변수 타입

Var table\_name.column\_name%TYPE; 해당 테이블의컬럼 타입대로 선언됨

Var table\_name%ROWTYPE; 해당테이블의 로루타입으로선언됨

Var.r 이러면 해당테이블의 r을 ..

선언부 DECALRE section

DECALARE

V\_id NUMBER(7);

C\_reginon\_id CONSTATNT NUMBER(8):=5;

…

실행부 Executeable section

BEGIN

….

Select \* from..

END;

완전체.

Decalar

Boolean ar;

Begin

ar:=true;

end;

디버그 켜기

DBMS\_OUTPUT.ENABLE; --여는 것

DBMS\_OUTPUT.ENABLE(9999999); --여는 것 버퍼많이준 것 보통이렇게 ㄴ

DBMS\_OUTPUT.DISABLE; --끄는 것

디버그 찍기

DBMS\_OUTPUT.PUT\_LINE('code\_name:2');

DBMS\_OUTPUT.PUT\_LINE('code\_name:32');

흐름제어.

이프문

IF s\_ild\_cd IS NULL THEN

…

End if

If A=’g’ THEN

…

ELSIF A=’B’ THEN

…

ELSE

…

END IF;

반복문

LOOP

…

END LOOP;

WHILE I<50 LOOP

I:=i+1;

END LOOP;

반복문 빠져나오는문EXIT

LOOP

Exit when a=’B’;

END LOOP;

1부터 MAX\_LOOPS 변수까지 증가 I변수는 카운트변수

FOR I in 1..max\_loops loop

End loop;

스토어드 프로시저, 펑션, 패키지

펑션

CREATE OR REPLACE z Fn\_getPoMaxDate(v\_ild\_cd in varchar2,

v\_plt\_cd in varchar2,

v\_year in varchar2)

RETURN varchar2 IS

r\_output VARCHAR2(10);

BEGIN

SELECT chr(32) || chr(40) || substr(AA.YMD,5,2) || '-' || substr(AA.YMD,7,2) || chr(41)

INTO r\_output

FROM (select YMD

FROM (SELECT TO\_CHAR(YMD, 'YYYYMMDD') YMD,

GREATEST(SUM(NVL(G01\_HR, 0)),

SUM(NVL(G02\_HR, 0)),

SUM(NVL(G03\_HR, 0)),

SUM(NVL(G04\_HR, 0)),

SUM(NVL(G05\_HR, 0)),

SUM(NVL(G06\_HR, 0)),

SUM(NVL(G07\_HR, 0)),

SUM(NVL(G08\_HR, 0)),

SUM(NVL(G09\_HR, 0)),

SUM(NVL(G010\_HR, 0)),

SUM(NVL(G011\_HR, 0)),

SUM(NVL(G012\_HR, 0)),

SUM(NVL(G013\_HR, 0)),

SUM(NVL(G014\_HR, 0)),

SUM(NVL(G015\_HR, 0)),

SUM(NVL(G016\_HR, 0)),

SUM(NVL(G017\_HR, 0)),

SUM(NVL(G018\_HR, 0)),

SUM(NVL(G019\_HR, 0)),

SUM(NVL(G020\_HR, 0)),

SUM(NVL(G021\_HR, 0)),

SUM(NVL(G022\_HR, 0)),

SUM(NVL(G023\_HR, 0)),

SUM(NVL(G024\_HR, 0))) GRE

FROM YTGENT05 t

WHERE 1 = 1

AND ILD\_CD = v\_ild\_cd

AND PLT\_CD = v\_plt\_cd

AND KIND\_CD = 'GN01'

AND TO\_CHAR(YMD, 'YYYY') = v\_year

group by TO\_CHAR(YMD, 'YYYYMMDD')) AA

ORDER BY AA.GRE DESC) AA

WHERE ROWNUM < 2;

RETURN r\_output;

EXCEPTION

WHEN OTHERS THEN

RETURN chr(32) || chr(40) || '-' || chr(41);

END Fn\_getPoMaxDate;

/

**스토어드프로시저**

CREATE OR REPLACE PROCEDURE "GENPROC1"

(s\_ild\_cd IN VARCHAR2,

s\_plt\_cd IN VARCHAR2,

s\_date IN VARCHAR2,

s\_date1 IN VARCHAR2)

IS

I Number(5) := 15;

BEGIN

..

END;

/

패키지.는 프로시저와 함수의 그릅입니다.

