[Skip to end of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=41888683" \l "page-metadata-end)

* 작성자 : [정관호 Kyle [X]](https://wiki.gsretail.com/display/~junggh) - [2020-11-24](https://wiki.gsretail.com/pages/viewpreviousversions.action?pageId=41888683)

[Go to start of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=41888683#page-metadata-start)

**DB접근제어(DB세이퍼) 솔루션을 통해 SBC에서 서비스 DB계정으로 DB접속을 제어하는 로직을 오라클 로그온-트리거를 통해 구현**

**SYS\_CONTEXT함수를 통한 클라이언트 세션 정보**

|  |
| --- |
|  |

**로그온 트리거**

|  |
| --- |
| -- 테스트용 계정  DROP USER GS\_TEST;  CREATE USER GS\_TEST Identified by "gs#ora!@34";  GRANT CONNECT TO GS\_TEST;    DROP USER I\_TEST;  CREATE USER I\_TEST identified by "gs#ora!@34";  GRANT CONNECT TO I\_TEST;      -- 한글 메세지  CREATE OR REPLACE TRIGGER SYS.DENY\_USER    AFTER LOGON ON DATABASE  BEGIN              IF (SYS\_CONTEXT('USERENV','HOST') LIKE 'HOSTWAY\SBC%')              THEN                  IF (SYS\_CONTEXT('USERENV','SESSION\_USER') IN('GSDBA','DBMON')) -- DBA 사용 계정이거나                  OR (SYS\_CONTEXT('USERENV','SESSION\_USER') LIKE '%ERD') -- ERD툴 계정이거나                  OR (SYS\_CONTEXT('USERENV','SESSION\_USER') LIKE'I@\_%' ESCAPE '@') -- 개인계정이면 허용                  THEN                      NULL;                  ELSE                      RAISE\_APPLICATION\_ERROR(-20001,CHR(10)                      ||'[경고] SBC를 통한 DB접속 시 개인계정(I\_XXX) 외 접근 불가'||CHR(10)                      ||' - SBC IP : '||SYS\_CONTEXT('USERENV','IP\_ADDRESS')||'   사용자 : '||SYS\_CONTEXT('USERENV','OS\_USER')||CHR(10)                      ||' - 접속계정 : '||SYS\_CONTEXT('USERENV','SESSION\_USER')||'   접속모듈 : '||SYS\_CONTEXT('USERENV','MODULE')||CHR(10)                      ||'예외 사항에 대해서는 DBA팀에 문의주세요!'                      );              END IF;        END IF;  END;  /    -- DB별 캐릭터셋이 상이한 이슈로 한글이 깨지는 경우 존재  CREATE OR REPLACE TRIGGER SYS.DENY\_USER    AFTER LOGON ON DATABASE  BEGIN              IF (SYS\_CONTEXT('USERENV','HOST') LIKE 'HOSTWAY\SBC%')              THEN                  IF (SYS\_CONTEXT('USERENV','SESSION\_USER') IN('GSDBA','DBMON')) -- DBA 사용 계정이거나                  OR (SYS\_CONTEXT('USERENV','SESSION\_USER') LIKE '%ERD') -- ERD툴 계정이거나                  OR (SYS\_CONTEXT('USERENV','SESSION\_USER') LIKE'I@\_%' ESCAPE '@') -- 개인계정이면 허용                  THEN                      NULL;                  ELSE                      RAISE\_APPLICATION\_ERROR(-20001,CHR(10)                      ||'[Warning] Unable to access database with service account.'||CHR(10)                      ||' Please use your personal DB account(I\_XXX).'||CHR(10)||CHR(10)                      ||' - SBC IP addr : '||SYS\_CONTEXT('USERENV','IP\_ADDRESS')||'   EMP NO : '||SYS\_CONTEXT('USERENV','OS\_USER')||CHR(10)                      ||' - DB Account : '||SYS\_CONTEXT('USERENV','SESSION\_USER')||'   Module : '||SYS\_CONTEXT('USERENV','MODULE')||CHR(10)||CHR(10)                      ||' Contact GSDBA Team for more information.'||CHR(10)||CHR(10)                      );              END IF;        END IF;  END;  /    -- 9i는 ESCAPE함수 에러발생하므로 제거  CREATE OR REPLACE TRIGGER SYS.DENY\_USER    AFTER LOGON ON DATABASE  BEGIN              IF (SYS\_CONTEXT('USERENV','HOST') LIKE 'HOSTWAY\SBC%')              THEN                  IF (SYS\_CONTEXT('USERENV','SESSION\_USER') IN('GSDBA','DBMON')) -- DBA 사용 계정이거나                  OR (SYS\_CONTEXT('USERENV','SESSION\_USER') LIKE '%ERD') -- ERD툴 계정이거나                  OR (SYS\_CONTEXT('USERENV','SESSION\_USER') LIKE'I%') -- 개인계정이면 허용                  THEN                      NULL;                  ELSE                      RAISE\_APPLICATION\_ERROR(-20001,CHR(10)                      ||'[Warning] Unable to access database with service account.'||CHR(10)                      ||' Please use your personal DB account(I\_XXX).'||CHR(10)||CHR(10)                      ||' - SBC IP addr : '||SYS\_CONTEXT('USERENV','IP\_ADDRESS')||'   EMP NO : '||SYS\_CONTEXT('USERENV','OS\_USER')||CHR(10)                      ||' - DB Account : '||SYS\_CONTEXT('USERENV','SESSION\_USER')||CHR(10)||CHR(10)                      ||' Contact GSDBA Team for more information.'||CHR(10)||CHR(10)                      );              END IF;        END IF;  END;  / |

[Skip to end of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=22256238#page-metadata-end)

* 작성자 : 알 수 없는 사용자 (jung.gh) - [2018-12-19](https://wiki.gsretail.com/pages/viewpreviousversions.action?pageId=22256238)

[Go to start of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=22256238#page-metadata-start)

**넷백업 코파일럿 기능을 사용하는 전제조건으로 공유저장소에 아카이브가 있어야 함**  
  
 - IBM GPFS 라이선스가 너무 비쌈  
 - NFS를 사용하는 것은 오라클이 권고하지 않는다.  
 - ACFS도 11.2.0.4 버전부터는 안정적이나, N/W를 사용하는 개념으로 현재 리눅스 기반이 아닌 DB서버에서 사용하고 있는 사례 없음  
 - 삼성 반도체/하이닉스 사용 예로 오라클 ASM을 사용하여 아카이브 저장(Archive\_dest1....2...3.. 에 필수/옵션 설정)

**작업 내역**

|  |
| --- |
| tst-hssmtcdb01:oragrid/ > asmcmd  Can not create path /oragrid/11.2.0/grid/log/diag/asmcmd/user\_oragrid/tst-hssmtcdb01/alert  Can not create path /oragrid/11.2.0/grid/log/diag/asmcmd/user\_oragrid/tst-hssmtcdb01/trace  mkdir -p /oragrid/11.2.0/grid/log/diag/asmcmd/user\_oragrid/tst-hssmtcdb01/alert  mkdir -p /oragrid/11.2.0/grid/log/diag/asmcmd/user\_oragrid/tst-hssmtcdb01/trace    sqlplus "/as sysasm"    SQL> create diskgroup DG\_ARCH external redundancy disk '/dev/r133vg06','/dev/r134vg06','/dev/r135vg06','/dev/r136vg06';  create diskgroup DG\_ARCH external redundancy disk '/dev/r133vg06','/dev/r134vg06','/dev/r135vg06','/dev/r136vg06'  \*  ERROR at line 1:  ORA-15018: diskgroup cannot be created  ORA-15031: disk specification '/dev/r136vg06' matches no disks  ORA-15014: path '/dev/r136vg06' is not in the discovery set  ORA-15031: disk specification '/dev/r135vg06' matches no disks  ORA-15014: path '/dev/r135vg06' is not in the discovery set  ORA-15031: disk specification '/dev/r134vg06' matches no disks  ORA-15014: path '/dev/r134vg06' is not in the discovery set  ORA-15031: disk specification '/dev/r133vg06' matches no disks  ORA-15014: path '/dev/r133vg06' is not in the discovery set    alter system set asm\_diskstring='/dev/r\*vg06';  SQL> show parameter asm  NAME                                 TYPE                              VALUE  ------------------------------------ --------------------------------- ------------------------------  asm\_diskgroups                       string                            DG\_ARCH  asm\_diskstring                       string                            /dev/r\*vg06  asm\_power\_limit                      integer                           1  asm\_preferred\_read\_failure\_groups    string    SQL> create diskgroup DG\_ARCH external redundancy disk '/dev/r133vg06','/dev/r134vg06','/dev/r135vg06','/dev/r136vg06';  Diskgroup created.  SQL> set line 200  SQL> col disk\_group for a10  SQL> col label for a20  SQL> col state for a10  SQL> select a.name as disk\_group, d.name "Label", a.state    2  from v$asm\_disk d, v$asm\_diskgroup a    3  where d.group\_number=a.group\_number    4  order by 2;  DISK\_GROUP Label                STATE  ---------- -------------------- ----------  DG\_ARCH    DG\_ARCH\_0000         MOUNTED  DG\_ARCH    DG\_ARCH\_0001         MOUNTED  DG\_ARCH    DG\_ARCH\_0002         MOUNTED  DG\_ARCH    DG\_ARCH\_0003         MOUNTED  SQL> col group\_number for 999  SQL> col disk\_number for 999  SQL> col name for a10  SQL> col mount\_status for a10  SQL> col path for a15  SQL> select group\_number, disk\_number, name, mount\_status, path, total\_mb from v$asm\_disk;  GROUP\_NUMBER DISK\_NUMBER NAME            MOUNT\_STATUS         PATH            File Size (MB)  ------------ ----------- --------------- -------------------- --------------- --------------             1           3 DG\_ARCH\_0003    CACHED               /dev/r136vg06           20,480             1           0 DG\_ARCH\_0000    CACHED               /dev/r133vg06           20,480             1           1 DG\_ARCH\_0001    CACHED               /dev/r134vg06           20,480             1           2 DG\_ARCH\_0002    CACHED               /dev/r135vg06           20,480                                                                                --------------  Grand Total:                                                                          81,920    ASMCMD> lsdg  State    Type    Rebal  Sector  Block       AU  Total\_MB  Free\_MB  Req\_mir\_free\_MB  Usable\_file\_MB  Offline\_disks  Voting\_files  Name  MOUNTED  EXTERN  N         512   4096  1048576     81920    81766                0           81766              0             N  DG\_ARCH/  tst-hssmtcdb01:oragrid/ > crsctl stat res -t  --------------------------------------------------------------------------------  NAME           TARGET  STATE        SERVER                   STATE\_DETAILS  --------------------------------------------------------------------------------  Local Resources  --------------------------------------------------------------------------------  ora.DATA.dg                 OFFLINE OFFLINE      tst-hssmtcdb01                 OFFLINE OFFLINE      tst-hssmtcdb02  ora.DG\_ARCH.dg                 ONLINE  ONLINE       tst-hssmtcdb01                 ONLINE  ONLINE       tst-hssmtcdb02  select name, group\_number, path,state, header\_status , mode\_status from v$asm\_disk;    # Node 1  LOG\_ARCHIVE\_DEST\_1 = 'LOCATION=/arch1' MANDATORY  LOG\_ARCHIVE\_DEST\_2 = 'LOCATION=+DG\_ARCH' OPTIONAL  # Node 2  LOG\_ARCHIVE\_DEST\_1 = 'LOCATION=/arch2' MANDATORY  LOG\_ARCHIVE\_DEST\_2 = 'LOCATION=+DG\_ARCH' OPTIONAL    SQL> archive log list;  Database log mode              Archive Mode  Automatic archival             Enabled  Archive destination            +DG\_ARCH  Oldest online log sequence     97  Next log sequence to archive   106  Current log sequence           106  # 참고  https://h391106.tistory.com/287#recentComments |

[Skip to end of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=22250771#page-metadata-end)

* 작성자 : 알 수 없는 사용자 (jung.gh) - [2018-08-06](https://wiki.gsretail.com/pages/viewpreviousversions.action?pageId=22250771)

