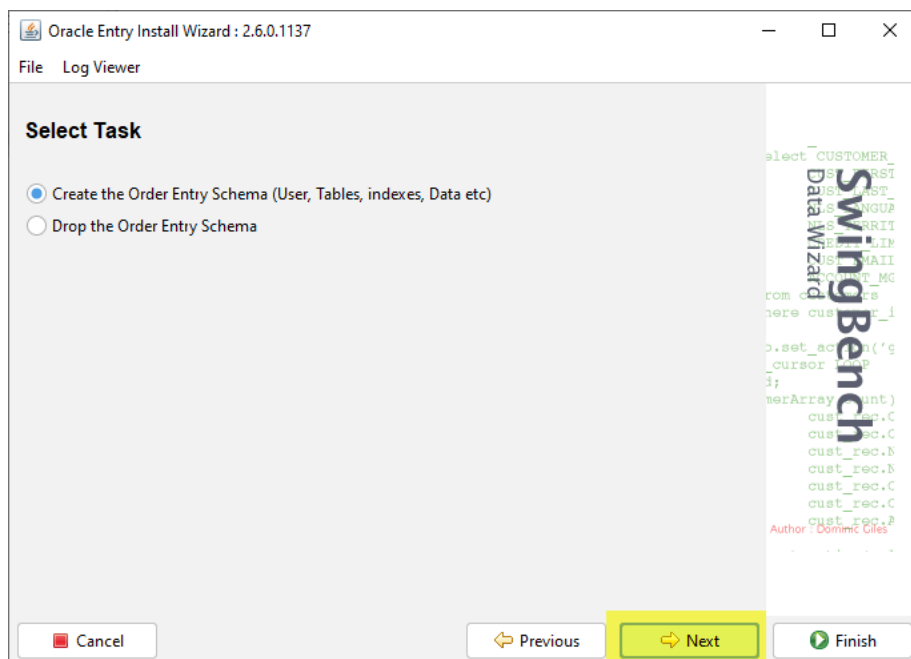
51. Start the "Order Entry Wizard".

This wizard creates a schema in the database (its name is `soe`) and fill it with sample data.

```
oewizard.bat
```

52. Respond to the wizard windows as follows to create SOE schema and load data into it.





Oracle Entry Install Wizard : 2.6.0.1137

File Log Viewer

Schema Details

Please enter the details of the schema you wish to create, this will contain all of the tables and indexes for the order entry benchmark.

Username: soe

Password: ABcd##1234

Schema's Tablespace: SOETBS

Tablespaces's Datafile:

☐ Meta Data Only Install

Cancel Previous Next Finish

```
select CUSTOMER
  from CUSTOMER
 where CUST_ID = 1;

set arraysize 100;
set echo on;
set feedback on;
set serveroutput on;
set termout on;
set verify on;

declare
  cursor c1 is
    select * from CUSTOMER;
  array customer_array(100) varchar2(50);
begin
  open c1;
  loop
    fetch c1 bulk collect into customer_array;
    for i in 1..customer_array.count
    loop
      dbms_output.put_line(customer_array(i));
    end loop;
  end loop;
  close c1;
end;
```

Oracle Entry Install Wizard : 2.6.0.1137

File Log Viewer

Database Options

Select the options you would like used during the creation of the schema. Certain options are additionally licensed by Oracle and others are only available on hardware such as Exadata. The recommended values are set by default.

