Kivételkezelés

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

set serveroutput on

begin

FOR err\_num IN 1..9999 LOOP

DBMS\_OUTPUT.PUT\_LINE(SQLERRM(-err\_num));

END LOOP;

end;

/

INVALID\_CURSOR -1001

INVALID\_NUMBER -1722 karakterlánc sikertelen számmá konvertálása

NO\_DATA\_FOUND -100 select .. into… SELECT utasitas nem ad vissza sort

TOO\_MANY\_ROWS -1422 egy sort kellett volna visszaadnia egy SELECT-nek, de többet kaptunk

VALUE\_ERROR -6502

ZERO\_DIVIDE -1476

- Felhasználói kivételek

- a DECLARE szakaszban: kivételnév EXCEPTION;

- végrehajtható szegmensben RAISE utasítással váltható ki;

DECLARE

kivetelnev EXCEPTION;

PRAGMA EXCEPTION\_INIT(kivetelnev, kód);

BEGIN

RAISE kivetelnev;

EXCEPTION

WHEN kivetelnev [OR kivetelnev….] THEN ut(ok)

[WHEN kivetelnev…..]

………..

[WHEN OTHERS THEN ut(ok)]

END;

/

RAISE\_APPLICATION\_ERROR( kód, üzenet[, ])

SQLCODE

SQLERRM

SQLERRM(kód)

DECLARE

A NUMBER NOT NULL:=1;

B NUMBER;

BEGIN

A:=B;

EXCEPTION

WHEN VALUE\_ERROR THEN

DBMS\_OUTPUT.PUT\_LINE('HOPPÁ!');

END;

/

DECLARE

X EMP.ENAME%TYPE;

BEGIN

SELECT ENAME INTO X FROM EMP WHERE job='&x';

DBMS\_OUTPUT.PUT\_LINE(x);

EXCEPTION

WHEN NO\_DATA\_FOUND OR TOO\_MANY\_ROWS

THEN RAISE\_APPLICATION\_ERROR(-20001,'Nem tudtam egyértelműen azonosítani a dolgozót');

END;

/

/\* 'president' 0 adat

'PRESIDENT' 1 adat (sor)

'MANAGER' több adat (sor) \*/

Példa:

DECLARE

v\_ber EMP.sal%TYPE;

v\_nev EMP.ename%TYPE;

BEGIN

v\_ber := 5000;

SELECT ename

INTO v\_nev

FROM EMP

WHERE sal = v\_ber;

DBMS\_OUTPUT.PUT\_LINE(v\_nev);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nincs ilyen fizetés');

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Több embernek is ez a fizetése');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Egyéb hiba');

END;/

Példa:

DECLARE

nincs\_vevo EXCEPTION;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Ez vegrehajtodik');

RAISE nincs\_vevo;

DBMS\_OUTPUT.PUT\_LINE('Ez nem hajtodik vegre');

EXCEPTION

WHEN nincs\_vevo THEN

DBMS\_OUTPUT.PUT\_LINE('Nincs ilyen vevo');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Egyéb hiba');

END;

**Triggerek**

CREATE [OR REPLACE] TRIGGER név

{ BEFORE | AFTER | INSTEAD OF }

{ DELETE | INSERT | UPDATE [OF oszlopok] }

[ OR { DELETE | INSERT | UPDATE [OF oszlopok] } ]

ON tábla

[ REFERENCING [OLD AS régi] [NEW AS új] ]

[ FOR EACH ROW ]

[WHEN (feltétel)]

PL/SQL programblokk;

**- Utasítás szintű trigger**

**Példa:**

CREATE TABLE tmp (

szam NUMBER(20)

);

CREATE OR REPLACE TRIGGER kiírás

BEFORE INSERT

ON tmp

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Új sort fogunk beszúrni');

END;

CREATE OR REPLACE TRIGGER kiiras2

AFTER INSERT

ON tmp

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Új sort szúrtunk be');

END;

INSERT INTO tmp VALUES(1122);

DROP TABLE tmp;

**- Sor szintű trigger**

**Példa:**

CREATE OR REPLACE TRIGGER nem\_csokken

AFTER UPDATE OF ber

ON munkatars

FOR EACH ROW

WHEN (OLD.ber > NEW.ber)

UPDATE munkatars SET ber = :OLD.ber

WHERE partner\_id = :NEW.partner\_id;

UPDATE munkatars SET ber = 1 WHERE partner\_id = 1;

CREATE TRIGGER NESZE

AFTER INSERT OR UPDATE OR DELETE ON dept

BEGIN

DBMS\_OUTPUT.PUT\_LINE(CASE

WHEN INSERTING THEN 'Beszúrtunk'

