Mastering RMAN Tasks

Using SQL , PL/SQL, RMAN and Bash Scripts

Asfaw Gedamu

Mastering RMAN: Essential Backup, Recovery, and Monitoring Techniques

Introduction

Oracle Recovery Manager (RMAN) is a powerful tool for backup and recovery in Oracle databases, offering a range of features to ensure data protection and continuity. This document outlines the procedures for performing full database backups, incremental database backups with block recovery, and tablespace backups using Oracle Recovery Manager (RMAN). These backups are crucial for ensuring data recovery in case of system failures or data corruption.

Prerequisites

- **RMAN configuration:** Ensure RMAN is configured with target database connection details, backup locations, and retention policies.
- **Database access:** The script executor requires DBA privileges to connect to the RMAN target and catalog databases.
- **Sufficient disk space:** Verify enough disk space is available for the backup files based on estimated database size.

Ideal Audience

- Database Administrators (DBAs) responsible for Oracle database backups.
- System Administrators supporting database environments.

1. RMAN Full Database Backup Script

PL/SQL Script:

```
DECLARE
  v_backup_set_name VARCHAR2(100);
BEGIN
  -- Set backup set name with timestamp
  v_backup_set_name := 'FULL_DB_BACKUP_' || TO_CHAR(SYSDATE,
'YYYYMMDD_HH24MISS');

  -- Allocate channels for parallel backup (adjust channel count
as needed)
```

```
  FOR i IN 1..4 LOOP
    EXECUTE IMMEDIATE 'ALLOCATE CHANNEL CH'||i||' DEVICE TYPE
DISK';
  END LOOP;

  -- Perform full database backup with compression (optional)
  RUN {
    SET UNTIL TIME 'SYSDATE + 1' COMPRESS BACKUPSET;
    BACKUP AS '+v_backup_set_name+' DATABASE;
  };

  -- Release allocated channels
  FOR i IN 1..4 LOOP
    EXECUTE IMMEDIATE 'RELEASE CHANNEL CH'||i;
  END LOOP;

  DBMS_OUTPUT.PUT_LINE('Full database backup completed
successfully.');
END;
```

/

Bash Script (call the above PL/SQL script):

```
sqlplus /nolog @full_db_backup.sql
```

RMAN Script

```
configure backup optimization on;
configure controlfile autobackup on;
configure controlfile autobackup format for device type disk to
'/archiva/backup/%F';
configure maxsetsize to unlimited;
configure device type disk parallelism 4;
run
{
allocate channel c1 type disk format '/archiva/backup/%I-%Y%M%D-
%U' maxpiecesize 3G;
allocate channel c2 type disk format '/archiva/backup/%I-%Y%M%D-
%U' maxpiecesize 3G;
allocate channel c3 type disk format '/archiva/backup/%I-%Y%M%D-
%U' maxpiecesize 3G;
allocate channel c4 type disk format '/archiva/backup/%I-%Y%M%D-
%U' maxpiecesize 3G;
backup as compressed backupset incremental level 0 check logical
database plus archivelog;
release channel c1 ;
release channel c2 ;
release channel c3 ;
release channel c4 ;
}
```

Explanation:

This RMAN script has two parts:

1. Configuration Section:

- configure backup optimization on;: Enables RMAN backup optimization, which skips backing up unchanged datafiles.

- configure controlfile autobackup on;: Enables automatic backups of the control file.
- configure controlfile autobackup format for device type disk to '/archiva/backup/%F';: Sets the format and location for automatic control file backups.
- configure maxsetsize to unlimited;: Removes the limitation on the size of backup sets.
- configure device type disk parallelism 4;: Enables parallel processing with 4 channels for disk backups.

2. Backup Execution Section:

- run { ... }: This block defines the actual backup operation.
  - allocate channel c1 ... c4 type disk ...;: Allocates 4 channels (c1, c2, c3, c4) for parallel disk backups.
  - backup as compressed backupset incremental level 0 ...;: Defines the backup type (compressed, incremental level 0), checks logical database consistency, and includes archive logs.

- release channel c1 ; ... c4 ;: Releases the allocated channels after the backup completes.

Overall, this script configures RMAN for optimized, parallel, and compressed incremental database backups (level 0 includes all data) along with archive logs.

2. RMAN Incremental Database Backup with Block Recovery

RMAN Script-1:

```
RUN {
  ALLOCATE CHANNEL CH1 DEVICE TYPE DISK;
  BACKUP AS '+INCREMENTAL_' || TO_CHAR(SYSDATE, 
'YYYYMMDD_HH24MISS')
          INCREMENTAL LEVEL 1 DATABASE
          BLOCKRECOVERY ALL;
  RELEASE CHANNEL CH1;
};
```

Explanation:

- This script performs an incremental backup capturing changes since the last full backup.
- `BLOCKRECOVERY ALL` enables block-level recovery for the entire database.