Partitioning Model: No Partitioning

Compression Used: No Compression

Tablespace Type: Normal Tablespace

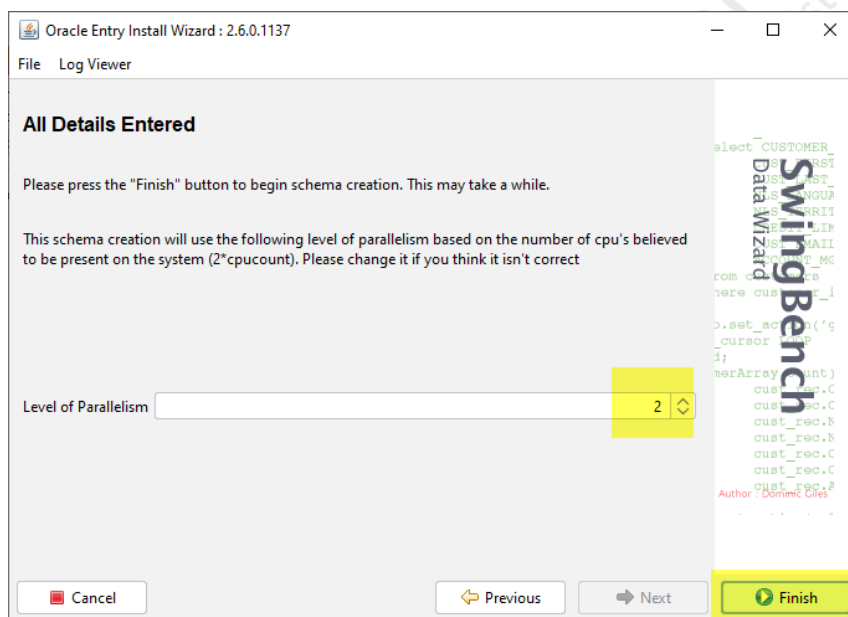
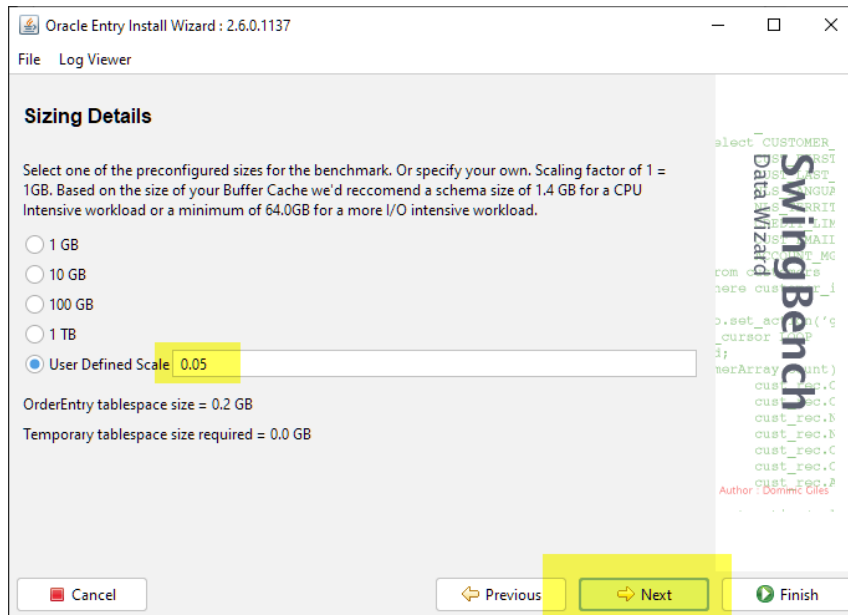
Indexing Used: All Indexes

Cancel Previous Next Finish

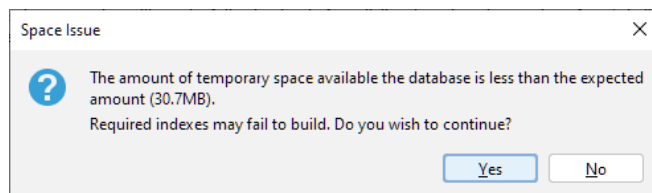
```
select CUSTOMER
  from CUSTOMER
 where CUST_ID = 1;

set arraysize 100;
set echo on;
set feedback on;
set serveroutput on;
set termout on;
set verify on;

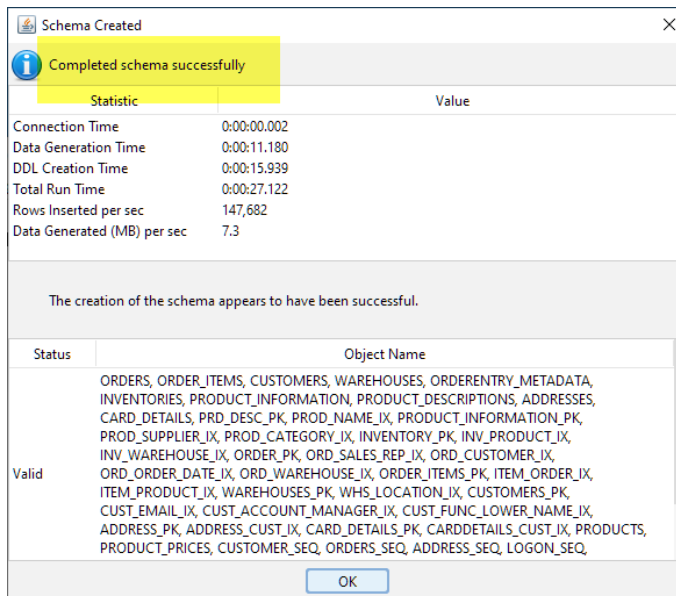
declare
  cursor c1 is
    select * from CUSTOMER;
  array customer_array(100) varchar2(50);
begin
  open c1;
  loop
    fetch c1 bulk collect into customer_array;
    for i in 1..customer_array.count
    loop
      dbms_output.put_line(customer_array(i));
    end loop;
  end loop;
  close c1;
end;
```



