**Jegyzőkönyv**

**Adatbázis rendszerek I.**

**Beadandó**

**Készítette: Uri-Kovács Bence**

**Neptunkód: O3AE4K**

**Kelt.: 2020.11.24.**

**A feladat leírása:**

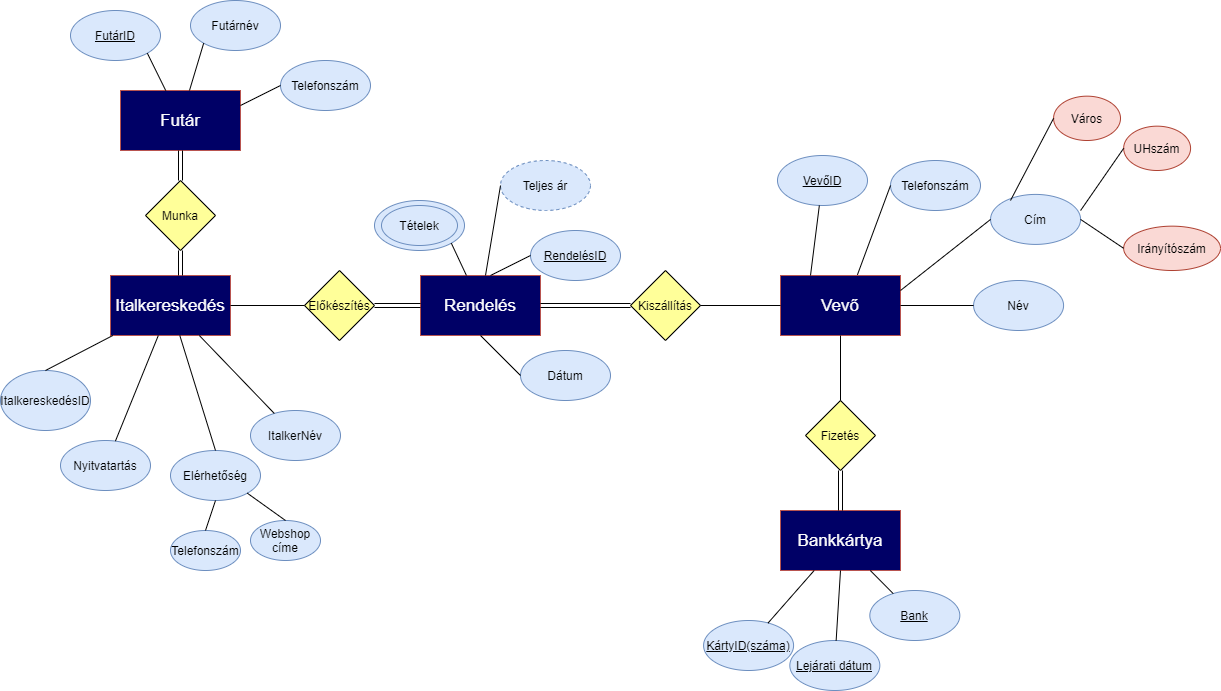
A beadandó témája egy olyan adatbázis, amely több italkereskedést kezel. A rendszer leírja a vevő adatait, az általa megrendelt termékeket és a kiszállítót is.

