– ALERT LOG FILE

데이터베이스의 Alert File에는 다음과 같은 메시지가 시간 순으로 기록되어 있습니다.

• 시작 시 사용된 기본값이 아닌 초기화 파라미터

• 발생한 모든 내부 오류(ORA-600), 블록 손상 오류(ORA-1578) 및 deadlock 오류(ORA-60)

• CREATE, ALTER, DROP DATABASE 및 TABLESPACE SQL 문과 STARTUP, SHUTDOWN,

ARCHIVE LOG 및 RECOVER Enterprise Manager 또는 SQL\*Plus 문과 같은 관리 작업

• Shared Server 및 디스패처 프로세스의 기능과 관련된 일부 메시지 및 오류

• Materialized View의 Automatic Refresh 중에 발생한 오류

ALERT LOG FILE 저장위치 확인

[oracle@edydr1p1 dbhome\_1]$ sqlplus / as sysdba

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> col name format a15

SQL> col value format a45

SQL> select name,value from v$diag\_info;

NAME VALUE

--------------- ---------------------------------------------

Diag Enabled TRUE

ADR Base /u01/app/oracle

ADR Home /u01/app/oracle/diag/rdbms/orcl/orcl

Diag Trace /u01/app/oracle/diag/rdbms/orcl/orcl/trace

Diag Alert /u01/app/oracle/diag/rdbms/orcl/orcl/alert

Diag Incident /u01/app/oracle/diag/rdbms/orcl/orcl/incident

Diag Cdump /u01/app/oracle/diag/rdbms/orcl/orcl/cdump

Health Monitor /u01/app/oracle/diag/rdbms/orcl/orcl/hm

Default Trace F /u01/app/oracle/diag/rdbms/orcl/orcl/trace/orilecl\_ora\_9510.trc

Active Problem 0

Count

Active Incident 0

Count

11 rows selected.

Elapsed: 00:00:00.02

SQL> exit

Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production

With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

[oracle@edydr1p1 dbhome\_1]$ cd /u01/app/oracle/diag/rdbms/orcl/orcl/trace

[oracle@edydr1p1 trace]$ ls alert\*

alert\_orcl.log

[oracle@edydr1p1 trace]$ gedit alert\_orcl.log

[oracle@edydr1p1 trace]$ cd /u01/app/oracle/diag/rdbms/orcl/orcl/alert

[oracle@edydr1p1 alert]$ ls log.xml

log.xml

[oracle@edydr1p1 alert]$ gedit log.xml

[oracle@edydr1p1 alert]$

<< TRACE FILE 조회 >>

[oracle@edydr1p1 alert]$ cd /u01/app/oracle/diag/rdbms/orcl/orcl/trace

[oracle@edydr1p1 trace]$ ps -ef|grep orcl

oracle 8950 1 0 16:37 ? 00:00:00 ora\_pmon\_orcl

oracle 8952 1 0 16:37 ? 00:00:01 ora\_psp0\_orcl

oracle 8954 1 1 16:37 ? 00:00:43 ora\_vktm\_orcl

oracle 8958 1 0 16:37 ? 00:00:00 ora\_gen0\_orcl

oracle 8960 1 0 16:37 ? 00:00:00 ora\_mman\_orcl

oracle 8964 1 0 16:37 ? 00:00:00 ora\_diag\_orcl

oracle 8966 1 0 16:37 ? 00:00:00 ora\_dbrm\_orcl

oracle 8968 1 0 16:37 ? 00:00:00 ora\_vkrm\_orcl

oracle 8970 1 0 16:37 ? 00:00:02 ora\_dia0\_orcl

oracle 8972 1 0 16:37 ? 00:00:00 ora\_dbw0\_orcl

oracle 8974 1 0 16:37 ? 00:00:00 ora\_lgwr\_orcl

oracle 8976 1 0 16:37 ? 00:00:01 ora\_ckpt\_orcl

oracle 8978 1 0 16:37 ? 00:00:00 ora\_lg00\_orcl

oracle 8980 1 0 16:37 ? 00:00:00 ora\_smon\_orcl

oracle 8982 1 0 16:37 ? 00:00:00 ora\_lg01\_orcl

oracle 8984 1 0 16:37 ? 00:00:00 ora\_reco\_orcl

oracle 8986 1 0 16:37 ? 00:00:00 ora\_lreg\_orcl

oracle 8988 1 0 16:37 ? 00:00:00 ora\_pxmn\_orcl

oracle 8990 1 0 16:37 ? 00:00:03 ora\_mmon\_orcl

oracle 8992 1 0 16:37 ? 00:00:02 ora\_mmnl\_orcl

oracle 8994 1 0 16:37 ? 00:00:00 ora\_d000\_orcl

oracle 8996 1 0 16:37 ? 00:00:00 ora\_s000\_orcl

oracle 9009 1 0 16:37 ? 00:00:00 ora\_tmon\_orcl

oracle 9011 1 0 16:37 ? 00:00:00 ora\_tt00\_orcl

oracle 9013 1 0 16:37 ? 00:00:00 ora\_smco\_orcl

oracle 9015 1 0 16:37 ? 00:00:00 ora\_w000\_orcl

oracle 9019 1 0 16:37 ? 00:00:00 ora\_aqpc\_orcl

oracle 9021 1 0 16:37 ? 00:00:01 ora\_cjq0\_orcl

oracle 9025 1 0 16:37 ? 00:00:00 ora\_p000\_orcl

oracle 9027 1 0 16:37 ? 00:00:00 ora\_p001\_orcl

oracle 9029 1 0 16:37 ? 00:00:00 ora\_p002\_orcl

oracle 9031 1 0 16:37 ? 00:00:00 ora\_p003\_orcl

oracle 9033 1 0 16:37 ? 00:00:00 ora\_p004\_orcl

oracle 9035 1 0 16:37 ? 00:00:00 ora\_p005\_orcl

oracle 9037 1 0 16:37 ? 00:00:00 ora\_p006\_orcl

oracle 9039 1 0 16:37 ? 00:00:00 ora\_p007\_orcl

oracle 9184 1 0 16:38 ? 00:00:00 ora\_qm02\_orcl

oracle 9188 1 0 16:38 ? 00:00:00 ora\_q002\_orcl

oracle 9190 1 0 16:38 ? 00:00:00 ora\_q003\_orcl

oracle 9468 1 0 17:05 ? 00:00:00 ora\_w002\_orcl

oracle 9574 7123 0 17:16 pts/0 00:00:00 grep orcl

[oracle@edydr1p1 trace]$ ls \*8974\*

orcl\_lgwr\_8974.trc orcl\_lgwr\_8974.trm

[oracle@edydr1p1 trace]$

<<DDL LOG 실습>>

[oracle@edydr1p1 ~]$ sqlplus / as sysdba

SQL> show parameter ddl

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

ddl\_lock\_timeout integer 0

enable\_ddl\_logging boolean FALSE

SQL> alter system set enable\_ddl\_logging=true

2 /

System altered.

Elapsed: 00:00:00.00

SQL> show parameter ddl

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

ddl\_lock\_timeout integer 0

enable\_ddl\_logging boolean TRUE

SQL> exit

[oracle@edydr1p1 log]$ sqlplus hr/oracle

SQL> create table ttt (id number)

2 /

Table created.

SQL> exit

[oracle@edydr1p1 ~]$ cd /u01/app/oracle/diag/rdbms/orcl/orcl/log

[oracle@edydr1p1 log]$ ls

ddl ddl\_orcl.log debug imdb test

[oracle@edydr1p1 log]$ cat ddl\_orcl.log

Mon Dec 20 19:13:50 2021

diag\_adl:create table ttt (id number)

[oracle@edydr1p1 ~]$ sqlplus / as sysdba

SQL> show parameter ddl

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

ddl\_lock\_timeout integer 0

enable\_ddl\_logging boolean TRUE

SQL> alter system set enable\_ddl\_logging=FALSE

2 /

System altered.

SQL> show parameter ddl

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

ddl\_lock\_timeout integer 0

enable\_ddl\_logging boolean FALSE

SQL> exit

DYNAMIC PERFORMANCE VIEW (동적성능 뷰)

: 현재 데이터베이스에서 실행되고 있는 모든 정보(SQL, 사용자 SESSION정보 등)을 실시간으로 조회 가능

: SGA 메모리의 SHARED POOL에 만들어져서 DB운영중에 정보가 실시간으로 계속 업데이트됨.

<< DATA DICTIONARY>>

오라클 데이터 딕셔너리는 데이터베이스에 대한 시스템 데이터를 저장하고 있는 테이블과 뷰

데이터 딕셔너리 테이블들(TAB$, AUD$ ..)에 기본 시스템정보들이 SYSTEM 테이블스페이스에 저장관리된다 SYS USER에 의해서 자동 관리한다.

데이터베이스에 있는 모든 객체의 정보를 포함합니다.

객체를 생성하거나 수정하면 해당 변경 사항을 반영하는

데이터 딕셔너리가 자동 갱신됩니다. 이 정보는 오라클 데이터베이스 서버에 의해 유지 관리되는

기본 테이블(base table, 데이터 딕셔너리 테이블들(TAB$, AUD$ ..))에 저장되지만

테이블을 직접 읽는 대신 미리 정의된 뷰(데이터 딕셔너리 뷰)를 사용하여

이러한 테이블에 액세스할 수 있습니다.

<< LISTENER >>

1. INSTANCE하고는 독립적으로 운영 (LSNRCTL 유틸리티)

2. REMOTE(원격)으로 데이터베이스에 연결하는 사용자 프로세스들은 반드시 LISTENER를 통해서 INSTANCE와 연결 가능

단, LOCAL(로컬)로 연결하는 사용자 프로세스들은 반드시 LISTENER가 필요하지 않다.

3. 모든 데이터베이스는 설치시 기본 리스너 한 개를 자동

생성하고 사용하게 된다 이 때 기본 리스너의 이름은 LISTENER가 된다.

4. LISTENER PROCESS는 필요하면 여러 개를 추가 생성해서 사용할 수 있다.

5. 하나의 LISTENER PROCESS는 여러 INSTANCE와 연결 가능하다.

6. 하나의 INSTANCE에는 여러 개의 SERVICE가 셋업될 수 있다.

7. 결국에는 USER PROCESS가 어떤 리스너를 사용하고 어떤 SERVICE를 이용해서

INSTANCE에 연결할 것인지를 결정할 수 있다.

추가 리스너가 필요한 이유

1. 백업 리스너의 역할 (FAIL OVER)

2. LOAD BALANCING (사용자 연결 요청을 분산 관리)

LISTENER에 SERVICE정보가 등록이 되어야 사용자들이 LISTENER를 통해 SERVICE정보를 통해

INSTANCE에 연결 가능.

LISTENER에 INSTANCE가 SERVICE로 자동 등록되는 경우

기본 LISTENER를 사용하고 있으면 해당 LISTENER를 포함하는 시스템에 같이 존재하는 INSTANCE의 경우에는INSTANCE가 시작이 될 때 LREG 백그라운드 프로세스가 자동으로 리스너에 서비스를 등록해 준다.만약 INSTANCE가 먼저 STARTUP이 되어 있고 이후에 기본 LISTENER가 시작이 된 경우 일정한 INTERVAL시간이 지나면 역시 LREG를 통해 현재의 INSTANCE는 SERVICE로 등록이 자동으로 된다.

추가 LISTENER의 경우에는 서비스정보가 자동등록이 안되기 때문에

수동으로 SERVICE정보를 등록해야 한다.

