DB Refresh-

Cold backup and Refresh

RECOVERY MANAGER (RMAN)

1. Introduction

- **RMAN** (Recovery Manager) is a backup and recovery manager supplied for Oracle databases (from version 8) created by the Oracle Corporation. It provides database backup, restore, and recovery capabilities addressing high availability and disaster recovery concerns. Oracle Corporation recommends RMAN as its preferred method for backup and recovery and has written command-line and graphical (via Oracle Enterprise Manager) interfaces for the product.
- RMAN is the backup methodology introduced in 8i which performs block level backup.

  Block level backup: Backup of only used blocks.

  RMAN is advanced recovery manager tool to take backup without shutting down the database.

  RMAN is an online backup tool and platform independent tool.

  The RMAN metadata backup operation information is stored in control file or metadata repository schema.
- **Advantages of RMAN:**
  a) Block level backup
  b) Parallelism
  c) Duplexing of archives
  d) Detection of corruption blocks
  e) Validating backup
  f) Incremental backup
  g) Recovery catalog etc.

- The RMAN environment consists of the utilities and databases that play a role in backing up your data. At a minimum, the environment for RMAN must include the following components:

- **Target database:**

  An Oracle database to which RMAN is connected with the TARGET keyword. A target database is a database on which RMAN is performing backup and recovery operations. RMAN always maintains metadata about its operations on a database in the control file of the database. The RMAN metadata is known as the **RMAN repository**.

- **RMAN client:**

  An Oracle Database executable that interprets commands, directs server sessions to execute those commands, and records its activity in the target database control file. The RMAN executable is automatically installed with the database and is typically located in the same directory as the other database executables. For example, the RMAN client on Linux is located in $ORACLE_HOME/bin.

Some environments use the following optional components:

- **Flash recovery area:**

  A disk location in which the database can store and manage files related to backup and recovery. You set the flash recovery area location and size with the DB_RECOVERY_FILE_DEST and DB_RECOVERY_FILE_DEST_SIZE initialization parameters.

  To enable the flash recovery area, you must set the two initialization parameters DB_RECOVERY_FILE_DEST_SIZE (which specifies the disk quota, or maximum space to use for flash recovery area files for this database) and DB_RECOVERY_FILE_DEST (which specifies the location of the flash recovery area).

  example -

```
SQL> show parameter db_recovery

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
```

db_recovery_file_dest string /opt/oracle/test01/dbs/arch
db_recovery_file_dest_size big integer 20G

- **Auxiliary database:**

  Auxiliary database is a cloned copy of target database.

- **Recovery catalog:**

  A separate database schema used to record RMAN activity against one or more target databases. A recovery catalog preserves RMAN repository metadata if the control file is lost, making it much easier to restore and recover following the loss of the control file. The database may overwrite older records in the control file, but RMAN maintains records forever in the catalog unless deleted by the user.

- **Media management software** required for RMAN to interface with backup devices such as tape drives.
- **Archive log mode should be enabled to take RMAN backup.**

2. RMAN configuration Parameters:

RMAN is in built with oracle software from Oracle Corporation. To configure RMAN in our database we need to configure some parameters in our database here are the details of RMAN configuration parameters.

a) **RETENSION POLICY:**

It tells that up to how many days our backup will be stored for using recovery. Again the RETENTION POLICYcan be based on 2 values.

i) Redundancy--- it will tell how many backups to be stored

ii) Recovery Window--- it will tell how many days backup to be retained

>CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;

Default value of RETENTION POLICY is set to be 7 days if we want to change the RETENTION POLICY we can change it.

b) BACKUP OPTIMIZATION:

Rman having advantage of skipping backup of unmodified datafiles from last backup.

To avoid the backup of unmodified datafiles just we need to set BACKUP OPTIMIZATION parameter on.

>CONFIGURE BACKUP OPTIMIZATION ON;

Above Parameter is BACKUP OPTIMIZATION parameter to Avoid backing up off unmodified datafiles in database from last backup.

c) DEVICE TYPE:

This parameter shows the device type to take backup of database.

