Practice Admin I

# Cài đặt, cấu hình DB

* Cài đặt Oracle trên linux red hat
* Tạo database bằng dbca
* Tình huống lỗi các file trong các giai đoạn startup và shutdown.

Testdb. Yêu cầu: file nào ảnh hưởng, hãy shutdo wn db, backup file đó và xóa file đi xem sẽ gặp lỗi gì, sau đó đưa ra phương án khắc phục.

**startup**

* + Nomount: file nào ảnh hưởng
  + Mount: file nào
  + Open: file nào.

**Shutdown:**

* + Normal: 🡪 ảnh hưởng gì
  + Immidiate: ảnh hưởng gì
  + Transaction: ảnh hưởng gì
  + Abort: ảnh hưởng gì?

Kiểm tra trạng thái DB đang ở chế độ nào?

# Lab về ASM

* Vào giao diện ASMCMD, theo dõi dung lượng ASM
* Tạo directory trên Asm
* Xem group disk, thống kê trên RDMBS instance và trên ASM instance

# Lab về cấu hình network

* Listener.ora: Tự động đăng ký DB với listener. Nhưng 1 số trường hợp ko đăng ký được 🡪 đăng ký static.
* $ORACLE\_HOME/network/admin/linstener.ora

LISTENER =

(ADDRESS\_LIST=

(ADDRESS=(PROTOCOL=tcp)(HOST=server1)(PORT=1571))

(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1571)))

# this is new

SID\_LIST\_LISTENER=

(SID\_LIST=

(SID\_DESC=

(GLOBAL\_DBNAME=oratest)

(SID\_NAME=oratest)

(ORACLE\_HOME=/usr/oracle/app/product/11.2.0/dbhome\_1)

)

)

* Start, stop, reload, status listener.
* Tnsnames.ora: Đặt ở client.
* $ORACLE\_HOME/network/admin/tnsnames.ora

TESTDB =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = db01.performancetest)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = testdb)

)

)

TESTDB1=

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = db01.performancetest)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = testdb)

)

)

* Sqlnet.ora: Đặt server. File cấu hình cách kết nối, thời gian time out…
* Giao diện: netca, net manager
* Cài đặt oracle client.

# Lab về user, information in DB

* Theo dõi tablespace
* $export ORACLE\_UNQNAME=oratest

Select \* from dba\_tablespace\_usage\_metrics;

* Tạo user, profile.

CREATE TABLESPACE Duong DATAFILE

'/opt/oracle/oradata/testdb/chien01.dbf' SIZE 35584K AUTOEXTEND ON NEXT 1280K MAXSIZE 8192M,

'/opt/oracle/oradata/testdb/chien02.dbf' SIZE 1M AUTOEXTEND ON NEXT 1280K MAXSIZE 8192M

LOGGING

ONLINE

PERMANENT

EXTENT MANAGEMENT LOCAL AUTOALLOCATE

BLOCKSIZE 8K

SEGMENT SPACE MANAGEMENT AUTO

* FLASHBACK ON;

Create user duong identified by 123456 default tablespace duong;

Gán quyền:

* User application thì gán cho quyền connect, resource, nếu có job thì gán scheduler\_admin.
* Chỉ gán dba cho admin.
* Xem một số tham số:

Grant <role> to <user>

* + O7\_DICTIONARY\_ACCESSIBILITY boolean FALSE
  + remote\_os\_authent boolean FALSE
  + resource\_limit boolean FALSE

Create profile:

CREATE PROFILE neverexpire LIMIT

SESSIONS\_PER\_USER DEFAULT

CPU\_PER\_SESSION DEFAULT

CPU\_PER\_CALL DEFAULT

CONNECT\_TIME DEFAULT

IDLE\_TIME DEFAULT

LOGICAL\_READS\_PER\_SESSION DEFAULT

LOGICAL\_READS\_PER\_CALL DEFAULT

COMPOSITE\_LIMIT DEFAULT

PRIVATE\_SGA DEFAULT

FAILED\_LOGIN\_ATTEMPTS DEFAULT

PASSWORD\_LIFE\_TIME UNLIMITED

PASSWORD\_REUSE\_TIME DEFAULT

PASSWORD\_REUSE\_MAX DEFAULT

PASSWORD\_LOCK\_TIME DEFAULT

PASSWORD\_GRACE\_TIME DEFAULT;

Alter user chien profile neverexpire;

Alter user test profile neverexpired;

* Lock conflict: Tìm lock, tìm session, resolve lock.