CREATE OR REPLACE package SMP\_MAINTENANCE IS

procedure RESET\_SYSMAN;

end;

/

**커서.**

**커서선언(DECLARE)**

**커서 오픈(OPEN)**

**커서 로우 가져옴(FETCH)**

**커서닫기(CLOSE)**

**커서선언(DECLARE)**

CURSOR c\_cap IS

**…**

*/\*===== ÆÇ¸ÅÀü·Â·® =====\*/*

CURSOR GN10 IS

SELECT ild\_cd,yyyy

, NVL(MAX(CASE WHEN mm = '01' THEN capa END),0) v\_01

, NVL(MAX(CASE WHEN mm = '02' THEN capa END),0) v\_02

, NVL(MAX(CASE WHEN mm = '03' THEN capa END),0) v\_03

, NVL(MAX(CASE WHEN mm = '04' THEN capa END),0) v\_04

, NVL(MAX(CASE WHEN mm = '05' THEN capa END),0) v\_05

, NVL(MAX(CASE WHEN mm = '06' THEN capa END),0) v\_06

, NVL(MAX(CASE WHEN mm = '07' THEN capa END),0) v\_07

, NVL(MAX(CASE WHEN mm = '08' THEN capa END),0) v\_08

, NVL(MAX(CASE WHEN mm = '09' THEN capa END),0) v\_09

, NVL(MAX(CASE WHEN mm = '10' THEN capa END),0) v\_10

, NVL(MAX(CASE WHEN mm = '11' THEN capa END),0) v\_11

, NVL(MAX(CASE WHEN mm = '12' THEN capa END),0) v\_12

, NVL(MAX(CASE WHEN mm = 'B01' THEN capa END),0) v\_B01

, NVL(MAX(CASE WHEN mm = 'B02' THEN capa END),0) v\_B02

, NVL(MAX(CASE WHEN mm = 'B03' THEN capa END),0) v\_B03

, NVL(MAX(CASE WHEN mm = 'B04' THEN capa END),0) v\_B04

, NVL(MAX(CASE WHEN mm = 'T01' THEN capa END),0) v\_T01

from

(

SELECT island\_cd\_doseo ild\_cd, SUBSTR(bill\_ym,1,4) yyyy,SUBSTR(bill\_ym,5,2) mm, NVL(SUM(whm\_usekwh),0) capa

FROM IIT0850 a , YTSYST21 b

WHERE TO\_CHAR(a.bill\_ym) BETWEEN TO\_CHAR(in\_yyyy ||'01') AND TO\_CHAR(in\_yyyy ||'12')

AND b.island\_cd\_doseo = GN00\_rec.ild\_cd

AND b.island\_cd\_si = a.island\_cd

AND b.office\_cd\_doseo = a.juris\_officecd

GROUP BY island\_cd\_doseo, bill\_ym

UNION ALL

SELECT island\_cd\_doseo ild\_cd, SUBSTR(bill\_ym,1,4) yyyy,'B0'||CEIL(SUBSTR(bill\_ym,5,2)/3) mm, NVL(SUM(whm\_usekwh),0) capa

FROM IIT0850 a , YTSYST21 b

WHERE TO\_CHAR(a.bill\_ym) BETWEEN TO\_CHAR(in\_yyyy ||'01') AND TO\_CHAR(in\_yyyy ||'12')

AND b.island\_cd\_doseo = GN00\_rec.ild\_cd

AND b.island\_cd\_si = a.island\_cd

AND b.office\_cd\_doseo = a.juris\_officecd

GROUP BY island\_cd\_doseo, SUBSTR(bill\_ym,1,4) ,CEIL(SUBSTR(bill\_ym,5,2)/3)

UNION ALL

SELECT island\_cd\_doseo ild\_cd, SUBSTR(bill\_ym,1,4) yyyy,'T0'||CEIL(SUBSTR(bill\_ym,5,2)/12) mm, NVL(SUM(whm\_usekwh),0) capa

FROM IIT0850 a , YTSYST21 b

WHERE TO\_CHAR(a.bill\_ym) BETWEEN TO\_CHAR(in\_yyyy ||'01') AND TO\_CHAR(in\_yyyy ||'12')

AND b.island\_cd\_doseo = GN00\_rec.ild\_cd

AND b.island\_cd\_si = a.island\_cd

AND b.office\_cd\_doseo = a.juris\_officecd

GROUP BY island\_cd\_doseo, SUBSTR(bill\_ym,1,4) ,CEIL(SUBSTR(bill\_ym,5,2)/12)

)group by ild\_cd,yyyy;

GN10\_rec GN10%ROWTYPE;

**파라미터 있는 커서**

CURSOR GN29(in\_v\_01 NUMBER,in\_v\_02…) IS

…

**커서 오픈(OPEN)**

**커서가 우선 오픈되면 커서 질의가 실행된것이다.**

OPEN c\_cap;

**커서 로우 가져옴(FETCH)**

**커서질의결과를 가져올수 있다.**

FETCH GN00 INTO GN00\_rec;

언제 루프에서 빠져나올지 정하는건은 exit로 정함니다.