[Go to start of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=22250771#page-metadata-start)

|  |
| --- |
| # 방법1: 내부 JAVA 클래스 사용  CREATE OR REPLACE PROCEDURE GSDBA.SLEEP (P\_MILLI\_SECONDS IN NUMBER)  AS LANGUAGE JAVA NAME 'java.lang.Thread.sleep(long)';  14:33:12 DEVSMTC SQL>SET SERVEROUTPUT ON ;  14:33:21 DEVSMTC SQL>BEGIN  14:33:21   2      DBMS\_OUTPUT.PUT\_LINE('Start ' || to\_char(SYSDATE, 'YYYY-MM-DD HH24:MI:SS'));  14:33:21   3      GSDBA.SLEEP(5 \* 1000);  14:33:21   4      DBMS\_OUTPUT.PUT\_LINE('End   ' || to\_char(SYSDATE, 'YYYY-MM-DD HH24:MI:SS'));  14:33:21   5  END;  14:33:21   6  /  Start 2018-08-06 14:33:21  End   2018-08-06 14:33:26  PL/SQL procedure successfully completed.  Elapsed: 00:00:05.37  # 방법2: DB 패키지 사용  grant execute on SYS.DBMS\_LOCK to GSDBA;  CREATE OR REPLACE PROCEDURE GSDBA.SP\_SLEEP IS  BEGIN     SYS.DBMS\_LOCK.sleep(10);  END;  /  SET SERVEROUTPUT ON ;  BEGIN      DBMS\_OUTPUT.PUT\_LINE('Start ' || to\_char(SYSDATE, 'YYYY-MM-DD HH24:MI:SS'));      GSDBA.SP\_SLEEP ;      DBMS\_OUTPUT.PUT\_LINE('End   ' || to\_char(SYSDATE, 'YYYY-MM-DD HH24:MI:SS'));  END;  /    # PLSQL Sleep 가이드  해당 프로시저 호출 시 10초간 Sleep 수행  GSDBA.SP\_SLEEP ;  Ex:>  DEVSMTC SQL>SET SERVEROUTPUT ON ;  DEVSMTC SQL>BEGIN     2    DBMS\_OUTPUT.PUT\_LINE('Start ' || to\_char(SYSDATE, 'YYYY-MM-DD HH24:MI:SS'));     3    GSDBA.SP\_SLEEP ;     4    DBMS\_OUTPUT.PUT\_LINE('End   ' || to\_char(SYSDATE, 'YYYY-MM-DD HH24:MI:SS'));     5  END;     6  /  Start 2018-08-06 14:43:55  End   2018-08-06 14:44:05  PL/SQL procedure successfully completed.  Elapsed: 00:00:10.01 |

[Skip to end of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=22259504#page-metadata-end)

* 작성자 : 알 수 없는 사용자 (jung.gh), 최근 변경 : [정관호 Kyle [X]](https://wiki.gsretail.com/display/~junggh) - [2020-07-31](https://wiki.gsretail.com/pages/diffpagesbyversion.action?pageId=22259504&selectedPageVersions=1&selectedPageVersions=2)

[Go to start of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=22259504#page-metadata-start)

참고: <http://plsqlexecoscomm.sourceforge.net/>

|  |
| --- |
| -- GSDBA계정으로  CREATE DATABASE LINK DL\_SMS\_PRD  CONNECT TO GSDBA identified by inomuski  USING '(DESCRIPTION=(ADDRESS\_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=218.237.122.41)(PORT=1521)))(CONNECT\_DATA=(SERVICE\_NAME=SMSPRD)))';    SQL> !pwd  /oracle/DBA/WORK/PLSQL\_OSCOMMAND/src/11g  SQL> @install  \*  \*      \*\*      \*\*  \*    \*\*\*\*    \*\*\*\*  \*      \*\*      \*\*      \*\*\*\*\*\*  \*      \*\*      \*\*    \*\*     \*\*  \*      \*\*      \*\*      \*\*\*\*\*\*\*  \*      \*\*      \*\*           \*\*  \*    \*\*\*\*\*\*  \*\*\*\*\*\*  \*\*\*\*\*\*\*  \*    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\* 1. Installing Java Code ...  \*\*  ... java source FILE\_TYPE  No errors.  ... compiling java FILE\_TYPE  No errors.  ... java source OS\_COMMAND  No errors.  ... compiling java OS\_COMMAND  No errors.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\* 2. PL/SQL Package Specs  \*\*  ... type spec FILE\_TYPE  No errors.  ... type spec FILE\_LIST\_TYPE  No errors.  ... package spec FILE\_PKG  No errors.  ... package spec OS\_COMMAND  No errors.  ... package spec LOB\_WRITER\_PLSQL (deprecated)  No errors.  ... package spec FILE\_SECURITY  No errors.  ... package spec FILE\_PKG\_VERSION  No errors.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\* 3. PL/SQL Package Bodys  \*\*  ... type body FILE\_TYPE  No errors.  ... package body FILE\_PKG  No errors.  ... package body OS\_COMMAND  No errors.  ... package body LOB\_WRITER\_PLSQL (deprecated) ...  No errors.  ... package body FILE\_SECURITY  No errors.  ... package body FILE\_PKG\_VERSION  No errors.    -- sys 계정으로  EXEC dbms\_java.grant\_permission('GSDBA','SYS:java.lang.RuntimePermission','writeFileDescriptor', '');  EXEC dbms\_java.grant\_permission('GSDBA','SYS:java.lang.RuntimePermission','readFileDescriptor', '');  EXEC dbms\_java.grant\_permission('GSDBA','SYS:java.io.FilePermission','/bin/ls', 'execute');  EXEC dbms\_java.grant\_permission('GSDBA','SYS:java.io.FilePermission','/bin/sh', 'execute');    -- gsdba계정으로 테스트    SQL> select os\_command.exec\_clob('/bin/ls -al /') from dual;    OS\_COMMAND.EXEC\_CLOB('/BIN/LS-AL/')  --------------------------------------------------------------------------------  �⑷퀎 150  dr-xr-xr-x.  32 root     root      4096 2018-12-04 15:23 .  dr-xr-x      SQL> select os\_command.exec\_clob('/bin/sh /oracle/DBA/MONITORING/SLACK\_SEND\_MESSEAGE.sh test') from dual;    OS\_COMMAND.EXEC\_CLOB('/BIN/SH/ORACLE/DBA/MONITORING/SLACK\_SEND\_MESSEAGE.SHTEST')  --------------------------------------------------------------------------------  ok |

[# 설치파일  
PLSQL\_OSCOMMAND\_1.0.zip](https://wiki.gsretail.com/download/attachments/22259504/PLSQL_OSCOMMAND_1.0.zip?version=1&modificationDate=1515058409000&api=v2)

[Skip to end of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=22256024#page-metadata-end)

* 알 수 없는 사용자 (jung.gh)님이 작성, [2020-02-27](https://wiki.gsretail.com/pages/diffpagesbyversion.action?pageId=22256024&selectedPageVersions=8&selectedPageVersions=9)에 최종 변경

[Go to start of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=22256024#page-metadata-start)

**쿠폰 대량 작업 시 대량 Delete에 대한 시간소비/DB부하를 줄이기 위해 개인계정으로 아래의 프로시저 수행 부탁드립니다.**

# 수행방법  
exec GSDBA.SP\_TRUUNC\_TMP\_TABLES('테이블명');  
ex>  
SQL> exec GSDBA.SP\_TRUUNC\_TMP\_TABLES('TEMP\_TEST');  
PL/SQL이 실행되었습니다.

# 실행 권한

GRANT EXECUTE ON GSDBA.SP\_TRUUNC\_TMP\_TABLES TO I\_CLEARPOND;  
GRANT EXECUTE ON GSDBA.SP\_TRUUNC\_TMP\_TABLES TO I\_YMLEE;  
GRANT EXECUTE ON GSDBA.SP\_TRUUNC\_TMP\_TABLES TO I\_PYJ77;  
GRANT EXECUTE ON GSDBA.SP\_TRUUNC\_TMP\_TABLES TO I\_DHBAEK;

 # 모니터링  
해당 프로시저 수행 시 DBA에게 슬랙을 통해 알림

|  |
| --- |
| CREATE OR REPLACE PROCEDURE GSDBA.SP\_TRUUNC\_TMP\_TABLES(vTB\_NM IN VARCHAR2)   IS   /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*    1.프로그램 명 : SP\_TRUUNC\_TMP\_TABLES    2.기 능 정 의 : 개발자가 작업테이블에 대한 Truncate수행    3.참 고 사 항 : 허가 된 테이블에 한해 수행되며, 수행 후 슬랙으로 메세지 발송됨    4.변 경 이 력 :     Version  작성자   소   속   일    자   요청자       내       용   --------  ------  --------  ----------  ------  ---------------------      1.0    정관호            2018.12.19  백동현  최초작성    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  -- 변수 선언부 Pre Fix s=String, t=Timestamp, n=Number, v=Variable  sModule\_NM VARCHAR2(100);  sAction\_NM VARCHAR2(100);  tStart\_Date TIMESTAMP;  tEnd\_Date TIMESTAMP;  sDB\_NM VARCHAR2(30);  sUSERNAME VARCHAR2(100);  sSQL\_Trunc VARCHAR2(1000);  SLACK\_MESSAGE VARCHAR2(1000);  SCALL\_SP\_SLACK\_SNED\_TEMP VARCHAR2(2000);  sErr\_Code VARCHAR2(20);  sErr\_Msg VARCHAR2(200);  OTHERS EXCEPTION;    BEGIN      tStart\_Date := SYSTIMESTAMP;      DBMS\_APPLICATION\_INFO.SET\_MODULE('SP\_TRUUNC\_TMP\_TABLES Procedure', 'Execute Truncate Table by Developer');      DBMS\_APPLICATION\_INFO.READ\_MODULE(sModule\_NM,sAction\_NM);        SELECT USER INTO sUSERNAME FROM DUAL;      SELECT UPPER(TRANSLATE(NAME,'+.0123456789',' ')) INTO sDB\_NM FROM V$DATABASE;            IF (vTB\_NM IN ('TEMP\_PRD','TEMP\_PRD\_NEW','TEMP\_PRD\_OLD','TEMP\_PRD2','TEMP\_PRD3', 'TEMP\_TEST')) THEN                -- 슬랙 메세지 발송              SLACK\_MESSAGE := '[DBA모니터링-'||sDB\_NM||']  테이블 Truncate 수행: '||vTB\_NM||'   수행유저: '||sUSERNAME;              sCALL\_SP\_SLACK\_SNED\_TEMP := q'[BEGIN DBMON.SP\_SLACK\_SNED\_TEMP@DL\_METADB\_DBMON(' ]'||SLACK\_MESSAGE||q'['); END;]';  --            DBMS\_OUTPUT.PUT\_LINE(sCALL\_SP\_SLACK\_SNED\_TEMP);              EXECUTE IMMEDIATE sCALL\_SP\_SLACK\_SNED\_TEMP;                -- Truncate수행              sSQL\_Trunc := 'TRUNCATE TABLE GSDBA.'||vTB\_NM;  --            DBMS\_OUTPUT.PUT\_LINE(sSQL\_Trunc);              EXECUTE IMMEDIATE sSQL\_Trunc;          ELSE              DBMS\_OUTPUT.PUT\_LINE('에러: '||vTB\_NM||' Truncate 권한이 없는 테이블입니다. DBA에게 문의 주세요');            END IF;      tEnd\_Date := SYSTIMESTAMP;  -- 에러 로그 기록 후 예외 처리      EXCEPTION          WHEN OTHERS THEN              tEnd\_Date := SYSTIMESTAMP;              DBMS\_OUTPUT.PUT\_LINE('ERROR : '||SQLERRM||'');              sErr\_Code := SQLCODE;              sErr\_Msg := SUBSTR(SQLERRM, 1, 200);   END; |

**추가 대상 테이블 있어 프로시저 변경함(2019.04.16)**

> 4개 테이블 에 대한 TRUNCATE 프로시져 4개 생성 (SMTC\_OWN.ORD\_DAILY\_ORD\_NO\_DEL\_D, SMTC\_OWN.ORD\_DAILY\_ORD\_ITEM\_DEL\_D, EPR\_OWN.EPR\_DAILY\_ORD\_NO\_DEL\_D, EPR\_OWN.EPR\_DAILY\_ORD\_ITEM\_DEL\_D)

**GSDBA 가 다른 계정 테이블 TRUNCATE 수행 시 DROP ANY TABLE 권한 필요함.**

> GRANT DROP ANY TABLE TO GSDBA;

