**Name: Avinash Singh [115408163]**

**Group 02**

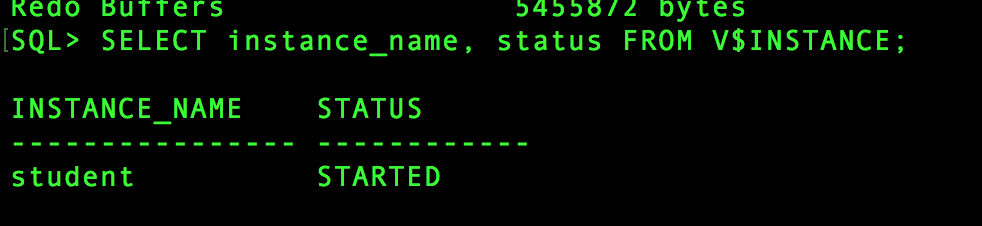
**DBA625 ASSIGNMENT - 1**

**One:**

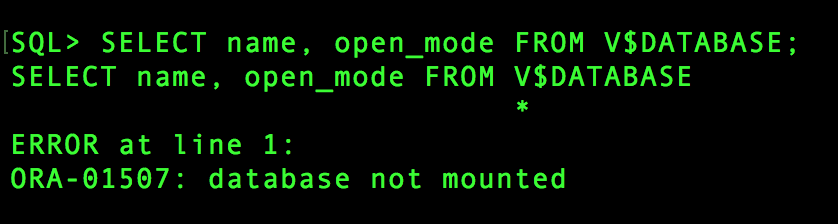
1. Your Database should be down initially. Perform the Instance Startup by using your PFILE and move through all 3 stages (one after another and without shutdown). AFTER each stage check TWO appropriate dynamic views, where first will show that you really are in that stage and second will show that you are not in the next stage (except when in OPEN stage, you should check only one view)

STAGE 1 : DATABASE STARTED but NOT MOUNTED.

1. sqlplus / as sysdba
2. STARTUP PFILE=initassignment.ora NOMOUNT
3. SELECT instance\_name, status FROM V$INSTANCE;



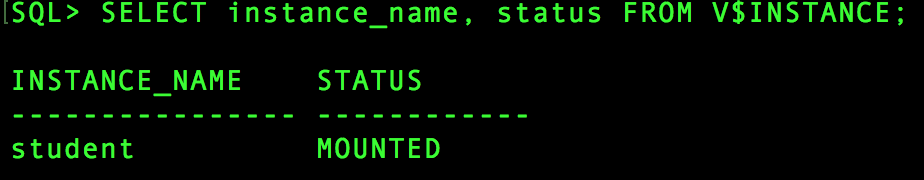
1. SELECT name, open\_mode FROM V$DATABASE;

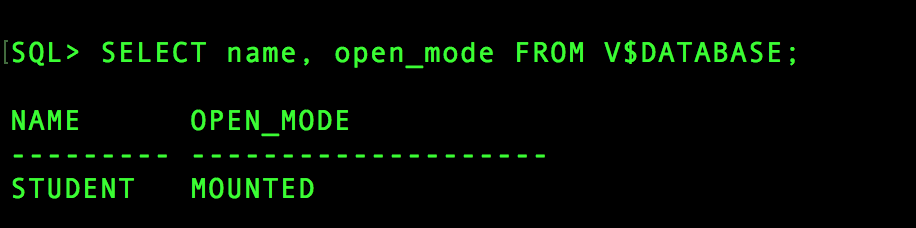


STAGE 2 : DATABASE MOUNTED but NIT OPENED.

1. ALTER DATABASE MOUNT;



1. SELECT instance\_name, status FROM V$INSTANCE; 
2. SELECT name, open\_mode FROM V$DATABASE;

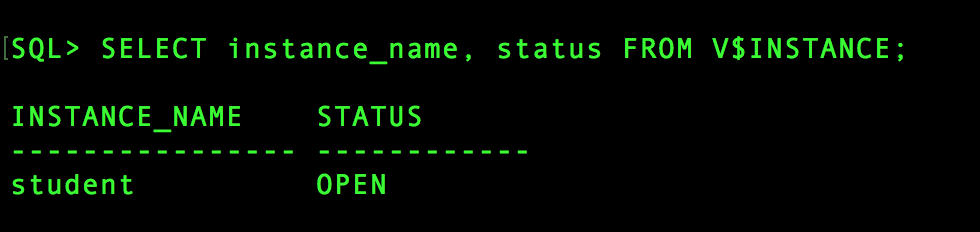


STAGE 3 : DATABASE MOUNTED AND OPENED.

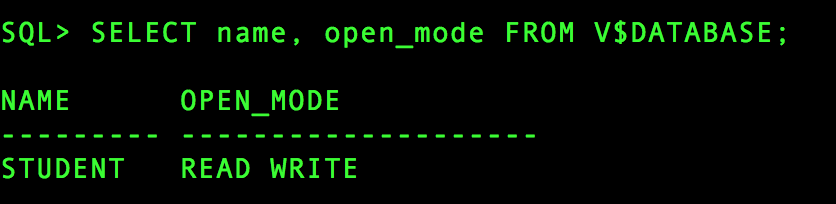
1. ALTER DATABASE OPEN.



