Waterford Institute *of* Technology

Oracle Database Administration

(Semester 2)

Eoin Dalton [20070289]

Bachelor of Science (Hons) in Information Technology Management

Waterford Institute of Technology

Contents

Introduction

The objective of this report is to take you through the steps involved in the administering of Oracle11g Express Version. Oracle builds highly sophisticated databases and administrator needs to understand the underlying architecture so that if something goes wrong or some adjustments need to be made that they can perform those operations. The big personage of this report will focus on the file, memory and process because this is what make the database work. Helped with the aid of screenshot and using SQLPlus a command line tool that comes with Oracle it will take you through some of the commands that can be used to controls these different sections of the database.

The last three sections of the report will look at locking and latching and concurrency control these are the two mechanisms that enable Oracle database to be used by many users. The last section will be a short section on some of the security features that Oracle offers, and the commands used to implement some of those security features. In order the write this report their sources will be used for information. The first is the Oracle help center, the second is a book from Tomas Kyte called Expert Oracle Database Architecture (Kyte, 2005) and finally a report done by a past student by the name of Jonathan O' Brian (O'Brien, 2014).

1. Installation Specifications

Windows Specifications

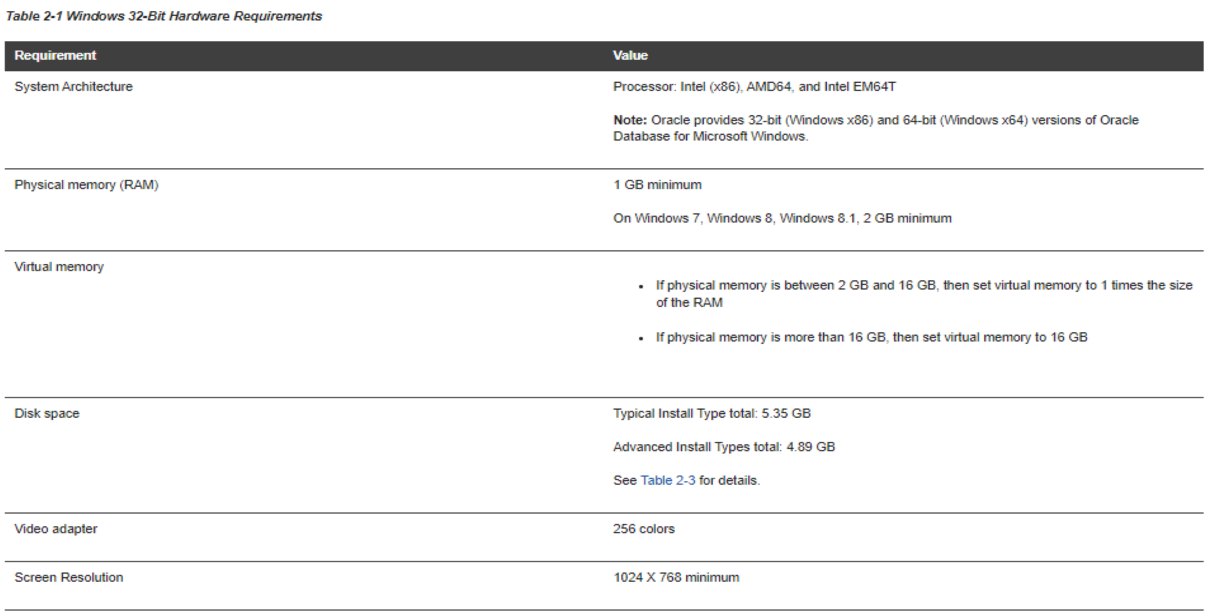


Table 2-1 Windows 32-Bit Hardware Requirements

Requirement	Value
System Architecture	Processor: Intel (x86), AMD64, and Intel EM64T **Note:** Oracle provides 32-bit (Windows x86) and 64-bit (Windows x64) versions of Oracle Database for Microsoft Windows.
Physical memory (RAM)	1 GB minimum On Windows 7, Windows 8, Windows 8.1, 2 GB minimum
Virtual memory	• If physical memory is between 2 GB and 16 GB, then set virtual memory to 1 times the size of the RAM • If physical memory is more than 16 GB, then set virtual memory to 16 GB
Disk space	Typical Install Type total: 5.35 GB Advanced Install Types total: 4.89 GB See Table 2-3 for details.
Video adapter	256 colors
Screen Resolution	1024 X 768 minimum

Figure 1: Specifications for Window 10 operating systems (Oracle, 2019).

Linux Specifications

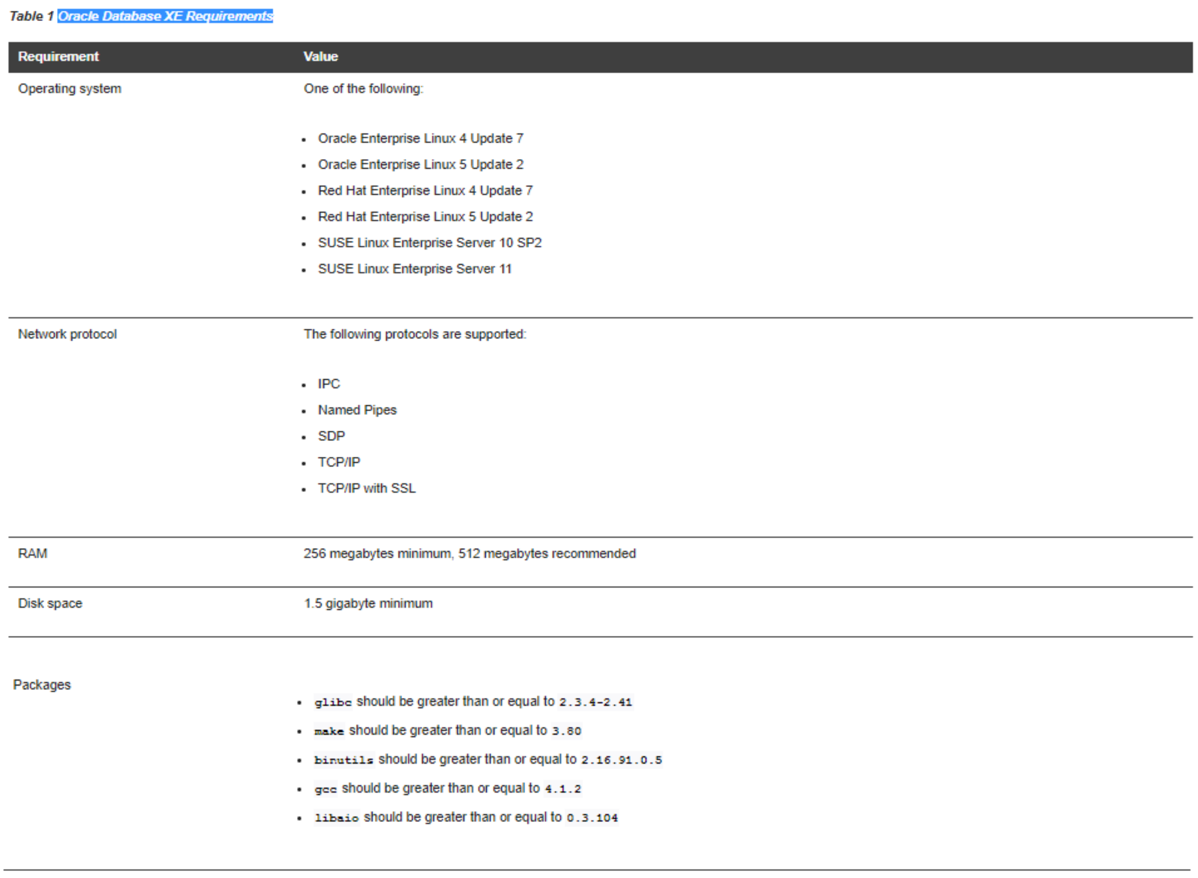


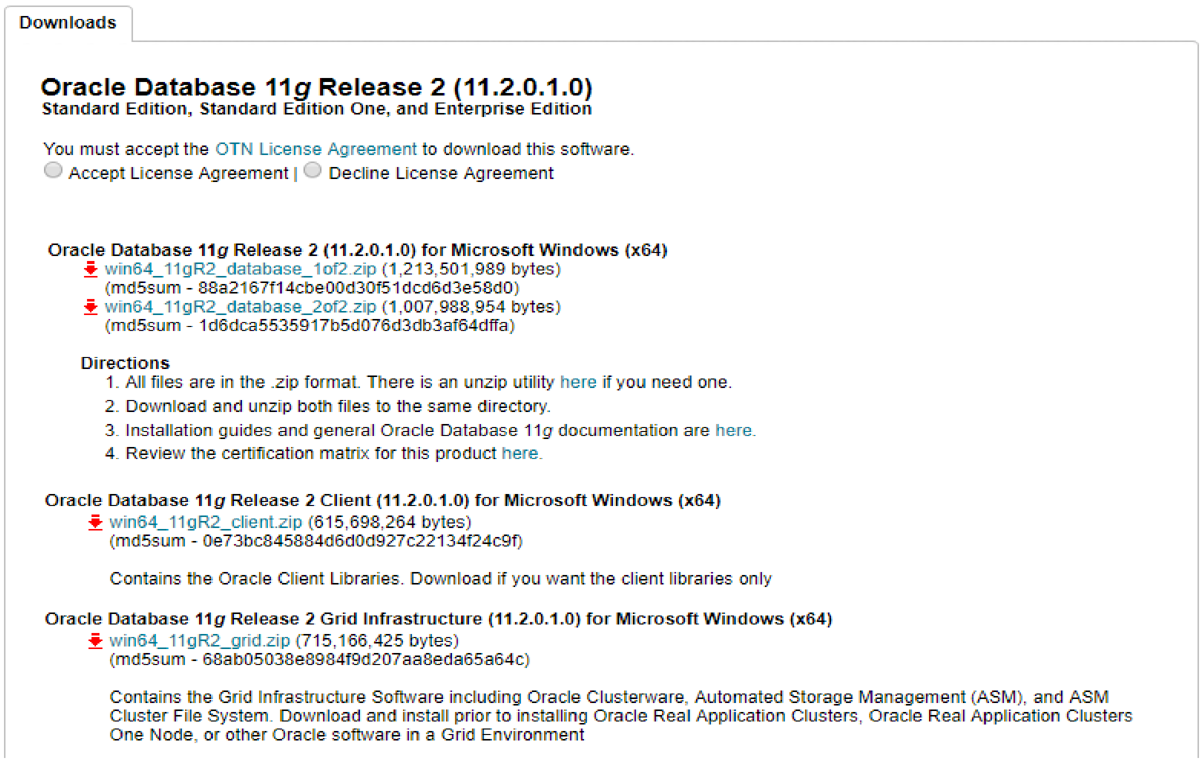
Table 1 Oracle Database XE Requirements

Requirement	Value
Operating system	One of the following: • Oracle Enterprise Linux 4 Update 7 • Oracle Enterprise Linux 5 Update 2 • Red Hat Enterprise Linux 4 Update 7 • Red Hat Enterprise Linux 5 Update 2 • SUSE Linux Enterprise Server 10 SP2 • SUSE Linux Enterprise Server 11
Network protocol	The following protocols are supported: • IPC • Named Pipes • SDP • TCP/IP • TCP/IP with SSL
RAM	256 megabytes minimum, 512 megabytes recommended
Disk space	1.5 gigabyte minimum
Packages	• `glibc` should be greater than or equal to `2.3.4-2.41` • `make` should be greater than or equal to `3.80` • `binutils` should be greater than or equal to `2.16.91.0.5` • `gcc` should be greater than or equal to `4.1.2` • `libaio` should be greater than or equal to `0.3.104`

Figure 2: Specifications for Linux operating system (Oracle, 2019) .

1.1 Windows installation

The installation process was aided by a video on YouTube found at the following link https://youtu.be/EvWNLgGHYfg . The class was asked to download Oracle 11g release 2 found on the Oracle downloads page. If you refer to figure 3 you will see many different options. As this machine is running Microsoft Windows with a 64bit processor was the correct option. The download at the top of the page was the most suited as we did not require extra libraries or clients. If you refer to figure 1 and 2 above it give the specification for installing Oracle in two operation system Window and Linux.



Downloads

Oracle Database 11*g* Release 2 (11.2.0.1.0)
Standard Edition, Standard Edition One, and Enterprise Edition

You must accept the OTN License Agreement to download this software.
Accept License Agreement | Decline License Agreement

Oracle Database 11*g* Release 2 (11.2.0.1.0) for Microsoft Windows (x64)
win64_11gR2_database_1of2.zip (1,213,501,989 bytes)
(md5sum - 88a2167f14cbe00d30f51dcd6d3e58d0)
win64_11gR2_database_2of2.zip (1,007,988,954 bytes)
(md5sum - 1d6dca5535917b5d076d3db3af64dffa)

Directions
1. All files are in the .zip format. There is an unzip utility here if you need one.
2. Download and unzip both files to the same directory.
3. Installation guides and general Oracle Database 11*g* documentation are here.
4. Review the certification matrix for this product here.

Oracle Database 11*g* Release 2 Client (11.2.0.1.0) for Microsoft Windows (x64)
win64_11gR2_client.zip (615,698,264 bytes)
(md5sum - 0e73bc845884d6d0d927c22134f24c9f)

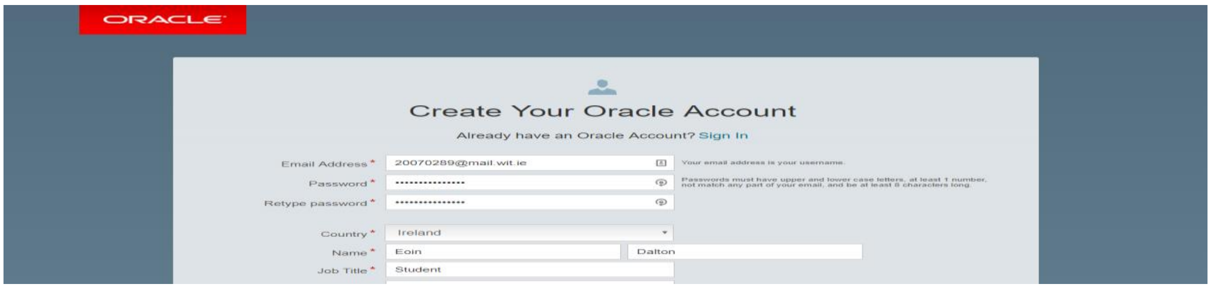
Contains the Oracle Client Libraries. Download if you want the client libraries only

Oracle Database 11*g* Release 2 Grid Infrastructure (11.2.0.1.0) for Microsoft Windows (x64)
win64_11gR2_grid.zip (715,166,425 bytes)
(md5sum - 68ab05038e8984f9d207aa8eda65a64c)

Contains the Grid Infrastructure Software including Oracle Clusterware, Automated Storage Management (ASM), and ASM Cluster File System. Download and install prior to installing Oracle Real Application Clusters, Oracle Real Application Clusters One Node, or other Oracle software in a Grid Environment

Figure 3: Oracle downloads page.

After agreeing to the licence terms and condition and selecting to download the file you get redirected to another Oracle page to sign up for a free account as seen in figure 4. After signing up for the account the download automatically started.



Create Your Oracle Account
Already have an Oracle Account? Sign In

Email Address * 20070289@mail.wit.ie — Your email address is your username.
Password * •••••••••••••••• — Passwords must have upper and lower case letters, at least 1 number, not match any part of your email, and be at least 8 characters long.
Retype password * ••••••••••••••••
Country * Ireland
Name * Eoin Dalton
Job Title * Student

Figure 4: Creating Oracle account.

There are two zipped files on the download page. If you refer to figure 5 you can see the two zipped files downloaded successfully.

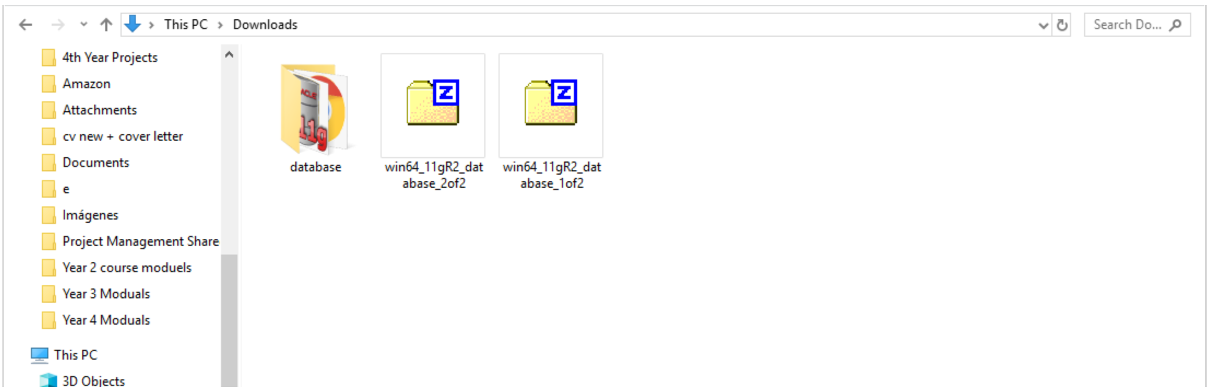
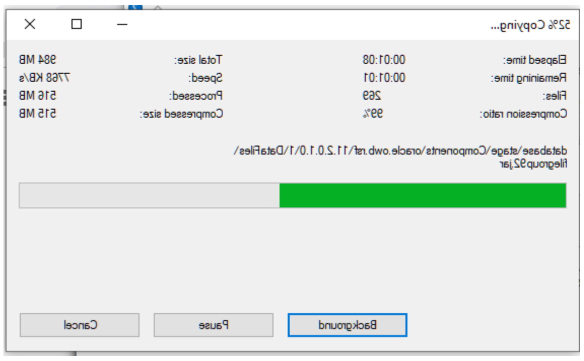
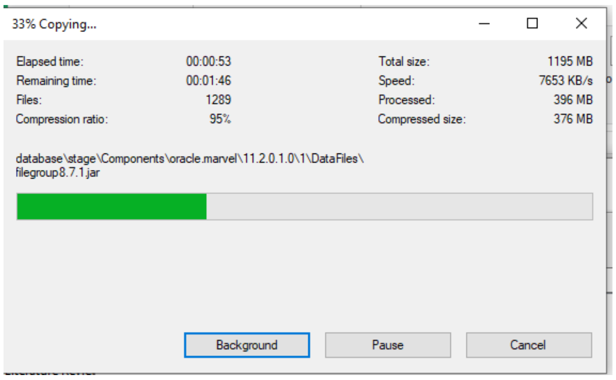


Figure 5: Zipped Oracle files.

After the two files have been extracted as seen in figure 6 and 7 you end up with a database file show in figure 8.



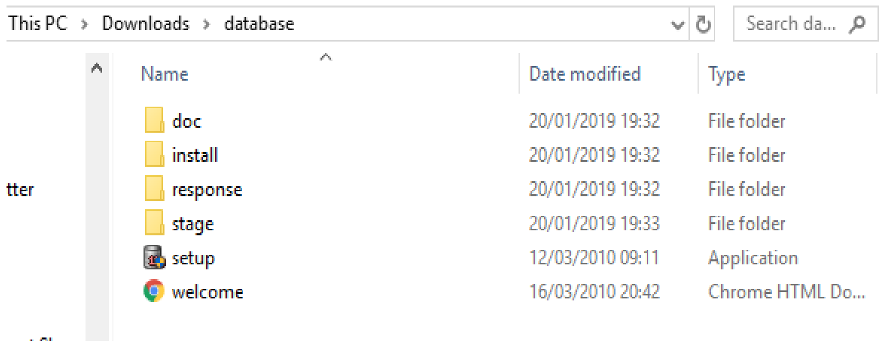
```
52% Copying...

Elapsed time:        00:01:08        Total size:        984 MB
Remaining time:      00:01:01        Speed:             7768 KB/s
Files:               269             Processed:         516 MB
Compression ratio:   99%             Compressed size:   515 MB

database/stage/Components/oracle.owb.rsf/11.2.0.1.0/1/DataFiles/
filegroup92.jar
```



```
33% Copying...

Elapsed time:        00:00:53        Total size:        1195 MB
Remaining time:      00:01:46        Speed:             7653 KB/s
Files:               1289            Processed:         396 MB
Compression ratio:   95%             Compressed size:   376 MB

database\stage\Components\oracle.marvel\11.2.0.1.0\1\DataFiles\
filegroup8.7.1.jar
```

Figure 7: Extracting folder 2

Figure 6: Extracting folder 1.

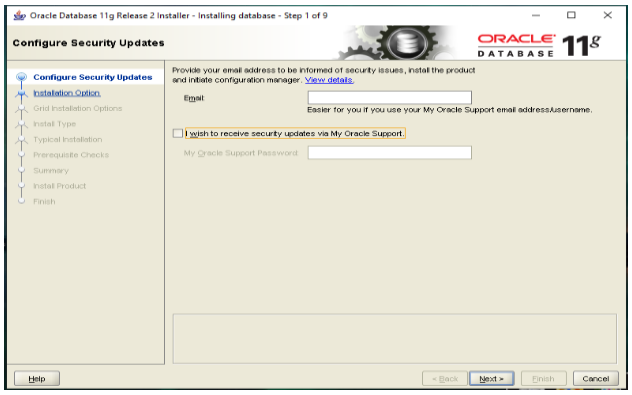


This PC › Downloads › database

Name	Date modified	Type
doc	20/01/2019 19:32	File folder
install	20/01/2019 19:32	File folder
response	20/01/2019 19:32	File folder
stage	20/01/2019 19:33	File folder
setup	12/03/2010 09:11	Application
welcome	16/03/2010 20:42	Chrome HTML Do...

Figure 8: Setup fie.

After clicking on the setup option users are presented with a command prompt for a few seconds while it runs some scripts, you are then presented with the install interface as show in figure 9. On the interface it offers options for email setup for information to be sent to you about updates and new letters, in the video the skip this option and move to the next step.


Oracle Database 11g Release 2 Installer - Installing database - Step 1 of 9

Configure Security Updates

- Configure Security Updates
- Installation Option
- Grid Installation Options
- Install Type
- Typical Installation
- Prerequisite Checks
- Summary
- Install Product
- Finish

Provide your email address to be informed of security issues, install the product and initiate configuration manager. View details.

Email:

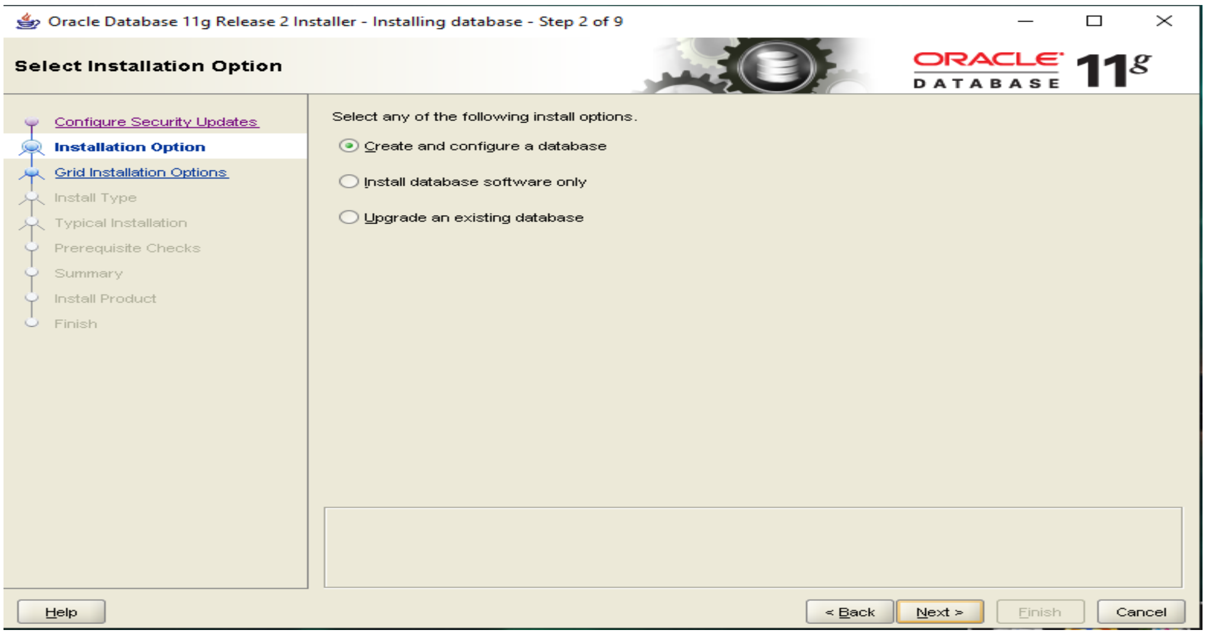
Easier for you if you use your My Oracle Support email address/username.

☐ I wish to receive security updates via My Oracle Support.

My Oracle Support Password:

Figure 9: Security update configuration

In the next step as show below is figure 10 is gives some install option like create and configure a database. Install the database software only or upgrade an existing database. The video suggested that because it's a clean install that the first option in most appropriate because it will install all components as well as updating the components as they install.


Oracle Database 11g Release 2 Installer - Installing database - Step 2 of 9

Select Installation Option

- Configure Security Updates
- Installation Option
- Grid Installation Options
- Install Type
- Typical Installation
- Prerequisite Checks
- Summary
- Install Product
- Finish

Select any of the following install options.

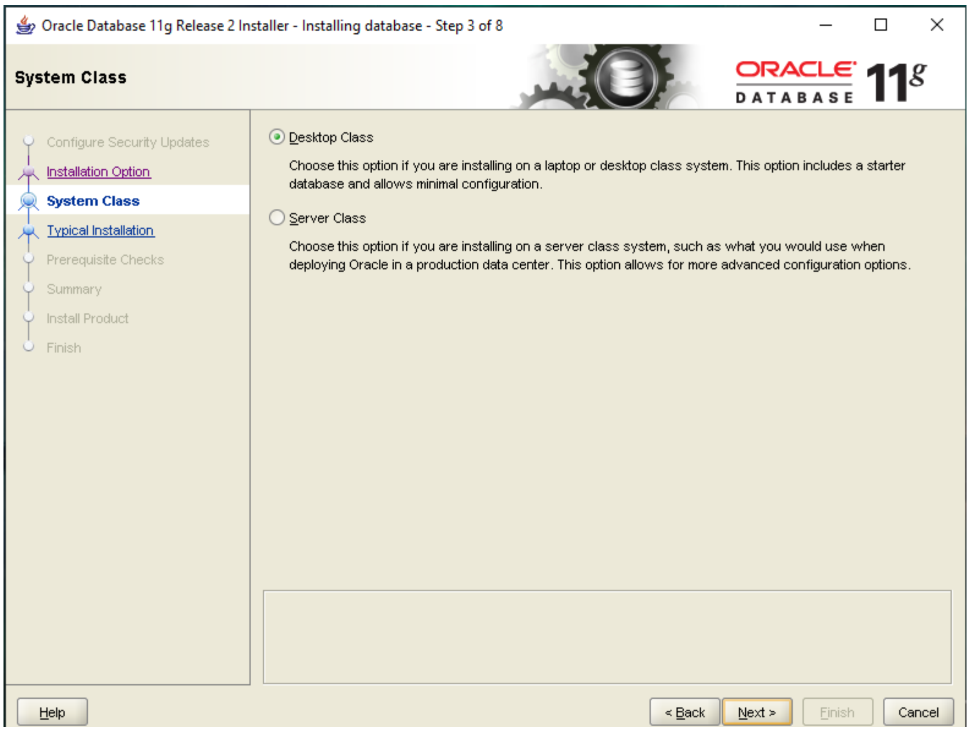
[✓ Create and configure a database]

Install database software only

Upgrade an existing database

Figure 10: Installation options.

The next configuration option was the install type. It gives two options. The first is a desktop class installation and the second option is a server class installation as show in figure 11. It was suggested in the video to run with the desktop install, this makes sense as there is no sever configured on this machine.


Oracle Database 11g Release 2 Installer - Installing database - Step 3 of 8

System Class

[✓ Desktop Class]

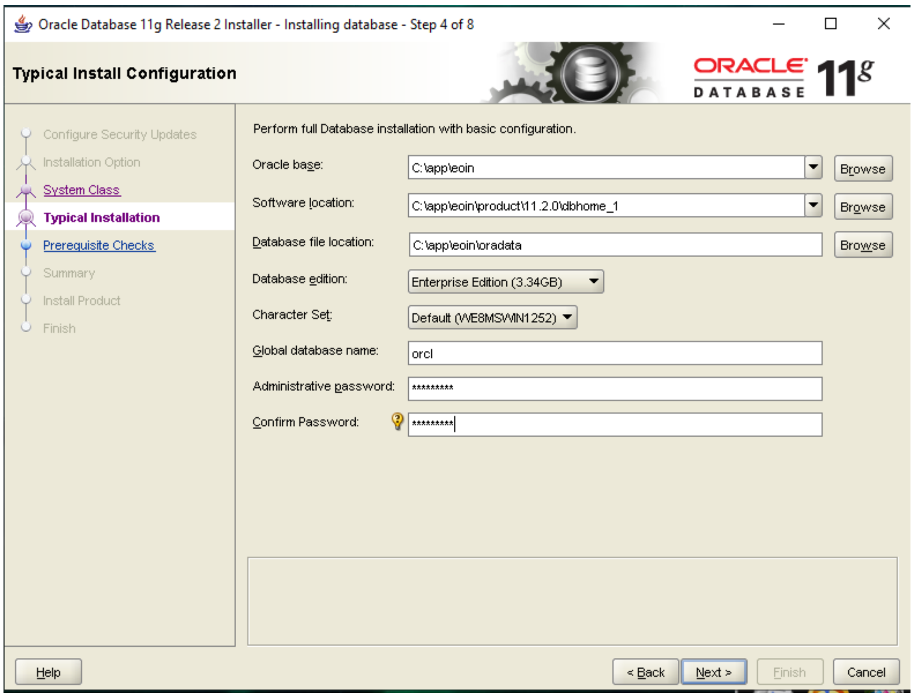
Choose this option if you are installing on a laptop or desktop class system. This option includes a starter database and allows minimal configuration.

Server Class

Choose this option if you are installing on a server class system, such as what you would use when deploying Oracle in a production data center. This option allows for more advanced configuration options.

Figure 11: Installation type.

The next stage of the install focused on configuration where the paths are set for the software location and database location. You specify the global database name so it can be identified by the Oracle software as seen in figure 12 (Oracle, 2019). The last step is to set an administrative password. This will be the password used by the system administrator to access the database. The administrator can all so grant access and permissions to other users as they need it.


Oracle Database 11g Release 2 Installer - Installing database - Step 4 of 8

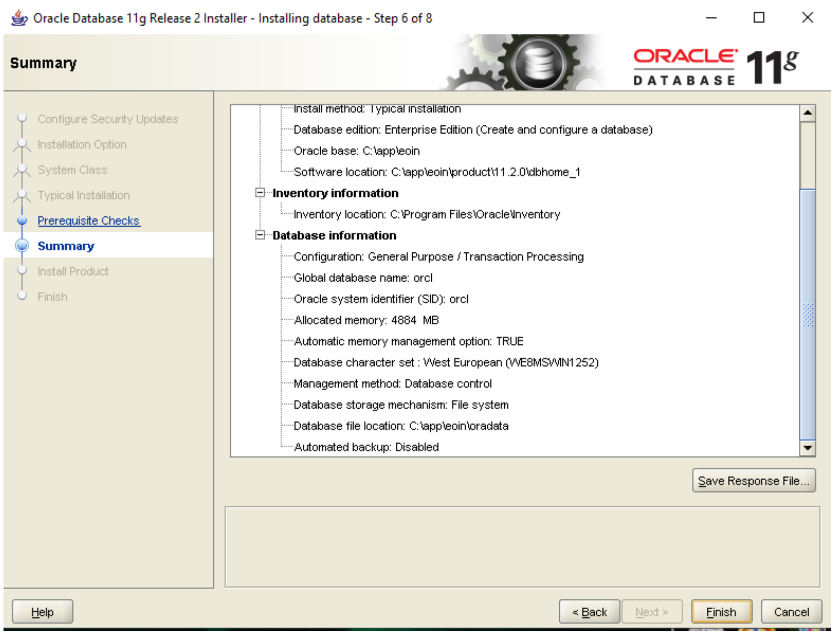
Typical Install Configuration

Perform full Database installation with basic configuration.

Oracle base:	C:\app\eoin
Software location:	C:\app\eoin\product\11.2.0\dbhome_1
Database file location:	C:\app\eoin\oradata
Database edition:	Enterprise Edition (3.34GB)
Character Set:	Default (WE8MSWIN1252)
Global database name:	orcl
Administrative password:	*********
Confirm Password:	*********

Figure 12: System configuration.

Show in figure 13 below is a summary of all the file and configurations that have taken place on the install so far. As you can see from the screenshot you have information for the database and inventory.

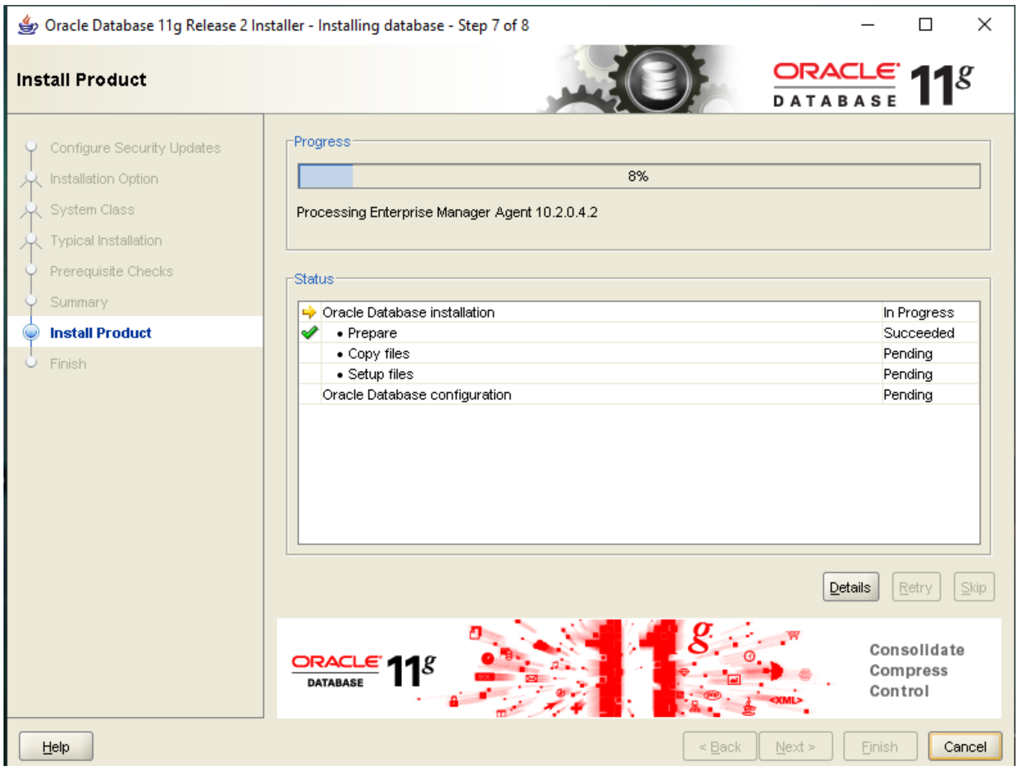

Oracle Database 11g Release 2 Installer - Installing database - Step 6 of 8

Summary

```
Install method: Typical installation
Database edition: Enterprise Edition (Create and configure a database)
Oracle base: C:\app\eoin
Software location: C:\app\eoin\product\11.2.0\dbhome_1
Inventory information
    Inventory location: C:\Program Files\Oracle\Inventory
Database information
    Configuration: General Purpose / Transaction Processing
    Global database name: orcl
    Oracle system identifier (SID): orcl
    Allocated memory: 4884  MB
    Automatic memory management option: TRUE
    Database character set : West European (WE8MSWIN1252)
    Management method: Database control
    Database storage mechanism: File system
    Database file location: C:\app\eoin\oradata
    Automated backup: Disabled
```

Figure 13: Summary of install.

If you refer to figure 14 you will see Oracle installing and the checklist that it needs to complete for the installation to complete. When it hits the Oracle database configuration it redirects to a different screen as show below in figure 15.


Oracle Database 11g Release 2 Installer - Installing database - Step 7 of 8

Install Product

Progress

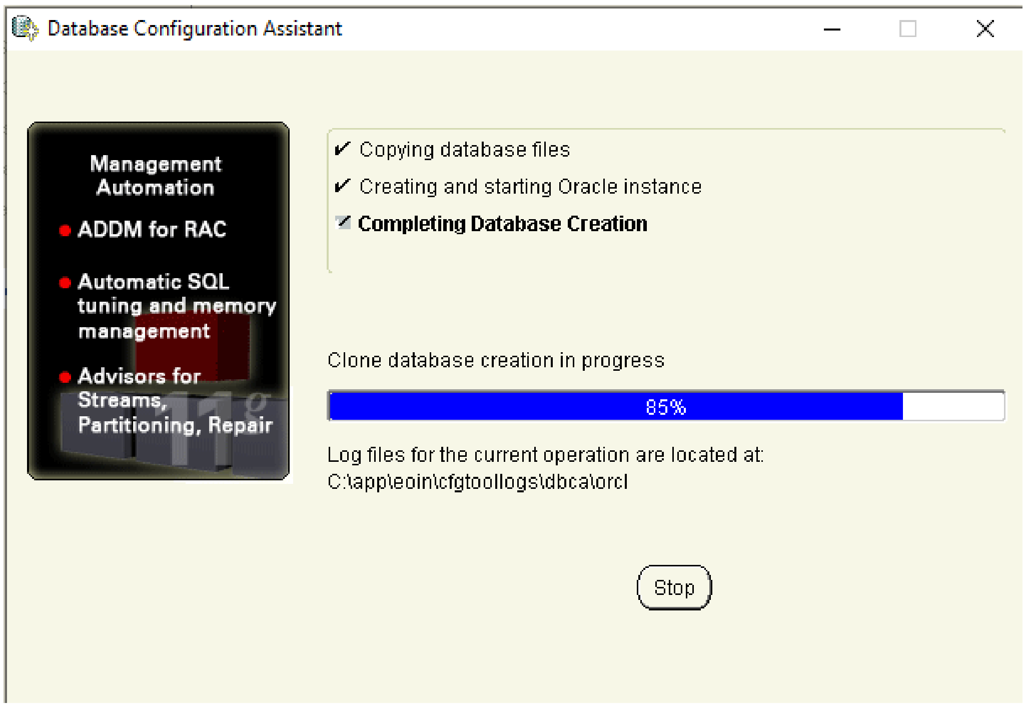
8%

Processing Enterprise Manager Agent 10.2.0.4.2

Status

Oracle Database installation	In Progress
• Prepare	Succeeded
• Copy files	Pending
• Setup files	Pending
Oracle Database configuration	Pending

Figure 14: Installing Oracle.


