

Jegyzőkönyv

Adatbázisrendszerek I.

Féléves Feladat: Klinikai rendelés

Készítette: Berki Viktor

Neptunkód: ZY5P7F

Gyakorlatvezető: Dr. Bednarik László

Miskolc, 2022

Tartalomjegyzék

| | |
|---|----|
| 1. A feladat leírása | 3 |
| 2. ER modell létrehozása | 6 |
| 3. ER modell konvertálása Relációs modellre | 7 |
| 4. Relációs séma létrehozása | 8 |
| 5. A adatbázis, táblák elkészítése..... | 9 |
| 6. Táblák feltöltése | 11 |
| 7. Lekérdezések | 17 |

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 résztvevőit, akik hatás körül a rendelés. Euminiszter_ID a kulcs, Eu_miniszter az egészségügyi miniszter 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 értelemszerűen 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.

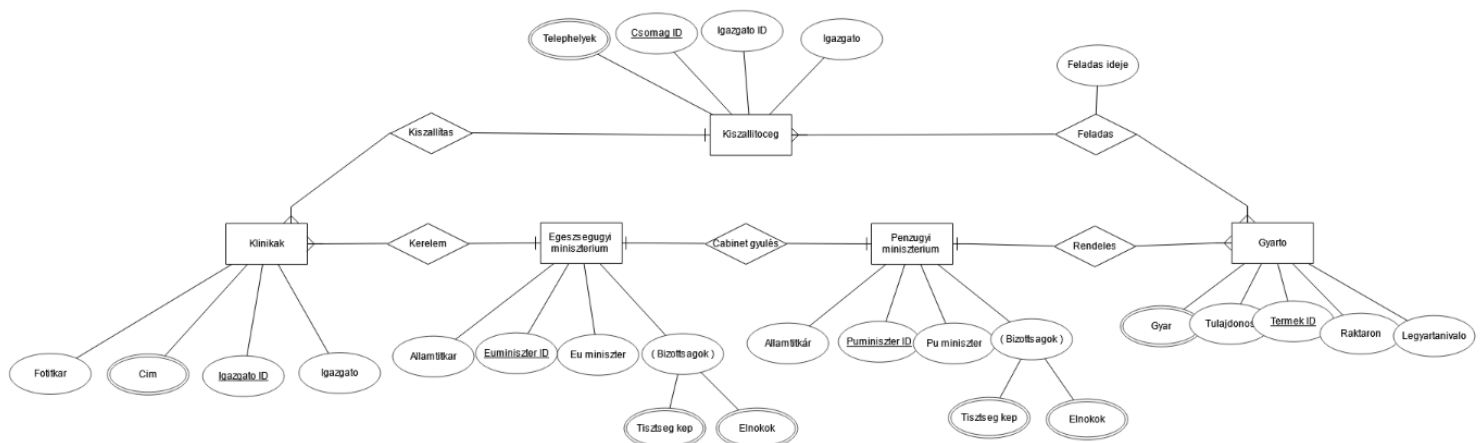
2. ER modell létrehozása

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 tulajdonságaikkal, 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 ként, amely n-m kapcsolatja van az adatbázisnak. Ha egy tulajdonságnak több tulajdonsággal rendelkezik külön táblát kap.

