

Oracle 11g : Flashback Data Archive

Author	유현재
Creation Date	2010-05
Last Updated	2010-09
Version	1.0
Copyright(C) 2004 Goodus Inc. All Rights Reserved	

Version	변경일자	변경자(작성자)	주요내용
1	2010-05	유현재	문서 최초 작성
2	2010-09	유현재	1 차 문서 보완
3			

목차

1. Introduction	3
2. Oracle Flashback Technology.....	3
2.1. 이전 버전에서의 FLASHBACK.....	3
3. FlashBack Data Archive.....	3
3.1. 주요 장점	3
4. Architecture	4
5. Understanding flashback Data Archive	5
5.1. Requirements	5
5.2. Step by Step Approach.....	6
5.3. Querying History Data	8
5.4. Managing Flashback Data Archives.....	9
5.5. Monitoring Views	15
6. Recommended Best Practices	16
7. Conclusion	16

1. Introduction

오라클은 Oracle 9iR2 부터 Flashback Query 라는 일종의 타임머신 기능을 소개했다. 이 기능은 언두 세그먼트에서 변경 이전의 데이터를 추출해 내는 방법으로 동작하게 되는데, 10g 에서는 Flashback Versions Query 라는 기능도 제공되어 언두 정보가 존재하는 한 로우의 변경내역도 조회가 가능하게 되었다. 하지만 이러한 기능은 언두 정보가 리사이클 처리 되거나 언두 세그먼트의 공간확보를 위해 삭제되면 더 이상 사용 할 수 없다.

이러한 문제 때문에 많은 DBA 들이 우회적인 방법으로 변경내역을 DB 에 저장하는 방법 등을 이용했는데 11g 에서는 Flashback Data Archive 라는 새로운 기능을 제공한다. Flashback 쿼리의 단순성, 강력한 기능을 제공하는 동시에 리로 로그와 같은 비영구적인 저장 공간의 문제점을 해결하여 Flashback Data Archive 는 영구적인 저장 공간인 Flashback Recovery Area 에 변경 내역을 저장하게 된다.

2. Oracle Flashback Technology

2.1.이전 버전에서의 FLASHBACK

Version	Flashback Operation	Implementation
Oracle 9i	Flashback Query	Undo
Oracle 10g	Flashback Version Query	Undo
	Flashback Table	Undo
	Flashback Database	Flashback logs + Redo logs
Oracle 11g	Flashback Data Archive	flashback archive

여기서 한 가지 짚고 넘어갈 점은 Flashback table, Flashback Database, Flashback Drop, Flashback Version Query, Flashback Transaction Query는 아래의 표와 같이 각기 다른 영역을 사용한다는 점이다.

