

# JEGYZŐKÖNYV

Adatbázis rendszerek I.

Féléves feladat

Egyetem

Készítette: Lovrity István

Neptunkód: HGQMMH

Dátum: 2024.11.

# Tartalomjegyzék

A feladat leírása: .....	3
1. feladat.....	4
1.1 Az adatbázis ER modell.....	4
1.2 Az ER modell konvertálása relációs modellre .....	5
1.3 Az adatbázis relációs séma .....	6
2. feladat: adatbázis létrehozása .....	7
2.1 Táblák létrehozása .....	7
2.2 Táblák feltöltése.....	9
3. feladat.....	13
3.1 Lekérdezések (SQL és relációs algebra) .....	13

## A feladat leírása:

Az alábbiakban bemutatásra kerül egy felsőoktatási intézmény számára készült adatbázis szerkezete, amely a hallgatók, képzések, tantárgyak, oktatók, mentorok és azok közötti kapcsolatok nyilvántartására szolgál. Az adatbázis lehetővé teszi a gyors és hatékony adatkezelést, az egyetemmel kapcsolatos adatok transzparens tárolását, valamint a különböző lekérdezések elvégzését.

A táblázatok megfelelő adatokkal való feltöltése után lehetőségünk van számos lekérdezést hatékonyan végrehajtani. Például hallgatók évfolyam szerinti csoportosítása, az oktatók által tanított tantárgyak listázása, vagy a képzésekhez tartozó tantárgyak megjelenítése. A lekérdezések és a kapcsolatok segítségével az adatbázis képes rugalmas és gyors válaszokat adni a felmerülő kérdésekre, legyen szó hallgatói információk kereséséről, vagy a tantárgyak elérhetőségének ellenőrzéséről.

Ez az adatbázis struktúra hatékonyan modellezi egy felsőoktatási intézmény különböző adatait és azok közötti kapcsolatokat. Az adatbázis lehetővé teszi a hallgatói, képzési és oktatási adatok gyors kezelését, és a különböző lekérdezések segítségével a szükséges információk könnyen hozzáférhetők.

Az adatbázis által biztosított kapcsolatokat és struktúrákat jól lehet alkalmazni az oktatási intézmények adminisztrációs folyamataiban, valamint a különböző oktatási és nyilvántartási feladatok során.

A séma el lett készítve szerkesztőprogram segítségével ER-modellben, ami utána át lett konvertálva relációs modellre. Fel van tüntetve minden egyed, az egyedek összes tulajdonsága, valamint meg van adva az azonosító kulcs, és a megfelelő modellben az idegen kulcs is.

Létre lett hozva SQL parancsok segítségével az adatbázis összes táblája, majd fel lettek töltve, és a kapcsolatokat reprezentáló referenciák csak utána lettek megadva. Erre lehetett volna több megoldás is akár, de az SQLite online-ban csak ilyen sorrendben lehet dolgozni a táblázatokkal. Minden adattípus következetesen lett megválasztva, és a méretük is megfelelően lett definiálva. A megfelelő mezők kapták az azonosító kulcs szerepet, ahol kellett ott meg lett adva az idegen kulcs is. A segéd táblák, avagy a gyenge egyedek, amik csak két táblázat összekapcsolására szolgálnak, nem rendelkeznek elsődleges kulccsal, csak két idegen kulccsal. Azok segítségével kapcsolnak össze két adott táblát.

A 10 SQL lekérdezés feladat folyamán törekedtem arra, hogy minél több típusú lekérdezés forduljon elő. Vannak sima, egytáblás lekérdezések, van több táblás lekérdezés, ott a közvetítő tábla segítségével lettek összekapcsolva, és utána lehetett adatokat lekérdezni, akár csoportosítás után, vagy akár 4-5 tábla összekapcsolása után. Relációs algebra formájában is le lett írva az összes lekérdezés

