Flashback in Oracle

- Flashback table restores the existing tables to earlier versions using timestamp and SCN number.
- It also helps to retrieve removed tables from the database, dropped using DROP and TRUNCATE commands.
- Tables altered using DDL statements can also not be restored to a prior state with flashback operation.
- One thing you must remember about flashback operation is that undo of data is controlled by UNDO_RETENTION parameter. In simpler words, if undo data is available, then only tables can be restored back.
- Once you trigger FLASHBACK TABLE statement on any table, it cannot be rolled back. Best practice is to record the current SCN number before issuing a FLASHBACK TABLE statement.
- You must enable row movement during the creation of the table to carry out flashback operation on it. Row movement can also be enabled using ALTER command later on.

Flashback Table Before Drop

- You can flashback a dropped table from recyclebin using flashback table command

```
SQL> SHOW RECYCLEBIN;
SQL> FLASHBACK TABLE "BIN$gk3lsj/3akk5hg3j2lkl5j3d==$0" TO BEFORE
DROP;
or
SQL> FLASHBACK TABLE SCOTT.FLASH_EMP TO BEFORE DROP;
```

- You can even rename table while flashing it back from recyclebin

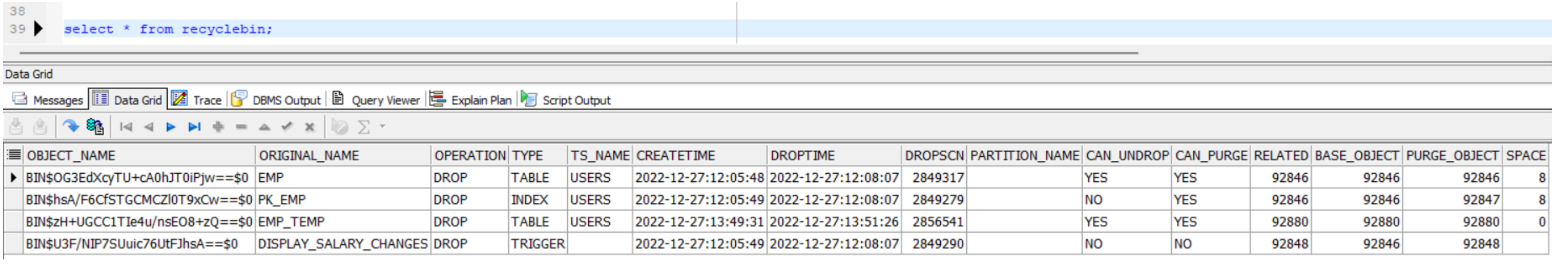
```
SQL> FLASHBACK TABLE SCOTT.FLASH_EMP TO BEFORE DROP RENAME
TO NEW_EMP;
```

Note: *Recyclebin must be enabled to use flashback table before drop*

Flashback in Oracle

*SQL> select * from recyclebin;*

- *The SCN is an internal number maintained by the database management system to log changes made to a database*
- *Whenever an application commits a transaction, the log writer process (LGWR)*
- *writes records from the redo log buffers in the (SGA) to the online redo logs on disk.*
- *LGWR also writes the transaction’s SCN to the online redo log file.*