EXIT WHEN GN00%NOTFOUND;

**/////////커서 여는거와 함께 변수까지 사용하는방법for 이용**

FOR I in c\_dgubun loop

DBMS\_OUTPUT.PUT\_LINE('code\_name: ' || I.code\_name);

End loop;

**OPEN로도 할수있다**

OPEN c\_dgubun;

LOOP

FETCH c\_dgubun INTO GN01\_rec;

EXIT WHEN c\_dgubun%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('code\_name: ' || GN01\_rec.code\_name);

END LOOP;

*/\*========================= ÆÇ¸ÅÀü·Â·® ==============================\*/*

OPEN GN10;

LOOP

FETCH GN10 INTO GN10\_rec;

EXIT WHEN GN10%NOTFOUND;

*/\*IF GN10\_rec.ild\_cd = '6101' THEN --- ¿ï¸ªµµ´Â 3°³¹ßÀü¼Ò·Î 1°³·Î Ã³¸®*

*w\_plt := '6102';*

*ELSE*

*w\_plt := GN10\_rec.ild\_cd;*

*END IF;*

*\*/*

pw\_01 := GN10\_rec.v\_01;

pw\_02 := GN10\_rec.v\_02;

pw\_03 := GN10\_rec.v\_03;

pw\_04 := GN10\_rec.v\_04;

pw\_05 := GN10\_rec.v\_05;

pw\_06 := GN10\_rec.v\_06;

pw\_07 := GN10\_rec.v\_07;

pw\_08 := GN10\_rec.v\_08;

pw\_09 := GN10\_rec.v\_09;

pw\_10 := GN10\_rec.v\_10;

pw\_11 := GN10\_rec.v\_11;

pw\_12 := GN10\_rec.v\_12;

pw\_B01 := GN10\_rec.v\_B01;

pw\_B02 := GN10\_rec.v\_B02;

pw\_B03 := GN10\_rec.v\_B03;

pw\_B04 := GN10\_rec.v\_B04;

pw\_T01 := GN10\_rec.v\_T01;

BEGIN

INSERT INTO YTGENT10 (ild\_cd, plt\_cd, YEAR, kind, g01\_mon, g02\_mon, g03\_mon, g04\_mon, g05\_mon,

g06\_mon, g07\_mon, g08\_mon, g09\_mon, g010\_mon, g011\_mon, g012\_mon,b01\_mon,b02\_mon,b03\_mon,b04\_mon,t01\_mon)

VALUES (v\_ild\_cd, v\_plt\_cd, v\_yyyy, 'GN10', pw\_01, pw\_02, pw\_03, pw\_04,

pw\_05, pw\_06, pw\_07, pw\_08, pw\_09, pw\_10, pw\_11, pw\_12, pw\_B01, pw\_B02, pw\_B03, pw\_B04, pw\_T01);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

UPDATE YTGENT10 SET g01\_mon = pw\_01, g02\_mon = pw\_02, g03\_mon = pw\_03, g04\_mon = pw\_04,

g05\_mon = pw\_05, g06\_mon = pw\_06, g07\_mon = pw\_07, g08\_mon = pw\_08,

g09\_mon = pw\_09, g010\_mon = pw\_10, g011\_mon = pw\_11, g012\_mon = pw\_12,

b01\_mon = pw\_B01, b02\_mon = pw\_B02, b03\_mon = pw\_B03, b04\_mon = pw\_B04, t01\_mon = pw\_T01

WHERE ild\_cd = v\_ild\_cd

AND plt\_cd = v\_plt\_cd

AND YEAR = v\_yyyy

AND kind = 'GN10' ;

WHEN OTHERS THEN

NULL;

END;

END LOOP;

CLOSE GN10;

**커서닫기(CLOSE)**

CLOSE GN00;

**커서의 속성 4가지**

**%NOTFOUND 커서가 반환할 행들의 유무 / 마지막행 리턴되면 true**

**%FOUND**

**%ROWCOUNT 처음로우0**

**%ISOPEN 열려있음 t 아님 f**

OPEN GN01;

LOOP

FETCH GN01 INTO GN01\_rec;

EXIT WHEN GN01%NOTFOUND;

tw\_01 := GN01\_rec.v\_01;

tw\_02 := GN01\_rec.v\_02;

tw\_03 := GN01\_rec.v\_03;

tw\_04 := GN01\_rec.v\_04;

tw\_05 := GN01\_rec.v\_05;

tw\_06 := GN01\_rec.v\_06;

tw\_07 := GN01\_rec.v\_07;

tw\_08 := GN01\_rec.v\_08;

tw\_09 := GN01\_rec.v\_09;

tw\_10 := GN01\_rec.v\_10;

tw\_11 := GN01\_rec.v\_11;

tw\_12 := GN01\_rec.v\_12;

tw\_B01 := GN01\_rec.v\_B01;

tw\_B02 := GN01\_rec.v\_B02;

tw\_B03 := GN01\_rec.v\_B03;

tw\_B04 := GN01\_rec.v\_B04;

tw\_T01 := GN01\_rec.v\_T01;

