Exam Objective: Configure and use Flashback Database

As an enhancement to recovery features, this new feature in Oracle 10g is used to revert an entire database to a state it was in at a previous point in time. This is particularly useful when you want to undo changes that occur as a result of user error or logical data corruptions.

Implementing Flashback Database

The flashback database is implemented using a new type of log file called '**Flashback Logs**'. These logs contain the before images of data blocks or flashback data along with some additional information. These before images are stored by the Oracle server periodically by a new background process called Recovery Writer.

The flashback logs are created in an area known as the Flash Recovery Area. Flashback of a database can be done using the Enterprise Manager Console, RMAN, the recovery wizard or SQL commands.

While performing a flashback database operation some tablespaces may be excluded if required. Setting the FLASHBACK_ON attribute of the tablespace can help achieve this. The command to be executed to exclude a tablespace is:

```
ALTER TABLESPACE tablespacename FLASHBACK OFF;
```

On issuing this command, the specific *tablespace* will not participate in the flashback operation.

Restrictions on using flashback database feature

Flashback database cannot be performed under the following situation:

- If the control file was restored or re-created during the span of the flashback.
- If a datafile has been manually resized using the RESIZE command. This restriction does not apply to datafiles that extend automatically as a result of their auto-extensible property.
- If a tablespace or datafile was dropped in the span of the time you are flashing back. The datafile will be added to the control file, and will be marked offline. It will not be flashed back. The tablespace or datafile would have to be restored or recovered after the completion of the flashback database.
- An incomplete recovery was performed during the span of the flashback and the database was opened with the RESETLOGS option.

Steps to configure the Flashback Database

Pre-requisites: The database must be in ARCHIVELOG mode to enable Flashback Database.

1. Specify values for the initialization parameters:
   - `DB_RECOVERY_FILE_DEST` - default database recovery file location.
   - `DB_RECOVERY_FILE_DEST_SIZE` - specifies (in bytes) the hard limit on the total space to be used by target database recovery files created in the flash recovery area.
   - `DB_FLASHBACK_RETENTION_TARGET` – specifies (in minutes) an upper limit as to how far back you want to flashback the database. The default value is 1440. The value depends on how much of flashback data has been kept in the flash recovery area. The value of this parameter is specified based on the current workload in your database.
2. Open the database in a `MOUNT EXCLUSIVE` mode and issue the command : `SQL>ALTER DATABASE FLASHBACK ON;`
3. At this point you would issue the FLASHBACK command with the appropriate option.
4. After the flashback database operation is complete, you would want to open the database in read only mode to verify if the correct target time or SCN was used, and that the database has been flashed back as desired. You must open the database with the RESETLOGS option after a flashback database is performed.

Commands used for flashback of a database

If performing flashback of the database using RMAN, you can use options:

```
1) RMAN> FLASHBACK DATABASE TO TIME = time format
2) RMAN> FLASHBACK DATABASE TO SCN=n;
3) RMAN> FLASHBACK DATABASE TO SEQUENCE=m THREAD=n;
```

If you use SQL to perform the Flashback of the database, you can either specify a timestamp or an SCN number. The database should be mounted for the command to work as given below:

```
SQL> STARTUP MOUNT;
SQL> FLASHBACK DATABASE TO TIMESTAMP (SYSDATE – 2/24);
SQL> FLASHBACK DATABASE TO SCN n;
```

Guidelines

1. The `V$FLASHBACK_DATABASE_LOG` view can be queried to retrieve the SCN or time against the columns `OLDEST_FLASHBACK_SCN` and `OLDEST_FLASHBACK_TIME.`

2. In the third option using SEQUENCE number, the redo log sequence number is an upper limit that is not inclusive. Database is recovered up to sequence number (m-1).

Important Data Dictionary Views

Relevant information required to perform a flashback database can be retrieved from the `V$FLASHBACK_DATABASE_LOG` view.
The columns of relevance in this view include:

- `OLDEST_FLASHBACK_SCN` – lowest system change number (SCN) in the flashback data.
- `OLDEST_FLASHBACK_TIME` – time of the lowest SNC in the flashback data.
- `FLASHBACK_SIZE` – Current size (in bytes) of the flashback data.
- `RETENTION_TARGET` – The target retention time in minutes.
- `ESTIMATED_FLASHBACK_SIZE` – The estimated size of flashback data needed for the current target retention.

Monitoring Flashback Logs

You can monitor the flashback logs by querying the `V$FLASHBACK_DATABASE_STAT` view.

This view contains 24 rows where each row represents an hour. It contains estimated flashback space needed based on the workload. Information from this view can help you decide how you would want to adjust the retention time or the flash recovery area size.

To display information from the view you could issue a query such as:

```
SQL> SELECT * FROM V$FLASHBACK_DATABASE_STAT;

BEGIN_TIM END_TIME  FLASHBACK_DATA    DB_DATA  REDO_DATA
--------- --------- -------------- ---------- ----------
ESTIMATED_FLASHBACK_SIZE
------------------------
21-JUL-04   21-JUL-04    1769472     868352     201728
                          0
```

- `REDO_DATA` – is amount redo in bytes, written during the time interval.
- `FLASHBACK_DATA` – is the amount of flashback data in bytes, written during the time interval.

