

# Configuring Backup Settings

# Objectives

RMAN usage :

- RMAN commands
- Channel allocation.
- Backup command.
  - Constraints.
  - Parallelization.
  - Backup types.
  - Backup data files.
  - Backup archived redolog files.
  - Whole database backup.
  - Using tags.
  - Backup options.

# Types of RMAN Commands

RMAN commands are of the following types:

- Stand-alone command:
  - Is executed individually at the RMAN prompt
  - Cannot appear as subcommands within RUN
- Job command:
  - Must be within the braces of a RUN command
  - Is executed as a group

Some commands can be executed as either a stand-alone or a job command.

# Job Commands: Example

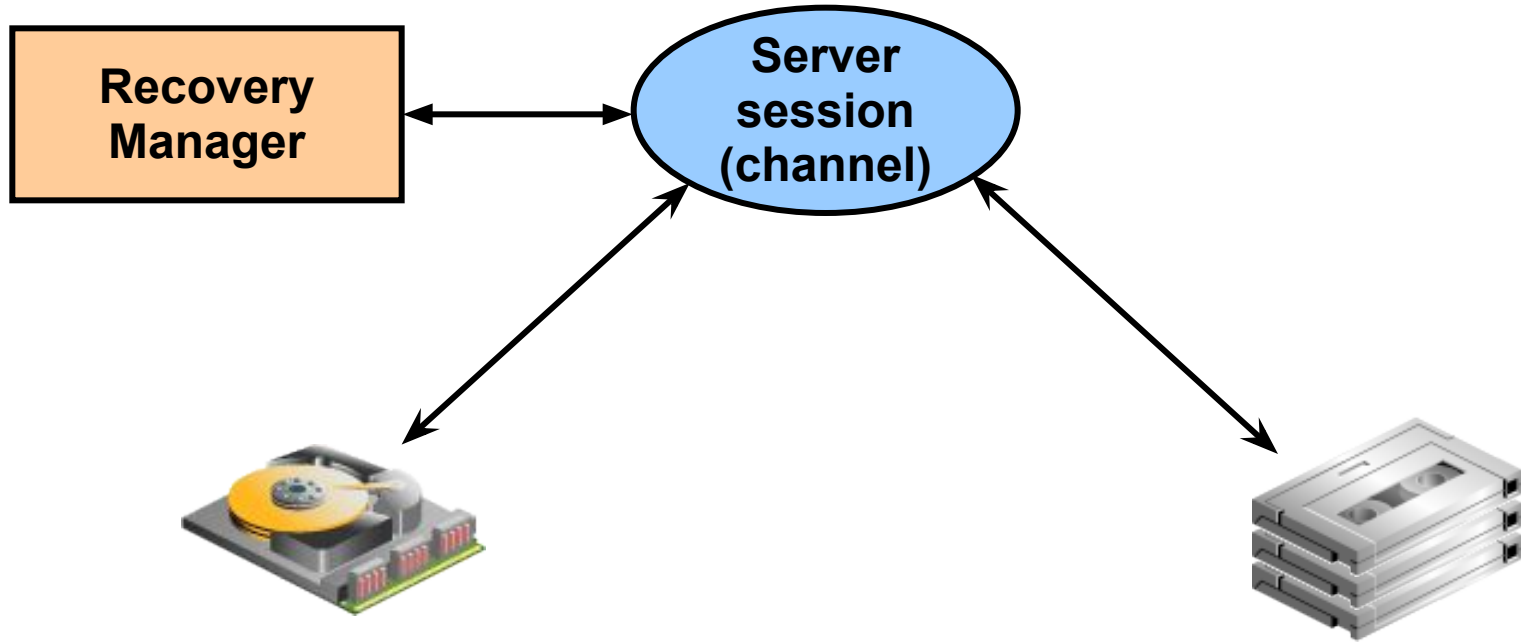
Job commands appear inside a RUN command block:

```
RMAN> RUN
2> {
3>   ALLOCATE CHANNEL c1 DEVICE TYPE DISK
4>   FORMAT "e:\backup\%U";
5>   BACKUP AS BACKUPSET DATABASE;
6>   SQL 'alter system archive log current';
7> }
```

