

Jegyzőkönyv
Adatbázis rendszerek I.
Féléves feladat

Készítette: Simon Dániel

Neptunkód: XUE9MH

A feladat leírása:

A feladatban egy olyan adatbázist hozok létre amely több üzlet adatait kezeli. Megtalálhatóak benne a beszállító, kiszállító (futár), vásárló és egyéb résztvevők adatai.

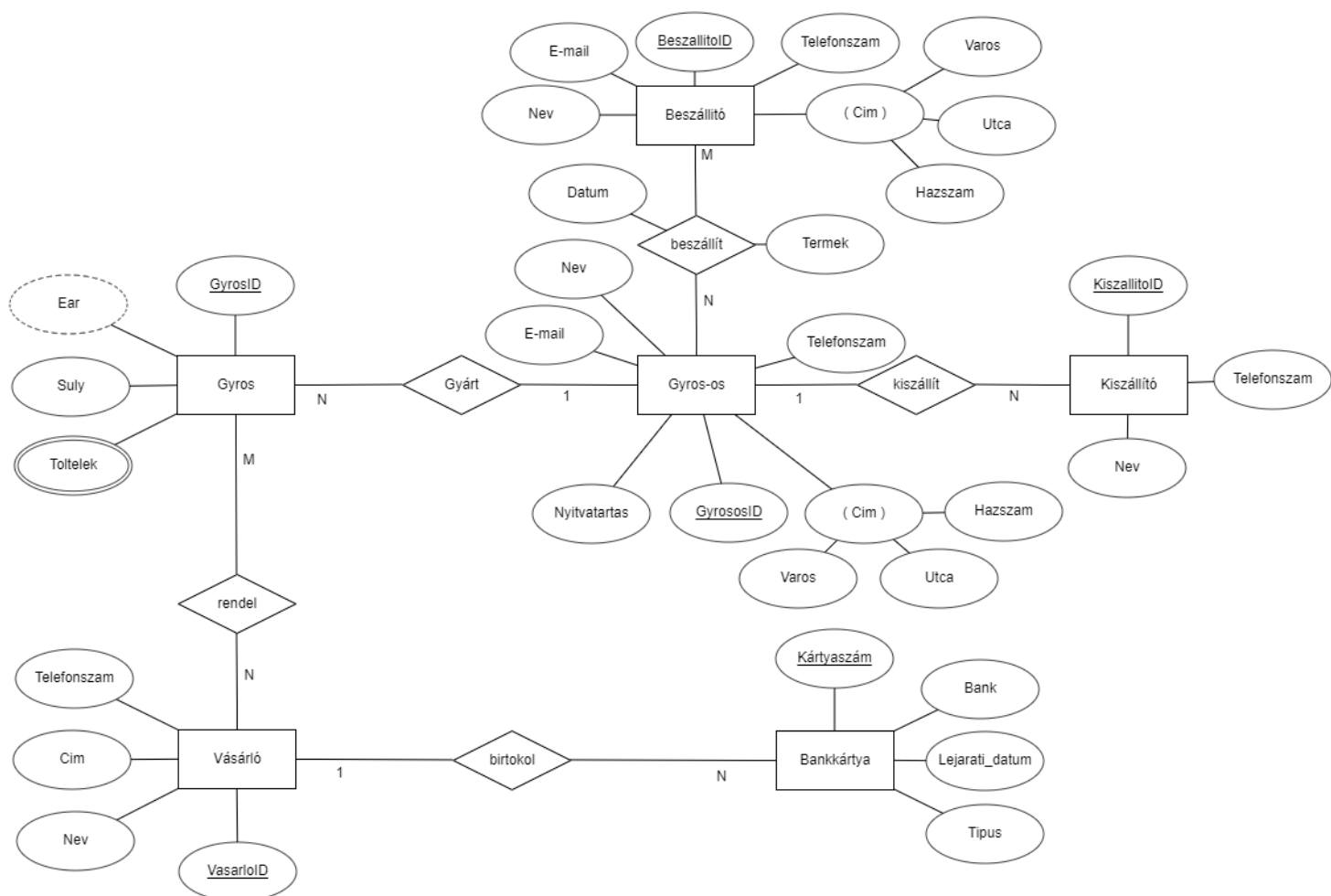
ER modelle egyedei és tulajdonságai:

