Adatbázisok

10. gyakorlat:

PL/SQL bemutatása

Változók, vezérlési szerkezetek

Kurzorok és kivételkezelés

Bevezetés a PL/SQL-be

- procedurális programnyelv,
- olyan feladatok megoldására, amelyekre az alap SQL nem, vagy csak körülményes módon ad megoldást,
- blokkstrukturált felépítésű,

Bevezetés a PL/SQL-be (2.)

- blokk részei:
  - deklarációs rész (opcionális),
  - végrehajtható rész,
  - kivételkezelő rész (opcionális),

```
[DECLARE
     változók deklarálása]
BEGIN
     utasítások
[EXCEPTION
     kivételkezelés ]
END;
/
```

Első lépések

```
SET serveroutput ON
SET verify OFF
BEGIN
 dbms_output.put_line('Hello
  World!');
END;
/
```

- Output beállítások: egy munkamenetben csak egyszer kell kiadni a parancsot
  - **SET serveroutput ON**
  - **SET verify OFF**

Feladat

- Kérjük be a felhasználó nevét, majd üdvözöld név szerint!

- ACCEPT
  - Nem a PL/SQL-hez tartozik az utasítás, hanem az SQL parancssorhoz.
  - ACCEPT nev
  - ACCEPT nev PROMPT 'Kérem a neved:'
  - változó használata: &nev

Megoldás

```
ACCEPT nev
  PROMPT 'Kérem a neved:'
BEGIN
  dbms_output.put_line('Hello
  ' || '&nev');
END;
/
```

Változók

```
ACCEPT x PROMPT 'Kérem a számot:'
DECLARE
  szam NUMBER;
BEGIN
  szam := &x;
  dbms_output.put_line(szam ||
  ' négyzete: ' ||szam*szam);
END;
/
```

Elágazások (IF utasítás)

```
IF logikai_feltétel
THEN -- a logikai feltétel
igaz volta esetén fut le
[ELSIF logikai feltétel2 THEN
-- a logikai feltétel2 igaz
volta esetén fut le]
[ELSE – minden egyéb más
esetben ez fut le]
END IF;
```

Elágazások (IF példa)

```
...
BEGIN
  szam := &x;
  IF szam>0 THEN
        dbms_output.put_line('Pozitív.');
  ELSIF szam<0 THEN
        dbms_output.put_line('Negatív.');
  ELSE
        dbms_output.put_line('Nulla.');
  END IF;
END;
/
```

Feladat

- Írja ki a számról, hogy osztható-e öttel.

Segítség:

MOD(x,y) : **x** mennyi maradékot ad **y**-nal osztva

Pl.: MOD(14,3) = 2

Ciklusok
(**loop** ciklus, feltétellel ellátott **exit** utasítással)

```
DECLARE
  tol NUMBER;
  ig  NUMBER;
BEGIN
  tol:=1;
  ig:=10;
  LOOP
        dbms_output.put_line(tol);
        EXIT WHEN tol=ig;
        tol:=tol+1;
  END LOOP;
END;
/
```

`for` és `while` ciklusváltozatok

```
WHILE feltétel                FOR i IN
  tól..ig
LOOP                        LOOP
  ....                          ....
END LOOP;                     END LOOP;
```

Feladat

- Alakítsd át az előző példát úgy, hogy:
  - FOR ciklust használjon;
  - a határokat a felhasználó adja meg;
  - egy sorba írja a számokat,
  - csak a páratlanokat írja ki.

Megoldás

```
ACCEPT mettol PROMPT 'Mettől:'
ACCEPT meddig PROMPT 'Meddig:'
DECLARE
  tol NUMBER;
  ig  NUMBER;
BEGIN
  tol:=&mettol;
  ig:=&meddig;
  FOR i in tol..ig
  LOOP
       IF MOD(i,2)=1 THEN
            dbms_output.put(i || ', ');
       END IF;
  END LOOP;
  dbms_output.put_line('');
END;
/
```

Lekérdezés PL/SQL blokkban

```
ACCEPT részleg PROMPT 'Kérem a részleg
  nevét:'
DECLARE
  átlag number;
BEGIN
  SELECT AVG(salary) INTO átlag
  FROM employees INNER JOIN departments
  USING(department_id)
  WHERE lower(department_name) =
    lower('&részleg');
  dbms_output.put_line(átlag);
END;
/
```

Feladat

- Kérj be egy munkakört. Írd ki az adott munkakörben dolgozók összfizetését.

KURZOROK

Kurzor fogalma és fajtái

Nem halmazként használjuk az adatokat, hanem olyan táblákként, amely táblákon rekordonként (soronként) végig lehet „lépkedni”.

