L-1 IDENTITY SOLUTIONS

RMAN-Backup and Recovery
The Most Essential But The Most Ignored

Asif Momen

Senior Oracle DBA

Who am I?

- Oracle ACE
- 12+ years of experience
- Member, Editorial Board, Oracle Connect, All India Oracle User Group (AIOUG)
- Oracle Certified Professional (OCP) 10g & 9i – Database & Oracle Forms
- M.S. (Software Systems) from BITS – Pilani, India
- Industries worked on: Education, Banking and High Technology
- Oracle Blogger at: http://momendba.blogspot.com

Agenda

- **What Survey has to say about Backup & Recovery**
- **Horror Stories**
- **Test Environment**
- **Recovery Scenario’s**
  - Loss of Parameter File (PFILE/SPFILE)
  - Loss of Control Files
  - Loss of Redo Log Files
  - Basic Recovery Solutions
  - Advanced Recovery Solutions
  - Unsupported Recovery Solutions
  - Data Unloader
- **References**

What Survey has to say about Backup & Recovery

- According to Symantec Survey, the average SMB only backs up 60% of customer data and doesn't even do that on a consistent schedule

- Research conducted by Acronis reveals 38% of UK SMEs never backup their data. Full story is here.

Horror Stories

- Ignacio Ruiz has recently put up a case where a customer was running a database with NO backups and a disaster hit them. Here is the complete story

- Disaster Recovery Stories by Alejandro Vargas

- OTN Forums is the other good place

Test Environment

- **Operating System**
  - Red Hat Enterprise Linux AS release 4 (Nahant Update 2)
  - Windows XP SP-2
- **Database**
  - Oracle Database 11g Release 2 (11.2.0.1)
  - Oracle Database 10g Release 2 Patch 3 (10.2.0.4)
  - Data Block Size – 8K
  - Archive Log Mode

Loss of Parameter File (PFILE/SPFILE)

Loss of Parameter File Scenario' s

- SPF1: Loss of PFILE/SPFILE When No Backup Exists
- SPF2: Restore SPFILE From Autobackup
- SPF3: Restore SPFILE From Autobackup in FRA
- SPF4: Restore SPFILE Using Recovery Catalog

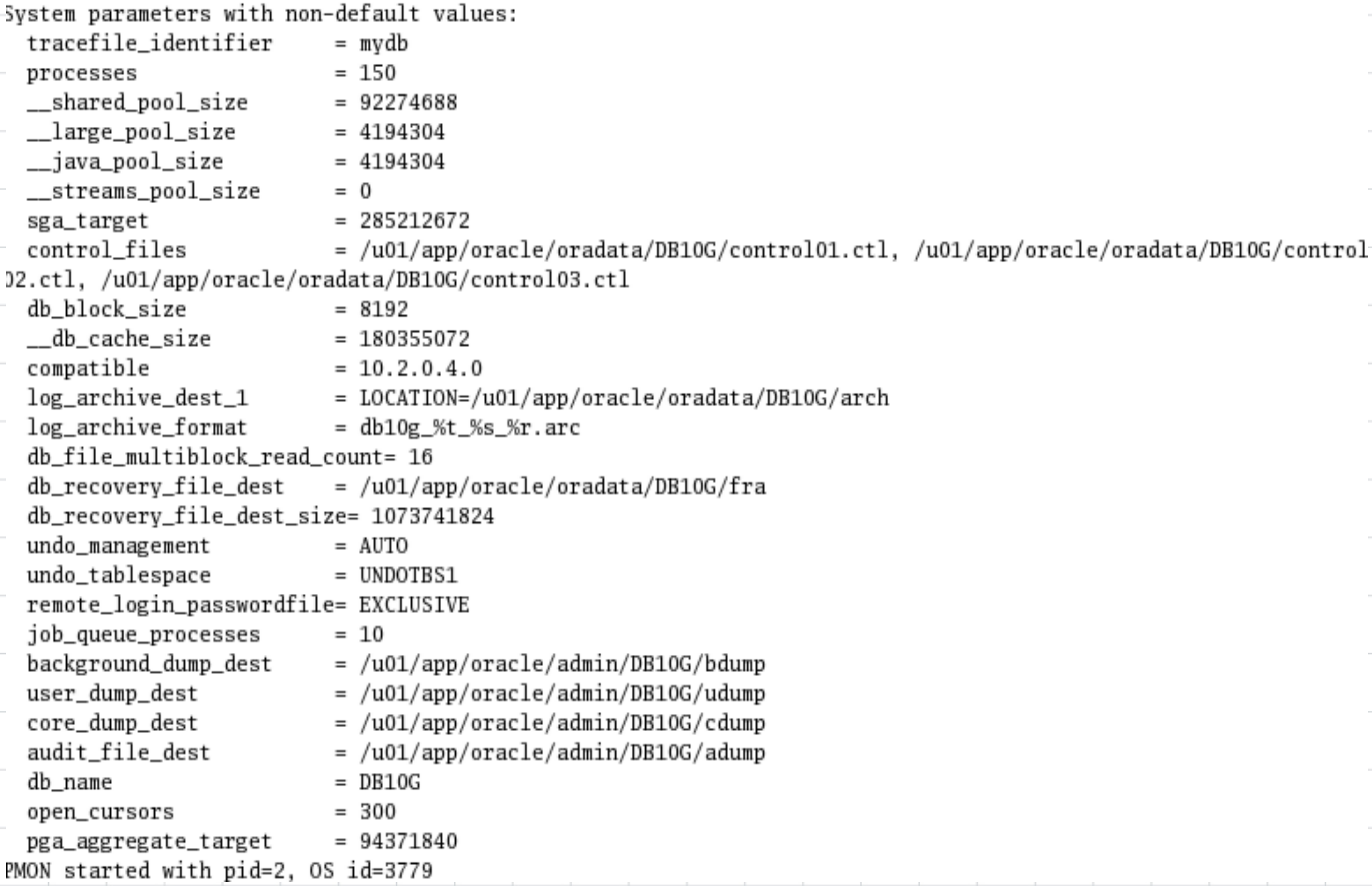
SPF1: Loss of PFILE/SPFILE When No Backup Exists

- **Problem:**
  - Loss of PFILE/SPFILE
  - You don’t have PFILE/SPFILE backup

- **Solution(s):**
  1) Check database alert.log bottom-up
  2) When database is UP
     1) PFILE → Query V$PARAMETER2 view
     2) SPFILE → Query V$SPPARAMETER
  3) Create PFILE/SPFILE from memory (Oracle 11g)

SPF1: Loss of PFILE/SPFILE When No Backup Exists

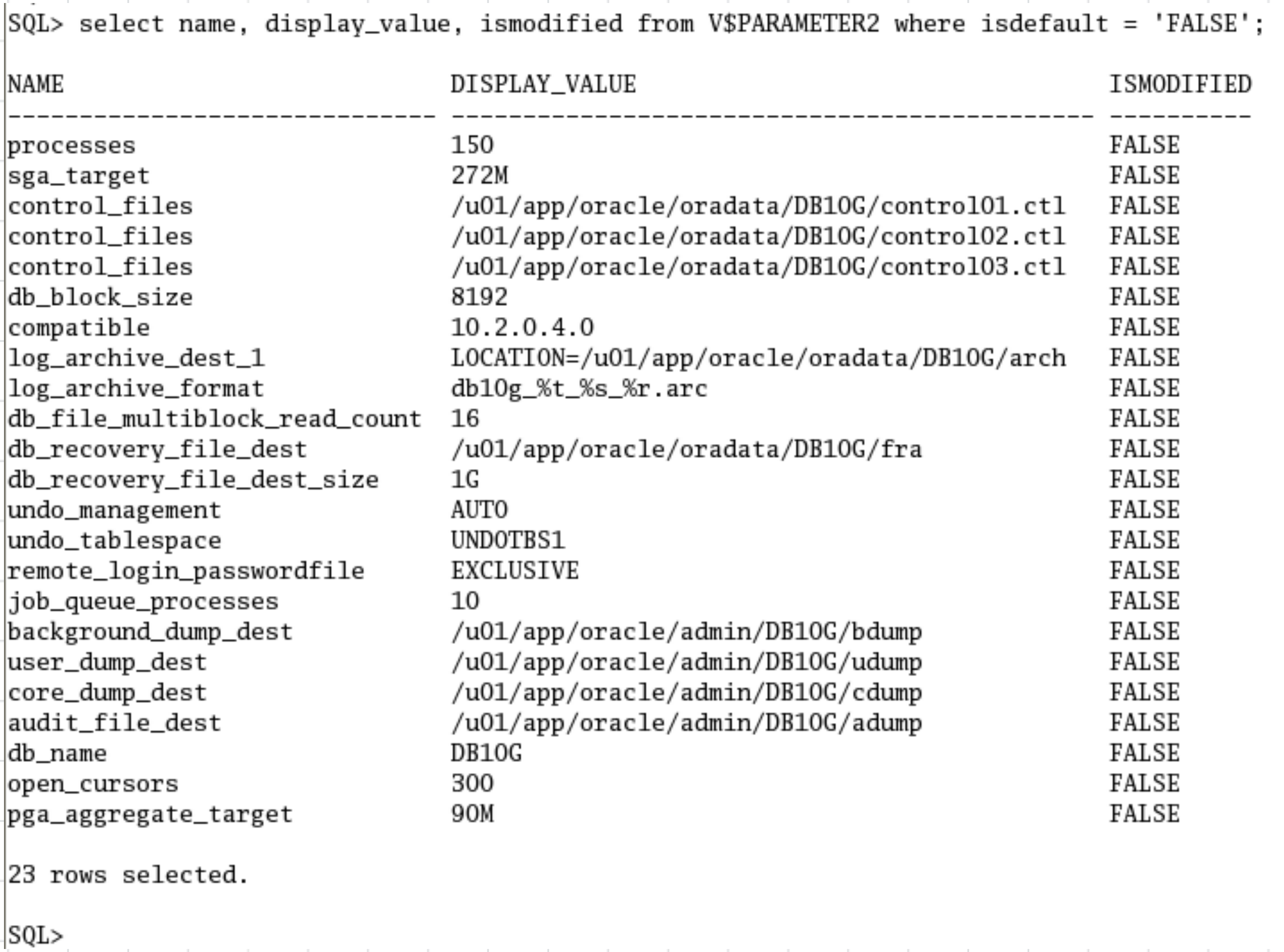
1) When Oracle instance starts, all non-default parameters are recorded in the database alert.log



```
System parameters with non-default values:
  tracefile_identifier     = mydb
  processes                = 150
  __shared_pool_size       = 92274688
  __large_pool_size        = 4194304
  __java_pool_size         = 4194304
  __streams_pool_size      = 0
  sga_target               = 285212672
  control_files            = /u01/app/oracle/oradata/DB10G/control01.ctl, /u01/app/oracle/oradata/DB10G/control
02.ctl, /u01/app/oracle/oradata/DB10G/control03.ctl
  db_block_size            = 8192
  __db_cache_size          = 180355072
  compatible               = 10.2.0.4.0
  log_archive_dest_1       = LOCATION=/u01/app/oracle/oradata/DB10G/arch
  log_archive_format       = db10g_%t_%s_%r.arc
  db_file_multiblock_read_count= 16
  db_recovery_file_dest    = /u01/app/oracle/oradata/DB10G/fra
  db_recovery_file_dest_size= 1073741824
  undo_management          = AUTO
  undo_tablespace          = UNDOTBS1
  remote_login_passwordfile= EXCLUSIVE
  job_queue_processes      = 10
  background_dump_dest     = /u01/app/oracle/admin/DB10G/bdump
  user_dump_dest           = /u01/app/oracle/admin/DB10G/udump
  core_dump_dest           = /u01/app/oracle/admin/DB10G/cdump
  audit_file_dest          = /u01/app/oracle/admin/DB10G/adump
  db_name                  = DB10G
  open_cursors             = 300
  pga_aggregate_target     = 94371840
PMON started with pid=2, OS id=3779
```

SPF1: Loss of PFILE/SPFILE When No Backup Exists

2) Output from V$PARAMETER2



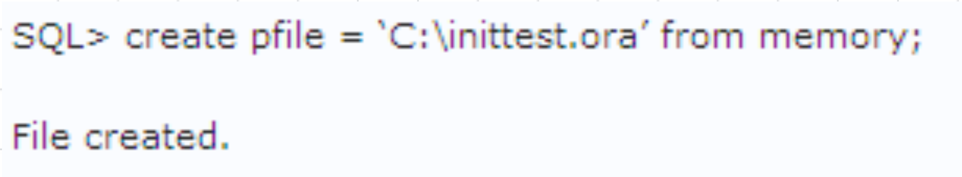
```
SQL> select name, display_value, ismodified from V$PARAMETER2 where isdefault = 'FALSE';

NAME                           DISPLAY_VALUE                                 ISMODIFIED
------------------------------ --------------------------------------------- ----------
processes                      150                                           FALSE
sga_target                     272M                                          FALSE
control_files                  /u01/app/oracle/oradata/DB10G/control01.ctl   FALSE
control_files                  /u01/app/oracle/oradata/DB10G/control02.ctl   FALSE
control_files                  /u01/app/oracle/oradata/DB10G/control03.ctl   FALSE
db_block_size                  8192                                          FALSE
compatible                     10.2.0.4.0                                    FALSE
log_archive_dest_1             LOCATION=/u01/app/oracle/oradata/DB10G/arch   FALSE
log_archive_format             db10g_%t_%s_%r.arc                            FALSE
db_file_multiblock_read_count  16                                            FALSE
db_recovery_file_dest          /u01/app/oracle/oradata/DB10G/fra             FALSE
db_recovery_file_dest_size     1G                                            FALSE
undo_management                AUTO                                          FALSE
undo_tablespace                UNDOTBS1                                      FALSE
remote_login_passwordfile      EXCLUSIVE                                     FALSE
job_queue_processes            10                                            FALSE
background_dump_dest           /u01/app/oracle/admin/DB10G/bdump             FALSE
user_dump_dest                 /u01/app/oracle/admin/DB10G/udump             FALSE
core_dump_dest                 /u01/app/oracle/admin/DB10G/cdump             FALSE
audit_file_dest                /u01/app/oracle/admin/DB10G/adump             FALSE
db_name                        DB10G                                         FALSE
open_cursors                   300                                           FALSE
pga_aggregate_target           90M                                           FALSE

23 rows selected.

SQL>
```

SPF1: Loss of PFILE/SPFILE When No Backup Exists

- Revert all the modified parameters to original values
- Construct a PFILE
- Create a SPFILE from the PFILE
- Backup the recovered PFILE/SPFILE

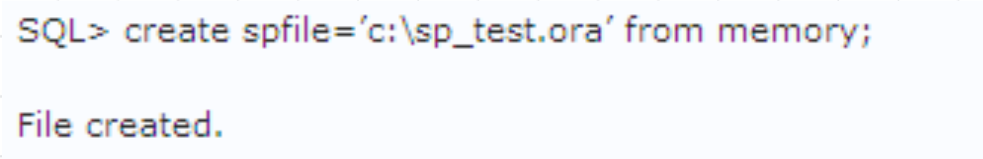
SPF1: Loss of PFILE/SPFILE When No Backup Exists

3) **Create PFILE/SPFILE from memory (Oracle 11g)**
   - PFILE



```
SQL> create pfile = ‘C:\inittest.ora’ from memory;

File created.
```

   - SPFILE



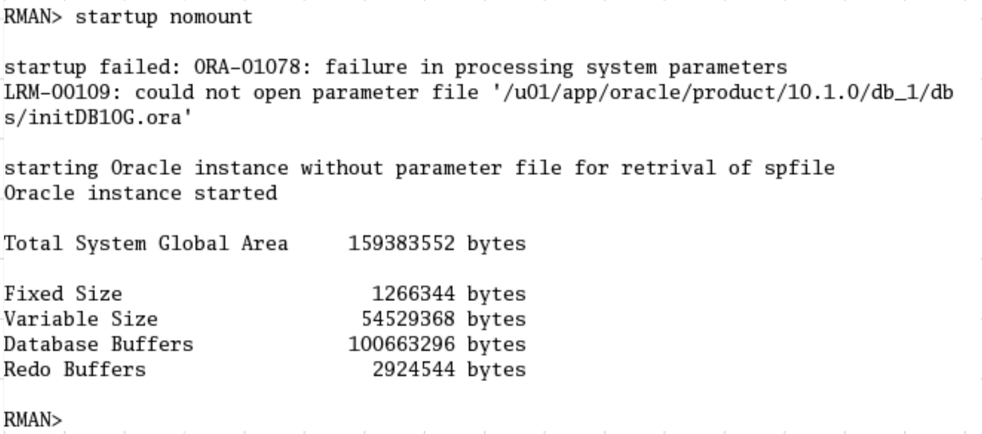
```
SQL> create spfile=’c:\sp_test.ora’ from memory;

File created.
```

SPF2: Restoring SPFILE From Autobackup

- **Problem:**
  - Database is DOWN
  - SPFILE is lost
  - Autobackup is configured
- **Solution:**
  - Start database instance in NOMOUNT mode without a parameter file
  - Set DBID
  - Restore SPFILE from Autobackup

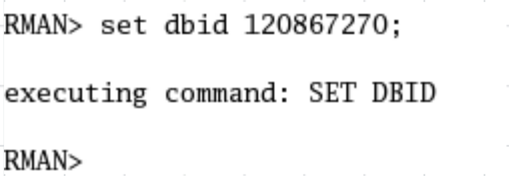
SPF2: Restoring SPFILE From Autobackup

- **Start database instance in NOMOUNT mode without a parameter file**



```
RMAN> startup nomount

startup failed: ORA-01078: failure in processing system parameters
LRM-00109: could not open parameter file '/u01/app/oracle/product/10.1.0/db_1/db
s/initDB10G.ora'

starting Oracle instance without parameter file for retrival of spfile
Oracle instance started

Total System Global Area     159383552 bytes

Fixed Size                     1266344 bytes
Variable Size                 54529368 bytes
Database Buffers             100663296 bytes
Redo Buffers                   2924544 bytes

RMAN>
```

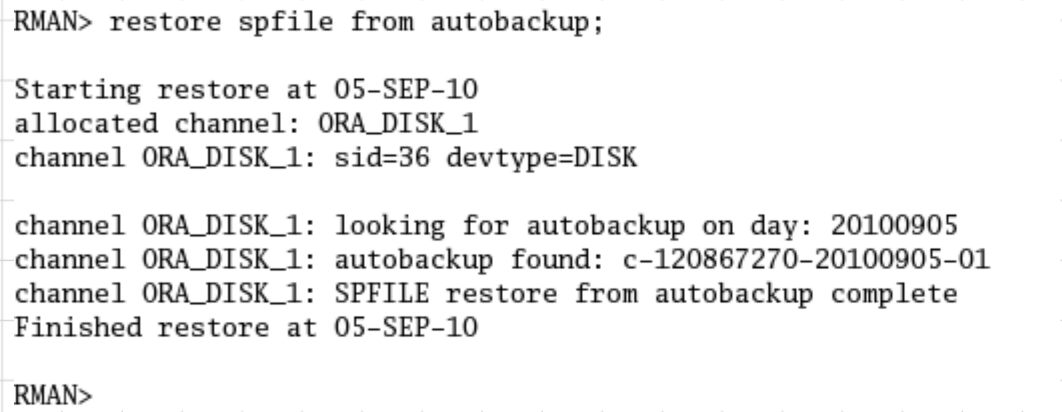
- **Set DBID**



```
RMAN> set dbid 120867270;

executing command: SET DBID

RMAN>
```

SPF2: Restoring SPFILE From Autobackup

- **Restore SPFILE from Autobackup**



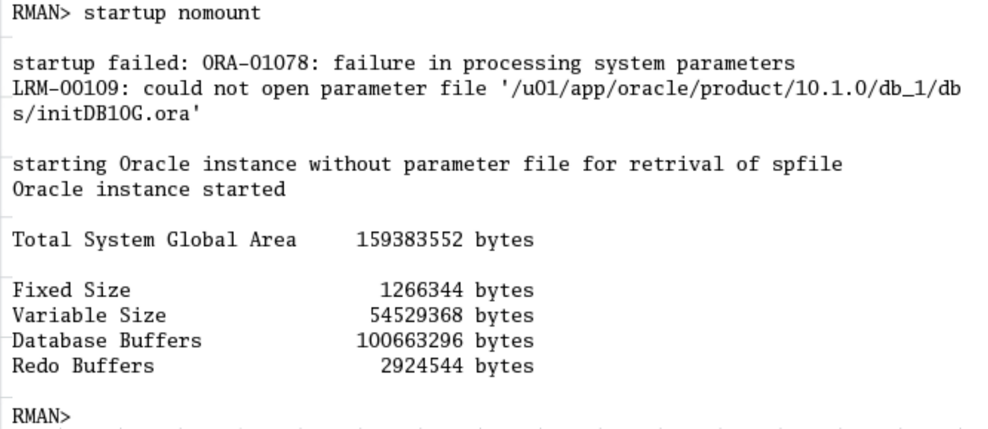
```
RMAN> restore spfile from autobackup;

Starting restore at 05-SEP-10
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=36 devtype=DISK

channel ORA_DISK_1: looking for autobackup on day: 20100905
channel ORA_DISK_1: autobackup found: c-120867270-20100905-01
channel ORA_DISK_1: SPFILE restore from autobackup complete
Finished restore at 05-SEP-10

RMAN>
```

SPF3: Restoring SPFILE From FRA

- **Problem:**
  - Database is DOWN
  - SPFILE is lost
  - FRA is enabled
- **Solution:**
  - Start database instance in NOMOUNT mode without a parameter file
  - Restore SPFILE from FRA

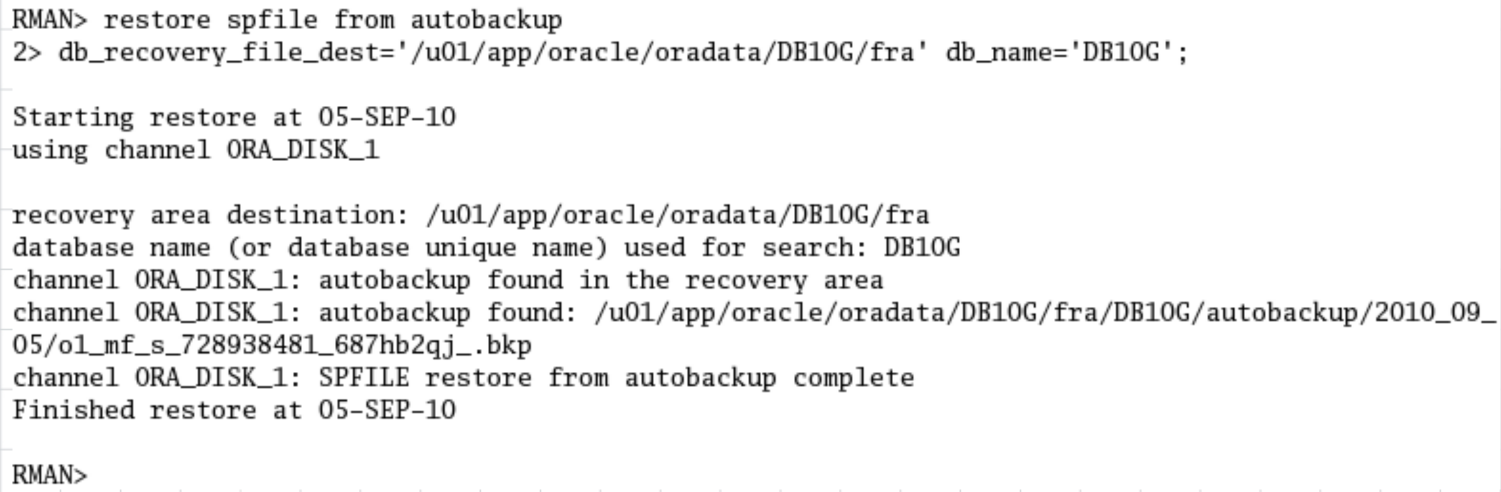
SPF3: Restoring SPFILE From FRA

- **Start database instance in NOMOUNT mode without a parameter file**



```
RMAN> startup nomount

startup failed: ORA-01078: failure in processing system parameters
LRM-00109: could not open parameter file '/u01/app/oracle/product/10.1.0/db_1/db
s/initDB10G.ora'

starting Oracle instance without parameter file for retrival of spfile
Oracle instance started

Total System Global Area     159383552 bytes

Fixed Size                     1266344 bytes
Variable Size                 54529368 bytes
Database Buffers             100663296 bytes
Redo Buffers                   2924544 bytes

RMAN>
```

- **Restore SPFILE from FRA**



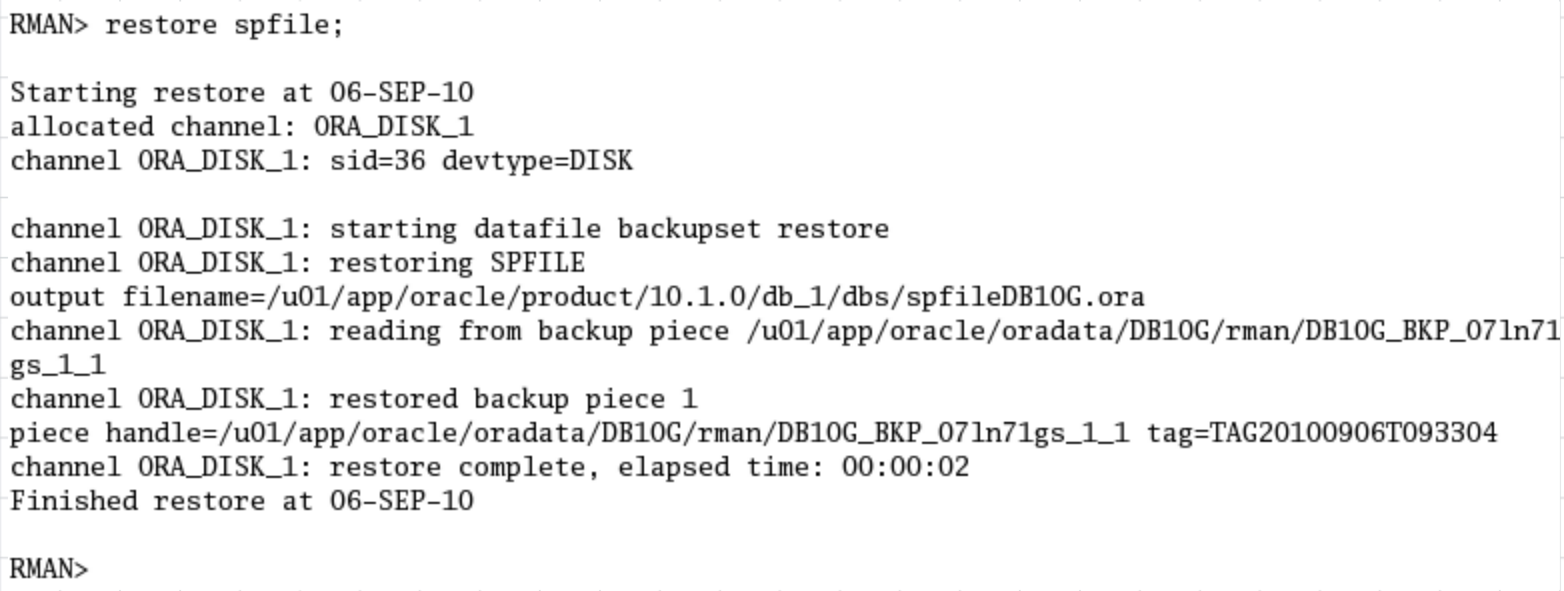
```
RMAN> restore spfile from autobackup
2> db_recovery_file_dest='/u01/app/oracle/oradata/DB10G/fra' db_name='DB10G';

Starting restore at 05-SEP-10
using channel ORA_DISK_1

recovery area destination: /u01/app/oracle/oradata/DB10G/fra
database name (or database unique name) used for search: DB10G
channel ORA_DISK_1: autobackup found in the recovery area
channel ORA_DISK_1: autobackup found: /u01/app/oracle/oradata/DB10G/fra/DB10G/autobackup/2010_09_
05/o1_mf_s_728938481_687hb2qj_.bkp
channel ORA_DISK_1: SPFILE restore from autobackup complete
Finished restore at 05-SEP-10

RMAN>
```

SPF4: Restore SPFILE Using Recovery Catalog

- **Problem:**
  - Database is DOWN
  - SPFILE is lost
  - Recovery Catalog Database is configured
- **Solution:**
  - Start database instance in NOMOUNT mode without a parameter file
  - Restore SPFILE

SPF4: Restore SPFILE Using Recovery Catalog

- **Restore SPFILE**



```
RMAN> restore spfile;

Starting restore at 06-SEP-10
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=36 devtype=DISK

channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: restoring SPFILE
output filename=/u01/app/oracle/product/10.1.0/db_1/dbs/spfileDB10G.ora
channel ORA_DISK_1: reading from backup piece /u01/app/oracle/oradata/DB10G/rman/DB10G_BKP_07ln71
gs_1_1
channel ORA_DISK_1: restored backup piece 1
piece handle=/u01/app/oracle/oradata/DB10G/rman/DB10G_BKP_07ln71gs_1_1 tag=TAG20100906T093304
channel ORA_DISK_1: restore complete, elapsed time: 00:00:02
Finished restore at 06-SEP-10

RMAN>
```

Loss of Control File

Loss of Control File

- CF1: Loosing one of the Multiplexed Control File’s
- CF2: Loosing all Multiplexed Control Files (without a backup)
- CF3: Restoring Control File From Autobackup
- CF4: Restore Control File From Recovery Catalog
- CF5: Restore Control File When FRA is Configured

CF1: Loosing one of the Multiplexed Control Files

- **Problem:**
  - Database is up
  - One of the multiplexed control file is lost
- **Solution(s):**

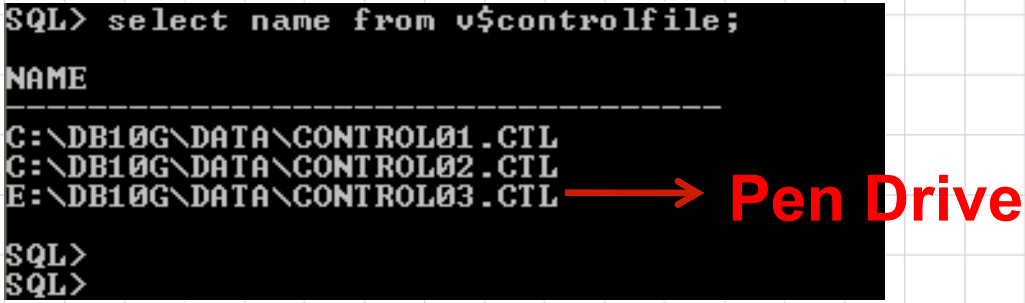
  1) Copy a good control file to the location of the missing control file

  2) Remove references to the missing control file from CONTROL_FILES initialization parameter

CF1: Loosing one of the Multiplexed Control Files

Simulating the problem:

- Place one of the Control Files on a Pen Drive



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

NAME
-----------------------------------
C:\DB10G\DATA\CONTROL01.CTL
C:\DB10G\DATA\CONTROL02.CTL
E:\DB10G\DATA\CONTROL03.CTL

SQL>
SQL>
```

Pen Drive

- Remove the pen drive

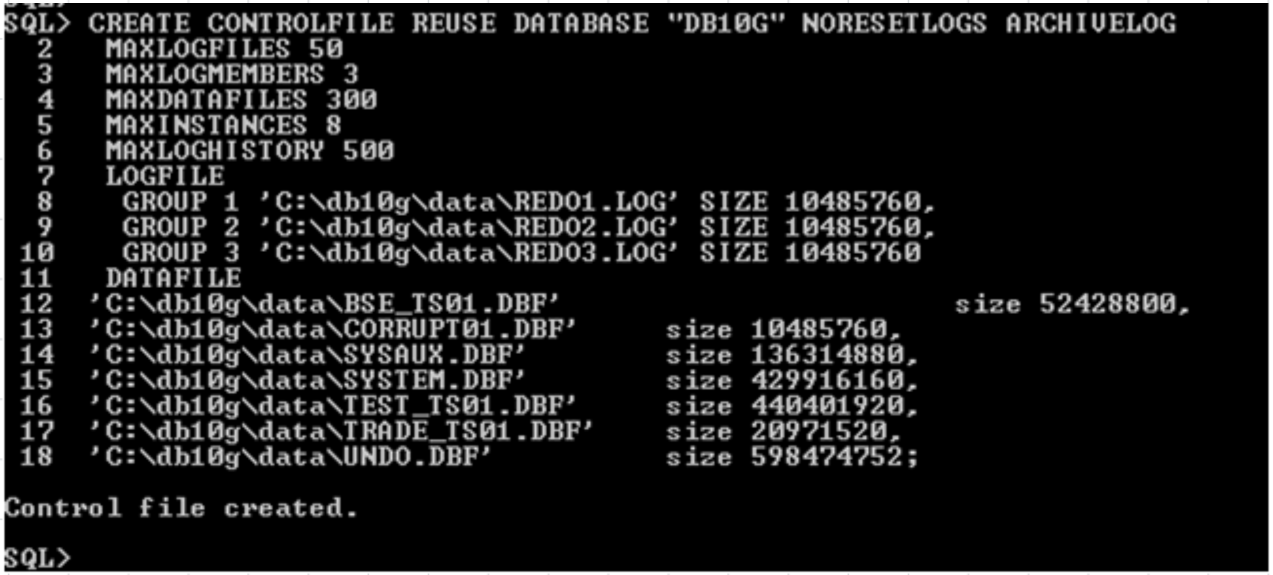
```
Errors in file c:\db10g\dump\db10g_ckpt_5836.trc:
ORA-00206: error in writing (block 3, # blocks 1) of control file
ORA-00202: control file: 'E:\DB10G\DATA\CONTROL03.CTL'
ORA-27072: File I/O error
OSD-04008: WriteFile() failure, unable to write to file
O/S-Error: (OS 1006) The volume for a file has been externally altered so that the opened file is no longer valid.
```

CF2: Loosing all Multiplexed Control Files (without a backup)

- **Problem:**
  - All Control Files are lost
  - No backup exists
- **Solution:**
  - Startup database instance in NOMOUNT mode
  - Create a new Control File
  - Open the database

CF2: Loosing all Multiplexed Control Files (without a backup)

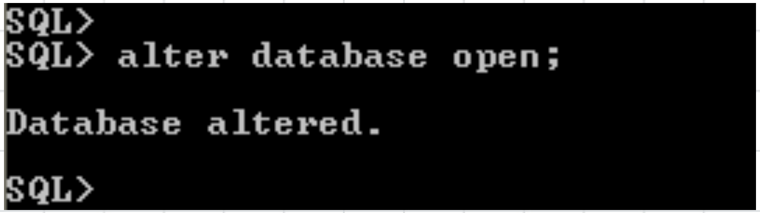
- **Creating a new Control File**
  - Startup database instance in NOMOUNT mode
  - Use NORESETLOGS option as online redo log files are still good



```
SQL> CREATE CONTROLFILE REUSE DATABASE "DB10G" NORESETLOGS ARCHIVELOG
  2    MAXLOGFILES 50
  3    MAXLOGMEMBERS 3
  4    MAXDATAFILES 300
  5    MAXINSTANCES 8
  6    MAXLOGHISTORY 500
  7    LOGFILE
  8     GROUP 1 'C:\db10g\data\REDO1.LOG' SIZE 10485760,
  9     GROUP 2 'C:\db10g\data\REDO2.LOG' SIZE 10485760,
 10     GROUP 3 'C:\db10g\data\REDO3.LOG' SIZE 10485760
 11    DATAFILE
 12   'C:\db10g\data\BSE_TS01.DBF'                          size 52428800,
 13   'C:\db10g\data\CORRUPT01.DBF'     size 10485760,
 14   'C:\db10g\data\SYSAUX.DBF'        size 136314880,
 15   'C:\db10g\data\SYSTEM.DBF'        size 429916160,
 16   'C:\db10g\data\TEST_TS01.DBF'     size 440401920,
 17   'C:\db10g\data\TRADE_TS01.DBF'    size 20971520,
 18   'C:\db10g\data\UNDO.DBF'          size 598474752;

Control file created.

SQL>
```

CF2: Loosing all Multiplexed Control Files (without a backup)

- Open Database



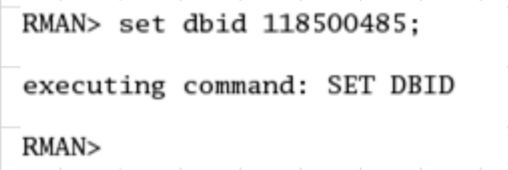
```
SQL>
SQL> alter database open;

Database altered.

