**Jegyzőkönyv**

Adatbázisrendszerek I.

Féléves Feladat: Klinikai rendelés

Készítette: **Berki Viktor**  
Neptunkód: **ZY5P7F**  
Gyakorlatvezető: **Dr. Bednarik László**

**Miskolc, 2022**

Tartalomjegyzék

[1. A feladat leírása 3](#_Toc121189033)

[2. ER modell létrehozása 6](#_Toc121189034)

[3. ER modell konvertálása Relációs modellre 7](#_Toc121189035)

[4. Relációs séma létrehozása 8](#_Toc121189036)

[5. A adatbázis, táblák elkészítése 9](#_Toc121189037)

[6. Táblák feltötése 11](#_Toc121189038)

[7. Lekérdezések 17](#_Toc121189039)

# 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
* Pénzügyiminisztérium
* 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 lekérdéseket 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 Euminiszer\_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\_miniszerium tábla:  
  
Tartalmazza a minisztérium részvevőit, 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 tartalmazz a Termek\_ID mint kulcs és Foreign kulcs és a Gyar, ami a gyár nevét tartalmazza.

Feladas tábla:

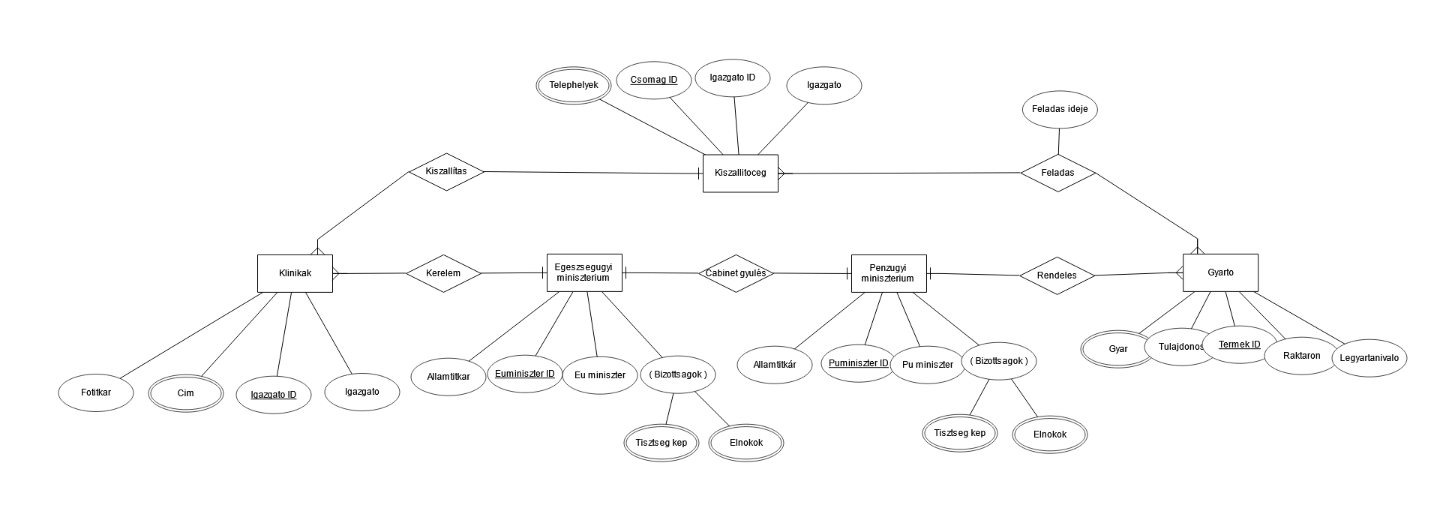
Ez egy n-m kapcsolat létre jötte miatt jött létre tábla.  
Ennek a táblának két foreign kulcs kell tartalmaznia és legalább egy tulajdonságnak. Itt a Termek\_ID és a Csomag\_ID nevét található, mint foreign kulcs és a Feladas\_ideje.  
  