**수행 방법**

> EXEC GSDBA.SP\_TRUNC\_TMP\_TABLES('SMTC\_OWN','ORD\_DAILY\_ORD\_NO\_DEL\_D');

|  |
| --- |
| CREATE OR REPLACE PROCEDURE GSDBA.SP\_TRUNC\_TMP\_TABLES(vOWNER IN VARCHAR2, vTB\_NM IN VARCHAR2)   IS   /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*    1.프로그램 명 : SP\_TRUNC\_GS\_POPUL\_PRD\_D\_TEST    2.기 능 정 의 : 상품판매지수 ordering 을 위한 배치에서 ETL이 수행했던 기능을 제거하고 배치(156)에서 수행   3.참 고 사 항 : 매일 짝수시 30분 ( 2시간 주기):  gsPopularPrdBatch.sh    4.변 경 이 력 :     Version  작성자   소   속   일    자   요청자       내       용   --------  ------  --------  ----------  ------  ---------------------      1.0    정관호            2018.12.19  백동현  최초작성      2.0    이정아            2019.04/16          OWNER 추가 및 대상 테이블 추가 4건    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  -- 변수 선언부 Pre Fix s=String, t=Timestamp, n=Number, v=Variable  sModule\_NM VARCHAR2(100);  sAction\_NM VARCHAR2(100);  tStart\_Date TIMESTAMP;  tEnd\_Date TIMESTAMP;  sDB\_NM VARCHAR2(30);  sUSERNAME VARCHAR2(100);  sSQL\_Trunc VARCHAR2(1000);  SLACK\_MESSAGE VARCHAR2(1000);  SCALL\_SP\_SLACK\_SNED\_TEMP VARCHAR2(2000);  sErr\_Code VARCHAR2(20);  sErr\_Msg VARCHAR2(200);  OTHERS EXCEPTION;    BEGIN      tStart\_Date := SYSTIMESTAMP;      DBMS\_APPLICATION\_INFO.SET\_MODULE('SP\_TRUNC\_TMP\_TABLES Procedure', 'Execute Truncate Table by Developer');      DBMS\_APPLICATION\_INFO.READ\_MODULE(sModule\_NM,sAction\_NM);        SELECT USER INTO sUSERNAME FROM DUAL;      SELECT UPPER(TRANSLATE(NAME,'+.0123456789',' ')) INTO sDB\_NM FROM V$DATABASE;            /\* 20190416 수정 대상 테이블 추가됨 \*/          IF (vTB\_NM IN ('TEMP\_PRD','TEMP\_PRD\_NEW','TEMP\_PRD\_OLD','TEMP\_PRD2','TEMP\_PRD3', 'TEMP\_TEST'                        ,'ORD\_DAILY\_ORD\_NO\_DEL\_D','ORD\_DAILY\_ORD\_ITEM\_DEL\_D','EPR\_DAILY\_ORD\_NO\_DEL\_D','EPR\_DAILY\_ORD\_ITEM\_DEL\_D')) THEN                -- 슬랙 메세지 발송              SLACK\_MESSAGE := '[DBA모니터링-'||sDB\_NM||']  테이블 Truncate 수행: '||vTB\_NM||'   수행유저: '||sUSERNAME;              sCALL\_SP\_SLACK\_SNED\_TEMP := q'[BEGIN DBMON.SP\_SLACK\_SNED\_TEMP@DL\_METADB\_DBMON(' ]'||SLACK\_MESSAGE||q'['); END;]';  --            DBMS\_OUTPUT.PUT\_LINE(sCALL\_SP\_SLACK\_SNED\_TEMP);              EXECUTE IMMEDIATE sCALL\_SP\_SLACK\_SNED\_TEMP;                -- Truncate수행              /\* 20190416 수정 OWNER 부분 추가됨 \*/              sSQL\_Trunc := 'TRUNCATE TABLE '||vOWNER||'.'||vTB\_NM;  --            DBMS\_OUTPUT.PUT\_LINE(sSQL\_Trunc);              EXECUTE IMMEDIATE sSQL\_Trunc;          ELSE              DBMS\_OUTPUT.PUT\_LINE('에러: '||vTB\_NM||' Truncate 권한이 없는 테이블입니다. DBA에게 문의 주세요');            END IF;      tEnd\_Date := SYSTIMESTAMP;  -- 에러 로그 기록 후 예외 처리      EXCEPTION          WHEN OTHERS THEN              tEnd\_Date := SYSTIMESTAMP;              DBMS\_OUTPUT.PUT\_LINE('ERROR : '||SQLERRM||'');              sErr\_Code := SQLCODE;              sErr\_Msg := SUBSTR(SQLERRM, 1, 200);   END;  / |

**WEBDB에 서비스계정의 Truncate 사용요구가 발생하여, 기존 개인계정에 한에 허용되던 Truc프로시저 변경**

1. **프로시저 내 IN절에 OWNER.TABLE\_NAME 이 정확한가?  
       2. 개인/서비스 계정 구분  
         2.1. 개인개정  
           - 슬랙발송  
         2.2. 서비스계정  
           - N/A**

**1.1. GSDBA.GSDBA\_TRUNC\_L 내 수행이력  
        1.2. Truncate 수행**

**- 테스트 결과**

|  |
| --- |
| -- 테스트  CREATE TABLE GSDBA.GSDBA\_TEST1 AS SELECT \* FROM DBA\_TABLES;  CREATE TABLE STORE.GSDBA\_TEST2 AS SELECT \* FROM DBA\_TABLES;  CREATE TABLE I\_JUNGGH.GSDBA\_TEST3 AS SELECT \* FROM DBA\_TABLES;    -- # DBA권한을 가진 GSDBA계정 소유의 프로시저로 프로시저 IN절 내 조건만 맞으면 모든 DB계정이 Trucate가능  -- 개인계정이 사용 시  SQL> conn I\_JUNGGH/"Wjdrhksgh!23"  Connected.    SQL> exec GSDBA.SP\_TRUNC\_TABLES('GSDBA','GSDBA\_TEST1');  TRUNCATE TABLE GSDBA.GSDBA\_TEST1  Truncate Table 성공  PL/SQL procedure successfully completed.    SQL> exec GSDBA.SP\_TRUNC\_TABLES('STORE','GSDBA\_TEST2');  TRUNCATE TABLE STORE.GSDBA\_TEST2  Truncate Table 성공  PL/SQL procedure successfully completed.    SQL> exec GSDBA.SP\_TRUNC\_TABLES('I\_JUNGGH','GSDBA\_TEST3');  TRUNCATE TABLE I\_JUNGGH.GSDBA\_TEST3  Truncate Table 성공  PL/SQL procedure successfully completed.    -- 서비스 계정도 당연히 사용 가능  SQL> conn store/store  Connected.  SQL> exec GSDBA.SP\_TRUNC\_TABLES('STORE','GSDBA\_TEST2');  TRUNCATE TABLE STORE.GSDBA\_TEST2  Truncate Table 성공  PL/SQL procedure successfully completed.    -- IN절의 조건(소유자+테이블명) 하나라도 불일치 시 권한 오류 발생  SQL> exec GSDBA.SP\_TRUNC\_TABLES('STORE','GSDBA\_TEST4');  에러: GSDBA\_TEST4 Truncate 권한이 없는 테이블입니다. DBA에게 문의 주세요  PL/SQL procedure successfully completed.    SQL> exec GSDBA.SP\_TRUNC\_TABLES('STORE\_TEST','GSDBA\_TEST2');  에러: GSDBA\_TEST2 Truncate 권한이 없는 테이블입니다. DBA에게 문의 주세요  PL/SQL procedure successfully completed. |

**개인계정이 trucate 수행 시 슬랙발송 내용**

**-Trucate 이력 저장 테이블**

|  |
| --- |
| DROP TABLE GSDBA.GSDBA\_TRUNC\_L;  CREATE TABLE GSDBA.GSDBA\_TRUNC\_L  (      DB\_NM           VARCHAR2(20),      USER\_TYP         VARCHAR2(20),      EXEC\_USERNAME   VARCHAR2(20),      EXEC\_DTM        DATE,      OWNER           VARCHAR2(20),      TABLE\_NAME      VARCHAR2(30),      HOST            VARCHAR2(30),      OS\_USER         VARCHAR2(30),      MODULE          VARCHAR2(30),      PGM\_PASS        VARCHAR2(30),      ERR\_CODE        VARCHAR2(200)  ) TABLESPACE ETCDATA;  COMMENT ON TABLE GSDBA.GSDBA\_TRUNC\_L IS 'GS홈쇼핑 DB관리용 - Truncate수행이력테이블';  COMMENT ON COLUMN GSDBA.GSDBA\_TRUNC\_L.DB\_NM IS 'DB명';  COMMENT ON COLUMN GSDBA.GSDBA\_TRUNC\_L.USER\_TYP IS '서비스/개인';  COMMENT ON COLUMN GSDBA.GSDBA\_TRUNC\_L.EXEC\_USERNAME IS '실행DB유저';  COMMENT ON COLUMN GSDBA.GSDBA\_TRUNC\_L.EXEC\_DTM IS '실행시간';  COMMENT ON COLUMN GSDBA.GSDBA\_TRUNC\_L.OWNER IS '대상오너';  COMMENT ON COLUMN GSDBA.GSDBA\_TRUNC\_L.TABLE\_NAME IS '대상테이블';  COMMENT ON COLUMN GSDBA.GSDBA\_TRUNC\_L.HOST IS '수행서버 hostname';  COMMENT ON COLUMN GSDBA.GSDBA\_TRUNC\_L.OS\_USER IS '수행서버 OS유저';  COMMENT ON COLUMN GSDBA.GSDBA\_TRUNC\_L.MODULE IS '수행서버 프로그램';  COMMENT ON COLUMN GSDBA.GSDBA\_TRUNC\_L.PGM\_PASS IS '성공여부';  COMMENT ON COLUMN GSDBA.GSDBA\_TRUNC\_L.ERR\_CODE IS '에러코드'; |

**- 프로시저 소스**

|  |
| --- |
| CREATE OR REPLACE PROCEDURE GSDBA.SP\_TRUNC\_TABLES(vOWNER IN VARCHAR2, vTB\_NM IN VARCHAR2)   IS   /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*    1.프로그램 명 : SP\_TRUNC\_TABLES    2.기 능 정 의 : 서비스/개인 DB계정으로 Truncate 가능하게 구현    3.참 고 사 항 :    4.변 경 이 력 :     Version  작성자   소   속   일    자   요청자       내       용   --------  ------  --------  ----------  ------  ---------------------      1.0    정관호            2020.02.27           최초작성,WEBDB용 Truncate 프로시저    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  -- 변수 선언부 Pre Fix s=String, d=date t=Timestamp, n=Number, v=Variable  sModule\_NM VARCHAR2(100);  sAction\_NM VARCHAR2(100);  dStart\_Date DATE;  dEnd\_Date DATE;  sDB\_NM VARCHAR2(30);  sSESSION\_USER VARCHAR2(100);  sHOST VARCHAR2(100);  sOS\_USER VARCHAR2(100);  sMODULE VARCHAR2(100);  sUSE\_TYP VARCHAR2(100);  sSQL\_Trunc VARCHAR2(1000);  SLACK\_MESSAGE VARCHAR2(1000);  SCALL\_SP\_SLACK\_SNED\_TEMP VARCHAR2(2000);  sErr\_Code VARCHAR2(20);  sErr\_Msg VARCHAR2(200);  OTHERS EXCEPTION;    BEGIN      dStart\_Date := SYSDATE;      DBMS\_APPLICATION\_INFO.SET\_MODULE('SP\_TRUNC\_TABLES Procedure', 'Execute Truncate Table by Developer');      DBMS\_APPLICATION\_INFO.READ\_MODULE(sModule\_NM,sAction\_NM);        SELECT UPPER(TRANSLATE(NAME,'+.0123456789',' ')) INTO sDB\_NM FROM V$DATABASE;      SELECT SYS\_CONTEXT ('USERENV', 'SESSION\_USER') INTO sSESSION\_USER FROM DUAL;      SELECT SYS\_CONTEXT ('USERENV', 'HOST') INTO sHOST FROM DUAL;      SELECT SYS\_CONTEXT ('USERENV', 'OS\_USER') INTO sOS\_USER FROM DUAL;      SELECT SYS\_CONTEXT ('USERENV', 'MODULE') INTO sMODULE FROM DUAL;            -- 1.허용된 테이블인지 확인          IF (vOWNER||'.'||vTB\_NM IN (          'GSDBA.GSDBA\_TEST1',          'STORE.GSDBA\_TEST2',          'I\_JUNGGH.GSDBA\_TEST3'          )) THEN                    -- 2.서버스용인지 개인계정인지 구분                  IF (sSESSION\_USER LIKE 'I#\_%' ESCAPE '#') THEN                      sUSE\_TYP := '개인';                          -- 슬랙 메세지 발송(개인계정 실행시에만)                          SLACK\_MESSAGE := '[DBA모니터링-'||sDB\_NM||']  테이블 Truncate 수행: '||vOWNER||'.'||vTB\_NM||CHR(10)||                                          ' Host명: '||sHOST||CHR(10)||                                          ' OS유저: '||sOS\_USER||CHR(10)||                                          ' 프로그램: '||sMODULE||CHR(10)||                                          ' 수행유저: '||sSESSION\_USER;                          sCALL\_SP\_SLACK\_SNED\_TEMP := q'[BEGIN DBMON.SP\_SLACK\_SNED\_TEMP@DL\_METADB\_DBMON(' ]'||SLACK\_MESSAGE||q'['); END;]';  --                        DBMS\_OUTPUT.PUT\_LINE(sCALL\_SP\_SLACK\_SNED\_TEMP);                          EXECUTE IMMEDIATE sCALL\_SP\_SLACK\_SNED\_TEMP;                  ELSE                      sUSE\_TYP := '서비스';                  END IF;          -- 사용이력저장(개인/서비스계정 모두 이력 저장)          INSERT INTO GSDBA.GSDBA\_TRUNC\_L(DB\_NM,USER\_TYP,EXEC\_USERNAME,EXEC\_DTM,OWNER,TABLE\_NAME,HOST,OS\_USER,MODULE,PGM\_PASS)          VALUES(sDB\_NM,sUSE\_TYP,sSESSION\_USER,dStart\_Date,vOWNER,vTB\_NM,sHOST,sOS\_USER,sMODULE,'성공');          COMMIT;            -- Truncate수행          sSQL\_Trunc := 'TRUNCATE TABLE '||vOWNER||'.'||vTB\_NM;          DBMS\_OUTPUT.PUT\_LINE('수행: '||sSQL\_Trunc);          EXECUTE IMMEDIATE sSQL\_Trunc;          DBMS\_OUTPUT.PUT\_LINE('Truncate Table 성공');            ELSE              DBMS\_OUTPUT.PUT\_LINE('에러: '||vTB\_NM||' Truncate 권한이 없는 테이블입니다. DBA에게 문의 주세요');              -- 사용이력저장(수행실패이력)              INSERT INTO GSDBA.GSDBA\_TRUNC\_L(DB\_NM,USER\_TYP,EXEC\_USERNAME,EXEC\_DTM,OWNER,TABLE\_NAME,HOST,OS\_USER,MODULE,PGM\_PASS,ERR\_CODE)              VALUES(sDB\_NM,sUSE\_TYP,sSESSION\_USER,dStart\_Date,vOWNER,vTB\_NM,sHOST,sOS\_USER,sMODULE,'실패','미 권한 테이블 수행시도');              COMMIT;          END IF;      dEnd\_Date := SYSTIMESTAMP;    -- 에러 로그 기록 후 예외 처리      EXCEPTION          WHEN OTHERS THEN              dEnd\_Date := SYSTIMESTAMP;              DBMS\_OUTPUT.PUT\_LINE('ERROR : '||SQLERRM||'');              sErr\_Code := SQLCODE;              sErr\_Msg := SUBSTR(SQLERRM, 1, 200);              -- 사용이력저장(수행실패이력)              INSERT INTO GSDBA.GSDBA\_TRUNC\_L(DB\_NM,USER\_TYP,EXEC\_USERNAME,EXEC\_DTM,OWNER,TABLE\_NAME,HOST,OS\_USER,MODULE,PGM\_PASS,ERR\_CODE)              VALUES(sDB\_NM,sUSE\_TYP,sSESSION\_USER,dStart\_Date,vOWNER,vTB\_NM,sHOST,sOS\_USER,sMODULE,'실패',sErr\_Msg);              COMMIT;   END; |

[좋아요](https://wiki.gsretail.com/pages/viewpage.action?pageId=22256024)처음으로 좋아하는 사람이 되볼까요?