1. SELECT instance\_name, status FROM V$INSTANCE;



1. SELECT name, open\_mode FROM V$DATABASE;

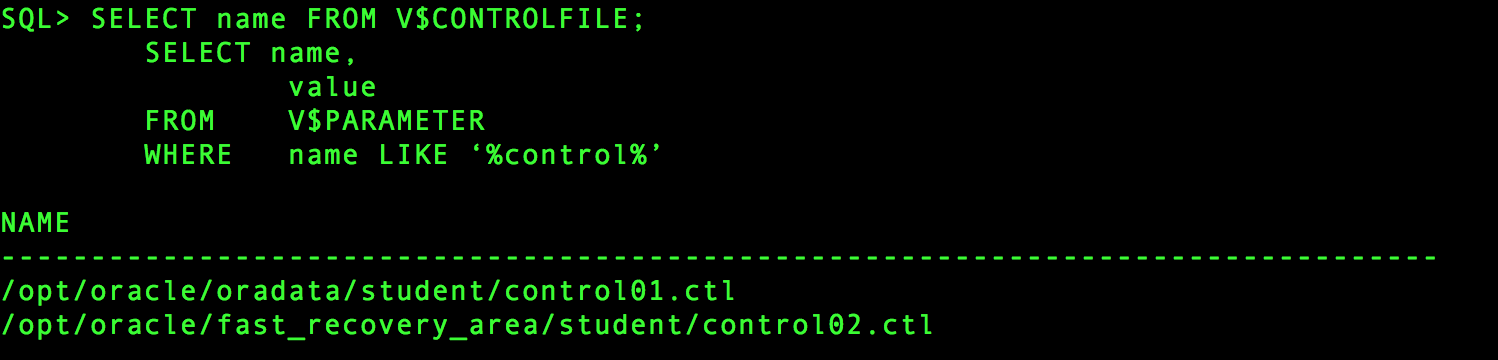


1. Show the full paths and names (just names) for all Control, Log and Data files by checking appropriate dynamic performance views.

CONTROL FILES: -

SELECT name FROM V$CONTROLFILE;

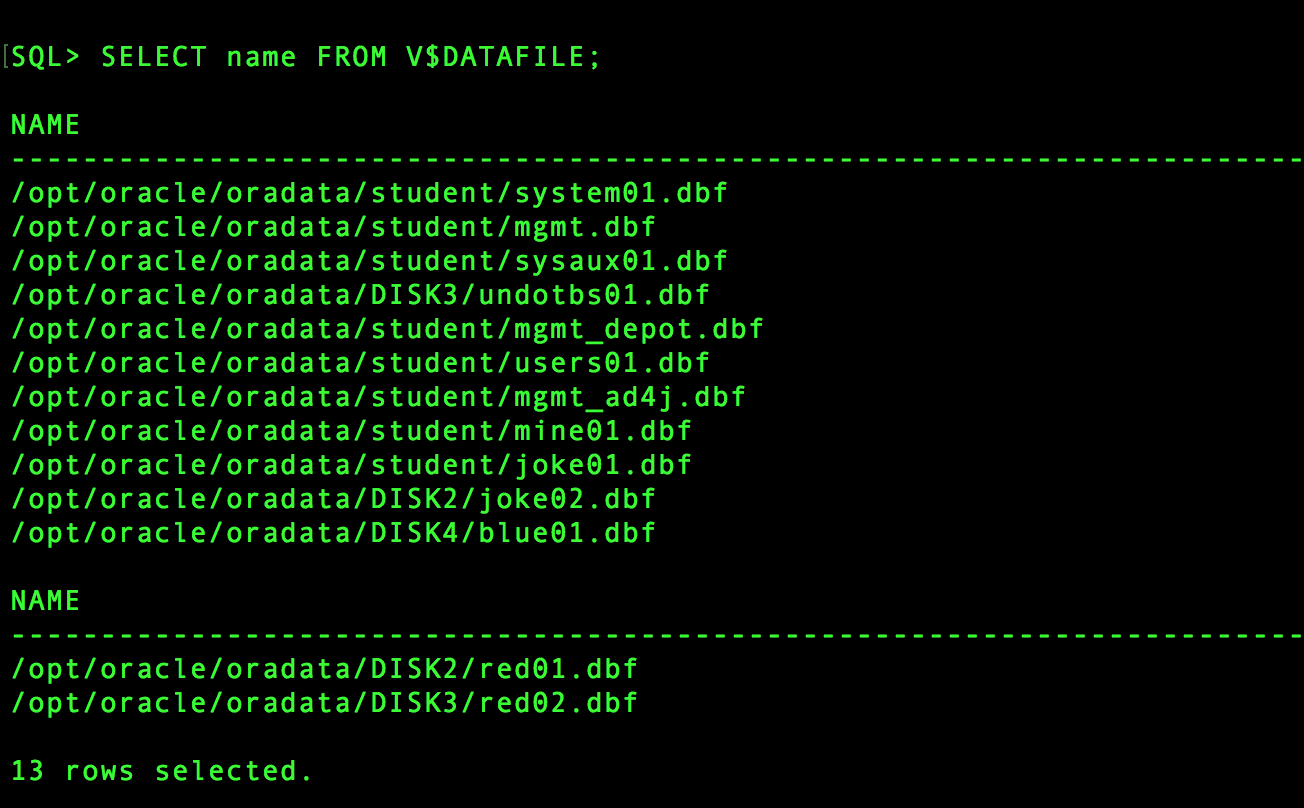
OUTPUT:



DATA FILE: -

SELECT name FROM V$DATAFILE;

OUTPUT:



LOG FILES: -

SELECT member FROM V$LOGFILE;

OUTPUT:

