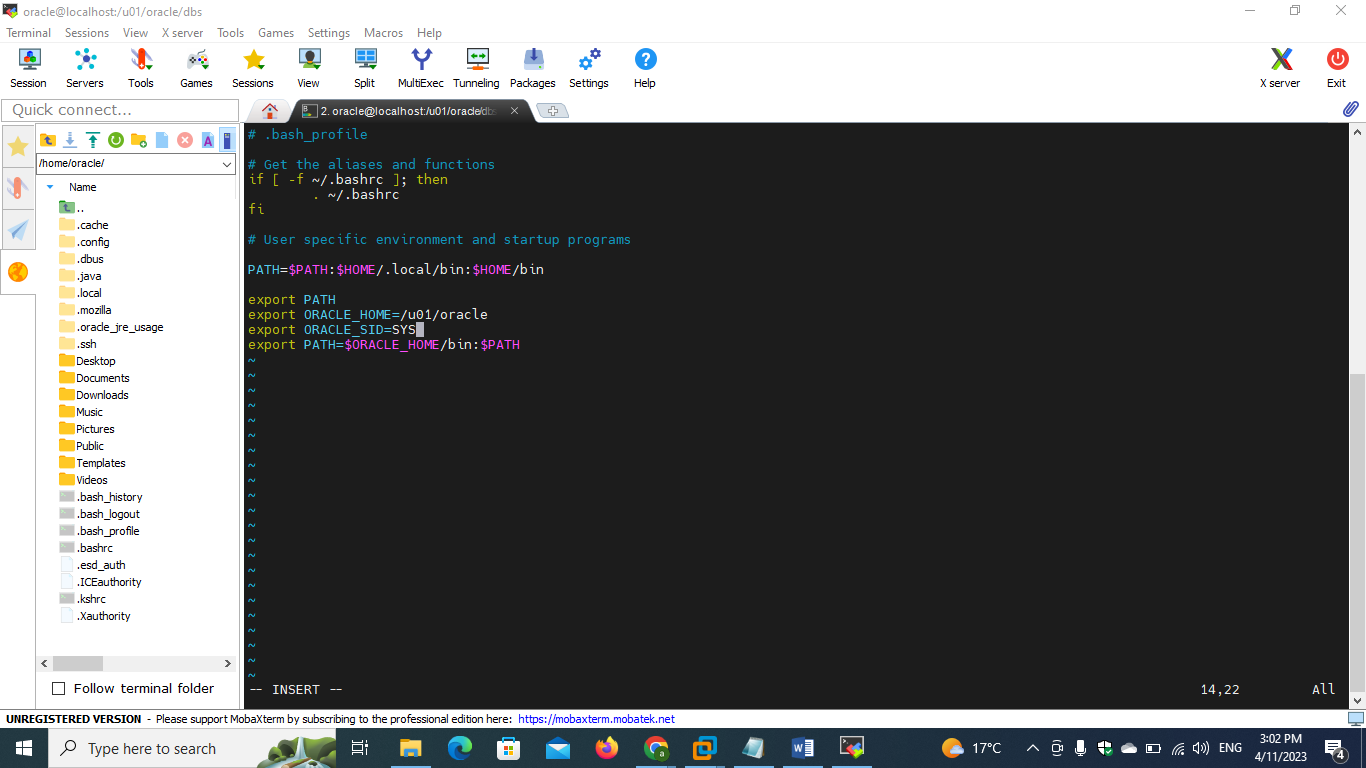
**Name : Aya Allah Ali Abbas Track : System Admin**

