

Adatbázisrendszerek I. BSc

5.gyak.

2021. 10. 20.

Készítette: László Andrea

Mérnökinformatikus

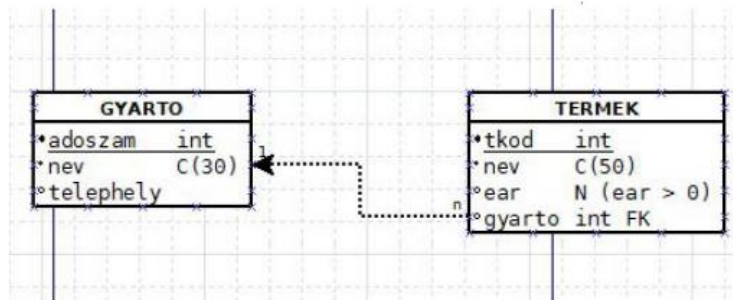
Neptunkód: DJ7PNE

Miskolc, 2021. 10. 20.

1 feladat

Hozza létre a táblákat SQL-ben az alábbi sémához.

Mentés: *neptunkod_6.1.sql*



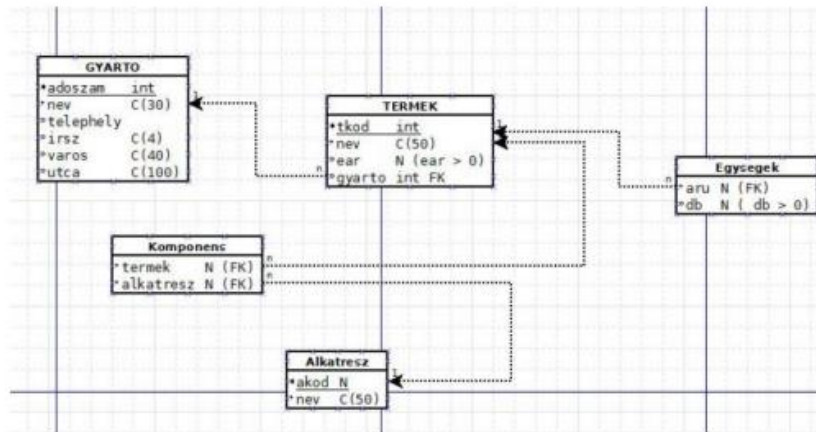
```
1 • CREATE DATABASE DJ7PNE;
2 • USE DJ7PNE;
3 • CREATE TABLE GYARTO(adoszam INT PRIMARY KEY, nev CHAR(30) NOT NULL, telephely CHAR(50));
4 • CREATE TABLE TERMEK(tkod INT PRIMARY KEY, nev CHAR(30) NOT NULL, ear INT CHECK(ear>0), gyarto INT REFERENCES GYARTO);
```

✓	1	13:03:32	CREATE DATABASE DJ7PNE	1 row(s) affected	0.015 sec
✓	2	13:03:32	USE DJ7PNE	0 row(s) affected	0.000 sec
✓	3	13:03:32	CREATE TABLE GYARTO(adoszam INT PRIMARY KEY, nev CHAR(30) NOT NULL, telephely CHAR(50))	0 row(s) affected	0.047 sec
✓	4	13:03:32	CREATE TABLE TERMEK(tkod INT PRIMARY KEY, nev CHAR(30) NOT NULL, ear INT CHECK(ear>0), gyarto I...	0 row(s) affected	0.063 sec

2. feladat

Hozza létre a táblákat SQL-ben az alábbi sémához.