Database Configuration Assistant
Management Automation
ADDM for RAC
Automatic SQL tuning and memory management
Advisors for Streams, Partitioning, Repair
11g
Stop

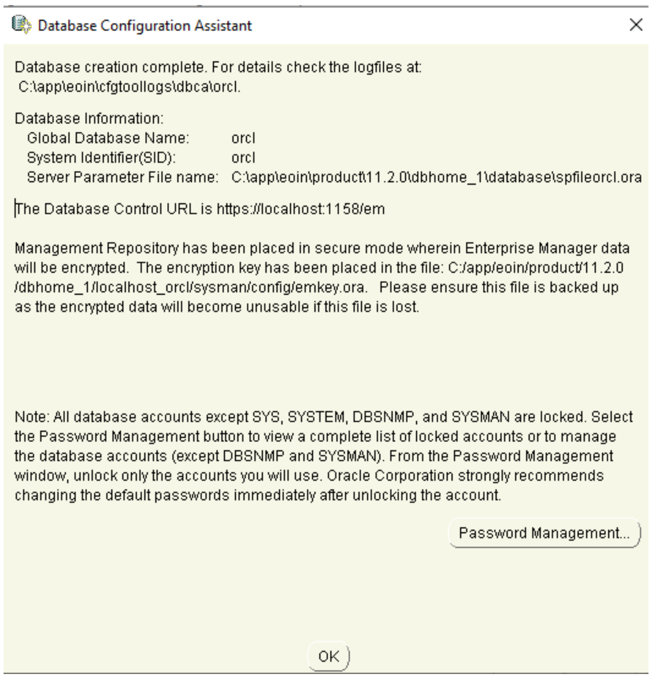
✔ Copying database files
✔ Creating and starting Oracle instance
Completing Database Creation

Clone database creation in progress

85%

Log files for the current operation are located at:
C:\app\eoin\cfgtoollogs\dbca\orcl

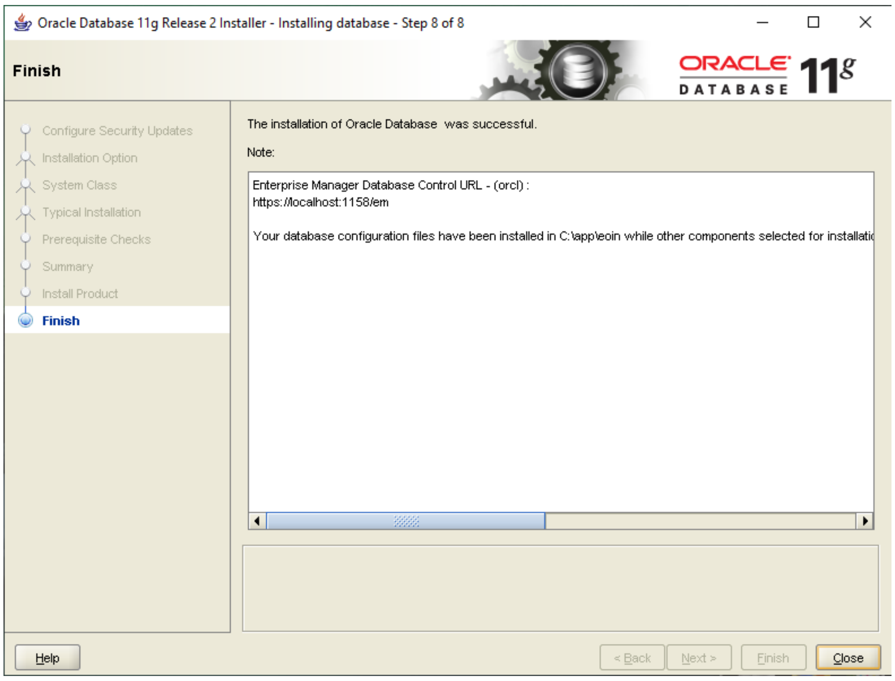
Figure 15: Configuration assistant.



```
Database creation complete. For details check the logfiles at:
 C:\app\eoin\cfgtoollogs\dbca\orcl.

Database Information:
   Global Database Name:       orcl
   System Identifier(SID):     orcl
   Server Parameter File name: C:\app\eoin\product\11.2.0\dbhome_1\database\spfileorcl.ora

The Database Control URL is https://localhost:1158/em

Management Repository has been placed in secure mode wherein Enterprise Manager data
will be encrypted.  The encryption key has been placed in the file: C:/app/eoin/product/11.2.0
/dbhome_1/localhost_orcl/sysman/config/emkey.ora.   Please ensure this file is backed up
as the encrypted data will become unusable if this file is lost.

Note: All database accounts except SYS, SYSTEM, DBSNMP, and SYSMAN are locked. Select
the Password Management button to view a complete list of locked accounts or to manage
the database accounts (except DBSNMP and SYSMAN). From the Password Management
window, unlock only the accounts you will use. Oracle Corporation strongly recommends
changing the default passwords immediately after unlocking the account.
```

Figure 16: Config assistant password management.

After the Oracle database configuration is completed you are presented with the screen show above in figure 16. This is similar to the summary page. It gives users details about the configurations they have made during the installation process for their database. It then allows users to manage password for other users they might want to add during the install process. If the database needs more than one administrator, they can be added by clicking on password management.



The installation of Oracle Database was successful.

Note:

Enterprise Manager Database Control URL - (orcl) :
https://localhost:1158/em

Your database configuration files have been installed in C:\app\eoin while other components selected for installati

Figure 17: Oracle Installed.

Shown in figure 17 is the last step confirming the install process is complete. After this section I will test the install to see if everything is working ok.

1.1.1 Connection Test Windows 10

Once install is complete, the next step is to test to see if the sever is running on the local host. To do this the users can navigate to where Oracle is installed, and they will see an option for Database control show in figure 18.

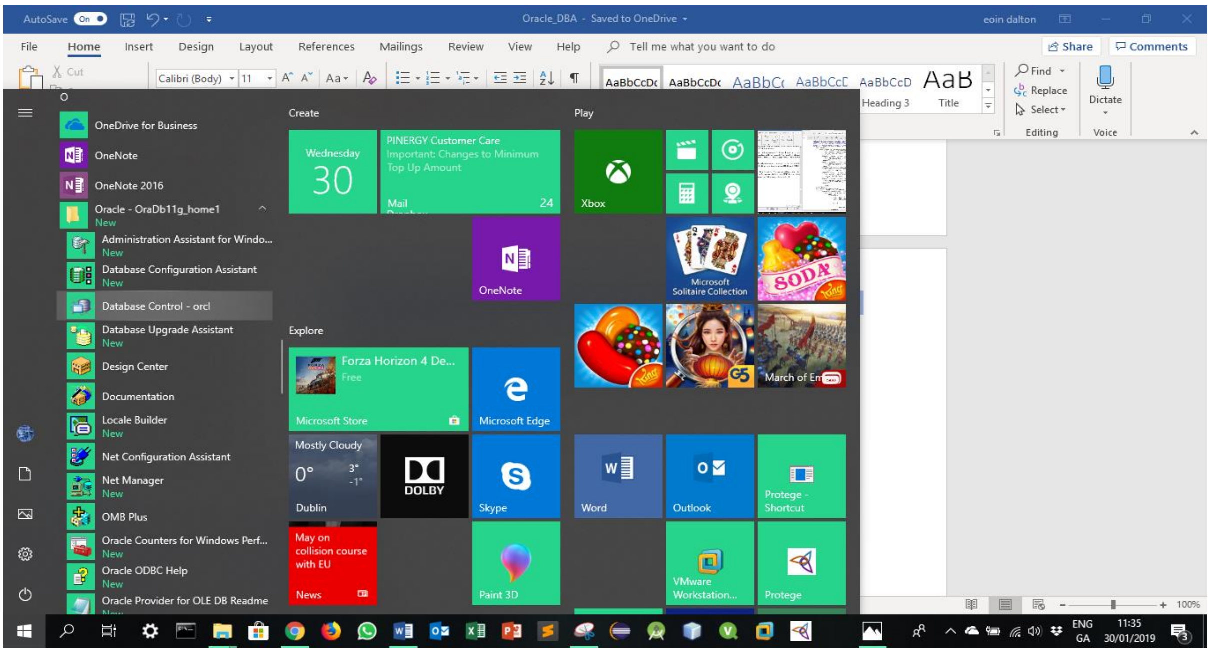
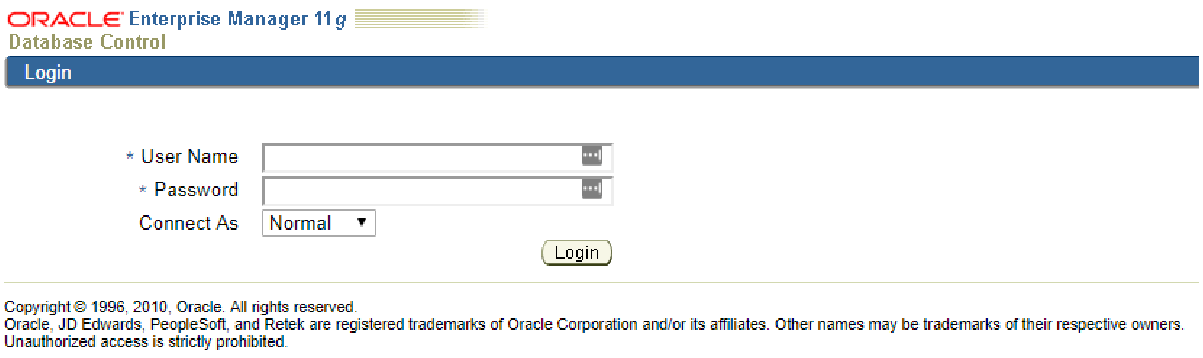


Figure 18: Database controller.

Once database control is selected users will be redirected to the Oracle enterprise manager as show in figure 19. User will use the username and password that they configured during the installation process. What they are logging into is the database instance that was setup on the install. After entering SYSTEM and my password users are presented with the screen shown below in figure 20.


ORACLE Enterprise Manager 11*g*
Database Control

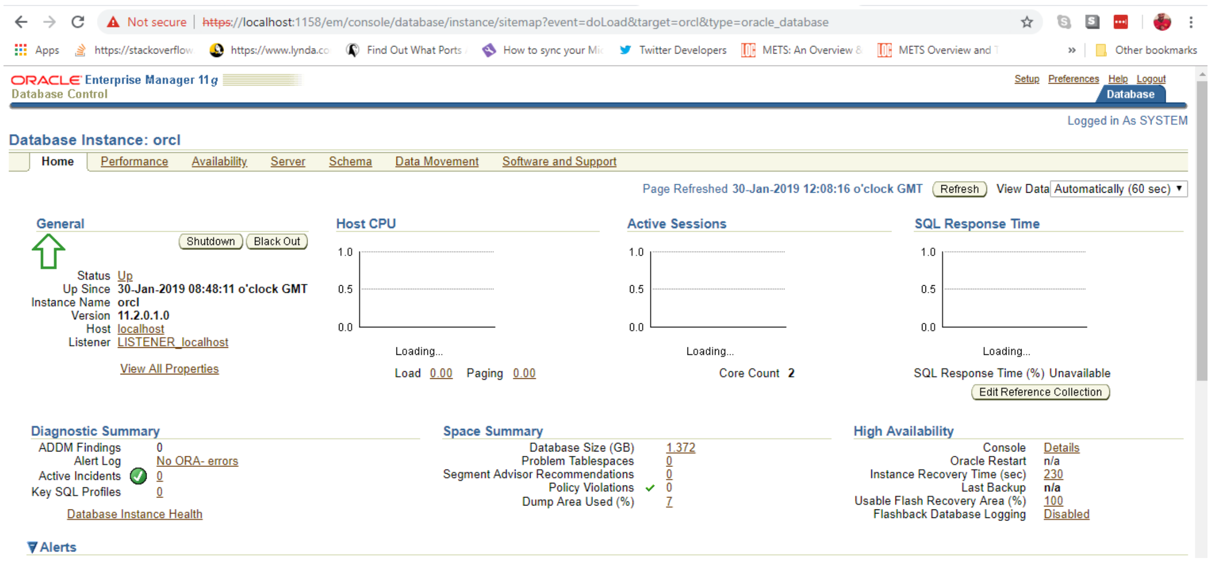
Login

* User Name
* Password
Connect As Normal

Login

Copyright © 1996, 2010, Oracle. All rights reserved.
Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.
Unauthorized access is strictly prohibited.

Figure 19: Oracle enterprise manager.



https://localhost:1158/em/console/database/instance/sitemap?event=doLoad&target=orcl&type=oracle_database

ORACLE Enterprise Manager 11g
Database Control

Logged in As SYSTEM

Database Instance: orcl

Home | Performance | Availability | Server | Schema | Data Movement | Software and Support

Page Refreshed 30-Jan-2019 12:08:16 o'clock GMT | View Data Automatically (60 sec)

General

Status	Up
Up Since	30-Jan-2019 08:48:11 o'clock GMT
Instance Name	orcl
Version	11.2.0.1.0
Host	localhost
Listener	LISTENER_localhost

Host CPU: Load 0.00 Paging 0.00

Active Sessions: Core Count 2

SQL Response Time: SQL Response Time (%) Unavailable

Diagnostic Summary

ADDM Findings	0
Alert Log	No ORA- errors
Active Incidents	0
Key SQL Profiles	0

Space Summary

Database Size (GB)	1.372
Problem Tablespaces	0
Segment Advisor Recommendations	0
Policy Violations	0
Dump Area Used (%)	7

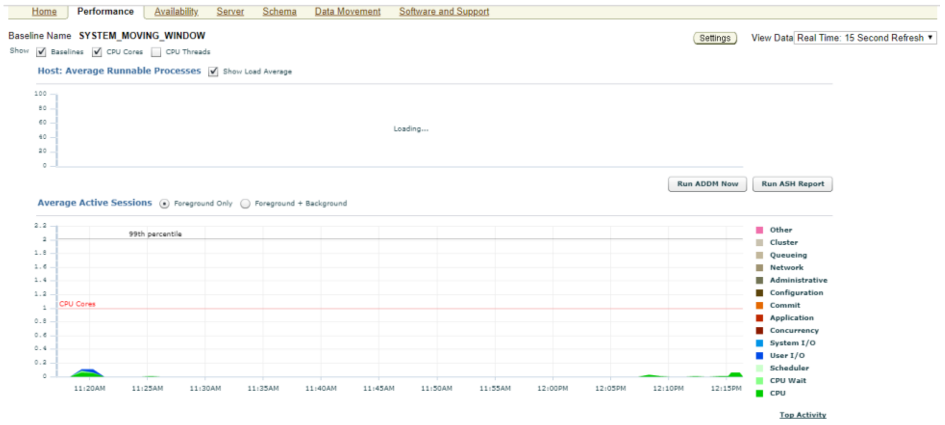
High Availability

Console	Details
Oracle Restart	n/a
Instance Recovery Time (sec)	230
Last Backup	n/a
Usable Flash Recovery Area (%)	100
Flashback Database Logging	Disabled

Alerts

Figure 20: Oracle localhost mangement.

If you refer to figure 20 in the search bar you will see that localhost is listening on port 1158. The management system allows users to monitor performance, configure availability, view schemas and the sever. If you refer to figure 21 you will see the performance screen. This gives administrators visuals of items like concurrency control, network, commits and much more. If a database was running user would see spikes for all the items below.

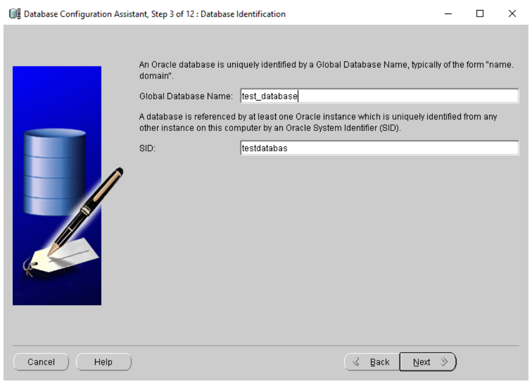


Baseline Name SYSTEM_MOVING_WINDOW

View Data Real Time: 15 Second Refresh

Figure 21: Performance.

Another test that was performed was to create a database and an admin user. To do this I navigated to the database configuration app. This was a 12-step process. If you refer to figure 22 you will see the naming of the database. The steps are straight forward to follow. I want to demonstrate how you can have many different admin users if you refer to figure 23 below step five I had four different user and I used the same password for all user just for convenience as this is only for practice.



Database Configuration Assistant, Step 3 of 12 : Database Identification

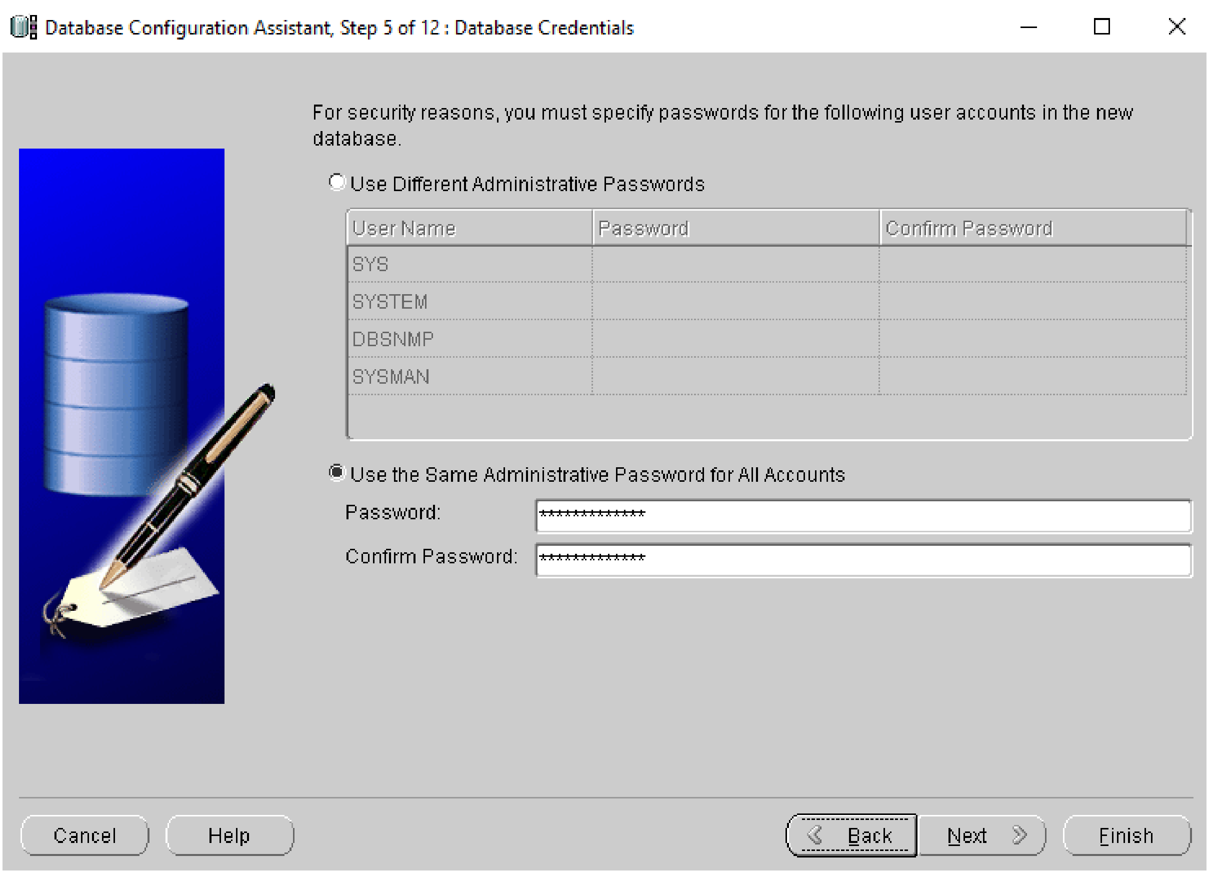
An Oracle database is uniquely identified by a Global Database Name, typically of the form "name. domain".

Global Database Name: test_database

A database is referenced by at least one Oracle instance which is uniquely identified from any other instance on this computer by an Oracle System Identifier (SID).

SID: testdatabas

Figure 22: Database name.



Database Configuration Assistant, Step 5 of 12 : Database Credentials

For security reasons, you must specify passwords for the following user accounts in the new database.

○ Use Different Administrative Passwords

User Name	Password	Confirm Password
SYS		
SYSTEM		
DBSNMP		
SYSMAN		

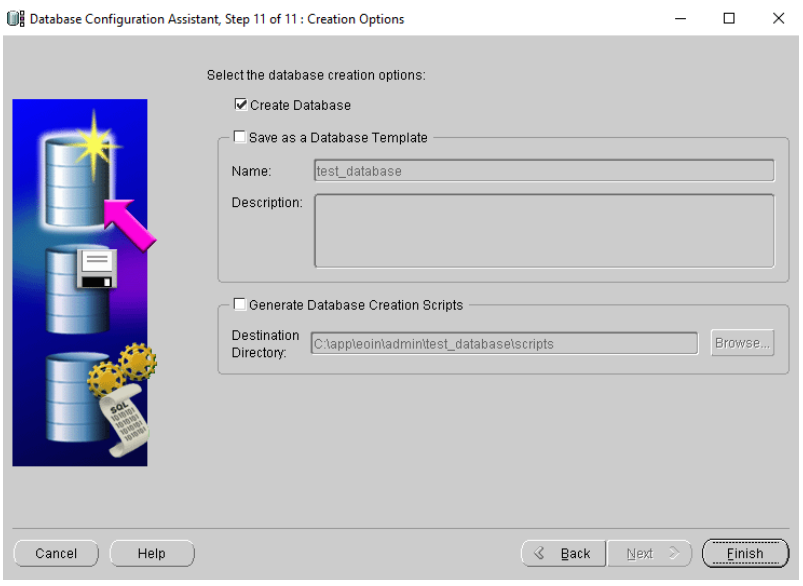
● Use the Same Administrative Password for All Accounts

Password: **************

Confirm Password: **************

Figure 23: Admin users and passwords.

Shown in figure 24 is the summary table once the database was installed. After the database was installed, I used the command line SQL plus to connect shown below in figure 25. I used user DBSNMP to connect to the database. This user can now issue SQL commands as they wish.



Database Configuration Assistant, Step 11 of 11 : Creation Options

Select the database creation options:

☑ Create Database

☐ Save as a Database Template

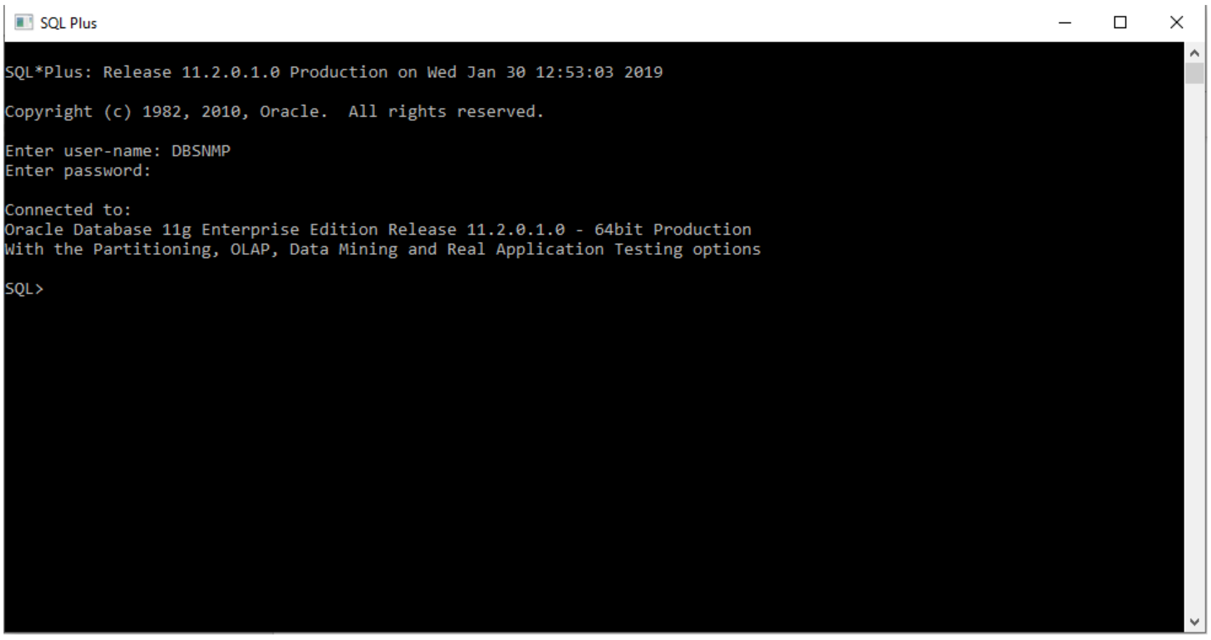
Name: test_database

Description:

☐ Generate Database Creation Scripts

Destination Directory: C:\app\eoin\admin\test_database\scripts

Figure 24: Database installed.



```
SQL*Plus: Release 11.2.0.1.0 Production on Wed Jan 30 12:53:03 2019

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

Enter user-name: DBSNMP
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL>
```

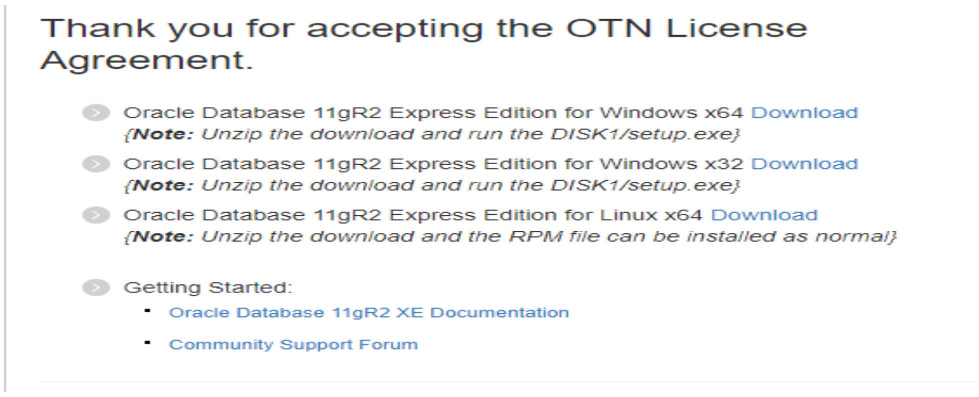
Figure 25: Connect to test_database using SQL Plus.

1.2. Ubuntu 18.04 installation

After installing Oracle 11g on my Window machine I noticed a decrease in performance during boot process, and it was extremely slow because of all the Oracle services that needed to start. I have Ubuntu 18.04 running on VMware and decided to delete the version from my Windows 10 machine and install Oracle on my virtual machine. This time I decided to install Oracle11g express version which is a more compact version of Oracle. The express version is a lightweight version compared with the standard version available. This is better suited as my machine has only an i3 processor.

In section 1 installation specifications you will notice that Oracle is built for three different version of Linux. Oracle, Red Hat and SUSE. Ubuntu does not feature on this list. After some searching, I found a tutorial on how to install Oracle 11g on a Debian version of Linux (Redressa, 2012). To install Oracle on Ubuntu it requires creating the required directories manually and creating some install scripts. On top of this the zip file that is downloaded is a .rpm file because Ubuntu is a .deb extension it requires converting the file to a .deb so the install can be executed. Luckily for me there is a software called Alien that is used to convert the .rpm file to .deb. So, let’s get started with the install.

The first step was to download the rpm file from Oracle’s website as show in figure 26.



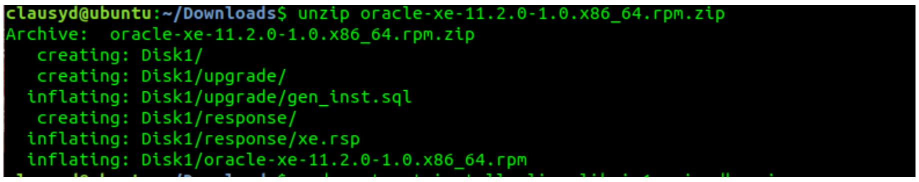
Thank you for accepting the OTN License Agreement.

- Oracle Database 11gR2 Express Edition for Windows x64 Download
  {**Note:** *Unzip the download and run the DISK1/setup.exe*}
- Oracle Database 11gR2 Express Edition for Windows x32 Download
  {**Note:** *Unzip the download and run the DISK1/setup.exe*}
- Oracle Database 11gR2 Express Edition for Linux x64 Download
  {**Note:** *Unzip the download and the RPM file can be installed as normal*}

- Getting Started:
  - Oracle Database 11gR2 XE Documentation
  - Community Support Forum

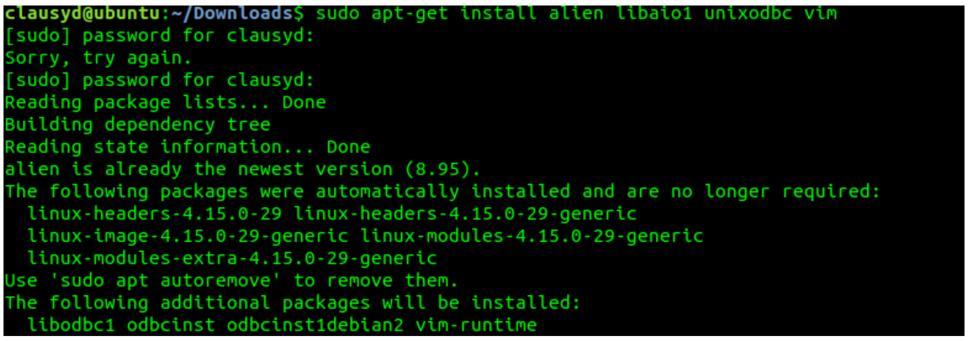
Figure 26: Oracle 11g R2 Express Edition for Linux.

Once the zip file is downloaded, I issued the unzip command as seen in figure 27. Once the file is decompressed, users will see a directory called Disk1 as shown below is figure 29. If you refer to figure 28 you will see me install the required libraries needed for installing Oracle 11g. The first tool is alien this is used to convert the .rpm file to a .deb file as I stated above.



```
clausyd@ubuntu:~/Downloads$ unzip oracle-xe-11.2.0-1.0.x86_64.rpm.zip
Archive:  oracle-xe-11.2.0-1.0.x86_64.rpm.zip
   creating: Disk1/
   creating: Disk1/upgrade/
  inflating: Disk1/upgrade/gen_inst.sql
   creating: Disk1/response/
  inflating: Disk1/response/xe.rsp
  inflating: Disk1/oracle-xe-11.2.0-1.0.x86_64.rpm
```

Figure 27: Unzipping folder containing Oracle 11g.



```
clausyd@ubuntu:~/Downloads$ sudo apt-get install alien libaio1 unixodbc vim
[sudo] password for clausyd:
Sorry, try again.
[sudo] password for clausyd:
Reading package lists... Done
Building dependency tree
Reading state information... Done
alien is already the newest version (8.95).
The following packages were automatically installed and are no longer required:
  linux-headers-4.15.0-29 linux-headers-4.15.0-29-generic
  linux-image-4.15.0-29-generic linux-modules-4.15.0-29-generic
  linux-modules-extra-4.15.0-29-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libodbc1 odbcinst odbcinst1debian2 vim-runtime
```

Figure 28: Installing the libraries needed for installation.

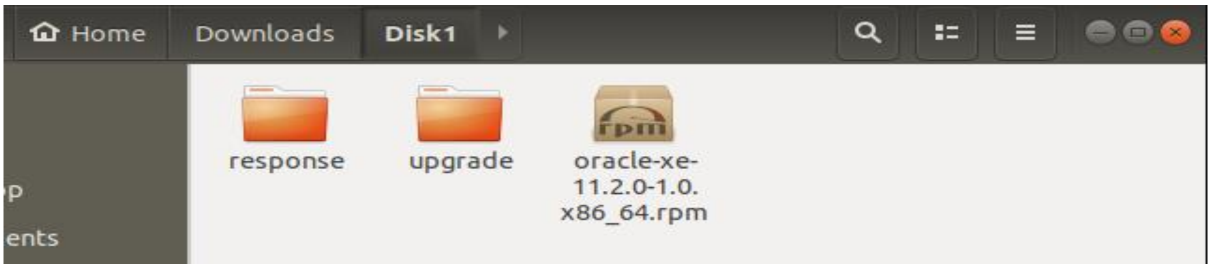


Figure 29: Decompressed .rpm file Disk1.

Once inside this folder they will see the .rpm file as seen in figure 29. After this I ran this command; sudo alien --scripts -d oracle-xe-11.2.0-1.0.x86_64.rpm, this command will convert the file if you refer to figure 30 you will see the new .deb file in the Disk1 directory. The –scripts means include all the script contained in the .rpm file.

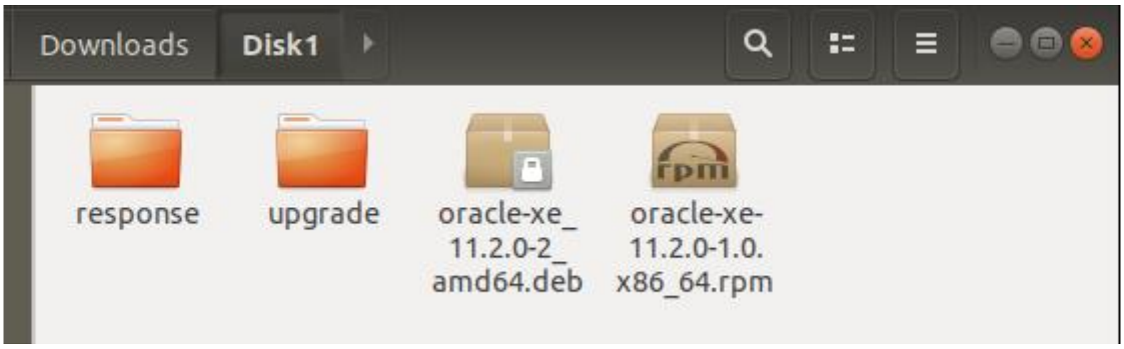
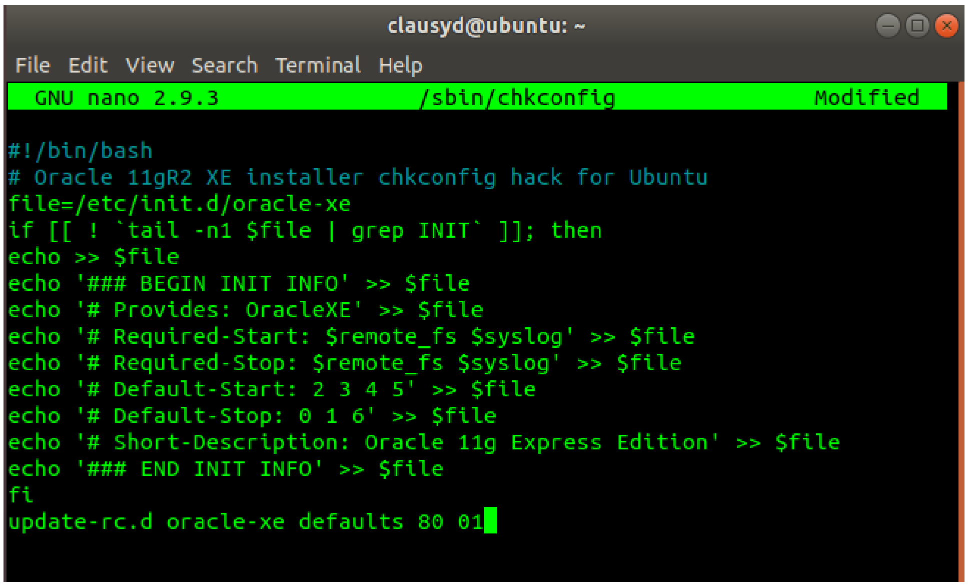


Figure 30: New .deb file is DIsk1 directory.

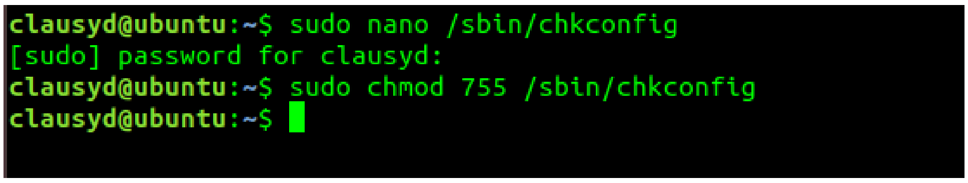
The next step was to create the install files needed for installing on Ubuntu. This is a simple bash file. In the tutorial the person uses the vim editor for me I prefer nano I find it easier to use. This script creates a file and makes it equal to this path show in figure 31. It then uses an if statement that if true will append the information to that file. All these commands are issued using the sudo command stating I am the administrator.



```
  GNU nano 2.9.3              /sbin/chkconfig              Modified

#!/bin/bash
# Oracle 11gR2 XE installer chkconfig hack for Ubuntu
file=/etc/init.d/oracle-xe
if [[ ! `tail -n1 $file | grep INIT` ]]; then
echo >> $file
echo '### BEGIN INIT INFO' >> $file
echo '# Provides: OracleXE' >> $file
echo '# Required-Start: $remote_fs $syslog' >> $file
echo '# Required-Stop: $remote_fs $syslog' >> $file
echo '# Default-Start: 2 3 4 5' >> $file
echo '# Default-Stop: 0 1 6' >> $file
echo '# Short-Description: Oracle 11g Express Edition' >> $file
echo '### END INIT INFO' >> $file
fi
update-rc.d oracle-xe defaults 80 01
```

Figure 31: Script needed for Ubuntu installation.

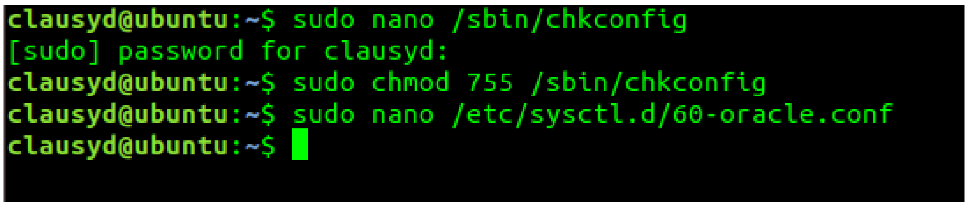
If you refer to figure 32, I am just using the chmod command to give the file executing privileges.



```
clausyd@ubuntu:~$ sudo nano /sbin/chkconfig
[sudo] password for clausyd:
clausyd@ubuntu:~$ sudo chmod 755 /sbin/chkconfig
clausyd@ubuntu:~$
```

Figure 32: Setting permissions for install script.

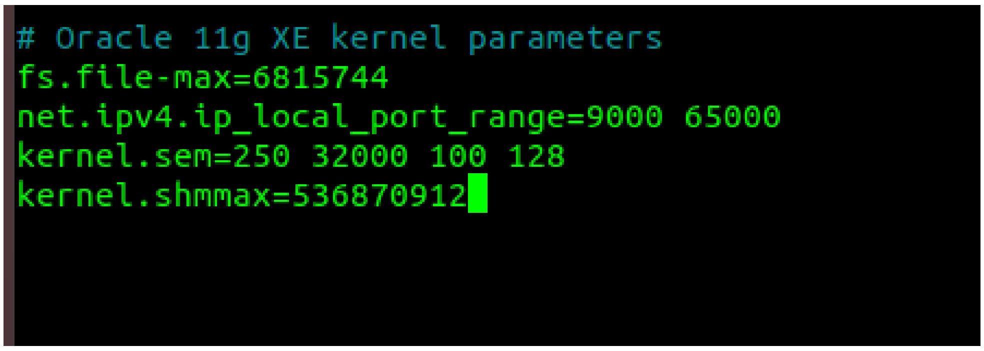
The next step is to configure the Oracle kernel parameters, again using sudo nano command I navigate to the oracle.conf file found in the etc directory under the system control directory.



```
clausyd@ubuntu:~$ sudo nano /sbin/chkconfig
[sudo] password for clausyd:
clausyd@ubuntu:~$ sudo chmod 755 /sbin/chkconfig
clausyd@ubuntu:~$ sudo nano /etc/sysctl.d/60-oracle.conf
clausyd@ubuntu:~$
```

Figure 33: File for Kernel parameters.

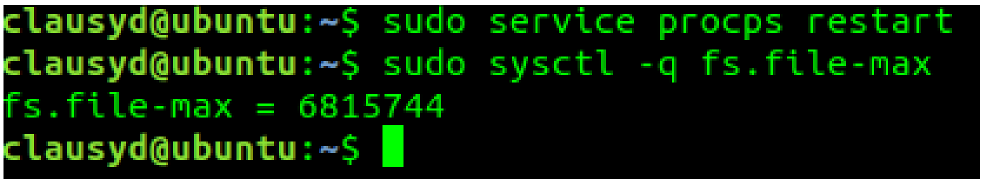
If you refer to figure 34 you will see the configuration script stating the size of the file, a port range. During the install I got an error relating to the kernel.shmmax. After a little researching I found the it was related to the size of the kernel.shmmax so I increased the size by changing the 53 to 54 and the install worked.



```
# Oracle 11g XE kernel parameters
fs.file-max=6815744
net.ipv4.ip_local_port_range=9000 65000
kernel.sem=250 32000 100 128
kernel.shmmax=536870912
```

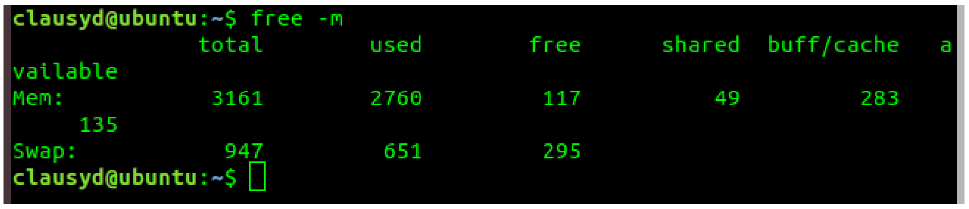
Figure 34: Setting kernel parameters.