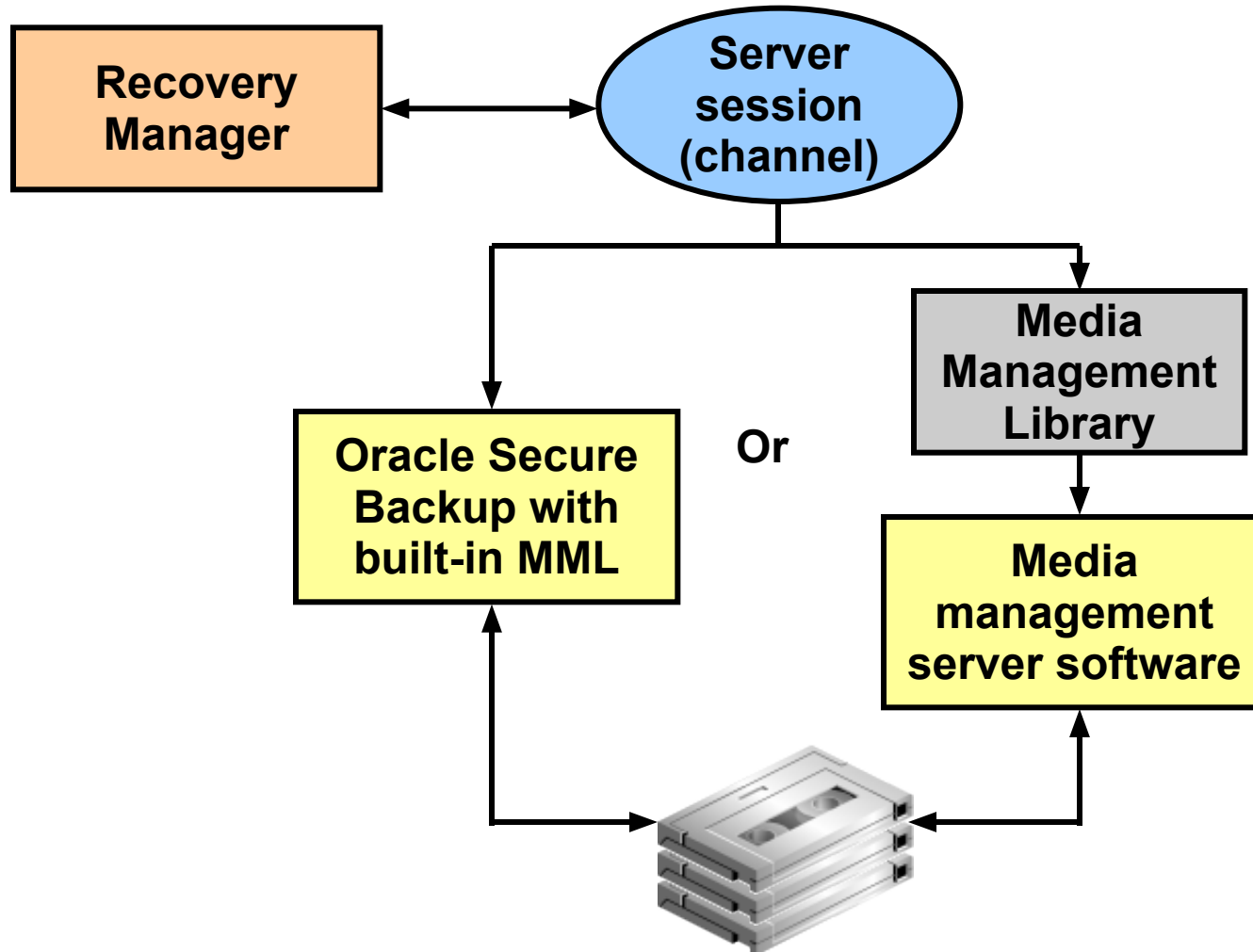
Execution of the entire block starts  
when this line is entered.