Disabling the flashback database feature

Using SQL you can disable flashback database by issuing
`SQL> ALTER DATABASE FLASHBACK OFF;`

With the execution of this command, all existing flashback logs will be deleted.

Exam Objective: Recover dropped tables with Flashback Drop Feature

The Flashback Drop can be used to recover from a table that has been accidentally dropped. If a user accidentally dropped a table in a earlier version of Oracle, you would have to perform a point-in-time recovery.

In Oracle 10g, the Oracle server maintains a RECYCLE BIN. An object that is dropped is put into the recycle bin and renamed with a unique name. As long as a table is in the recycle bin it can be "undropped". It remains in the recycle bin until one of the following situations occurs:

- The tablespace that contains the dropped object cannot accommodate any more data.
- The datafile extends automatically.

Reclamation of space for objects in the recycle bin is dependent on the space utilization in the tablespace. The objects that were the first to enter the recycle bin are the first to leave. Oracle automatically purges the recycle bin when there is no space in the tablespace or the user's quota is met.

Syntax for the FLASHBACK DROP command

The command that is used to recover a table is:

```
FLASHBACK TABLE tablename TO BEFORE DROP
[RENAME TO newname];
```

The rename clause is used to rename the object after restoring it. When using this clause the original name of the table needs to be specified.

Example

A user accidentally issues a DROP TABLE command.

```
SQL> DROP TABLE xyz;
```

The user can then issue the command given below to restore the table called *xyz*.

```
SQL> FLASHBACK TABLE xyz TO BEFORE DROP;
```

Or

```
SQL> FLASHBACK TABLE xyz TO BEFORE DROP RENAME TO newname;
```

Syntax changes to the DROP TABLE command

In Oracle 10g, the DROP TABLE command takes the following forms:

1. `DROP TABLE tablename ;`

Stores the objects in the recycle bin. Since the extents remain in the datafile, they continue to count towards the quota the user has been given in the tablespace. All constraints and triggers are preserved when the object is dropped.

2. `DROP TABLE tablename PURGE;`

This command drops the object permanently and deallocates the objects' associated extents.

The RECYCLE BIN

An object when dropped is sent to the recycle bin.
There is no guarantee how long an object remains in the recycle bin. An object that is dropped is renamed with globally unique identifier. The naming convention takes the form BIN$unique_id$version.unique_id. The version is the version number of the database server. For *eg*. **BIN$Eo8hhIYhQceT5IQc01hI3g==$1**

You can query the USER_RECYCLEBIN or RECYCLEBIN data dictionary views to see the objects you dropped that are currently in the recycle bin. The DBA can query the DBA_RECYCLEBIN to query all objects that were dropped system-wide.

You can still query objects that are in the recycle bin. However you would have to issue the SELECT against the new name given to it.

```
SQL>SELECT *
      FROM BIN$xyieMDw/dhrwEEJ/wed[4d=r3=$0";
```

Deleting Objects Permanently

The PURGE command can be used to permanently remove an object from the database. All the extents allocated it will also be released and can be reused in the datafile. The entry about the object is removed from the RECYCLE BIN.

```
PURGE TABLE table_name;
```

As an example, the table renamed as BIN$xyieMDw/dhrwEEJ/wed[4d=r3=$0 will be permanently deleted.

```
SQL> PURGE TABLE BIN$xyieMDw/dhrwEEJ/wed[4d=r3=$0";
```

1. The purge command can also be used against a tablespace. All objects present in the recycle bin that belong to the specified tablespace will be dropped.

```
PURGE TABLESPACE tablespacename;
```

2. The purge command can be used to drop objects belonging to a specific user in a tablespace. If some dependent objects such as LOBs, nested tables, or partitions exist in the tablespace that is being purged, the parent objects that may be stored in another tablespace will also be purged.

```
SQL> PURGE TABLESPACE tablespacename USER username ;
```

3. The recycle bin can be purged using:

```
SQL> PURGE RECYCLEBIN;
SQL> PURGE DBA_RECYCLEBIN; -- you need SYSDBA to do this
```

Dropping a Tablespace or User

When the DROP TABLESPACE command is issued the tablespace must be empty. Any objects of this tablespace currently in the recycle bin will get purged.

If a tablespace is dropped using the DROP TABLESPACE *tablespacename* INCLUDING CONTENTS option, all objects in the tablespace are purged without being sent to the recycle bin.

When a user is dropped using the DROP USER username CASCADE command, the user along with his objects are permanently deleted or purge. Any objects that belong to the user in the recycle bin will also get purged.