Kiszallitoceg tábla:

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

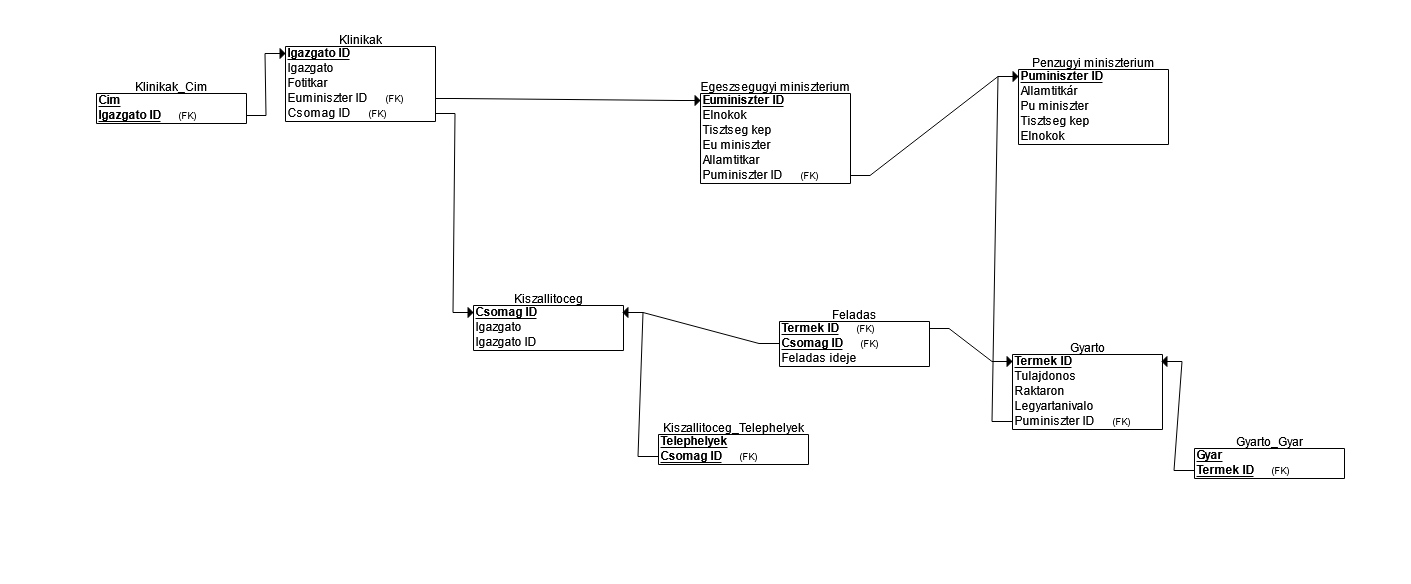
Kiszallitoceg tábla:  
  
Szintén több tulajdonság miatt jött létre.  
Itt található a Csomag\_ID mint foreign kulcs és a Telephelyek.

# 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 írjukbe.  
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.