*-- dbms\_output.put\_line('GN01>>>> '||' tw\_01 '||tw\_01||' tw\_02 '||tw\_02||' tw\_03 '||tw\_03||' tw\_04 '||tw\_04||' tw\_05 '||tw\_05||' tw\_06 '||tw\_06||' tw\_07 '||tw\_07||' tw\_08 '||tw\_08||' tw\_09 '||tw\_09||' tw\_10 '||tw\_10||' tw\_11 '||tw\_11||' tw\_12 '||tw\_12);*

BEGIN

INSERT INTO YTGENT10 (ild\_cd, plt\_cd, YEAR, kind, g01\_mon, g02\_mon, g03\_mon, g04\_mon, g05\_mon,

g06\_mon, g07\_mon, g08\_mon, g09\_mon, g010\_mon, g011\_mon, g012\_mon, b01\_mon, b02\_mon, b03\_mon, b04\_mon, t01\_mon)

VALUES (v\_ild\_cd, v\_plt\_cd, v\_yyyy, 'GN01', tw\_01, tw\_02, tw\_03, tw\_04,

tw\_05, tw\_06, tw\_07, tw\_08, tw\_09, tw\_10, tw\_11, tw\_12,tw\_B01,tw\_B02,tw\_B03,tw\_B04,tw\_T01);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

UPDATE YTGENT10 SET g01\_mon = tw\_01, g02\_mon = tw\_02, g03\_mon = tw\_03, g04\_mon = tw\_04,

g05\_mon = tw\_05, g06\_mon = tw\_06, g07\_mon = tw\_07, g08\_mon = tw\_08,

g09\_mon = tw\_09, g010\_mon = tw\_10, g011\_mon = tw\_11, g012\_mon = tw\_12,

b01\_mon = tw\_B01,b02\_mon = tw\_B02, b03\_mon = tw\_B03, b04\_mon = tw\_B04,t01\_mon =tw\_T01

WHERE ild\_cd = GN01\_rec.ild\_cd

AND plt\_cd = v\_plt\_cd

AND YEAR = v\_yyyy

AND kind = 'GN01' ;

END;

END LOOP;

CLOSE GN01;

**예외처리 Exception**

**BEGIN**

**…**

**EXCEPTION**

**WHEN TOO\_MANY\_ROWS THEN**

**…**

**WHEN OTHERS THEN**

**Null;**

**END;**

**/**

**사전에 선언된 예외처리 들**

**DUP\_VAL\_ON\_INDEX 존재하는 칼람에 중복값 작성할 때.**

**NO\_DATA\_FOUND SELECT문이 다금과 같이 어떠한 로우도 반환하지 않은경우 발생**

**TOO\_MANY\_ROWS 값이 오버해서 들어갔을 때 4 칸인데 5개들어갔다거나..**

**OTHERS 그외.**

**셀렉트문으로 into 변수 지정할수있다**

SELECT CASE

WHEN sisul\_rec.CAP > 0 AND

NVL(sisul\_rec.CAP, 0) > 0 THEN

ROUND(NVL(i\_sumg, 0) /

(NVL(sisul\_rec.CAP, 0) \* 24 \* 1) \* 100,

0)

ELSE

0

END

INTO i\_erat

FROM DUAL;

**있는 것**

CREATE OR REPLACE PROCEDURE "TEST\_H"( s\_date\_c IN VARCHAR2) IS

s\_date\_y VARCHAR2(8); */\* ÀüÀÏ³¯Â¥ \*/*

CURSOR c\_dgubun IS *--µµ¼­±¸ºÐ*

SELECT \* FROM YTSYST05 where rownum <5;

GN01\_rec c\_dgubun%ROWTYPE;

BEGIN

DBMS\_OUTPUT.ENABLE;

OPEN c\_dgubun;

LOOP

FETCH c\_dgubun INTO GN01\_rec;

EXIT WHEN c\_dgubun%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('code\_name: ' || GN01\_rec.code\_name);

END LOOP;

CLOSE c\_dgubun;

*/\**

*FOR I in c\_dgubun loop*

*DBMS\_OUTPUT.PUT\_LINE('code\_name: ' || I.code\_name);*

*End loop;*

*\*/*

*--DBMS\_OUTPUT.DISABLE;*

COMMIT;

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('ERR CODE : ' || TO\_CHAR(SQLCODE));

DBMS\_OUTPUT.PUT\_LINE('ERR MESSAGE : ' || SQLERRM);

