

# Jegyzőkönyv

Adatbázisrendszerek I.

Féléves Feladat: Klinikai rendelés

Készítette: Berki Viktor

Neptunkód: ZY5P7F

Gyakorlatvezető: Dr. Bednarik László

Miskolc, 2022

## 1. A feladat leírása

A feladat lényege, hogy bemutassa az államilag finanszírozott klinikák rendelés folyamatát adatbázis eljárás folyamatban.

A feladatban található 5 egyed és mindegyik legalább rendelkezik 4 tulajdonsággal, ezek az egyedek kapcsolatban állnak egymással. Egyikük 1-1, 1-n, vagy n-m kapcsolatban.

Nézzük is meg azt az öt egyedet:

- Klinikák
- Egészségügyi minisztérium
- Pénzügyminisztérium
- Gyártó
- Kiszállítócég

Ebben a dokumentumban bemutatóm, hogyan valósítottam meg az adatbázisomat ezek a lépések szerint.

Először is létre hoztam az ER modellem, amely során lehetet tudni, hogy milyen egyedek, tulajdonságok és kapcsolatok lesznek.

Majd Relációs adatmodellre alakítottam, hogy megállapítható lehessen a táblák kinézete és olyan táblák kinézete, ami n-m kapcsolatja van.

Ez után létre hoztam a Relációs sémáját amivel megvizsgálható szöveges formátumban az adatmodell.

Ezt követően létre hoztam az adatbázist, táblákat, amelyekben beállítottam a kívánt adat típusát.

Később feltöltöttem az oszlopokat adatokkal, majd lekérdeéseket hajtottam végre.

Ezeket a részleteket be is fogom mutatni a következő pontokban.

Klinikak tábla:

Tartalmazza klinikai igazgatók azonosító számát, igazgató nevét, annak a főtitkárát, a Euminiszter\_ID foregin kulcsát és a Csomag\_ID-t.

Ebben a táblázatban található az összes rendelő igazgató.

Klinikak\_Cim tábla:

Tartalmazza a klinikák címét, ahova az igazgatók rendeltek.

Egeszsegugyi\_miniszterium tábla:

Tartalmazza a minisztérium részvevőit, akik hatás körül a rendelés. Euminiszter\_ID a kulcs, Eu\_miniszter a miniszter elnököt tartalmazza, míg az Allamtitkar pedig az államtitkárt tartalmazza. Elnokok a bizottság elnökei nevét tartalmazza, míg a tisztseg\_kep pedig a tisztség képviselőket.

Foreign kulcsa Puminiszter\_ID

Penzugyi\_miniszterium tábla:

Ugyanazt tartalmazza, mint az Egeszsegugyi\_miniszterium csak az Euminiszter\_ID helyett Puminiszter ID található és az Eu\_miniszter helyett Pu\_miniszter.

Foreign kulcs itt nem található, ez az első tábla, ami létrehozásra került.

Gyarto tábla:

Ebben a táblában található a Termék\_ID, ami szerint rendelték a termékeket és egyben a kulcs is. Megtalálható a Tulajdonos, Raktaron és a Legyartanivalo, ami értelmű szerűen tudni lehet mit tartalmaz. Majd a Puminiszter\_ID mint foreign kulcs.

Gyarto\_gyar:

Ez egy olyan tábla, ami a több tulajdonságot tartalmazó tulajdonság miatt jött létre, ez tartalmazza a Termek\_ID mint kulcs és Foreign kulcs és a Gyar, ami a gyár nevét tartalmazza.

Feladas tábla:

Ez egy n-m kapcsolat létre jötté miatt jött létre tábla. Ennek a táblának két foreign kulcs kell tartalmaznia és legalább egy tulajdonságnak. Itt a Termek\_ID és a Csomag\_ID nevét található mint foreign kulcs és a Feladas\_ideje.

Kiszallitoceg tábla:

Itt a kulcs a Csomag\_ID viszont nem megtevesztésnek, de található Igazgato\_ID is, de csak az igazgatót azonosítja. Ezen belül már a Csomag\_nev is megtalálható.

Kiszallitoceg tábla:

Szintén több tulajdonság miatt jött létre.  
Itt található a Csomag\_ID mint foreign kulcs és a Telephelyek.