WHEN UPDATING THEN 'Módosítottunk'

WHEN DELETING THEN 'Töröltünk'

END);

END;

/

SET SERVEROUT ON

INSERT INTO dept VALUES (80,'reszl1','loc1');

INSERT INTO dept VALUES (81,'reszl2','loc2');

DELETE FROM dept WHERE deptno=80 OR deptno=81;

SELECT DESCRIPTION FROM user\_triggers;

SELECT line, text FROM user\_source

WHERE UPPER(TYPE)='TRIGGER' AND

UPPER(NAME)='NESZE';

DROP TRIGGER nesze;

CREATE TRIGGER minimálbér

AFTER UPDATE OF sal ON emp

REFERENCING OLD AS régi NEW AS új

FOR EACH ROW

WHEN (régi.sal>új.sal AND új.sal<500)

begin

UPDATE emp SET sal=:régi.sal

WHERE empno=:régi.empno;

End;

/

alter trigger tnev enable;

set serveroutput on

CREATE OR REPLACE TRIGGER mitcsinaltunk

AFTER INSERT OR UPDATE OR DELETE ON dept

BEGIN

DBMS\_OUTPUT.PUT\_LINE(CASE

WHEN INSERTING THEN 'Beszúrtunk.'

WHEN UPDATING THEN 'Módosítottunk.'

WHEN DELETING THEN 'Töröltünk.'

END);

END;

/

show errors

INSERT INTO dept VALUES (80, 'reszl1', 'loc1');

INSERT INTO dept VALUES (81, 'reszl2', 'loc2');

DELETE FROM dept WHERE deptno = 80 or deptno = 81;

select description from user\_triggers;

select line, text FROM user\_source where upper(type)='TRIGGER' and UPPER(name) ='MITCSINALTUNK';

DROP TRIGGER mitcsinaltunk;

set serveroutput on

CREATE OR REPLACE TRIGGER rakerdez

AFTER INSERT OR UPDATE OR DELETE ON dept

BEGIN

DBMS\_OUTPUT.PUT\_LINE(CASE

WHEN INSERTING THEN 'Beszúrtunk.'

WHEN UPDATING THEN 'Módosítottunk.'

WHEN DELETING THEN 'Töröltünk.'

END);

END;

/

show errors

INSERT INTO dept VALUES (80, 'reszl1', 'loc1');

INSERT INTO dept VALUES (81, 'reszl2', 'loc2');

DELETE FROM dept WHERE deptno = 80 or deptno = 81;

select description from user\_triggers;

select line, text FROM user\_source where upper(type)='TRIGGER' and UPPER(name) ='RAKERDEZ';

DROP TRIGGER rakerdez;

CREATE OR REPLACE TRIGGER dml\_trg

AFTER UPDATE OF o1

ON t1

FOR EACH ROW

WHEN (NEW.o1 >= 1000)

DECLARE

v\_local VARCHAR2(20);

BEGIN

v\_local := :new.o1;

END;

/

CREATE OR REPLACE TRIGGER dml\_trg2

BEFORE INSERT OR DELETE OR UPDATE

ON t2

BEGIN

null;

END;

/

CREATE OR REPLACE TRIGGER dml\_trg3

INSTEAD OF UPDATE

ON v1

FOR EACH ROW -- enelkul is sortrigger lesz az instead of miatt

BEGIN

null;

END;

/

CREATE OR REPLACE TRIGGER db\_trg

AFTER SERVERERROR OR LOGON OR STARTUP OR SUSPEND

ON DATABASE

DECLARE -- A PL/SQL blokknak lehet deklarációs része is

i number;

BEGIN

null;

END ;

/

CREATE OR REPLACE TRIGGER db\_trg2

BEFORE LOGOFF OR SHUTDOWN

ON DATABASE

BEGIN

null;

END ;

/

CREATE OR REPLACE TRIGGER any\_ddl\_trg

BEFORE DDL

ON DATABASE

BEGIN

null;

END ;

/

CREATE OR REPLACE TRIGGER spec\_ddl\_trg

BEFORE ALTER OR ANALYZE OR ASSOCIATE STATISTICS OR AUDIT OR CREATE OR DROP OR GRANT

ON DATABASE

BEGIN

null;

END ;

/

CREATE OR REPLACE TRIGGER sajat\_ddl\_trg

BEFORE DDL ON nikovits.SCHEMA

BEGIN

null;

END ;

/

CREATE OR REPLACE TRIGGER sajat\_logon\_trg

AFTER LOGON ON SCHEMA

BEGIN

null;

END ;

/

CREATE OR REPLACE TRIGGER any\_logon\_trg

AFTER LOGON ON DATABASE

BEGIN

null;

END ;

/