END;

/

**Function**

CREATE OR REPLACE FUNCTION USRSKILL.GET\_BIZ\_NO(stop\_officecd IN VARCHAR2, stop\_comp\_cd IN VARCHAR2)

RETURN VARCHAR2

IS

stop\_biz\_no VARCHAR2(10);

BEGIN

SELECT biz\_no INTO stop\_biz\_no

FROM YKT9010\_TMP

WHERE officecd = stop\_officecd

AND comp\_cd = stop\_comp\_cd;

return stop\_biz\_no;

END;

/

**다이나믹 쿼리**

**Dynamic query**

CREATE OR REPLACE PROCEDURE USRSKILL.Rcnt IS

l\_str varchar2(100);

l\_cnt number;

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*NAME: Rcnt*

*PURPOSE:*

*REVISIONS:*

*Ver Date Author Description*

*--------- ---------- --------------- ------------------------------------*

*1.0 2007.03.02 1. Created this procedure.*

*NOTES:*

*Automatically available Auto Replace Keywords:*

*Object Name: Rcnt*

*Sysdate: 2007.03.02*

*Date and Time: 2007.03.02, 오전 10:37:46, and 2007.03.02 오전 10:37:46*

*Username: (set in TOAD Options, Procedure Editor)*

*Table Name: (set in the "New PL/SQL Object" dialog)*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

BEGIN

for c1 in (select table\_name from user\_tables)

loop

l\_str := 'select count(\*) from '||c1.table\_name;

execute immediate l\_str into l\_cnt;

***dbms\_output.put\_line***(rpad(c1.table\_name,20,' ')||l\_cnt);

end loop;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

NULL;

WHEN OTHERS THEN

*-- Consider logging the error and then re-raise*

RAISE;

END Rcnt;

/

**Exception 및 for**

CREATE OR REPLACE PROCEDURE "GENPROC1A\_WHILE"( start\_ymd IN VARCHAR2,end\_ymd IN VARCHAR2) IS

start\_date date; */\* 전일날짜 \*/*

end\_date date;

date\_num number;

BEGIN

start\_date := to\_date(start\_ymd,'yyyymmdd');

end\_date := to\_date(end\_ymd,'yyyymmdd');

date\_num := end\_date - start\_date;

*--max\_loops integer :=100;*

***DBMS\_OUTPUT.DISABLE***; *--끄는 것*

*--DBMS\_OUTPUT.ENABLE(999999999999);*

FOR I in 0..date\_num LOOP

*--exec GENPROC1A(to\_char(start\_date+I,'yyyymmdd'));*

GENPROC1A(to\_char(start\_date+I,'yyyymmdd'));

*--DBMS\_OUTPUT.PUT\_LINE(to\_char(start\_date+I,'yyyymmdd'));*

END LOOP;

EXCEPTION

WHEN OTHERS THEN

***DBMS\_OUTPUT.PUT\_LINE***('ERR CODE : ' || TO\_CHAR(SQLCODE));

***DBMS\_OUTPUT.PUT\_LINE***('ERR MESSAGE : ' || SQLERRM);

END;

/

**트리거**

CREATE OR REPLACE TRIGGER USRBQM.TRI\_YKT0901

BEFORE DELETE OR INSERT OR UPDATE

ON USRBQM.YKT0901

REFERENCING NEW AS NEW OLD AS OLD

FOR EACH ROW

declare

MAX\_SNAPTIME number;

MAX\_KEY\_NUMBER number;

DMLTYPE char(1);

begin

*------------------------------------------------------------------------------------------------*

*-- 기능인력시스템에 자격정보의 추가/수정/삭제를 실행한다.*

if (inserting) then

delete from USRSKILL.YKT9040

where QUAL\_NO = :new.CERT\_NO

and RESID\_NO = :new.SRL\_NO

and QUAL\_KND\_CD = :new.MJR\_CD;

insert into USRSKILL.YKT9040 (

QUAL\_KND\_CD, RESID\_NO, QUAL\_NO, QUAL\_ACQ\_YMD, QUAL\_END\_YMD,

CRS\_ORG\_CD, PBLSH\_ORG\_CD, INYMD, INID, EDU\_YMD,

QUAL\_STATUS\_CD, FST\_ACQ\_YMD ,ISATTACH, ATTACH\_FILE, IDX\_KEY

) values (

:new.MJR\_CD, :new.SRL\_NO, :new.CERT\_NO, :new.CERT\_START\_YMD, :new.CERT\_END\_YMD,

:new.EDU\_ORG\_CD, :new.ISU\_ORG\_CD, :new.REG\_YMD, substrb(:new.REG\_ID,1,8), :new.EDU\_YMD,

