

**Systems and Database Administration –Journal Entry 4**

**DT211C**

**BSc in Computer Science**

**Tim Browning**

**C14519547**

School of Computing

Dublin Institute of Technology

****

Contents

[Introduction 3](#_Toc508570944)

[Auditing 3](#_Toc508570945)

[Partitioning 5](#_Toc508570946)

[Backup and Recovery 8](#_Toc508570947)

[Reflection and Conclusion 12](#_Toc508570948)

# Table of Figures

[Figure 1 - table\_audit show update and insert privs 4](#_Toc508570949)

[Figure 2 - FGA\_POLICY Query 5](#_Toc508570950)

[Figure 3 - partitions for table RPStudent 7](#_Toc508570951)

[Figure 4 - Before Commit 9](#_Toc508570952)

[Figure 5 - After Commit 9](#_Toc508570953)

[Figure 6 - log files 10](#_Toc508570954)

[Figure 7 – Inactive 11](#_Toc508570955)

[Figure 8 – Active 11](#_Toc508570956)

[Figure 9 - enabled archived log mode 11](#_Toc508570957)

[Figure 10 – logfiles 12](#_Toc508570958)

[Figure 11 - log files on system 12](#_Toc508570959)

# Introduction

This assignment will show Partitioning, Auditing and recovery within Oracle Linux. The commands are run in SQLDeveloper.

## Auditing

**Set up an audit trail that audits your database administrator**

The first step is to create a table with at least two columns, one of which has a value you can trace.

*“create table table\_audit (first\_name varchar(40), age number);”*

You will now have to login as SYSTEM. Set the AUDIT\_TRAIL instance parameter to enable auditing to the data dictionary.

“*alter system set audit\_trail = DB,EXTENDED scope = spfile*”

You will need to restart to apply changes.

To enable database auditing of select and update access to the table set up by USER1 enter:

“*Audit insert,update by user1.table\_audit*”

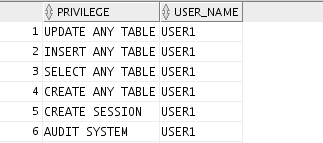


Figure 1 - table\_audit show update and insert privs

Execute dml statements:

*“Insert into table\_audit value (‘jack, 16);”*

The following should appear in the dba\_audit\_trail table.

*Select username, owner, timestamp, action\_name, priv\_used ,sql\_text from dba\_audit\_trail where username = ‘USER1’;*



I had a problem entering the username. I entered it in lowercase instead of uppercase causing an error, which outputted could not find user. Sqldeveloper requires all username fields to be in capitals.

**Set up fine-grained auditing on one of your users; Run some queries as the user and check the audit report**

The following code must be executed as a transaction.

BEGIN

DBMS\_FGA.ADD\_POLICY (

object\_schema => ‘USER1’,

object\_name => table\_audit,

policy\_name => 'mypolicy1',

audit\_column => 'age',

handler\_schema => NULL,

handler\_module => NULL,

enable => TRUE,

statement\_types => 'SELECT',

audit\_column\_opts => DBMS\_FGA.ANY\_COLUMNS,

policy\_owner => 'sec\_admin);

END;

To give user2 access – ‘*grant all on USER1.table\_audit to USER2;*’

Insert into user1 table from user2. Make sure to commit or the data will not save.

‘*insert into user1.table\_audit values (‘Nick’, 50);*’

‘*insert into user1.table\_audit values (‘luke, 76);*’

‘commit;’

Query the fine-grained audit-trail:

“*Select username, owner, timestamp, action, sql\_text from dba\_audit\_trail where username = ‘USER1’*”

This query will show you the username that performed the query, the owner of the table, the date and the SQL of the query.