Lệnh xem lock:

**Session 1: hr/hr**

Update, insert, delete jobs.

Ko commit.

**Session 2: sys / as sysdba**

**Lệnh xem wait:**

select SID, SERIAL#, USERNAME

from V$SESSION where SID in

(select BLOCKING\_SESSION from V$SESSION);

Lệnh kill session lock:

ALTER SYSTEM KILL SESSION ‘sid, serial#’

Hoặc xem lock:

SELECT l.session\_id||','||v.serial# sid\_serial, v.osuser,v.MACHINE,v.PROGRAM,v.LOGON\_TIME,

l.ORACLE\_USERNAME ora\_user,

o.object\_name,

o.object\_type,

DECODE(l.locked\_mode,

0, 'None',

1, 'Null',

2, 'Row-S (SS)',

3, 'Row-X (SX)',

4, 'Share',

5, 'S/Row-X (SSX)',

6, 'Exclusive',

TO\_CHAR(l.locked\_mode)

) lock\_mode,

o.status,

to\_char(o.last\_ddl\_time,'dd.mm.yy') last\_ddl

FROM dba\_objects o,v$locked\_object l, v$session v

WHERE o.object\_id = l.object\_id

and l.SESSION\_ID=v.sid

order by 2,3;

Trong đó sid và serial# được select từ lệnh trên, hoặc từ v$session với xác nhận là gây lock. Kiểm tra kỹ trước khi gõ lệnh kill.

* Undo data

Lab recovery undo tablespace:

Quy trình thực hiện recovery undo ts là recreate undo tablespace.

(trả lời câu hỏi vì sao có thể recreate undo tablespace mà không ảnh hưởng đến an toàn của database? Trường hợp nào ảnh hưởng, trường hợp nào không ảnh hưởng?)

Quy trình thực hiện lab:

$ sqlplus / as sysdba

SQL> show parameter undo

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

undo\_management string AUTO

undo\_retention integer 900

undo\_tablespace string UNDOTBS1

SQL>

The current undo tablespace as suggested by the initialization parameter undo\_tablespace is UNDOTBS1. Leave this sysdba as is, open another console, log in as user SCOTT and initiate a transaction.

$ sqlplus scott/tiger

update emp set sal = sal + 1000 where empno=7839;

1 row updated.

With an update on emp table we have initiated a transaction. The undo data is written to a segment in the UNDOTBS1 tablespace. Now leave this SCOTT's session intact and go back to the sysdba console without issuing any COMMIT or ROLLBACK.

CREATE UNDO TABLESPACE undotbs2

DATAFILE '/d01/apps/oradata/oraxpo/undotbs201.dbf'

SIZE 50M AUTOEXTEND ON NEXT 5M;

Tablespace created.

-- We created a new UNDO tablespace named UNDOTBS2

ALTER SYSTEM SET UNDO\_TABLESPACE=UNDOTBS2 SCOPE=BOTH;

System altered.

-- Switch the database to the new UNDO tablespace.

SQL> DROP TABLESPACE undotbs1 INCLUDING CONTENTS AND DATAFILES;

DROP TABLESPACE undotbs1 INCLUDING CONTENTS AND DATAFILES

\*

ERROR at line 1:

ORA-30013: undo tablespace 'UNDOTBS1' is currently in use

-- Try to drop the tablespace but failed.