Deallocated after the  
RUN block completes

# Channel allocation



# Using a Media Manager



# Configuring and Allocating Channels

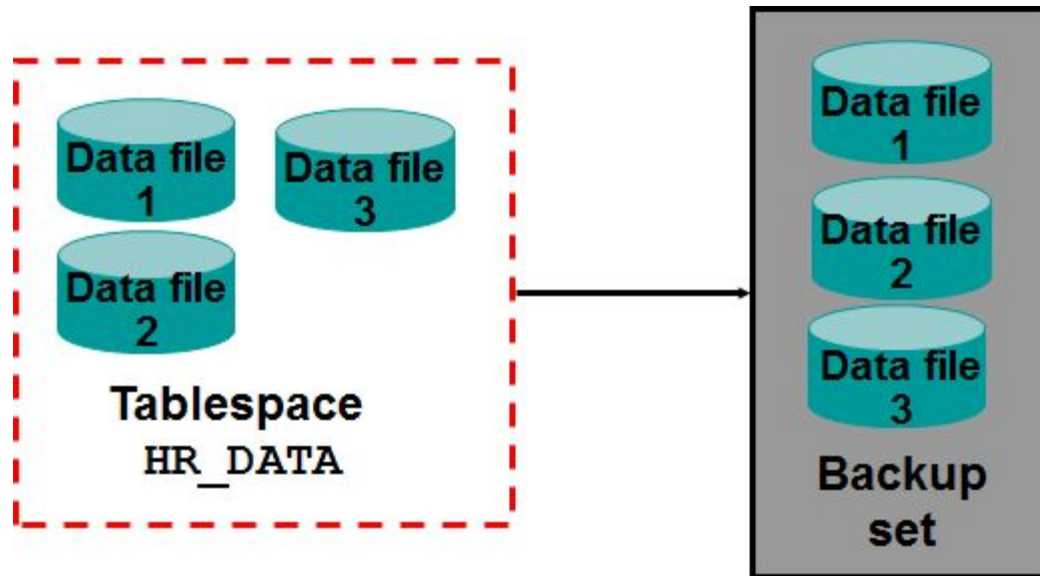
- Configure automatic channels with the CONFIGURE command:

```
RMAN> CONFIGURE DEVICE TYPE sbt;  
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO sbt;  
RMAN> CONFIGURE CHANNEL DEVICE TYPE sbt ...  
RMAN> BACKUP DATABASE;
```

- Allocate channels manually with the ALLOCATE CHANNEL command within a RUN block:

```
RMAN> RUN  
{  
  ALLOCATE CHANNEL ch1 DEVICE TYPE DISK;  
  BACKUP DATABASE PLUS ARCHIVELOG;  
}
```