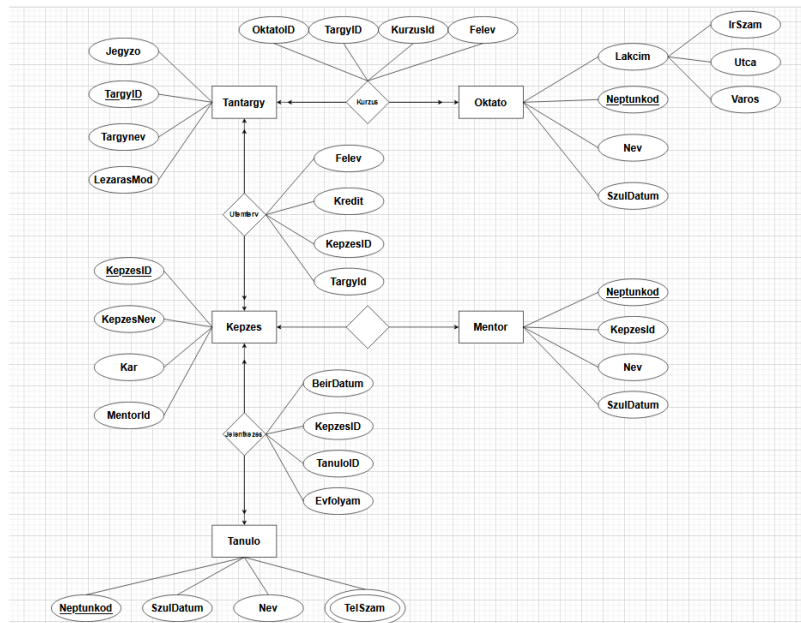
# 1. feladat

## 1.1 Az adatbázis ER modell

Az adatbázisban alkalmazott struktúra és táblák a következők alapján épülnek fel:

A Tanulo tábla a hallgatók adatait tárolja. Ez az entitás tartalmazza a hallgatók egyedi azonosítóját (Neptunkod), nevét, születési dátumát és telefonszámát. A Kepzes tábla a különböző képzéseket és azok jellemzőit rögzíti, ide tartozik a képzés azonosítója, ami egyben a public key, a kar neve és a mentor ID-ja. A Jelentkezes tábla a hallgatók képzésre történő jelentkezését tartalmazza, rögzítve a beiratkozás dátumát, a képzés

azonosítóját és az évfolyamot, amelyen a hallgató szerepel. Ez a tábla köti össze a Tanulo és a Kepzes egyedet, amelyek között M:N kapcsolat van.

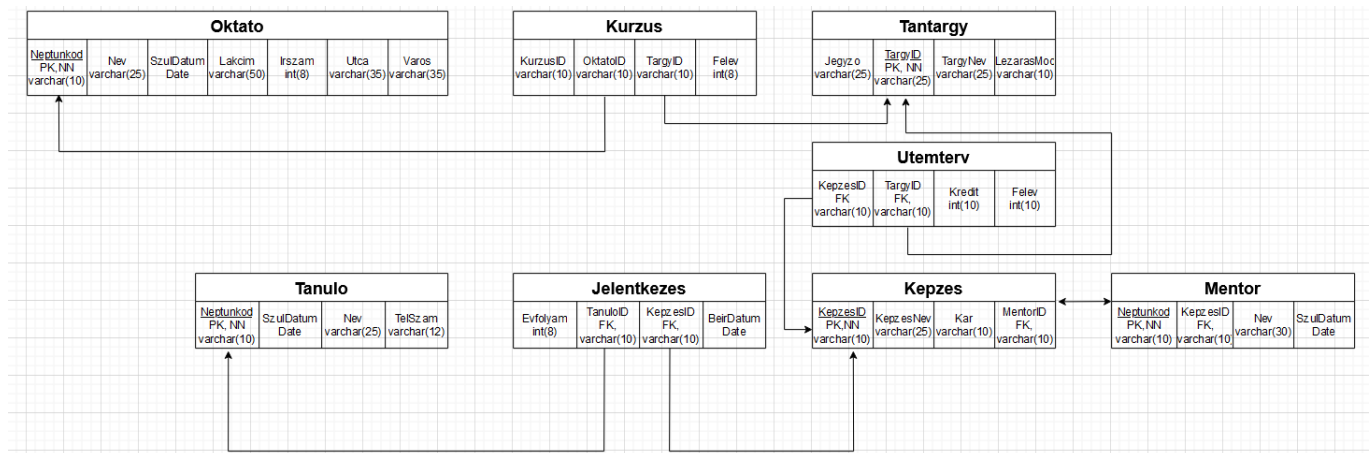


A Tantargy tábla a különböző tantárgyakat tartalmazza, minden tantárgyhoz hozzárendelve a tantárgy nevét, a jegyző oktatót, valamint a lezárás módját (például kollokvium vagy gyakorlat). Az Utemterv tábla kapcsolódik a képzésekhez és a tantárgyakhoz, tárolva a tantárgyak kreditértékét és a félévet, amelyben azok elérhetők. Az Utemterv tábla létesíti az M:N kapcsolatot a Tantargy és a Kepzes egyedek között. Az Oktato tábla az oktatók adatait tárolja, beleértve az oktató neptunkódját és egyben egyedi azonosítóját, nevét, születési dátumát és lakcímét. A Kurzus tábla az oktatókat és a tantárgyakat kapcsolja össze, M:N kapcsolatot biztosítva közöttük. A Mentor tábla az aktuális mentorokat tartalmazza. 1:1 kapcsolatban áll a képzés táblával, így nyomon tudjuk követni, hogy melyik szaknak ki a mentora, és melyik mentor melyik szakhoz tartozik.