* 레이블 없음
* [레이블 편집](https://wiki.gsretail.com/pages/viewpage.action?pageId=22256024)

1 개의 댓글

알 수 없는 사용자 (leech)

* + AI 센터 요청
  + 테이블 추가 : STORE\_OWN.DQ\_EVENT\_DIVER
  + 권한 추가 : GRANT EXECUTE ON GSDBA.SP\_TRUNC\_TABLES TO PLF\_APP;

[Skip to end of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=22261156#page-metadata-end)

* 알 수 없는 사용자 (jung.gh)님이 작성, [2018-02-14](https://wiki.gsretail.com/pages/diffpagesbyversion.action?pageId=22261156&selectedPageVersions=3&selectedPageVersions=4)에 최종 변경

[Go to start of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=22261156#page-metadata-start)

대량의 데이터 삭제 시 커밋주기 조절하여 삭제

**GSDBA.SP\_GS\_DBA\_DEL** 원본 접기

|  |
| --- |
| GRANT EXECUTE ON SYS.DBMS\_SYSTEM TO GSDBA; -- 실시간 로깅 시 필요  GRANT DELETE ANY TABLE  TO GSDBA; -- 직접 권한 필요    ALTER SESSION SET DB\_FILE\_MULTIBLOCK\_READ\_COUNT=256; --Fast Full Scan 시 멀티블록 수 상향      CREATE OR REPLACE PROCEDURE GSDBA.SP\_GS\_DBA\_DEL(V\_COMMIT\_CNT IN NUMBER DEFAULT 1000)  AUTHID CURRENT\_USER  IS  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   1.프로그램 명 : SP\_GS\_DBA\_DEL 대량 Row 삭제 프로시저   2.기 능 정 의 : 대상 테이블의 Commit 주기 설정하여 데이터 삭제   3.참 고 사 항 :   4.변 경 이 력 :    Version  작성자   소   속   일    자   요청자       내       용   --------  ------  --------  ----------  ------  ---------------------      1.0    이치호            2017.07.17          최초 생성      2.0    정관호            2018.02.13          실시간 작업이력, 진행상황, 남은 실행수/시간 표시   \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  -- 변수 선언  nTotal\_CNT NUMBER := 0; -- 전체 Row 수  nTotal\_Del\_CNT NUMBER := 0; -- 삭제 Row 수  nCommit\_CNT NUMBER := 0; -- Commit 된 Row 수  sMSG VARCHAR2(500) := ' ';--진행상황 메세지  t\_Start\_Date TIMESTAMP;  t\_End\_Date TIMESTAMP;  t\_Start\_Time NUMBER;  t\_End\_Time NUMBER;  nRemain\_Execs NUMBER := 0;  nExecution\_Time NUMBER := 0;  nRemain\_Sec NUMBER := 0;  sRemain\_Time\_SQL VARCHAR2(100);  sRemain\_Time VARCHAR2(100);  CURSOR c1 IS      SELECT /\*+ index\_ffs(A) parallel\_index(A IX\_GS\_CARD\_CERT\_L\_01) \*/ ROWID RID, COUNT(\*) OVER () CT      FROM GS\_CARD\_CERT\_L A      WHERE REG\_DTM < TO\_DATE('20170101 000000','YYYYMMDD HH24MISS');  BEGIN      /\*ORA-20000: ORU-10027: buffer overflow, limit of 20000 bytes\*/  --    DBMS\_OUTPUT.DISABLE;      DBMS\_OUTPUT.ENABLE(1000000);      DBMS\_APPLICATION\_INFO.SET\_MODULE('SP\_GS\_DBA\_DEL PROCEDURE', 'DELETE DATA BY DBA BATCH');      BEGIN      t\_Start\_Date := SYSTIMESTAMP;      t\_Start\_Time := DBMS\_UTILITY.get\_time;      FOR cur\_rec IN c1 LOOP              -- 삭제 수행              DELETE FROM STORE\_OWN.GS\_CARD\_CERT\_L A              WHERE ROWID = cur\_rec.RID;                nTotal\_CNT := cur\_rec.CT;              nTotal\_Del\_CNT := nTotal\_Del\_CNT + 1;              nCommit\_CNT := nCommit\_CNT + 1;                -- 중간 COMMIT 수행              IF MOD(nCommit\_CNT, V\_COMMIT\_CNT) = 0 THEN                  COMMIT;                  t\_End\_Time := DBMS\_UTILITY.get\_time;                  nRemain\_Execs := ((nTotal\_CNT/V\_COMMIT\_CNT)+1) - (nTotal\_Del\_CNT/V\_COMMIT\_CNT); -- 남은 실행 수                  nExecution\_Time := (t\_End\_Time-t\_Start\_Time)/100; -- 1사이클 수행시간                  nRemain\_Sec := nRemain\_Execs \* nExecution\_Time; -- 남은 시간                  sRemain\_Time\_SQL := 'SELECT TO\_CHAR(TO\_DATE(TO\_CHAR('||nRemain\_Sec||',''99999999''),''SSSSS''),''HH24:MI:SS'') FROM DUAL'; -- 초->시간 변경                  EXECUTE IMMEDIATE sRemain\_Time\_SQL INTO sRemain\_Time;                    sMSG := '->'||TO\_CHAR(nCommit\_CNT,'9,999,999')||' 건 삭제완료('||nExecution\_Time||'초)   진행상황:'||TO\_CHAR(nTotal\_Del\_CNT/nTotal\_CNT\*100,'99.00')||'%('||nTotal\_Del\_CNT||'/'||nTotal\_CNT||')   남은 실행수: '||nRemain\_Execs||'   남은 시간: '||sRemain\_Time;                  SYS.DBMS\_SYSTEM.ksdwrt(1, sMSG);                  DBMS\_OUTPUT.PUT\_LINE(sMSG);                    nCommit\_CNT := 0; -- Commit Count 초기화                  t\_Start\_Time := DBMS\_UTILITY.get\_time;              END IF;          END LOOP;        EXCEPTION          WHEN OTHERS THEN          sMSG := '[Error] '||TO\_CHAR(SYSDATE,'YYYY-MM-DD HH24:MI:SS')||' >>>'||SQLERRM;          sys.DBMS\_SYSTEM.ksdwrt(1, sMSG);          DBMS\_OUTPUT.PUT\_LINE(sMSG);      END;  COMMIT;      t\_End\_Date := SYSTIMESTAMP;      sMSG := ('----------------------------------------------------------------------------------------');      SYS.DBMS\_SYSTEM.ksdwrt(1, sMSG);      DBMS\_OUTPUT.PUT\_LINE(sMSG);      sMSG := ('# 시작시간: '||TO\_CHAR(t\_Start\_Date,'YYYY-MM-DD HH24:MI:SS')||' 종료시간: '||TO\_CHAR(t\_End\_Date,'YYYY-MM-DD HH24:MI:SS')||' 전체 삭제 건수: '||nTotal\_Del\_CNT||' 건' );      SYS.DBMS\_SYSTEM.ksdwrt(1, sMSG);      DBMS\_OUTPUT.PUT\_LINE(sMSG);      sMSG := ('# 잔여 건수: = '||TO\_CHAR(nTotal\_CNT)||' - '||TO\_CHAR(nTotal\_Del\_CNT)||' = '||TO\_CHAR(nTotal\_CNT-nTotal\_Del\_CNT)||' 건');      SYS.DBMS\_SYSTEM.ksdwrt(1, sMSG);      DBMS\_OUTPUT.PUT\_LINE(sMSG);  END; |

[좋아요](https://wiki.gsretail.com/pages/viewpage.action?pageId=22261156)처음으로 좋아하는 사람이 되볼까요?

* 레이블 없음
* [레이블 편집](https://wiki.gsretail.com/pages/viewpage.action?pageId=22261156)

1 개의 댓글

알 수 없는 사용자 (jung.gh)

loop문과 if문의 잦은 사용은 성능 저하가 발생하므로 자제

[Skip to end of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=30125691#page-metadata-end)

* 작성자 : [정관호 Kyle [X]](https://wiki.gsretail.com/display/~junggh) - [2020-07-11](https://wiki.gsretail.com/pages/viewpreviousversions.action?pageId=30125691)

[Go to start of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=30125691#page-metadata-start)

**RMAN 주요 명령어**

###########################  
# CROSSCHECK 명령어  
###########################  
컨트롤 파일을 사용하는 NoCatalog Mode에서도 자주 사용되는 명령이다.  
백업 파일이나 아카이브드 로그파일을 RMAN을 사용하지 않고 삭제하였을 경우인데 RMAN 세션에서 관련 메시지가 발생한다.  
이와 같은 경우 CROSSCHECK 명령어를 사용하여 RMAN catalog 혹은 control file의 리스트를 현 상태와 동일하게 업데이트하게 할 수 있다.  
​  
이 명령어는 실제 V$BACKUP\_SET, V$DATAFILE\_COPY, RC\_BACKUP\_SET, RC\_DATAFILE\_COPY 뷰들의 Status 컬럼에 업데이트를 실행한다.  
특히 주목할 부분은 V$BACKUP\_PIECE의 STATUS 컬럼 값 (X, D, A) 이다.  
- A(Available) : RMAN 이 판단하기에 해당 백업 파일이 존재한다는 의미  
- X(Expired) : RMAN 이 판단하기에 해당 백업 파일이 존재하지 않는 의미  
- D(Deleted) : 해당 파일이 삭제됨  
​※ 주의할 점은 CROSSCHECK 명령어는 백업셋 피스의 상태를 DELETE 에서 AVAILABLE 로 변경할 수 없음(DELETE로 설정된 것은 어떠한 값으로도 변경이 불가)  
​  
RMAN> CROSSCHECK BACKUP;  
RMAN> crosscheck backupset;  
RMAN> crosscheck backup of datafile 1;  
RMAN> crosscheck backup of tablespace users;  
RMAN> crosscheck backup of controlfile;  
RMAN> crosscheck backup of spfile;  
RMAN> crosscheck backup tag ='FULL\_BACKUP';  
RMAN> crosscheck backup completed after sysdate-2';  
RMAN> crosscheck backup completed between'sysdate 5'and'sysdate 2';  
RMAN> crosscheck backup device type sbt;  
RMAN> crosscheck backup device type disk;  
RMAN> CROSSCHECK ARCHIVELOG ALL;  
RMAN> crosscheck archivelog from sequence 12;  
RMAN> crosscheck archivelog until sequence 2432;  
RMAN> crosscheck archivelog from time "to\_date('11-10-2004','mm-dd-yyyy')";  
RMAN> crosscheck archivelog until time "to\_date( 11-10-2004', mm-dd-yyyy')";  
RMAN> crosscheck archivelog'c:\backup'ARC003212.002';  
RMAN> crosscheck archivelog like'ARC 0001.log';  
RMAN> crosscheck archivelog like'% ARC00143.002 ;  
RMAN> crosscheck copy of datafile 10;  
RMAN> crosscheck datafilecopy'c:\backup\user\_tbs\_copy01.dbf';

###########################  
# LIST 명령어  
###########################  
백업 완료된 RMAN Repository 정보를 참조할 때 사용하는 명령어다.  
-- 백업된 모든 데이터베이스 파일을 표시한다.  
RMAN> list backup of database;  
RMAN> list backup summary;  
RMAN> list backup by file;  
RMAN> list backupset;  
RMAN> LIST EXPIRED BACKUP;  
RMAN> LIST EXPIRED ARCHIVELOG ALL;  
RMAN> list incarnation;  
RMAN> list backup of tablespace users;  
RMAN> list backup of datafile 3;  
RMAN> LIST ARCHIVELOG ALL;  
RMAN> list backup of archivelog all;  
RMAN> list backup of controlfile;  
RMAN> list backup of spfile;  
RMAN> list copy;  
RMAN> list copy of database;

-- system 테이블스페이스의 데이터파일 복사본을 표시 한다.  
RMAN> list copy of tablespace system;  
RMAN> list copy of archivelog all;  
RMAN> list copy of archivelog from sequence 33;  
RMAN> list copy of controlfile;

###########################  
# REPORT 명령어  
###########################  
RMAN Repository 정보를 참조하여 백업이 필요한 데이터 객체에 대한 분석 정보를 표시해 주는 명령어다.  
RMAN>report schema; # 데이터베이스의 구조는  
RMAN>report need backup; # 백업이 필요한 파일은

-- incremental level 3 일 경우 백업이 필요한지 report  
RMAN> report need backup incremental = 3;  
RMAN> report need backup days = 3; # 3일이 초과되어 백업이 필요한 파일은?