# RMAN backup command



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



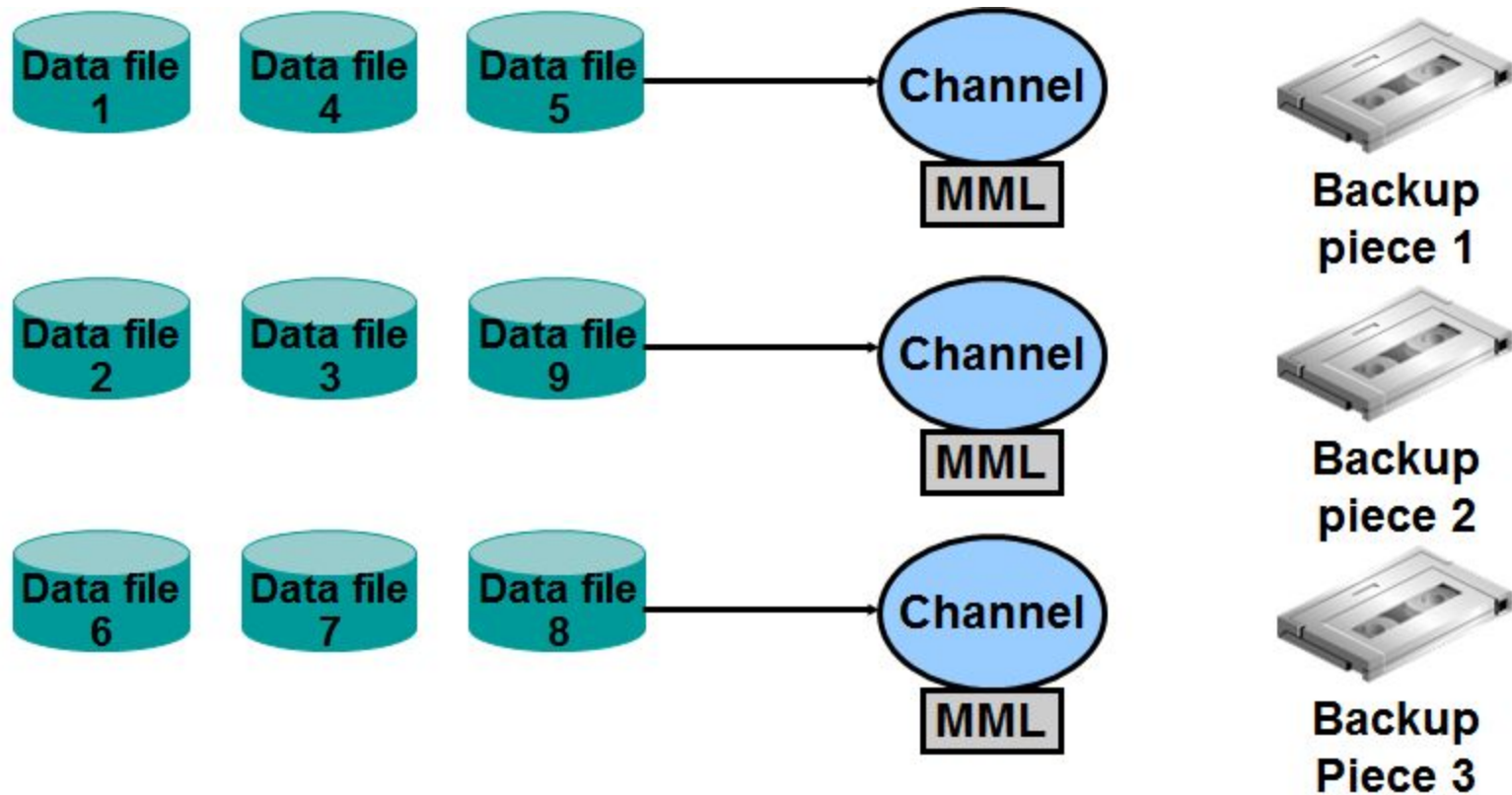
# RMAN backup command

- Type of backup
  - FULL
  - incremental
  - automatically backs up the control file and the current server parameter file
- What to backup
  - DATABASE
  - DATAFILE
  - TABLESPACE
  - ARCHIVELOG
  - CURRENT CONTROLFILE, or SPFILE.
- Whether to backup
  - image copy
  - backup set

# Constraints

- The database must be mounted or open.
- Online redo log backups are not supported
- Only “clean” backups are usable in NOARCHIVELOG mode
- Only “current” data file backups are usable in ARCHIVELOG mode

# Parallelization



# Example

```
RMAN>
```

```
RUN
```

```
{
```

```
  ALLOCATE CHANNEL c1 DEVICE TYPE DISK
```

```
  FORMAT 'e:\backup\df_%d';
```

```
  ALLOCATE CHANNEL c2 DEVICE TYPE disk
```

```
  FORMAT 'd:\ora2\df_%d';
```

```
  BACKUP as backupset database
```