A táblák között többféle kapcsolat valósul meg, amelyek az adatbázis integritását biztosítják. A Jelentkezes tábla például a hallgatók és a képzések közötti kapcsolatokat tárolja. Tartalmazza továbbá, hogy egyes tanulók melyik évben kezdték tanulmányaikat az adott képzésen, és mikor iratkoztak be. Az Utemterv a képzéseket és a tantárgyakat kapcsolja össze. Az oktatókat és tantárgyakat a Kurzus tábla kapcsolja, lehetővé téve a tantárgyak oktatójának nyomon követését. A Mentor és a Kepzes egyedek között 1:1 kapcsolat van, közük nem kellett segéd tábla M:N kapcsolat végett, csak egy idegen kulcs mindkét táblázatba, amivel egymásra tudnak hivatkozni.

A hallgatói adatok, képzések, tantárgyak és oktatók példáján keresztül szemléltethetjük, hogyan működik az adatbázis a gyakorlatban. Például a Tanulo tábla hallgatói adatokat tartalmaz, mint például a hallgató Neptunkódja és neve. A Kepzes tábla a különböző képzések neveit és az azokat vezető mentorok azonosítóit tárolja. Az Utemterv tábla a tantárgyak kreditértékeit és féléveit tartalmazza, míg a Kurzus tábla az oktatókat és a tantárgyakat kapcsolja össze, hogy nyomon követhessük, mely oktató mely tantárgyat tanít.

## 1.2 Az ER modell konvertálása relációs modellre

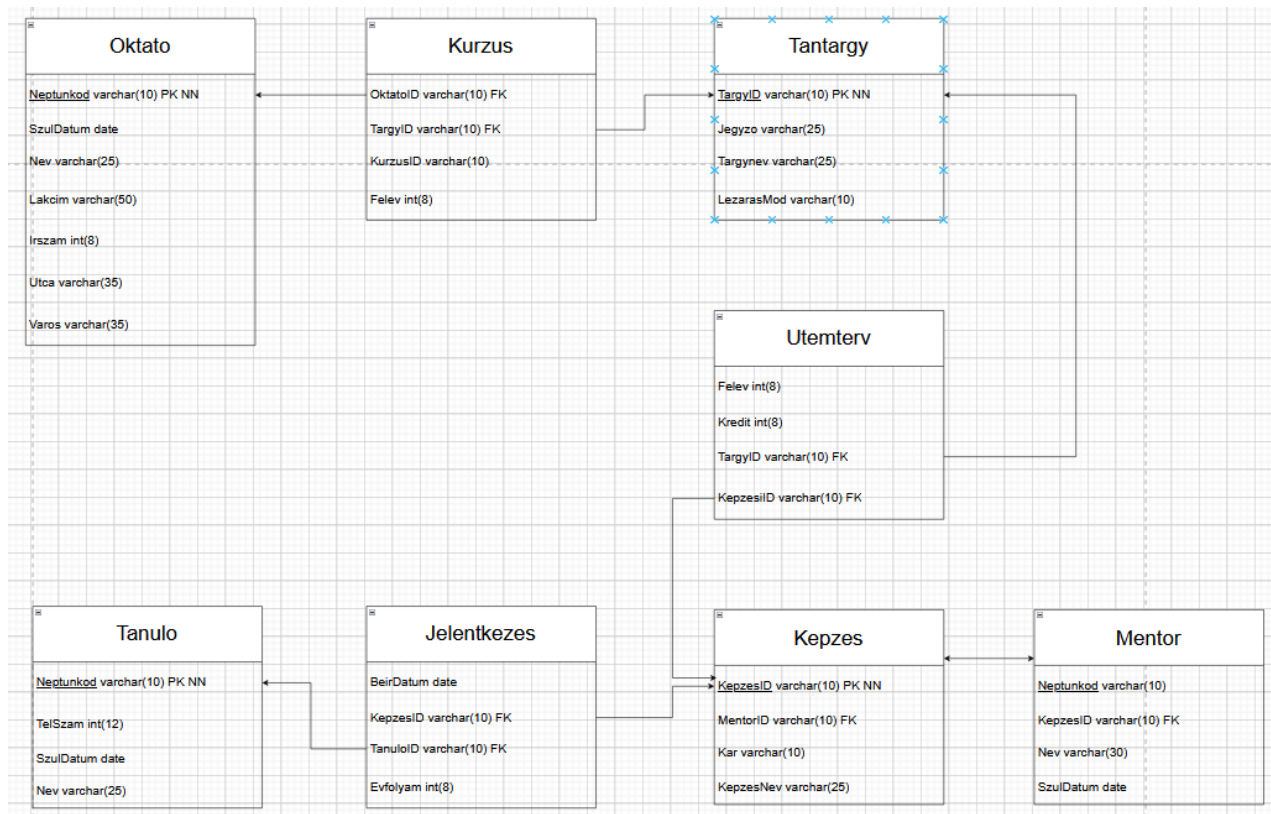


A már létező ER modell lett átkonvertálva relációs modellre (RM modell), ahogy a képen is látható. Lényegében ugyan az az ábra, csak más szerkezetben lett ábrázolva, valamint több információt tartalmaz az egyedekről és tulajdonságaikról. Minden táblának ugyan úgy fel van tüntetve a neve és a tulajdonságaik neve, mint eddig, valamint a tulajdonságokról bővebb információk lettek megadva.