SQL>

With the alter system set undo\_tablespace=UNDOTBS2, the database UNDO tablespace is changed and any new transaction's undo data will go to the new tablespace i.e. UNDOTBS2. But the undo data for already pending transaction (e.g. the one initiated by SCOTT before the database UNDO tablespace switch) is still in the old tablespace with a status of PENDING OFFLINE. As far as it is there you cannot drop the old tablespace.

set lines 10000

column name format a10

SELECT a.name,b.status

FROM v$rollname a,v$rollstat b

WHERE a.usn = b.usn

AND a.name IN (

SELECT segment\_name

FROM dba\_segments

WHERE tablespace\_name = 'UNDOTBS1'

);

NAME STATUS

---------- ---------------

\_SYSSMU8$ PENDING OFFLINE

The above query shows the name of the UNDO segment in the UNDOTBS1 tablespace and its status. Now lets see which users/sessions are running this pending transaction.

column username format a6

SELECT a.name,b.status , d.username , d.sid , d.serial#

FROM v$rollname a,v$rollstat b, v$transaction c , v$session d

WHERE a.usn = b.usn

AND a.usn = c.xidusn

AND c.ses\_addr = d.saddr

AND a.name IN (

SELECT segment\_name

FROM dba\_segments

WHERE tablespace\_name = 'UNDOTBS1'

);

NAME STATUS USERNA SID SERIAL#

---------- --------------- ------ ---------- ----------

\_SYSSMU8$ PENDING OFFLINE SCOTT 147 4

So this is SCOTT with SID=147 and SERIAL#=4. Since we know now the user, we can go to him/her and request to end the transaction gracefully i.e. issue a ROLLBACK or COMMIT. However, if this is not possible (say the user initiated the transaction and left for annual leave :) and trust me this happens) you may go ahead and kill the session to release the undo segments in the UNDOTBS1 tablespace.

SQL> alter system kill session '147,4';

System altered.

SELECT a.name,b.status , d.username , d.sid , d.serial#

FROM v$rollname a,v$rollstat b, v$transaction c , v$session d

WHERE a.usn = b.usn

AND a.usn = c.xidusn

AND c.ses\_addr = d.saddr

AND a.name IN (

SELECT segment\_name

FROM dba\_segments

WHERE tablespace\_name = 'UNDOTBS1'

);

no rows selected

As we can see once the session is kills we don't see anymore segments occupied in the UNDOTBS1 tablespace. Lets drop UNDOTBS1.

SQL> DROP TABLESPACE undotbs1 INCLUDING CONTENTS AND DATAFILES;

DROP TABLESPACE undotbs1 INCLUDING CONTENTS AND DATAFILES

\*

ERROR at line 1:

ORA-30013: undo tablespace 'UNDOTBS1' is currently in use

If you are retaining undo data then you still won't be able to drop the tablespace because it is still in use by undo\_retention. Let the UNDO\_RETENTION time pass and then try to drop the tablespace. In my case it is 900 seconds i.e. 15 minutes.

-- After 15 minutes.

SQL> DROP TABLESPACE undotbs1 INCLUDING CONTENTS AND DATAFILES;

Tablespace dropped.

# Lab về monitoring, audit, maintain, performance

* 1. Đặt audit

Có 2 option đặt audit là by access và by session. Giải thích hai cách này? và thực hiện đặt audit như sau:

Đặt audit cho user fireid:

Kiểm tra chế độ audit hiện tại của DB:

Show parameter audit\_trail

Audit toàn bộ user:

AUDIT ALL BY fireid BY ACCESS;

Audit các thao tác select, update, insert, delete:

AUDIT SELECT TABLE, UPDATE TABLE, INSERT TABLE, DELETE TABLE BY fireid BY ACCESS;

Audit procedure execute:

AUDIT EXECUTE PROCEDURE BY fireid BY ACCESS;

Audit trên 1 bảng:

AUDIT INSERT, UPDATE, DELETE ON table\_name BY ACCESS;

View Audit Trail

The audit trail is stored in the SYS.AUD$ table. It's contents can be viewed directly or via the following views.