RMAN Script-2:

```
configure backup optimization on;
configure controlfile autobackup on;
configure controlfile autobackup format for device type disk to 
'/archiva/backup/%F';
configure maxsetsize to unlimited;
configure device type disk parallelism 4;
run
{
allocate channel c1 type disk format '/archiva/backup/%I-%Y%M%D-
%U' maxpiecesize 3G;
allocate channel c2 type disk format '/archiva/backup/%I-%Y%M%D-
%U' maxpiecesize 3G;
allocate channel c3 type disk format '/archiva/backup/%I-%Y%M%D-
%U' maxpiecesize 3G;
allocate channel c4 type disk format '/archiva/backup/%I-%Y%M%D-
%U' maxpiecesize 3G;
backup as compressed backupset incremental level 1 check logical
database plus archivelog;
release channel c1 ;
release channel c2 ;
release channel c3 ;
release channel c4 ;
}
```

3. RMAN Tablespace Backup

RMAN Script-1:

```
RUN {
  ALLOCATE CHANNEL CH1 DEVICE TYPE DISK;
  BACKUP TABLESPACE <tablespace_name> FORMAT
'<backup_location>/%U_<tablespace_name>_%d_%T_%s_%p';
  RELEASE CHANNEL CH1;
};
```

Replace:

- `<tablespace_name>` with the specific tablespace to be backed up.
- `<backup_location>` with the desired directory path for storing backups.

RMAN Script-2:

```
configure controlfile autobackup on;
configure controlfile autobackup format for device type disk to
'/archiva/backup/%F';
configure maxsetsize to unlimited;
configure device type disk parallelism 4;
run
{
allocate channel c1 type disk format '/archiva/backup/%I-%Y%M%D-
%U' maxpiecesize 3G;
allocate channel c2 type disk format '/archiva/backup/%I-%Y%M%D-
%U' maxpiecesize 3G;
backup tablespace USERS,TOOLS;
release channel c1 ;
release channel c2 ;
}
```

4. RMAN Datafile(s) Backup Run Block

RMAN Script-1:

```
RUN {
  ALLOCATE CHANNEL CH1 DEVICE TYPE DISK;
  BACKUP DATAFILE <datafile_id>|<datafile_spec> FORMAT
'<backup_location>/%U_<datafile_name>_%d_%T_%s_%p';
  RELEASE CHANNEL CH1;
};
```

Replace:

- <datafile_id> with the specific datafile ID (e.g., 1, 2).
- <datafile_spec> with the datafile location specification (e.g., '/u01/app/oracle/oradata/orcl/datafile/system01.dbf').
- <backup_location> with the desired directory path for storing backups.

Explanation:

This script allows backing up individual datafiles based on ID or file specification. Modify the script to include multiple datafiles or use a loop to iterate through a list.

RMAN Script-2:

```
configure controlfile autobackup on;
configure controlfile autobackup format for device type disk to
'/archiva/backup/%F';
configure maxsetsize to unlimited;
configure device type disk parallelism 4;
run
{
allocate channel c1 type disk format '/archiva/backup/%I-%Y%M%D-
%U' maxpiecesize 3G;
allocate channel c2 type disk format '/archiva/backup/%I-%Y%M%D-
%U' maxpiecesize 3g;
backup datafile 3,4;
release channel c1 ;
release channel c2 ;
}
```

5. Delete Archive Logs Older Than 1 Day

SQL Script:

```
BEGIN
  DBMS_SCHEDULER.DROP_JOB(job_name => 'RMAN_DELETE_ARCHIVELOG');
  DBMS_SCHEDULER.CREATE_JOB(
    job_name => 'RMAN_DELETE_ARCHIVELOG',
    job_type => 'SCHEDULED',
    schedule_type => 'INTERVAL',
    interval_expr => 'FREQ=DAILY',
    enabled => TRUE,
    program_name => 'dbms_scheduler.run_sql(statement =>
''DELETE FROM ARC$ WHERE CREATION_DATE < SYSDATE - 1'')'
  );
END;
/
```

Explanation:

This script creates a scheduled job named "RMAN_DELETE_ARCHIVELOG" that runs daily (modify `INTERVAL_EXPR` for different schedules) and deletes archive logs older than 1 day using `DBMS_SCHEDULER`.