53. If you receive this message, click on Yes because the temporary tablespace in our database is auto-extensible.



54. The process should finish with the message like the following:

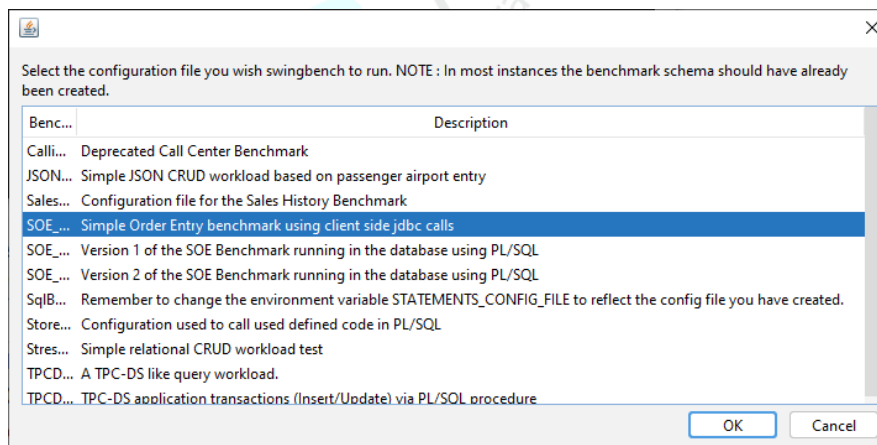


55. Click on **OK** button and close the wizard window.

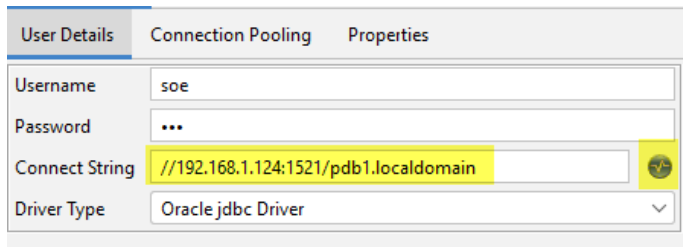
56. Start Swingbench by issuing the following command in the command prompt:

```
swingbench.bat
```

57. When you see the following window, select the option "**SOE_Client_Side**".

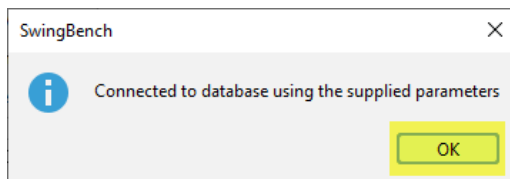


58. Under the **User Details** tab, set the fields to the values as in the following screenshot. Click on **Test Connection** button.

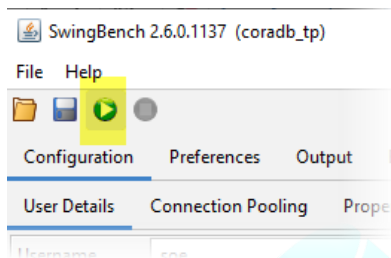


User Details	Connection Pooling	Properties
Username	soe	
Password	...	
Connect String	//192.168.1.124:1521/pdb1.localdomain	
Driver Type	Oracle jdbc Driver	

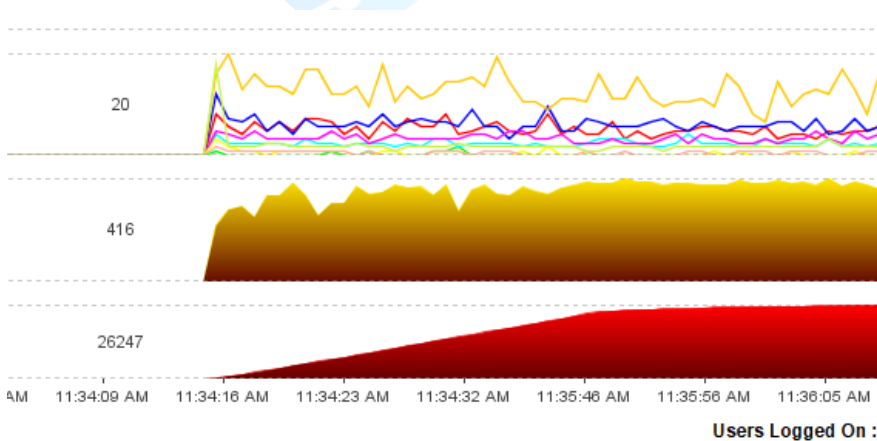
59. You should see the following successful message:



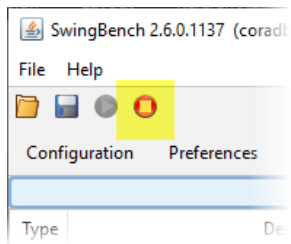
60. Click on **Start Benchmark Run** button:



61. Observe that the **"Transactions Per Minute"** chart is increasing by time and it eventually gets saturated.



- 62.** Stop the Benchmark Run by clicking on its button.