* DBA\_AUDIT\_EXISTS
* DBA\_AUDIT\_OBJECT
* DBA\_AUDIT\_SESSION
* DBA\_AUDIT\_STATEMENT
* DBA\_AUDIT\_TRAIL
* DBA\_OBJ\_AUDIT\_OPTS
* DBA\_PRIV\_AUDIT\_OPTS
* DBA\_STMT\_AUDIT\_OPTS

The audit trail contains a lot of data, but the following are most likely to be of interest.

* USERNAME : Oracle Username.
* TERMINAL : Machine that the user performed the action from.
* TIMESTAMP : When the action occured.
* OBJECT\_OWNER : The owner of the object that was interacted with.
* OBJECT\_NAME : The name of the object that was interacted with.
* ACTION\_NAME : The action that occured against the object. (INSERT, UPDATE, DELETE, SELECT, EXECUTE)

Select \* from SYS.AUD$;

SELECT USERNAME, LOGOFF\_TIME, LOGOFF\_LREAD, LOGOFF\_PREAD,

LOGOFF\_LWRITE, LOGOFF\_DLOCK

FROM DBA\_AUDIT\_SESSION;

SELECT \* FROM DBA\_OBJ\_AUDIT\_OPTS

WHERE OWNER = 'LAUREL' AND OBJECT\_NAME LIKE 'EMP%';

select 'standard audit', sessionid,

    proxy\_sessionid, statementid, entryid, extended\_timestamp, global\_uid,

    username, client\_id, null, os\_username, userhost, os\_process, terminal,

    instance\_number, owner, obj\_name, null, new\_owner,

    new\_name, action, action\_name, audit\_option, transactionid, returncode,

    scn, comment\_text, sql\_bind, sql\_text,

    obj\_privilege, sys\_privilege, admin\_option, grantee, priv\_used,

    ses\_actions, logoff\_time, logoff\_lread, logoff\_pread, logoff\_lwrite,

    logoff\_dlock, session\_cpu

  from

  dba\_audit\_trail;

# Theo dõi hệ thống (Maintain and performance)

Theo dõi các process, số process, số session, resource hiện tại của hệ thống:

Set line 200

Select \* from v$resource\_limit;

Sử dụng khi nào? Theo dõi hiện trạng của hệ thống so với mức cấu hình lớn nhất, so với mức đã đạt max. Nếu thường xuyên theo dõi: 🡪 Tải của hệ thống thường xuyên đạt ngưỡng nào 🡪 Giúp DBA có dữ liệu để tuning, reporting. 🡪 Tìm nguyên nhân và khắc phục lỗi.

Show parameter session

Show parameter processes

Alter system set processes =1000 scope=spfile;

Các session đang kết nối đến DB, lấy tên máy, program,…:

Select sid, serial#, machine, program, state from v$session;

Dung lượng DB and tempfile:

select sum(bytes/1024/1024/1024) from v$datafile; --GB

select sum(bytes/1024/1024/1024) from v$tempfile; --GB

Long transaction:

col OPNAME for a30

select OPNAME,SOFAR/TOTALWORK\*100 PCT, trunc(TIME\_REMAINING/60) MIN\_RESTANTES,

trunc(ELAPSED\_SECONDS/60) MIN\_ATEAGORA

from v$session\_longops where TOTALWORK>0 order by pct asc;

Dung lượng từng Tablespace :

select ts.name, sum(df.bytes/1024/1024) from v$tablespace ts, v$datafile df where ts.TS#=df.TS# group by ts.name;

Sử dụng EM để Đặt chính sách cho AWR và lấy file AWR. Cách xem file AWR.

Nhắc lại khái niệm Maintenance Window opens.

Xem tham số về statistic:

Show parameter STATISTICS\_LEVEL

Sử dụng EM để theo dõi performance.

# Kiểm tra tham số và đặt chế độ archive log mode

* 1. Verifying control files:

Show parameter controlfile

* 1. Configuring a default fast recovery area
  2. 2 tham số cho FRA.