- **A Gyros-os** egyed tulajdonságai
 - GyrososID – az egyed elsődleges kulcsa
 - Nyitvatartás – az üzlet nyitvatartása
 - Cím – összetett tulajdonság, üzlet címe
 - Telefonszam – az üzlet telefonos elérhetősége
 - E-mail – az üzlet telefonos elérhetősége
 - Név – üzlet neve
- **A Kiszállító** egyed tulajdonságai
 - KiszallitoID – az egyed elsodleges kulcsa
 - Telefonszam – a kiszállító(futár) elérhetősége
 - Név – a kiszállító(futár) neve
- **A Beszállító** egyed tulajdonságai
 - BeszallitoID – az egyed elsődlehes kulcsa
 - Telefonszam – a beszállító telefonos elérhetősége
 - E-mail – a beszállító internetes elérhetősége
 - Név – a beszállító neve
 - Cím – összetett tulajdonság, beszállító címe
- **A Vásárló** egyed tulajdonságai
 - VasarloID – az egyed elsődleges kulcsa
 - Telefonszam – a vásárló elérhetősége
 - Cim – vásárló címe
 - Név – vásárló neve
- **A Bankkártya** egyed tulajdonságai
 - Kartyaszám – az egyed elsődleges kulcsa
 - Bank – a kártyát kiadó bank neve
 - Lejarati_datum – a kártya lejárat dátuma
 - Tipus – a kártya típusa
- **A Gyros** egyed tulajdonságai
 - GyrosID – az egyed elsődleges kulcsa
 - Ear – a termék egységára, származtatott tulajdonság
 - Suly – a gyros súlya
 - Toltelek – többértékű tulajdonság, gyros-ban lévő húsok és zöldségek

Egyedek közötti kapcsolat:

- **Gyros és Vásárló** – ezek között az egyedek közt több-több kapcsolat van, mivel egy vásárló több gyros-t is rendelhet és egy gyros-t több vásárló rendelhet
- **Vásárló és Bankkártya** – ezek között az egyedek közt egy-több kapcsolat van, mivel egy vásárló több kártyával rendelkezhet de egy kártyának csupán egy tulajdonosa lehet
- **Gyros-os és Beszállító** – ezek között az egyedek közt több-több kapcsolat van, mivel egy üzlet több beszállítóval is rendelkezhet és egy beszállító több üzletnek is szállíthat. Kapcsolat paraméterei a szállítás Dátuma és a szállított Termék
- **Gyros-os és Kiszállító** – ezek között az egyedek között egy-több kapcsolat van, mivel egy üzlet több futárt is alkalmazhat, de egy futár csak egy üzletnek dolgozik
- **Gyros-os és Gyros** – ezek az egyedek közt egy-több kapcsolat van, mivel egy üzletnek lehet több gyros-a de egy gyros csak egy üzlethez tartozhat

Az adatbázis ER modellje:



Az adatbázis konvertálása relációs modellre:

A **Beszállító** egyedből egy tábla lesz. Elsődleges kulcsa a **BeszallitoID** lesz, attribútumaiból és a cím komponenseiből lesznek a tábla mezői.

A **beszállít** kapcsolat egy külön tábla lesz, aminek a paraméterei lesznek a mezők. Két idegen kulcsa lesz, a **BeszallitoID** ami a **Beszállító** elsődleges kulcsával áll kapcsolatban és a **GyrososID**, amely a **Gyros-os** elsődleges kulcsához kapcsolódik.

A **Gyros-os** egyedből egy tábla lesz. Elsődleges kulcsa a **GyrososID** lesz, attribútumaiból és a cím komponenseiből lesznek a tábla mezői.

A **Kiszállító** egyedből egy tábla lesz. Elsődleges kulcsa a **KiszallitoID** lesz, valamint lesz egy idegen kulcsa, a **GyrososID** ami a **Gyros-os** elsődleges kulcsához kapcsolódik. Attribútumaiból lesznek mezői.