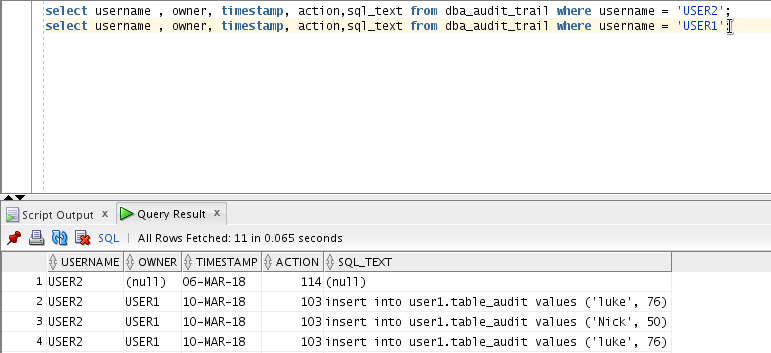


Figure 2 - FGA\_POLICY Query

To cancel audit execute the statement:

*“NOAUDIT SELECT on USER1.TABLE\_AUDIT;”*

This will stop Auditing of Queries on the table *“TABLE\_AUDIT”.*

To delete or drop a policy execute the following transaction

BEGIN

*DBMS\_FGA.DROP\_POLICY (*

*object\_schema => ‘USER1’,*

*object\_name => 'table\_audit',*

*policy\_name => 'mypolicy1');*

*END;*

## Partitioning

Download SetupTable.sql from webcourses and open it with SQldeveloper.

**Set up new tablespaces for your partitioned tables**

‘*create tablespace ts1 datafile ’/u01/app/oracle/oradata/tim/ts1 size 30m*;’`

‘*create tablespace ts2 datafile ’/u01/app/oracle/oradata/tim/ts2 size 30m*;’

‘*create tablespace ts3 datafile ’/u01/app/oracle/oradata/tim/ts3 size 30m*;’

‘*create tablespace ts4 datafile ’/u01/app/oracle/oradata/tim/ts4 size 30m*;’

‘*create tablespace ts5 datafile ’/u01/app/oracle/oradata/tim/ts5 size 30m*;’

To view tablespaces enter : “*select name from v$datafile;”*

**Set up a student table RPSTUDENT range partitioned on stage no**

I have created the following tables in USER3.

“*create table rpstudent*

*(*

*stage\_no integer not null,*

*prog\_code varchar2(5),*

*stage\_mentor varchar2(25),*

*remaining\_places integer,*

*primary key (prog\_code,stage\_no),*

*foreign key (prog\_code) references programme*

*)*

*partition by range (stage\_no)*

*(PARTITION part1 values less than (3)*

*tablespace ts1,*

*partition part2 values less than (MAXVALUE)*

*tablespace ts2);*2

This table is partitioned into two parts. The first part shows the rows with stage number less than 3 and the second part displays rows with values less than the max value.

Part1 has been allocated to tablespace ts1

Part2 has been allocated to tablespace ts2

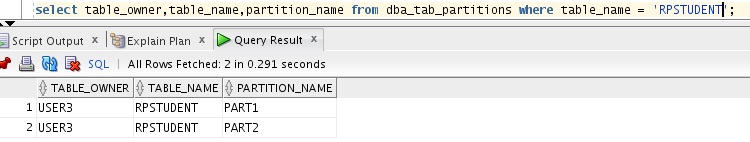


Figure 3 - partitions for table RPStudent

**Set up a table HSTUDENT hash partitioned on student no.**

“*create table HSTUDENT*

*(*

*s\_studentNo varchar2(9) not null,*

*prog\_code varchar2(5) not null,*

*stage\_no integer not null,*

*studentname varchar2(40),*

*studentAddress varchar2(60),*

*primary key (s\_studentNo),*

*foreign key (prog\_code,stage\_no) references rpstudent(prog\_code,stage\_no)*

*)*

*partition by hash (s\_studentNo)*

*(partition part1 tablespace ts3,*

*partition part2 tablespace ts4);”*

The hash key is used to distribute rows evenly across the different partitions

**Set up a table LSTUDENT list partitioned on programme code**

*“Create table LSTUDENT*

*(*

*prog\_code varchar2(5) not null,*

*prog\_name varchar2(34),*

*course\_chairperson varchar2(25),*

*total\_s\_students integer default 0,*

*primary key (prog\_code)*

*)*

*partition by list (prog\_code)*

*(partition part1 values ('DT211','DT228','DT249')*

*tablespace ts5);*

*--*

*insert into LSTUDENT values ('DT228','Degree in Computer Science','SarahJane Delany',200);*

*insert into LSTUDENT values ('DT211','Degree in Computing','Ken O''Brien',100);*

*insert into LSTUDENT values ('DT249','Degree in Information Technology','Ciaran O''Leary',100);”*

You should use list partitioning when you want to specifically map rows to partitions based on discrete values. In this case the table LSTUDENT maps the rows with the values 'DT211','DT228','DT249'.

