The What, How and Why of ORACLE RMAN

Maxym Kharchenko
Suncoast ORACLE Users Group
January 2009

Who am I

- Maxym Kharchenko, maxym@softcomputer.com
- a Scorpio …
- Used to be a database internals developer
  - Does anybody remember the name dbVISTA? …
    I did not think so ;-)
- Have known RMAN since release 8 … closely …
- ORACLE Certified Master
  - … RMAN was a big part of the exam … shh …

What is it that DBA does ?

- **DBA is someone, who …**
  - **Keeps the database running** …
    - *“If your program doesn’t stop producing 10 Gb of archived logs every minute, the system will die in 30 minutes … Just FYI …”*
  - … **Reasonably fast** …
    - *“You might think of using WHERE clause in that SQL” …*
  - **Protects the data …**
    - *“How many tables did you say your script dropped by mistake?”*
  - **… and keeps it reasonably secure**
    - *“Sorry, the CEO asked NOT to make his salary world viewable” …*
  - Etc etc etc

Recover that data OR die!

- All DBA responsibilities are important
- But NONE is as important as …
  - **Be able to recover data when necessary !**
- People will (reasonably) **tolerate**:
  - System slowness
  - Lack of security
  - Even downtime …
- But … **if you lose production data and CANNOT recover it → you are really screwed !**
  - Nobody will care about the reasons
  - One strike and you are out!

ORACLE Backup and Recovery are NOT for the average Joe

```
-- Backup data files
Get tbs list/exclude not needed
FOR i IN 'all tablespaces' DO
ALTER tablespace $I BEGIN BACKUP;
 Get the list of tbs files
 Copy tbs files
 ALTER TABLESPACE $I END BACKUP;
DONE
-- Backup archived logs
Get the list of ARC destinations
Copy archived logs
```

How to make a 'disaster' recovery

- Let's face it – ORACLE backups and recoveries are pretty *complex* …
- Making a mistake is easy if you do NOT understand what you are doing …
  - … and sometimes even if you do …
- Successful recovery TEST does NOT mean that future recovery will work
  - Surprise, surprise …

Why DBAs make mistakes (with backups) ?

- There are 2 major reasons:
  - **Junior DBAs** → the problem is ***complexity***
    - You need to know a lot and experience a lot
    - Some things will only come with experience

    It is EASY to make a mistake if you don't know enough …
  - **Senior DBAs** → the problem is that the process is ***too mundane***
    - Repeating the same 40 operations over and over is *boring*
    - People tend to *optimize* the work and skip the *unnecessary* steps
    - Documentation is usually ignored

    It is even EASIER to make a mistake if you know too much …

Coming of ‘R-man’

- To recover the data in all cases … guaranteed, you need to:
  - Know it all
  - Do NOT forget anything
- The (sad) conclusion is:
  - **No human will be able to do it …**
- That is why, we need a ROBOT

The WHAT of RMAN

RMAN Makes Things Simple

1

```
ALTER DATABASE BEGIN BACKUP;
Get the list of database files
(optional) Exclude read only/unneeded
Copy database files
Check that copy is successful
ALTER DATABASE END BACKUP;
ALTER SYSTEM SWITCH ARCHIVE LOG CURRENT;
Get the list of ARC destinations
Copy archived logs
Check that copy is successful
ALTER DATABASE BACKUP CONTROLFILE TO …;
Copy controlfile
Copy parameter/spfile/password file
Check that copy is successful
```

```
backup full database plus archivelog;
```

RMAN is a Detail Junkie

2

EXECUTES:

- **Every single step in the procedure …**
- **… Including implied (‘*switch that log*’) …**

REMEMBERS:

- **What backups to keep, remove …**
- **Special cases (i.e. KEEP FOREVER)**

KEEPS TRACK AND VALIDATES:

- **Is my database recoverable ?**
- **Are my backups ‘good’ ?**

ADVISES AND MAKES DECISIONS:

- **What to backup ?**
- **How to recover ?**

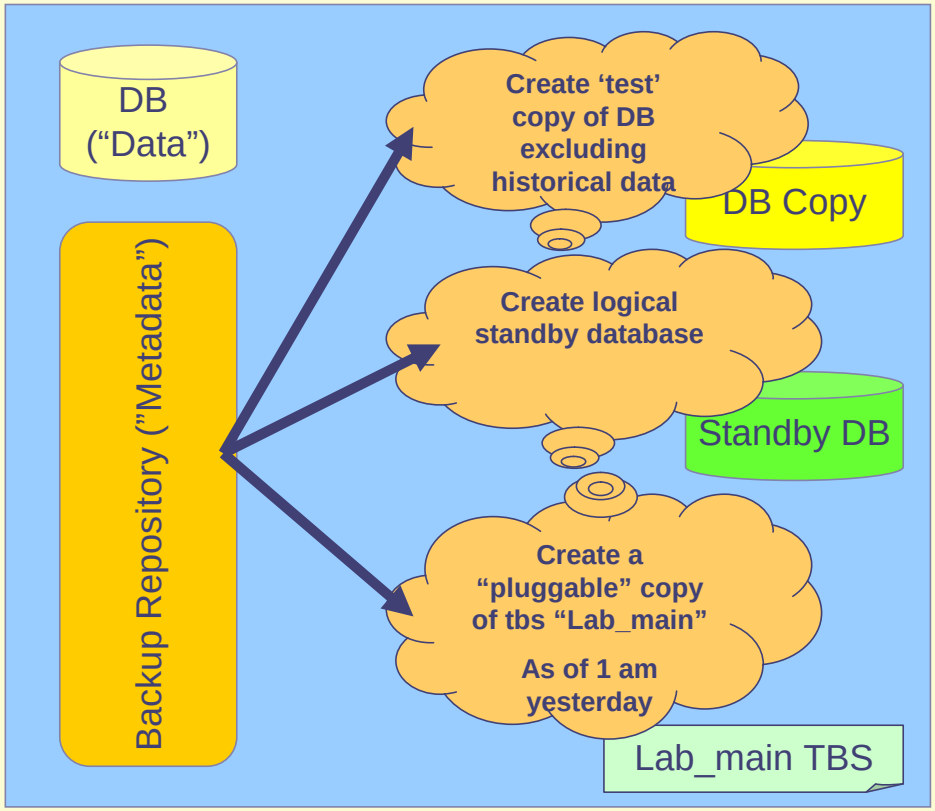
RMAN has Advanced Tools

3

- Incremental Backups
- Compressed Backups
- Encrypted Backups
- Block Recovery
- Backup Flow Control
- Etc …