# 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 hozzunk 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.

# 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]

# A adatbázis, táblák elkészítése

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

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

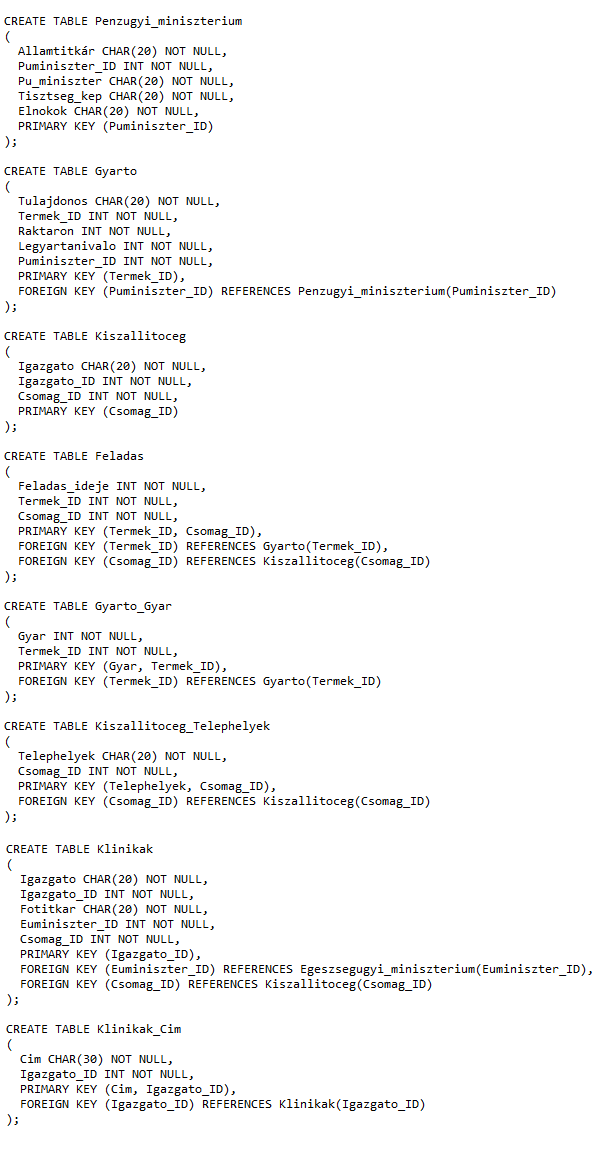
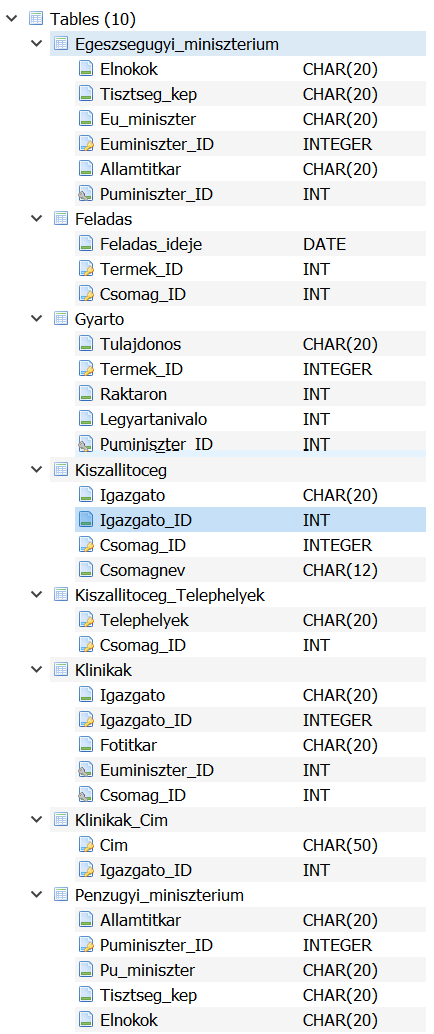
(

Tulajdonság neve adattípus kötelező\_cella\_kitöltés,

PRIMARY KEY (tábla kulcsa),

FOREIGN KEY (táblának a külföldi kulcs neve) REFERENCES az\_a\_tábla\_ahhonan\_származik(táblának a külföldi kulcs neve)

);



# Táblák feltötése

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(Allamtitkár,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)

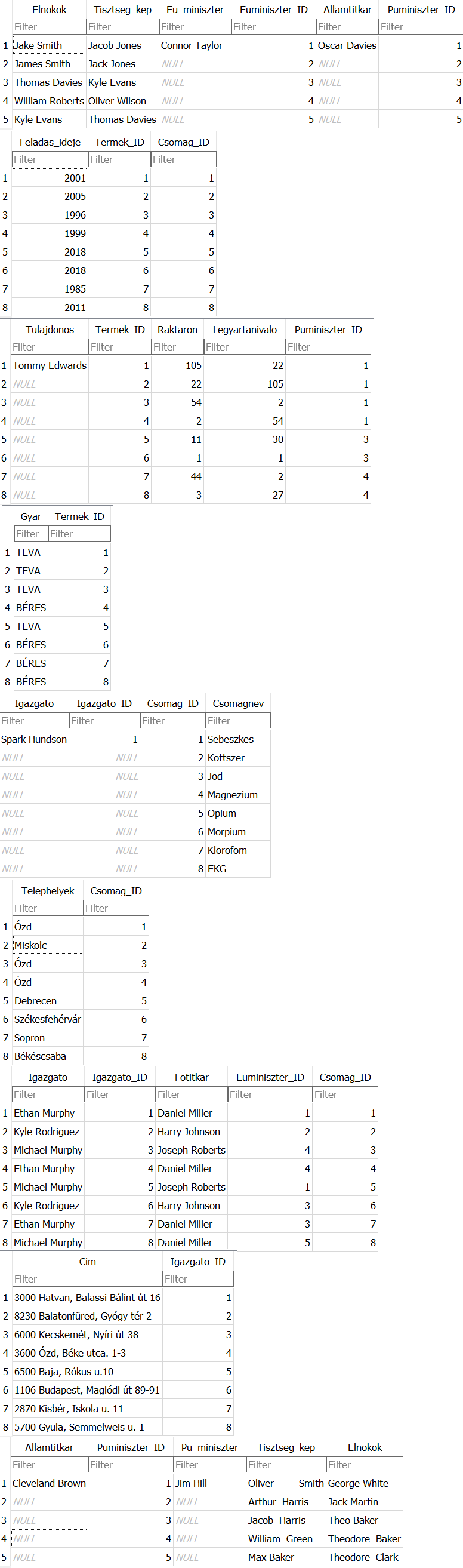
VALUES ('1106 Budapest, Maglódi út 89-91',6);