Shown in figure 35 I am restarting the procps service and verifying the size of the fs.file size which was the same as the tutorial that is was following. *If you refer to figure 36, I was checking the size to my swap partition as it requires at least 2GB of free space for the install as you can see, I had just enough configured when setting up the virtual machine. If I didn't, I would have had to increase the size which would require me to follow another tutorial.*



```
clausyd@ubuntu:~$ sudo service procps restart
clausyd@ubuntu:~$ sudo sysctl -q fs.file-max
fs.file-max = 6815744
clausyd@ubuntu:~$
```

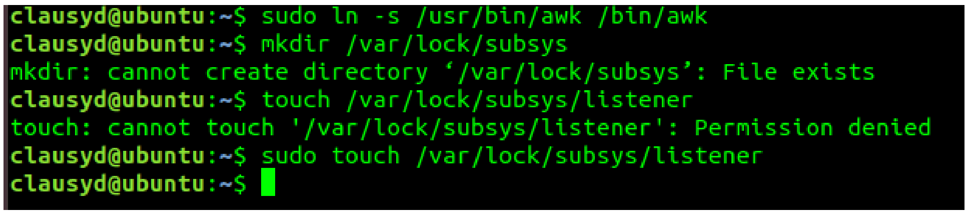
Figure 35: Restarting props service & verifying fs.file size.



```
clausyd@ubuntu:~$ free -m
              total        used        free      shared  buff/cache   a
vailable
Mem:           3161        2760         117          49         283
     135
Swap:           947         651         295
clausyd@ubuntu:~$
```

Figure 36: Verifying swap partition is greater than 2GB.

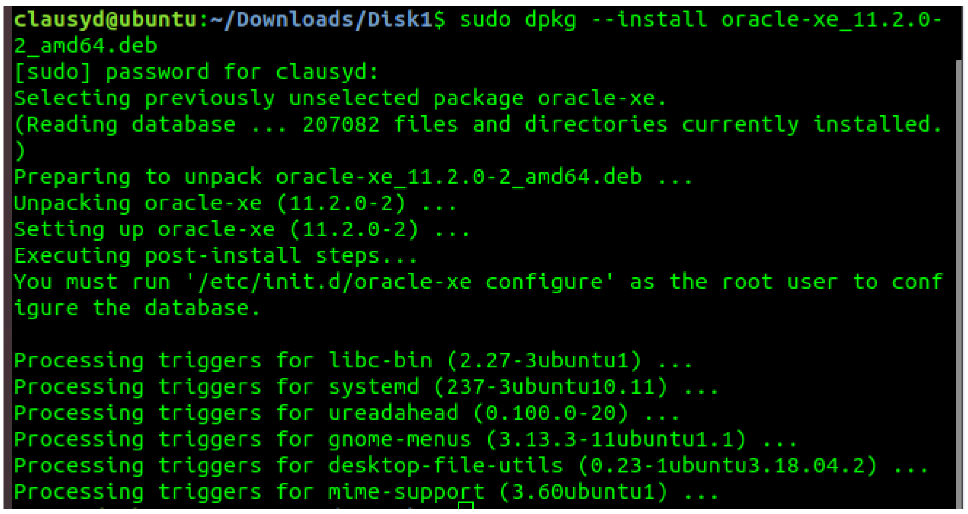
The next step was to create the required directories for the install process. For some reason some of these directories where already created as seen in figure 37. The first command is creating a link between paths, then the file is created a directory, but it said the directory already exists. The last command touch was to create a file for the listener.



```
clausyd@ubuntu:~$ sudo ln -s /usr/bin/awk /bin/awk
clausyd@ubuntu:~$ mkdir /var/lock/subsys
mkdir: cannot create directory ‘/var/lock/subsys’: File exists
clausyd@ubuntu:~$ touch /var/lock/subsys/listener
touch: cannot touch '/var/lock/subsys/listener': Permission denied
clausyd@ubuntu:~$ sudo touch /var/lock/subsys/listener
clausyd@ubuntu:~$ 
```

Figure 37: Creating required directories.

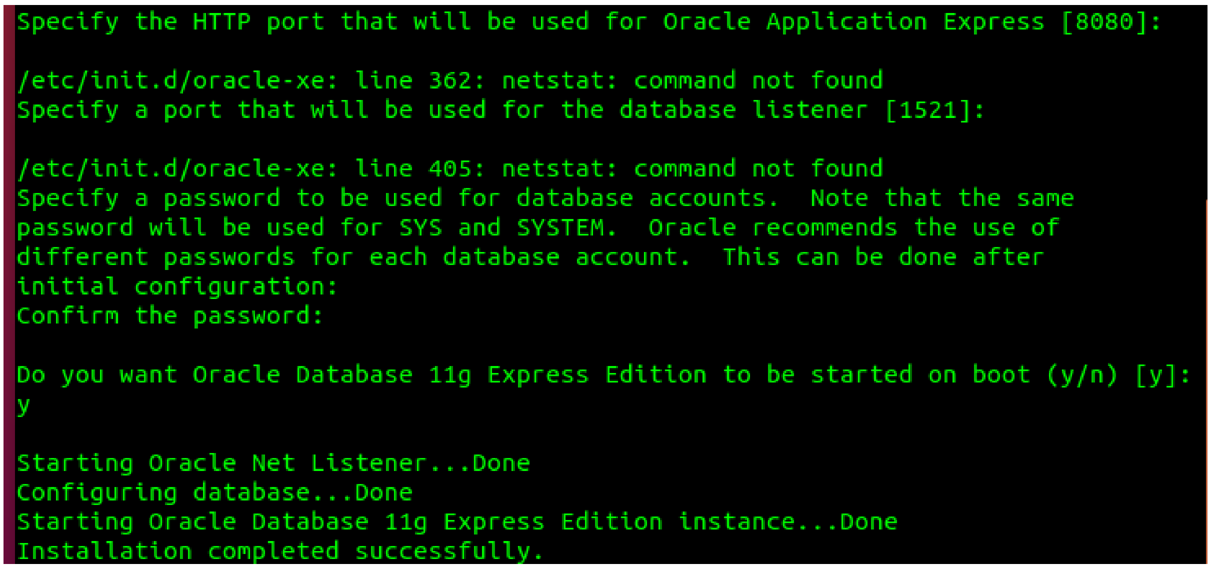
Finelly I was ready for installation. If you refer to figure 38, I can now use the dpkg –install command because the file has been converted to a .deb file. That command is used by Debien versions of Linux to install packages. The installation of the file and directories only took a few moments to compleate.



```
clausyd@ubuntu:~/Downloads/Disk1$ sudo dpkg --install oracle-xe_11.2.0-
2_amd64.deb
[sudo] password for clausyd:
Selecting previously unselected package oracle-xe.
(Reading database ... 207082 files and directories currently installed.
)
Preparing to unpack oracle-xe_11.2.0-2_amd64.deb ...
Unpacking oracle-xe (11.2.0-2) ...
Setting up oracle-xe (11.2.0-2) ...
Executing post-install steps...
You must run '/etc/init.d/oracle-xe configure' as the root user to conf
igure the database.

Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.11) ...
Processing triggers for ureadahead (0.100.0-20) ...
Processing triggers for gnome-menus (3.13.3-11ubuntu1.1) ...
Processing triggers for desktop-file-utils (0.23-1ubuntu3.18.04.2) ...
Processing triggers for mime-support (3.60ubuntu1) ...
```

Figure 38: Installing Oracle 11g R2.

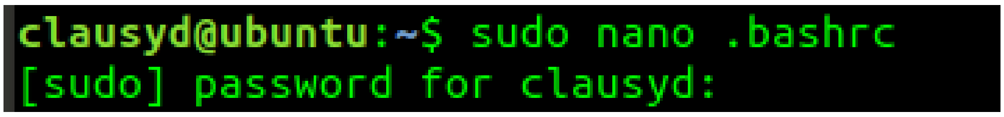
The next step was to configure Oracle the same as I done for the install in Window 10. I had to configure a port for the listener, the http port for the local host and to set a password for the administrator. After this the installation completed as shown in figure 39.



```
Specify the HTTP port that will be used for Oracle Application Express [8080]:

/etc/init.d/oracle-xe: line 362: netstat: command not found
Specify a port that will be used for the database listener [1521]:

/etc/init.d/oracle-xe: line 405: netstat: command not found
Specify a password to be used for database accounts.  Note that the same
password will be used for SYS and SYSTEM.  Oracle recommends the use of
different passwords for each database account.  This can be done after
initial configuration:
Confirm the password:

Do you want Oracle Database 11g Express Edition to be started on boot (y/n) [y]:
y

Starting Oracle Net Listener...Done
Configuring database...Done
Starting Oracle Database 11g Express Edition instance...Done
Installation completed successfully.
```

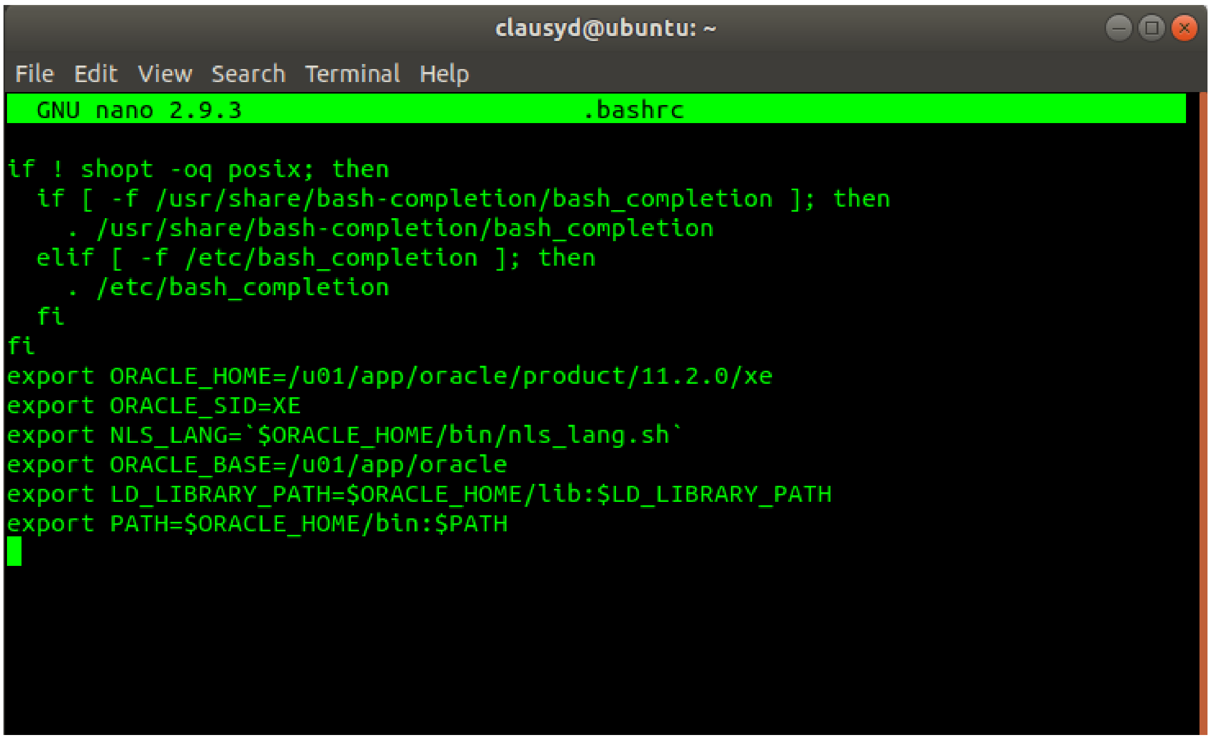
Figure 39: Configuration of Oracle database.

The last step was to set the environment variables. To do this I need to navigate to the .bashrc file shown in figure 40 and paste the script show below in figure 41.



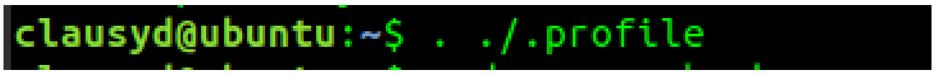
```
clausyd@ubuntu:~$ sudo nano .bashrc
[sudo] password for clausyd:
```

Figure 40: File for environment variables.



```
  GNU nano 2.9.3                      .bashrc

if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
export ORACLE_HOME=/u01/app/oracle/product/11.2.0/xe
export ORACLE_SID=XE
export NLS_LANG=`$ORACLE_HOME/bin/nls_lang.sh`
export ORACLE_BASE=/u01/app/oracle
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
export PATH=$ORACLE_HOME/bin:$PATH
```

Figure 41: Setting environment variables.



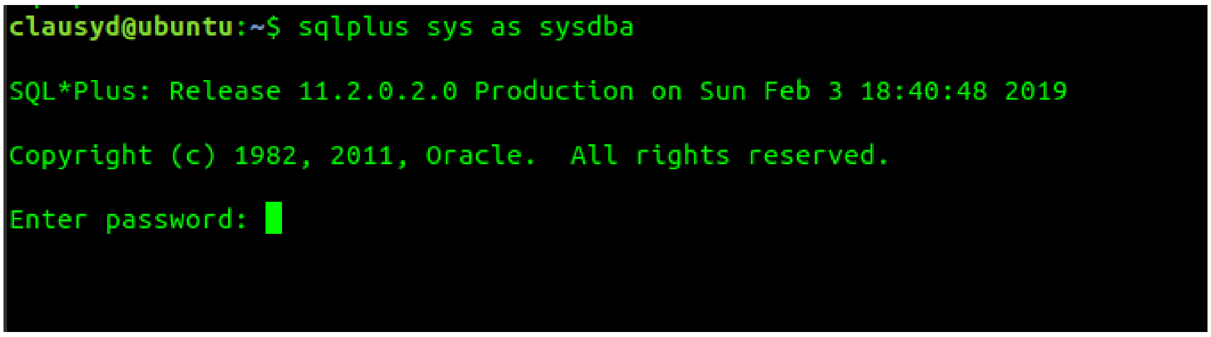
```
clausyd@ubuntu:~$ . ./.profile
```

Figure 42: Reloading user profile.

For these changes to take effect it was just a matter of reloading my profile show above in figure 42, after this I was ready the test connection to Oracle 11g.

1.2.1 Connection Test Ubuntu 18.04

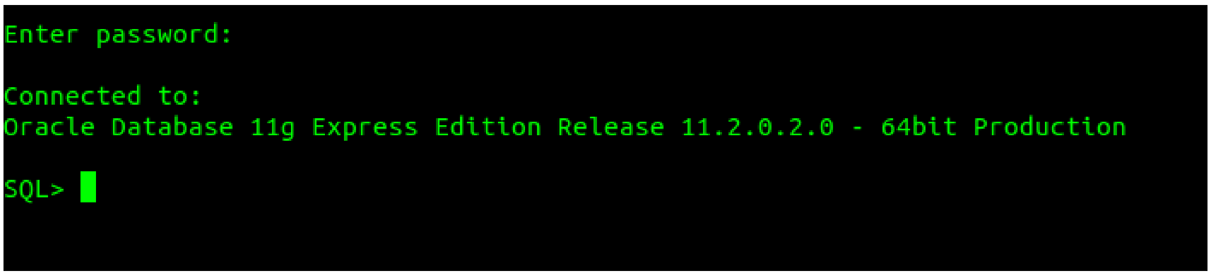
The last step in the installation process involved testing to see if the connection worked. If you refer to figure 43 you will see in am using SQLPlus and connecting as system administrator and using the password I configured during the configuration phase of the install.



```
clausyd@ubuntu:~$ sqlplus sys as sysdba

SQL*Plus: Release 11.2.0.2.0 Production on Sun Feb 3 18:40:48 2019

Copyright (c) 1982, 2011, Oracle.  All rights reserved.

Enter password:
```

Figure 43: Starting SQLPlus.

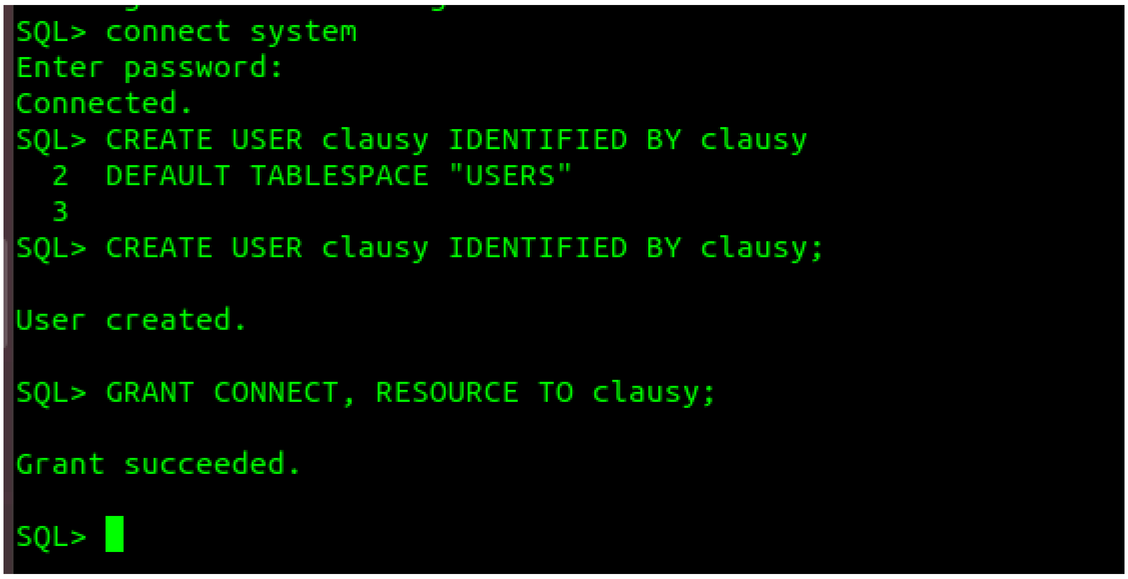
As you will see in figure 44 the connection was successful.



```
Enter password:

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL>
```

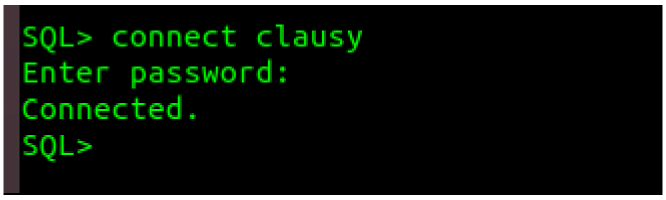
Figure 44: Connected to Oracle 11g R2

The next test involved creating a user and granting them some rights. If you refer to figure 45 below you will see I created a user called clausy and I granted, then privileges such as connect and resource this allows this user to connect to Oracle 11g. The resource is a powerful privilege as it allows the user to create DBA's. Shown in figure 46 below you will see that the connection by user clausy was successful. Now it is time to move on the file section of this report.



```
SQL> connect system
Enter password:
Connected.
SQL> CREATE USER clausy IDENTIFIED BY clausy
  2  DEFAULT TABLESPACE "USERS"
  3
SQL> CREATE USER clausy IDENTIFIED BY clausy;

User created.

SQL> GRANT CONNECT, RESOURCE TO clausy;

Grant succeeded.

SQL>
```

Figure 45: Creating admin user.



```
SQL> connect clausy
Enter password:
Connected.
SQL>
```

Figure 46: Connecting to admin user.

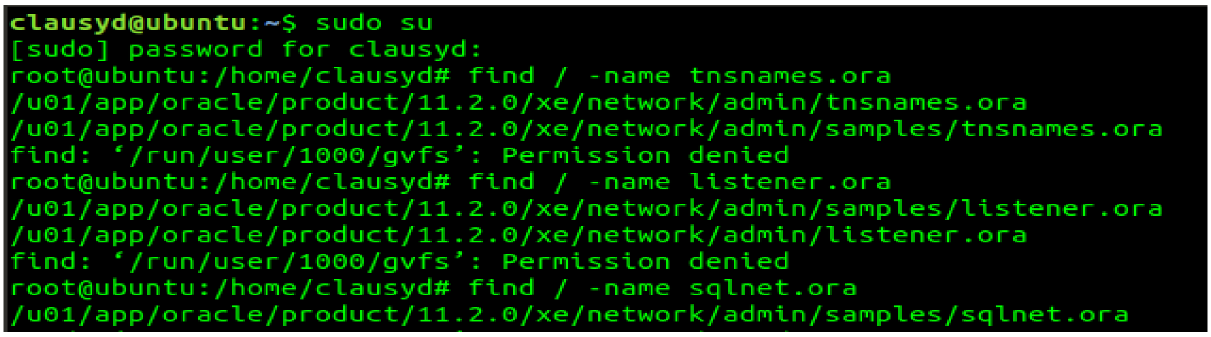
2. Files Management

2.1 Parameter Files

The parameter files indicate to the Oracle instance where the location of the control files is stored. Parameter file may also describe some initialization parameter the explain how big some memory structures may be. An Oracle database is made up of many different parameter files such as:

- tnsnames.ora
- listener.ora
- sqlnet.ora
- cman.ora
- ldap.ora

using the find command in Linux it is straight forward to locate the paths to these files. If you refer to figure 47 you will see the paths to these files. If you notice I am getting permission denied for some of the files. The most important file contained in the parameter file is the database parameter file. Without this file the database will not start.

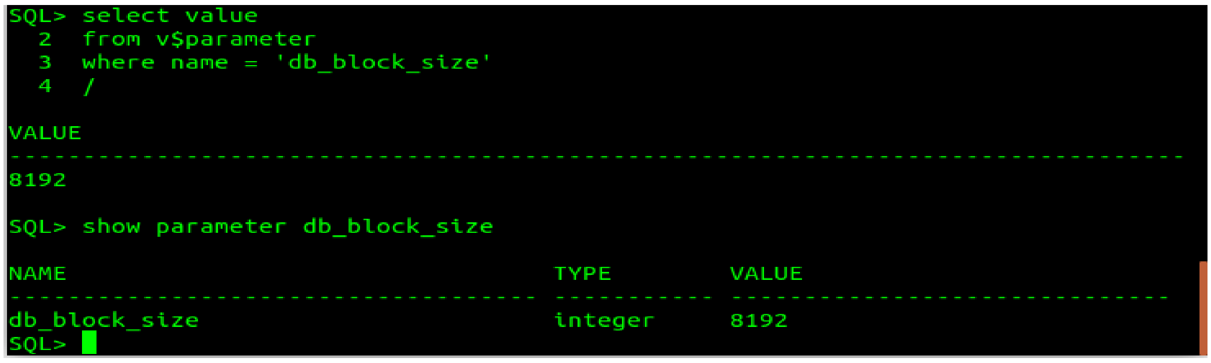


```
clausyd@ubuntu:~$ sudo su
[sudo] password for clausyd:
root@ubuntu:/home/clausyd# find / -name tnsnames.ora
/u01/app/oracle/product/11.2.0/xe/network/admin/tnsnames.ora
/u01/app/oracle/product/11.2.0/xe/network/admin/samples/tnsnames.ora
find: '/run/user/1000/gvfs': Permission denied
root@ubuntu:/home/clausyd# find / -name listener.ora
/u01/app/oracle/product/11.2.0/xe/network/admin/samples/listener.ora
/u01/app/oracle/product/11.2.0/xe/network/admin/listener.ora
find: '/run/user/1000/gvfs': Permission denied
root@ubuntu:/home/clausyd# find / -name sqlnet.ora
/u01/app/oracle/product/11.2.0/xe/network/admin/samples/sqlnet.ora
```

Figure 47: Associated parameter files & paths.

Init.ora file or sometimes referred to as the init file due to the historic default name which is init<ORACLE_SID>.ora. Its historic because after Oracle9i release 1 they vastly improved the method of storing parameter settings for their database. They did this by introducing the server parameter file or SPFILE. The default name is spfile<ORACLE_SID>.ora (Kyte, 2005).

The database parameter file can be seen as a key/value couple. (Kyte, 2005) explained in the book that the db_name is the key, where value may be the version of oracle such as ora11g. That is referred to in the book as the key value pair. To view the value of an instance parameter we can use V$PARAMETER or simply use the SHOW PARAMETER command is SQLPlus.

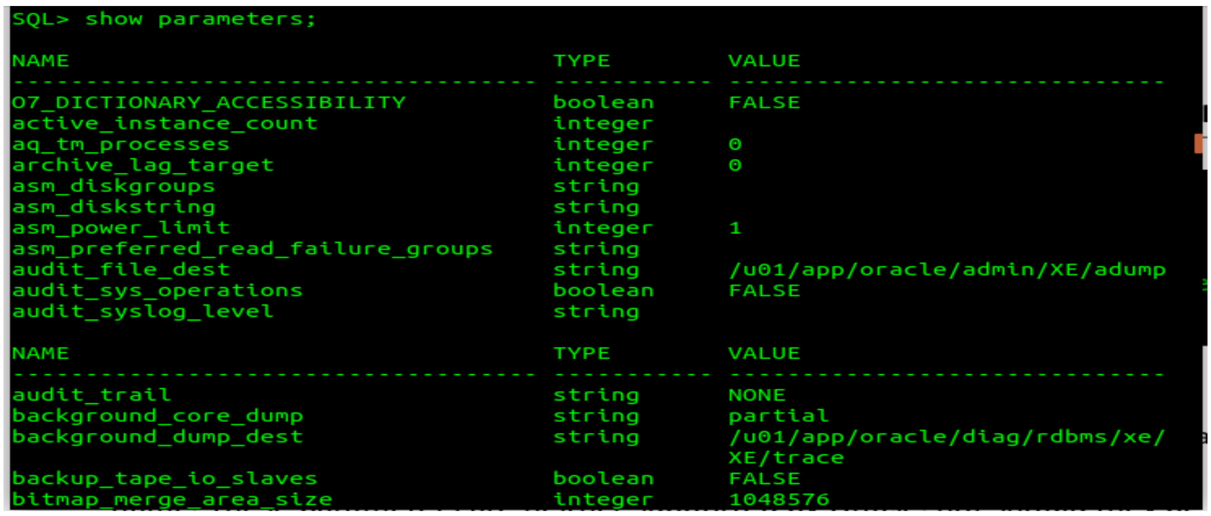


```
SQL> select value
  2  from v$parameter
  3  where name = 'db_block_size'
  4  /

VALUE
--------------------------------------------------------------------------------
8192

SQL> show parameter db_block_size

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
db_block_size                        integer     8192
SQL>
```

Figure 48: Using V$PARAMETER command to show database block size.

The db_block_size refers to the database block size, these are values of 4096 and 8192. These values are set when the database is setup and they cannot be modified. "The value of this parameter must be a multiple of the physical block size at the device level" (Oracle, 2019).

Another very useful command is the show parameter. If you refer to figure 49 this command will list every parameter and their associated values.

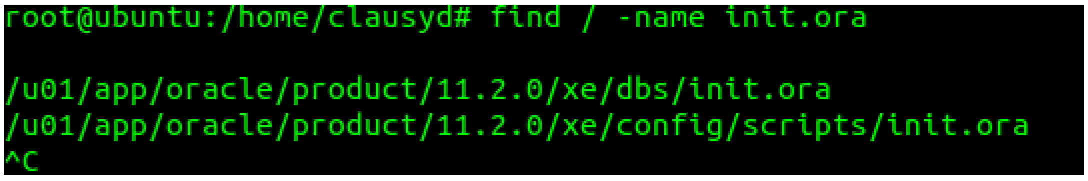


```
SQL> show parameters;

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
O7_DICTIONARY_ACCESSIBILITY          boolean     FALSE
active_instance_count                integer
aq_tm_processes                      integer     0
archive_lag_target                   integer     0
asm_diskgroups                       string
asm_diskstring                       string
asm_power_limit                      integer     1
asm_preferred_read_failure_groups    string
audit_file_dest                      string      /u01/app/oracle/admin/XE/adump
audit_sys_operations                 boolean     FALSE
audit_syslog_level                   string

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
audit_trail                          string      NONE
background_core_dump                 string      partial
background_dump_dest                 string      /u01/app/oracle/diag/rdbms/xe/
                                                 XE/trace
backup_tape_io_slaves                boolean     FALSE
bitmap_merge_area_size               integer     1048576
```

Figure 49: SHOW PARAMETER command.

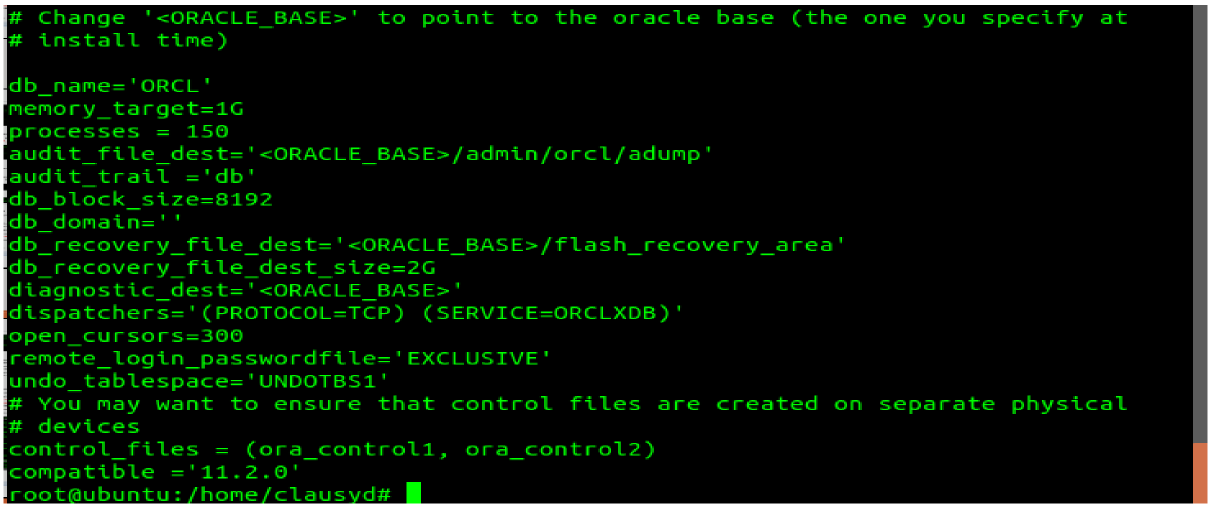
2.1.1 Legacy INIT

To locate the path to the init.ora file I used the find command in Linux. If you refer the figure 50 you will notice that there are two file paths shown. If you refer to figure 51 this file is provided by Oracle Corporation to help you start by providing a starting point to customize your RDBMS installation for your site.



```
root@ubuntu:/home/clausyd# find / -name init.ora

/u01/app/oracle/product/11.2.0/xe/dbs/init.ora
/u01/app/oracle/product/11.2.0/xe/config/scripts/init.ora
^C
```

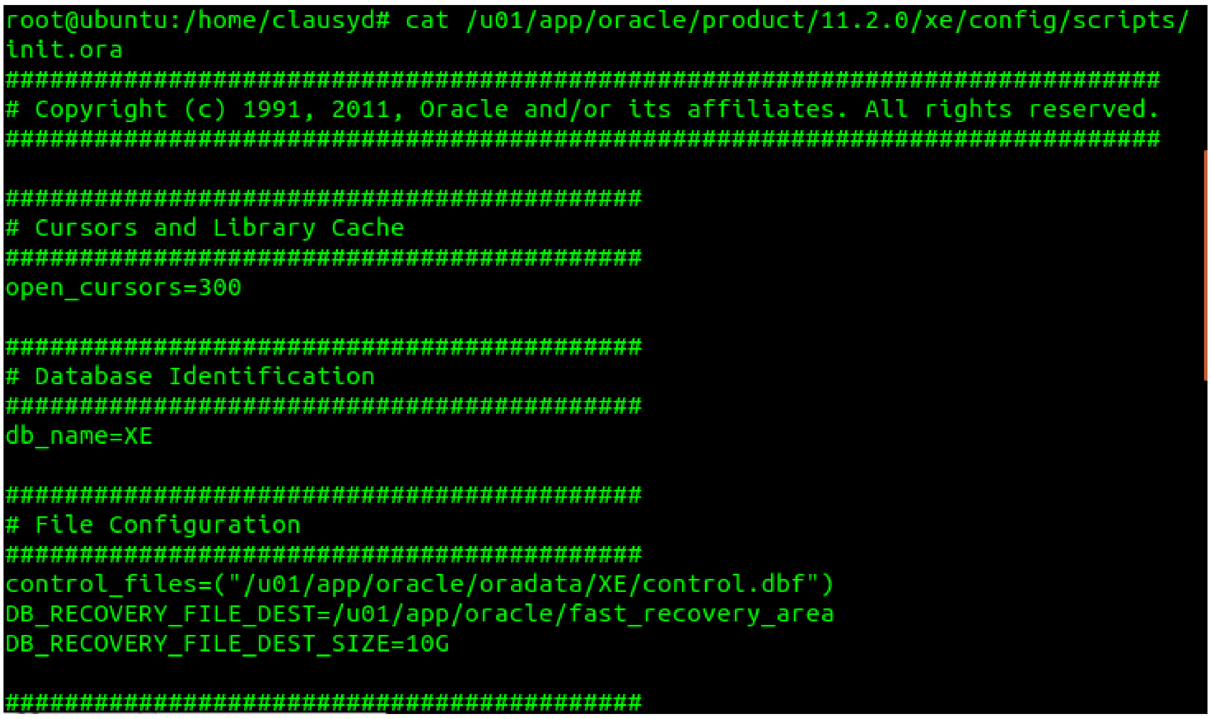
Figure 50: Finding the path for init.ora.



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

db_name='ORCL'
memory_target=1G
processes = 150
audit_file_dest='<ORACLE_BASE>/admin/orcl/adump'
audit_trail ='db'
db_block_size=8192
db_domain=''
db_recovery_file_dest='<ORACLE_BASE>/flash_recovery_area'
db_recovery_file_dest_size=2G
diagnostic_dest='<ORACLE_BASE>'
dispatchers='(PROTOCOL=TCP) (SERVICE=ORCLXDB)'
open_cursors=300
remote_login_passwordfile='EXCLUSIVE'
undo_tablespace='UNDOTBS1'
# You may want to ensure that control files are created on separate physical
# devices
control_files = (ora_control1, ora_control2)
compatible ='11.2.0'
root@ubuntu:/home/clausyd#
```

Figure 51: ORCL database setup on install.

Oracle init.ora is a fairly basic script. If you refer to figure 52 you will notice the key/value pair mentioned above. You will see the db_name is equal to the default xe as I have not created a database as of yet. Some of the other values it is telling the database is locations of the control file the recovery file and its size. It is initializing the parameters for the database.



```
root@ubuntu:/home/clausyd# cat /u01/app/oracle/product/11.2.0/xe/config/scripts/
init.ora
##############################################################################
# Copyright (c) 1991, 2011, Oracle and/or its affiliates. All rights reserved.
##############################################################################

###########################################
# Cursors and Library Cache
###########################################
open_cursors=300

###########################################
# Database Identification
###########################################
db_name=XE

###########################################
# File Configuration
###########################################
control_files=("/u01/app/oracle/oradata/XE/control.dbf")
DB_RECOVERY_FILE_DEST=/u01/app/oracle/fast_recovery_area
DB_RECOVERY_FILE_DEST_SIZE=10G

###########################################
```

Figure 52: init.ora file.

2.1.2 SPFILE

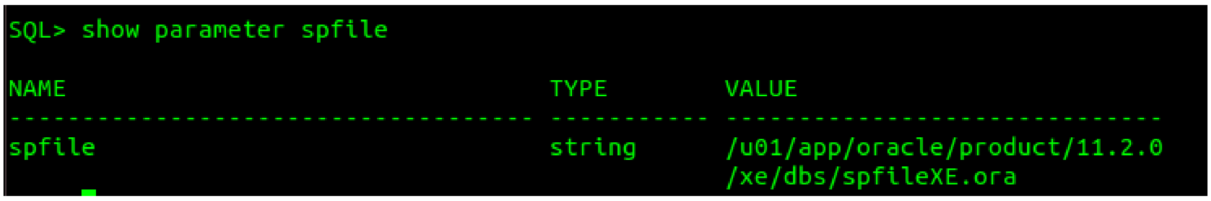
The SPFILE represented a big change in the way Oracle accesses and maintains the parameter setting for an instance. By introducing the SPFILE the removed two big problems with legacy files:

- It removes the proliferation of parameter files

- It removes the need to manually maintain parameter files using text editors outside of the database.

The first bullet points state that the SPFILE must be present on the server machine and not on the client machine. This mean that there is only a single source for the parameter settings. The second bullet point refers to the fact administrators don't have to maintain all of the parameter file, i.e. meaning that Oracle will automatically handle the file for them.

Using the show parameter spfile command you can see the location of the SPFILE as seen in figure 53. This is a binary file that contain the same information as the old PFILE. The SPFILE allow for dynamic change without restarting the instance. This is done through the alter command in SQLPlus and can't be done using text editors on Linux and Windows operating system (Oracle, 2019).

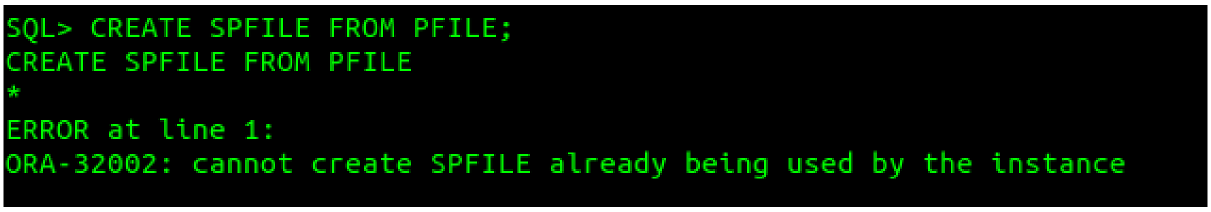


```
SQL> show parameter spfile

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
spfile                               string      /u01/app/oracle/product/11.2.0
                                                 /xe/dbs/spfileXE.ora
```

Figure 53: Showing parameter SPFILE.

The PFILE was used in Oracle until version 9i, this is when the SPFILE was introduced. By default, during install Oracle will default to SPFILE but the PFILE is still around and is needed for legacy system running older versions of Oracle. In my system the PFILE is found at /u01/app/oracle/admin/XE/pfile.

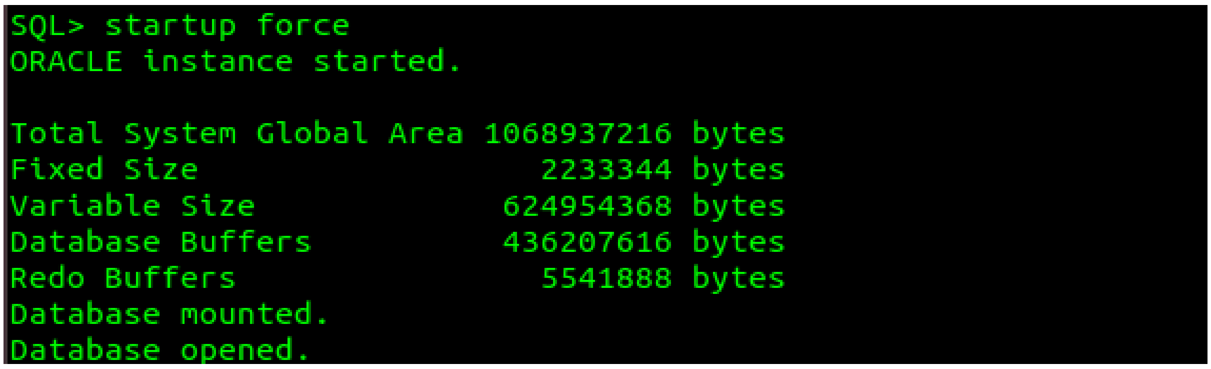
To test to see if the SPFILE was created during the install I used the create the SPFILE form the PFILE command but as shown in figure 54 you can see that the SPFILE is already in use.