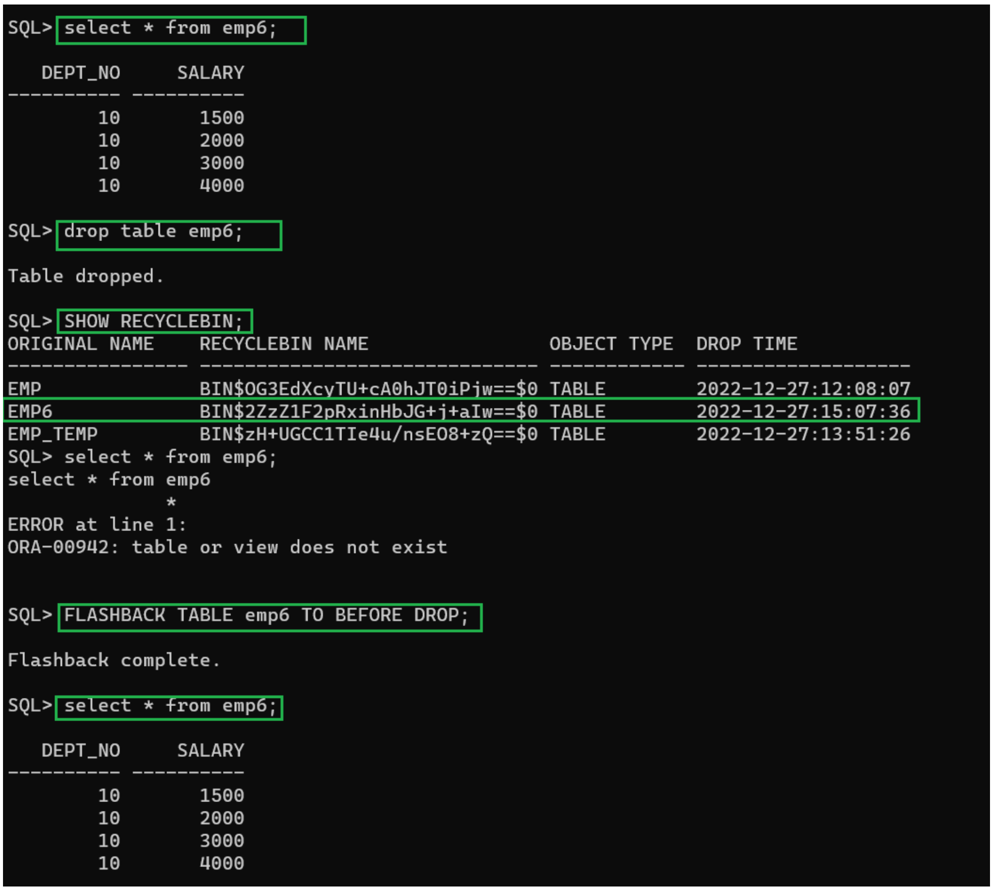
```
select * from recyclebin;
```

OBJECT_NAME	ORIGINAL_NAME	OPERATION	TYPE	TS_NAME	CREATETIME	DROPTIME	DROPSCN	PARTITION_NAME	CAN_UNDROP	CAN_PURGE	RELATED	BASE_OBJECT	PURGE_OBJECT	SPACE
BIN$OG3EdXcyTU+cA0hJT0iPjw==$0	EMP	DROP	TABLE	USERS	2022-12-27:12:05:48	2022-12-27:12:08:07	2849317		YES	YES	92846	92846	92846	8
BIN$hsA/F6CfSTGCMCZl0T9xCw==$0	PK_EMP	DROP	INDEX	USERS	2022-12-27:12:05:49	2022-12-27:12:08:07	2849279		NO	YES	92846	92846	92847	8
BIN$zH+UGCC1TIe4u/nsEO8+zQ==$0	EMP_TEMP	DROP	TABLE	USERS	2022-12-27:13:49:31	2022-12-27:13:51:26	2856541		YES	YES	92880	92880	92880	0
BIN$U3F/NIP7SUuic76UtFJhsA==$0	DISPLAY_SALARY_CHANGES	DROP	TRIGGER		2022-12-27:12:05:49	2022-12-27:12:08:07	2849290		NO	NO	92848	92846	92848	

