

Adatbázis rendszerek II.

Gyakorlati feladatsor

Készítette:
Oravecz Áron
D3U3EE

Dátum:
2022 04.13

1. Készítse el csomag1 néven a csomag specifikációt, amely tartalmaz egy publikus változót (x int:=5;) és eljárást (procedure kiir;), majd futtassa. **csomag1.sql**

1a Készítse el csomag1 néven a csomag törzsét, amely tartalmaz egy konzolra való kiírást (dbms_output.put_line('x= '||x;)) majd futtassa. **csomag1Torzs.sql**

2. Készítse el elj.auto néven a következő alprogramot, amely törli a megadott kor-tól idősebb autókat. **elj.auto.sql**

2a. Készítse el fvg.auto.maxar néven a következő alprogramot, amely kiírja az autó táblából a maxár autót. **fvg.auto.maxar.sql**

Készítse el elj.auto.sql és a fvg.auto.maxar.sql alprogramot egy csomagba. Futtatás. **aut_csomag1.sql**

Készítse el aut_csomag1Torzs néven a csomag törzsét, ahol szerepel az eljárás és a függvény is - Futtatás. **aut_csomag1Torzs.sql**

3.

1. Hozza létre az alábbi táblát:

Hallgato [ID int pk, Nev Varchar(25), Fizetes int]

2. Szűrjön be 4-5 rekordot a Szemely táblába csoporttársak. (HallgatoFeltolt.sql)

3. Készítsen egy tárolt eljárást (ModF), mely módosítja egy paraméterként megadott ID személy fizetését egy paraméterként megadott értékre. (ModF.sql)

4. Készítsen egy tárolt függvényt (KerN), mely visszaadja egy paraméterként megadott ID hallgató nevét. (KerN.sql)

A két alprogramot helyezze el egy csomagban

5. Készítse el a csomag specifikációt (H_cs.sql)

6. Készítse el a csomag törzsét (H_csTorzs.sql)

7. Futtassa a csomagban lévő alprogramokat. H_cs.ModF_fut.sql

H_cs.KerN_fut.sql

Kurzor készítése

1. A korábban elkészített *Kategoria* táblát használjuk kurzor készítéshez.

Készítsen egy *tlp* nevű tárolt eljárást, amely a *Kategoria* tábla adatait írja ki a konzolra. (*kurzor_kategoria.sql*) - Futtatás.

1.

create or replace procedure tlp as

```
    cursor cur is select * from kategoria;
```

```
    cv cur%rowtype;
```

```
begin
```

```
    open cur;
```

```
    loop
```

```
        fetch cur into cv;
```

```
        exit when cur%notfound;
```

```
        dbms_output.put_line('Név: ' || cv.nev);
```

```
        dbms_output.put_line('Feldolgozva: ' || cur%rowcount);
```

```
    end loop;
```

```
    close cur;
```

```
end;
```

2. Készítsen egy olyan *tlpfor* nevű tárolt eljárást, amely a *Kategoria* tábla adatait dolgozza fel FOR ciklussal. (kurzorfor_kategoria.sql) – Futtatás.

2.

create or replace procedure tlpfor as

```
    cursor cur is select * from kategoria;
```

```
begin
```

```
    for cv in cur
```

```
    loop
```

```
        dbms_output.put_line('Név: '||cv.nev);
```

```
        dbms_output.put_line('Feldolgozva: '||cur%rowcount);
```

```
    end loop;
```

```
end;
```

3. Készítse el a *Piros_Auto* táblát!

Piros_Auto (rsz varchar2 (20), tipus varchar2 (20), szin varchar2 (20), kor number (3, 0), ar number (10,0));

```
create table piros_auto(rsz varchar(20),
tipus varchar(20), szin varchar(20), kor
number, ar number);
```

```
insert into piros_auto values('ABC500',
'Opel Corsa', 'piros', 8, 720000);
```

```
insert into piros_auto values('MCC325',
'Opel Insignia', 'piros', 10, 26800000);
```

-----\n\n\n

3a) Töltse fel adatokkal a Piros_Auto adattáblát (*Piros_Auto_feltolt.sql*)