**Az ER modell egyedei és tuladjonságai:**

* **A Futár egyed tulajdonságai**
  + FutárID: A Futár egyed azonosítója. Elsődleges kulcs.
  + Telefonszám: A futár telefonszáma.
  + Név: A futár neve.
* **A Italkereskedés egyed tulajdonságai**
  + ItalkereskedésID: Az italkereskedés egyed elsődleges kulcsa, azonosítója.
  + Nyitvatartás: A italkereskedés nyitva tartási ideje.
  + Név: A italkereskedés neve.
  + Elérhetőség: Összetett tulajdonság. Az elérhetőség közé tartozik egy telefonszám, illetve az italkereskedés webshopjának címe.
* **A Bankkártya egyed tulajdonságai**
  + Kártyaszám/KártyID: A Bankkártya egyed elsődleges kulcsa.
  + Bank: A bank neve, amelyhez a bankkártya tartozik.
  + Lejárati dátum: A kártya lejárati dátuma.
* **A Vevő egyed tulajdonságai**
  + VevőID: A Vevő egyed elsődleges kulcsa.
  + Név: A vevő neve.
  + Telefonszám: A vevő telefonszáma.
  + Cím: Összetett tulajdonság. A vevő címe (irányítószám, utcs-házszám, és város).
* **A Rendelés egyed tulajdonságai**
  + RendelésID: A rendelés egyedet azonosítja, elsődleges kulcs.
  + Teljes ár: A rendelt tételek teljes ára. Származtatott tulajdonság.
  + Dátum: A rendelés dátuma
  + Tételek: Többszörös tulajdonság. A rendelésben szereplő tételek.

**Az ER modell egyedei és tuladjonságai:**

* **Futár és Italkereskedés**
  + Ebben a kapcsolatban több-több kapcsolatról beszélünk, hiszen egy Italbolt alkalmazhat több futárt, és feltételezzük, hogy a futár munka az diákmunka, így egy futár több helyen betöltheti ezt a pozíciót, és végezhet szállító munkát.
* **Italkereskedés és Rendelés**
  + Az egyedek között fennálló kapcsolat az előkészítés, mikor az alkalmazottak összeszedik, milyen tételekből tevődik össze egy-egy megrendlés. Itt is egy-több kapcsolat áll fennt, hiszen egy rendelés egyszerre csak egy kereskedéshez kapcsolódhat, de egy kereskedésnek lehet több megrendelése.
* **Rendelés és Vevő**
  + A két egyed között a kapcsolatot a kiszállítás teremti meg. Egy vevőnek lehet több megrendelése, míg egy megrendelés csak egy vevőhöz kapcsolódhat, így a kapcsolat 1:N típusú.
* **Vevő és Bankkártya**
  + Legjobban a birtoklás írja le a kapcsolatot, melynél ugyancsak 1:N kapcsolatot realizálhatunk, hiszen ugyanaz a kártya nem tartozhat több emberhez, de egy embernek lehet több bankkártyája is. Ezt az életben látott példák is igazolják.



**Az adatbázis konvertálása relációs modellre:**

A ***Vevő*** egyedből egy tábla lesz. A tulajdonságaiból és a *Cím* komponenseiből lesznek a mezők. A VevőID lesz az elsődleges kulcsa.

A ***Futár*** egyedből egy tábla lesz. A tulajdonságaiból lesznek a mezők. A FutárID lesz az elsődleges kulcsa, valamint lesz egy idegen kulcsa, a IKID, amely az ***Italkereskedés*** elsődleges kulcsához kapcsolódik.

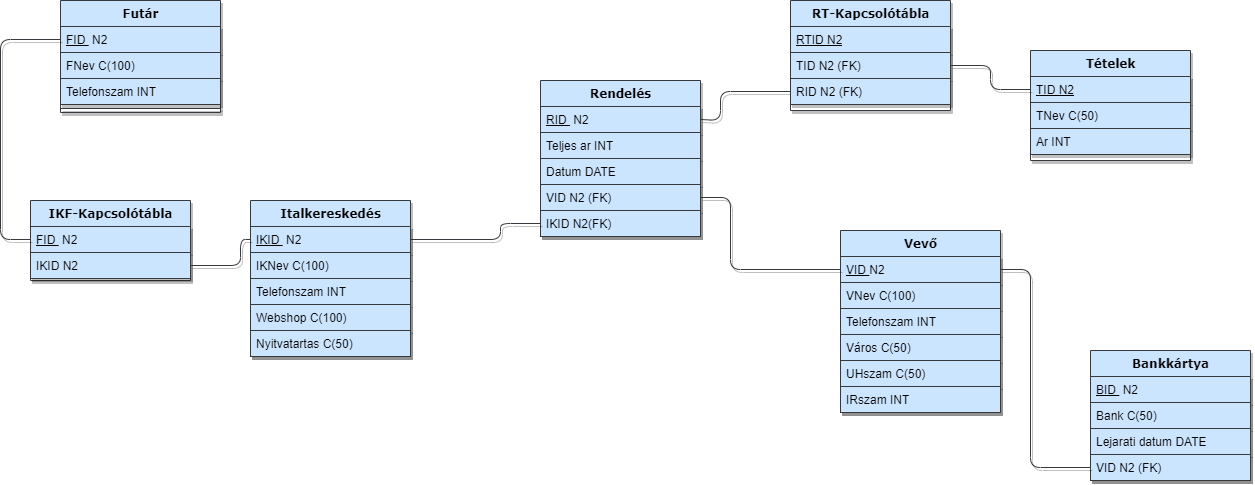
A ***Italkereskedés*** egyedből egy tábla lesz. A tulajdonságaiból és az *Elérhetőség* komponenseiből lesznek a mezők. A IKID lesz az elsődleges kulcsa.

