**Lab -Performing Database Recovery**

**Please ensure you have completed the previous lab work on Recovery and Backup Settings before starting this lab.**

**Background:** Many failures of the Oracle database can be traced to some sort of media failure,

such as disk or controller failure. Let us now recover your database from a variety of simulated media failures:

1. Recover from the loss of a control file.

2. Recover from the loss of a redo log member.

3. Recover from the loss of an application data file.

1. **Recover from the loss of a control file.**
2. Firstly make a copy of the CONTROL01.ctl and put in on the desktop for later recovery you will find this on **/home/oracle/app/oracle/oradata/orcl**
3. Now we need to shut the database down.

First logon using SQL\*PLUS in the DOS window, this is the TERMINAL on your desktop in the VM.

Graphical user interface, application

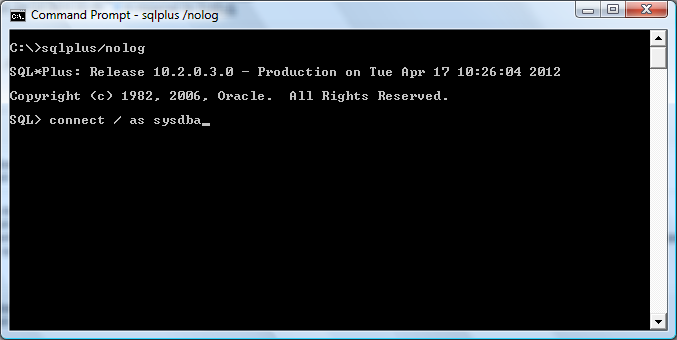
Description automatically generated

Type in **sqlplus /nolog;** Note there is a space between sqlplus and the forward slash.

connectHIT ENTER

You then get an **SQL >** ( prompt)

Then type in **connect / as sysdba;** Note there is a space between connect and the forward slash. **SEE SCREEN BELOW.**



When connected shutdown the database by issuing the command:

**SQL> shutdown immediate;**

|  |
| --- |
| **INSERT SCREEN HERE:**  **Text  Description automatically generated** |

**JUST AS A TEST FOLLOW THIS WITH THE COMMAND**

**SQL > STARTUP;**

|  |
| --- |
| **INSERT SCREEN HERE:**  **Text  Description automatically generated** |

You have now successful shut down the database and started it up again using the SQL commands on the interface.

Now shutdown the database again (shutdown immediate)

1. Rename the control file CONTROL01.ctl to TEST CONTROL01.ctl to simulate a loss or a corrupted file you will find this file on ***/home/oracle/app/oracle/oradata/orcl***
2. The Help desk begins receiving calls saying that the database is down. Troubleshoot and recover as necessary. Use SQL\*Plus, startup the database. Use the command **startup;**
3. Show screen shot of error received below.

|  |
| --- |
| Answer:  Text  Description automatically generated |

**f)** The startup of the instance fails with Enterprise Manager, and you can get no other information to explain the failure. So you will now use the command-line tools.

* Connect to the instance with SQL\*Plus as sysdba and check the current status of the instance by issuing the command:

***select status from v$instance;***

Insert the message you get below.

|  |
| --- |
| Answer: |

* The instance status is STARTED, which means that the database is in the NOMOUNT stage. Attempt to mount the database by entering this:

SQL> alter database mount;

What error message do you get?

|  |
| --- |
| Answer:  A picture containing text  Description automatically generated |

**e)** The instance cannot move to the MOUNT stage because it cannot find one of the

control files. Check the last 10 rows of the alert log to see which control file is the

problem. What did it tell you? It can be found at

The alert log can be found here: ***/home/oracle/app/oracle/diag/rdbms/orcl/orcl/trace***

Make sure to scroll to the end to see error.

|  |
| --- |
| Answer:  Text  Description automatically generated |

**f).** Restore the missing control file by renaming an existing control file, from where you changed the name earlier then mount and open the database. E.g. ALTER DATABASE MOUNT, ALTER DATABASE OPEN. Summarise the steps you took.

