

Adatbázis Rendszerek I. - Féléves Feladat  
Nagy Áron – USVGM7

# Bevezetés

A téma egy földrajzi-társadalmi adatbázis, ami az alábbi egyedeket tartalmazza:

## Ország

**Ország\_ID** ez az egyed *elsődleges kulcsa*.

*Tulajdonságai:* **Országnév, Hivatalos nyelv, Néesség, Terület, Vallás**

## Régió

**Régió\_ID** ez az egyed *elsődleges kulcsa*.

*Tulajdonságai:* **Régiónév, Néesség, Terület**

## Megye

**Megye\_ID** ez az egyed *elsődleges kulcsa*.

*Tulajdonságai:* **Megyenév, Néesség, Terület, Gazdaság**

## Város

**Város\_ID** ez az egyed *elsődleges kulcsa*

*Tulajdonságai:* **Városnév, Néesség, Terület, Alapítás éve, Polgármester** utóbbi egy *összetett tulajdonság* ami a **Vezetéknév, Keresztnév** tulajdonságokból áll elő.

## Városrész

**Városrész\_ID** ez az egyed *elsődleges kulcsa*

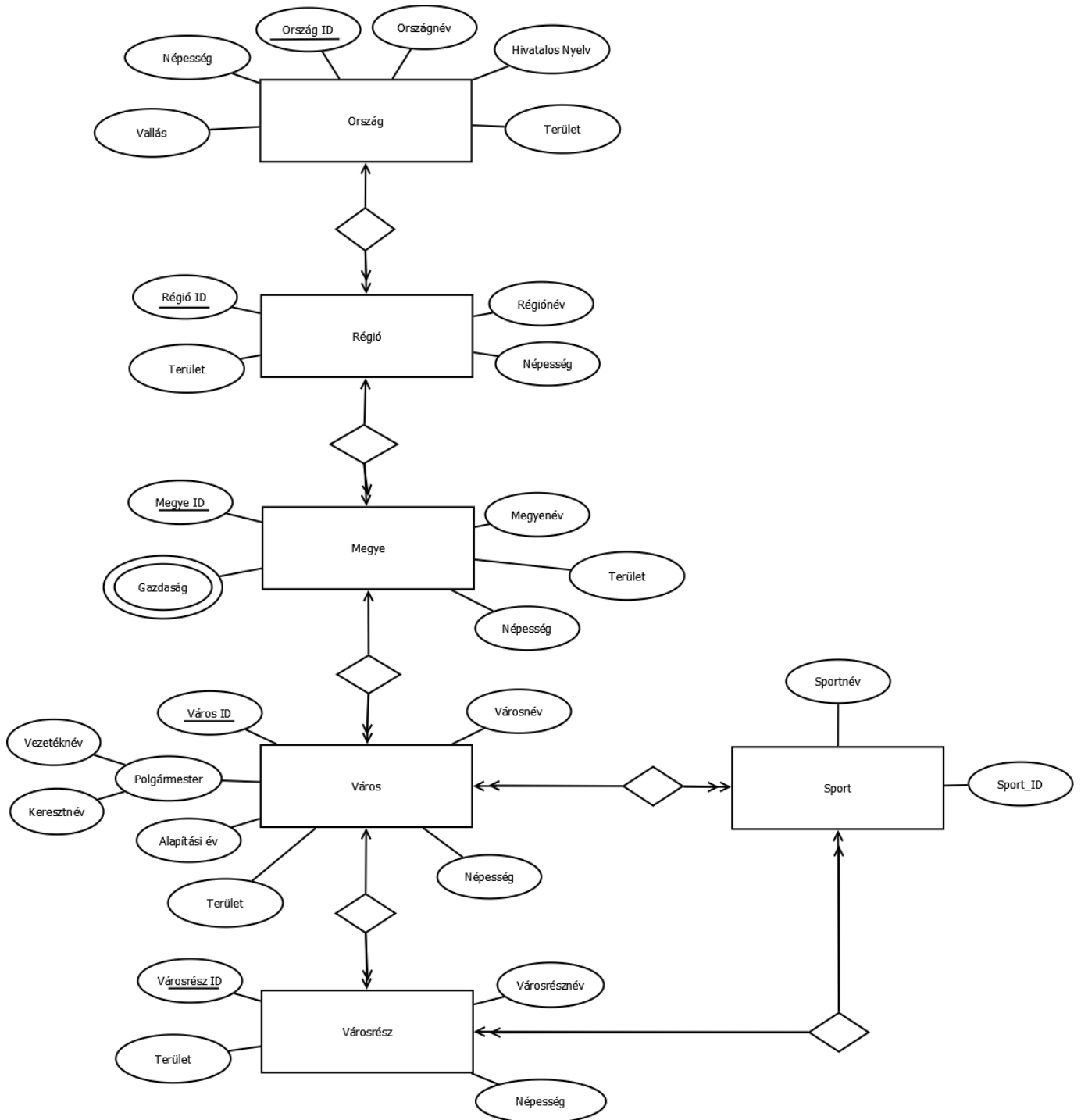
*Tulajdonságai:* **Városrésznév, Néesség, Terület**