Kurzor fajták:

- Implicit: nem kell külön deklarálni, a szövegkörnyezetből kiderül,
- Explicit: létrehozunk egy kurzort a lekérdezés típusa alapján

Implicit kurzor

```
DECLARE
  egysor employees%ROWTYPE;
BEGIN
   FOR egysor IN (SELECT * FROM
  employees)
   LOOP
     dbms_output.put('Név: ' ||
  egysor.last_name);
     dbms_output.put_line(',  Fizetés:
  ' || egysor.salary);
   END LOOP;
END;
/
```

Explicit kurzor

```
ACCEPT also PROMPT 'Kérem az alsó határt:'
ACCEPT felso PROMPT 'Kérem az felső határt:'
DECLARE
  CURSOR kurzor IS select * from employees;
  rekord employees%ROWTYPE;
BEGIN
  FOR rekord IN kurzor
  LOOP
    IF rekord.salary BETWEEN &also AND &felso
    THEN
 dbms_output.put_line(rekord.last_name||',  '
 || rekord.salary);
    END IF;
  END LOOP;
END;
/
```

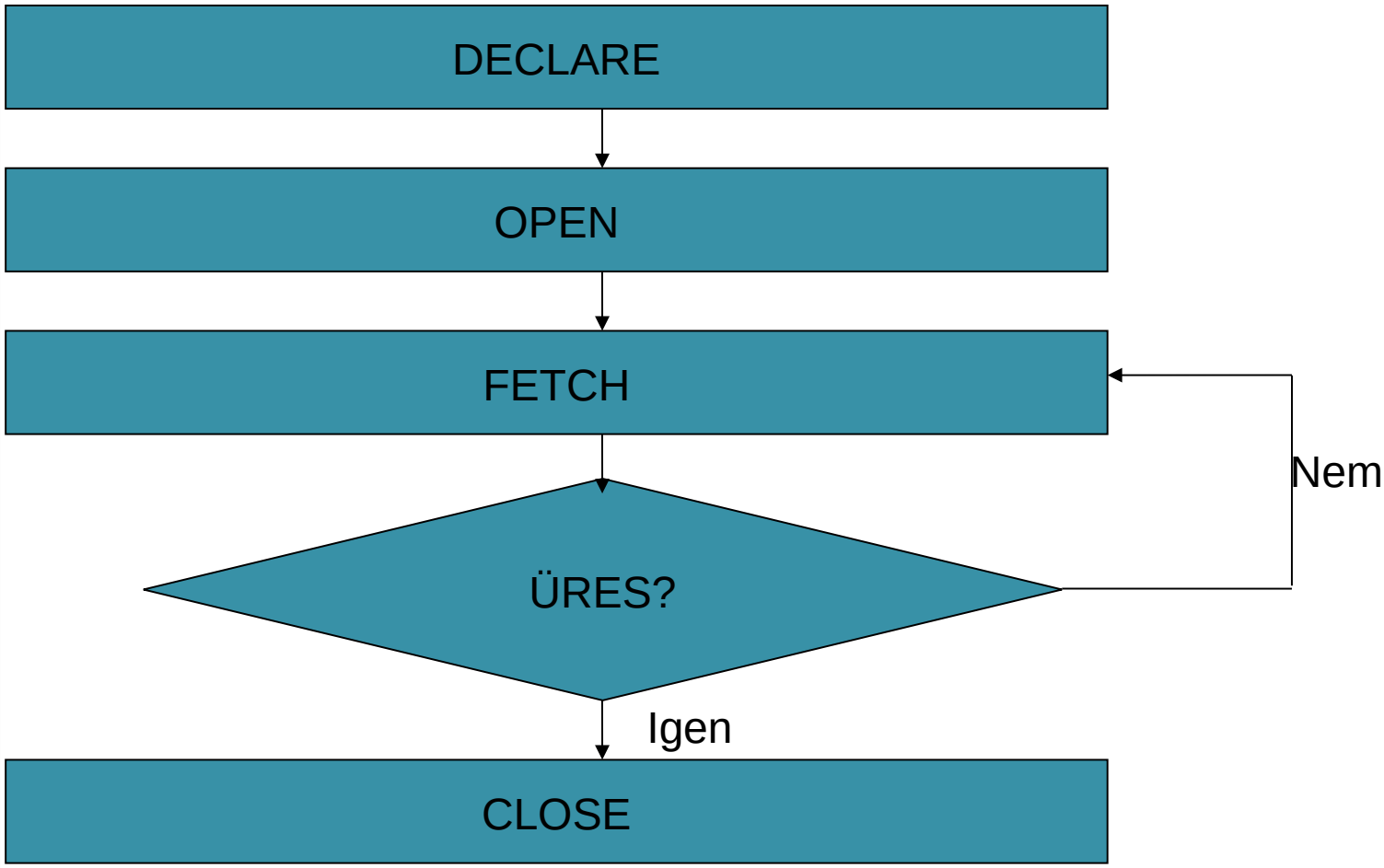
Feladat

- Listázd azon dolgozók nevét, fizetését és részlegük nevét, akiknek a fizetése nagyobb a részlegük átlagos fizetésénél.