|  |
| --- |
| Answer:   1. Went to directory on ***/home/oracle/app/oracle/oradata/orcl*** 2. *Found the testcontrol01.ctl file* 3. *Named it back to control01.ctl* 4. *Issued the alter command got the result below*   *Text  Description automatically generated*   1. Then issued the command alter database open and got the result below   Graphical user interface, text  Description automatically generated |

**g)** Why did you have to use two commands to move the instance state from NOMOUNT to

OPEN?

|  |
| --- |
| Answer:  **This was because there was a corruption so to find out what was happening we had already started the instance to see what the status was and thus to complete the process the step of mount and open are then required.** |

**2. Recover from the loss of a redo log member.**

You are going to delete one of your redo log files to simulate a corrupt disk. You can also see the error in the alert log and the recover from it. Let us get started.

**a)** Remove one of your redo log files to simulate a corrupt file. First, let’s look at the current redo log. You can do this from your SQL\*PLUS Dos window:

select rownum "NUM", member from v$logfile where group#=(select group#

from v$log where status = 'CURRENT') order by rownum

Show the result of this command here. Ensure you understand what this command does. View the full table (Select \* from v$logfile and Select \* from v$log to see full data)

|  |
| --- |
| Answer:  shuitText  Description automatically generated |

**b)** Shutdown the database and now Delete one of the members

SHUTDOWN IMMEDIATE;

**c)** Now delete (or rename)one of the members of the online REDO Log at ***/home/oracle/app/oracle/oradata/orcl***. After deleting a file now open the database for

business

STARTUP OPEN;

**d)** The database continues to function normally, and no users are complaining. But during

a routine check of the alert log for errors, you notice the following message:

Check the last 25 rows of the alert log to see which redo logfile is the problem. What did it tell you? It can be found at

***/home/oracle/app/oracle/diag/rdbms/orcl/orcl/trace***

- Note the error about the missing redo log file:

|  |
| --- |
| Answer:  Text  Description automatically generated |

**e)** Find and resolve the problem.

- Examine the .trc file reference in the alert log to identify the redo log that is missing and get more detailed information on the error.

**f)** Exit SQL\*Plus, and copy the remaining member of the log group to replace the missing log file (You need to substitute the appropriate file name.). Make sure you copy the correctly numbered file.

**g)** *Question:* Why doesn’t Enterprise Manager produce a critical alert for a missing log file?

|  |
| --- |
| Answer:  **This is because the log file is not a system critical file. The database can run without it or on a reduced number of logfiles.** |

**3. Recover from the loss of an application data file.**

1. The Help desk has received a call from a user who is unable to access the COUNTRIES table in the SH application schema. The SH schema is stored in the Example tablespace physically stored in ***/home/oracle/app/oracle/oradata/orcl*** /**example01.dbf**.
2. Find then countries table in Enterprise Manager. Screen shot this below.

|  |
| --- |
| Answer:  Graphical user interface, table  Description automatically generated |

**Logout of Enterprise manager.**

**b)** Let us simulate a data loss. **Shutdown** the data base in SQL PLUS, then rename **example01.dbf**. and call it **example01HOLD.dbf**.