- 63.** Exit from Swingbench.

We let Swingbench connect to PDB1 using Easy Connect method. In real life, Naming method is more commonly used by applications for connecting to Oracle databases rather than Easy Connect.



Ahmed Baraka
Oracle Database Administrator

Installing Oracle Instant Client

At the moment, we cannot use the Naming method for connecting Swingbench with the database in `srv1` because Oracle Net is not installed in the hosting PC. To enable Swinbench to connect to the database using Naming method, we must install Oracle Net in the PC. The easiest way to install Oracle Net in a machine is by installing Oracle Instant Client. And that is what you're going to do in the following steps.

- 64.** Oracle Instant Client for Windows platform requires to have **Microsoft Visual Studio 2017 Redistributable** installed in your PC. If you do not have this package installed in your PC, download the file **vc_redist.x64.zip** from the lecture downloadable resources, extract it, and install the package file in it. This file was downloaded from this [link](#).
- 65.** Download the following files from the lecture downloadable resources. They are the files that we need for installing Oracle Instant Client in the hosting PC.

```
instantclient-basic-windows.x64-19.14.0.0.0dbru.zip  
instantclient-sqlplus-windows.x64-19.14.0.0.0dbru.zip  
instantclient-tools-windows.x64-19.14.0.0.0dbru.zip
```

Note: Those files were originally downloaded from this [link](#).

Note: For just connecting to the database, we just need to install the first file. The other two files provide the SQL*Plus and some other admin tools.

- 66.** Extract the file `instantclient-basic-windows.x64-19.14.0.0.0dbru.zip` into your favorite drive letter. In my case, I extracted the file in the C drive. When I extract the file into the C drive, I will have the following folder:

```
C:\instantclient_19_14
```

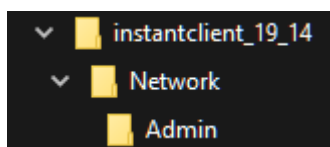
- 67.** Extract the other two files into the same drive letter. They will automatically be extracted into the same folder.

- 68.** Create the following directory structure **under** the Oracle Instant Client installation folder:

```
Network\Admin
```

Note: We could create any folder but the directory structure that we created is the default one.

The directory structure should look like the following:



69. In the `Admin` directory created in the preceding step, create the following file:

```
tnsnames.ora
```

70. Open the file with Notepad.

71. In `srv1`, open the `tnsnames.ora` file in it and copy the naming connection for `PDB1`.

```
host cat $ORACLE_HOME/network/admin/tnsnames.ora
```

72. Paste the `pdb1` connection setting into the `tnsnames.ora` file in the Oracle Instant Client. Replace the `srv1` hostname with its IP address because the hostname `srv1` is not defined locally in the PC.

```
PDB1 =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.124)(PORT = 1521))
  )
  (CONNECT_DATA =
    (SERVICE_NAME = pdb1.localdomain)
  )
)
```

73. Add the Oracle Instant Client installation folder `C:\instantclient_19_14` to the `PATH` environment variable:

73.1. Open the File Explorer

73.2. Right-click on the **This PC** icon

73.3. Select **Properties**

73.4. Click on the **Advanced system settings** link on the right side

73.5. On **System Properties** window, Click on **Advanced** tab

73.6. Click on **Environment Variables** button

73.7. Under **System Variables** panel, double click on **Path** variable

73.8. Click on **New** button

73.9. Add the Oracle Instant Client directory to it

73.10. Click on **OK** button

74. Add a new environment variable named as `TNS_ADMIN` and make its value the full directory name of the `tnsnames.ora` file.

74.1. Under **System Variables** panel, click on **New** button

74.2. Set the Variable Name to **TNS_ADMIN**

74.3. Set the Variable value to `C:\instantclient_19_14\Network\Admin`

74.4. Click on **OK** button and close all the opened windows

75. In the hosting PC, open a new command prompt window and issue the following command to make sure that the `TNS_ADMIN` variable is set.