Ý nghĩa của mỗi tham số:

show parameter DB\_RECOVERY\_FILE \_DEST🡪

Show parameter DB\_RECOVERY\_FILE\_DEST\_SIZE 🡪

alter system set db\_recovery\_file\_dest\_size =5G;

* 1. Multiplexing redo log groups

Đặt multi redo log file:

# Resizing / Recreating Online Redo Log Files

by Jeff Hunter, Sr. Database Administrator

One of the best ways I have found to resize or recreate online redo log files and keep the current sequence is to perform it online. In this example, we will resize all online redo logs from 100MB to 250MB while the database is running and use SQL\*Plus to drop/recreate them in stages.

Before looking at the tasks involved to perform the resize, let's look at the current online redo log groups and their sizes:

|  |
| --- |
| SQL> **SELECT a.group#, a.member, b.bytes/1024/1024** **FROM v$logfile a,**  **v$log b WHERE a.group# = b.group#;**  GROUP# MEMBER BYTES  ---------- ---------------------------------------- ------------  1 /u03/app/oradata/ORA920/redo\_g01a.log 104,857,600  1 /u04/app/oradata/ORA920/redo\_g01b.log 104,857,600  1 /u05/app/oradata/ORA920/redo\_g01c.log 104,857,600  2 /u03/app/oradata/ORA920/redo\_g02a.log 104,857,600  2 /u04/app/oradata/ORA920/redo\_g02b.log 104,857,600  2 /u05/app/oradata/ORA920/redo\_g02c.log 104,857,600  3 /u03/app/oradata/ORA920/redo\_g03a.log 104,857,600  3 /u04/app/oradata/ORA920/redo\_g03b.log 104,857,600  3 /u05/app/oradata/ORA920/redo\_g03c.log 104,857,600  9 rows selected. |

Now let's take a look at the steps involved to resize / recreate all online redo log groups:

1. **Make the last redo log CURRENT**

Force a log switch until the last redo log is marked "CURRENT" by issuing the following command:

|  |
| --- |
| SQL> **select group#, status from v$log;**  GROUP# STATUS  ---------- ----------------  **1 CURRENT**  2 INACTIVE  3 INACTIVE  SQL> **alter system switch logfile;**  SQL> **alter system switch logfile;**  SQL> **select group#, status from v$log;**  GROUP# STATUS  ---------- ----------------  1 INACTIVE  2 INACTIVE  **3 CURRENT** |

1. **Drop first redo log**

After making the last online redo log file the CURRENT one, drop the first online redo log:

|  |
| --- |
| SQL> **alter database drop logfile group 1;**  Database altered. |

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | http://www.idevelopment.info/images/popup_dialog_information_mark.gif | As a DBA, you should already be aware that if you are going to drop a logfile group, it cannot be the*current* logfile group. I have run into instances; however, where attempting to drop the logfile group resulted in the following error as a result of the logfile group having an *active* status:  SQL> **ALTER DATABASE DROP LOGFILE GROUP 1;**  ALTER DATABASE DROP LOGFILE GROUP 1  \*  ERROR at line 1:  ORA-01624: log 1 needed for crash recovery of instance ORA920 (thread 1)  ORA-00312: online log 1 thread 1: '*<file\_name>*'  Easy problem to resolve. Simply perform a checkpoint on the database:  SQL> **ALTER SYSTEM CHECKPOINT GLOBAL;**  System altered.  SQL> **ALTER DATABASE DROP LOGFILE GROUP 1;**  Database altered. | |

1. **Re-create dropped online redo log group**

Re-create the dropped redo log group with different size (if desired):

|  |
| --- |
| SQL> **alter database add logfile group 1 (**  2 **'/u03/app/oradata/ORA920/redo\_g01a.log',**  3 **'/u04/app/oradata/ORA920/redo\_g01b.log',**  4 **'/u05/app/oradata/ORA920/redo\_g01c.log') size 250m reuse;**  Database altered. |