```
SQL> CREATE SPFILE FROM PFILE;
CREATE SPFILE FROM PFILE
*
ERROR at line 1:
ORA-32002: cannot create SPFILE already being used by the instance
```

Figure 54: Creating SPFILE from PFILE error.

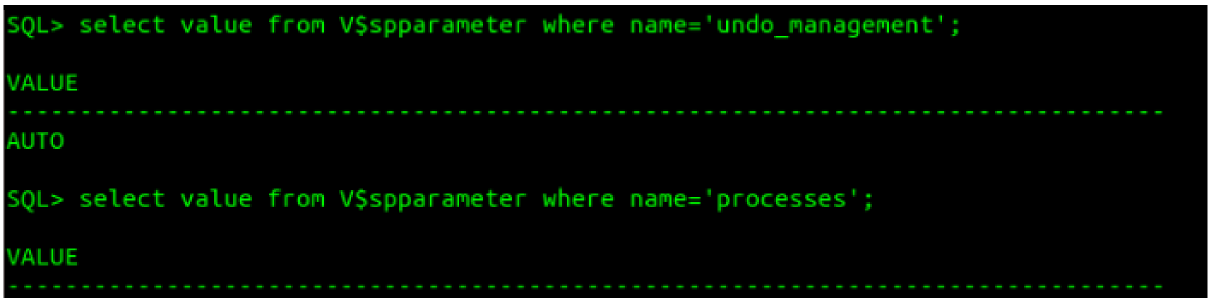
If a user wanted to upgrade a legacy system that was still running the PFILE to the newer version SPFILE you can perform a test by issuing the startup force command as seen in figure 55. If the reboot is successful, the system will reboot without problem.



```
SQL> startup force
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2233344 bytes
Variable Size             624954368 bytes
Database Buffers          436207616 bytes
Redo Buffers                5541888 bytes
Database mounted.
Database opened.
```

Figure 55: Forcing start-up command.

If the user was to run the show parameter pfile the output should be the SPFILE. And this would tell the dba that the system is now upgraded to the newer SPFILE.



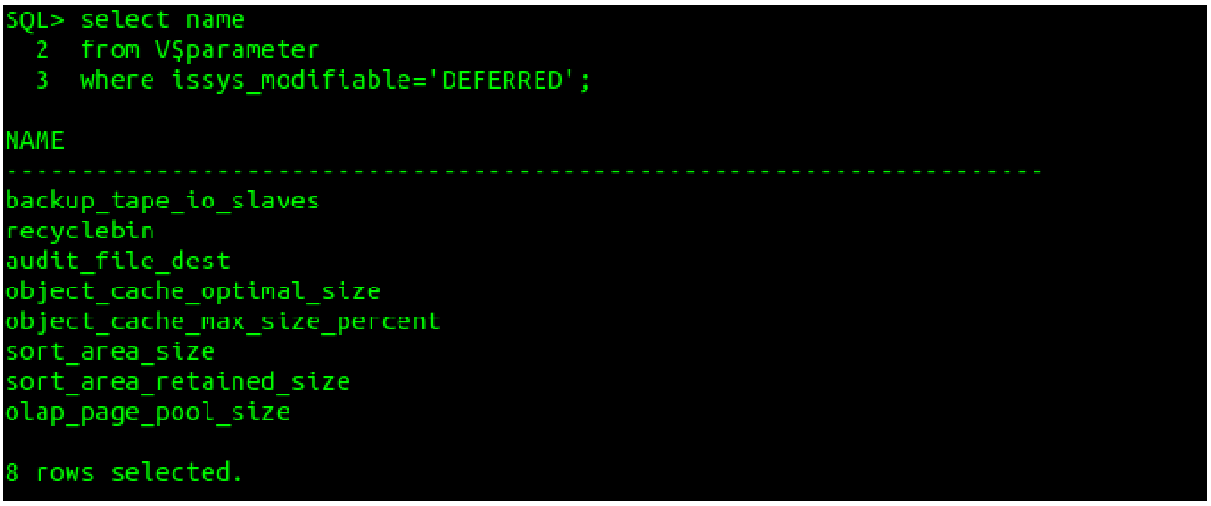
```
SQL> select value from V$spparameter where name='undo_management';

VALUE
--------------------------------------------------------------------------------
AUTO

SQL> select value from V$spparameter where name='processes';

VALUE
--------------------------------------------------------------------------------
```

Figure 56: Checking different values in the V$PARAMETER file.

By using the V$PARAMETER command users can display information about what is stored is the server parameter file. Show in figure 56 I am looking for to parameters of the server file the undo management which is set to auto and the parameter process. There is no value shown because there were no processes running on the server.

2.1.3 Setting Values in SPFILES

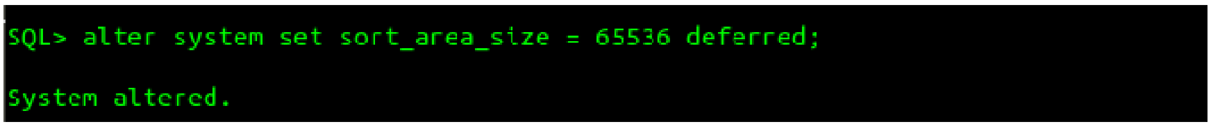
As the SPFILE is a binary file it can't be edited from a text editor on Ubuntu like nano or vim. But the SPFILE can be edited through SQLPlus using the alter system set command. First, we need to identify the parameter that can be altered using SQLPlus. Shown in figure 57 the select command is used. It is looks for the all the values by name that can be modified and that will be deferred until the next session starts.



```
SQL> select name
  2  from V$parameter
  3  where issys_modifiable='DEFERRED';

NAME
--------------------------------------------------------------------------------
backup_tape_io_slaves
recyclebin
audit_file_dest
object_cache_optimal_size
object_cache_max_size_percent
sort_area_size
sort_area_retained_size
olap_page_pool_size

8 rows selected.
```

Figure 57: Parameters that can be modified.

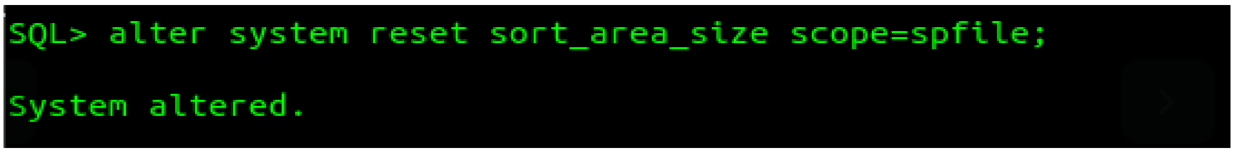
The sort area size monitors the amount of memory that Oracle uses for sorting. Shown in figure 58 the sort area size is changed to 65536 using the alter system command below and it completed successfully.



```
SQL> alter system set sort_area_size = 65536 deferred;

System altered.
```

Figure 58: Altering the sort_area_size in SPFILES.

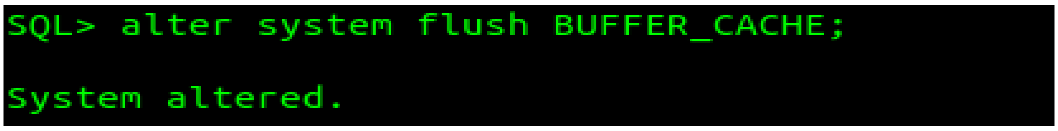
In figure 59 below I used the reset command to change the sort area size back to the default. This is done using the scope command. The scope of the SPFILE will be still default setting that is set up on install.



```
SQL> alter system reset sort_area_size scope=spfile;

System altered.
```

Figure 59: Resetting sort_area_size value.

The last step is the flush the buffer cache as shown in figure 60.



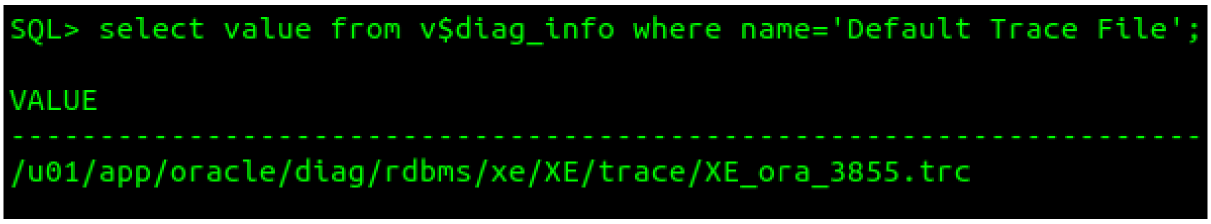
```
SQL> alter system flush BUFFER_CACHE;

System altered.
```

Figure 60: Flushing the buffer cache.

2.2 Trace Files

Trace files can be used for debugging. When the system has an error, the server produces a trace of performance related information. When Oracle built the system, there developer built a kernel and left all the debugging code in so the system administrators could track where the errors where coming from. Signs of this can been seen all over the system such as V$ view and sql_trace and Oracle events (Kyte, 2005).

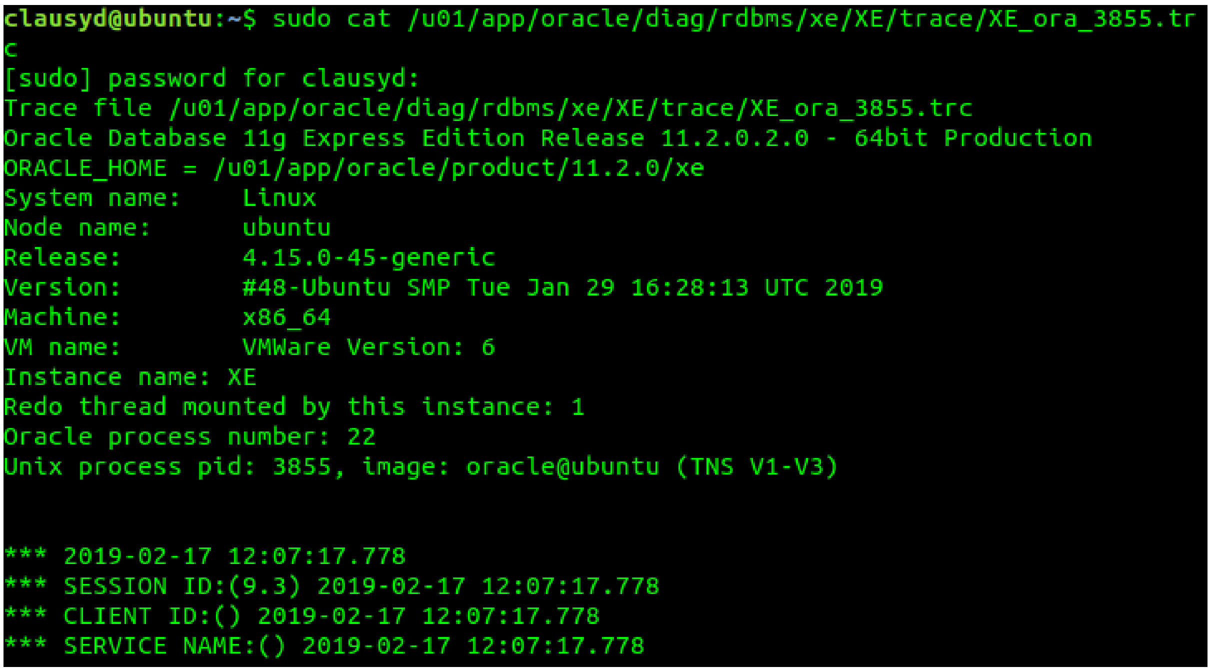
In figure 61 user can get the path to the trace file using v$diag_info command and naming the trace file they require. In the case below I looked for the default trace file.



```
SQL> select value from v$diag_info where name='Default Trace File';

VALUE
--------------------------------------------------------------------------------
/u01/app/oracle/diag/rdbms/xe/XE/trace/XE_ora_3855.trc
```

Figure 61: Finding the path to the default trace file.

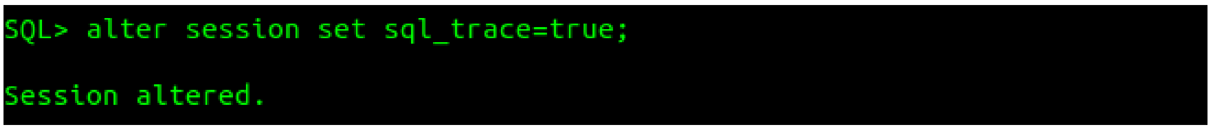
Using the cat command that Ubuntu provides users can view the contents of the trace file. Show below it has information about my system and the Oracle version I am currently running. Where the default folder is the process that are running and the number of instances that are running.



```
clausyd@ubuntu:~$ sudo cat /u01/app/oracle/diag/rdbms/xe/XE/trace/XE_ora_3855.tr
c
[sudo] password for clausyd:
Trace file /u01/app/oracle/diag/rdbms/xe/XE/trace/XE_ora_3855.trc
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
ORACLE_HOME = /u01/app/oracle/product/11.2.0/xe
System name:    Linux
Node name:      ubuntu
Release:        4.15.0-45-generic
Version:        #48-Ubuntu SMP Tue Jan 29 16:28:13 UTC 2019
Machine:        x86_64
VM name:        VMWare Version: 6
Instance name: XE
Redo thread mounted by this instance: 1
Oracle process number: 22
Unix process pid: 3855, image: oracle@ubuntu (TNS V1-V3)


*** 2019-02-17 12:07:17.778
*** SESSION ID:(9.3) 2019-02-17 12:07:17.778
*** CLIENT ID:() 2019-02-17 12:07:17.778
*** SERVICE NAME:() 2019-02-17 12:07:17.778
```

Figure 62: Using cat command to view the default trace file.

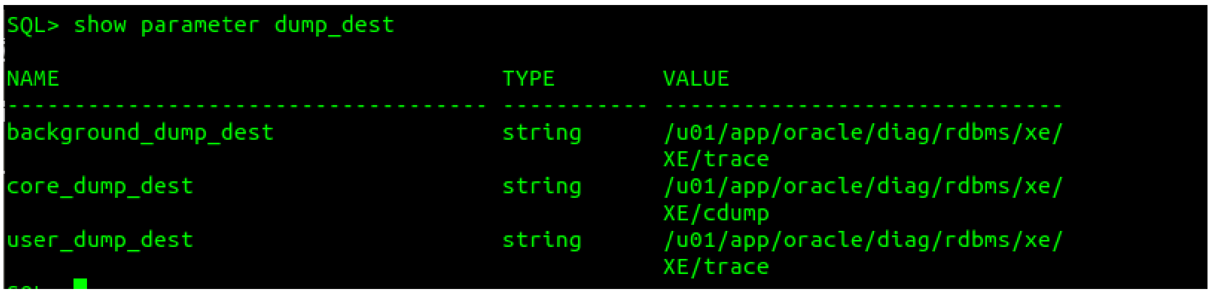
If users want to enable traces for the current session, they can use the command shown below in figure 63.



```
SQL> alter session set sql_trace=true;

Session altered.
```

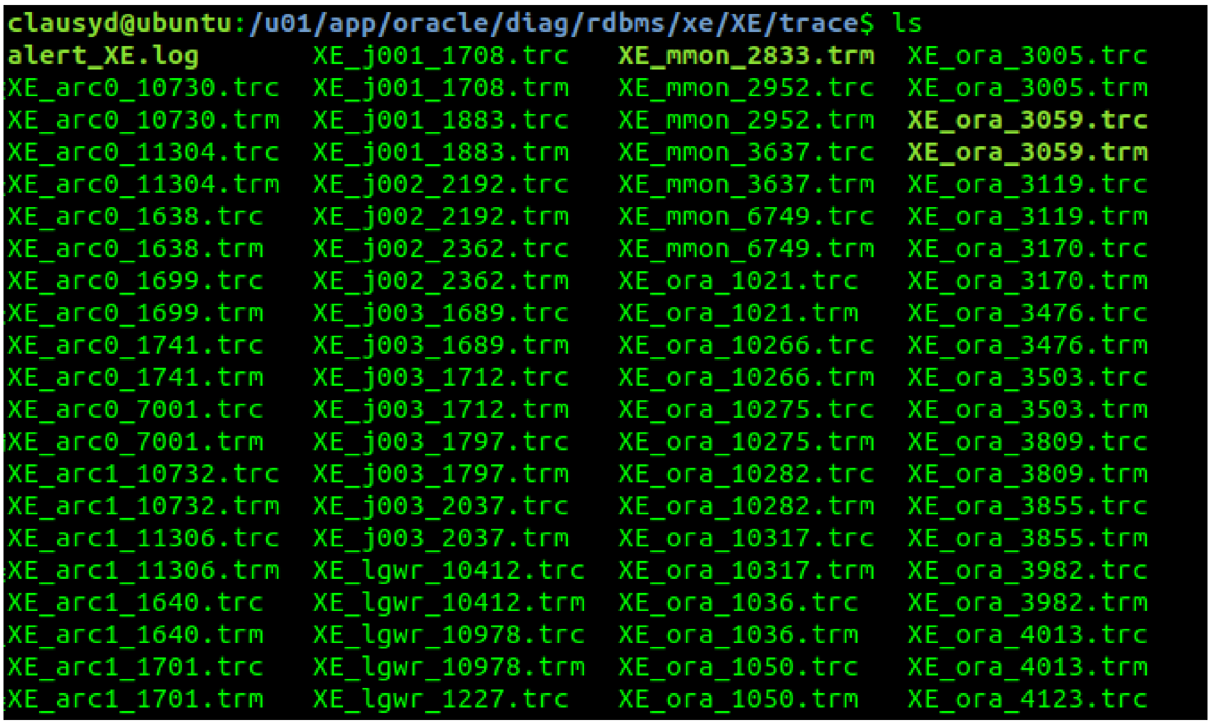
Figure 63: Enabling trace logging.

To see the location of all of the trace files dba's can issue the command referred to in figure 64. Shown in figure 65 is a ls command on the user dump destination. As you see it contains a lot of files. Oracle offers a way to format the file so that they are readable to the dba's using the ADRCI command.



```
SQL> show parameter dump_dest

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
background_dump_dest                 string      /u01/app/oracle/diag/rdbms/xe/
                                                 XE/trace
core_dump_dest                       string      /u01/app/oracle/diag/rdbms/xe/
                                                 XE/cdump
user_dump_dest                       string      /u01/app/oracle/diag/rdbms/xe/
                                                 XE/trace
```

Figure 64: Path values to Trace dump files.



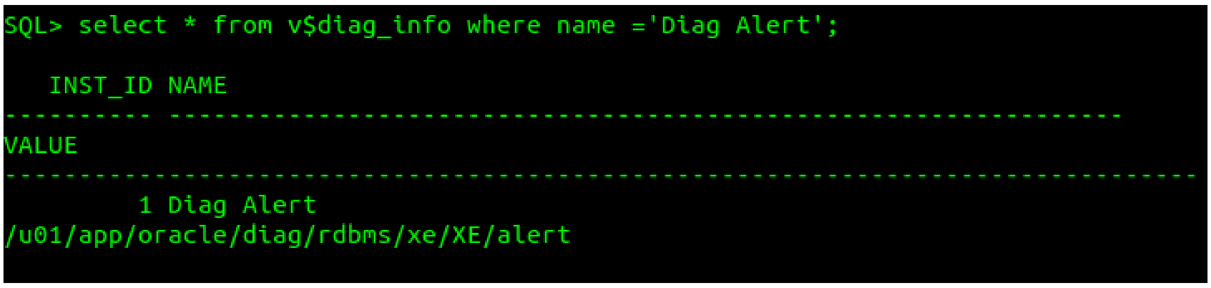
```
clausyd@ubuntu:/u01/app/oracle/diag/rdbms/xe/XE/trace$ ls
alert_XE.log          XE_j001_1708.trc    XE_mmon_2833.trm    XE_ora_3005.trc
XE_arc0_10730.trc     XE_j001_1708.trm    XE_mmon_2952.trc    XE_ora_3005.trm
XE_arc0_10730.trm     XE_j001_1883.trc    XE_mmon_2952.trm    XE_ora_3059.trc
XE_arc0_11304.trc     XE_j001_1883.trm    XE_mmon_3637.trc    XE_ora_3059.trm
XE_arc0_11304.trm     XE_j002_2192.trc    XE_mmon_3637.trm    XE_ora_3119.trc
XE_arc0_1638.trc      XE_j002_2192.trm    XE_mmon_6749.trc    XE_ora_3119.trm
XE_arc0_1638.trm      XE_j002_2362.trc    XE_mmon_6749.trm    XE_ora_3170.trc
XE_arc0_1699.trc      XE_j002_2362.trm    XE_ora_1021.trc     XE_ora_3170.trm
XE_arc0_1699.trm      XE_j003_1689.trc    XE_ora_1021.trm     XE_ora_3476.trc
XE_arc0_1741.trc      XE_j003_1689.trm    XE_ora_10266.trc    XE_ora_3476.trm
XE_arc0_1741.trm      XE_j003_1712.trc    XE_ora_10266.trm    XE_ora_3503.trc
XE_arc0_7001.trc      XE_j003_1712.trm    XE_ora_10275.trc    XE_ora_3503.trm
XE_arc0_7001.trm      XE_j003_1797.trc    XE_ora_10275.trm    XE_ora_3809.trc
XE_arc1_10732.trc     XE_j003_1797.trm    XE_ora_10282.trc    XE_ora_3809.trm
XE_arc1_10732.trm     XE_j003_2037.trc    XE_ora_10282.trm    XE_ora_3855.trc
XE_arc1_11306.trc     XE_j003_2037.trm    XE_ora_10317.trc    XE_ora_3855.trm
XE_arc1_11306.trm     XE_lgwr_10412.trc   XE_ora_10317.trm    XE_ora_3982.trc
XE_arc1_1640.trc      XE_lgwr_10412.trm   XE_ora_1036.trc     XE_ora_3982.trm
XE_arc1_1640.trm      XE_lgwr_10978.trc   XE_ora_1036.trm     XE_ora_4013.trc
XE_arc1_1701.trc      XE_lgwr_10978.trm   XE_ora_1050.trc     XE_ora_4013.trm
XE_arc1_1701.trm      XE_lgwr_1227.trc    XE_ora_1050.trm     XE_ora_4123.trc
```

Figure 65: Trace files.

2.3 Alert File

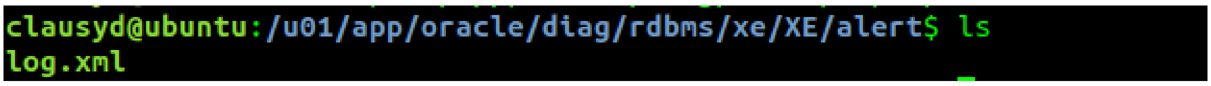
The alert file is a text file documenting a chronological history, from the day Oracle was installed until the day it is deleted from the system or put out of service. DBA's can let this file grow as large as they like. They can archive the data daily, monthly, yearly and so on. It stores useful information like internal error and log switching among a list of other useful information (Kyte, 2005).

Using the select * from v$diag_info command users can get the path to where the alert file is stored as referred to in figure 66. Going back to the Ubuntu bash I used cd to navigate to that file path and performed a ls and you can see the alert file as referred to in figure 67.



```
SQL> select * from v$diag_info where name ='Diag Alert';

   INST_ID NAME
---------- ----------------------------------------------------------------
VALUE
--------------------------------------------------------------------------------
         1 Diag Alert
/u01/app/oracle/diag/rdbms/xe/XE/alert
```

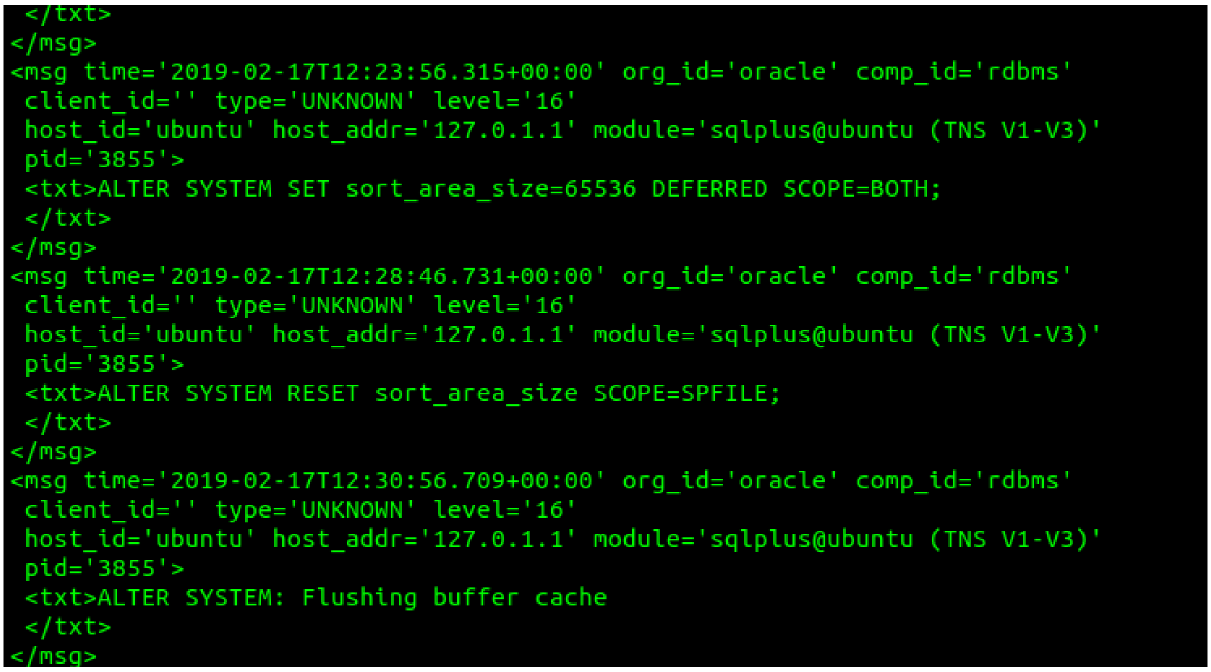
Figure 66: Getting the path to where the alert file is stored.



```
clausyd@ubuntu:/u01/app/oracle/diag/rdbms/xe/XE/alert$ ls
log.xml
```

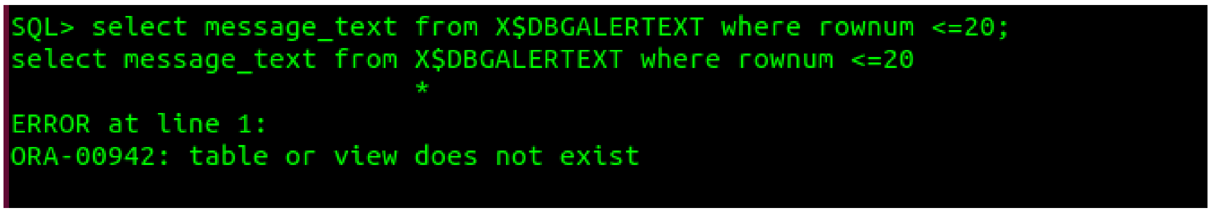
Figure 67: Path to the alert file and using the ls command to see what is inside the directory.

Using cat log.xml users can view this file but as you can see it is not very readable for the human eye. The sample report that was on Moodle spoke about X$DBGALERTEXT to view this file using SQL and its very readable. This is a fixed table that started in Oracle11g but unfortunately, if you refer to figure 69 it was not available on my machine. This could be down to the fact that Oracle 11g is not built to support the Debian Linux distro.



```
 </txt>
</msg>
<msg time='2019-02-17T12:23:56.315+00:00' org_id='oracle' comp_id='rdbms'
 client_id='' type='UNKNOWN' level='16'
 host_id='ubuntu' host_addr='127.0.1.1' module='sqlplus@ubuntu (TNS V1-V3)'
 pid='3855'>
 <txt>ALTER SYSTEM SET sort_area_size=65536 DEFERRED SCOPE=BOTH;
 </txt>
</msg>
<msg time='2019-02-17T12:28:46.731+00:00' org_id='oracle' comp_id='rdbms'
 client_id='' type='UNKNOWN' level='16'
 host_id='ubuntu' host_addr='127.0.1.1' module='sqlplus@ubuntu (TNS V1-V3)'
 pid='3855'>
 <txt>ALTER SYSTEM RESET sort_area_size SCOPE=SPFILE;
 </txt>
</msg>
<msg time='2019-02-17T12:30:56.709+00:00' org_id='oracle' comp_id='rdbms'
 client_id='' type='UNKNOWN' level='16'
 host_id='ubuntu' host_addr='127.0.1.1' module='sqlplus@ubuntu (TNS V1-V3)'
 pid='3855'>
 <txt>ALTER SYSTEM: Flushing buffer cache
 </txt>
</msg>
```

Figure 68: The xml alert file.



```
SQL> select message_text from X$DBGALERTEXT where rownum <=20;
select message_text from X$DBGALERTEXT where rownum <=20
                         *
ERROR at line 1:
ORA-00942: table or view does not exist
```

Figure 69: Using SQL to read alert file.

2.4 Data Files

(Kyte, 2005) say that "Data files, along with redo log files, are the most important type of files in the database. Data file are made up of tablespaces, segments, extents and blocks and is where all the data is stored. If you refer to figure 70 it will give you an idea of what a tablespace is made up of.

- **Tablespaces** are the containers for segments which in turn hold the extent and the extents hold the blocks. Each segment belongs to only one tablespace and never cross tablespace boundaries.
- **Segments** are made up of the database object such as tables and indexes and layout the structure within the tablespace.
- **Extents** are logically continuous allocation of space in the file. But is reality this may not be the case because with the use of RAID technology files may be spread out over many disks. Segments will be made up of one or many extents.
- **Block** are contained within the extents and for a DB developer would be seen as the rows of data in the database.

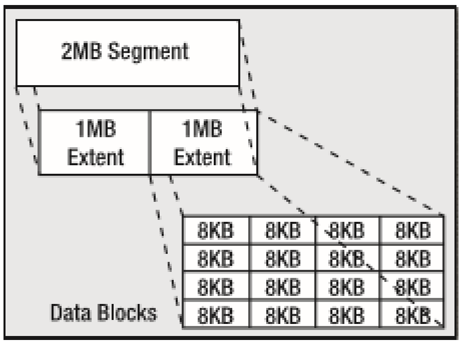
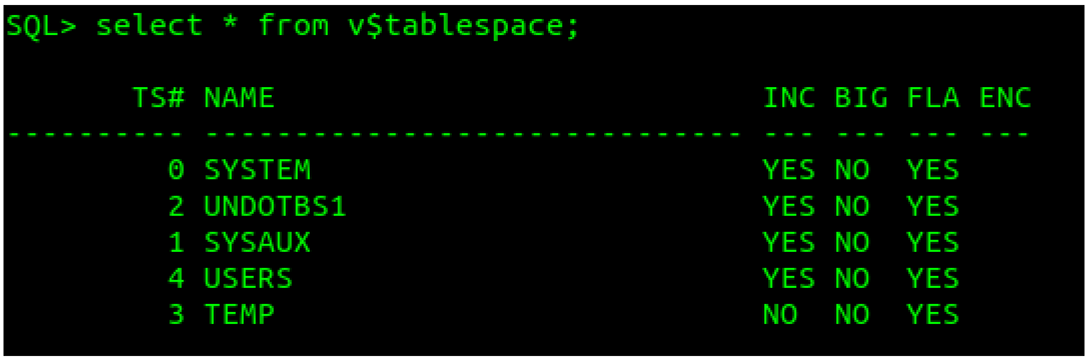


Figure 70: Segments, extents and blocks.

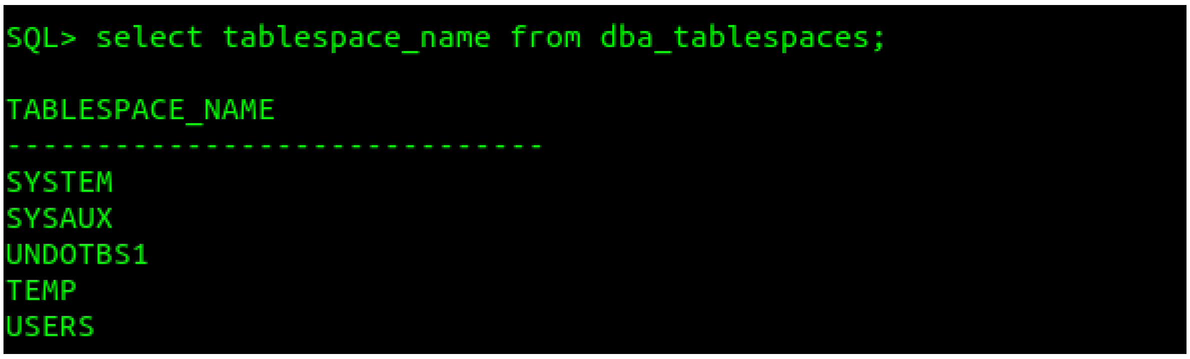
Using the simple command CREATE DATABASE will create three separate data files, system, users and sysaux. If you refer to figure 71 you will see all the values return for a select statement on the tablespace. You will notice the three files I referred to above along with the temp and undotbs1 files.



```
SQL> select * from v$tablespace;

       TS# NAME                           INC BIG FLA ENC
---------- ------------------------------ --- --- --- ---
         0 SYSTEM                         YES NO  YES
         2 UNDOTBS1                       YES NO  YES
         1 SYSAUX                         YES NO  YES
         4 USERS                          YES NO  YES
         3 TEMP                           NO  NO  YES
```

Figure 71: Listing all table-spaces.

Referred to in figure 72 I am being more specific in my search terms by stating only to return the tablespace name for the database tablespaces.



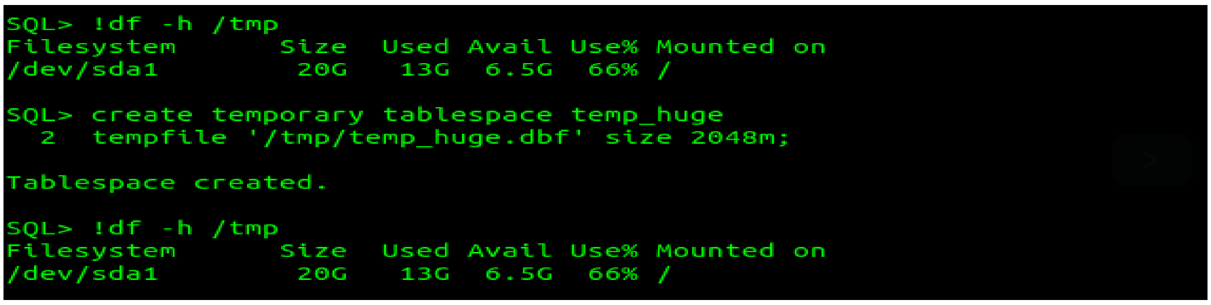
```
SQL> select tablespace_name from dba_tablespaces;

TABLESPACE_NAME
------------------------------
SYSTEM
SYSAUX
UNDOTBS1
TEMP
USERS
```

Figure 72: Select only table-space names from database table-spaces.

2.5 Temp Files

Temp file or temporary files are used by Oracle to store the intermediate result of big sort operations as well as hash functions. Temp file may also store global temporary data if there isn't enough RAM to hold it all. Data that Oracle sees as permanent like tables or indexes will never be stored in temp files.

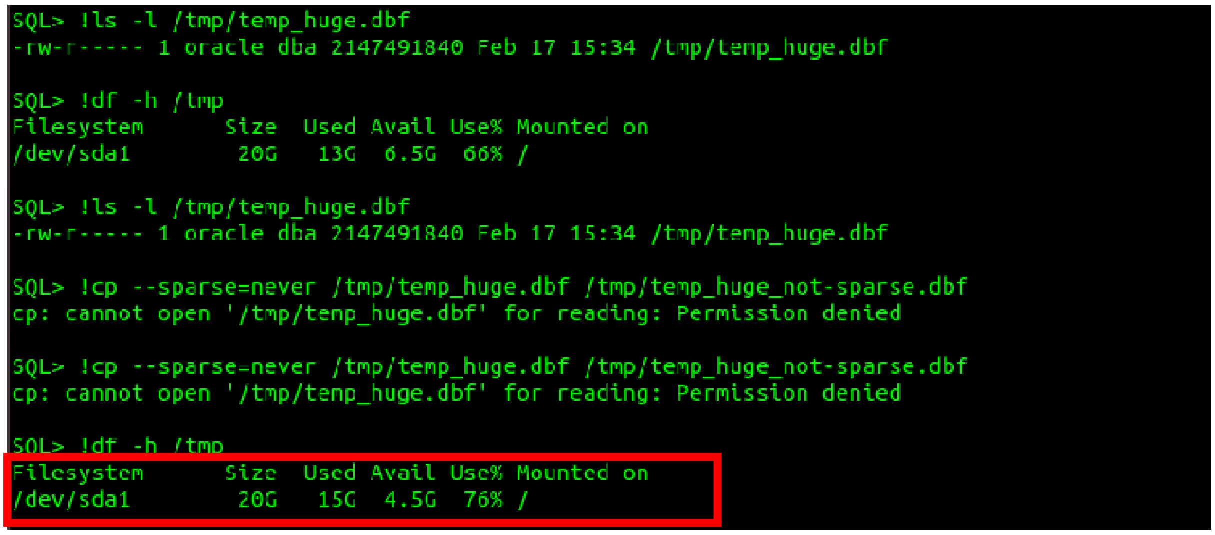
In SQLPlus by using the ! sign users can interact with the command line interface on Linux system this is very useful as there is no need to open a new terminal. Temporary files are created sparse, this means they don't consume any disk space until they need to (Kyte, 2005). By using the !df -h users can see the disk space and its statistics here I am looking at the tmp file. To test if it's sparse we will create a table space that is 2GB in size and use the command again and you will see the used disk space didn't change.



```
SQL> !df -h /tmp
Filesystem       Size  Used Avail Use% Mounted on
/dev/sda1         20G   13G  6.5G  66% /

SQL> create temporary tablespace temp_huge
  2  tempfile '/tmp/temp_huge.dbf' size 2048m;

Tablespace created.

SQL> !df -h /tmp
Filesystem       Size  Used Avail Use% Mounted on
/dev/sda1         20G   13G  6.5G  66% /
```

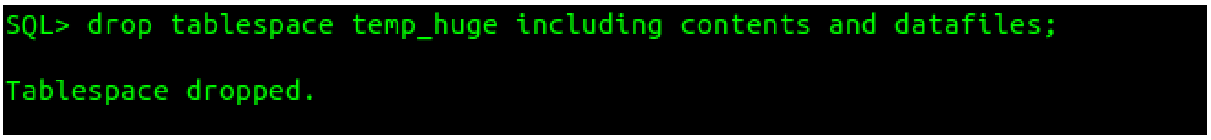
Figure 73: Creating a temp file. Using ! to interact with Ubuntu bash.

If you refer to figure 74, I am using the ls command on the directory that was created, and you will see that it is just over the 2GB size. What I done then was to try and copy the temp_huge file into a new directory called temp_huge_not_sparse.dbf file. For some reason I was getting permission denied so I cheated a little and opened a new command prompt and used the exact same cp command and if you refer to the bottom of figure 74 highlighted in red, you will see that the used disk space increased to 15G this is because the temp file in use. Finally, in figure 75 the drop command was used to drop the tablespace and all its contents.



```
SQL> !ls -l /tmp/temp_huge.dbf
-rw-r----- 1 oracle dba 2147491840 Feb 17 15:34 /tmp/temp_huge.dbf

SQL> !df -h /tmp
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1        20G   13G  6.5G  66% /

SQL> !ls -l /tmp/temp_huge.dbf
-rw-r----- 1 oracle dba 2147491840 Feb 17 15:34 /tmp/temp_huge.dbf

SQL> !cp --sparse=never /tmp/temp_huge.dbf /tmp/temp_huge_not-sparse.dbf
cp: cannot open '/tmp/temp_huge.dbf' for reading: Permission denied

SQL> !cp --sparse=never /tmp/temp_huge.dbf /tmp/temp_huge_not-sparse.dbf
cp: cannot open '/tmp/temp_huge.dbf' for reading: Permission denied

SQL> !df -h /tmp
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1        20G   15G  4.5G  76% /
```

Figure 74: Showing the change through sparse command.