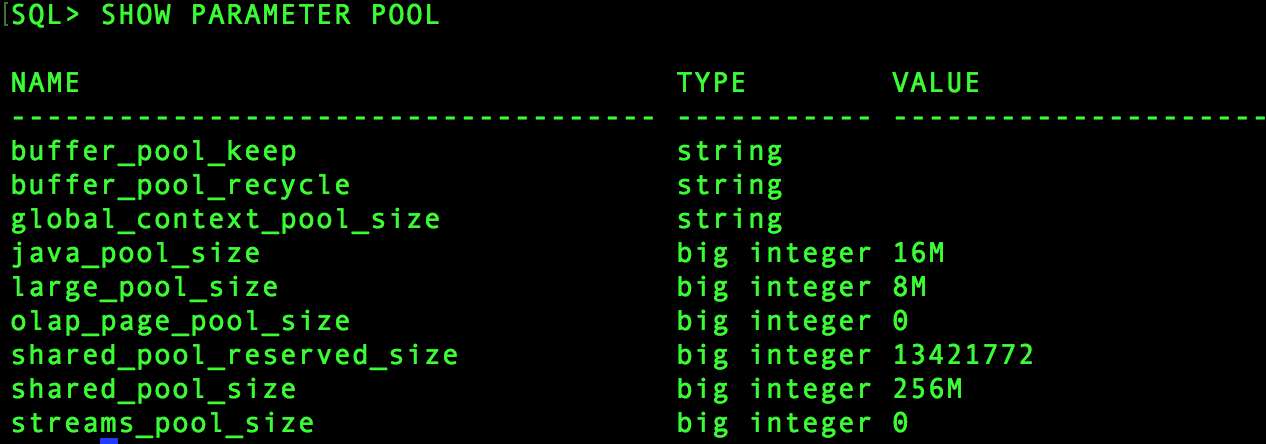


c)  Show values of all six important SGA parameters by using the “Quick” method three times (without specifying SELECT statement).

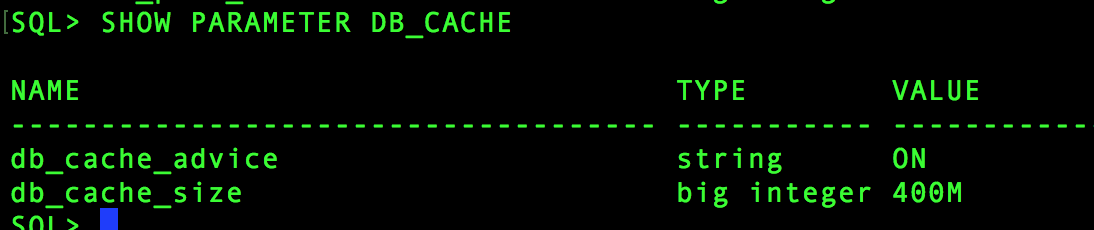
SHOW PARAMETER LOG\_BUFFER



SHOW PARAMETER POOL

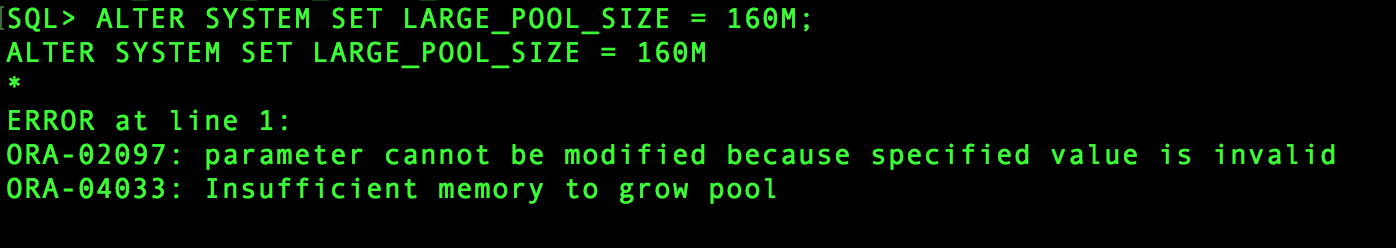


SHOW PARAMETER DB\_CACHE



1. Then try to increase the value of your Large Pool promptly to 160M. What happened? Explain.

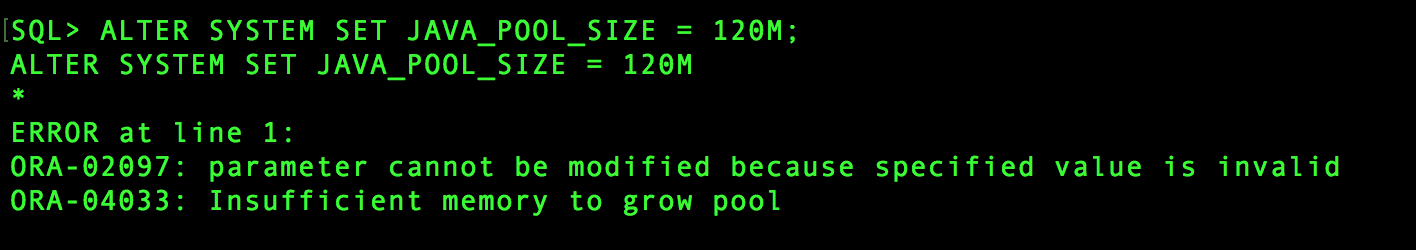
ALTER SYSTEM SET LARGE\_POOL\_SIZE = 160M;



Explanation: Since the total Size of SGA\_TARGET is 804M and out that DB\_CACHE\_SIZE is 400M there is not much space left (space left after all allocation is 116M) to allocate 160M to LARGE\_POOL, we would need to decrease the Size of DB\_CACHE in multiples of 4 to allocate desired space to LARGE\_POOL.

1. Now, try to increase the value of your Java Pool promptly to 120M.   
          What happened? Explain.

ALTER SYSTEM SET JAVA\_POOL\_SIZE = 120M;



Explanation: Since the total Size of SGA\_TARGET is 804M and out that DB\_CACHE\_SIZE is 400M there is not much space left (space left after all allocation is 116M) to allocate 160M to LARGE\_POOL, we would need to decrease the Size of DB\_CACHE in multiples of 4 to allocate desired space to LARGE\_POOL.

NOTE:    Promptly means NOW and NOT LATER after the new startup.

1. Deduct the minimal amount of memory from Large Pool, so that you can achieve the target of 120M for Java Pool. Perform both tasks (decrease followed by increase).
2. Decreasing Large pool size.