1. **Force another log switch**

After re-creating the online redo log group, force a log switch. The online redo log group just created should become the "CURRENT" one:

|  |
| --- |
| SQL> **select group#, status from v$log;**  GROUP# STATUS  ---------- ----------------  1 UNUSED  2 INACTIVE  **3 CURRENT**  SQL> **alter system switch logfile;**  SQL> **select group#, status from v$log;**  GROUP# STATUS  ---------- ----------------  **1 CURRENT**  2 INACTIVE  3 ACTIVE |

1. **Loop back to Step 2 until all logs are rebuilt**

After re-creating an online redo log group, continue to re-create (or resize) all online redo log groups until all of them are rebuilt.

After rebuilding (resizing) all online redo log groups, here is a snapshot of all physical files:

|  |
| --- |
| SQL> **SELECT a.group#, a.member, b.bytes**  **FROM v$logfile a, v$log b WHERE a.group# = b.group#;**  GROUP# MEMBER BYTES  ---------- ---------------------------------------- ------------  1 /u03/app/oradata/ORA920/redo\_g01a.log 262,144,000  1 /u04/app/oradata/ORA920/redo\_g01b.log 262,144,000  1 /u05/app/oradata/ORA920/redo\_g01c.log 262,144,000  2 /u03/app/oradata/ORA920/redo\_g02a.log 262,144,000  2 /u04/app/oradata/ORA920/redo\_g02b.log 262,144,000  2 /u05/app/oradata/ORA920/redo\_g02c.log 262,144,000  3 /u03/app/oradata/ORA920/redo\_g03a.log 262,144,000  3 /u04/app/oradata/ORA920/redo\_g03b.log 262,144,000  3 /u05/app/oradata/ORA920/redo\_g03c.log 262,144,000  9 rows selected. |

* 1. Placing your database in ARCHIVELOG mode and Ensuring that redundant archive logs are created

Quy trình thực hiện:

export ORACLE\_SID=orcl

1. set parameter log\_archive\_dest

ALTER SYSTEM SET log\_archive\_dest\_1='location=/usr/oracle/archivelog/';

ALTER SYSTEM SET log\_archive\_dest\_state\_1 ='ENABLE';

ALTER SYSTEM SET log\_archive\_format='arch\_%T\_%t\_%s\_%r.dbf' scope =spfile;

Kiểm tra xem còn tham số nào cần thực hiện đồng thời cần phải tắt DB không?

2-- shutdown immediate all database instances

sqlplus / as sysdba

SQL> shutdown immediate;

3-- startup database in mount mode

SQL> startup mount;

4-- enable archvive logging

sql> alter database archivelog;

sql> exit;

5-- stop database

sqlplus / as sysdba

SQL> shutdown immediate;

6-- restart database instances

sQL> startup

7-- verify archiving is enabled by means of:

sql> archive log list;

8-- backup database

# Lab về backup và recovery

* Sử dụng RMAN:
  + Backup trong khi db đang hoạt động. Backup online.
  + Đặt lịch backup bằng scheduler trên EM và trên linux.
  + Viết script backup full, backup level 0, level 1.
  + Phân biệt các loại backup và viết script backup cho từng loại.
  + Đặt database từ chế độ archive log mode sang chế độ no archive log và ngược lại.
  + Theo dõi file, tình huống nào thì switch log.

run {

BACKUP BLOCKS ALL AS COMPRESSED BACKUPSET INCREMENTAL LEVEL 0 DATABASE FORMAT '/usr/oracle/backup/dbbk/db0\_%T\_%d\_%u\_%s' FILESPERSET 4 TAG FULLBKP;

}

run{

BACKUP AS COMPRESSED BACKUPSET ARCHIVELOG ALL FORMAT '/usr/oracle/backup/arcbk/arc0\_%T\_%d\_%u\_%s\_%U' FILESPERSET 4 DELETE INPUT TAG ARCH;

}

run{

BACKUP AS COMPRESSED BACKUPSET CURRENT CONTROLFILE FORMAT '/opt/oracle/backup/ctrbk/bkcontrolfile0\_%T\_%d\_%u\_%s' TAG CTLFILE;

}

RMAN> Restore controlfile from ‘/opt/oracle/backup/orcl/bkcontrolfile0\_3u203ụ23u04’;

Level 1:

BACKUP INCREMENTAL LEVEL 1 CUMULATIVE DATABASE; # blocks changed since level 0

Default là differential.