**Lab1**

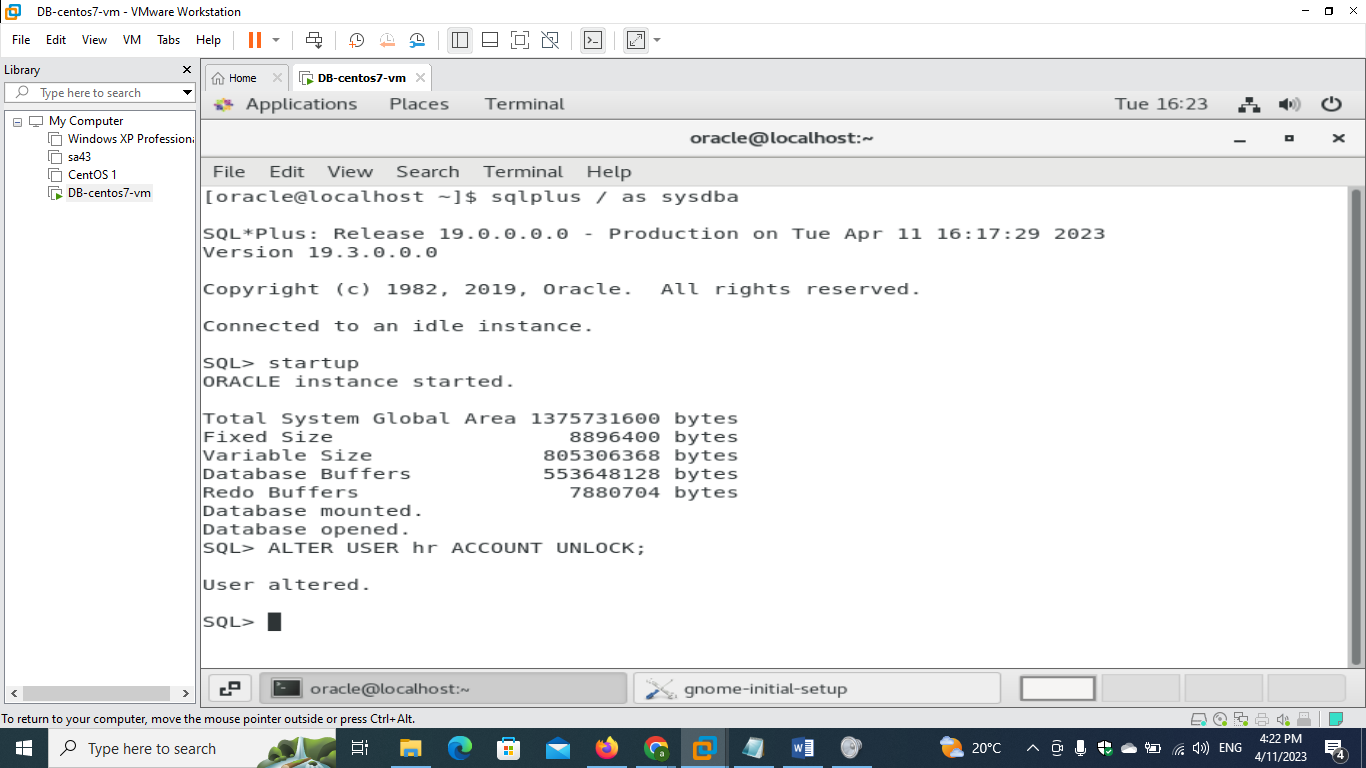
**First export env variables**

**vi ~/.bash\_profile**

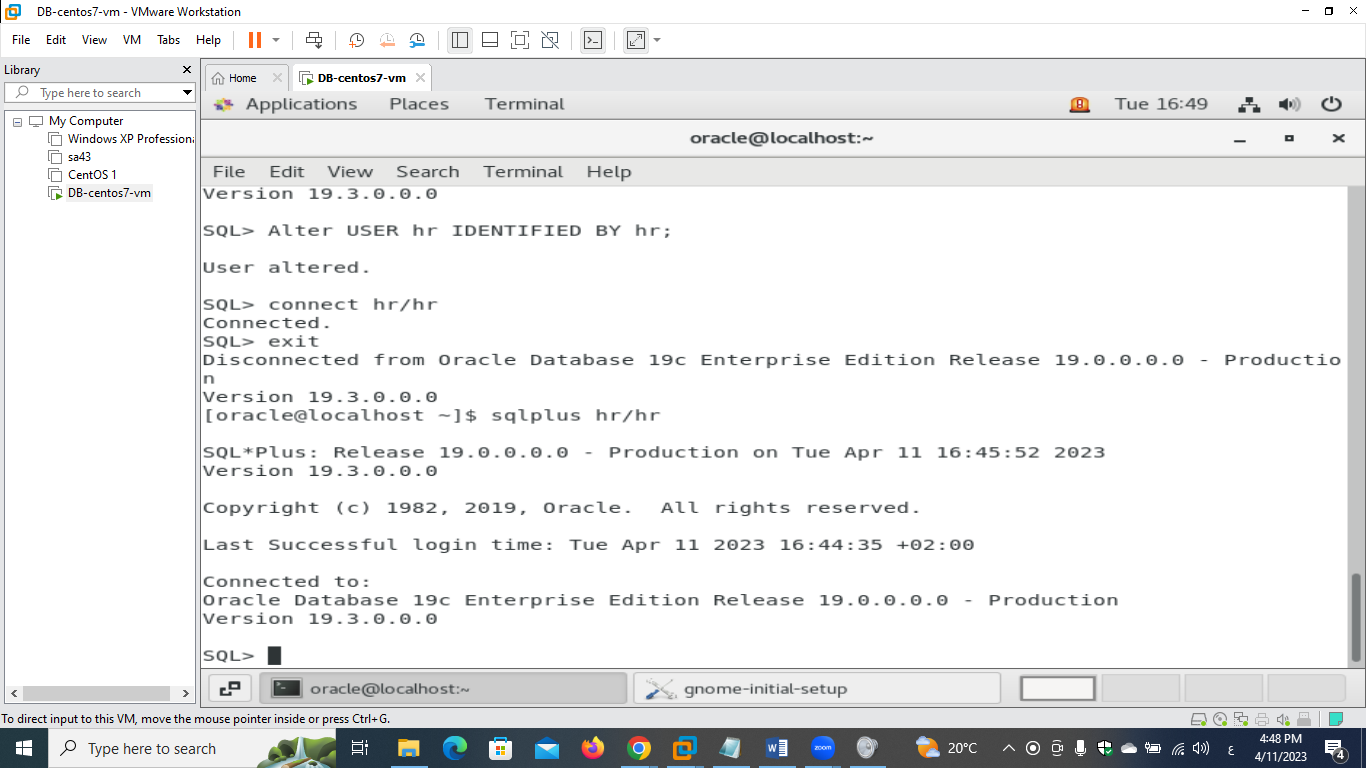


**1)**

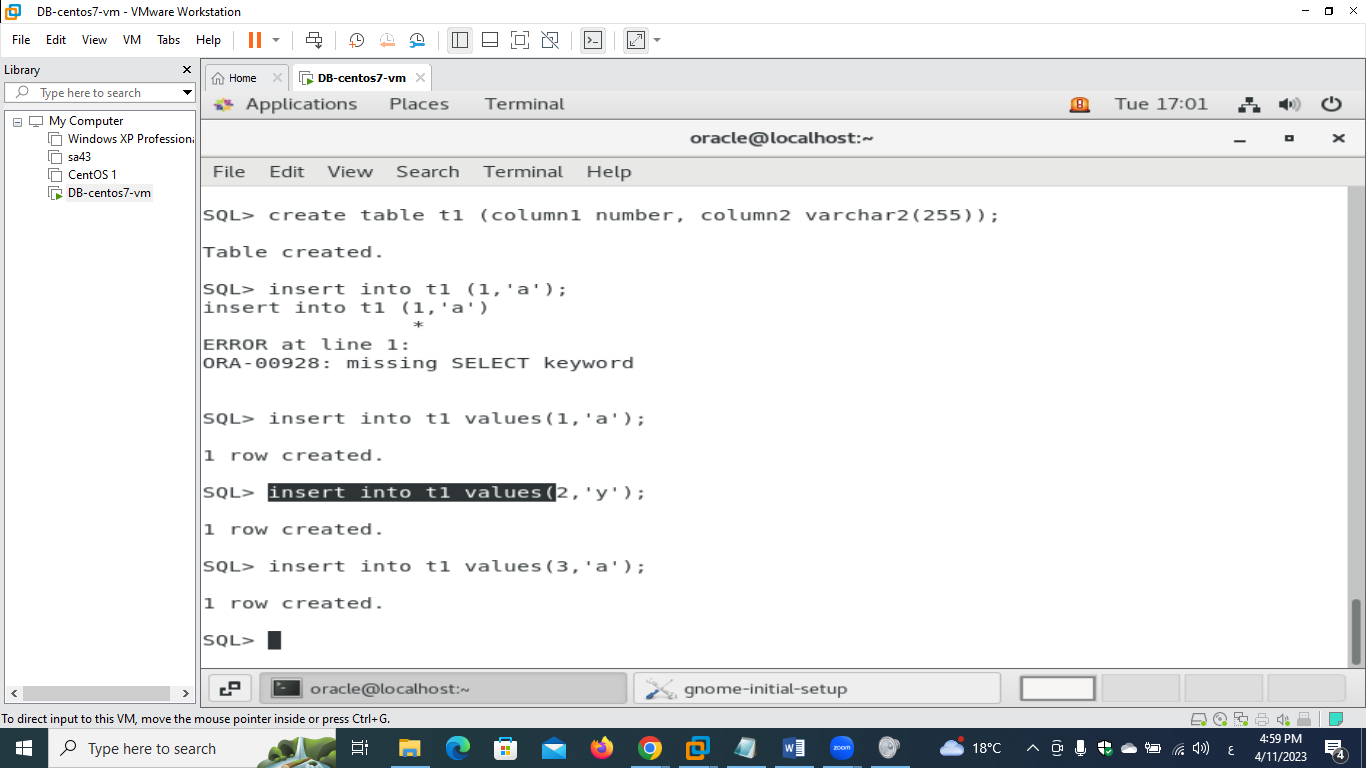
**- unlock hr user**



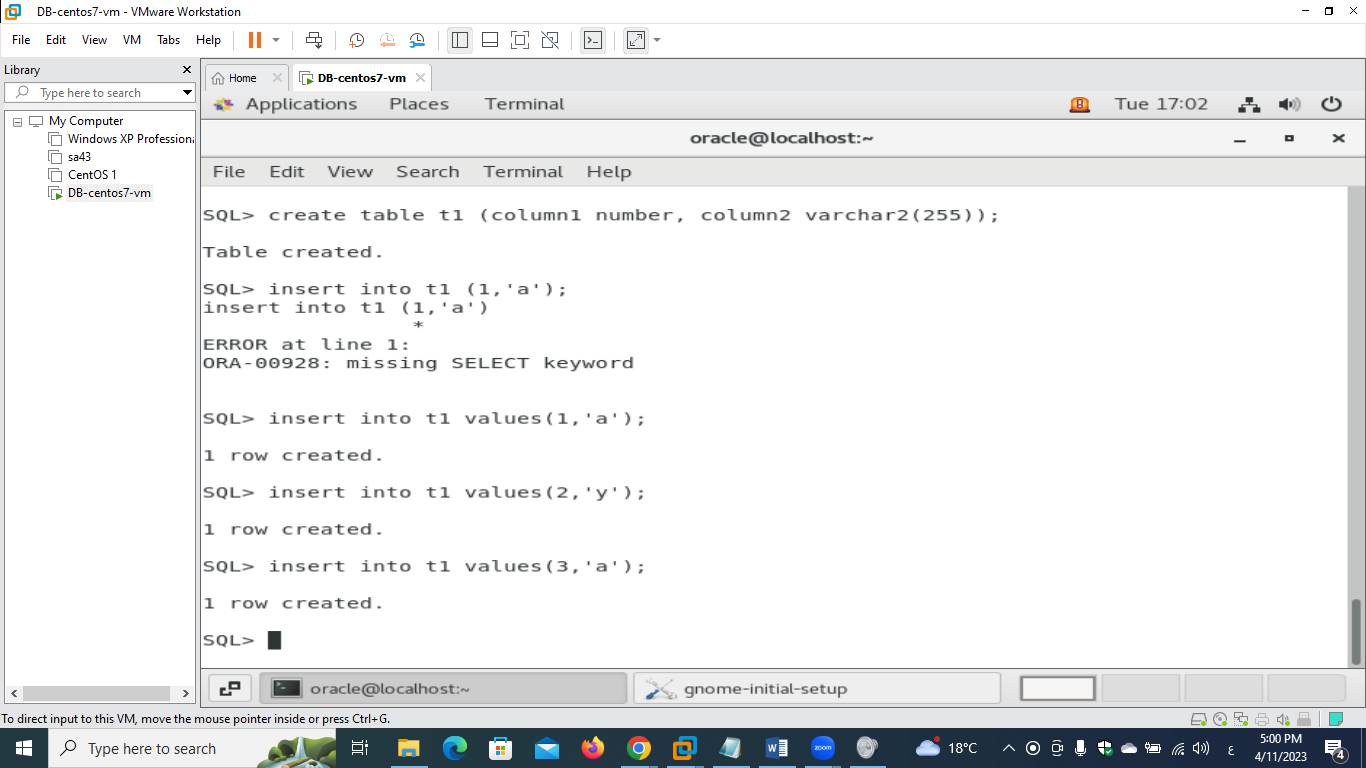
**- connect as HR user**



**- create table t1 with two columns , column 1 number , column2 varchar2**

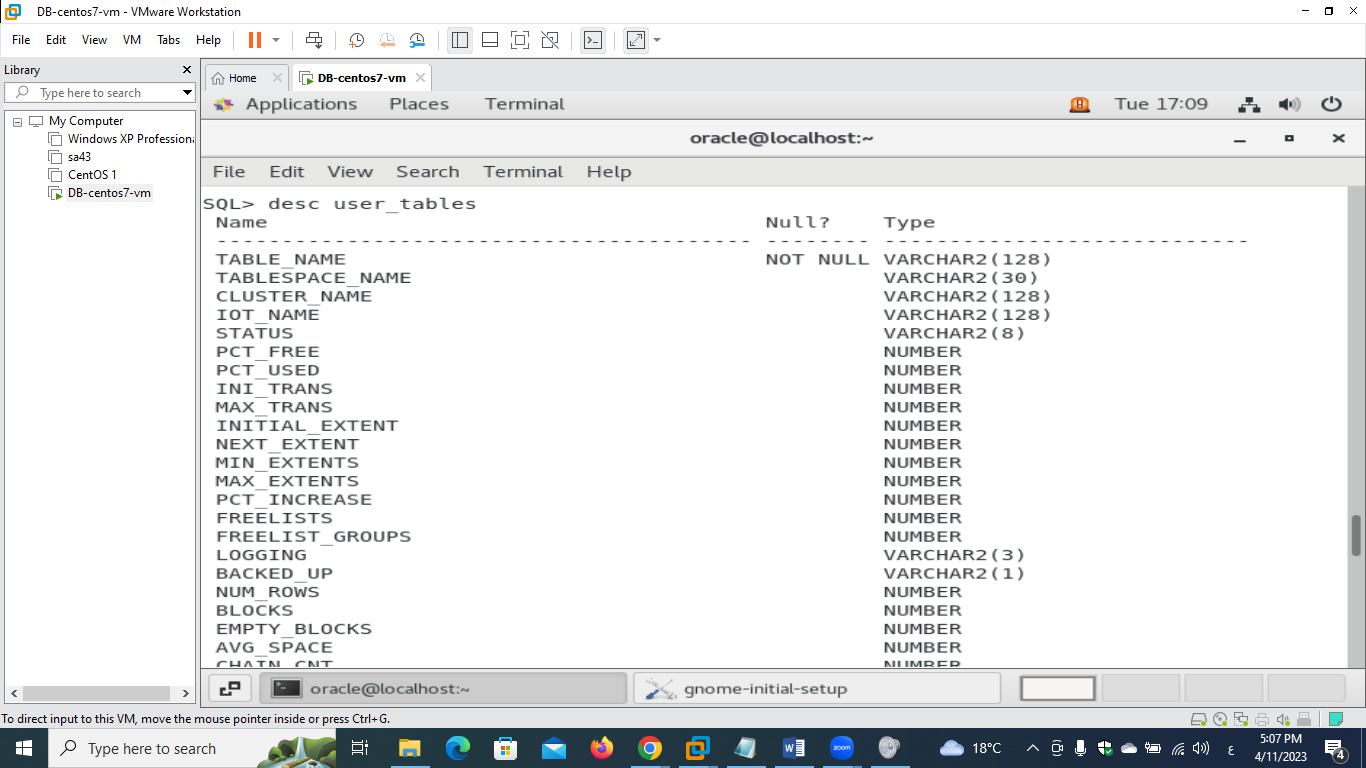


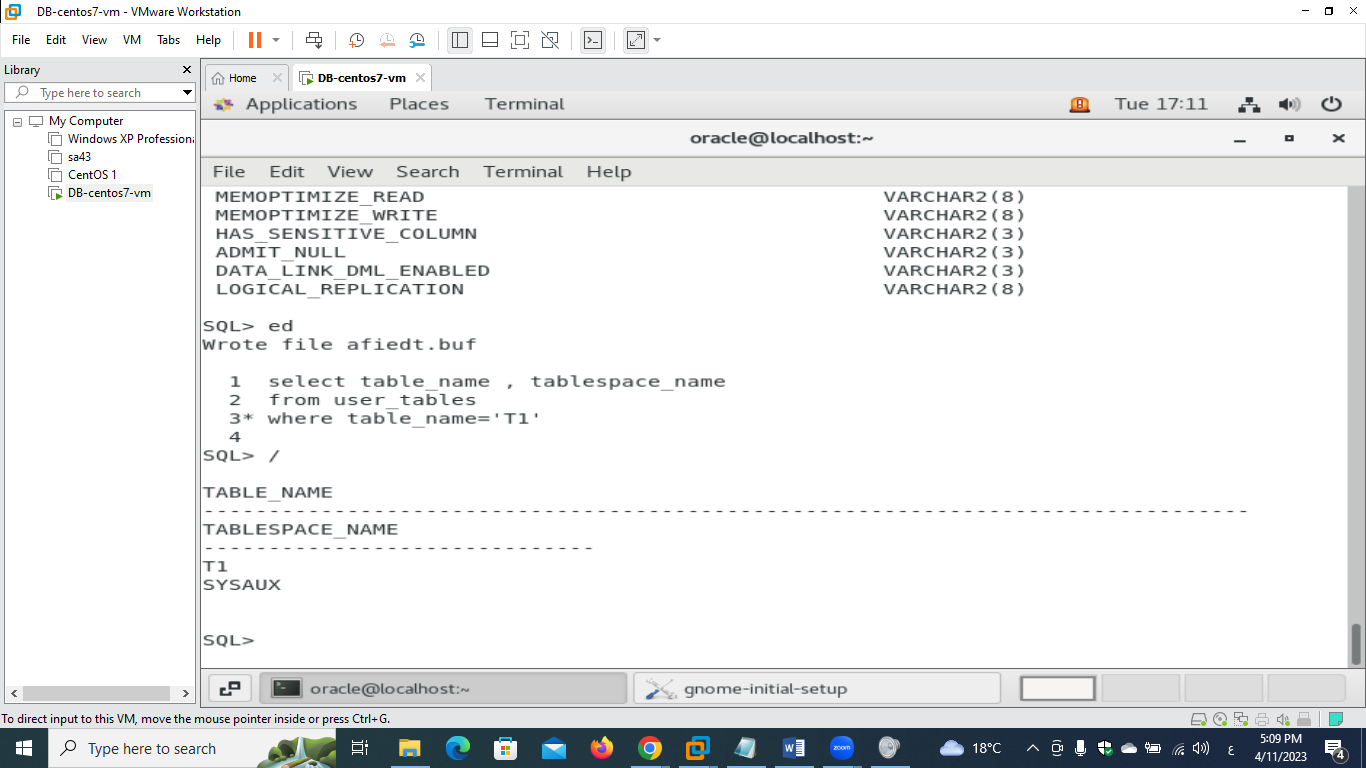
**- insert 3 rows in this table**



**- find information related to this table, and in which tablespace it saved.**

**Use desc to see the name of the required col name from user\_tables which contains info about all the created tables**

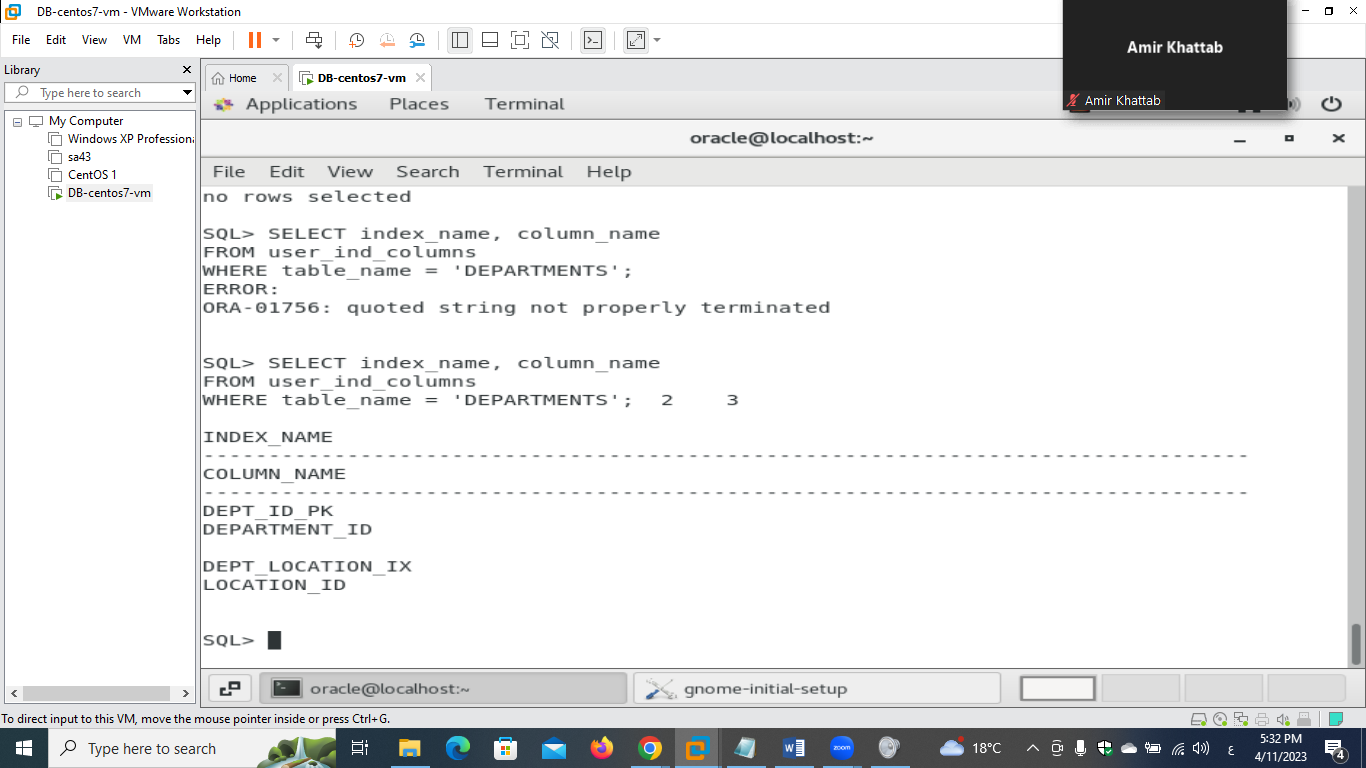