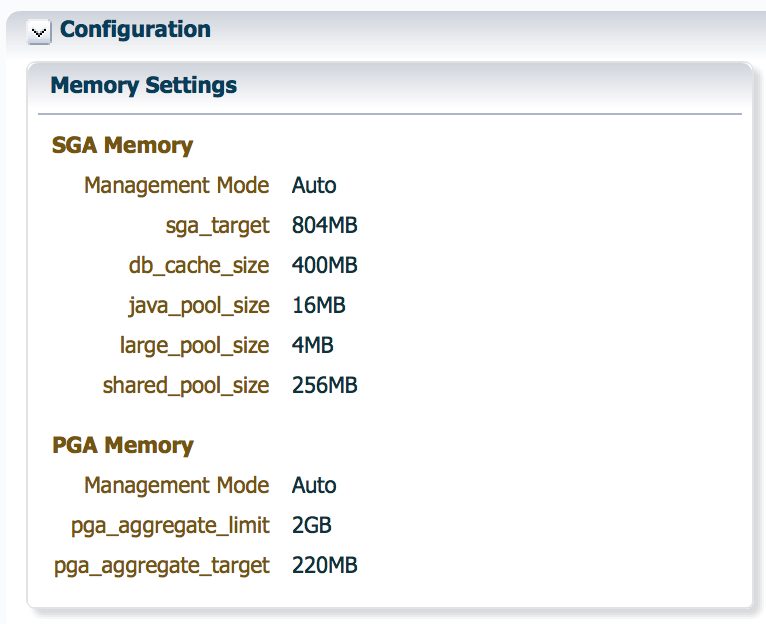
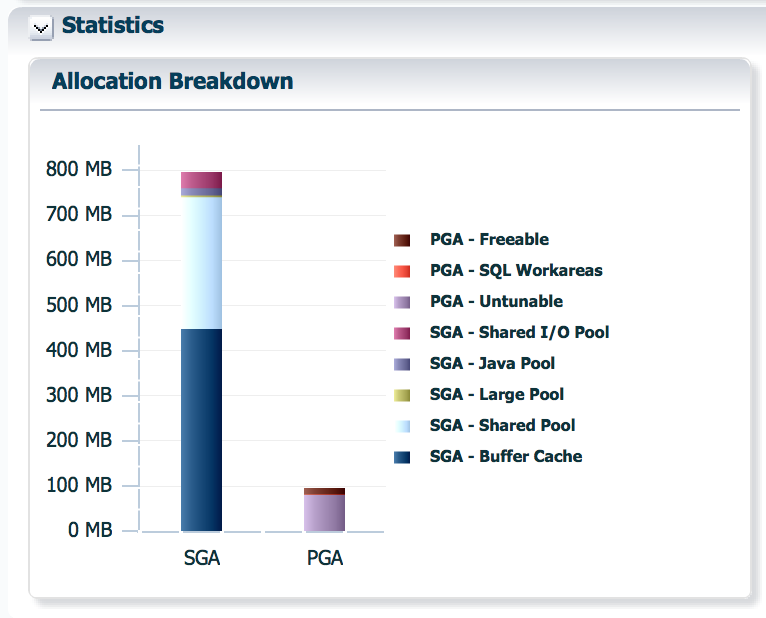
ALTER SYSTEM SET LARGE\_POOL\_SIZE = 4M;

1. Increasing Java pool size.

ALTER SYSTEM SET JAVA\_POOL\_SIZE = 120M;