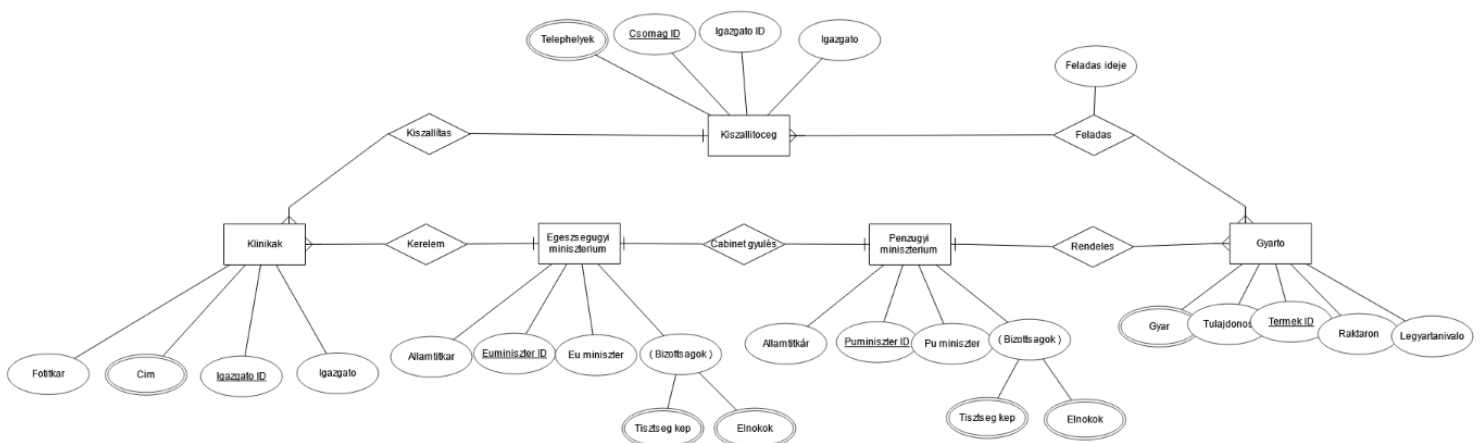
## 2. ER modell létrehozása

Először is kiválasztottam a megfelelő programot a rajz elkészítése, ugyan bár kézzel is megrajzolható, de én az ERDplus weboldalon készítettem el.

Egyedeket a négyzettel jelöljük, amelybe a táblának a nevét írjuk be.

Következő lépésben az egyedeket egyenes vonallal kötjük össze a tulajdonságainkkal, ami a tábla oszlopai lesznek. A tulajdonságokat pedig egy fektetett lapos körrel ábrázoljuk. Bizonyos egyedek tulajdonságainak lehet több saját tulajdonsága vagy pedig több értékű is lehet. Minden egyednek legalább van 1 darab kulcsa, ami összeköteti a többi egyeddel.

Ha ezekkel megvolnánk, akkor jönnek a kapcsolatok, amiket egyedfelé mutató nyíllal kötünk össze vagy pedig egyenes vonallal, amelynek az egyed felől egy merőleges vonal áthúzzuk. Természetesen jelölni kell, hogy a kapcsolat egy vagy több irányú kapcsolat, mint pl.: egy autónak több tulajdonosa lehet. Ez esetben, ha az adott egyednek több kapcsolata van, akkor a másik egyednél 2 nyíllal jelöljük. Ha viszont a vonal áthúzásos megoldás alkalmaztuk akkor annak az egyednek, amelynek több irányú lehet a másik egyednél az áthúzásra a kapcsolat felé egy háromszöget rajzolunk. Ha a kapcsolat n-m, akkor saját tulajdonsággal is rendelkezhet.



### 3. ER modell konvertálása Relációs modellre

Hogy jobban kilátható lehessen és jobban értelmezni tudjuk, majd a kapcsolatoknak, hogy valójában hogyan is fog létre jönni ezek a táblák át konvertáljuk őket Relációs modelleké.