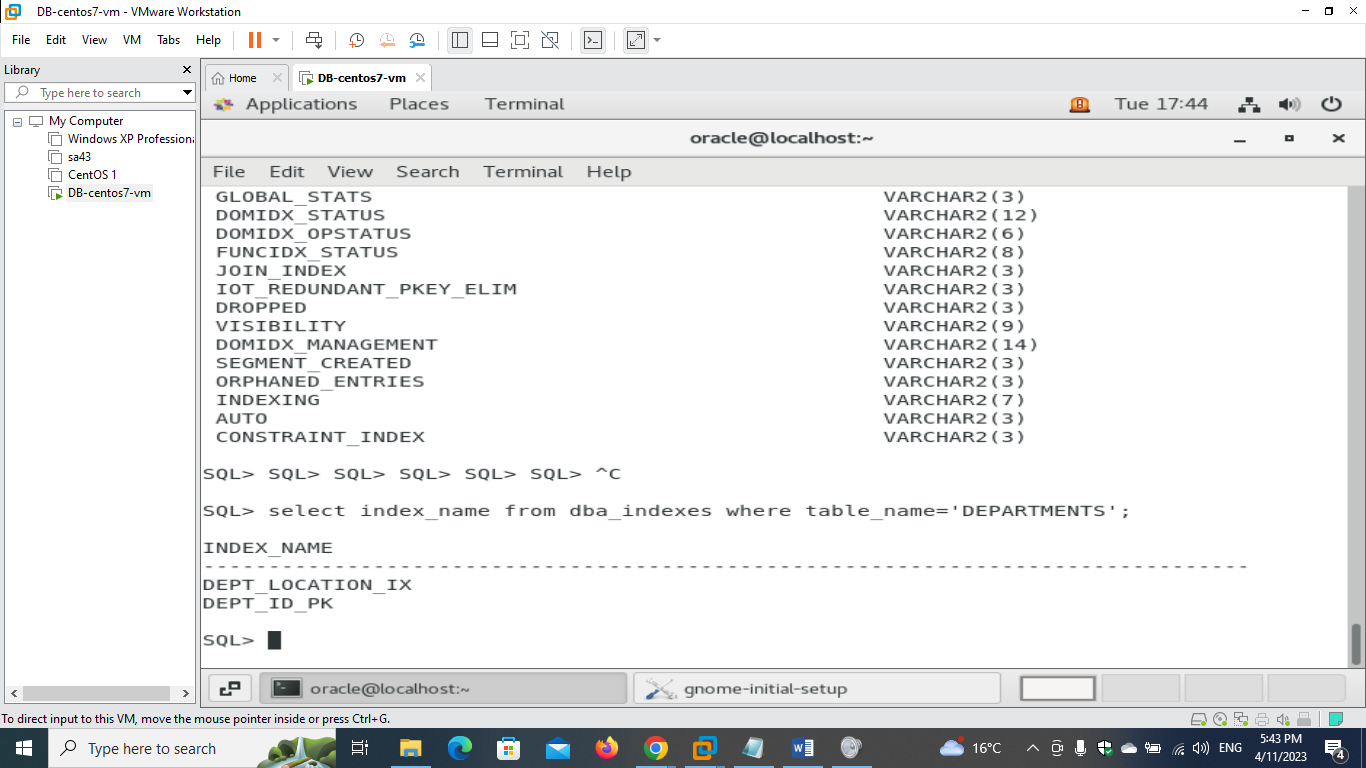
**2)**

**- find the indexes for table hr.departments**

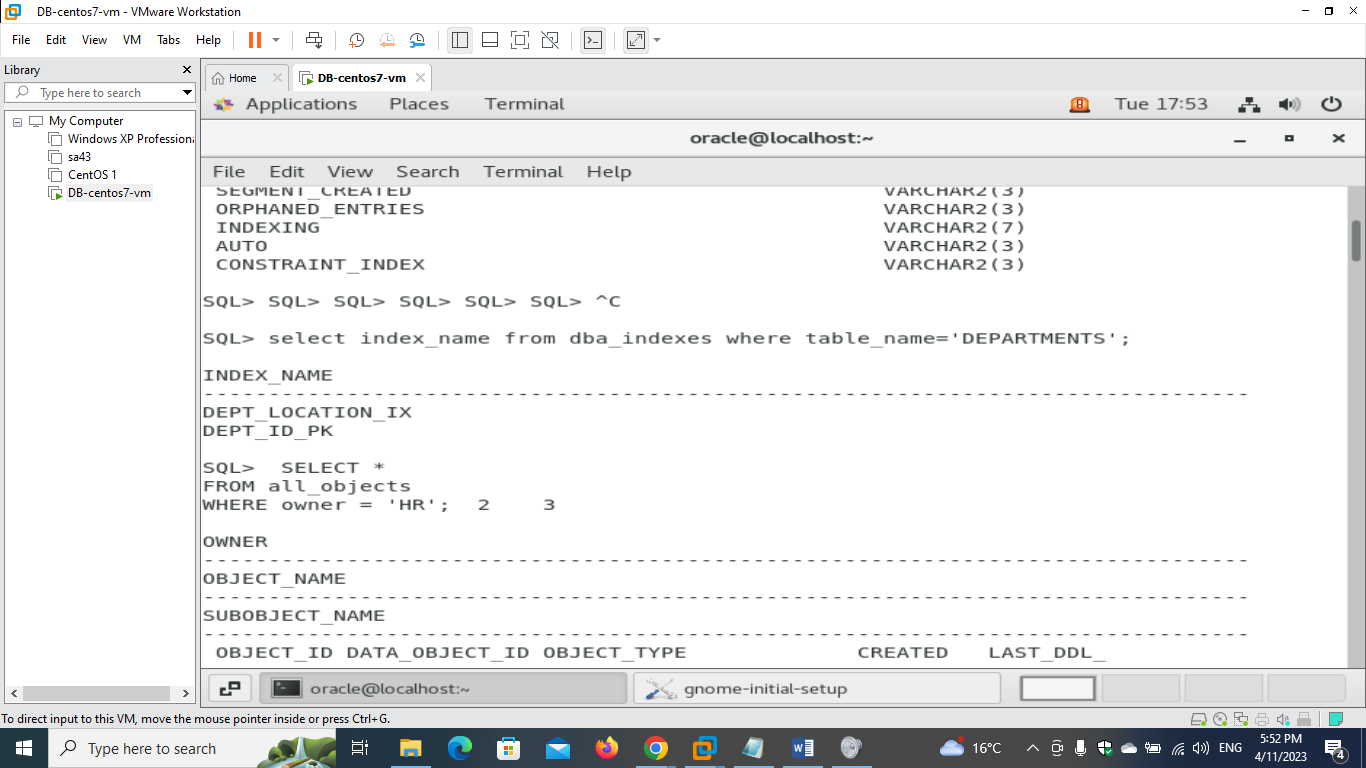
**As hr user**

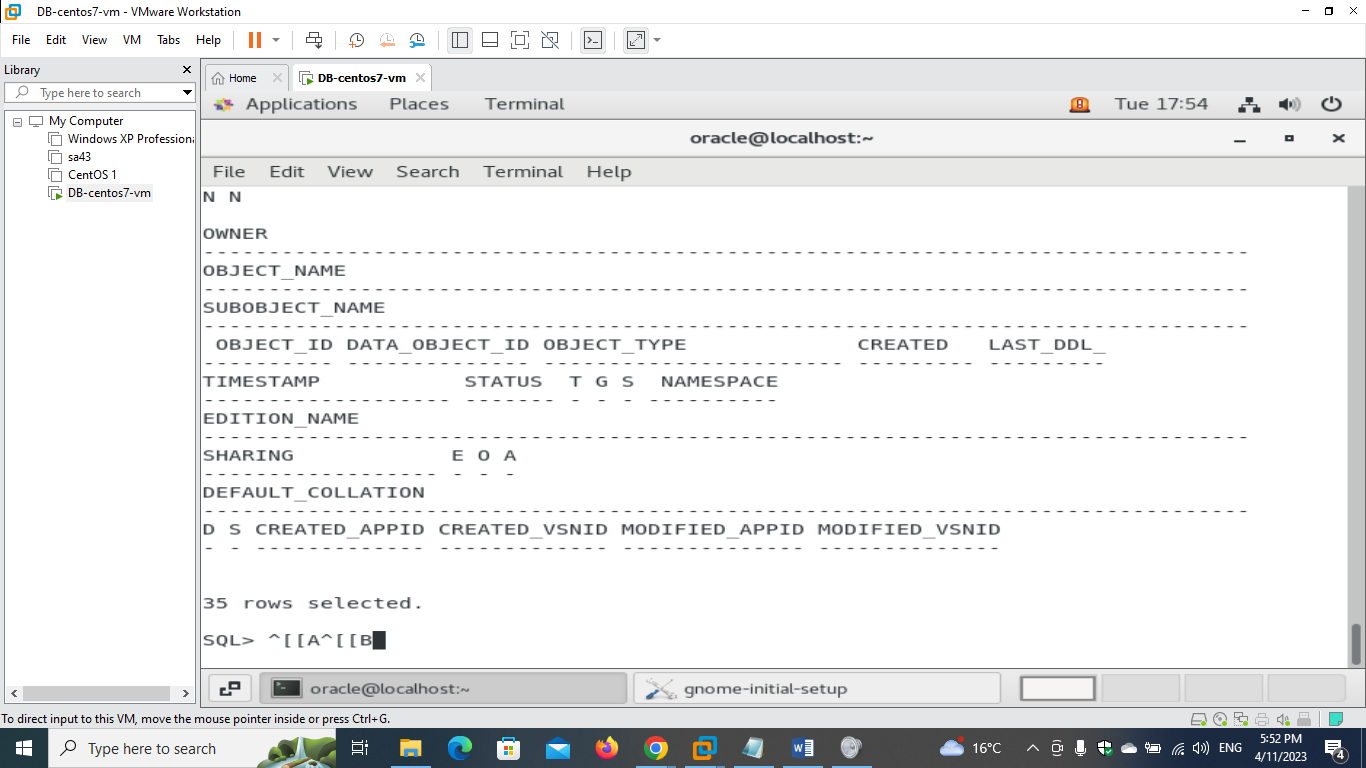


**As sysadmin**



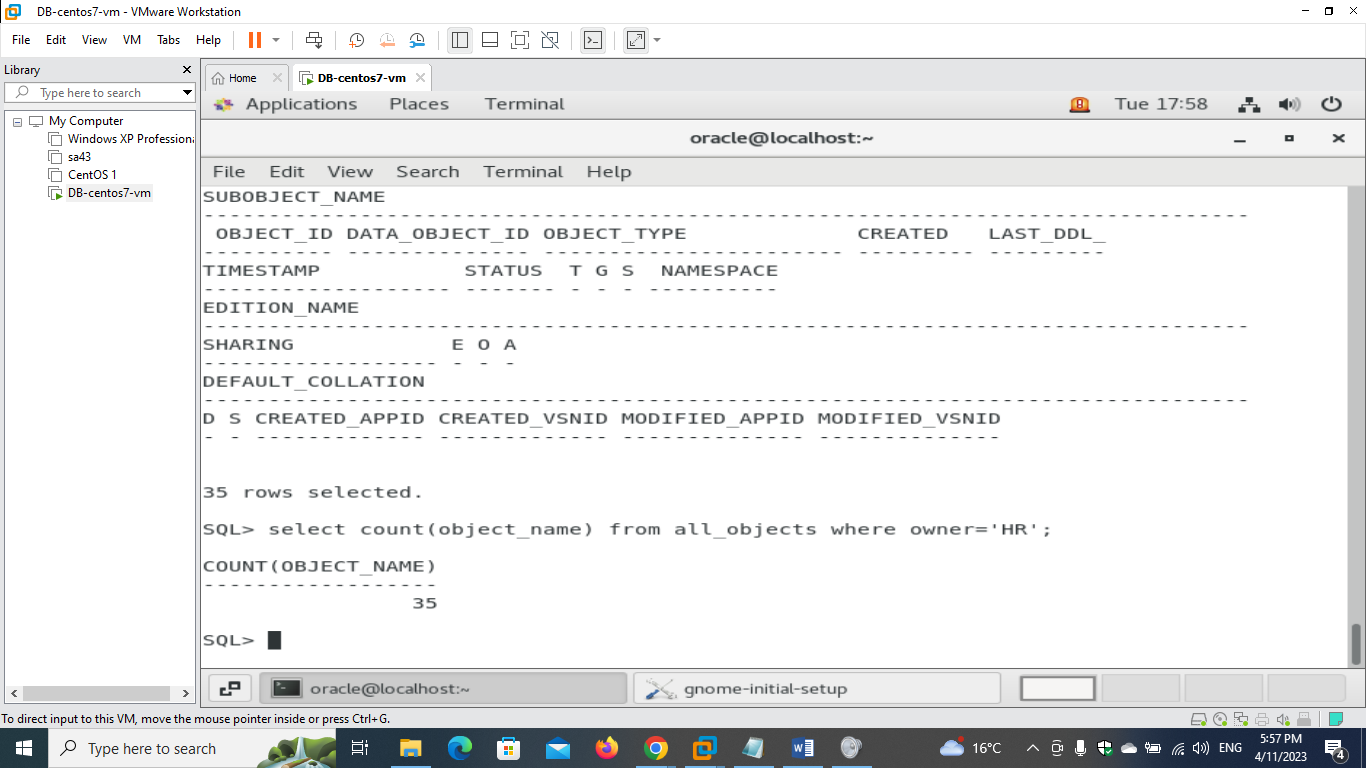
**- find all objects related to HR schema**





**As the output was large I used the following cmd to show count**

**select count(object\_name) from all\_objects where owner='HR';**



**- rebuild one of this indexes "****DEPT\_ID\_PK"**

**I connected back as hr to rebuild this index**

**I could have used hr. DEPT\_ID\_PK if I wanted to continue using sysdba.**

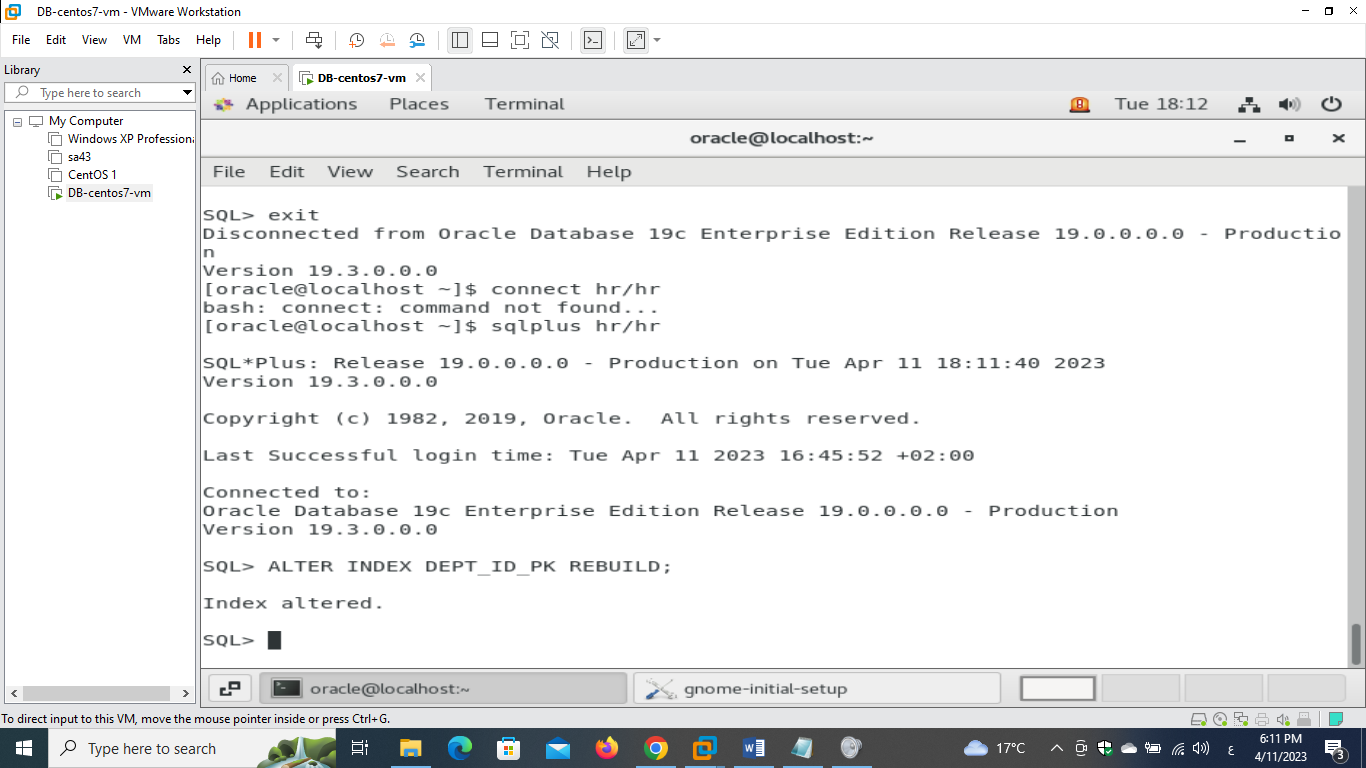