A ***Rendelés*** egyed is egy táblát fog képezni, melynek elsődleges kulcsa a RendelésID lesz. Mezői a fent említett tulajdonságok. Ez lesz a kapcsoló tábla a Vevő és a kereskedés között.

Egy rendeléshez több tétel tartozik alapesetben. Ehhez is szükséges egy kapcsolótábla mivel ugyanaz a termék megnevezés több rendeléshez is kapcsolódhat.

A Munka kapcsolatból a ***Italkereskedés*** és a ***Futár*** között egy kapcsolótábla lesz, hiszen a N:M típusú. Ebben a az esetben a IKID az üzeltet, míg a FID a futárt azonosítja.

A ***Bankkártya*** is egy táblát fog jelenteni, ahol a kártya száma lesz az ID és a többi tulajdonsága is mezőket jelent majd.



**Táblák létrehozása:**

Elsőként az idegen kulcs nélküli táblákat érdemes létrehozni, ezt követően kell azokat, melyekben idegenkulcs található. Az idegenkulcsot tartalmazó mezők típusának meg kell egyeznie a másik táblában lévő mező típusával, amire mutat.

Fontos megemlíteni, hogy a telefonszám 7 karaktere van csak letárolva, hiszen bármily

CREATE TABLE VEVO (

VID NUMBER (2) PRIMARY KEY,

VNEV CHAR(100),

TELEFONSZAM NUMBER(11),

VAROS CHAR(50),

UHSZAM CHAR(50),

IRSZAM NUMBER(4)

);

CREATE TABLE FUTAR (

FID NUMBER (2) PRIMARY KEY,

FNEV CHAR(100),

TELEFONSZAM NUMBER(11)

);

CREATE TABLE TETEL (

TID NUMBER (2) PRIMARY KEY,

TNEV CHAR(50),

AR NUMBER(4)

);

CREATE TABLE ITALKERESKEDES (

IKID NUMBER (2) PRIMARY KEY,

IKNEV CHAR(100),

TELEFONSZAM NUMBER(11),

WEBSHOP CHAR(100),

NYITVATARTAS CHAR(50)

);

CREATE TABLE IKF\_KAPCSOLO (

FID NUMBER (2),

IKID NUMBER (2),

FOREIGN KEY (FID) REFERENCES FUTAR (FID),

FOREIGN KEY (IKID) REFERENCES ITALKERESKEDES (IKID)

);

CREATE TABLE RENDELES (

RID NUMBER (2) PRIMARY KEY,

TELJESAR NUMBER (10),

DATUM CHAR(20),

VID NUMBER(2),

IKID NUMBER(2),

FOREIGN KEY (VID) REFERENCES VEVO (VID),

FOREIGN KEY (IKID) REFERENCES ITALKERESKEDES (IKID)

);