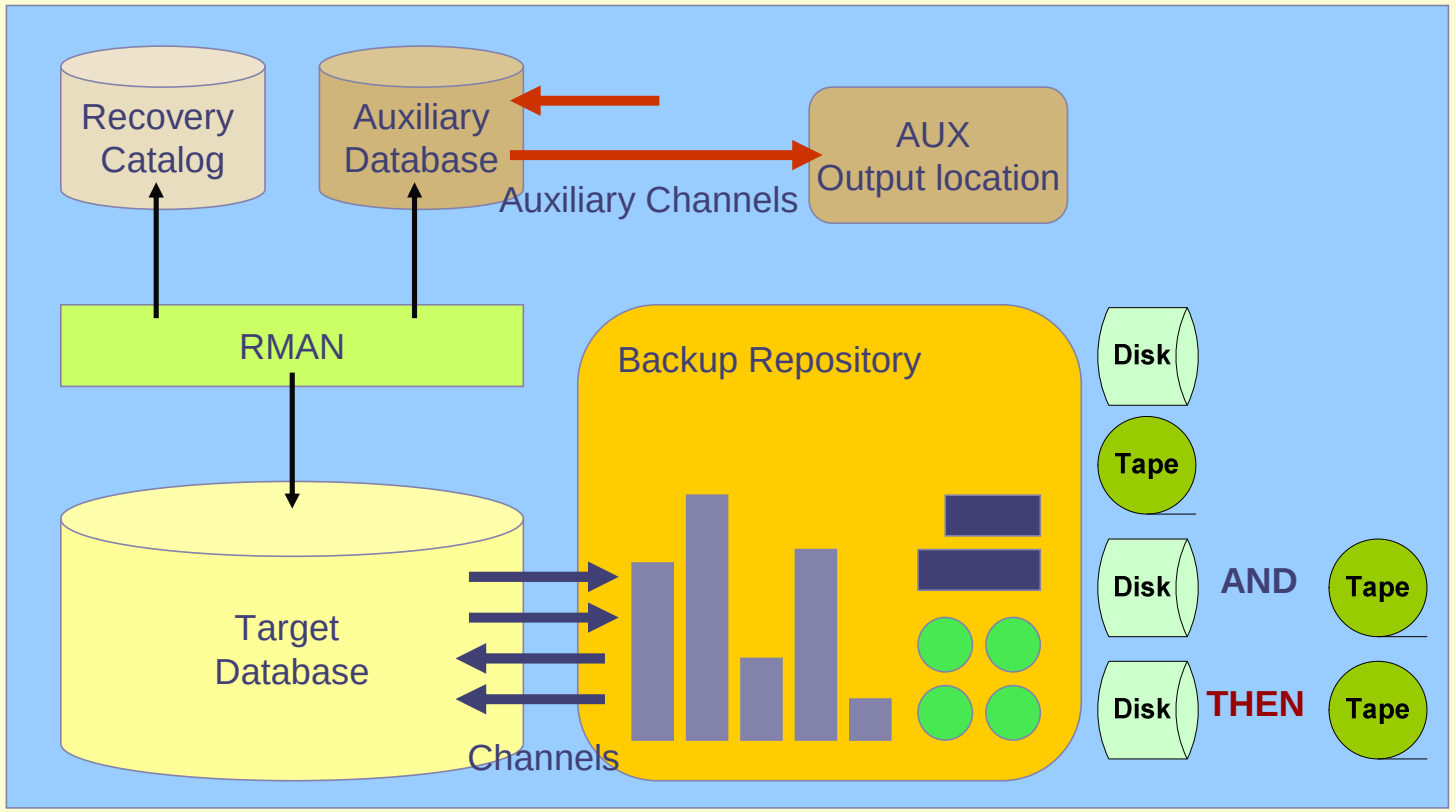
RMAN Manages Binary Metadata

4

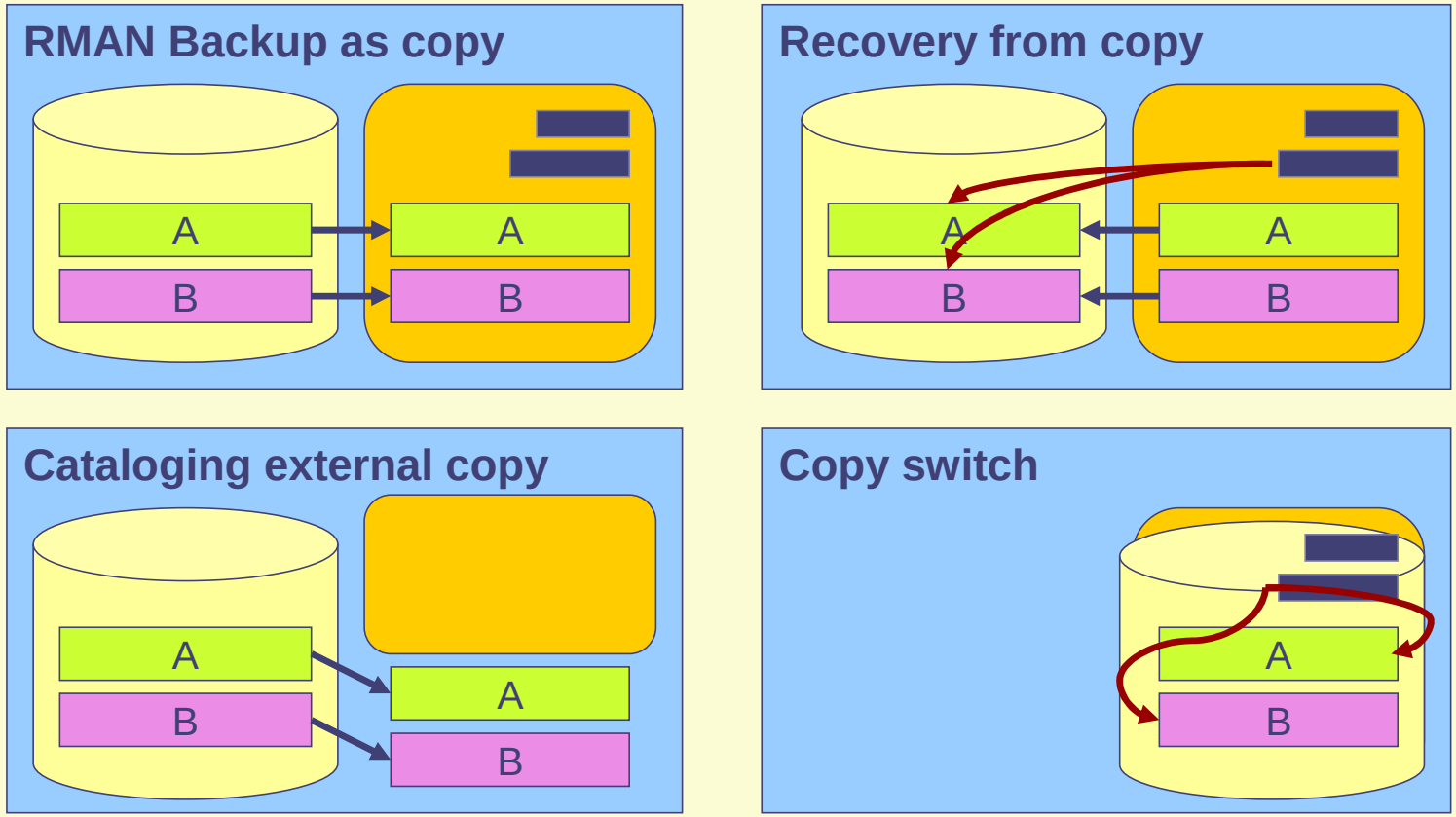


The HOW of RMAN

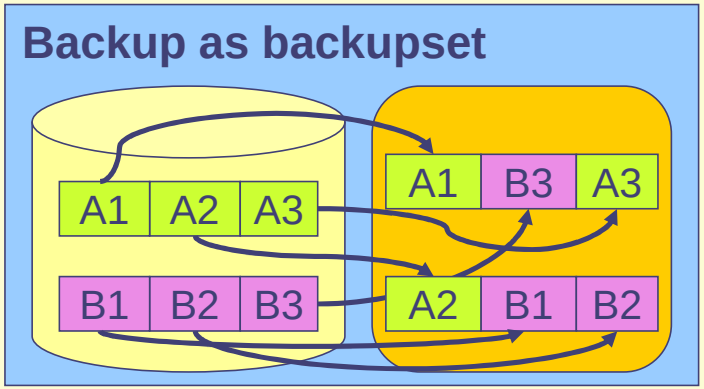
RMAN configuration in a nutshell

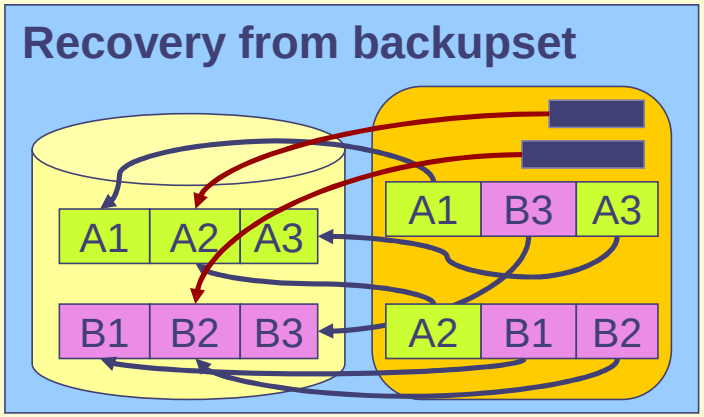


RMAN backup – Data Copies



RMAN backup – Backupsets

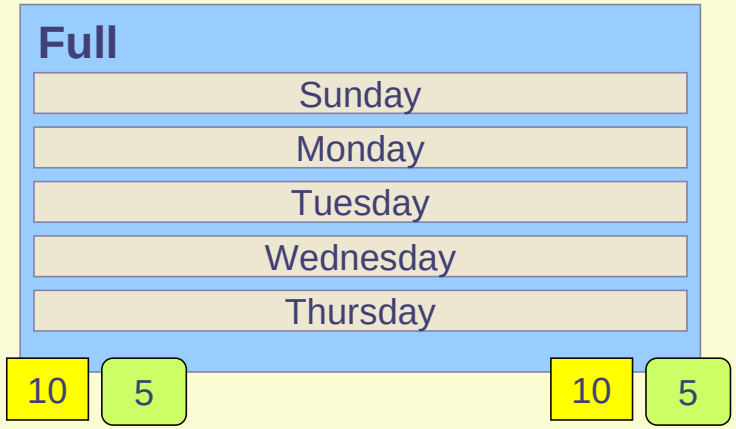


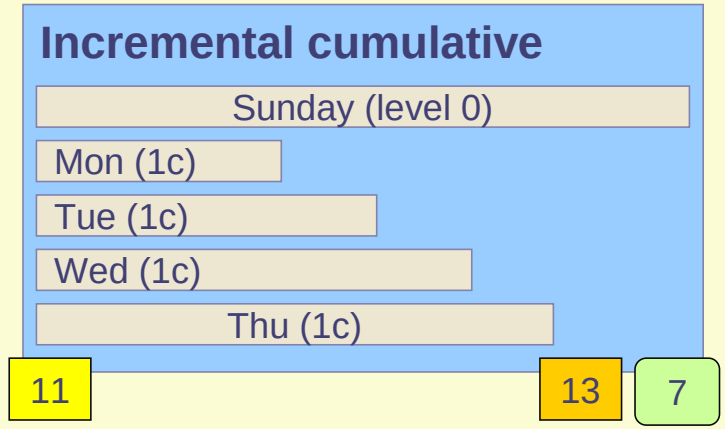