## Sport

**Sport\_ID** ez az egyed *elsődleges kulcsa*

*Tulajdonságai:* **Sportnév, Néesség, Terület**

# Az ER-modell



# Konvertáció ER-Modellből Relációs Modellbe

Tábla képződik az **Ország** egyedből. A tulajdonságok változatlanul megmaradnak.

Tábla képződik a **Régió** egyedből. A tulajdonságok változatlanul megmaradnak. A tábla kiegészül egy idegen kulccsal (**Ország\_ID**) amely az **Ország** tábla elsődleges kulcsát tartalmazza.

Tábla képződik a **Megye** egyedből. A tulajdonságok változatlanul megmaradnak. A tábla kiegészül egy idegen kulccsal (**Régió\_ID**) amely a **Régió** tábla elsődleges kulcsát tartalmazza.

Tábla képződik a **Város** egyedből. A tulajdonságok változatlanul megmaradnak kivéve a **Polgármestert** amelyet a **Vezetéknév** és **Keresztnév** tulajdonságok határoznak meg. A tábla kiegészül egy idegen kulccsal (**Megye\_ID**) amely a **Megye** tábla elsődleges kulcsát tartalmazza.

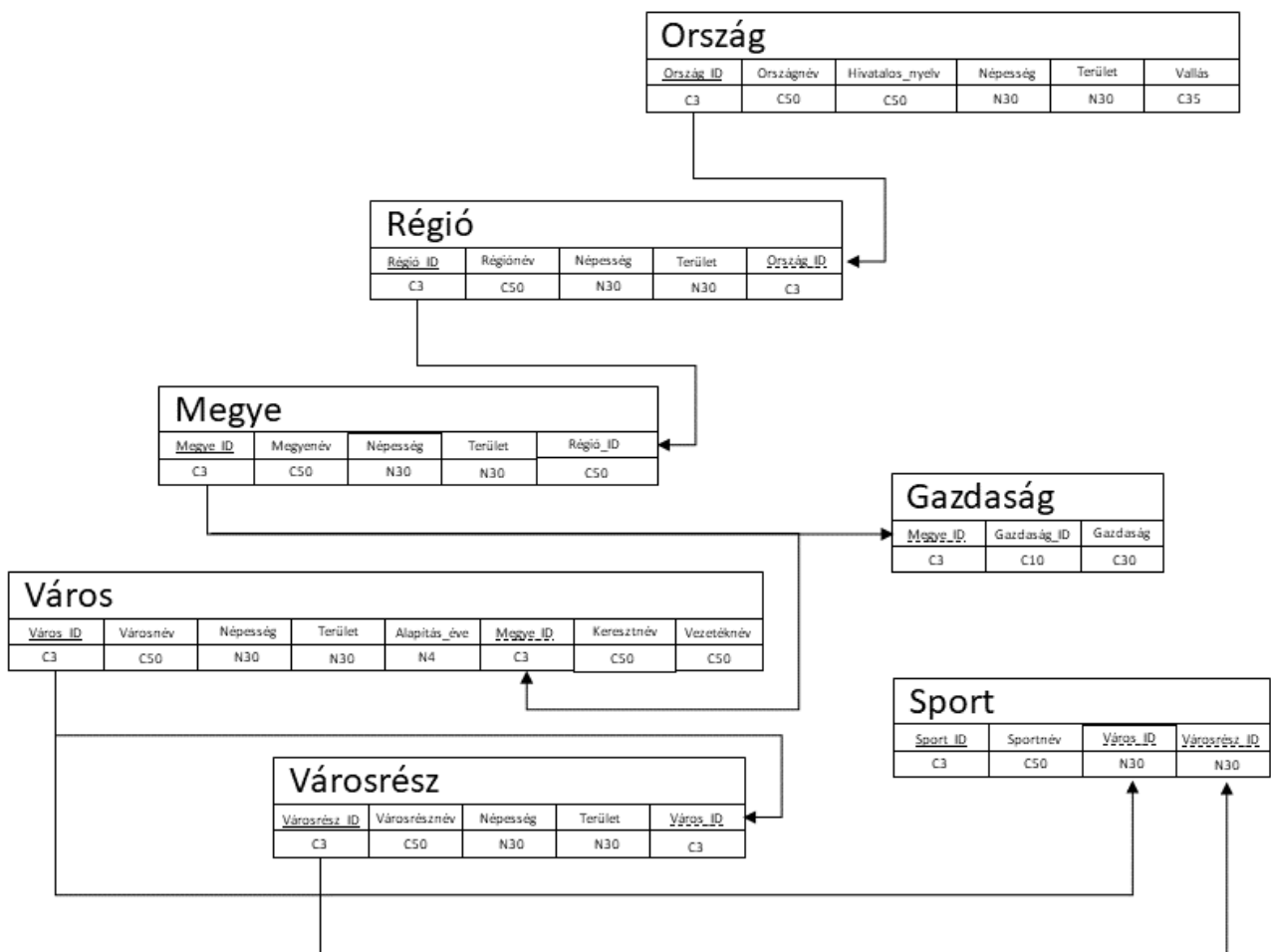
Tábla képződik a **Városrész** egyedből. A tulajdonságok változatlanul megmaradnak. A tábla kiegészül egy idegen kulccsal (**Város\_ID**) amely a **Város** tábla elsődleges kulcsát tartalmazza.