CREATE TABLE RT\_KAPCSOLO (

RTID NUMBER (2) PRIMARY KEY,

TID NUMBER (2),

RID NUMBER(2),

FOREIGN KEY (RID) REFERENCES RENDELES (RID),

FOREIGN KEY (TID) REFERENCES TETEL (TID)

);

CREATE TABLE BANKKARTYA (

BID NUMBER (2) PRIMARY KEY,

BANK CHAR (50),

LEJARAT CHAR(20),

VID NUMBER(2),

FOREIGN KEY (VID) REFERENCES VEVO (VID)

);

**Táblák feltöltése:**

BEGIN

INSERT INTO vevo VALUES (2, 'Tóth Zoltán', 06701234567, 'Miskolc', 'Zoltán u. 16.', 3525 );

INSERT INTO vevo VALUES (3, 'Tóth Pista', 06305678976, 'Miskolc', 'József Attila u. 9.', 3525 );

INSERT INTO vevo VALUES (4, 'Kacsó Attila', 06701234433, 'Budapest', 'Felszabadítók u. 10.', 1019 );

INSERT INTO vevo VALUES (5, 'Veres Zsombor', 06701264563, 'Szirma', 'Berekkert u. 6.', 3525 );

INSERT INTO vevo VALUES (6, 'Halász András', 06701238567, 'Győr', 'Petőfi Sándor u. 16.', 9012 );

INSERT INTO vevo VALUES (7, 'Kertész Attila', 06701234999, 'Budapest', 'Ilona u. 10.', 1012);

end



BEGIN

INSERT INTO futar VALUES (1, 'Kiss Zoltán', 06701234567 );

INSERT INTO futar VALUES (2, 'Tóth Zsófia', 06203334567 );

INSERT INTO futar VALUES (3, 'Kiss Dorka', 06701637577 );

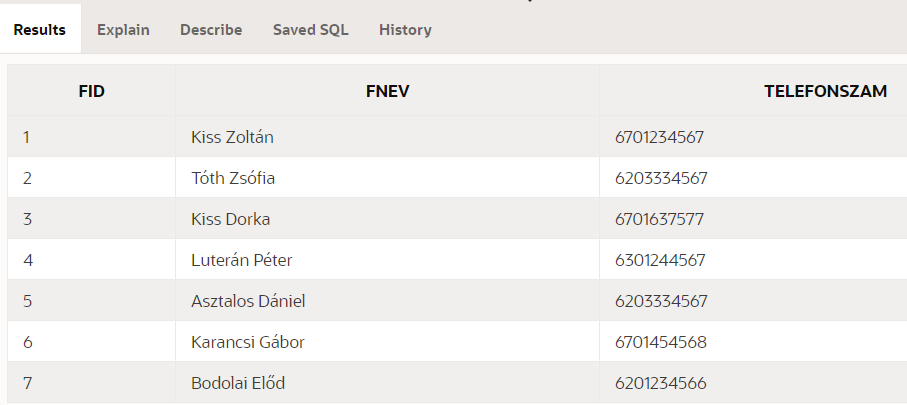
INSERT INTO futar VALUES (4, 'Luterán Péter', 06301244567 );

INSERT INTO futar VALUES (5, 'Asztalos Dániel', 06203334567 );

INSERT INTO futar VALUES (6, 'Karancsi Gábor', 06701454568 );

INSERT INTO futar VALUES (7, 'Bodolai Előd', 06201234566 );

end



BEGIN

INSERT INTO tetel VALUES (1, 'Bor', 350 );

INSERT INTO tetel VALUES (2, 'Ballantines', 4450 );

INSERT INTO tetel VALUES (3, 'Mátyás keserű', 230 );

INSERT INTO tetel VALUES (4, 'Kőbányai', 180 );

INSERT INTO tetel VALUES (5, 'Soproni', 300 );

INSERT INTO tetel VALUES (6, 'Soproni meggy ale', 330 );

INSERT INTO tetel VALUES (7, 'Tátra tea', 7800 );