Az elsődleges kulcsok ugyan úgy alá vannak húzva, valamint az idegen kulcsok is külön fel lettek tüntetve. Abból is látszanak ezek, ahogyan össze vannak kötve az egyedek. Ahol segéd tábla található - avagy gyenge egyed mivel nincs elsődleges kulcsa, csak két idegen kulcsa, amivel össze köt két másik táblázatot – lehet látni, hogy ilyen esetben egy idegen kulcs mindig egy másik egyed elsődleges kulcsára mutat. A Mentor-Kepzes kapcsolat az másabb eset, mivel az 1:1 kapcsolat, nem pedig N:M mint a többi. Fel lettek tüntetve ezek mellett az egyedek tulajdonságainak adattípusai, valamint méretük egyaránt.

Amit még érdemes megfigyelni, hogy ez a modell táblázat formában tartalmazza az egyedeket és tulajdonságaikat. Felül van az egyed neve, alatta pedig egymástól horizontálisan elhelyezkedve a tulajdonságok.

### 1.3 Az adatbázis relációs séma



A relációs séma szerkezetében nagyon hasonlít a relációs modellhez. Ugyan úgy táblázat formájában vannak feltüntetve az egyedek, legfelül az egyedek neve, alatta pedig a tulajdonságok, minden hozzájuk tartozó információt is feltüntetve, mint az adattípus, méret, vagy éppen kulcs. Ebben az esetben viszont a tulajdonságok egymás alatt helyezkednek el.

A kapcsolatok szintén ugyan úgy vannak feltüntetve, az idegen kulcsok a másik táblában lévő, hozzájuk tartozó elsődleges kulcsra mutatnak M:N kapcsolat esetében. 1:1 kapcsolat esetén pedig ugyan úgy egymásra mutatnak, és kölcsönösen létre lett hozva egymásra mutató idegen kulcsok, mint tulajdonságok.

A következő feladatrészben SQL formátumban lesz bemutatva minden itt, valamint korábban látható feladatrész.

## 2. feladat: adatbázis létrehozása

### 2.1 Táblák létrehozása

```
CREATE TABLE Tanulo (  
    Neptunkod varchar(10) PRIMARY KEY NOT NULL,  
    TelSzam varchar(12),  
    SzulDatum date,  
    Nev varchar(25));  
  
CREATE TABLE Kepzes (  
    KepzesID varchar(10) PRIMARY KEY NOT NULL,  
    MentorID varchar(10),  
    Kar varchar(10),  
    KepzesNev varchar(25));  
  
CREATE TABLE Jelentkezes (  
    BeirDatum date,  
    KepzesID varchar(10),  
    TanuloID varchar(10),  
    Evfolyam int(8));  
  
CREATE TABLE Tantargy (  
    TargyID varchar(10) PRIMARY KEY NOT NULL,  
    Jegyzo varchar(25),  
    TargyNev varchar(25),  
    LezarasMod varchar(10));  
  
CREATE TABLE Utemterv (  
    Felev int(8),  
    Kredit int(8),  
    KepzesID varchar(10),  
    TargyID varchar(10));  
  
CREATE TABLE Oktato (  
    Neptunkod varchar(10) PRIMARY KEY NOT NULL,  
    SzulDatum date,  
    Nev varchar(25),  
    Lakcim varchar(50),  
    Irszam int(8),  
    Utca varchar(35),  
    Varos varchar(35));
```

```
CREATE TABLE Kurzus (
```

```
    OktatoID varchar(10),  
    TargyID varchar(10),  
    KurzusID varchar(10),  
    Felev int(8));
```

```
CREATE TABLE Mentor (
```

```
    Neptunkod varchar(10) PRIMARY KEY NOT NULL,  
    KepzesID varchar(10),  
    Nev varchar(30),  
    SzulDatum date);
```

Mindegyik tábla ugyan úgy, a CREATE TABLE paranccsal került létrehozásra. Létre lett hozva az 5 egyed tábla, valamint a 3 segédtábla, amik az egyedek közötti kapcsolat végett voltak szükségesek. Ennek a 3 táblázatnak, név szerint Kurzus, Utemterv, Jelentkezés, nincs elsődleges kulcsuk szóval gyenge egyedek.

A többi táblában meg lett adva az azonosító kulcs, valamint hogy azok nem vehetnek fel null értéket. Karakterlánc (string-varchar), egész szám (int), valamint dátum (date) típusú tulajdonságokat adtam az egyedeknek, a méretekre külön figyeltem, hogy ne legyen túlszordulás.