INSERT INTO Klinikak\_Cim(Cim,Igazgato\_ID)

VALUES ('2870 Kisbér, Iskola u. 11',7);

INSERT INTO Klinikak\_Cim(Cim,Igazgato\_ID)

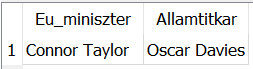
VALUES ('5700 Gyula, Semmelweis u. 1',8);

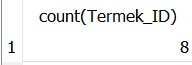
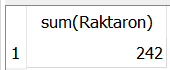
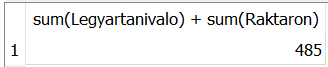


# Lekérdezések

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:** π elnokok, tisztseg\_kep egeszsegugyi\_miniszterium  
   **Parancs:**  
   SELECT Elnokok, Tisztseg\_kep  
   FROM Egeszsegugyi\_miniszterium;
2. **Le kérdezzük az egészségügyi minisztert és az államtitkárját az egészségügyi minisztériumban.  
   Relációs algebra:** π eu\_miniszter, allamtitkar   
    σ NOT (eu\_miniszter = "Not null") 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:** π pu\_miniszter, allamtitkar   
    σ 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:** π elnokok, tisztseg\_kep penzugyi\_miniszterium  
   **Parancs:**  
   SELECT Elnokok, Tisztseg\_kep  
   FROM Penzugyi\_miniszterium;
5. **Le kérdezzük hogy hány különféle terméket rendeltek a gyártótól  
   Relációs algebra:** π COUNT (termek\_id)   
    γ COUNT (termek\_id) gyarto  
   **Parancs:**  
   SELECT count(Termek\_ID)  
   FROM Gyarto;
6. **Le kérdezzük hogy összesen mennyit van raktáron a gyártónak  
   Relációs algebra:** π SUM (raktaron)   
    γ SUM (raktaron) gyarto  
   **Parancs:**  
   SELECT sum(Raktaron)  
   FROM Gyarto;
7. **Le kérdezzük hogy összesen mennyit kell legyártani a gyártónak  
   Relációs algebra:** π SUM (legyartanivalo)   
    γ SUM (legyartanivalo) gyarto  
   **Parancs:**  
   SELECT sum(Legyartanivalo)  
   FROM Gyarto;
8. **Le kérdezzük hogy összesen mennyit rendeltek a gyártótól  
   Relációs algebra:** π 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:** π klinikak . igazgato, kiszallitoceg . csomagnev (klinikak ⋈ klinikak . 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:** π klinikak\_cim . cim, klinikak . igazgato, kiszallitoceg . csomagnev (klinikak ⋈ klinikak . csomag\_id = kiszallitoceg . csomag\_id kiszallitoceg ⋈ klinikak\_cim . igazgato\_id = klinikak . 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:** π gyarto\_gyar . gyar, kiszallitoceg . csomagnev, gyarto . legyartanivalo (feladas ⋈ feladas . csomag\_id = kiszallitoceg . csomag\_id kiszallitoceg ⋈ gyarto . termek\_id = feladas . termek\_id gyarto ⋈ 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**: π gyarto\_gyar . gyar, kiszallitoceg . csomagnev, gyarto . legyartanivalo   
     σ gyar = "TEVA" (feladas ⋈ feladas . csomag\_id = kiszallitoceg . csomag\_id kiszallitoceg ⋈ gyarto . termek\_id = feladas . termek\_id gyarto ⋈ 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";
13. A képen asztal látható

    Automatikusan generált leírás
14. 
15. 
16. A képen asztal látható

    Automatikusan generált leírás
17. 
18. 
19. 
20. 
21. A képen asztal látható

    Automatikusan generált leírás
22. A képen asztal látható

    Automatikusan generált leírás
23. A képen asztal látható

    Automatikusan generált leírás
24. A képen asztal látható

    Automatikusan generált leírás