3. FlashBack Data Archive

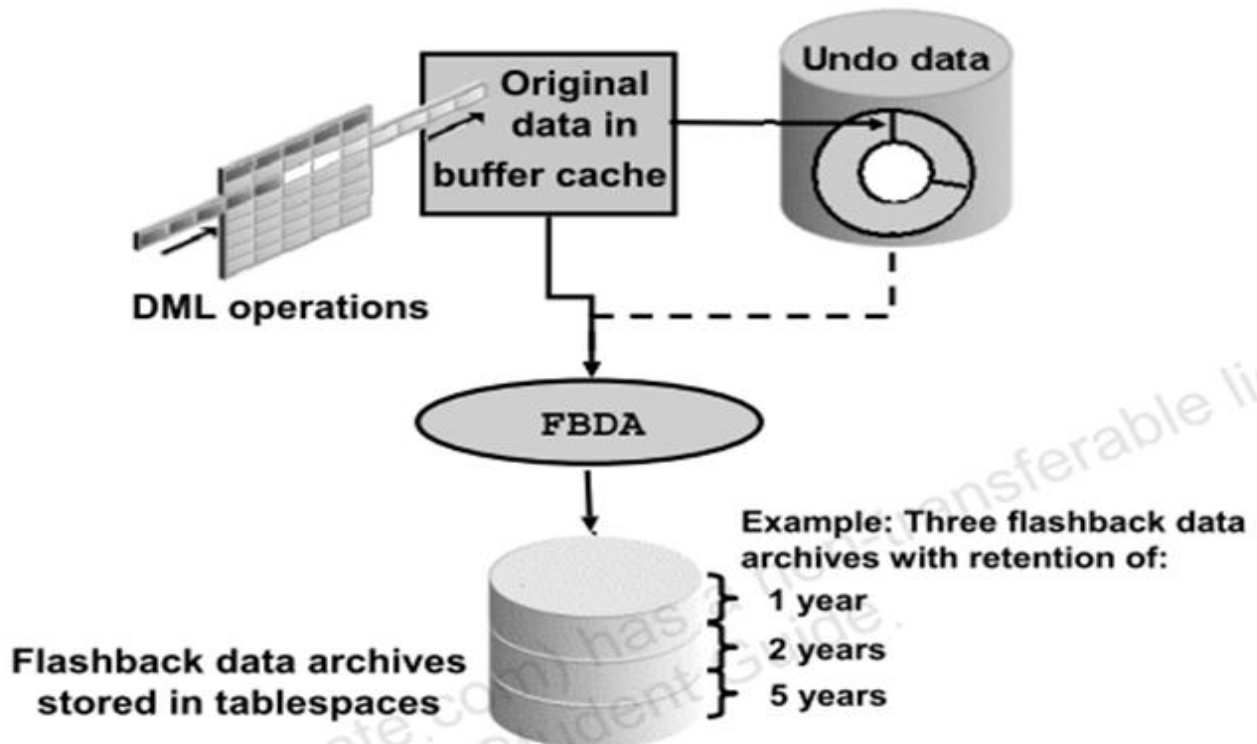
3.1.주요 장점

어플리케이션 이식성	하나 이상의 테이블에 대한 플래시백 데이터를 남길 때 어플리케이션에 대한 수정이 필요 없다.
데이터 접근 용이성	Flashback SQL 가 비슷한 구조의 쿼리를 통하여 저장된 기록에 쉽게 접근할 수 있다. 또한 저장된 플래시백 데이터 중에 특정 과거 시점의 데이터에 대한 조회도 가능하다.
보안	한번 생성된 플래시백 데이터는 모든 유저로부터 변경이 불가능하며, 내부 로그 테이블은 read only 로 접근가능하며 관리자를 포함한 모든 유저의 DML 문은 허용되지 않는다. 어플리케이션에서도 Flashback Query 외에 직접 쿼리를 날리는 것은 불가능하다.
DB 성능 최소 부하	백그라운드 프로세스를 통해서 비동기적으로 플래시백 데이터를 생성하는데, 통상적인 트랜잭션에서는 무시해도 좋을 정도의 작은 영향이 있다
스토리지 최적화	플래시백 데이터는 내부적으로 파티셔닝(range-partition) 되고 압축되어 스토리지 공간 활용을 최대화 시키는데 ,이런 작업들은 자동화 되어 있고 관리자의 작업이 필요 없다.

관리 용이성	대상 테이블에 대한 그룹핑과 보관정책을 지원하여, 관리자와 불필요한 관리업무를 줄여줍니다. 대부분의 작업을 자동화하여 수동관리에 대한 위험부담도 피한다
--------	--

4. Architecture

Flashback Data Archive



FBDA process

- 비동기적으로 row 캡처
- row업데이트가 아무리 많아도 해당테이블의 update는 모두 history table에 저장
- 최대 10개의 FBDA Slaves가 생성되어 FBDA를 도움
- fbda는 항상 떠 있는 것이 아니고, undo record 생성량에 따라 자동으로 sleep interval를 조정하는데 트랜잭션이 늘어나면 interval time이 줄어든다. Default time은 5분이다.
- 모든 트랜잭션이 아카이브 되는 것을 보장하기 위해서 모든 history가 생성되어 DB에 저장되기 전까지는 undo records는 recycle되지 않는다.

History Table

- 시간과 볼륨에 따라 자동 파티셔닝 되고, 압축되어 있다.
- Data retention time 조절 가능하며, retention time을 지나면 자동 삭제 된다.
- 언두 정보의 효율적인 저장과 검색
- 언제든지 가능한 row 조회하나, 변조는 불가능하다.
- 대상 테이블의 history data가 저장되기 전까지는 언두 정보는 recycle되지 않는다.
- Insert 는 history data 를 만들지 않는다.