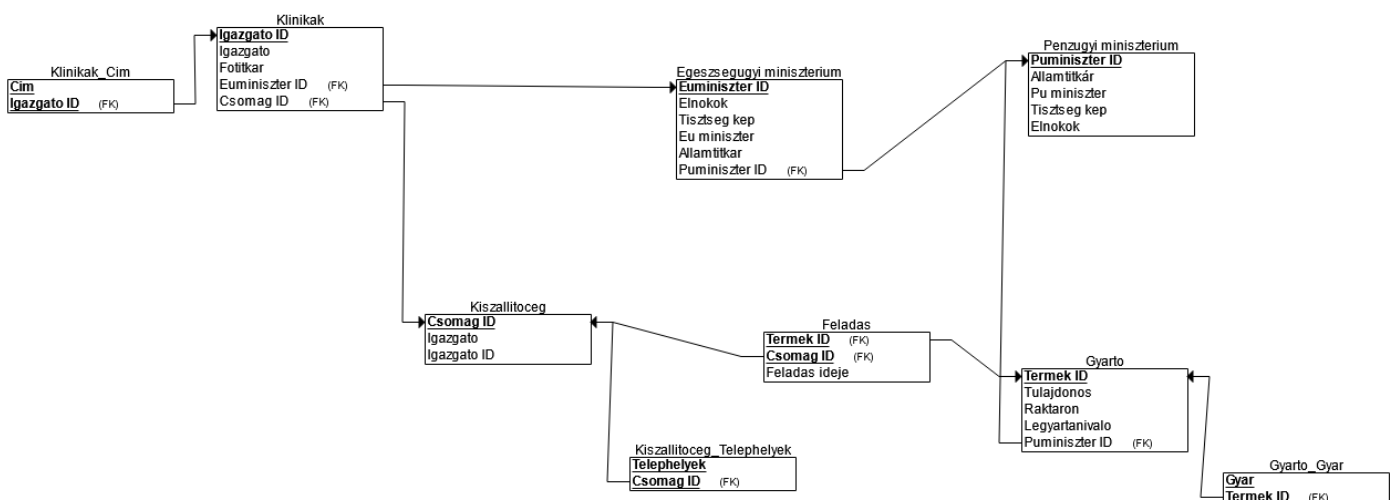
A feladat során létre hozunk annyi táblát, amennyi egyed van és annyi kapcsolatot tábla két amely n-m kapcsolatja van. Ha egy tulajdonságnak több tulajdonsággal rendelkezik külön táblát kap. Ezekbe a táblákban az első sorában az egyed neve található, a n-m kapcsolat neve, vagy a több tulajdonság neve.

Többi adat a táblában az egyed tulajdonságaival töltjük fel, az n-m kapcsolat esetén a kapcsolat tulajdonságaival, ha rendelkezik és a több tulajdonság tulajdonságaival.

Ezeket a táblákat a kulcs tulajdonságaival kötjük össze.

Ennek megvalósításához minden táblának a kapcsolatának megfelelő egyednek tartalmaznia kell a másik egyed kulcsát, amit „foregin” avagy külföldi kulcsnak nevezzünk.

Mi előtt feltöltenék minden adatot foregin kulcsokkal legalább egyetlen egy táblának nem szabad tartalmaznia, hogy a sql tábla létrehozása kor ne az lehessen a hibánk, hogy egy nem létező táblára próbáljunk referenciát létre hozni. A tábláknak a kulcsait össze kötjük nyilakkal a másik táblák foregin kulcsaival és így létre hoztuk a relációs modellünket.



#### 4. Relációs séma létrehozása

Klinikak[Igazgato\_ID, Igazgato, Fotitkar, Euminiszter\_ID, Csomag\_ID]

Klinikak\_Cim[Cim, Igazgato\_ID]

Egeszsegugyi\_miniszterium[Euminiszter\_ID, Elnokok, Tisztseg\_kep, Eu\_miniszter, Allamtitkar, Puminiszter\_ID]

Penzugyi\_miniszterium[Puminiszter\_ID, Elnokok, Tisztseg\_kep, Pu\_miniszter, Allamtitkar]

Gyarto[Termek\_ID, Tulajdonos, Raktaron, Legyartanivalo, Puminiszter\_ID]

Gyarto\_Gyar[Gyar, Termek\_ID]

Feladas[Termek\_ID, Csomag\_ID, Feladas\_ideje]

Kiszallitoceg[Csomag\_ID, Igazgato, Igazgato\_ID, Csomagnev]

Kiszallitoceg\_Telephelyek[Telephelyek, Csomag\_ID]

## 5. A adatbázis, táblák elkészítése

Ahhoz, hogy létre jöjjön az adatbázisunk egy szimpla SQL parancsot kell alkalmazunk.

```
CREATE DATABASE adatbázis_neve;
```