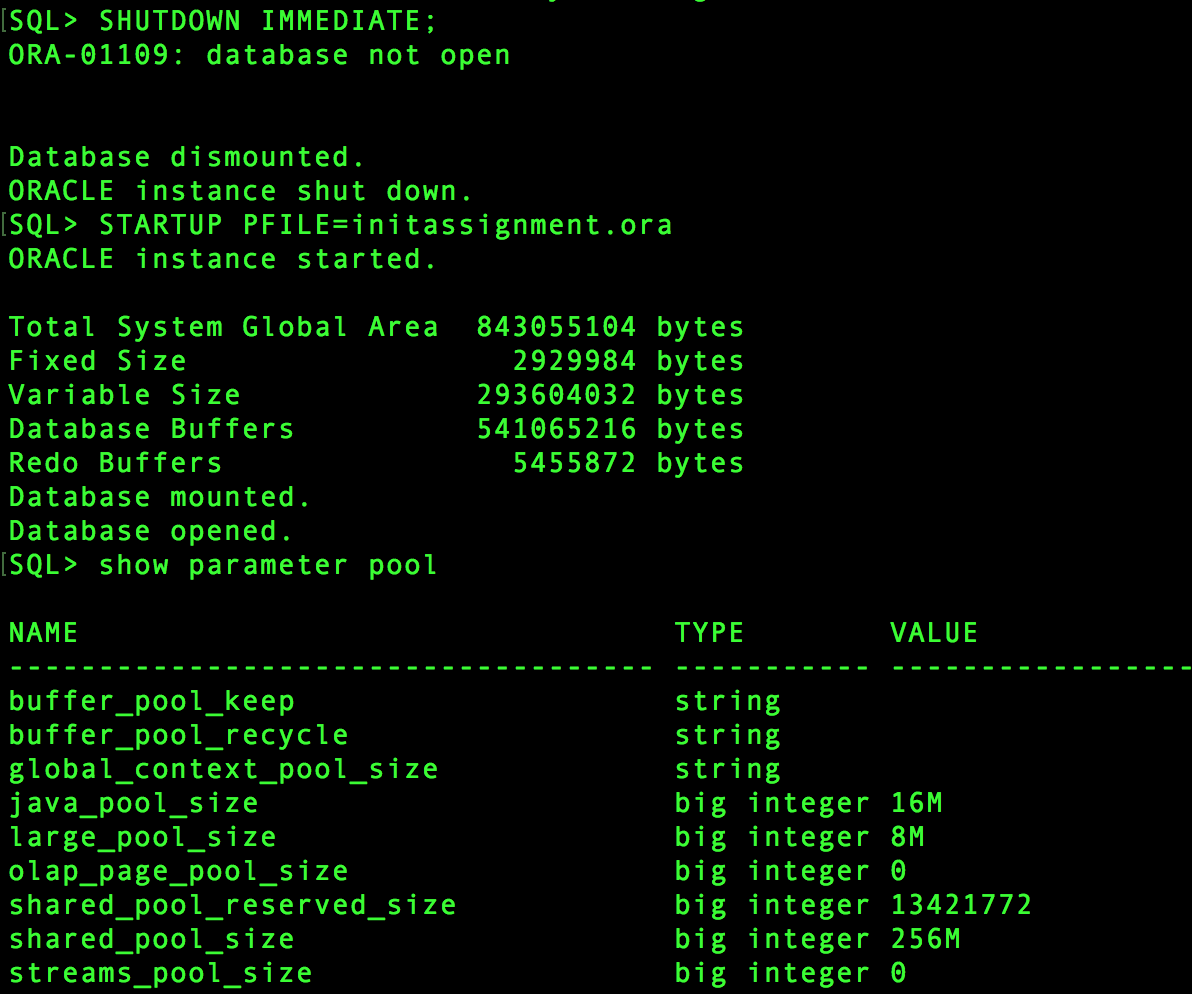


1. In DB Express figure out what is the Current Usage of SGA and then print what portion of SGA is still unused.



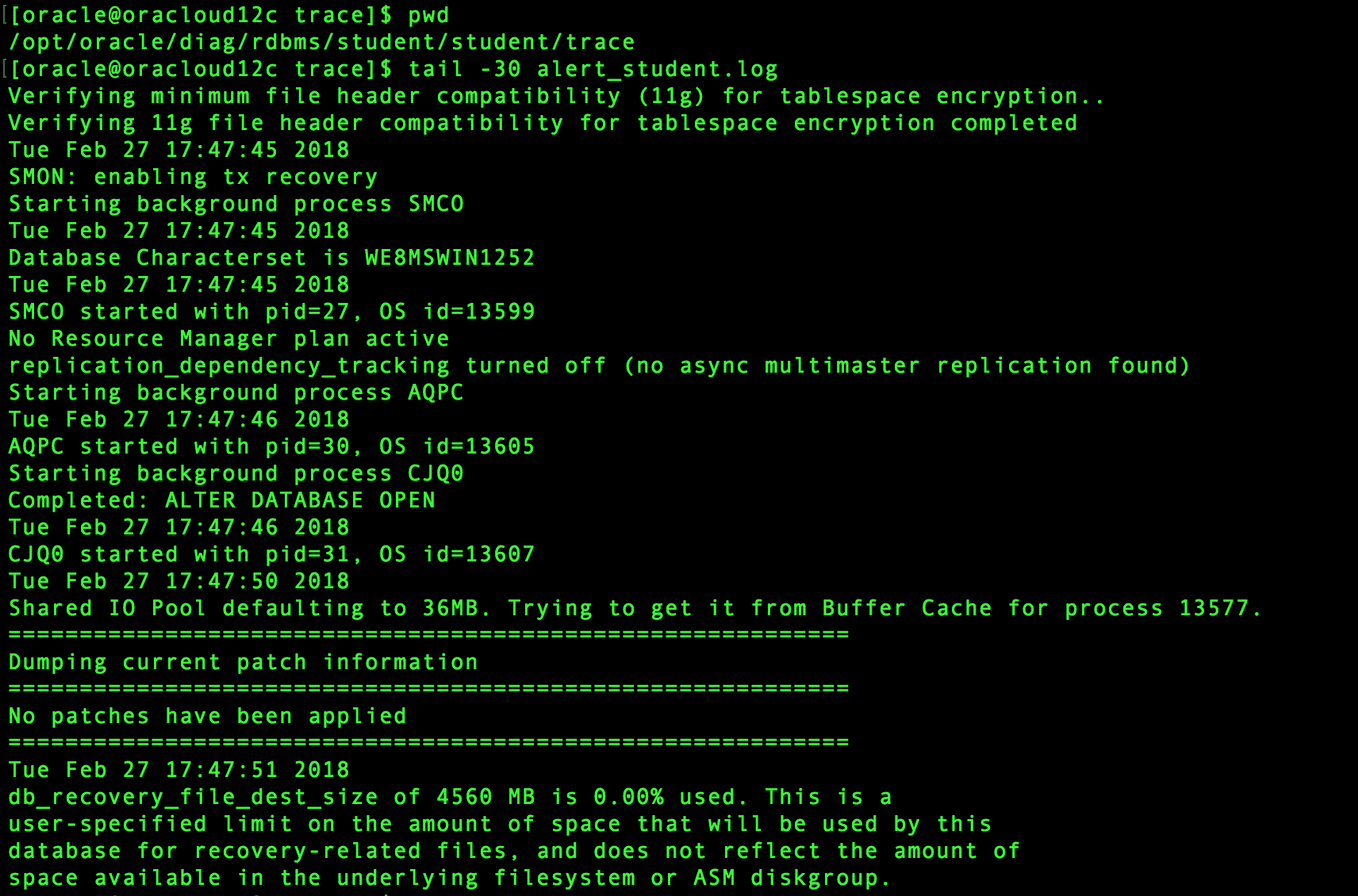
1. Close your Database so that Checkpoint will NOT happen and all transactions will be undone. Then start it by using your PFILE again. What happened with changes for Large and Java Pool?

