

## Practice 7

# Reporting and Monitoring RMAN Backups and Jobs

### Practice Target

In this practice, you will examine the RMAN reporting commands and the V\$ views that can be used to obtain information about RMAN backups and monitor RMAN operations.

### Practice Overview

In this practice, you will perform the following tasks:

- Use `LIST` command to retrieve list of the following:
  - backupset file and image copies registered in RMAN repository
  - archived redo log files and their backups
  - expired backup files
- Use `CROSSCHECK` to scan the backup files and check on their expiry status
- Use `REPORT` command to perform the following:
  - display the physical structure of the database
  - report datafiles that do not satisfy retention policy
  - report datafiles that are affected by unrecoverable operations
  - report obsolete backup files
- Monitor RMAN jobs using `v$` views

### Assumptions

This practice assumes the `srv1` appliance is up and running and its database `ORADB` is running in `OPEN` state.

### Note

I recommend taking a snapshot of the appliances before you start implementing the practice.

## A. Preparing for the Practice

In the following steps, you will produce some backup files to prepare for this practice.

1. Open Putty and login to `srv1` as `oracle`. In the rest of this practice, this Putty window will be referred to as **RMAN window**. You will use it to issue RMAN commands.
2. Invoke RMAN and connect to the local database as target.

```
rman target /
```

3. Perform the following RMAN tasks.

```
# Set the retention policy to redundancy 2
CONFIGURE RETENTION POLICY TO REDUNDANCY 2;

# Take backup of the database as backupset
BACKUP AS BACKUPSET DATABASE TAG 'DB_FULL_BS';

# Take backup of users tablespace twice
BACKUP TABLESPACE users TAG 'USERS_BS';

# Take backup of the database as copy in the shared folder
BACKUP AS COPY DATABASE FORMAT '/media/sf_extdisk/%U' TAG 'DB_FULL_CP';

# Take backup of the archived redo log files as backupset. Do not delete them.
BACKUP ARCHIVELOG ALL TAG 'ARC_BS';

# Take backup of the archived redo log files as backupset and as copy in the shared folder
BACKUP AS BACKUPSET ARCHIVELOG ALL FORMAT '/media/sf_extdisk/%U' TAG 'ARC_BS';
BACKUP AS COPY ARCHIVELOG ALL FORMAT '/media/sf_extdisk/%U';
```

4. Open another Putty session and login to `srv1` as `oracle`. In the rest of this practice, this Putty window will be referred to as the **SQL window**. You will use it to issue SQL commands.
5. Start SQL\*Plus and login as sysdba to `ORADB`

```
sqlplus / as sysdba
```

## B. Obtain Information about RMAN Backups using LIST Command

In the following steps you will use `LIST` command to obtain information about RMAN backups.

### Listing RMAN backup files.

In the following steps, you will use the `LIST` command to list RMAN backup files.

- Issue the following `LIST` commands.

```
# The following commands display the backupsets but not the image copies
LIST BACKUP;
LIST BACKUP SUMMARY;

# The following command displays the image copies, but not the backupsets.
LIST COPY;

# To display the backupsets that contain backup of users tablespace:
LIST BACKUP OF TABLESPACE USERS;

# To display the image copies of users tablespace:
LIST COPY OF TABLESPACE USERS;

# To display backupsets of specific TAG:
LIST BACKUPSET TAG 'ARC_BS';
```

- In the SQL window, issue the following query. It retrieves information about the backupsets registered in the RMAN repository.

```
set linesize 180
col backup_type for a4 heading "TYPE"
col incremental_level heading "INCR LEVEL"
col pieces for 99 heading "PCS"
col elapsed_seconds heading "SECONDS"
col output_mbytes for 999,999 heading "MBYTES"

SELECT
  D.BS_KEY, D.BACKUP_TYPE, D.INCREMENTAL_LEVEL, D.PIECES,
  TO_CHAR(D.START_TIME, 'DD-MM-RR HH24:MI:SS') START_TIME,
  TO_CHAR(D.COMPLETION_TIME, 'DD-MM-RR HH24:MI:SS') COMPLETION_TIME,
  ROUND(D.ELAPSED_SECONDS,3) ELAPSED_SECONDS, (D.OUTPUT_BYTES/1024/1024) OUTPUT_MBYTES
FROM V$BACKUP_SET_DETAILS D
  JOIN V$BACKUP_SET S ON S.SET_STAMP = D.SET_STAMP AND S.SET_COUNT = D.SET_COUNT
ORDER BY D.BS_KEY;
```

- In the SQL windows, issue the following query. It retrieves information about the datafile copies registered in the RMAN repository.

```
col TAG format a18
SELECT FILE#, STATUS, CREATION_TIME, COMPLETION_TIME,
TAG, BLOCKS*BLOCK_SIZE/1024/1024 SIZE_MB, INCREMENTAL_LEVEL
FROM V$DATAFILE_COPY
WHERE DELETED = 'NO'
ORDER BY FILE#, RECID;
```

## Listing Archivelog files

In the following steps, you will use the `LIST` command to list the archivelog files and their backups.