A **Vásárló** egyedből egy tábla lesz. Az elsődleges kulcsa a **VasarloID** lesz, attribútumaiból lesznek a mezők.

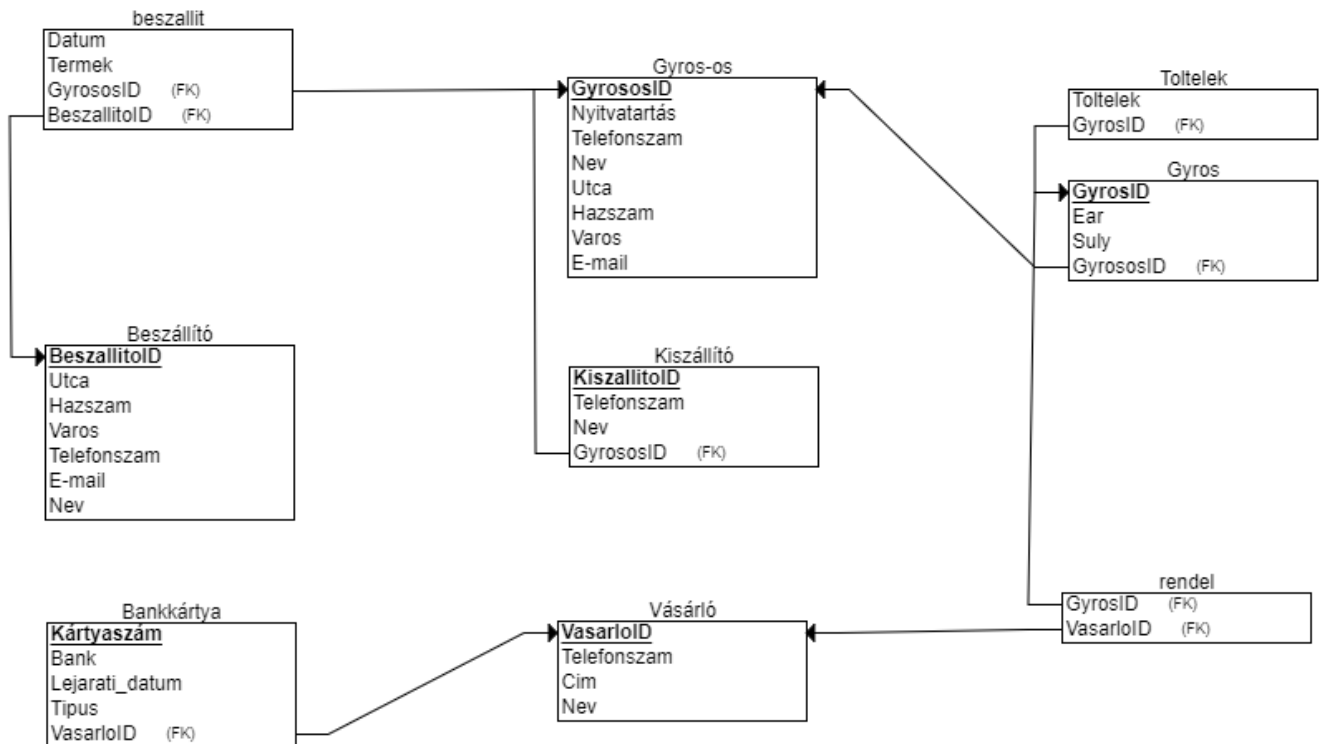
A **Gyros** egyedből egy tábla lesz. A **GyrosID** lesz az elsődleges kulcsa és lesz egy idegen kulcsa, a **GyrososID** amely a **Gyros-os** elsődleges kulcsához kapcsolódik. Attribútumai lesznek a tábla mezői, kivéve a **Toltelek**-et amely külön tábla lesz, mivel többérték tulajdonság.

A **Bankkártya** egyedből egy tábla lesz. Elsődleges kulcsa a **Kartyaszám** lesz, valamint lesz egy idegen kulcsa, a **VasarloID**, amely a **Vásárló** tábla elsődleges kulcsához kapcsolódik.