- connect to the DB in SQLPLUS as follows: **connect sys / as sysdba;**

**-** you will be prompted for you password which is **oracle.**

**-** you should be connected to an idle instance now.

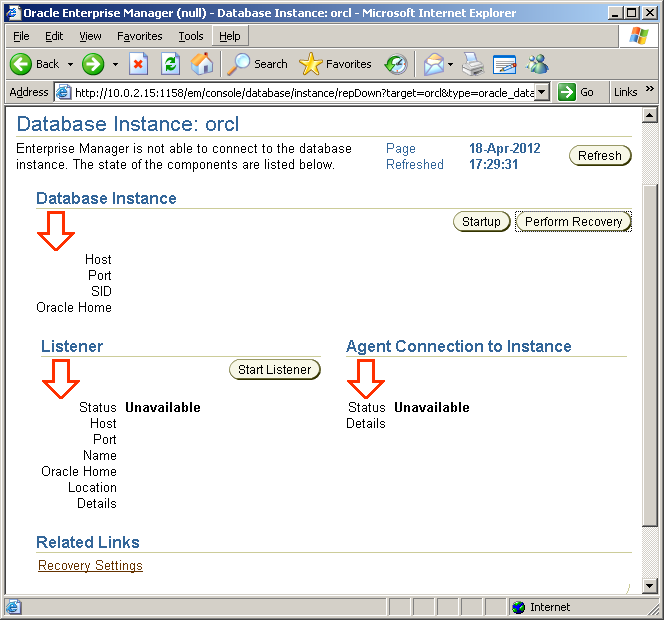
- mount the database : ***startup mount;*** *(if the instance is running shut it down first)*

- Open Enterprise Manager.

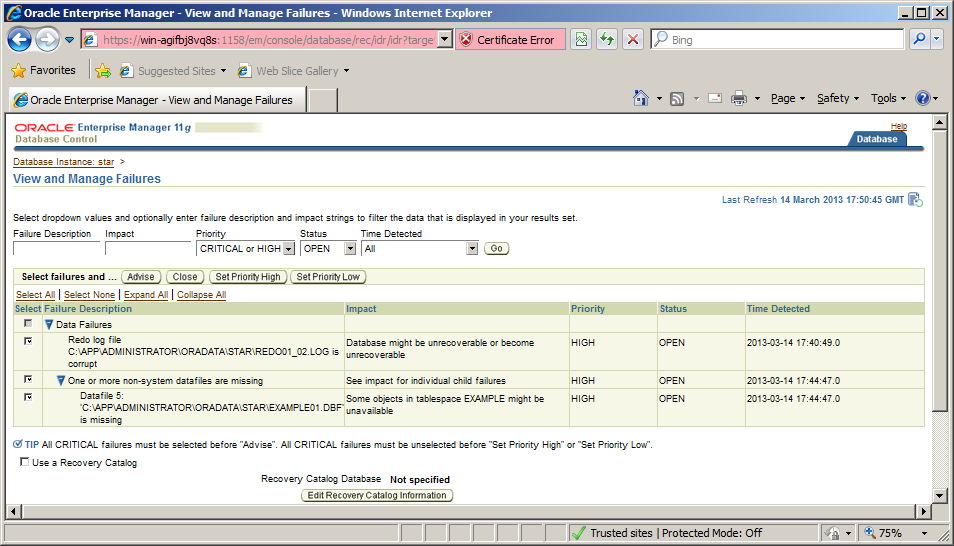
- Enter the host credentials i.e. oracle/oracle