**Set up a range-hash partitioned table RHSTUDENT, partitioned by range on Stage-no and sub-partitioned by hash on student no.**

I have changed the HStudent table to RHStudent to show partitioning on stageno and sub-partitioning by hash on student no.

“create table RHSTUDENT

(

s\_studentNo varchar2(9) not null,

prog\_code varchar2(5) not null,

stage\_no integer not null,

studentname varchar2(40),

studentAddress varchar2(60),

primary key (s\_studentNo),

foreign key (prog\_code,stage\_no) references rpstudent(prog\_code,stage\_no)

)

partition by range (stage\_no)

subpartition by hash (s\_studentNo)

(partition part1 values less than (3)

tablespace ts1,

partition part2 values less than (MAXVALUE)

tablespace ts2);”

**Set up a range-list partitioned table RLstudent partitioned by stage-no and sub-partitioned by programme code.**

“*create table RLSTUDENT*

*(*

*s\_studentNo varchar2(9) not null,*

*prog\_code varchar2(5) not null,*

*stage\_no integer not null,*

*studentname varchar2(40),*

*studentAddress varchar2(60),*

*primary key (s\_studentNo),*

*foreign key (prog\_code,stage\_no) references rpstudent(prog\_code,stage\_no)*

*)*

*partition by range (stage\_no)*

*subpartition by hash (prog\_code)*

*(partition part1 values less than (3)*

*tablespace ts1,*

*partition part2 values less than (MAXVALUE)*

*tablespace ts2);”*

## Backup and Recovery

**Instance Recovery & MTTR**

Set FAST\_START\_MTTR\_TARGET = 0 to disable checkpoint tuning.

Select \* from v$instance\_recovery;

Set table that has two parameters “*first\_name”* and “age”.

Execute a transaction:

*“BEGIN*

*Insert into table1 values (4,’’tim’,21);*

*END;”*

To view the recovery table enter “*select \* from v$instance\_recovery*;”. This monitors the mechanisms available to users to limit recovery I/O. The RECOVERY\_ESTIMATED\_IOS, ACTUAL\_REDO\_BLKS and TARGET\_REDO\_BLKS have higher values. A higher RECOVERY\_ESTIMATED\_IOS means that there is more dirty buffers in the buffer cache. A higher ACTUAL\_REDO\_BLKS means there is more redo blocks for recovery and the TARGET\_REDO\_BLKS are the target number of redo blocks that must be processed for recovery



Figure 4 - Before Commit

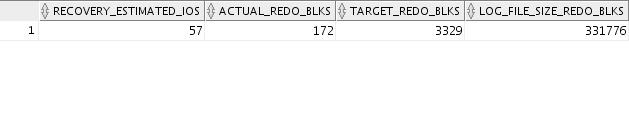
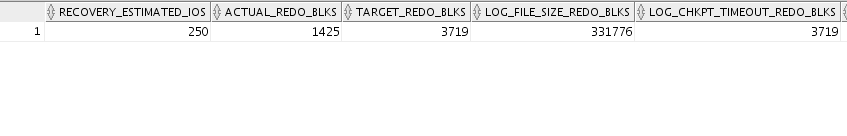


Figure 5 - After Commit

“*Create restore point point1*” – This takes a few seconds to complete

Do another enter “*select \* from v$instance\_recovery*;”



There is a significant increase in RECOVERY\_ESTIMATED\_IOS, ACTUAL\_REDO\_BLKS and TARGET\_REDO\_BLKS. This is because there is an increase in redo blocks. (More redo writes to recover)

“*Drop table table1*”

**Multiplexing the Redo Log**

“select \* from v$log;” – displays the current and inactive logs

“select \* from v$logfile;” – Displays the file location of the redo logs

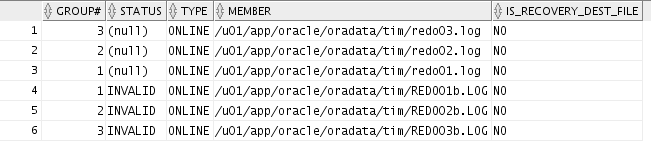


Figure 6 - log files

“*ALTER DATABASE ADD LOGFILE MEMBER /u01/app/oracle/oradata/tim/redo01b.LOG TO GROUP 1;”*

This command add a group member

Querying the groups shows that the group members have changed from 1 to 2. The new log files have an invalid status.