SQL>
```

CF3: Restoring Control File From Autobackup

- **Problem:**
  - All Control Files are lost
  - Autobackup is configured
- **Solution:**
  - Start database instance in NOMOUNT mode
  - Set DBID in RMAN
  - Restore control file from autobackup
  - MOUNT the database
  - Recover database
  - Open the database with RESETLOGS option

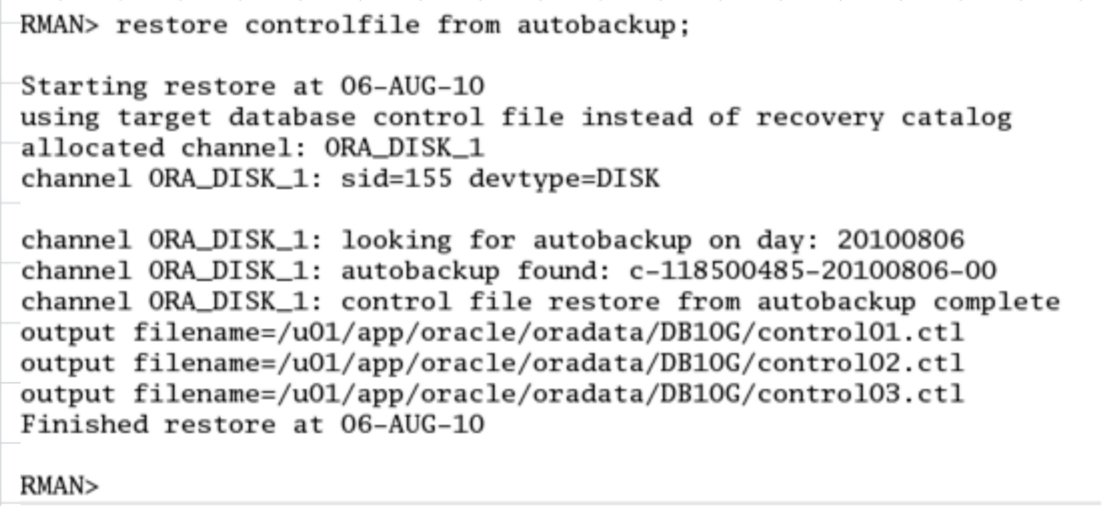
CF3: Restoring Control File From Autobackup

- Start database instance in NOMOUNT mode
- Set DBID in RMAN



```
RMAN> set dbid 118500485;

executing command: SET DBID

RMAN>
```

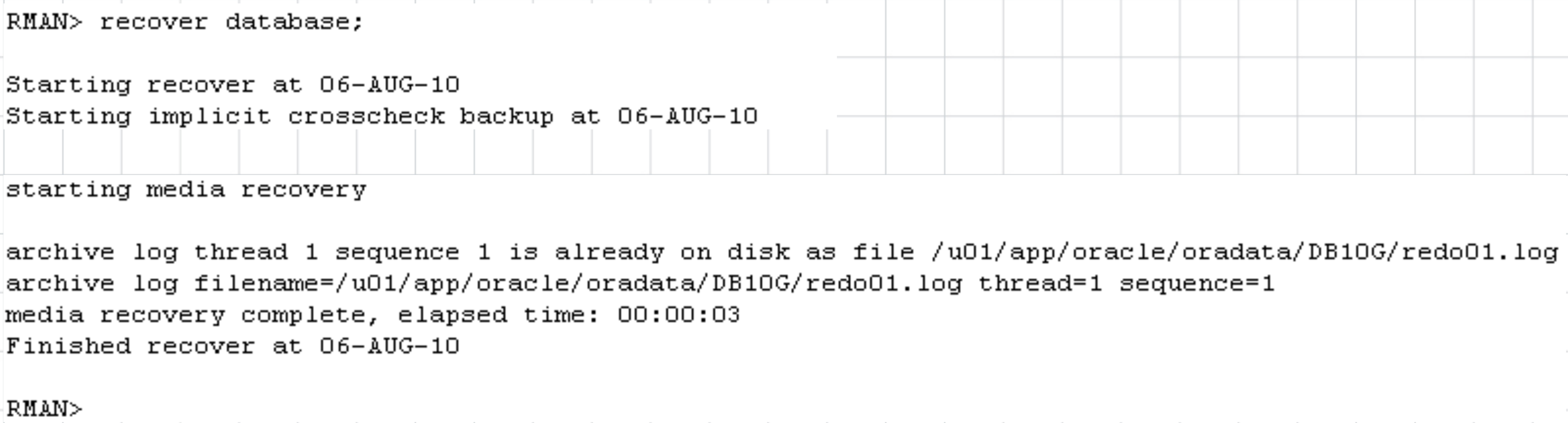
- Restore Control File



```
RMAN> restore controlfile from autobackup;

Starting restore at 06-AUG-10
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=155 devtype=DISK

channel ORA_DISK_1: looking for autobackup on day: 20100806
channel ORA_DISK_1: autobackup found: c-118500485-20100806-00
channel ORA_DISK_1: control file restore from autobackup complete
output filename=/u01/app/oracle/oradata/DB10G/control01.ctl
output filename=/u01/app/oracle/oradata/DB10G/control02.ctl
output filename=/u01/app/oracle/oradata/DB10G/control03.ctl
Finished restore at 06-AUG-10

RMAN>
```

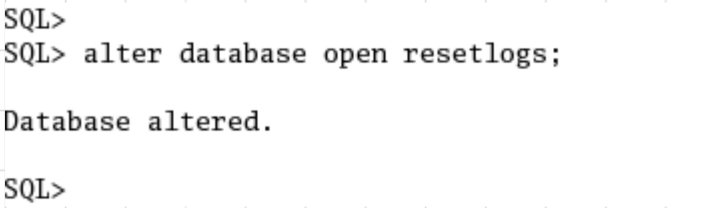
CF3: Restoring Control File From Autobackup

- **MOUNT database**
- **Recover database**



```
RMAN> recover database;

Starting recover at 06-AUG-10
Starting implicit crosscheck backup at 06-AUG-10

starting media recovery

archive log thread 1 sequence 1 is already on disk as file /u01/app/oracle/oradata/DB10G/redo01.log
archive log filename=/u01/app/oracle/oradata/DB10G/redo01.log thread=1 sequence=1
media recovery complete, elapsed time: 00:00:03
Finished recover at 06-AUG-10

RMAN>
```

- **Open database with RESETLOGS option**



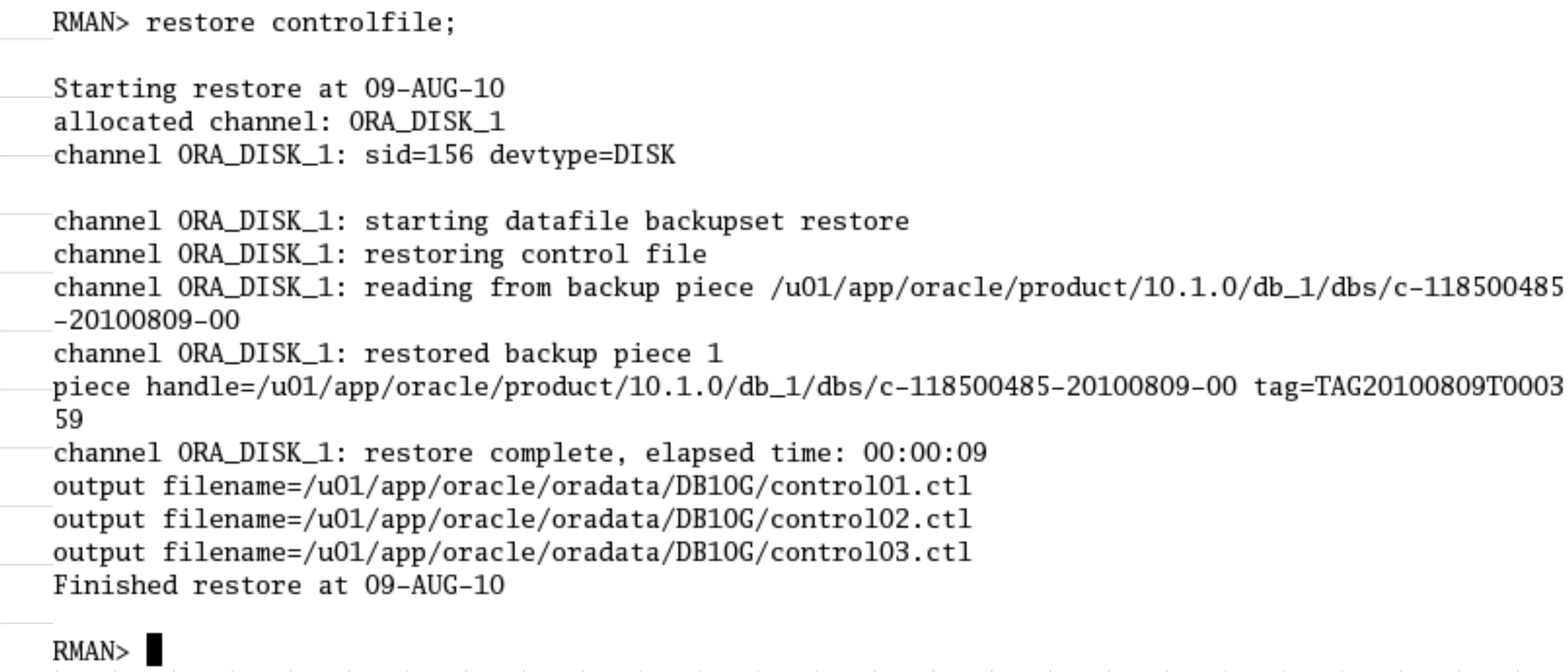
```
SQL>
SQL> alter database open resetlogs;

Database altered.

SQL>
```

CF4: Restore Control File From Recovery Catalog

- **Scenario:**
  - All Control Files are lost
  - Recover Catalog is configured
- **Solution(s):**
  - Start database instance in NOMOUNT mode
  - Restore control file
  - MOUNT the database
  - Recover database
  - Open the database with RESETLOGS option

CF4: Restore Control File From Recovery Catalog

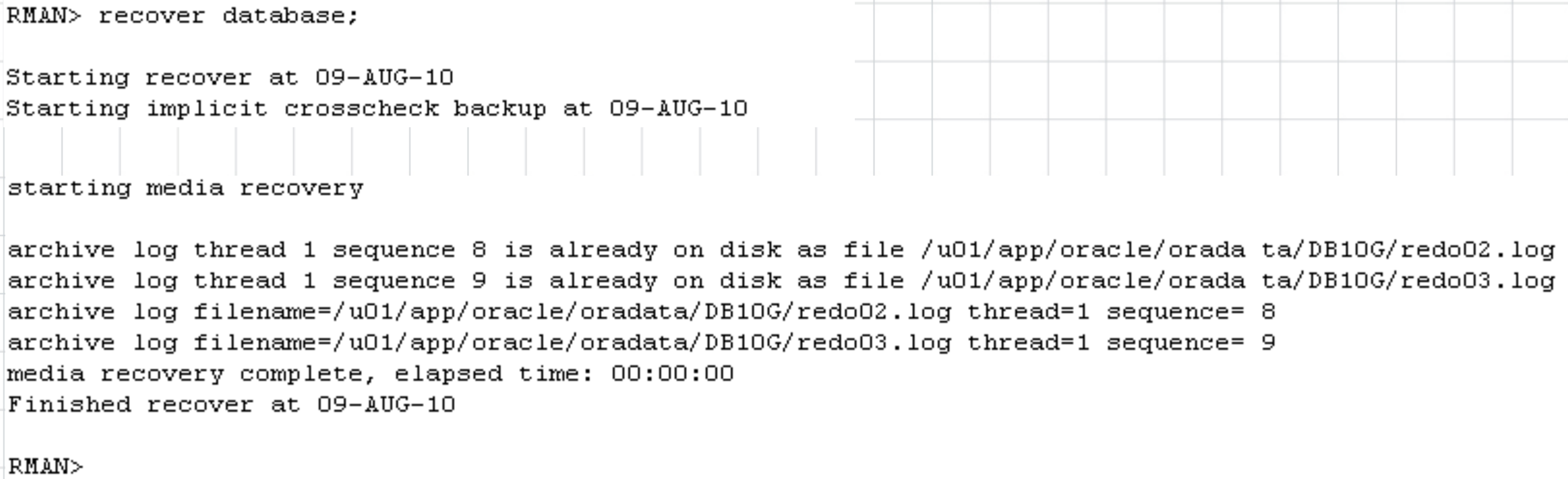
- **Start database instance in NOMOUNT mode**
- **Restore Control File**
  - Setting DBID is not required as recover catalog is configured



```
RMAN> restore controlfile;

Starting restore at 09-AUG-10
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=156 devtype=DISK

channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: restoring control file
channel ORA_DISK_1: reading from backup piece /u01/app/oracle/product/10.1.0/db_1/dbs/c-118500485
-20100809-00
channel ORA_DISK_1: restored backup piece 1
piece handle=/u01/app/oracle/product/10.1.0/db_1/dbs/c-118500485-20100809-00 tag=TAG20100809T0003
59
channel ORA_DISK_1: restore complete, elapsed time: 00:00:09
output filename=/u01/app/oracle/oradata/DB10G/control01.ctl
output filename=/u01/app/oracle/oradata/DB10G/control02.ctl
output filename=/u01/app/oracle/oradata/DB10G/control03.ctl
Finished restore at 09-AUG-10

RMAN> █
```

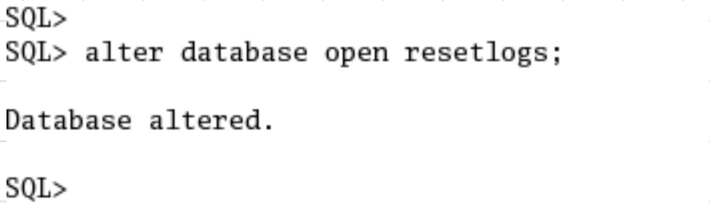
CF4: Restore Control File From Recovery Catalog

- **Recover database**



```
RMAN> recover database;

Starting recover at 09-AUG-10
Starting implicit crosscheck backup at 09-AUG-10


starting media recovery

archive log thread 1 sequence 8 is already on disk as file /u01/app/oracle/orada ta/DB10G/redo02.log
archive log thread 1 sequence 9 is already on disk as file /u01/app/oracle/orada ta/DB10G/redo03.log
archive log filename=/u01/app/oracle/oradata/DB10G/redo02.log thread=1 sequence= 8
archive log filename=/u01/app/oracle/oradata/DB10G/redo03.log thread=1 sequence= 9
media recovery complete, elapsed time: 00:00:00
Finished recover at 09-AUG-10

RMAN>
```

- **Open database with RESETLOGS option**



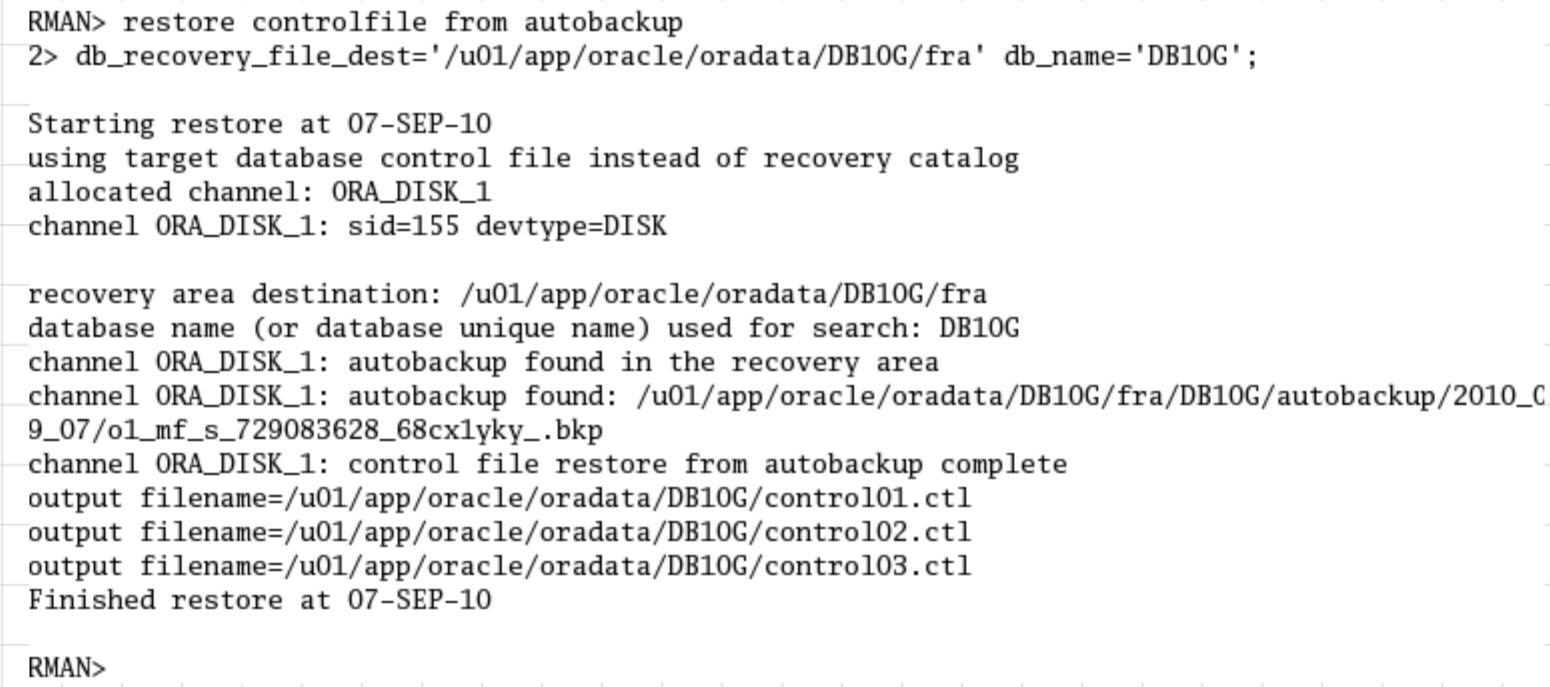
```
SQL>
SQL> alter database open resetlogs;

Database altered.

SQL>
```

CF5: Restore Control File When FRA is Configured

- **Problem:**
  - All Control Files are lost
  - Flash Recovery Area (FRA) is configured
  - Autobackup feature is disabled
- **Solution(s):**
  - Start database instance in NOMOUNT mode
  - Restore control file
  - MOUNT the database
  - Recover database
  - Open the database with RESETLOGS option

CF5: Restore Control File When FRA is Configured

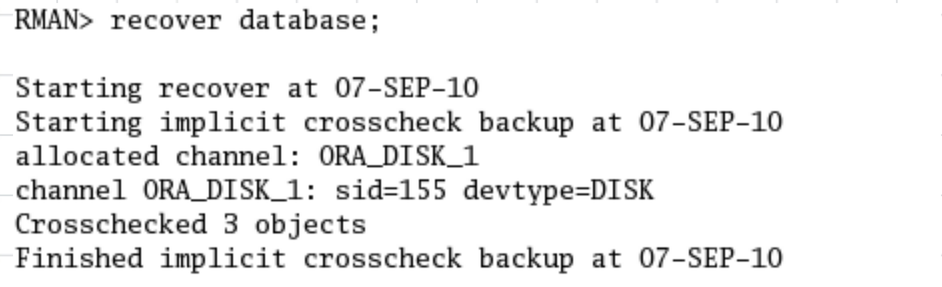
- **Start database instance in NOMOUNT mode**
- **Restore Control File**



```
RMAN> restore controlfile from autobackup
2> db_recovery_file_dest='/u01/app/oracle/oradata/DB10G/fra' db_name='DB10G';

Starting restore at 07-SEP-10
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=155 devtype=DISK

recovery area destination: /u01/app/oracle/oradata/DB10G/fra
database name (or database unique name) used for search: DB10G
channel ORA_DISK_1: autobackup found in the recovery area
channel ORA_DISK_1: autobackup found: /u01/app/oracle/oradata/DB10G/fra/DB10G/autobackup/2010_0
9_07/o1_mf_s_729083628_68cx1yky_.bkp
channel ORA_DISK_1: control file restore from autobackup complete
output filename=/u01/app/oracle/oradata/DB10G/control01.ctl
output filename=/u01/app/oracle/oradata/DB10G/control02.ctl
output filename=/u01/app/oracle/oradata/DB10G/control03.ctl
Finished restore at 07-SEP-10

RMAN>
```

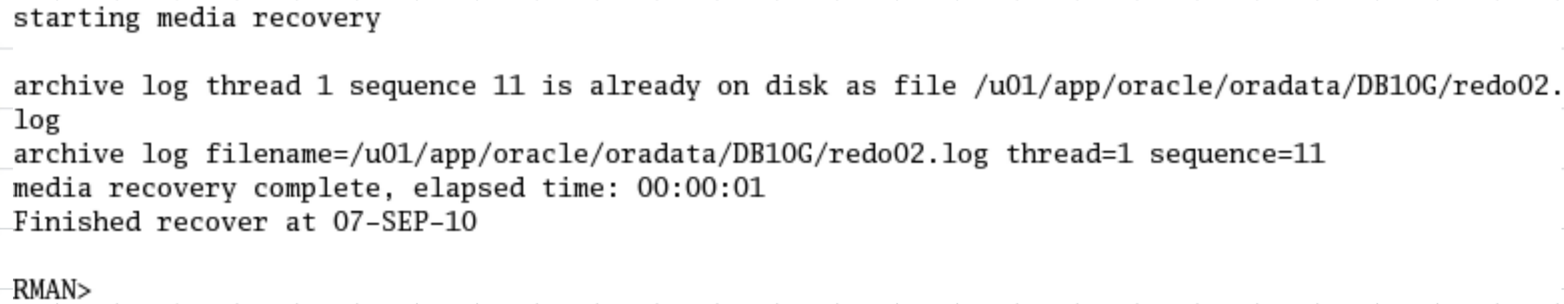
CF5: Restore Control File When FRA is Configured

- **MOUNT database**
- **Recover database**



```
RMAN> recover database;

Starting recover at 07-SEP-10
Starting implicit crosscheck backup at 07-SEP-10
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=155 devtype=DISK
Crosschecked 3 objects
Finished implicit crosscheck backup at 07-SEP-10
```

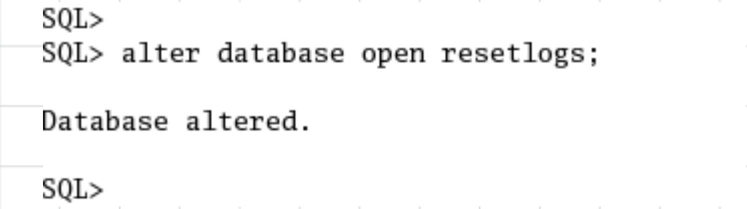
:



```
starting media recovery

archive log thread 1 sequence 11 is already on disk as file /u01/app/oracle/oradata/DB10G/redo02.
log
archive log filename=/u01/app/oracle/oradata/DB10G/redo02.log thread=1 sequence=11
media recovery complete, elapsed time: 00:00:01
Finished recover at 07-SEP-10

RMAN>
```

CF5: Restore Control File When FRA is Configured

- **Open database with RESETLOGS option**



```
SQL>
SQL> alter database open resetlogs;

Database altered.

SQL>
```

Loss of Redo Log Files

Loss of Redo Log Files

- Understanding STATUS column of V$LOG and V $LOGFILE views
- RLF1: Loosing a Member of Multiplexed Redo Log Files
- RLF2: Loosing INACTIVE Redo Log Files
- RLF3: Loosing CURRENT Redo Log Files
- RLF4: Loosing ACTIVE Redo Log Files

Understanding STATUS of Redo Log Group (V$LOG)

The STATUS column of V$LOG view reflects the status of the *log group:*

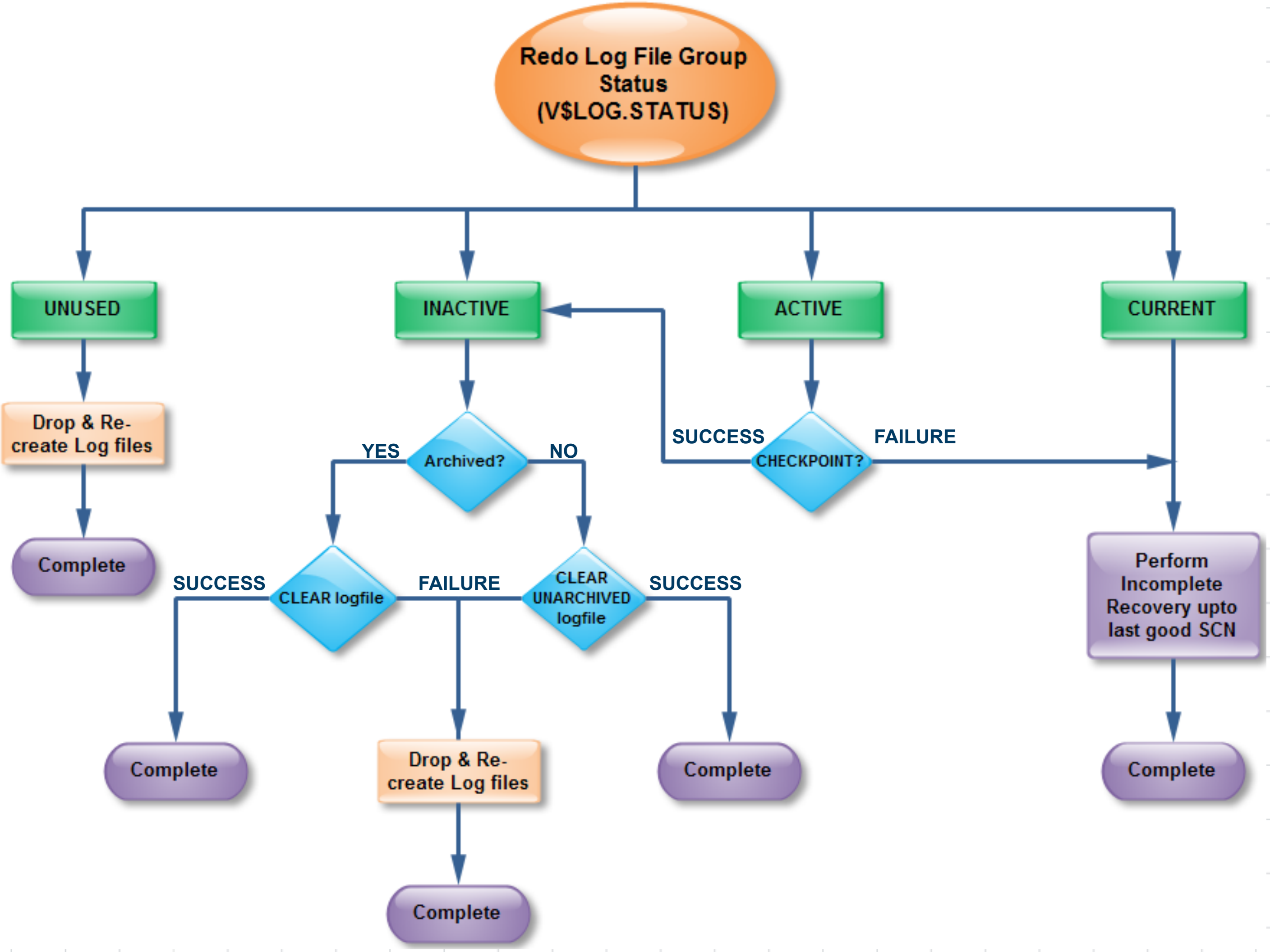
- **CURRENT:** The log group that is currently being written to by the log writer.
- **ACTIVE:** The log group is required for crash recovery and may or may not have been archived.
- **INACTIVE:** The log group isn’t needed for crash recovery and may or may not have been archived.
- **UNUSED:** The log group has never been written to as it was recently created.

Understanding STATUS of Redo Log Files (V$LOG)

The STATUS column of V$LOGFILE reports the status of a *online redo log file member :*

- **INVALID:** The log file member is inaccessible, or it has been recently created.
- **NULL:** The log file member is being used by the database.

Recovering from Redo Log File Failures - Flowchart

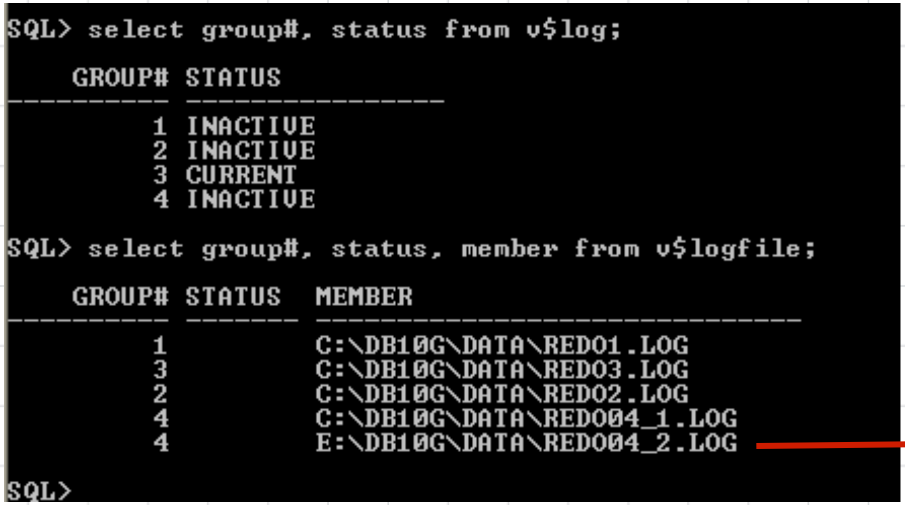


RLF1: Loosing a Member of Multiplexed Redo Log Files

- **Problem:**
  - A member of multiplexed redo log group is lost
  - Database is UP
- **Solution(s):**
  - Fix the media or
  - Drop the affected Redo Log File and Create a new one in a different location

RLF1: Loosing a Member of Multiplexed Redo Log Files

- **Simulating Media Failure**
  - Place one of the Redo Log member on a pen drive



```
SQL> select group#, status from v$log;

    GROUP# STATUS
---------- ----------------
         1 INACTIVE
         2 INACTIVE
         3 CURRENT
         4 INACTIVE

SQL> select group#, status, member from v$logfile;

    GROUP# STATUS  MEMBER
---------- ------- ------------------------------
         1         C:\DB10G\DATA\REDO1.LOG
         3         C:\DB10G\DATA\REDO3.LOG
         2         C:\DB10G\DATA\REDO2.LOG
         4         C:\DB10G\DATA\REDO04_1.LOG
         4         E:\DB10G\DATA\REDO04_2.LOG

SQL>
```

→ **Pen Drive**

- **Unplug the pen drive while database is still in open mode**
- **Oracle Instance remains up and continues to function normally while reporting errors in alert.log**

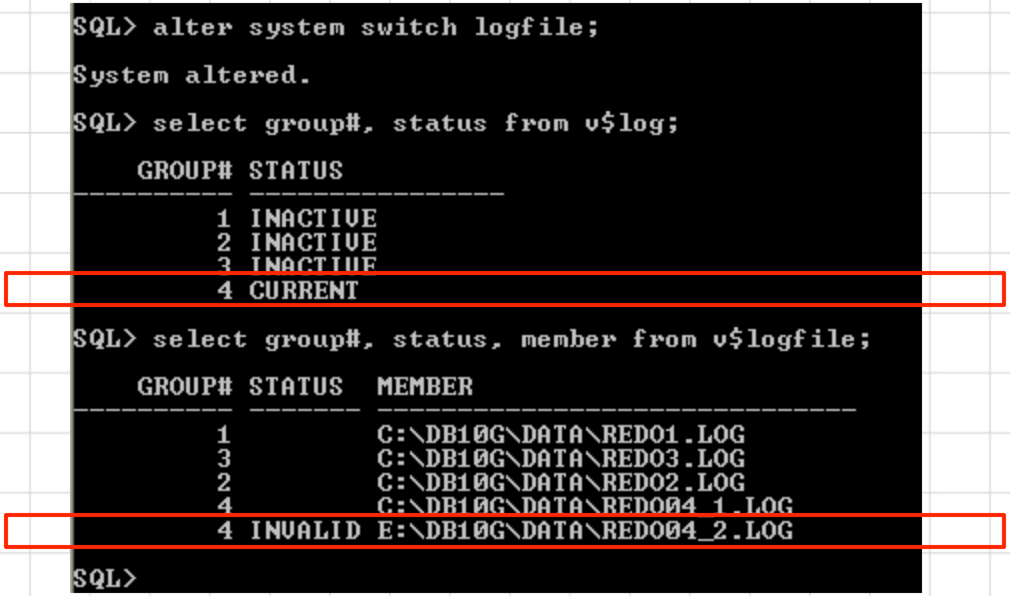
RLF1: Loosing a Member of Multiplexed Redo Log Files

- **Perform few log switches and monitor alert.log**

```
Errors in file c:\db10g\dump\db10g_lgwr_3724.trc:
ORA-00321: log 4 of thread 1, cannot update log file header
ORA-00312: online log 4 thread 1: 'E:\DB10G\DATA\REDO04_2.LOG'
ORA-27091: unable to queue I/O
ORA-27070: async read/write failed
OSD-04008: WriteFile() failure, unable to write to file
O/S-Error: (OS 1006) The volume for a file has been externally altered so that the opened file is no longer valid.

Sat Aug 14 20:43:50 2010
Errors in file c:\db10g\dump\db10g_lgwr_3724.trc:
ORA-00313: open failed for members of log group 4 of thread 1
```

RLF1: Loosing a Member of Multiplexed Redo Log Files

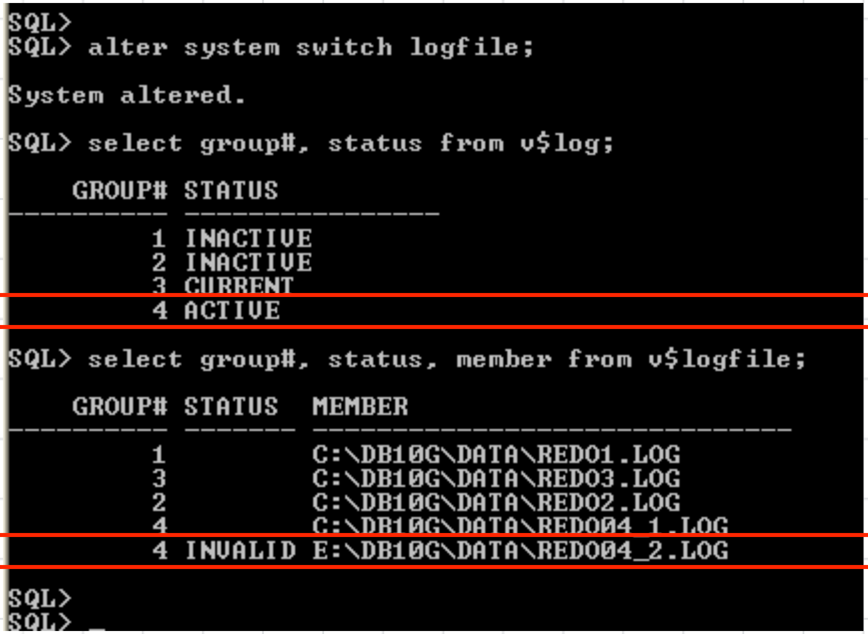
- Oracle marks the unavailable Redo Log file as INVALID
- Database remains UP



```
SQL> alter system switch logfile;

System altered.

SQL> select group#, status from v$log;

    GROUP# STATUS
---------- ----------------
         1 INACTIVE
         2 INACTIVE
         3 INACTIVE
         4 CURRENT

SQL> select group#, status, member from v$logfile;

    GROUP# STATUS  MEMBER
---------- ------- ------------------------------
         1         C:\DB10G\DATA\REDO1.LOG
         3         C:\DB10G\DATA\REDO3.LOG
         2         C:\DB10G\DATA\REDO2.LOG
         4         C:\DB10G\DATA\REDO04_1.LOG
         4 INVALID E:\DB10G\DATA\REDO04_2.LOG

SQL>
```

RLF1: Loosing a Member of Multiplexed Redo Log Files

- **Force a log switch**
  - Log Group Status = ACTIVE
  - Member Status = INVALID



```
SQL>
SQL> alter system switch logfile;

System altered.