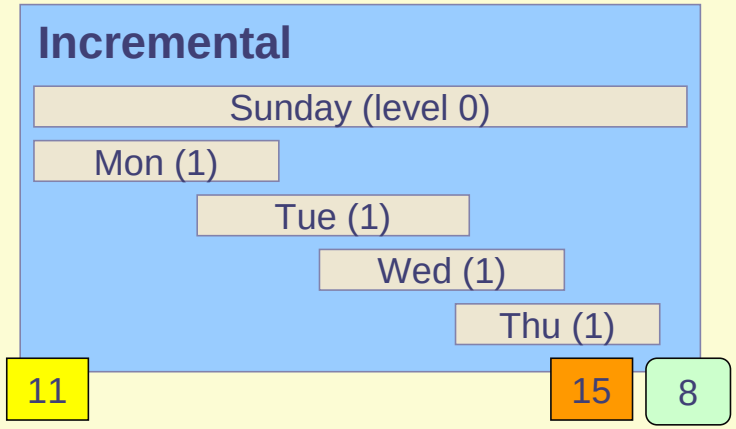
1. **Backups can be smaller:**
   - Incremental, Compressed
2. **(some) Backups can be faster:**
   - Incremental with B.C.T.
3. **(some) Recoveries can be faster:**
   - Incremental vs. ARC
4. **(some) Backup operations can be optimized**
   - Zero block, UNDO optimization
5. **System can be less affected**
   - Native RMAN, MINIMIZE LOAD
6. **Can backup directly to “tape”**