**To check if the index exists**

**SELECT index\_name FROM user\_indexes WHERE index\_name = 'DEPT\_ID\_PK';**



**The index is shown**

**Now rebuilding the index**



**3)**

**- change parameter audit\_trail from none to DB.**

**(changing the AUDIT\_TRAIL parameter to "DB" will enable database-level auditing, which logs all auditable events to the database audit trail)**

**Using this cmd**

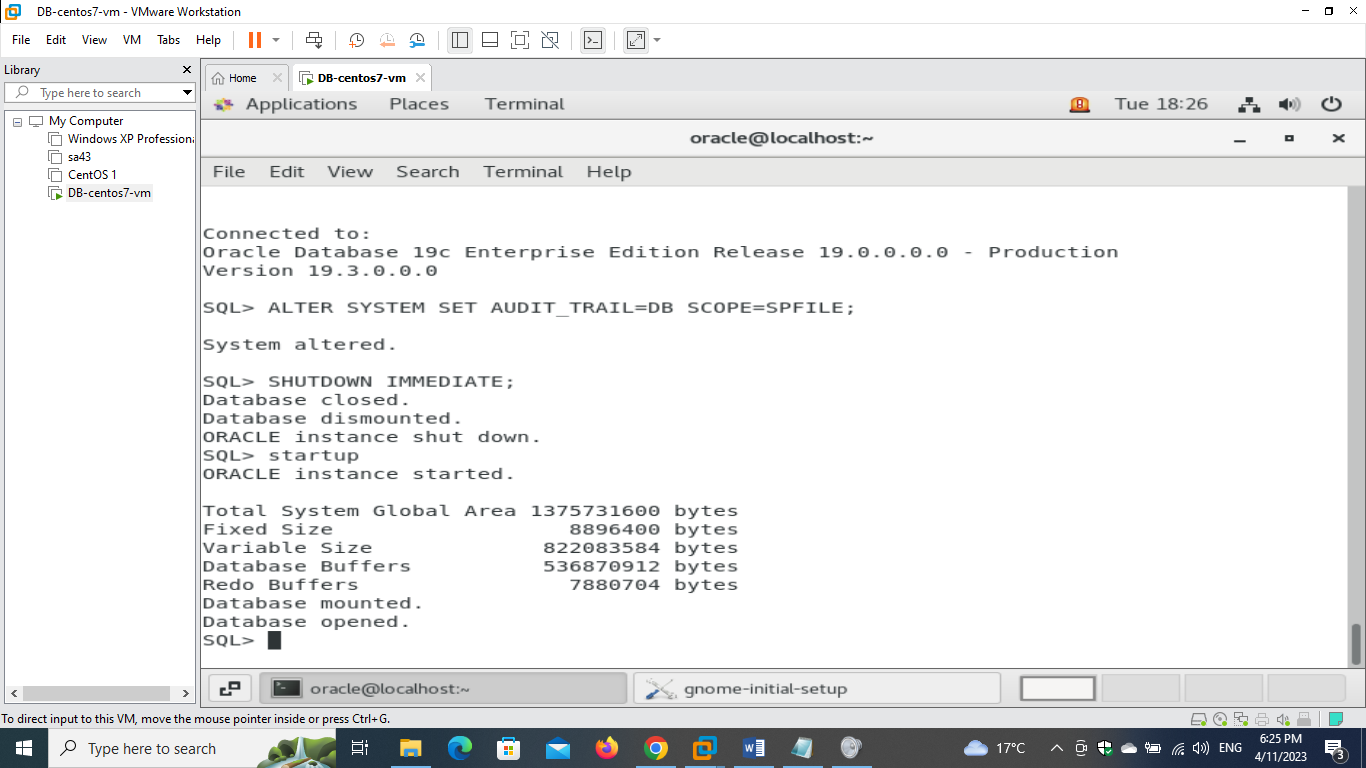
**ALTER SYSTEM SET AUDIT\_TRAIL=DB SCOPE=SPFILE;**

**This command changes the value of the AUDIT\_TRAIL parameter to "DB" and sets the scope to the server parameter file (SPFILE), which means that the change will persist even after the database is restarted.**

**After executing this command, I will need to restart the database for the changes to take effect. Using the following SQL command:**