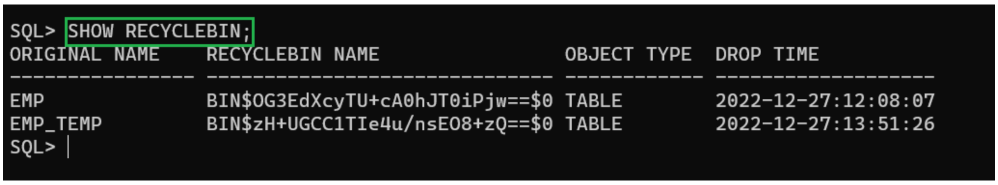
*SQL> select * from recyclebin;*

- *You can do this command(purge recyclebin) to make the recyclebin empty;*

Flashback in Oracle



```
SQL> select * from emp6;

   DEPT_NO     SALARY
---------- ----------
        10       1500
        10       2000
        10       3000
        10       4000

SQL> drop table emp6;

Table dropped.

SQL> SHOW RECYCLEBIN;
ORIGINAL NAME    RECYCLEBIN NAME                OBJECT TYPE  DROP TIME
---------------- ------------------------------ ------------ -------------------
EMP              BIN$OG3EdXcyTU+cA0hJT0iPjw==$0 TABLE        2022-12-27:12:08:07
EMP6             BIN$2ZzZ1F2pRxinHbJG+j+aIw==$0 TABLE        2022-12-27:15:07:36
EMP_TEMP         BIN$zH+UGCC1TIe4u/nsEO8+zQ==$0 TABLE        2022-12-27:13:51:26
SQL> select * from emp6;
select * from emp6
              *
ERROR at line 1:
ORA-00942: table or view does not exist


SQL> FLASHBACK TABLE emp6 TO BEFORE DROP;

Flashback complete.

SQL> select * from emp6;

   DEPT_NO     SALARY
---------- ----------
        10       1500
        10       2000
        10       3000
        10       4000
```



```
SQL> SHOW RECYCLEBIN;
ORIGINAL NAME    RECYCLEBIN NAME                OBJECT TYPE  DROP TIME
---------------- ------------------------------ ------------ -------------------
EMP              BIN$OG3EdXcyTU+cA0hJT0iPjw==$0 TABLE        2022-12-27:12:08:07
EMP_TEMP         BIN$zH+UGCC1TIe4u/nsEO8+zQ==$0 TABLE        2022-12-27:13:51:26
SQL> |
```

RECYCLEBIN and USER_RECYCLEBIN

- USER_RECYCLEBIN is the view from which you can retrieve the tables that are dropped or truncated.

```
SQL> SELECT * FROM user_recyclebin;
```

- RECYCLEBIN is the synonym of USER_RECYCLEBIN view.

TO SCN Clause

- The table can be returned to a point in time of mentioned SCN number using TO SCN clause.

- SCN number can be retrieved from the v$database table using below query

Flashback in Oracle

```
SQL> SELECT current_scn FROM v$database;
```

- You can also convert a timestamp to SCN number and vice versa using SCN_TO_TIMESTAMP and TIMESTAMP_TO_SCN functions

```
SQL> SELECT TIMESTAMP_TO_SCN(SYSTIMESTAMP – INTERVAL '2' MINUTE) FROM dual;
```

- SCN_TO_TIMESTAMP(3187302511937) will retrieve the timestamp of this SCN number.

TO TIMESTAMP Clause

- In order to restore the table back to particular timestamp, use TO TIMESTAMP clause.

TO BEFORE DROP Clause

- The clause TO BEFORE DROP retrieves back the dropped table.
- However, the tables dropped using PURGE option cannot be retrieved back.
- TO BEFORE DROP clause is also unable to recover tables dropped TRUNCATE TABLE command.

RENAME TO Clause

- RENAME TO clause retrieves the table from recycle bin with a new name during flashback operation.

Retrieve Table After TRUNCATE TABLE

- Tables dropped by TRUNCATE TABLE can be restored using TO TIMESTAMP and TO SCN clause. Recovery of lost data is not possible in this case.

ENABLE or DISABLE TRIGGERS

- By default Oracle database disable triggers during the flashback. These triggers are enabled when flashback operation is completed.
- You need to specify ENABLE TRIGGERS to keep the triggers enabled during flashback operation.

```
SQL> FLASHBACK TABLE emp_temp TO SCN (3187302511937) ENABLE TRIGGERS;
```

- However, the triggers that are already disabled before flashback operation cannot be enabled using this option.
- DISABLE TRIGGERS acts the same as a default behaviour in flashback operation.