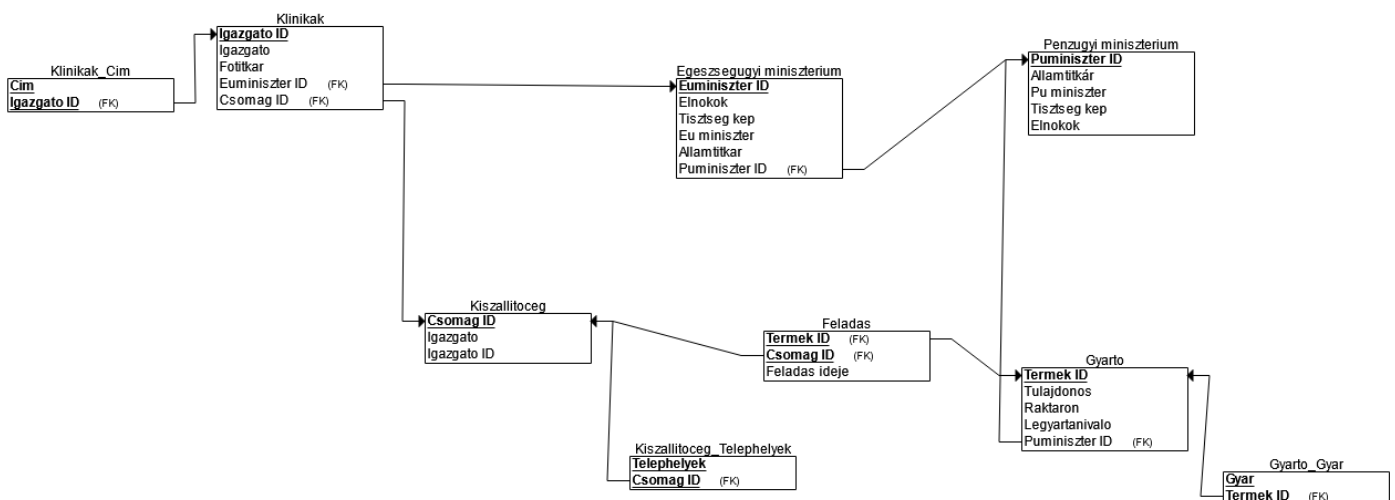
Ezekbe a táblákban az első sorában az egyed neve található, a n-m kapcsolat neve, vagy a több tulajdonság neve.

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 összekötjük 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 hozzunk 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 szintaxisa:

```
CREATE TABLE Gyarto
```

```
(
```

```
    Tulajdonság neve adattípus kötelező_cella_kitöltés,
```

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

```
    FOREIGN KEY (táblának a külföldi kulcs neve) REFERENCES  
    az_a_tábla_ahhona_n_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.

Szintaxisa:

```
INSERT INTO tábla_név(tulajdonság_név,tulajdonság_név)
VALUES ('char típusú adat' , adat);
```

Tábla feltöltés során használt parancsok:

```
INSERT INTO Kiszallitoceg (Igazgato,Igazgato_ID,Csomagnev)
VALUES ('Spark Hundson',1,'Sebeszkes');

INSERT INTO Kiszallitoceg (Csomagnev)
VALUES ('Kottszer');

INSERT INTO Kiszallitoceg (Csomagnev)
VALUES ('Jod')

INSERT INTO Kiszallitoceg (Csomagnev)
VALUES ('Magnezium');

INSERT INTO Kiszallitoceg (Csomagnev)
VALUES ('Opium');

INSERT INTO Kiszallitoceg (Csomagnev)
VALUES ('Morpium');

INSERT INTO Kiszallitoceg (Csomagnev)
VALUES ('Klorofom');

INSERT INTO Kiszallitoceg (Csomagnev)
VALUES ('EKG');

INSERT INTO Kiszallitoceg (Csomagnev)
VALUES ('Sebesz lampa');

INSERT INTO Kiszallitoceg (Csomagnev)
VALUES ('Sebtu');

INSERT INTO Penzugyi_miniszterium(Allamtitkar,Puminiszter_ID,Pu_miniszter,Tisztseg_kep,Elnokok)
VALUES ('Cleveland Brown',1,'Jim Hill','Oliver Smith','George White');

INSERT INTO Penzugyi_miniszterium(Tisztseg_kep,Elnokok)
```

```

VALUES ('Arthur Harris','Jack Martin');

INSERT INTO Penzugyi_miniszterium(Tisztseg_kep,Elnokok)

VALUES ('Jacob Harris','Theo Baker');

INSERT INTO Penzugyi_miniszterium(Tisztseg_kep,Elnokok)

VALUES ('William Green','Theodore Baker');