Exam Objective: Retrieve row history information with the Flashback Versions Query feature

Flashback Query was a feature that was introduced in Oracle 9i. Using this feature a user could issue queries that were consistent as of a certain time or user-specified system change number (SCN). The functionality of this feature has been expanded in Oracle 10g.

The Flashback Versions Query feature enables you to retrieve all the versions of rows that exist or existed between the time of the query and a previous point in time. The VERSIONS BETWEEN clause is used to implement this type of flashback feature. The rows that are returned are a list of all committed changes made to the rows across transactions, excluding the current transaction.

Configuring Flashback Versions Query

To use the flashback versions query feature you need to specify the following initialization parameters:
UNDO_TABLESPACE – the tablespace that will contain the undo data for the database.
UNDO_RETENTION - specifies in seconds the amount of time undo data should remain in the database.

- The VERSIONS BETWEEN cannot produce versions of rows across structural changes make to the table.
- The VERSIONS BETWEEN clause cannot be used against a view however you can use the clause in the definition of the view.

- The VERSIONS BETWEEN cannot produce versions of rows across certain DDL commands.
- The VERSIONS BETWEEN can be used in subqueries in DDL and DML statements, such as,

```
CREATE TABLE XX
SELECT * FROM YY
VERSIONS BETWEEN TIMESTAMP
TO_TIMESTAMP('2004-10-3','YYYY-MM-DD') AND
TO_TIMESTAMP('2004-12-3','YYYY-MM-DD');
```

- The FLASHBACK and SELECT object privileges are required to use this feature.
- To issue queries on all tables you would need the FLASHBACK ANY TABLE privilege.

SYNTAX

```
SELECT column, column
FROM table
VERSIONS BETWEEN
SCN minvalue AND maxvalue
[WHERE condition ] ;
```

(In the syntax displayed above you could either specify a lower SCN number and upper SCN number or an identifier name that holds a number).

```
Or
SELECT column, column, …
FROM table
VERSIONS BETWEEN TIMESTAMP
      timestamp1 AND timestamp2
[WHERE condition ];
```

(In the syntax above *timestamp1* and *timestamp2* could take a timestamp value in a format such as TO_TIMESTAMP('2004-10-3','YYYY-MM-DD').

Exam Objective: Audit or recover from transactions with the Flashback Transactions Query feature

Flashback Transactions Query is another new feature in Oracle 10g that is used to view changes made to the database at a transaction level. It is particularly useful to audit changes made by transactions within a specified time frame.

Implementing Flashback Transactions Query

To configure the Flashback Transactions Query you need to set the following initialization parameters:

- UNDO_TABLESPACE – the tablespace that will contain the undo data for the database.
- UNDO_RETENTION – specifies in seconds the amount of time undo data should remain in the database.

Retrieving transaction information

The FLASHBACK_TRANSACTION_QUERY view is used to determine what changes were made by a specific transaction or what changes were made during a specific time interval.

The flashback transactions query can be used in conjunction with Flashback Versions Query to recover the database from user or application errors at a transaction level. The Flashback Versions Query helps to determine the transaction ID that modified the row of interest. The transaction ID can then be used to determine the changes made by the transaction. This is auditing mechanism and is much faster than using the LogMiner utility.

An example is displayed below:

```
SQL> SELECT VERSIONS_XID, column
       FROM table
       VERSIONS BETWEEN SCN
        scn1  AND scn2
        [WHERE Condition];

SQL> SELECT OPERATION, UNDO_SQL, TABLE_NAME
       FROM flashback_transaction_query
       WHERE xid = 'n';
```

Note: *n* was returned as the value for the VERSIONS_XID column in the previous query above. The UNDO_SQL column can be used to rollback changes made by the transaction.

Exam Objectives: Recover tables to a point in time with the Flashback Table feature

The Flashback Table is a new Oracle 10g feature that enables you to recover a table or tables to a previous point in time without restoring a backup. A user can easily and quickly recover from accidental modifications to tables without the involvement of the DBA.

Implementing Flashback table

The initialization parameters that need to be configured for implementing the flashback table feature are:

- UNDO_TABLESPACE – specifies the tablespace that will hold the undo data.
- UNDO_RETENTION – specifies the amount of time in seconds that undo data should be retained in the database. You would set it based on how far back you would need to flash back.
- You must enable row movement on a table to be able to flashback the table. A command you could use is:
  SQL> ALTER TABLE *tablename* ENABLE ROW MOVEMENT;
- You use the FLASHBACK TABLE command to flash back one or more tables to a specified SCN or previous point in time, within the retention period. To find out the current SCN number you can query the V$DATABASE using :
  SQL> SELECT CURRENT_SCN FROM V$DATABASE;

SQL commands to Flashback a table

The following SQL commands can be used to flashback a table.

SQL> FLASHBACK TABLE *schema.tablename*
TO SCN n;

SQL> FLASHBACK TABLE *schema.tablename*
TO TIMESTAMP *timestamp_to_flashback_to*;

timestamp_to_flashback_to : can take the form TO_TIMESTAMP('2004-10-03','YYYY-MM-DD')