INSERT INTO tetel VALUES (8, 'Coca-Cola', 310 );

INSERT INTO tetel VALUES (9, 'Pepsi', 350 );

INSERT INTO tetel VALUES (10, 'Szobi Almalé', 260 );

INSERT INTO tetel VALUES (11, 'Szobi Baracklé', 260 );

INSERT INTO tetel VALUES (12, 'Szobi Baracklé', 260 );

INSERT INTO tetel VALUES (13, 'Szobi Szőlőlé', 260 );

INSERT INTO tetel VALUES (14, 'Muscat bor', 860 );

INSERT INTO tetel VALUES (15, 'Szentkirályi Ásványvíz', 170 );

INSERT INTO tetel VALUES (16, 'Fuse tea Citrom', 360 );

END

BEGIN

INSERT INTO ITALKERESKEDES VALUES (1, 'Italdiszkont', 06701234567, 'https://neptun32.uni-miskolc.hu', '8-18');

INSERT INTO ITALKERESKEDES VALUES (2, 'Italdiszkont\_Barcika', 06701233367, 'https://neptun32.uni-miskolc.hu', '8-18');

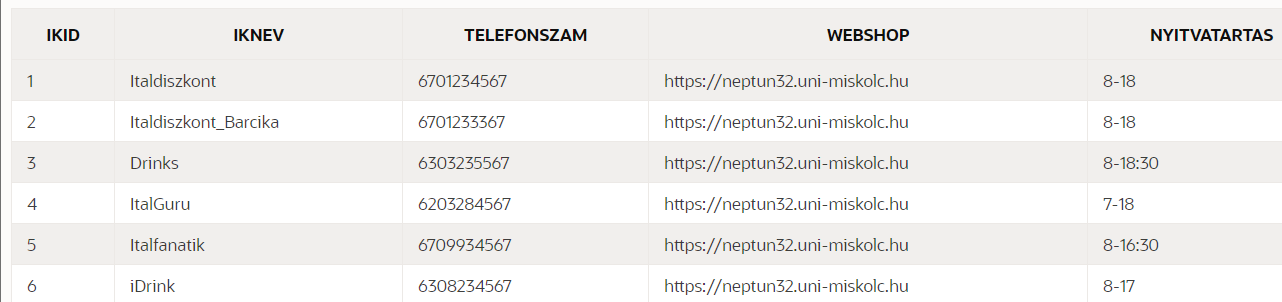
INSERT INTO ITALKERESKEDES VALUES (3, 'Drinks', 06303235567, 'https://neptun32.uni-miskolc.hu', '8-18:30');

INSERT INTO ITALKERESKEDES VALUES (4, 'ItalGuru', 06203284567, 'https://neptun32.uni-miskolc.hu', '7-18');

INSERT INTO ITALKERESKEDES VALUES (5, 'Italfanatik', 06709934567, 'https://neptun32.uni-miskolc.hu', '8-16:30');

INSERT INTO ITALKERESKEDES VALUES (6, 'iDrink', 06308234567, 'https://neptun32.uni-miskolc.hu', '8-17');

END



A tesztadatok fikcionalitását megtartva a webshop adatként a Neptun elérhetőségét adtam meg. Erre a célra bármi más is megfelelt volna, a logikán nem változtat.

BEGIN

INSERT INTO IKF\_KAPCSOLO  VALUES (1, 1);

INSERT INTO IKF\_KAPCSOLO  VALUES (1, 6);

INSERT INTO IKF\_KAPCSOLO  VALUES (1, 3);

INSERT INTO IKF\_KAPCSOLO  VALUES (2, 1);

INSERT INTO IKF\_KAPCSOLO  VALUES (2, 2);

INSERT INTO IKF\_KAPCSOLO  VALUES (3, 4);

INSERT INTO IKF\_KAPCSOLO  VALUES (4, 5);

INSERT INTO IKF\_KAPCSOLO  VALUES (5, 1);

