

7. gyakorlat

I. PLSQL

1. Blokk

A PL/SQL alapvető egysége a blokk. Lehet nevet adni neki, ekkor az adatbázisban tárolódik, és többször meghívható újra. Maradhat név nélkül is, ekkor nem tárolódik az adatbázisban, és nem használható fel újra, hanem azonnal végrehajtható.

Egy blokk általános kinézete:

```
DECLARE
    Variable declaration
BEGIN
    Program Execution
EXCEPTION
    Exception handling
END;
```

Declare: típus, nevesített konstans, változó, kivétel, kurzor

Szintaxisa:

```
identifier [CONSTANT] datatype [NOT NULL] [:= |
DEFAULT expr];
```

Pl.:

konstans érték megadása

```
konstansom CONSTANT NUMBER NOT NULL:=42;
```

kezdeti érték megadása:

```
valtozom VARCHAR2(40) := 'ezakezdetiérték';
```

2. A % típusú attribútumok

Mindig tudni kell, milyen típusú adatokkal dolgozunk. Erre jó a %TYPE.

Segítségével a % előtt meghatározott tábla meghatározott attribútumának típusát „vesszük át”.

Pl.:

```
idtype mb18____db.users.id%TYPE;
```

az „idtype” nevű változónk típusa az lesz, ami a „users” táblában az „id” attribútumé.

A%ROWTYPE egy egész sor típusát megában foglalja, vagyis tulajdonképpen a fejléct.

Pl.: `users_egysor mb18___db.users%ROWTYPE;`

A „users_egysor” nevű változónk elbír annyi, és olyan típusú adatot, amelyen sorok a „users” táblánkban szerepelnek.

Rekord: a programozó által definiálható változó (struct)

Kurzor: SELECT állítás a deklarációs részben definiálva és elnevezve, azaz:

```
CURSOR kurzor_neve IS SELECT ... FROM...  
WHERE...;
```

Program: amit a blokk végrehajt (pl. feltétel, ciklus, SQL parancs)

Lekérdezés blokkban:

```
DECLARE  
    v_ename VARCHAR2(10);  
    v_id NUMBER(3);  
BEGIN  
    SELECT ename, id  
    INTO v_ename, v_id  
    FROM emp  
    WHERE id = '112';  
END;
```

Állítsuk be, hogy a számokat át lehessen konvertálni:
`ALTER SESSION SET NLS_NUMERIC_CHARACTERS = '.,';`

II. Gyakorló feladatok

1. Készítsük egy hello world nevű programot, ami egy változó segítségével kiírja azt, hogy hello world!
(`dbms_output.put_line();`)

Kiíratás

```
set serveroutput on;  
dbms_output.put_line();
```

2. Bővítsük a programot, úgy hogy egy ciklus segítségével 10x írja ki azt!

Ciklus

egyszerű ciklus:

```
LOOP mitörténjen  
EXIT WHEN kilépési feltétel  
END LOOP;
```

for ciklus:

```
FOR i IN 1..10 LOOP  
mitörténjen  
END LOOP;
```

while ciklus:

```
WHILE i<|>|==valami LOOP  
mitörténjen  
END LOOP;
```

3. Írassuk ki a páros számok négyzetét 1 és 15 között.
4. Iteráljunk végig a dcdb.components táblán egy kurzor segítségével és írjuk ki azon komponensek id-ját és nevét, ahol a chemical_formula **NEM** 'Not Available'

Kurzor:

Cursor általános alakja:

```
DECLARE  
    variables;  
    records;  
    create a cursor;  
BEGIN  
    OPEN cursor;  
    FETCH cursor;  
    process the records;  
    CLOSE cursor;  
END;
```

Elágazás

Ha a kiválasztott sorra valami igaz vagy nem igaz, ez vagy az történjen.

Szintaxisa:

```
IF feltétel THEN utasítás [utasítás]...  
[ELSIF feltétel THEN utasítás [utasítás]...]...  
[ELSE utasítás [utasítás]...]
```

```
END IF;
```

Case utasítás egy elágazó utasítás, ahol az egymást követő kölcsönösen kizáró tevékenységek közül egy kifejezés értékei, vagy feltételek teljesülése szerint lehet választani.

Szintaxisa:

```
CASE [kifejezés] --kifejezés: amit a  
feltételekkel összehasonlítunk  
    WHEN {kifejezés | feltétel} THEN utasítás  
    [utasítás]...  
    [WHEN {kifejezés | feltétel} THEN utasítás  
    [utasítás]...]...  
    [ELSE utasítás [utasítás]...]  
END CASE;
```

5. Hozzunk létre egy függvényt, mely kiszámolja egy r sugarú kör területét!
6. Hívjuk meg ezt a függvényt, a visszatérési értékét írassuk ki!

Függvények létrehozása

```
CREATE OR REPLACE FUNCTION fgvneve  
    (paraméterlista)  
RETURN visszatérési érték típusa  
IS  
    begin  
    end
```

7. Készítsen egy eljárást, ami egy adott típusú (bemenő paraméter) komponensnek kiadja a kémiai formuláját!

Eljárás létrehozása

```
CREATE OR REPLACE PROCEDURE procneve  
    (paraméterlista)  
IS  
    innentől egy rendes blokk...
```

8. Írjunk egy PL/SQL szkriptet, amiben egy megadott tábla egy megadott oszlopában lévő különböző értékek számát íratjuk ki. (A tábla és az oszlop nevét változóban adjuk meg!)