5. Understanding flashback Data Archive

5.1. Requirements

- 자동 세그먼트 공간 관리 (ASSM)으로 관리되는 TABLESPACE
- 자동 언두 관리 (AUM)

```
SQL> select tablespace_name, status , extent_management, segment_space_management from dba_tablespaces;
```

TABLESPACE_NAME	STATUS	EXTENT_MAN	SEGMENT
SYSTEM	ONLINE	LOCAL	MANUAL
SYSAUX	ONLINE	LOCAL	AUTO
UNDOTBS1	ONLINE	LOCAL	MANUAL
TEMP	ONLINE	LOCAL	MANUAL
USERS	ONLINE	LOCAL	AUTO
PART01	ONLINE	LOCAL	AUTO
PART02	ONLINE	LOCAL	AUTO
PART03	ONLINE	LOCAL	AUTO
HJTEST	ONLINE	LOCAL	AUTO
FBDA	ONLINE	LOCAL	AUTO
AA	ONLINE	LOCAL	AUTO

11 rows selected.

```
SQL> show parameter undo
```

NAME	TYPE	VALUE
undo_management	string	AUTO
undo_retention	integer	900
Undo_tablespace	string	UNDOTBS1

5.2. Step by Step Approach

- 전체 테스트 진행순서
 - FDA history table 을 저장할 tablespaces 생성
 - 해당 유저에 필요 권한 부여
 - Flashback data archive (FDA) 생성
 - Table 단위로 FDA enable
 - 모니터링 및 query FDA

1) ASSM 방식의 테이블 스페이스를 생성하고 새로운 계정에 필요한 권한을 부여한다.

```
SQL> create tablespace fbda datafile '/oracle/oracle/noncloud/fbda01.dbf' size 100m segment  
spacemanagement auto;
```

Tablespace created.

```
SQL> create user fba identified by fba default tablespace fbda;
```

User created.

```
SQL> grant resource, connect to fba;
```

Grant succeeded.

```
SQL> grant flashback archive administer to fba;
```

Grant succeeded.

< privilege >

- FLASHBACK ARCHIVE ADMINISTER : FDA 를 관리하기 위한 새로운 system 권한
 - FDA 생성, 수정 , 삭제 가능
 - RETENTION set 가능
 - Tablespace add/remove
 - Disable tracking for tracked table

<grant >

- Grant system privileges to allow to create FDA

```
SQL> grant flashback archive adminster to FDA_ADMIN;' or SYSDBA is required.
```

- Grant object privileges to allow to create FDA

```
SQL> grant flashback archive on <FBDA_DFA> to <user>;
```

2) 새로운 flashback data archive 를 생성한다.

```
SQL> create flashback archive fda02 tablespace fbda [quota XX m ] retention 2 day;
```

Flashback archive created.

: fda02 라는 flashback archive 를 fbda tablespace 에 생성한다.

: TBS 에 대한 QUOTA 를 지정 안했기 때문에 디폴트로 제한이 없다. 보관주기를 2 일로 한다.

: QUOTA 는 M | G | T | P 로 지정 가능

: RETENTION 는 DAY | MONTH | YEAR 로 지정 가능

```
SQL> select owner, name, flashback_archive_name, retention_in_days, status from dba_flashback_archive;
```

OWNER_NAME	FLASHBACK_ARCHIVE_NAME	RETENTION_IN_DAYS
SYS	FDA02	2
FBA	FDA01	2 DEFAULT

3) 특정 테이블에 기능 활성화

- 테이블 생성 시 지정

- default flashback archive 가 지정 되어 있을 때, 안되어 있을 때로 방식이 나뉘어 진다.

```
SQL> alter flashback archive fda01 set default;
```

Flashback archive altered.

```
SQL> select flashback_archive_name, status from dba_flashback_archive
```

FLASHBACK_ARCHIVE_NAME	STATUS
FDA02	
FDA01	DEFAULT

```
SQL> create table t3
```

```
( name varchar2(20) ) flashback archive;
```

➔ Fda01 로 지정됨

```
SQL> create table t3
```

```
( name varchar2(20) ) flashback archive fda02;
```

- 기존에 존재하는 테이블에 설정할 때

```
SQL> alter table t1 flashback archive ; or alter table t1 flashback archive fda02;
```

Table altered

- Default flashback data archive 가 없을 때

```
SQL> alter table t1 flashback archive;
```

```
ORA-55608 : default flashback archive does not exist
```

5.3. Querying History Data

< DML History >

T2	10:52	11:00	11:19	11:35
Count(*)	1202	2404	2004	4008
DML		Insert 1202	Delete 400	Insert 2004

table 에서 DML 문이 일어나도 history table 에 바로 적용되지는 않는다.

