5

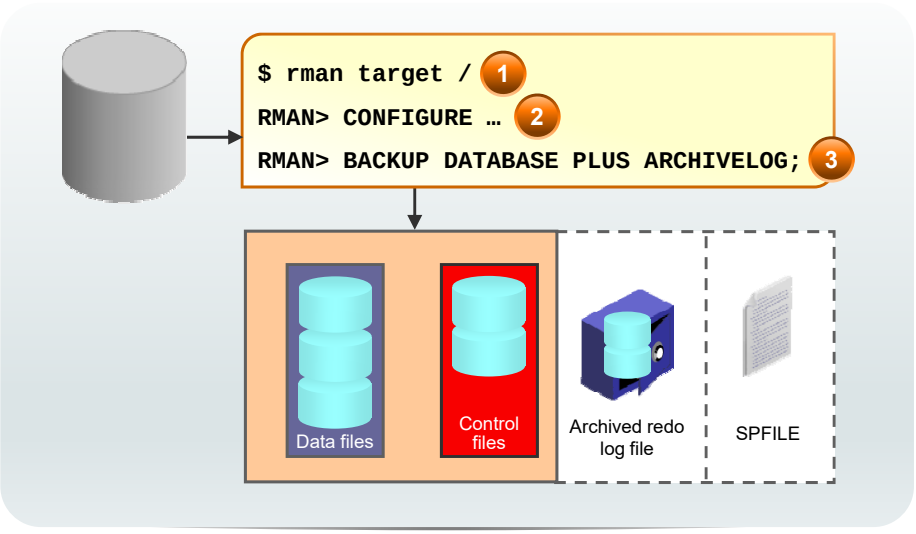
Creating Database Backups

Objectives

After completing this lesson, you should be able to:

- Create whole backups
- Create full and incremental backups
- Configure block change tracking
- Use Oracle-suggested backup strategy
- Back up the control file to a trace file
- Report and manage backups

Using RMAN Commands to Create Backups



Syntax and Clauses in RMAN

```
$ export ORACLE_SID=cdb1
$ rman TARGET /                    $ rman TARGET jim@pdb1
```

- DATABASE keyword operates on all PDBs and CDB root or on only one PDB.

```
RMAN> BACKUP DATABASE;
RMAN> RECOVER DATABASE;
```

- PLUGGABLE DATABASE keywords operate on individual PDBs.

```
RMAN> BACKUP PLUGGABLE DATABASE hr_pdb, sales_pdb;
RMAN> RECOVER PLUGGABLE DATABASE hr_pdb;
```

- Back up, restore, recover the CDB root using CDB$ROOT keyword.

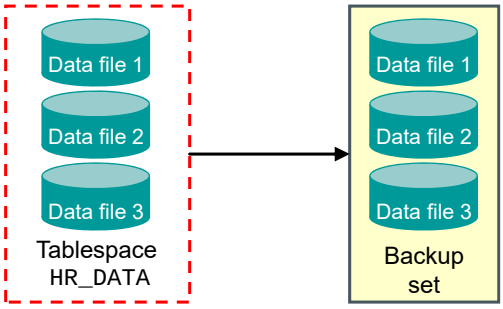
```
RMAN> BACKUP PLUGGABLE DATABASE "CDB$ROOT";
```

- Qualify tablespace of PDB with PDB name.

```
RMAN> BACKUP TABLESPACE sales_pdb:tbs2;
RMAN> RESTORE TABLESPACE system;
```

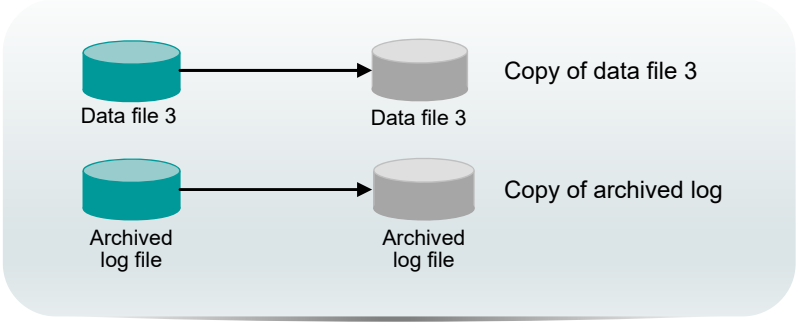
Creating Backup Sets

```
RMAN> BACKUP AS BACKUPSET
   2> FORMAT '/BACKUP/df_%d_%s_%p.bus'
   3> TABLESPACE hr_data;
```