RMAN Script-2:

```
DELETE ARCHIVELOG ALL COMPLETED BEFORE 'sysdate-1';
CROSSCHECK ARCHIVELOG ALL;
DELETE EXPIRED ARCHIVELOG ALL;
```

6. Backup Archive Logs Using RMAN

RMAN Script-1:

```
RUN {
  ALLOCATE CHANNEL CH1 DEVICE TYPE DISK BACKUP RETENTION POLICY
TO RECOVERY WINDOW OF 1 DAYS;
  BACKUP ARCHIVELOG ALL FORMAT
'<backup_location>/%U_arch_%T_%s_%p';
```

```
  RELEASE CHANNEL CH1;
};
```

Replace:

- `<backup_location>` with the desired directory path for storing backups.

Explanation:

This script backs up all archive logs with a retention policy of 1 day using `RECOVERY WINDOW`. Adjust the retention period as needed.

RMAN Script-2:

```
--Backup all archivelogs known to controlfile
backup archivelog all;
--Backup all archivelogs known to controlfile and delete them
once backed up
backup archivelog all delete input ;
--Backup archivlogs known to controlfile and the logs which
haven't backed up once also
backup archivelog all not backed up 1 times;
```

Bash Script (call the above SQL and RMAN scripts):

```
sqlplus /nolog @create_arch_delete_job.sql
rman target /nolog @backup_archivelogs.rman
```

7. Copy Archive from ASM to Mount Point

RMAN Script-1:

```
RUN {
  ALLOCATE CHANNEL CH1 DEVICE TYPE DISK;
  COPY ARCHIVE FROM '/u02/asm/oracle/archivelog/<archive_name>'
TO '/path/to/mount/point/<archive_name>';
  RELEASE CHANNEL CH1;
};
```

Replace:

- `<archive_name>` with the specific archive log filename (e.g., arch_2024_03_26.arc).
- `/u02/asm/oracle/archivelog/` with the ASM archive log location on your system.
- `/path/to/mount/point/` with the desired mount point directory for storing the copied archive.

Explanation:

This script copies a specific archive log from ASM to a designated mount point. Modify the script with a loop to copy multiple archives or use wildcards for filenames.

RMAN Script-2:

```
--- Copy archive log from ASM to regular mount point using RMAN:
--- Connect to RMAN in RAC db
RMAN> copy archivelog '+B2BSTARC/thread_2_seq_34.933' to
'/data/thread_2_seq_34.933';
```

8. Backup Archive Between 2 Sequence Numbers

RMAN Script-1:

```
RUN {
  ALLOCATE CHANNEL CH1 DEVICE TYPE DISK;
  BACKUP ARCHIVELOG FROM SEQUENCE <start_seq_num> TO SEQUENCE
<end_seq_num> FORMAT '<backup_location>/%U_arch_%T_%s_%p';
  RELEASE CHANNEL CH1;
};
```

Replace:

- `<start_seq_num>` with the starting sequence number of the archive logs to be backed up.
- `<end_seq_num>` with the ending sequence number of the archive logs to be backed up.
- `<backup_location>` with the desired directory path for storing backups.

Explanation:

This script enables selective backup of archive logs based on their sequence number range.

RMAN Script-2:

```
--For taking backup of archivelog between seq number 1000 to
1050
RMAN> backup format '/archive/%d_%s_%p_%c_%t.arc.bkp'
archivelog from sequence 1000 until sequence 1050;
--For RAC ,need to mention the thread number also
RMAN> backup format '/archive/%d_%s_%p_%c_%t.arc.bkp'
archivelog from sequence 1000 until sequence 1050 thread 2;
```

9. Enable Trace for RMAN

RMAN Script:

```
-- To diagnose rman script, use trace as below.
spool trace to '/tmp/rman_trace.out';
report schema;
list backup summary;
list backup of datafile 1;
list copy of datafile 1;
spool trace off;
```

Bash Script:

```
rman target /nolog << EOF
  SET TRACE ON;
  -- Your RMAN commands here
  SET TRACE OFF;
EOF
```

Explanation:

This script enables RMAN tracing within a Bash script. Include your desired RMAN commands between `SET TRACE ON` and `SET TRACE OFF` for detailed logging of RMAN operations.

10. Recover Dropped Table with RMAN 12c

Oracle 12c introduced the `RECOVER TABLE` command that allows recovering dropped tables using RMAN. However, you cannot directly use a script for this. Here's the process:

1. Identify the relevant backup containing the table before it was dropped.
2. Use the `RECOVER DATABASE` command with the `UNTIL TIME` clause specifying a point in time before the drop. This recovers the entire database to an auxiliary instance.
3. Export the recovered table data from the auxiliary instance and import it back into the primary database.
4. Clean up the auxiliary instance.