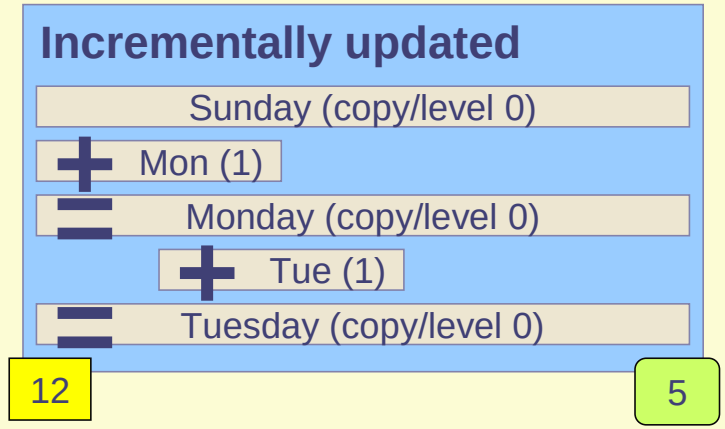
Full backup and recovery will be slower

How to be Incremental

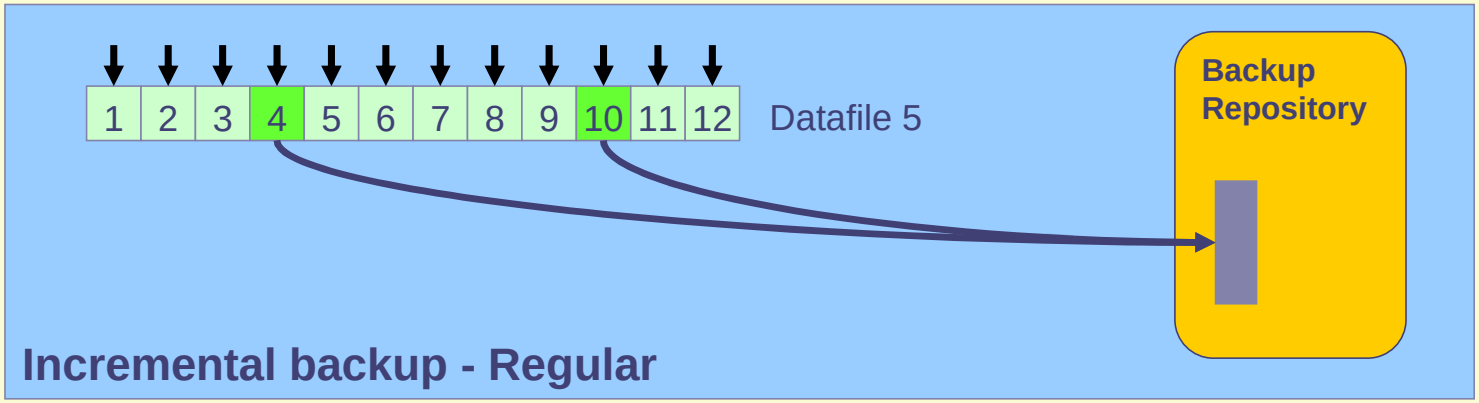


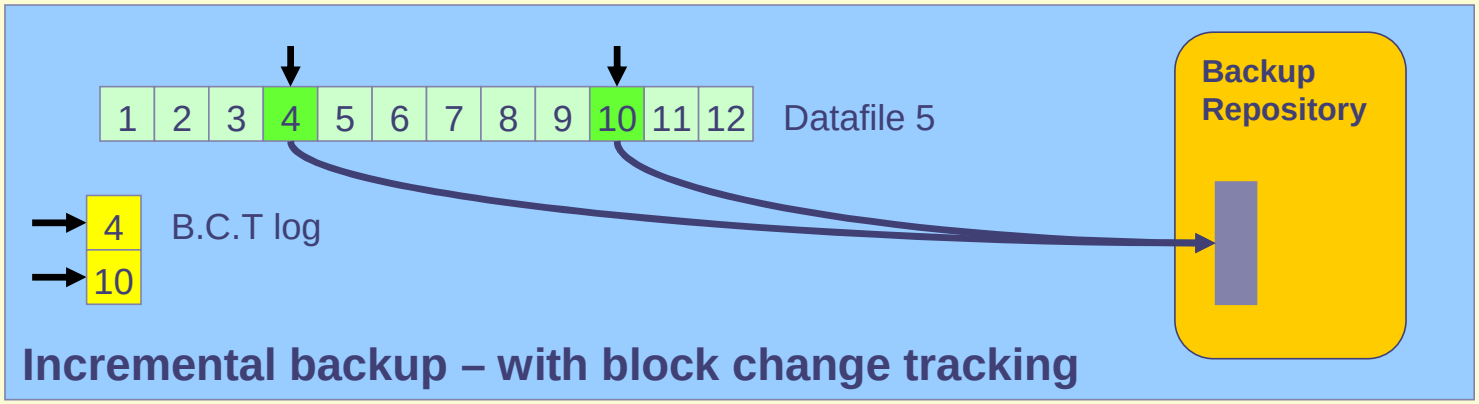






Speeding Up Incremental Backups





RMAN – Making a Backup

```
RMAN> run {
   allocate channel ch1 type sbt
      parms 'ENV=(TDPO_OPTFILE=/opt/tsm/tdpo.opt)' rate=100M
   allocate channel ch2 type sbt
      parms 'ENV=(TDPO_OPTFILE=/opt/tsm/tdpo.opt)' rate=100M;

   backup incremental level 0 database format '%d_s%s_p%p_%T.bkpdf';
   backup archivelog all format '%d_s%s_p%p_%T.bkpal';

   delete noprompt archivelog all until time "sysdate-2/24";

   backup current controlfile format '%d_s%s_p%p_%T.bkpcf';

   release channel ch1;
   release channel ch2;
}
```

```
RMAN> backup incremental level 0 database plus archivelog;
```

RMAN – The Beauty of Stored Parameters

```
RMAN> show all;


CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;
CONFIGURE BACKUP OPTIMIZATION ON;
CONFIGURE DEFAULT DEVICE TYPE TO 'SBT_TAPE';
CONFIGURE CONTROLFILE AUTOBACKUP ON;
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE SBT_TAPE
   TO 'autocf_%d_%F'; # default
CONFIGURE DEVICE TYPE 'SBT_TAPE' PARALLELISM 2
   BACKUP TYPE TO BACKUPSET;
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE'
   PARMS  'ENV=(TDPO_OPTFILE=/opt/tsm/tdpo.opt)'