Ennek léte hozása után táblákat hozunk létre, aminek a parancsában tartalmazni fogja tábla nevét, tulajdonság nevét és adat típusát. A tulajdonságok lesznek az oszlopok és az oszlopokban olyan adatok kerülhetnek, amelyek egyeznek a létrehozáskor az adattípusával. Az adattípus után megadhatjuk, hogy a cellákat kötelező vagy nem kötelező kitölteni egy sorban. A tulajdonság végén megadjuk a kulcsot. Ha pedig rendelkezik foregin kulccsal, akkor azt is megadjuk. A parancs syntaxisa:

```
CREATE TABLE Gyarto
```

```
(
```

```
    Tulajdonság neve adattípus kötelező_cella_kitöltés,
```

```
    PRIMARY KEY (tábla kulcsa),
```

```
    FOREIGN KEY (táblának a külföldi kulcs neve) REFERENCES  
    az_a_tábla_ahhonaan_származik(táblának a külföldi kulcs neve)
```

```
);
```



```

CREATE TABLE Penzugyi_miniszterium
(
    Allamtitkar CHAR(20) NOT NULL,
    Puminiszter_ID INT NOT NULL,
    Pu_miniszter CHAR(20) NOT NULL,
    Tisztseg_kep CHAR(20) NOT NULL,
    Elnokok CHAR(20) NOT NULL,
    PRIMARY KEY (Puminiszter_ID)
);

CREATE TABLE Gyarto
(
    Tulajdonos CHAR(20) NOT NULL,
    Termek_ID INT NOT NULL,
    Raktaron INT NOT NULL,
    Legyartanivalo INT NOT NULL,
    Puminiszter_ID INT NOT NULL,
    PRIMARY KEY (Termek_ID),
    FOREIGN KEY (Puminiszter_ID) REFERENCES Penzugyi_miniszterium(Puminiszter_ID)
);

CREATE TABLE Kiszallitoceg
(
    Igazgato CHAR(20) NOT NULL,
    Igazgato_ID INT NOT NULL,
    Csomag_ID INT NOT NULL,
    PRIMARY KEY (Csomag_ID)
);

CREATE TABLE Feladas
(
    Feladas_ideje INT NOT NULL,
    Termek_ID INT NOT NULL,
    Csomag_ID INT NOT NULL,
    PRIMARY KEY (Termek_ID, Csomag_ID),
    FOREIGN KEY (Termek_ID) REFERENCES Gyarto(Termek_ID),
    FOREIGN KEY (Csomag_ID) REFERENCES Kiszallitoceg(Csomag_ID)
);

CREATE TABLE Gyarto_Gyar
(
    Gyar INT NOT NULL,
    Termek_ID INT NOT NULL,
    PRIMARY KEY (Gyar, Termek_ID),
    FOREIGN KEY (Termek_ID) REFERENCES Gyarto(Termek_ID)
);

CREATE TABLE Kiszallitoceg_Telephelyek
(
    Telephelyek CHAR(20) NOT NULL,
    Csomag_ID INT NOT NULL,
    PRIMARY KEY (Telephelyek, Csomag_ID),
    FOREIGN KEY (Csomag_ID) REFERENCES Kiszallitoceg(Csomag_ID)
);

CREATE TABLE Klinikak
(
    Igazgato CHAR(20) NOT NULL,
    Igazgato_ID INT NOT NULL,
    Fotitkar CHAR(20) NOT NULL,
    Euminiszter_ID INT NOT NULL,
    Csomag_ID INT NOT NULL,
    PRIMARY KEY (Igazgato_ID),
    FOREIGN KEY (Euminiszter_ID) REFERENCES Egeszsegugyi_miniszterium(Euminiszter_ID),
    FOREIGN KEY (Csomag_ID) REFERENCES Kiszallitoceg(Csomag_ID)
);

CREATE TABLE Klinikak_Cim
(
    Cim CHAR(30) NOT NULL,
    Igazgato_ID INT NOT NULL,
    PRIMARY KEY (Cim, Igazgato_ID),
    FOREIGN KEY (Igazgato_ID) REFERENCES Klinikak(Igazgato_ID)
);

```

## ▼ Tables (10)

### ▼ Egeszsegugyi\_miniszterium

Elnokok	CHAR(20)
Tisztseg_kep	CHAR(20)
Eu_miniszter	CHAR(20)
Euminiszter_ID	INTEGER
Allamtitkar	CHAR(20)
Puminiszter_ID	INT

### ▼ Feladas

Feladas_ideje	DATE
Termek_ID	INT
Csomag_ID	INT

### ▼ Gyarto

Tulajdonos	CHAR(20)
Termek_ID	INTEGER
Raktaron	INT
Legyartanivalo	INT
Puminiszter_ID	INT

### ▼ Kiszallitoceg

Igazgato	CHAR(20)
Igazgato_ID	INT
Csomag_ID	INTEGER
Csomagnev	CHAR(12)