INSERT INTO Penzugyi_miniszterium(Tisztseg_kep,Elnokok)

VALUES ('Max Baker','Theodore Clark');

INSERT INTO Gyarto(Tulajdonos,Raktaron,Legyartanivalo,Puminiszter_ID)

VALUES (105,22,1);

INSERT INTO Gyarto(Raktaron,Legyartanivalo,Puminiszter_ID)

VALUES (22,105,1);

INSERT INTO Gyarto(Raktaron,Legyartanivalo,Puminiszter_ID)

VALUES (54,2,1);

INSERT INTO Gyarto(Raktaron,Legyartanivalo,Puminiszter_ID)

VALUES (2,54,1);

INSERT INTO Gyarto(Raktaron,Legyartanivalo,Puminiszter_ID)

VALUES (11,30,3);

INSERT INTO Gyarto(Raktaron,Legyartanivalo,Puminiszter_ID)

VALUES (1,1,3);

INSERT INTO Gyarto(Raktaron,Legyartanivalo,Puminiszter_ID)

VALUES (44,2,4);

INSERT INTO Gyarto(Raktaron,Legyartanivalo,Puminiszter_ID)

VALUES (3,27,4);

INSERT INTO Gyarto_Gyar(Gyar,Termek_ID)

VALUES ('TEVA',1);

INSERT INTO Gyarto_Gyar(Gyar,Termek_ID)

VALUES ('TEVA',2);

INSERT INTO Gyarto_Gyar(Gyar,Termek_ID)

VALUES ('TEVA',3);

INSERT INTO Gyarto_Gyar(Gyar,Termek_ID)

VALUES ('BÉRES',4);

INSERT INTO Gyarto_Gyar(Gyar,Termek_ID)

VALUES ('TEVA',5);

INSERT INTO Gyarto_Gyar(Gyar,Termek_ID)

VALUES ('BÉRES',6);

INSERT INTO Gyarto_Gyar(Gyar,Termek_ID)

```

```

VALUES ('BÉRES',7);
INSERT INTO Gyarto_Gyar(Gyar,Termek_ID)
VALUES ('BÉRES',8);
INSERT INTO Feladas(Feladas_ideje,Termek_ID,Csomag_ID)
VALUES (2022-10-11,1,1)
INSERT INTO Feladas(Feladas_ideje,Termek_ID,Csomag_ID)
VALUES (2021-09-07,2,2)
INSERT INTO Feladas(Feladas_ideje,Termek_ID,Csomag_ID)
VALUES (2021-03-22,3,3)
INSERT INTO Feladas(Feladas_ideje,Termek_ID,Csomag_ID)
VALUES (2022-07-16,4,4)
INSERT INTO Feladas(Feladas_ideje,Termek_ID,Csomag_ID)
VALUES (2020-01-01,5,5)
INSERT INTO Feladas(Feladas_ideje,Termek_ID,Csomag_ID)
VALUES (2020-01-01,6,6)
INSERT INTO Feladas(Feladas_ideje,Termek_ID,Csomag_ID)
VALUES (2017-05-27,7,7)
INSERT INTO Feladas(Feladas_ideje,Termek_ID,Csomag_ID)
VALUES (2022-08-03,8,8)
INSERT INTO Kiszallitoceg_Telephelyek(Telephelyek,Csomag_ID)
VALUES('Ózd',1);
INSERT INTO Kiszallitoceg_Telephelyek(Telephelyek,Csomag_ID)
VALUES('Miskolc',2);
INSERT INTO Kiszallitoceg_Telephelyek(Telephelyek,Csomag_ID)
VALUES('Ózd',3);
INSERT INTO Kiszallitoceg_Telephelyek(Telephelyek,Csomag_ID)
VALUES('Ózd',4);
INSERT INTO Kiszallitoceg_Telephelyek(Telephelyek,Csomag_ID)
VALUES('Debrecen',5);
INSERT INTO Kiszallitoceg_Telephelyek(Telephelyek,Csomag_ID)
VALUES('Székesfehérvár',6);
INSERT INTO Kiszallitoceg_Telephelyek(Telephelyek,Csomag_ID)
VALUES('Sopron',7);
INSERT INTO Kiszallitoceg_Telephelyek(Telephelyek,Csomag_ID)
VALUES('Békéscsaba',8);
INSERT INTO Egeszsegugyi_miniszterium(Elnokok,Tisztseg_kep,Eu_miniszter,Allamtitkar,Puminiszter_ID)

```