## 2.2 Táblák feltöltése

```
INSERT INTO Tanulo (Neptunkod, TelSzam, SzulDatum, Nev) VALUES
('QWERTZ', '06707896321', '2001-05-15', 'Kovács István'),
('ASDFGH', '06703214789', '2002-07-21', 'Nagy Beáta'),
('FGH4J5', '06305828796', '2000-12-03', 'Szabó Bence'),
('RFV5BH', '06204569831', '1999-03-11', 'Tóth Ottó'),
('MNO5L6', '06704719673', '1998-08-29', 'Fekete Viktor'),
('PQRPO9', '06304379842', '2003-10-14', 'Varga Attila');
```

Neptunkod	TelSzam	SzulDatum	Nev
ASDFGH	06703214789	2002-07-21	Nagy Beáta
FGH4J5	06305828796	2000-12-03	Szabó Bence
MNO5L6	06704719673	1998-08-29	Fekete Viktor
PQRPO9	06304379842	2003-10-14	Varga Attila
QWERTZ	06707896321	2001-05-15	Kovács István
RFV5BH	06204569831	1999-03-11	Tóth Ottó

```
INSERT INTO Kepzes (KepzesID, MentorID, Kar, KepzesNev) VALUES
('GEIK1', 'M4', 'GEIK', 'Informatika'),
('GTK1', 'M1', 'GTK', 'Adatelemzes'),
('BTK1', 'M6', 'BTK', 'Gyogypedagogia'),
('GTK2', 'M2', 'GTK', 'Szamvitel'),
('GEIK2', 'M3', 'GEIK', 'Villamosmernok'),
('ETK1', 'M5', 'ETK', 'Gyogytestnevelés');
```

KepzesID	MentorID	Kar	KepzesNev
BTK1	M6	BTK	Gyogypedagogia
ETK1	M5	ETK	Gyogytestnevelés
GEIK1	M4	GEIK	Informatika
GEIK2	M3	GEIK	Villamosmernok
GTK1	M1	GTK	Adatelemzes
GTK2	M2	GTK	Szamvitel

```
INSERT INTO Jelentkezes (BeirDatum, KepzesID, TanuloID, Evfolyam) VALUES
('2023-09-10', 'GEIK1', 'MNO5L6', 2023),
('2022-08-15', 'ETK1', 'QWERTZ', 2022),
('2024-09-05', 'GTK1', 'RFV5BH', 2024),
('2022-09-08', 'ETK1', 'ASDFGH', 2022),
('2017-08-29', 'ETK1', 'PQRPO9', 2017),
('2023-09-01', 'GEIK2', 'FGH4J5', 2023);
```

BeirDatum	KepzesID	TanuloID	Evfolyam
2023-01-10	GEIK1	MNO5L6	2023
2023-01-15	ETK1	QWERTZ	2022
2022-09-05	GTK1	RFV5BH	2024
2022-09-08	ETK1	ASDFGH	2020
2021-02-12	ETK1	PQRPO9	2017
2021-05-20	GEIK2	FGH4J5	2023

```
INSERT INTO Tantargy (TargyID, Jegyzo, TargyNev, LezarasMod) VALUES
('T1', 'Kiss Anita', 'Matematika', 'Kollokvium'),
('T2', 'Nagy Árpád', 'Fizika', 'Gyakorlat'),
('T3', 'Szabó Teréz', 'Biológia', 'Gyakorlat'),
('T4', 'Fekete István', 'Kémia', 'Kollokvium'),
('T5', 'Pető Vilmos', 'Informatika', 'Gyakorlat'),
('T6', 'Magyar Gyula', 'Gépészet', 'Kollokvium');
```

TargyID	Jegyzo	TargyNev	LezarasMod
T1	Kiss Anita	Matematika	Kollokvium
T2	Nagy Árpád	Fizika	Gyakorlat
T3	Szabó Teréz	Biológia	Gyakorlat
T4	Fekete István	Kémia	Kollokvium
T5	Pető Vilmos	Informatika	Gyakorlat
T6	Magyar Gyula	Gépészet	Kollokvium

```
INSERT INTO Utemterv (Felev, Kredit, KepzesID, TargyID) VALUES
(1, 4, 'GEIK1', 'T5'),
(2, 5, 'GTK1', 'T1'),
(3, 4, 'ETK1', 'T3'),
(4, 3, 'GTK2', 'T1'),
(5, 5, 'GTK2', 'T5'),
(6, 2, 'BTK1', 'T3');
```