9. Issue the following `LIST` commands against the archive redo logs:

```
# display all the archive redo logs
LIST ARCHIVELOG ALL;

# display archivelogs generated in the last 24 hours
LIST ARCHIVELOG TIME BETWEEN 'SYSDATE-1' AND 'SYSDATE';

# you can use the following format:
LIST ARCHIVELOG SEQUENCE BETWEEN <n> AND <m>;

# display backups of archivelogs:
LIST BACKUP OF ARCHIVELOG ALL;

# display backup of archivelogs whose transactions time is within the last 24 hours
LIST BACKUP OF ARCHIVELOG TIME BETWEEN 'SYSDATE-1' AND 'SYSDATE';

# archivelog files backed up at least twice
LIST ARCHIVELOG ALL BACKED UP 2 TIMES TO DEVICE TYPE disk;
```

## Managing expired backup files

In the following steps, you will use the `LIST` command to obtain a report of the expired backup files. Expired backup files are backup files that are marked available in RMAN repository but they actually are not, probably because they have been deleted or moved to some other location.

10. Confirm that there is no expired backup file at this stage.

```
LIST EXPIRED BACKUPSET;
```

11. List the backupsets of tablespace users. Take a note of the backup piece in the last backupset in the displayed list.

```
LIST BACKUP OF TABLESPACE USERS;
```

12. Rename the chosen backup piece.

```
host 'mv <current file name> <renamed file name>';
```

13. List the expired backupsets.

No backup file has been reported by expired, although one backup piece is not actually available.

```
LIST EXPIRED BACKUPSET;
```

14. Issue the following command.

```
CROSSCHECK BACKUPSET;
```

15. List the expired backupsets.

The renamed backup piece is retrieved by the command.

```
LIST EXPIRED BACKUPSET;
```

16. Rename the backup piece to its original name.

**Note:** If in a real life scenario, the expired file cannot be made available again, you can remove it from RMAN repository using the command `DELETE EXPIRED`.

17. Crosscheck the backupset files and make sure the backup piece is **not** reported as expired any more.

```
CROSSCHECK BACKUPSET;
```

```
LIST EXPIRED BACKUPSET;
```

## C. Obtain Information about RMAN Backups using REPORT Command

In this section of the practice you will use `REPORT` command to obtain information about RMAN backups. Sometimes you will take actions on the backups reported by the command.

### Report the database datafiles

In the following steps, you will use the `REPORT` command to obtain information about the datafiles in the target database. This includes the datafile names and numbers, the tablespaces they are assigned to, the size of the datafiles, and whether the datafiles contain rollback segments.

**18.** Issue the following command.

The `SCHEMA` in this command means the database physical structure, not the database users.

```
REPORT SCHEMA;
```

### Report datafiles that do not satisfy retention policy

In the following steps you will use `REPORT` command to know the datafiles that should be backed up because they do not satisfy the retention policy.

**19.** Create a testing tablespace.

`SQL` is a command that you can use to execute SQL commands from RMAN command prompt. That was needed in the releases before 12c. Now, you can issue SQL statements directly from within RMAN command prompt.

This freshly created tablespace has not been involved in any backup before.

```
SQL 'CREATE TABLESPACE TEST_TBS';
```

**20.** Report the datafiles that should be backed up now because they do not satisfy the retention policy.

Observe that the command returns the datafile of `TEST_TBS` tablespace.

```
REPORT NEED BACKUP;
```

**21.** Take backup of `TEST_TBS` tablespace.

```
BACKUP TABLESPACE TEST_TBS;
```

**22.** Issue the same `REPORT` command again.

Observe that the command again returns the datafile of `TEST_TBS` tablespace. Having a single backup of the datafile does not satisfy the retention policy.

```
REPORT NEED BACKUP;
```

**23.** Take another backup of `TEST_TBS` tablespace and issue the same `REPORT` command again.

Now the datafile is not retrieved by the `REPORT` command.

```
BACKUP TABLESPACE TEST_TBS;  
REPORT NEED BACKUP;
```

- 24.** Check if tablespace `users` should be backed up now to satisfy the default retention policy.

```
REPORT NEED BACKUP TABLESPACE users;
```

- 25.** Check if tablespace `users` should be backed up if the default retention policy is of redundancy 3.

```
REPORT NEED BACKUP REDUNDANCY 3 TABLESPACE users;
```

- How can you know how many backups are registered in RMAN repository to tablespace `users`? Remember, backups could be as backupsets or as image copy.