Mentés: *neptunkod_6.2.sql*



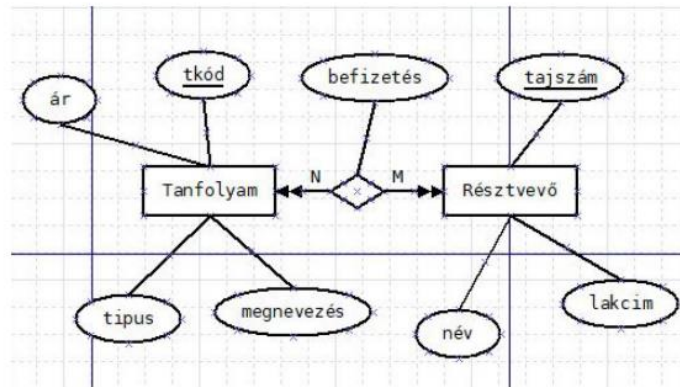
```
1 • CREATE DATABASE DJ7PNE;
2 • USE DJ7PNE;
3 • CREATE TABLE GYARTO(adoszam INT PRIMARY KEY, nev CHAR(30) NOT NULL, telephely CHAR(50));
4 • CREATE TABLE TERMEK(tkod INT PRIMARY KEY, nev CHAR(30) NOT NULL, ear INT CHECK(ear>0), gyarto INT REFERENCES GYARTO);
5 • CREATE TABLE EGYSEGEK(aru INT REFERENCES TERMEK, db INT CHECK(db>0));
6 • CREATE TABLE ALKATRESZ(akod INT PRIMARY KEY, nev CHAR(30) NOT NULL);
7 • CREATE TABLE KOMPONENS(termek INT REFERENCES TERMEK, alkatresz INT REFERENCES ALKATRESZ);
```

✓	1	13:04:35	CREATE DATABASE DJ7PNE	1 row(s) affected	0.016 sec
✓	2	13:04:35	USE DJ7PNE	0 row(s) affected	0.000 sec
✓	3	13:04:35	CREATE TABLE GYARTO(adoszam INT PRIMARY KEY, nev CHAR(30) NOT NULL, telephely CHAR(50))	0 row(s) affected	0.047 sec
✓	4	13:04:35	CREATE TABLE TERMEK(tkod INT PRIMARY KEY, nev CHAR(30) NOT NULL, ear INT CHECK(ear>0), gyarto I...	0 row(s) affected	0.046 sec
✓	5	13:04:35	CREATE TABLE EGYSEGEK(aru INT REFERENCES TERMEK, db INT CHECK(db>0))	0 row(s) affected	0.047 sec
✓	6	13:04:35	CREATE TABLE ALKATRESZ(akod INT PRIMARY KEY, nev CHAR(30) NOT NULL)	0 row(s) affected	0.032 sec
✓	7	13:04:35	CREATE TABLE KOMPONENS(termek INT REFERENCES TERMEK, alkatresz INT REFERENCES ALKATRESZ)	0 row(s) affected	0.046 sec

3. feladat

Hozza létre a táblákat SQL-ben az alábbi ER sémához úgy, hogy kiemeli a PK, FK megkötéseket.

Mentés: *neptunkod_6.3.sql*



```

1 • CREATE DATABASE DJ7PNE;
2 • USE DJ7PNE;
3 • CREATE TABLE TANFOLYAM(tkód INT, ár INT check(ÁR>0), típus CHAR(30), megnevezés CHAR(30), PRIMARY KEY(tkód));
4 • CREATE TABLE RESZTVEVO(tajsám INT, név CHAR(30) not null, lakcím CHAR(40), PRIMARY KEY(tajsám));
5 • CREATE TABLE BEFIZETES(tanfolyamok INT REFERENCES TANFOLYAM, résztvevők INT REFERENCES RESZTVEVO, befizetes INT);
    
```

1	13:05:24	CREATE DATABASE DJ7PNE	1 row(s) affected	0.015 sec
2	13:05:24	USE DJ7PNE	0 row(s) affected	0.000 sec
3	13:05:24	CREATE TABLE TANFOLYAM(tkód INT, ár INT check(ÁR>0), típus CHAR(30), megnevezés CHAR(30), PRIMARY KEY(tkód));	0 row(s) affected	0.047 sec
4	13:05:24	CREATE TABLE RESZTVEVO(tajsám INT, név CHAR(30) not null, lakcím CHAR(40), PRIMARY KEY(tajsám));	0 row(s) affected	0.047 sec
5	13:05:24	CREATE TABLE BEFIZETES(tanfolyamok INT REFERENCES TANFOLYAM, résztvevők INT REFERENCES RESZTVEVO, befizetes INT);	0 row(s) affected	0.047 sec

4. feladat

Adott az alábbi tábla:

```
CREATE TABLE termekek (tkod INT PRIMARY KEY, nev VARCHAR2(50) NOT NULL, ear INT CHECK (EAR > 0), kategoria CHAR(20));
```