**SHUTDOWN IMMEDIATE;**

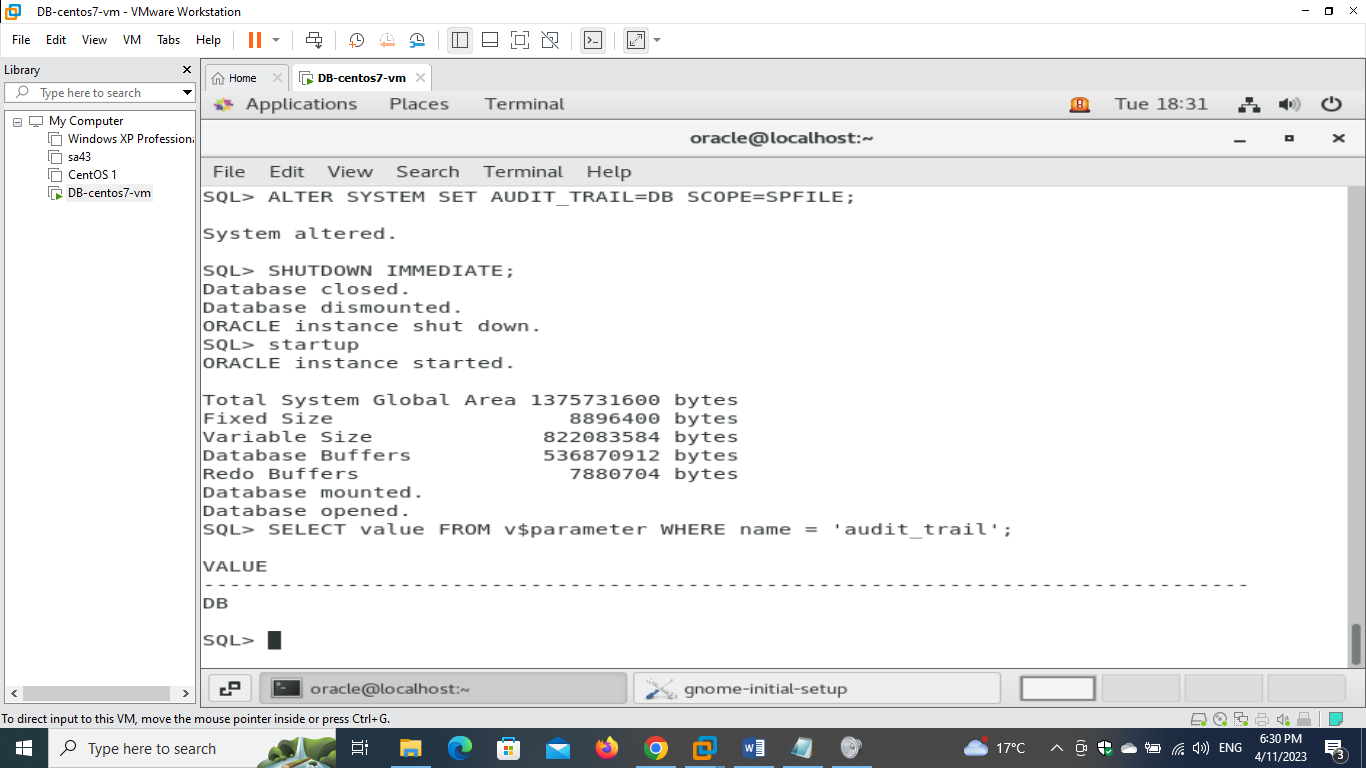
**STARTUP;**



**- restart the database to find the changes**

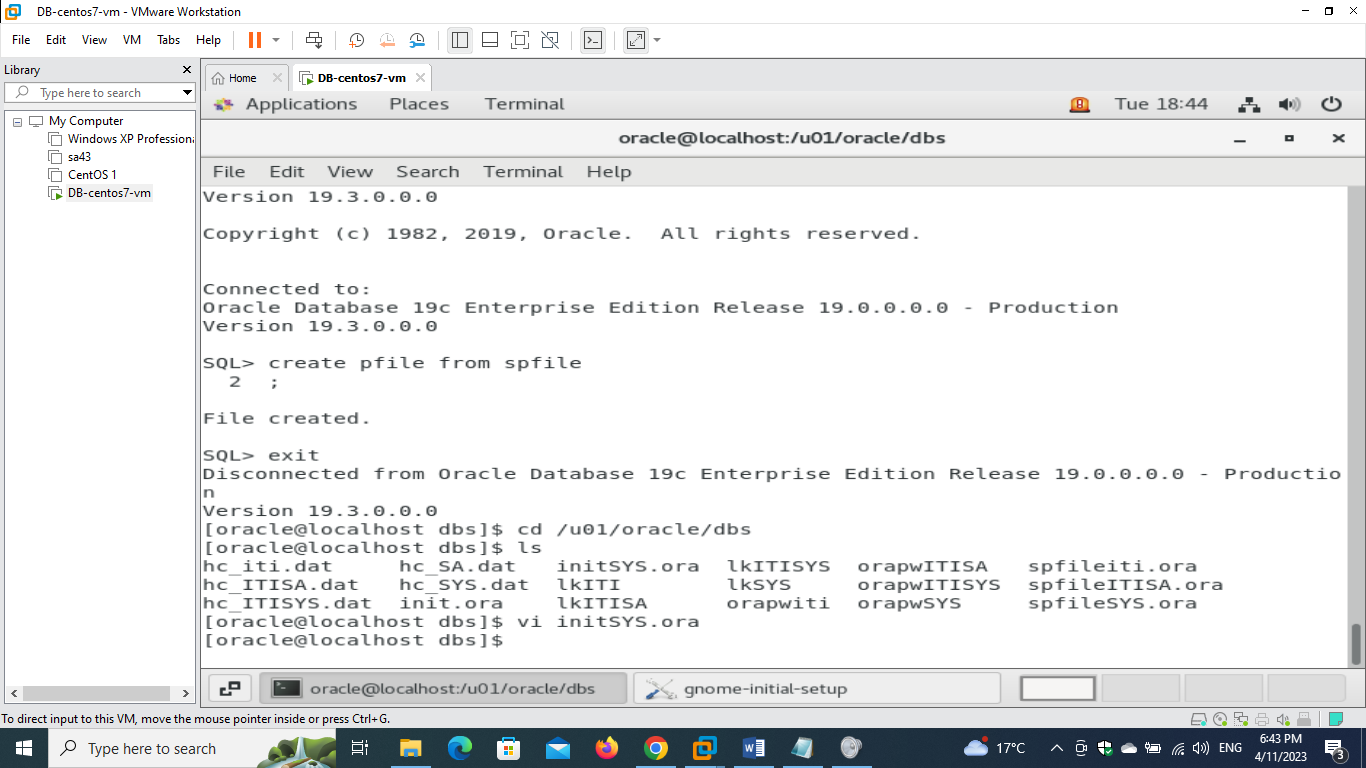
**Verify that the parameter has been changed after startup**

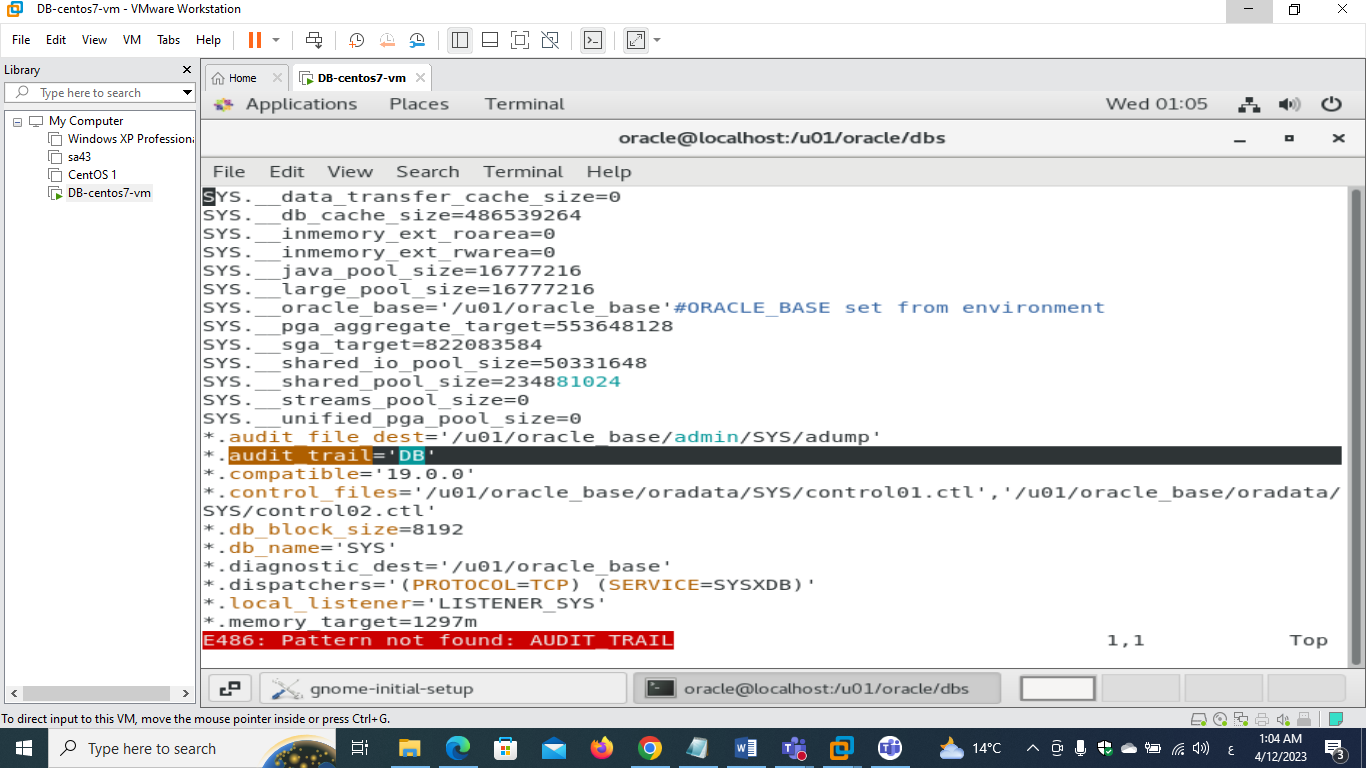
**SELECT value FROM v$parameter WHERE name = 'audit\_trail';**



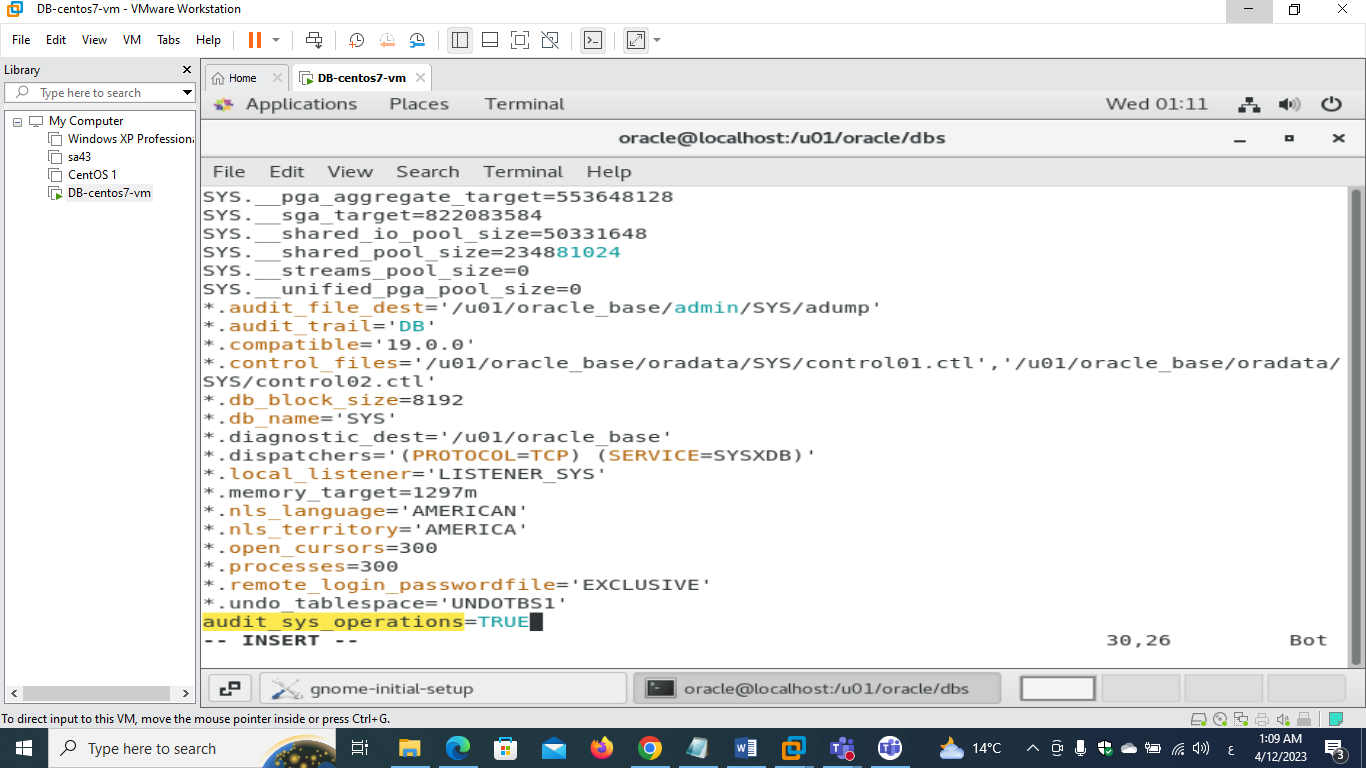
**After completing these steps, the database will be running with the AUDIT\_TRAIL parameter set to "DB", which will enable database-level auditing and log auditable events to the database audit trail.**