Bash Script (Template):

```
rman target /nolog << EOF
  RUN {
    ALLOCATE CHANNEL CH1 DEVICE TYPE DISK;
    RECOVER DATABASE UNTIL TIME
'TO_DATE('2024_03_25_DROP_TIME','YYYY_MM_DD_HH24_MI_SS')'
AUXILIARY DESTINATION '/aux_data';
    RELEASE CHANNEL CH1;
  };
EOF
```

Replace:

- `2024_03_25_DROP_TIME` with the estimated time the table was dropped (adjust format as needed).

Additional Steps:

- Export and import the recovered table data manually using tools like `expdp` and `impdp`.
- Clean up the auxiliary instance using `RMAN DROP AUXILIARY DATABASE`.

RMAN Script:

```
RMAN>recover table SCOTT.SALGRADE until time
“to_date(’08/09/2016 18:49:40’,’mm/dd/yyyy hh24:mi:ss’)”
auxiliary destination ‘/u03/arch/TEST/BACKUP’
datapump destination ‘/u03/arch/TEST/BACKUP';
```

```
Auxiliary destination – Location where all the related files for auxiliary
instance will be placed

Datapump destination – Location where the export dump of the table will be
placed

NOTE - This feature is available only in oracle 12c and later.
```

11. Monitor RMAN Backup Progress

There are several ways to monitor RMAN backup progress:

- **RMAN Output:** The RMAN command prompt displays progress information during backup execution.
- **v$session_longops view:** This view provides detailed information on running database operations, including RMAN backups. You can query this view in SQL*Plus.
- **Alerting:** Configure RMAN to send email or pager notifications upon backup completion or errors.

Bash Script (Example using v$session_longops):

```
sqlplus -s /nolog << EOF
  SET HEADING OFF
  SELECT sid, username, operation, progress, SOFAR/TOTALWORK *
100 AS PERCENT_COMPLETE
  FROM v\$session_longops
  WHERE operation LIKE '%RMAN%'
  ORDER BY sid;
EOF
```

SQL Script:

```
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;
```

12. Restore Archive Log from RMAN Tape

RMAN can directly restore archive logs from tape:

RMAN Script-1:

```
RUN {
  ALLOCATE CHANNEL CH1 DEVICE TYPE TAPE NOREWIND;
  RESTORE ARCHIVELOG FROM '/path/to/tape/device' LIKE 'arch_%';
-- Adjust LIKE pattern for specific archive names
```

```
  RELEASE CHANNEL CH1;
};
```

Replace:

- `/path/to/tape/device` with the specific tape device name for your system.

RMAN Script-2:

```
connect target sys/******@CRM_DB
connect catalog RMAN_tst/*****@catdb
run
{
allocate channel t1 type SBT_TAPE parms
'ENV=(NSR_SERVER=nwwerpw,NSR_CLIENT=tsc_test01,NSR_DATA_VOLUME_P
OOL=DD086A1)'connect sys/****@CRM_DB;
set archivelog destination to '/dumparea/';
restore archivelog from sequence 7630 until sequence 7640;
release channel t1;
}
```

```
Note: This script will restore the archive sequences from 7630 to 7640 to
/dumparea location
```

13. Check the Syntax of RMAN Commands

Here are ways to check RMAN command syntax:

- **RMAN Command Prompt:** Enter the command at the RMAN prompt. If there are syntax errors, RMAN will display an error message.
- **SQL*Plus Script (Limited):** Wrap the RMAN command within a BEGIN and END; block in a SQL*Plus script. However, this only validates basic syntax and may not catch all errors.

Bash Script (Limited):

```
rman target /nolog << EOF
  -- Your RMAN command here
EOF
```

Important Note: This Bash script executes the command but won't necessarily highlight syntax errors. It's for testing basic functionality.

RMAN Script:

```
--check the syntax of RMAN commands interactively without actually
executing the commands
```

```
$ rman checksyntax
Recovery Manager: Release 12.1.0.2.0 - Production on Sun Jan 29
12:04:24 2017
Copyright (c) 1982, 2014, Oracle and/or its affiliates. All
rights reserved.
```

```
--Now put the command for checking syntax
```

```
RMAN> backup database;

The command has no syntax errors
```

Conclusion

Regular RMAN backups are essential for a robust database disaster recovery strategy. Implementing these scripts ensures consistent and reliable database backups, minimizing downtime and data loss in the event of unexpected outages. By following these procedures and customizing scripts as needed, DBAs can effectively safeguard valuable database information.

Note: These scripts are basic examples. Remember to adjust them based on your specific environment, backup requirements, and retention policies. Hence:

- Customize these scripts based on your environment and retention policies.
- Schedule RMAN backups and archive log deletion for automated execution.

- Refer to Oracle documentation for detailed information on RMAN commands and options.
- RMAN trace files can become large. Enable tracing only when needed for troubleshooting purposes and disable it afterward.