INSERT INTO IKF\_KAPCSOLO  VALUES (6, 2);

INSERT INTO IKF\_KAPCSOLO  VALUES (7, 6);

END

BEGIN

INSERT INTO bankkartya  VALUES (1, 'K&H', '09/21', 1);

INSERT INTO bankkartya  VALUES (2, 'K&H', '10/26', 2);

INSERT INTO bankkartya  VALUES (3, 'K&H', '02/22',3);

INSERT INTO bankkartya  VALUES (4, 'K&H', '08/24', 4);

INSERT INTO bankkartya  VALUES (5, 'K&H', '01/23', 5);

INSERT INTO bankkartya  VALUES (6, 'K&H', '09/23', 5);

INSERT INTO bankkartya  VALUES (7, 'K&H', '03/22', 6);

INSERT INTO bankkartya  VALUES (8, 'K&H', '11/21', 7);

END



BEGIN

INSERT INTO RENDELES  VALUES ( 1, 2000, '2020.10.23.', 2, 1);

INSERT INTO RENDELES  VALUES ( 2, 20000, '2020.10.23.', 1, 1);

INSERT INTO RENDELES  VALUES ( 3, 3400, '2020.10.23.', 5, 2);

INSERT INTO RENDELES  VALUES ( 4, 6000, '2020.10.23.', 3, 1);

INSERT INTO RENDELES  VALUES ( 5, 12000, '2020.10.23.', 4, 6);

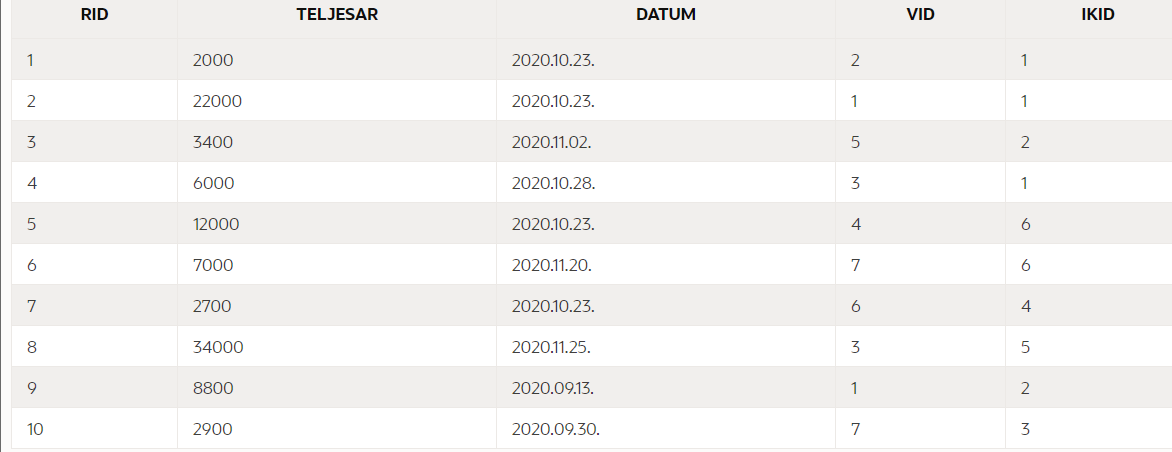
INSERT INTO RENDELES  VALUES ( 6, 7000, '2020.10.23.', 7, 6);

INSERT INTO RENDELES  VALUES ( 7, 2700, '2020.10.23.', 6, 4);

INSERT INTO RENDELES  VALUES ( 8, 34000, '2020.10.23.', 3, 5);

INSERT INTO RENDELES  VALUES ( 9, 8800, '2020.10.23.', 1, 2);

INSERT INTO RENDELES  VALUES ( 10, 2900, '2020.10.23.', 7, 3);

END

BEGIN

INSERT INTO rt\_kapcsolo  VALUES (1, 3, 1);

INSERT INTO rt\_kapcsolo  VALUES ( 2, 4,1);