-- 3번 이상 백업되어 새로이 백업이 필요한 파일을 표시  
RMAN> report need backup redundancy = 3;  
RMAN> report need backup recovery window of 2 days;  
RMAN> report obsolete  
RMAN> report unrecoverable

###########################  
# DELETE 명령어  
###########################  
DELETE 명령으로 RMAN 툴로 디스크나 테이프에 백업하였지만 쓸모없는 백업 (RMAN backups)이나 Archived Redo Logs 파일을 삭제한다.  
control 파일에 백업파일들의 상태를 D(Deleted)로 갱신한다.  
NOPROMPT 옵션을 지정하지 않으면 삭제할 때 확인한다.  
DELETE OBSOLETE 명령은 더 이상 필요 없는 백업이나 복사된 파일을 지울 때 사용된다.

-- 백업파일들의 보존정책을 넘어선 백업본들을 삭제한다.  
RMAN>DELETE OBSOLETE;

-- 파일의 상태가 X인 것들을 지운다.(OS에서 파일 삭제 후 CROSSCHECK)  
RMAN> DELETE EXPIRED BACKUP;  
RMAN> delete backupset;  
RMAN> delete backupset 55; # v$backup\_set의 recid 값  
RMAN> delete backup;  
RMAN> delete noprompt expired backup; # 과거 백업본중 만료된 백업본 삭제.  
RMAN> delete noprompt expired copy;  
RMAN> delete archivelog all;  
RMAN> delete noprompt archivelog all;  
RMAN> delete expired archivelog all;  
RMAN> DELETE NOPROMPT EXPIRED ARCHIVELOG ALL;  
RMAN> delete noprompt expired backup of tablespace user\_data completed before 'sysdate-14';  
RMAN> delete backup of database like '/tmp%';  
RMAN> delete archivelog all backed up 2 times to device type sbt;  
RMAN> delete backupset 101, 102, 103;  
RMAN> delete controlfilecopy '/tmp/cntrlfile.copy';  
RMAN> delete backup of spfile tablespace users device type sbt;

# [**서버 font생성 사이트**](https://wiki.gsretail.com/pages/viewpage.action?pageId=30117751)

[Skip to end of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=30117751#page-metadata-end)

* [정관호 Kyle [X]](https://wiki.gsretail.com/display/~junggh)님이 작성, [2020-06-18](https://wiki.gsretail.com/pages/diffpagesbyversion.action?pageId=30117751&selectedPageVersions=1&selectedPageVersions=2)에 최종 변경

[Go to start of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=30117751#page-metadata-start)

<http://patorjk.com/software/taag/#p=testall&f=Graffiti&t=Prod%20TongDB01>

로 접속하여 원하는 글자 타이핑 후 입맛대로 사용

관호가 선호하는 타입

Colossal

TongDB 운영은 아래걸로 선정

Font Name: ANSI Shadow

PLANMON\_BASE=/oracle/DBA/plan\_mon

today=`date +%y%m%d`

cd $PLANMON\_BASE

sqlplus gsdba/inomuski << EOF

set echo off

spool /oracle/DBA/plan\_mon/mon\_$today.bak

@daily\_plan\_check\_07.sql

spool off

EXIT

EOF

## 사전 테이블 생성

CREATE TABLE GSDBA.SMTC\_DAILY\_PLAN\_CHECK

(

    CHECK\_DATE  DATE,

    SQL\_ID      VARCHAR2(13),

    OLD\_PS      DATE,

    OLD\_PLAN    NUMBER,

    NEW\_PS      DATE,

    NEW\_PLAN    NUMBER(20,2),

    DIFF\_BUFF   NUMBER(20,2),

    DIFF\_CPU    NUMBER(20,2),

    DIFF\_ELAP   NUMBER(20,2),

    OLD\_ELAP    NUMBER(20,2),

    NEW\_ELAP    NUMBER(20,3),

    AVG\_EXEC    NUMBER(20,2),

    MODULE      VARCHAR2(64)

)

TABLESPACE TS\_ETC\_D

;

hssmtcdb01:(SMTCPRD11):/oracle/DBA/plan\_mon]cat daily\_plan\_check\_07.sql

col DBID new\_value DB\_ID

SELECT DBID

  FROM V$DATABASE ;

col MAX\_SNAP new\_value MAX\_S

col MIN\_SNAP new\_value MIN\_S

select MAX\_SNAP , MIN\_SNAP

  FROM (SELECT MAX(SNAP\_ID) AS MAX\_SNAP, MAX(END\_INTERVAL\_TIME), MIN(SNAP\_ID) AS MIN\_SNAP , MIN(END\_INTERVAL\_TIME)

          FROM (SELECT SNAP\_ID , END\_INTERVAL\_TIME

                 FROM DBA\_HIST\_SNAPSHOT

                 WHERE TO\_CHAR(END\_INTERVAL\_TIME , 'YYYYMMDD HH24:MI:SS') BETWEEN TO\_CHAR(SYSDATE-1, 'YYYYMMDD')||' 05:50:00' AND TO\_CHAR(SYSDATE, 'YYYYMMDD')||' 06:10:00') );

INSERT INTO SMTC\_DAILY\_PLAN\_CHECK

WITH DAILY\_PLAN\_CHECK AS (

SELECT to\_CHAR(sysdate , 'YYYYMMDD') AS CHECK\_DATE , SPPT.TS AS PARSE\_TIME, SEH.\* ,

       ROW\_NUMBER() OVER (PARTITION BY SPPT.SQL\_ID ORDER BY SPPT.TS ) AS RK ,

       ROW\_NUMBER() OVER (PARTITION BY SPPT.SQL\_ID ORDER BY SPPT.TS DESC ) AS D\_RK

  from (SELECT SQL\_ID , PLAN\_HASH\_VALUE , TS

          FROM (SELECT SP.SQL\_ID , SP.PLAN\_HASH\_VALUE , TO\_CHAR(TIMESTAMP , 'YYYYMMDD HH24:MI:SS') AS TS ,

                       ROW\_NUMBER() OVER (PARTITION BY DHSP.TIMESTAMP ORDER BY DHSP.TIMESTAMP ) AS RK

                  FROM (SELECT SQL\_ID , PLAN\_HASH\_VALUE

                          FROM (SELECT SQL\_ID , PLAN\_HASH\_VALUE , COUNT(\*) OVER (PARTITION BY SQL\_ID ORDER BY SQL\_ID ) AS RK

                                  FROM DBA\_HIST\_SQLSTAT STAT

                                 WHERE STAT.DBID = &DB\_ID

                                   AND STAT.SNAP\_ID BETWEEN &MIN\_S and &MAX\_S

                                   AND STAT.INSTANCE\_NUMBER in(1,2,3,4,5)

                                   and PLAN\_HASH\_VALUE <> '0'

                                   AND STAT.EXECUTIONS\_DELTA > = 1

                                 GROUP BY SQL\_ID ,

                                       PLAN\_HASH\_VALUE )

                         WHERE RK <> 1) SP ,

                       DBA\_HIST\_SQL\_PLAN DHSP

                 WHERE DHSP.DBID = &DB\_ID

                   AND DHSP.SQL\_ID = SP.SQL\_ID

                   AND DHSP.PLAN\_HASH\_VALUE = SP.PLAN\_HASH\_VALUE

--           AND TO\_CHAR(TIMESTAMP , 'YYYYMMDD HH24:MI:SS') BETWEEN to\_char(sysdate - 2 , 'YYYYMMDD')||' 05:50:00' and to\_char(sysdate-1 , 'YYYYMMDD')||' 06:10:00'

                       )

         WHERE RK = 1

               ) SPPT ,

       (select SQL\_ID ,

               PLAN\_HASH\_VALUE,

               round(AVG(EXECUTIONS\_DELTA), 2) AS AVG\_EXEC ,

               round(AVG(BUFFER\_GETS\_DELTA), 2) AS AVG\_BUFF ,

               round(AVG(BUFFER\_GETS\_DELTA/decode(EXECUTIONS\_DELTA, 0, 1, EXECUTIONS\_DELTA)), 2) AS AVG\_PER\_BUFF ,

               round(AVG(CPU\_TIME\_DELTA), 2) AS AVG\_CPU ,

               round(AVG(CPU\_TIME\_DELTA/decode(EXECUTIONS\_DELTA, 0, 1, EXECUTIONS\_DELTA)), 2) AS AVG\_PER\_CPU ,

               round(AVG(ELAPSED\_TIME\_DELTA), 2) AS AVG\_ELAPS ,

               round(AVG(ELAPSED\_TIME\_DELTA/decode(EXECUTIONS\_DELTA, 0, 1, EXECUTIONS\_DELTA)), 2) AS AVG\_PER\_ELAP

          FROM DBA\_HIST\_SQLSTAT STAT

         WHERE STAT.DBID = &DB\_ID

           AND STAT.SNAP\_ID BETWEEN &MIN\_S AND &MAX\_S

           AND STAT.INSTANCE\_NUMBER IN(1,2,3,4,5)

           AND STAT.EXECUTIONS\_DELTA > = 1

           AND STAT.CPU\_TIME\_DELTA > 0

           AND STAT.MODULE IS not null

         group by /\*STAT.INSTANCE\_NUMBER ,\*/STAT.SQL\_ID , STAT.PLAN\_HASH\_VALUE ) SEH

 WHERE SPPT.SQL\_ID = SEH.SQL\_ID

   AND SPPT.PLAN\_HASH\_VALUE = SEH.PLAN\_HASH\_VALUE

 order by SPPT.sql\_id , TS

 )

SELECT SYSDATE  AS CHECK\_DATE , INFO.\* ,

       S\_STAT.MODULE

  FROM (select \*

          from (select A.SQL\_ID AS SQL\_ID ,

                       TO\_DATE(A.PARSE\_TIME,'YYYYMMDD HH24:MI:SS') AS OLD\_PS ,

                       A.PLAN\_HASH\_VALUE AS OLD\_PLAN ,

                       TO\_DATE(B.PARSE\_TIME,'YYYYMMDD HH24:MI:SS') AS NEW\_PS,

                       B.PLAN\_HASH\_VALUE AS NEW\_PLAN ,

                       round(B.AVG\_PER\_BUFF/A.AVG\_PER\_BUFF , 2) AS DIFF\_BUFF ,

                       round(B.AVG\_PER\_CPU / a.AVG\_PER\_CPU , 2) AS DIFF\_CPU ,

                       round(B.AVG\_PER\_ELAP / a.AVG\_PER\_ELAP , 2) AS DIFF\_ELAP ,

                       TRUNC(a.AVG\_PER\_ELAP/1000000,3) AS OLD\_ELAP ,

                       TRUNC(B.AVG\_PER\_ELAP/1000000,3) AS NEW\_ELAP ,

                       TRUNC(B.AVG\_EXEC,3) AVG\_EXEC

                  from (select \*

                          from DAILY\_PLAN\_CHECK

                         where CHECK\_DATE = to\_char(sysdate , 'YYYYMMDD')

                           and RK = 1) A ,

                       (select \*

                          from DAILY\_PLAN\_CHECK

                         where CHECK\_DATE = to\_char(sysdate , 'YYYYMMDD')

                           and D\_RK = 1

                           AND PARSE\_TIME BETWEEN to\_char(sysdate - 1 , 'YYYYMMDD')||' 06:50:00' and to\_char(sysdate , 'YYYYMMDD')||' 07:10:00'