Backup lv 1 cumulative

run {

BACKUP BLOCKS ALL AS COMPRESSED BACKUPSET INCREMENTAL LEVEL 1 CUMULATIVE DATABASE FORMAT '/usr/oracle/backup/dbbk/db0\_%T\_%d\_%u\_%s' FILESPERSET 4 TAG FULLBKP;

}

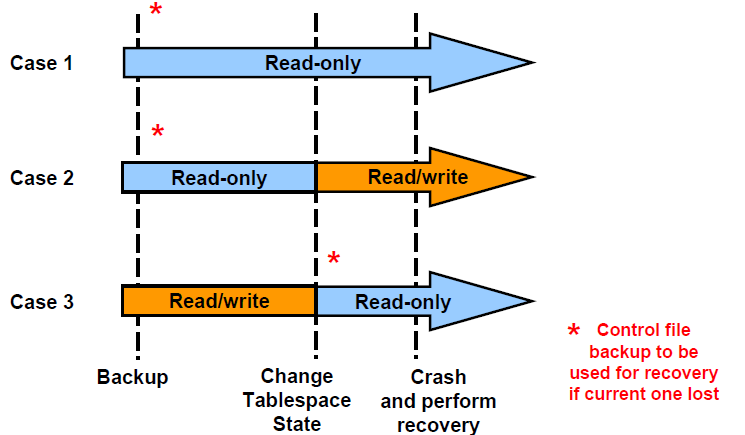
* + Backup full database, chuyển tất cả các file backup đó sang 1 server khác, restore and recovery database sau đó open database.

==================/

* + Backup tablespace, giả sử lỗi và cần restore tablespace đó
  + Backup datafile, giả sử datafile lỗi và cần restore.

================/

* + Restore read-only tablespace
    - * 1. Recover read-only tablespace (3 trường hợp)



***Hình 2.2.10: Các trường hợp recover read-only tablespace***

**Quy trình:**

|  |  |  |
| --- | --- | --- |
| Các trường hợp | Còn control files hiện tại | Sử dụng backup control files |
| Trường hợp 1 | 1. Restore data files 2. Mount database 3. Recover database 4. Open database | 1. Restore toàn bộ database files 2. Mount database 3. [Offline read-only tablespace](#OFfline_tablespace) 4. RECOVER DATABASE USING BACKUP CONTROL FILE; 5. Open database với tùy chọn RESETLOGS 6. [Online read-only tablespace](#Online_tablespace) 7. [Tạo thêm temp file](#add_temp_file) 8. [Backup database](#backup_full) 9. [Cập nhật vào nhật ký DBA](#xemnhatkyDBA) |
| Trường hợp 2 |  | 1. Restore toàn bộ database files 2. Mount database 3. Recover database using backup control file 4. Open database với tùy chọn RESETLOGS 5. [Tạo thêm temp file](#add_temp_file) 6. Backup database 7. Cập nhật vào nhật ký DBA |
| Trường hợp 3 |  | 1. Restore toàn bộ database files 2. Mount the database 3. Recover database using backup control file 4. Offline tablespace Read - Only 5. Open database với tùy chọn RESETLOGS 6. Online tablespace Read – Only 7. Backup database 8. Cập nhật vào nhật ký DBA |

*Bảng 2.2.2: Các trường hợp recover read-only tablespace*

* + Restore controlfile: Copy file k lỗi và đổi tên hoặc sử dụng RNAN
  + Restore redo log file
  + Nắm được quy tắc restore datafile ở chế độ noarchive log mode và archivelog mode.
  + Point in time recovery database. 🡪 Admin II.
* Sử dụng flashback
  + Flashback table: sử dụng khi nào, dữ liệu nào ảnh hưởng
  + Flashback drop
  + Flashback database. 🡪 admin II
  + Flashback query, transaction. 🡪 admin II