Flashback Table

- You can flashback table to a particular SCN or time in the past. Before you can flashback table, you must enable row movement

```
SQL> ALTER TABLE test.emp_temp ENABLE ROW MOVEMENT;
```

- Now you are ready to flashback table to SCN or timestamp

```
SQL> FLASHBACK TABLE EMP TO SCN <scn_no>;

SQL> FLASHBACK TABLE HR.EMPLOYEES TO TIMESTAMP
     TO_TIMESTAMP('2016-05-12 18:30:00', 'YYYY-MM-DD HH24:MI:SS');
```

Note: *for flashback table, enabling FLASHBACK DATABASE is not required at all*

Log:-

```
SQL> select * from test.test;

SQL> commit;

SQL> select to_char(sysdate, 'DD-MM-YYYY:HH24:MI:SS') from dual;
```

Flashback in Oracle

```
SQL> update
test
set salary=salary+100
where dept_no=10;

SQL>commit;

SQL> ALTER TABLE test.test ENABLE ROW MOVEMENT;

SQL> FLASHBACK TABLE test.test TO TIMESTAMP
        TO_TIMESTAMP('28-12-2022:12:58:38', 'DD-MM-YYYY:HH24:MI:SS');


SQL> select * from test.test;

   DEPT_NO     SALARY
---------- ----------
        10       1800
        10       2300
        10       3300
        10       4300

SQL> commit;

Commit complete.

SQL> select to_char(sysdate, 'DD-MM-YYYY:HH24:MI:SS') from dual;

TO_CHAR(SYSDATE,'DD
-------------------
28-12-2022:12:58:38

SQL> update
  2  test
  3  set salary=salary+100
  4  where dept_no=10;

4 rows updated.

SQL> commit;

Commit complete.
```

Flashback in Oracle

```
SQL> select to_char(sysdate, 'DD-MM-YYYY:HH24:MI:SS') from dual;

TO_CHAR(SYSDATE,'DD
-------------------
28-12-2022:12:58:52

SQL> select * from test.test;

   DEPT_NO     SALARY
---------- ----------
        10       1900
        10       2400
        10       3400
        10       4400
```

```
SQL> FLASHBACK TABLE test.test TO TIMESTAMP
  2  TO_TIMESTAMP('28-12-2022:12:58:38', 'DD-MM-YYYY:HH24:MI:SS');
FLASHBACK TABLE test.test TO TIMESTAMP
                *
ERROR at line 1:
ORA-08189: cannot flashback the table because row movement is not enabled

SQL> ALTER TABLE test.test ENABLE ROW MOVEMENT;

Table altered.

SQL> FLASHBACK TABLE test.test TO TIMESTAMP
  2  TO_TIMESTAMP('28-12-2022:12:58:38', 'DD-MM-YYYY:HH24:MI:SS');


Flashback complete.
```

Flashback in Oracle

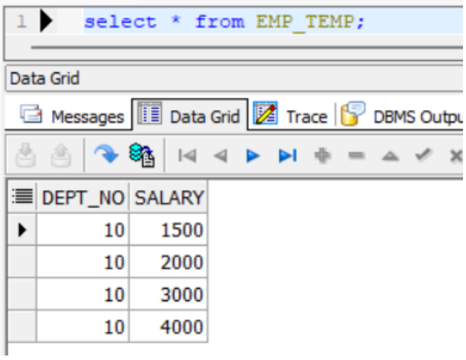
```
SQL> select * from test.test;

  DEPT_NO    SALARY
---------- ----------
        10       1800
        10       2300
        10       3300
        10       4300
```

END LOG:-

Oracle Flashback Version Query

- *Use Oracle Flashback Version Query to retrieve the different versions of specific rows that existed during a given time interval.*
- *A row version is created whenever a COMMIT statement is executed.*

SQL> Select * from EMP_TEMP;