It did not change.



i) Now display last 30 lines from your “Database Journal”

tail -30 alert\_student.log



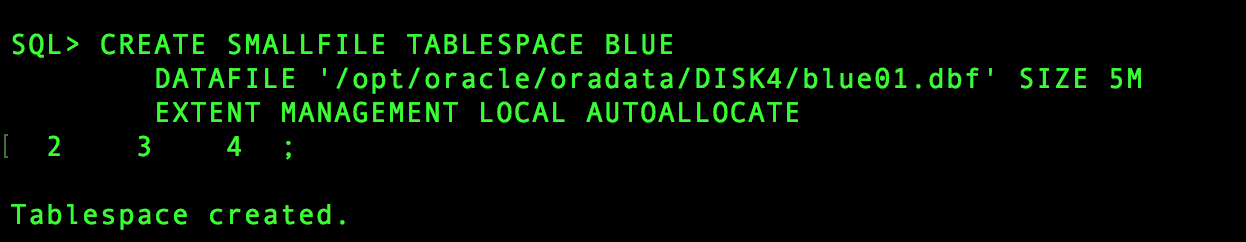
**Two:**

a)    Create a new Tablespace called BLUE with an initial size of 5M and that all extents are automatically sized and allocated by Server. Datafile **blue01.dbf** should go to DISK4 folder.

CREATE SMALLFILE TABLESPACE BLUE

DATAFILE '/opt/oracle/oradata/DISK4/blue01.dbf' SIZE 5M

EXTENT MANAGEMENT LOCAL AUTOALLOCATE;



b)    Create a new Tablespace called RED with an initial size of 2M with all extents of equal size 200K. Datafile **red01.dbf** should go to DISK2 folder and should have auto-extend option turned on with all new extents of 128K in size and with the limit of 5M.

CREATE SMALLFILE TABLESPACE RED

DATAFILE '/opt/oracle/oradata/DISK2/red01.dbf' SIZE 2M

AUTOEXTEND ON NEXT 128K MAXSIZE 5M

UNIFORM SIZE 200K;

c)    Add a second datafile to Tablespace RED with an initial size of 1M.  
       Datafile **red02.dbf** should go to DISK3 folder and try to make this file with all extents uniformed in size of 48K.   
       Explain why that was not successful? Then just add this file without that option.

ALTER TABLESPACE RED

ADD DATAFILE '/opt/oracle/oradata/DISK3/red02.dbf' SIZE 1M

UNIFORM SIZE 48K;

REASON: Because there is no such option available with ALTER TABLESPACE.

ALTER TABLESPACE RED

ADD DATAFILE '/opt/oracle/oradata/DISK3/red02.dbf' SIZE 1M;

d)    Use DBA\_TABLESPACES view to display all tablespace names, their available status, can they store data or not, management mode, how the size of the extents is managed, their actual size, but only for freshly created tablespaces and 'USERS'.

SELECT TABLESPACE\_NAME,

STATUS,

EXTENT\_MANAGEMENT

FROM DBA\_TABLESPACES

WHERE TABLESPACE\_NAME IN ('USERS','BLUE','RED');

e)    Use DBA\_DATA\_FILES view to display file numbers, their names, tablespace names they belong to, current size (in blocks only) and their auto-extend information (with 3 components), but only for freshly created files and file from the 'USERS' tablespace.

SELECT FILE\_ID,

FILE\_NAME,

TABLESPACE\_NAME,

BLOCKS,

AUTOEXTENSIBLE

FROM DBA\_DATA\_FILES

WHERE TABLESPACE\_NAME IN ('USERS','BLUE','RED');

f)    Resize the second datafile in Tablespace RED to 2M and make it auto-extendable with extent size of 128K and with the limit of 4M.

ALTER DATABASE DATAFILE

'/opt/oracle/oradata/DISK3/red02.dbf' RESIZE 2M;

ALTER DATABASE DATAFILE

'/opt/oracle/oradata/DISK3/red02.dbf'

AUTOEXTEND ON NEXT 128K MAXSIZE 4M;

g)    Try to make the file in Tablespace BLUE auto-extendable with the extent size 2 times greater than it was the case with the original extent. Do not specify the limit. Was it successful? What will be the limit value in this case - check the appropriate view and show only that column for Tablespace BLUE.

ALTER DATABASE DATAFILE

'/opt/oracle/oradata/DISK4/blue01.dbf'

AUTOEXTEND ON NEXT 10M;

SELECT tablespace\_name, max\_extents

FROM dba\_tablespaces

WHERE tablespace\_name = 'BLUE';

h)    Try to limit auto-extension from the previous question to 10M without specifying the NEXT extent value. Was it successful?   
       What is now value of that (NEXT) parameter -- check the appropriate view and show only that column for Tablespace BLUE.

ALTER DATABASE DATAFILE

'/opt/oracle/oradata/DISK4/blue01.dbf'

AUTOEXTEND ON MAXSIZE 10M;

SELECT tablespace\_name,

increment\_by

FROM dba\_data\_files

WHERE tablespace\_name = 'BLUE';

i)    Move datafile **red02.dbf** to the DISK4 folder by using the appropriate scenario steps (when dealing with non-critical datafiles).

1. ALTER TABLESPACE RED OFFLINE;
2. HOST
3. [oracle@oracloud12c pfile]$ mv /opt/oracle/oradata/DISK2/red02.dbf /opt/oracle/oradata/DISK4
4. exit
5. ALTER TABLESPACE joke RENAME DATAFILE '/opt/oracle/oradata/DISK2/red02.dbf' TO'/opt/oracle/oradata/DISK4/red02.dbf' ;
6. ALTER TABLESPACE RED ONLINE;

j)    Repeat step e)

SELECT FILE\_ID,

FILE\_NAME,

TABLESPACE\_NAME,

BLOCKS,

AUTOEXTENSIBLE

FROM DBA\_DATA\_FILES

WHERE TABLESPACE\_NAME IN ('USERS','BLUE','RED');