SQL> select group#, status from v$log;

    GROUP# STATUS
---------- ----------------
         1 INACTIVE
         2 INACTIVE
         3 CURRENT
         4 ACTIVE

SQL> select group#, status, member from v$logfile;

    GROUP# STATUS  MEMBER
---------- ------- ------------------------------
         1         C:\DB10G\DATA\REDO1.LOG
         3         C:\DB10G\DATA\REDO3.LOG
         2         C:\DB10G\DATA\REDO2.LOG
         4         C:\DB10G\DATA\REDO04_1.LOG
         4 INVALID E:\DB10G\DATA\REDO04_2.LOG

SQL>
SQL> _
```

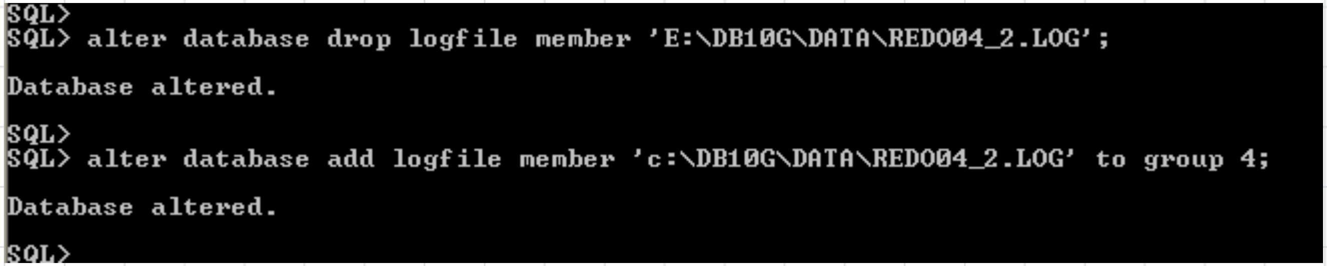
RLF1: Loosing a Member of Multiplexed Redo Log Files

Solution(s):

1) Fix the media (plug in the pen drive)

- Log Writer continues writing to the Redo Log File as if the problem never existed
- The V$LOG.STATUS is updated to NULL

2) Drop and recreate the affected member to a different location



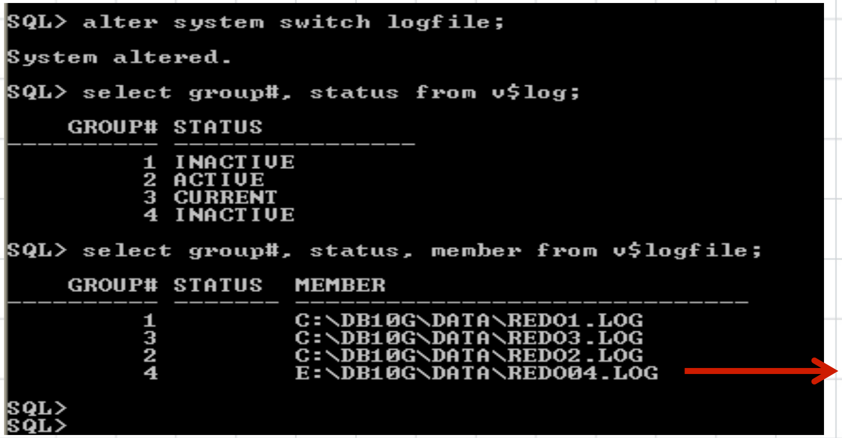
```
SQL>
SQL> alter database drop logfile member 'E:\DB10G\DATA\REDO04_2.LOG';

Database altered.

SQL>
SQL> alter database add logfile member 'c:\DB10G\DATA\REDO04_2.LOG' to group 4;

Database altered.

SQL>
```

RLF2: Loosing INACTIVE Redo Log Files

- **Problem:**
  - The only member of the redo log group is lost
  - Status of the Redo Log Group is INACTIVE
  - Database is Up
- **Solution(s):**
  1) If this is a temporary media failure, fix the issue and start database
  2) If the redo log file is lost while the media remains available then clear the archvied/unarchived log file
  3) If the media failure is permanent then drop and re-create the redo log group to a new location

RLF2: Loosing INACTIVE Redo Log Files

- **Simulating Media Failure**
  - Create a Redo Log Group with a single member on a pen drive



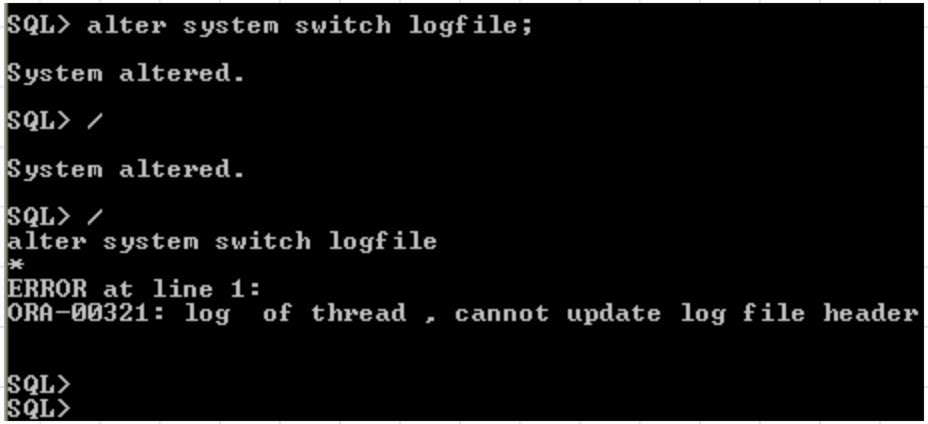
```
SQL> alter system switch logfile;

System altered.

SQL> select group#, status from v$log;

    GROUP# STATUS
---------- ----------------
         1 INACTIVE
         2 ACTIVE
         3 CURRENT
         4 INACTIVE

SQL> select group#, status, member from v$logfile;

    GROUP# STATUS  MEMBER
---------- ------- ------------------------------
         1         C:\DB10G\DATA\REDO1.LOG
         3         C:\DB10G\DATA\REDO3.LOG
         2         C:\DB10G\DATA\REDO2.LOG
         4         E:\DB10G\DATA\REDO04.LOG

SQL>
SQL>
```

Pen Drive

  - Unplug the pen drive while the database is still open and V $LOG.STATUS = INACTIVE
  - Perform log switches until Oracle tries to reuse the redo log file residing in the pen drive

RLF2: Loosing INACTIVE Redo Log Files

– LGWR terminates the instance with ORA-00321 error as shown:



```
SQL> alter system switch logfile;

System altered.

SQL> /

System altered.

SQL> /
alter system switch logfile
*
ERROR at line 1:
ORA-00321: log  of thread , cannot update log file header


SQL>
SQL>
```

– Alert.log →

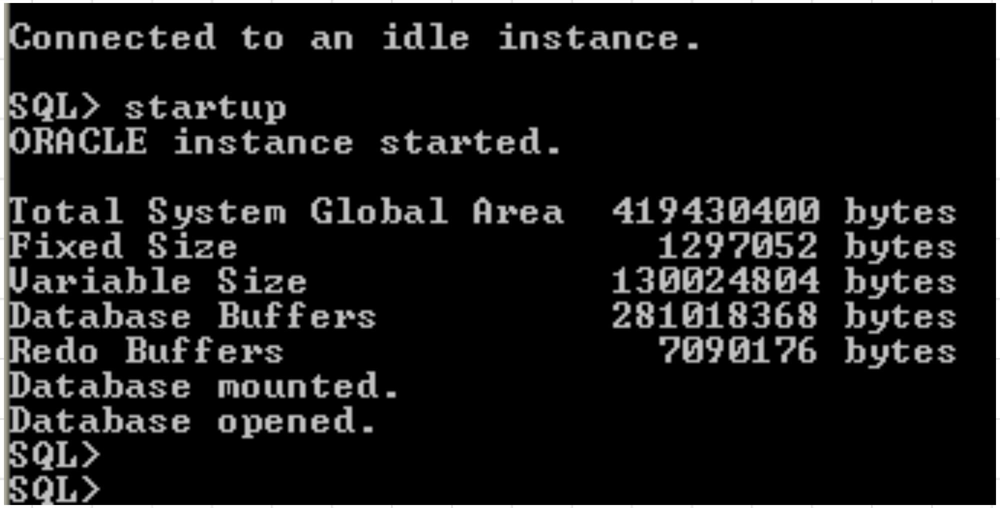
```
Errors in file c:\db10g\dump\db10g_lgwr_1348.trc:
ORA-00321: log 4 of thread 1, cannot update log file header
ORA-00312: online log 4 thread 1: 'E:\DB10G\DATA\REDO04.LOG'
ORA-27091: unable to queue I/O
ORA-27070: async read/write failed
OSD-04008: WriteFile() failure, unable to write to file
O/S-Error: (OS 1006) The volume for a file has been externally altered so that the opened file is no
        longer valid.

Sat Aug 14 21:06:15 2010
Errors in file c:\db10g\dump\db10g_lgwr_1348.trc:
ORA-00321: log 4 of thread 1, cannot update log file header

LGWR: terminating instance due to error 321
Sat Aug 14 21:06:16 2010
Errors in file c:\db10g\dump\db10g_q001_4008.trc:
ORA-00321: log  of thread , cannot update log file header
```

RLF2: Loosing INACTIVE Redo Log Files

Solution(s):

1) Fix the media (plug in the pen drive)

- Start database
- Oracle performs crash recovery behind the scenes
- The V$LOG.STATUS is updated to NULL



```
Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area  419430400 bytes
Fixed Size                  1297052 bytes
Variable Size             130024804 bytes
Database Buffers          281018368 bytes
Redo Buffers                7090176 bytes
Database mounted.
Database opened.
SQL>
SQL>
```

RLF2: Loosing INACTIVE Redo Log Files

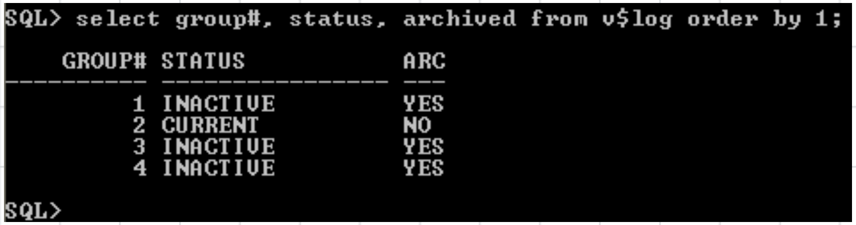
- Crash recovery information in database alert.log

```
ALTER DATABASE OPEN
Sat Aug 28 01:32:17 2010
Beginning crash recovery of 1 threads
 parallel recovery started with 2 processes
Sat Aug 28 01:32:17 2010
Started redo scan
Sat Aug 28 01:32:18 2010
Completed redo scan
 0 redo blocks read, 0 data blocks need recovery
Sat Aug 28 01:32:18 2010
Started redo application at
 Thread 1: logseq 907, block 2, scn 90480632
Sat Aug 28 01:32:18 2010
Recovery of Online Redo Log: Thread 1 Group 2 Seq 907 Reading mem 0
  Mem# 0: C:\DB10G\DATA\REDO2.LOG
Sat Aug 28 01:32:18 2010
Completed redo application
Sat Aug 28 01:32:18 2010
Completed crash recovery at
 Thread 1: logseq 907, block 2, scn 90500633
 0 data blocks read, 0 data blocks written, 0 redo blocks read
Sat Aug 28 01:32:19 2010
```

RLF2: Loosing INACTIVE Redo Log Files

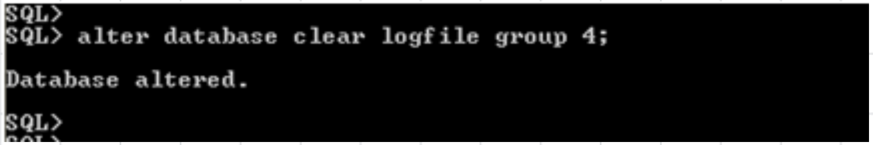
2) If the redo log file is lost while the media remains available then clear the archived/unarchived log file

- Identify whether redo log was archived or not by querying V$LOG view



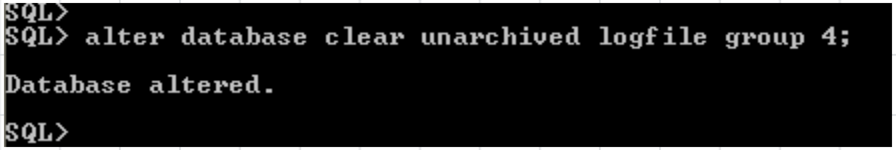
```
SQL> select group#, status, archived from v$log order by 1;

    GROUP# STATUS           ARC
---------- ---------------- ---
         1 INACTIVE         YES
         2 CURRENT          NO
         3 INACTIVE         YES
         4 INACTIVE         YES

SQL>
```

- If the redo log file is archived then use CLEAR ARCHIVED command



```
SQL>
SQL> alter database clear logfile group 4;

Database altered.

SQL>
```

- If the redo log file is not archived then use CLEAR UNARCHIVED command



```
SQL>
SQL> alter database clear unarchived logfile group 4;

Database altered.

SQL>
```

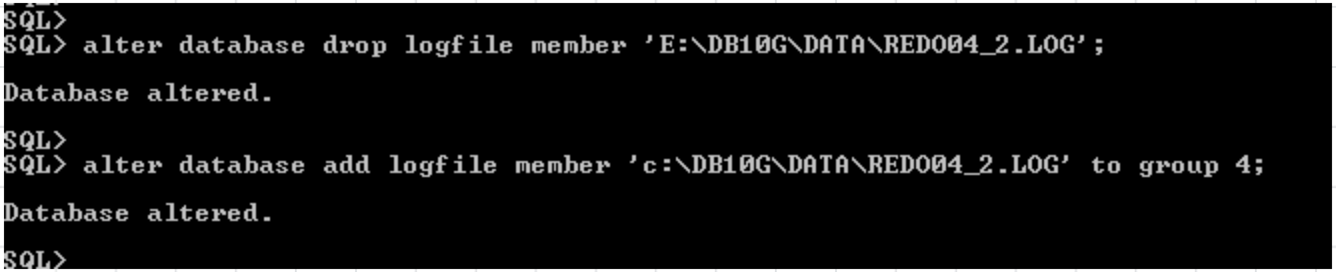
- Open database

RLF2: Loosing INACTIVE Redo Log Files

Solution(s):

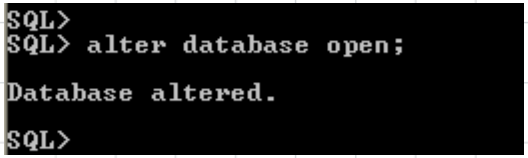
3) If the media failure is permanent then drop and re-create the redo log group to a new location

- Drop and Re-create



```
SQL>
SQL> alter database drop logfile member ’E:\DB10G\DATA\REDO04_2.LOG’;

Database altered.

SQL>
SQL> alter database add logfile member ’c:\DB10G\DATA\REDO04_2.LOG’ to group 4;

Database altered.

SQL>
```

- Open Database



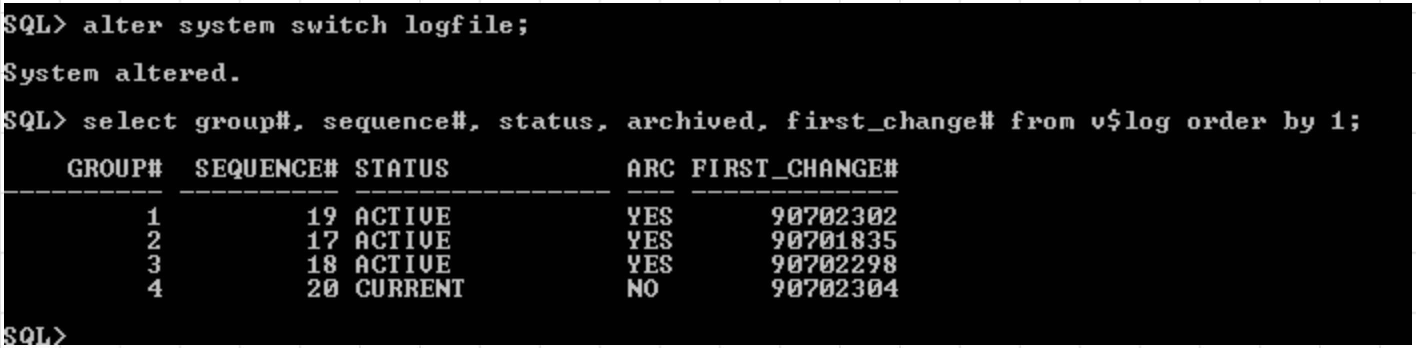
```
SQL>
SQL> alter database open;

Database altered.

SQL>
```

RLF3: Loosing CURRENT Redo Log Files

- **Problem:**
  - All the member of an CURRENT redo log group are lost
  - Valid database backup exist
- **Solution:**
  - Startup database in MOUNT mode
  - Identify the last good SCN
  - Restore database until last good SCN
  - Recover database until last good SCN
  - Re-create the redo log group to a different location
  - Open database with RESETLOGS option

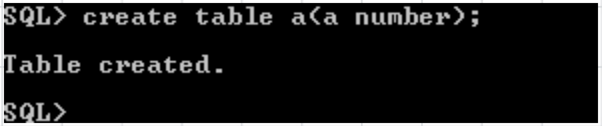
RLF3: Loosing CURRENT Redo Log Files

- **Simulating Media Failure**
  - Create a Redo Log Group with a single member on a pen drive
  - Switch archive logs until the status of redo log residing in pen drive changes to CURRENT



```
SQL> alter system switch logfile;

System altered.

SQL> select group#, sequence#, status, archived, first_change# from v$log order by 1;

    GROUP#  SEQUENCE# STATUS           ARC FIRST_CHANGE#
---------- ---------- ---------------- --- -------------
         1         19 ACTIVE           YES      90702302
         2         17 ACTIVE           YES      90701835
         3         18 ACTIVE           YES      90702298
         4         20 CURRENT          NO       90702304

SQL>
```

  - Create a test table

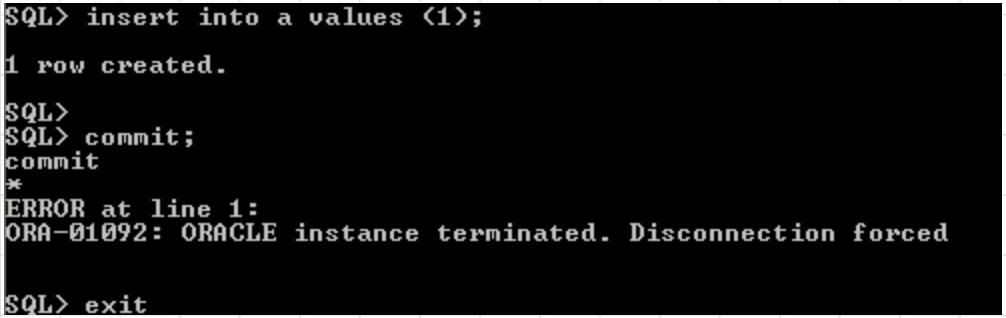


```
SQL> create table a(a number);

Table created.

SQL>
```

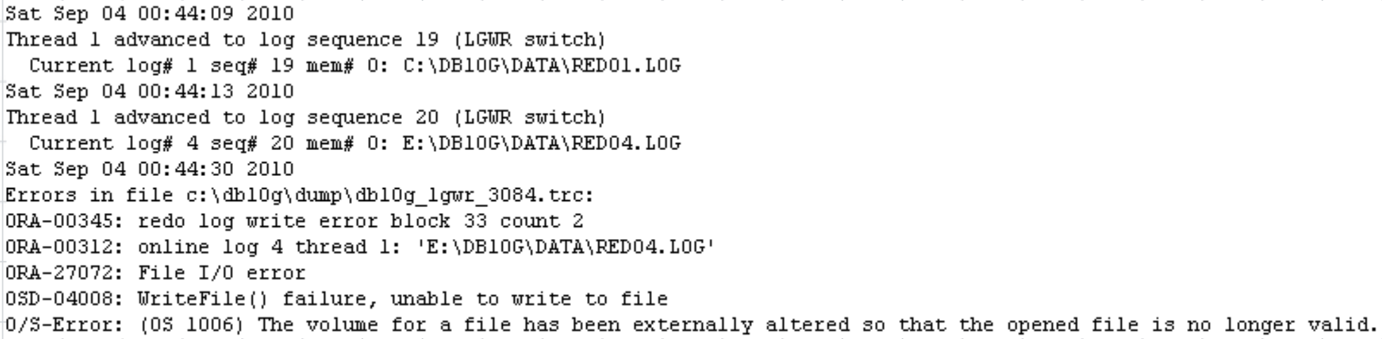
  - Unplug the pen drive

RLF3: Loosing CURRENT Redo Log Files

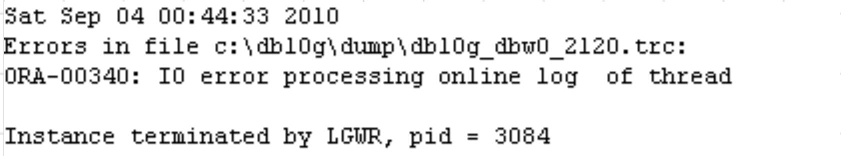
- **Simulating Media Failure**
  - LGWR terminates the instance as it is not able to write to the CURRENT online redo log group



```
SQL> insert into a values (1);

1 row created.

SQL>
SQL> commit;
commit
*
ERROR at line 1:
ORA-01092: ORACLE instance terminated. Disconnection forced


SQL> exit
```



```
Sat Sep 04 00:44:09 2010
Thread 1 advanced to log sequence 19 (LGWR switch)
  Current log# 1 seq# 19 mem# 0: C:\DB10G\DATA\RED01.LOG
Sat Sep 04 00:44:13 2010
Thread 1 advanced to log sequence 20 (LGWR switch)
  Current log# 4 seq# 20 mem# 0: E:\DB10G\DATA\RED04.LOG
Sat Sep 04 00:44:30 2010
Errors in file c:\db10g\dump\db10g_lgwr_3084.trc:
ORA-00345: redo log write error block 33 count 2
ORA-00312: online log 4 thread 1: 'E:\DB10G\DATA\RED04.LOG'
ORA-27072: File I/O error
OSD-04008: WriteFile() failure, unable to write to file
O/S-Error: (OS 1006) The volume for a file has been externally altered so that the opened file is no longer valid.
```

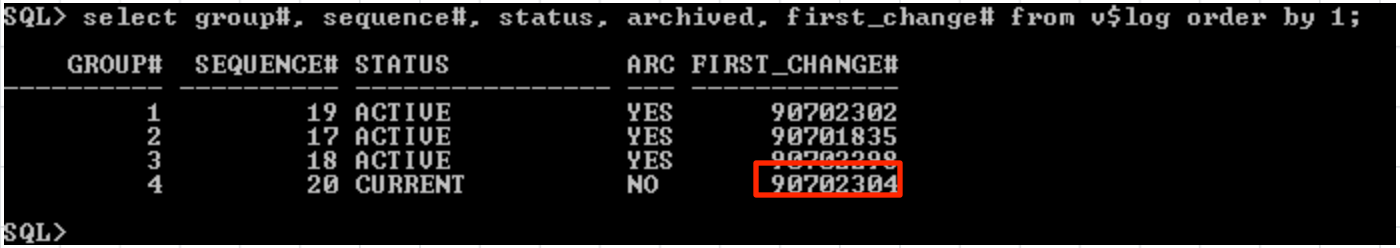


```
Sat Sep 04 00:44:33 2010
Errors in file c:\db10g\dump\db10g_dbw0_2120.trc:
ORA-00340: IO error processing online log  of thread

Instance terminated by LGWR, pid = 3084
```

RLF3: Loosing CURRENT Redo Log Files

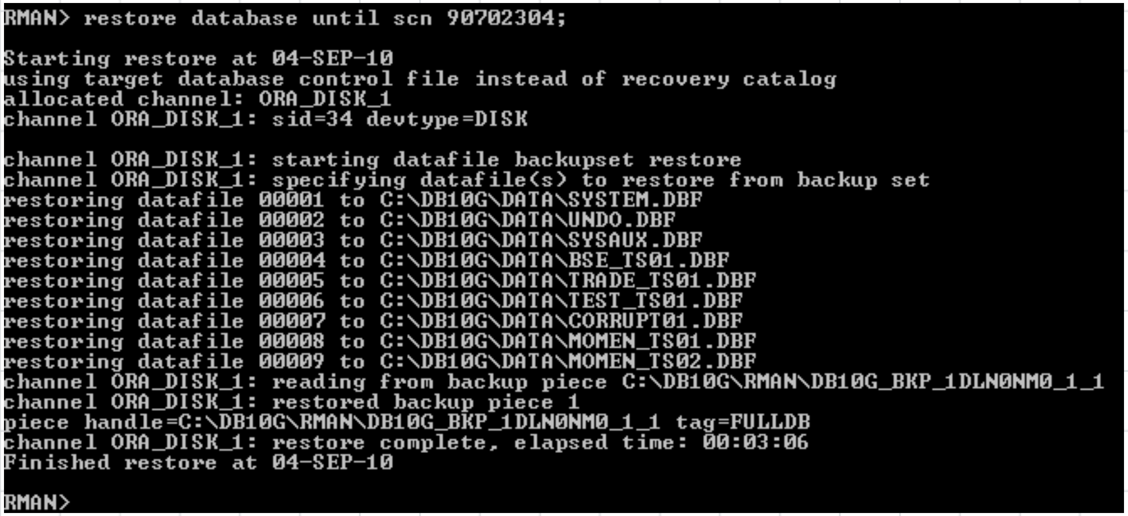
Solution:

- Startup database in MOUNT mode
- Identify the last good SCN



```
SQL> select group#, sequence#, status, archived, first_change# from v$log order by 1;

    GROUP#  SEQUENCE# STATUS           ARC FIRST_CHANGE#
---------- ---------- ---------------- --- -------------
         1         19 ACTIVE           YES      90702302
         2         17 ACTIVE           YES      90701835
         3         18 ACTIVE           YES      90702298
         4         20 CURRENT          NO       90702304

SQL>
```

- Restore database until last good SCN (90702304)

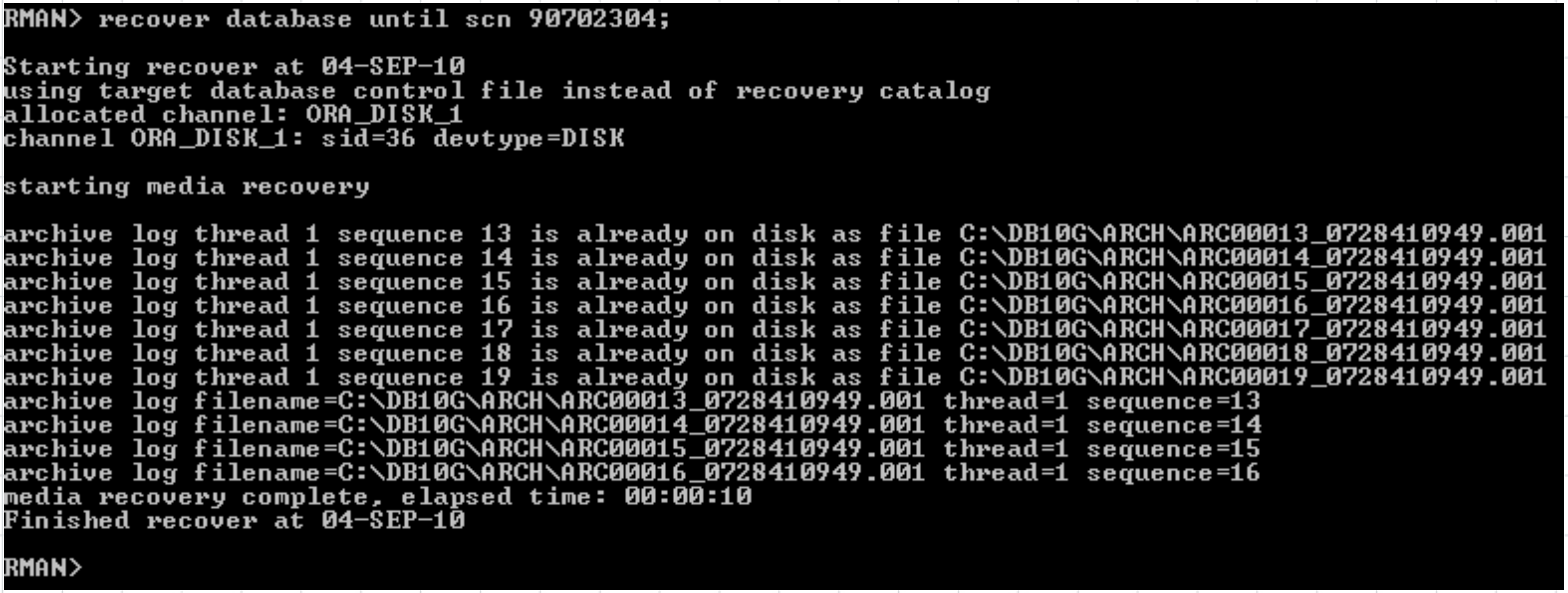


```
RMAN> restore database until scn 90702304;

Starting restore at 04-SEP-10
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=34 devtype=DISK

channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
restoring datafile 00001 to C:\DB10G\DATA\SYSTEM.DBF
restoring datafile 00002 to C:\DB10G\DATA\UNDO.DBF
restoring datafile 00003 to C:\DB10G\DATA\SYSAUX.DBF
restoring datafile 00004 to C:\DB10G\DATA\BSE_TS01.DBF
restoring datafile 00005 to C:\DB10G\DATA\TRADE_TS01.DBF
restoring datafile 00006 to C:\DB10G\DATA\TEST_TS01.DBF
restoring datafile 00007 to C:\DB10G\DATA\CORRUPT01.DBF
restoring datafile 00008 to C:\DB10G\DATA\MOMEN_TS01.DBF
restoring datafile 00009 to C:\DB10G\DATA\MOMEN_TS02.DBF
channel ORA_DISK_1: reading from backup piece C:\DB10G\RMAN\DB10G_BKP_1DLN0NM0_1_1
channel ORA_DISK_1: restored backup piece 1
piece handle=C:\DB10G\RMAN\DB10G_BKP_1DLN0NM0_1_1 tag=FULLDB
channel ORA_DISK_1: restore complete, elapsed time: 00:03:06
Finished restore at 04-SEP-10

RMAN>
```

RLF3: Loosing CURRENT Redo Log Files

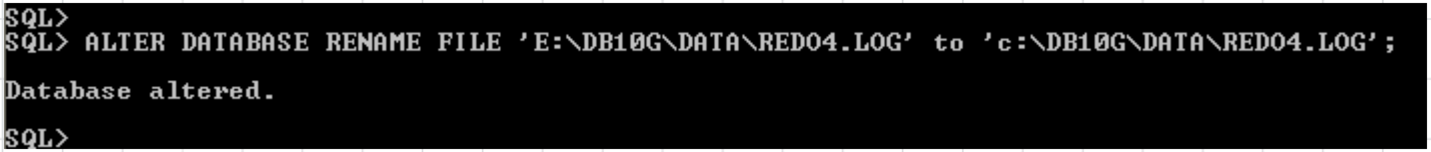
Solution:

- Recover database until the last good SCN (90702304)



```
RMAN> recover database until scn 90702304;

Starting recover at 04-SEP-10
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=36 devtype=DISK

starting media recovery

archive log thread 1 sequence 13 is already on disk as file C:\DB10G\ARCH\ARC00013_0728410949.001
archive log thread 1 sequence 14 is already on disk as file C:\DB10G\ARCH\ARC00014_0728410949.001
archive log thread 1 sequence 15 is already on disk as file C:\DB10G\ARCH\ARC00015_0728410949.001
archive log thread 1 sequence 16 is already on disk as file C:\DB10G\ARCH\ARC00016_0728410949.001
archive log thread 1 sequence 17 is already on disk as file C:\DB10G\ARCH\ARC00017_0728410949.001
archive log thread 1 sequence 18 is already on disk as file C:\DB10G\ARCH\ARC00018_0728410949.001
archive log thread 1 sequence 19 is already on disk as file C:\DB10G\ARCH\ARC00019_0728410949.001
archive log filename=C:\DB10G\ARCH\ARC00013_0728410949.001 thread=1 sequence=13
archive log filename=C:\DB10G\ARCH\ARC00014_0728410949.001 thread=1 sequence=14
archive log filename=C:\DB10G\ARCH\ARC00015_0728410949.001 thread=1 sequence=15
archive log filename=C:\DB10G\ARCH\ARC00016_0728410949.001 thread=1 sequence=16
media recovery complete, elapsed time: 00:00:10
Finished recover at 04-SEP-10

RMAN>
```

- Re-create the redo log group to a different location

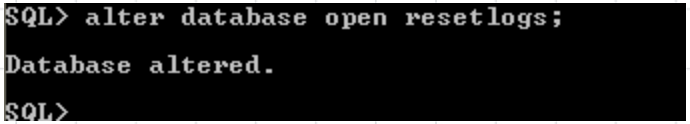


```
SQL>
SQL> ALTER DATABASE RENAME FILE 'E:\DB10G\DATA\REDO4.LOG' to 'c:\DB10G\DATA\REDO4.LOG';

Database altered.

SQL>
```

RLF3: Loosing CURRENT Redo Log Files

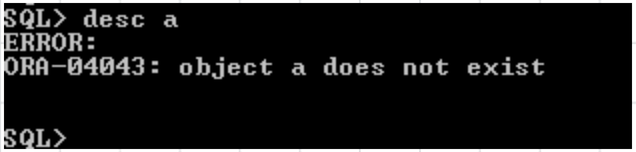
Solution:

- Open database with RESTELOGS options



```
SQL> alter database open resetlogs;

Database altered.

SQL>
```

- Test the existence of the test table ("A")



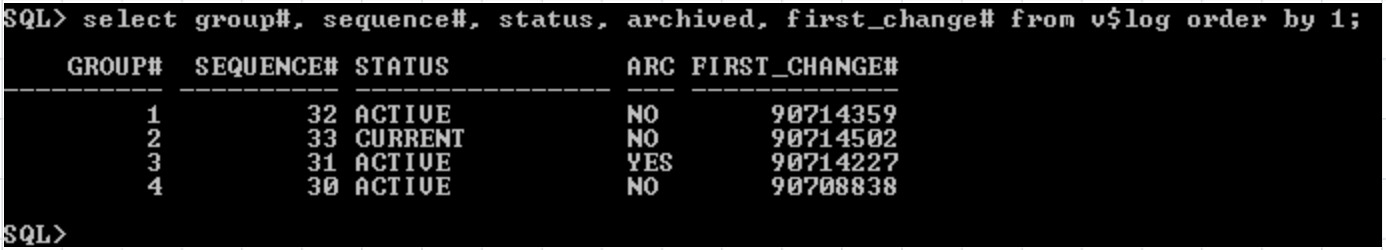
```
SQL> desc a
ERROR:
ORA-04043: object a does not exist


SQL>
```

RLF4: Loosing ACTIVE Redo Log Files

- **Problem:**
  - All the member of an ACTIVE redo log group are lost
  - Database is Up
- **Solution:**
  - Issue a Checkpoint
  - Check redo log status
  - If Checkpoint is SUCCESS then CLEAR redo log group.
  - If Checkpoint FAILS to complete then perform incomplete recovery by identifying the last good SCN

RLF4: Loosing ACTIVE Redo Log Files

- **Simulating Media Failure**
  - Create a Redo Log Group with a single member on a pen drive



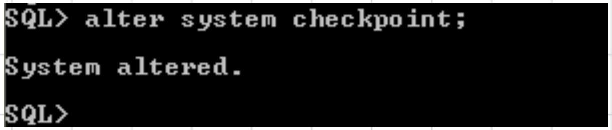
```
SQL> select group#, sequence#, status, archived, first_change# from v$log order by 1;

    GROUP#  SEQUENCE# STATUS           ARC FIRST_CHANGE#
---------- ---------- ---------------- --- -------------
         1         32 ACTIVE           NO       90714359
         2         33 CURRENT          NO       90714502
         3         31 ACTIVE           YES      90714227
         4         30 ACTIVE           NO       90708838

SQL>
```

  - Unplug the pen drive while the database is in open mode and V $LOG.STATUS = ACTIVE

RLF4: Loosing ACTIVE Redo Log Files

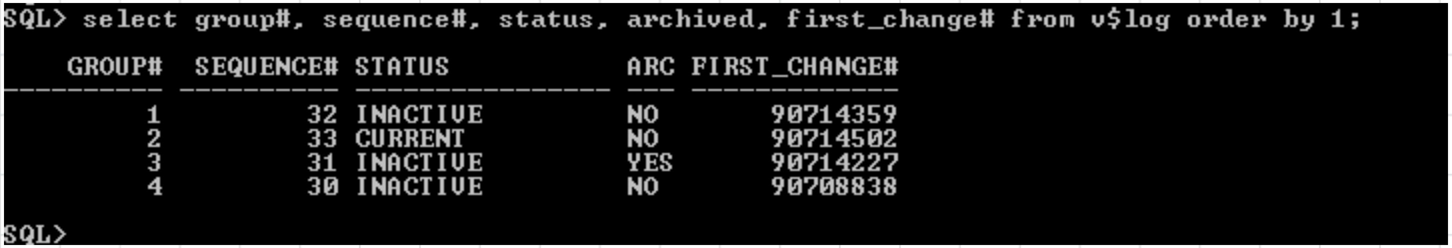
Solution:

- Issue a Checkpoint



```
SQL> alter system checkpoint;

System altered.