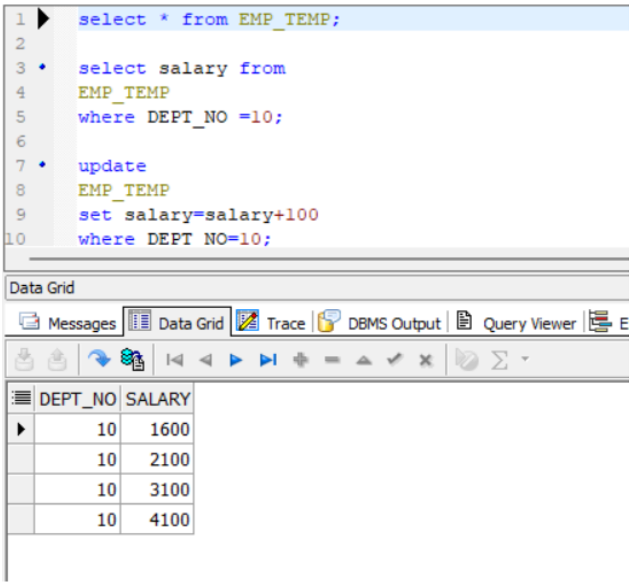


```
select * from EMP_TEMP;
```

DEPT_NO	SALARY
10	1500
10	2000
10	3000
10	4000

```
 SQL>  update
EMP_TEMP
set salary=salary+100
where DEPT_NO=107;
commit;
```

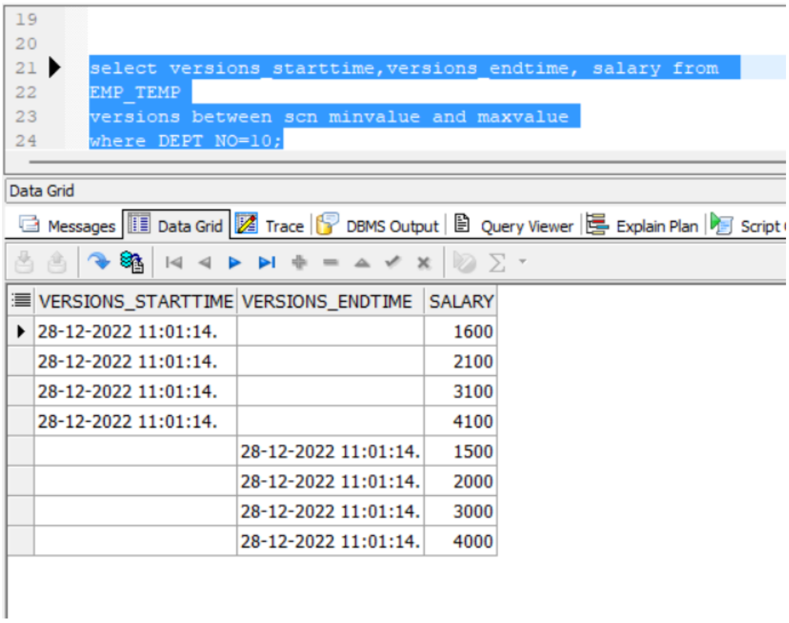
Flashback in Oracle



```
select * from EMP_TEMP;

select salary from
EMP_TEMP
where DEPT_NO =10;

update
EMP_TEMP
set salary=salary+100
where DEPT_NO=10;
```

DEPT_NO	SALARY
10	1600
10	2100
10	3100
10	4100

SQL> select versions_starttime,versions_endtime, salary from
EMP_TEMP
versions between scn minvalue and maxvalue
where DEPT_NO=107;



```
select versions_starttime,versions_endtime, salary from
EMP_TEMP
versions between scn minvalue and maxvalue
where DEPT_NO=10;
```

VERSIONS_STARTTIME	VERSIONS_ENDTIME	SALARY
28-12-2022 11:01:14.		1600
28-12-2022 11:01:14.		2100
28-12-2022 11:01:14.		3100
28-12-2022 11:01:14.		4100
	28-12-2022 11:01:14.	1500
	28-12-2022 11:01:14.	2000
	28-12-2022 11:01:14.	3000
	28-12-2022 11:01:14.	4000