Felev	Kredit	KepzesID	TargyID
1	4	GEIK1	T5
2	5	GTK1	T1
3	4	ETK1	T3
4	3	GTK2	T1
5	5	GTK2	T5
6	2	BTK1	T3

```
INSERT INTO Oktato (Neptunkod, SzulDatum, Nev, Lakcim, Irszam, Utca, Varos) VALUES
('QUKRTH', '1980-01-12', 'Kiss János', '1011 Budapest Alkotás u. 11', 1011, 'Alkotás u.', 'Budapest'),
('LFTHGQ', '1975-03-22', 'Nagy Éva', '4025 Debrecen Hunyadi u. 12', 4025, 'Hunyadi u.', 'Debrecen'),
('PLAQ3D', '1985-09-18', 'Szabó Péter', '6000 Kecskemét Petőfi tér 2', 6000, 'Petőfi tér', 'Kecskemét'),
('SLJ3V2', '1978-07-09', 'Tóth Erika', '7621 Pécs Diófa u. 23', 7621, 'Diófa u.', 'Pécs'),
('QWLLJ1', '1982-12-30', 'Varga Attila', '9400 Sopron Kossuth Lajos u. 10', 9400, 'Kossuth Lajos u.', 'Sopron'),
('QQMJH3', '1990-05-16', 'Fekete Eszter', '8000 Székesfehérvár Bartók Béla u. 30', 8000, 'Bartók Béla u.', 'Székesfehérvár');
```

Neptunkod	SzulDatum	Nev	Lakcim	Irszam	Utca	Varos
LFTHGQ	1975-03-22	Nagy Éva	4025 Debrecen...	4025	Hunyadi u.	Debrecen
PLAQ3D	1985-09-18	Szabó Péter	6000 Kecskem...	6000	Petőfi tér	Kecskemét
QQMJH3	1990-05-16	Fekete Eszter	8000 Székesfe...	8000	Bartók Béla u.	Székesfehérvár
QUKRTH	1980-01-12	Kiss János	1011 Budapest ...	1011	Alkotás u.	Budapest
QWLLJ1	1982-12-30	Varga Attila	9400 Sopron K...	9400	Kossuth Lajos u.	Sopron
SLJ3V2	1978-07-09	Tóth Erika	7621 Pécs Dióf...	7621	Diófa u.	Pécs

```
INSERT INTO Kurzus (OktatoID, TargyID, KurzusID, Felev) VALUES
('QUKRTH', 'T1', 'K1', 1),
('QQMJH3', 'T3', 'K2', 2),
('SLJ3V2', 'T3', 'K3', 3),
('LFTHGQ', 'T6', 'K4', 4),
('SLJ3V2', 'T1', 'K5', 5),
('QQMJH3', 'T5', 'K6', 6);
```

OktatoID	TargyID	KurzusID	Felev
QUKRTH	T1	K1	1
QQMJH3	T3	K2	2
SLJ3V2	T3	K3	3
LFTHGQ	T6	K4	4
SLJ3V2	T1	K5	5
QQMJH3	T5	K6	6

```
INSERT INTO Mentor (Neptunkod, KepzesID, Nev, SzulDatum) VALUES
('M1', 'GTK1', 'Kovács Lajos', '1973-02-15'),
('M2', 'GTK2', 'Nagy Tímea', '1980-06-22'),
('M3', 'GEIK2', 'Szabó Gábor', '1985-11-03'),
('M4', 'GEIK1', 'Tóth Zsuzsanna', '1977-01-14'),
('M5', 'ETK1', 'Fekete István', '1968-08-30'),
('M6', 'BTK1', 'Varga Klára', '1990-09-05');
```

Neptunkod	KepzesID	Nev	SzulDatum
M1	GTK1	Kovács Lajos	1973-02-15
M2	GTK2	Nagy Tímea	1980-06-22
M3	GEIK2	Szabó Gábor	1985-11-03
M4	GEIK1	Tóth Zsuzsanna	1977-01-14
M5	ETK1	Fekete István	1968-08-30
M6	BTK1	Varga Klára	1990-09-05

```
ALTER TABLE Jelentkezes  
ADD FOREIGN KEY (kepzesid) REFERENCES Kepzes(kepzesid);
```

```
ALTER TABLE Jelentkezes  
ADD FOREIGN KEY (tanuloid) REFERENCES Tanulo(neptunkod);
```

```
ALTER TABLE Utemterv  
ADD FOREIGN KEY (kepzesid) REFERENCES Kepzes(kepzesid);
```

```
ALTER TABLE Utemterv  
ADD FOREIGN KEY (targyid) REFERENCES Tantargy(targyid);
```

```
ALTER TABLE Kurzus  
ADD FOREIGN KEY (oktatoid) REFERENCES Oktato(neptunkod);
```

```
ALTER TABLE Kurzus  
ADD FOREIGN KEY (targyid) REFERENCES Tantargy(targyid);
```

```
ALTER TABLE Kepzes  
ADD FOREIGN KEY (mentoid) REFERENCES Mentor(neptunkod);
```

```
ALTER TABLE Mentor  
ADD FOREIGN KEY (kepzesid) REFERENCES Kepzes(kepzesid);
```

Ebben a feladatrészben manuálisan meg lett adva minden adat, természetesen az adattípusnak megfelelően. Ezt az INSERT INTO paranccsal kezdeményeztem, majd zárójelben megadtam az oszlopok nevét.