SQL>
```

- Check redo log status

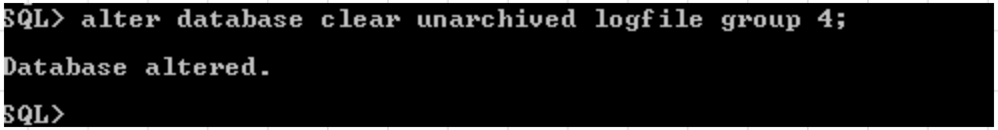


```
SQL> select group#, sequence#, status, archived, first_change# from v$log order by 1;

    GROUP#  SEQUENCE# STATUS           ARC FIRST_CHANGE#
---------- ---------- ---------------- --- -------------
         1         32 INACTIVE         NO       90714359
         2         33 CURRENT          NO       90714502
         3         31 INACTIVE         YES      90714227
         4         30 INACTIVE         NO       90708838

SQL>
```

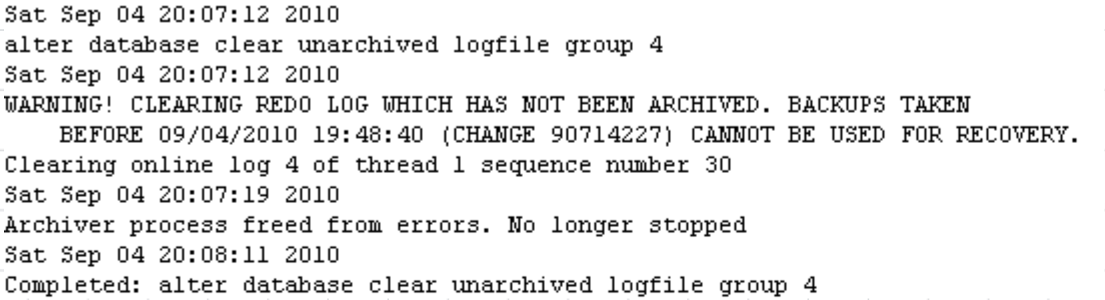
RLF4: Loosing ACTIVE Redo Log Files

Solution:

- If Checkpoint is SUCCESS then CLEAR redo log group.



```
SQL> alter database clear unarchived logfile group 4;

Database altered.

SQL>
```



```
Sat Sep 04 20:07:12 2010
alter database clear unarchived logfile group 4
Sat Sep 04 20:07:12 2010
WARNING! CLEARING REDO LOG WHICH HAS NOT BEEN ARCHIVED. BACKUPS TAKEN
    BEFORE 09/04/2010 19:48:40 (CHANGE 90714227) CANNOT BE USED FOR RECOVERY.
Clearing online log 4 of thread 1 sequence number 30
Sat Sep 04 20:07:19 2010
Archiver process freed from errors. No longer stopped
Sat Sep 04 20:08:11 2010
Completed: alter database clear unarchived logfile group 4
```

- If there is a failure to complete the checkpoint then perform incomplete recovery by identifying the last good SCN as discussed in RLF3-Scenario.

Basic Recovery Solutions

Basic Recovery Solutions

- Datafile Recovery
- Tablespace Recovery
- Recovering Read-Only Tablespace
- Recovering Temporary Tablespace
- Tablespace Point-In-Time Recovery (TSPITR)
- Flashback Database

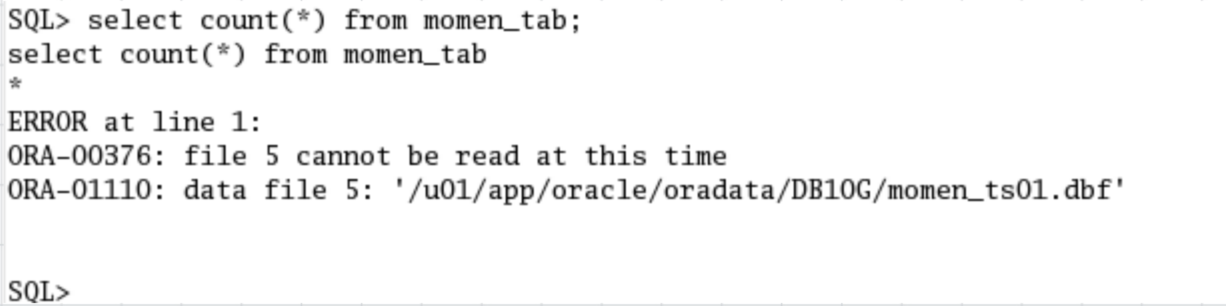
Datafile Recovery

- **Problem:**
  - One of the Datafiles is lost
  - Database is Up
  - Valid backups exist
- **Solution:**
  - Offline all the required datafiles
  - Restore the affected datafiles
  - Recover datafiles
  - Bring back the datafiles to online state
  - Verify restore

Datafile Recovery

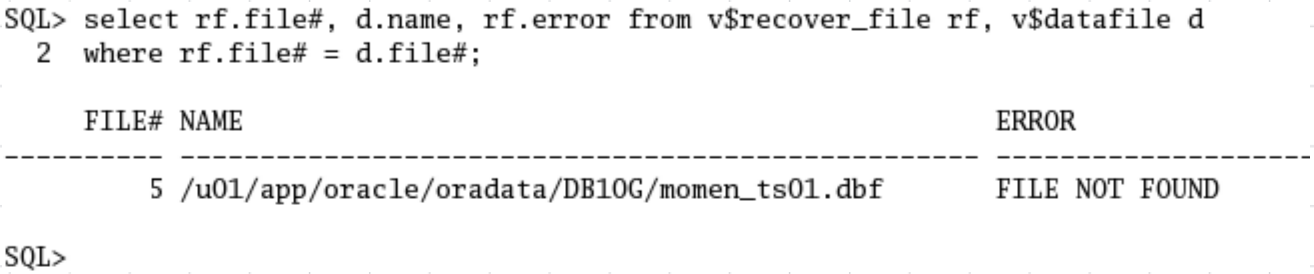
Solution:

- Query any of the tables residing in the datafile



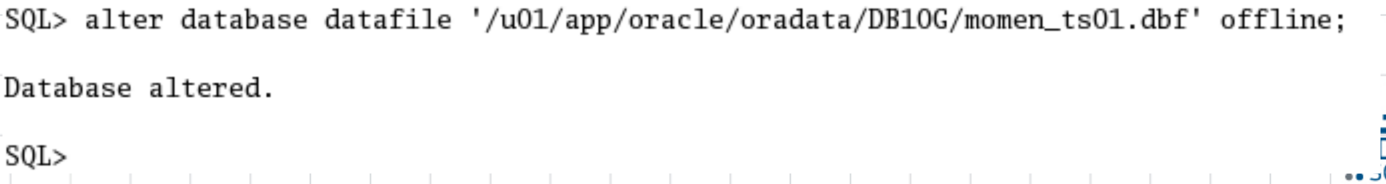
```
SQL> select count(*) from momen_tab;
select count(*) from momen_tab
*
ERROR at line 1:
ORA-00376: file 5 cannot be read at this time
ORA-01110: data file 5: '/u01/app/oracle/oradata/DB10G/momen_ts01.dbf'


SQL>
```

- Find all datafiles that need recovery



```
SQL> select rf.file#, d.name, rf.error from v$recover_file rf, v$datafile d
  2  where rf.file# = d.file#;

     FILE# NAME                                                ERROR
---------- --------------------------------------------------- ---------------------
         5 /u01/app/oracle/oradata/DB10G/momen_ts01.dbf        FILE NOT FOUND

SQL>
```

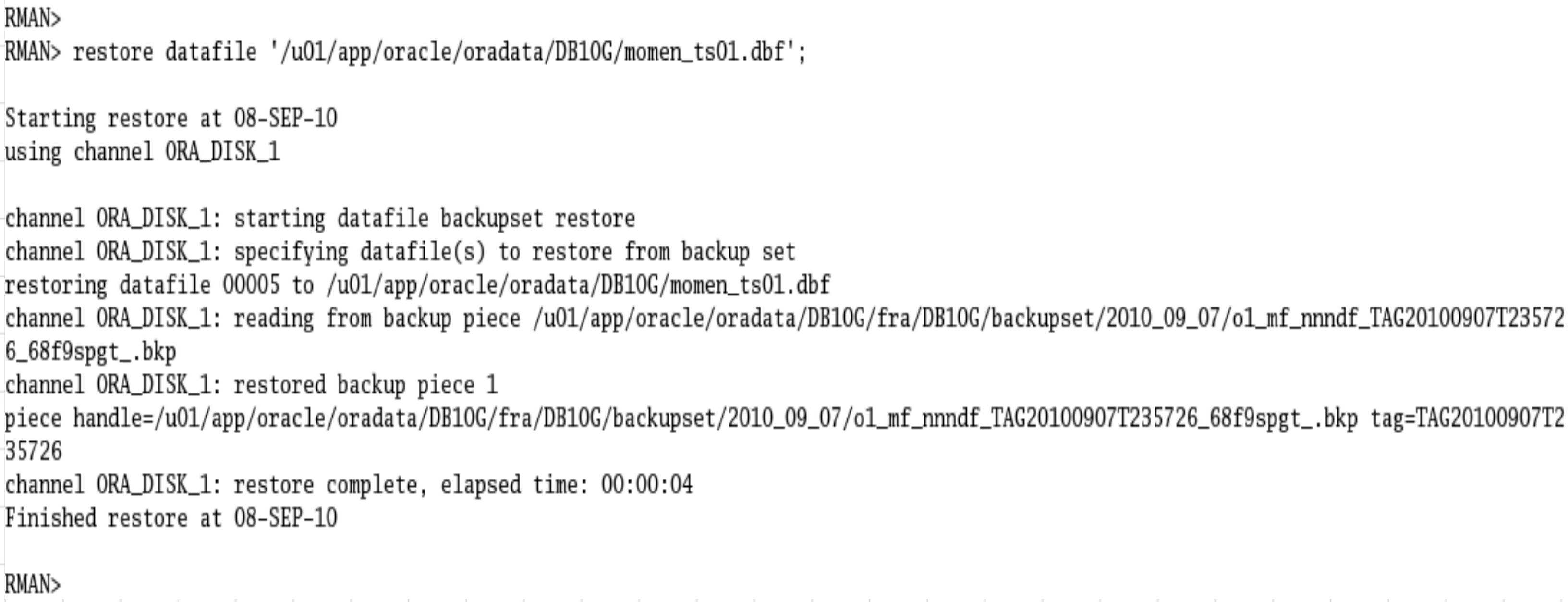
- Take the datafile(s) offline



```
SQL> alter database datafile '/u01/app/oracle/oradata/DB10G/momen_ts01.dbf' offline;

Database altered.

SQL>
```

Datafile Recovery

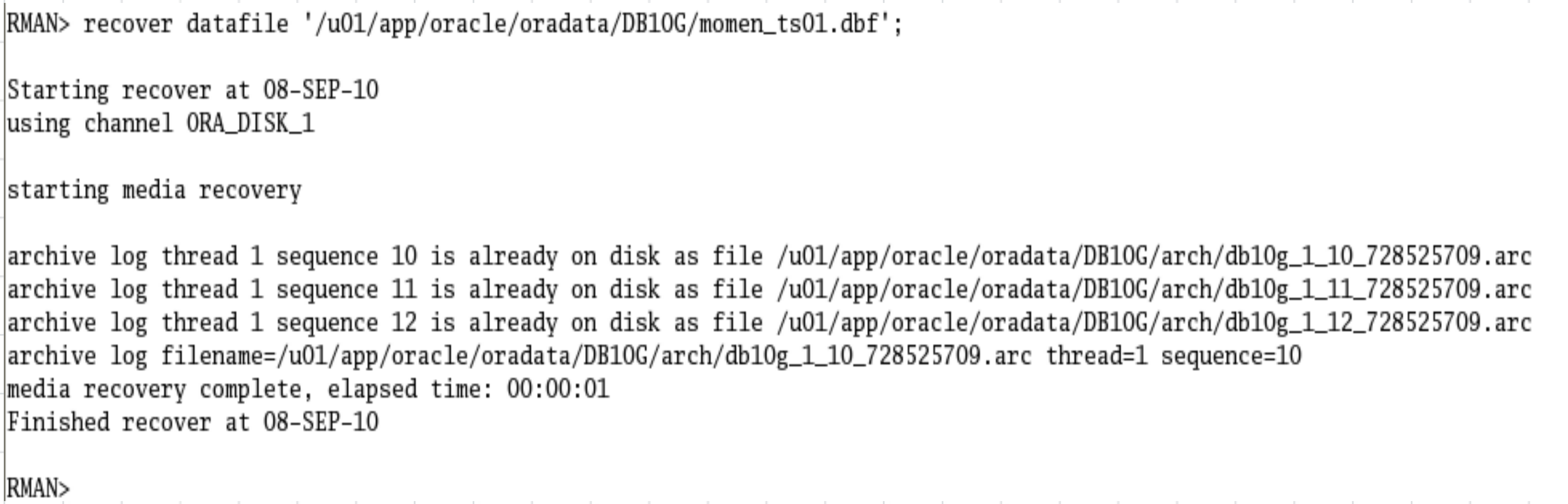
– Restore datafiles affected by media failure



```
RMAN>
RMAN> restore datafile '/u01/app/oracle/oradata/DB10G/momen_ts01.dbf';

Starting restore at 08-SEP-10
using channel ORA_DISK_1

channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
restoring datafile 00005 to /u01/app/oracle/oradata/DB10G/momen_ts01.dbf
channel ORA_DISK_1: reading from backup piece /u01/app/oracle/oradata/DB10G/fra/DB10G/backupset/2010_09_07/o1_mf_nnndf_TAG20100907T23572
6_68f9spgt_.bkp
channel ORA_DISK_1: restored backup piece 1
piece handle=/u01/app/oracle/oradata/DB10G/fra/DB10G/backupset/2010_09_07/o1_mf_nnndf_TAG20100907T235726_68f9spgt_.bkp tag=TAG20100907T2
35726
channel ORA_DISK_1: restore complete, elapsed time: 00:00:04
Finished restore at 08-SEP-10

RMAN>
```

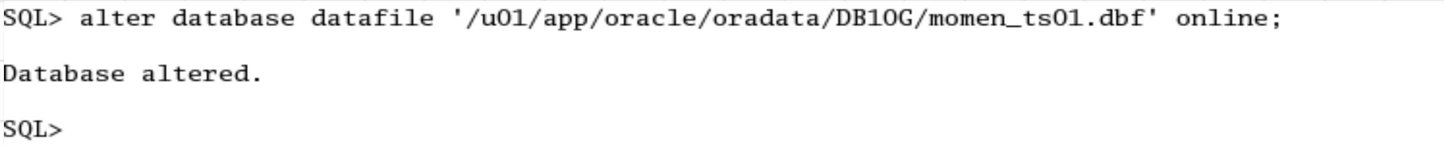
Datafile Recovery

- Recover datafiles



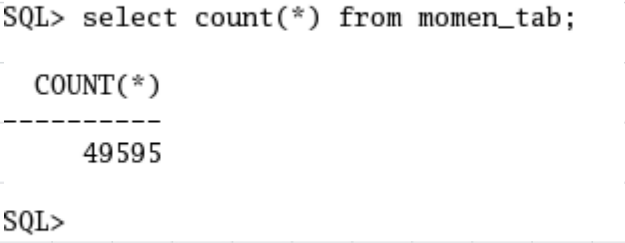
```
RMAN> recover datafile '/u01/app/oracle/oradata/DB10G/momen_ts01.dbf';

Starting recover at 08-SEP-10
using channel ORA_DISK_1

starting media recovery

archive log thread 1 sequence 10 is already on disk as file /u01/app/oracle/oradata/DB10G/arch/db10g_1_10_728525709.arc
archive log thread 1 sequence 11 is already on disk as file /u01/app/oracle/oradata/DB10G/arch/db10g_1_11_728525709.arc
archive log thread 1 sequence 12 is already on disk as file /u01/app/oracle/oradata/DB10G/arch/db10g_1_12_728525709.arc
archive log filename=/u01/app/oracle/oradata/DB10G/arch/db10g_1_10_728525709.arc thread=1 sequence=10
media recovery complete, elapsed time: 00:00:01
Finished recover at 08-SEP-10

RMAN>
```

- Bring back the datafiles to online state



```
SQL> alter database datafile '/u01/app/oracle/oradata/DB10G/momen_ts01.dbf' online;

Database altered.

SQL>
```

- Verify restore



```
SQL> select count(*) from momen_tab;

  COUNT(*)
----------
     49595

SQL>
```

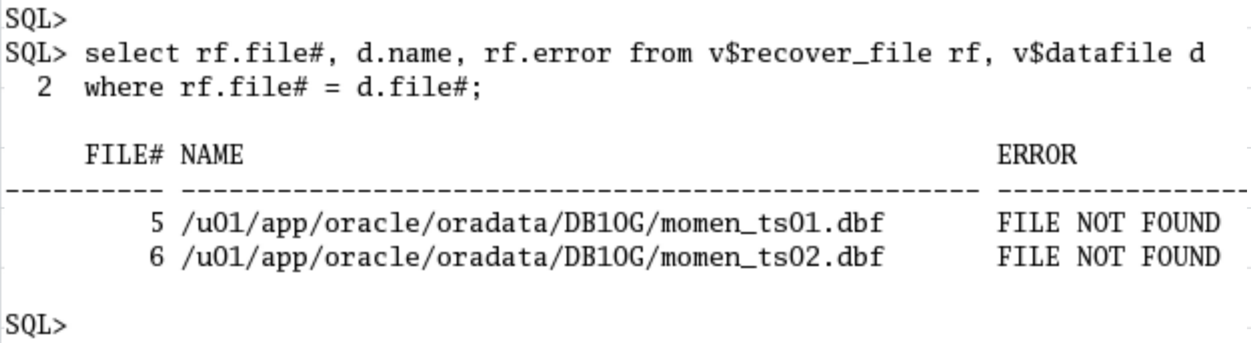
Tablespace Recovery

- **Problem:**
  - Several datafiles of a tablespace are affected
  - Database is Up
  - Valid backups exist
- **Solution:**
  - Offline the affected tablespace
  - Restore tablespace
  - Recover tablespace
  - Bring back the tablespace to online state
  - Verify restore

Tablespace Recovery

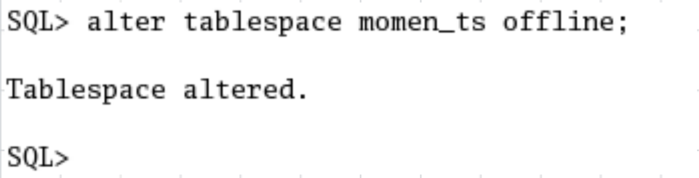
Solution:

- Query V$RECOVER_FILE to view datafiles that need to be recovered



```
SQL>
SQL> select rf.file#, d.name, rf.error from v$recover_file rf, v$datafile d
  2  where rf.file# = d.file#;

     FILE# NAME                                               ERROR
---------- -------------------------------------------------- ----------------
         5 /u01/app/oracle/oradata/DB10G/momen_ts01.dbf       FILE NOT FOUND
         6 /u01/app/oracle/oradata/DB10G/momen_ts02.dbf       FILE NOT FOUND

SQL>
```

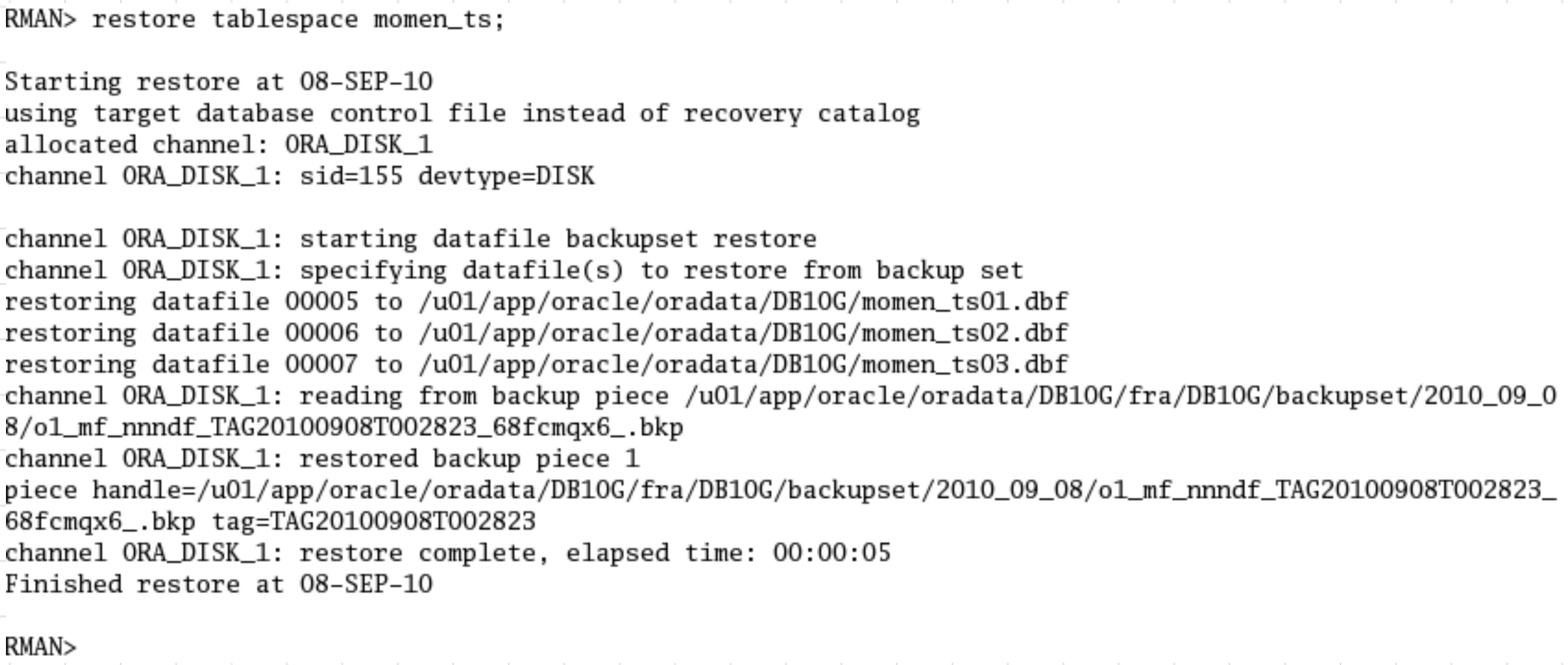
- Take the tablespace(s) offline



```
SQL> alter tablespace momen_ts offline;

Tablespace altered.

SQL>
```

Tablespace Recovery

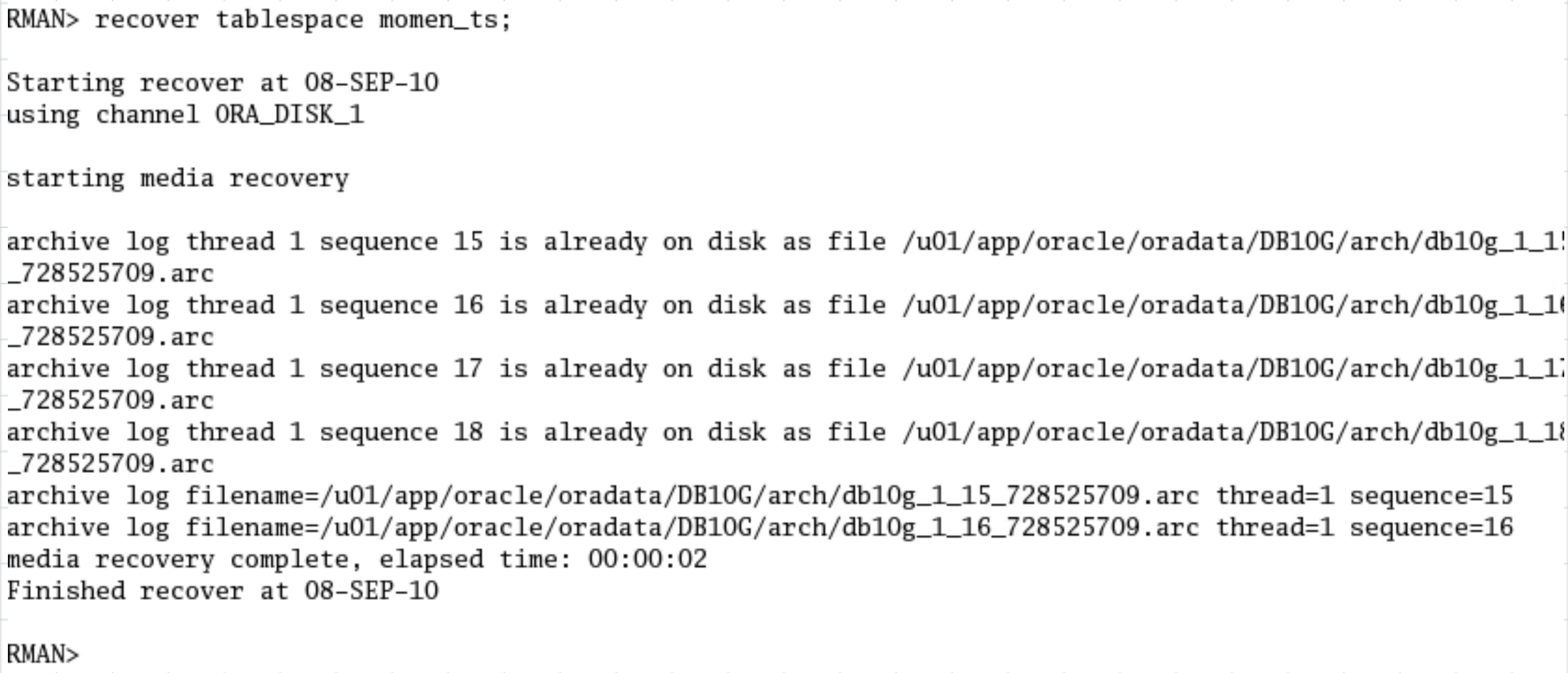
– Restore tablespace



```
RMAN> restore tablespace momen_ts;

Starting restore at 08-SEP-10
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=155 devtype=DISK

channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
restoring datafile 00005 to /u01/app/oracle/oradata/DB10G/momen_ts01.dbf
restoring datafile 00006 to /u01/app/oracle/oradata/DB10G/momen_ts02.dbf
restoring datafile 00007 to /u01/app/oracle/oradata/DB10G/momen_ts03.dbf
channel ORA_DISK_1: reading from backup piece /u01/app/oracle/oradata/DB10G/fra/DB10G/backupset/2010_09_0
8/o1_mf_nnndf_TAG20100908T002823_68fcmqx6_.bkp
channel ORA_DISK_1: restored backup piece 1
piece handle=/u01/app/oracle/oradata/DB10G/fra/DB10G/backupset/2010_09_08/o1_mf_nnndf_TAG20100908T002823_
68fcmqx6_.bkp tag=TAG20100908T002823
channel ORA_DISK_1: restore complete, elapsed time: 00:00:05
Finished restore at 08-SEP-10

RMAN>
```

Tablespace Recovery

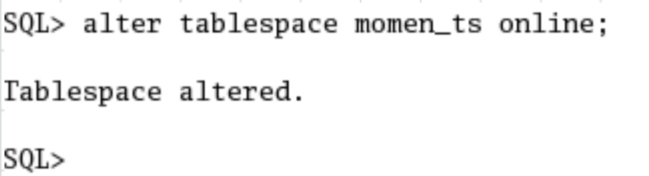
– Recover tablespace



```
RMAN> recover tablespace momen_ts;

Starting recover at 08-SEP-10
using channel ORA_DISK_1

starting media recovery

archive log thread 1 sequence 15 is already on disk as file /u01/app/oracle/oradata/DB10G/arch/db10g_1_15
_728525709.arc
archive log thread 1 sequence 16 is already on disk as file /u01/app/oracle/oradata/DB10G/arch/db10g_1_16
_728525709.arc
archive log thread 1 sequence 17 is already on disk as file /u01/app/oracle/oradata/DB10G/arch/db10g_1_17
_728525709.arc
archive log thread 1 sequence 18 is already on disk as file /u01/app/oracle/oradata/DB10G/arch/db10g_1_18
_728525709.arc
archive log filename=/u01/app/oracle/oradata/DB10G/arch/db10g_1_15_728525709.arc thread=1 sequence=15
archive log filename=/u01/app/oracle/oradata/DB10G/arch/db10g_1_16_728525709.arc thread=1 sequence=16
media recovery complete, elapsed time: 00:00:02
Finished recover at 08-SEP-10

RMAN>
```

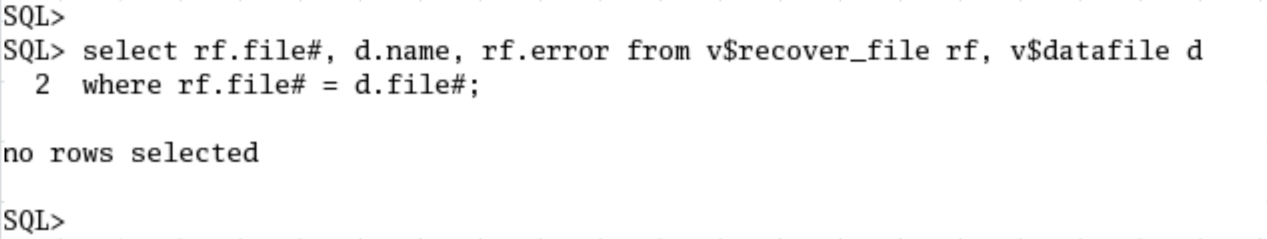
Tablespace Recovery

– Bring the tablespace to online state



```
SQL> alter tablespace momen_ts online;

Tablespace altered.

SQL>
```

– Check any datafiles still need to be recovered



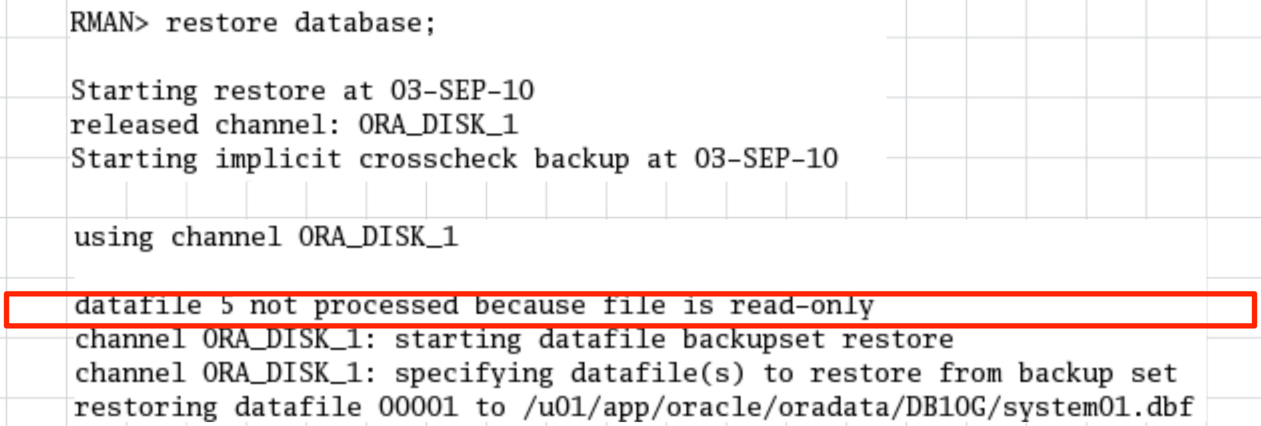
```
SQL>
SQL> select rf.file#, d.name, rf.error from v$recover_file rf, v$datafile d
  2  where rf.file# = d.file#;

no rows selected

SQL>
```

Recovering Read-Only Tablespaces

- **Problem:**
  - Full database restore is performed
  - Read-only tablespaces were ignored by RMAN during restore
- **Solution:**
  - Use CHECK READONLY option during database restore
  - Alternatively, explicitly restore the tablespaces which are in read-only mode

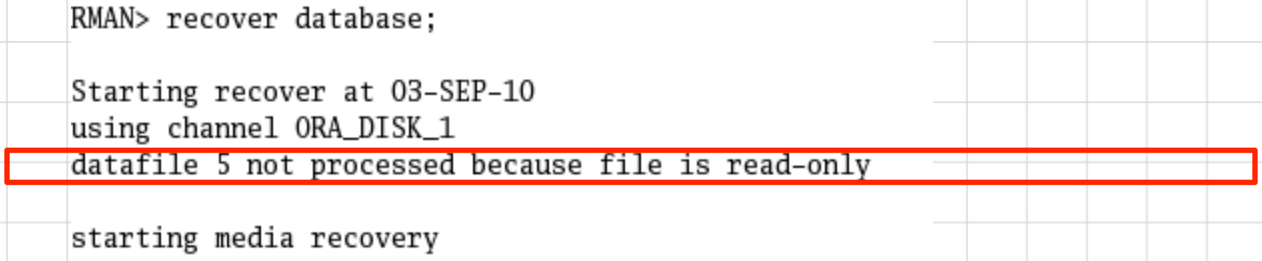
Recovering Read-Only Tablespaces

- By default “restore database” will ignore tablespaces in read-only mode



```
RMAN> restore database;

Starting restore at 03-SEP-10
released channel: ORA_DISK_1
Starting implicit crosscheck backup at 03-SEP-10

using channel ORA_DISK_1

datafile 5 not processed because file is read-only
channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
restoring datafile 00001 to /u01/app/oracle/oradata/DB10G/system01.dbf
```

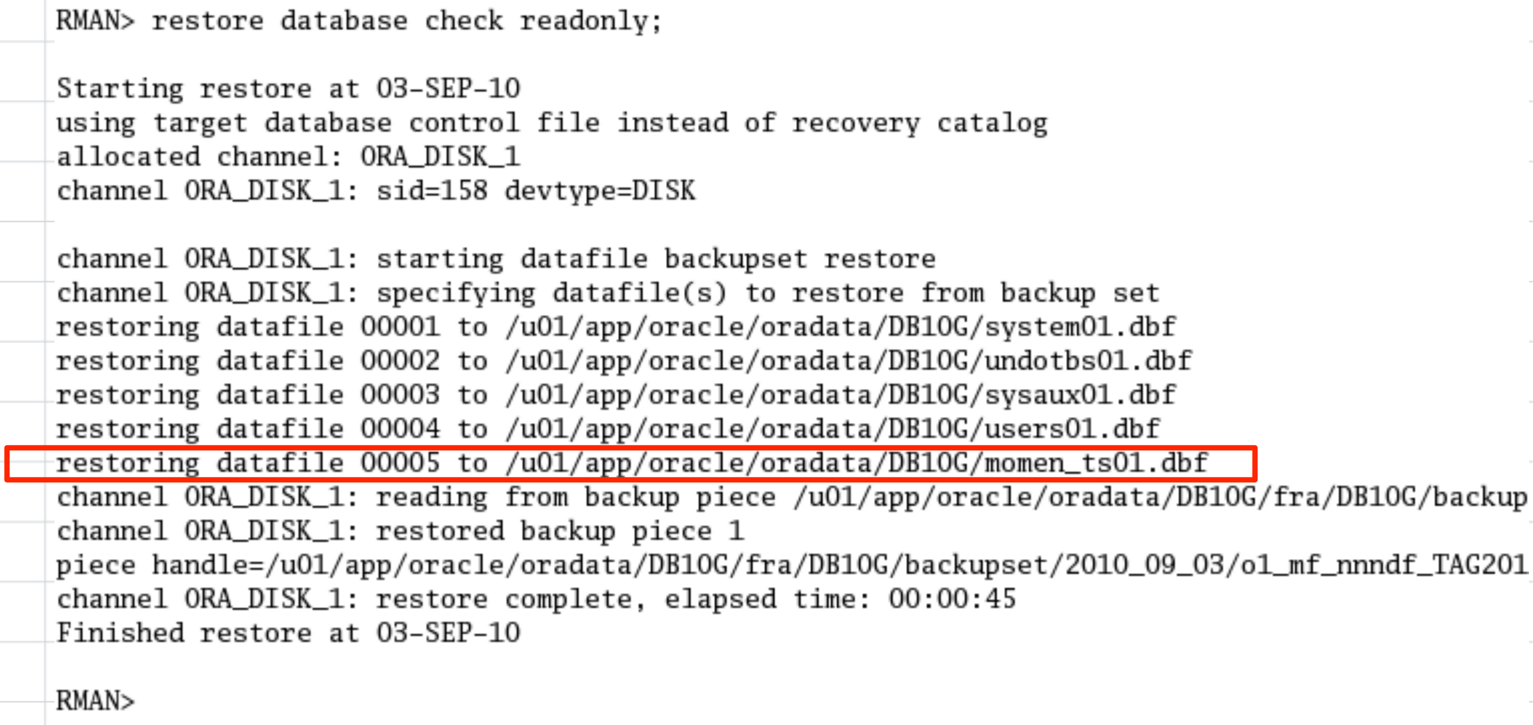
- Also “recover database” command will ignore tablespaces in read-only mode



```
RMAN> recover database;

Starting recover at 03-SEP-10
using channel ORA_DISK_1
datafile 5 not processed because file is read-only

starting media recovery
```

Recovering Read-Only Tablespaces

- **Restore database using CHECK READONLY option**



```
RMAN> restore database check readonly;