Flashback in Oracle

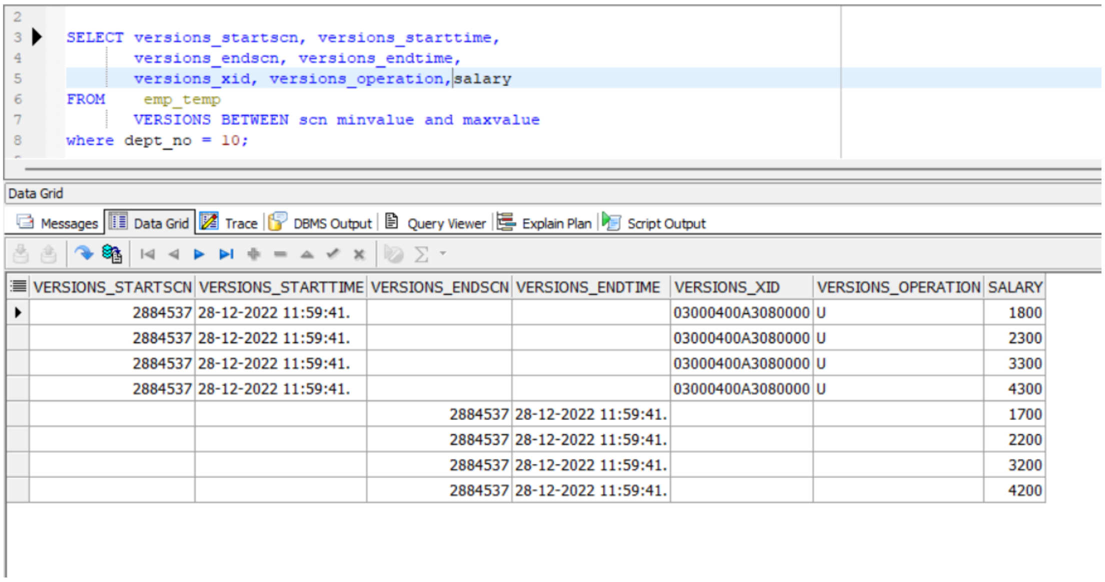
- You can also use this for more information like here we find VERSIONS_SCN AND VERSIONS_TIMES and which kind of operation perform (update,insert).

```
SQL> SELECT versions_startscn, versions_starttime,
       versions_endscn, versions_endtime,
       versions_xid, versions_operation,salery
FROM   emp_test
       VERSIONS BETWEEN scn minvalue and maxvalue
WHERE  dept_no = 10;
```

OR

<u>Note:- here we use timestamp.</u>

```
SQL> SELECT versions_startscn, versions_starttime,
       versions_endscn, versions_endtime,
       versions_xid, versions_operation,salery
FROM   emp_test
       VERSIONS BETWEEN TIMESTAMP TO_TIMESTAMP('2004-03-29 14:59:08', 'YYYY-
MM-DD HH24:MI:SS')
                  AND TO_TIMESTAMP('2004-03-29 14:59:36', 'YYYY-MM-DD HH24:MI:SS')
WHERE  dept_no = 10;
```



```
SELECT versions_startscn, versions_starttime,
       versions_endscn, versions_endtime,
       versions_xid, versions_operation,salary
FROM    emp_temp
       VERSIONS BETWEEN scn minvalue and maxvalue
where dept_no = 10;
```

VERSIONS_STARTSCN	VERSIONS_STARTTIME	VERSIONS_ENDSCN	VERSIONS_ENDTIME	VERSIONS_XID	VERSIONS_OPERATION	SALARY
2884537	28-12-2022 11:59:41.			03000400A3080000	U	1800
2884537	28-12-2022 11:59:41.			03000400A3080000	U	2300
2884537	28-12-2022 11:59:41.			03000400A3080000	U	3300
2884537	28-12-2022 11:59:41.			03000400A3080000	U	4300
		2884537	28-12-2022 11:59:41.			1700
		2884537	28-12-2022 11:59:41.			2200
		2884537	28-12-2022 11:59:41.			3200
		2884537	28-12-2022 11:59:41.			4200