**c)** Recover the data file to the current time, specifying the missing data file to be

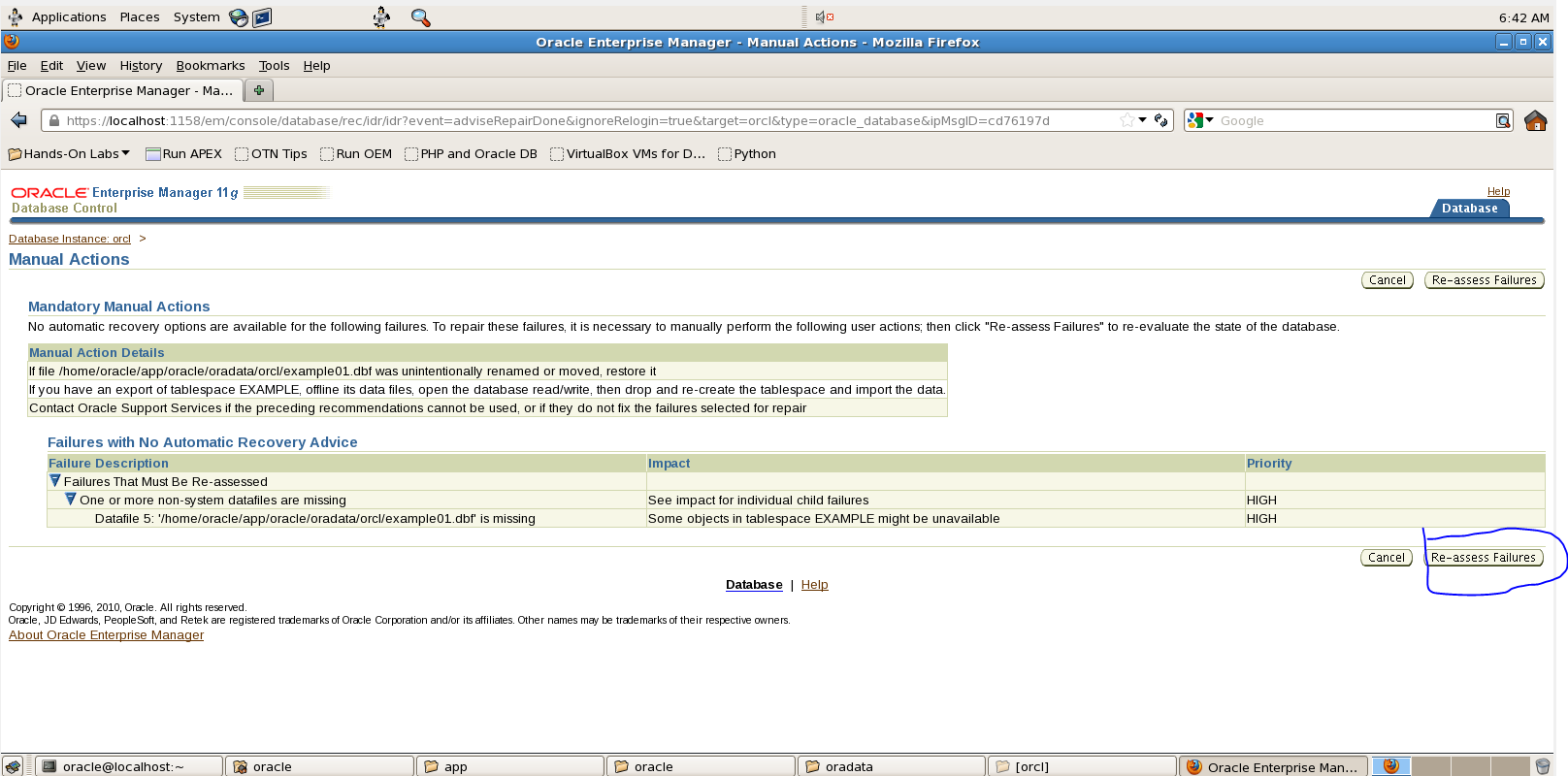
recovered. In Enterprise Manager, select **Perform Recovery**.



* Login as requested with **host** and **system credentials**
* Click Advise on Recover button
* On the View and Manage Failures page, click the plus(+) icon under the failure description. You should see a failure like the following:



* Click the **Advise button.**



* + You will need to manually fix the example01.dbf by renaming it back to it’s original name. Leave this window but leave it open and the database running.
  + When the file is re-named click on the **re-assess button**.
  + Now go to the main page and **startup** the database. In effect you are only opening the DB as it is already mounted. You can do this in SQL PLUS either if you wish.
  + Once the DB is open then Login to SQLPLUS and Verify that the COUNTRIES table is now accessible.

SQL> select count(\*) from HR.COUNTRIES;

Insert the result of this command here:

|  |
| --- |
| Answer:  Text  Description automatically generated |

**Note:** we have recovered the user datafile without shutting down the database