```
SQL> drop tablespace temp_huge including contents and datafiles;

Tablespace dropped.
```

Figure 75: Dropping temp file.

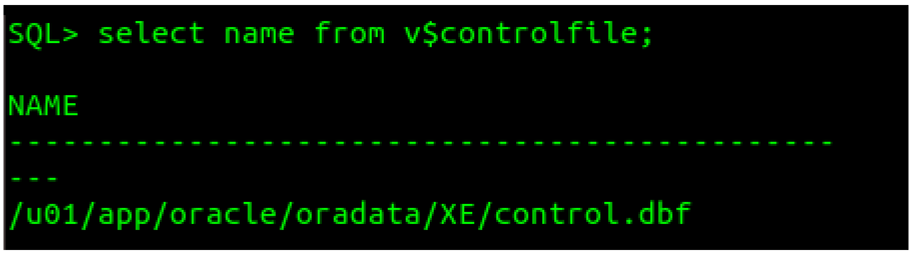
2.6 Control Files

Control file are usually small in size but in some case may grow up to 64mb in size (Kyte, 2005). Every Oracle database will contain a control file. It is a binary file and will contain details such as:

- Database name
- Time stamp of database creation
- Names and locations of data files
- Names and locations of redo log files
- The current log sequence number
- Checkpoint information
- Recent RMAN backups taken

(Oracle, 2019)

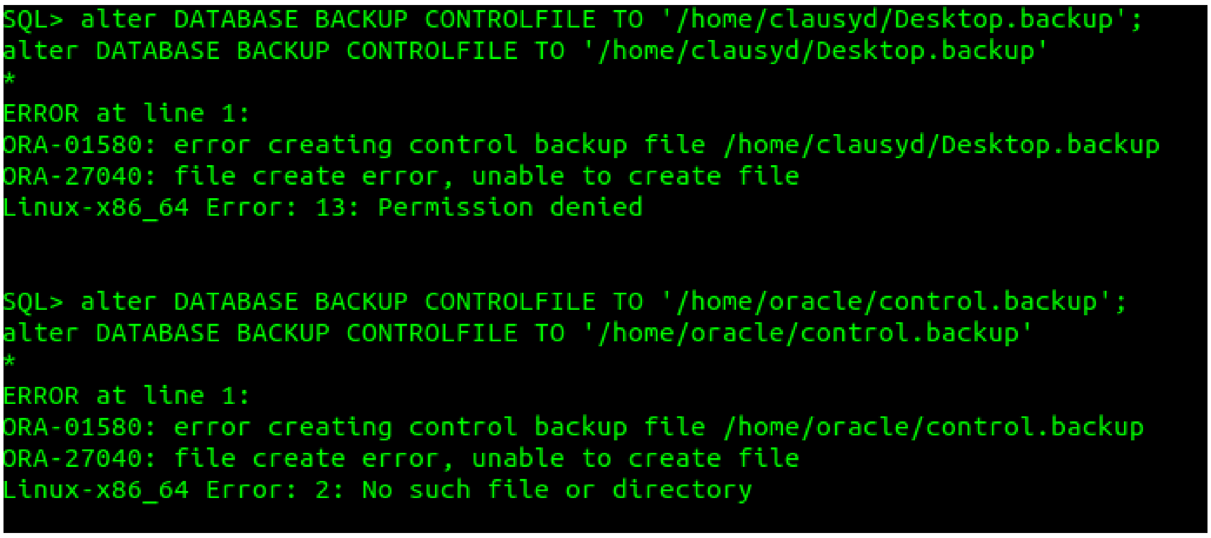
If you refer to figure 76 the select name from controlfile will return the path to where this file is stored. (Experts Exchange,2010) state that more the one copy of the control file should be stored and should be stored in different locations to avoid the risk of the being deleted from the machine. If you notice my Ubuntu distro only has on copy of the control file. I will try and attempt to make a backup of the control file using the alter command.



```
SQL> select name from v$controlfile;

NAME
--------------------------------------------------
---
/u01/app/oracle/oradata/XE/control.dbf
```

Figure 76: Path to control file.

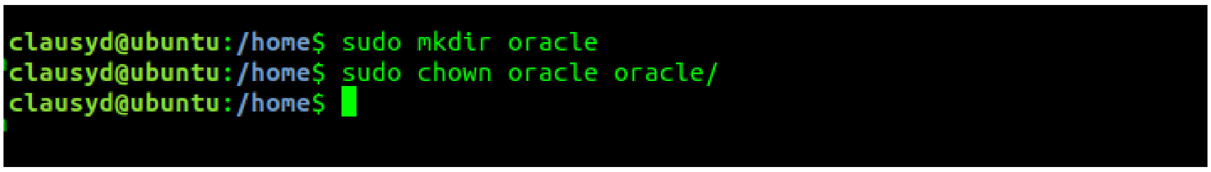
If you refer to figure 77 you will see that my OS is not giving permission to make this backup file and say its unable to create the file. I then tried to create a backup of the control file in a directory called Oracle in my home directory again this was unsuccessful and there was no file or directory.



```
SQL> alter DATABASE BACKUP CONTROLFILE TO '/home/clausyd/Desktop.backup';
alter DATABASE BACKUP CONTROLFILE TO '/home/clausyd/Desktop.backup'
*
ERROR at line 1:
ORA-01580: error creating control backup file /home/clausyd/Desktop.backup
ORA-27040: file create error, unable to create file
Linux-x86_64 Error: 13: Permission denied


SQL> alter DATABASE BACKUP CONTROLFILE TO '/home/oracle/control.backup';
alter DATABASE BACKUP CONTROLFILE TO '/home/oracle/control.backup'
*
ERROR at line 1:
ORA-01580: error creating control backup file /home/oracle/control.backup
ORA-27040: file create error, unable to create file
Linux-x86_64 Error: 2: No such file or directory
```

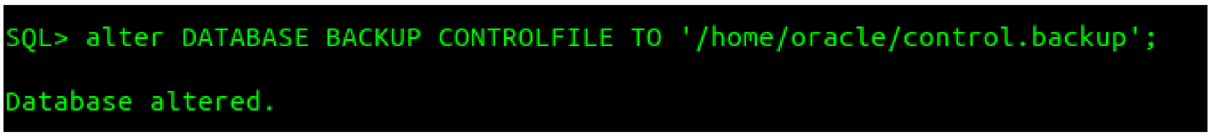
Figure 77: Error altering control file.

To make the backup, Oracle must own the folder that the backup is being placed in. Next, I made a new directory using the sudo mkdir command to state that I am the root and the name of the folder. Then I used the chown command to give ownership off the folder to Oracle as seen in figure 78. If you refer to figure 79 you will see that the backup was no successful.



```
clausyd@ubuntu:/home$ sudo mkdir oracle
clausyd@ubuntu:/home$ sudo chown oracle oracle/
clausyd@ubuntu:/home$
```

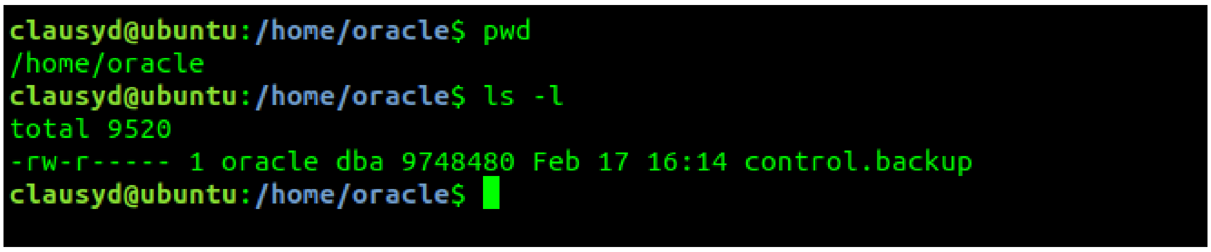
Figure 78: Creating directory for control file and giving ownership to Oracle.



```
SQL> alter DATABASE BACKUP CONTROLFILE TO '/home/oracle/control.backup';

Database altered.
```

Figure 79: control file now altered.

If you refer to figure 80 using the pwd command which is present working directory, then using the ls -l command to list what is inside the directory and all the directory details and you can see that the backup was successful. (O'Brien, 2014)



```
clausyd@ubuntu:/home/oracle$ pwd
/home/oracle
clausyd@ubuntu:/home/oracle$ ls -l
total 9520
-rw-r----- 1 oracle dba 9748480 Feb 17 16:14 control.backup
clausyd@ubuntu:/home/oracle$
```

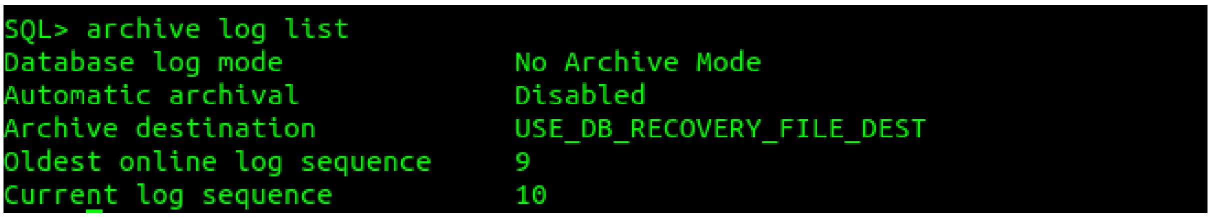
Figure 80: Displaying back-up of control file.

2.7 Redo Log Files

“Redo log files are crucial to the Oracle database”. These are operation logs for the database. They are used for recovery when something goes wrong with the database. They can all so be used for replication, standby database processing and media recovery after a data file restore from a backup (Kyte, 2005). If there is a power failure or human error such as deletion of table or disks containing data become corrupted the point of failure can be recovered by the redo log files. “Virtually every operation you perform in Oracle generates some amount of redo to be written to the online redo log file” (Kyte, 2005).

The archive log will send redo logs files to an archive after a certain amount of time depending on the business need for data. If archiving is disabled as shown is figure 81 Oracle will overwrite redo log file and data will be lost forever. If you only do a backup on a Sunday night and on a Friday, the disk becomes corrupted then the only option may be to roll back to the previous Sunday losing a weeks’ worth of data which depending on the business could be catastrophic. Having archiving mode enabled you would simply get another disk and restore the file that are affected.

By enabling this feature businesses can track all changes that happen to the database over the course of its life time by storing all the data in archive files.

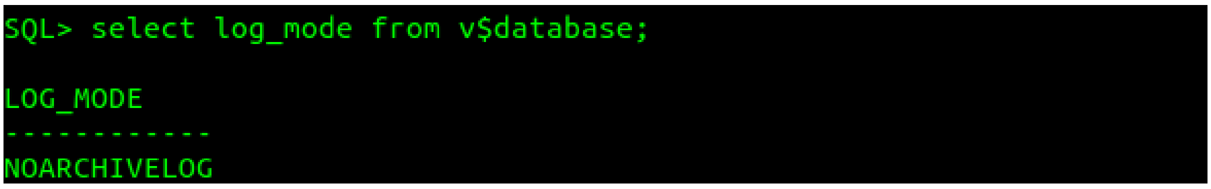


```
SQL> archive log list
Database log mode              No Archive Mode
Automatic archival             Disabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     9
Current log sequence           10
```

Figure 81; Archive log list.

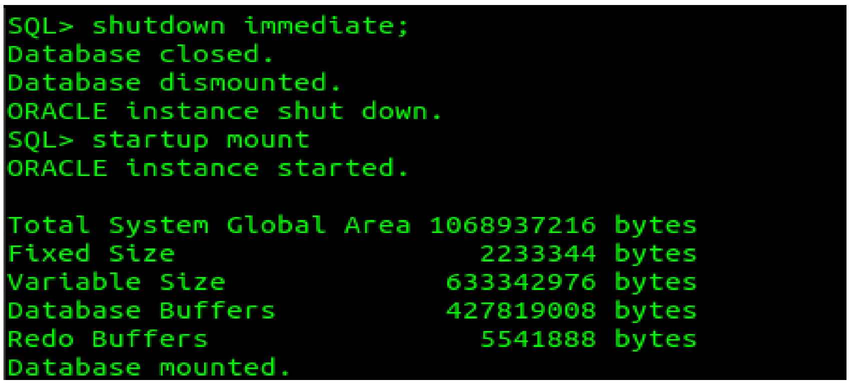
2.7.1 Enabling Archive Mode

To enable this feature first we check to see what mode is enable by using a select statement for log mode from the database as seen in figure 82. We then perform a shut down and a start-up as shown in figure 83 below.



```
SQL> select log_mode from v$database;

LOG_MODE
------------
NOARCHIVELOG
```

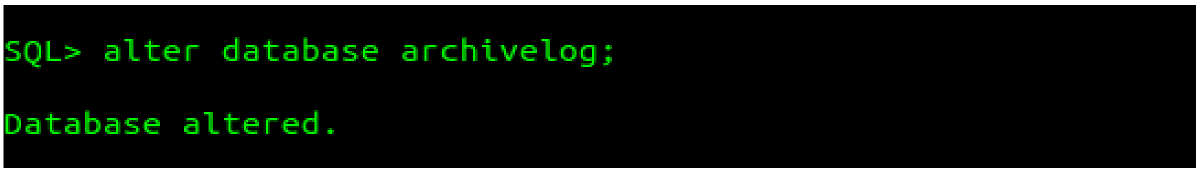
Figure 82: Checking log_mode configuration.



```
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2233344 bytes
Variable Size             633342976 bytes
Database Buffers          427819008 bytes
Redo Buffers                5541888 bytes
Database mounted.
```

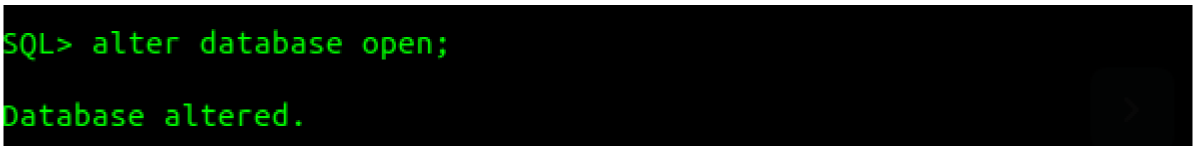
Figure 83: Performing a reboot.

Next the alter database command is used to activate archivelog and then a database open command as referred to in figure 85 and 85.



```
SQL> alter database archivelog;

Database altered.
```

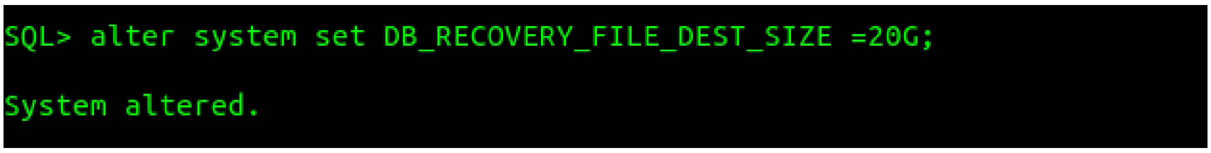
Figure 84: Altering archive log.



```
SQL> alter database open;

Database altered.
```

Figure 85: Checking database is open.

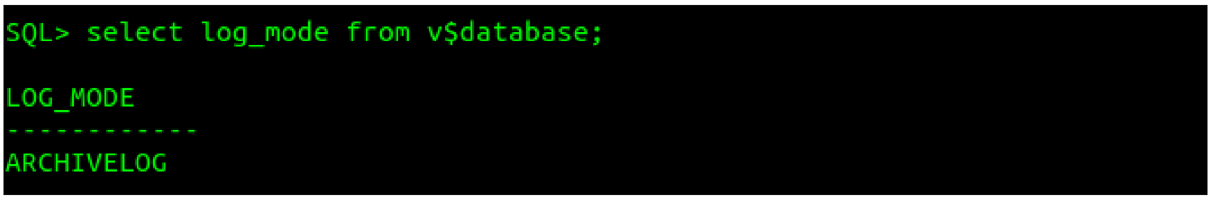
To give extra space for the redo log file the following command was used to increase the size to 20GB.



```
SQL> alter system set DB_RECOVERY_FILE_DEST_SIZE =20G;

System altered.
```

Figure 86: Altering database recovery file destination size.

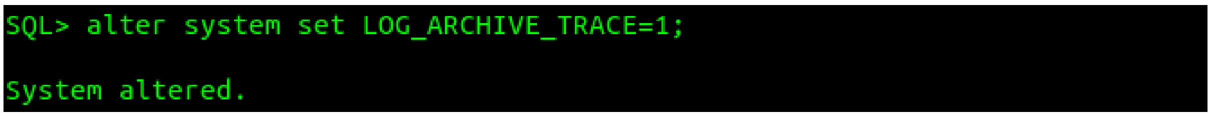
To confirm the change, I issued the command select log mode from the database a referred to in figure 87.



```
SQL> select log_mode from v$database;

LOG_MODE
------------
ARCHIVELOG
```

Figure 87: Confirming alteration of archive log.

If you refer to figure 88, I am altering the log archive trace values. Users can control what is outputted using this command (Oracle, 2019). Below is a list of values that I retrieve from the Oracle help center that the trace can be altered to. The number in front is there identifier.



```
SQL> alter system set LOG_ARCHIVE_TRACE=1;

System altered.
```

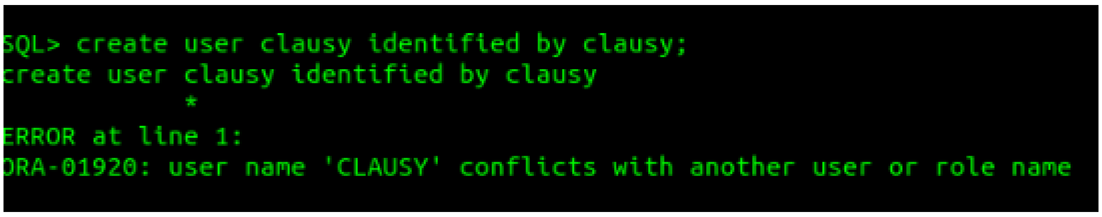
Figure 88: Altering log archive trace.

- 0: Disable archivelog tracing (this is the default)
- 1: Track archival of redo log file
- 2: Track archival status of each archivelog destination
- 4: Track archival operational phase
- 8: Track archivelog destination activity
- 16: Track detailed archivelog destination activity
- 32: Track archivelog destination parameter modifications
- 64: Track ARC*n* process state activity
- 128: Track FAL (fetch archived log) server related activities
- 256: Track RFS Logical Client
- 512: Track LGWR redo shipping network activity
- 1024: Track RFS Physical Client
- 2048: Track RFS/ARCn Ping Heartbeat
- 4096: Track Real Time Apply
- 8192: Track Redo Apply (Media Recovery or Physical Standby)

(Oracle, 2019)

2.8 Password Files

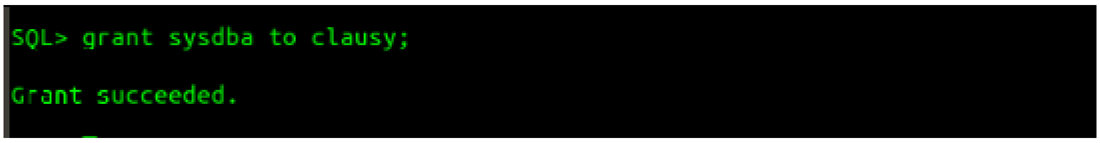
This is an optional file that give access for remote access by system DBA's or admins to gain access to the database. During is install Oracle11g during the configuration process I was prompted for a password that the SYSTEM and SYS accounts would use upon initialization of the database. With these accounts I can create group or user that may access that database. During the testing phase for the install I created a user called clausy and grant them some privileges. If you refer to figure 89, I tried to create that used again and received an error because the user is already found in the password file.



```
SQL> create user clausy identified by clausy;
create user clausy identified by clausy
            *
ERROR at line 1:
ORA-01920: user name 'CLAUSY' conflicts with another user or role name
```

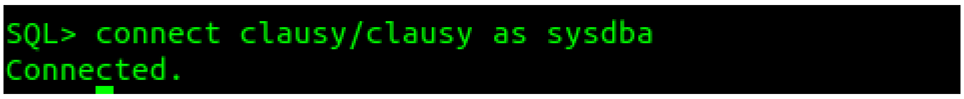
Figure 89: Creating user that already exists.

Show in figure 90 I granted this user sysdba rights this gives the users a large amount of privileges such as creating other users. As you can see below in figure 91 the connection a sysdba was successful.



```
SQL> grant sysdba to clausy;

Grant succeeded.
```

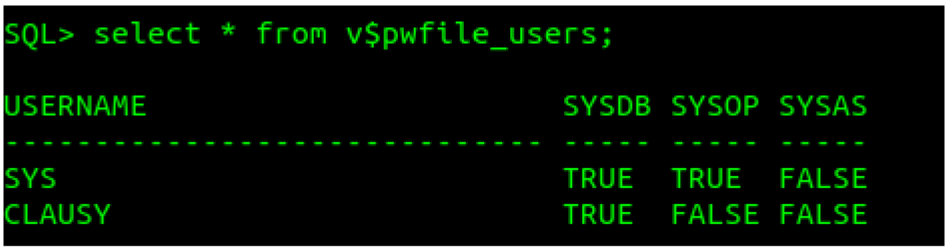
Figure 90: Granting dba to user clausy.



```
SQL> connect clausy/clausy as sysdba
Connected.
```

Figure 91: Connecting as sysdba.

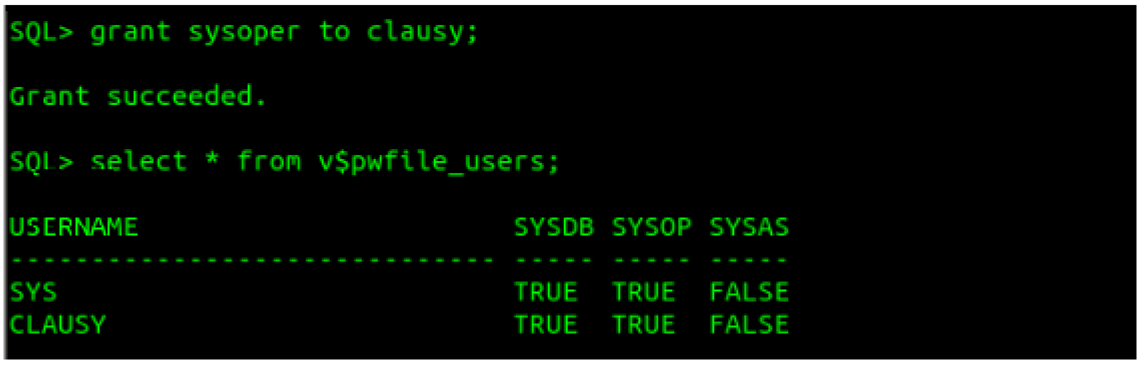
Users may view the password file using the select statement shown below in figure 92. As you can see user clausy only has limited access.



```
SQL> select * from v$pwfile_users;

USERNAME                       SYSDB SYSOP SYSAS
------------------------------ ----- ----- -----
SYS                            TRUE  TRUE  FALSE
CLAUSY                         TRUE  FALSE FALSE
```

Figure 92; Viewing password file users.

If you refer to figure 93 below, I granted system operation access to user clausy. The system operators allow users to perform basic operational task, but they can’t look at user data. SYSDBA and SYSOP can access database instances even when the database is not open.

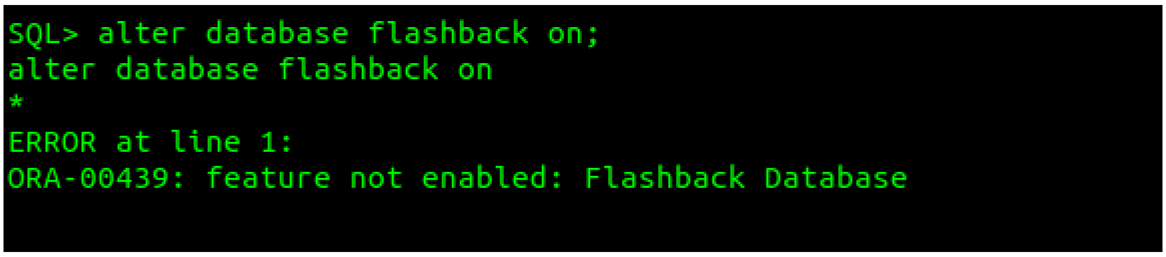


```
SQL> grant sysoper to clausy;

Grant succeeded.

SQL> select * from v$pwfile_users;

USERNAME                       SYSDB SYSOP SYSAS
------------------------------ ----- ----- -----
SYS                            TRUE  TRUE  FALSE
CLAUSY                         TRUE  TRUE  FALSE
```

Figure 93: Granting and confirming system operator for clausy.

2.9 Flashback Log Files

Flashback log files will contain images of before modifications where made to the database blocks. The flash back log files can be used to return the database to the way it was before the modifications were made. The Flashback Database command was introduced to speed up the slow process of point in time database recovery. If you refer to figure 94 you will notice that this feature is not available for the Express version of Oracle.



```
SQL> alter database flashback on;
alter database flashback on
*
ERROR at line 1:
ORA-00439: feature not enabled: Flashback Database
```

Figure 94: Flashback log file.

.

3. Memory Structures

Memory structures and defined by three major areas:

- System Global Area (SGA).
- Process Global Area (PGA).
- User Global Area (UGA).

These three structures can be used is two different ways. They can be used in manual mode. This is where the DBA is in charge of all memory sizes, from the memory for sorting data to the memory needed for the buffer cache for cached data blocks. The other mode is automatic. If you are running versions before Oracle11g DBA need to set the size of the SGA and PGA memory regions. Any version after Oracle11g the DBA's only need to set the memory target value and the database will decide how to size all the different regions.

3.1 The System Global Area (SGA)

All Oracle instances will have a memory structure called the system global area. This memory structure can be quite large and is shared by Oracle processes as referred to in figure 95. All oracle processes will access the SGA at some time (Kyte, 2005). It will differ in size and may be a few megabytes to gigabytes depending on the size of the system using it. The SGA will keep the data and contain the control information for an Oracle instance. If many users are connected to an instance at the same time than the data in the SGA will be shared among users (Center, 2019).

If there is many users using the system, then Oracle will have many processes and the processes can read the information from inside the SGA and there will be many more processes doing the writing to the database instances SGA. Some of the SGA will contain information about the state of the database and the instance. The background processes will use this information and it is called the fixed SGA. When an instance is started Oracle will automatically give the SGA some memory when the instance is shut down the host operation system will take back the allocated memory. (Center, 2019). If you refer to figure 95 you will see the SGA in the middle it is being accessed by the PGA through the server processes.

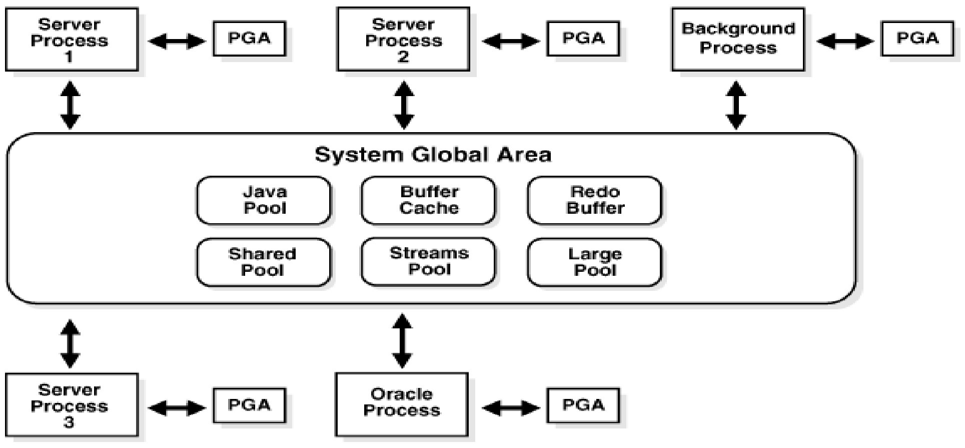
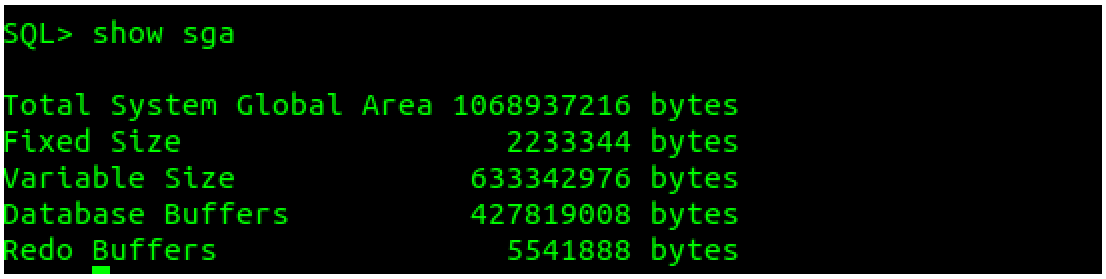


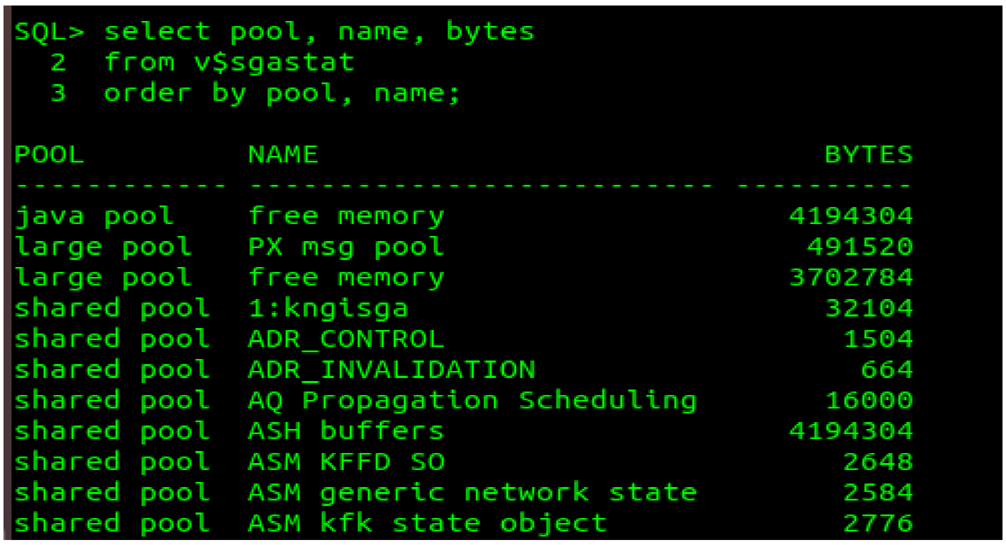
Figure 95: SGA and interaction with Oracle processes (Center, 2019).

Users can view the SGA from SQLPlus with the simple command show SGA as see in figure 96. This view only shows a few different values such as total SGA . If users want a more detailed view, they can use the command shown in figure 97.



```
SQL> show sga

Total System Global Area 1068937216 bytes
Fixed Size                  2233344 bytes
Variable Size             633342976 bytes
Database Buffers          427819008 bytes
Redo Buffers                5541888 bytes
```

Figure 96: Showing SGA sizes.



```
SQL> select pool, name, bytes
  2  from v$sgastat
  3  order by pool, name;

POOL         NAME                            BYTES
------------ -------------------------- ----------
java pool    free memory                   4194304
large pool   PX msg pool                    491520
large pool   free memory                   3702784
shared pool  1:kngisga                       32104
shared pool  ADR_CONTROL                      1504
shared pool  ADR_INVALIDATION                  664
shared pool  AQ Propagation Scheduling       16000
shared pool  ASH buffers                   4194304
shared pool  ASM KFFD SO                      2648
shared pool  ASM generic network state        2584
shared pool  ASM kfk state object             2776
```

Figure 97: Detailed view of SGA pools.

3.1.1 SGA Pools

As shown above in figure 95 the SGA is broken into many pools.

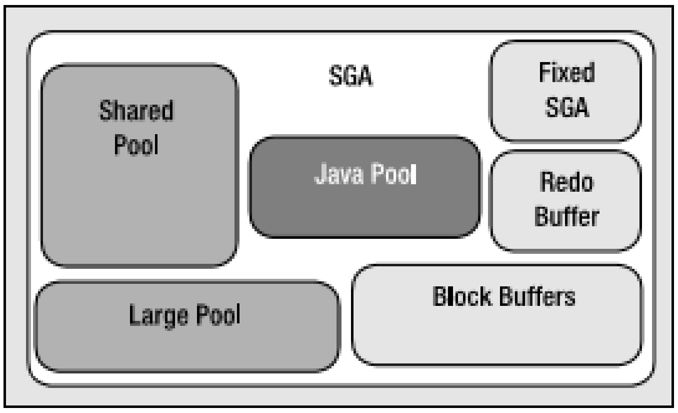


Figure 98: SGA pools.

- **Java Pool:** this is the memory that is giving to the java virtual machine when the database is running. The memory size won't change and in previous version the size could be changed while the database was running.
- **Large Pool:** this is used during connections between client and server for session memory. It may also be used when SQL statements are running in parallel for the message buffers. May also be used by RMAN backup for disk I/O buffers.

- **Shared Pool:** this contains data such as shared cursors, state objects, dictionary caches, and stored procedures. It is one of the most important pieces of memory when considering for scalability and performance. Make it to small and it may appear as if the system is hanging. And make it to big will have the same effect.
- **Streams Pool:** is a data sharing tool used only by Oracle that contains a pool of memory.
- **Fixed SGA:** this component can differ in size from system to system. It contains a "set of variables the points to other components and variables that contain values for different parameters" (Center, 2019).

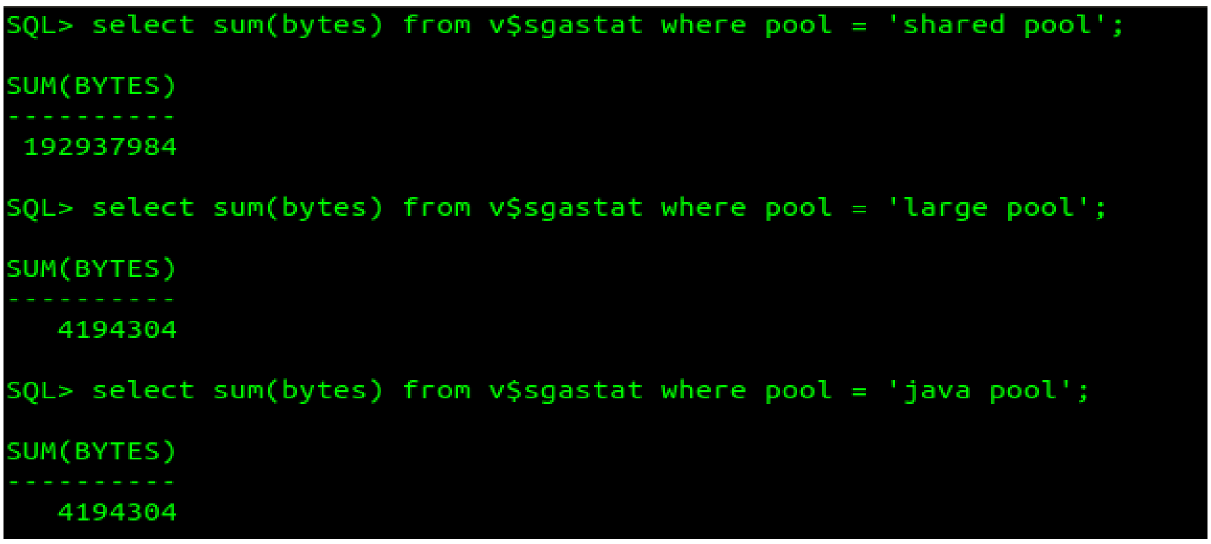
The **redo buffer** log was explained in the previous section of the report.

Block Buffer Cache: we spoke about blocks in the previous section under data files and how they might represent a table row in a database. The block buffer cache is where the blocks is stored when read in and out of the data files. This is an important area of the SGA because if it is to small then queries may take an age and make it too big and there won't be enough room for the creation of the PGA which in turn will starve the other processes. The block buffer cache may be huge compared to the other components in the SGA pools. The SGA has three places where it may store cached blocks:

- **Default pool:** usually the location of the segment blocks.
- **Keep pool:** this is used as an alternative to the default cache. If there are blocks that are queried very often, they may be assigned here.
- **Recycle pool:** where large segments may be assigned when they are accessed at random times.

(Kyte, 2005)

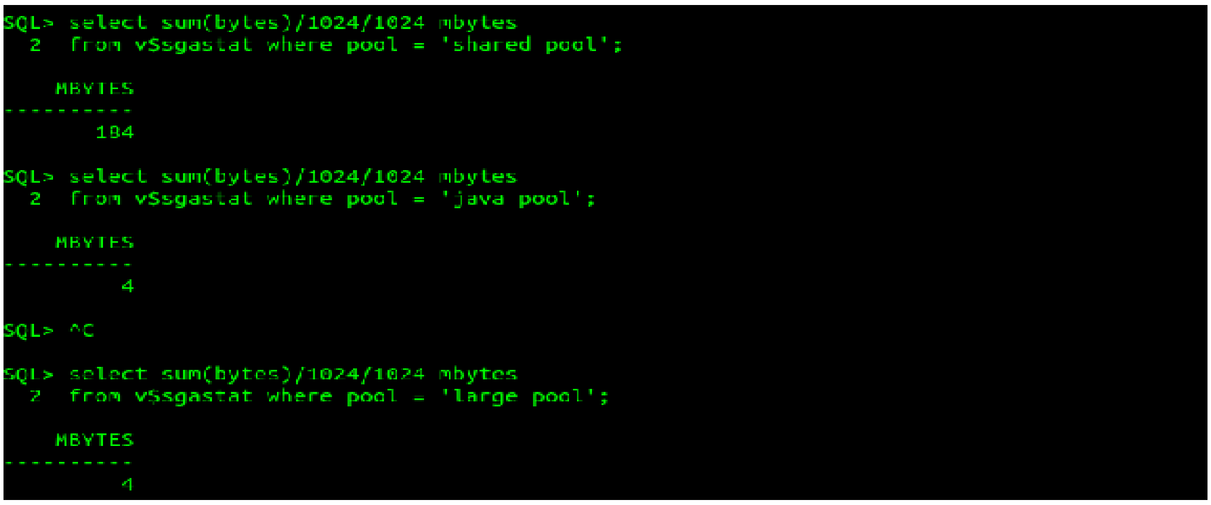
If you refer to figure 99 you will see a select statement from the V$SGASTAT. The statement is returning values from the various pool's sizes in bytes.