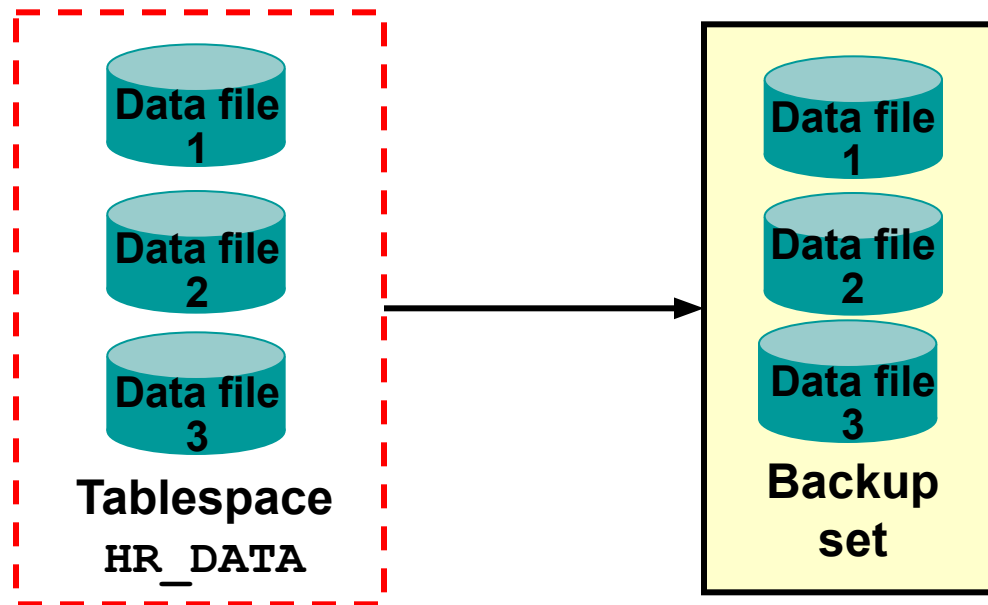
```
  (DATAFILE 1,4,5 CHANNEL c1)
```

```
  (DATAFILE 2,3,6 CHANNEL c2) ;
```

```
  }
```

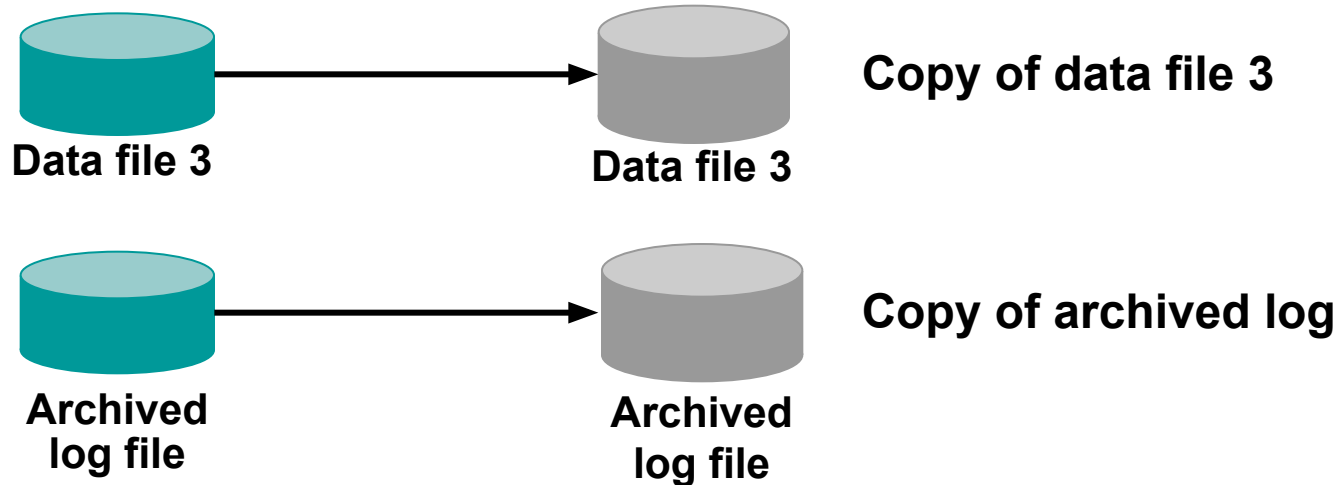
# Creating Backup Sets

```
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%';
```



# Backup archived redo log files

```
RMAN> BACKUP FORMAT 'C: \disk1\backup\ar_%t_%s_%p'  
ARCHIVELOG FROM SEQUENCE=234  
DELETE INPUT;
```