## Clean up

- 26.** Drop the tablespace `TEST_TBS` with its datafile.

```
SQL 'DROP TABLESPACE TEST_TBS INCLUDING CONTENTS AND DATAFILES';
```

## Report datafiles that are affected by unrecoverable operations

In the following steps, you will use the `REPORT` command to know the datafiles that contain data from unrecoverable operations. Normally, you should take backup of those datafiles, if you want to protect that data.

- 27.** Open another Putty session and connect to `srv1` as `oracle`

- 28.** Invoke SQL\*Plus and login as `sysdba`

```
sqlplus / as sysdba
```

- 29.** Issue the following command to execute an unrecoverable operation.

```
CREATE TABLE testme NOLOGGING TABLESPACE USERS AS SELECT * FROM USER_TABLES ;
```

- 30.** Exit from the SQL\*Plus Putty session.

- 31.** In RMAN session, issue the following command.

The command should report that the tablespace `users` is unrecoverable. In such cases, you should take backup of the tablespace.

```
REPORT UNRECOVERABLE;
```

- 32.** Take backup of the tablespace and issue the same `REPORT` command again.

The `REPORT` command should not report any datafile.

```
BACKUP TABLESPACE users TAG 'USERS_BS';  
REPORT UNRECOVERABLE;
```

### Report obsolete backup files

In the following steps, you will use the `REPORT` command to obtain a report of the obsolete backup files. Obsolete backup files are backup files that are not needed any more to satisfy the retention policy. Normally, you delete those files.

- 33.** Issue the following command.

Observe that the command first displays the current active retention policy.

```
REPORT OBSOLETE;
```

- 34.** Issue the following command to delete the obsolete backup files. This command is commonly used in routine RMAN backup script.

```
DELETE OBSOLETE;
```

### Clean up

- 35.** Delete all the backup files.

```
DELETE BACKUPSET;
```

```
DELETE COPY;
```

Ahmed Baraka  
Oracle Database Administrator



## D. Monitoring RMAN Jobs

In this section of the practice, you will issue RMAN jobs and monitor their activities.

### Monitor all RMAN jobs

In the following steps, you will start a backup job and monitor its operation.

- 36.** In the RMAN window, execute the following RMAN job. Do **not** wait for its execution to finish. Go to the next step straight away after starting the job.

If the `run` block finishes before you execute the queries, delete the backup sets tagged as 'DB\_FULL' and try again.

```
run{
  ALLOCATE CHANNEL C1 DEVICE TYPE DISK;
  ALLOCATE CHANNEL C2 DEVICE TYPE DISK;
  BACKUP DATABASE TAG 'DB_FULL';
}
```

- 37.** In the SQL window, issue the following query.

Observe that RMAN job is represented by two sessions. Each session represents an RMAN channel.

```
SELECT S.SID, P.SPID, S.SID, S.SERIAL#, S.CLIENT_INFO
FROM   V$PROCESS P, V$SESSION S WHERE P.ADDR = S.PADDR
AND    CLIENT_INFO LIKE 'rman%';
```

- 38.** In the SQL window, issue the following query.

This query displays the progress of the RMAN job.

```
SELECT SID, SERIAL#, CONTEXT, SOFAR, TOTALWORK,
       ROUND(SOFAR/TOTALWORK*100,2) "%_COMPLETE"
FROM   V$SESSION_LONGOPS
WHERE  OPNAME LIKE 'RMAN%'
AND    OPNAME NOT LIKE '%aggregate%'
AND    TOTALWORK != 0
AND    SOFAR <> TOTALWORK;
```

### Clean up

- 39.** Wait for the backup to finish.  
**40.** Delete the backupset files produced.

```
DELETE BACKUPSET TAG 'DB_FULL';
```

## Using V\$RMAN\_BACKUP\_JOB\_DETAILS

V\$RMAN\_BACKUP\_JOB\_DETAILS provides information about RMAN backup jobs that cannot be found somewhere else. However, to understand the meaning of its figures, you need to understand what RMAN job means to this view perspective.