```
SQL> select sum(bytes) from v$sgastat where pool = 'shared pool';

SUM(BYTES)
----------
 192937984

SQL> select sum(bytes) from v$sgastat where pool = 'large pool';

SUM(BYTES)
----------
   4194304

SQL> select sum(bytes) from v$sgastat where pool = 'java pool';

SUM(BYTES)
----------
   4194304
```

Figure 99: Show each pool value.

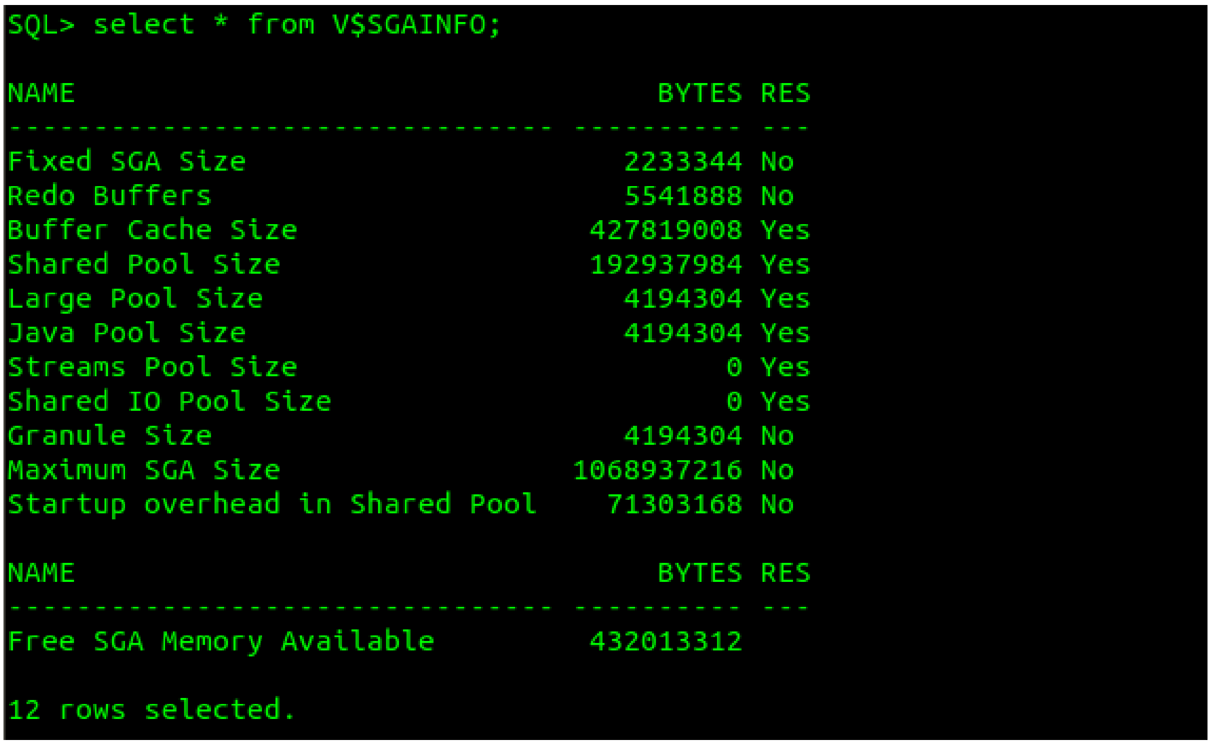
In figure 99 the command is calculation the number of bytes for each pool. As you will notice the number a quite large because its show in bytes. If you refer to figure 100 below the following command will display the total number in megabytes by dividing 1024 which is one megabyte into the total sum of bytes. This round the number to a more readable figure for the users.



```
SQL> select sum(bytes)/1024/1024 mbytes
  2  from v$sgastat where pool = 'shared pool';

    MBYTES
----------
       184

SQL> select sum(bytes)/1024/1024 mbytes
  2  from v$sgastat where pool = 'java pool';

    MBYTES
----------
         4

SQL> ^C

SQL> select sum(bytes)/1024/1024 mbytes
  2  from v$sgastat where pool = 'large pool';

    MBYTES
----------
         4
```

Figure 100: Show each pool in megabytes.

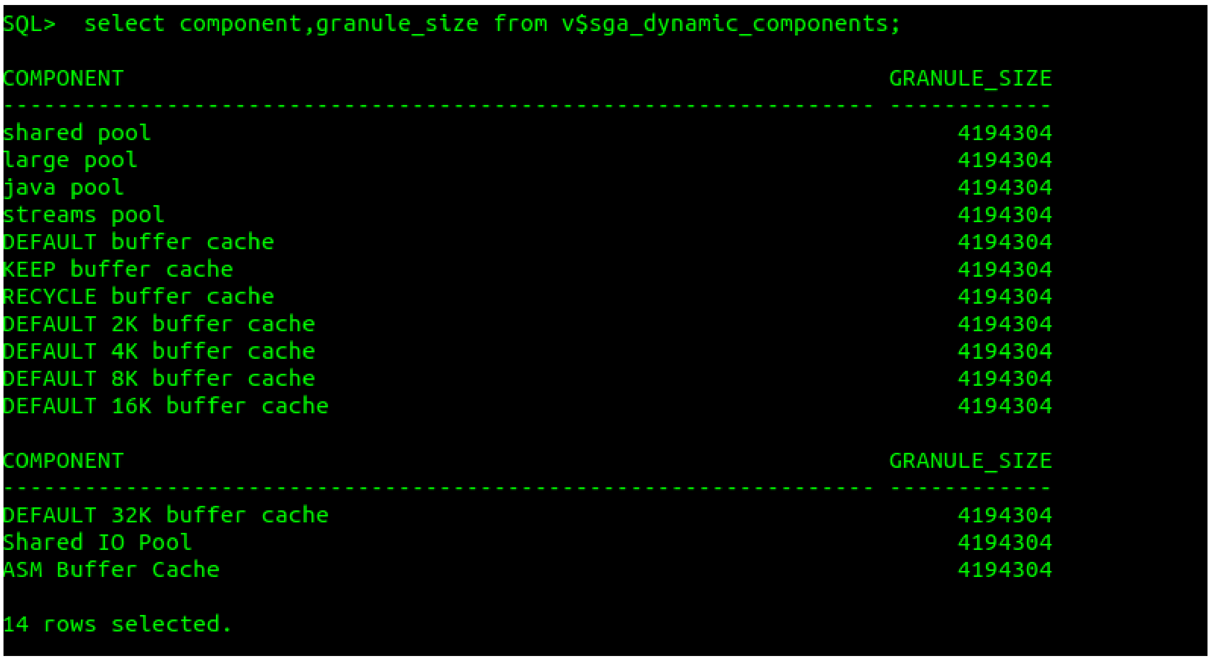
Users can gain more information by using a select all from V$SGAINFO. This will return the name, bytes whether each pool may be resized. If you notice the fixed SGA can't be resized. This is because it is compiled into Oracle binary at the time of installation and why it has the name fixed (Kyte, 2005).



```
SQL> select * from V$SGAINFO;

NAME                                  BYTES RES
-------------------------------- ---------- ---
Fixed SGA Size                      2233344 No
Redo Buffers                        5541888 No
Buffer Cache Size                 427819008 Yes
Shared Pool Size                  192937984 Yes
Large Pool Size                     4194304 Yes
Java Pool Size                      4194304 Yes
Streams Pool Size                         0 Yes
Shared IO Pool Size                       0 Yes
Granule Size                        4194304 No
Maximum SGA Size                 1068937216 No
Startup overhead in Shared Pool    71303168 No

NAME                                  BYTES RES
-------------------------------- ---------- ---
Free SGA Memory Available         432013312

12 rows selected.
```

Figure 101: Showing SGA information.

Shown below in figure 102 the select statement in retuning the component name and its granule size. The granule size may differ is size. This is memory is allocated to different pools in the SGA and they are called granules. One granule may be 4, 8, or 16MB. If the pool is bigger than the nearest granule size, then Oracle will automatically assign the next granule size the is available. For example, if the large java pool was 9MB in size and the granule was 8MB then Oracle would allocate 16MB for the large pool (Kyte, 2005).



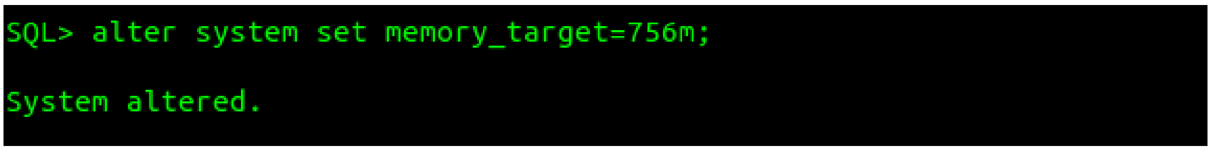
```
SQL>  select component,granule_size from v$sga_dynamic_components;

COMPONENT                                                        GRANULE_SIZE
---------------------------------------------------------------- ------------
shared pool                                                           4194304
large pool                                                            4194304
java pool                                                             4194304
streams pool                                                          4194304
DEFAULT buffer cache                                                  4194304
KEEP buffer cache                                                     4194304
RECYCLE buffer cache                                                  4194304
DEFAULT 2K buffer cache                                               4194304
DEFAULT 4K buffer cache                                               4194304
DEFAULT 8K buffer cache                                               4194304
DEFAULT 16K buffer cache                                              4194304

COMPONENT                                                        GRANULE_SIZE
---------------------------------------------------------------- ------------
DEFAULT 32K buffer cache                                              4194304
Shared IO Pool                                                        4194304
ASM Buffer Cache                                                      4194304

14 rows selected.
```

Figure 102: Showing each component & their size.

3.1.2 SGA Memory Management Mode

There are two ways of managing memory in Oracle. Automictic and manual. Manual is where the user must set all the parameters for each SGA pool. From 11g onwards users may leg Oracle managed the Memory, all the users may need to set is the memory target, and this will be the memory for SGA and PGA together this is automatic mode. Before Oracle11g users needed to set both SGA and PGA targets and the database would allocate the memory based on the sizes configured.

If you refer to figure 103, I set the memory target to 756 megabytes.



```
SQL> alter system set memory_target=756m;

System altered.
```

Figure 103: Altering memory target.

3.2 Process Global Area (PGA)

PGA is a "private region that contains the data and control information for a single server process" (Oracle, 2019). The memory can't be accessed by another process or thread in that region as shown in figure 104. The Oracle database will do the read and writes of the data from the PGA on behalf of the sever process. PGA will never be found in the SGA and will always be allocated by the process. If you refer to figure 104 again you will see that the PGA is allocated by two separate server processes. You will also notice the session memory, this is the interaction between the UGA and the PGA. (Kyte, 2005) states the location of the UGA is dependent on how a client connects to Oracle. If users connect via a shared server then the UGA will be in a memory structure that all the shared servers have access to. If the connection is via a dedicated sever then the UGA will be present within the PGA as is the case shown below in figure 104.

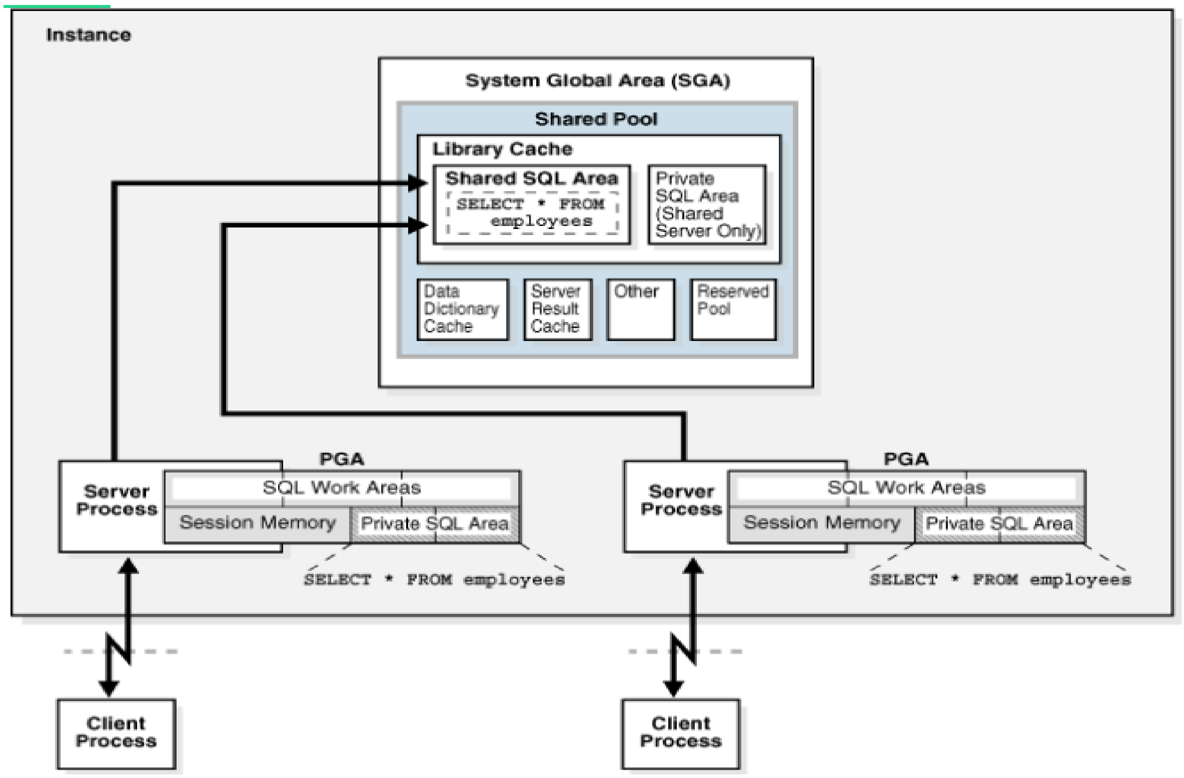
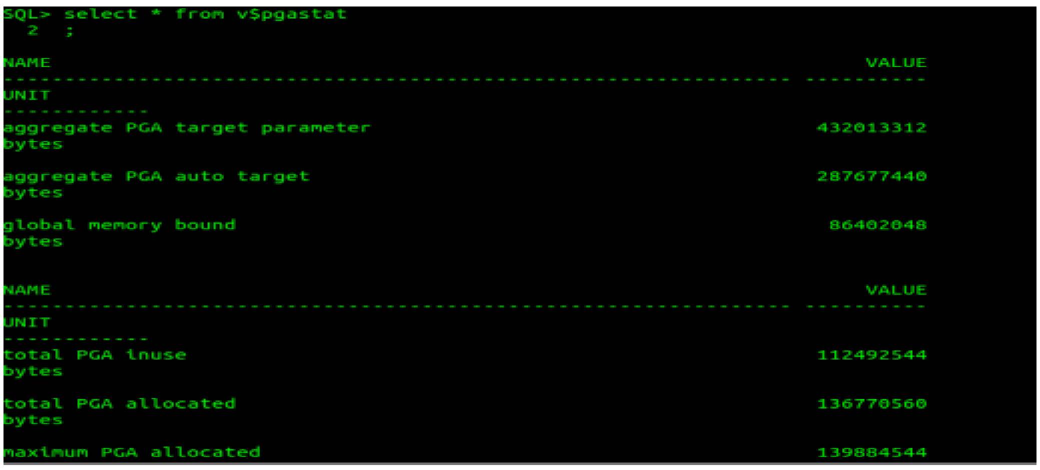


Figure 104: PGA process (Oracle, 2019).

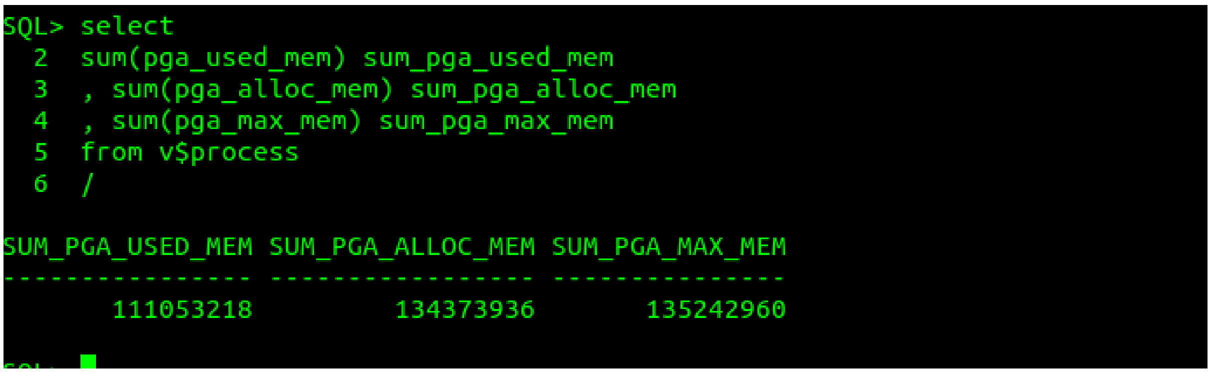
If you refer to figure 105 you will see a select statement, that is returning information about PGA using v$pgastat command.



```
SQL> select * from v$pgastat
  2  ;

NAME                                                                  VALUE
-------------------------------------------------------------------- ----------
UNIT
------------
aggregate PGA target parameter                                         432013312
bytes

aggregate PGA auto target                                              287677440
bytes

global memory bound                                                     86402048
bytes


NAME                                                                  VALUE
-------------------------------------------------------------------- ----------
UNIT
------------
total PGA inuse                                                        112492544
bytes

total PGA allocated                                                    136770560
bytes

maximum PGA allocated                                                  139884544
```

Figure 105: show PGA stats.

If you refer to figure 106 the select statement looks a little more complicated but all it is doing is adding up the values of the PGA usage per process.

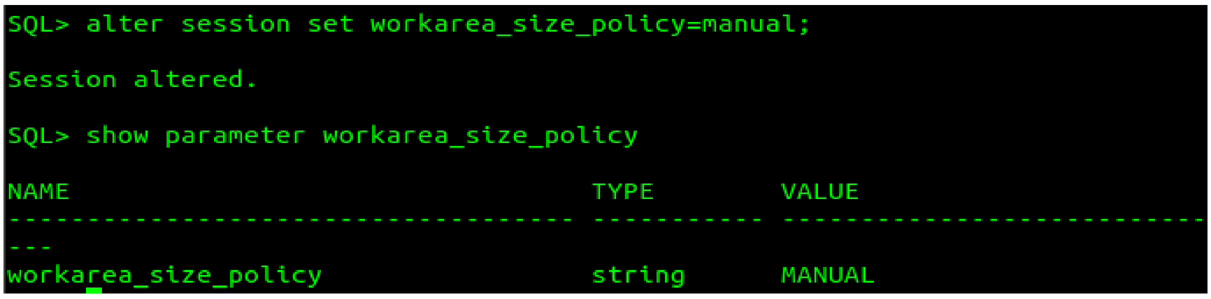


```
SQL> select
  2  sum(pga_used_mem) sum_pga_used_mem
  3  , sum(pga_alloc_mem) sum_pga_alloc_mem
  4  , sum(pga_max_mem) sum_pga_max_mem
  5  from v$process
  6  /

SUM_PGA_USED_MEM SUM_PGA_ALLOC_MEM SUM_PGA_MAX_MEM
---------------- ----------------- ---------------
       111053218         134373936       135242960
```

Figure 106: Showing the sum of all PGA usage per process.

3.2.1 Manual

As you seen with SGA there is two way to handle PGA memory manual and automatic. In manual the administrator in responsible for handling memory allowance for particular processes. If you refer to figure 107 below you will see a parameter of work area size policy. This is a "session and system level parameter" and is and is used to enable and disable PGA memory management with the "default being auto" (Oracle, 2019).

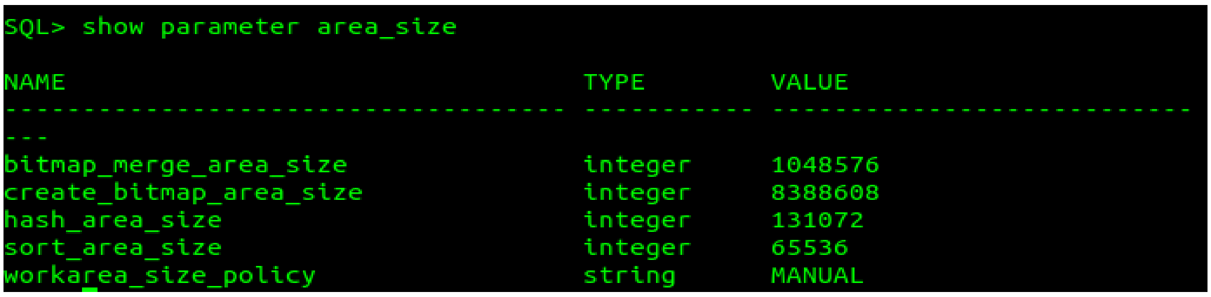


```
SQL> alter session set workarea_size_policy=manual;

Session altered.

SQL> show parameter workarea_size_policy

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
workarea_size_policy                 string      MANUAL
```

Figure 107: Changing work area size policy to manual.

If you refer to figure 108 after a user sets the work area policy to manual, then there is some important parameter that must be set manually. These parameters will have the biggest impact on the size of the PGA, besides the memory that is allocated for the PL/SQL tables (Kyte, 2005).

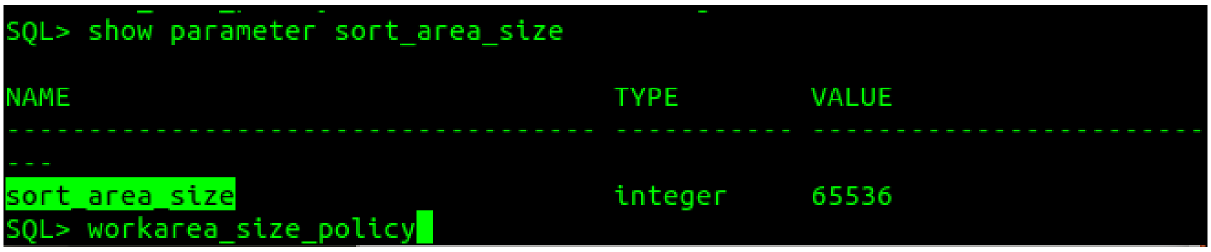
- **Sort_Area_Size:** this parameter is the total amount of RAM that will be used for sorting until swapping to disk.
- **Sort_Area_Retained_Size:** specifies the amount of RAM needed to hold the sorted data after the sort is complete.
- **Hash_Area_Size:** this parameter is used for server process and states how much this process can use to store hash tables in memory.



```
SQL> show parameter area_size

NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------------------------
---
bitmap_merge_area_size               integer     1048576
create_bitmap_area_size              integer     8388608
hash_area_size                       integer     131072
sort_area_size                       integer     65536
workarea_size_policy                 string      MANUAL
```

Figure 108: Showing area size values.

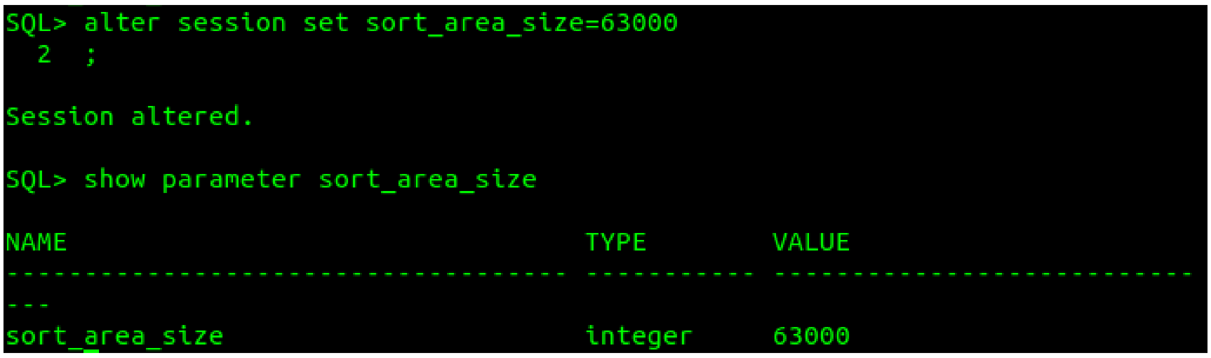
The following command shown in figure 109 is used to identify the values for each of the parameter that where just explained.



```
SQL> show parameter sort_area_size

NAME                                 TYPE        VALUE
------------------------------------ ----------- -------------------------
---
sort_area_size                       integer     65536
SQL> workarea_size_policy
```

Figure 109: Showing sort are size value.

If users need to allocate more memory to one of the parameters, they would use the alter session command as shown in figure 110.



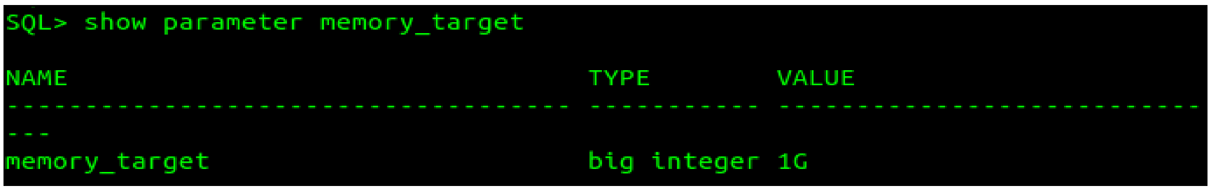
```
SQL> alter session set sort_area_size=63000
  2  ;

Session altered.

SQL> show parameter sort_area_size

NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------------------------
---
sort_area_size                       integer     63000
```

Figure 110: Altering sort area size.

3.2.2 Automatic

From version 9i Release 1 Oracle developed a new way the manage PGA memory. It was a way to avoid setting the parameters seen in the manual memory management such Hash_Area_Size and sort_area_size and addressed a few issues at the same time. There were some issues around ease of use and especially how to set the parameter mentioned above. Many users found it difficult to understand how the parameter works and how much memory to allocate. Manual allocation was another problem. Depending on what users where doing will determine how much memory they need. But with manual they all got the same unless a DBA was manually adjusting for all users which was unlikely.

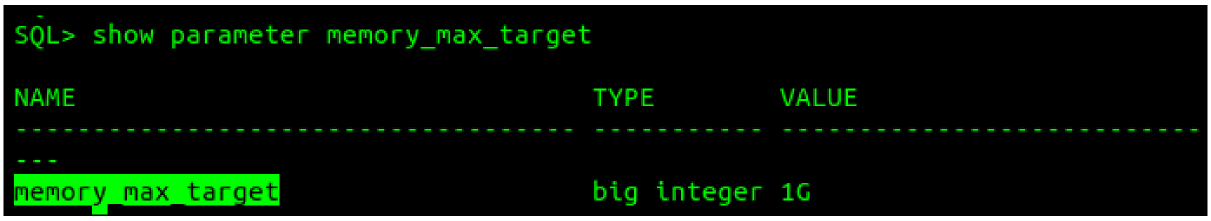
The new way that was developed was automatic PGA memory management. At that time, they needed to set the pga_target but from 11g onward it was just a matter of setting the memory target and Oracle allocate memory to the SGA and PGA. Users can view the memory target by using the command show in figure 111. In this case I had 1 gigabyte of memory that could be allocated.



```
SQL> show parameter memory_target

NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------------------------
---
memory_target                        big integer 1G
```

Figure 111: Showing memory target.

User may also view the maximum memory target by using the show command referred to in figure 112.



```
SQL> show parameter memory_max_target

NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------------------------
---
memory_max_target                    big integer 1G
```

Figure 112: Showing memory max target.

3.3 User Global Area

UGA is the session memory. This is the memory that is used for session variables as referred to in figure 113, these could be variables such as user login information and other bit of information needed for a database session (Oracle, 2019). The UGA would be similar to connections made between user and server on the internet.

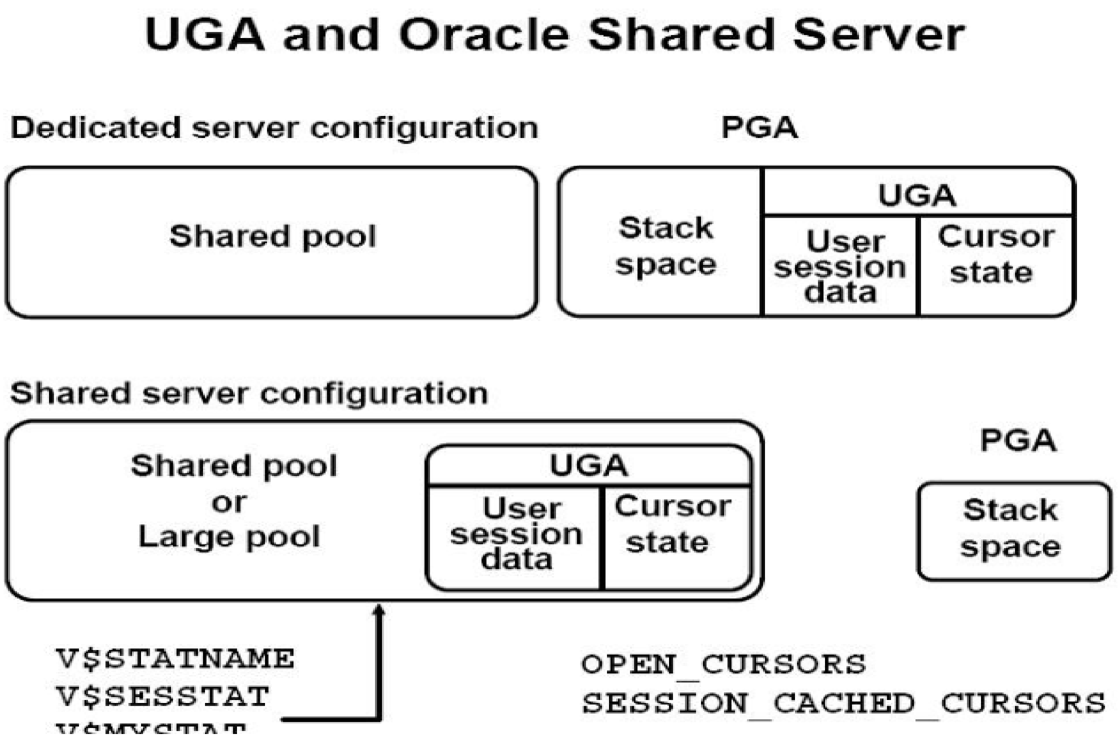
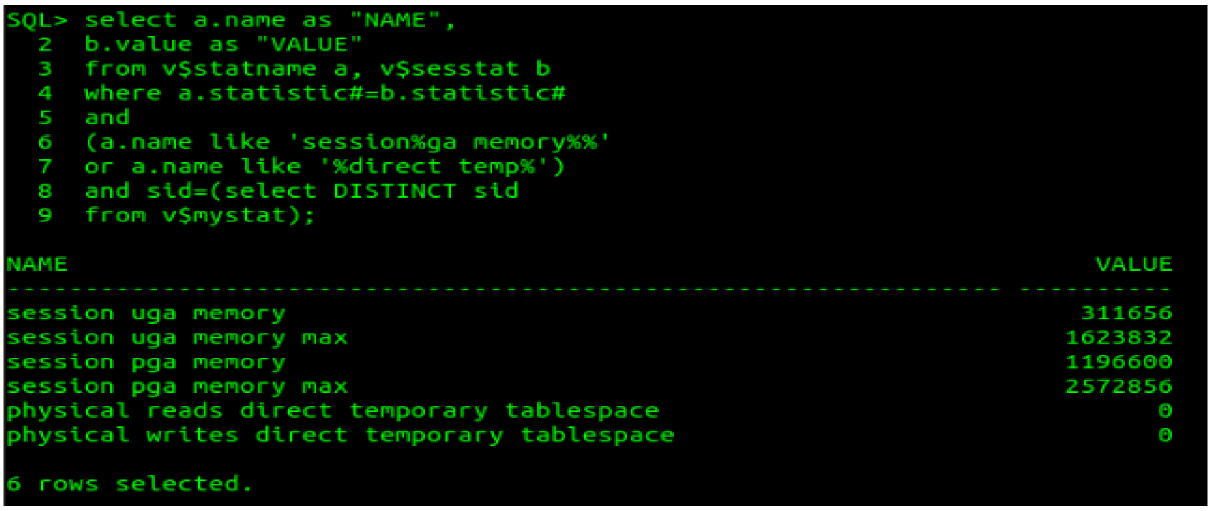


Figure 113: UGA shared/dedicated sever mode.

The UGA has to be available to the database for the life time of the session. As stated in the PGA section and seen in figure 113 the UGA will have different locations depending weather the connection is a shared or dedicated. So, if it's a shared server then the UGA can't be stored in the PGA because the PGA belongs to only one sever process. In this case it stored in the SGA but if it's a dedicated server it stored in the UGA (Oracle, 2019)

The following script seen in figure 114 is used to return the values of the UGA memory and UGA max memory.



```
SQL> select a.name as "NAME",
  2  b.value as "VALUE"
  3  from v$statname a, v$sesstat b
  4  where a.statistic#=b.statistic#
  5  and
  6  (a.name like 'session%ga memory%%'
  7  or a.name like '%direct temp%')
  8  and sid=(select DISTINCT sid
  9  from v$mystat);

NAME                                                                  VALUE
---------------------------------------------------------------- ----------
session uga memory                                                   311656
session uga memory max                                              1623832
session pga memory                                                  1196600
session pga memory max                                              2572856
physical reads direct temporary tablespace                                0
physical writes direct temporary tablespace                               0

6 rows selected.
```

Figure 114: Script to look at UGA values.

4. Processes

Processes in Oracle are broken into three main classes of processes with each process performing a particular task or set of tasks. Each task will have internal memory allocated to it to perform its job.

- Server processes
- Background Processes
- Slave processes

4.1 Server processes

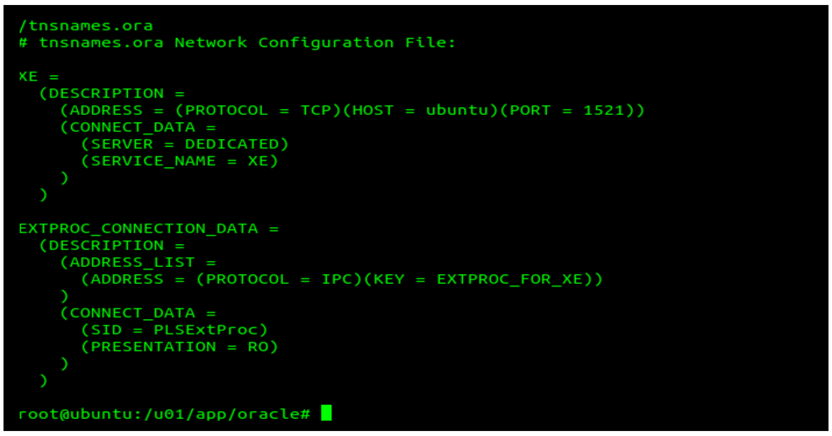
The server processes the do work on behalf of the client session. These are processes that will perform the SQL statements used by our application to send and retrieve data from the Oracle database. As mentions above there are two types of servers:

- Dedicated server.
- Shared server.

The implementation of these servers is transparent to the user as both the dedicated and shared server both perform the same job, but the setup is far different as dedicated sever uses a one to one mapping and the shared server uses a many to one mapping. The dedicated server mode is the recommend configuration by Oracle and will suit non-OLTP environment where there could be long running transactions. The shared server mode is better equipped to handle sort transactions.

As stated above they both perform the same job the dedicated/shared server process parses the query and puts them in the shared pool or else it locates the query already in the shared pool. This process to make a query plan and executes the plan by finding the data in the buffer cache or reading the data from the disk into the buffer cache (Kyte, 2005). These server processes are the worker processes and have a high CPU consumption because they are doing the sorts, joins and sum on the system.

If you refer to figure 115 you will see the output tnsnames.ora file and you will notice that the server type is dedicated.



```
/tnsnames.ora
# tnsnames.ora Network Configuration File:

XE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = ubuntu)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = XE)
    )
  )

EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC_FOR_XE))
    )
    (CONNECT_DATA =
      (SID = PLSExtProc)
      (PRESENTATION = RO)
    )
  )

root@ubuntu:/u01/app/oracle#
```

Figure 115: Network configuration.

4.1.1 Dedicated Server

In the dedicated server mode, each server will map to one client connection. The more connections the more processes will be running on behalf of each connection. If you refer to figure 116 you will see three servers in total each with a direct line from the client. The servers are utilizing the SGA pools for executing the query's coming from the clients. Oracle net is the API that is provided by Oracle in order to communicate with the databases.

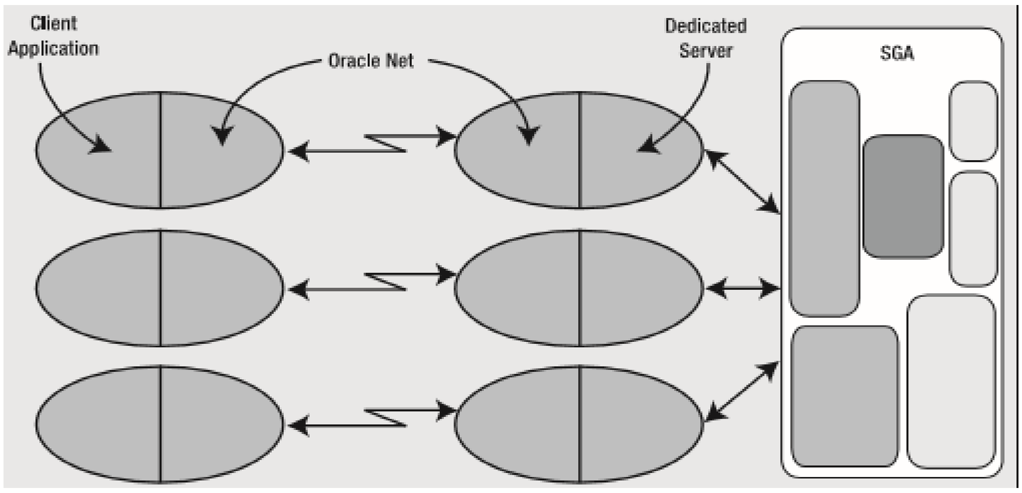
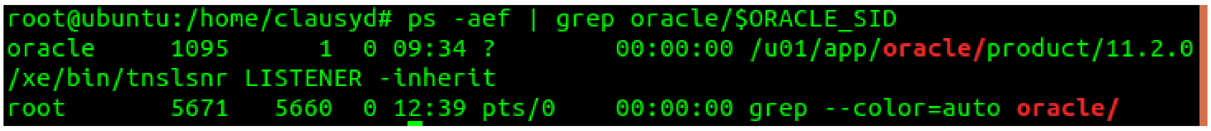


Figure 116: Dedicated server mode (Kyte, 2005).

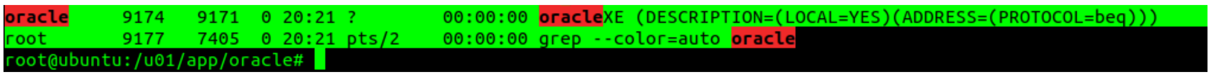
Shown in figure 117 is the command to view the currents processes running on my ubuntu VM using the Linux command grep for searching processes that are linked to Oracle. As you will see there is one Oracle process running this starts automatically when I booth the Ubuntu VM.



```
root@ubuntu:/home/clausyd# ps -aef | grep oracle/$ORACLE_SID
oracle      1095       1  0 09:34 ?        00:00:00 /u01/app/oracle/product/11.2.0
/xe/bin/tnslsnr LISTENER -inherit
root        5671    5660  0 12:39 pts/0    00:00:00 grep --color=auto oracle/
```

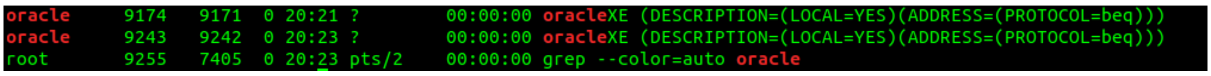
Figure 117: No SQLPlus process running.

Once I started SQLPlus you can see in figure 118 another process running and the description of that process. It is running on my local machine and the protocol is bequeath. This is a protocol that is used for local connections when server and client reside on the same machine. Shown in figure 119 I open a second connection in SQLPlus, and you can see the second process running.



```
oracle     9174   9171  0 20:21 ?        00:00:00 oracleXE (DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))
root       9177   7405  0 20:21 pts/2    00:00:00 grep --color=auto oracle
root@ubuntu:/u01/app/oracle#
```

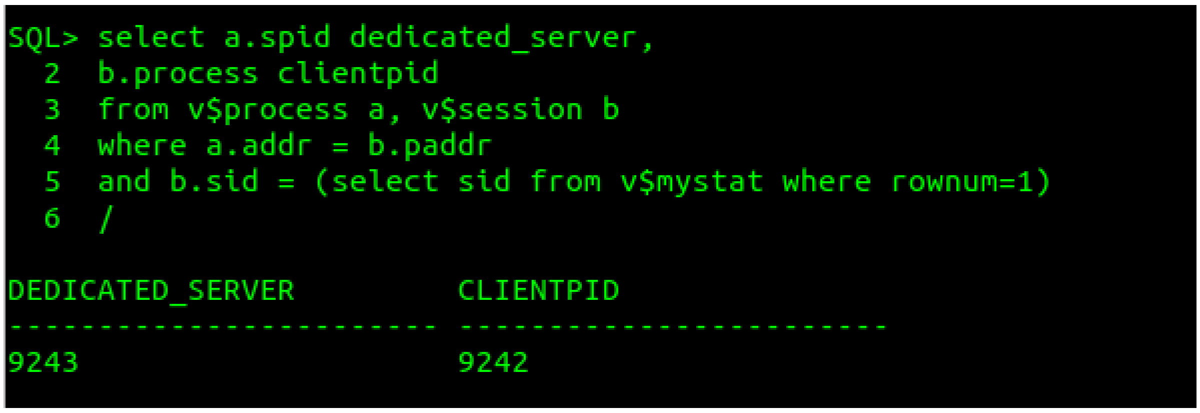
Figure 118: Processes running.