**- create pfile from the spfile and find the change you made on it.**



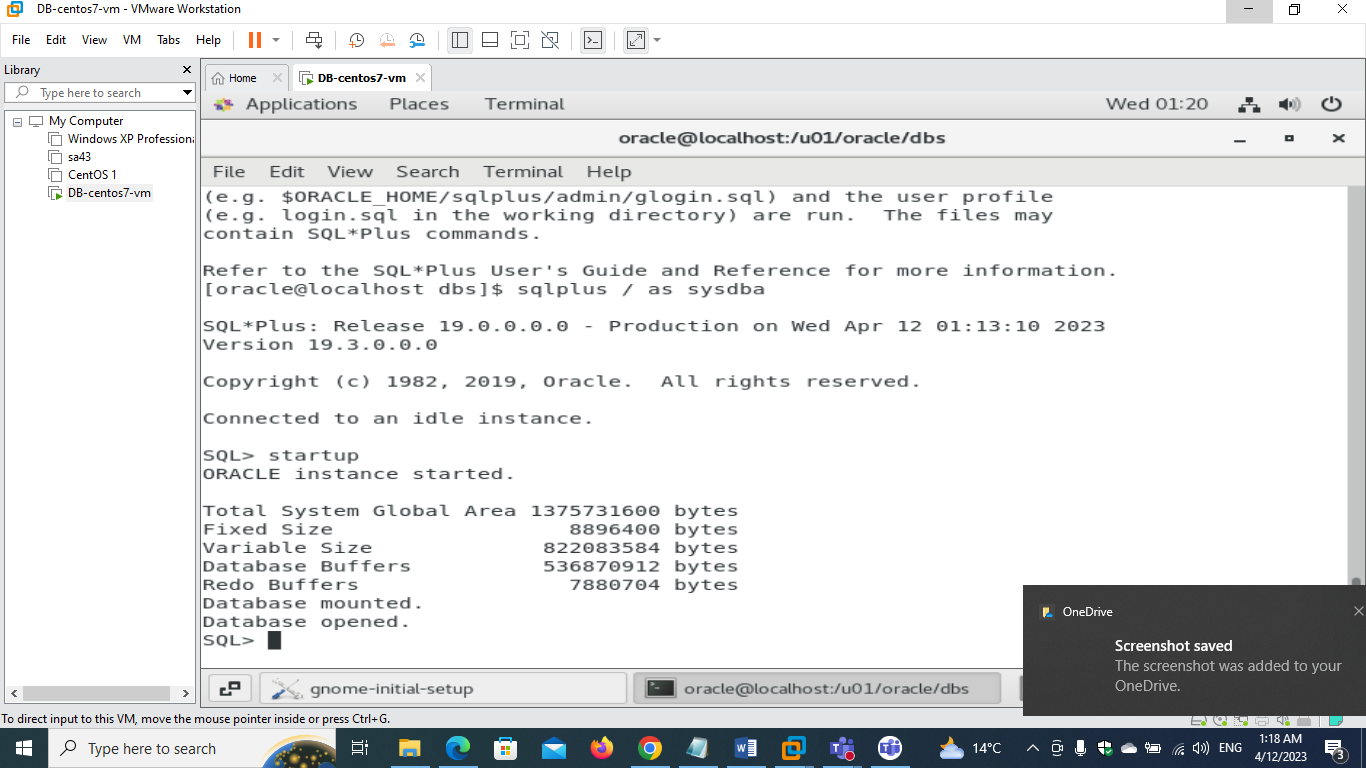


**- edit the pfile and make the audit\_sys\_operations=true**

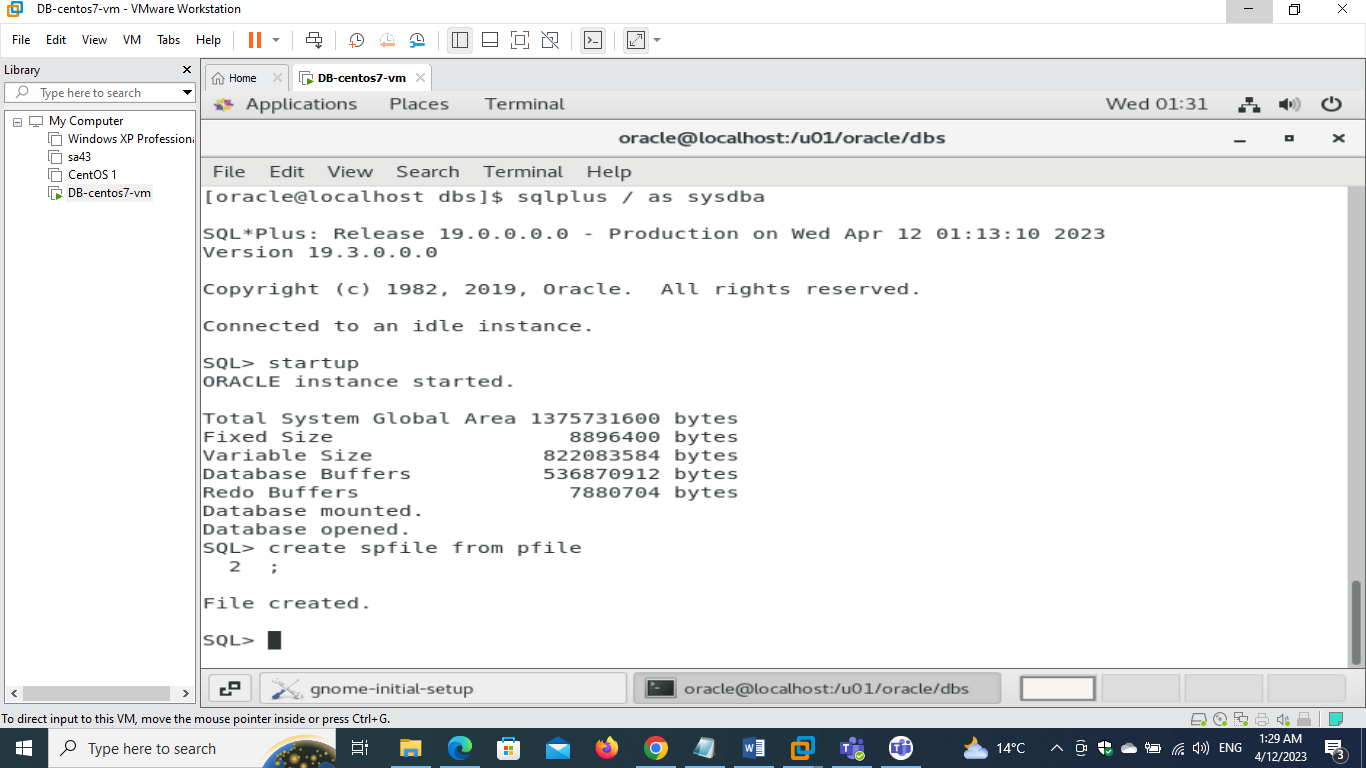


**- start the database using this pfile.**

**Remove spfile and shutdown and then startup the database**



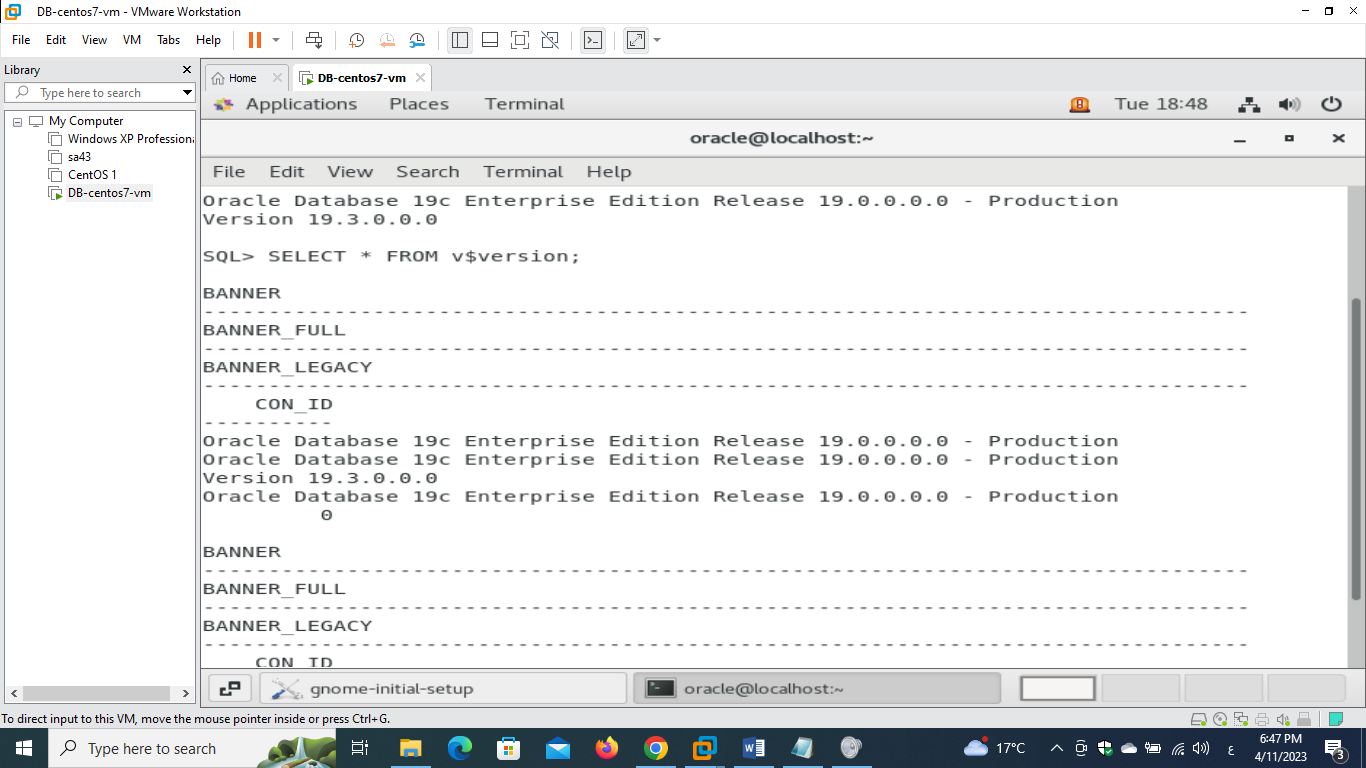
**- create spfile from the pfile.**



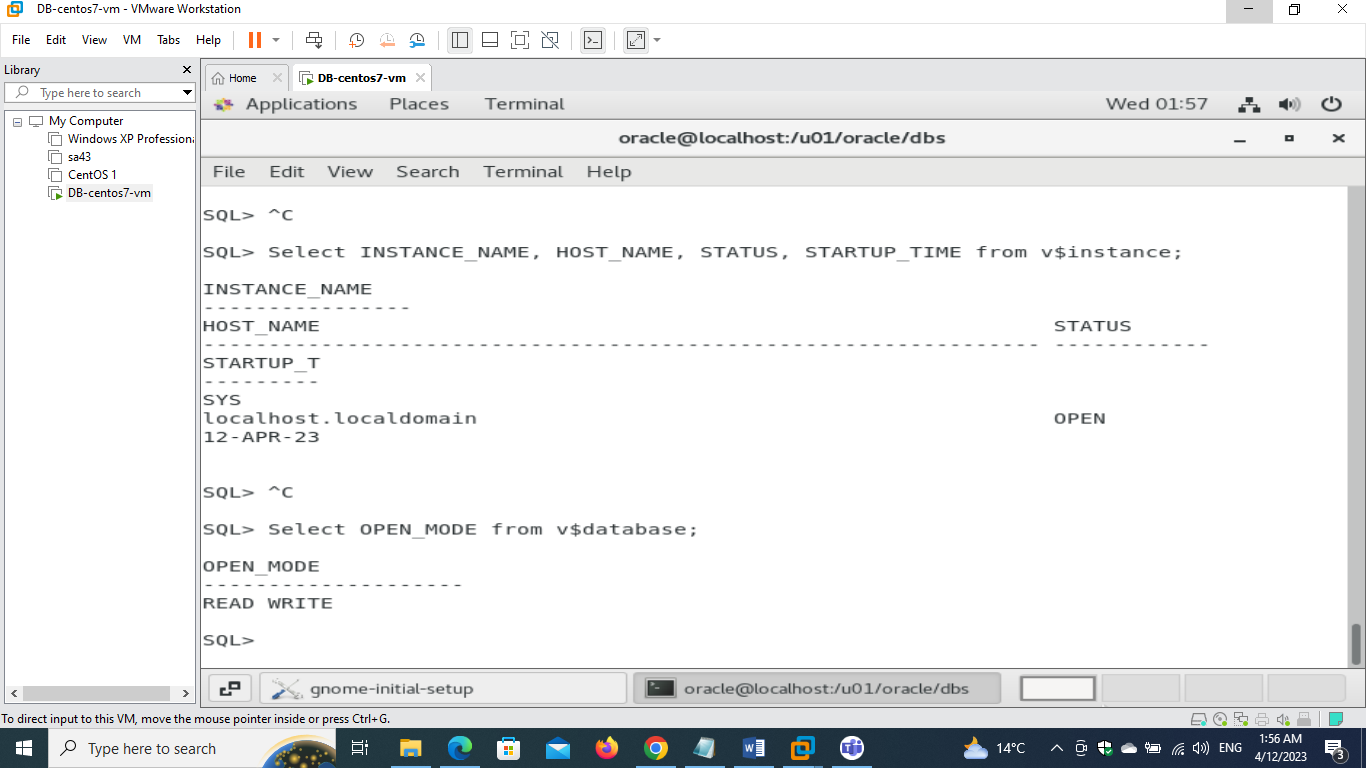
**4)**

**- show the version of your database using the view "v$version"**

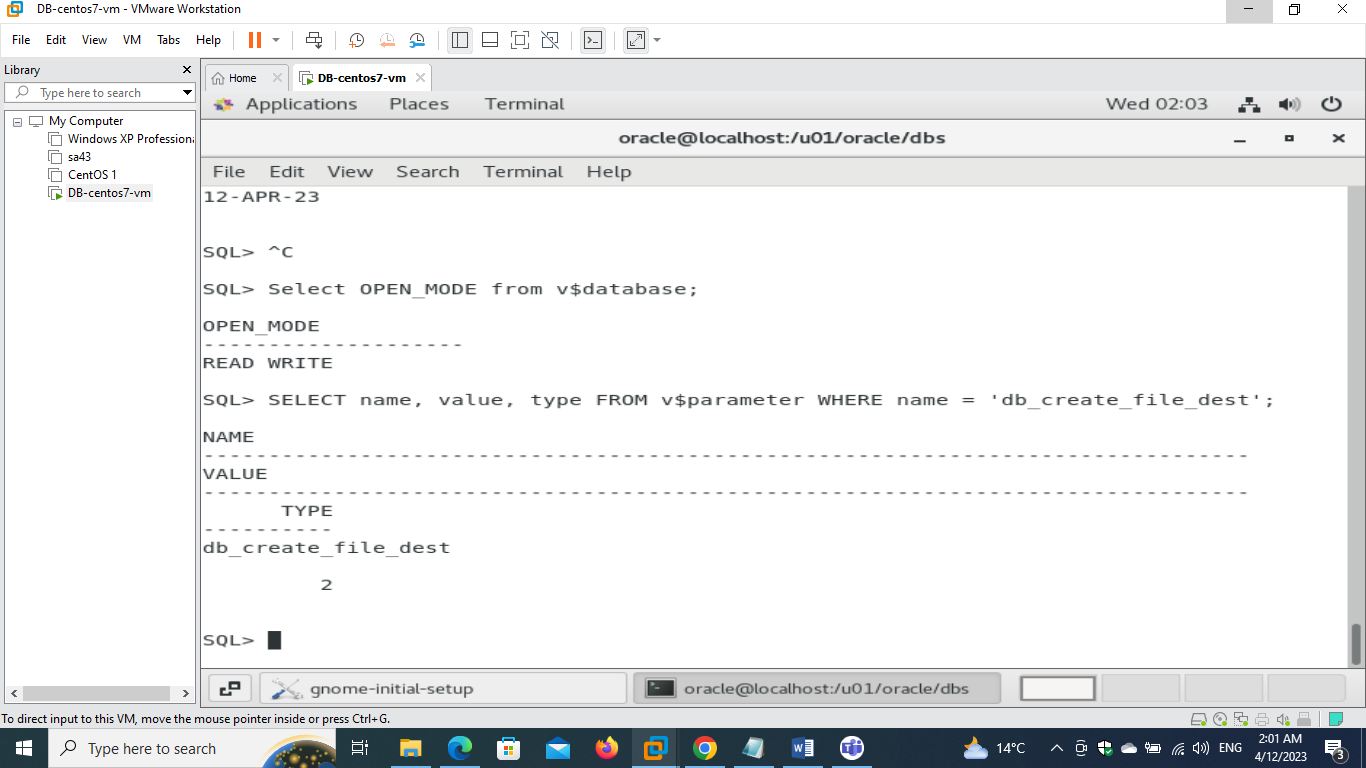
**SELECT \* FROM v$version;**



**- show hostname, instance name, status, open mode and startup\_time using the views "v$instance" and "v$database"**



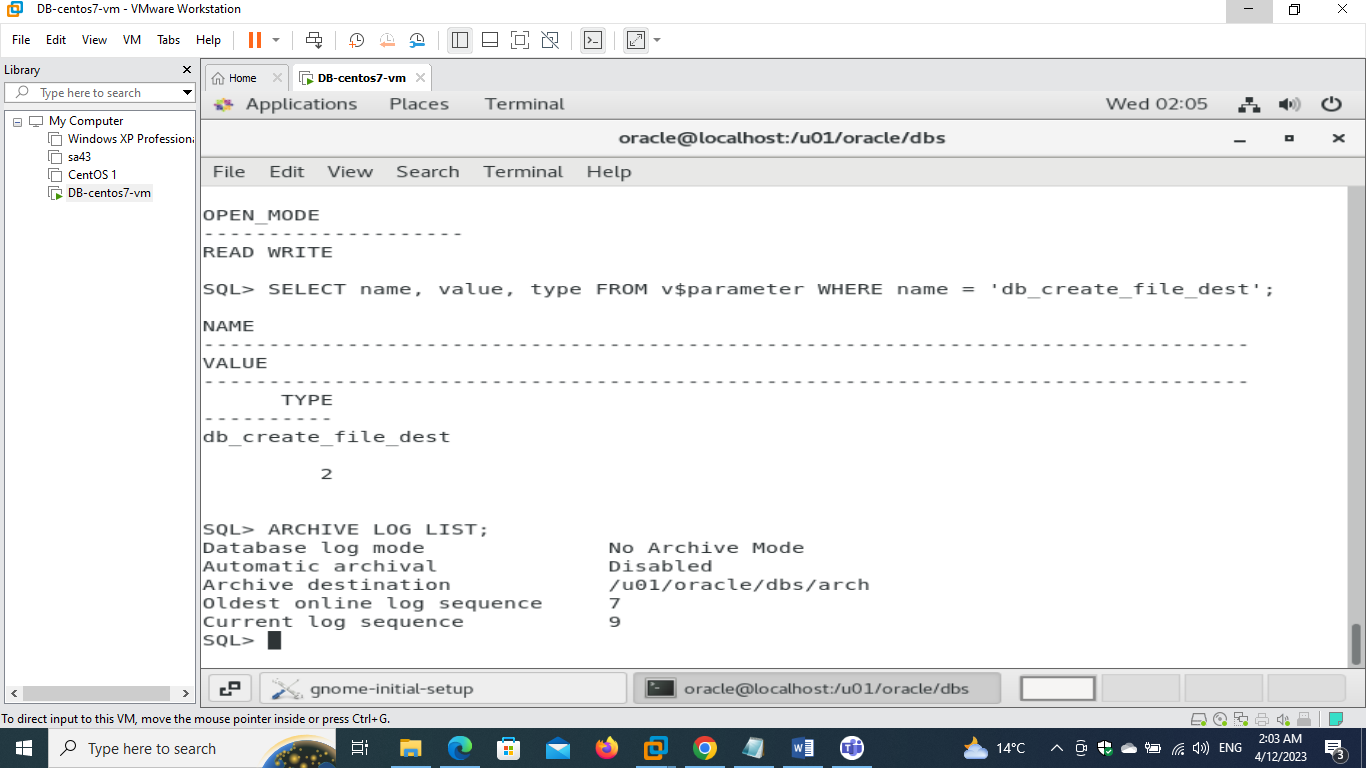