:new.CERT\_STATE, :new.CERT\_YMD, decode(:new.PROOF\_FILE,null,'N','Y'), :new.PROOF\_FILE

,:new.IDX\_KEY

);

delete from SMARTUSER001.YKT9040

where QUAL\_NO = :new.CERT\_NO

and RESID\_NO = :new.SRL\_NO

and QUAL\_KND\_CD = :new.MJR\_CD;

insert into SMARTUSER001.YKT9040 (

QUAL\_KND\_CD, RESID\_NO, QUAL\_NO, QUAL\_ACQ\_YMD, QUAL\_END\_YMD,

CRS\_ORG\_CD, PBLSH\_ORG\_CD, INYMD, INID, EDU\_YMD,

QUAL\_STATUS\_CD, FST\_ACQ\_YMD ,ISATTACH, ATTACH\_FILE, IDX\_KEY, UPDATE\_YMD2

) values (

:new.MJR\_CD, :new.SRL\_NO, :new.CERT\_NO, :new.CERT\_START\_YMD, :new.CERT\_END\_YMD,

:new.EDU\_ORG\_CD, :new.ISU\_ORG\_CD, :new.REG\_YMD, substrb(:new.REG\_ID,1,8), :new.EDU\_YMD,

:new.CERT\_STATE, :new.CERT\_YMD, decode(:new.PROOF\_FILE,null,'N','Y'), :new.PROOF\_FILE

,:new.IDX\_KEY, TO\_CHAR(SYSDATE,'YYYYMMDD')

);

*/\**

*select nvl(max(KEY\_NUMBER),0)+1 into MAX\_KEY\_NUMBER from YKT0901\_EX;*

*INSERT INTO YKT0901\_EX*

*( REG\_YMD , CERT\_STATE , CERT\_YMD , EDU\_YMD , EDU\_ORG\_CD*

*,CERT\_END\_YMD, CERT\_START\_YMD, CERT\_NO , MJR\_CD , SRL\_NO*

*,PROOF\_FILE , STOP\_FROM\_YMD , ISU\_ORG\_CD, UPDATE\_ID, UPDATE\_YMD*

*,UPDATE\_WHERE, REG\_ID , KEY\_NUMBER)*

*VALUES(:new.REG\_YMD , :new.CERT\_STATE , :new.CERT\_YMD , :new.EDU\_YMD , :new.EDU\_ORG\_CD*

*,:new.CERT\_END\_YMD , :new.CERT\_START\_YMD, :new.CERT\_NO , :new.MJR\_CD , :new.SRL\_NO*

*,:new.PROOF\_FILE , :new.STOP\_FROM\_YMD , :new.ISU\_ORG\_CD, :new.UPDATE\_ID , :new.UPDATE\_YMD*

*,:new.UPDATE\_WHERE , :new.REG\_ID , MAX\_KEY\_NUMBER);*

*\*/*

*--이력관리*

elsif (updating) then

update USRSKILL.YKT9040 set

QUAL\_KND\_CD = :new.MJR\_CD,

QUAL\_NO = :new.CERT\_NO,

QUAL\_ACQ\_YMD = :new.CERT\_START\_YMD,

QUAL\_END\_YMD = :new.CERT\_END\_YMD,

CRS\_ORG\_CD = :new.EDU\_ORG\_CD,

PBLSH\_ORG\_CD = :new.ISU\_ORG\_CD,

INYMD = :new.REG\_YMD,

FST\_ACQ\_YMD = :new.CERT\_YMD,

INID = substrb(:new.REG\_ID,1,8),

EDU\_YMD = :new.EDU\_YMD,

QUAL\_STATUS\_CD = :new.CERT\_STATE,

ISATTACH = decode(:new.PROOF\_FILE,null,'N','Y'),

ATTACH\_FILE = :new.PROOF\_FILE

where QUAL\_NO = :old.CERT\_NO

and RESID\_NO = :old.SRL\_NO

and QUAL\_KND\_CD = :old.MJR\_CD;

update SMARTUSER001.YKT9040 set

QUAL\_KND\_CD = :new.MJR\_CD,

QUAL\_NO = :new.CERT\_NO,

QUAL\_ACQ\_YMD = :new.CERT\_START\_YMD,

QUAL\_END\_YMD = :new.CERT\_END\_YMD,

CRS\_ORG\_CD = :new.EDU\_ORG\_CD,

PBLSH\_ORG\_CD = :new.ISU\_ORG\_CD,

INYMD = :new.REG\_YMD,

FST\_ACQ\_YMD = :new.CERT\_YMD,

INID = substrb(:new.REG\_ID,1,8),

EDU\_YMD = :new.EDU\_YMD,

QUAL\_STATUS\_CD = :new.CERT\_STATE,

ISATTACH = decode(:new.PROOF\_FILE,null,'N','Y'),

ATTACH\_FILE = :new.PROOF\_FILE,

UPDATE\_YMD2 = TO\_CHAR(SYSDATE,'YYYYMMDD')

where QUAL\_NO = :old.CERT\_NO

and RESID\_NO = :old.SRL\_NO

and QUAL\_KND\_CD = :old.MJR\_CD;