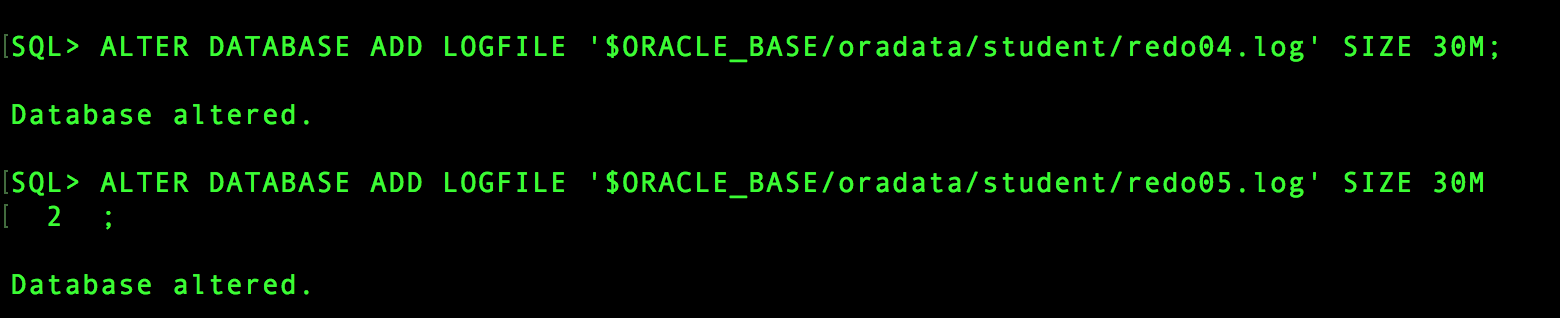
k) Display in DB Express current Tablespace / Datafile situation.

**Three:**

a) Create a fourth and fifth Log Group and place them in the same directory where the existing log groups are already. Size for both new groups should be 30M each.

ALTER DATABASE ADD LOGFILE '$ORACLE\_BASE/oradata/student/redo04.log' SIZE 30M

ALTER DATABASE ADD LOGFILE '$ORACLE\_BASE/oradata/student/redo04.log' SIZE 30M



b)  Create a second Log Member for each Log File (5 of them) in a subdirectory DISK3.

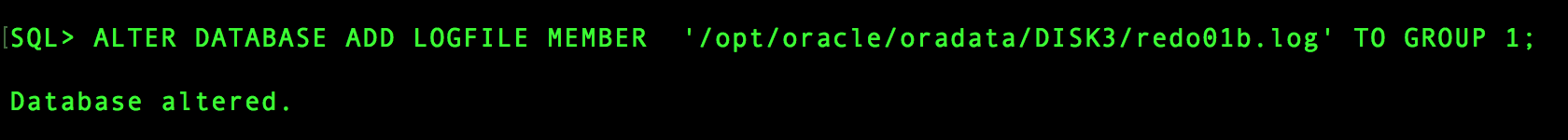
ALTER DATABASE ADD LOGFILE MEMBER'/opt/oracle/oradata/DISK3/redo01b.log' TO GROUP 1

ALTER DATABASE ADD LOGFILE MEMBER '/opt/oracle/oradata/DISK3/redo02b.log' TO GROUP 2

ALTER DATABASE ADD LOGFILE MEMBER '/opt/oracle/oradata/DISK3/redo03b.log' TO GROUP 3

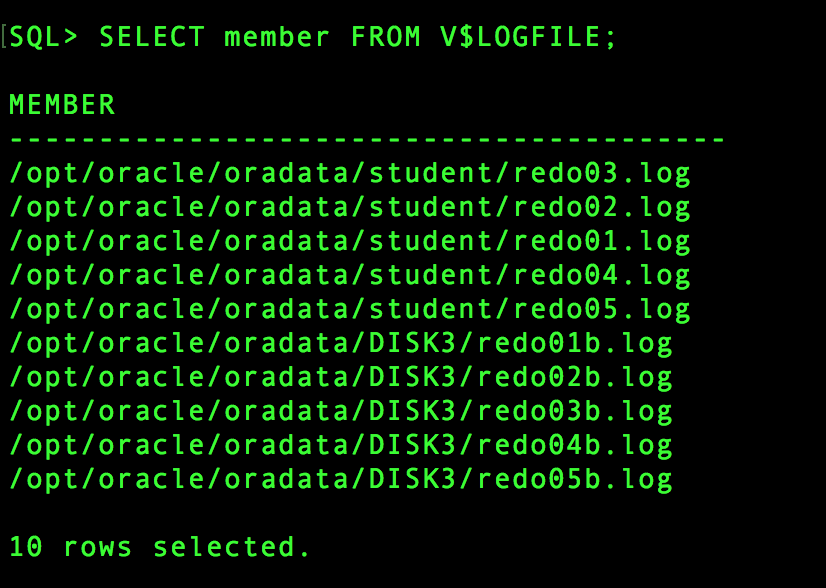
ALTER DATABASE ADD LOGFILE MEMBER '/opt/oracle/oradata/DISK3/redo04b.log' TO GROUP 4

ALTER DATABASE ADD LOGFILE MEMBER '/opt/oracle/oradata/DISK3/redo05b.log' TO GROUP 5



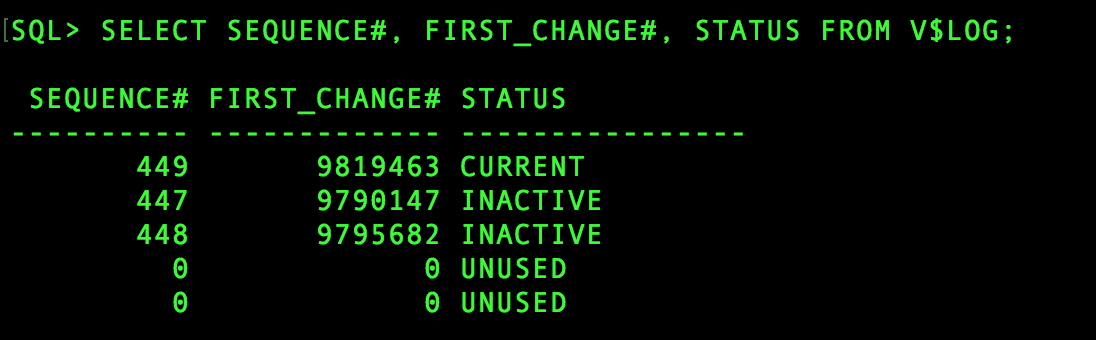
c)  Show the full paths and names for all Log Files by using an appropriate dynamic performance view.