**- show the name and type of parameter "db\_create\_file\_dest" using view "v$parameter"**



**- show your archiving status then enable the archiving mode if not enabled and show your archiving status again.**

**ARCHIVE LOG LIST;**

**This command will display the current archiving status of the database, showing whether archiving is enabled or disabled, the name and location of the current archive log destination, and other related information**.



**ALTER DATABASE ARCHIVELOG;**

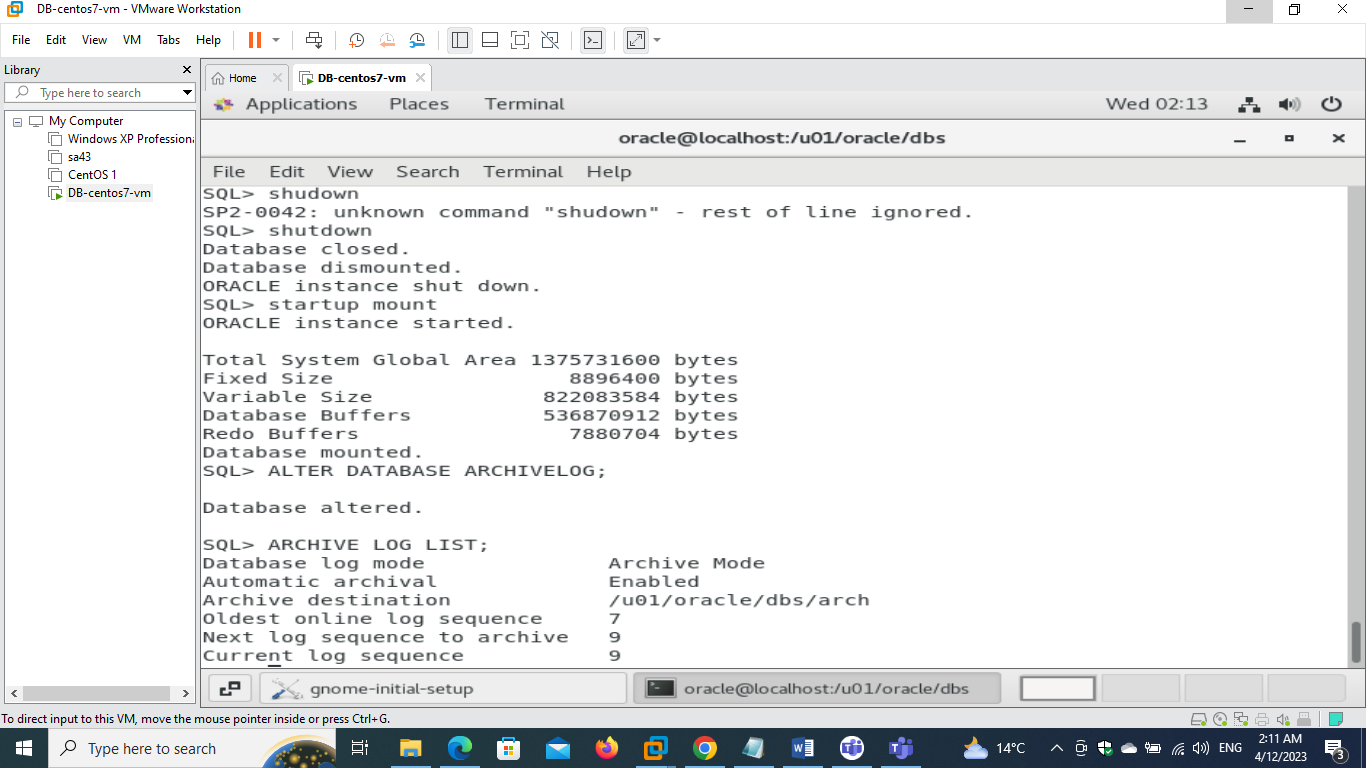
**This command will enable archiving mode for the database and start automatically archiving redo log files to the specified archive log destination**