                               ) B

                 where a.sql\_id = B.sql\_id

                       ) A

         --where DIFF\_BUFF >= 1.2

         --   OR DIFF\_CPU >= 1.2

         --   OR DIFF\_ELAP >= 1.2

         ) INFO ,

       (select

               SQL\_ID ,               STAT.MODULE

          from dba\_hist\_snapshot SNAP ,

               dba\_hist\_sqlstat STAT

         where SNAP.DBID = &DB\_ID

           and STAT.dbid = &DB\_ID

           and SNAP.SNAP\_ID between &MIN\_S and &MAX\_S

           and STAT.snap\_ID between &MIN\_S and &MAX\_S

           and SNAP.SNAP\_ID = STAT.SNAP\_ID

           and SNAP.INSTANCE\_NUMBER in(1,2,3,4,5)

           and STAT.INSTANCE\_NUMBER in(1,2,3,4,5)

           and stat.sql\_id not in ('g245w9vsy6mkt')

           and STAT.INSTANCE\_NUMBER = SNAP.INSTANCE\_NUMBER

           and SNAP.SNAP\_ID = STAT.SNAP\_ID

         group by SQL\_ID,

               MODULE ) S\_STAT

 WHERE S\_STAT.SQL\_ID = INFO.SQL\_ID

 order by INFO.SQL\_ID ;

 COMMIT;

 hssmtcdb01:(SMTCPRD11):/oracle/DBA/plan\_mon]cat /oracle/DBA/RUN\_CRON/run\_Stat\_daily.sh

. /oracle/.profile

STATMON\_BASE=/oracle/DBA/RUN\_CRON

today=`date +%y%m%d`

cd $STATMON\_BASE

sqlplus gsdba/inomuski<< EOF

set echo off

set serveroutput on

spool /oracle/DBA/RUN\_CRON/log/day\_$today.bak

@run\_Stat\_daily.sql

spool off

EXIT

EOF

hssmtcdb01:(SMTCPRD11):/oracle/DBA/RUN\_CRON]cat run\_Stat\_daily.sql

DECLARE

V\_OWNER  VARCHAR2(100);

V\_TABLENAME VARCHAR2(100);

CURSOR c\_sql IS

WITH PART\_CHK AS(

SELECT /\*+ MATERIALIZE \*/ TABLE\_OWNER,TABLE\_NAME FROM DBA\_TAB\_PARTITIONS

WHERE TABLE\_OWNER  IN ('SMTC\_OWN','QA\_OWN','WMS\_OWN','WMS\_INF','PAY\_OWN')

 AND TABLE\_NAME NOT LIKE 'BIN$%'

 GROUP BY TABLE\_OWNER,TABLE\_NAME

 ) , STALE\_CHK AS(

 SELECT /\*+ MATERIALIZE \*/ OWNER,TABLE\_NAME  FROM DBA\_TAB\_STATISTICS

 WHERE OWNER IN ('SMTC\_OWN','QA\_OWN','WMS\_OWN','WMS\_INF','PAY\_OWN')

   AND STATTYPE\_LOCKED IS NOT NULL

 ) , SEG\_CHK AS(

 SELECT /\*+ MATERIALIZE \*/ OWNER,SEGMENT\_NAME  FROM DBA\_SEGMENTS

 WHERE OWNER IN ('SMTC\_OWN','QA\_OWN','WMS\_OWN','WMS\_INF','PAY\_OWN')

   AND SEGMENT\_TYPE = 'TABLE'

)

SELECT OWNER, TABLE\_NAME

FROM DBA\_TABLES

WHERE 1=1

AND OWNER  IN ('SMTC\_OWN','QA\_OWN','WMS\_OWN','WMS\_INF','PAY\_OWN')--('GSDBA','BATCH\_STORE','BATCH\_INSU')

AND ( NUM\_ROWS is null or NUM\_ROWS <= 10)

AND TABLE\_NAME NOT  LIKE 'Z/\_%' escape '/'

AND TABLE\_NAME NOT  LIKE 'MLOG$%' escape '/'

AND TABLE\_NAME NOT  LIKE 'MV/\_%' escape '/'

AND PARTITIONED ='NO'

AND TEMPORARY ='N'

AND  (OWNER,TABLE\_NAME) NOT IN(SELECT \* FROM PART\_CHK)

AND  (OWNER,TABLE\_NAME) NOT IN(SELECT \* FROM STALE\_CHK)

AND  (OWNER,TABLE\_NAME) IN(SELECT \* FROM SEG\_CHK)

AND TABLESPACE\_NAME NOT IN('TS\_MIG\_TEST1','TS\_MIG\_TEST2')

ORDER BY  1,2 DESC;

BEGIN

-- LAST trucnate

EXECUTE IMMEDIATE 'TRUNCATE TABLE GSDBA.SYS\_STATS\_BAC\_LAST' ;

--1dya back

INSERT INTO GSDBA.SYS\_STATS\_BAC\_LAST

SELECT \* FROM GSDBA.SYS\_STATS\_BACKUP;

COMMIT;

--1day TRUNCATE

EXECUTE IMMEDIATE 'TRUNCATE TABLE GSDBA.SYS\_STATS\_BACKUP' ;

 DBMS\_STATS.EXPORT\_SCHEMA\_STATS ('SMTC\_OWN','SYS\_STATS\_BACKUP','STAT\_TDAY', 'GSDBA');

 DBMS\_STATS.EXPORT\_SCHEMA\_STATS ('QA\_OWN','SYS\_STATS\_BACKUP','STAT\_TDAY', 'GSDBA');

 DBMS\_STATS.EXPORT\_SCHEMA\_STATS ('WMS\_OWN','SYS\_STATS\_BACKUP','STAT\_TDAY', 'GSDBA');

 DBMS\_STATS.EXPORT\_SCHEMA\_STATS ('WMS\_INF','SYS\_STATS\_BACKUP','STAT\_TDAY', 'GSDBA');

 DBMS\_STATS.EXPORT\_SCHEMA\_STATS ('PAY\_OWN','SYS\_STATS\_BACKUP','STAT\_TDAY', 'GSDBA');

 DBMS\_STATS.EXPORT\_SCHEMA\_STATS ('CST\_INF','SYS\_STATS\_BACKUP','STAT\_TDAY', 'GSDBA');

 FOR r\_sql IN c\_sql LOOP

   BEGIN

       GSDBA.PC\_GATHER\_STATS(r\_sql.OWNER,r\_sql.TABLE\_NAME) ;

   EXCEPTION

     WHEN OTHERS THEN   EXECUTE IMMEDIATE V\_TABLENAME||','''||SUBSTR(SQLERRM, 11, 200)||''')';

   END;

END LOOP;

END;

/

CREATE TABLE GSDBA.SYS\_STATS\_BAC\_LAST

(

    STATID  VARCHAR2(30),

    TYPE    CHAR(1),

    VERSION NUMBER,

    FLAGS   NUMBER,

    C1      VARCHAR2(30),

    C2      VARCHAR2(30),

    C3      VARCHAR2(30),

    C4      VARCHAR2(30),

    C5      VARCHAR2(30),

    N1      NUMBER,

    N2      NUMBER,

    N3      NUMBER,

    N4      NUMBER,

    N5      NUMBER,

    N6      NUMBER,

    N7      NUMBER,

    N8      NUMBER,

    N9      NUMBER,

    N10     NUMBER,

    N11     NUMBER,

    N12     NUMBER,

    D1      DATE,

    R1      RAW(32),

    R2      RAW(32),

    CH1     VARCHAR2(1000),

    CL1     CLOB

)

TABLESPACE TS\_ETC\_D

STORAGE

(

    INITIAL 10M

    NEXT 10M

)

NOCOMPRESS

LOB (CL1) STORE AS BASICFILE

(

    STORAGE

    (

        INITIAL 5M

        NEXT 5M

    )

    CHUNK 8192

    RETENTION

);

CREATE TABLE GSDBA.SYS\_STATS\_BACKUP

(

    STATID  VARCHAR2(30),

    TYPE    CHAR(1),

    VERSION NUMBER,

    FLAGS   NUMBER,

    C1      VARCHAR2(30),

    C2      VARCHAR2(30),

    C3      VARCHAR2(30),

    C4      VARCHAR2(30),

    C5      VARCHAR2(30),

    N1      NUMBER,

    N2      NUMBER,

    N3      NUMBER,

    N4      NUMBER,

    N5      NUMBER,

    N6      NUMBER,

    N7      NUMBER,

    N8      NUMBER,

    N9      NUMBER,

    N10     NUMBER,

    N11     NUMBER,

    N12     NUMBER,

    D1      DATE,

    R1      RAW(32),

    R2      RAW(32),

    CH1     VARCHAR2(1000),

    CL1     CLOB

)

TABLESPACE TS\_ETC\_D

STORAGE

(

    INITIAL 10M

    NEXT 10M

)

NOCOMPRESS

LOB (CL1) STORE AS BASICFILE

(

    STORAGE

    (

        INITIAL 5M

        NEXT 5M

    )

    CHUNK 8192

    RETENTION

);

CREATE INDEX GSDBA."BIN$pzm/jkCIAwDgU6XzzcJ+Pg==$0"

ON GSDBA.SYS\_STATS\_BACKUP

(

    STATID,

    TYPE,

    C5,

    C1,

    C2,

    C3,

    C4,

    VERSION

)

TABLESPACE TS\_ETC\_D

STORAGE

(

    INITIAL 10M

    NEXT 10M

);

-- DDL 구문은 Orange for ORACLE(으)로 생성되었습니다.

-- 'GSDBA@PSMTC1'세션(2021/11/20 05:57:44.000000) 사용

CREATE OR REPLACE PROCEDURE GSDBA.PC\_GATHER\_STATS (

  P\_TABLE\_OWNER      IN VARCHAR2,

  P\_TABLE\_NAME       IN VARCHAR2,

  P\_PARTITION\_NAME   IN VARCHAR2 DEFAULT NULL,

  P\_DEGREE           IN NUMBER DEFAULT NULL,

  P\_ESTIMATE\_PERCENT IN NUMBER DEFAULT NULL )

IS

  v\_start\_tm date ;

  v\_sql      varchar2(2000) ;

  v\_errno    number ;

  v\_err\_msg  varchar2(2000) ;

  V\_DEGREE           number := 8 ;

  V\_ESTIMATE\_PERCENT number := 1 ;

  V\_PARTITION\_TYPE   varchar2(30) ;

  V\_PARTITION\_NAME   varchar2(30) ;

  V\_SUBPARTITION\_NAME   varchar2(30) ;

  V\_IOT\_FLAG         number := 0 ;

  V\_CLUSTER\_FLAG     number := 0 ;

  l\_error\_pos        VARCHAR2(100);

begin

  l\_error\_pos := '10.----->>>  변수 초기화';

  v\_start\_tm := sysdate ;

  l\_error\_pos := '11.----->>>  IOT Check';

  select count(\*) into V\_IOT\_FLAG

  from dba\_tables

  where owner   = upper(P\_TABLE\_OWNER)

  and table\_name = upper(P\_TABLE\_NAME)

  and iot\_type = 'IOT';

  l\_error\_pos := '12.----->>>  Cluster Check';

  select count(\*) into V\_CLUSTER\_FLAG

  from dba\_tables

  where owner   = upper(P\_TABLE\_OWNER)

  and table\_name = upper(P\_TABLE\_NAME)

  and cluster\_name is not null;

  l\_error\_pos := '13.----->>> OPTIMIZER\_MODE 변경';

-- DBMS\_UTILITY.EXEC\_DDL\_STATEMENT('ALTER SESSION SET OPTIMIZER\_MODE=ALL\_ROWS');

--  DBMS\_UTILITY.EXEC\_DDL\_STATEMENT('ALTER SESSION SET "\_DB\_FAST\_OBJ\_TRUNCATE"=TRUE');

  l\_error\_pos := '14.----->>> 일반 테이블 및 파티션 테이블 검사';

  IF (V\_IOT\_FLAG = 0 AND V\_CLUSTER\_FLAG = 0) THEN  -- Normal Table inclue Partiton

  begin

    l\_error\_pos := '15.----->>> 일반 테이블 및 파티션 테이블 Sample 크기 및 병렬도 계산';

    select  case   when sum(bytes)/(1024\*1024) <= 100    then  100

                   when sum(bytes)/(1024\*1024) <= 500    then  100

                   when sum(bytes)/(1024\*1024) <= 1000   then  100

                   when sum(bytes)/(1024\*1024) <= 50000  then   30

                   else 30 end  est\_pct,

            case   when sum(bytes)/(1024\*1024) <= 100    then    4

                   when sum(bytes)/(1024\*1024) <= 500    then    4

                   when sum(bytes)/(1024\*1024) <= 1000   then    4

                   when sum(bytes)/(1024\*1024) <= 50000  then    8

                   else 8 end degree,

            max( segment\_type )

           into V\_ESTIMATE\_PERCENT, V\_DEGREE, V\_PARTITION\_TYPE

           from dba\_segments

          where owner        = upper(P\_TABLE\_OWNER)

            and segment\_name = upper(P\_TABLE\_NAME)

          group by owner, segment\_name

    ;

    l\_error\_pos := '16.----->>> 파티션 지정시 병렬도 설정';

    IF ( P\_PARTITION\_NAME is not NULL and V\_PARTITION\_TYPE like '%TABLE PARTITION%') THEN

       select   case   when sum(bytes)/(1024\*1024) <= 100    then    4

                       when sum(bytes)/(1024\*1024) <= 500    then    4

                       when sum(bytes)/(1024\*1024) <= 1000   then    4

                       when sum(bytes)/(1024\*1024) <= 50000  then    8

                       else 8 end degree

       into  V\_DEGREE

       from  dba\_tab\_partitions p, dba\_segments s

      where  p.table\_owner    = upper(P\_TABLE\_OWNER)

        and  p.table\_name     = upper(P\_TABLE\_NAME)

        and  p.partition\_name = upper(P\_PARTITION\_NAME)

        and  p.table\_owner    = s.owner

        and  p.table\_name     = s.segment\_name

        and  p.partition\_name = s.partition\_name

      group by s.owner, s.segment\_name, s.partition\_name

      ;