```
oracle     9174   9171  0 20:21 ?        00:00:00 oracleXE (DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))
oracle     9243   9242  0 20:23 ?        00:00:00 oracleXE (DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))
root       9255   7405  0 20:23 pts/2    00:00:00 grep --color=auto oracle
```

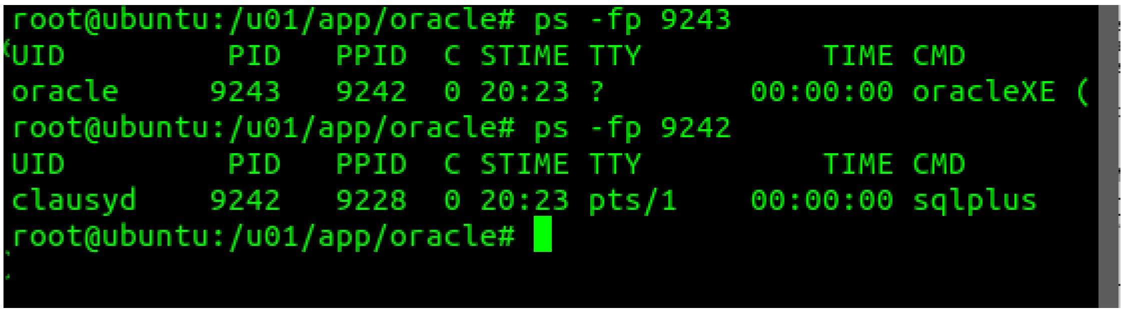
Figure 119: Two SQLPlus processes running.

The script referred to in figure 120 is used to find the process ID that is associated with the dedicated server. When a process is forked is SQLPlus a child process can then be seen in the host machine a seen below in figure 121. This is the parent/child processes the (Kyte, 2005) referred to in the book.



```
SQL> select a.spid dedicated_server,
  2  b.process clientpid
  3  from v$process a, v$session b
  4  where a.addr = b.paddr
  5  and b.sid = (select sid from v$mystat where rownum=1)
  6  /

DEDICATED_SERVER         CLIENTPID
------------------------ ------------------------
9243                     9242
```

Figure 120: Parent and child process in Unix



```
root@ubuntu:/u01/app/oracle# ps -fp 9243
UID          PID    PPID  C STIME TTY          TIME CMD
oracle      9243    9242  0 20:23 ?        00:00:00 oracleXE (
root@ubuntu:/u01/app/oracle# ps -fp 9242
UID          PID    PPID  C STIME TTY          TIME CMD
clausyd     9242    9228  0 20:23 pts/1    00:00:00 sqlplus
root@ubuntu:/u01/app/oracle#
```

Figure 121: Identifying processes through Ubuntu bash.

4.1.2 Shared Server

Shared server connections take instruction from Oracle Net even if client and server are hosted on the same machine, as is the case for me. The Oracle TNS listener is always used when using shared server mode. The listener sits between the client and dispatcher and is used to handle the traffic of to the dispatcher. If you refer to figure 122 the client makes a connection, the TSL listener redirects the traffic to the dispatcher who then places the request in the request queue in the SGA. The first available server process from the pool of shared servers will append the request to the UGA which is the box in with S (highlighted in red below) of the associated session. The shared server will then process the request and place the output in the response queue.

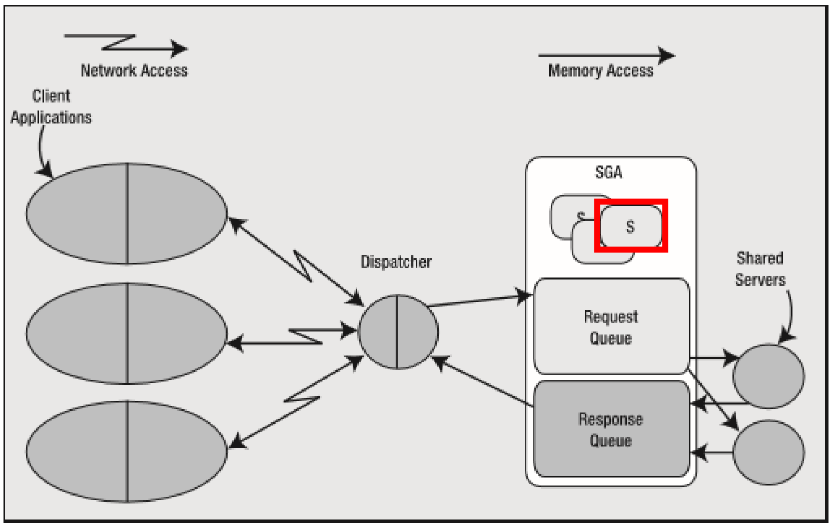
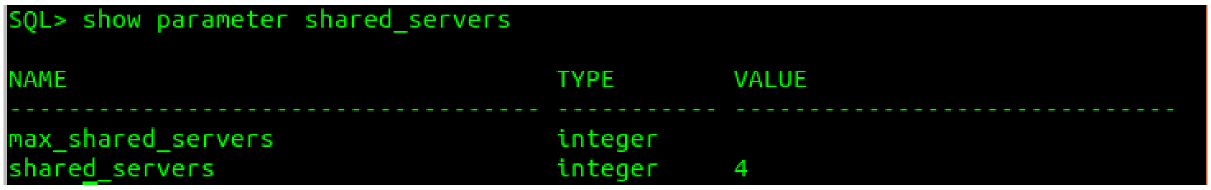


Figure 122: Shared sever (Kyte, 2005).

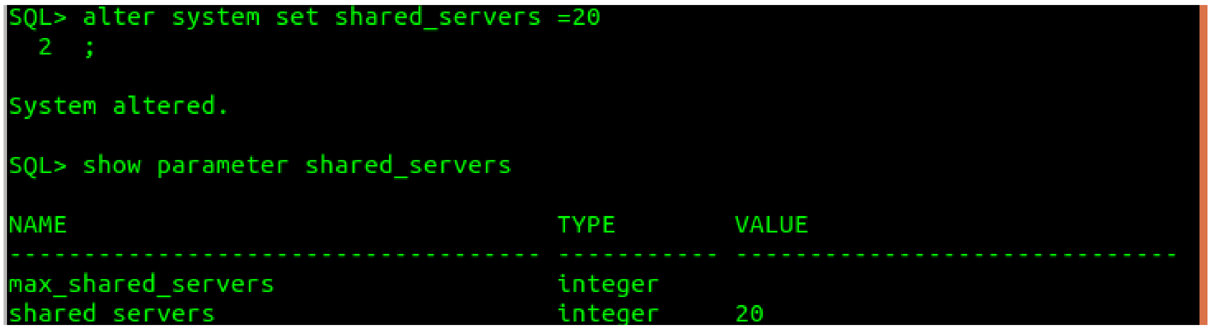
The dispatcher will always be listening to the response queue ready to pass the result back the client. As stated above the client will not no weather the connection is via a shared or dedicated server as they appear the same to the client. There may be many dispatchers in use by an instance, but it is not unusual for one dispatcher to be receiving hundreds of connections from users (Kyte, 2005).

If you refer to figure 123 the show parameter command is being used to show the shared servers, the values are the number of shared servers in use. If you refer to figure 124 below, I used the alter system command to increase the number shared server that can be used. In this case the number was increased to 20.



```
SQL> show parameter shared_servers

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
max_shared_servers                   integer
shared_servers                       integer     4
```

Figure 123: Shared servers.



```
SQL> alter system set shared_servers =20
  2  ;

System altered.

SQL> show parameter shared_servers

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
max_shared_servers                   integer
shared_servers                       integer     20
```

Figure 124: Altering shared severs.

The other important component of the shared server process is the dispatchers. Referred to in figure 125 you will see the properties of the dispatchers. The dispatcher value is passed as a string these values can be changed. If you refer to figure 126, I alter the amount of configured dispatcher to a total of 30. You will notice the value change in figure 127 for the max_diapatchers.

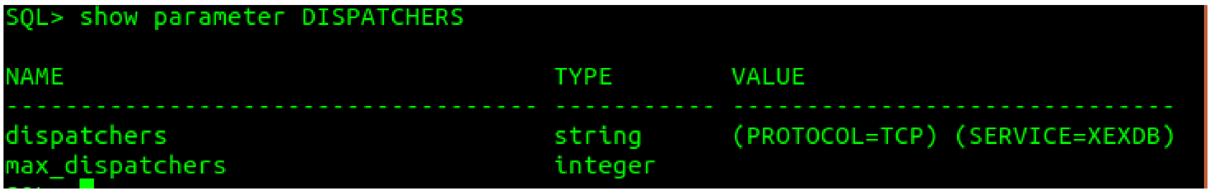

```
SQL> show parameter DISPATCHERS

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
dispatchers                          string      (PROTOCOL=TCP) (SERVICE=XEXDB)
max_dispatchers                      integer
```

Figure 125: Showing parameter DISPATCHERS.

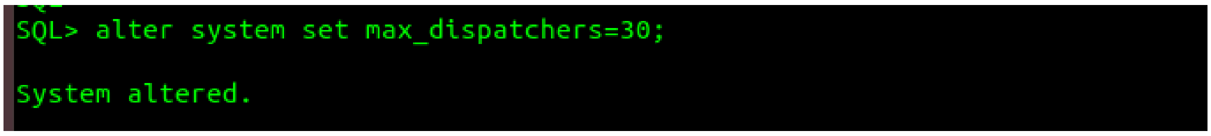

```
SQL> alter system set max_dispatchers=30;

System altered.
```

Figure 126: Altering DISPATCHERS, protocol & dispatcher.

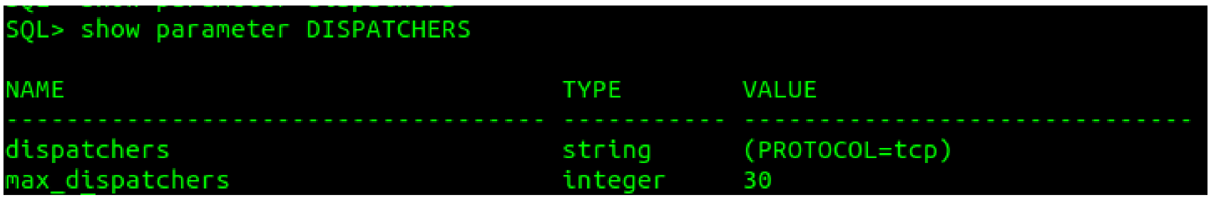

```
SQL> show parameter DISPATCHERS

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
dispatchers                          string      (PROTOCOL=tcp)
max_dispatchers                      integer     30
```

Figure 127: Parameter for the dispatchers.

If you refer to figure 128 users can change the confirmation file to enable shared server mode. Below I am using nano editor on Ubuntu to edit the file this must be done as root or else you will get an error an only root user has permission to edit files found on the system.

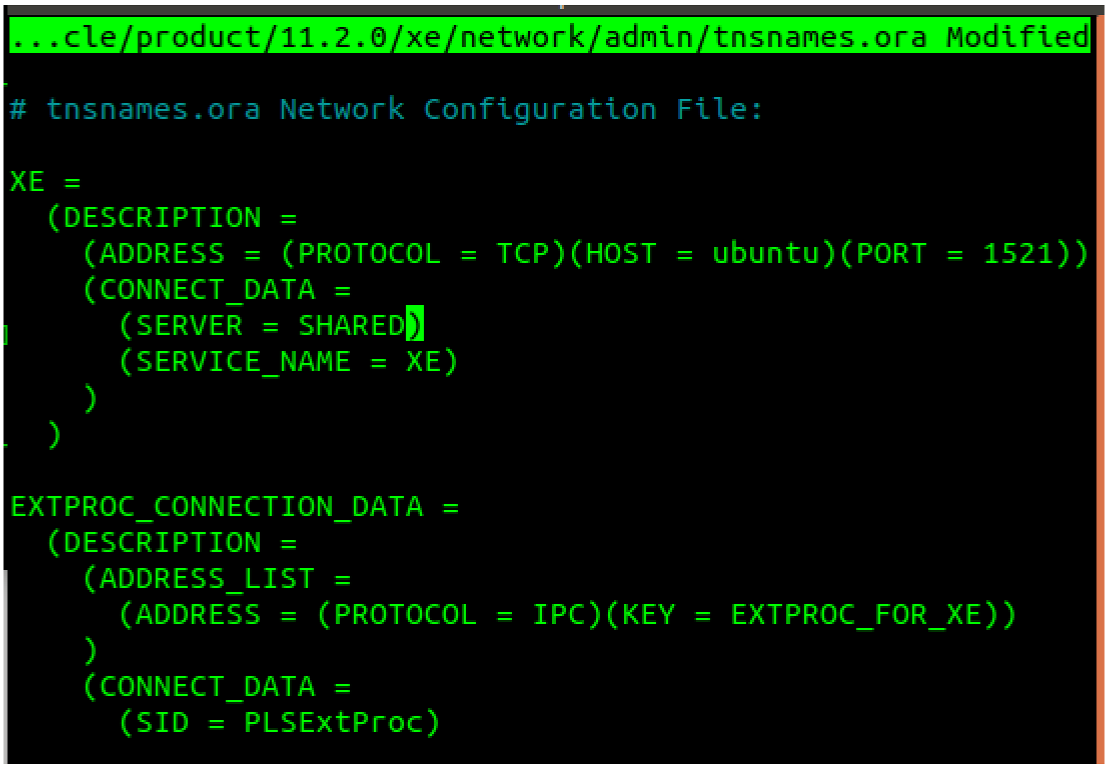

```
...cle/product/11.2.0/xe/network/admin/tnsnames.ora Modified

# tnsnames.ora Network Configuration File:

XE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = ubuntu)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = SHARED)
      (SERVICE_NAME = XE)
    )
  )

EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC_FOR_XE))
    )
    (CONNECT_DATA =
      (SID = PLSExtProc)
```

Figure 128: Altering server mode with nano editor. .

4.2 Background Processes

The background processes are the processes that perform the routine task needed to keep the instance ticking over. We may have one background process maintaining the buffer cache another writing blocks out to the data files and another process writing to the redo log file. All the processes deal with a specific task, but they work as one. What I mean by this is that when is that you have one process writing out to files when this process is finished writing to the particular file is will notify the archive process that it has finished and the archive will start doing it specific job of archiving the file (Kyte, 2005).

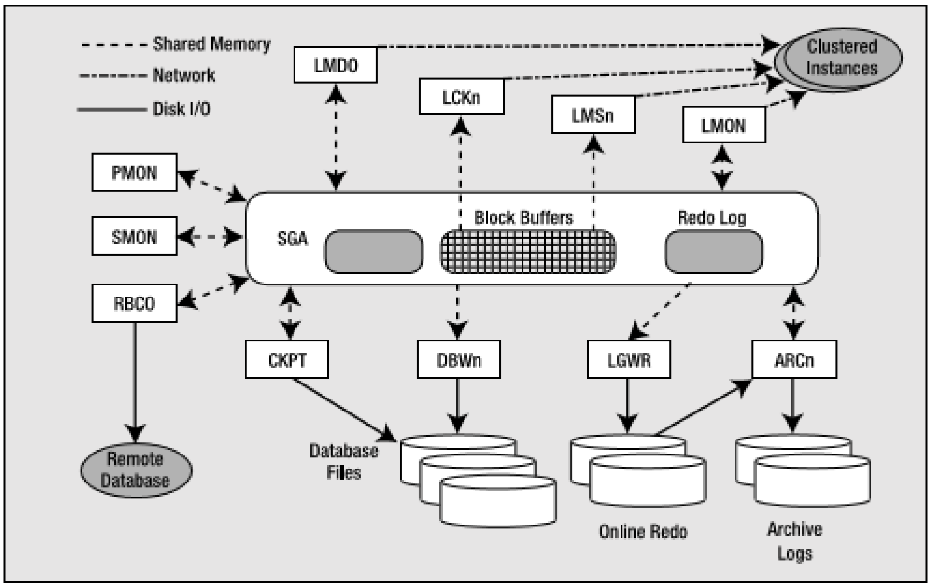


Figure 129:Background processes (Kyte, 2005).

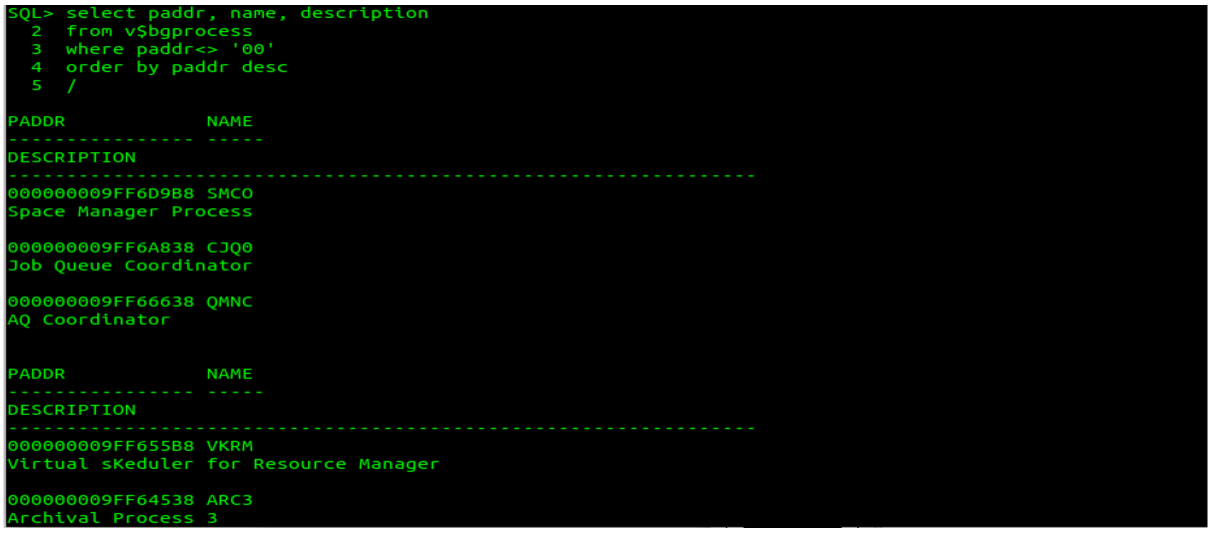
If you refer to figure 124 you will see all the processes in action and how the interact with the different components of the database instance. I will give a brief description of each background process below.

- **(PMON) Process Monitor:** This processes job is to clean up when a connection is unexpectedly terminated. The process will recover by releasing locks, resources perform roll backs and free up SGA resources that have been assigned for the process that failed.
- **(SMON) System Monitor:** This process looks after the job from a system perspective. (Kyte, 2005) referred to it as the garbage cleaner for the database. It cleans temporary space, merges free space and recovers transaction in case of system failure.
- **(RECO) Distributed Database Recovery:** This process has a very specific job, when a system crashes and transactions were fixed in a prepare state for a two-phase commit it will recover the transaction.
- **(CKPT) Checkpoint Process:** "This process is used to assist with the checkpointing process by updating file headers of the data files" (Kyte, 2005).
- **(DBWn) Database Block Writer:** The is the process that write the data from the buffer cache to the data files. This is usually done to free up space in the cache so the buffers can perform the reads of other data (Kyte, 2005).

- **(LGWR) Log Writer:** The job of this process is to write to the redo log file all the contents of the redo log buffer. It does this when commits are issued, or the redo buffer is 1/3 full or has 1mb of buffer data in it or every three second.
- **(ARCn) Archive Process:** This process is used to write redo log file data to the location where the rest of the archived data is stored.
- **(DIAG) Diagnosability Process:** This process was used prior to version 11g. This process is now named (ADR) or advanced diagnostic repository and is used to monitor the health of the system. It will keep information about system failures.

(Kyte, 2005)

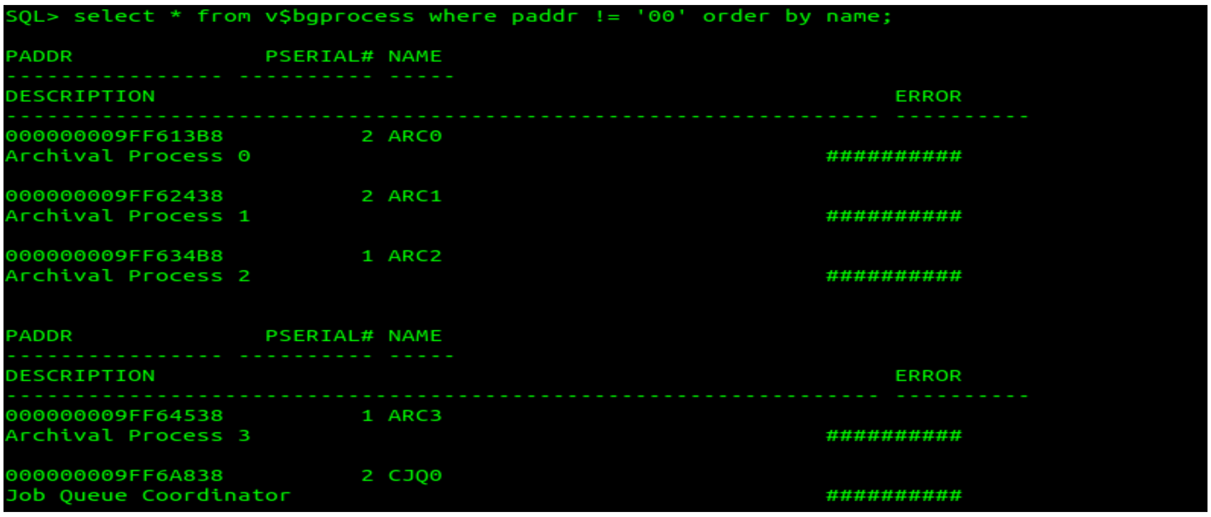
If you refer to figure 130, I am performing a select statement and returning the values for process address, the name of the process and a description of the process.



```
SQL> select paddr, name, description
  2  from v$bgprocess
  3  where paddr<> '00'
  4  order by paddr desc
  5  /

PADDR            NAME
---------------- -----
DESCRIPTION
----------------------------------------------------------------
000000009FF6D9B8 SMCO
Space Manager Process

000000009FF6A838 CJQ0
Job Queue Coordinator

000000009FF66638 QMNC
AQ Coordinator


PADDR            NAME
---------------- -----
DESCRIPTION
----------------------------------------------------------------
000000009FF655B8 VKRM
Virtual sKeduler for Resource Manager

000000009FF64538 ARC3
Archival Process 3
```

Figure 130: Background processes.

Using the select statement as seen in figure 131 we can get more information on the current running process and by ordering by name it makes it far more readable for the human eye.



```
SQL> select * from v$bgprocess where paddr != '00' order by name;

PADDR               PSERIAL# NAME
---------------- ---------- -----
DESCRIPTION                                                       ERROR
---------------------------------------------------------------- ----------
000000009FF613B8          2 ARC0
Archival Process 0                                               ##########

000000009FF62438          2 ARC1
Archival Process 1                                               ##########

000000009FF634B8          1 ARC2
Archival Process 2                                               ##########


PADDR               PSERIAL# NAME
---------------- ---------- -----
DESCRIPTION                                                       ERROR
---------------------------------------------------------------- ----------
000000009FF64538          1 ARC3
Archival Process 3                                               ##########

000000009FF6A838          2 CJQ0
Job Queue Coordinator                                            ##########
```

Figure 131: More detail on background processes.

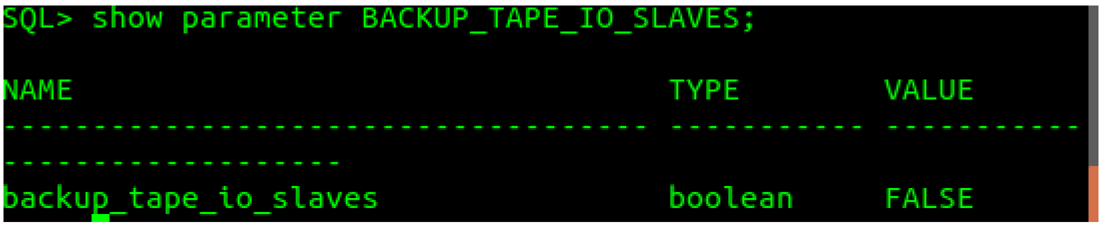
4.3 Slave Processes

Salve process can be broken into two categories I/O salve and parallel query slaves.

4.3.1 I/O Slaves

I/O slave are used to emulate asynchronous input/output for systems of devices that don't support it. With I/O salves they can replicate what operating systems provide for disk drives which can be useful for writing out to tape devices which are much slower then disk drives. I/O slaves can be utilized by the DBWn and LGWR to replicate asynchronous I/O or by RMAN for writing to tape drives.

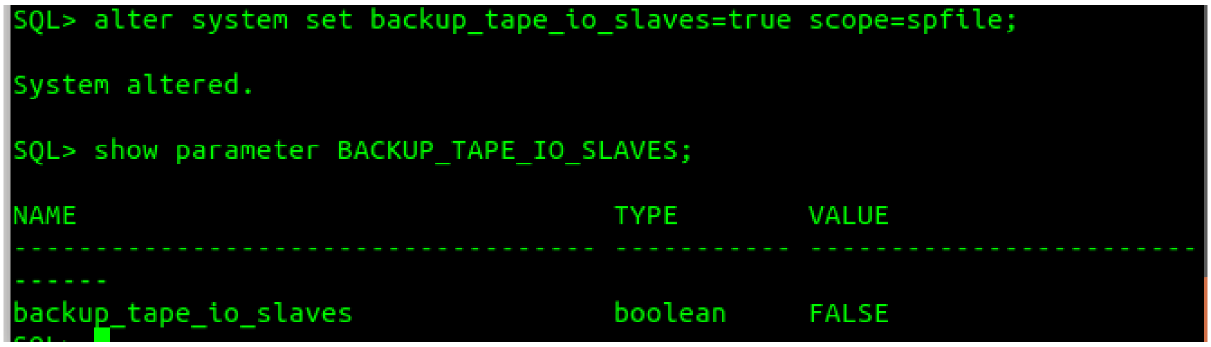
Shown below in figure 132 I am looking as the parameter of the BACKUP_TAPE_IO_SLAVES. This parameter will say whether I/O salves are being used by RMAN to back-up, copy or restore data to a tape. RMAN will start as many slaves that are need for the for writing out to physical devices being used by the instance (Kyte, 2005).



```
SQL> show parameter BACKUP_TAPE_IO_SLAVES;

NAME                                 TYPE        VALUE
------------------------------------ ----------- -----------
--------------------
backup_tape_io_slaves                boolean     FALSE
```

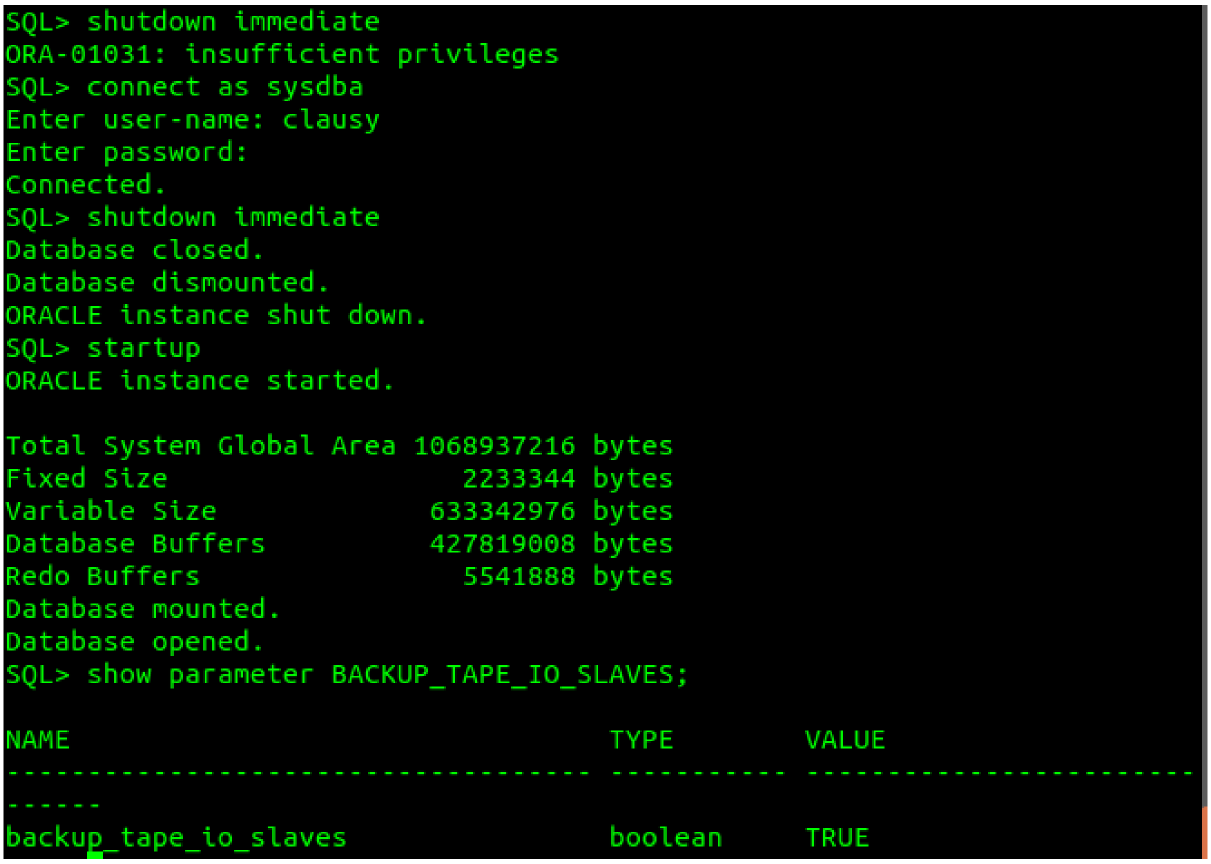
Figure 132: Backup slaves

If you refer to figure 133, I am enabling the BACKUP_TAPE_IO_SLAVES by setting them to true and using the scope of the spfile. When I showed the parameter to confirm the changed it still showed up a false but if you refer to figure 134 after I performed a system reboot the change then took effect.



```
SQL> alter system set backup_tape_io_slaves=true scope=spfile;

System altered.

SQL> show parameter BACKUP_TAPE_IO_SLAVES;

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------
------
backup_tape_io_slaves                boolean     FALSE
```

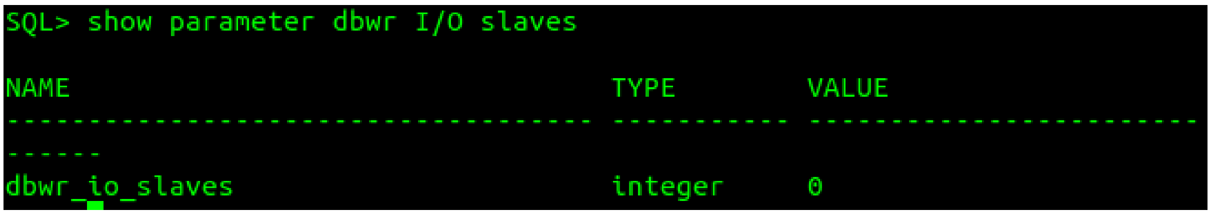
Figure 133: Altering backup slave.



```
SQL> shutdown immediate
ORA-01031: insufficient privileges
SQL> connect as sysdba
Enter user-name: clausy
Enter password:
Connected.
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2233344 bytes
Variable Size             633342976 bytes
Database Buffers          427819008 bytes
Redo Buffers                5541888 bytes
Database mounted.
Database opened.
SQL> show parameter BACKUP_TAPE_IO_SLAVES;

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------
------
backup_tape_io_slaves                boolean     TRUE
```

Figure 134: Restarting system for backup slave changes to take effect.

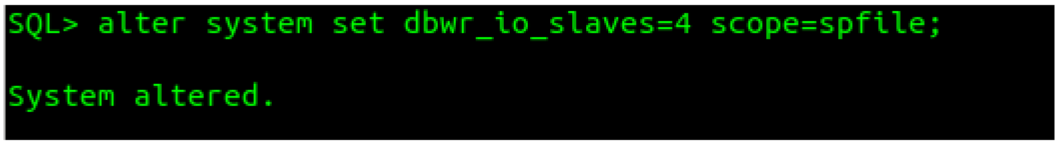
As mentioned above the database writer and log writer can make use of the I/O slave. The database writer process can have many slaves writing to the dirty data block. If you refer to figure 135 you can see the dbwr value is zero meaning that dbwr has no I/O slaves.



```
SQL> show parameter dbwr I/O slaves

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------
------
dbwr_io_slaves                       integer     0
```

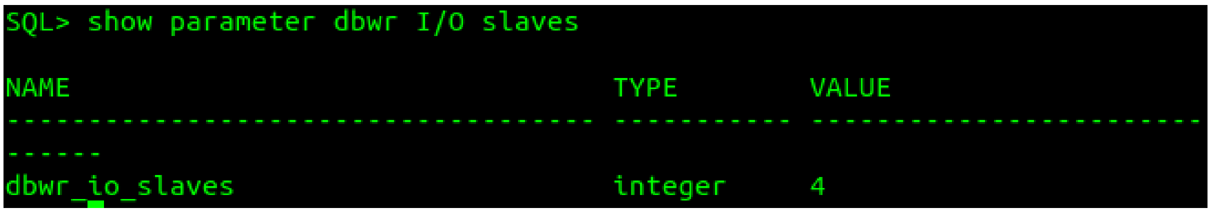
Figure 135: Showing database writer slaves.

User may add slaves to the dbwr. If you refer to figure 136, I added 4 I/O slave to the dbwr. Users then will need to perform a reboot on the system for the alteration to take effect as we are using changing the scope of the spfile as seen in figure 137.



```
SQL> alter system set dbwr_io_slaves=4 scope=spfile;

System altered.
```

Figure 136: Altering database writer slaves.



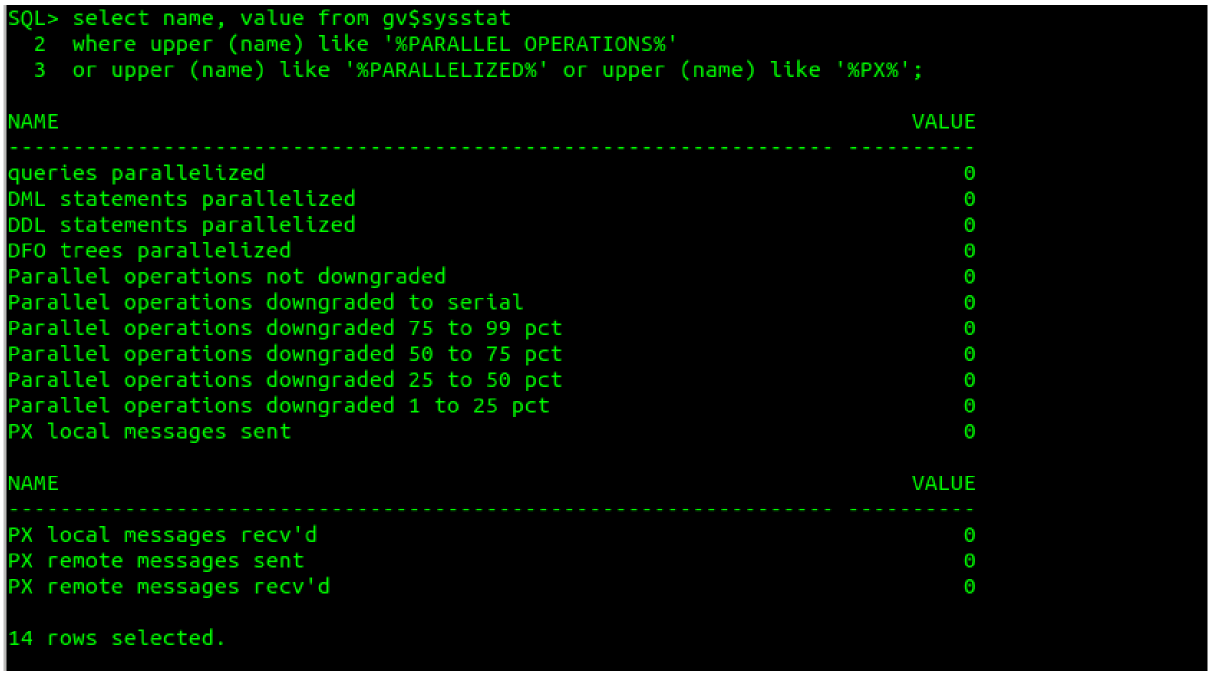
```
SQL> show parameter dbwr I/O slaves

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------
------
dbwr_io_slaves                       integer     4
```

Figure 137: Database writer slaves altered.

4.3.2 Parallel Query Slaves

Parallel query slaves do exactly what the name say. It will run SQL queries such as selects, create table, create index and updates in parallel. If you have a large amount of CPU at your disposal then it is possible to break the query up into smaller queries and run the smaller queries at the same time. The output from each query will then be merged together. If performing something like an adhoc query this will speed up the length of time it takes for the query the execute.

User can see what queries are running in parallel by using the following code referred to in figure 138.