Tábla képződik a **Sport** egyedből. A tulajdonságok változatlanul megmaradnak. A tábla kiegészül két idegen kulccsal (**Város\_ID**) amely a **Város** tábla elsődleges kulcsát tartalmazza.

A **Gazdaság** mivel egy többértékű tulajdonság ezért ebből is egy tábla képződik. Megtalálható benne egy **Megye\_ID** nevű külső kulcs egy **Gazdaság\_ID** elsődleges kulcs és egy **Gazdaság** tulajdonság.

(Megjegyzés: Ahol egy táblában megtalálható a **Népesség** és a **Terület** ott összetett tulajdonságként fog elő állni az ún. **Népsűrűség** mező a **Népesség** és a **Terület** osztlékából.)

## Relációs Modell



## Relációs sémák:

Ország[Ország\_ID, Orzágnév, Hivatalos\_nyelv, Néesség, Terület, Vallás]

Régió[Régió\_ID, Régiónev, Néesség, Terület, Ország\_ID]

Megye[Megye\_ID, Megyenév, Néesség, Terület, Régió\_ID]

Város[Város\_ID, Városnev, Néesség, Terület, Alapítás\_éve, Megye\_ID, Keresztnév, Vezetéknév, Keresztnév]

Városrész[Városrész\_ID, Városrésznév, Néesség, Terület, Város\_ID]

Gazdaság[Megye\_ID, Gazdaság\_ID, Gazdaság]

## Implementáció SQL-be

Az eddig felsorolt sorrendben jönnek létre a *táblák*.

Erre azon egyszerű okból van szükség hiszen az **Ország** *tábla* kivételével minden *tábla* tartalmaz *idegen kulcsot*, aminek egy már létrehozott *táblára* kell mutatnia.

Ez nyilván megköveteli a *típus konverziót* a két *kulcs* (Az *idegen kulcs* és az adott másik *tábla kulcsa*) között.