<< 기본 리스너 환경 확인 >>

[oracle@edydr1p1 ~]$ cd $ORACLE\_HOME/network/admin

[oracle@edydr1p1 admin]$ ls

listener.ora samples shrept.lst sqlnet.ora tnsnames.ora

[oracle@edydr1p1 admin]$ cat listener.ora

# listener.ora Network Configuration File: /u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/listener.ora

# Generated by Oracle configuration tools.

LISTENER =

(DESCRIPTION\_LIST =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = edydr1p1.us.oracle.com)(PORT = 1521))

(ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))

)

)

[oracle@edydr1p1 admin]$ lsnrctl status

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edydr1p1.us.oracle.com)(PORT=1521)))

TNS-12541: TNS:no listener

TNS-12560: TNS:protocol adapter error

TNS-00511: No listener

Linux Error: 111: Connection refused

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))

TNS-12541: TNS:no listener

TNS-12560: TNS:protocol adapter error

TNS-00511: No listener

Linux Error: 111: Connection refused

[oracle@edydr1p1 admin]$ lsnrctl start

Copyright (c) 1991, 2014, Oracle. All rights reserved.

Starting /u01/app/oracle/product/12.1.0/dbhome\_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.2.0 - Production

System parameter file is /u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/listener.ora

Log messages written to /u01/app/oracle/diag/tnslsnr/edydr1p1/listener/alert/log.xml

Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edydr1p1.us.oracle.com)(PORT=1521)))

Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edydr1p1.us.oracle.com)(PORT=1521)))

STATUS of the LISTENER

------------------------

Alias LISTENER

Version TNSLSNR for Linux: Version 12.1.0.2.0 - Production

Start Date 26-MAR-2022 10:34:11

Uptime 0 days 0 hr. 0 min. 0 sec

Trace Level off

Security ON: Local OS Authentication

SNMP OFF

Listener Parameter File /u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/listener.ora

Listener Log File /u01/app/oracle/diag/tnslsnr/edydr1p1/listener/alert/log.xml

Listening Endpoints Summary...

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edydr1p1.us.oracle.com)(PORT=1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

The listener supports no services

The command completed successfully

[oracle@edydr1p1 admin]$ lsnrctl status

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edydr1p1.us.oracle.com)(PORT=1521)))

STATUS of the LISTENER

------------------------

Alias LISTENER

Version TNSLSNR for Linux: Version 12.1.0.2.0 - Production

Start Date 26-MAR-2022 10:34:11

Uptime 0 days 0 hr. 0 min. 57 sec

Trace Level off

Security ON: Local OS Authentication

SNMP OFF

Listener Parameter File /u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/listener.ora

Listener Log File /u01/app/oracle/diag/tnslsnr/edydr1p1/listener/alert/log.xml

Listening Endpoints Summary...

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edydr1p1.us.oracle.com)(PORT=1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edydr1p1.us.oracle.com)(PORT=5500))(Presentation=HTTP)(Session=RAW))

Services Summary...

Service "orcl" has 1 instance(s).

Instance "orcl", status READY, has 1 handler(s) for this service...

Service "orclXDB" has 1 instance(s).

Instance "orcl", status READY, has 1 handler(s) for this service...

The command completed successfully

[oracle@edydr1p1 admin]$ sqlplus / as sysdba

SQL\*Plus: Release 12.1.0.2.0 Production on Sat Mar 26 10:35:45 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> show parameter service\_name

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

service\_names string orcl

SQL> EXIT

<< 리스너를 통해 INSTANCE에 연결하는 예 >>

[oracle@edydr1p1 admin]$ cd $ORACLE\_HOME/network/admin

[oracle@edydr1p1 admin]$ ls

listener.ora samples shrept.lst sqlnet.ora tnsnames.ora

[oracle@edydr1p1 admin]$ cat tnsnames.ora

# tnsnames.ora Network Configuration File: /u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/tnsnames.ora

# Generated by Oracle configuration tools.

EM12REP =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = edydr1p1.us.oracle.com)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = em12rep)

)

)

ORCL =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = edydr1p1.us.oracle.com)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = orcl)

)

)

Tnsnames.ora 파일을 아래와 같이 편집기로 열어서 연결내용을 추가

[oracle@edydr1p1 admin]$ gedit tnsnames.ora

# tnsnames.ora Network Configuration File: /u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/tnsnames.ora

# Generated by Oracle configuration tools.

EM12REP =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = edydr1p1.us.oracle.com)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = em12rep)

)

)

ORCL =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = edydr1p1.us.oracle.com)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = orcl)

)

)

**A =**

**(DESCRIPTION =**

**(ADDRESS = (PROTOCOL = TCP)(HOST = edydr1p1.us.oracle.com)(PORT = 1521))**

**(CONNECT\_DATA =**

**(SERVER = DEDICATED)**

**(SERVICE\_NAME = orcl)**

**)**

**)**

**파일을 저장하고 exit하고 아래내용을 실행**

[oracle@edydr1p1 admin]$ sqlplus hr/oracle@A

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>

SQL>exit

Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production

With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

[oracle@edydr1p1 admin]$ lsnrctl stop

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edydr1p1.us.oracle.com)(PORT=1521)))

The command completed successfully

[oracle@edydr1p1 admin]$ lsnrctl status

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edydr1p1.us.oracle.com)(PORT=1521)))

TNS-12541: TNS:no listener

TNS-12560: TNS:protocol adapter error

TNS-00511: No listener

Linux Error: 111: Connection refused

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))

TNS-12541: TNS:no listener

TNS-12560: TNS:protocol adapter error

TNS-00511: No listener

Linux Error: 2: No such file or directory

[oracle@edydr1p1 admin]$ sqlplus hr/oracle@A

ERROR:

ORA-12541: TNS:no listener

Enter user-name: ^C

[oracle@edydr1p1 admin]$ sqlplus hr/oracle

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>

SQL> exit

Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production

With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

[oracle@edydr1p1 admin]$ lsnrctl start

Starting /u01/app/oracle/product/12.1.0/dbhome\_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.2.0 - Production

System parameter file is /u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/listener.ora

Log messages written to /u01/app/oracle/diag/tnslsnr/edydr1p1/listener/alert/log.xml

Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edydr1p1.us.oracle.com)(PORT=1521)))

Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edydr1p1.us.oracle.com)(PORT=1521)))

STATUS of the LISTENER

------------------------

Alias LISTENER

Version TNSLSNR for Linux: Version 12.1.0.2.0 - Production

Start Date 26-MAR-2022 10:44:52

Uptime 0 days 0 hr. 0 min. 0 sec

Trace Level off

Security ON: Local OS Authentication

SNMP OFF

Listener Parameter File /u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/listener.ora

Listener Log File /u01/app/oracle/diag/tnslsnr/edydr1p1/listener/alert/log.xml

Listening Endpoints Summary...

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edydr1p1.us.oracle.com)(PORT=1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

The listener supports no services

The command completed successfully

[oracle@edydr1p1 admin]$

<< 간단한 연결(Easy Connect) >>

TNSNAMES.ORA 파일이 필요없이 아래와 같이 리스너로 연결가능

[oracle@edydr1p1 admin]$ sqlplus hr/oracle@edydr1p1.us.oracle.com:1521/orcl

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>

SQL>EXIT

<<추가 리스너 셋업 예제>>

[oracle@edydr1p1 admin]cd $ORACLE\_HOME/network/admin

[oracle@edydr1p1 admin]$ gedit listener.ora

# listener.ora Network Configuration File: /u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/listener.ora

# Generated by Oracle configuration tools.

**LIS2 =**

**(DESCRIPTION =**

**(ADDRESS = (PROTOCOL = TCP)(HOST = edydr1p1.us.oracle.com)(PORT = 1522))**

**)**

**ADR\_BASE\_LIS2 = /u01/app/oracle**

LISTENER =

(DESCRIPTION\_LIST =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = edydr1p1.us.oracle.com)(PORT = 1521))

)

(DESCRIPTION =

(ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))

)

)

ADR\_BASE\_LISTENER = /u01/app/oracle

**SID\_LIST\_LIS2 =**

**(SID\_LIST =**

**(SID\_DESC =**

**(GLOBAL\_DBNAME = orcl)**

**(ORACLE\_HOME = /u01/app/oracle/product/12.1.0/dbhome\_1)**

**(SID\_NAME = orcl)**

**)**

**)**

[oracle@edydr1p1 admin]$ lsnrctl start lis2

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 20-DEC-2021 21:01:52

Copyright (c) 1991, 2014, Oracle. All rights reserved.

Starting /u01/app/oracle/product/12.1.0/dbhome\_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.2.0 - Production

System parameter file is /u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/listener.ora

Log messages written to /u01/app/oracle/diag/tnslsnr/edydr1p1/lis2/alert/log.xml

Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edydr1p1.us.oracle.com)(PORT=1522)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edydr1p1.us.oracle.com)(PORT=1522)))

STATUS of the LISTENER

------------------------

Alias lis2

Version TNSLSNR for Linux: Version 12.1.0.2.0 - Production

Start Date 20-DEC-2021 21:01:52

Uptime 0 days 0 hr. 0 min. 0 sec

Trace Level off

Security ON: Local OS Authentication

SNMP OFF

Listener Parameter File /u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/listener.ora

Listener Log File /u01/app/oracle/diag/tnslsnr/edydr1p1/lis2/alert/log.xml

Listening Endpoints Summary...

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edydr1p1.us.oracle.com)(PORT=1522)))

Services Summary...

Service "orcl" has 1 instance(s).

Instance "orcl", status UNKNOWN, has 1 handler(s) for this service...

The command completed successfully

[oracle@edydr1p1 admin]$ gedit tnsnames.ora

# tnsnames.ora Network Configuration File: /u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/tnsnames.ora

# Generated by Oracle configuration tools.

EM12REP =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = edydr1p1.us.oracle.com)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = em12rep)

)

)

ORCL =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = edydr1p1.us.oracle.com)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = orcl)

)

)

A =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = edydr1p1.us.oracle.com)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = orcl)

)

)

**B =**

**(DESCRIPTION =**

**(ADDRESS = (PROTOCOL = TCP)(HOST = edydr1p1.us.oracle.com)(PORT = 1522))**

**(CONNECT\_DATA =**

**(SERVER = DEDICATED)**

**(SERVICE\_NAME = orcl)**

**)**

**)**

[oracle@edydr1p1 admin]$ sqlplus hr/oracle**@B**

Last Successful login time: Mon Dec 20 2021 20:38:16 +09:00

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>

SQL> EXIT

[oracle@edydr1p1 admin]$ lsnrctl stop lis2

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 20-DEC-2021 21:03:47

Copyright (c) 1991, 2014, Oracle. All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edydr1p1.us.oracle.com)(PORT=1522)))

The command completed successfully

[[oracle@edydr1p1 admin]$ sqlplus hr/oracle@B

SQL\*Plus: Release 12.1.0.2.0 Production on Mon Dec 20 21:04:16 2021

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

ERROR:

ORA-12541: TNS:no listener

Enter user-name:

<< EASY CONNECTION 예제>>

[oracle@edydr1p1 admin]$ lsnrctl start lis2

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 20-DEC-2021 21:19:47

Copyright (c) 1991, 2014, Oracle. All rights reserved.