A **Toltelek** egy külön tábla lesz, aminek egy **GyrosID** idegen kulcsa lesz, amely a **Gyros** tábla elsődleges kulcsával lesz kapcsolatban, emellett a Toltelekek attribútumból lesz a mezője.

A **rendel** kapcsolat egy külön tábla lesz. Két idegen kulcsa lesz, a **GyrosID** amely a **Gyros** tábla elsődleges kulcsához kapcsolódik valamint a **VasarloID** amely a **Vásárló** tábla elsődleges kulcsához kapcsolódik.

Az adatbázis relációs modellje:



Az adatbázis relációs sémája:

Gyrosos [GyrososID, Nyitvatartas, Telefonszam, Nev, Utca, Hazszam, Varos, Email]

Beszallito [BeszallitoID, Utca, Hazszam, Varos, Telefonszam, Email, Nev]

Beszallit [Datum, Termek, GyrososID, BeszallitoID]

Kiszallito [KiszallitoID, Telefonszam, Nev, GyrososID]

Gyros [GyrosID, Ear, Suly, GyrososID]

Toltelek [Toltelek, GyrosID]

Vasarlot [VasarlotID, Telefonszam, Cim, nev]

Bankkartya [Kartyaszam, Bank, Lejarati_datum, Tipus, VasarlotID]

Rendel [GyrosID, VasarlotID]

A táblák létrehozása:

```
CREATE TABLE Gyrosos (  
  GyrososID INT NOT NULL,  
  Nyitvatartas VARCHAR(20),  
  Telefonszam INT,  
  Nev VARCHAR(30),  
  Utca VARCHAR(30),  
  Hazszam NUMBER,  
  Varos VARCHAR(30),  
  Email VARCHAR(50),  
  PRIMARY KEY (GyrososID)  
);
```

```
CREATE TABLE Beszallito (  
  BeszallitoID INT NOT NULL,  
  Utca VARCHAR(30),  
  Hazszam NUMBER,  
  Varos VARCHAR(30),  
  Telefonszam INT,  
  Email VARCHAR(50),  
  Nev VARCHAR(30),  
  PRIMARY KEY (BeszallitoID)  
);
```

```
CREATE TABLE Beszallit (  
  Datum DATE,  
  Termek VARCHAR(20),  
  GyrososID INT,  
  BeszallitoID INT,  
  FOREIGN KEY (GyrososID) REFERENCES Gyrosos(GyrososID),  
  FOREIGN KEY (BeszallitoID) REFERENCES Beszallito(BeszallitoID)  
);
```

```
CREATE TABLE Kiszallito (  
  KiszallitoID INT NOT NULL,  
  Telefonszam INT,  
  Nev VARCHAR(30),  
  GyrososID INT,  
  PRIMARY KEY (KiszallitoID),  
  FOREIGN KEY (GyrososID) REFERENCES Gyrosos(GyrososID)  
);
```

```
CREATE TABLE Gyros (  
  GyrosID INT NOT NULL,  
  Ear VARCHAR(7),  
  Suly INT NOT NULL,  
  GyrososID INT NOT NULL,  
  PRIMARY KEY (GyrosID),  
  FOREIGN KEY (GyrososID) REFERENCES Gyrosos(GyrososID)  
);
```

```
CREATE TABLE Toltelek (  
  ToltelekID INT NOT NULL,
```

```
Toltelek VARCHAR(25),
GyrosID INT,
FOREIGN KEY (GyrosID) REFERENCES Gyros(GyrosID)
);
```

```
CREATE TABLE Vasarlo (
VasarloID INT NOT NULL,
Telefonszam INT,
Cim VARCHAR (70),
Nev VARCHAR(30),
PRIMARY KEY (VasarloID)
);
```

```
CREATE TABLE Bankkartya (
Kartyaszam INT NOT NULL,
Bank VARCHAR(30),
Lejarati_datum DATE,
Tipus VARCHAR(20),
VasarloID INT,
PRIMARY KEY (Kartyaszam),
FOREIGN KEY (VasarloID) REFERENCES Vasarlo(VasarloID)
);
```

```
CREATE TABLE Rendel (
GyrosID INT,
VasarloID INT,
FOREIGN KEY (GyrosID) REFERENCES Gyros(GyrosID),
FOREIGN KEY (VasarloID) REFERENCES Vasarlo(VasarloID)
);
```