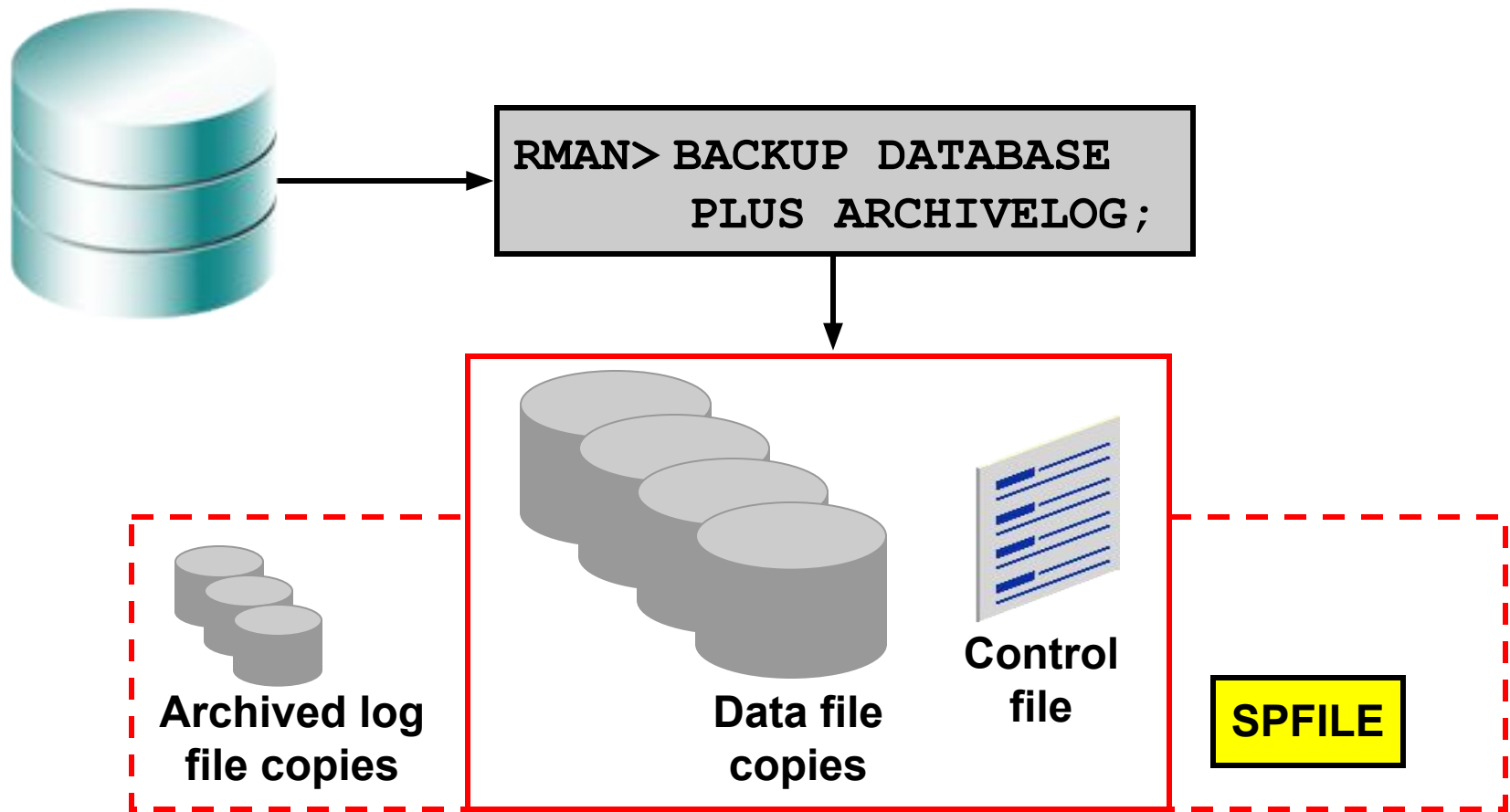
Where:

- %t is thread number
- %s is the sequence number
- %p is the piece number

```
RMAN> BACKUP ARCHIVELOG ALL;  
RMAN> BACKUP ARCHIVELOG  
FROM SEQUENCE 121  
UNTIL SEQUENCE 125;
```

```
RMAN> BACKUP ARCHIVELOG  
FROM TIME 'SYSDATE-30'  
UNTIL TIME 'SYSDATE-7';
```

# Creating a Whole Database Backup





# Backup options

- Check for physical block corruptions
- Scan for logical corruptions in addition to physical corruptions
- Set a threshold on the number of detected corruptions allowed before aborting
- Validate the target input files before performing a backup operation
- Duplex the backup set
- Overwrite an existing backup set or image copy