*“ALTER SYSTEM SWITCH LOGFILE;” – switches logfile*

When switched three times the status changes to active and the sequence numbers change.

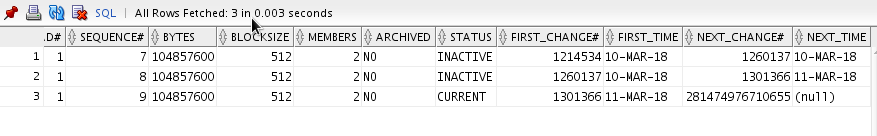


Figure 7 – Inactive

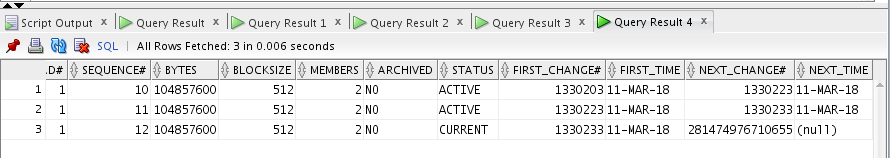


Figure 8 – Active

**Archivelog Mode**

Create two directories: /u01/app/oracle/archive1 and /u01/app/oracle/archive2.

Go to the folder pfile and add:

log\_archive\_dest\_1='location=/u01/app/oracle/Archive1 '

log\_archive\_start=TRUE

shutdown immediate

startup mount

alter database archivelog;

archive log list;

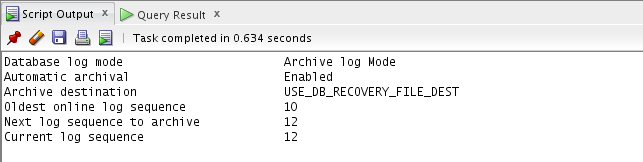


Figure 9 - enabled archived log mode

*“select \* from archived\_log;”*

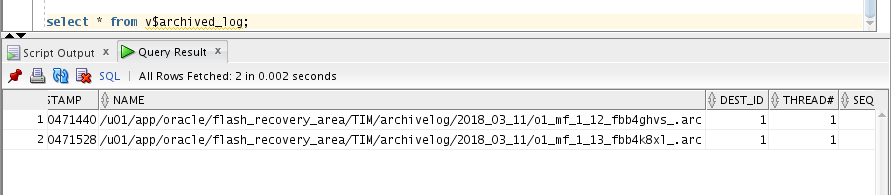


Figure 10 – logfiles

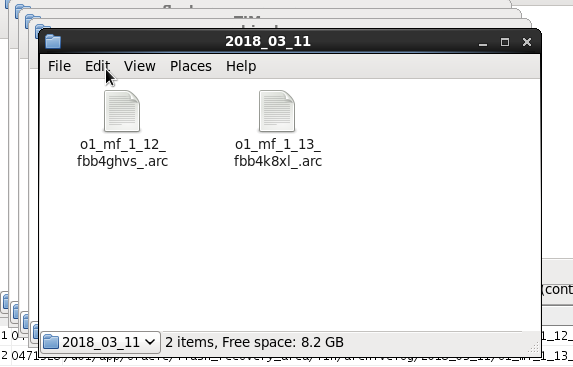


Figure 11 - log files on system

# Reflection and Conclusion

The partitioning part to this assignment was challenging, as it required a lot of research. Luckily, I found a YouTube video that demonstrated how to partition correctly.

<https://www.youtube.com/watch?v=m3q4lrE671Y>. I learnt about list, hash and range partitioning. This video showed a real project example.

A problem occurred where I could not access my database because it was in the process of shutting down. This occurred because I entered the shutdown command in SQLdeveloper. To fix this problem I opened command prompt, entered the command shutdown immediate, and then reopened the database.

My host computer ran out of battery, which corrupted the database. To fix this I ran dbca and reinstalled the database.