Here default device type is DISK, it can be changed based on our requirement and based on environment. It also having a parameter format of the backup and location of backup.

CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default

CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '/app12/Rman_bkp/db/%d_%T_%s_%p';

d) CONTROLFILE AUTOBACKUP:

Above configuration parameter is used to take the backup of control files and spfile with database backup.

If the CONFIGURATION AUTOBACKUP is ON, in that case only it took the backup of control files and spfile, if it's not on, it don't touch the control files and spfile.

The configuration parameter for this as bellow.

CONFIGURE CONTROLFILE AUTOBACKUP ON;

If auto backup is on it took the backup of control files and spfile at the end of backup.

And it also having one more configuration parameter while taking the backup of contorlfiles. The parameter indicates the format of backup.

CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '/app12/Rman_bkp/db/%F';

By using above parameter we can edit the format and location of backed up controlfiles.

e) PARALLELISM:

Above configuration parameter used to create multiple processes to speed up backup and it also shows the backup type.

By default PARALLELISM is set to 1 if we want speed up our backup we can change the PARALLELISM value based on our requirement.

CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default

CONFIGURE SNAPSHOT CONTROLFILE:

This parameter is a point in time copy of the database controlfile that is taken during RMAN backup operations.

This ensures that the backup is consistent to a given point in time.

CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/db12/oracle12/VIS/db/tech_st/11.1.0/dbs/snapcf_VIS.f'; # default

There are more configuration parameters in RMAN but it's not used in a regular RMAN backup here are the details of those parameters.

CONFIGURE MAXSETSIZE TO UNLIMITED; # default:

It simply indicates the size of backup set if we are mentioned any value like 1 GB then new file creates after 1 GB.

CONFIGURE ENCRYPTION FOR DATABASE OFF; # default:

It simply used for the secure backup.

CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default:

This parameter automatically deletes the Archive logs based on time

CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default:

It gives where to store the backup of datafiles.

CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default:

It gives the where to store the backup of archive logs.

3. Connecting to RMAN:

RMAN is inbuilt with our oracle software and we can directly connect from our OS level.

Before connecting to RMAN we have to check the status of database.

We want to connect to RMAN our database is in at least nomount or mount stage.

COMMAND:

$ rman target /

Here rman is command to connect rman

Target is our existing database

'/' is the sysuser of target database.

Example:

[oravis@ebs12trn ~]$ rman target/

Recovery Manager: Release 11.1.0.7.0 - Production on Wed Mar 26 20:42:21 2014

Copyright (c) 1982, 2007, Oracle. All rights reserved.

connected to target database: VIS (DBID=385036269)

Once connected to the rman we want to see the existing configuration parameters we can use following command from RMAN.

RMAN>show all;

It shows the all existing parameter values in our target database.

Example:

RMAN> show all;

using target database control file instead of recovery catalog

RMAN configuration parameters for database with db_unique_name VIS are:

CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;

CONFIGURE BACKUP OPTIMIZATION ON;

CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default

CONFIGURE CONTROLFILE AUTOBACKUP ON;

CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '/app12/Rman_bkp/db/%F';

CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default

CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default

CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default

CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '/app12/Rman_bkp/db/%d_%T_%s_%p';

CONFIGURE MAXSETSIZE TO UNLIMITED; # default

CONFIGURE ENCRYPTION FOR DATABASE OFF; # default

CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default

CONFIGURE COMPRESSION ALGORITHM 'BZIP2'; # default

CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default

CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/db12/oracle12/VIS/db/tech_st/11.1.0/dbs/snapcf_VIS.f'; # default

4. RMAN Backup Concepts:

Here in this RMAN Backup concepts we are mainly discussing about different types of RMAN backups.

This RMAN Backup concepts describes the general concepts that you need to understand to make any type of RMAN backup.