INSERT INTO rt\_kapcsolo  VALUES (3,  2,1);

INSERT INTO rt\_kapcsolo  VALUES (4, 9,1);

INSERT INTO rt\_kapcsolo  VALUES (5,  10,1);

INSERT INTO rt\_kapcsolo  VALUES (6, 16,1);

INSERT INTO rt\_kapcsolo  VALUES (7, 2, 2);

INSERT INTO rt\_kapcsolo  VALUES (8, 4,2);

INSERT INTO rt\_kapcsolo  VALUES ( 9,11,2);

INSERT INTO rt\_kapcsolo  VALUES ( 10,12,3);

INSERT INTO rt\_kapcsolo  VALUES ( 11, 15,4);

INSERT INTO rt\_kapcsolo  VALUES ( 12,1,4);

INSERT INTO rt\_kapcsolo  VALUES ( 13, 2,4);

INSERT INTO rt\_kapcsolo  VALUES ( 14, 4,5);

INSERT INTO rt\_kapcsolo  VALUES ( 15,5, 5);

INSERT INTO rt\_kapcsolo  VALUES ( 16,5, 5);

INSERT INTO rt\_kapcsolo  VALUES ( 17, 2, 6);

INSERT INTO rt\_kapcsolo  VALUES ( 18,4,7);

INSERT INTO rt\_kapcsolo  VALUES ( 19,5,7);

INSERT INTO rt\_kapcsolo  VALUES ( 20,9,8);

INSERT INTO rt\_kapcsolo  VALUES ( 21,8, 8);

INSERT INTO rt\_kapcsolo  VALUES ( 22,2,8);

INSERT INTO rt\_kapcsolo  VALUES ( 23,3,9);

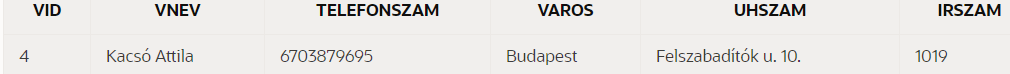
INSERT INTO rt\_kapcsolo  VALUES ( 24,  3,10);

END

**SQL parancsok:**

* Módosítások:

UPDATE VEVO SET TELEFONSZAM='06703879695' WHERE

VID = 4;

UPDATE TETEL SET AR=360 WHERE

TNEV='Bor';

UPDATE ITALKERESKEDES SET WEBSHOP='https://users.iit.uni-miskolc.hu/' WHERE

IKNEV='ItalGuru';

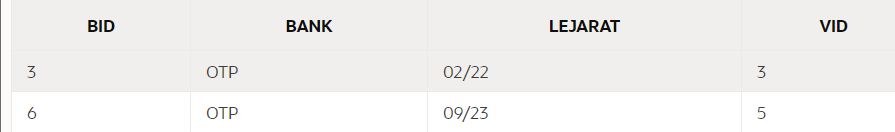
UPDATE BANKKARTYA SET BANK='OTP' WHERE BID=3;

UPDATE BANKKARTYA SET BANK='OTP' WHERE BID=6;

UPDATE RENDELES SET TELJESAR=22000 WHERE RID=2;

* Lekérdezések:

1. Listázzuk ki az OTP Bankhoz tartozó kártyákat

select \* from BANKKARTYA WHERE BANK ='OTP';

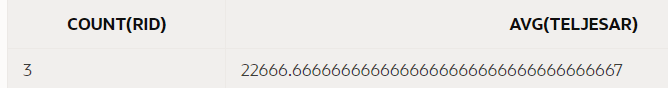
π BID, BANK, LEJARAT, VID[σ BANK=’OTP’ (BANKKÁRTYA)]

1. Listázzuk ki az 10000 Ft-nál nagyobb rendelések dátumát és összegét

SELECT RID, TELJESAR, DATUM FROM RENDELES WHERE TELJESAR > 10000;