Starting /u01/app/oracle/product/12.1.0/dbhome\_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.2.0 - Production

System parameter file is /u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/listener.ora

Log messages written to /u01/app/oracle/diag/tnslsnr/edydr1p1/lis2/alert/log.xml

Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edydr1p1.us.oracle.com)(PORT=1522)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edydr1p1.us.oracle.com)(PORT=1522)))

STATUS of the LISTENER

------------------------

Alias lis2

Version TNSLSNR for Linux: Version 12.1.0.2.0 - Production

Start Date 20-DEC-2021 21:19:47

Uptime 0 days 0 hr. 0 min. 0 sec

Trace Level off

Security ON: Local OS Authentication

SNMP OFF

Listener Parameter File /u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/listener.ora

Listener Log File /u01/app/oracle/diag/tnslsnr/edydr1p1/lis2/alert/log.xml

Listening Endpoints Summary...

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edydr1p1.us.oracle.com)(PORT=1522)))

Services Summary...

Service "orcl" has 1 instance(s).

Instance "orcl", status UNKNOWN, has 1 handler(s) for this service...

The command completed successfully

[oracle@edydr1p1 admin]$ sqlplus hr/oracle@edydr1p1.us.oracle.com:1522/orcl

SQL\*Plus: Release 12.1.0.2.0 Production on Mon Dec 20 21:20:02 2021

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

Last Successful login time: Mon Dec 20 2021 21:03:21 +09:00

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>

SQL> EXIT

<< TNSNAME이 2개이상의 리스너를 가리킬 수 있다>>

[oracle@edydr1p1 admin]$ gedit tnsnames.ora

# tnsnames.ora Network Configuration File: /u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/tnsnames.ora

# Generated by Oracle configuration tools.

B =

(DESCRIPTION =

(ADDRESS\_LIST =

(ADDRESS = (PROTOCOL = TCP)(HOST = edydr1p1.us.oracle.com)(PORT = 1522))

(ADDRESS = (PROTOCOL = TCP)(HOST = edydr1p1.us.oracle.com)(PORT = 1521))

)

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = orcl)

)

)

<< DB 계정 생성 예>>

CREATE USER USERA IDENTIFIED BY oracle

DEFAULT TABLESPACE USERS

QUOTA 50M ON USERS

QUOTA 50M ON EXAMPLE

TEMPORARY TABLESPACE TEMP – SORTING작업시 사용하는 임시공간

PROFILE DEFAULT

ACCOUNT LOCK – 계정을 잠긴 상태로 생성, ALTER USER USERA ACCOUNT UNLOCK; (LOCK을 풀어주는 명령)

PASSWORD EXPIRE; – 최초 로그인시 암호를 변경하도록 셋업

SELECT문에 ORDER BY절을 사용하면

조회결과가 PGA메모리로 올라와서 정렬 작업을 진행하는데

PGA메모리 영역이 부족한 경우에는 일부 데이터가 TEMPORARY TABLESPACE로 내려와서 정렬작업을

임시로 진행하게 된다. 정렬작업이 끝나면 결과는 사용자에게 보내어지고 지워진다.

<< SYS, SYSBACKUP USER 권한 정보 조회>>

[oracle@edydr1p1 admin]$ sqlplus / as sysdba

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> show user

USER is "SYS"

SQL> select count(\*) from session\_privs;

COUNT(\*)

----------

234

1 row selected.

SQL> exit

[oracle@edydr1p1 admin]$ sqlplus / as sysbackup

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> show user

USER is "SYSBACKUP"

SQL> select count(\*) from session\_privs;

COUNT(\*)

----------

14

1 row selected.

SQL>

<< PASSWORD FILE>>

PASSWORD FILE에는 기본적으로 관리자 권한을 가지는 DB계정 정보가 저장된다.

기본적으로는 SYS 계정만 등록이 되어 있다.

Remote로 sysdba권한을 가지고 db system에 연결하고자 하는 요청이 있을 때

연결 대상인 Db system에서는 password file안에 sys user가 등록되어 있는지를 확인하고

Sys user가 등록되어 있으면 remote로 sys user로 connect하는 것을 허용하게 된다.

반면 password file에 문제가 있다거나 password file안에 sys user가 등록되어 있지 않으면

remote로 sys user로 connect하는 것을 허용되지 않는다.

HR USER에게 SYSDBA권한을 부여하면 PASSWORD FILE에 HR USER가 등록이 되고

REMOTE로 HR USER를 이용해서 SYSDBA권한으로 현재 DB에 CONNECT가 가능해진다.

만약 PASSWORD FILE이 삭제되면 orapwd utility 명령으로 새로 생성 가능.

<<PASSWORD FILE 실습>>

[oracle@edydr1p1 ~]$ sqlplus / as sysdba

SQL> select username from dba\_users

2 where username in ('SYS','SYSTEM','SYSDG','SYSBACKUP','SYSKM');

USERNAME

--------------------------------------------------------------------------------------------------------------------------------

SYSTEM

SYS

SYSDG

SYSKM

SYSBACKUP

5 rows selected.

Elapsed: 00:00:00.01

SQL> EXIT

[oracle@edydr1p1 ~]$ cd $ORACLE\_HOME/dbs

[oracle@edydr1p1 dbs]$ ls

hc\_em12rep.dat init.ora lkORCL **orapworcl** spfileorcl.ora

hc\_orcl.dat lkEM12REP orapwem12rep spfileem12rep.ora

[oracle@edydr1p1 dbs]$ sqlplus / as sysdba

SQL> select \* from v$pwfile\_users;

USERNAME SYSDB SYSOP SYSAS SYSBA SYSDG SYSKM CON\_ID

------------------------------ ----- ----- ----- ----- ----- ----- ----------

SYS TRUE TRUE FALSE FALSE FALSE FALSE 0

1 row selected.

=> remote로 sys user를 이용해서 sysdba, sysoper 권한으로 로그인할 수 있다,

SQL> grant sysdba to hr;

SQL> select \* from v$pwfile\_users;

USERNAME SYSDB SYSOP SYSAS SYSBA SYSDG SYSKM CON\_ID

------------------------------ ----- ----- ----- ----- ----- ----- ----------

SYS TRUE TRUE FALSE FALSE FALSE FALSE 0

HR TRUE FALSE FALSE FALSE FALSE FALSE 0

=> remote로 sys user를 이용해서 sysdba, sysoper 권한으로 로그인할 수 있다,

=> remote로 hr user를 이용해서 sysdba권한으로 로그인할 수 있다

SQL> revoke sysdba from hr;

Revoke succeeded.

SQL> select \* from v$pwfile\_users;

USERNAME SYSDB SYSOP SYSAS SYSBA SYSDG SYSKM CON\_ID

------------------------------ ----- ----- ----- ----- ----- ----- ----------

SYS TRUE TRUE FALSE FALSE FALSE FALSE 0

1 row selected.

=> remote로 sys user를 이용해서 sysdba, sysoper 권한으로 로그인할 수 있다,

=> remote로 hr user를 이용해서 sysdba권한으로 로그인할 수 없다.

SQL> EXIT

<< 관리자 권한 예제>>

-- 관리자 권한을 OS 인증을 통해 사용.

예를 들면 SYSDBA권한을 OS GROUP인 dba 그룹(그룹이름은 달라질 수 있다)에 부여

Dba 그룹에 속해있는 oracle os user가 sysdba권한을 갖게 된다.

sqlplus sys/oracle as sysdba => sys user를 이용해서 sys user가 가지고 있는 sysdba권한으로 db log on

sqlplus / as sysdba => 현재 logon되어 있는 os user(oracle user)가 소유하고 있는 sysdba권한을 가지고 log on

현재 logon한 user가 oracle user가 아닌 경우에는

sqlplus / as sysdba 명령을 실행하면 error 발생

sqlplus sys/oracle as sysdba 으로 실행해야 log on 가능

<< OS USER oracle 유저가 dba os group에 연결되어 있고 dba os group은 sysdba 권한을 할당받은 상태이기 때문에 Oracle user로 로그인하면 아래와 같이 sysdba권한을 가진 sys user로 로그인이 가능해진다. 이런 인증방식을 OS 인증방식이라고 한다>>

[oracle@edydr1p1 dbs]$ id

uid=54321(oracle) gid=54321(oinstall) groups=54321(oinstall),491(vboxsf),54322(dba),54323(oper),54324(backupdba),54325(dgdba),54326(kmdba),54327(asmdba)

[oracle@edydr1p1 dbs]$ sqlplus / as sysdba

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> show user

USER is "SYS"

SQL>

<<DB 계정 생성 예제>>

CREATE USER USERA IDENTIFIED BY oracle

DEFAULT TABLESPACE USERS

QUOTA 50M ON USERS

QUOTA 50M ON EXAMPLE

TEMPORARY TABLESPACE TEMP

PROFILE DEFAULT

ACCOUNT LOCK

PASSWORD EXPIRE;

<< HR USER 의 정보 조회 및 관리 예 >>

[oracle@edydr1p1 dbs]$ sqlplus / as sysdba

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> show user

USER is "SYS"

SQL> select username, default\_tablespace,temporary\_tablespace

2 created, profile, account\_status

3 FROM DBA\_USERS

4 WHERE USERNAME = 'HR';

USERNAME

--------------------------------------------------------------------------------------------------------------------------------

DEFAULT\_TABLESPACE CREATED

------------------------------ ------------------------------

PROFILE

--------------------------------------------------------------------------------------------------------------------------------

ACCOUNT\_STATUS

--------------------------------

HR

USERS TEMP

DEFAULT

OPEN

SQL> ALTER USER HR ACCOUNT LOCK;

User altered.

SQL> SELECT USERNAME, ACCOUNT\_STATUS

2 FROM DBA\_USERS

3 WHERE USERNAME = 'HR'

4 /

USERNAME

--------------------------------------------------------------------------------------------------------------------------------

ACCOUNT\_STATUS

--------------------------------

HR

LOCKED

SQL> ALTER USER HR ACCOUNT UNLOCK;

User altered.

SQL> SELECT USERNAME, ACCOUNT\_STATUS

2 FROM DBA\_USERS

3 WHERE USERNAME = 'HR'

4 /

USERNAME

--------------------------------------------------------------------------------------------------------------------------------

ACCOUNT\_STATUS

--------------------------------

HR

OPEN

1 row selected.

SQL> ALTER USER HR IDENTIFIED BY oracle2

2 /

User altered.

SQL> connect hr/oracle2

Connected.

SQL> show user

USER is "HR"

SQL> connect / as sysdba

Connected.

SQL> show user

USER is "SYS"

SQL> ALTER USER HR IDENTIFIED BY oracle

2 /

User altered.

SQL> connect hr/oracle

Connected.

SQL> show user

USER is "HR"

SQL> connect / as sysdba

Connected.

SQL>

<< DB 계정 생성 예제>>

[oracle@edydr1p1 ~]$ sqlplus / as sysdba

Connected to an idle instance.

SQL> startup

ORACLE instance started.

Total System Global Area 536870912 bytes

Fixed Size 2926472 bytes

Variable Size 234883192 bytes

Database Buffers 293601280 bytes

Redo Buffers 5459968 bytes

Database mounted.

Database opened.

SQL> exit

[oracle@edydr1p1 ~]$ gedit a.sql

CREATE USER USERA IDENTIFIED BY oracle

DEFAULT TABLESPACE USERS

QUOTA 50M ON USERS

QUOTA 50M ON EXAMPLE

TEMPORARY TABLESPACE TEMP

PROFILE DEFAULT

ACCOUNT LOCK

PASSWORD EXPIRE;

[oracle@edydr1p1 ~]$ sqlplus / as sysdba

SQL> @a.sql

User created.

SQL> select ACCOUNT\_STATUS from dba\_users

2 where username = 'USERA'

3 /

ACCOUNT\_STATUS

--------------------------------

EXPIRED & LOCKED

SQL> ALTER USER USERA ACCOUNT UNLOCK;

User altered.