- Consistent and Inconsistent RMAN Backups
- Online Backups and Backup Mode
- Backup Sets

- Image Copies
- Multiple Copies of RMAN Backups
- Control File and Server Parameter File Auto backups
- Incremental Backups
- Backup Retention Policies

- **Consistent and Inconsistent RMAN Backups:**

The RMAN command for making backups is BACKUP

RMAN>backup

The RMAN BACKUP command supports backing up the following types of files:

- Datafiles and control files
- Server parameter file
- Archived redo logs
- RMAN backups

Although the database depends on other types of files, such as network configuration files, password files, and the contents of the Oracle home, you cannot back up these files with RMAN. Likewise, some features of Oracle, such as external tables, may depend upon files other than the datafiles, control files, and redo log. RMAN cannot back up these files. Use some non-RMAN backup solution for any files not in the preceding list.

When you execute the BACKUP command in RMAN, the output is always either one or more backup sets or one or more image copies.

A **backup set** is an RMAN-specific proprietary format, whereas an **image copy**is a bit-for-bit copy of a file. By default, RMAN creates backup sets.

Incremental Backups

By default, RMAN makes full backups. A full backup of a datafile includes every allocated block in the file being backed up. A full backup of a datafile can be an image copy, in which case every data block is backed up. It can also be stored in a backup set, in which case datafile blocks not in use may be skipped.

A full backup is the default type of RMAN backup. A full backup has no effect on subsequent incremental backups and is not considered a part of an incremental backup strategy. Image copies are always full backups because they include every data block in a datafile. A backup set is by default a full backup because it can potentially include every data block in a datafile, although unused block compression means that blocks never used are excluded and, in some cases, currently unused blocks are excluded (see "Block Compression for Backup Sets").

In contrast to a full backup, an incremental backup copies only those data blocks that have changed since a previous backup. You can use RMAN to create incremental backups of datafiles, tablespaces, or the whole database. A full backup cannot be part of an incremental backup strategy; that is, it cannot be the parent for a subsequent incremental backup.

Multilevel Incremental Backups

RMAN can create multilevel incremental backups. Each incremental level is denoted by a value of 0 or 1. A level 0 incremental backup, which is the base for subsequent incremental backups, copies all blocks containing data. You can create a level 0 database backup as backup sets or image copies.

The only difference between a level 0 incremental backup and a full backup is that a full backup is never included in an incremental strategy. Thus, an incremental level 0 backup is a full backup that happens to be the parent of incremental backups whose level is greater than 0.

A level 1 incremental backup can be either of the following types:

- **Differential incremental backup**, which backs up all blocks changed after the most recent incremental backup at level 1 or 0
- **A cumulative incremental backup**, which backs up all blocks changed after the most recent incremental backup at level 0

Incremental backups are differential by default.

Note:
Cumulative backups are preferable to differential backups when recovery time is more important than disk space, because fewer incremental backups need to be applied during recovery.