   FORMAT '%d_s%s_p%p_%T.bkpset';
CONFIGURE CHANNEL 1 TYPE 'SBT_TAPE' RATE 100M;
CONFIGURE CHANNEL 2 TYPE 'SBT_TAPE' RATE 100M;
```

Disk or Tape ?

```
RMAN> backup device type DISK full database plus archivelog;
```

```
RMAN> backup device type SBT full database plus archivelog;
```

```
Sunday:
RMAN> backup device type SBT incremental level 0 database
   plus archivelog;

Monday:
RMAN> backup device type DISK incremental level 1 database;
RMAN> backup device type SBT archivelog all;
```

```
RMAN> backup incremental level 0 database plus archivelog;
RMAN> backup recovery area;
```

RMAN - Recovering Your Data

```
set dbid 2226857552;
startup force nomount
restore spfile from autobackup;

shutdown immediate
startup nomount
restore controlfile from autobackup;
alter database mount;

# (optional): catalog start with '…';
# Find latest available SCN, i.e. from v$archived_log
restore database until scn 1234567;
recover database until scn 1234567;

alter database open resetlogs;
```

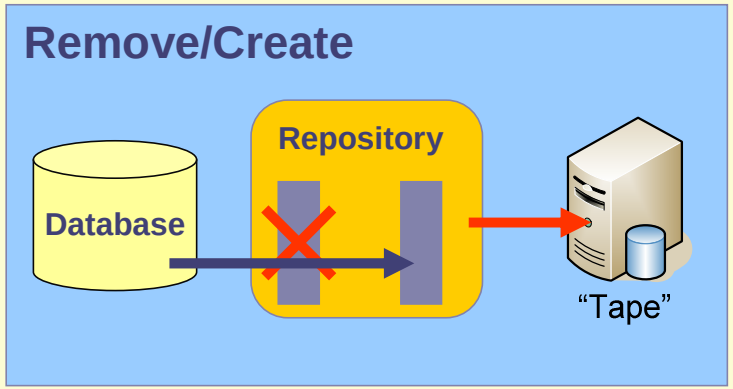
```
RMAN> restore spfile from '/ora05/flash/DEV10/autobackup/
   2008_12_21/o1_mf_s_674071460_4nxlvpdk_.bkp';
```

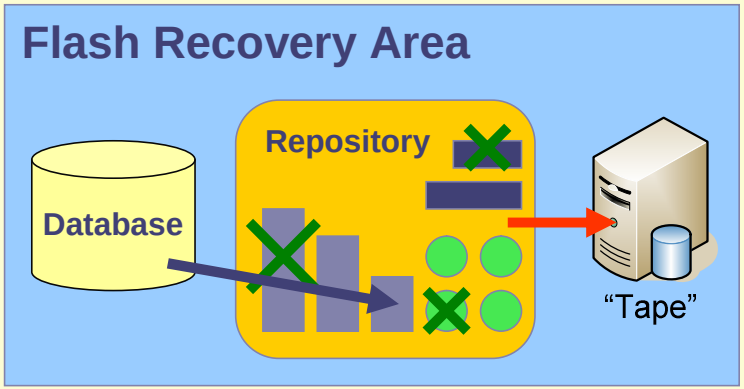
```
RMAN> restore controlfile from autobackup
   db_recovery_file_dest='/ora05/flash' db_name='orcl';
```