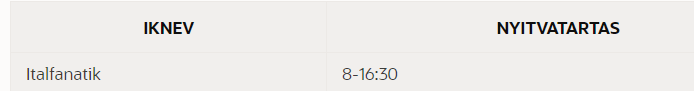
π RID, TELJESAR, DATUM[σ TELJESAR > 10000’ (RENDELES)]

1. Számolja meg, hány rendelés került az átlagnál többe.

SELECT count(rid),AVG(TELJESAR) FROM RENDELES WHERE TELJESAR>(SELECT avg(TELJESAR) FROM RENDELES);

πCOUNT(RID),AVG(TELJESAR) [σ TELJESAR > (π AVG(TELJESAR) (RENDELES))(RENDELES)]

1. Listázza ki azt/azokat az üzleteket, ahol a nyitva tartás 8-16:30-ig tart.

select IKNEV from ITALKERESKEDES WHERE NYITVATARTAS ='8-16:30';

πIKNEV, NYITVATARTAS [σ NYITVATARTAS = ’8-16:30’(ITALKERESKEDES)]

1. Listázza ki, hogy a vevőknek hány kártyája van regisztrálva.

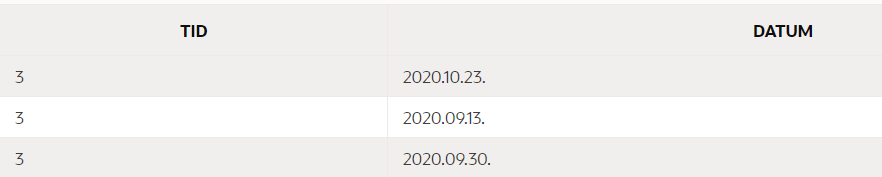
SELECT vnev,count(bid) FROM BANKKARTYA INNER JOIN VEVO ON BANKKARTYA.VID = VEVO.VID GROUP BY VNEV;

// ALIASOK: BANKKARTYA – BK

VEVO - V

πVNEV, COUNT(BID)Γ count(BID),VNEVVNEV (BANKKÁRTYA⨝BK.VID=V.VID VEVO)

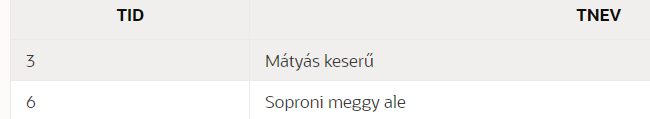
1. Listázza ki mikor rendeltek Mátyás keserűt

SELECT TID,DATUM FROM RENDELES INNER JOIN RT\_KAPCSOLO ON RENDELES.RID = RT\_KAPCSOLO.RID WHERE TID='3' ;

πTID, DATUM [σ TID = 3(RENDELES ⨝RENDELES.RID = RT\_KAPCSOLO.RID RT\_KAPCSOLO)]

1. Listázza ki azokat a tételeket, amiket 2-nél többször rendeltek

sELECT TID,TNEV,RID FROM TETEL INNER JOIN RT\_KAPCSOLO ON TETEL.TID = RT\_KAPCSOLO.TID WHERE (SELECT COUNT(RID),TID FROM RT\_KAPCSOLO GROUP  BY TID)>2 ;



//ALIASOK: RT\_KAPCSOLO – RT\_K

πTID, TNEV, RID [σ ((Γ count(RID),TIDTID RT\_KAPCSOLO) > 2)TETEL⨝TETEL.TID=RT\_K.TID RT\_KAPCSOLO)]

* Rekordok törlése:

DELETE FROM futar WHERE fid=4;

DELETE FROM BANKKARTYA WHERE lejarat='09/21';

DELETE FROM tetel WHERE tnev='Szobi Almalé';

* Táblák törlése:

DROP TABLE BANKKARTYA;

DROP TABLE RT\_KAPCSOLO;

DROP TABLE RENDELES;

DROP TABLE IKF\_KAPCSOLO;

DROP TABLE ITALKERESKEDES;

DROP TABLE TETEL;

DROP TABLE FUTAR;

DROP TABLE VEVO;