A táblák feltöltése:

```
BEGIN
```

```
INSERT INTO Gyrosos VALUES (1, '0/24', 132456789, 'Mutchen-dutchen', 'WallStreet', 66,
'New York', 'valami@mail.alien');
```

```
INSERT INTO Gyrosos VALUES (2, '0/24', 234567891, 'Hammas', 'WallStreet', 22, 'New
York', 'valamimas@mail.alien');
```

```
INSERT INTO Gyrosos VALUES (3, '0/24', 345678912, 'Hesbollah', 'WallStreet', 33, 'New
York', 'valamiegyebb@mail.alien');
```

```
INSERT INTO Gyrosos VALUES (4, '0/24', 456789123, 'MC.Donalds', 'Hlavne', 11, 'Kosice',
'egyedi@mail.alien');
```

```
INSERT INTO Gyrosos VALUES (5, '0/24', 567891234, 'KFC', 'Brookstreet', 6, 'London',
'valamiegyedi@mail.alien');
```

```
END;
```

```
BEGIN
```

```
INSERT INTO Beszallito VALUES (1, 'Utca1', 1, 'Varos1', 123456789, 'szallito1@mail.ru',
'Szallito1');
```

```
INSERT INTO Beszallito VALUES (2, 'Utca2', 2, 'Varos2', 234567891, 'szallito2@mail.ru',
'Szallito2');
```

```
INSERT INTO Beszallito VALUES (3, 'Utca3', 3, 'Varos3', 345678912, 'szallito3@mail.ru',  
'Szallito3');  
INSERT INTO Beszallito VALUES (4, 'Utca4', 4, 'Varos4', 456789123, 'szallito4@mail.ru',  
'Szallito4');  
INSERT INTO Beszallito VALUES (5, 'Utca5', 5, 'Varos5', 567891234, 'szallito5@mail.ru',  
'Szallito5');
```

END;

BEGIN

```
INSERT INTO Beszallit VALUES (TO_DATE('2011-11-11','YYYY-MM-DD'), 'Hus', 1, 3);  
INSERT INTO Beszallit VALUES (TO_DATE('2022-02-22','YYYY-MM-DD'), 'Zoldseg', 1, 1);  
INSERT INTO Beszallit VALUES (TO_DATE('2033-03-23','YYYY-MM-DD'), 'Pita', 5, 3);
```

END;

BEGIN

```
INSERT INTO Kiszallito VALUES (1, 123456789, 'Nev1', 2);  
INSERT INTO Kiszallito VALUES (2, 234567891, 'Nev2', 2);  
INSERT INTO Kiszallito VALUES (3, 345678912, 'Nev3', 3);  
INSERT INTO Kiszallito VALUES (4, 456789123, 'Nev4', 3);  
INSERT INTO Kiszallito VALUES (5, 567891234, 'Nev5', 1);
```

END;

BEGIN

```
INSERT INTO Gyros VALUES (1,'1010HUF',150,5);  
INSERT INTO Gyros VALUES (2,'2020HUF',175,4);  
INSERT INTO Gyros VALUES (3,'5EUR',160,3);  
INSERT INTO Gyros VALUES (4,'1500HUF',160,2);  
INSERT INTO Gyros VALUES (5,'1000HUF',135,1);
```

END;

BEGIN

```
INSERT INTO Toltelek VALUES ('minden',2);  
INSERT INTO Toltelek VALUES ('nemcsipos',4);  
INSERT INTO Toltelek VALUES ('nincshagyma',3);
```

END;

BEGIN

```
INSERT INTO Vasarlo VALUES (1,123456789,'Cim1','Nev1');  
INSERT INTO Vasarlo VALUES (2,234567891,'Cim2','Nev2');  
INSERT INTO Vasarlo VALUES (3,345678912,'Cim3','Nev3');  
INSERT INTO Vasarlo VALUES (4,456789123,'Cim2','Nev4');  
INSERT INTO Vasarlo VALUES (5,567891234,'Cim1','Nev2');
```