## Táblák létrehozása

```
CREATE TABLE Ország(  
    Ország_ID CHAR(3) PRIMARY KEY,  
    Orzágnév VARCHAR(50) NOT NULL,  
    Hivatalos_nyelv VARCHAR(50) NOT NULL,  
    Néesség NUMBER(30) NOT NULL,  
    Terület NUMBER(30) NOT NULL,  
    Vallás VARCHAR(35)  
);
```

```
CREATE TABLE Régió(  
    Régió_ID CHAR(3) PRIMARY KEY,  
    Régiónev VARCHAR(50) NOT NULL,  
    Néesség NUMBER(7) NOT NULL,  
    Terület NUMBER(30) NOT NULL,  
    Ország_ID CHAR(3), FOREIGN KEY(Ország_ID) REFERENCES Ország(Ország_ID);  
);
```

```
CREATE TABLE (  
    Megye_ID CHAR(3) PRIMARY KEY,  
    Megyenév VARCHAR(50) NOT NULL,  
    Néesség NUMBER(7) NOT NULL,  
    Terület NUMBER(30) NOT NULL,  
    Gazdaság VARCHAR(30),  
    Régió_ID CHAR(3), FOREIGN KEY(Régió_ID) REFERENCES Régió(Régió_ID);  
);
```

```
CREATE TABLE Város(
  Város_ID CHAR(3) PRIMARY KEY,
  Városnév VARCHAR(50) NOT NULL,
  Néesség NUMBER(7) NOT NULL,
  Terület NUMBER(30) NOT NULL,
  Alapítás_éve NUMBER(4),
  Megye_ID CHAR(3), FOREIGN KEY(Megye_ID) REFERENCES Megye(Megye_ID);
  Vezetéknév VARCHAR(30),
  Keresztnév VARCHAR(30),
);
```

```
CREATE TABLE Városrész(
  Városrész_ID CHAR(3) PRIMARY KEY,
  Városrésznév VARCHAR(50) NOT NULL,
  Néesség NUMBER(7) NOT NULL,
  Terület NUMBER(30) NOT NULL,
  Város_ID CHAR(3), FOREIGN KEY(Város_ID) REFERENCES Város(Város_ID);
);
```

```
CREATE TABLE Sport(
  Sport_ID NUMBER(30) PRIMARY KEY,
  Sportnév VARCHAR(30),
  Város_ID CHAR(3), FOREIGN KEY(Város_ID) REFERENCES Város(Város_ID),
  Városrész_ID CHAR(3), FOREIGN KEY(Városrész_ID) REFERENCES Városrész(Városrész_ID));
```

```
CREATE TABLE Gazdaság (
  Megye_ID CHAR(3), FOREIGN KEY(Megye_ID) REFERENCES Megye(Megye_ID),
  Gazdaság_ID VARCHAR(10) PRIMARY KEY,
  Gazdaság VARCHAR(30)
);
```