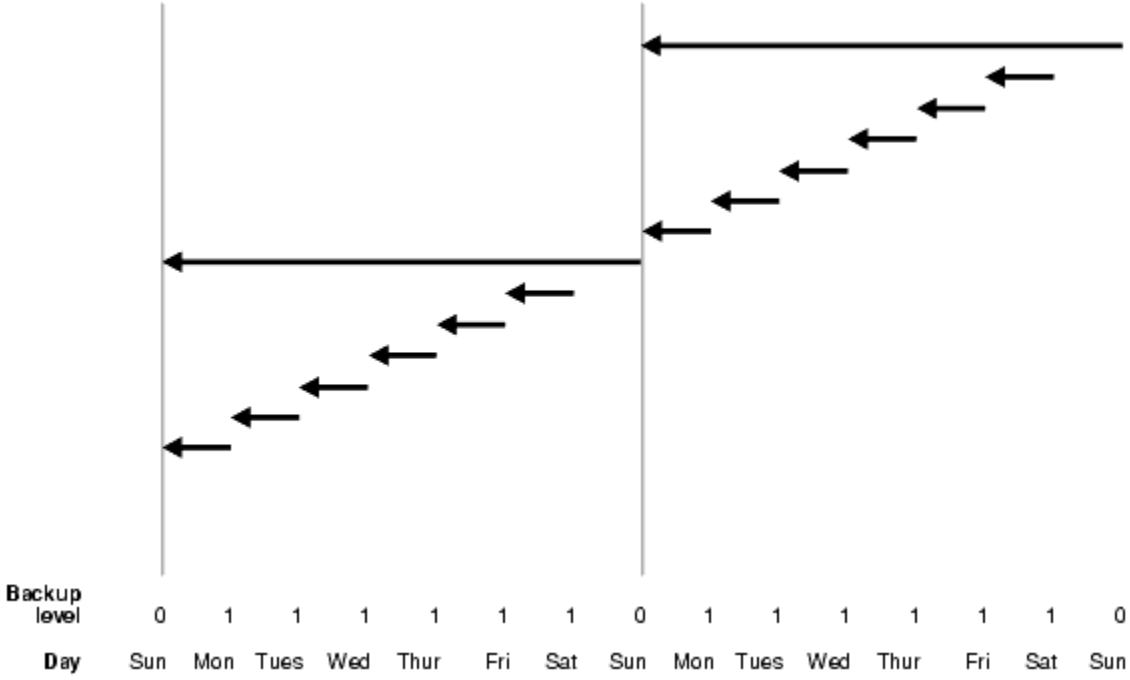
The size of the backup file depends solely upon the number of blocks modified, the incremental backup level, and the type of incremental backup (differential or cumulative).

Differential Incremental Backups

In a differential level 1 backup, RMAN backs up all blocks that have changed since the most recent incremental backup at level 1 (cumulative or differential) or level 0. For example, in a differential level 1 backup, RMAN determines which level 1 backup occurred most recently and backs up all blocks modified after that backup. If no level 1 is available, then RMAN copies all blocks changed since the base level 0 backup.

If no level 0 backup is available in either the current or parent incarnation, then the behavior varies with the compatibility mode setting. If compatibility is >=10.0.0, RMAN copies all blocks that have been changed since the file was created. Otherwise, RMAN generates a level 0 backup.

Differential Incremental Backups



the following activity occurs each week:

- Sunday

  An incremental level 0 backup backs up all blocks that have ever been in use in this database.

- Monday through Saturday

On each day from Monday through Saturday, a differential incremental level 1 backup backs up all blocks that have changed since the most recent incremental backup at level 1 or 0. The Monday backup copies blocks changed since Sunday level 0 backup, the Tuesday backup copies blocks changed since the Monday level 1 backup, and so forth.

Cumulative Incremental Backups

In a cumulative level 1 backup, RMAN backs up all blocks used since the most recent level 0 incremental backup in either the current or parent incarnation. Cumulative incremental backups reduce the work needed for a restore operation by ensuring that you only need one incremental backup from any particular level. Cumulative backups require more space and time than differential backups because they duplicate the work done by previous backups at the same level.