END;

BEGIN

```
INSERT INTO Bankkartya VALUES (12341234,'Bank1', TO_DATE('2001-11-11','YYYY-MM-DD'),  
'Visa',5);  
INSERT INTO Bankkartya VALUES (23452345,'Bank2', TO_DATE('2002-12-22','YYYY-MM-DD'),  
'Visa',4);  
INSERT INTO Bankkartya VALUES (34563456,'Bank3', TO_DATE('2003-03-23','YYYY-MM-DD'),  
'Visa',3);  
INSERT INTO Bankkartya VALUES (45674567,'Bank4', TO_DATE('2004-04-24','YYYY-MM-DD'),  
'Visa',2);
```



```
INSERT INTO Bankkartya VALUES (56785678,'Bank5', TO_DATE('2005-05-25','YYYY-MM-DD'),'Visa',1);  
END;
```

```
BEGIN  
    INSERT INTO Rendel VALUES (5,1);  
    INSERT INTO Rendel VALUES (4,2);  
    INSERT INTO Rendel VALUES (4,4);  
END;
```

Módosítások:

1. UPDATE Gyrosos SET Nyitvatartas='24/7' WHERE GyrososID='2';
2. UPDATE Kiszallito SET Telefonszam='66666666' WHERE KiszallitoID='3';
3. UPDATE Gyros SET Ear='1800HUF' WHERE GyrosID='3';
4. UPDATE Bankkartya SET Lejarati_datum='2003-03-13' WHERE VasarloID='3';
5. UPDATE Gyrosos SET Utca='Time square' WHERE GyrososID='3';

Lekérdezések:

- 1.Hány darab Gyros-os van 'New York' -ban?

```
SELECT count(*) FROM Gyrosos WHERE Varos='New York';
```

- 2.Vásárló táblából kilistázza a neveket és címeket.

```
SELECT Nev, Cim FROM Vasarlo;
```

- 3.Kilistázni az olyan Gyros-ok listáját amik kevesebbek mint 160 g.

```
SELECT * FROM Gyros WHERE suly < 160;
```

4. Irja ki a listajat azoknak a Gyros-osoknak ahol 160g alatti Gyros-okat árulnak.

```
SELECT Nev FROM Gyrosos JOIN Gyros USING (GyrososID) WHERE suly < 160;
```

5. Átlagosnál nagyobb Gyrosok listája.

```
SELECT * FROM Gyros WHERE suly > (SELECT avg(suly) FROM Gyros);
```

- 6.Melyik városban hány Gyros-os van?

```
SELECT Varos, count(*)"Gyrososok száma" FROM Gyrosos GROUP BY Varos;
```

- 7.Melyik Városban étkeztek Gyros-t?

```
SELECT Vasarlo.nev, Varos FROM Vasarlo LEFT JOIN Rendel USING(VasarloID) LEFT  
JOIN Gyros USING( GyrosID) LEFT JOIN Gyrosos USING( GyrososID) ORDER BY  
Vasarlo.nev;
```

8.Irjuk ki a lejárt kártyákkal rendelkező vásárlók neveit.

```
SELECT Nev, Lejarati_datum FROM Bankkartya JOIN Vasarlo USING(VasarloID)
WHERE Lejarati_datum < sysdate;
```

9.Kik azok a vásárlók akik átlag súlytól kisebb Gyros-t ettek?

```
SELECT Nev, Suly FROM Vasarlo JOIN rendel USING(VasarloID) JOIN Gyros
USING(GyrosID) WHERE Suly < ( SELECT AVG(Suly) FROM Gyros);
```

10.Milyen banknál van fiókjuk azoknak a vásárlóknak akik az ID=2 Gyrosos-nál ettek?

```
SELECT Bank, Vasarlo.Nev"Vásárló neve", Toltelek, Gyrosos.Nev"Gyrosos neve" FROM
Bankkartya JOIN Vasarlo USING (VasarloID) JOIN rendel USING(VasarloID) JOIN Gyros
USING(GyrosID) JOIN Toltelek USING(GyrosID) JOIN Gyrosos USING(GyrososID)
WHERE GyrososID=2;
```