Flashback in Oracle

<u>QUESTION:- until when i can see these records ???</u>

This depend on many factors

- Tablespace type (fixed or auto extend)
- The retention period UNDO_RETENTION
- Retention Guarantee

<u>Flashback Database:</u>

- We can move an entire database back in time to a particular SCN or a timestamp. Flashback Database must be already enabled on the database to user this feature.

<u>Enable Flashback Database</u>

- Make sure <u>DB_RECOVERY_FILE_DEST</u> parameter is set. This is the location where Oracle will store flashback logs.

```
SQL> alter system set db_recovery_file_dest=' C:\spool\flash_logs' SCOPE=spfile;
```

- Set <u>DB_RECOVERY_FILE_DEST</u> parameter as per requirement

```
SQL> alter system set db_recovery_file_dest_size=50G SCOPE=spfile;
```

- Set the <u>DB_FLASHBACK_RETENTION_TARGET</u> parameter which specifies the upper limit (in minutes) on how far back in time the database can be flashed back

Flashback in Oracle

```
SQL> alter system set db_flashback_retention_target=2880;
```

- Enable flashback database which requires database bounce

```
SQL> shutdown immediate;
SQL> startup mount;
SQL> alter database flashback on;
SQL> alter database open;

SQL> select flashback_on from v$database;
```

Create Sample User

- Let us capture the database SCN number before we create a user

```
SQL> SELECT current_scn, SYSTIMESTAMP FROM v$database;
```

```
Current SCN: 2703232
```

- Create a user FLASH_USR and try to connect the database with same user

```
SQL> create user flash_usr identified by flash_usr;
SQL> grant connect, resource to flash_usr;
SQL> conn flash_usr/flash_usr;
```

Flashback Database to SCN or Timestamp

- Assume that the user has been created by mistake and you want to flashback database to the SCN just before the user creation. Shutdown DB and startup mount

```
SQL> shut immediate;
SQL> startup mount;
```

Flashback database to SCN before user creation and open database with resetlogs

```
SQL> Flashback database to scn 2703232;
SQL> Alter database open resetlogs;
```

- You can flashback database to particular timestamp too

```
FLASHBACK DATABASE TO TIMESTAMP
TO_TIMESTAMP(‘2016-05-12 18:30:00’, ‘YYYY-MM-DD HH24:MI:SS’);
```

Flashback in Oracle

LOG:-

```
Microsoft Windows [Version 10.0.22621.963]
(c) Microsoft Corporation. All rights reserved.

C:\Users\GSL-GGN-LT-05>cd\

C:\>sqlplus sys/syspwd@KINITH2 as sysdba

SQL*Plus: Release 12.1.0.2.0 Production on Fri Dec 30 13:14:49 2022

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> SELECT FLASHBACK_ON FROM V$DATABASE;
FLASHBACK_ON
------------------
NO

SQL> select * from v$flashback_database_log;
no rows selected

SQL> select * from v$flashback_database_stat;
no rows selected

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> startup mount;
ORACLE instance started.

Total System Global Area 2499805184 bytes
Fixed Size                  3048728 bytes
Variable Size             671091432 bytes
Database Buffers         1811939328 bytes
Redo Buffers               13725696 bytes
Database mounted.

SQL> alter database flashback on;
Database altered.

SQL> alter database open;
Database altered.
```