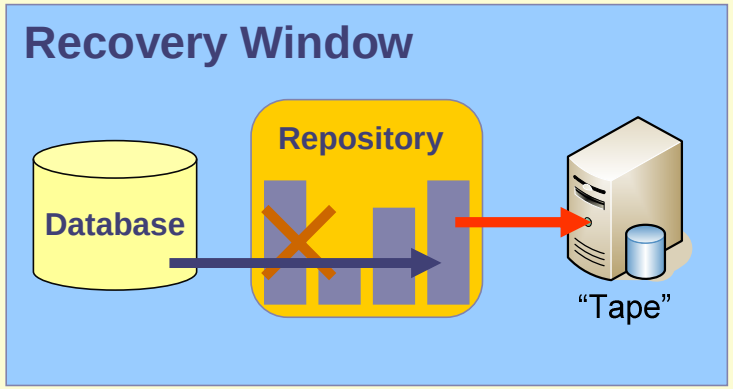
Block Recovery

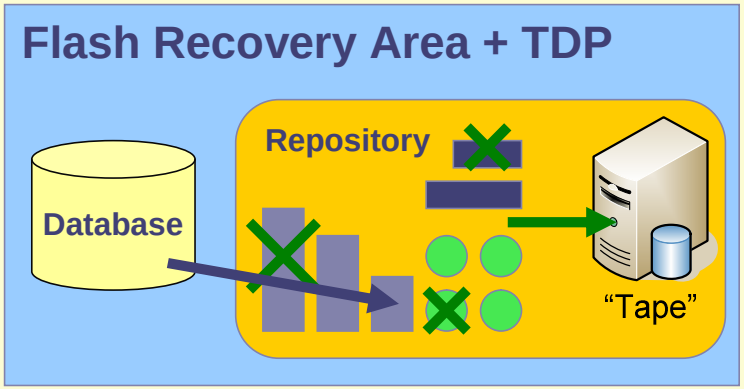
```
# First of all - find which blocks are corrupted

# 1. Alert log
2. RMAN> backup validate check logical datafile N;
   SQL> SELECT * FROM v$database_block_corruption;
3. RMAN> backup check logical datafile N;
4. UNIX> dbv data_file

# Then, repair it like this:
RMAN> blockrecover datafile X block Y;

# Or, like this:
RMAN> blockrecover corruption list;
```