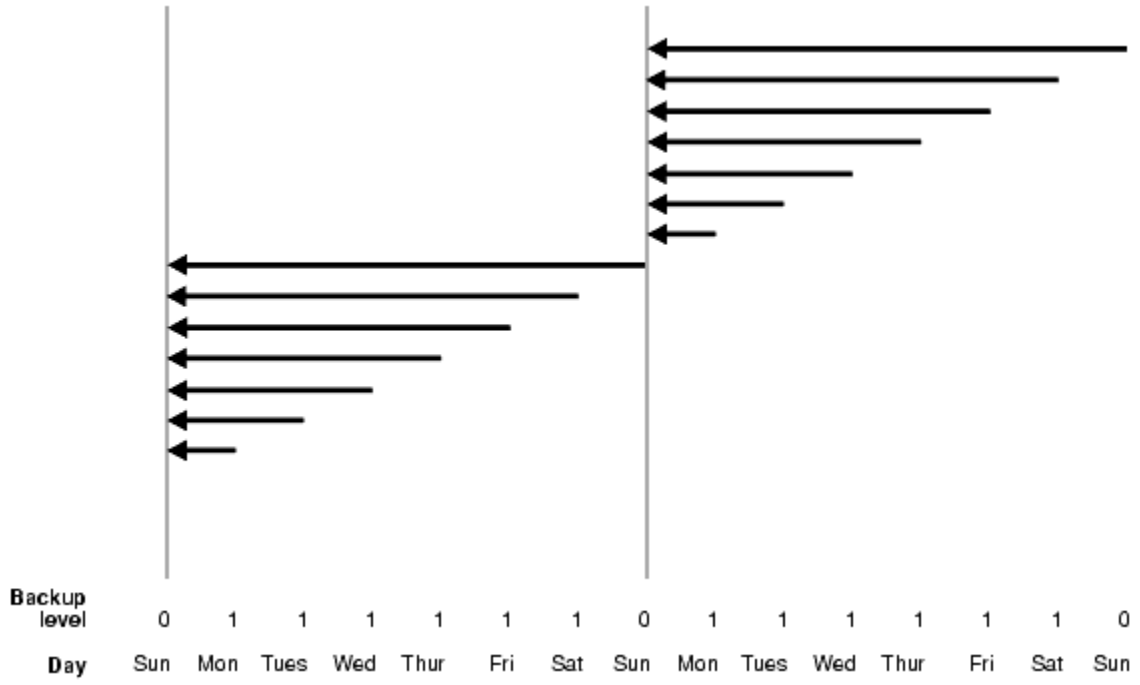
- Sunday

  An incremental level 0 backup backs up all blocks that have ever been in use in this database.

- Monday - Saturday

  A cumulative incremental level 1 backup copies all blocks changed since the most recent level 0 backup. Because the most recent level 0 backup was created on Sunday, the level 1 backup on each day Monday through Saturday backs up all blocks changed since the Sunday backup.

Cumulative Incremental Backups



Block Change Tracking

The block change tracking feature for incremental backups improves incremental backup performance by recording changed blocks in each datafile in a block change tracking file. This file is a small binary file stored in the database area. RMAN tracks changed blocks as redo is generated.

If block change tracking is enabled, then RMAN uses the change tracking file to identify changed blocks for incremental backups, thus avoiding the need to scan every block in the datafile. RMAN only uses block change tracking when the incremental level is greater than 0, because a level 0 incremental backup includes all blocks.

Backup Retention Policies

You can use the CONFIGURE RETENTION POLICY command to create a persistent and automatic backup retention policy. When a backup retention policy is in effect, RMAN considers a backup of datafiles or control files as an obsolete backup, that is, no longer needed for

recovery, according to criteria specified in the CONFIGURE command. You can use theREPORT OBSOLETE command to view obsolete files and the DELETE OBSOLETE command to delete them.

As you produce backups over time, older backups become obsolete as they are no longer needed to satisfy the retention policy. RMAN can identify the obsolete files for you, but it does not automatically delete them. You must use the DELETE OBSOLETE command to delete files that are no longer needed to satisfy the retention policy.

Recovery Window

A recovery window is a period of time that begins with the current time and extends backward in time to the point of recoverability. The point of recoverability is the earliest time for a hypothetical point-in-time recovery, that is, the earliest point to which you can recover following a media failure. For example, if you implement a recovery window of 1 week, then RMAN retains full backups and required incremental backups and archived logs so that the database can be recovered up to 7 days in the past. You implement this retention policy as follows:

CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;

This command ensures that for each datafile, one backup that is older than the point of recoverability is retained. For example, if the recovery window is 7, then there must always exist one backup of each datafile that satisfies the following condition:

SYSDATE - BACKUP CHECKPOINT TIME >= 7

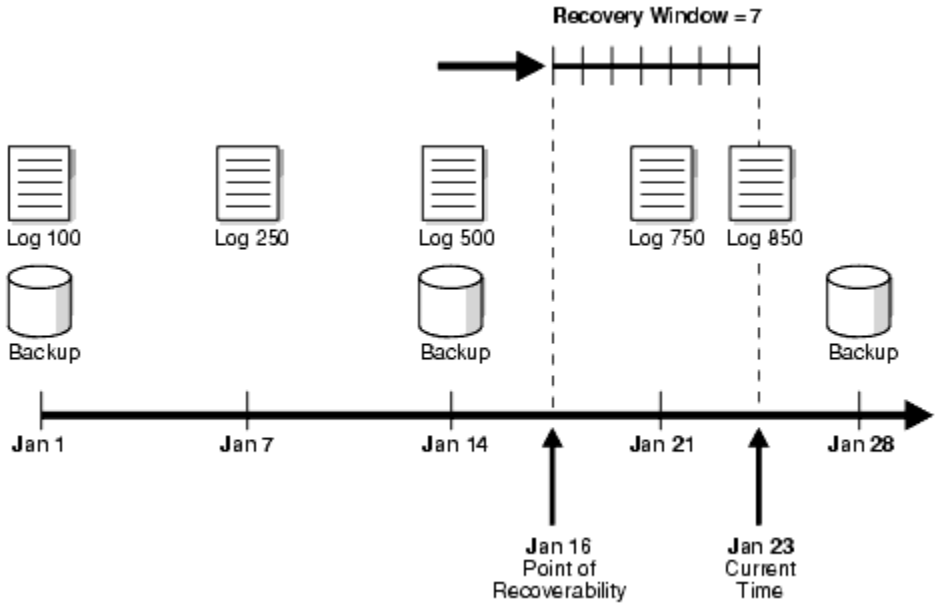
All backups older than the most recent backup that satisfied this condition are obsolete.

The retention policy has the following aspects:

- The recovery window is 7 days.
- Database backups are scheduled every two weeks on these days:
    - January 1
    - January 15
    - January 29
    - February 12

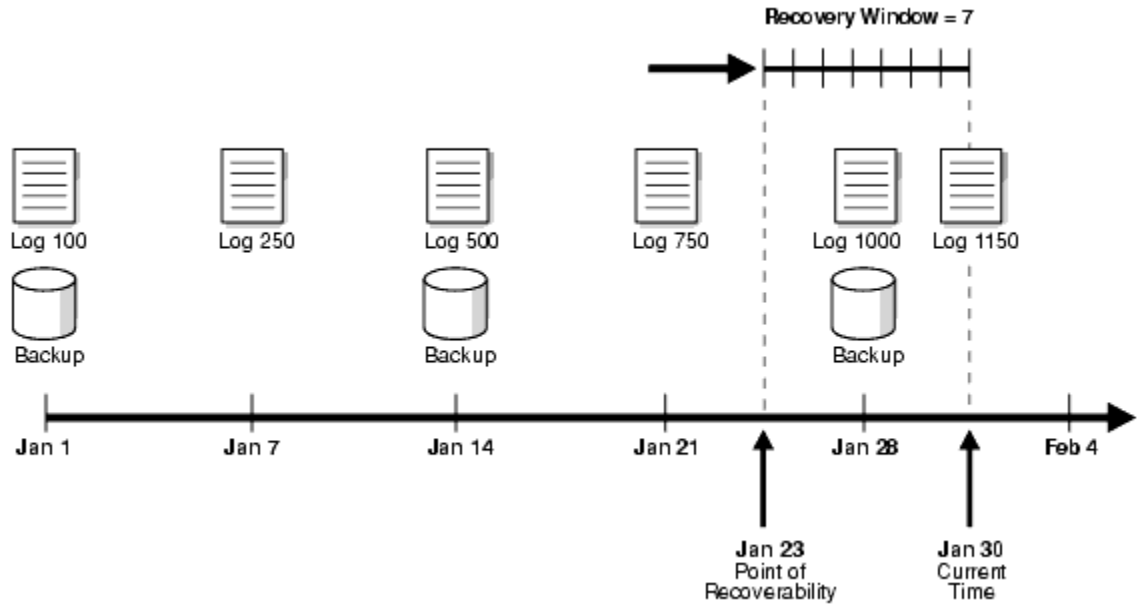
- The database runs in ARCHIVELOG mode, and archived logs are saved on disk only as long as needed for the retention policy.

