

Adatbázisrendszerek I. BSc

8.gyak.

2021. 11. 17.

**Készítette: Garamszegi
Márton**

Mérnökinformatikus

Neptunkód: AJYKQ3

Miskolc, 2021. 11. 17.

1. feladat

```
CREATE DATABASE AJYKQ3;
USE AJYKQ3;
CREATE TABLE GYARTO(adoszam INT PRIMARY KEY, nev CHAR(30) NOT NULL, iranyitoszam CHAR(4), varos char(40), utac char(100));
CREATE TABLE EGYSEGEK(aru INT REFERENCES TERMEK, db INT CHECK(db>0));
CREATE TABLE ALKATRESZ(akod INT PRIMARY KEY, nev CHAR(30) NOT NULL);
CREATE TABLE KOMPONENS(termek INT REFERENCES TERMEK, alkatresz INT REFERENCES ALKATRESZ);
CREATE TABLE TERMEK(tkod INT PRIMARY KEY, nev CHAR(30) NOT NULL, ear INT CHECK(ear>0), gyarto INT REFERENCES GYARTO);

INSERT INTO GYARTO VALUES(11111111, "mediamarkt", 4090, NULL, NULL);
INSERT INTO GYARTO VALUES(33333333, "lidl", 6024, NULL, NULL);
INSERT INTO GYARTO VALUES(44444444, "penny", 1046, NULL, NULL);
INSERT INTO GYARTO VALUES(22222222, "obi", 3021, "Hajdúszoboszló", "Petőfi utca");

INSERT INTO TERMEK VALUES(2, "lámpa", 3000, 11111111);
INSERT INTO TERMEK VALUES(3, "monitor", 15000, 11111111);
INSERT INTO TERMEK VALUES(0, "billentyűzet", 5000, 11111111);
INSERT INTO TERMEK VALUES(1, "ásó", 2000, 22222222);
INSERT INTO TERMEK VALUES(4, "szőnyeg", 10000, 22222222);

INSERT INTO EGYSEGEK VALUES(0, 10);
INSERT INTO EGYSEGEK VALUES(1, 18);
INSERT INTO EGYSEGEK VALUES(4, 40);
INSERT INTO EGYSEGEK VALUES(2, 34);
INSERT INTO EGYSEGEK VALUES(3, 100);

INSERT INTO ALKATRESZ VALUES(03, "BILLENTYŰ");
INSERT INTO ALKATRESZ VALUES(04, "FESTÉK");
INSERT INTO ALKATRESZ VALUES(00, "FOGASKERÉK");
INSERT INTO ALKATRESZ VALUES(01, "BOT");
INSERT INTO ALKATRESZ VALUES(02, "VAS");

INSERT INTO KOMPONENS VALUES(0, 00);
INSERT INTO KOMPONENS VALUES(0, 03);
INSERT INTO KOMPONENS VALUES(3, 03);
INSERT INTO KOMPONENS VALUES(1, 02);
INSERT INTO KOMPONENS VALUES(3, 00);
INSERT INTO KOMPONENS VALUES(2, 04);
INSERT INTO KOMPONENS VALUES(1, 01);
INSERT INTO KOMPONENS VALUES(4, 04);

ALTER TABLE ALKATRESZ ADD(ar INT, CHECK(ar>0));
SELECT * FROM ALKATRESZ;
SELECT nev FROM gyarto WHERE adoszam NOT IN (SELECT gyarto FROM termek);
UPDATE termek SET ear = 1.1*ear WHERE tkod IN (SELECT termek FROM komponens WHERE alkatresz = (SELECT akod FROM alkatresz WHERE nev = "X"));
CREATE VIEW et AS SELECT MAX(t.nev) nev, t.tkod, COUNT(*) db FROM termek t INNER JOIN egysegek e ON e.aru = t.tkod GROUP BY t.tkod;
SELECT nev FROM et WHERE db = (SELECT MAX(db) FROM et);
```

2. feladat

```
CREATE DATABASE AJYKQ3;
USE AJYKQ3;
CREATE TABLE RESZTVEVO(tajszám INT, név CHAR(30) not null, lakcím CHAR(40), PRIMARY KEY(tajszám));
CREATE TABLE BEFIZETES(tanfolyamok INT REFERENCES TANFOLYAM, résztvevők INT REFERENCES RESZTVEVO, befizetes INT);
CREATE TABLE TANFOLYAM(tkód INT, ár INT check(ár>0), típus CHAR(30), megnevezés CHAR(30), PRIMARY KEY(tkód));

insert into tanfolyam values(0,120000,"egyetem","analízis");
insert into tanfolyam values(1,50000,"középiskola","biológia");
insert into tanfolyam values(2,25000,"általános","történelem");
insert into tanfolyam values(3,150000,"egyetem","adatbázisrendszerek");

insert into résztvevo values(111222333, "Kiss Pista", "Budapest");
insert into résztvevo values(000111222, "Tóth Tibor", "Miskolc");
insert into résztvevo values(010102020, "Balogh Rebeka", NULL);
insert into résztvevo values(222333444, "Nagy Gergő", NULL);

insert into befizetes values(0,222333444,10000);
insert into befizetes values(3,000111222,15000);
insert into befizetes values(0,111222333,10000);
insert into befizetes values(2,010102020,2500);
insert into befizetes values(1,111222333,2500);

SELECT típus, AVG(ár), MAX(ár) - MIN(ár) FROM tanfolyam GROUP BY típus;
SELECT megnevezés, CASE WHEN ár > 100000 THEN "drága" ELSE "olcsó" END arkatgoria FROM tanfolyam;
CREATE VIEW tdb AS SELECT MAX(t.megnevezés) név, COUNT(*) db FROM tanfolyam t INNER JOIN befizetes b ON b.tanfolyamok = t.tkód GROUP BY t.tkód;
SELECT név, CASE WHEN db > 10 THEN "népszerű" ELSE "nem népszerű" END résztvevők FROM tdb;
```