	RSZ	TIPUS	SZIN	KOR	AR
1	ABC500	Opel Corsa	piros	8	800000
2	MCC325	Opel Insignia	fekete	2	6800000
3	BBM104	Suzuki Swift	piros	5	1500000
4	CHR411	Renault Twingo	piros	12	700000
5	JRZ932	BMW M3	fekete	5	4500000
6	JEG113	Opel Corsa	piros	7	900000
7	DER842	Seat Ibiza	szürke	14	500000
8	BAB422	Lada 1300S	fehér	28	220000
9	UFF666	Audi TT	fekete	4	7500000

```
insert into piros_auto values('BBM104', 'Suzuki Swift','piros', 5, 150000);
```

```
insert into piros_auto values('CHR411', 'Renault Twingo', 'piros', 12, 630000);
```

```
insert into piros_auto values('JRZ932', 'BMW M3', 'fekete', 5,4500000);
```

```
insert into piros_auto values('DER842', 'Seat Ibiza', 'szürke', 14, 45000);
```

```
create table MasikPiros_Auto (rsz varchar2 (20), tipus varchar2 (20), szin varchar2 (20), kor number
(3,0), ar number (10,0));
```

3b) Készítsen egy *MasikPiros_Auto* táblát!

```
MasikPiros_Auto (rsz varchar2 (20), tipus varchar2 (20), szin, varchar2 (20), kor number (3,
0), ar number (10,0));
```

3c) Írjon egy PL/SQL blokkot a *piros autók lekérdezésére* és külön táblában (*MasikPiros_Auto*) való eltárolására (*Piros.Auto.sql*).

A kurzor neve: piros.

Futtassa le, majd frissítse a MasikPiros_Auto táblát – a táblában megjelenik a piros autók listája.

4. Írjon egy PL/SQL blokkot a piros autók lekérdezésére és külön táblában való eltárolására, de: FOR ciklussal! (*PirosAutoFor.sql*)

A szkript futtatása után a tábla frissítése!

5. Írjon egy PL/SQL tárolt eljárást, amely egy adott színű autó árát növeli meg adott százalékkal. (*ModCurAuto.sql*)

Eljárás neve: aut_arnov

Kurzor neve: cur_a

Változó neve: a

A szkript futtatása után a tábla frissítése: *ModCurAuto_hiv.sql*

```
begin
  open cur_a;
  loop
    fetch cur_a into a;
    exit when cur_a%notfound;
    update autok set ar=a.ar*(1+ert/100) where current of cur_a;
  end loop;
  close cur_a;
end;
```

```
begin

  aut_arnov('piros', 10);

end;
```

6. Írjon egy PL/SQL tárolt függvényt For ciklussal, amely az adott színű autó árát növeli meg adott százalékkal (*ModCurAutoFgv.sql*)

Függvény neve: aut_arnov2

Kurzor neve: cur_a

Változó neve: db

Futtatás után a tábla frissítése: *ModCurAutoFgv_fut.sql*

create or replace procedure aut_arnov2(szinbe in char, ert in int) is

```
    cursor cur_a is select * from autok where szin=szinbe
    for update of ar;
    db cur_a%rowtype;
begin
    open cur_a;
    for db in cur_a
    loop
        fetch cur_a into db;
        exit when cur_a%notfound;
        update autok set ar=db.ar*(1+ert/100) where current of cur_a;
    end loop;
    close cur_a;
end;
```

7. Írjon egy PL/SQL tárolt eljárást, amely egy adott színű autó árát növeli meg adott százalékkal és kiírja mennyi auto árát növelte meg.(ArvaltCurAuto.sql)

Eljárás neve: aut_arnov3

Kurzor neve: cur_a

Változó neve: db

A szkript futtatása után a tábla frissítése: *ArvaltCurAuto_hiv.sql*

create or replace procedure aut_arnov3(szinbe in char, ert in int) is

```
    cursor cur_b is select * from autok where szin=szinbe
    for update of ar;
    b cur_b%rowtype;
    darab number := 0;
begin
    if cur_b%isopen then
        close cur_b;
```



```
end if;
open cur_b;
loop
    fetch cur_b into b;
    exit when cur_b%notfound;
    update autok set ar=b.ar*(1+crt/100) where current of cur_b;
    darab:=darab+1;
end loop;
close cur_b;

dbms_output.put_line('Darabok: ' || darab);

end;
```