USING timestamp

```
SQL> select count(*) from t2 as of timestamp to_timestamp ('2010-06-07 10:53','yyy-mm-dd hh24:mi');
```

COUNT(*)

1202

```
SQL> select count(*) from t2 as of timestamp to_timestamp ('2010-06-07 11:01','yyy-mm-dd hh24:mi');
```

COUNT(*)

1202

➔ T2 table 에 insert 작업 후 1 분 뒤에 history table 을 조회하였으나 아직 적용되지 않았다.

```
SQL> select count(*) from t2 as of timestamp to_timestamp ('2010-06-07 11:05','yyy-mm-dd hh24:mi');
```

COUNT(*)

2404

```
SQL> select count(*) from t2 as of timestamp to_timestamp ('2010-06-07 11:36','yyy-mm-dd hh24:mi');
```



```
COUNT(*)
```

```
-----  
2004
```

```
SQL> select count(*) from t2 as of timestamp to_timestamp ('2010-06-07 11:50','yyyy-mm-dd hh24:mi');
```

```
COUNT(*)
```

```
-----  
2004
```

→ Undo record 생성량에 따라 FBDA 의 sleep interval time 이 결정되기 때문에 35 분에 실행된 insert 작업에 대한 내용이 15 분 후에 archiving 되었다.

```
SQL> select count(*) from t2 as of timestamp to_timestamp ('2010-06-07 12:00','yyyy-mm-dd hh24:mi');
```

```
COUNT(*)
```

```
-----  
4008
```

USING SCN

```
SQL> select count(*) from t2 as of scn 100;
```

VERSIONS BETWEEN TIMESTAMP AND TIMESTAMP

```
SQL> select * from t2 versions between timestamp to_timestamp ('2010-06-07 11:50','yyyy-mm-dd hh24:mi')  
2 and to_timestamp ('2010-06-07 12:00','yyyy-mm-dd hh24:mi');
```

5.4.Managing Flashback Data Archives

● INTERNAL

```
SQL> select table_name, flashback_archive_name, archive_table_name from  
2 user_flashback_archive_tables  
3 where table_name='T2';
```

TABLE_NAME	FLASHBACK_ARCHIVE_NAME	ARCHIVE_TABLE_NAME
T2	FDA01	SYS_FBA_HIST_75787

SQL> select object_name , object_type, object_id from user_objects;

OBJECT_NAME	OBJECT_TYPE	OBJECT_ID
T2	TABLE	75787
SYS_FBA_DDL_COLMAP_75787	TABLE	75788
SYS_FBA_HIST_75787	TABLE PARTITION	75790
SYS_FBA_HIST_75787	TABLE	75789
SYS_FBA_TCRV_75787	TABLE	75793
SYS_FBA_TCRV_IDX_75787	INDEX	75794
T1	TABLE	74656

➔ FDA 가 enable 되어 있는 table 의 History table/index 확인

SQL> select owner, tablespace_name, table_name, temporary from dba_tables where table_name like '%FBA%';

(혹은 select table_name, from user_tables where table_name like '%FBA%';

OWNER	TABLESPACE_NAME	TABLE_NAME	T
RHJ		SYS_FBA_HIST_75787	N
SYS		SYS_MFBA_NHIST_75787	Y
SYS		SYS_MFBA_STAGE_RID	Y
SYS		SYS_MFBA_TRACKED_TXN	Y
SYS		SYS_MFBA_NROW	Y
SYS		SYS_MFBA_NCHANGE	Y
SYS		SYS_MFBA_NTCRV	Y
RHJ	FBDA	SYS_FBA_TCRV_75787	N
RHJ	FBDA	SYS_FBA_DDL_COLMAP_75787	N
SYS	SYSTEM	SYS_FBA_DL	N
SYS	SYSTEM	SYS_FBA_USERS	N
SYS	SYSTEM	SYS_FBA_PARTITIONS	N
SYS	SYSTEM	SYS_FBA_TRACKEDTABLES	N
SYS	SYSTEM	SYS_FBA_BARRIERSCN	N
SYS	SYSTEM	SYS_FBA_TSFA	N
SYS	SYSTEM	SYS_FBA_FA	N

SQL> select table_name, partition_name , compression from user_tab_partitions ;