SQL> select ACCOUNT\_STATUS from dba\_users

2 where username = 'USERA'

3 /

ACCOUNT\_STATUS

--------------------------------

EXPIRED

SQL> EXIT

[oracle@edydr1p1 ~]$ gedit b.sql

CREATE USER USERB IDENTIFIED BY oracle

DEFAULT TABLESPACE USERS

QUOTA 50M ON USERS

QUOTA 50M ON EXAMPLE

TEMPORARY TABLESPACE TEMP

PROFILE DEFAULT

ACCOUNT LOCK

PASSWORD EXPIRE;

[oracle@edydr1p1 ~]$ sqlplus / as sysdba

SQL> @b.sql

User created

SQL> col username format a10

SQL> select username, default\_tablespace,ACCOUNT\_STATUS

2 FROM dba\_users

3 where username in ('USERA','USERB')

4 /

USERNAME DEFAULT\_TABLESPACE ACCOUNT\_STATUS

---------- ------------------------------ --------------------------------

USERB USERS LOCKED

USERA USERS EXPIRED

SQL> connect usera/oracle

ERROR:

ORA-28001: the password has expired

Changing password for usera

New password:

Retype new password:

ERROR:

ORA-01045: user USERA lacks CREATE SESSION privilege; logon denied

Password changed

Warning: You are no longer connected to ORACLE.

SQL> connect / as sysdba

Connected.

SQL> grant create session to usera

2 /

Grant succeeded.

SQL> connect usera/oracle

Connected.

SQL> show user

USER is "USERA"

SQL> select \* from session\_privs;

PRIVILEGE

----------------------------------------

CREATE SESSION

1 row selected.

SQL> create table a (id number)

2 /

create table a (id number)

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL> connect / as sysdba

Connected.

SQL> grant create table to usera

2 /

Grant succeeded.

SQL> connect usera/oracle

Connected.

SQL> select \* from session\_privs;

PRIVILEGE

----------------------------------------

CREATE TABLE

CREATE SESSION

SQL> create table a (id number);

Table created.

SQL> connect userb/oracle

ERROR:

ORA-28000: the account is locked

SQL> connect / as sysdba

Connected.

SQL> alter user userb account unlock;

User altered.

SQL> grant create session, create table to userb;

Grant succeeded.

SQL> connect userb/oracle

Connected.

SQL> show user

USER is "USERB"

SQL> select table\_name from user\_tables;

no rows selected

SQL> connect usera/oracle

Connected.

SQL> show user

USER is "USERA"

SQL> select table\_name from user\_tables;

TABLE\_NAME

--------------------------------------------------------------------------------------------------------------------------------

A

<< 객체권한(object privilege) 예제>>

[oracle@edydr1p1 ~]$ sqlplus usera/oracle

SQL> show user

USER is "USERA"

SQL> select employee\_id from hr.employees;

select employee\_id from hr.employees

\*

ERROR at line 1:

ORA-00942: table or view does not exist

SQL> connect hr/oracle

Connected.

SQL> show user

USER is "HR"

SQL> grant select on employees to usera

2 /

Grant succeeded.

SQL> connect / as sysdba

Connected.

SQL> show user

USER is "SYS"

SQL> grant update on hr.employees to usera;

Grant succeeded.

SQL> connect usera/oracle

Connected.

SQL> select employee\_id

2 from hr.employees

3 where employee\_id=100

4 /

EMPLOYEE\_ID

-----------

100

1 row selected.

SQL> col grantee format a10

SQL> col owner format a10

SQL> col table\_name format a10

SQL> col grantor format a10

SQL> select \* from user\_tab\_privs

2 /

GRANTEE OWNER TABLE\_NAME GRANTOR PRIVILEGE GRA HIE COM TYPE

---------- ---------- ---------- ---------- ---------------------------------------- --- --- --- ------------------------

USERA HR EMPLOYEES HR SELECT NO NO NO TABLE

USERA HR EMPLOYEES HR UPDATE NO NO NO TABLE

PUBLIC SYS USERA USERA INHERIT PRIVILEGES NO NO NO USER

3 rows selected.

SQL> revoke update on hr.employees from usera;

SQL> col grantee format a10

SQL> col owner format a10

SQL> col table\_name format a10

SQL> col grantor format a10

SQL> select \* from user\_tab\_privs

2 /

GRANTEE OWNER TABLE\_NAME GRANTOR PRIVILEGE GRA HIE COM TYPE

---------- ---------- ---------- ---------- ---------------------------------------- --- --- --- ------------------------

USERA HR EMPLOYEES HR SELECT NO NO NO TABLE

PUBLIC SYS USERA USERA INHERIT PRIVILEGES NO NO NO USER

– 시스템권한의 경우

GRANT 시스템권한 TO USERA **WITH ADMIN OPTION**

ADMIN OPTION은 DBA(SYS USER)가

일반 USERA에게 SYSTEM 권한을 부여하면 권한을 받은 USER(USERA)가

또 다른 USER(예를 들면 USERB)에게 자신이 받은 시스템권한을 GRANT할 수 있다.

DBA가 시스템권한을 USER(USERA)로부터 뺏으면

(REVOKE명령을 이용해서 USERA로부터 권한 삭제) USERB에게 부여된 권한은 같이

삭제되지 않는다. 그래서 DBA USER는 USERB로부터 별도로 부여된 시스템권한을 뺏는 작업을

해야 한다.

– 객체권한 (OBJECT 권한)

GRANT 객체권한 TO USERA **WITH GRANT OPTION**

GRANT OPTION은 USERC가 일반 USERA에게 객체 권한을 부여하면 권한을 받은 USERA가 다른

USERB에게 USERA가 받은 권한을 GRANT할 수 있다.

USERC가 객체권한을 뺏으려면 REVOKE명령을 이용해서 USERA한테서만 뺏으면

USERB에게 간접적으로 전달된 권한도 자동으로 같이 뺏기게 된다.

<<ROLE>>

권한이나 다른 ROLE을 포함할 수 있는 그룹 개념

<< ROLE 정보 조회 >>

[oracle@edydr1p1 dbs]$ sqlplus / as sysdba

SQL> desc dba\_roles

Name Null? Type

----------------------------------------------------------------------------------- -------- --------------------------------------------------------

ROLE NOT NULL VARCHAR2(128)

PASSWORD\_REQUIRED VARCHAR2(8)

AUTHENTICATION\_TYPE VARCHAR2(11)

COMMON VARCHAR2(3)

ORACLE\_MAINTAINED VARCHAR2(1)

SQL> desc role\_sys\_privs

Name Null? Type

----------------------------------------------------------------------------------- -------- --------------------------------------------------------

ROLE VARCHAR2(128)

PRIVILEGE VARCHAR2(40)

ADMIN\_OPTION VARCHAR2(3)

COMMON VARCHAR2(3)

SQL> select count(\*) from role\_sys\_privs

2 where role = 'DBA'

3 /

COUNT(\*)

----------

220

1 row selected.

SQL> select count(\*) from role\_sys\_privs

2 where role = 'CONNECT'

3 /

COUNT(\*)

----------

2

1 row selected.

SQL>

<< ROLE 셋업 예제>>

[oracle@edydr1p1 dbhome\_1]$ sqlplus / as sysdba

SQL> create role test\_role;

Role created.

SQL> grant create view, create table to test\_role;

Grant succeeded.

SQL> grant select, insert on hr.employees to test\_role;

Grant succeeded.

SQL> grant test\_role to usera

2 /

Grant succeeded.

SQL> connect usera/oracle

Connected.

SQL> show user

USER is "USERA"

SQL> select \* from session\_roles;

ROLE

--------------------------------------------------------------------------------------------------------------------------------

TEST\_ROLE

1 row selected.

SQL> select PRIVILEGE from role\_sys\_privs

2 where role = 'TEST\_ROLE';

PRIVILEGE

----------------------------------------

CREATE TABLE

CREATE VIEW

2 rows selected.

SQL> SELECT TABLE\_NAME, PRIVILEGE FROM ROLE\_TAB\_PRIVS

2 WHERE ROLE = 'TEST\_ROLE';

TABLE\_NAME

--------------------------------------------------------------------------------------------------------------------------------

PRIVILEGE

----------------------------------------

EMPLOYEES

INSERT

EMPLOYEES

SELECT

SQL> SHOW USER

USER is "USERA"

SQL> SELECT EMPLOYEE\_ID, LAST\_NAME FROM HR.EMPLOYEES

2 WHERE EMPLOYEE\_ID=200;

EMPLOYEE\_ID LAST\_NAME

----------- -------------------------

200 Whalen

1 row selected.

<< dba role grant 예제>>

[oracle@edydr1p1 ~]$ sqlplus / as sysdba

SQL> grant dba to hr;

Grant succeeded.

SQL> connect hr/oracle

Connected.

SQL> select count(\*) from session\_privs;

COUNT(\*)

----------

222

1 row selected.

SQL> connect / as sysdba

Connected.

SQL> revoke dba from hr;

Revoke succeeded.

SQL> connect hr/oracle

Connected.

SQL> select count(\*) from session\_privs;

COUNT(\*)

----------

13

1 row selected.

<< SET ROLE 예제 >>

– ROLE에 대해서 보안설정

[oracle@edydr1p1 ~]$ sqlplus / as sysdba

SQL> create role test\_a;

Role created.

SQL> create role test\_b;

Role created.

SQL> grant select on hr.employees to test\_a;

Grant succeeded.

SQL> grant update on hr.employees to test\_b;

Grant succeeded.

SQL> grant test\_a,test\_b to usera

2 /

Grant succeeded.

SQL> connect usera/oracle

Connected.

SQL> show user

USER is "USERA"

SQL> select \* from session\_roles;

ROLE

--------------------------------------------------------------------------------------------------------------------------------

TEST\_A

TEST\_B

2 rows selected.

SQL> connect / as sysdba

Connected.

SQL> alter user usera

2 default role test\_a;

User altered.

SQL> connect usera/oracle

Connected.

SQL> show user

USER is "USERA"

SQL> select \* from session\_roles;

ROLE

--------------------------------------------------------------------------------------------------------------------------------

TEST\_A

1 row selected.

SQL> update hr.employees

2 set salary=9999

3 where employee\_id=200;

update hr.employees

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL> set role test\_b; => TEST\_B ROLE을 활성화

Role set.

SQL> update hr.employees set salary=9999 where employee\_id=200

2 /

1 row updated.

SQL> set role test\_a,test\_b;

Role set.

SQL> select \* from session\_roles;

ROLE

--------------------------------------------------------------------------------------------------------------------------------

TEST\_A

TEST\_B

<< PASSWORD가 걸린 ROLE의 활성화>>

SQL> connect / as sysdba

Connected.

SQL> create role test\_c **identified by oracle;**

Role created.

SQL> grant delete on hr.employees to test\_c;

Grant succeeded.

SQL> grant test\_c to usera;

Grant succeeded.

SQL> set role test\_a,test\_b,test\_c identified by oracle

2 /

Role set.

SQL> select \* from session\_roles;

ROLE

--------------------------------------------------------------------------------------------------------------------------------

TEST\_A

TEST\_B

TEST\_C

<< PROFILE의 예>>

CREATE PROFILE A\_PROFILE LIMIT

FAILED\_LOGIN\_ATTEMPTS 6 => 6번이상 암호입력을 잘못하면 계정을 잠근다.

PASSWORD\_LIFE\_TIME 60 => 60일 암호변경 주기

PASSWORD\_REUSE\_TIME 60 => 같은 암호를 60일간은 사용하지 못한다.