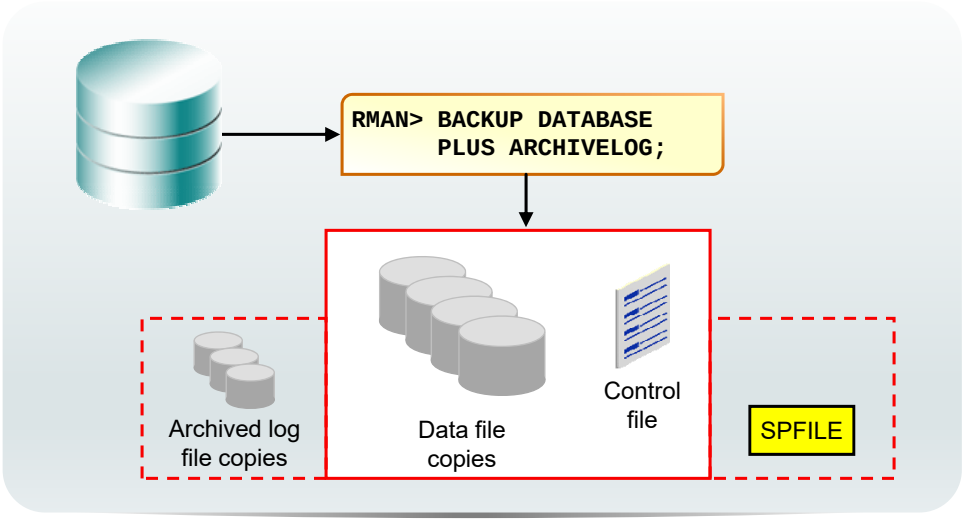


Creating Image Copies

```
RMAN> BACKUP AS COPY DATAFILE '/ORADATA/users_01_db01.dbf';
RMAN> BACKUP AS COPY ARCHIVELOG LIKE '/arch%';
```



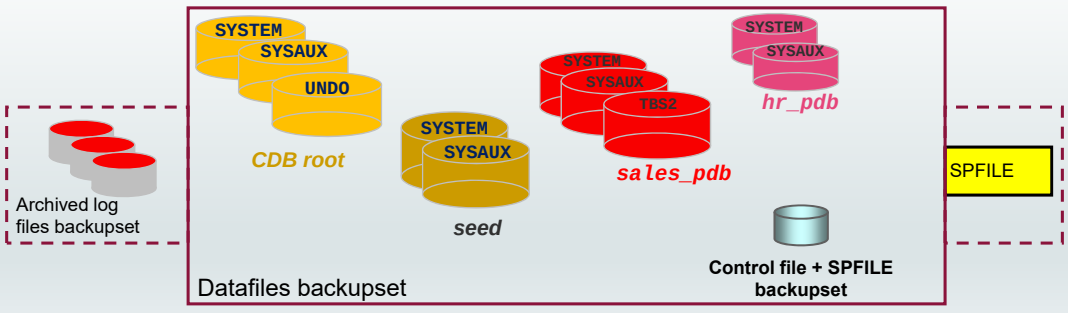
Creating a Whole Database Backup



CDB Backup: Whole CDB Backup

Back up all PDBs datafiles and CDB root files.

```
$ rman TARGET /
RMAN> BACKUP DATABASE PLUS ARCHIVELOG;
```

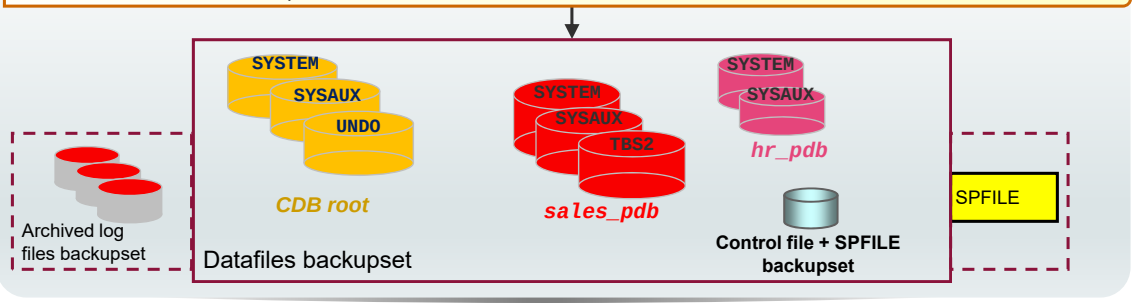


CDB Backup: Partial CDB Backup

Back up the CDB root and/or individual PDBs.