## Adatok feltöltése

BEGIN

```
INSERT INTO Ország VALUES('HUN' , 'Magyarország' , 'magyar' , 9817958 , 93036 , 'kereszténység');
INSERT INTO Ország VALUES('SLO' , 'Szlovákia' , 'szlovák' , 5426252 , 49036 , 'kereszténység');
INSERT INTO Ország VALUES('AUS' , 'Ausztria' , 'német' , 8572895 , 83879 , 'kereszténység');
INSERT INTO Ország VALUES('RUS' , 'Oroszország' , 'orosz' , 146804372 , 17098240 , 'kereszténység');
```

```
INSERT INTO Régió VALUES('EMA' , 'Észak-Magyarország' , 1209142 , 13428 , 'HUN');
INSERT INTO Régió VALUES('EAL' , 'Észak-Alföld' , 1492502 , 17749 , 'HUN');
INSERT INTO Régió VALUES('DAL' , 'Dél-Alföld' , 1318214 , 18339 , 'HUN');
INSERT INTO Régió VALUES('KMA' , 'Közép-Magyarország' , 2951436 , 6919 , 'HUN');
INSERT INTO Régió VALUES('KDU' , 'Közép-Dunántúl' , 1098654 , 11237 , 'HUN');
INSERT INTO Régió VALUES('NDU' , 'Nyugat-Dunántúl' , 996390 , 11209 , 'HUN');
INSERT INTO Régió VALUES('DDU' , 'Dél-Dunántúl' , 947986 , 14169 , 'HUN');
INSERT INTO Régió VALUES('WSL' , 'Nyugat-Szlovákia' , 1870599 , 12554 , 'SLO');
INSERT INTO Régió VALUES('MSL' , 'Közép-Szlovákia' , 1945300 , 20744 , 'SLO');
INSERT INTO Régió VALUES('ESL' , 'Nyugat-Szlovákia' , 1580539 , 15746 , 'SLO');
```

```

INSERT INTO Megye VALUES('BAZ' , 'Borsod-Abaúj-Zemplén' , 684793 , 7247 , 'EMA');
INSERT INTO Megye VALUES('HEV' , 'Heves' , 307985 , 3637 , 'EMA');
INSERT INTO Megye VALUES('SSB' , 'Szabolcs-Szatmár-Bereg' , 555496 , 5936 , 'EAL');
INSERT INTO Megye VALUES('KOS' , 'Kassai Terület' , 766012 , 6753 , 'ESL');
INSERT INTO Megye VALUES('PRE' , 'Eperjesi Terület' , 814527 , 8993 , 'ESL');
INSERT INTO Megye VALUES('BRA' , 'Pozsonyi Terület' , 602436 , 2053 , 'WSL');

INSERT INTO Város VALUES('MIS' , 'Miskolc' , 159554 , 237 , 1173 , 'BAZ' , 'Krizsa' , 'Ákos');
INSERT INTO Város VALUES('EGE' , 'Eger' , 54609 , 92 , 670 , 'HEV' , 'Habis' , 'László');
INSERT INTO Város VALUES('KOS' , 'Kassa' , 240688 , 243 , 1230 , 'KOS' , 'Raši' , 'Richard');
INSERT INTO Város VALUES('PRE' , 'Eperjes' , 91498 , 70 , 1132 , 'PRE' , 'Turčanová' , 'Andrea');
INSERT INTO Város VALUES('BRA' , 'Pozsony' , 417389 , 368 , 907 , 'BRA' , 'Nesrovnal' , 'Ivo');
INSERT INTO Város VALUES('NYI' , 'Nyíregyháza' , 118125 , 275 , 1209 , 'SSB' , 'Kovács' , 'Ferenc');

INSERT INTO Városrész VALUES('KEL' , 'Kelet-városrész' , 8302 , 5 , 'EGE');
INSERT INTO Városrész VALUES('BOR' , 'Borbánya' , 8302 , 5 , 'EGE');
INSERT INTO Városrész VALUES('DZU' , 'Dzsungel' , 671 , 1 , 'KOS');
INSERT INTO Városrész VALUES('KVP' , 'KVP Lakótelep' , 24858 , 2 , 'KOS');
INSERT INTO Városrész VALUES('DAR' , 'Dargói Hősök Lakótelep' , 26955 , 11 , 'KOS');
INSERT INTO Városrész VALUES('ABA' , 'Abaszéplak' , 4739 , 20 , 'BRA');
INSERT INTO Városrész VALUES('RUZ' , 'Főrév' , 68574 , 40 , 'BRA');
INSERT INTO Városrész VALUES('DEV' , 'Dévény' , 1096 , 14 , 'BRA');
INSERT INTO Városrész VALUES('DIO' , 'Diósgyőr' , 18065 , 17 , 'MIS');
INSERT INTO Városrész VALUES('PER' , 'Pereces' , 4102 , 9 , 'MIS');

INSERT INTO Sport VALUES('1' , 'Gördeszkázás' , 4102 , 9 , 'MIS' , 'DIO');
INSERT INTO Sport VALUES('2' , 'Labdarúgás' , 4102 , 9 , 'MIS' , 'DIO');
INSERT INTO Sport VALUES('3' , 'Gördeszkázás' , 4102 , 9 , 'KOS' , 'KVP');

INSERT INTO Gazdaság VALUES('BAZ' , 1 , 'könnyűipar');
INSERT INTO Gazdaság VALUES('BAZ' , 2 , 'vegyipar');
INSERT INTO Gazdaság VALUES('HEV' , 3 , 'borászat');
INSERT INTO Gazdaság VALUES('HEV' , 4 , 'turizmus');
INSERT INTO Gazdaság VALUES('SSB' , 5 , 'mezőgazdaság');

```

END;

## Módosítások

```

ALTER TABLE Ország ADD Népsűrűség NUMBER(30);
UPDATE Ország SET Népsűrűség = Néesség/Terület;

```

```

ALTER TABLE Régió ADD Népsűrűség NUMBER(30);
UPDATE Régió SET Népsűrűség = Néesség/Terület