PASSWORD\_REUSE\_MAX 5 => 최근 사용했던 암호 5개를 기억하면서 사용 못하게 한다.

PASSWORD\_LOCK\_TIME 1/24 => 계정이 잠겨있는 기간

PASSWORD\_GRACE\_TIME 10 => 암호변경주기의 유예기간

PASSWORD\_VERIFY\_FUNCTION DEFAULT; => 암호에 대한 규칙

SESSIONS\_PER\_USER UNLIMITED => 같은 USERNAME으로 동시에 연결가능한 CONNECT수

CPU\_PER\_SESSION UNLIMITED ⇒ 한 세션에서 사용할 수 있는 최대 CPU 시간

CPU\_PER\_CALL 3500 => 한 세션안에 하나의 SQL명령이 사용할 수 있는 최대 CPU 시간

CONNECT\_TIME 50 => 한 세션에 대해서 작업가능 최대시간

LOGICAL\_READS\_PER\_SESSION DEFAULT => 한 세션에서 읽을 수 있는 데이터의 최대 크기

LOGICAL\_READS\_PER\_CALL 1200 => 한 세션안에 하나의 SQL명령이 읽을 수 있는 데이터의 최대 크기

PRIVATE\_SGA 20K => SHARED SERVER환경에서 UGA안에서 한 세션이 잡을 수 있는 최대 크기

RESOURCE\_LIMIT가 TRUE로 설정되어야 프로파일에서 리소스 제한을 적용할 수 있습니다.

CREATE USER TEST\_USER

PROFILE A\_PROFILE;

ALTER USER TEST\_USER

PROFILE B\_PROFILE;

ALTER PROFILE app\_user LIMIT

SESSIONS\_PER\_USER UNLIMITED

CPU\_PER\_SESSION UNLIMITED

CPU\_PER\_CALL 3500

CONNECT\_TIME 50

LOGICAL\_READS\_PER\_SESSION DEFAULT

LOGICAL\_READS\_PER\_CALL 1200

기본적으로 사용되는 프로파일의 이름은 DEFAULT

CREATE USER 명령을 통해서 USER 생성시 별도로 PROFILE을 지정하지 않으면

DEFAULT PROFILE이 적용

ALTER PROFILE 명령으로 DEFAULT PROFILE의 내용을 변경하면

DEFAULT PROFILE을 사용하는 모든 사용자들에게 변경내용이 적용

DBA\_PROFILES 데이터 딕셔너리 뷰

=> 현재 만들어져 있는 PROFILE의 정보

DBA\_USERS

=> 어떤 USER가 어떤 PROFILE에 연결되어 있는지 조회 가능

PROFILE의 내용 조회

https://docs.oracle.com/en/database/oracle/oracle-database/12.2/dbseg/managing-security-for-oracle-database-users.html#GUID-94816191-5539-4D2F-A1AC-B41989840B03

GRANT unlimited tablespace to usera;

=> usera는 데이터베이스 안에 있는 모든 tablespace공간을 다 사용 가능.

– SYSDBA권한을 가지고 있는 USER가 조회 가능

– USERA가 어느 테이블스페이스를 사용할 수 있고 해당 테이블스페이스안에서 사용할 수 있는

– MAX SIZE(QUOTA) 확인

SELECT USERNAME,TABLESPACE\_NAME,MAX\_BYTES

FROM DBA\_TS\_QUOTAS

WHERE USERNAME='USERA';

ALTER USER USERA QUOTA 100M ON USERS;

=> USERA가 USERS 테이블스페이스에서 사용할 수 있는 QUOTA의 조정

– TEST 테이블의 데이타가 저장되는 TABLESPACE정보 조회

SELECT TABLE\_NAME, TABLESPACE\_NAME

FROM USER\_TABLES

WHERE TABLE\_NAME='TEST';

O7\_DICTIONARY\_ACCESSIBILITY=FALSE

=> DICTIONARY TABLE의 내용을 수동으로(직접) 변경 불가.

O7\_DICTIONARY\_ACCESSIBILITY=TRUE => 보안위험

=> DICTIONARY TABLE의 내용을 수동으로 변경 가능.

O7\_DICTIONARY\_ACCESSIBILITY=FALSE

: data dictionary table 안에 있는 시스템 데이터는 수동으로 변경(insert/update/delete 명령) 불가능

: 예외적인 경우에만 이 값을 TRUE로 설정이 되면 수동으로 변경(insert/update/delete 명령) 가능

(시스템의 유지보수나 에러 수정을 위해)

PUBLIC에서 불필요한 권한 취소

: PUBLIC은 DB안에 모든 사용자를 가리키는 이름

: GRANT 권한 TO PUBLIC; (DB안에 모든 사용자들에게 일괄적으로 권한 부여)

<< 데이터베이스 공간 관리>>

하나의 데이터베이스 에는 하나 이상의 테이블스페이스로 구성되고

하나의 테이블스페이스는 하나 이상의 데이터파일로 구성 가능

하나의 테이블스페이스에는 하나 이상의 세그먼트들로 구성된다.

세그먼트는 보통 일반적인 경우에는 1 테이블당 1개의 세그먼트가 생긴다.

예외적으로 1개의 테이블당 여러개의 세그먼트가 생기는 경우도 있다. (PARTITION TABLE의 경우)

세그먼트는 실제 데이터가 저장되는 공간의 개념.

테이블이 생성이 된다고 해서 바로 세그먼트가 생성되는 것은 아니다.

테이블에 실제 데이터가 입력이 되면 세그먼트가 생성된다.

세그먼트가 만들어지는 단위가 EXTENT이다

하나의 세그먼트는 하나 이상의 EXTENT들로 구성된다.

EXTENT를 저장 공간의 할당단위

하나의 EXTENT는 하나이상의 데이터가 없는 BLOCK들(FREE BLOCK)로 구성이 된다.

하나의 블럭의 크기가 2K,4K,8K,16K,32K중 하나로 잡히고 블럭의 크기는 데이타베이스가 생성이 될 때 결정된다.

DEFAULT는 8K로 셋업된다. 그리고 보통 하나의 EXTENT는 DEFAULT로 64K로 잡힌다.

기본적으로는 하나의 EXTENT에 8개의 블럭이 포함된다.

테이블스페이스의 정보 조회

DBA\_TABLESPACES

DBA\_DATA\_FILES

<< TABLESPACE정보 조회>>

[oracle@edydr1p1 ~]$ cd

[oracle@edydr1p1 ~]$ pwd

/home/oracle

[oracle@edydr1p1 ~]$ gedit c.sql

CREATE TABLESPACE TEST\_TABLESPACE1

DATAFILE '/home/oracle/test1.dbf' size 60m;

[oracle@edydr1p1 ~]$ sqlplus / as sysdba

SQL> @c.sql

Tablespace created.

SQL> col file\_name format a40

SQL> col tablespace\_name format a20

SQL> SELECT FILE\_NAME, TABLESPACE\_NAME

2 FROM DBA\_DATA\_FILES

3 WHERE TABLESPACE\_NAME = 'TEST\_TABLESPACE1'

4 /

FILE\_NAME TABLESPACE\_NAME

-------------------- --------------------------------------

/home/oracle/test1.dbf TEST\_TABLESPACE1

SQL> ALTER TABLESPACE TEST\_TABLESPACE1

2 ADD DATAFILE '/home/oracle/test2.dbf' size 70m

3 /

Tablespace altered.

SQL> SELECT FILE\_NAME, TABLESPACE\_NAME

2 FROM DBA\_DATA\_FILES

3 WHERE TABLESPACE\_NAME = 'TEST\_TABLESPACE1'

4 /

FILE\_NAME TABLESPACE\_NAME

-------------------- --------------------

/home/oracle/test1.dbf TEST\_TABLESPACE1

/home/oracle/test2.dbf TEST\_TABLESPACE1

SQL> col file\_name format a60

SQL> select tablespace\_name, file\_name

2 from dba\_data\_files

3 /

TABLESPACE\_NAME FILE\_NAME

-------------------- ----------------------------------------

SYSTEM /u01/app/oracle/oradata/orcl/system01.dbf

SYSAUX /u01/app/oracle/oradata/orcl/sysaux01.dbf

EXAMPLE /u01/app/oracle/oradata/orcl/example01.dbf

USERS /u01/app/oracle/oradata/orcl/users01.dbf

TEST\_TABLESPACE1 /home/oracle/test1.dbf

UNDOTBS1 /u01/app/oracle/oradata/orcl/undotbs01.dbf

TEST\_TABLESPACE1 /home/oracle/test2.dbf

SQL> select tablespace\_name, file\_name

2 from dba\_temp\_files

3 /

TABLESPACE\_NAME FILE\_NAME

-------------------- ----------------------------------------

TEMP /u01/app/oracle/oradata/orcl/temp01.dbf

SQL> CREATE USER TESTA IDENTIFIED BY oracle

2 DEFAULT TABLESPACE TEST\_TABLESPACE1

3 TEMPORARY TABLESPACE TEMP

4 QUOTA 5M ON TEST\_TABLESPACE1

5 QUOTA 10M ON USERS

6 /

User created.

SQL> GRANT CONNECT, RESOURCE TO TESTA

2 /

Grant succeeded.

SQL> connect testa/oracle

Connected.

SQL> show user

USER is "TESTA"

SQL> create table testa1 (id number)

2 /

Table created.

SQL> col tablespace\_name format a20

SQL> col table\_name format a10

SQL> select table\_name, tablespace\_name

2 from user\_tables

3 /

TABLE\_NAME TABLESPACE\_NAME

---------- --------------------

TESTA1 TEST\_TABLESPACE1

1 row selected.

SQL> create table testb (id number) tablespace users;

Table created.

SQL> select table\_name, tablespace\_name

2 from user\_tables

3 /

TABLE\_NAME TABLESPACE\_NAME

---------- --------------------

TESTA1 TEST\_TABLESPACE1

TESTB USERS

SQL> create table testc (id number) tablespace example

2 /

Table created.

SQL> connect testa/oracle

Connected.

SQL> show user

USER is "TESTA"

SQL> col segment\_name format a10

SQL> col tablespace\_name format a15

SQL> select segment\_name, tablespace\_name

2 from user\_segments

3 where segment\_name = 'TESTA1';

no rows selected

SQL> select segment\_name, tablespace\_name, extent\_id

2 from user\_extents

3 /

no rows selected

SQL> insert into testa1 values (1000)

2 /

1 row created.

SQL> commit;

Commit complete.

SQL> select segment\_name, tablespace\_name

2 from user\_segments

3 where segment\_name = 'TESTA1';

SEGMENT\_NA TABLESPACE\_NAME

---------- ---------------

TESTA1 TEST\_TABLESPACE

SQL> select segment\_name, tablespace\_name, extent\_id

2 from user\_extents

3 /

SEGMENT\_NA TABLESPACE\_NAME EXTENT\_ID

---------- --------------- ----------

TESTA1 TEST\_TABLESPACE 0

SQL> insert into testa1

2 select \* from testa1

3 /

1 row created.

SQL> /

2 rows created.

SQL> /

4 rows created.

SQL> /

8 rows created.

SQL> /

16 rows created.

SQL> /

32 rows created.

SQL> /

64 rows created.

SQL> /

128 rows created.

SQL> /

256 rows created.

SQL> /

512 rows created.

SQL> /

1024 rows created.

SQL> /

2048 rows created.

SQL> commit;

Commit complete.

SQL> select count(\*) from testa1;

COUNT(\*)

----------

4096

1 row selected.