Flashback in Oracle

```
SQL> select flashback_on from v$database;

FLASHBACK_ON
------------------
YES

SQL> select * from v$flashback_database_log;

OLDEST_FLASHBACK_SCN OLDEST_FL RETENTION_TARGET FLASHBACK_SIZE ESTIMATED_FLASHBACK_SIZE
CON_ID
-------------------- --------- ---------------- --------------  ------------------------ ----------
 3124348          30-DEC-22   2880             104857600       0             0


SQL> select * from v$flashback_database_stat;

BEGIN_TIM END_TIME  FLASHBACK_DATA    DB_DATA  REDO_DATA ESTIMATED_FLASHBACK_SIZE     CON_ID
--------- --------- -------------- ---------- ----------  ------------------------ ----------
30-DEC-22 30-DEC-22       2113536    1671168     876032   0                       0
```

```
SQL> select * from v$flashback_database_logfile;

NAME                                                        LOG# THREAD#       SEQUENCE#       BYTES    FIRST_CHANGE#
             FIRST_TIME           TYPE      CON_ID
----------------------------------------------------------------------------  ---  ------  ----------  ---------  --------------     -----
----------      ----------- ---------
D:\ORACLE\DATABASE\FAST_RECOVERY_AREA\KINITH2\FLASHBACK\O1_MF_KTX3DLJ1_.FLB            1           1
             1               52428800      3124407              30/12/2022 12:34:51     NORMAL           0
D:\ORACLE\DATABASE\FAST_RECOVERY_AREA\KINITH2\FLASHBACK\O1_MF_KTX3DMLL_.FLB           2           1
             1               52428800       0                                  RESERVED          0


SQL> select * from v$flashback_database_log;

OLDEST_FLASHBACK_SCN OLDEST_FL RETENTION_TARGET FLASHBACK_SIZE ESTIMATED_FLASHBACK_SIZE
CON_ID
-------------------- --------- ---------------- -------------- ----------------------- ----------
```

```
3124348          30-DEC-22          2880     104857600    386678784            0


SQL> show user;
USER is "SYS"

SQL> SELECT current_scn, SYSTIMESTAMP FROM v$database;

CURRENT_SCN  SYSTIMESTAMP
-----------  ---------------------------------------------------------------------------
    3126032  30-DEC-22 12.58.08.524000 PM +05:30


SQL> create user flash_usr identified by flash_usr;
User created.

SQL> grant connect, resource to flash_usr;
Grant succeeded.

SQL> connect flash_usr/flash_usr@KINITH2
Connected.

SQL> show user;
USER is "FLASH_USR"

SQL> connect sys/syspwd@KINITH2 as sysdba
Connected.

SQL> shut immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount;
ORACLE instance started.

Total System Global Area 2499805184 bytes
Fixed Size                  3048728 bytes
Variable Size             671091432 bytes
Database Buffers         1811939328 bytes
Redo Buffers               13725696 bytes
Database mounted.


SQL> Flashback database to scn 3126032;
Flashback complete.

SQL> Alter database open resetlogs;
Database altered.

SQL> select username from dba_users;
```

```
USERNAME
--------------------------------------------------------------------------------
SCOTT
ORACLE_OCM
OJVMSYS
SYSKM
XS$NULL
GSMCATUSER
MDDATA
SYSBACKUP
DIP
SYSDG
APEX_PUBLIC_USER

USERNAME
--------------------------------------------------------------------------------
SPATIAL_CSW_ADMIN_USR
TEST
SPATIAL_WFS_ADMIN_USR
GSMUSER
AUDSYS
SCOTT2
FLOWS_FILES
DVF
MDSYS
ORDSYS
DBSNMP

USERNAME
--------------------------------------------------------------------------------
WMSYS
APEX_040200
APPQOSSYS
GSMADMIN_INTERNAL
ORDDATA
CTXSYS
ANONYMOUS
XDB
ORDPLUGINS
DVSYS
SI_INFORMTN_SCHEMA

USERNAME
--------------------------------------------------------------------------------
OLAPSYS
LBACSYS
OUTLN
SYSTEM
SYS
```

```
38 rows selected.

SQL> connect flash_usr/flash_usr@KINITH;
ERROR:
ORA-01017: invalid username/password; logon denied


Warning: You are no longer connected to ORACLE.
```