### ▼ Kiszallitoceg\_Telephelyek

Telephelyek	CHAR(20)
Csomag_ID	INT

### ▼ Klinikak

Igazgato	CHAR(20)
Igazgato_ID	INTEGER
Fotitkar	CHAR(20)
Euminiszter_ID	INT
Csomag_ID	INT

### ▼ Klinikak\_Cim

Cim	CHAR(50)
Igazgato_ID	INT

### ▼ Penzugyi\_miniszterium

Allamtitkar	CHAR(20)
Puminiszter_ID	INTEGER
Pu_miniszter	CHAR(20)
Tisztseg_kep	CHAR(20)
Elnokok	CHAR(20)

## 6. Táblák feltöltése

A táblák feltöltéséhez több paramétert megadhatunk melyik cellákba szeretnénk és tölthetünk fel úgy is sorokat, hogy nem töltünk fel bizonyos cellákat, kivétel abban az esetben, ha az oszlop egyik tulajdonsága az, hogy kötelező kitölteni az oszlop celláját, ebben az esetben mindig úgy tudunk feltölteni, ha a parancs paraméterében benne van az a tulajdonság és adatja, amit kötelező kitölteni.

Syntaxa:

```
INSERT INTO tábla_név(tulajdonság_név,tulajdonság_név)
```

```
VALUES ('char típusú adat' , adat);
```

	Elnokok	Tisztseg_kep	Eu_miniszter	Euminiszter_ID	Allamtitkar	Puminiszter_ID
	Filter	Filter	Filter	Filter	Filter	Filter
1	Jake Smith	Jacob Jones	Connor Taylor		1 Oscar Davies	1
2	James Smith	Jack Jones	NULL		2 NULL	2
3	Thomas Davies	Kyle Evans	NULL		3 NULL	3
4	William Roberts	Oliver Wilson			4 NULL	4
5	Kyle Evans	Thomas Davies	NULL		5 NULL	5

	Feladas_ideje	Termek_ID	Csomag_ID
	Filter	Filter	Filter
1	2001	1	1
2	2005	2	2
3	1996	3	3
4	1999	4	4
5	2018	5	5
6	2018	6	6
7	1985	7	7
8	2011	8	8

	Tulajdonos	Termek_ID	Raktaron	Legyartanivalo	Puminiszter_ID
	Filter	Filter	Filter	Filter	Filter
1	Tommy Edwards	1	105	22	1
2	NULL	2	22	105	1
3	NULL	3	54	2	1
4	NULL	4	2	54	1
5	NULL	5	11	30	3
6	NULL	6	1	1	3
7	NULL	7	44	2	4
8	NULL	8	3	27	4

	Gyar	Termek_ID
	Filter	Filter
1	TEVA	1
2	TEVA	2
3	TEVA	3
4	BÉRES	4
5	TEVA	5
6	BÉRES	6
7	BÉRES	7
8	BÉRES	8

	Igazgato	Igazgato_ID	Csomag_ID	Csomagnev
	Filter	Filter	Filter	Filter
	Spark Hundson	1	1	Sebeszkes
	NULL	NULL	2	Kottszer
	NULL	NULL	3	Jod
	NULL	NULL	4	Magnezium
	NULL	NULL	5	Opium
	NULL	NULL	6	Morpium
	NULL	NULL	7	Klorofom
	NULL	NULL	8	EKG

	Telephelyek	Csomag_ID
	Filter	Filter
1	Ózd	1
2	Miskolc	2
3	Ózd	3
4	Ózd	4
5	Debrecen	5
6	Székesfehérvár	6
7	Sopron	7
8	Békéscsaba	8

	Igazgato	Igazgato_ID	Fotitkar	Euminiszter_ID	Csomag_ID
	Filter	Filter	Filter	Filter	Filter
1	Ethan Murphy	1	Daniel Miller	1	1
2	Kyle Rodriguez	2	Harry Johnson	2	2
3	Michael Murphy	3	Joseph Roberts	4	3
4	Ethan Murphy	4	Daniel Miller	4	4
5	Michael Murphy	5	Joseph Roberts	1	5
6	Kyle Rodriguez	6	Harry Johnson	3	6
7	Ethan Murphy	7	Daniel Miller	3	7
8	Michael Murphy	8	Daniel Miller	5	8