  - Tipp: a rekord változó típusa nem csak egy tábla sorának felelhet meg, hanem egy kurzor sorának is!

    rekord **kurzor%ROWTYPE**

Kurzorműveletek



Kurzor-attribútumok

- **%FOUND** – Sikeres volt az előző FETCH művelet?
- **%NOTFOUND** – Sikertelen volt az előző FETCH művelet?
- **%ROWCOUNT** – Feldolgozott sorok darabszáma
- **%ISOPEN** – Volt OPEN művelet?

Kurzor használata

```
ACCEPT also PROMPT 'Kérem az alsó határt:'
ACCEPT felso PROMPT 'Kérem az felső határt:'
DECLARE
  CURSOR kurzor IS select * from employees;
  rekord employees%ROWTYPE;
BEGIN
 OPEN kurzor;
 LOOP
    FETCH kurzor INTO rekord;
    EXIT WHEN kurzor%NOTFOUND;
    IF rekord.salary BETWEEN &also AND &felso
        THEN dbms_output.put_line(rekord.last_name||', '
  ||rekord.salary);
    END IF;
 END LOOP;
 CLOSE kurzor;
END;
/
```

Feladat

- Módosítsd az előző példát úgy, hogy csak az első 3 találatot listázza.

Kurzor használata módosításra

```
CURSOR kurzor IS
    SELECT * FROM táblanév
    FOR UPDATE [OF Oszlopnév]
[NOWAIT];
```

- **NOWAIT**: ne várakozzon a zárolt sorokra

Kurzor használata módosításra

```
Declare
  cursor kurzor is select*from employees2 for update of
  salary;
  rekord kurzor%ROWTYPE;
  fizetes number;
BEGIN
 OPEN kurzor;
  LOOP
    FETCH kurzor INTO rekord;
    EXIT WHEN kurzor%NOTFOUND;
    fizetes := rekord.salary * 1.2;
    UPDATE employees2 set salary = fizetes WHERE
  CURRENT OF kurzor;
  END LOOP;
 CLOSE kurzor;
END;
/
```

KIVÉTELKEZELÉS

Kivételkezelés

```
...
EXCEPTION
 WHEN kivételnév THEN
     utasítások;
END;
```

Néhány, előre definiált rendszerkivétel

NO_DATA_FOUND

- SELECT INTO nem adott vissza sort
- Vigyázat! Csoportfüggvények nem dobnak ilyet!

TOO_MANY_ROWS

- SELECT INTO egynél több sort adott vissza

INVALID_NUMBER

- számmá konvertálás hibája

ZERO_DIVIDE

- nullával osztás

OTHERS

- "Minden más"

Példa

```
ACCEPT azon PROMPT 'Kérem az azonosítót:'
DECLARE
  név employees.last_name%TYPE;
BEGIN
  SELECT last_name INTO név
  FROM employees WHERE employee_id = &azon;
  dbms_output.put_line('Dolgozó neve: '||
  név);
  EXCEPTION
    WHEN NO_DATA_FOUND THEN
      dbms_output.put_line('Nincs ilyen
  dolgozó!');
    WHEN OTHERS THEN
      dbms_output.put_line('Ismeretlen
  hiba.');
END;
/
```

Példa

```
ACCEPT részleg PROMPT 'Kérem a részleg nevét:'
DECLARE
 átlag number;
BEGIN
  SELECT AVG(salary) INTO átlag
  FROM employees INNER JOIN departments
  USING(department_id)
  WHERE lower(department_name) = lower('&részleg');
  dbms_output.put_line(átlag);
  EXCEPTION WHEN NO_DATA_FOUND THEN
 dbms_output.put_line('Nincs ilyen részleg!');
END;
/
```

- Nem dob **NO_DATA_FOUND**-ot!

Példa saját kivételre

```
ACCEPT szam PROMPT 'Szám:'
DECLARE
  nulla EXCEPTION;
BEGIN
  IF &szam=0 THEN
    RAISE nulla;
  END IF;
  EXCEPTION WHEN nulla THEN
       dbms_output.put_line('Ez nulla!');
END;
/
```