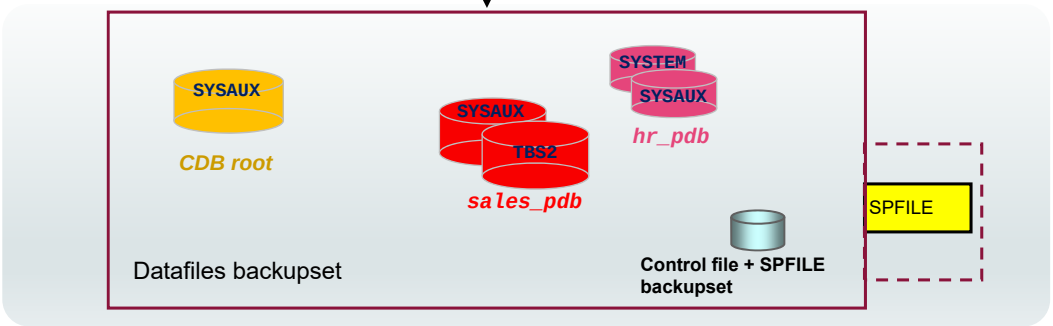
```
$ rman TARGET /
RMAN> BACKUP PLUGGABLE DATABASE "CDB$ROOT", sales_pdb;
RMAN> BACKUP PLUGGABLE DATABASE hr_pdb PLUS ARCHIVELOG;
```

```
$ rman TARGET sys@hr_pdb
RMAN> BACKUP DATABASE;
```



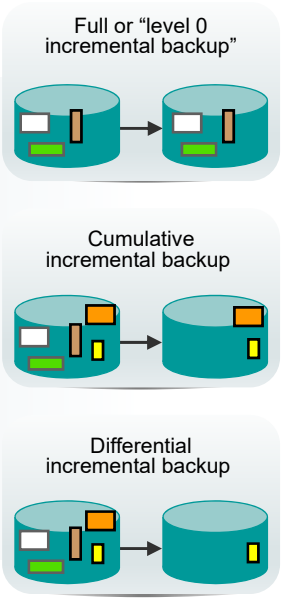
PDB Backup: Partial PDB Backup

```
$ rman TARGET /
RMAN> REPORT SCHEMA;
RMAN> BACKUP TABLESPACE sales_pdb:tbs2;
RMAN> BACKUP TABLESPACE hr_pdb:system, sales_pdb:sysaux;
RMAN> BACKUP TABLESPACE sysaux, hr_pdb:sysaux;
```



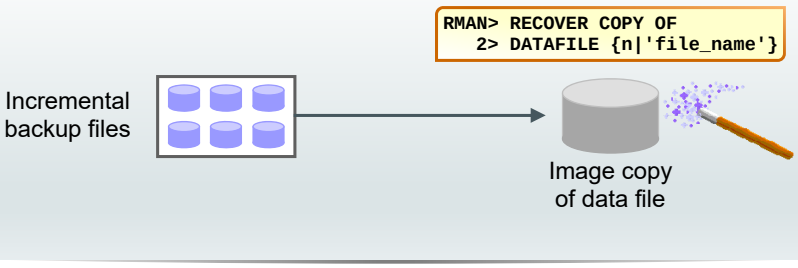
Review: RMAN Backup Types

- A *full backup* contains all used data file blocks.
- A *level 0 incremental backup* is equivalent to a full backup that has been marked as level 0.
- A *cumulative level 1 incremental backup* contains only blocks modified since the last level 0 incremental backup.
- A *differential level 1 incremental backup* contains only blocks modified since the last incremental backup.



Incrementally Updated Backups

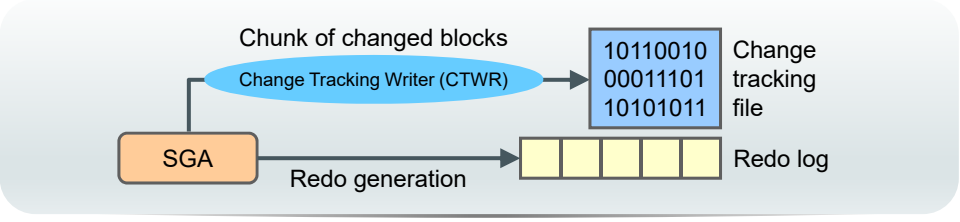
- Image copies are updated with all changes up to the incremental backup SCN.
- Incremental backup reduces the time required for media recovery.
- With incrementally updated backups, you can use the `SWITCH` command during the recovery operation.