    END IF ;

    l\_error\_pos := '17.----->>> Sub파티션 확인';

    IF ( P\_PARTITION\_NAME is not NULL and V\_PARTITION\_TYPE like '%SUBPARTITION%' ) THEN

       BEGIN

         select   case   when sum(bytes)/(1024\*1024) <= 100    then    4

                         when sum(bytes)/(1024\*1024) <= 500    then    4

                         when sum(bytes)/(1024\*1024) <= 1000   then    4

                         when sum(bytes)/(1024\*1024) <= 50000  then    8

                         else 8 end degree

         into  V\_DEGREE

         from  dba\_tab\_subpartitions sp, dba\_segments s

        where  sp.table\_owner       = upper(P\_TABLE\_OWNER)

          and  sp.table\_name        = upper(P\_TABLE\_NAME)

          and  sp.partition\_name    = upper(P\_PARTITION\_NAME)

          and  sp.table\_owner       = s.owner

          and  sp.table\_name        = s.segment\_name

          and  sp.subpartition\_name = s.partition\_name

        group by sp.table\_owner, sp.table\_name, sp.partition\_name

        ;

       l\_error\_pos := '18.----->>> Execption 처리: Sub파티션이 없는 경우 파티션 이름 설정';

       V\_PARTITION\_NAME := P\_PARTITION\_NAME ;

       EXCEPTION WHEN NO\_DATA\_FOUND THEN

          V\_PARTITION\_NAME := P\_PARTITION\_NAME ;

       END ;

    ELSE

       l\_error\_pos := '19.----->>> Sub파티션이 없는 경우 파티션 이름 설정';

       V\_PARTITION\_NAME := P\_PARTITION\_NAME ;

    END IF ;

    l\_error\_pos := '20.----->>> 병렬도 수동 설정';

    IF ( P\_DEGREE is not NULL ) THEN

       V\_DEGREE := P\_DEGREE ;

    END IF ;

    l\_error\_pos := '21.----->>> Sampleing Percent 수동 설정';

    IF ( P\_ESTIMATE\_PERCENT is not NULL ) THEN

       V\_ESTIMATE\_PERCENT := P\_ESTIMATE\_PERCENT ;

    END IF ;

    l\_error\_pos := '22.----->>> 파티션 미설정시 통계정보 수행 Check';

    IF ( P\_PARTITION\_NAME is NULL ) THEN

      l\_error\_pos := '23.----->>> 피티션 미설정 Logging을 위한 SQL 저장';

      v\_sql := 'DBMS\_STATS.GATHER\_TABLE\_STATS ( ownname => ''' || P\_TABLE\_OWNER || ''', tabname => '''||p\_table\_name ||

                 ''', estimate\_percent=> ' || V\_ESTIMATE\_PERCENT ||

                 ', granularity=>''ALL'', degree=> ' || V\_DEGREE || ' , cascade=>true ) ' ;

      l\_error\_pos := '23.----->>> 파티션 미설정 통계정보 수행';

      DBMS\_STATS.GATHER\_TABLE\_STATS (

         ownname => P\_TABLE\_OWNER

        ,tabname => P\_TABLE\_NAME

        ,estimate\_percent=> V\_ESTIMATE\_PERCENT

        ,granularity=>'ALL'

        ,degree=> V\_DEGREE

        ,cascade=>true );

    ELSE

      l\_error\_pos := '24.----->>> 파티션 설정 Logging을 위한 SQL 저장';

      v\_sql := 'DBMS\_STATS.GATHER\_TABLE\_STATS ( ownname => ''' || P\_TABLE\_OWNER || ''', tabname => '''||p\_table\_name ||

                ''', partname => '''||v\_partition\_name || ''', estimate\_percent=> ' || V\_ESTIMATE\_PERCENT ||

                ', granularity=>''PARTITION'', degree=> ' || V\_DEGREE || ' , cascade=>true ) ' ;

      l\_error\_pos := '25.----->>> 파티션 설정 통계정보 수행';

      DBMS\_STATS.GATHER\_TABLE\_STATS (

         ownname => P\_TABLE\_OWNER

        ,tabname => P\_TABLE\_NAME

        ,partname => V\_PARTITION\_NAME

        ,estimate\_percent=> V\_ESTIMATE\_PERCENT

        ,granularity=>'PARTITION'

        ,degree=> V\_DEGREE

        ,cascade=>true );

      IF ( V\_PARTITION\_TYPE like '%SUBPARTITION%') THEN

          l\_error\_pos := '26.----->>> Sub파티션 설정 Logging을 위한 SQL 저장';

          v\_sql := 'DBMS\_STATS.GATHER\_TABLE\_STATS ( ownname => ''' || P\_TABLE\_OWNER || ''', tabname => '''||p\_table\_name ||

                    ''', partname => '''||v\_partition\_name || ''', estimate\_percent=> ' || V\_ESTIMATE\_PERCENT ||

                    ', granularity=>''SUBPARTITION'', degree=> ' || V\_DEGREE || ' , cascade=>true ) ' ;

          l\_error\_pos := '27.----->>> Sub파티션 설정 통계정보 수행';

          DBMS\_STATS.GATHER\_TABLE\_STATS (

             ownname => P\_TABLE\_OWNER

            ,tabname => P\_TABLE\_NAME

            ,partname => V\_PARTITION\_NAME

            ,estimate\_percent=> V\_ESTIMATE\_PERCENT

            ,granularity=>'SUBPARTITION'

            ,degree=> V\_DEGREE

            ,cascade=>true );

      END IF;

    END IF;

  END;

  END IF;

  l\_error\_pos := '30.----->>> IOT 테이블 검사';

  IF (V\_IOT\_FLAG >= 1) THEN  -- IOT Table

  BEGIN

    l\_error\_pos := '31.----->>> IOT 테이블 Sample 크기 및 병렬도 계산';

    select max(est\_pct) est\_pct, max(degree) degree

    into V\_ESTIMATE\_PERCENT, V\_DEGREE

    from ( select  case   when sum(bytes)/(1024\*1024) <= 100    then  100

                          when sum(bytes)/(1024\*1024) <= 500    then  100

                          when sum(bytes)/(1024\*1024) <= 1000   then  100

                          when sum(bytes)/(1024\*1024) <= 50000  then   30

                          else 30 end  est\_pct,

                   case   when sum(bytes)/(1024\*1024) <= 100    then    4

                          when sum(bytes)/(1024\*1024) <= 500    then    4

                          when sum(bytes)/(1024\*1024) <= 1000   then    4

                          when sum(bytes)/(1024\*1024) <= 50000  then    8

                          else 8 end degree

           from dba\_tables T, dba\_indexes I, dba\_segments S

           where T.owner        = upper(P\_TABLE\_OWNER)

           and T.table\_name     = upper(P\_TABLE\_NAME)

           and T.owner          = I.TABLE\_OWNER

           and T.TABLE\_NAME     = I.TABLE\_NAME

           and I.INDEX\_NAME     = S.SEGMENT\_NAME

           and S.SEGMENT\_TYPE   like 'INDEX%'

           and I.owner          = S.OWNER

           group by S.owner, S.segment\_name

         ) iot\_all\_segments

    ;

    l\_error\_pos := '32.----->>> IOT 테이블 Logging을 위한 SQL 저장';

    v\_sql := 'DBMS\_STATS.GATHER\_TABLE\_STATS ( ownname => ''' || P\_TABLE\_OWNER || ''', tabname => '''||p\_table\_name ||

                 ''', estimate\_percent=> ' || V\_ESTIMATE\_PERCENT ||

                 ', granularity=>''ALL'', degree=> ' || V\_DEGREE || ' , cascade=>true ) ' ;

    l\_error\_pos := '33.----->>> IOT 테이블 통계정보 수행';

    DBMS\_STATS.GATHER\_TABLE\_STATS (

         ownname => P\_TABLE\_OWNER

        ,tabname => P\_TABLE\_NAME

        ,estimate\_percent=> V\_ESTIMATE\_PERCENT

        ,granularity=>'ALL'

        ,degree=> V\_DEGREE

        ,cascade=>true );

  END;

  END IF;

  l\_error\_pos := '34.----->>> Cluster 테이블 검사';

  IF (V\_CLUSTER\_FLAG >= 1) THEN  -- Cluster Table

  BEGIN

  l\_error\_pos := '35.----->>> Cluster 테이블 Sample 크기 및 병렬도 계산';

    select  case   when sum(bytes)/(1024\*1024) <= 100    then  100

                   when sum(bytes)/(1024\*1024) <= 500    then  100

                   when sum(bytes)/(1024\*1024) <= 1000   then  100

                   when sum(bytes)/(1024\*1024) <= 50000  then   30

                   else 30 end  est\_pct,

            case   when sum(bytes)/(1024\*1024) <= 100    then    4

                   when sum(bytes)/(1024\*1024) <= 500    then    4

                   when sum(bytes)/(1024\*1024) <= 1000   then    4

                   when sum(bytes)/(1024\*1024) <= 50000  then    8

                   else 8 end degree

           into V\_ESTIMATE\_PERCENT, V\_DEGREE

           from dba\_tables T, dba\_segments S

           where T.owner  =  upper(P\_TABLE\_OWNER)

           and T.table\_name = upper(P\_TABLE\_NAME)

           and T.cluster\_name= S.SEGMENT\_NAME

           and T.owner = S.OWNER

           group by S.owner, S.segment\_name

           ;

    l\_error\_pos := '36.----->>> Cluster 테이블 Logging을 위한 SQL 저장';

    v\_sql := 'DBMS\_STATS.GATHER\_TABLE\_STATS ( ownname => ''' || P\_TABLE\_OWNER || ''', tabname => '''||p\_table\_name ||

                 ''', estimate\_percent=> ' || V\_ESTIMATE\_PERCENT ||

                 ', granularity=>''ALL'', degree=> ' || V\_DEGREE || ' , cascade=>true ) ' ;

    l\_error\_pos := '37.----->>> Cluster 테이블 통계정보 수행';

    DBMS\_STATS.GATHER\_TABLE\_STATS (

         ownname => P\_TABLE\_OWNER

        ,tabname => P\_TABLE\_NAME

        ,estimate\_percent=> V\_ESTIMATE\_PERCENT

        ,granularity=>'ALL'

        ,degree=> V\_DEGREE

        ,cascade=>true );

  END;

  END IF;

  v\_errno   := SQLCODE ;

  v\_err\_msg := SQLERRM ;

  insert into GSDBA.APP\_DDL\_LOG (PROC\_NAME, USER\_NAME, RUN\_DATE, TABLE\_OWNER, TABLE\_NAME, RUNING\_TIME, ERR\_NO, ERR\_MSG, SQL\_TEXT )

          values ( 'PC\_GATHER\_STATS', USER, SYSDATE, P\_TABLE\_OWNER, P\_TABLE\_NAME, (SYSDATE - v\_start\_tm)\*3600\*24, v\_errno, v\_err\_msg, l\_error\_pos||v\_sql ) ;

  commit ;

EXCEPTION

   when others then

       v\_errno   := SQLCODE ;

       v\_err\_msg := SQLERRM ;

       PC\_DDL\_LOG( 'PC\_GATHER\_STATS', USER, P\_TABLE\_OWNER, P\_TABLE\_NAME, (SYSDATE - v\_start\_tm)\*3600\*24, v\_errno, v\_err\_msg, l\_error\_pos||v\_sql ) ;

       raise\_application\_error(-20005, P\_TABLE\_NAME ||'에 대한 통계정보 생성이 실패하였습니다.' || chr(10)

                   || '           원인은 아래와 같습니다.'|| chr(10)|| v\_err\_msg  );

END PC\_GATHER\_STATS;

/

GRANT EXECUTE ON GSDBA.PC\_GATHER\_STATS TO SYS;

hssmtcdb01:(SMTCPRD11):/oracle/DBA/RUN\_CRON]cat /oracle/DBA/RUN\_CRON/run\_Stat\_week.sh

. /oracle/.profile

STATMON\_BASE=/oracle/DBA/RUN\_CRON

today=`date +%y%m%d`

cd $STATMON\_BASE

sqlplus gsdba/inomuski<< EOF

set echo off

set serveroutput on

spool /oracle/DBA/RUN\_CRON/log/week\_$today.bak

@run\_Stat\_week.sql

spool off

EXIT

EOF

hssmtcdb01:(SMTCPRD11):/oracle/DBA/RUN\_CRON]cat run\_Stat\_week.sql

DECLARE

V\_OWNER  VARCHAR2(100);

V\_TABLENAME VARCHAR2(100);