```
SQL> select name, value from gv$sysstat
  2  where upper (name) like '%PARALLEL OPERATIONS%'
  3  or upper (name) like '%PARALLELIZED%' or upper (name) like '%PX%';

NAME                                                                  VALUE
---------------------------------------------------------------- ----------
queries parallelized                                                      0
DML statements parallelized                                               0
DDL statements parallelized                                               0
DFO trees parallelized                                                    0
Parallel operations not downgraded                                        0
Parallel operations downgraded to serial                                  0
Parallel operations downgraded 75 to 99 pct                               0
Parallel operations downgraded 50 to 75 pct                               0
Parallel operations downgraded 25 to 50 pct                               0
Parallel operations downgraded 1 to 25 pct                                0
PX local messages sent                                                    0

NAME                                                                  VALUE
---------------------------------------------------------------- ----------
PX local messages recv'd                                                  0
PX remote messages sent                                                   0
PX remote messages recv'd                                                 0

14 rows selected.
```

Figure 138:Parallel query slaves

5. Locking & Latching

Locking and Latching are very important in today's database. The reason for this is because in recent years nearly all database has become multiuser. What is meant be this is that, two people can access a resource on a database at the same time, they will to be able to perform their tasks without problems. "The locking mechanisms that allow this to happen are key features in any database" (Kyte, 2005). Most database will perform locking on different levels. An Example of this will be if a business is performing a batch update, they may choose to lock all the tables, so nobody has access. If a user is updating a record, the row that the data is contained in may be just locked. “Locks are used in the database to permit concurrent access to these shared resources while at the same time providing data integrity and consistency” (Kyte, 2005). Oracle database has some key points when it comes to locking that is goes by:

- Transactions are what database are all about.
- You should hold locks on data for as long as you need.
- There is no overhead involved in row locking in Oracle.
- You never escalate a lock (use a table lock instead of a row lock when trying to help the system).
- Concurrency and consistency can be achieved simultaneously.

5.1 Locking Issues

When database is designed badly it can lead to the locking system in a database becoming absolute. One of the big problems is lost updates. This happen when two people access the same resource. The first user updates the address of a person in the database before they can commit another user tries to update the phone number then the first user commits and the update states that is complete Then the other user finishes there update and commits. Because there was no lock in place on the row the first user update is gone because the second user had accessed the same record while the transaction was currently in process. This is called a lost update and happens because the developer refreshes all the field after the update instead of checking which field had been updated.

5.2 Optimistic Locking & Pessimistic Locking

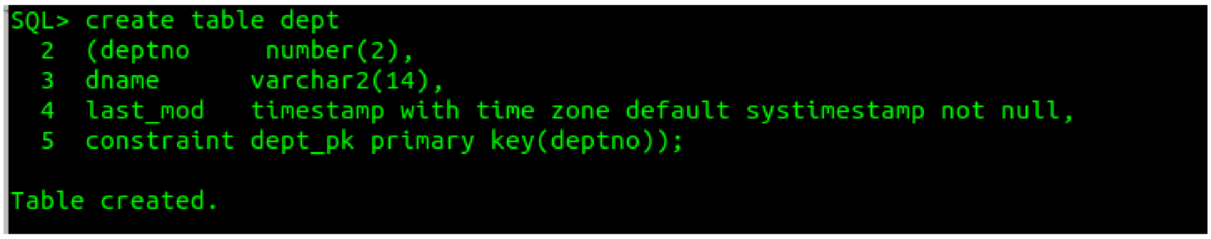
Oracle focuses on two types of locking optimistic and pessimistic. Pessimistic locking is put in place the minute before a user updates a value on the screen. A row lock would be put in places when a user states they are going to log money to their bank account. Pessimistic locking as the name would suggest always will assume the worst. Once an update is selected, they whole record is locked and no other user can read, write to the record in question (Oracle, 2019). Pessimistic locking would work better in databases where update is not a common practice.

Optimistic as the name might suggest is the complete opposite. This holds of locks right up until the update is performed. They are optimistic that the data would not be updated by someone else at the exact sometime. This way of locking work in well is all systems but is can lead to lost update. A lost update, this is when a user goes to update a row and finds that it’s already been altered and has to perform the update again.

There are a few different ways this can be achieved.

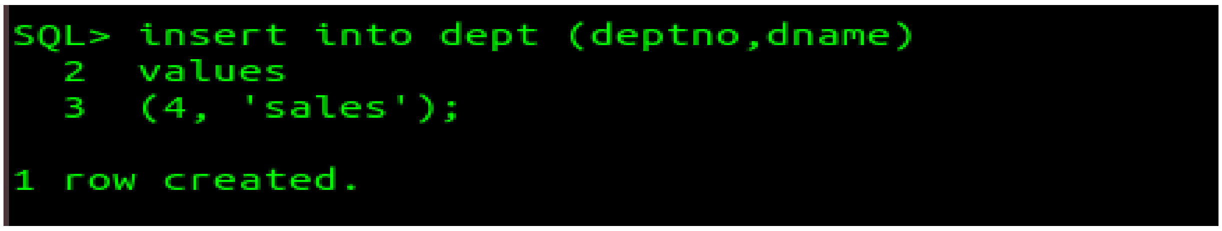
- Time Stamps
- Checksums
- ORA_ROWSCN

If you refer to figure 139 below you will see optimistic locking with a timestamp. This works by adding an extra column called last_mod. Once the extra Column is added there is a trigger or alarm set on the row so that every time the row is updated that a timestamp is taken.



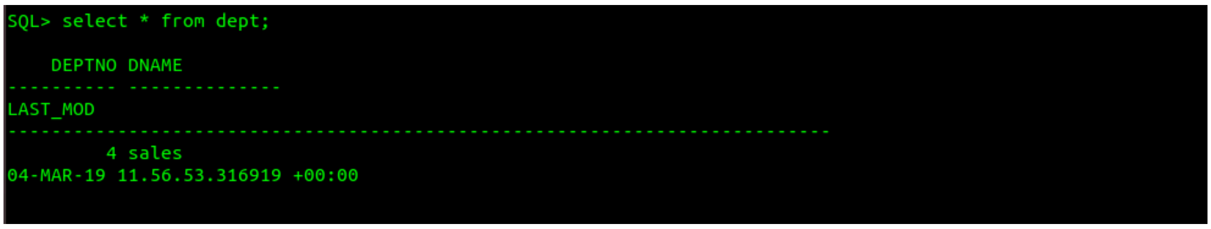
```
SQL> create table dept
  2  (deptno      number(2),
  3  dname       varchar2(14),
  4  last_mod    timestamp with time zone default systimestamp not null,
  5  constraint dept_pk primary key(deptno));

Table created.
```

Figure 139: Optimistic locking with timestamp.



```
SQL> insert into dept (deptno,dname)
  2  values
  3  (4, 'sales');

1 row created.
```

Figure 140: Insert record to show timestamp.



```
SQL> select * from dept;

    DEPTNO DNAME
---------- --------------
LAST_MOD
---------------------------------------------------------------------------
         4 sales
04-MAR-19 11.56.53.316919 +00:00
```

Figure 141: Timestamp show using select all statement.

If you refer to figure 140 and 141 you will see an insert in a table called dept and then a select all from that dept table and it shows the exact time and data that the transaction took place. By storing that value when the next transaction takes place a check can be done against that timestamp. If there happened to be a transaction at the same time, they can check the timestamp to see which transaction happened first and the could possibly merge or roll back on the transactions.

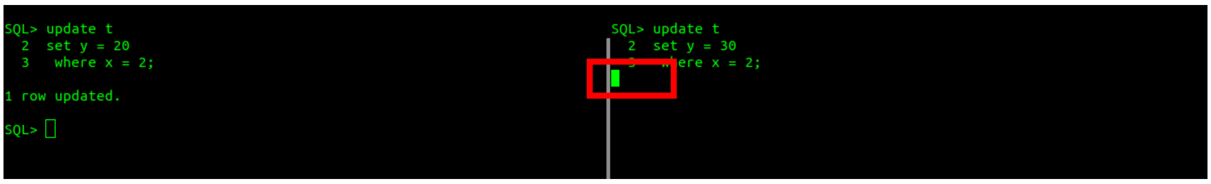
The next method focuses on generation of hash values or checksums to identify the transaction. This is done by using sum function to generate the checksum or hash value. They can then check the hash values and if there is a change in the output value of the hash value, they would know the record has been altered in some way.

The last way of optimistic locking is by using ORA_ROWSCN. This uses the system clock and anytime a commit is made the clock advances, it never goes back. This has drawback in that if you don't create a table with that function that you may get many false positive. This is because ORA_ROWSCN works at block level and you update a row on a block and that block contains 50 other rows then all the rows will have that ORA_ROWSCN value.

(Kyte, 2005)

5.3 Blocking

"Blocking happens when one session holds a lock on a resource that another session is trying to access, when this happens the system will hang" (Kyte, 2005). This may happen when perform any of the following queries in Oracle, INSERT, UPDATE, DELETE, MEARGE and SELECT FOR UPDATE. To prevent blocking while performing a SELECT FOR UPDATE state all users need to do is add a NOWWAIT clause to the statements. Blocked inserts can happen if you have linked tables through referential integrity constraints when you try to insert into the child table, the user may become blocked because the record needs to be created in the parent table first (Kyte, 2005). In order to test the locking process, I was able to create a table that had two columns both integers. I then opened two SQLPlus sessions. As shown in figure 142. The session on the left was started first and I tried to set some value. Before the where clause was executed the other update, session was started. I then executed the where clause on the first session with no problem 1 row was updated. Then tried executing the last where clause on the second session and the session was left hanging as seen highlighted in red (Oracle, 2019).



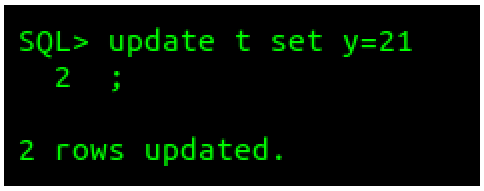
```
SQL> update t
  2  set y = 20
  3   where x = 2;

1 row updated.

SQL>
```

```
SQL> update t
  2  set y = 30
  3   where x = 2;
```

Figure 142: Testing locking

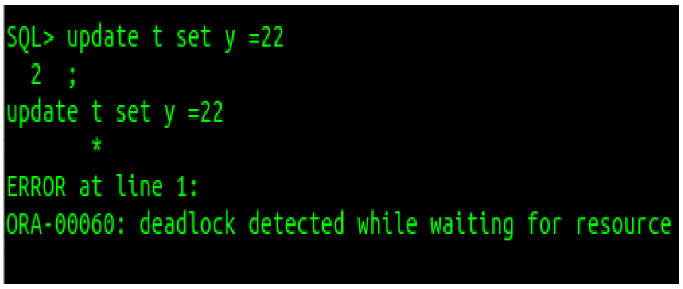
5.4 Deadlock

Deadlock happens when two sessions are active, each session has accessed a resource that the session want, and this creates a deadlock. Luckily Oracle handles deadlock where by the first session that realises, they have deadlock will be rolled back the entire query by Oracle. If refer to figure 143 and 144 below where I created a deadlock. Again, there was two sessions opened. The deadlock happened because I selected to update y in the table. But I started the query first in session 2 then I done the same thing, but I executed the query in full in session 1. Session two realised first that they were deadlocked and was waiting for the resource to become available to perform the update.



```
SQL> update t set y=21
  2  ;

2 rows updated.
```

Figure 143: Session 1 of deadlock test.



```
SQL> update t set y =22
  2  ;
update t set y =22
       *
ERROR at line 1:
ORA-00060: deadlock detected while waiting for resource
```

Figure 144: Session 2 of dead lock test.

6. Concurrency Control

In single user database uses are able to modify data without concern of other users modifying the same data. The same can't be said for multi-user database. Designers of multi-user database need to maximize concurrent access an also make sure that all the users can read and modify the data in a consistent fashion. This is performed using the locking talked about in section 5. Locking is a core mechanism by which Oracle regulated concurrent access to a shared database. As well as locking Oracle implements multi-versioning this is where many versions of the same data is kept. This means that the readers of the data will never be blocked by writer of the data. This is where Oracle differs from another database.

The main objective of a transaction in a database is to take the database from one consistent state to the next. There is standard set by the ISO for SQL, and they specify various transaction isolation levels. The isolation levels are defined by how sensitive a transaction is the greater the sensitivity the greater the degree of isolation. The Standards are known as the ANSI/ISO SQL standards and they define four states of transaction isolation. The states of transaction isolation are the broken into three categories' known as the phenomena these categories are either permitted or have no isolation level. If you refer to figure 145 you will see how they are broken down. As the (Kyte, 2005) refers to in the book some don't have an isolation level.

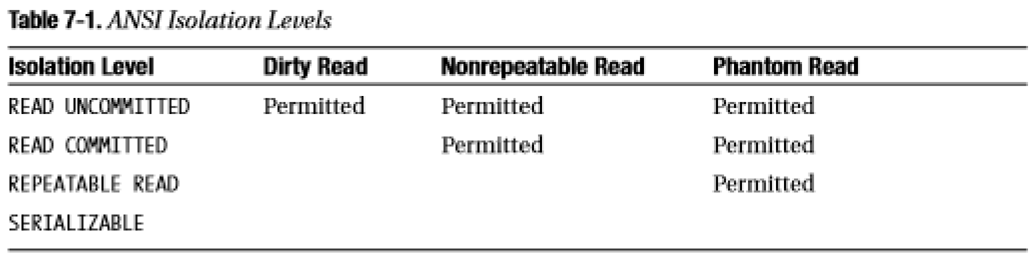


Table 7-1. *ANSI Isolation Levels*

Isolation Level	Dirty Read	Nonrepeatable Read	Phantom Read
READ UNCOMMITTED	Permitted	Permitted	Permitted
READ COMMITTED		Permitted	Permitted
REPEATABLE READ			Permitted
SERIALIZABLE			

Figure 145: ANSI isolation levels (Kyte, 2005).

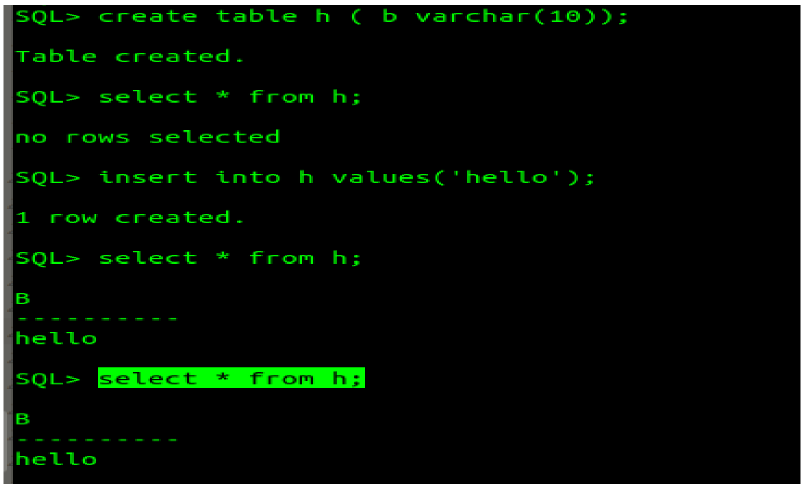
Let's look at the table is more detail first we look at the phenomena.

- Dirty Read – means that you can read uncommitted data. This happens when a file is opened that someone is reading to writing. If this happens the data integrity is compromised, foreign keys are violated, and unique constraints are ignored.
- Nonrepeatable Read – this means that when you read the data at a certain time and come back later and read the data again that it may have changed, disappeared or have been updated.
- Phantom Read – means that if a user executes a query at a certain time and then execute the query later on the data may have been added to the database and this may affect the query that your trying to execute.

7.1 Read Uncommitted

If you refer to figure 145 you will see that Read Uncommitted isolation levels allows dirty reads. The read is not used by Oracle and Oracle doesn't even allow for dirty reads. "The function of the Read Uncommitted isolation is to provide a standards-based definition that caters for non-blocking reads" (Kyte, 2005). Oracle provide these types of read by default.

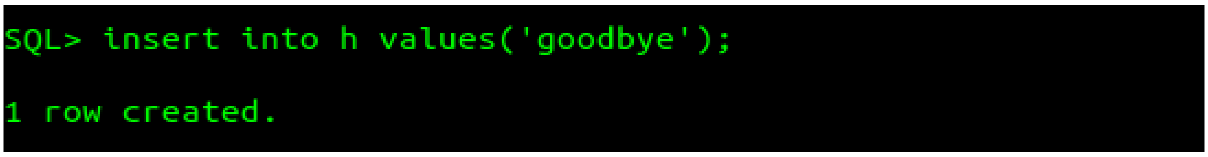
If you refer to figure 146 you will see that a table was create called h. Performed a select all query and it tells us that no row are inserted so the next step is to insert some values as shown is figure 146 and then perform the same query again you can now see the value hello in row one.



```
SQL> create table h ( b varchar(10));

Table created.

SQL> select * from h;

no rows selected

SQL> insert into h values('hello');

1 row created.

SQL> select * from h;

B
----------
hello

SQL> select * from h;

B
----------
hello
```

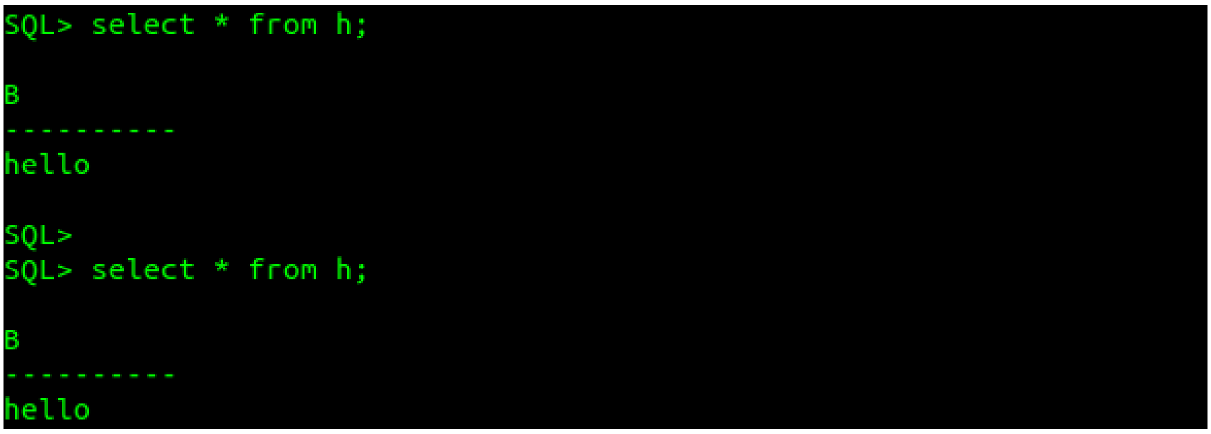
Figure 146: Creating table to demonstrate Read Uncommitted.

To perform this demonstration, you need two session running in SQLPlus. If you refer to figure 147 you can see the second session running and a command to insert a new value into h. If you refer to figure 148 you will see that the new value is not showing that is because the transaction is not committed, and Oracle does not allow for dirty reads.



```
SQL> insert into h values('goodbye');

1 row created.
```

Figure 147: Inserting new value to test Read Uncommitted.



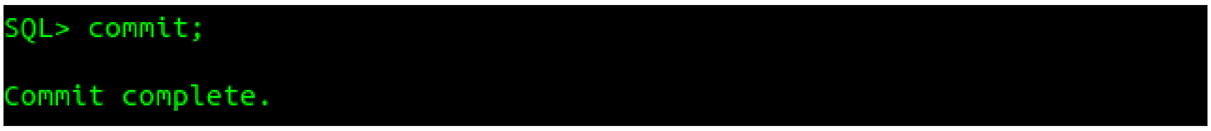
```
SQL> select * from h;

B
----------
hello

SQL>
SQL> select * from h;

B
----------
hello
```

Figure 148: Proving that Oracle doesn't support dirty reads.

7.2 Read Committed

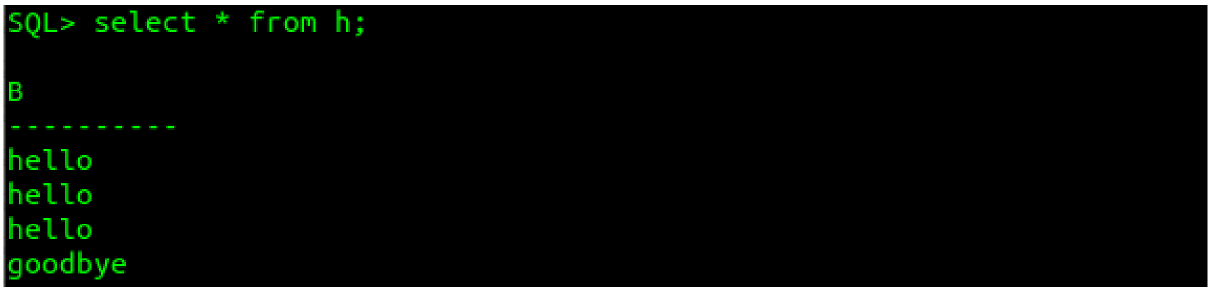
The isolation level say that a transaction can only read data that has been committed to the database. As stated above there is no dirty reads. Read committed is the most commonly used isolation level is all database and is the default mode for Oracle database.

Following on from our last example where I had yet to commit the transactions. If you refer to figure 149 a commit command was issued and now you will see that all the commands that were issued in the Read Uncommitted section of the document are now propagating the rows of data shown is figure 150.



```
SQL> commit;

Commit complete.
```

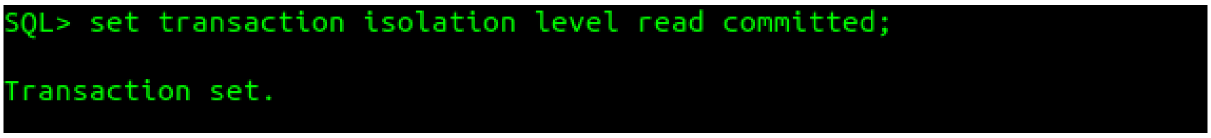
Figure 149: Issuing a commit to test Read Committed.



```
SQL> select * from h;

B
----------
hello
hello
hello
goodbye
```

Figure 150: Showing committed transaction.

If you refer to figure 151 this command may be used of there was a different isolation level set like Repeatable Read. This will set the mode back to Read Committed.



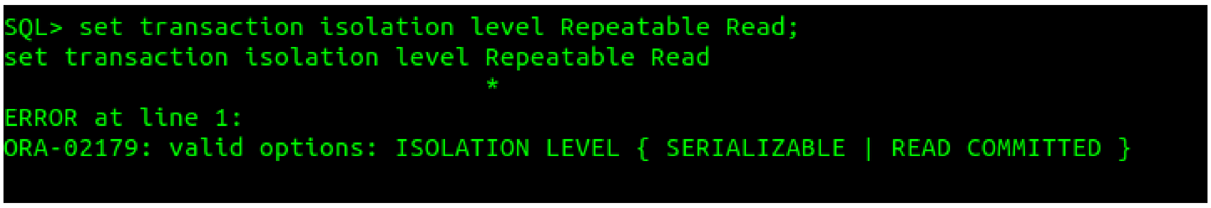
```
SQL> set transaction isolation level read committed;

Transaction set.
```

Figure 151: Setting Read Committed mode in Oracle.

7.3 Repeatable Read

This from of isolation level gives a consistent, correct answers and to prevent lost updates. Any results of queries should be consistent with some point in time. Most database will use row level shared locks these give answers to queries from when the query is finished. Oracle with their multi-versioning took a different approach by giving the answer before the query was executed (Kyte, 2005).

If you refer to figure 152 Oracle doesn’t allow for Repeatable Reads and you can only set to levels of isolation and that is Read Committed and Serializable.



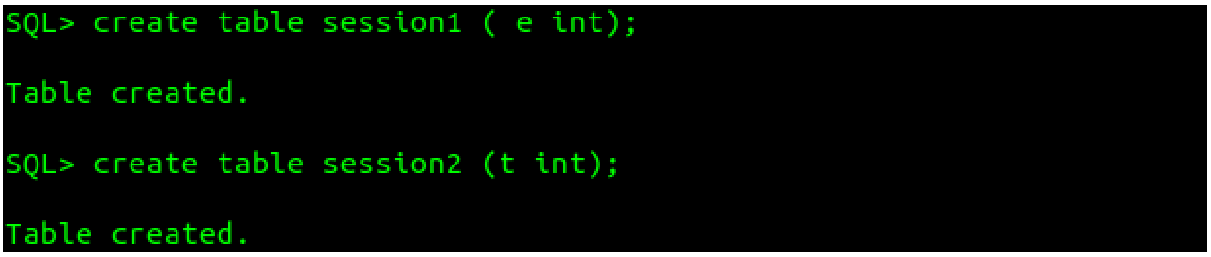
```
SQL> set transaction isolation level Repeatable Read;
set transaction isolation level Repeatable Read
                                *
ERROR at line 1:
ORA-02179: valid options: ISOLATION LEVEL { SERIALIZABLE | READ COMMITTED }
```

Figure 152:Trying to set Repeatable Reads in Oracle.

7.4 Serializable

This level of isolation is seen as the most restrictive of all the isolation levels. But the advantage is that it provides the highest degree of isolation. These transactions work in surroundings that makes them appear as they are not been modified by any other users. Rows read by users will be the same after the are reread and when queries execute it will bring back the same result for the length of the transaction (Kyte, 2005).

If you refer to figure 153 to test the serializable isolation, we create two table session 1 and session 2.

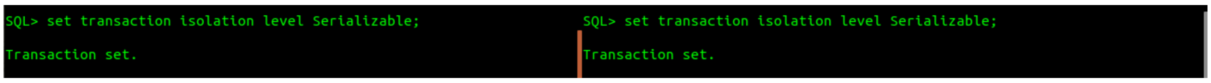


```
SQL> create table session1 ( e int);

Table created.

SQL> create table session2 (t int);

Table created.
```

Figure 153: Creating tables to test Serializable isolation.

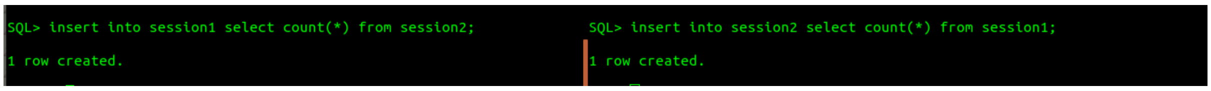
If you refer to figure 154 you will see two sessions happening. The first step is to set the isolation level as Read Commit is set by default in Oracle. Once this is done the next commands shown in figure 155 will perform a count of all the rows in each table and it will insert the returned value into the opposite table. After this the commit command is issued on both tables as shown in figure 156 below.

Session 2 **Session 1**



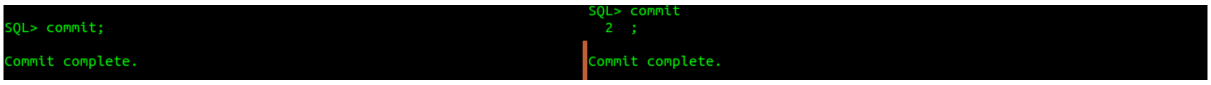
```
SQL> set transaction isolation level Serializable;       SQL> set transaction isolation level Serializable;

Transaction set.                                          Transaction set.
```

Figure 154: Setting isolation level to Serializable.



```
SQL> insert into session1 select count(*) from session2;    SQL> insert into session2 select count(*) from session1;

1 row created.                                              1 row created.
```

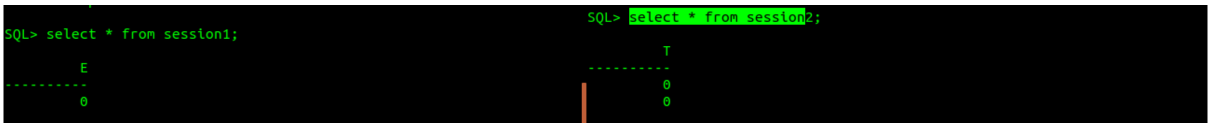
Figure 155: Inserting values into tables session1 and session2.



```
                                                            SQL> commit
SQL> commit;                                                  2  ;

Commit complete.                                            Commit complete.
```

Figure 156; Issuing commit to session tables.

There is one extra value shown in session 1 when the select all command is issued on the two tables shown in figure 156. I issued the insert command twice by accident. But you will notice the value didn't change when it really should have changed to one because there is the second row in session1 table. This is what (Kyte, 2005) stated in the book that it makes it appear as if no one else modified the tables.



```
                                                            SQL> select * from session2;
SQL> select * from session1;
                                                                     T
         E                                                  ----------
----------                                                           0
         0                                                           0
```

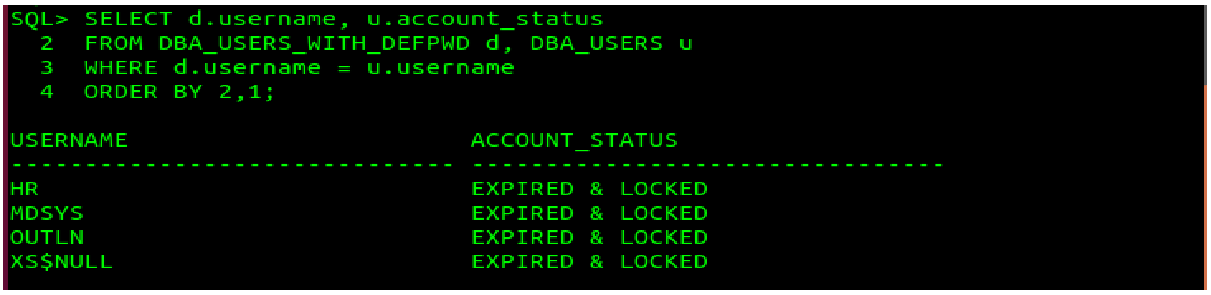
Figure 157: Confirming Serialization is working.

7. Security

Security in Oracle11g can be handled on different levels. The first and most basic is create users that has special privileges. In the installation process I created a user called clausy then in section 2.8 password files I granted clausy more privileges like system dba, system database and system operations.

Authentication is another way of making access more secure. There are three levels where authentication can be set, at database level, operating system level and network. Oracle by default will encrypt all password using Advanced Encryption Standard (AES) on client to server connections this means if a man in the middle attack was to occur that they would need to break the encryption to access to database. Oracle provides a few other methods of securing their stored passwords. They limit the amount of time that someone can tries a password before they are locked. Password are also case sensitive meaning the password must be typed exactly how it is stored. Hashes are taken of all passwords using a sha-1 hashing function this along with AES encryption make Oracle database very difficult to break.

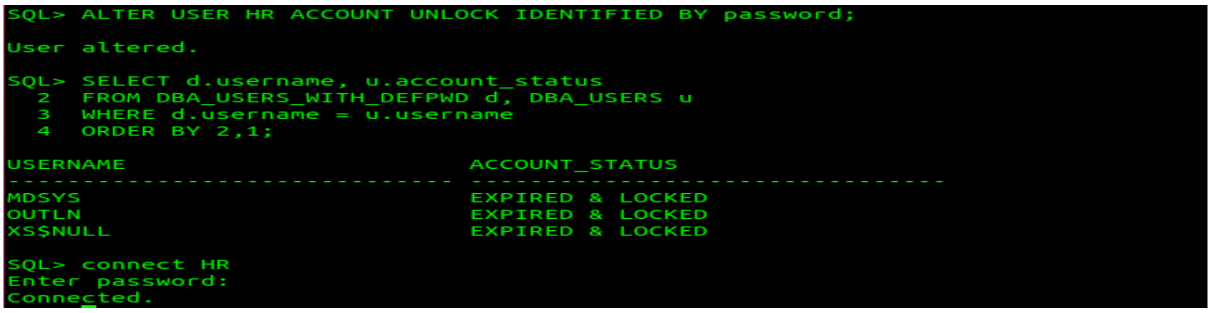
By default, when a new database is created all the default accounts that Oracle provides are locked. If you refer to figure 158 you will see you can check all the users accounts that have default password set. Default password will make the database vulnerable to attack because the default password is well known.



```
SQL> SELECT d.username, u.account_status
  2  FROM DBA_USERS_WITH_DEFPWD d, DBA_USERS u
  3  WHERE d.username = u.username
  4  ORDER BY 2,1;

USERNAME                       ACCOUNT_STATUS
------------------------------ --------------------------------
HR                             EXPIRED & LOCKED
MDSYS                          EXPIRED & LOCKED
OUTLN                          EXPIRED & LOCKED
XS$NULL                        EXPIRED & LOCKED
```

Figure 158: Locked Oracle default accounts (Oracle, 2019).

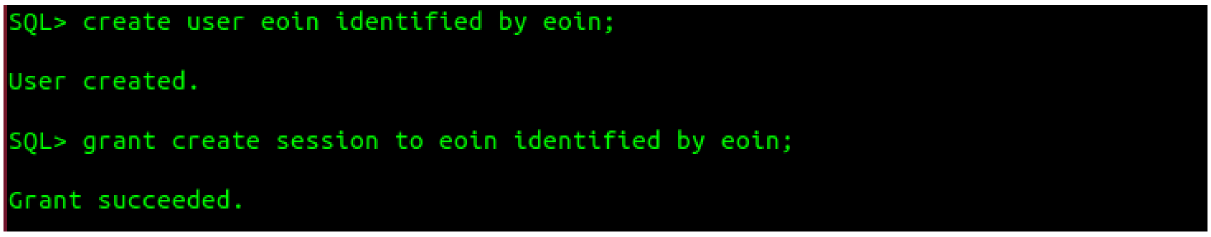
If you refer to figure 158 you will see the first command is user to unlock the HR account, I have also assigned it a new password called password not very secure but will do for a test. You will see that again I checked for default password and you will notice the user HR is now gone from the list. I then connected as that user and it was successful. This is something that should be checked upon creation of a database because these are defaults users for all database and could be easily unlocked.



```
SQL> ALTER USER HR ACCOUNT UNLOCK IDENTIFIED BY password;

User altered.

SQL> SELECT d.username, u.account_status
  2  FROM DBA_USERS_WITH_DEFPWD d, DBA_USERS u
  3  WHERE d.username = u.username
  4  ORDER BY 2,1;

USERNAME                       ACCOUNT_STATUS
------------------------------ --------------------------------
MDSYS                          EXPIRED & LOCKED
OUTLN                          EXPIRED & LOCKED
XS$NULL                        EXPIRED & LOCKED

SQL> connect HR
Enter password:
Connected.
```

Figure 159: Changing default password for HR.

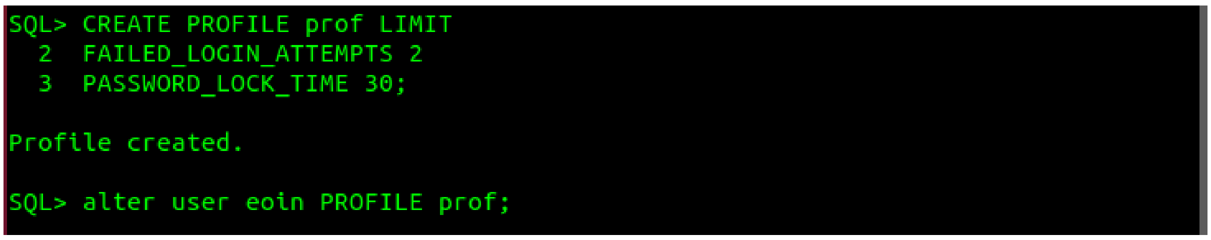
Another option in Oracle is to use profiles. “A profile is a collection of parameters that sets limits on database resources” (Oracle, 2019). You can then add users to those profiles with limits set on the parameter. Like of you have a sysdba you might not want then to have as many logins tries then a normal database user. In the example shown below, first shown in figure 160 I am creating a user called eoin and giving him permission to open up a user session.



```
SQL> create user eoin identified by eoin;

User created.

SQL> grant create session to eoin identified by eoin;

Grant succeeded.
```

Figure 160: Creating a new user eoin.

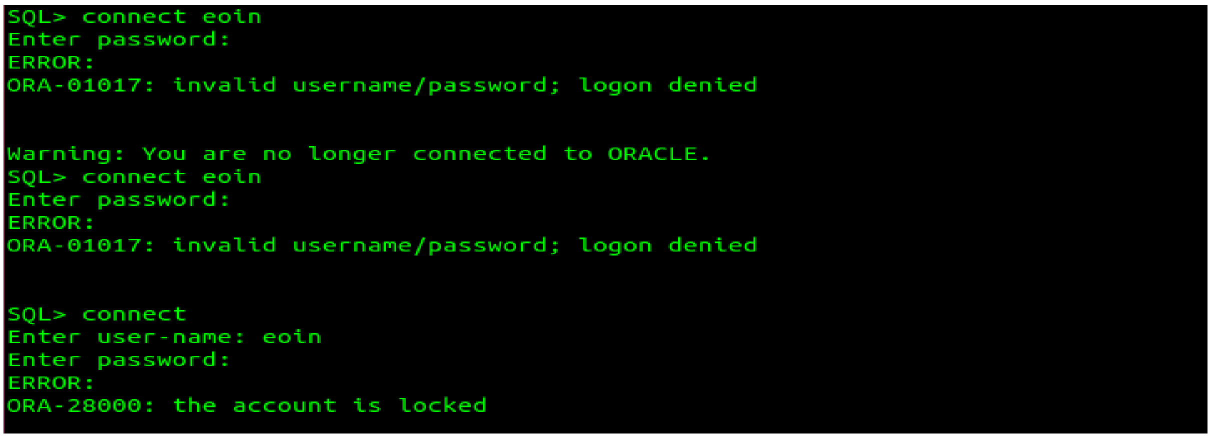
After this the next step is to create a profile. If you refer to figure 161 a profile called prof was created that would limit any users in this profile to have to login attempt before that are locked out for 30 minutes. The last command seen adds the user eoin to the profile using the alter command.



```
SQL> CREATE PROFILE prof LIMIT
  2  FAILED_LOGIN_ATTEMPTS 2
  3  PASSWORD_LOCK_TIME 30;

Profile created.

SQL> alter user eoin PROFILE prof;
```

Figure 161: Creating profile.

If you refer to figure 162 to test this setting eoin logged in three time with the wrong password and the account was locked



```
SQL> connect eoin
Enter password:
ERROR:
ORA-01017: invalid username/password; logon denied


Warning: You are no longer connected to ORACLE.
SQL> connect eoin
Enter password:
ERROR:
ORA-01017: invalid username/password; logon denied


SQL> connect
Enter user-name: eoin
Enter password:
ERROR:
ORA-28000: the account is locked
```

Figure 162: User eoin account locked.

Some other really useful command used is the password setting is PASSWORD_LIFE_TIME and PASSWORD_REUSE_MAX. What the first command will do is set a time limit on the password so if you users don’t change their password in the giving time limit, they will be locked out. The second command look at password reuse. It set so that user can’t use the same password over and over again. These are really useful controls to have on the database.

8. Conclusion

The purpose of this report when to understand the underlying architecture of an Oracle database. Once the underlying architecture was understood we would then be able to administer the database. This was a really useful exercise as it gave me great understanding of database. What I mean by this is up until now we have been developing database but after doing this module you can see there is so much more then developing. It gives a good understanding of how database is implemented on a business level. It also gives a great understanding on how a database work underneath the hood.

Using SQLPlus was good fun as I find command line stuff very interesting. Up until now we had only used MySQL database, so I was great to get an understanding of a different type of database. After using MySQL all along my opinion is that Oracle is a superior database.

9. Bibliography

Center, O. H., 2019. *Database Concepts 8 Memory Structures.* [Online]
Available at: https://docs.oracle.com/cd/B19306_01/server.102/b14220/memory.htm#i10093
[Accessed 21 02 2019].

Kyte, T., 2005. *Expert Oracle Database Architecture 9i and 10g Programming Techniques and Solutions.* 2 ed. s.l.:Apress.

O'Brien, J., 2014. *Database Administration,* Waterford: s.n.

Oracle, 2019. *Application Server TopLink Application Developer's Guide.* [Online]
Available at: https://docs.oracle.com/cd/B14099_19/web.1012/b15901/dataaccs008.htm
[Accessed 04 03 2019].

Oracle, 2019. *AskTom.* [Online]
Available at: https://asktom.oracle.com/pls/apex/asktom.search?tag=oracle-locksblocksdeadlocks
[Accessed 03 04 2019].

Oracle, 2019. *Configuring Memory Mangement.* [Online]
Available at:
https://docs.oracle.com/cd/B28359_01/server.111/b28310/memory004.htm#BGBBHGDD
[Accessed 21 02 2019].

Oracle, 2019. *Database Express Edition Installation Guide.* [Online]
Available at: https://docs.oracle.com/cd/E17781_01/install.112/e18802/toc.htm#XEINL110
[Accessed 04 02 2019].

Oracle, 2019. *Database Perfromance Tuning Guide.* [Online]
Available at: https://docs.oracle.com/database/121/TGDBA/tune_pga.htm#TGDBA346
[Accessed 21 02 2019].

Oracle, 2019. *Database Referance Oracle Help Center.* [Online]
Available at: https://docs.oracle.com/cd/B19306_01/server.102/b14237/initparams107.htm
[Accessed 19 02 2019].

Oracle, 2019. *Database Security Guide.* [Online]
Available at:
https://docs.oracle.com/cd/E11882_01/network.112/e36292/authentication.htm#DBSEG33223
[Accessed 06 03 2019].

Oracle, 2019. *Memory Architecture.* [Online]
Available at: https://docs.oracle.com/database/121/CNCPT/memory.htm#CNCPT007
[Accessed 21 02 2019].

Oracle, 2019. *Oracle Database Memory Management.* [Online]
Available at: https://docs.oracle.com/cd/E11882_01/server.112/e40540/memory.htm#BGBCBGIB
[Accessed 21 02 2019].

Oracle, 2019. *Oracle Database Preinstallation Requirements.* [Online]
Available at: https://docs.oracle.com/cd/E11882_01/install.112/e47798/reqs.htm#NTDBI2690
[Accessed 04 02 2019].

Oracle, 2019. *Oracle FAQ's.* [Online]
Available at: http://www.orafaq.com/wiki/Control_file
[Accessed 19 02 2019].

Oracle, 2019. *Oracle Help Center.* [Online]
Available at: https://docs.oracle.com/en/database/oracle/oracle-database/12.2/hpdbi/about-assigning-global-database-names-during-installation.html#GUID-1AC9E18F-0AE4-4263-B3CC-A83B859A178F
[Accessed 25 01 2019].

Oracle, 2019. *Oracle Help Center.* [Online]
Available at:
https://docs.oracle.com/cd/B19306_01/server.102/b14237/initparams041.htm#REFRN10031
[Accessed 12 02 2019].

Redressa, E., 2012. *Me and My Ubuntu.* [Online]
Available at: http://meandmyubuntulinux.blogspot.com/2012/05/installing-oracle-11g-r2-express.html
[Accessed 04 02 2019].

10. Appendix