Starting restore at 03-SEP-10
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=158 devtype=DISK

channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
restoring datafile 00001 to /u01/app/oracle/oradata/DB10G/system01.dbf
restoring datafile 00002 to /u01/app/oracle/oradata/DB10G/undotbs01.dbf
restoring datafile 00003 to /u01/app/oracle/oradata/DB10G/sysaux01.dbf
restoring datafile 00004 to /u01/app/oracle/oradata/DB10G/users01.dbf
restoring datafile 00005 to /u01/app/oracle/oradata/DB10G/momen_ts01.dbf
channel ORA_DISK_1: reading from backup piece /u01/app/oracle/oradata/DB10G/fra/DB10G/backup
channel ORA_DISK_1: restored backup piece 1
piece handle=/u01/app/oracle/oradata/DB10G/fra/DB10G/backupset/2010_09_03/o1_mf_nnndf_TAG201
channel ORA_DISK_1: restore complete, elapsed time: 00:00:45
Finished restore at 03-SEP-10

RMAN>
```

Recovering Temporary Tablespaces

- As tempfiles aren't checkpointed, we don't need to back them up. We can recreate them at any point after the database has been restored, recovered and opened

- Temporary datafiles that belong to locally managed temporary tablespaces are automatically re-created during database recovery. This eliminates the need to manually create temporary tablespaces after recovery

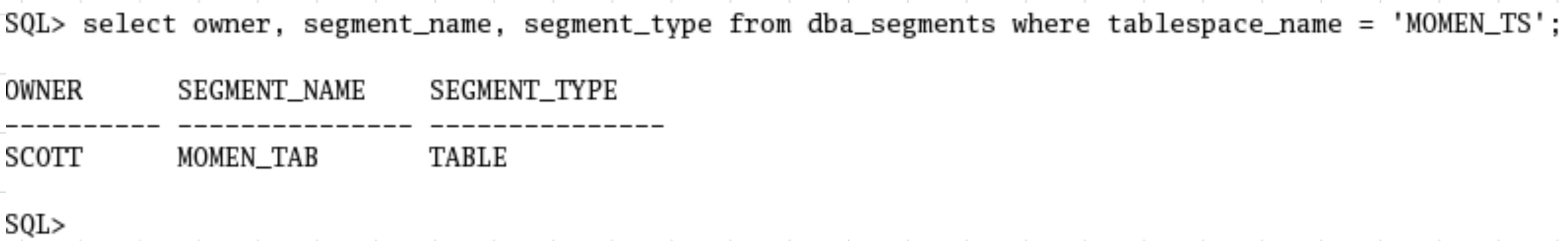
Tablespace Point-In-Time Recovery (TSPITR)

- **Problem:**
  - A TRUNCATE TABLE statement was erroneously run in production
  - Database is UP
  - Valid database backup’s exist
- **Solution:**
  - List all the objects residing in the affected tablespace
  - Identify and resolve any dependencies
  - Backup all the objects that will be lost
  - Create an auxiliary destination
  - Recover the tablespace
  - Bring the tablespace online

Tablespace Point-In-Time Recovery (TSPITR)

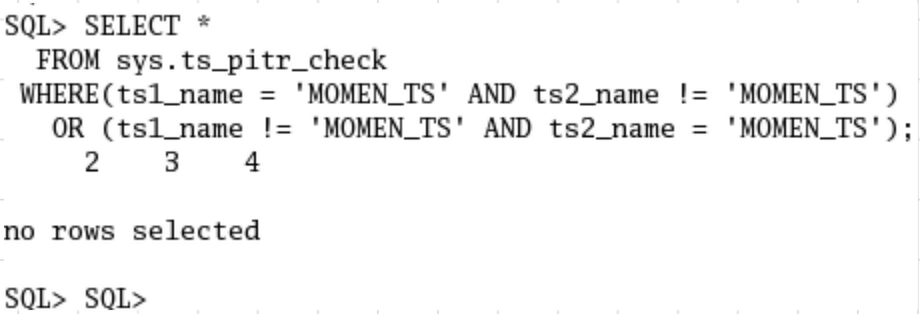
Solution:

- List all the objects residing in the tablespace



```
SQL> select owner, segment_name, segment_type from dba_segments where tablespace_name = 'MOMEN_TS';

OWNER      SEGMENT_NAME     SEGMENT_TYPE
---------- ---------------- ---------------
SCOTT      MOMEN_TAB        TABLE

SQL>
```

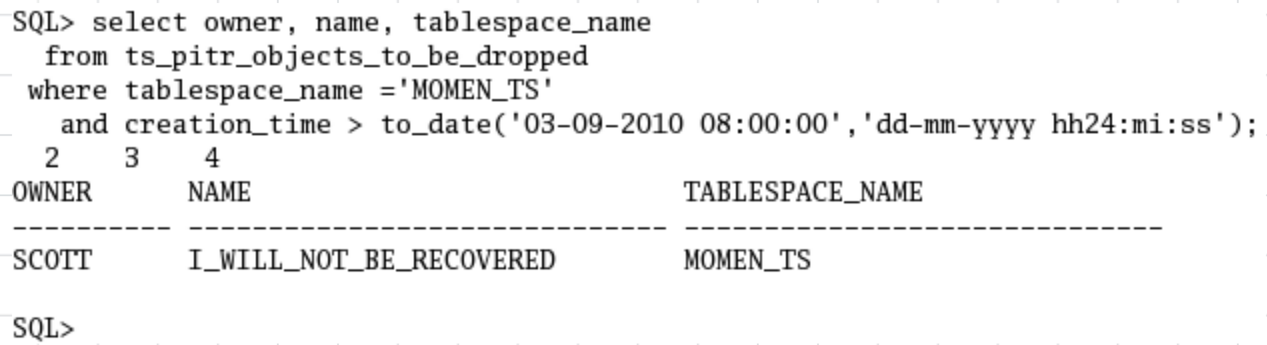
- Identify and resolve any dependencies



```
SQL> SELECT *
  FROM sys.ts_pitr_check
 WHERE(ts1_name = 'MOMEN_TS' AND ts2_name != 'MOMEN_TS')
   OR (ts1_name != 'MOMEN_TS' AND ts2_name = 'MOMEN_TS');
     2    3    4

no rows selected

SQL> SQL>
```

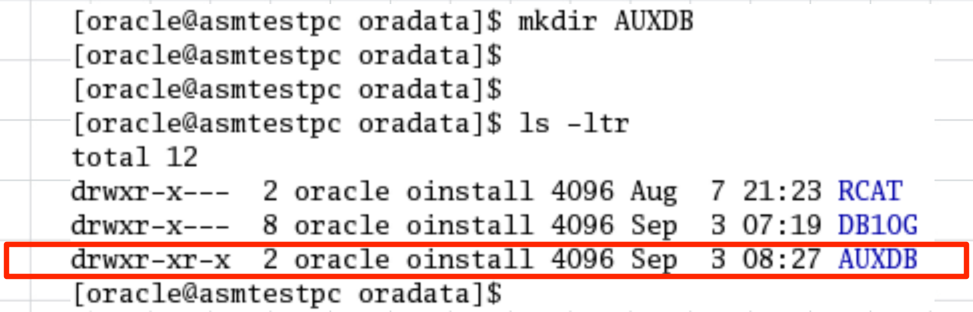
Tablespace Point-In-Time Recovery (TSPITR)

- Backup all the objects that will be lost
  - Identify objects what will be lost during TSPITR by querying TS_PITR_OBJECTS_TO_BE_DROPPED view on the primary database



```
SQL> select owner, name, tablespace_name
  from ts_pitr_objects_to_be_dropped
 where tablespace_name ='MOMEN_TS'
   and creation_time > to_date('03-09-2010 08:00:00','dd-mm-yyyy hh24:mi:ss');
  2    3    4
OWNER      NAME                           TABLESPACE_NAME
---------- ------------------------------ ------------------------------
SCOTT      I_WILL_NOT_BE_RECOVERED        MOMEN_TS

SQL>
```

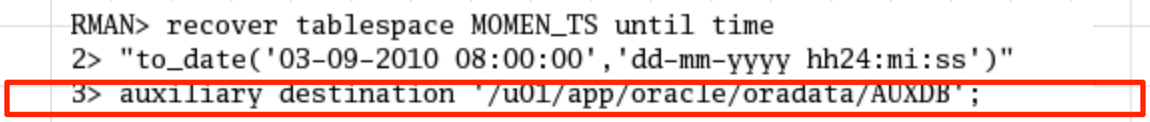
- Create an Auxiliary destination



```
[oracle@asmtestpc oradata]$ mkdir AUXDB
[oracle@asmtestpc oradata]$
[oracle@asmtestpc oradata]$
[oracle@asmtestpc oradata]$ ls -ltr
total 12
drwxr-x---  2 oracle oinstall 4096 Aug  7 21:23 RCAT
drwxr-x---  8 oracle oinstall 4096 Sep  3 07:19 DB10G
drwxr-xr-x  2 oracle oinstall 4096 Sep  3 08:27 AUXDB
[oracle@asmtestpc oradata]$
```

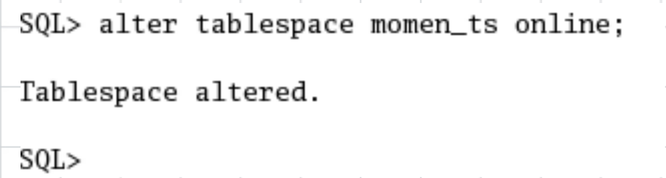
Tablespace Point-In-Time Recovery (TSPITR)

- Recover the tablespace



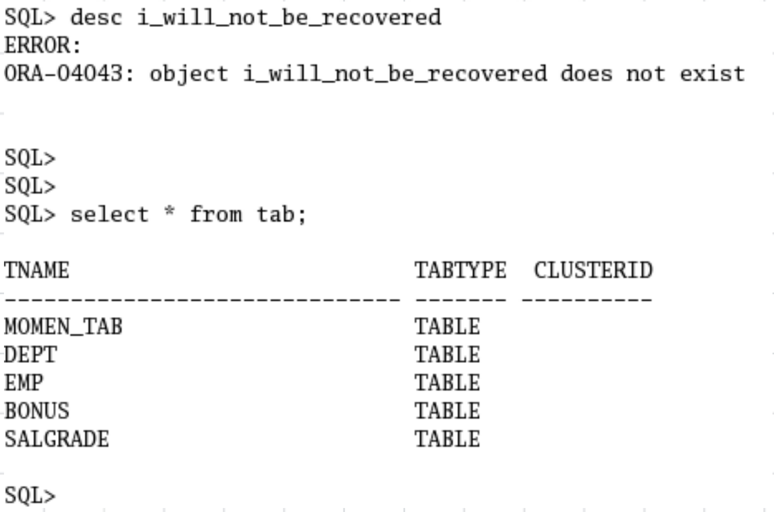
```
RMAN> recover tablespace MOMEN_TS until time
2> "to_date('03-09-2010 08:00:00','dd-mm-yyyy hh24:mi:ss')"
3> auxiliary destination '/u01/app/oracle/oradata/AUXDB';
```

- Bring the tablespace online



```
SQL> alter tablespace momen_ts online;

Tablespace altered.

SQL>
```

- Verify TSPITR process



```
SQL> desc i_will_not_be_recovered
ERROR:
ORA-04043: object i_will_not_be_recovered does not exist


SQL>
SQL>
SQL> select * from tab;

TNAME                          TABTYPE  CLUSTERID
------------------------------ ------- ----------
MOMEN_TAB                      TABLE
DEPT                           TABLE
EMP                            TABLE
BONUS                          TABLE
SALGRADE                       TABLE

SQL>
```

Tablespace Point-In-Time Recovery (TSPITR)

How the Recover command in TSPITR works?

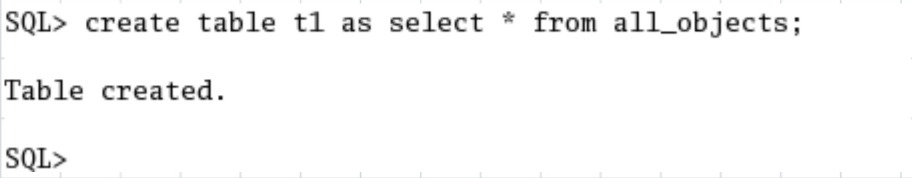
1. Creates an automatic instance
2. Restores Control File in the auxiliary (AUXDB) location
3. Restores and recovers the tablespaces SYSTEM, UNDO, and data tablespace (MOMEN_TS)
4. Export tablespace metadata in Recovery Set
5. Cleans auxiliary instance

Flashback Database

- **Problem:**
  - An erroneous transaction was performed (TRUNCATE)
  - Database is in Flashback mode
- **Solution:**
  - Identify the SCN to flashback to
  - Shutdown database
  - Start database in MOUNT mode
  - Flashback database
  - Open database with RESETLOGS option
  - Verify table contents

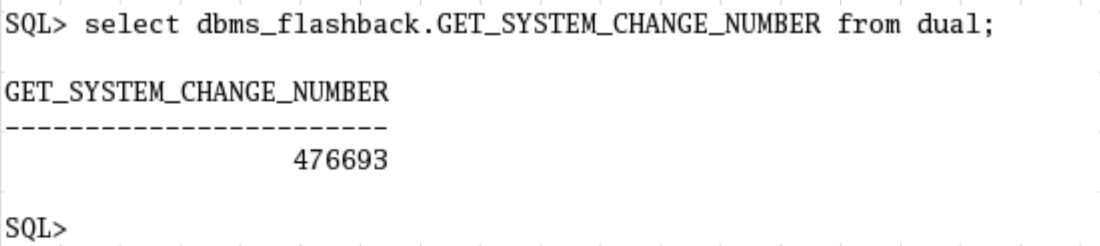
Flashback Database

- **Scenario:**
  - Create a table (T1) and insert some records



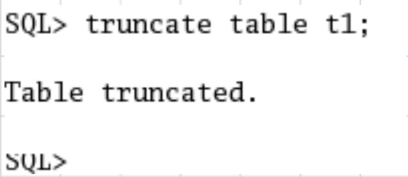
```
SQL> create table t1 as select * from all_objects;

Table created.

SQL>
```

  - Make a note of SCN



```
SQL> select dbms_flashback.GET_SYSTEM_CHANGE_NUMBER from dual;

GET_SYSTEM_CHANGE_NUMBER
------------------------
                  476693

SQL>
```

  - TRUNCATE table T1



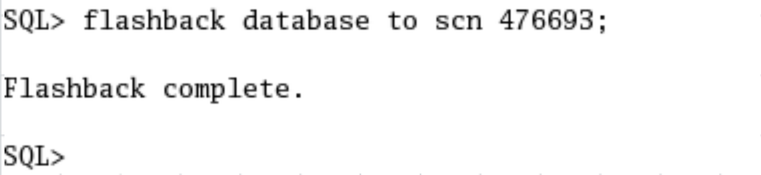
```
SQL> truncate table t1;

Table truncated.

SQL>
```

Flashback Database

- **Solution:**
  - Identify the SCN to flashback to
    - SCN = 476693
  - Shutdown database
  - Start database in MOUNT mode

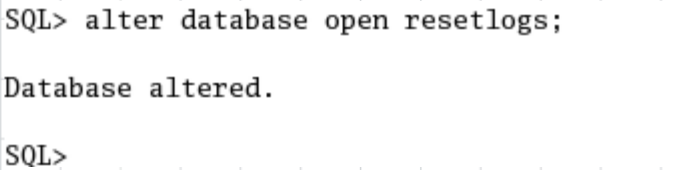
Flashback Database

- **Solution:**
  - Flashback database



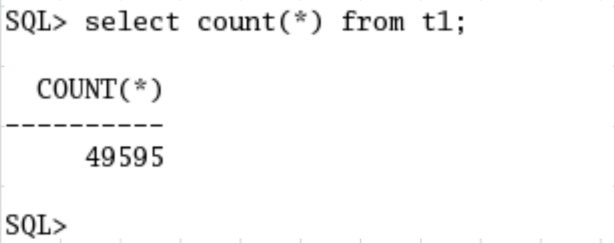
```
SQL> flashback database to scn 476693;

Flashback complete.

SQL>
```

  - Open database with RESETLOGS option



```
SQL> alter database open resetlogs;

Database altered.

SQL>
```

  - Verify results



```
SQL> select count(*) from t1;

  COUNT(*)
----------
     49595

SQL>
```

Advanced Recovery Solutions

Advanced Recovery Solutions

- Recovering Datafiles Not Backed Up
- Recovering through RESETLOGS
- Recovering to a Restore Point
- Recovering to a Previous Incarnation
- Partial Restore of a Database
- Block Recovery

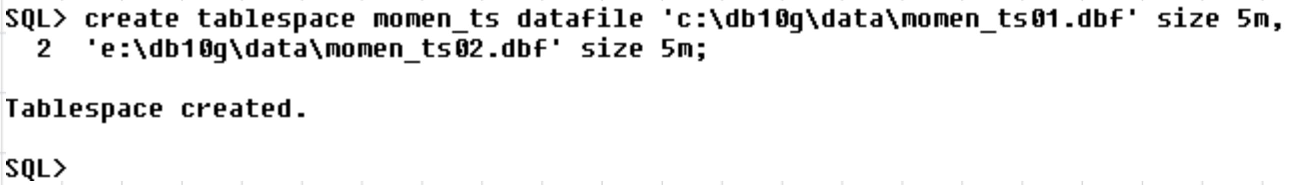
Recovering Datafiles Not Backed Up

- **Problem:**
  - New datafile is added to a tablespace (MOMEN_TS)
  - Datafile was lost before it could be backed up
  - Valid database backup exists
  - All Archive logs exist
  - Database is UP
- **Solution:**
  - List datafiles that need recovery
  - Restore datafile
  - Recover tablespace
  - Verify contents

Recovering Datafiles Not Backed Up

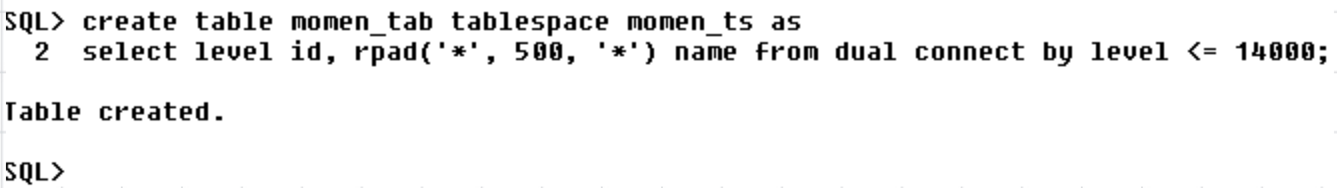
Simulating Loss of Datafile

- Create a tablespace (MOMEN_TS) with one of the datafiles residing in pen drive



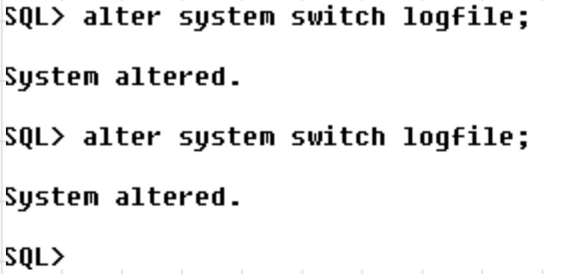
```
SQL> create tablespace momen_ts datafile 'c:\db10g\data\momen_ts01.dbf' size 5m,
  2  'e:\db10g\data\momen_ts02.dbf' size 5m;

Tablespace created.

SQL>
```

- Create and populate a table (MOMEN_TAB) in the above tablespace



```
SQL> create table momen_tab tablespace momen_ts as
  2  select level id, rpad('*', 500, '*') name from dual connect by level <= 14000;

Table created.

SQL>
```

- Switch archive logs

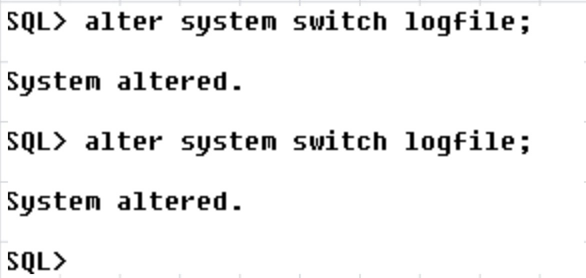


```
SQL> alter system switch logfile;

System altered.

SQL> alter system switch logfile;

System altered.

SQL>
```

Recovering Datafiles Not Backed Up

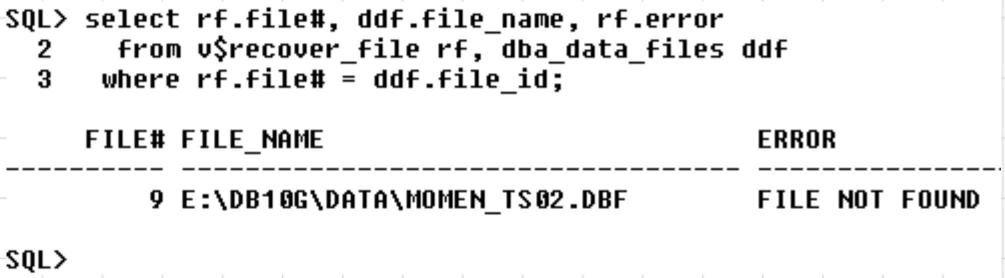
Simulating Loss of Datafile (cont …)

- Unplug the pen drive
- Switch archive logs



```
SQL> alter system switch logfile;

System altered.

SQL> alter system switch logfile;

System altered.

SQL>
```

- Verify the status

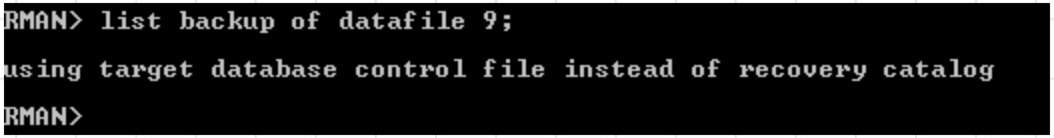


```
SQL> select rf.file#, ddf.file_name, rf.error
  2    from v$recover_file rf, dba_data_files ddf
  3   where rf.file# = ddf.file_id;

     FILE# FILE_NAME                                  ERROR
---------- ------------------------------------------ ---------------
         9 E:\DB10G\DATA\MOMEN_TS02.DBF               FILE NOT FOUND

SQL>
```

Recovering Datafiles Not Backed Up

Simulating Loss of Datafile (cont …)

- Datafile backup exists?



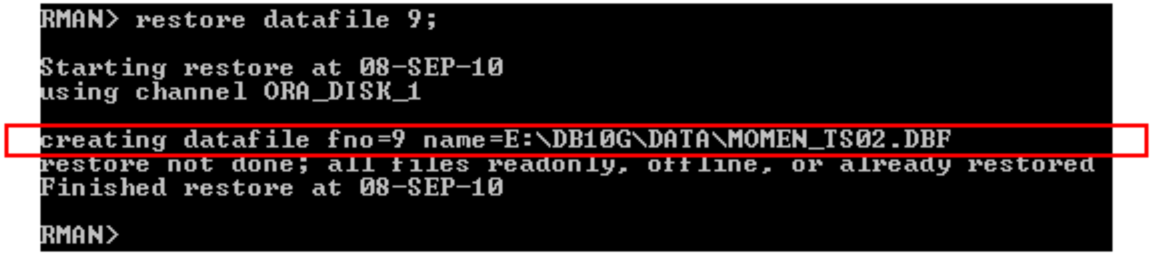
```
RMAN> list backup of datafile 9;
using target database control file instead of recovery catalog
RMAN>
```

- Plug the pen drive and delete the datafile (MOMEN_TS02.DBF)

Recovering Datafiles Not Backed Up

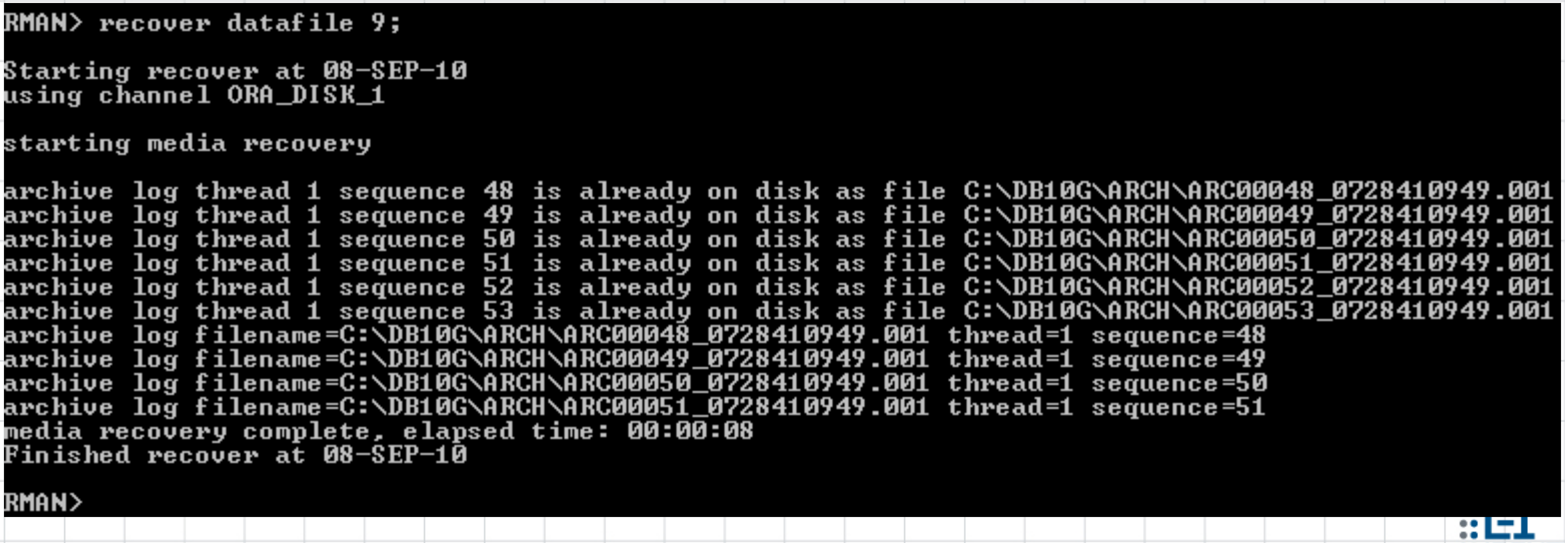
Solution:

- Restore datafile



```
RMAN> restore datafile 9;

Starting restore at 08-SEP-10
using channel ORA_DISK_1

creating datafile fno=9 name=E:\DB10G\DATA\MOMEN_TS02.DBF
restore not done; all files readonly, offline, or already restored
Finished restore at 08-SEP-10

RMAN>
```

- Recover datafile

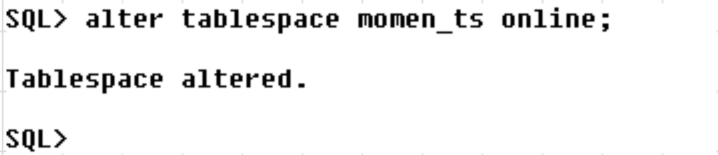


```
RMAN> recover datafile 9;

Starting recover at 08-SEP-10
using channel ORA_DISK_1

starting media recovery

archive log thread 1 sequence 48 is already on disk as file C:\DB10G\ARCH\ARC00048_0728410949.001
archive log thread 1 sequence 49 is already on disk as file C:\DB10G\ARCH\ARC00049_0728410949.001
archive log thread 1 sequence 50 is already on disk as file C:\DB10G\ARCH\ARC00050_0728410949.001
archive log thread 1 sequence 51 is already on disk as file C:\DB10G\ARCH\ARC00051_0728410949.001
archive log thread 1 sequence 52 is already on disk as file C:\DB10G\ARCH\ARC00052_0728410949.001
archive log thread 1 sequence 53 is already on disk as file C:\DB10G\ARCH\ARC00053_0728410949.001
archive log filename=C:\DB10G\ARCH\ARC00048_0728410949.001 thread=1 sequence=48
archive log filename=C:\DB10G\ARCH\ARC00049_0728410949.001 thread=1 sequence=49
archive log filename=C:\DB10G\ARCH\ARC00050_0728410949.001 thread=1 sequence=50
archive log filename=C:\DB10G\ARCH\ARC00051_0728410949.001 thread=1 sequence=51
media recovery complete, elapsed time: 00:00:08
Finished recover at 08-SEP-10

RMAN>
```

Recovering Datafiles Not Backed Up

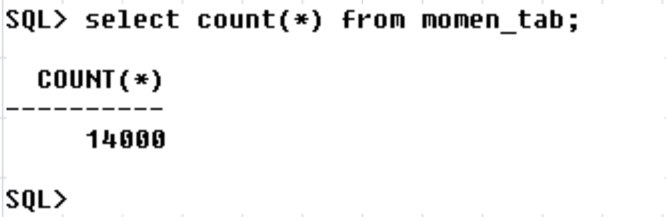
Solution:

- Bring the tablespace online



```
SQL> alter tablespace momen_ts online;

Tablespace altered.

SQL>
```

- Verify restore and recovery



```
SQL> select count(*) from momen_tab;

  COUNT(*)
----------
     14000

SQL>
```

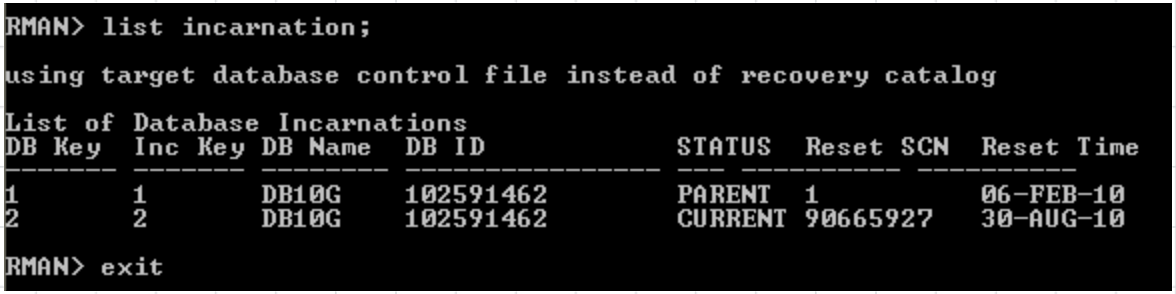
Recovering through RESETLOGS

- **Problem:**
  - Incomplete database recovery was performed (RESETLOGS)
  - Soon after restore completed, you suffered from another media failure
  - Backup was not performed after opening database with RESETLOGS option
  - All the generated archive logs exist
- **Solution:**
  - Start database in NOMOUNT mode
  - Restore Control File
  - MOUNT database
  - Restore database
  - Recover database
  - Open database with RESETLOGS option

Recovering through RESETLOGS

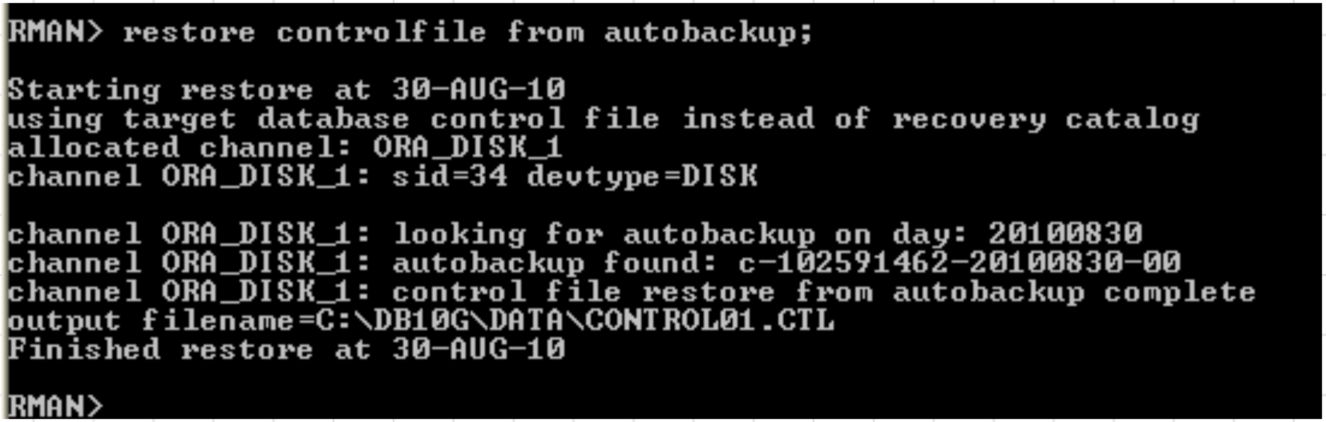
Solution:

- List incarnations of the current database



```
RMAN> list incarnation;

using target database control file instead of recovery catalog

List of Database Incarnations
DB Key  Inc Key DB Name  DB ID            STATUS  Reset SCN  Reset Time
------- ------- -------- ---------------- --- ---------- ----------
1       1       DB10G    102591462        PARENT  1          06-FEB-10
2       2       DB10G    102591462        CURRENT 90665927   30-AUG-10

RMAN> exit
```

- Restore Control File

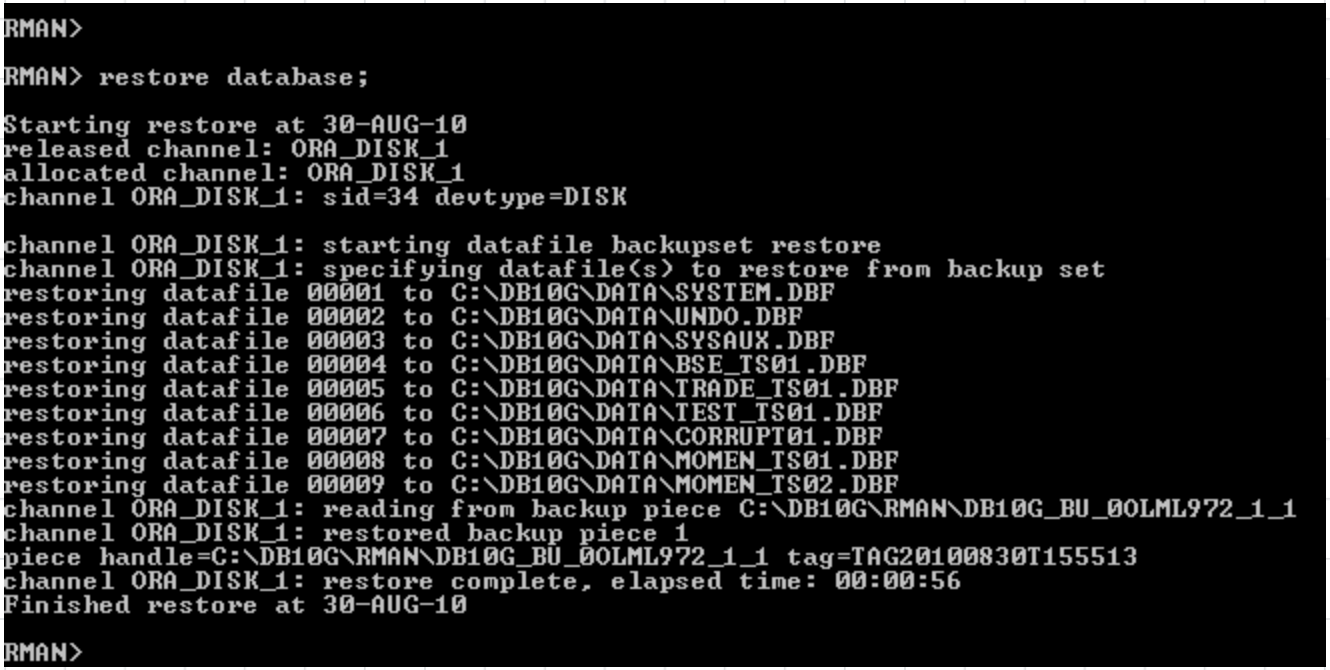


```
RMAN> restore controlfile from autobackup;

Starting restore at 30-AUG-10
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=34 devtype=DISK

channel ORA_DISK_1: looking for autobackup on day: 20100830
channel ORA_DISK_1: autobackup found: c-102591462-20100830-00
channel ORA_DISK_1: control file restore from autobackup complete
output filename=C:\DB10G\DATA\CONTROL01.CTL
Finished restore at 30-AUG-10