Recovery Window, Part 1



the current time is January 23 and the point of recoverability is January 16. Hence, the January 14 backup is needed for recovery, and so are the archived logs from log sequence 500 through 850. The logs before 500 and the January 1 backup are obsolete because they are not needed for recovery to a point within the window.

Recovery Window, Part 2



In this scenario, the current time is January 30 and the point of recoverability is January 23. Note how the January 14 backup is not obsolete even though a more recent backup (January 28) exists in the recovery window. This situation occurs because restoring the January 28 backup does not enable you to recover to the earliest time in the window, January 23. To ensure recoverability to any point in the window, you must save the January 14 backup and all archived logs from sequence 500 to 1150.

Backup Redundancy

In some cases using a recovery window can complicate disk space planning because the number of backups that must be retained is not constant and depends on the backup schedule. In contrast, a redundancy-based retention policy specifies how many backups of each datafile must be retained. For example, you can configure a redundancy of 2 as follows:

CONFIGURE RETENTION POLICY TO REDUNDANCY 2;

The default retention policy is configured to REDUNDANCY 1.

4.1. Consistent Backups:

You can use the BACKUP command to make consistent and inconsistent backups of the database.

A **consistent backup** occurs when the database is in a consistent state. A database is in a consistent state after being shut down with the SHUTDOWNNORMAL, SHUTDOWN IMMEDIATE, or SHUTDOWN TRANSACTIONAL commands.

A consistent shutdown guarantees that all redo has been applied to the datafiles. If you mount the database and make a backup at this point, then you can restore the database backup later and open it without performing media recovery.

4.2. Inconsistent Backups:

- Any database backup that is not consistent is an **inconsistent backup**. A backup made when the database is open is inconsistent, as is a backup made after an instance failure or SHUTDOWN ABORT command. When a database is restored from an inconsistent backup.

- Oracle must perform media recovery before the database can be opened, applying any pending changes from the redo logs.
- RMAN does not permit you to make inconsistent backups when the database is in NOARCHIVELOG mode.
- As long as the database runs in ARCHIVELOG mode, and you back up the archived redo logs and datafiles, inconsistent backups can be the foundation for a sound backup and recovery strategy.
- Inconsistent backups offer superior availability because you do not have to shut down the database to make backups that fully protect the database.
- To perform recovery from inconsistent backup we need archive logs for recovery, so we need to put our database in archive log mode. Without archive log mode RMAN inconsistent backup is not possible.

COMMANDS

 # To connect to RMAN

[oracle@server1 ~]$ rman target /

 # To see configuration parameter values

RMAN> show all;

 # To change any configuration parameter

RMAN> configure retention policy to redundancy 5;

 # To backup the database

RMAN> backup database;

 # To backup archivelogs

RMAN> backup archivelog all;

 # To backup both database and archivelogs

RMAN> backup database plus archivelog;

Note: By default in 10g, rman backup will go to flash recovery area. To override that, use below command

 # To take backup to specified area

RMAN> backup format='/u03/rmanbkp/fulldbbkp_%t.bkp' database;

 # To see backup information

RMAN> list backup;

The above command will get the information from controlfile of the database

 # To find & delete expired backups

RMAN> crosscheck backup;

RMAN> delete expired backup;

RMAN> delete noprompt expired backup;

 # To find and delete expired archivelogs

RMAN> crosscheck archivelog all;

RMAN> delete expired archivelog all;

RMAN> delete noprompt expired archivelog all;

 # To find and delete unnecessary backups

RMAN> report obsolete;

RMAN> delete obsolete;

RMAN> delete noprompt obsolete;

 # To take physical image copy of database

RMAN> backup as copy database;

 # To validate the backup

RMAN> restore database validate;

 # To validate a particular backupset

RMAN> validate backupset 1234;