Fast Incremental Backup

Implemented by block change tracking, which:

- Maintains a record of block chunks that have changed since the last backup
- Writes this record to a file, as redo is generated
- Is automatically accessed when a backup is done and can make the backup complete more quickly
- Is optimized for up to eight incremental backups
- Is recommended if the changes are less than 20 percent



Maintaining the Block Change Tracking File

- The `DB_CREATE_FILE_DEST` initialization parameter provides the default destination.
- Enable or disable with:

```
ALTER DATABASE
{ENABLE|DISABLE} BLOCK CHANGE TRACKING
[USING FILE '...']
```

- Rename the block change tracking file with the `ALTER DATABASE RENAME` command. (The database must be in `MOUNT` state.)

Monitoring Block Change Tracking

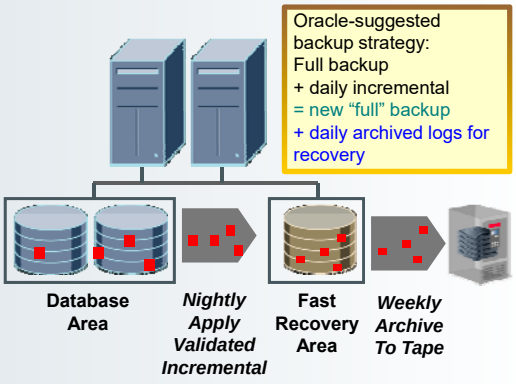
```
SQL> SELECT filename, status, bytes
  2  FROM   v$block_change_tracking;
```

```
SQL> SELECT file#, avg(datafile_blocks),avg(blocks_read),
  2         avg(blocks_read/datafile_blocks) * 100 AS PCT_READ_FOR_BACKUP,
  3         avg(blocks)
  4  FROM   v$backup_datafile
  5  WHERE  used_change_tracking = 'YES' AND incremental_level > 0
  6  GROUP  BY file#;

FILE#  BLOCKS_IN_FILE  BLOCKS_READ PCT_READ_FOR_BACKUP BLOCKS_BACKED_UP
-----  --------------  ----------- ------------------ ----------------
    1           56320        4480                  7              462
    2            3840        2688                 70             2408
    3           49920       16768                 33             4457
```


Automatic Disk-to-Disk Backup and Recovery

- Integrated disk-to-disk backup and recovery:
  Low-cost disks used for fast recovery area
- Fast incremental backups:
  Back up only changed blocks



Integrated backup-storage tiering

Backing Up the Control File to a Trace File

- A control file trace backup contains the SQL statement required to re-create the control files in the event that all control files are lost.
- It is recommended to do after each change in the physical structure of the database.
- Control file trace backups may be used to recover from loss of all control files.
- Choose your DBA tool: EM Express, Cloud Control, or command line.

Backing Up the Control File to a Trace File

- Control files can be backed up to a trace file, generating a SQL command to re-create the control file.
- Control file trace backups may be used to recover from the loss of all control files.

```
ALTER DATABASE BACKUP CONTROLFILE TO TRACE
```

Cataloging Additional Backup Files

Using the CATALOG command:

- To catalog existing backup files that are no longer listed in the control file
- To catalog files that were never included in the control file or recovery catalog
- To add the following file types to the recovery catalog:
  - CONTROLFILECOPY: Control file copies
  - DATAFILECOPY: Data file copies
  - BACKUPPIECE: Backup pieces
  - ARCHIVELOG: Archived redo log files
- With the START WITH option:

```
RMAN> CATALOG ARCHIVELOG '/disk1/arch_logs/archive1_731.log',
'/disk1/arch_logs/archive1_732.log';
RMAN> CATALOG START WITH '/tmp/arch_logs/';
```


Reporting on Backups

RMAN commands:

- LIST: Displays information about backup sets, proxy copies, and image copies recorded in the repository
- REPORT: Produces a detailed analysis of the repository
- REPORT NEED BACKUP: Lists all data files that require a backup
- REPORT OBSOLETE: Identifies files that are no longer needed to satisfy backup retention policies

Enterprise Manager Cloud Control:

- Graphical, customizable interface

Using Dynamic Views

Query the following dynamic views in the target database to obtain information about your backups:

- V$BACKUP_SET: Backup sets created
- V$BACKUP_PIECE: Backup pieces that exist
- V$DATAFILE_COPY: Copies of data files on disk
- V$BACKUP_FILES: Information about all files created when creating backups

Summary

In this lesson, you should have learned how to:

- Create whole backups
- Create full and incremental backups
- Configure block change tracking
- Use Oracle-suggested backup strategy
- Back up the control file to a trace file
- Report and manage backups

Practice Overview

- Backing up the Control File
- Verifying Automatic Backups of the Control File and SPFILE
- Creating a Whole Database Backup
- Creating Partial Database Backups
- Configuring Block Change Tracking
- Using Incremental Backup
- Backing Up Additional Database Files