SQL> select segment\_name, tablespace\_name, extent\_id, blocks

2 from user\_extents

3 /

SEGMENT\_NA TABLESPACE\_NAME EXTENT\_ID BLOCKS

---------- --------------- ---------- ----------

TESTA1 TEST\_TABLESPACE 0 8

TESTA1 TEST\_TABLESPACE 1 8

2 rows selected.

<< TABLESPACE 관련 문법>>

– smallfile tablespace : DEFAULT TYPE, 하나의 테이블스페이스가 여러개의 데이터파일을 가질 수 있다.

CREATE TABLESPACE 이름

DATAFILE '/directory/file\_name.dbf' size 50m

– BIGfile tablespace : 하나의 테이블스페이스가 한 개의 데이터파일만 가질 수 있다

– 데이터파일의 크기가 TB단위까지 가질 수 있다.

CREATE BIGFILE TABLESPACE 이름

DATAFILE '/directory/file\_name.dbf' size 50m

<<undo tablespace>>

CREATE UNDO TABLESPACE 이름

DATAFILE '/directory/file\_name.dbf' size 50m

<<temporary tablespace>>

CREATE TEMPORARY TABLESPACE TEMP2

TEMPFILE '/home/oracle/temp2.dbf' size 50m;

SQL> SELECT TABLESPACE\_NAME, FILE\_NAME

2 FROM DBA\_TEMP\_FILES;

TABLESPACE\_NAME FILE\_NAME

------------------------------ --------------------------------------------------

TEMP /u01/app/oracle/oradata/orcl/temp01.dbf

TEMP2 /home/oracle/temp2.dbf

SQL> create user testyy identified by oracle

2 default tablespace users

3 quota unlimited on users

4 temporary tablespace temp2;

User created.

ALTER DATABASE DATAFILE '/home/oracle/test\_tablespace1.dbf' RESIZE 100M;

ALTER DATABASE DATAFILE '/home/oracle/test\_tablespace1.dbf' AUTOEXTEND ON

NEXT 10M

MAXSIZE 300M;

ALTER TABLESPACE USERS OFFLINE; – USERS TABLESPACE를 임시로 사용불가능한 상태로 전환

ALTER TABLESPACE USERS ONLINE; – USERS TABLESPACE를 임시로 사용가능한 상태로 전환

ALTER TABLESPACE USERS READ ONLY; - USERS TABLESPACE안에 데이터를 조회만 가능

ALTER TABLESPACE USERS READ WRITE; - USERS TABLESPACE안에 데이터를 조회와 변경 가능

ALTER DATABASE DATAFILE '/u02/oracle/rbdb1/users03.dbf'

AUTOEXTEND OFF;

테이블스페이스 내에서 사용되는 DATA BLOCK의 크기는 DB 기본 BLOCK SIZE와는 다른 BLOCK SIZE로

설정이 가능하다. BLOCK SIZE를 다르게 하는 이유는 BLOCK SIZE가 SQL실행의 성능에 영향을 줄 수 있기 때문이다. 예를 들어 BLOCK SIZE가 크게 되면 많은 데이터를 적은 수의 BLOCK으로 검색할 수 있기 때문에 조회속도가 빨라질 수 있다. 상대적으로 적은 데이터들을 자주 검색하거나 변경하는 경우에는 BLOCK을 크기가 작은 것이 성능을 좋게 할 수 있다.

BLOCK SIZE는 기본적으로 DB 생성시 기본 BLOCK SIZE가 설정된다.

이 기본 BLOCK SIZE가 이후에 추가되는 테이블스페이스안에 만들어지는 기본 블럭SIZE가 된다.

그런데 필요하다면 테이블스페이스별로 블럭SIZE를 별도로 SETTING할 수 있다.

블럭의 SIZE가 크면 좋은 경우는 많은 데이터들이 한꺼번에 입력이 되고 한번에 많은 데이터가

조회되는 경우 (BIG DATA환경)

테이블스페이스안에 생성되는 EXTENT들에 대한 관리방법이 AUTO방식과 UNIFORM방식이 있다.

AUTO방식이 오라클이 RECOMMEND하는 방식

AUTO방식은 EXTENT의 크기를 ORACLE내에서 자동조정하는 기능

<< 테이블스페이스 삭제 예제>>

SQL> CREATE TABLESPACE TESTU

2 DATAFILE '/home/oracle/testu.dbf' size 50m

3 /

Tablespace created.

SQL> EXIT

[oracle@edydr1p1 ~]$cd /home/oracle

[oracle@edydr1p1 ~]$ ls

afiedt.buf labs solns  **testu.dbf**  Videos

bea Music temp2.dbf undo2.dbf

Desktop oradiag\_oracle Templates usera.sql

Documents Pictures test\_tablespace1-1.dbf userb.sql

Downloads Public test\_tablespace1.dbf userb.sql~

[oracle@edydr1p1 ~]$ sqlplus / as sysdba

SQL> drop tablespace testu

2 /

Tablespace dropped.

SQL> exit

[oracle@edydr1p1 ~]$ ls

afiedt.buf labs solns  **testu.dbf** Videos

bea Music temp2.dbf undo2.dbf

Desktop oradiag\_oracle Templates usera.sql

Documents Pictures test\_tablespace1-1.dbf userb.sql

Downloads Public test\_tablespace1.dbf userb.sql~

[oracle@edydr1p1 ~]$ rm -r testu.dbf

테이블스페이스와 관련된 데이터파일들도 같이 삭제하고자 하는 경우

만약 테이블스페이스안에 테이블 데이터가 저장되어 있는 경우에는 반드시 아래의 명령을

사용해야만 테이블스페이스를 DROP 할 수 있게 된다.

DROP TABLESPACE 이름 **INCLUDING CONTENTS AND DATAFILES;**

<< OMF (Oracle Managed File) >>

[oracle@edydr1p1 ~]$ sqlplus / as sysdba

SQL> alter system set db\_create\_file\_dest='/home/oracle';

System altered.

– /home/oracle 위치에 100m 크기의 datafile이 자동으로 생성.

SQL> create tablespace testp;

Tablespace created.

SQL> col file\_name format a50

SQL> select tablespace\_name, file\_name

2 from dba\_data\_files

3 where tablespace\_name='TESTP';

TABLESPACE\_NAME FILE\_NAME

------------------------------ --------------------------------------------------

TESTP /home/oracle/ORCL/datafile/o1\_mf\_testp\_jysco7st\_.dbf

1 row selected.

SQL> alter tablespace testp add datafile '/home/oracle/tt.dbf' size 50m;

EXTENT공간의 관리와 DB BLOCK 공간의 관리방법에 따른 테이블스페이스의 종류

1. LOCAL MANAGED TABLESPACE (LMT) : 12c 버전에서 DEFAULT로 사용됨

CREATE TABLESPACE 이름

EXTENT MANAGEMENT LOCAL

SEGMENT SPACE MANAGEMENT AUTO

EXTENT MANAGEMENT LOCAL이 의미하는 내용은

DATAFILE의 HEADER에서 BITMAP으로 파일안에 있는 블럭들의 상태를 자체 관리하게 된다.

데이터가 입력가능한 FREE BLOCK들의 정보를 빨리 확인이 가능하기 때문에

새로운 EXTENT를 만드는 작업이 빠르게 실행될 수 있다.

SEGMENT SPACE MANAGEMENT AUTO로 설정하면

각 SEGMENT마다 SEGMENT HEADER에서 BITMAP으로

기존의 EXTENT안에 비어있는 BLOCK을 관리를 하게 되고

비어 있는 BLOCK에 대한 검색하는 속도가 빨라져서 데이타의 입력작업의 속도가 빨라질 수 있다.

BLOCK안의 저장공간에 대한 관리방법이 DICTIONARY MANAGED TABLESPACE와는 다르게

관리가 된다. PCTUSED에 대한 관리방법에 차이가 생긴다.

2.DICTIONARY MANAGED TABLESPACE (DMT)

CREATE TABLESPACE 이름

EXTENT MANAGEMENT DICTIONARY

SEGMENT SPACE MANAGEMENT MANUAL

데이터 파일안에 있는 블럭들의 상태를 SYSTEM 테이블스페이스 안에 있는

DATA DICTIONARY 테이블로 저장 관리

결국은 데이타가 입력가능한 FREE BLOCK들의 정보를 빨리 확인하는데 시간이 걸리는 구조로 옛날 방식의 공간관리 방법

그래서 새로운 EXTENT를 만드는 작업이 LOCAL TABLESPACE보다 느리게 실행될 수 있다.

SEGMENT SPACE MANAGEMENT MANUAL로 설정하면

기존의 EXTENT안에 비어있는 BLOCK을 SYSTEM 테이블스페이스 안에 있는

DATA DICTIONARY 테이블로 저장 관리하게 되고

비어 있는 BLOCK에 대한 검색하는 속도가 상대적으로 느리게 되서

데이타의 입력작업의 속도가 LOCAL TABLESPACE에서보다 느릴 수 있다.

DICTIONARY 관리 테이블스페이스에서는

SEGMENT SPACE MANAGEMENT AUTO 방식 사용 불가능

SEGMENT SPACE MANAGEMENT MANUAL 방식이 사용 가능

행 체인화 (ROW CHAINING), 행 이전 (ROW MIGRATION)

: 하나의 RECORD(row)가 2개이상의 데이타블럭에 나뉘어져 보관되는 현상

: UPDATE시 기존 데이터 보다 큰 데이타가 업데이트되는 경우 ROW CHAINING 발생 가능

: INSERT시 입력될 때 블록의 FREE SPACE가 부족하면 ROW MIGRATION 발생 가능

위의 작업들이 많이 진행되면 데이타의 검색속도가 느려지므로

위의 작업들이 많이 진행된 테이블의 경우에는 테이블의 데이터들을 다시 정리하는 작업이 필요.

<<CHAINED ROWS 조회 QUERY>>

column c1 heading "Owner" format a9;

column c2 heading "Table" format a12;

column c3 heading "PCTFREE" format 99;

column c4 heading "PCTUSED" format 99;

column c5 heading "avg row" format 99,999;

column c6 heading "Rows" format 999,999,999;

column c7 heading "Chains" format 999,999,999;

column c8 heading "Pct" format .99;

set heading off;

select 'Tables with migrated/chained rows and no RAW columns.' from dual;

set heading on;

select

owner c1,

table\_name c2,

pct\_free c3,

pct\_used c4,

avg\_row\_len c5,

num\_rows c6,

chain\_cnt c7,

chain\_cnt/num\_rows c8

from dba\_tables

where table\_name ='EMP';

ALTER TABLE EMP MOVE TABLESPACE USERS;

ALTER TABLE EMP MOVE;

<< 압축방법>>

Direct path 삽입 작업의 경우 기본 압축

Direct path 삽입 작업이란 데이터가 테이블에 INSERT될때 DB BUFFER CACHE를 통하지 않고

DISK에 직접 삽입하는 작업이다. 이 작업을 진행하기 위해서는 일반 INSERT문장으로는 불가능하고

별도의 문법을 사용해야 한다. (예: SQL LOADER를 이용한 대량의 데이타 삽입작업)

고급압축의 경우에는 일반 INSERT문장을 사용하는 경우에도 압축이 가능하다.

HWM(HIGH WATER MARK): SEGMENT안에 데이터를 포함하고 있는 마지막 EXTENT를 가리키는

내부 POINTER.

HWM 안에 포함되어 있는 공간은 TABLESPACE LEVEL에서 보았을 때 사용중인 공간으로

인식된다.

HWM를 이동시켜서 만들어지는 빈공간을 TABLESPACE에서 사용가능한 공간으로 만드는

작업이 SHRINK작업이다.