	Cim	Igazgato_ID
	Filter	Filter
1	3000 Hatvan, Balassi Bálint út 16	1
2	8230 Balatonfüred, Gyógy tér 2	2
3	6000 Kecskemét, Nyíri út 38	3
4	3600 Ózd, Béke utca. 1-3	4
5	6500 Baja, Rókus u.10	5
6	1106 Budapest, Maglódi út 89-91	6
7	2870 Kisdér, Iskola u. 11	7
8	5700 Gyula, Semmelweis u. 1	8

	Allamtitkar	Puminiszter_ID	Pu_miniszter	Tisztseg_kep	Elnokok
	Filter	Filter	Filter	Filter	Filter
1	Cleveland Brown	1	Jim Hill	Oliver Smith	George White
2	NULL	2	NULL	Arthur Harris	Jack Martin
3	NULL	3	NULL	Jacob Harris	Theo Baker
4	NULL	4	NULL	William Green	Theodore Baker
5	NULL	5	NULL	Max Baker	Theodore Clark

## 7. Lekérdezések

Ahogy látható, az adatbázisunk létre jött, viszont, ahhoz hogy adatokat is letudjunk kérdezni, parancsokat kell alkalmaznunk. Viszont többféle paraméter és számítás szerint is lekérdezhetünk. Kérdezhetünk le úgy is, hogy bizonyos adattípus vagy tartalmaz a cella ilyen adatot vagy ilyen adat részletet, vagy nem tartalmaz. A variációkból rengetek van. De az alapvetően minden lekérdezésnek tartalmaznia kell, hogy milyen oszlopban keresünk, milyen táblában.

Az alapvető lekérdezés syntaxisa:

```
SELECT oszlop_név  
FROM tábla_név;
```

```
SELECT Elnokok, Tisztseg_kep FROM Egeszsegugyi_miniszterium;
```

```
SELECT Eu_miniszter, Allamtitkar  
FROM Egeszsegugyi_miniszterium  
WHERE NOT Eu_miniszter="Not null";
```

```
SELECT Pu_miniszter, Allamtitkar  
FROM Penzugyi_miniszterium  
WHERE NOT Pu_miniszter="Not null";
```

```
SELECT Elnokok, Tisztseg_kep FROM Penzugyi_miniszterium;
```

```
SELECT count(Termek_ID)  
FROM Gyarto;
```

```
SELECT sum(Raktaron)  
FROM Gyarto;
```

```
SELECT sum(Legyartanivalo)  
FROM Gyarto;
```

```
SELECT sum(Legyartanivalo) - sum(Raktaron)  
FROM Gyarto;
```

```
SELECT Klinikak.Igazgato, Kiszallitoceg.Csomagnev  
FROM Klinikak  
INNER JOIN Kiszallitoceg ON Klinikak.Csomag_ID = Kiszallitoceg.Csomag_ID;
```

```
SELECT Klinikak_Cim.Cim, Klinikak.Igazgato, Kiszallitoceg.Csomagnev  
FROM Klinikak  
INNER JOIN Kiszallitoceg ON Klinikak.Csomag_ID = Kiszallitoceg.Csomag_ID INNER JOIN Klinikak_Cim ON Klinikak_Cim.Igazgato_ID = Klinikak.Igazgato_ID;
```

```
SELECT Klinikak_Cim.Cim, Klinikak.Igazgato, Kiszallitoceg.Csomagnev  
FROM Klinikak  
INNER JOIN Kiszallitoceg ON Klinikak.Csomag_ID = Kiszallitoceg.Csomag_ID INNER JOIN Klinikak_Cim ON Klinikak_Cim.Igazgato_ID = Klinikak.Igazgato_ID;
```

```
SELECT Gyarto_Gyar.Gyar, Kiszallitoceg.Csomagnev, Gyarto.Legyartanivalo  
FROM Feladas  
INNER JOIN Kiszallitoceg ON Feladas.Csomag_ID = Kiszallitoceg.Csomag_ID INNER JOIN Gyarto ON Gyarto.Termek_ID = Feladas.Termek_ID INNER JOIN Gyarto_Gyar ON Gyarto_Gyar.Termek_ID = Gyarto.Termek_ID;
```

```
SELECT Gyarto_Gyar.Gyar, Kiszallitoceg.Csomagnev, Gyarto.Legyartanivalo  
FROM Feladas  
INNER JOIN Kiszallitoceg ON Feladas.Csomag_ID = Kiszallitoceg.Csomag_ID INNER JOIN Gyarto ON Gyarto.Termek_ID = Feladas.Termek_ID INNER JOIN Gyarto_Gyar ON Gyarto_Gyar.Termek_ID = Gyarto.Termek_ID  
WHERE Gyar="TEVA";
```