SELECT member FROM V$LOGFILE;



d)  Show Log Sequence Number, First System Change Number and Status for all Log files by using an appropriate dynamic performance view.

SELECT SEQUENCE#, FIRST\_CHANGE#, STATUS FROM V$LOG;

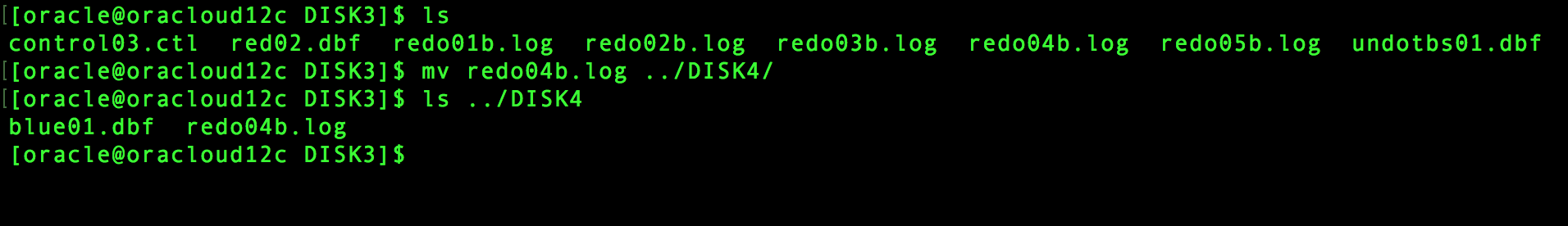


e)  Relocate (move)  just the second member of the fourth group to subdirectory DISK4.

1. Host

2. mv /opt/oracle/oradata/DISK3/redo04b.log /opt/oracle/oradata/DISK4/

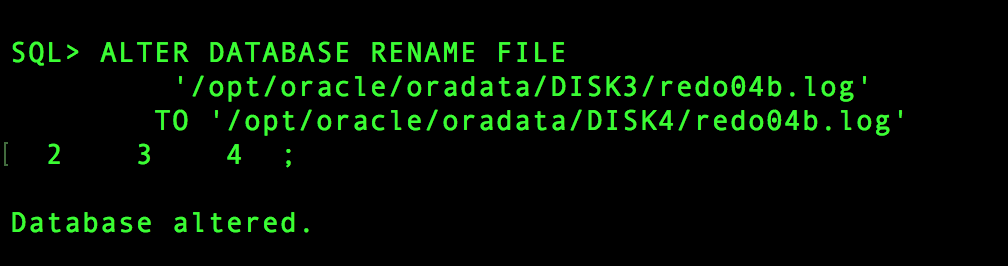
3. exit.



ALTER DATABASE RENAME FILE

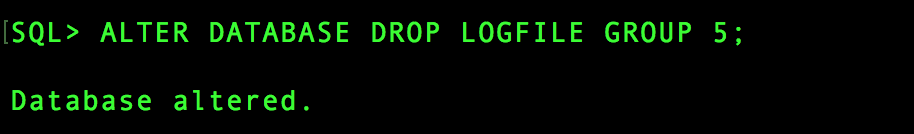
'/opt/oracle/oradata/DISK3/redo04b.log'

TO '/opt/oracle/oradata/DISK4/redo04b.log'

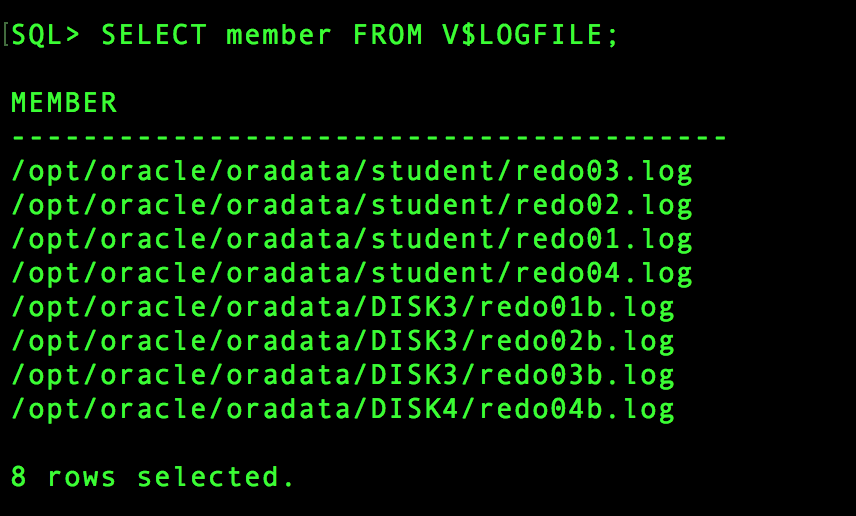


f)   Drop the fifth group completely. Then repeat step c)

ALTER DATABASE DROP LOGFILE GROUP 5;



SELECT member FROM V$LOGFILE;



g)  Perform two manual log switches. Then repeat step d)

h) Check the status of all Log Files, and if there is one or more ACTIVE ones, then perform manual command, so that all these active files become inactive. Then repeat step d)

**And do NOT forget to have a FUN!!!**