TABLE_NAME	PARTITION_NAME	COMPRESS
------------	----------------	----------

```
SYS_FBA_HIST_75787          HIGH_PART          ENABLED
```

```
SQL> select table_name, partitioning_type, partition_count from user_part_tables;
```

```
TABLE_NAME                PARTITION PARTITION_COUNT
-----
SYS_FBA_HIST_75787        RANGE          1
```

● SPACE MANAGEMENT

<공간 추가>

```
SQL> alter flashback archive FDA01 add tablespace FBDA2 ;
```

Flashback archive altered.

```
SQL> select * from dba_flashback_archive_ts;
```

```
FLASHBACK_ARCHIVE_NAME    FLASHBACK_ARCHIVE# TABLESPACE_NAME
QUOTA_IN_MB
```

```
-----
FDA01                3 FBDA2
FDA02                    4 FBDA
FDA01                    3 FBDA
```

<공간 제거>

primary tbs 와 last tbs 는 제거할 수 없다.

```
SQL> alter flashback archive FDA01 remove tablespace FBDA2;
```

```
SQL> select * from dba_flashback_archive_ts;
```

```
FLASHBACK_ARCHIVE_NAME    FLASHBACK_ARCHIVE# TABLESPACE_NAME
QUOTA_IN_MB
```

```
-----
FDA01                3 FBDA2
FDA02                    4 FBDA
FDA01                    3 FBDA
```

➔ FBDA2 가 제거 되지 않았다.

```
SQL> alter flashback archive FDA01 remove tablespace FBDA;
```

```
alter flashback archive FDA01 remove tablespace FBDA
```

*

ERROR at line 1:

ORA-55626: Cannot remove the Flashback Archive's primary tablespace

➔ FBDA 역시 제거 되지 않았다.

SQL> alter flashback archive FDA01 add tablespace fbda3;

Flashback archive altered.

SQL> select * from dba_flashback_archive_ts;

FLASHBACK_ARCHIVE_NAME	FLASHBACK_ARCHIVE#	TABLESPACE_NAME	QUOTA_IN_MB
------------------------	--------------------	-----------------	-------------

FDA01	3	FBDA2	
FDA02	4	FBDA	
FDA01	3	FBDA	
FDA01	3	FBDA3	

SQL> alter flashback archive FDA01 remove tablespace FBDA2;

Flashback archive altered.

SQL>

SQL> select * from dba_flashback_archive_ts;

FLASHBACK_ARCHIVE_NAME	FLASHBACK_ARCHIVE#	TABLESPACE_NAME	QUOTA_IN_MB
------------------------	--------------------	-----------------	-------------

FDA02	4	FBDA	
FDA01	3	FBDA	
FDA01	3	FBDA3	

➔ 세 번 째 TBS 를 추가하고 2 번째 TBS 삭제를 시도하니 삭제된다.

➔ Tablespace 안에 history table 이 존재한다면 retention 값을 지나지 않은 data 가 있다고 하더라도 같이 삭제된다.

QUOTA 와 retention 값을 조절하여 공간사용을 줄일 수 있다.

SQL> alter flashback archive fda01 modify tablespace FBDA3 quota 5m;

Flashback archive altered.

```
SQL> alter flashback archive fda01 modify retention 1 day;
```

Flashback archive altered.

삭제하는 경우

```
SQL> drop flashback archive fda02;
```

Flashback archive dropped.

할당받은 공간이나 테이블스페이스를 모두 사용했을 때

- alert log 에 ORA-55617 (tbs) 나 ORA-55623 (tbs quota) 를 발생 시키며 archiving 과 tracking 을 중단한다.
- tbs 안에 해당 flashback archive 의 Quota 를 변경해주거나 해당 tbs datafile resize or add datafile 작업으로 해결한다.

● purging

- Retention 값이 지나 하루가 지나면 자동으로 purge 된다.
- 아래 세가지 옵션 사용하여 수동으로 purging 작업이 가능하다.
- PURGE ALL | PURGE BEFORE SCN | PURGE BEFORE TIMESTAMP

수동으로 purging 작업을 하기 위해서는 FLASHBACK ADMINISTER 권한이 있어야 한다.

```
SQL> ALTER FLASHBACK ARCHIVE FDA01 PURGE ALL;
```

하루 지난 히스토리 데이터를 모두 PURGE 시키기

```
SQL> ALTER FLASHBACK ARCHIVE FDA01 purge before timestamp ( systimestamp - interval '1' day ) ;
```

