

Adatbázis rendszerek II.

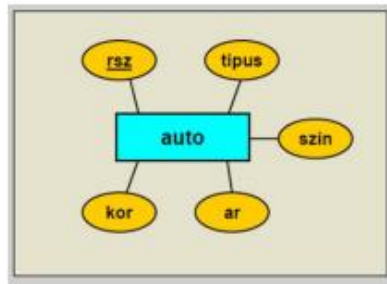
Gyakorlati feladatsor

Készítette:
Oravecz Áron
D3U3EE

Dátum:
2022 03.30

1. Az ER modell alapján készítse el a következő táblát: Mentés: **neptunkod_auto.sql**

Hozzon létre egy *auto* nevű táblát a *PLSQL* fejlesztővel.



```
create table auto1(  
    rsz int,  
    tipus VARCHAR2(20),  
    szín varchar2(20),  
    ar int,  
    kor int,  
    constraint rsz_pk primary key(rsz)  
);
```

```
1 create table auto1(  
2     rsz int,  
3     tipus VARCHAR2(20),  
4     szín varchar2(20),  
5     ar int,  
6     kor int,  
7     constraint rsz_pk primary key(rsz)  
8 );
```

Results Explain Describe Saved SQL History

Table created.

0.03 seconds

2. Írjon egy PL/SQL blokkot egy autó rekord felvitelére, ahol a mezőértékeket memóriaváltozókon keresztül adjuk meg, melyek típusa az AUTO tábla mezőinek típusával megegyező. **Mentés:** *autos_feltoltegy sor.sql*

auto				
rsz	tipus	szin	kor	ar

3. Töltse fel a táblát a minta alapján. **Mentés:** *autos_feltolt.sql*

RSZ	TIPUS	SZIN	KOR	AR
1 ABC500	Opel Corsa	piros	8	720000
2 MCC325	Opel Insignia	fekete	2	6800000
3 BBM104	Suzuki Swift	piros	5	1500000
4 CHR411	Renault Twingo	piros	12	630000
5 JRZ932	BMW M3	fekete	5	4500000
6 JEG113	Opel Corsa	piros	7	810000
7 DER842	Seat Ibiza	szürke	14	450000
8 BAB422	Lada 1300S	fehér	28	198000
9 UFF666	Audi TT	fekete	4	7500000
10 ABC124	Fiat	Piros	3	18000000

```
EXECUTE UDO('ABC500', 'Opel Corsa', 720000, 'piros', 8);
```

```
EXECUTE UDO('MCC325', 'Opel Insignia', 6800000,
'fekete', 2);
```

```
EXECUTE UDO('BBM104', 'Suzuki Swift', 150000, 'piros',
5);
```

```
EXECUTE UDO('CHR411', 'Renault Twingo', 630000, 'piros',
12);
```

```
EXECUTE UDO('JRZ932', 'BMW M3', 4500000, 'fekete', 5);
```

```
EXECUTE UDO('DER842', 'Seat Ibiza', 45000, 'szürke', 14);
```

4. Írjon egy PL/SQL blokk-t, mely egy memóriaváltozóban megadott kornál idősebb autók (7 év) árát csökkenti 10%-al. **Mentés:** *auto_ar_csokk_10.sql*

```
create view vkor as select rsz, kor from autok;
```

```
declare
```

```
    x number(2) := 7;
```

begin

update autok set ar=ar*0.9 where rsz in (select rsz from vkor where kor > x);

end;

5. Írjon egy PL/SQL blokk-t az autók átlagárának lekérdezésére és kiírására.

6. Írjon egy PL/SQL blokk-t, amely a memóriaváltozóban 10 évnél idősebb auto törlésére.

Mentés: *autos_kor_torles.sql*

declare

r autok.kor%type := 10;

begin

delete from autok where kor>r;

end;

7. Írjon egy PL/SQL blokk-t az autók darabszámának lekérdezésére és kiírására!

declare

atlag number;

BEGIN

select avg(ar) into atlag from autok;

DBMS_OUTPUT.put_line(atlag);

END;

8. Készítsen egy tárolt eljárás a 10 évnél idősebb autók törlésére!

declare

r autok.kor%type := 10;

begin

delete from autok where kor>r;

end;

Tárolt eljárás, függvény

1. Készítsen egy tárolt eljárást, amely paraméterként megadott kornál idősebb autók törlését végzi az Auto táblából. *elj_torol_parameter.sql*

Tárolt eljárás –hívás: *elj_torol_parameter_hiv.sql*

2. Készítsen egy **vevo** táblát (ugyfelszam, nev, e-mail) mezőkkel.

3. Készítsen egy Napló (ujx, sysdate, user) táblát, melybe eltároljuk a megnövelt ujk értéket
Mentés: *5-tel. naplo.sql*

a) Készítsen egy tárolt függvény -t, mellyel egy numerikus érték növelhető 5-el.
t_fgv_novel5.sql

b) Készítsen egy tárolt **függvény**-t a novel5 függvény meghívására, mellyel egy numerikus érték növelhető 5-el – (dbms_output.put_line kiírja).

c) Készítsen egy tárolt **függvény**-t a novel5 függvény meghívására, mellyel egy numerikus érték növelhető 5-el. Kiírás: Select novel5(13) from dual;

Mentés: 5-tel. novel5_torol.sql

4. Készítsen egy tárolt **eljárás**-t , mellyel egy változó (numerikus) érték növelhető 5-el, majd íratassa ki a konzolra. (*t_fgv_novelot.sql*)

5. Készítsen egy tárolt eljárás-t, mely a legmagasabb ár kiírását végzi el az autó táblából - dbms-re való kiírás. (*tar_elj_maxar_auto.sql*)

6. Készítsen egy tárolt függvény-t , mellyel a legmagasabb ár lekérésére az autó táblából.
(*tar_fgv_maxarleker_auto.sql*)

7. Készítse el a *Kategoria* táblát! (*kategoria create.sql*), majd töltse fel adatokkal (*Kategoria_feltolt.sql*)

Kategoria	
Kkod	Nev

Kkod	Nev
k01	Ka1a
k02	P1a
k03	Ruha
k04	Egyeb

8. Készítsen egy *UjKategoria* nevű tárolt eljárást, mely paraméterként megkapja egy új kategória adatait, és beszúrja a *Kategoria* táblába (k05 Édesség). (*elj_ujkategoria.sql*)

9. Készítsen egy *KategoriaTorol* nevű tárolt eljárást, mely kitöröl egy K05 kódú kategóriát. (*elj_KategoriaTorol.sql*)

10. Készítsen *KategoriaDarab* nevű tárolt függvényt, mely visszaadja a kategóriák darabszámát! (*elj_KategoriaDarab.sql*)

11. Készítsen egy *UjKategoria2* nevű tárolt eljárást, mely paraméterként megkapja egy új kategória adatait, és beszúrja a *Kategoria* táblába.

Ha már létező kategóriakódot kap paraméterként, írjon üzenetet a képernyőre (Van már ilyen kategória: Kkod), és ne hozza létre a kategóriát. (*elj_UjKategoria2.sql*)