< SHRINK>

하나의 테이블안에 EXTENT로 잡혀 있는 공간중에 비어있는 블록들이 있는 경우

테이블스페이스 레벨에서 보았을때 비어있는 블록들이 FREE SPACE로 CHECK되지 않는다.

비어있는 블록들을 FREE SPACE로 만들어서 다른 테이블이 새로운 EXTENT를 만들 수 있는 공간을 생성하는 작업

LOCAL TABLESPACE안에 있는 SEGMENT에 대해서만 SHRINK 작업 가능.

<<SHRINK 실습 예제>>

-- select Blocks including real data

SQL> select count(distinct dbms\_rowid.rowid\_block\_number(rowid)) block from employees;

BLOCK

----------

556

-- select Hi-water mark block

SQL> select blocks,extents

2 from user\_segments

3 where segment\_name='EMPLOYEES';

BLOCKS EXTENTS

---------- ----------

640 20

SQL> delete from employees

2 where department\_id=50;

23040 rows deleted.

SQL> commit;

Commit complete.

-- some changed count of blocks

SQL> select count(distinct dbms\_rowid.rowid\_block\_number(rowid)) block from employees;

BLOCK

----------

555

-- not changed hi-water mark

SQL> select blocks,extents

2 from user\_segments

3 where segment\_name='EMPLOYEES';

BLOCKS EXTENTS

---------- ----------

640 20

SQL> alter table employees enable row movement;

Table altered.

SQL> alter table employees shrink space compact;

Table altered.

-- some changed count of blocks

SQL> select count(distinct dbms\_rowid.rowid\_block\_number(rowid)) block from employees;

BLOCK

----------

297

-- not changed hi-water mark

SQL> select blocks,extents

2 from user\_segments

3 where segment\_name='EMPLOYEES';

BLOCKS EXTENTS

---------- ----------

640 20

SQL> alter table employees shrink space;

Table altered.

SQL> select count(distinct dbms\_rowid.rowid\_block\_number(rowid)) block from employees;

BLOCK

----------

296

SQL> select blocks,extents

2 from user\_segments

3 where segment\_name='EMPLOYEES';

BLOCKS EXTENTS

---------- ----------

312 18

ASSM 세그먼트 : LOCAL TABLESPACE안에 만들어져 있는 SEGMENT

ASSM(AUTO SEGMENT SPACE MANAGEMENT):LOCAL TABLESPACE에서의 기본 공간 관리방식

<<RESUMABLE SPACE ALLOCATION 예제>>

[oracle@edydr1p1 ~]$ sqlplus / as sysdba

SQL> create tablespace test66

2 datafile '/u01/app/oracle/oradata/orcl/test66.dbf' size 1m;

Tablespace created.

SQL> create table emp\_test

2 (empno number(10),ename char(10))

3 tablespace test;

Table created.

SQL> insert into emp\_test values(100,'a');

SQL> insert into emp\_test

2 select \* from emp\_test

3 /

33 rows created.

SQL> /

66 rows created.

SQL> /

132 rows created.

SQL> /

264 rows created.

SQL> /

528 rows created.

SQL> /

1056 rows created.

SQL> /

2112 rows created.

SQL> /

4224 rows created.

SQL> /

8448 rows created.

SQL> /

16896 rows created.

SQL> /

33792 rows created.

SQL> /

67584 rows created.

SQL>/

insert into emp\_test

\*

ERROR at line 1:

ORA-01653: unable to extend table SYS.EMP\_TEST by 128 in tablespace TEST

SQL> TRUNCATE TABLE EMP\_TEST;

SQL> ALTER SESSION ENABLE RESUMABLE TIMEOUT 3600;

Session altered.

SQL> INSERT INTO EMP\_TEST

2 VALUES(1111,'TEST');

1 row created.

SQL> INSERT INTO EMP\_TEST

2 SELECT \* FROM EMP\_TEST;

1 row created.

SQL> /

2 rows created.

SQL> /

4 rows created.

SQL> /

8 rows created.

SQL> /

16 rows created.

SQL> /

32 rows created.

SQL> /

64 rows created.

SQL> /

128 rows created.

SQL> /

256 rows created.

SQL> /

512 rows created.

SQL> /

1024 rows created.

SQL> /

2048 rows created.

SQL> /

4096 rows created.

SQL> /

8192 rows created.

SQL> /

16384 rows created.

SQL> /

32768 rows created.

SQL> /

65536 rows created.

SQL> /

-- pause

-- open another terminal

[oracle@edydr1p1 ~]$ sqlplus / as sysdba

SQL> alter tablespace test

2 add datafile '/home/oracle/test2.dbf' size 6m;

Tablespace altered.

– then

SQL> /

131072 rows created.

SQL> alter session disable resumable;

Session altered.

# UNDO DATA는 데이터를 변경하는 작업 (INSERT,UPDATE,DELETE) 작업시에 발생하는 OLD DATA(기존의 데이터)를 가리키는 것이고

이 데이터는 DB BUFFER CACHE와 UNDO TABLESPACE안에 UNDO SEGMENT에서 관리된다.

UNDO TABLESPACE는 아래와 같은 명령으로 생성한다.

CREATE UNDO TABLESPACE 이름 DATAFILE '/directory/file' size 100m

UNDO TABLESPACE는 하나의 데이터베이스에 대해서 여러개가 생성가능

그러나 실제로 사용될 수 있는 UNDO TABLESPACE는 어느 한 시점에서

하나의 UNDO TABLESPACE만 사용 가능

현재 사용중인 UNDO TABLESPACE를 가리키는 파라미터가 UNDO\_TABLESPACE이다.

하나의 UNDO TABLESPACE가 여러 사용자들에 의해 공유된다.

UNDO TABLESPACE를 생성하면 그 안에 UNDO SEGMENT들이 자동으로 생성

만약에 UNDO TABLESPACE를 변경하고자 할때는

ALTER SYSTEM SET UNDO\_TABLESPACE='새로운 UNDO TABLESPACE이름' SCOPE=SPFILE

– SNAPSHOT TOO OLD

READ CONSISTENCY (읽기 일관성): 모든 SELECT문장은 SELECT 시작 시간을 기준 데이타를 봐야 한다.

10:00 A USER SELECT 시작 EMP TABLE ~10:30 (30분동안 진행하는 QUERY 라고 가정)

10:10 B USER UPDATE EMP -> COMMIT 10:15 -> UNDO DATA가 지워짐.

-> A USER SELECT가 B USER가 UPDATE한 데이터와 관련된 UNDO DATA를 참조해야 하는 경우에

COMMIT작업으로 인해 UNDO DATA가 사라지게 되면 A USER에게 "SNAPSHOT TOO OLD" ERROR 발생한다

이러한 ERROR가 나지 않도록 하기 위해서는 UNDO RETENTION 시간을 설정해서

COMMIT에 되더라도 UNDO DATA를 일정시간동안은 계속 유지하도록 해서 위와 같이

UNDO DATA를 계속 참조해서 SELECT를 진행해야 하는 경우 ERROR가 발생되지 않도록 할 수 있다.

UNDO TABLESPACE에 대해서 UNDO ADVISOR 프로그램을 통해 지금까지의 발생된 UNDO DATA의 발생 패턴이나 비율을 기준으로 UNDO TABLESPACE의 적절한 크기와 적절한 UNDO\_RETENTION(보존기간)의 설정값에 대한 가이드를 받을 수 있다. 가이드를 받을 때에는 현재 UNDO TABLESPACE를 고정크기로 잡고(AUTO EXTEND OPTION 은 비활성하고) ADVISE를 받는 것이 좋다.

<< UNDO TABLESPACE >>

관련된 파라미터

[oracle@edydr1p1 ~]$ sqlplus / as sysdba

SQL> show parameter undo

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

temp\_undo\_enabled boolean FALSE

undo\_management string AUTO => UNDO TABLESPACE안에 UNDO SEGMENT에 대한 관리가 자동화

undo\_retention integer 900 => UNDO 데이터의 보존기간 설정

undo\_tablespace string UNDOTBS1 => 현재 사용중인 UNDO TABLESPACE

SQL> SELECT COUNT(\*) FROM DBA\_SEGMENTS

2 WHERE TABLESPACE\_NAME='UNDOTBS1';

=> UNDOTBS1 테이블스페이스 안에 기본적으로 만들어져 있는 UNDO SEGMENT의 갯수

COUNT(\*)

----------

10

<<UNDO TABLESPACE 변경 실습>>

SQL> CREATE UNDO TABLESPACE UNDOT2

2 DATAFILE '/home/oracle/undo2.dbf' size 100m;

Tablespace created.

SQL> show parameter undo

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

temp\_undo\_enabled boolean FALSE

undo\_management string AUTO

undo\_retention integer 900

undo\_tablespace string UNDOTBS1

SQL> alter system set undo\_tablespace='UNDOT2' SCOPE=SPFILE;

System altered.

SQL> SHOW PARAMETER UNDO

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

temp\_undo\_enabled boolean FALSE

undo\_management string AUTO

undo\_retention integer 900

undo\_tablespace string UNDOTBS1

SQL> SHUTDOWN IMMEDIATE

Database closed.

Database dismounted.

ORACLE instance shut down.

SQL> STARTUP

ORACLE instance started.

Total System Global Area 536870912 bytes

Fixed Size 2926472 bytes

Variable Size 276826232 bytes

Database Buffers 251658240 bytes

Redo Buffers 5459968 bytes

Database mounted.

Database opened.

SQL> SHOW PARAMETER UNDO

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

temp\_undo\_enabled boolean FALSE

undo\_management string AUTO

undo\_retention integer 900

undo\_tablespace string **UNDOT2**

SQL> ALTER SYSTEM SET UNDO\_TABLESPACE=UNDOTBS1 SCOPE=SPFILE;

System altered.

SQL> SHUTDOWN IMMEDIATE

Database closed.

Database dismounted.

ORACLE instance shut down.

SQL> STARTUP

ORACLE instance started.

Total System Global Area 536870912 bytes

Fixed Size 2926472 bytes

Variable Size 276826232 bytes

Database Buffers 251658240 bytes

Redo Buffers 5459968 bytes

Database mounted.

Database opened.

SQL> SHOW PARAMETER UNDO

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

temp\_undo\_enabled boolean FALSE

undo\_management string AUTO

undo\_retention integer 900

undo\_tablespace string UNDOTBS1

<< FLASHBACK 기술>>

– Retrieving a Row with Flashback Query

– last\_name = 'Chung' 조건에 해당하는 사원에 대한 2004-04-04 09:30:00 시점의

데이터(UNDO DATA)를 조회

SELECT \* FROM employees **AS OF TIMESTAMP**

**TO\_TIMESTAMP('2004-04-04 09:30:00', 'YYYY-MM-DD HH:MI:SS')**

WHERE last\_name = 'Chung';

INSERT INTO employees

SELECT \* FROM employees AS **OF TIMESTAMP**

**TO\_TIMESTAMP('2004-04-04 09:30:00', 'YYYY-MM-DD HH:MI:SS')**

WHERE last\_name = 'Chung';

--Using Flashback Version Query

--A given row version is valid starting at its time VERSIONS\_START\* up to,

--but not including, its time VERSIONS\_END\*.

--That is, it is valid for any time t such

--that VERSIONS\_START\* <= t < VERSIONS\_END\*. For example,

--the following output indicates that the salary was 10243 from September 9, 2002,

--included, to November 25, 2003, not included.