 # To take backup when using tape

```
RMAN> run
        {
          allocate channel c1 device type sbt_tape;
          backup database plus archivelog;
        }
```

 # To specify duration during backup

RMAN> backup duration 5 database;

The above command will run backup for 5 hours and will pause after that. It will continue on next day at scheduled time

RMAN RECOVERY SCENARIOS

STEPS to recover a datafile

```
RMAN> run
{
sql 'alter tablespace mydata offline';
restore tablespace mydata;
recover tablespace mydata;
sql 'alter tablespace mydata online';
}
```

STEPS to recover a system datafile

```
RMAN>run
{
shutdown immediate;
startup mount;
restore datafile 1;
recover datafile 1;
sql 'alter database open';
}
```

STEPS to recover a redolog files

```
RMAN> run
{
shutdown immediate;
startup mount;
set until scn 1234; or set until time "to_date('2011-01-05 11:30:00','YYYY-MM-DD
hh24:mi:ss')";
recover database;
sql 'alter database open resetlogs';
}
```

STEPS to recover a controlfiles

```
RMAN> run
{
```

```
shutdown immediate;
startup nomount;
restore controlfile from autobackup;
sql 'alter database mount';
recover database;
sql 'alter database open resetlogs';
}
```

RECOVERY CATALOG

1. RMAN will store the backup information in target database controlfile. If we lost this controlfile and perform either complete or incomplete recovery, we will loose backup info even though physically backups are available
2. To avoid this situation RMAN introduced recovery catalog. It is a database which stores target database backup information
3. Single recovery catalog can support multiple target databases
4. We cannot obtain recovery catalog information from target but vice versa is possible

Steps for Configuring Recover Catalog

```
SQL> create tablespace rmantbs
       datafile '/datafiles/prod/rmantbs01.dbf' size 50m;

SQL> create user rman_rc identified by rman_rc
       default tablespace rmantbs
       temporary tablespace temp;

SQL> grant connect,resource,recovery_catalog_owner to rman_rc;

[oracle@server1 ~]$ rman catalog rman_rc/rman_rc@rc

RMAN> create catalog;

[oracle@server1 ~]$ rman target / catalog rman_rc/rman_rc@rc

RMAN> register database;
```

INCREMENTAL BACKUP

1. Taking backup of very large database (VLDB) will take time if the backup size is increasing frequently

2. In such cases, we can go for incremental backup which will take backup of any changes happend from last full backup to till date
3. Incremental backups are two types
    a. Differential (default)
    b. Cumulative
4. Both incremental backup types will have level 0 and level 1 (level 0 –full backup, level 1- incremental backup)
5. First time incremental backup will do level 0 backup always
6. RMAN will perform incremental backup by identifying changed blocks with the help of block SCN
7. We cannot recover database using level 1 backup applying on full database backup
8. We can apply level 1 backup on image copies and can recover the database
9. 10g RMAN can perform faster incremental backups using block change tracker. With this whenever any block changes CTWR (change track writer) background process will write that information to a tracking file
10. The change tracking file resides in DB_CREATE_FILE_DEST

COMMANDS

 # To take full backup in incremental mode

RMAN> backup incremental level 0 database;

RMAN> backup cumulative incremental level 0 database;

 # To take differential backup

RMAN> backup incremental level 1 backup;

 # To take cumulative backup

RMAN> backup cumulative incremental level 1 database;

 # To enable change tracking

SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;

You can also create the change tracking file in a location you choose yourself, using the following SQL statement:

```
SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING  USING FILE
'/mydir/rman_change_track.f' REUSE;
```

The REUSE option tells Oracle to overwrite any existing file with the specified name.

 # To disable change tracking

```
SQL> ALTER DATABASE DISABLE BLOCK CHANGE TRACKING;
```

 # To apply incremental backup to datafile copy or level 0 backup

```
RMAN> RUN {
  RECOVER COPY OF DATABASE WITH TAG 'incr_update';
  BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY WITH TAG 'incr_update'
     DATABASE;
  }
```