RMAN>
```

Recovering through RESETLOGS

Solution:

- MOUNT database
- Restore database

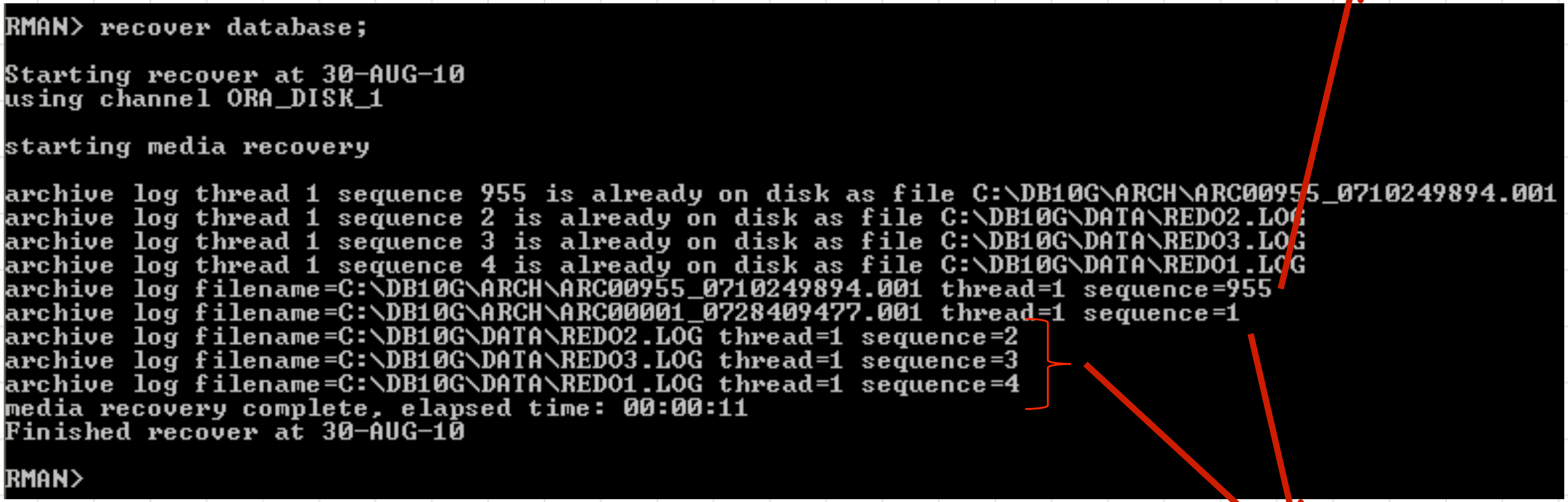


```
RMAN>

RMAN> restore database;

Starting restore at 30-AUG-10
released channel: ORA_DISK_1
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=34 devtype=DISK

channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
restoring datafile 00001 to C:\DB10G\DATA\SYSTEM.DBF
restoring datafile 00002 to C:\DB10G\DATA\UNDO.DBF
restoring datafile 00003 to C:\DB10G\DATA\SYSAUX.DBF
restoring datafile 00004 to C:\DB10G\DATA\BSE_TS01.DBF
restoring datafile 00005 to C:\DB10G\DATA\TRADE_TS01.DBF
restoring datafile 00006 to C:\DB10G\DATA\TEST_TS01.DBF
restoring datafile 00007 to C:\DB10G\DATA\CORRUPT01.DBF
restoring datafile 00008 to C:\DB10G\DATA\MOMEN_TS01.DBF
restoring datafile 00009 to C:\DB10G\DATA\MOMEN_TS02.DBF
channel ORA_DISK_1: reading from backup piece C:\DB10G\RMAN\DB10G_BU_0OLML972_1_1
channel ORA_DISK_1: restored backup piece 1
piece handle=C:\DB10G\RMAN\DB10G_BU_0OLML972_1_1 tag=TAG20100830T155513
channel ORA_DISK_1: restore complete, elapsed time: 00:00:56
Finished restore at 30-AUG-10

RMAN>
```

Recovering through RESETLOGS

Solution:

- Recover database



```
RMAN> recover database;

Starting recover at 30-AUG-10
using channel ORA_DISK_1

starting media recovery

archive log thread 1 sequence 955 is already on disk as file C:\DB10G\ARCH\ARC00955_0710249894.001
archive log thread 1 sequence 2 is already on disk as file C:\DB10G\DATA\REDO2.LOG
archive log thread 1 sequence 3 is already on disk as file C:\DB10G\DATA\REDO3.LOG
archive log thread 1 sequence 4 is already on disk as file C:\DB10G\DATA\REDO1.LOG
archive log filename=C:\DB10G\ARCH\ARC00955_0710249894.001 thread=1 sequence=955
archive log filename=C:\DB10G\ARCH\ARC00001_0728409477.001 thread=1 sequence=1
archive log filename=C:\DB10G\DATA\REDO2.LOG thread=1 sequence=2
archive log filename=C:\DB10G\DATA\REDO3.LOG thread=1 sequence=3
archive log filename=C:\DB10G\DATA\REDO1.LOG thread=1 sequence=4
media recovery complete, elapsed time: 00:00:11
Finished recover at 30-AUG-10

RMAN>
```

Archive Log (955) from PREVIOUS Incarnation

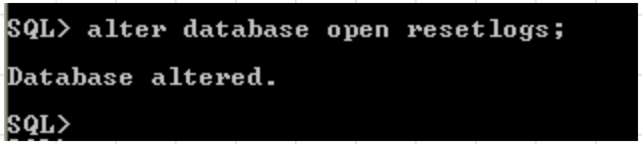
Archive Logs (1, 2, 3, & 4) from CURRENT Incarnation

- Open database with RESETLOGS option

Recovering through RESETLOGS

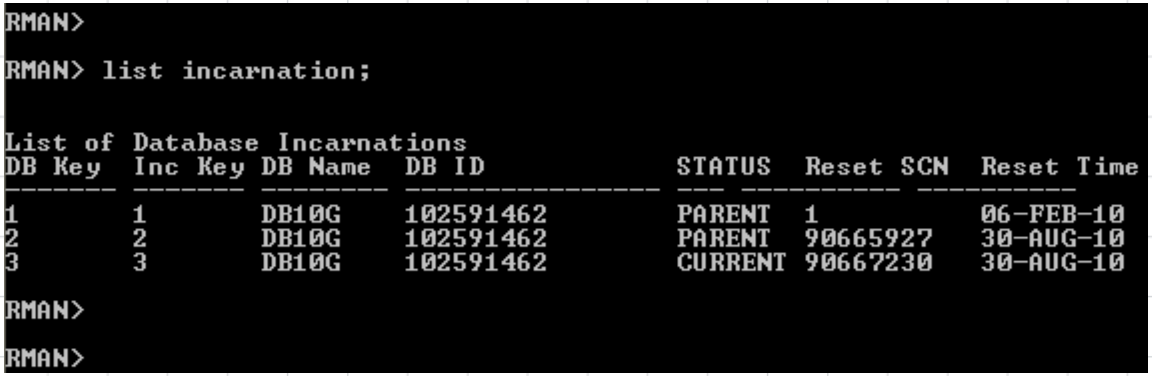
Solution:

- Open database with RESETLOGS option



```
SQL> alter database open resetlogs;

Database altered.

SQL>
```

- List database incarnations



```
RMAN>

RMAN> list incarnation;


List of Database Incarnations
DB Key  Inc Key DB Name  DB ID            STATUS  Reset SCN  Reset Time
------- ------- -------- ---------------- --- ---------- ----------
1       1       DB10G    102591462        PARENT  1          06-FEB-10
2       2       DB10G    102591462        PARENT  90665927   30-AUG-10
3       3       DB10G    102591462        CURRENT 90667230   30-AUG-10

RMAN>

RMAN>
```

Recovering to a Restore Point

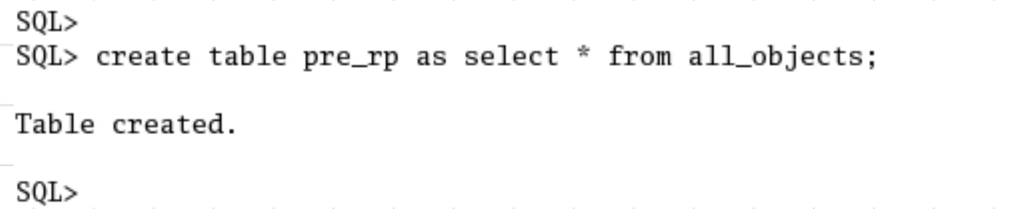
- **Problem:**
  - You have created a Restore Point
  - You want to restore database to the created Restore Point
  - Valid database backup exists

- **Solution:**
  - List Restore Points
  - MOUNT database
  - Restore database until Restore Point
  - Recover database until Restore Point
  - Open database with RESETLOGS option
  - Verify restore

Recovering to a Restore Point

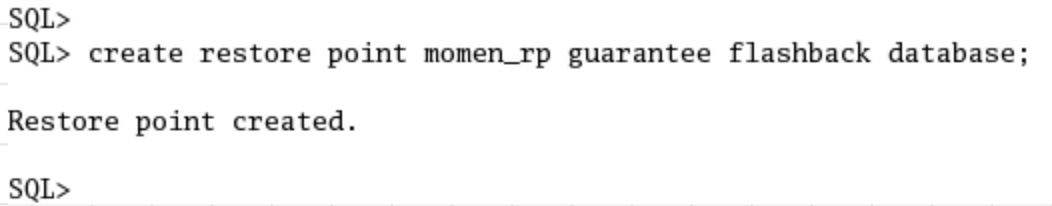
Simulating:

- Create a table (PRE_RP) and dump some records



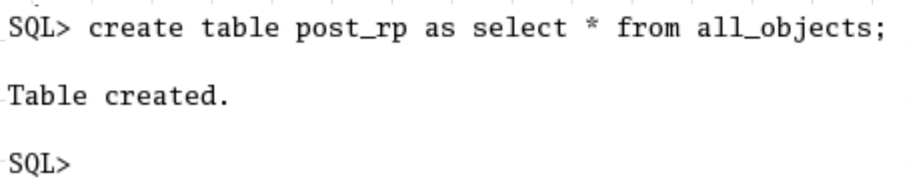
```
SQL>
SQL> create table pre_rp as select * from all_objects;

Table created.

SQL>
```

- Create a Restore Point



```
SQL>
SQL> create restore point momen_rp guarantee flashback database;

Restore point created.

SQL>
```

- Create a table (POST_RP) and dump some records

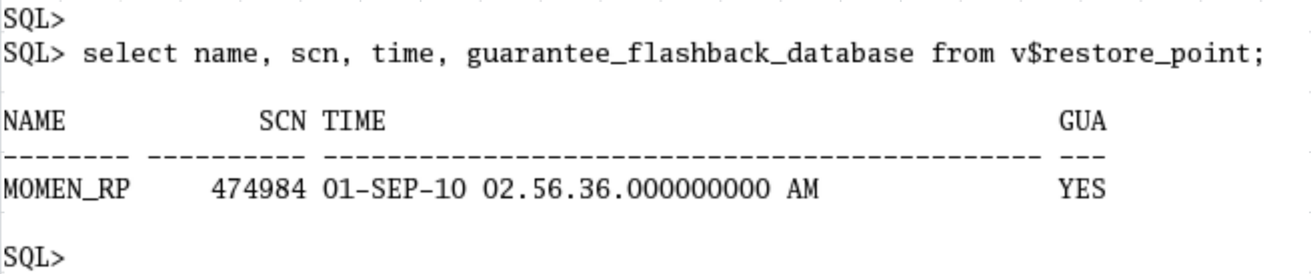


```
SQL> create table post_rp as select * from all_objects;

Table created.

SQL>
```

Recovering to a Restore Point

Solution:

- List Restore Points



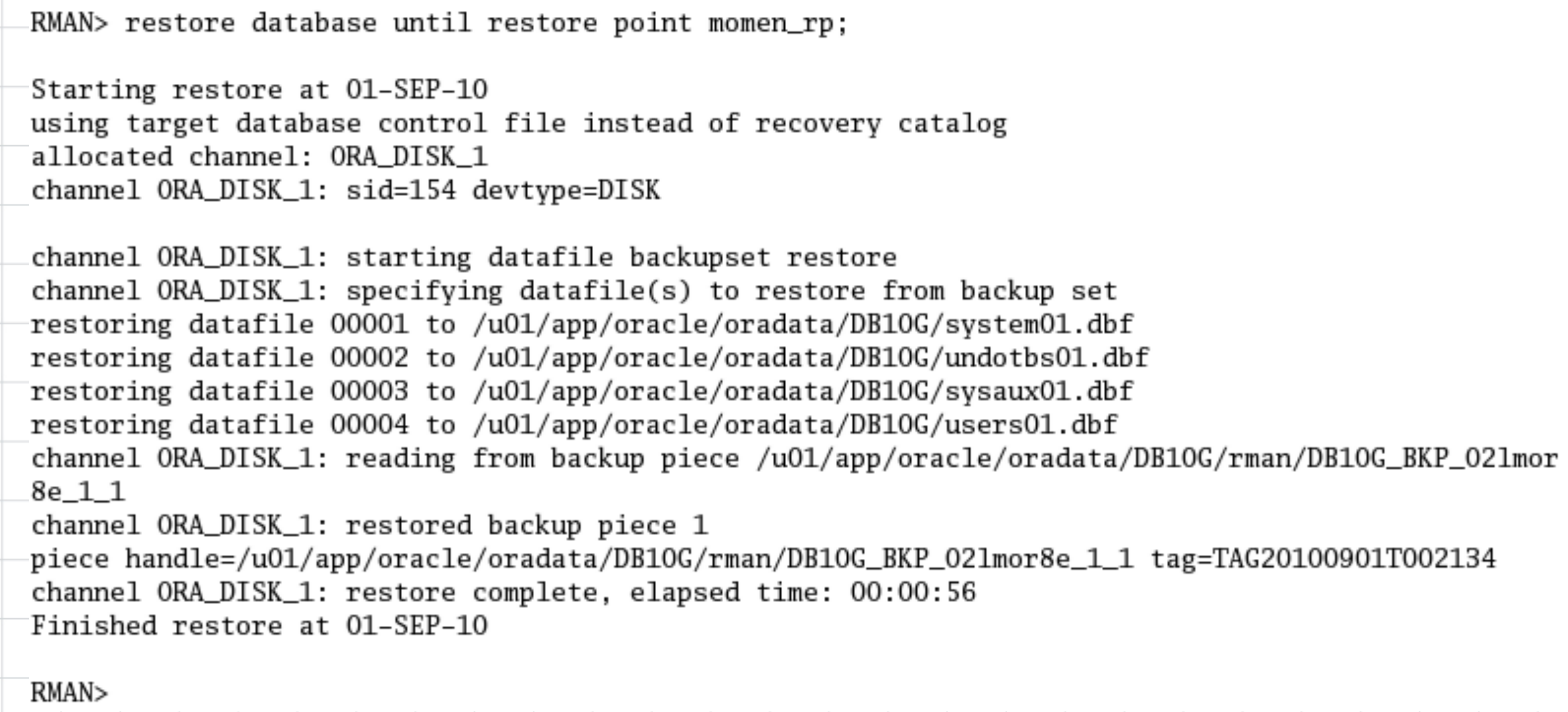
```
SQL>
SQL> select name, scn, time, guarantee_flashback_database from v$restore_point;

NAME            SCN TIME                                           GUA
-------- ---------- ---------------------------------------------- ---
MOMEN_RP     474984 01-SEP-10 02.56.36.000000000 AM                YES

SQL>
```

- Shutdown and start database in MOUNT mode

Recovering to a Restore Point

Solution:

- Restore database until Restore Point

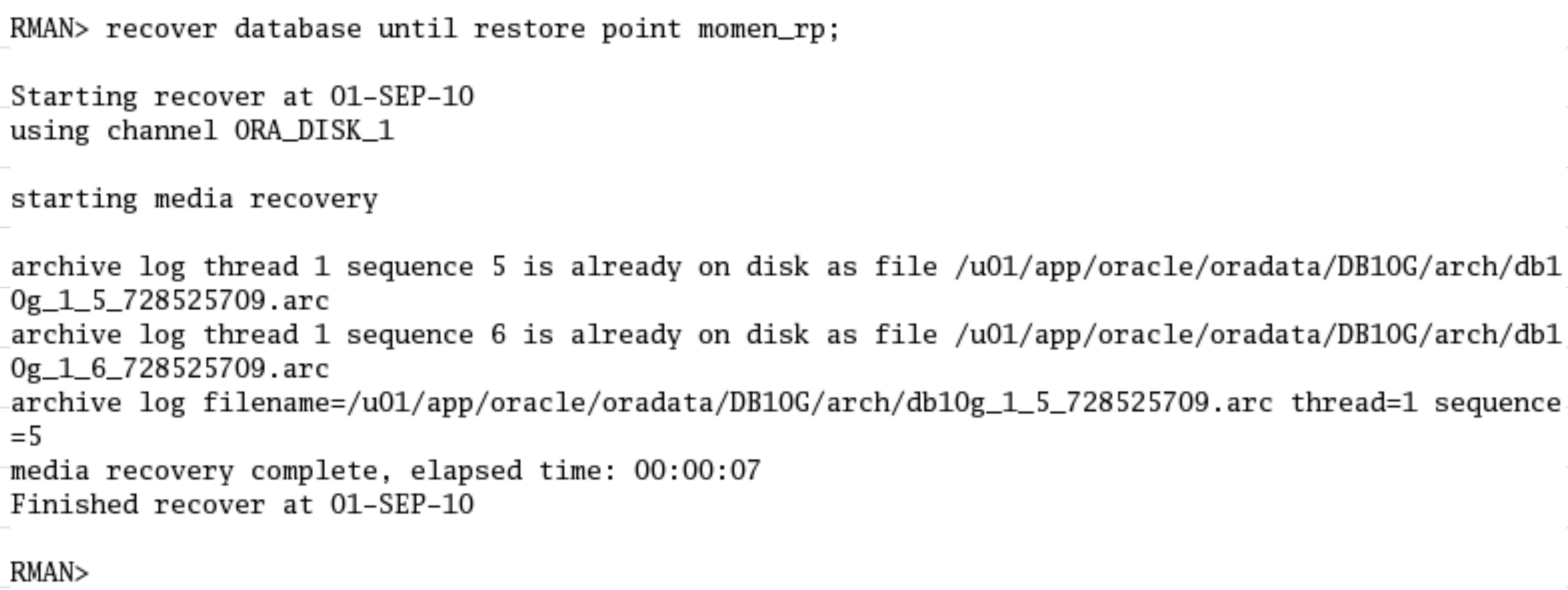


```
RMAN> restore database until restore point momen_rp;

Starting restore at 01-SEP-10
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=154 devtype=DISK

channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
restoring datafile 00001 to /u01/app/oracle/oradata/DB10G/system01.dbf
restoring datafile 00002 to /u01/app/oracle/oradata/DB10G/undotbs01.dbf
restoring datafile 00003 to /u01/app/oracle/oradata/DB10G/sysaux01.dbf
restoring datafile 00004 to /u01/app/oracle/oradata/DB10G/users01.dbf
channel ORA_DISK_1: reading from backup piece /u01/app/oracle/oradata/DB10G/rman/DB10G_BKP_02lmor
8e_1_1
channel ORA_DISK_1: restored backup piece 1
piece handle=/u01/app/oracle/oradata/DB10G/rman/DB10G_BKP_02lmor8e_1_1 tag=TAG20100901T002134
channel ORA_DISK_1: restore complete, elapsed time: 00:00:56
Finished restore at 01-SEP-10

RMAN>
```

Recovering to a Restore Point

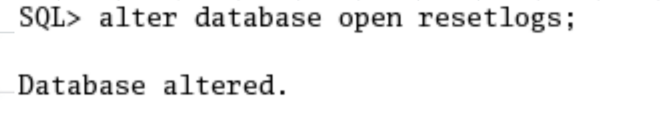
Solution:

- Recover database until Restore Point



```
RMAN> recover database until restore point momen_rp;

Starting recover at 01-SEP-10
using channel ORA_DISK_1

starting media recovery

archive log thread 1 sequence 5 is already on disk as file /u01/app/oracle/oradata/DB10G/arch/db1
0g_1_5_728525709.arc
archive log thread 1 sequence 6 is already on disk as file /u01/app/oracle/oradata/DB10G/arch/db1
0g_1_6_728525709.arc
archive log filename=/u01/app/oracle/oradata/DB10G/arch/db10g_1_5_728525709.arc thread=1 sequence
=5
media recovery complete, elapsed time: 00:00:07
Finished recover at 01-SEP-10

RMAN>
```

- Open database with RESETLOGS option

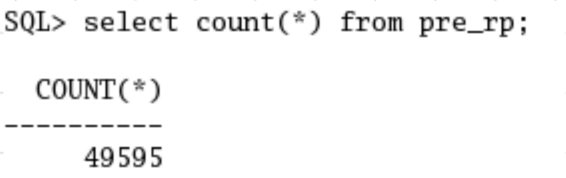


```
SQL> alter database open resetlogs;

Database altered.
```

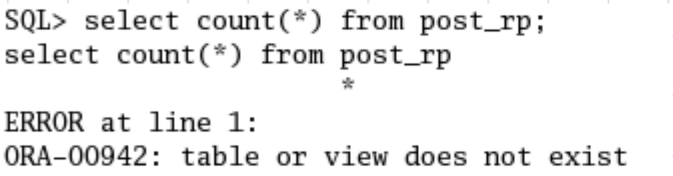
Recovering to a Restore Point

Solution:

- Verify Restore

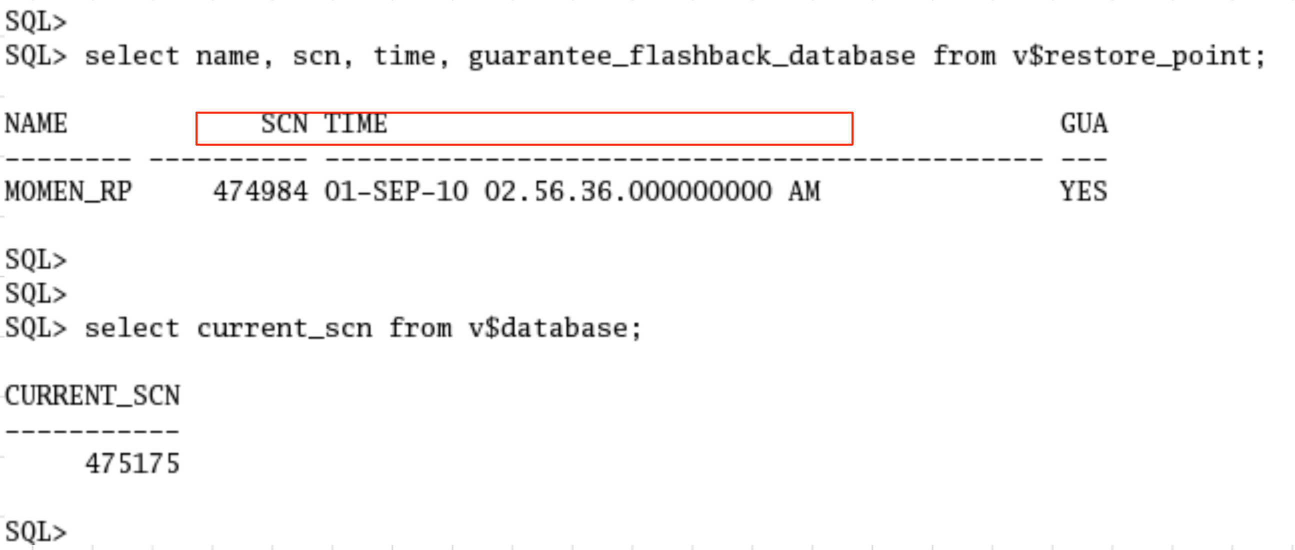


```
SQL> select count(*) from pre_rp;

  COUNT(*)
----------
     49595
```



```
SQL> select count(*) from post_rp;
select count(*) from post_rp
                     *
ERROR at line 1:
ORA-00942: table or view does not exist
```

- Make a note of the time as this will be used for the next recovery scenario



```
SQL>
SQL> select name, scn, time, guarantee_flashback_database from v$restore_point;

NAME            SCN TIME                                          GUA
-------- ---------- --------------------------------------------- ---
MOMEN_RP     474984 01-SEP-10 02.56.36.000000000 AM               YES

SQL>
SQL>
SQL> select current_scn from v$database;

CURRENT_SCN
-----------
     475175

SQL>
```

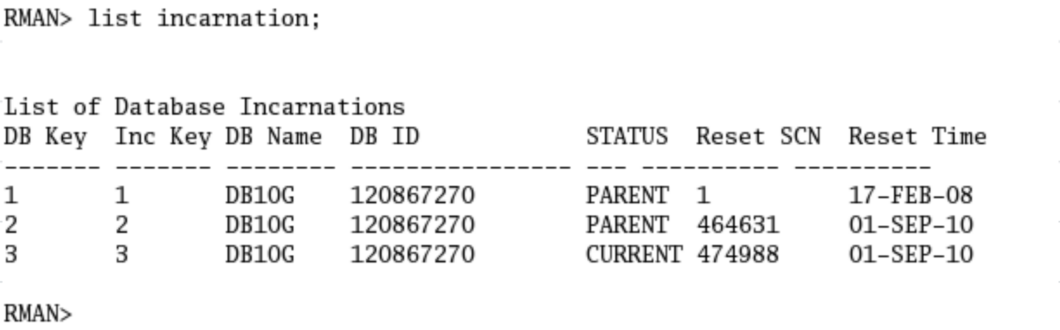
Recovering to a Previous Incarnation

- **Problem:**
  - You performed an incomplete recovery
  - Now you want to go back in time prior to the time when database was opened with resetlogs
  - Valid database backup exists
- **Solution:**
  - List database incarnations
  - Restore Control File
  - MOUNT database
  - Reset database incarnation
  - Restore database until Time
  - Recover database until time
  - Open database with RESETLOGS option
  - Verify restore

Recovering to a Previous Incarnation

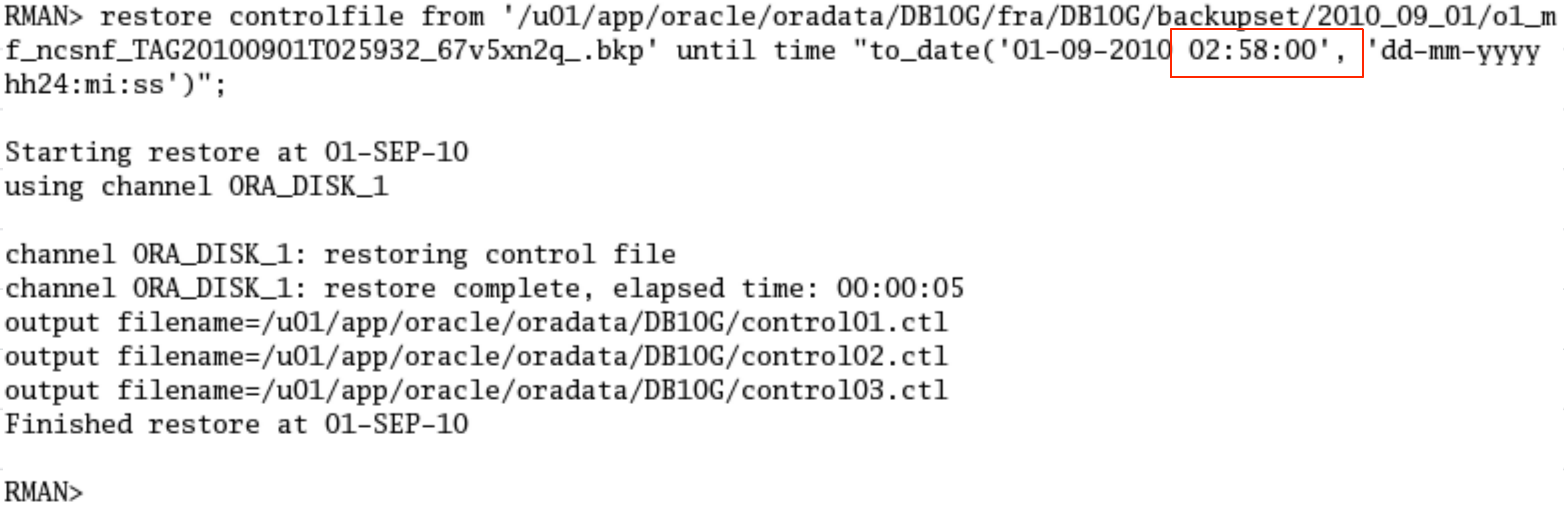
Solution:

- List database incarnations



```
RMAN> list incarnation;


List of Database Incarnations
DB Key  Inc Key DB Name  DB ID            STATUS  Reset SCN  Reset Time
------- ------- -------- ---------------- --- ---------- ----------
1       1       DB10G    120867270        PARENT  1          17-FEB-08
2       2       DB10G    120867270        PARENT  464631     01-SEP-10
3       3       DB10G    120867270        CURRENT 474988     01-SEP-10

RMAN>
```

- Restore Control File

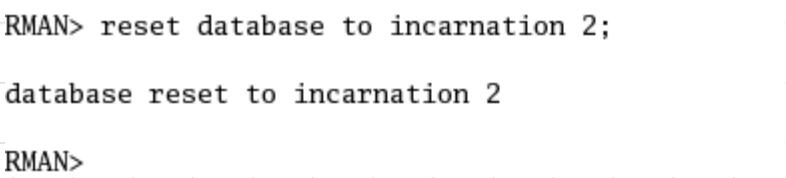


```
RMAN> restore controlfile from '/u01/app/oracle/oradata/DB10G/fra/DB10G/backupset/2010_09_01/o1_m
f_ncsnf_TAG20100901T025932_67v5xn2q_.bkp' until time "to_date('01-09-2010 02:58:00', 'dd-mm-yyyy
hh24:mi:ss')";

Starting restore at 01-SEP-10
using channel ORA_DISK_1

channel ORA_DISK_1: restoring control file
channel ORA_DISK_1: restore complete, elapsed time: 00:00:05
output filename=/u01/app/oracle/oradata/DB10G/control01.ctl
output filename=/u01/app/oracle/oradata/DB10G/control02.ctl
output filename=/u01/app/oracle/oradata/DB10G/control03.ctl
Finished restore at 01-SEP-10

RMAN>
```

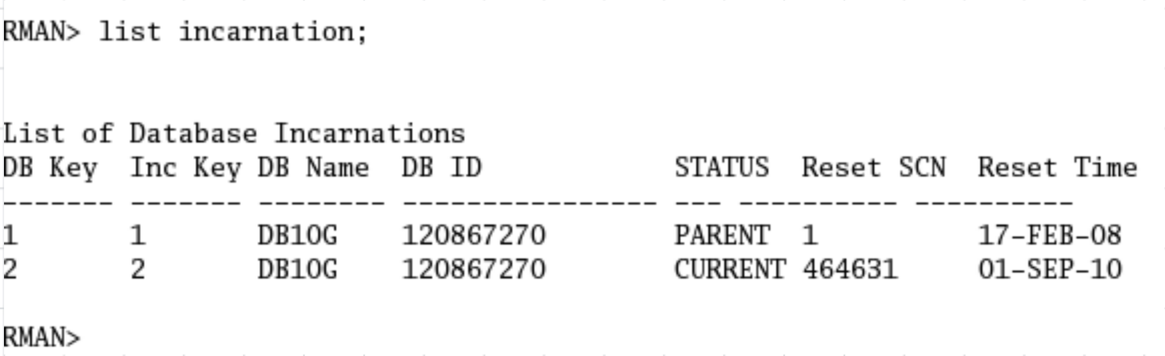
Recovering to a Previous Incarnation

Solution:

- MOUNT database
- Reset database incarnation



```
RMAN> reset database to incarnation 2;

database reset to incarnation 2

RMAN>
```

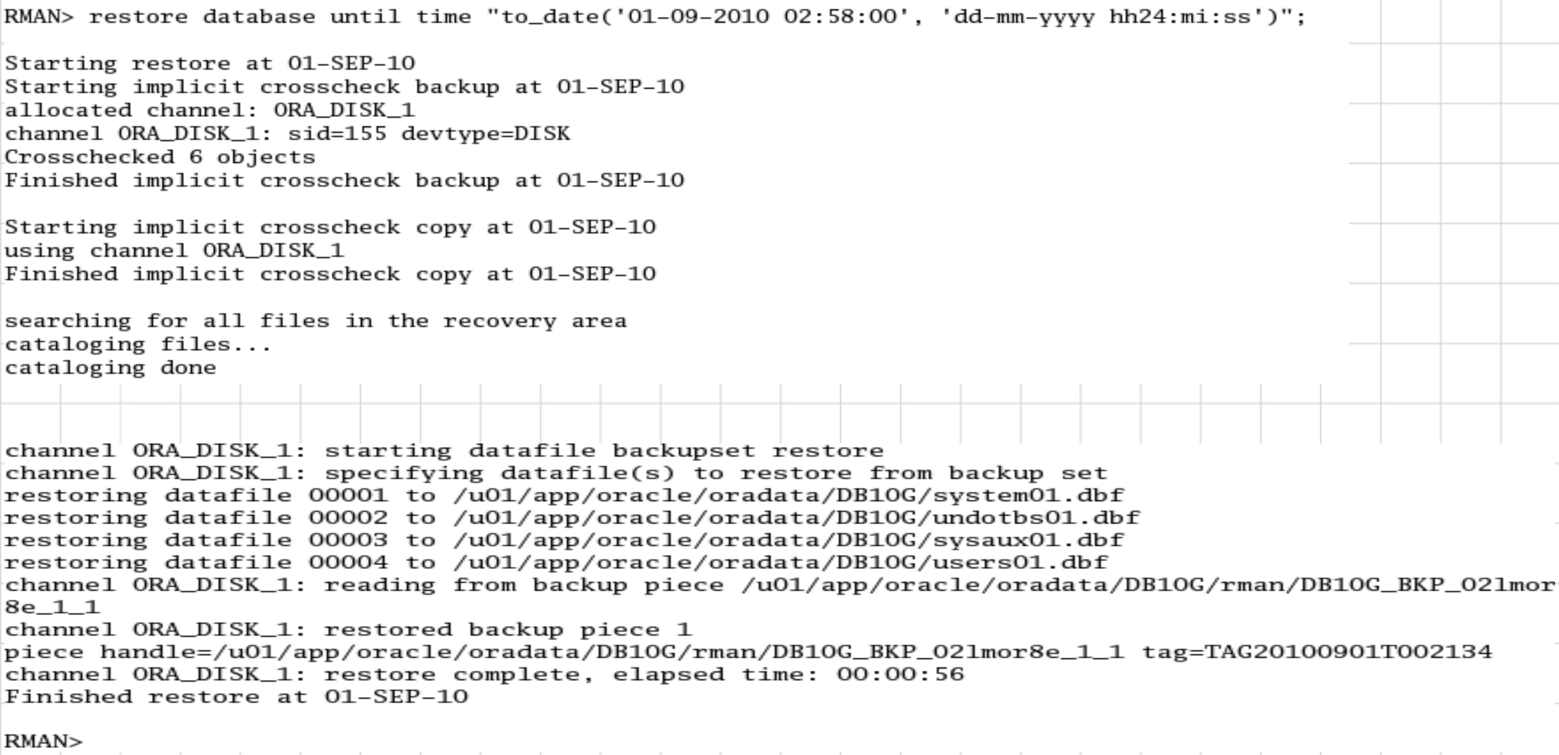


```
RMAN> list incarnation;


List of Database Incarnations
DB Key  Inc Key DB Name  DB ID            STATUS  Reset SCN  Reset Time
------- ------- -------- ---------------- --- ---------- ----------
1       1       DB10G    120867270        PARENT  1          17-FEB-08
2       2       DB10G    120867270        CURRENT 464631     01-SEP-10

RMAN>
```

Recovering to a Previous Incarnation

Solution:

– Restore database until Time

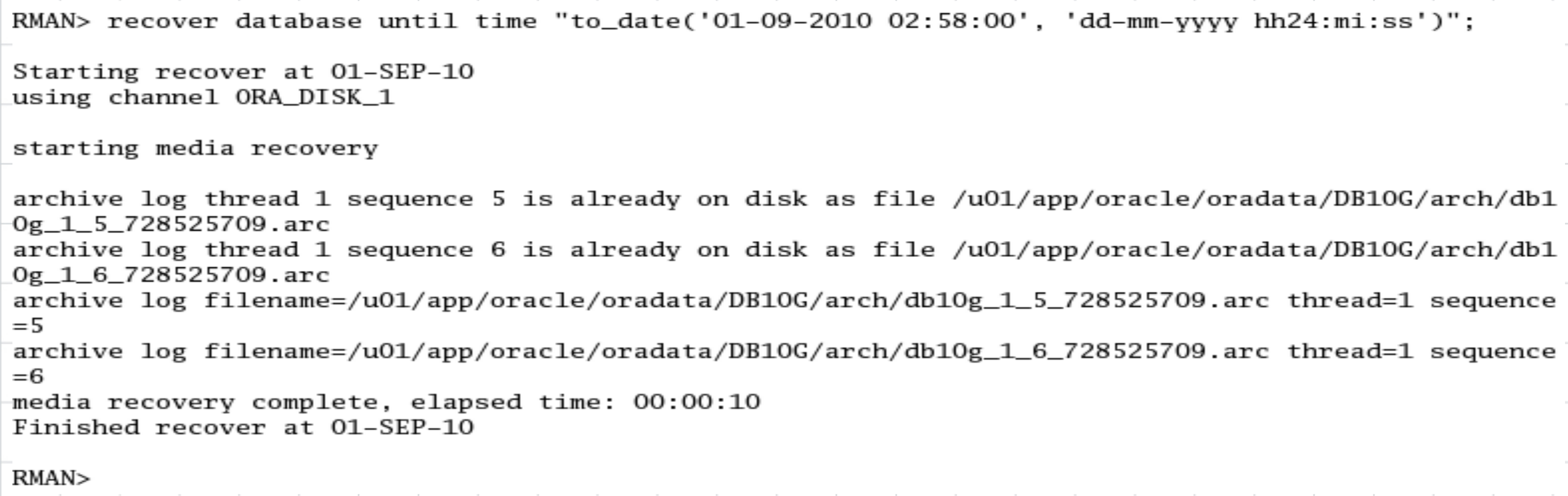