```

```

ALTER TABLE Megye ADD Népsűrűség NUMBER(30);
UPDATE Megye SET Népsűrűség = Néesség/Terület

```

```

ALTER TABLE Város ADD Népsűrűség NUMBER(30);
UPDATE Város SET Népsűrűség = Néesség/Terület

```

```
ALTER TABLE Városrész ADD Népsűrűség NUMBER(30);  
UPDATE Városrész SET Népsűrűség = Népség/Terület
```

Tegyük fel, hogy választások után vagyunk és új ember lesz a polgármester! Ekkor:

```
UPDATE Város SET Vezetéknév='Szerencsés' , Keresztnév='Szilárd' WHERE Város_ID='MIS';  
Természetesen ezt vissza is lehet (jelen esetben kell is) csinálni.  
UPDATE Város SET Vezetéknév='Krizsa' , Keresztnév='Ákos' WHERE Város_ID='MIS';
```

## Lekérdezések

*Azon országok neve és hivatalos nyelve ahol nem a magyar a hivatalos nyelv:*

```
SELECT Országnév, Hivatalos_nyelv FROM Ország WHERE NOT Hivatalos_nyelv='magyar';
```

*A magyarországi régiók neve és azonosítója:*

```
SELECT Régiónév, Régió_ID FROM Régió WHERE Ország_ID='HUN';
```

*Egy országban található régiók száma:*

```
SELECT Országnév, COUNT(Régiónév) FROM Ország LEFT OUTER JOIN Régió ON  
Régió.Ország_ID=Ország.Ország_ID GROUP BY Országnév;
```

*A Szlovák városokat általában e körül alapították:*

```
SELECT AVG(Alapítás_éve) FROM Város INNER JOIN Megye ON Város.Megye_ID=Megye.Megye_ID  
INNER JOIN Régió On Megye.Régió_ID=Régió.Régió_ID WHERE Ország_ID='SLO';
```

*A legnépesebb város neve és lakossága:*

```
SELECT Városnév, Népség FROM Város WHERE Népség=(SELECT MAX(Népség) FROM  
Város);
```

*Heves megye gazdasági jellemzői:*

```
SELECT Gazdaság FROM Gazdaság WHERE Megye_ID='HEV';
```

*A vegyiparral foglalkozó megyék neve:*

```
SELECT Megyenév FROM Megye INNER JOIN Gazdaság ON Megye.Megye_ID=Gazdaság.Megye_ID  
WHERE Gazdaság='vegyipar';
```

*Azok a városok amelyek városrészeiről egyáltalán nincs adat:*

```
SELECT V.Városnév FROM Város V WHERE NOT EXISTS(SELECT * FROM Városrész VR WHERE  
VR.Város_ID=V.Város_ID);
```

*Azon megye neve aminek kevesebb mint kettő gazdasági jellemzője van:*

```
SELECT Megyenév, COUNT(*) FROM Megye INNER JOIN Gazdaság ON  
Megye.Megye_ID=Gazdaság.Megye_ID GROUP BY Megyenév HAVING COUNT(*)<2;
```

*Szlovákia vagy Ausztria népsűrűsége közül amelyiknek kisebb a népsűrűsége:*

```
SELECT Országnév FROM Ország WHERE Népsűrűség=(SELECT MIN(Népsűrűség) FROM Ország  
WHERE Ország_ID='SLO' OR Ország_ID='AUS');
```

*Ezt le is tudjuk ellenőrizni az alábbi módon:*

```
SELECT Országnév FROM Ország WHERE Népsűrűség=(SELECT MAX(Népsűrűség) FROM Ország  
WHERE Ország_ID='SLO' OR Ország_ID='AUS');
```



*Azon városok neve ahol a gördeszkázás az egyik sport:*

```
SELECT Városnév FROM Város INNER JOIN Sport ON Város.Város_ID=Sport.Város_ID WHERE  
Sport.Város_ID IN (SELECT Város_ID FROM Sport WHERE Sportnév='Gördeszkázás') GROUP BY  
Városnév;
```

*Miskolcon az alábbi sportok találhatóak:*

```
SELECT Sportnév FROM Város INNER JOIN Sport ON Sport.Város_ID=Város.Város_ID WHERE  
Sport.Város_ID='MIS';
```

*A feladatban felhasznált adatok valódiak. Forrás: Wikipédia, KSH*