**But first database must be mounted**

**So I used the following cmd**

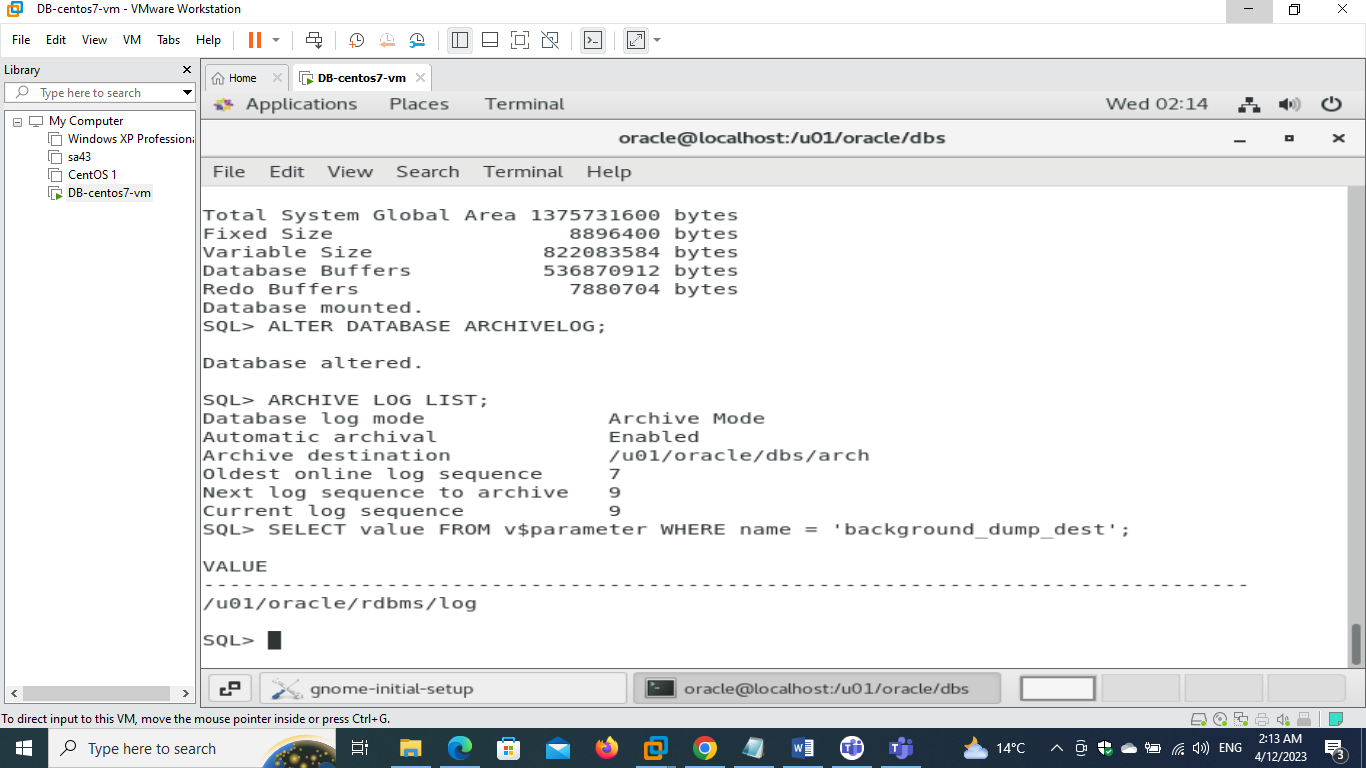
**shutdown**

**startup mount**



**- Identify the location of the alert log file.**

**SELECT value FROM v$parameter WHERE name = 'background\_dump\_dest';**

 **the path is /u01/oracle/rdbms/log**

**- shutdown your database with the most safe and fast method and watch your alert log while doing this (use "tail -f" command).**

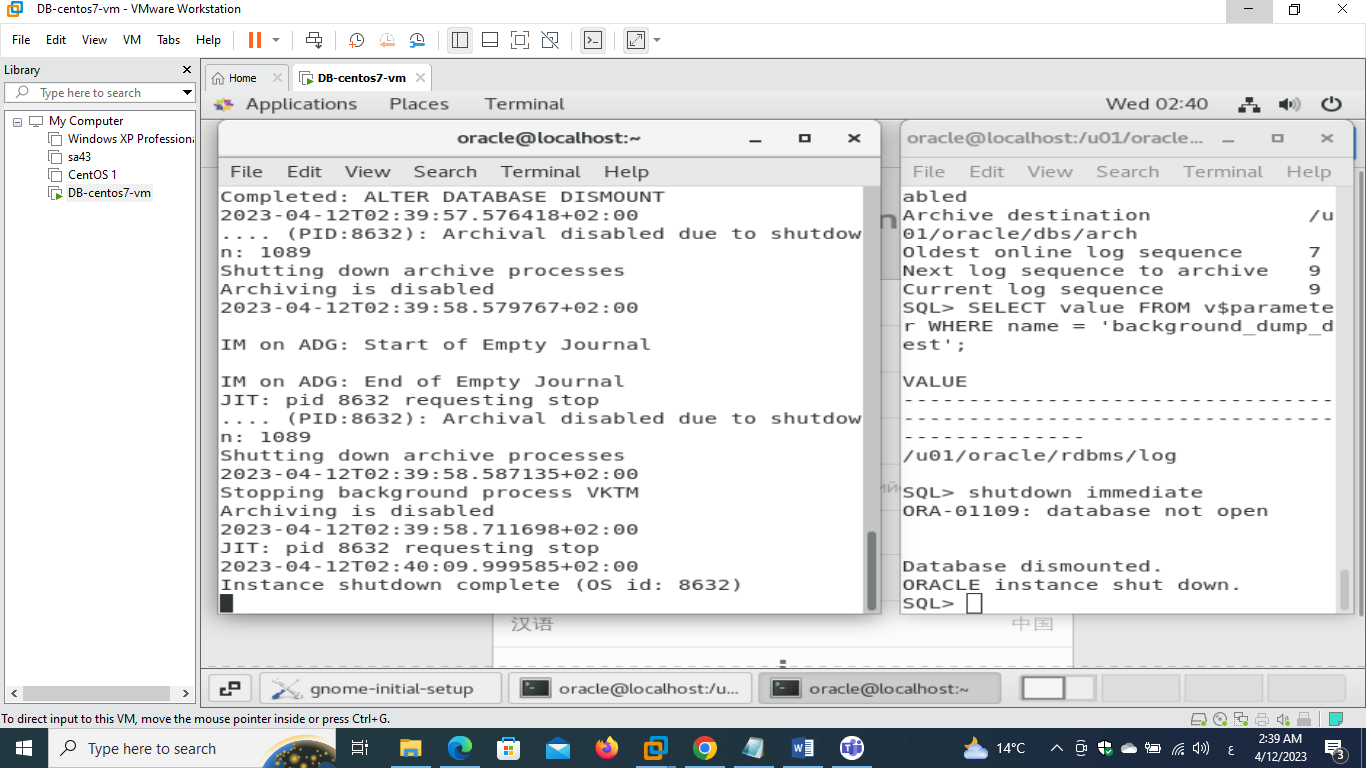
**Open two terminals**

**Run in one terminal**

**tail -f /u01/oracle\_base/diag/rdbms/sys/SYS/trace/alert\_SYS.log**

**and other terminal to run**

**shutdown immediate**

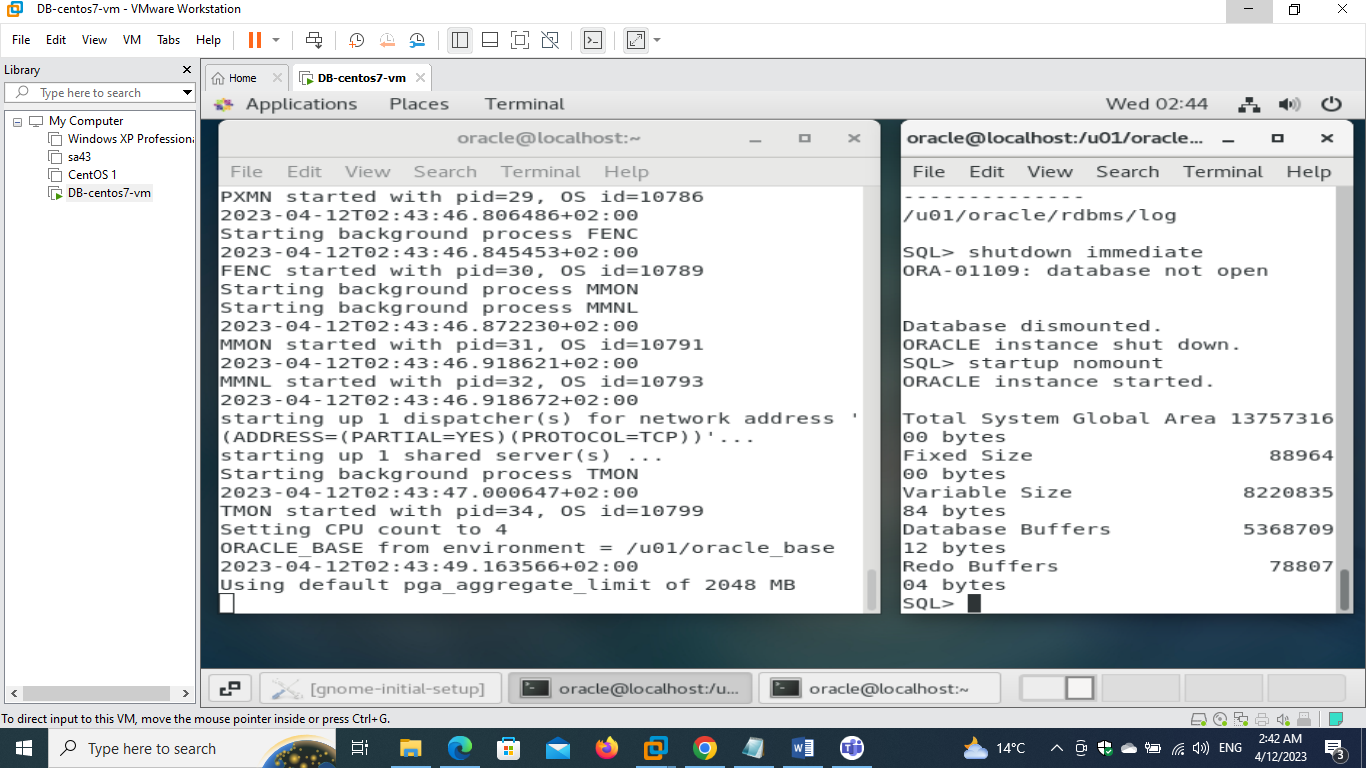


**- Start up your database step by step and watch your alert log while doing this (use "tail -f" command).**

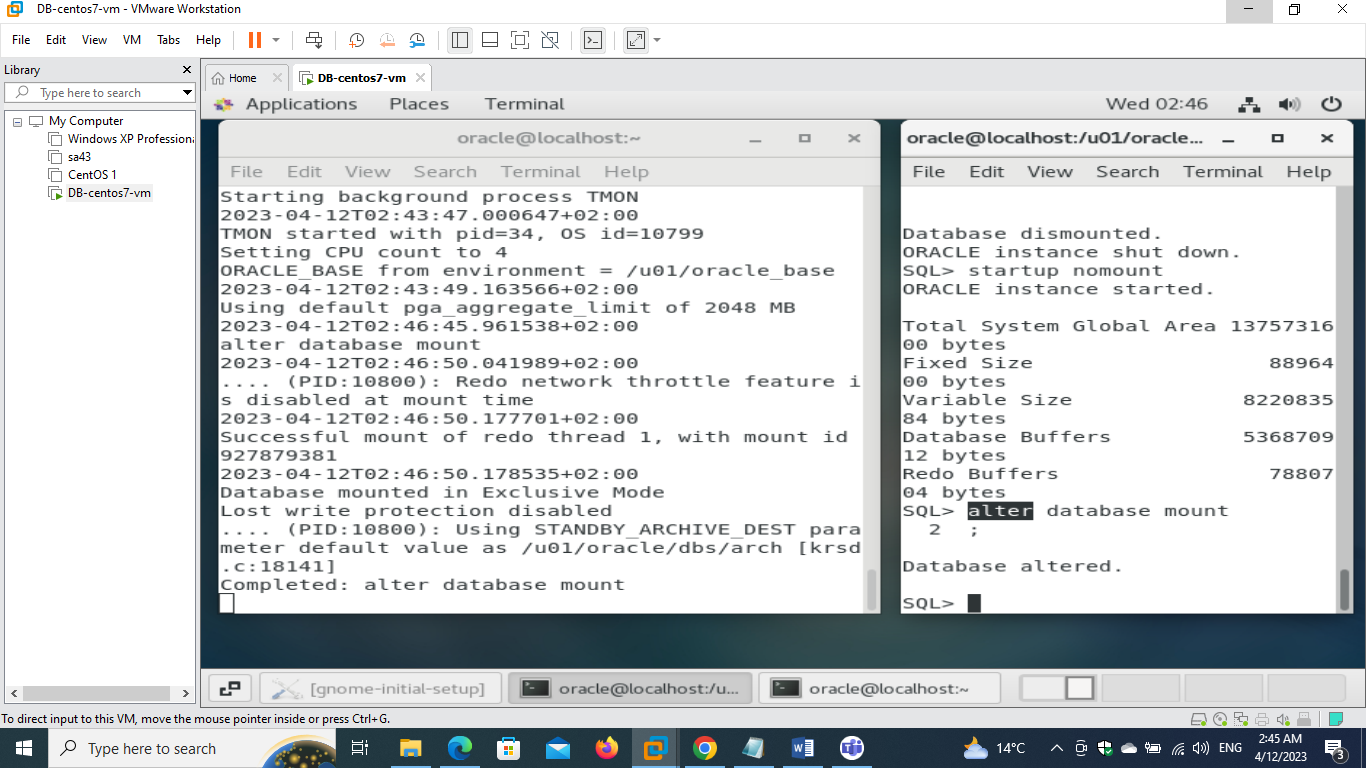
**Open two terminals**

**One for alert log and the other for exec cmd**

**Startup nomount**



**Alter database mount**



**Alter database open**