```
RMAN> restore database until time "to_date('01-09-2010 02:58:00', 'dd-mm-yyyy hh24:mi:ss')";

Starting restore at 01-SEP-10
Starting implicit crosscheck backup at 01-SEP-10
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=155 devtype=DISK
Crosschecked 6 objects
Finished implicit crosscheck backup at 01-SEP-10

Starting implicit crosscheck copy at 01-SEP-10
using channel ORA_DISK_1
Finished implicit crosscheck copy at 01-SEP-10

searching for all files in the recovery area
cataloging files...
cataloging done


channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
restoring datafile 00001 to /u01/app/oracle/oradata/DB10G/system01.dbf
restoring datafile 00002 to /u01/app/oracle/oradata/DB10G/undotbs01.dbf
restoring datafile 00003 to /u01/app/oracle/oradata/DB10G/sysaux01.dbf
restoring datafile 00004 to /u01/app/oracle/oradata/DB10G/users01.dbf
channel ORA_DISK_1: reading from backup piece /u01/app/oracle/oradata/DB10G/rman/DB10G_BKP_021mor
8e_1_1
channel ORA_DISK_1: restored backup piece 1
piece handle=/u01/app/oracle/oradata/DB10G/rman/DB10G_BKP_021mor8e_1_1 tag=TAG20100901T002134
channel ORA_DISK_1: restore complete, elapsed time: 00:00:56
Finished restore at 01-SEP-10

RMAN>
```

Recovering to a Previous Incarnation

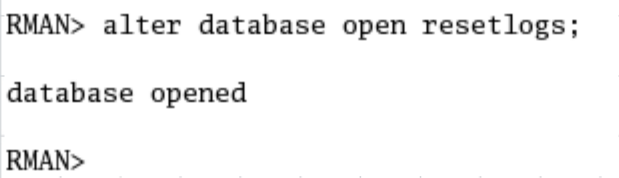
Solution:

- Recover database until time



```
RMAN> recover database until time "to_date('01-09-2010 02:58:00', 'dd-mm-yyyy hh24:mi:ss')";

Starting recover at 01-SEP-10
using channel ORA_DISK_1

starting media recovery

archive log thread 1 sequence 5 is already on disk as file /u01/app/oracle/oradata/DB10G/arch/db1
0g_1_5_728525709.arc
archive log thread 1 sequence 6 is already on disk as file /u01/app/oracle/oradata/DB10G/arch/db1
0g_1_6_728525709.arc
archive log filename=/u01/app/oracle/oradata/DB10G/arch/db10g_1_5_728525709.arc thread=1 sequence
=5
archive log filename=/u01/app/oracle/oradata/DB10G/arch/db10g_1_6_728525709.arc thread=1 sequence
=6
media recovery complete, elapsed time: 00:00:10
Finished recover at 01-SEP-10

RMAN>
```

- Open database with RESETLOGS option

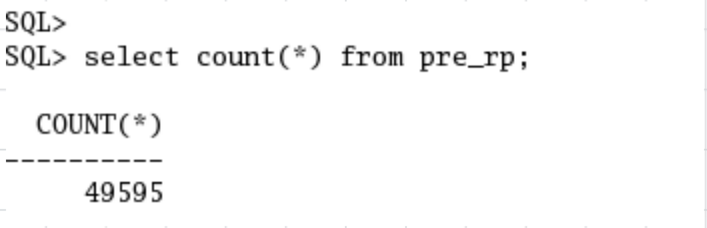


```
RMAN> alter database open resetlogs;

database opened

RMAN>
```

Recovering to a Previous Incarnation

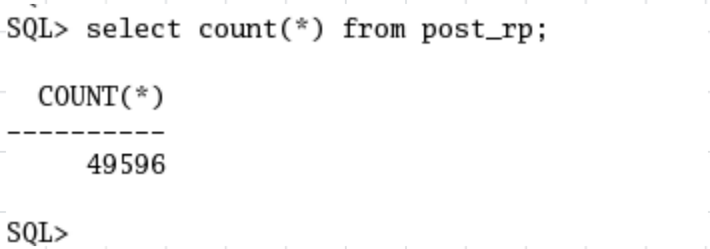
Solution:

- Verify restore



```
SQL>
SQL> select count(*) from pre_rp;

  COUNT(*)
----------
     49595
```

- We have managed to restore the POST_RP table which was created immediately after creating the Restore Point (MOMEN_RP)



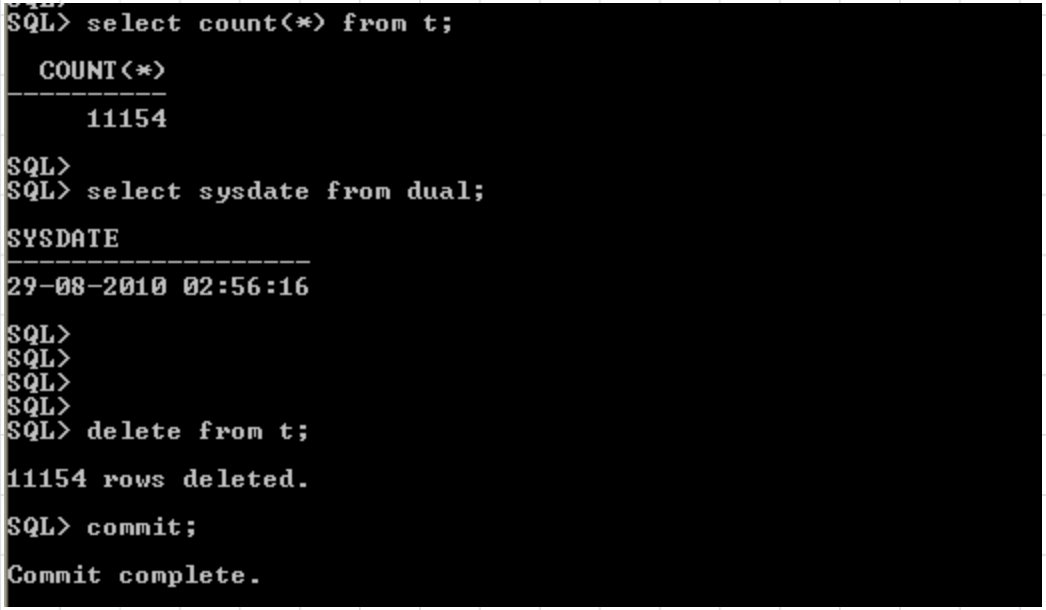
```
SQL> select count(*) from post_rp;

  COUNT(*)
----------
     49596

SQL>
```

Partial Restore of a Database

- **Problem:**
  - Database is very large
  - Need to recover few tables due to erroneous transaction
  - Valid database backups exist
- **Solution:**
  - Identify required tablespaces and datafiles
  - Copy init.ora to a different host
  - Start instance in NOMOUNT state
  - Restore control file
  - Restore required tablespaces
  - Delete the existing control file and create a new one with the required datafiles only
  - Recover database
  - Open database with RESETLOGS
  - Confirm contents of table T
  - Export the table (T) and import into production database

Partial Restore of a Database

- **Scenario:**
  - Note the number of records in the test table “T”
  - Make a note of the time
  - Perform erroneous transaction

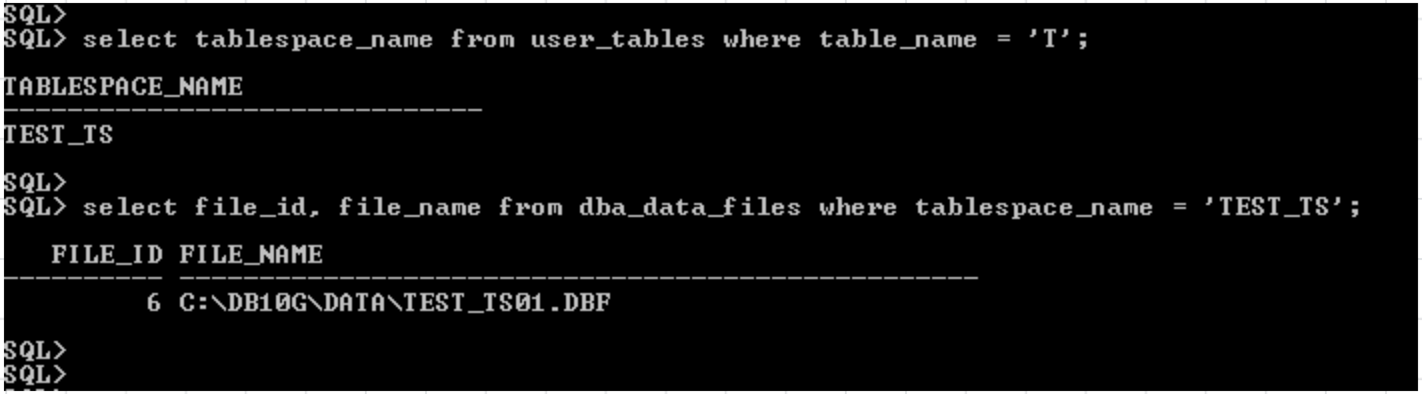


```
SQL> select count(*) from t;

  COUNT(*)
----------
     11154

SQL>
SQL> select sysdate from dual;

SYSDATE
-------------------
29-08-2010 02:56:16

SQL>
SQL>
SQL>
SQL>
SQL> delete from t;

11154 rows deleted.

SQL> commit;

Commit complete.
```

Partial Restore of a Database

Solution:

- Identify required tablespaces and datafiles



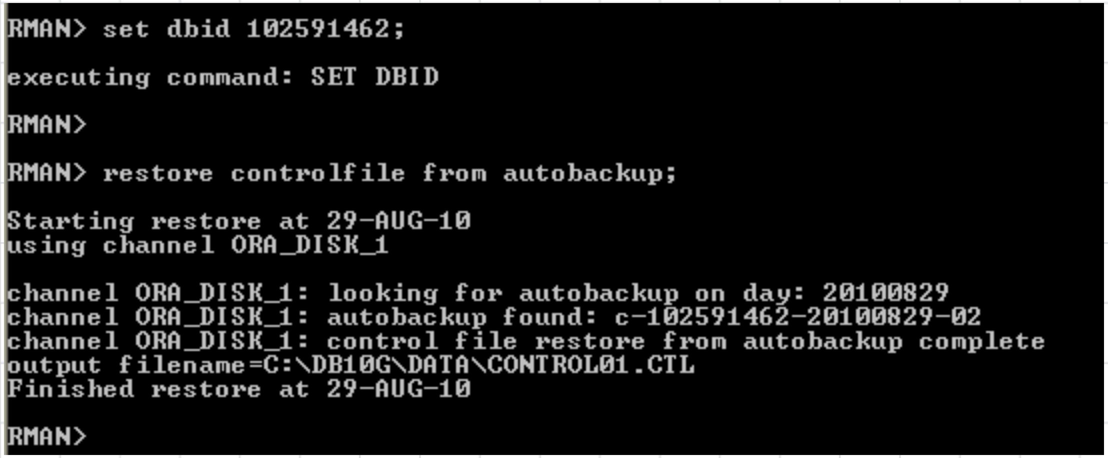
```
SQL>
SQL> select tablespace_name from user_tables where table_name = 'T';

TABLESPACE_NAME
------------------------------
TEST_TS

SQL>
SQL> select file_id, file_name from dba_data_files where tablespace_name = 'TEST_TS';

   FILE_ID FILE_NAME
---------- --------------------------------------------------
         6 C:\DB10G\DATA\TEST_TS01.DBF

SQL>
SQL>
```

- Copy init.ora to a different host
- Start instance in NOMOUNT state

Partial Restore of a Database

Solution:

- Restore Control File

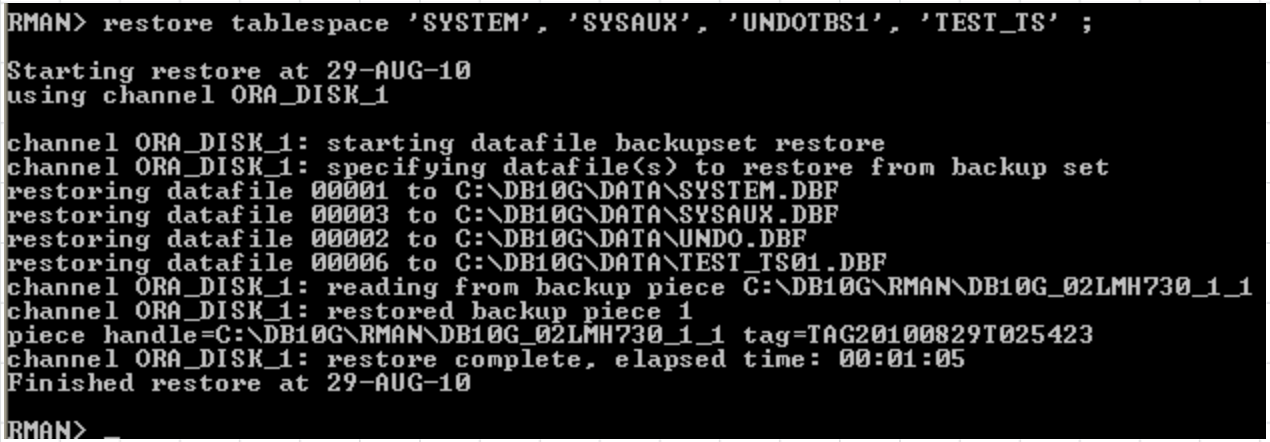


```
RMAN> set dbid 102591462;

executing command: SET DBID

RMAN>

RMAN> restore controlfile from autobackup;

Starting restore at 29-AUG-10
using channel ORA_DISK_1

channel ORA_DISK_1: looking for autobackup on day: 20100829
channel ORA_DISK_1: autobackup found: c-102591462-20100829-02
channel ORA_DISK_1: control file restore from autobackup complete
output filename=C:\DB10G\DATA\CONTROL01.CTL
Finished restore at 29-AUG-10

RMAN>
```

Partial Restore of a Database

Solution:

- Restore required tablespaces

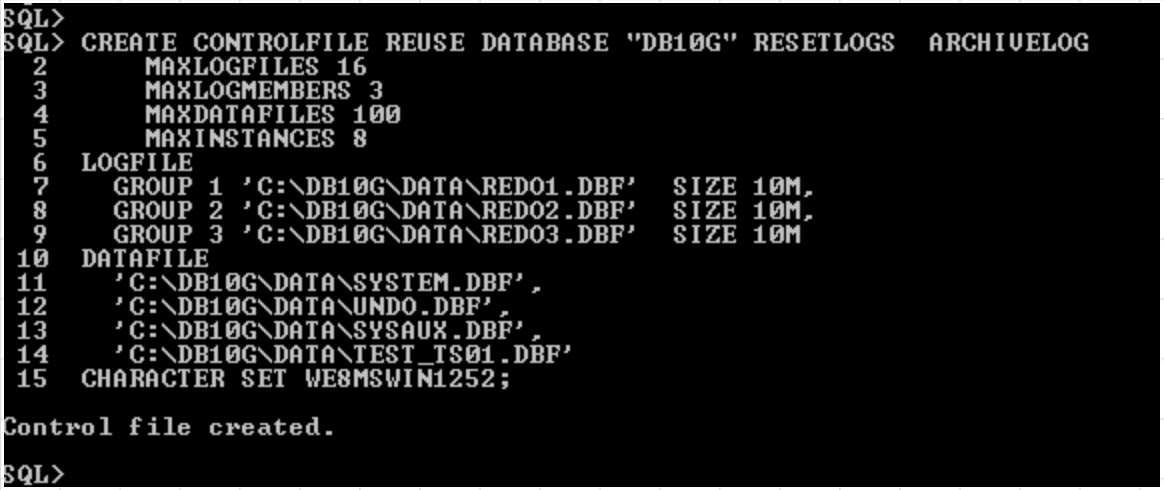


```
RMAN> restore tablespace 'SYSTEM', 'SYSAUX', 'UNDOTBS1', 'TEST_TS' ;

Starting restore at 29-AUG-10
using channel ORA_DISK_1

channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
restoring datafile 00001 to C:\DB10G\DATA\SYSTEM.DBF
restoring datafile 00003 to C:\DB10G\DATA\SYSAUX.DBF
restoring datafile 00002 to C:\DB10G\DATA\UNDO.DBF
restoring datafile 00006 to C:\DB10G\DATA\TEST_TS01.DBF
channel ORA_DISK_1: reading from backup piece C:\DB10G\RMAN\DB10G_02LMH730_1_1
channel ORA_DISK_1: restored backup piece 1
piece handle=C:\DB10G\RMAN\DB10G_02LMH730_1_1 tag=TAG20100829T025423
channel ORA_DISK_1: restore complete, elapsed time: 00:01:05
Finished restore at 29-AUG-10

RMAN> _
```

Partial Restore of a Database

Solution:

- Delete the existing control file and create a new one with the required datafiles only

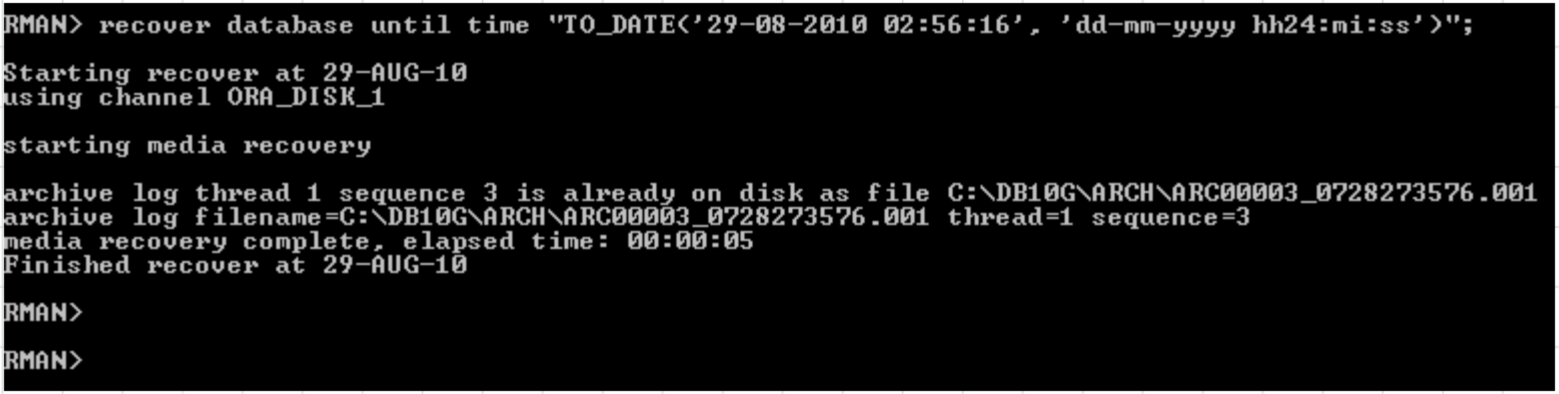


```
SQL>
SQL> CREATE CONTROLFILE REUSE DATABASE "DB10G" RESETLOGS  ARCHIVELOG
  2      MAXLOGFILES 16
  3      MAXLOGMEMBERS 3
  4      MAXDATAFILES 100
  5      MAXINSTANCES 8
  6  LOGFILE
  7    GROUP 1 'C:\DB10G\DATA\REDO1.DBF'  SIZE 10M,
  8    GROUP 2 'C:\DB10G\DATA\REDO2.DBF'  SIZE 10M,
  9    GROUP 3 'C:\DB10G\DATA\REDO3.DBF'  SIZE 10M
 10  DATAFILE
 11    'C:\DB10G\DATA\SYSTEM.DBF',
 12    'C:\DB10G\DATA\UNDO.DBF',
 13    'C:\DB10G\DATA\SYSAUX.DBF',
 14    'C:\DB10G\DATA\TEST_TS01.DBF'
 15  CHARACTER SET WE8MSWIN1252;

Control file created.

SQL>
```

Partial Restore of a Database

Solution:

- Recover database

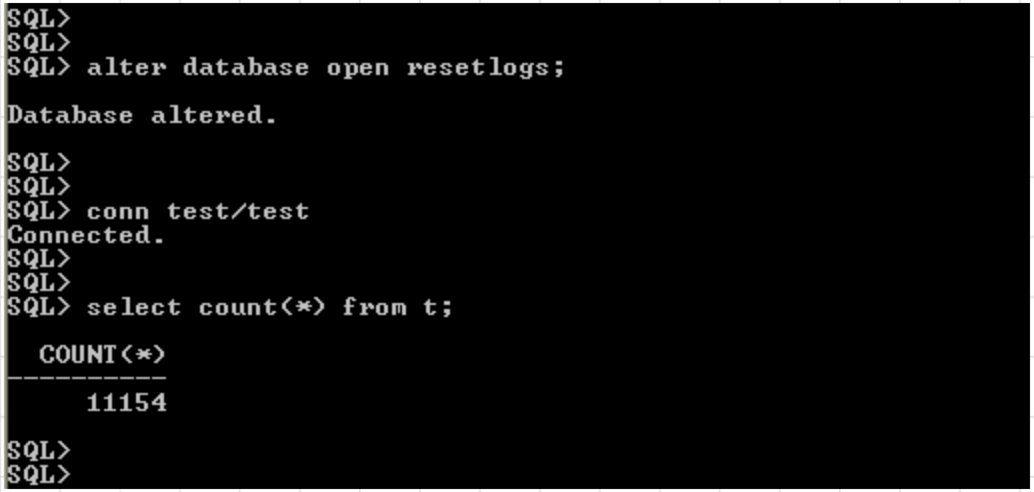


```
RMAN> recover database until time "TO_DATE('29-08-2010 02:56:16', 'dd-mm-yyyy hh24:mi:ss')";

Starting recover at 29-AUG-10
using channel ORA_DISK_1

starting media recovery

archive log thread 1 sequence 3 is already on disk as file C:\DB10G\ARCH\ARC00003_0728273576.001
archive log filename=C:\DB10G\ARCH\ARC00003_0728273576.001 thread=1 sequence=3
media recovery complete, elapsed time: 00:00:05
Finished recover at 29-AUG-10

RMAN>

RMAN>
```

Partial Restore of a Database

Solution:

- Open database with RESETLOGS
- Confirm contents of table T
- Export the table (T) and import into production database



```
SQL>
SQL>
SQL> alter database open resetlogs;

Database altered.

SQL>
SQL>
SQL> conn test/test
Connected.
SQL>
SQL>
SQL> select count(*) from t;

  COUNT(*)
----------
     11154

SQL>
SQL>
```

Block Recovery

- **Problem:**
  - Few data blocks are reported as corrupt
  - RMAN Backups are safe
- **Solution:**
  - Identify and list corrupt blocks
  - Perform Block Recovery
  - Verify results

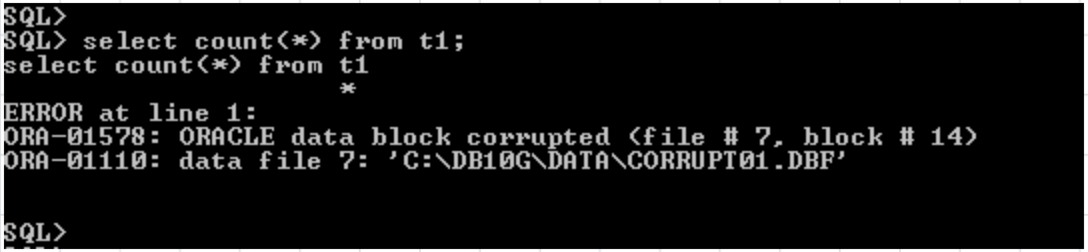
Block Recovery

- **Scenario:**
  - Create a test table (T) in tablespace “X” and insert dummy data
  - Take the tablespace (“X”) offline
  - Open the data file that belongs to tablespace “X” using UltraEdit
  - Make changes to the data, save and close the data file
  - Bring the tablespace online
  - Querying table T should report data corruption
  - For more details read my post Practicing Block Recovery

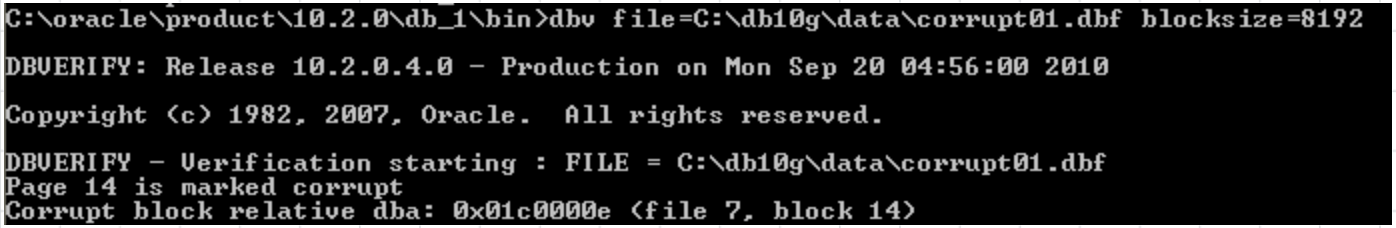
Block Recovery

Solution:

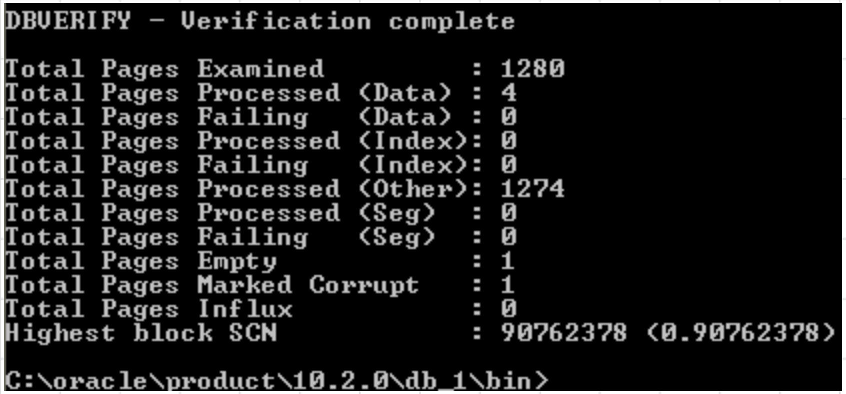
- Identify and list corrupt blocks



```
SQL>
SQL> select count(*) from t1;
select count(*) from t1
                     *
ERROR at line 1:
ORA-01578: ORACLE data block corrupted (file # 7, block # 14)
ORA-01110: data file 7: 'C:\DB10G\DATA\CORRUPT01.DBF'


SQL>
```



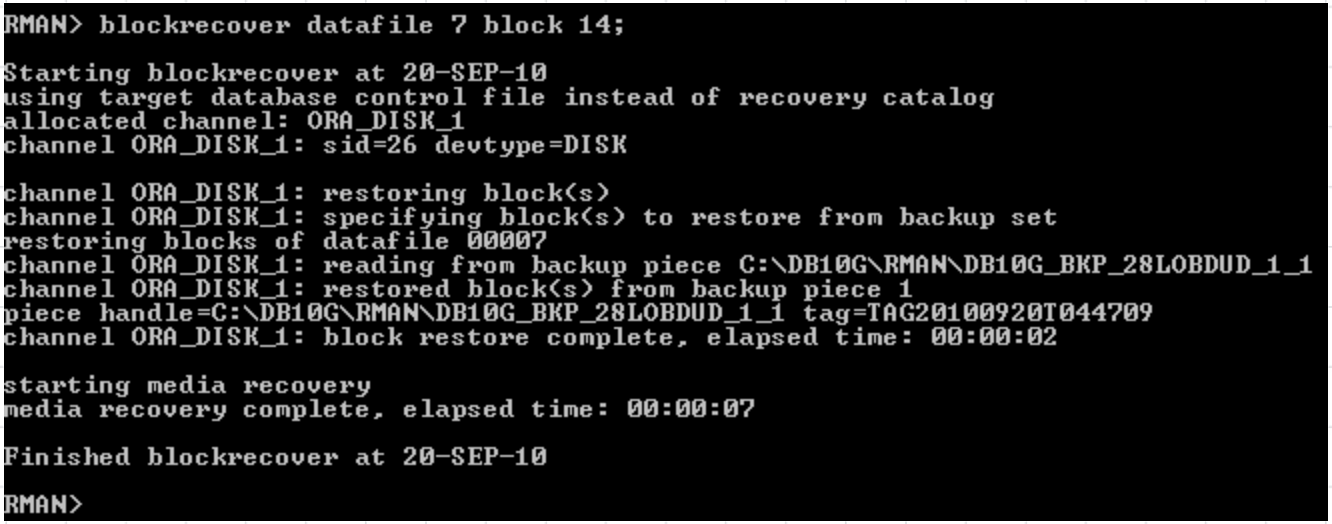
```
C:\oracle\product\10.2.0\db_1\bin>dbv file=C:\db10g\data\corrupt01.dbf blocksize=8192

DBVERIFY: Release 10.2.0.4.0 - Production on Mon Sep 20 04:56:00 2010

Copyright (c) 1982, 2007, Oracle.  All rights reserved.

DBVERIFY - Verification starting : FILE = C:\db10g\data\corrupt01.dbf
Page 14 is marked corrupt
Corrupt block relative dba: 0x01c0000e (file 7, block 14)
```



```
DBVERIFY - Verification complete

Total Pages Examined         : 1280
Total Pages Processed (Data) : 4
Total Pages Failing   (Data) : 0
Total Pages Processed (Index): 0
Total Pages Failing   (Index): 0
Total Pages Processed (Other): 1274
Total Pages Processed (Seg)  : 0
Total Pages Failing   (Seg)  : 0
Total Pages Empty            : 1
Total Pages Marked Corrupt   : 1
Total Pages Influx           : 0
Highest block SCN            : 90762378 (0.90762378)

C:\oracle\product\10.2.0\db_1\bin>
```

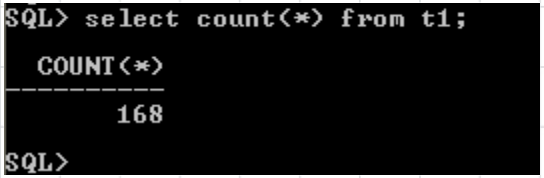
Block Recovery

– Perform Block Recovery



```
RMAN> blockrecover datafile 7 block 14;

Starting blockrecover at 20-SEP-10
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=26 devtype=DISK

channel ORA_DISK_1: restoring block(s)
channel ORA_DISK_1: specifying block(s) to restore from backup set
restoring blocks of datafile 00007
channel ORA_DISK_1: reading from backup piece C:\DB10G\RMAN\DB10G_BKP_28LOBDUD_1_1
channel ORA_DISK_1: restored block(s) from backup piece 1
piece handle=C:\DB10G\RMAN\DB10G_BKP_28LOBDUD_1_1 tag=TAG20100920T044709
channel ORA_DISK_1: block restore complete, elapsed time: 00:00:02

starting media recovery
media recovery complete, elapsed time: 00:00:07

Finished blockrecover at 20-SEP-10

RMAN>
```

– Verify results



```
SQL> select count(*) from t1;

  COUNT(*)
----------
       168

SQL>
```

Unsupported Recovery Solutions

Unsupported Recovery Solutions

- Recovering From RMAN Backup Pieces
- Recovering an Inconsistent Database

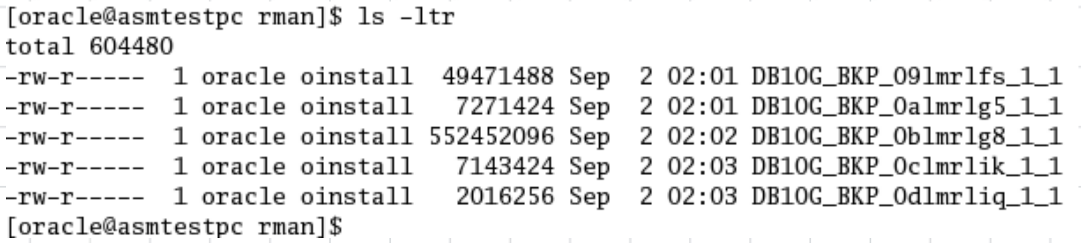
Recovering from RMAN Backup Pieces

- **Problem:**
  - All you have is RMAN backup pieces
  - SPFILE, Control File are included in the backup pieces
- **Solution:**
  - List all backup pieces
  - Startup database instance in NOMOUNT mode without parameter file
  - Extract SPFILE from the backup pieces
  - Extract Control File from the backup pieces
  - Shutdown and start database in MOUNT mode using restored SPFILE and Control Files
  - Restore and recover database
  - Open database with RESETLOGS option

Recovering from RMAN Backup Pieces

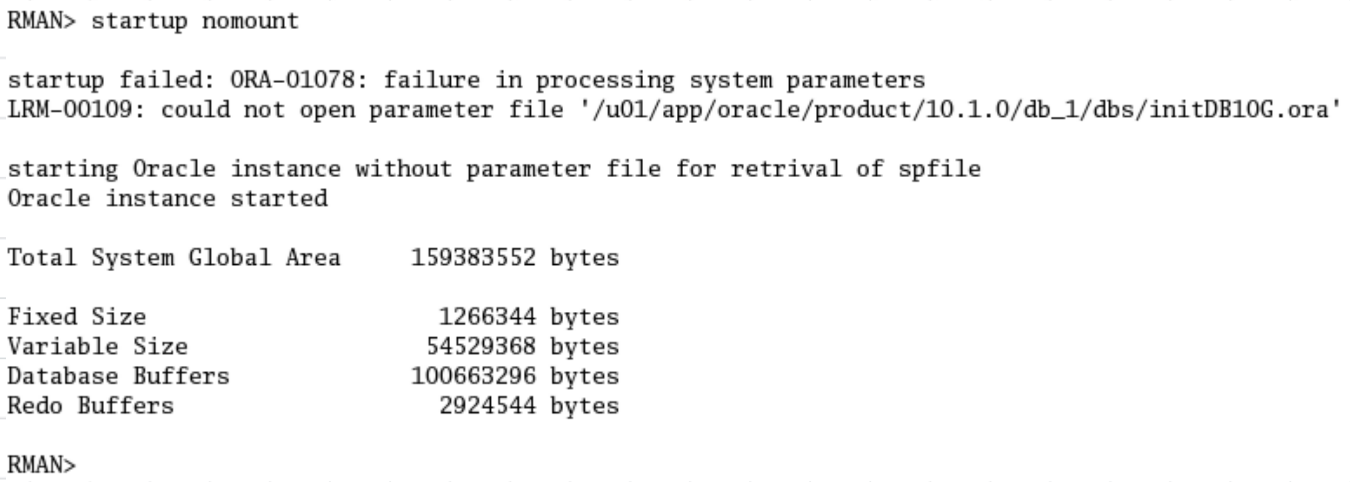
Solution:

- List all backup pieces



```
[oracle@asmtestpc rman]$ ls -ltr
total 604480
-rw-r-----  1 oracle oinstall  49471488 Sep  2 02:01 DB10G_BKP_09lmrlfs_1_1
-rw-r-----  1 oracle oinstall   7271424 Sep  2 02:01 DB10G_BKP_0almrlg5_1_1
-rw-r-----  1 oracle oinstall 552452096 Sep  2 02:02 DB10G_BKP_0blmrlg8_1_1
-rw-r-----  1 oracle oinstall   7143424 Sep  2 02:03 DB10G_BKP_0clmrlik_1_1
-rw-r-----  1 oracle oinstall   2016256 Sep  2 02:03 DB10G_BKP_0dlmrliq_1_1
[oracle@asmtestpc rman]$
```

- Startup database instance in NOMOUNT mode without parameter file

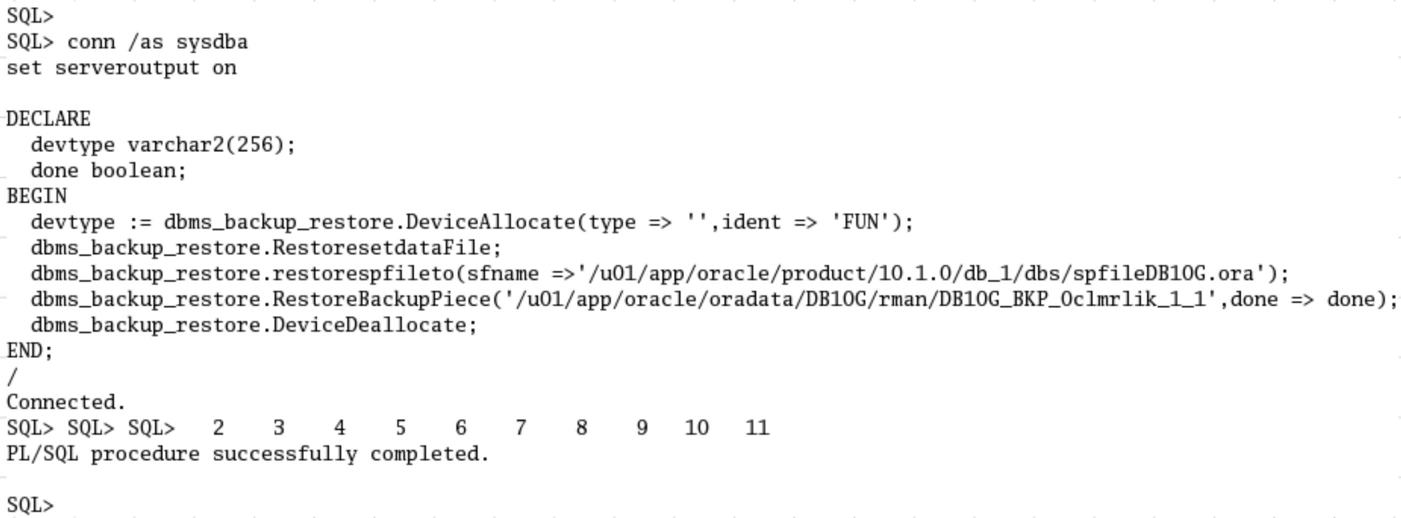


```
RMAN> startup nomount

startup failed: ORA-01078: failure in processing system parameters
LRM-00109: could not open parameter file '/u01/app/oracle/product/10.1.0/db_1/dbs/initDB10G.ora'

starting Oracle instance without parameter file for retrival of spfile
Oracle instance started

Total System Global Area     159383552 bytes

Fixed Size                     1266344 bytes
Variable Size                 54529368 bytes
Database Buffers             100663296 bytes
Redo Buffers                   2924544 bytes

RMAN>
```

Recovering from RMAN Backup Pieces

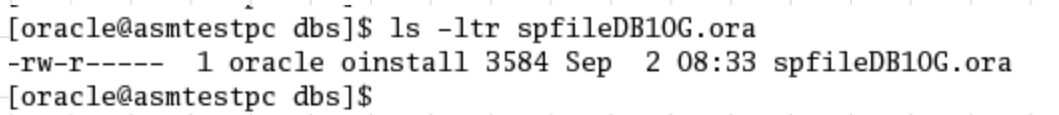
Solution:

- Extract SPFILE from the backup pieces using DBMS_BACKUP_RESTORE package



```
SQL>
SQL> conn /as sysdba
set serveroutput on

DECLARE
  devtype varchar2(256);
  done boolean;
BEGIN
  devtype := dbms_backup_restore.DeviceAllocate(type => '',ident => 'FUN');
  dbms_backup_restore.RestoresetdataFile;
  dbms_backup_restore.restorespfileto(sfname =>'/u01/app/oracle/product/10.1.0/db_1/dbs/spfileDB10G.ora');
  dbms_backup_restore.RestoreBackupPiece('/u01/app/oracle/oradata/DB10G/rman/DB10G_BKP_0clmrlik_1_1',done => done);
  dbms_backup_restore.DeviceDeallocate;
END;
/
Connected.
SQL> SQL> SQL>   2    3    4    5    6    7    8    9   10   11
PL/SQL procedure successfully completed.