CURSOR c\_sql IS

WITH  PART\_CHK AS(

SELECT /\*+ MATERIALIZE \*/ TABLE\_OWNER,TABLE\_NAME FROM DBA\_TAB\_PARTITIONS

WHERE TABLE\_OWNER  IN ('SMTC\_OWN','QA\_OWN','WMS\_OWN','WMS\_INF','PAY\_OWN')

 AND TABLE\_NAME NOT LIKE 'BIN$%'

 GROUP BY TABLE\_OWNER,TABLE\_NAME

 ) ,TAB AS(

SELECT OWNER,TABLE\_NAME,NUM\_ROWS FROM DBA\_TAB\_STATISTICS

WHERE OWNER IN ('SMTC\_OWN','QA\_OWN','WMS\_OWN','WMS\_INF','PAY\_OWN')

AND PARTITION\_NAME IS NULL

AND NUM\_ROWS between 11 and 50000

AND STATTYPE\_LOCKED IS NULL

AND TABLE\_NAME NOT  LIKE 'Z/\_%' escape '/'

AND TABLE\_NAME NOT  LIKE 'MLOG$%' escape '/'

AND TABLE\_NAME NOT  LIKE 'MV/\_%' escape '/'

AND (OWNER,TABLE\_NAME) NOT IN(SELECT \* FROM PART\_CHK)

AND TABLE\_NAME NOT IN('DTR\_ORD\_SPEED\_IMPROV\_T') ---- 2017.08.14 plan change

)

SELECT OWNER,TABLE\_NAME  FROM(

SELECT TAB.OWNER,TAB.TABLE\_NAME, ((INSERTS-DELETES) /NUM\_ROWS)\* 100 PER

FROM TAB TAB , DBA\_TAB\_MODIFICATIONS DTM

WHERE TAB.OWNER =  DTM.TABLE\_OWNER

 AND  TAB.TABLE\_NAME =  DTM.TABLE\_NAME

 AND DTM.PARTITION\_NAME IS NULL

)

WHERE PER >= 15;

BEGIN

 FOR r\_sql IN c\_sql LOOP

   BEGIN

       GSDBA.PC\_GATHER\_STATS(r\_sql.OWNER,r\_sql.TABLE\_NAME) ;

   EXCEPTION

     WHEN OTHERS THEN   EXECUTE IMMEDIATE V\_TABLENAME||','''||SUBSTR(SQLERRM, 11, 200)||''')';

   END;

END LOOP;

END;

/

[Skip to end of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=38571922#page-metadata-end)

* 알 수 없는 사용자 (22980)님이 작성, [2020-10-05](https://wiki.gsretail.com/pages/diffpagesbyversion.action?pageId=38571922&selectedPageVersions=2&selectedPageVersions=3)에 최종 변경

[Go to start of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=38571922#page-metadata-start)

* 1[설치](https://wiki.gsretail.com/pages/viewpage.action?pageId=38571922#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4%EC%84%B1%EB%8A%A5%ED%85%8C%EC%8A%A4%ED%8A%B8%ED%88%B4Locust-%EC%84%A4%EC%B9%98)
* 2[호출 할 Web Page 생성 - Django](https://wiki.gsretail.com/pages/viewpage.action?pageId=38571922#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4%EC%84%B1%EB%8A%A5%ED%85%8C%EC%8A%A4%ED%8A%B8%ED%88%B4Locust-%ED%98%B8%EC%B6%9C%ED%95%A0WebPage%EC%83%9D%EC%84%B1-Django)
* 3[Locust 부하 Script](https://wiki.gsretail.com/pages/viewpage.action?pageId=38571922#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4%EC%84%B1%EB%8A%A5%ED%85%8C%EC%8A%A4%ED%8A%B8%ED%88%B4Locust-Locust%EB%B6%80%ED%95%98Script)
* 4[부하 테스트 UI](https://wiki.gsretail.com/pages/viewpage.action?pageId=38571922#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4%EC%84%B1%EB%8A%A5%ED%85%8C%EC%8A%A4%ED%8A%B8%ED%88%B4Locust-%EB%B6%80%ED%95%98%ED%85%8C%EC%8A%A4%ED%8A%B8UI)
* 5[성능 결과](https://wiki.gsretail.com/pages/viewpage.action?pageId=38571922#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4%EC%84%B1%EB%8A%A5%ED%85%8C%EC%8A%A4%ED%8A%B8%ED%88%B4Locust-%EC%84%B1%EB%8A%A5%EA%B2%B0%EA%B3%BC)

참고 : <https://docs.locust.io/en/stable/index.html>

# **설치**

|  |
| --- |
| pip3 install locust |

# **호출 할 Web Page 생성 - Django**

* WEBDB TOP Qeury 중 하나를 테스트 대상으로 선정
* API 호출 주소 [http://10.52.124.77:8081/database/ApiTest/](http://smartdba.gshs.co.kr:90/database/ApiTest/)

 원본 펼치기

# **Locust 부하 Script**

|  |
| --- |
| import time  from locust import HttpUser, task, between, User, TaskSet, events  import random  import cx\_Oracle as oci      class QuickstartUser(HttpUser):        list\_prd\_cd = []      wait\_time = between(0, 1)      @task      def view\_item(self):          # print(len(self.list\_prd\_cd))          if len(self.list\_prd\_cd) < 1 :              self.interrupt()          self.client.get("/?fname=%i" % self.list\_prd\_cd.pop(random.randrange(len(self.list\_prd\_cd)))[0], name="/?fname=[id]")      def on\_start(self):            conn = oci.connect('gsdba/inomuski@10.52.221.44:1530/DEVEC')          cursor = conn.cursor()          sql = """SELECT PRD\_CD FROM  PRD\_PRD\_M WHERE MOD\_DTM > SYSDATE - 15"""          cursor.execute(sql)          self.list\_prd\_cd = cursor.fetchall()          cursor.close() #          conn.close |

# **부하 테스트 UI**

* 최대 부하 : 1000 user
* 증가율 : 초당 100 user 씩 증가
* 대상 DB : 개발WEBDB
* 테스트 SQL : WEBDB TOP SQL

# **성능 결과**

개발 WEBDB 대상으로 1000 user 부하 테스트 결과 아래 결과를 확인할 수 있음 ( 운영WEBDB 에서는 평균 응답시간 0.01초 )

* Requests per Second(TPS) : 61.3
* Response Time(평균 응답시간) : 9883ms → 9.8초 ( 최대 33초 )

**<요약>**

**<시간 별 모니터링>**

[Skip to end of metadata](https://wiki.gsretail.com/display/DB/DBSAFER+Needs#page-metadata-end)

* 작성자 : 알 수 없는 사용자 (jung.gh), 최근 변경 : [백승엽/클라우드팀](https://wiki.gsretail.com/display/~sybaek) - [2022-10-19](https://wiki.gsretail.com/pages/diffpagesbyversion.action?pageId=22269849&selectedPageVersions=5&selectedPageVersions=6)

[Go to start of metadata](https://wiki.gsretail.com/display/DB/DBSAFER+Needs#page-metadata-start)

**DBSAFER Enterprise 주요기능**

**참고자료**

<http://www.pnpsecure.com/PRODUCTS/3-dbsafer-juyogineung/3-1-jeobgeunjeeo-mich-gwonhanjeeo>

* **접근통제** DBSAFER는 가상계정, 사용자 IP(IP 대역 포함), Application(TOAD / Orange / Golden / SQL-PLUS / 기타), DB 계정, 날짜 / 시간, PC Host Name, Login User 등 다양한 정보를 AND 조건으로 접근통제 정책을 설정하며, Telnet, FTP, SSH 서비스에 대한 접근통제 기능도 제공합니다.   
  사용자 IP / MAC / Host name 별 제어   
  - 차단하고자 하는 사용자의 IP를 Single 또는 Range로 지정하여 차단 및 허용  
  DBMS계정(Login User)차단   
  - DBMS에 접속하는 계정에 따라 접속을 차단 및 허용  
  사용자 IP / MAC / Host name 별 제어   
  - DBMS에 접속하는 Application에 따라 접속을 차단 및 허용  
  일정 기간 및 시간을 지정하여 허용 및 차단 기능 수행

**=> 개발자 SBC에서 DB계정(AP)을 통한 접속은 모두 불가,  개인DB계정(I\_XXX,DML\_XXX)은 개발자 SBC IP를 통해서만 접속 가능  
=> DB서버 용도별 정책 그룹핑(운영-개인정보보유DB / 운영-개인정보미보유DB / 검증DB / 개발DB**

* **권한통제** DBSAFER는 SQL 분석을 통해 DB Object(Table, Columu, DB link, Schema, Object, Alias) 등에 대한 접근통제 기능을 제공하며, 명령(CREATE, DROP, UPDATE)을 차단할 경우에도 접속된 세션을 유지합니다. Return Data에 대한 제한 설정을 통해 특정 사이즈 이상의 결과값은 파일로 저장하여 모든 로그를 조회할 수 있습니다.  
  - 특정 명령어와 Table 및 Columu 명까지 조합하여 명령어에 대해 사전 차단  
  - DCL, DML, DDL 등에 대한 명령을 차단할 경우에도 접속된 세션을 유지하며 SQL 명령어, DB Table 등의 모든 조건을 이용하여 권한을 통제  
  - 지정된 시간을 초과해도 DB 서버가 사용자에게 요청 값을 확보하지 못할 경우 해당 사용자 세션 강제 Kill 기능, 사용하지 않는 세션 자동 종료 기능, 기존 연결 세션에 대한 실시간 접근통제 적용

=> **개인DB계정(I\_XXX,DML\_XXX)로 DDL불가(수행 시도 시 SMS발송), 디비세이퍼 Timeout 정책 통일(현재 중구남방)**

* **모니터링 및 이력관리\_01** DBSAFER는 사용자별, 세션별 접속 이력을 실시간으로 모니터링하며 DBMS, SSH, Telnet, FTP 등을 통합으로 모니터링하고 로깅합니다. 대용량의 로그 조회 시 빠른 결과값을 전달합니다.  
  - 사용자 접속현황, 세션별 접속이력 및 쿼리에 대한 실시간 모니터링 (단, 보안관리자의 접근내역은 별도 관리)  
  - 통합 Dashboard와 통합 관리화면을 통하여 감시대상 서버, DB에 대한 세션, SQL 변화 추이 및 상태 정보를 그래프로 제공  
  - 로그에서 특정 로그를 검색하는 기능  
  - 모든 DB 접속 SQL 및 BIND 값 로깅 및 결과값(Option) 로깅

**=>** **개인DB계정(I\_XXX,DML\_XXX)에 대한 DB접속 및 수행 SQL(Bind포함) 이력 기록 시 DB서버의 부하정도는?  
=> SSH/TELNET/FTP에 대한 정책 추가 필요( 현재 WEBDB/ODS 운영DB서버에 개발자가 OS로 직접접속하여 Crontab/Script 변경 작업을 하며, 별도의 감사기록이 없음)**

* **모니터링 및 이력관리\_02** DBSAFER는 Telnet / SSH / FTP를 통해 DB 서버에 접근하여 실행한 명령어 및 모든 결과값을 로그에 저장합니다. DDL, CREATE, TABLE 별과 같은 다양한 조건에 의하여 DB 접근행위를 분석 및 감사하는 기능을 제공합니다. 정책에 따라 선별적으로 로깅하여 실패 또는 오류에 대한 메시지도 로깅합니다.   
  - 로그 검색기능을 통해 다양한 검색조건에 맞는 특정 데이터베이스 Object에 대해 언제, 누가 어떤 작업을 했는지 추적할 수 있습니다.
* **데이터마스킹 기능** DBSAFER는 어플리케이션에 관계없이 중요정보에 대해 데이터마스킹 처리하여 정보유출을 사전에 방지합니다. 특정 패턴(주민번호)의 경우 Masking의 오류를 최소화하기 위해 "13번째" Checksum을 인식하여 Checksum이 맞는 경우에만 데이터마스킹 기능이 동작하고, 지정한 패턴이 모두 마스킹이 되는 것이 아니라 패턴 중에서도 정책에서 지정한 일부분만 마스킹(예:"701125-\*\*\*\*\*\*\*")처리됩니다.

**=> 마스킹 대상에 대한 그룹핑과 마스킹 레벨이 다르게 설정이 가능? EX> 레벨1: 고객정보그룹, 레벨2: 협력사정보그룹, 레벨3: 임직원정보그룹**

**=> 마스킹 정책에 대해 엑셀 포맷으로 업로드/다운로드 관리 가능? (약 13개 DB에 6000개의 컬럼이 대상)**

* **결재기능** DBSAFER의 결재시스템은 사번 / User ID / User IP 및 테이블 단위 등 여러 가지 조건을 조합하여 쿼리에 대한 결재 기능을 제공하며 관련 이력보고서를 제공합니다.

**=> N/A**

* **리포트**    
   DBSAFER는 IP / 계정 / 데이터 / APP / Bind / Error Code 등 다양한 검색조건에 의한 리포팅 기능을 제공하고, 생성된 리포트는 외부 데이터(DOC, HWP, PDF, HTML, XML, XLS, TXT등...)로 Export하는 기능을 제공합니다. Report Viewer 포로그램을 통하여 그래픽 기반의 리포트를 제공하며, 다양한 기본 리포트 폼을 제공합니다.

**=> N/A**