Vigyen fel 5 rekordot a táblába.

tkod	nev	ear	kategoria
1	lapát	2000	K1
2	seprű	4000	K1
3	mosogató gél	1500	K2
4	szappan	1000	K2
5	pohár	2400	K3

Mentés: *neptunkod_6.4.sql*

```

1 • CREATE DATABASE DJ7PNE;
2 • USE DJ7PNE;
3 • CREATE TABLE termekek (tkod INT PRIMARY KEY, nev VARCHAR(50) NOT NULL, ear INT CHECK (EAR > 0), kategoria CHAR(20));
4 • INSERT INTO termekek VALUES(0, billentyűzet, 5000, elektronika);
5 • INSERT INTO termekek VALUES(1, ásó, 2000, kertészeti);
6 • INSERT INTO termekek VALUES(2, lámpa, 3000, elektronika);
7 • INSERT INTO termekek VALUES(3, monitor, 15000, elektronika);
8 • INSERT INTO termekek VALUES(4, szőnyeg, 10000, valami);
    
```

#	Time	Action	Message	Duration / Fetch
1	13:16:40	CREATE DATABASE DJ7PNE	1 row(s) affected	0.015 sec
2	13:16:40	USE DJ7PNE	0 row(s) affected	0.000 sec
3	13:16:40	CREATE TABLE termekek (tkod INT PRIMARY KEY, nev VARCHAR(50) NOT NULL, ear INT CHECK (EAR > 0), kategoria CHAR(20));	0 row(s) affected	0.031 sec
4	13:16:40	INSERT INTO termekek VALUES(0, 'billentyűzet', 5000, 'elektronika')	1 row(s) affected	0.016 sec
5	13:16:40	INSERT INTO termekek VALUES(1, 'ásó', 2000, 'kertészeti')	1 row(s) affected	0.000 sec
6	13:16:40	INSERT INTO termekek VALUES(2, 'lámpa', 3000, 'elektronika')	1 row(s) affected	0.015 sec
7	13:16:40	INSERT INTO termekek VALUES(3, 'monitor', 15000, 'elektronika')	1 row(s) affected	0.000 sec
8	13:16:40	INSERT INTO termekek VALUES(4, 'szőnyeg', 10000, 'valami')	1 row(s) affected	0.000 sec

5. feladat Adjon meg az előző táblához olyan felviteli parancsokat, melyet nem fog végrehajtani az adatbázismotor.
INSERT INTO termek VALUES (1, valami, 2000, 'K1'); // létező kulcs
INSERT INTO termek VALUES (8, NULL, 4000, 'K1'); // üres név;
INSERT INTO termek VALUES (15, 'pohár A', 0, 'K3'); // hibás ár

6. feladat Adottak az alábbi tábla: KÖNYV (isbn C(20) PK, cím C(40), tárgy C(30), ár INT) Adja meg az alábbi műveletek relációs algebrai alakját.

- könyvek címe: Π cím (könyv)
- a 2000-nél drágább könyvek adatai: σ ár > 2000 (könyv)
- az 1000-nél olcsóbb könyvek címei: Π cím (σ ár < 1000 (könyv))
- a tárgyak nevei: Π tárgy (könyv)
- az 'AB' tárgyú könyvek címe és ára: Π cím, ár (σ tárgy = 'AB' (könyv))

7. feladat Adottak az alábbi tábla: KÖNYV (isbn C(20) PK, cím C(40), tárgy C(30), ár INT) Adja meg az alábbi műveletek relációs algebrai alakját.

- a könyvek darabszáma: Γ count(*) (könyv)
- a könyvek átlagára: Γ avg(ár) (könyv)
- a legolcsóbb könyv ára: Γ min(ár) (könyv)
- az 'AB' kategóriájú könyvek darabszáma: Γ count(*) (σ tárgy = 'AB' (könyv))
- a legdrágább AB kategóriájú könyv ára: Γ max(ár) (σ tárgy = 'AB' (könyv))
- az átlagárnál drágább könyvek címei: Π cím (σ (ár > (Γ {avg(ár)} (könyv))) (könyv))
- az átlagárnál drágább könyvek darabszáma: Γ count(*) (σ (ár > (Γ {avg(ár)} (könyv))) (könyv))