```

VALUES('Jake Smith','Jacob Jones','Connor Taylor','Oscar Davies',1);
INSERT INTO Egeszsegugyi_miniszterium(Elnokok,Tisztseg_kep,Puminiszter_ID)
VALUES('James Smith','Jack Jones',2);
INSERT INTO Egeszsegugyi_miniszterium(Elnokok,Tisztseg_kep,Puminiszter_ID)
VALUES('Thomas Davies','Kyle Evans',3);
INSERT INTO Egeszsegugyi_miniszterium(Elnokok,Tisztseg_kep,Puminiszter_ID)
VALUES('William Roberts','Oliver Wilson',4);
INSERT INTO Egeszsegugyi_miniszterium(Elnokok,Tisztseg_kep,Puminiszter_ID)
VALUES('Kyle Evans','Thomas Davies',5);
INSERT INTO Klinikak(Igazgato,Fotitkar,Euminiszter_ID,Csomag_ID)
VALUES ('Ethan Murphy','Daniel Miller',1,1);
INSERT INTO Klinikak(Igazgato,Fotitkar,Euminiszter_ID,Csomag_ID)
VALUES ('Kyle Rodriguez','Harry Johnson',2,2);
INSERT INTO Klinikak(Igazgato,Fotitkar,Euminiszter_ID,Csomag_ID)
VALUES ('Michael Murphy','Joseph Roberts',4,3);
INSERT INTO Klinikak(Igazgato,Fotitkar,Euminiszter_ID,Csomag_ID)
VALUES ('Ethan Murphy','Daniel Miller',4,4);
INSERT INTO Klinikak(Igazgato,Fotitkar,Euminiszter_ID,Csomag_ID)
VALUES ('Michael Murphy','Joseph Roberts',1,5);
INSERT INTO Klinikak(Igazgato,Fotitkar,Euminiszter_ID,Csomag_ID)
VALUES ('Kyle Rodriguez','Harry Johnson',3,6);
INSERT INTO Klinikak(Igazgato,Fotitkar,Euminiszter_ID,Csomag_ID)
VALUES ('Ethan Murphy','Daniel Miller',3,7);
INSERT INTO Klinikak(Igazgato,Fotitkar,Euminiszter_ID,Csomag_ID)
VALUES ('Michael Murphy','Daniel Miller',5,8);
INSERT INTO Klinikak_Cim(Cim,Igazgato_ID)
VALUES ('3000 Hatvan, Balassi Bálint út 16',1);
INSERT INTO Klinikak_Cim(Cim,Igazgato_ID)
VALUES ('8230 Balatonfüred, Gyógy tér 2',2);
INSERT INTO Klinikak_Cim(Cim,Igazgato_ID)
VALUES ('6000 Kecskemét, Nyíri út 38',3);
INSERT INTO Klinikak_Cim(Cim,Igazgato_ID)
VALUES ('3600 Ózd, Béke utca. 1-3',4);
INSERT INTO Klinikak_Cim(Cim,Igazgato_ID)
VALUES ('6500 Baja, Rókus u.10',5);

```

```
INSERT INTO Klinikak_Cim(Cim,Igazgato_ID)
VALUES ('1106 Budapest, Maglódi út 89-91',6);
INSERT INTO Klinikak_Cim(Cim,Igazgato_ID)
VALUES ('2870 Kisbér, Iskola u. 11',7);
INSERT INTO Klinikak_Cim(Cim,Igazgato_ID)
VALUES ('5700 Gyula, Semmelweis u. 1',8);
```

| | 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 Kiskőr, 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 több féle 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 szintaxisa:

```
SELECT oszlop_név  
FROM tábla_név;
```

1. Le kérdezzük a bizottság elnököket és tisztség képviselőket az egészségügyi minisztériumban.

Relációs algebra: $\pi_{elnokok, tisztség_kep} \text{egeszsegugyi_miniszterium}$

Parancs:

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

2. Le kérdezzük az egészségügyi minisztert és az államtitkárját az egészségügyi minisztériumban.

Relációs algebra: $\pi_{eu_miniszter, allamtitkar}$

$\sigma_{NOT (eu_miniszter = "Not null")} \text{egeszsegugyi_miniszterium}$

Parancs:

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

3. Le kérdezzük a pénzügyi minisztert és az államtitkárját a pénzügyi minisztériumban.

Relációs algebra: $\pi_{pu_miniszter, allamtitkar}$

$\sigma_{NOT (pu_miniszter = "Not null")}$ *penzugyi_miniszterium*

Parancs:

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

4. Le kérdezzük a bizottság elnököket és tisztség képviselőket a pénzügyi minisztériumban.

Relációs algebra: $\pi_{elnokok, tisztség_kep}$ *penzugyi_miniszterium*

Parancs:

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

5. Le kérdezzük hogy hány különféle terméket rendeltek a gyártótól

Relációs algebra: $\pi_{COUNT (termek_id)}$

$\gamma_{COUNT (termek_id)}$ *gyarto*

Parancs:

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

6. Le kérdezzük hogy összesen mennyit van raktáron a gyártónak

Relációs algebra: $\pi_{SUM (raktaron)}$

$\gamma_{SUM (raktaron)}$ *gyarto*

Parancs:

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

7. Le kérdezzük hogy összesen mennyit kell legyártani a gyártónak

Relációs algebra: $\pi_{SUM (legyartanivalo)}$

$\gamma_{SUM (legyartanivalo)}$ *gyarto*

Parancs:

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

8. Le kérdezzük hogy összesen mennyit rendeltek a gyártótól

Relációs algebra: $\pi_{SUM(legyartanivalo) + SUM(raktaron)} gyarto$

Parancs:

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

9. Le kérdezzük hogy melyik klinikai igazgató, milyen terméket rendelt.

Relációs algebra: $\pi_{kliniak . igazgato, kiszallitoceg . csomagnev} (kliniak \bowtie_{kliniak . csomag_id = kiszallitoceg . csomag_id} kiszallitoceg)$

Parancs:

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

10. Le kérdezzük hogy melyik klinikai igazgató, milyen terméket rendelt, és hova.

Relációs algebra: $\pi_{kliniak_cim . cim, kliniak . igazgato, kiszallitoceg . csomagnev} (kliniak \bowtie_{kliniak . csomag_id = kiszallitoceg . csomag_id} kiszallitoceg \bowtie_{kliniak_cim . igazgato_id = kliniak . igazgato_id} klinikak_cim)$

Parancs:

```
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;
```

11. Le kérdezzük hogy melyik gyár, mit gyárt le és mennyit.

Relációs algebra: $\pi_{gyarto_gyar.gyar, kiszallitoceg.csomagnev, gyarto.legyartanivalo}$ (feladas $\bowtie_{feladas.csomag_id = kiszallitoceg.csomag_id}$ *kiszallitoceg* $\bowtie_{gyarto.termek_id = feladas.termek_id}$ *gyarto* $\bowtie_{gyarto_gyar.termek_id = gyarto.termek_id}$ *gyarto_gyar*)

Parancs:

```
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;
```

12. Le kérdezzük hogy a TEVA gyár mennyit és mit gyárt le.

Relációs algebra: $\pi_{gyarto_gyar.gyar, kiszallitoceg.csomagnev, gyarto.legyartanivalo}$
 $\sigma_{gyar = "TEVA"}$ (feladas $\bowtie_{feladas.csomag_id = kiszallitoceg.csomag_id}$ *kiszallitoceg* $\bowtie_{gyarto.termek_id = feladas.termek_id}$ *gyarto* $\bowtie_{gyarto_gyar.termek_id = gyarto.termek_id}$ *gyarto_gyar*)

Parancs:

```
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";
```

1.

| | Gyar | Csomagnev | Legyartanivalo |
|---|------|-----------|----------------|
| 1 | TEVA | Sebeszkes | 22 |
| 2 | TEVA | Kottszer | 105 |
| 3 | TEVA | Jod | 2 |
| 4 | TEVA | Opium | 30 |

2.

| | Eu_miniszter | Allamtitkar |
|---|---------------|--------------|
| 1 | Connor Taylor | Oscar Davies |

3.

| | Pu_miniszter | Allamtitkar |
|---|--------------|-----------------|
| 1 | Jim Hill | Cleveland Brown |

4.

| | Elnokok | Tisztseg_kep |
|---|----------------|---------------|
| 1 | George White | Oliver Smith |
| 2 | Jack Martin | Arthur Harris |
| 3 | Theo Baker | Jacob Harris |
| 4 | Theodore Baker | William Green |
| 5 | Theodore Clark | Max Baker |

5.

| | count(Termek_ID) |
|---|------------------|
| 1 | 8 |

6.

| | sum(Raktaron) |
|---|---------------|
| 1 | 242 |

7.

| | sum(Legyartanivalo) |
|---|---------------------|
| 1 | 243 |

8.

| | |
|---|-------------------------------------|
| | sum(Legyartanivalo) + sum(Raktaron) |
| 1 | 485 |

9.

| | Igazgato | Csomagnev |
|---|----------------|-----------|
| 1 | Ethan Murphy | Sebeszkes |
| 2 | Kyle Rodriguez | Kottszer |
| 3 | Michael Murphy | Jod |
| 4 | Ethan Murphy | Magnezium |
| 5 | Michael Murphy | Opium |
| 6 | Kyle Rodriguez | Morpium |
| 7 | Ethan Murphy | Klorofom |
| 8 | Michael Murphy | EKG |

10.

| | Cim | Igazgato | Csomagnev |
|---|-----------------------------------|----------------|-----------|
| 1 | 3000 Hatvan, Balassi Bálint út 16 | Ethan Murphy | Sebeszkes |
| 2 | 8230 Balatonfüred, Gyógy tér 2 | Kyle Rodriguez | Kottszer |
| 3 | 6000 Kecskemét, Nyíri út 38 | Michael Murphy | Jod |
| 4 | 3600 Ózd, Béke utca. 1-3 | Ethan Murphy | Magnezium |
| 5 | 6500 Baja, Rókus u.10 | Michael Murphy | Opium |
| 6 | 1106 Budapest, Maglódi út 89-91 | Kyle Rodriguez | Morpium |
| 7 | 2870 Kisbér, Iskola u. 11 | Ethan Murphy | Klorofom |
| 8 | 5700 Gyula, Semmelweis u. 1 | Michael Murphy | EKG |

11.

| | Gyar | Csomagnev | Legyartanivalo |
|---|-------|-----------|----------------|
| 1 | TEVA | Sebeszkes | 22 |
| 2 | TEVA | Kottszer | 105 |
| 3 | TEVA | Jod | 2 |
| 4 | BÉRES | Magnezium | 54 |
| 5 | TEVA | Opium | 30 |
| 6 | BÉRES | Morpium | 1 |
| 7 | BÉRES | Klorofom | 2 |
| 8 | BÉRES | EKG | 27 |

12.

| | Gyar | Csomagnev | Legyartanivalo |
|---|------|-----------|----------------|
| 1 | TEVA | Sebeszkes | 22 |
| 2 | TEVA | Kottszer | 105 |
| 3 | TEVA | Jod | 2 |
| 4 | TEVA | Opium | 30 |