We need to open a new command prompt window so that the variable values changes take effect.

```
echo %TNS_ADMIN%
```

76. Run the following code to verify that the installation and the network configuration work fine.

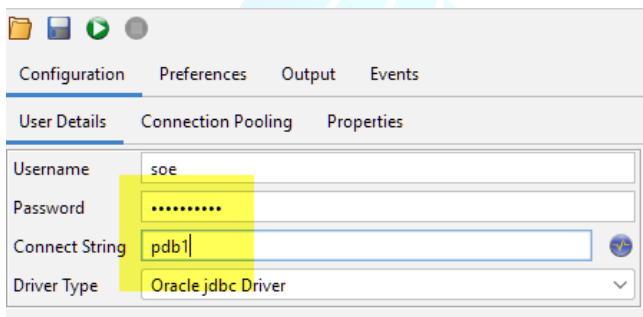
```
sqlplus system/ABcd##1234@PDB1
```

77. Exit from SQL*Plus.

78. Go to the Swingbench `bin` folder and run Swingbench again. Open the **SOE Client side** option.

```
cd C:\swingbench2.6\winbin
swingbench.bat
```

79. Try connecting to the database using the network service name defined in the `tnsnames.ora` file.



80. Click on **Test** Connection button to test the connection.

The test should succeed. This connection attempt used the naming connections defined locally.

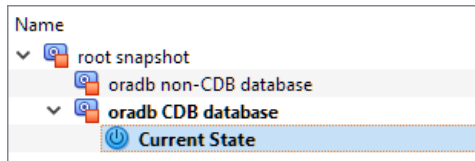
81. Run Swingbench sessions to test further that everything is going well.

82. Close the sessions and exit from Swingbench.

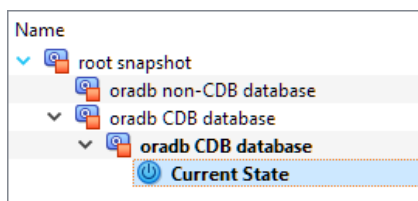
Saving the `srv1` Current State Snapshot

We want to save the changes that we made so far on `srv1` in its snapshot. Therefore, in this section of the practice, you will create a new "**oradb CDB database**" snapshot and delete the existing one.

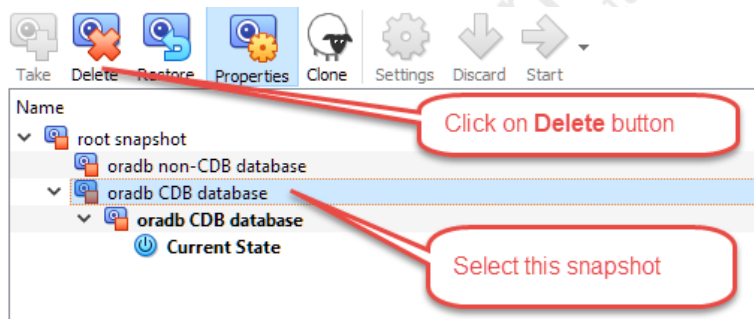
83. In Oracle VirtualBox manager, make sure the Current State is pointed under the "oradb CDB database" snapshot as follows:



84. Click on **Take** button and name the snapshot as "oradb CDB database". you should have the snapshot structure as follows:

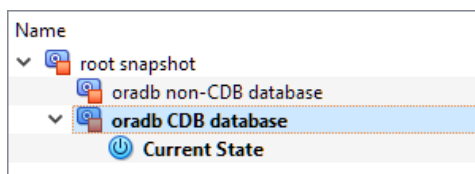


85. Select the **old** "oradb CDB database" snapshot then click on **Delete** button, as follows:



86. Click on **Delete** button on the confirmation message.

87. The Snapshot structure should now look like the following:



Summary

- Easy Connect method allows us to connect to Oracle database with no Oracle Net configuration on the client side. But it must be enabled in the database side.
- Using `netca` utility or by directly editing the `tnsnames.ora` file, we can manage the naming connection names in the `tnsnames.ora` file at the database side.
- Using `netca` utility or by directly editing the `sqlnet.ora` file, we can control which connection method is available in the database side.
- Using `lsnrctl`, we can control and manage the Oracle Net Listener
- We installed Swingbench in the hosting PC to use it as a sample application in this course.
- If a client application wants to connect to an Oracle database using Naming method, it should have the Oracle Net installed in the client machine. The easiest way to have Oracle Net in a machine is by installing the Oracle instant client.



Ahmed Baraka
Oracle Database Administrator