3. feladat

ALTER TABLE termek DROP COLUMN gyarto; CREATE TABLE gyartja (termek INT REFERENCES termek, gyarto INT REFERENCES GYARTO); 4. feladat ALTER TABLE termek ADD (kategoria CHAR(50)); CREATE VIEW kgy AS SELECT g.adoszam, t.kategoria FROM gyarto g LEFT OUTER JOIN termek t ON g.adoszam = t.gyarto GROUP BY g.adoszam, t.kategoria; CREATE VIEW kgy2 AS SELECT kategoria, COUNT(adoszam) db FROM kgy GROUP BY kategoria; SELECT kategoria FROM kgy2 WHERE db = (SELECT COUNT(*) FROM gyarto);

4. feladat

```
CREATE DATABASE AJYKQ3;
USE AJYKQ3;
CREATE TABLE GYARTO(adoszam INT PRIMARY KEY, nev CHAR(30) NOT NULL, telephely CHAR(50));
CREATE TABLE TERMEK(tkod INT PRIMARY KEY, nev CHAR(30) NOT NULL, ear INT CHECK(ear>0), gyarto INT REFERENCES GYARTO);

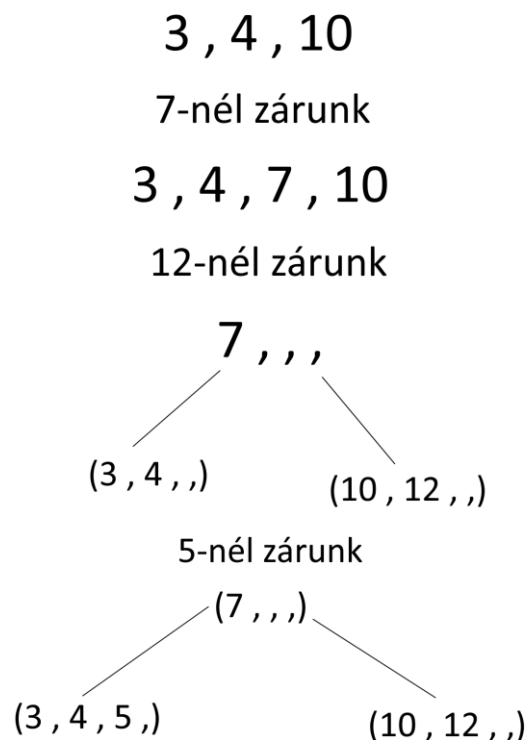
INSERT INTO GYARTO VALUES(11111111, "mediamarkt", 4090);
INSERT INTO GYARTO VALUES(33333333, "lidl", 6024);
INSERT INTO GYARTO VALUES(44444444, "penny", 1046);
INSERT INTO GYARTO VALUES(22222222, "obi", 3021);

ALTER TABLE termek ADD (kategoria CHAR(50));

INSERT INTO TERMEK VALUES(0, "billentyűzet", 5000, 11111111, "ELEKTRONIKA");
INSERT INTO TERMEK VALUES(5, "ELEM", 300, 44444444, "ELEKTRONIKA");
INSERT INTO TERMEK VALUES(2, "lámpa", 3000, 22222222, "ELEKTRONIKA");
INSERT INTO TERMEK VALUES(1, "ásó", 2000, 22222222, "MINDENNAPOK");
INSERT INTO TERMEK VALUES(3, "monitor", 15000, 33333333, "ELEKTRONIKA");
INSERT INTO TERMEK VALUES(4, "szőnyeg", 10000, 22222222, "MINDENNAPOK");

CREATE VIEW kgy AS SELECT g.adoszam, t.kategoria FROM gyarto g LEFT OUTER JOIN termek t ON g.adoszam = t.gyarto GROUP BY g.adoszam, t.kategoria;
CREATE VIEW kgy2 AS SELECT kategoria, COUNT(adoszam) db FROM kgy GROUP BY kategoria;
SELECT kategoria FROM kgy2 WHERE db = (SELECT COUNT(*) FROM gyarto);
```

5. feladat



19 elemet kell hozzáadni a fa magasságának növeléséhez.