Ezután szabványosan be lett írva minden adat, amiket le is lehet kérdezni tetszés szerint. Ahogyan korábban írtam is, az SQLite nem engedte, hogy két kapcsolatban lévő tábla tartalmát módosítsam, emiatt ennek a feladatnak a végén kapcsoltam össze a táblákat. Ezt az ALTER TABLE, majd az ADD FOREIGN KEY és REFERENCES kulcsszavakkal valósítottam meg. Ezzel azt értem el, hogy először megmondtam, hogy módosítani fogom az adott táblázatot, majd megadtam, hogy melyik mező legyen az idegen kulcs, és melyik másik táblázat, melyik bizonyos tulajdonságára hivatkozzon.

## 3. feladat

### 3.1 Lekérdezések (SQL és relációs algebra)

- Kérjünk le minden mezőt a Kepzes táblából, ahol a Kar 'TK'-ra végződik.

- `SELECT * FROM Kepzes WHERE Kepzes.Kar LIKE '%TK';`
- $\sigma_{\text{kepzes.kar LIKE "%TK"}} \text{kepzes}$

KepzesID	MentorID	Kar	KepzesNev
BTK1	M6	BTK	Gyogypedagogia
ETK1	M5	ETK	Gyogytestnevelés
GTK1	M1	GTK	Adatelemzés
GTK2	M2	GTK	Számvitel

- A Tantargy táblát csoportosítsuk a Lezárás módja szerint, és kérdezzük le a Lezárás módját, valamint hogy hány tantárgy van lezárás módonként (Num).

- `SELECT Tantargy.LezarasMod, COUNT(*) AS Num FROM Tantargy GROUP BY Tantargy.LezarasMod;`
- $\pi_{\text{tantargy.lezarasmod, COUNT(*)} \rightarrow \text{num}} \gamma_{\text{lezarasmod, COUNT(*)} \text{tantargy}}$

LezarasMod	Num
Gyakorlat	3
Kollokvium	3

- Kérdezzük le, hogy az egyes tanulók melyik képzésen indultak, és mikor iratkoztak be.
  - `SELECT Tanulo.Nev, Kepzes.KepzesNev, Jelentkezes.BeirDatum FROM Jelentkezes INNER JOIN Tanulo ON Tanulo.Neptunkod = Jelentkezes.TanuloID INNER JOIN Kepzes ON Kepzes.KepzesID = Jelentkezes.KepzesID;`
  - $\pi \text{ tanulo . nev, kepzes . kepzesnev, jelentkezes . beirdatum (jelentkezes } \bowtie \text{ tanulo . neptunkod = jelentkezes . tanuloid tanulo } \bowtie \text{ kepzes . kepzeseid = jelentkezes . kepzeseid kepzes)}$

Nev	KepzesNev	BeirDatum
Fekete Viktor	Informatika	2023-01-10
Kovács István	Gyógytestnevelés	2023-01-15
Tóth Ottó	Adatelemzés	2022-09-05
Nagy Beáta	Gyógytestnevelés	2022-09-08
Varga Attila	Gyógytestnevelés	2021-02-12
Szabó Bence	Villamosmernok	2021-05-20

- Kérdezzük le, hogy az egyes tanulók melyik képzésen indultak, és ez a képzés melyik karon van.
  - `SELECT Tanulo.Nev, Kepzes.KepzesNev, Kepzes.Kar FROM Jelentkezes INNER JOIN Tanulo ON Tanulo.Neptunkod = Jelentkezes.TanuloID INNER JOIN Kepzes ON Kepzes.KepzesID = Jelentkezes.KepzesID;`
  - $\pi \text{ tanulo . nev, kepzes . kepzesnev, kepzes . kar (jelentkezes } \bowtie \text{ tanulo . neptunkod = jelentkezes . tanuloid tanulo } \bowtie \text{ kepzes . kepzeseid = jelentkezes . kepzeseid kepzes)}$

Nev	KepzesNev	Kar
Fekete Viktor	Informatika	GEIK
Kovács István	Gyógytestnevelés	ETK
Tóth Ottó	Adatelemzés	GTK
Nagy Beáta	Gyógytestnevelés	ETK
Varga Attila	Gyógytestnevelés	ETK
Szabó Bence	Villamosmernok	GEIK

- Kérdezzük le, hogy évfolyamonként hány tanuló jelentkezett képzésre.
  - `SELECT Jelentkezes.Evfolyam, COUNT(*) AS Num FROM Jelentkezes INNER JOIN Tanulo ON Tanulo.Neptunkod = Jelentkezes.TanuloID INNER JOIN Kepzes ON Kepzes.KepzesID = Jelentkezes.KepzesID GROUP BY Jelentkezes.Evfolyam;`
  - $\pi$  jelentkezes . evfolyam, COUNT (\*)  $\rightarrow$  num  
 $\gamma$  evfolyam, COUNT (\*) (jelentkezes  $\bowtie$  tanulo . neptunkod = jelentkezes . tanuloID  
 tanulo  $\bowtie$  kepzes . kepzesid = jelentkezes . kepzesid kepzes)