*/\* select nvl(max(KEY\_NUMBER),0)+1 into MAX\_KEY\_NUMBER from YKT0901\_EX;*

*INSERT INTO YKT0901\_EX*

*( REG\_YMD , CERT\_STATE , CERT\_YMD , EDU\_YMD , EDU\_ORG\_CD*

*,CERT\_END\_YMD, CERT\_START\_YMD, CERT\_NO , MJR\_CD , SRL\_NO*

*,PROOF\_FILE , STOP\_FROM\_YMD , ISU\_ORG\_CD, UPDATE\_ID, UPDATE\_YMD*

*,UPDATE\_WHERE, REG\_ID , KEY\_NUMBER)*

*VALUES(:old.REG\_YMD , :old.CERT\_STATE , :old.CERT\_YMD , :old.EDU\_YMD , :old.EDU\_ORG\_CD*

*,:old.CERT\_END\_YMD , :old.CERT\_START\_YMD, :old.CERT\_NO , :old.MJR\_CD , :old.SRL\_NO*

*,:old.PROOF\_FILE , :old.STOP\_FROM\_YMD , :old.ISU\_ORG\_CD, :old.UPDATE\_ID , :old.UPDATE\_YMD*

*,:old.UPDATE\_WHERE , :old.REG\_ID , MAX\_KEY\_NUMBER);*

*\*/*

elsif (deleting) then

delete from USRSKILL.YKT9040

where QUAL\_NO = :old.CERT\_NO

and RESID\_NO = :old.SRL\_NO

and QUAL\_KND\_CD = :old.MJR\_CD;

delete from SMARTUSER001.YKT9040

where QUAL\_NO = :old.CERT\_NO

and RESID\_NO = :old.SRL\_NO

and QUAL\_KND\_CD = :old.MJR\_CD;

end if;

*------------------------------------------------------------------------------------------------*

*------------------------------------------------------------------------------------------------*

*-- 이력을 추가한다.*

if inserting then DMLTYPE := 'I';

elsif updating then DMLTYPE := 'U';

elsif deleting then DMLTYPE := 'D';

end if;

select nvl(max(SNAPTIME),0)+1 into MAX\_SNAPTIME from YKT0901\_TMP;

insert into YKT0901\_TMP (

SNAPTIME, DMLTYPE,

OLD\_REG\_ID, OLD\_REG\_YMD, OLD\_CERT\_STATE, OLD\_CERT\_YMD, OLD\_EDU\_YMD, OLD\_EDU\_ORG\_CD, OLD\_CERT\_END\_YMD,

OLD\_CERT\_START\_YMD, OLD\_CERT\_NO, OLD\_MJR\_CD, OLD\_SRL\_NO, OLD\_PROOF\_FILE, OLD\_STOP\_FROM\_YMD,

NEW\_REG\_ID, NEW\_REG\_YMD, NEW\_CERT\_STATE, NEW\_CERT\_YMD, NEW\_EDU\_YMD, NEW\_EDU\_ORG\_CD, NEW\_CERT\_END\_YMD,

NEW\_CERT\_START\_YMD, NEW\_CERT\_NO, NEW\_MJR\_CD, NEW\_SRL\_NO, NEW\_PROOF\_FILE, NEW\_STOP\_FROM\_YMD,

OCCR\_YMD

) values (

MAX\_SNAPTIME, DMLTYPE,

:old.REG\_ID, :old.REG\_YMD, :old.CERT\_STATE, :old.CERT\_YMD, :old.EDU\_YMD, :old.EDU\_ORG\_CD, :old.CERT\_END\_YMD,

:old.CERT\_START\_YMD, :old.CERT\_NO, :old.MJR\_CD, :old.SRL\_NO, :old.PROOF\_FILE, :old.STOP\_FROM\_YMD,

:new.REG\_ID, :new.REG\_YMD, :new.CERT\_STATE, :new.CERT\_YMD, :new.EDU\_YMD, :new.EDU\_ORG\_CD, :new.CERT\_END\_YMD,

:new.CERT\_START\_YMD, :new.CERT\_NO, :new.MJR\_CD, :new.SRL\_NO, :new.PROOF\_FILE, :new.STOP\_FROM\_YMD,

sysdate

);

*------------------------------------------------------------------------------------------------*

end;