SCN 기준으로 삭제

```
SQL> ALTER FLASHBACK ARCHIVE FDA01 PURGE BEFORE SCN 123412
```

● DDL

<지원하는 DDL>

- Adds | drops | renames | modifies a Column
- Adds | drops | renames a Constraint
- Drop | Truncates a Partition or Subpartition
- TRUNCATE TABLE
- RENAME TABLE
- PARTITION OR SUBPARTITION

<지원하지 않는 DDL>

- 파티션이나 서브파티션을 변경하는 ALTER TABLE 구문
- DROP TABLE
-

< 11g R1 과 R2 에서 지원되는 DDL 이 다르다. >

	11g R1	11g R2
Truncate	X	O
Alter table	X	O
Drop	X	X

11gR2 에서 test 를 진행함

SQL> desc t2;

```
Name                Null?    Typ
-----
C1                                NUMBER
C2                                NUMBER
C3                                DATE
C4                                NUMBER
```

SQL> alter table t2 add (name varchar2(10));

Table altered.

SQL> desc t2;

```
Name
Null?    Type
-----
C1                                NUMBER
C2                                NUMBER
C3                                DATE
C4                                NUMBER
NAME                                VARCHAR2(10)
```

SQL> alter table t2 drop (name);

Table altered.

SQL> desc t2;

```
Name
```

Null?	Type
C1	NUMBER
C2	NUMBER
C3	DATE
C4	NUMBER

SQL>

SQL> drop table t2;

drop table t2

*

ERROR at line 1:

ORA-55610: Invalid DDL statement on history-tracked table

SQL> truncate table t2;

Table truncated.

-

SQL> alter table t2 shrink space;

alter table t2 shrink space

*

ERROR at line 1:

ORA-55610: Invalid DDL statement on history-tracked table

5.5 Monitoring Views

SQL> desc dba_flashback_archive_ts

Name	Null?	Type
FLASHBACK_ARCHIVE_NAME	NOT NULL	VARCHAR2(255)
FLASHBACK_ARCHIVE#	NOT NULL	NUMBER
TABLESPACE_NAME	NOT NULL	VARCHAR2(30)
QUOTA_IN_MB		VARCHAR2(40)

SQL> desc dba_flashback_archive

Name	Null?	Type
OWNER_NAME		VARCHAR2(30)
FLASHBACK_ARCHIVE_NAME	NOT NULL	VARCHAR2(255)

FLASHBACK_ARCHIVE#	NOT NULL NUMBER
RETENTION_IN_DAYS	NOT NULL NUMBER
CREATE_TIME	TIMESTAMP(9)
LAST_PURGE_TIME	TIMESTAMP(9)
STATUS	VARCHAR2(7)

SQL> desc dba_flashback_archive_tables

Name	Null?	Type
TABLE_NAME	NOT NULL	VARCHAR2(30)
OWNER_NAME	NOT NULL	VARCHAR2(30)
FLASHBACK_ARCHIVE_NAME	NOT NULL	VARCHAR2(255)
ARCHIVE_TABLE_NAME		VARCHAR2(53)
STATUS		VARCHAR2(8)

6. Recommended Best Practices

- **Internal history table에 대한 INDEX를 생성**

- Flashback data archive는 자동으로 basic table의 index를 internal history table로 복제하지 않는다.
 - DBA/USER_FLASHBACK_ARCHIVE_TABLE view를 조회하여 internal history table에 대한 index 생성
- ➔ 프로세스가 트랜잭션 change를 수집하고 archiving 하는데 성능 상의 많은 영향을 미친다.

- **하나의 FLASHBACK DATA ARCHIVE 당 하나의 TABLESPACE를 정하여 사용**

- QUOTA 옵션은 사용하지 않고 default 옵션인 unlimited를 유지
 - 각각의 retention 값으로 각 Flashback archive data을 관리
- ➔ 공간관리의 계획과 유지가 용이해진다.

- **Flashback archive administer 권한에 대한 접근 제한**

- ➔ Object 권한인 flashback archive 권한과 분리하여 사용한다.

7. Conclusion

Oracle Database 11g 는 아카이브/온라인 리두 로그의 비영구성이라는 문제를 해결하고 Flashback Archive 에 변경 내역을 영구적으로 저장할 수 있는 기능을 추가로 제공하고 있다. 이제 Flashback Recovery Area 기능이 제공하는 간단한 몇 가지 명령만으로 전체 테이블의 변경 내역을 관리하고 복구할 수 있게 되었다.