Evfolyam	Num
2017	1
2020	1
2022	1
2023	2
2024	1

- Listázzuk ki, hogy a 'QQMJH3' neptunkóddal rendelkező tanár milyen tárgyakat tanít.
  - `SELECT Oktato.Nev, Tantargy.TargyNev FROM Kurzus INNER JOIN Oktato ON Kurzus.OktatoID = Oktato.Neptunkod INNER JOIN Tantargy ON Kurzus.TargyID = Tantargy.TargyID WHERE Kurzus.OktatoID = 'QQMJH3';`
  - $\pi$  oktato . nev, tantargy . targynev  
 $\sigma$  kurzus . oktatoID = "QQMJH3" (kurzus  $\bowtie$  kurzus . oktatoID = oktato . neptunkod  
 oktato  $\bowtie$  kurzus . targyid = tantargy . targyid tantargy)

Nev	TargyNev
Fekete Eszter	Biológia
Fekete Eszter	Informatika

- Listázzuk ki, hogy melyik képzéshez milyen tantárgyak tartoznak.
  - `SELECT Kepzes.KepzesNev, Tantargy.TargyNev FROM Utemterv INNER JOIN Kepzes ON Kepzes.KepzesID = Utemterv.KepzesID INNER JOIN Tantargy ON Utemterv.TargyID = Tantargy.TargyID;`
  - $\pi \text{ kepzes} . \text{kepzesnev}, \text{tantargy} . \text{targynev} (\text{utemterv} \bowtie \text{kepzes} . \text{kepzesid} = \text{utemterv} . \text{kepzesid} \text{ kepzes} \bowtie \text{utemterv} . \text{targyid} = \text{tantargy} . \text{targyid} \text{ tantargy})$

KepzesNev	TargyNev
Informatika	Informatika
Adatelemzes	Matematika
Gyogytestnevelés	Biológia
Számvitel	Matematika
Számvitel	Informatika
Gyogypedagogia	Biológia

- Listázzuk ki az összes mentor nevét, és a hozzájuk tartozó képzést.
  - `SELECT Mentor.Nev, Kepzes.KepzesNev FROM Mentor INNER JOIN Kepzes ON Kepzes.MentorID = Mentor.Neptunkod;`
  - $\pi \text{ mentor} . \text{nev}, \text{kepzes} . \text{kepzesnev} (\text{mentor} \bowtie \text{kepzes} . \text{mentorid} = \text{mentor} . \text{neptunkod} \text{ kepzes})$

Nev	KepzesNev
Varga Klára	Gyogypedagogia
Fekete István	Gyogytestnevelés
Tóth Zsuzsanna	Informatika
Szabó Gábor	Villamosmernok
Kovács Lajos	Adatelemzes
Nagy Tímea	Számvitel



- Listázzuk ki, hogy az egyes oktatók mely karokon tanítanak.
  - SELECT Oktato.Nev AS OktNev, Kepzes.Kar AS Kar FROM Oktato INNER JOIN Kurzus ON Oktato.Neptunkod = Kurzus.OktatoID INNER JOIN Tantargy ON Kurzus.TargyID = Tantargy.TargyID INNER JOIN Utemterv ON Tantargy.TargyID = Utemterv.TargyID INNER JOIN Kepzes ON Utemterv.KepzesID = Kepzes.KepzesID GROUP BY Oktato.Nev, Kepzes.Kar ORDER BY Oktato.Nev;
  - $\tau$  oktato . nev  
 $\pi$  oktato . nev  $\rightarrow$  oktnév, kepzes . kar  $\rightarrow$  kar  
 $\gamma$  nev, kar, (oktato  $\bowtie$  oktato . neptunkod = kurzus . oktatoid kurzus  $\bowtie$  kurzus . targyid = tantargy . targyid tantargy  $\bowtie$  tantargy . targyid = utemterv . targyid utemterv  $\bowtie$  utemterv . kepzesid = kepzes . kepzesid kepzes)

OktNev	Kar
Fekete Eszter	BTK
Fekete Eszter	ETK
Fekete Eszter	GEIK
Fekete Eszter	GTK
Kiss János	GTK
Tóth Erika	BTK
Tóth Erika	ETK
Tóth Erika	GTK

- Listázzuk ki az összes tanuló nevét, aki 'GEIK' karon lévő képzésre jelentkezett.
  - SELECT Nev AS TanuloNev FROM Tanulo WHERE Neptunkod IN ( SELECT TanuloID FROM Jelentkezes WHERE KepzesID IN ( SELECT KepzesID FROM Kepzes WHERE Kar = 'GEIK'));
  - $\pi$ Nev( $\sigma$ Neptunkod ( $\pi$ TanuloID( $\sigma$ KepzesID ( $\pi$ KepzesID( $\sigma$ Kar='GEIK' (Kepzes))(Jelentkezes))))(Tanulo))

TanuloNev
Szabó Bence
Fekete Viktor