RMAN – How to Maintain Backups

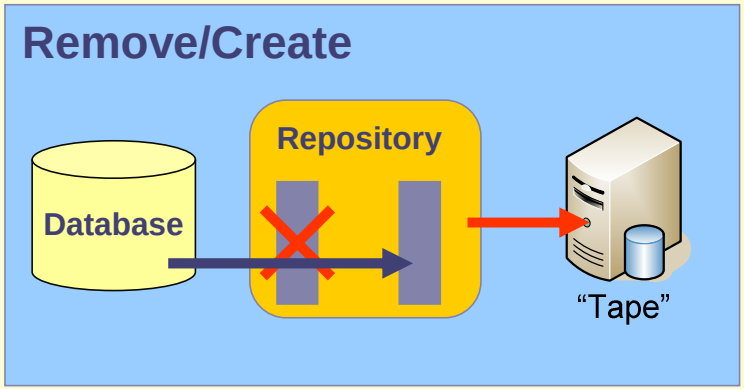


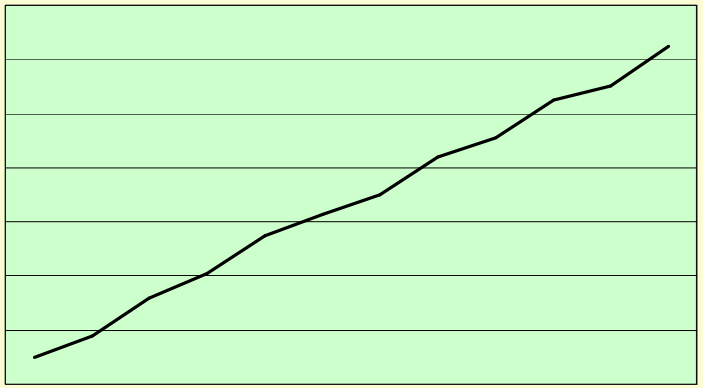


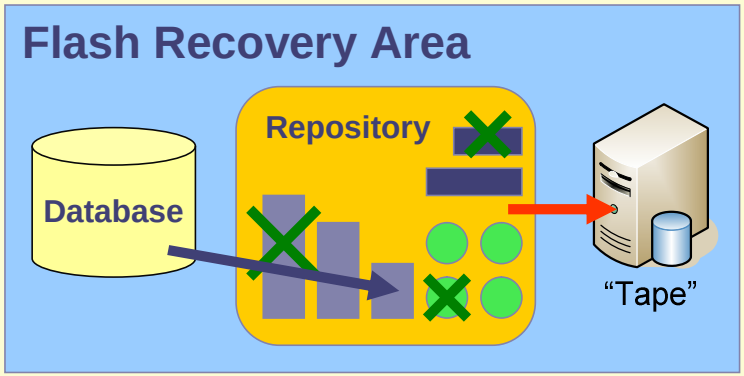


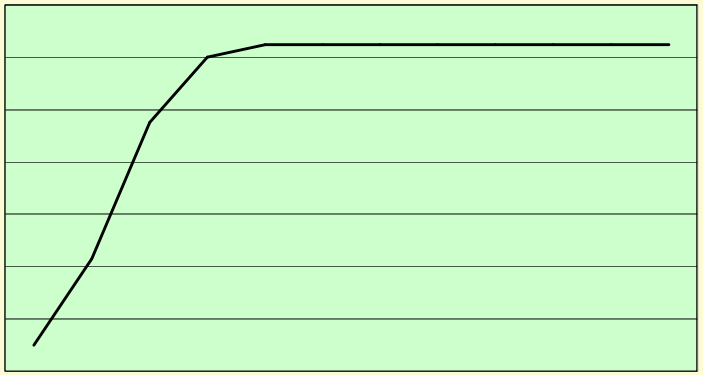


FRA Sizing Considerations









Is your database recoverable ?

```
RMAN> list backup of database summary;
RMAN> list backup of database by file;
```

```
RMAN> report need backup database;
RMAN> report unrecoverable database;
```

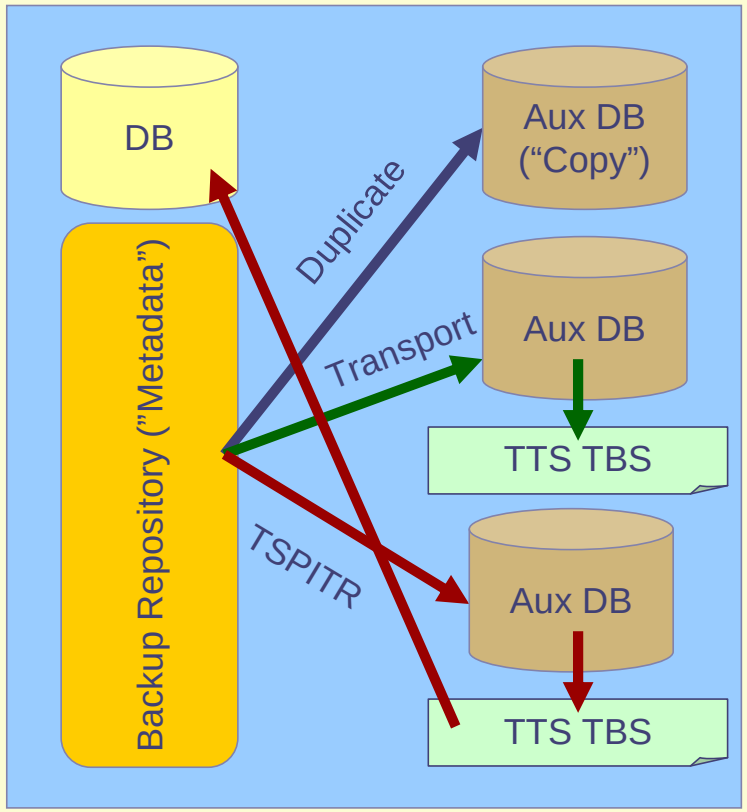
```
RMAN> restore database preview;
RMAN> recover database test;
```

```
RMAN> crosscheck backup;
RMAN> crosscheck copy;
```

```
RMAN> backup incremental level 0 check logical database;
SQL> SELECT * FROM v$database_block_corruption;
```

```
RMAN> restore database validate check logical;
SQL> SELECT * FROM v$database_block_corruption;
```

RMAN - Managing Your Binary Metadata




```
duplicate target database
   to testdb skip tablespace tbs1
   until time 'sysdate-1'
   logfile
   group 1 ('/testdb/redo01.arc')
      size 100M reuse,
   group 2 ('/testdb/redo01.arc')
      size 100M reuse;
```

```
transport tablespace lab
   tablespace destination '/tbs'
   auxiliary destination '/aux'
   until time 'sysdate-1';
```

```
recover tablespace users, tools
   auxiliary destination '/aux'
   until time 'sysdate-1';
```

RMAN Extras

- Database command console
- ASM data manager
- Datafile *platform* conversion
- Script processor

The WHY of RMAN

Why use RMAN?

- **RMAN** utility:
  - Can simplify many backup and recovery operations
    - Doing them faster and, in many cases, better than *'regular'* backup tools
  - Let's you do some cool things with ORACLE
  - Helps avoid mistakes
- But most importantly, RMAN **allows a DBA to be LAZY** (in a good way)
  - And spend your valuable time on *other* cool things ORACLE
    - Because, let's admit it … backups **ARE** somewhat … *boring* ;-)

Any Questions ?