# Moving Data

* Directory: Tạo và gán quyền, hiểu về owner.

create directory dump\_dir as '/opt/oracle/backup/export';

grant read, write on directory dump\_dir to system;

select \* from dba\_directories;

* Không có owner của người tạo directory. Owner luôn là sys. Mặc dù user tạo ra directory là user khác.
* SQL Loader: Tạo file sample và sử dụng sqlldr để load vào database. Viết controlfile.

Control file: ======================

load data

CHARACTERSET UTF8

infile '/home/oracle/datafrommMssql/test1data.txt'

append into table chien.test1

fields terminated by X'9' TRAILING NULLCOLS

( id, name )

sqlldr chien/chien control=/home/oracle/test\_sqlloader.ctl

test1data.txt

1 so mot test sql load

2 so hai test sql load

3 so ba test sql load

Mã ascii <http://www.ascii.cl/>

FIELDS TERMINATED BY x'A7' --> §

FIELDS TERMINATED BY x'9' --> tab

FIELDS TERMINATED BY ','

FIELDS TERMINATED BY '|'

SELECT ASCII('a') FROM dual;

SELECT ASCII('§') FROM dual;

SELECT chr(65) FROM dual;

SELECT TO\_NUMBER('7A', 'XX')FROM DUAL; --> chu z

SQL> SELECT TO\_NUMBER('A7', 'XX')FROM DUAL; --> chu §

TO\_NUMBER('A7','XX')

--------------------

167

SQL> SELECT chr(167) FROM dual;

C

-

§

* Sử dụng expdp và impdp, viết script và chạy trên DB.

nohup expdp userid=chien/chien tables=chien.test1 directory= dump\_dir DUMPFILE=20131016test1.dmp LOGFILE=20131016test1.log JOB\_NAME=20131016test1.expdp &

nohup impdp userid=chien/chien tables= chien.test1 directory= dump\_dir DUMPFILE=20131016test1.dmp LOGFILE=20131016test1.impdp.log JOB\_NAME=20131016test1.impdp REMAP\_TABLE=test1: test1copy &

* Đặt lịch expdp hoặc impdp cho DB. (bằng EM)
* Cấu hình một số lệnh remap. Chú ý việc đổi tên 1 schema.

Remap\_schema=oldname:newname.

* External table. Tạo và insert dữ liệu, query dữ liệu.

external\_table.txt

1,'Xin chao heoheie

2,'hello world'

3,'Ve cua quy khach da duoc xac nhan: Ma: 50171. Phim: 3D 48FS - The Hobbit.Ngay: 5/01/2013. Thoi gian:12:35PM. Rap: Pico Mall'

4,'ok khong nao'

5,'ve cua quy khach da duoc xac nhan: Ma: 50171. Phim: 3D 48FS-The Hobbit.

Ngay: 5/01/2013. Thoi gian:12:35PM.Rap: Pico Mall'

6, 'vo ly, tai sao lai khong duoc hehe'

================

CREATE TABLE oldemp ( fname char(25), lname CHAR(200))

ORGANIZATION EXTERNAL (TYPE ORACLE\_LOADER DEFAULT DIRECTORY dump\_dir ACCESS PARAMETERS

(RECORDS DELIMITED BY NEWLINE NOBADFILE NOLOGFILE FIELDS TERMINATED BY ','

(fname CHAR, lname CHAR)) LOCATION ('external\_table.txt')) PARALLEL 5 REJECT LIMIT 200;

* Giải thích **TRANSFORM=STORAGE:n 🡪 bỏ qua storage.**

Enables you to alter object creation DDL for objects being imported.

STORAGE - If the value is specified as y, the storage clauses are included, with appropriate DDL. The default is y. This parameter is ignored if SEGMENT\_ATTRIBUTES=n.