VERSIONS\_START\_TIME VERSIONS\_END\_TIME SALARY

------------------- ----------------- ------

09-SEP-2003 25-NOV-2003 10243

--Here is a typical Flashback Version Query:

- name = 'JOE' 조건에 해당하는 사원의 RECORD에 대해서

2003-07-18 14:00:00 와 2003-07-18 17:00:00 사이에

변경된 작업 이력정보 조회 (UNDO DATA조회)

SELECT versions\_startscn, versions\_starttime,

versions\_endscn, versions\_endtime,

versions\_xid, versions\_operation,

name, salary

FROM employees

**VERSIONS BETWEEN TIMESTAMP**

**TO\_TIMESTAMP('2003-07-18 14:00:00', 'YYYY-MM-DD HH24:MI:SS')**

**AND TO\_TIMESTAMP('2003-07-18 17:00:00', 'YYYY-MM-DD HH24:MI:SS')**

WHERE name = 'JOE';

Using Flashback Transaction Query – undo data 활용

A Flashback Transaction Query is a query on the view FLASHBACK\_TRANSACTION\_QUERY.

You use a Flashback Transaction Query to obtain transaction information,

including SQL code that you can use to undo each of the changes made by the transaction.

SELECT xid, operation, start\_scn,commit\_scn, logon\_user, undo\_sql

FROM flashback\_transaction\_query

WHERE xid = HEXTORAW('000200030000002D');

SELECT xid, logon\_user FROM flashback\_transaction\_query

WHERE xid IN (SELECT versions\_xid FROM employees VERSIONS BETWEEN TIMESTAMP

TO\_TIMESTAMP('2003-07-18 14:00:00', 'YYYY-MM-DD HH24:MI:SS') AND

TO\_TIMESTAMP('2003-07-18 17:00:00', 'YYYY-MM-DD HH24:MI:SS'));

<<임시테이블(TEMPORARY TABLE)>>

1.임시테이블은 특정 SESSION에서만 테이블에 INSERT/UPDATE한

데이터의 결과를 볼 수 있는 테이블, 데이터는 DB에 저장되는 것이 아니라 메모리에서만 유지된다.

-- COMMIT을 하면 작업한 결과 데이터가 자동으로 삭제된다.

CREATE GLOBAL TEMPORARY TABLE TEST\_TEMP2

(ID NUMBER(10), NAME VARCHAR2(20)) **ON COMMIT DELETE ROWS**;

INSERT INTO TEST\_TEMP2 VALUES (10,'A');

INSERT INTO TEST\_TEMP2 VALUES (20,'B');

SELECT \* FROM TEST\_TEMP2;

– 입력된 데이타가 조회됨

COMMIT; -- 임시 테이블의 데이터가 자동 삭제

SELECT \* FROM TEST\_TEMP2;

– 입력된 데이타가 조회가 안된다.

2. 임시테이블은 특정 SESSION에서만 테이블에 INSERT/UPDATE한 데이터의 결과를 볼 수

있는 테이블

-- COMMIT을 하면 작업 결과 데이타가 자동으로 삭제되지는 않지만 여전히 결과 데이타는

현재 작업하고 있는 세션(CONNECTION)에서만 조회

-- 현재 CONNECTION을 종료하게 되면 임시 테이블의 결과 데이터를 자동으로 삭제된다.

CREATE GLOBAL TEMPORARY TABLE TEST\_TEMP3 (ID NUMBER(10), NAME

VARCHAR2(20)) **ON COMMIT PRESERVE ROWS**;

INSERT INTO TEST\_TEMP3 VALUES (10,'A');

INSERT INTO TEST\_TEMP3 VALUES (20,'B');

SELECT \* FROM TEST\_TEMP3;

– 입력된 데이타가 조회됨

COMMIT; -- 자동 삭제 되지 않고

SELECT \* FROM TEST\_TEMP3;

– 입력된 데이타가 조회됨

-- DB CONNECT를 종료(LOGOUT)할 때 데이터가 자동으로 삭제

<< 임시테이블 예제>>

[oracle@edydr1p1 ~]$ sqlplus hr/oracle

SQL\*Plus: Release 12.1.0.2.0 Production on Sat Jul 31 09:30:36 2021

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

Last Successful login time: Sun Jul 25 2021 09:36:09 +09:00

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> create global temporary table test\_temp (id number) 3 on commit delete rows;

Table created.

Elapsed: 00:00:00.04

SQL> insert into test\_temp values(1000);

1 row created.

Elapsed: 00:00:00.00

SQL> select \* from test\_temp;

ID

----------

1000

1 row selected.

Elapsed: 00:00:00.00

SQL> commit;

Commit complete.

Elapsed: 00:00:00.01

SQL> select \* from test\_temp;

no rows selected

Elapsed: 00:00:00.00

SQL>

create global temporary table test\_temp2

(id number)

**on commit preserve rows;**

Table created.

SQL> insert into test\_temp2

2 values(100);

1 row created.

SQL> select \* from test\_temp2;

ID

----------

100

1 row selected.

SQL> commit;

Commit complete.

SQL> select \* from test\_temp2;

ID

----------

100

1 row selected.

SQL> exit

[oracle@edydr1p1 ~]$ sqlplus hr/oracle

SQL> select \* from test\_temp2;

no rows selected

SQL>EXIT

<<LOCK>>

RX (ROW EXCLUSIVE) LOCK :

INSERT/UPDATE/DELETE작업시 작업하고 있는 RECORD에 걸리는 LOCK

TM LOCK(TABLE MAINTENANCE LOCK) : RX LOCK이 걸려있는 TABLE에 대해서

구조를 변경하는 작업(ALTER TABLE)이 진행되지 못하게 자동으로 걸리는 LOCK.

DML(INSERT/UPDATE/DELETE)를 실행하면

해당 작업대상인 RECORD(ROW)에 대해 RX(ROW EXCLUSIVE) LOCK이 걸린다.

RX LOCK이 걸리게 되면 다른 세션에서 같은 RECORD에 대해서 UPDATE,DELETE를

실행할 수 없다

COMMIT이나 ROLLBACK을 하면 LOCK이 풀리게 된다.

LOCK과 관련해서 사용자 SESSION들은 LOCK이 걸려 있는 ROW에대해

작업을 요청한 순서대로 데이타베이스 안에 있는 QUEUE에 대기하게 된다.

LOCK이 풀리면 대기했던 순서대로 작업을 진행하게 된다. 이것을 ENQUEUE

메커니즘(자동큐 관리)이라고 한다.

lock table employees in exclusive mode (X LOCK)

:TABLE 전체 데이터에 LOCK이 걸린다.

DEADLOCK은 두개의 세션이 서로 LOCK을 걸어놓은 ROW에 대해서 변경작업을 시도하는

경우 양쪽 세션이 모두 WAIT하는 상태를 말한다..

이 경우 오라클은 두 개의 세션중에 하나를 선택해서 마지막 작업을 자동으로

ROLLBACK해서 LOCK을 풀어준다.

<< AUDIT>>

EVALUATE PER SESSION : 한 SESSION에서 처음에 실행된 하나의 명령만 감사하겠다.

EVALUATE PER STATEMENT : 한 SESSION에서 실행되는 모든 작업내용을 감사하겠다.

EVALUATE PER INSTANCE : INSTANCE가 STARTUP한 이후에 처음 실행된 내용만 감사.

<<값 기준 감사방법>>

표준 데이터베이스 감사의 경우는 데이터에 대한 감사는 불가.

값기준 감사의 경우에는 데이타에 대한 감사 가능

값기준 감사의 경우에는 별도의 정책을 사용하지 않고 트리거(TRIGGER) 프로그램을 사용한다.

트리거는 데이터베이스 안에 저장되는 프로그램 (PL/SQL 프로그래밍 언어로 개발)

트리거 프로그램은 이벤트(INSERT/UPDATE/DELETE)에 의해서 실행이 된다

예를 들면 아래와 같이 TRIGGER 프로그램을 통해 변경되고 있는 데이터에 대한 정보를

별도로 다른 테이블에 저장한다던가 LOG 파일에 써 준다거나 하는 LOGIC으로 변경되는

데이터에 대한 추적이 가능하게 하는 방법이 값 기준 AUDIT 방법이다.

CREATE TRIGGER 프로그램이름

AFTER UPDATE ON HR.EMPLOYEES (이벤트)

FOR EACH ROW

WHEN (:NEW.SALARY > 5000)

BEGIN

INSERT INTO TRACE\_TABLE

VALUES( :OLD.SALARY, :NEW.SALARY);

END;

UPDATE HR.EMPLOYEES SET SALARY=8000 ---- (명령을 실행하면 TRIGGER가

자동으로 실행)

OLD:3000 NEW:8000

<< FGA 감사 지침>>

audit\_condition=> NULL (모든 행을 대상으로 감사한다)

audit\_column => NULL (모든 COLUMN을 대상으로 감사한다)

audit\_condition=> 'DEPTID=10' 잘못된 COLUMN정보를 입력했을 경우에는

SELECT \* FROM EMPLOYEES; ORA-28112 발생

audit\_column => 'SAL' 잘못된 COLUMN정보를 주었을 경우 AUDIT기능이 비활성화된다.

SELECT \* FROM EMPLOYEES; 에러가 나지 않지만 AUDIT가 되지 않는다

<< FLASHBACK DROP TABLE>>

[oracle@edydr1p1 ~]$ sqlplus / as sysdba

SQL> show parameter recycle

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

buffer\_pool\_recycle string

db\_recycle\_cache\_size big integer 0

recyclebin string on (활성화가 되어 있어야 한다)

SQL>GRANT UNLIMITED TABLESPACE TO HR;

SQL> connect hr/oracle

Connected.

SQL> create table emp100

2 as select \* from employees;

Table created.

SQL> drop table emp100;

Table dropped.

SQL> show recyclebin

ORIGINAL NAME RECYCLEBIN NAME OBJECT TYPE DROP TIME

---------------- ------------------------------ ------------ -------------------

EMP100 BIN$VD9a3oHAsVjgQL4KfWI/DQ==$0 TABLE 2008-08-12:16:38:16

SQL> flashback table emp100 to before drop;

Flashback complete.

SQL> show recyclebin

SQL> SELECT \* FROM emp100; – 복원된 것을 확인

SQL> create index emp100\_index

2 on emp100(sal);

Index created.

SQL> drop table emp100;

Table dropped.

SQL> select index\_name from user\_indexes where table\_name = 'EMP100';

no rows selected – 테이블이 삭제되면 인덱스도 같이 삭제됨

SQL> flashback table emp100 to before drop; 인덱스도 같이 복원

Flashback complete.

SQL> select index\_name from user\_indexes where table\_name = 'EMP100';

INDEX\_NAME

------------------------------

BIN$VD9a3oHBsVjgQL4KfWI/DQ==$0

SQL> ALTER INDEX "BIN$VD9a3oHBsVjgQL4KfWI/DQ==$0"

RENAME TO EMP100\_SAL;

<<FLASHBACK TABLE 예제>>

SQL> select systimestamp from dual;

SYSTIMESTAMP

---------------------------------------------------------------------------

12-AUG-08 04.48.16.727251 PM +09:00

SQL> delete from emp100;

14 rows deleted.

SQL> commit;

Commit complete.

SQL> alter table emp100 enable row movement;

Table altered.

SQL> flashback table emp100 to timestamp(systimestamp - interval '6' minute);

– 현재기준으로 emp100 테이블의 데이터를 6분전의 상태(delete 전의 상태)로 되돌려라

– UNDO DATA가 이용된다.

Flashback complete.

SQL> select count(\*) from emp100; – 6분전 데이터 상태로 복원이 됨을 확인할 수 있다.