SQL>
```

- Verify the extracted SPFILE

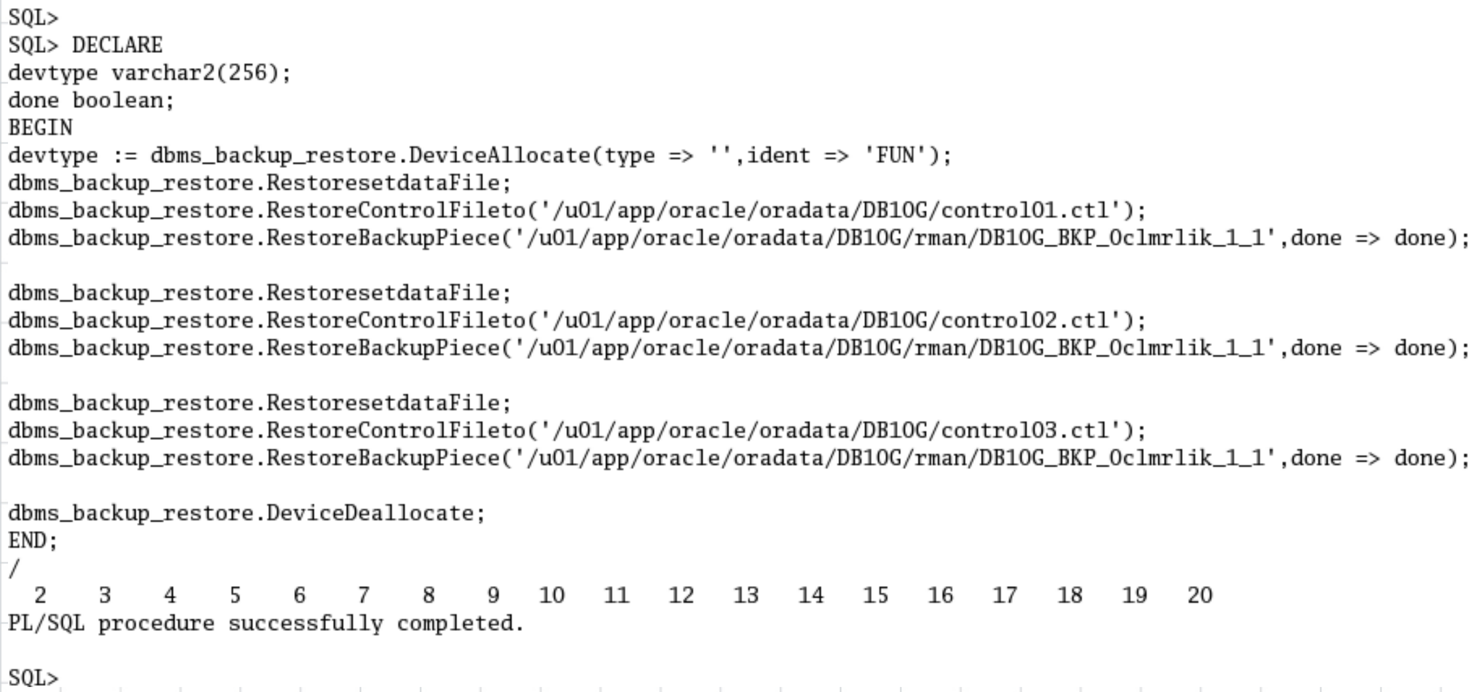


```
[oracle@asmtestpc dbs]$ ls -ltr spfileDB10G.ora
-rw-r-----  1 oracle oinstall 3584 Sep  2 08:33 spfileDB10G.ora
[oracle@asmtestpc dbs]$
```

Recovering from RMAN Backup Pieces

Solution:

- Extract Control Files from the backup pieces using DBMS_BACKUP_RESTORE package

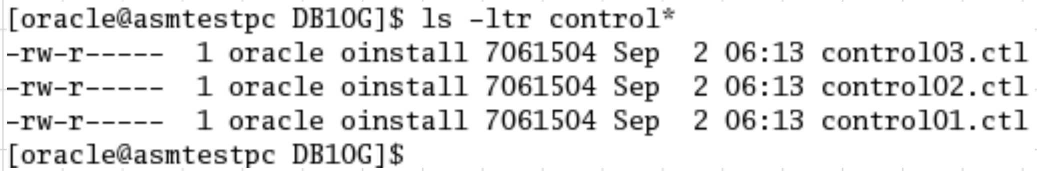


```
SQL>
SQL> DECLARE
devtype varchar2(256);
done boolean;
BEGIN
devtype := dbms_backup_restore.DeviceAllocate(type => '',ident => 'FUN');
dbms_backup_restore.RestoresetdataFile;
dbms_backup_restore.RestoreControlFileto('/u01/app/oracle/oradata/DB10G/control01.ctl');
dbms_backup_restore.RestoreBackupPiece('/u01/app/oracle/oradata/DB10G/rman/DB10G_BKP_0clmrlik_1_1',done => done);

dbms_backup_restore.RestoresetdataFile;
dbms_backup_restore.RestoreControlFileto('/u01/app/oracle/oradata/DB10G/control02.ctl');
dbms_backup_restore.RestoreBackupPiece('/u01/app/oracle/oradata/DB10G/rman/DB10G_BKP_0clmrlik_1_1',done => done);

dbms_backup_restore.RestoresetdataFile;
dbms_backup_restore.RestoreControlFileto('/u01/app/oracle/oradata/DB10G/control03.ctl');
dbms_backup_restore.RestoreBackupPiece('/u01/app/oracle/oradata/DB10G/rman/DB10G_BKP_0clmrlik_1_1',done => done);

dbms_backup_restore.DeviceDeallocate;
END;
/
  2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20
PL/SQL procedure successfully completed.

SQL>
```

Recovering from RMAN Backup Pieces

Solution:

- Verify extracted Control Files



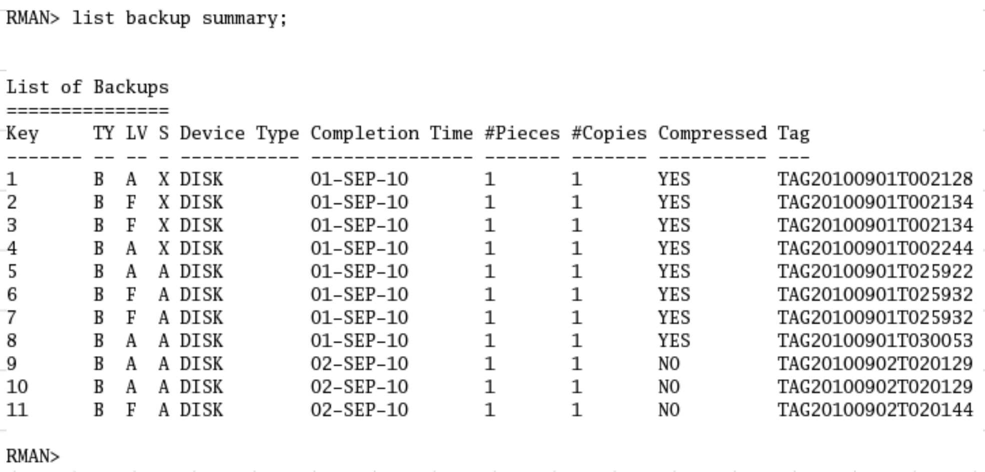
```
[oracle@asmtestpc DB10G]$ ls -ltr control*
-rw-r-----   1 oracle oinstall 7061504 Sep  2 06:13 control03.ctl
-rw-r-----   1 oracle oinstall 7061504 Sep  2 06:13 control02.ctl
-rw-r-----   1 oracle oinstall 7061504 Sep  2 06:13 control01.ctl
[oracle@asmtestpc DB10G]$
```

- Shutdown and start database in MOUNT mode using restored SPFILE and Control Files
- View backup information from the control file
- Restore and recover database
- Open database with RESETLOGS option

Recovering from RMAN Backup Pieces

Solution:

- Shutdown and start database in MOUNT mode using restored SPFILE and Control Files
- View backup information from the control file



```
RMAN> list backup summary;


List of Backups
===============
Key     TY LV S Device Type Completion Time #Pieces #Copies Compressed Tag
------- -- -- - ----------- --------------- ------- ------- ---------- ---
1       B  A  X DISK        01-SEP-10       1       1       YES        TAG20100901T002128
2       B  F  X DISK        01-SEP-10       1       1       YES        TAG20100901T002134
3       B  F  X DISK        01-SEP-10       1       1       YES        TAG20100901T002134
4       B  A  X DISK        01-SEP-10       1       1       YES        TAG20100901T002244
5       B  A  A DISK        01-SEP-10       1       1       YES        TAG20100901T025922
6       B  F  A DISK        01-SEP-10       1       1       YES        TAG20100901T025932
7       B  F  A DISK        01-SEP-10       1       1       YES        TAG20100901T025932
8       B  A  A DISK        01-SEP-10       1       1       YES        TAG20100901T030053
9       B  A  A DISK        02-SEP-10       1       1       NO         TAG20100902T020129
10      B  A  A DISK        02-SEP-10       1       1       NO         TAG20100902T020129
11      B  F  A DISK        02-SEP-10       1       1       NO         TAG20100902T020144

RMAN>
```

- Restore and recover database
- Open database with RESETLOGS option

Recovering an Inconsistent Database

- **Problem:**
  - Full database backup exists as of day “X”
  - A backup of SYSTEM, SYSAUX, and data tablespace (MOMEN_TS) exists as of day “Y”
  - Few Archive log’s are missing between day “X” and day “Y”
  - Your database is in inconsistent mode

Recovering an Inconsistent Database

- **Solution:**
  - Check SCN of datafiles
  - Edit init<sid>.ora to include hidden parameters and modify undo management
  - Start database instance in MOUNT mode
  - Bring all datafiles online
  - Perform a fake recovery
  - Open database with RESETLOGS
  - If the instance crashes then set “10015” event
  - Open database with RESETLOGS

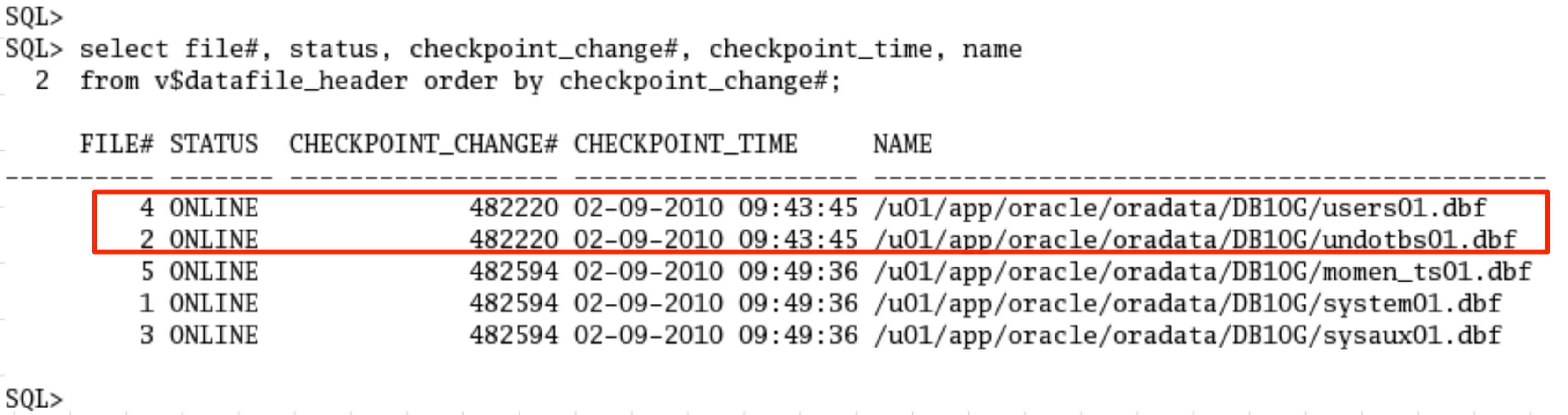
Recovering an Inconsistent Database

Simulating:

- **How to make an inconsistent database**
  - Take a full database backup
  - Make some changes (like create a table)
  - Delete at least one archivelog
  - Take backup of SYSTEM, SYSAUX, & MOMEN_TS tablespaces
  - Restore database

Recovering an Inconsistent Database

- **Solution:**
  - Check SCN of datafiles

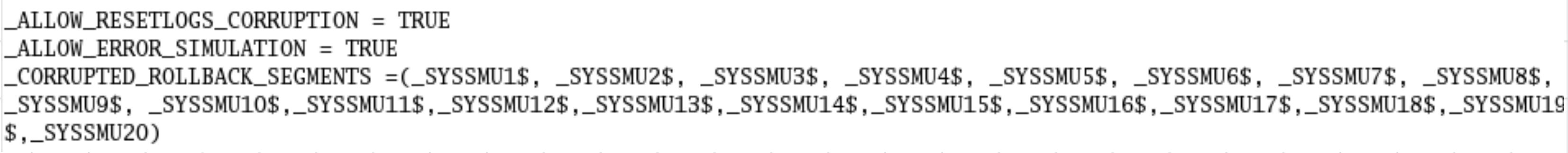


```
SQL>
SQL> select file#, status, checkpoint_change#, checkpoint_time, name
  2  from v$datafile_header order by checkpoint_change#;

     FILE# STATUS  CHECKPOINT_CHANGE# CHECKPOINT_TIME     NAME
---------- ------- ------------------ ------------------- ---------------------------------------------
         4 ONLINE              482220 02-09-2010 09:43:45 /u01/app/oracle/oradata/DB10G/users01.dbf
         2 ONLINE              482220 02-09-2010 09:43:45 /u01/app/oracle/oradata/DB10G/undotbs01.dbf
         5 ONLINE              482594 02-09-2010 09:49:36 /u01/app/oracle/oradata/DB10G/momen_ts01.dbf
         1 ONLINE              482594 02-09-2010 09:49:36 /u01/app/oracle/oradata/DB10G/system01.dbf
         3 ONLINE              482594 02-09-2010 09:49:36 /u01/app/oracle/oradata/DB10G/sysaux01.dbf

SQL>
```

Recovering an Inconsistent Database

Solution:

- Edit init<sid>.ora to include hidden parameters and modify undo management
    - _ALLOW_RESETLOGS_CORRUPTION=TRUE
    - _ALLOW_ERROR_SIMULATION=TRUE
    - _CORRUPTED_ROLLBACK_SEGMENTS=(comma separated list of Automatic Undo segments)
    - UNDO_MANAGEMENT=MANUAL



```
_ALLOW_RESETLOGS_CORRUPTION = TRUE
_ALLOW_ERROR_SIMULATION = TRUE
_CORRUPTED_ROLLBACK_SEGMENTS =(_SYSSMU1$, _SYSSMU2$, _SYSSMU3$, _SYSSMU4$, _SYSSMU5$, _SYSSMU6$, _SYSSMU7$, _SYSSMU8$,
_SYSSMU9$, _SYSSMU10$,_SYSSMU11$,_SYSSMU12$,_SYSSMU13$,_SYSSMU14$,_SYSSMU15$,_SYSSMU16$,_SYSSMU17$,_SYSSMU18$,_SYSSMU19
$,_SYSSMU20)
```

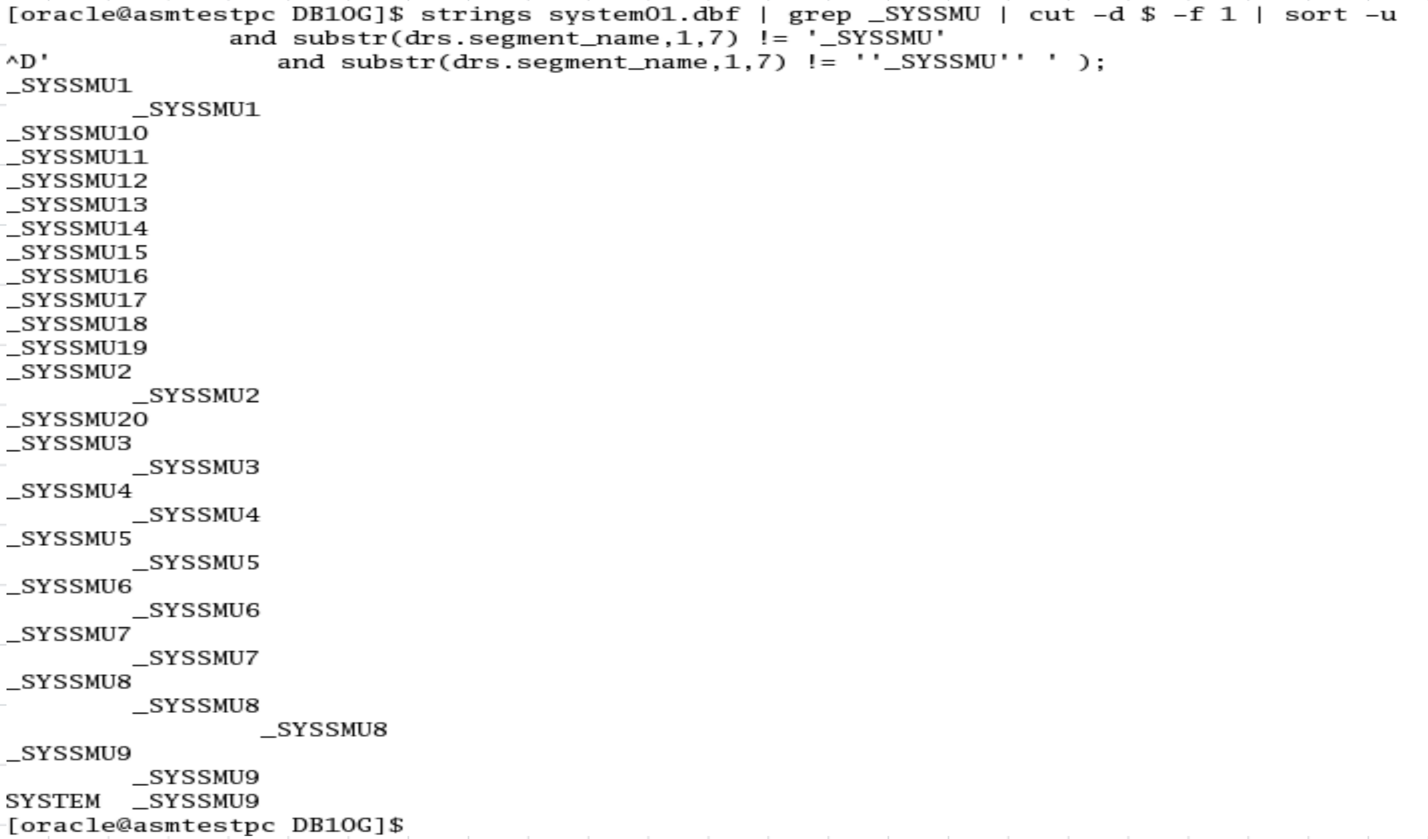
- **_ALLOW_RESETLOGS_CORRUPTION**
  - This parameter forces the opening of the datafiles even if their SCNs do not match up
  - Allow RESETLOGS even if it will cause corruption
- **_ALLOW_ERROR_SIMULATION**
  - Allow error simulation for testing
- **_CORRUPTED_ROLLBACK_SEGMENTS**
  - Corrupted undo segment list

Recovering an Inconsistent Database

Solution:

- Run the following command on UNIX to get undo segment names

  $ strings system01.dbf | grep _SYSSMU | cut -d $ -f 1 | sort –u

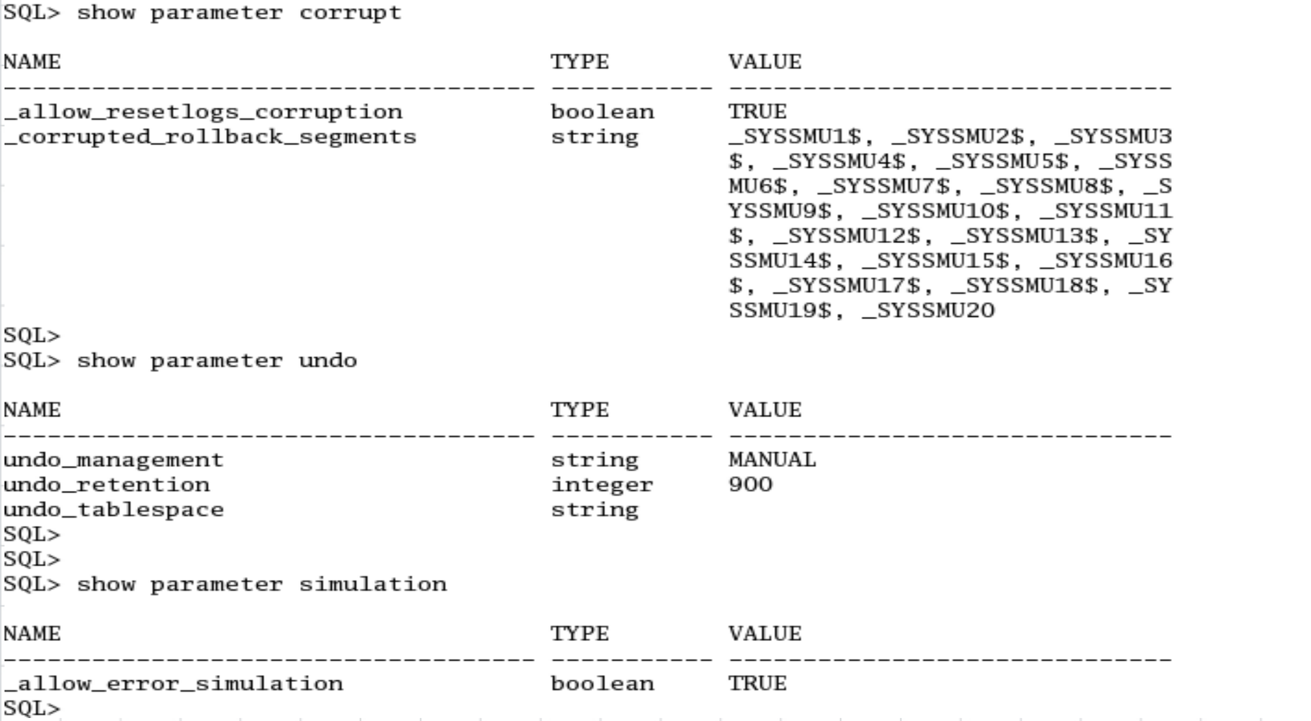


```
[oracle@asmtestpc DB10G]$ strings system01.dbf | grep _SYSSMU | cut -d $ -f 1 | sort -u
              and substr(drs.segment_name,1,7) != '_SYSSMU'
^D'              and substr(drs.segment_name,1,7) != ''_SYSSMU'' ' );
_SYSSMU1
        _SYSSMU1
_SYSSMU10
_SYSSMU11
_SYSSMU12
_SYSSMU13
_SYSSMU14
_SYSSMU15
_SYSSMU16
_SYSSMU17
_SYSSMU18
_SYSSMU19
_SYSSMU2
        _SYSSMU2
_SYSSMU20
_SYSSMU3
        _SYSSMU3
_SYSSMU4
        _SYSSMU4
_SYSSMU5
        _SYSSMU5
_SYSSMU6
        _SYSSMU6
_SYSSMU7
        _SYSSMU7
_SYSSMU8
        _SYSSMU8
                _SYSSMU8
_SYSSMU9
        _SYSSMU9
SYSTEM  _SYSSMU9
[oracle@asmtestpc DB10G]$
```

Recovering an Inconsistent Database

Solution:

- Start database instance in MOUNT mode
- Confirm the new parameters are set properly

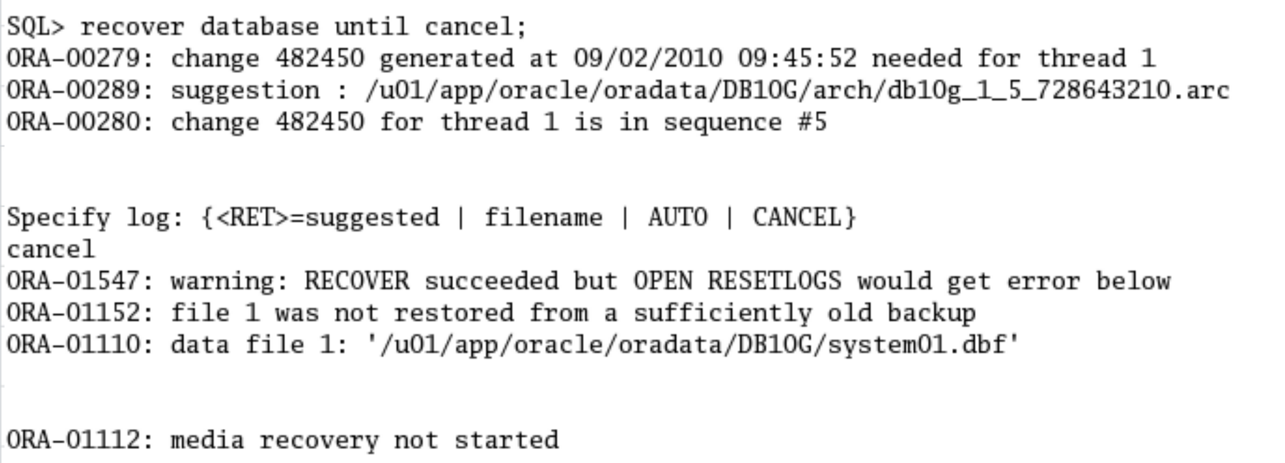


```
SQL> show parameter corrupt

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
_allow_resetlogs_corruption          boolean     TRUE
_corrupted_rollback_segments         string      _SYSSMU1$, _SYSSMU2$, _SYSSMU3
                                                 $, _SYSSMU4$, _SYSSMU5$, _SYSS
                                                 MU6$, _SYSSMU7$, _SYSSMU8$, _S
                                                 YSSMU9$, _SYSSMU10$, _SYSSMU11
                                                 $, _SYSSMU12$, _SYSSMU13$, _SY
                                                 SSMU14$, _SYSSMU15$, _SYSSMU16
                                                 $, _SYSSMU17$, _SYSSMU18$, _SY
                                                 SSMU19$, _SYSSMU20
SQL>
SQL> show parameter undo

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
undo_management                      string      MANUAL
undo_retention                       integer     900
undo_tablespace                      string
SQL>
SQL>
SQL> show parameter simulation

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
_allow_error_simulation              boolean     TRUE
SQL>
```

Recovering an Inconsistent Database

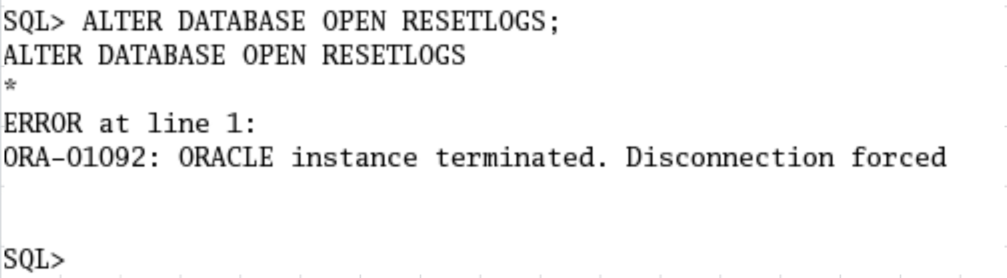
Solution:

- Bring all datafiles online
- Perform a fake recovery



```
SQL> recover database until cancel;
ORA-00279: change 482450 generated at 09/02/2010 09:45:52 needed for thread 1
ORA-00289: suggestion : /u01/app/oracle/oradata/DB10G/arch/db10g_1_5_728643210.arc
ORA-00280: change 482450 for thread 1 is in sequence #5


Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
cancel
ORA-01547: warning: RECOVER succeeded but OPEN RESETLOGS would get error below
ORA-01152: file 1 was not restored from a sufficiently old backup
ORA-01110: data file 1: '/u01/app/oracle/oradata/DB10G/system01.dbf'


ORA-01112: media recovery not started
```

- Open database with RESETLOGS

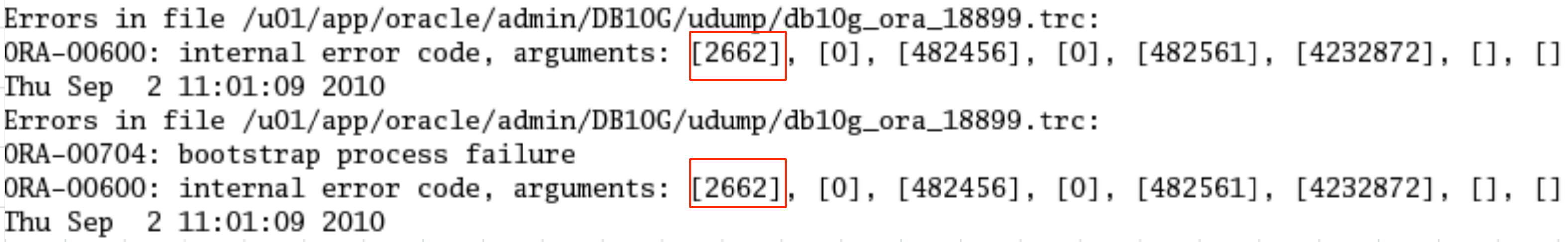


```
SQL> ALTER DATABASE OPEN RESETLOGS;
ALTER DATABASE OPEN RESETLOGS
*
ERROR at line 1:
ORA-01092: ORACLE instance terminated. Disconnection forced


SQL>
```

Recovering an Inconsistent Database

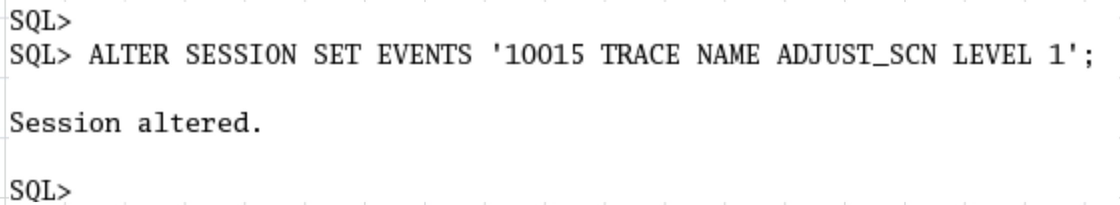
Solution:

- If the instance crashes then check the trace files and alert log for ORA-00600 [2662] error in it.



```
Errors in file /u01/app/oracle/admin/DB10G/udump/db10g_ora_18899.trc:
ORA-00600: internal error code, arguments: [2662], [0], [482456], [0], [482561], [4232872], [], []
Thu Sep  2 11:01:09 2010
Errors in file /u01/app/oracle/admin/DB10G/udump/db10g_ora_18899.trc:
ORA-00704: bootstrap process failure
ORA-00600: internal error code, arguments: [2662], [0], [482456], [0], [482561], [4232872], [], []
Thu Sep  2 11:01:09 2010
```

- MOUNT the database and set “10015” event

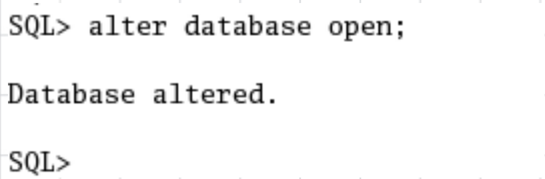


```
SQL>
SQL> ALTER SESSION SET EVENTS '10015 TRACE NAME ADJUST_SCN LEVEL 1';

Session altered.

SQL>
```

Recovering an Inconsistent Database

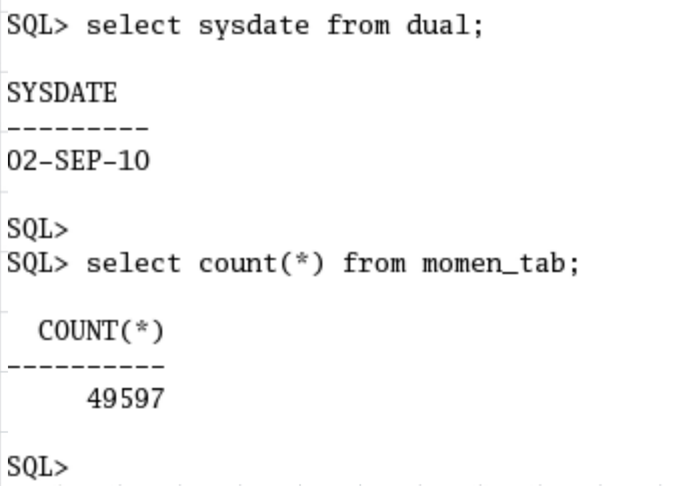
Solution:

- Open database



```
SQL> alter database open;

Database altered.

SQL>
```

- Confirm that the database is functional



```
SQL> select sysdate from dual;

SYSDATE
---------
02-SEP-10

SQL>
SQL> select count(*) from momen_tab;

  COUNT(*)
----------
     49597

SQL>
```

Recovering an Inconsistent Database

- **If the instance crashes again, check the trace file for another ORA-00600 [2662] error.**
- **If so, increment the LEVEL by 1 and repeat (Increment 10015 Event & Open database) until you can successfully open the database.**
- **If you need to go beyond 'LEVEL 6' or 'LEVEL 7', then the database is probably beyond forcing open ☹**

Data Unloader

Data Unloader (DUL)

- DUL is the process of extracting data from Oracle data files directly.
- DUL completely bypassing the Oracle Kernel.
- Contact Oracle Support
- For more information on third party Data Unloader products you may read DUL & Desperation: The Trials and Tribulations of Corruption by Jonah H. Harris

References

- Oracle Documentation (Release 10g R2 and 11g R1 & R2)
- Oracle database recovery with data unloading by Ignacio Ruiz
- Disaster Recovery Stories by Alejandro Vargas
- Dell Simplifies Backup And Recovery For SMBs by Symantec
- UK SMEs in data recovery failure by Acronis
- DUL & Desperation: The Trials and Tribulations of Corruption by Jonah H. Harris
- When you lost your controlfile backups by Coskan Gundogar
- Practicing Block Recovery by Asif Momen
- RMAN Recipes for Oracle Database 11g by Darl Kuhn, Sam Alapati, and Arup Nanda
- Oracle Support

Thank you for your interest

For more information and to provide feedback please contact me

My e-mail address is:

asif.momen@gmail.com

My blog address is:

http://momendba.blogspot.com