/

**트리거2**

CREATE OR REPLACE TRIGGER USRSKILL.TRI\_YKT8060

AFTER DELETE OR INSERT OR UPDATE

ON USRSKILL.YKT8060

REFERENCING NEW AS NEW OLD AS OLD

FOR EACH ROW

DECLARE

license\_seq number(10);

biz\_no varchar2(10);

pause\_str\_date VARCHAR2 (8) ;

pause\_end\_date VARCHAR2 (8) ;

cancel\_date VARCHAR2 (8) ;

status VARCHAR2 (1) ;

BEGIN

*--dbms\_output.ENABLE(99999999999);*

IF (INSERTING) THEN

SELECT TO\_CHAR(sysdate, 'YYYY') || TO\_CHAR(sysdate, 'MM') || NO INTO license\_seq

FROM (

SELECT CASE WHEN MAX(license\_seq) IS NULL

THEN '0001'

ELSE LPAD(substr(max(license\_seq), 7, 4)+1, 4, '0')

END NO

FROM PSI\_REV\_LICENSE@KEBS2 AA

WHERE substr(license\_seq, 0, 6) = TO\_CHAR(sysdate,'YYYYMM')

) ;

biz\_no := GET\_BIZ\_NO(:new.stop\_officecd, :new.stop\_comp\_cd);

select CASE WHEN :new.stop\_kind = 'JL001' or :new.stop\_kind = 'JL002' THEN :new.stop\_start\_ymd END into pause\_str\_date from dual;

select CASE WHEN :new.stop\_kind = 'JL001' or :new.stop\_kind = 'JL002' THEN :new.stop\_end\_ymd END into pause\_end\_date from dual;

select CASE WHEN :new.stop\_kind = 'JL003' THEN :new.stop\_start\_ymd END into cancel\_date from dual;

select CASE WHEN :new.stop\_kind = 'JL001' or :new.stop\_kind = 'JL002' THEN 'P'

WHEN :new.stop\_kind = 'JL003' THEN 'D'

ELSE '' END into pause\_end\_date from dual;

*-- dbms\_output.put\_line( 'biz\_no=====>'||biz\_no);*

INSERT INTO PSI\_REV\_LICENSE@KEBS2

(

license\_seq

,biz\_no

,license\_regno

,government\_office

,license\_regdate

,license\_end\_date

,pause\_str\_date

,pause\_end\_date

,cancel\_date

,status

,workdate

)VALUES(

license\_seq

,biz\_no

,license\_seq

,'한국전력공사'

,'' *--인증시작일자*

,'' *--인증만료일자*

,pause\_str\_date

,pause\_end\_date

,cancel\_date

,cancel\_date

,sysdate

)

;

END IF;

END;

/

**TRIGGER 2**

CREATE OR REPLACE TRIGGER USRSKILL.TRI\_YKT8010

AFTER DELETE OR INSERT OR UPDATE

ON USRSKILL.YKT8010

REFERENCING NEW AS NEW OLD AS OLD

FOR EACH ROW

DECLARE

license\_seq number(10);

biz\_no varchar2(10);

pause\_str\_date VARCHAR2 (8) ;

pause\_end\_date VARCHAR2 (8) ;

cancel\_date VARCHAR2 (8) ;

status VARCHAR2 (1) ;

BEGIN

IF (INSERTING) OR (UPDATING) THEN

IF :new.CNCL\_DATE IS NULL THEN

status :='Y';

ELSIF :new.CNCL\_DATE IS NULL AND:new.APPY\_CNCL\_DATE IS NOT NULL THEN

RETURN;

ELSE

status :='D';

END IF ;

SELECT NO INTO license\_seq

FROM (

SELECT CASE WHEN MAX(license\_seq) IS NULL

THEN 1

ELSE max(license\_seq)+1

END NO

FROM PSI\_REV\_LICENSE@KEBS2 AA

) ;

biz\_no := GET\_BIZ\_NO(:new.officecd, :new.comp\_cd);

INSERT INTO PSI\_REV\_LICENSE@KEBS2

(

license\_seq

,biz\_no

,LICENSE\_KIND

,license\_regno

,government\_office

,license\_regdate

,license\_end\_date

,pause\_str\_date

,pause\_end\_date

,cancel\_date

,status

,workdate

)VALUES(

license\_seq

,biz\_no

,:new.CONFIRM\_GUBUN *--인증종류 M 겸임 J 지중배전전문*

,:new.confirm\_no *--인증번호*

,'한국전력공사' *--면저관청*

,TO\_DATE(:new.confirm\_date,'YYYYMMDD') *--인증시작일자*

,:new.confirm\_end\_date *--인증만료일자*

,'' *-- 면허정지시작일*

,'' *--면허정지정료일*

,'' *--면허취소일*

,status *--상태(Y:인증(등록), D:취소, P:정지)*

,sysdate *--작업일자*

)

;

END IF;

END;

/