### Understanding how V\$RMAN\_BACKUP\_JOB\_DETAILS represents its data

41. In the SQL window, issue the following query:

```
col S_KEY format a4
col STATUS format a11
SELECT TO_CHAR(SESSION_KEY) S_KEY, INPUT_TYPE, STATUS,
       TO_CHAR(START_TIME, 'dd/mm/yy hh24:mi') START_TIME,
       TO_CHAR(END_TIME, 'dd/mm/yy hh24:mi') END_TIME,
       ELAPSED_SECONDS/60 MINS
FROM V$RMAN_BACKUP_JOB_DETAILS
WHERE START_TIME > SYSDATE-1 -- this condition is just to reduce the output
ORDER BY SESSION_KEY;
```

42. In the RMAN window, exit from RMAN, start RMAN, and login again to ORADB as target.

43. In the SQL window, run the query again.

You should not observe any changes in the output.

44. In the RMAN window, issue the following BACKUP command.

```
BACKUP TABLESPACE users TAG 'USERS_TBS';
```

45. In the SQL window, run the query again. You should see an additional row is displayed in the list. Observe its data.

46. Wait for a couple of minutes then in RMAN window, issue the same BACKUP command again.

```
BACKUP TABLESPACE users TAG 'USERS_TBS';
```

47. In SQL window, run the query again.

You will observe that no additional row is added and the last retrieved row is updated. This means a backup job in this view is not a backup command. It is a group of backup commands of the same type (full or incremental) executed in a single RMAN session.

Observe the time (ELAPSED\_SECONDS) is updated but it does not reflect the time taken by executing the backup command. It is the time between the start time of the first backup command and the time at which the last backup command finished execution in RMAN session.

48. In the RMAN window, issue the following BACKUP command.

```
BACKUP INCREMENTAL LEVEL 0 DATABASE TAG 'DBLVL0';
```

49. In the SQL window, run the query again. You should see either the INPUT\_TYPE of the last row changed or an additional row is displayed in the list.

In conclusion, the figures in the view V\$RMAN\_BACKUP\_JOB\_DETAILS for backup jobs are accumulated within an RMAN session. The idle time between backup commands, especially if they are of the same type, is included in the ELAPSED\_SECONDS.

## Clean up

- 50. Wait for the backup to finish.
- 51. Delete the backupset files produced.

```
DELETE BACKUPSET TAG 'DBLVL0';  
DELETE BACKUPSET TAG 'USERS_TBS';
```

## Using V\$RMAN\_BACKUP\_JOB\_DETAILS view

V\$RMAN\_BACKUP\_JOB\_DETAILS view can be used in different ways to obtain a lot of valuable information about RMAN backup jobs.

- 52. In the RMAN window, issue the following BACKUP command. Do not wait for it to finish, go to the next step.

```
BACKUP database tag 'DB_FULL';
```

- 53. In the SQL window, issue the following query to view the speed of RMAN backup job.

```
col SESSION_KEY for 999  
col COMP for 99.9999  
col IN_SPEED for a10  
col OUT_SPEED for a10  
col TIME_TAKEN_DISPLAY for a15  
  
SELECT SESSION_KEY, OPTIMIZED, round(COMPRESSION_RATIO,4) COMP,  
       INPUT_BYTES_PER_SEC_DISPLAY IN_SPEED,  
       OUTPUT_BYTES_PER_SEC_DISPLAY OUT_SPEED,  
       TIME_TAKEN_DISPLAY  
FROM   V$RMAN_BACKUP_JOB_DETAILS  
WHERE  STATUS='RUNNING'  
ORDER BY SESSION_KEY;
```

- 54. In the SQL window, issue the following query to view the size of RMAN backup job.

```
col SESSION_KEY for 999  
col IN_SIZE for a15  
col OUT_SIZE for a15  
  
SELECT SESSION_KEY, INPUT_TYPE, ROUND(COMPRESSION_RATIO,5) COMP_R,  
       INPUT_BYTES_DISPLAY IN_SIZE,  
       OUTPUT_BYTES_DISPLAY OUT_SIZE  
FROM   V$RMAN_BACKUP_JOB_DETAILS  
WHERE  STATUS='RUNNING'  
ORDER BY SESSION_KEY;
```

## Clean up

- 55. Wait for the backup to finish.
- 56. Delete all the backup files.  

```
DELETE BACKUPSET;  
DELETE COPY;
```
- 57. Consider deleting the appliance snapshot, if you created one in the start of the practice.

## Summary

In this practice, you performed the following tasks:

- Use `LIST` command to retrieve list of the following:
  - backupset file and image copies registered in RMAN repository
  - archived redo log files and their backups
  - expired backup files
- Use `CROSSCHECK` to scan the backup files and check on their expiry status
- Use `REPORT` command to perform the following:
  - display the physical structure of the database
  - report datafiles that do not satisfy retention policy
  - report datafiles that are affected by unrecoverable operations
  - report obsolete backup files
- Monitor RMAN jobs using `v$` views

Ahmed Baraka  
Oracle Database Administrator