

# Több táblából álló adatbázis. Az SQL SELECT utasítása: oszlopválogatás, sorválogatás; SFW különböző lehetőségei.

## Több táblából álló adatbázisok



HALLGATO (neptunkod, nev, szakazon)

SZAK (szkod, sznev, szfelelos)

### Kapcsolt táblák létrehozása:

```
CREATE TABLE szak(  
  szkod char(1) PRIMARY KEY,  
  sznev varchar(25),  
  szfelelos varchar(30));
```

```
CREATE TABLE hallgato(  
  neptunkod char(6) PRIMARY KEY,  
  nev varchar(30),  
  szakazon char(1),  
  FOREIGN KEY(szakazon) REFERENCES szak(szkod));
```

### Táblák feltöltése adatokkal:

A hallgatók felvételekor már léteznie kell a megadott szaknak a hivatkozott táblában!

*Különben:*

The query could not be executed.

!	Description	ErrorNr.
!	Cannot add or update a child row: a foreign key constraint fails ('A_HTESZT`.`hallgato`, CONSTRAINT `hallgato_ibfk_1` FOREIGN KEY (`szakazon`) REFERENCES `szak` (`szkod`))	1452

Ezért először a *szak* táblát töltjük fel, és csak utána hivatkozhatunk a benne szereplő szakokra a *hallgato* táblában adatfelvitelkor!

```
INSERT INTO szak VALUES('i','informatika','Dr. Kiss Géza');
```

```
INSERT INTO hallgato VALUES('AAAAAA', 'Kiss Tibor', 'i');
```

## Több táblából álló adatbázis. Az SQL SELECT utasítása: oszlopválogatás, sorválogatás; SFW különböző lehetőségei.

---

### Rekordok törlése:

A hivatkozott táblában csak akkor törölhető egy adott rekord, ha a hivatkozó táblában nincs olyan rekord, amely rá hivatkozna. Azaz a *szak* táblából csak akkor törölhetek egy szakot, ha a *hallgato* táblában nincs olyan szakos hallgató.

*Helyes sorrend:*

```
DELETE FROM hallgato WHERE szakazon='i';
```

```
DELETE FROM szak WHERE szkod='i';
```

### Kaszádolt törlés és módosítás:

Ha a *FOREIGN KEY* megszorítás végén megadjuk az **ON DELETE CASCADE** opciót, akkor a szülő rekord törlésével a gyermek rekordok is törlődnek a hivatkozó táblában. Azaz, ha pl.: törölöm az 'i' szakkódot a *szak* táblából, akkor a *hallgato* táblából törlődik minden olyan hallgató, aki ilyen szakos.

Ha a *FOREIGN KEY* megszorítás végén megadjuk az **ON UPDATE CASCADE** opciót, akkor a szülő rekord módosításával a gyermek rekordok is módosulnak a hivatkozó táblában. Azaz, ha pl.: módosítom az 'i' szakkódot a *szak* táblában valamilyen más kódra, akkor a *hallgato* táblában módosul minden olyan hallgató szakazonosítója, aki informatika szakos.

*Kaszádolt törlés és frissítés megadása egy lépésben tábladefiniáláskor:*

```
CREATE TABLE hallgato(  
neptunkod char(6) PRIMARY KEY,  
nev varchar(30),  
szakazon char(1),  
FOREIGN KEY(szakazon) REFERENCES szak(szkod) ON UPDATE CASCADE ON DELETE CASCADE);
```

# Több táblából álló adatbázis. Az SQL SELECT utasítása: oszlopválogatás, sorválogatás; SFW különböző lehetőségei.

## Gyakorló feladat

### 1. Feladat

Készítse el az alábbi táblákat, és a köztük lévő kapcsolatokat!

#### Hallgato

Hkod	Nev	Irszam	Varos	Utca_hsz	Telefon	Szdatum
1	Török Anna	7620	Pécs	Pipacs u. 12.	(75) 111-222	1982.01.20
2	Németh Ábel	2513	Ercsi	Fa u.123.	(75) 111-223	1980.01.21
3	Nagy Elvira	7400	Paks	Dózsa Gy. 23.	(75) 111-224	1980.11.22
4	Kiss Jutka	8500	Pápa	Sóház u.120.	(75) 111-225	1982.01.22
5	Szabadi Imre	8500	Pápa	Szőlő u. 16.	(75) 111-226	1980.01.20
6	Vendég Anna	2514	Adony	Fő tér 12/B.	(75) 111-227	1980.11.25
7	Haragos Lajos	2580	Tata	Kossuth u. 16.	(75) 111-228	1982.01.25

#### Tantargy

Tkod	Tnev
10	Angol
20	Algebra
30	Német
40	Matematika
50	Logika
60	Történelem
70	Fizika

#### Tanul

Hkod	Tkod	Jegy
1	10	4
1	20	1
2	20	5
2	30	1
3	60	2
4	70	5

HALLGATO (Hkod, Nev, Irszam, Varos, Utca, Telefon, Szdatum)

TANUL (Hkod, Tkod, Jegy)

TANTARGY (Tkod, Tnev)

#### Hallgato tábla mezőtípusai:

- **Hkod:** karakteres fix 1 hosszú, elsődleges kulcs
- **Nev:** karakteres, változó hossz, max 30 karakter, kötelező kitölteni
- **Irszam:** egész típus
- **Varos:** karakteres, változó hossz, max 25 karakter
- **Utca\_hsz:** karakteres, változó hossz, max 50 karakter
- **Telefon:** karakteres, változó hossz, max 12 karakter
- **Szdatum:** dátum típus

## Több táblából álló adatbázis. Az SQL SELECT utasítása: oszlopválogatás, sorválogatás; SFW különböző lehetőségei.

---

### Tantargy tábla mezőtípusai:

- **Tkod:** karakteres fix 2 hosszú, elsődleges kulcs
- **Tnev:** karakteres, változó hossz, max 30 karakter, kötelező kitölteni

### Tanul tábla mezőtípusai:

- **Hkod:** karakteres fix 1 hosszú
- **Tkod:** karakteres fix 2 hosszú
- **Jegy:** egész típus

*Ahol a Hkod és a Tkod  
elsődleges kulcsok,  
és egyben idegen kulcsok is!*

## Több táblából álló adatbázis. Az SQL SELECT utasítása: oszlopválogatás, sorválogatás; SFW különböző lehetőségei.

---

### A település tábla létrehozása

```
CREATE TABLE telepules (  
    helysegnev VARCHAR(255),  
    tipus VARCHAR(255),  
    megye VARCHAR(255),  
    jaras VARCHAR(255),  
    terület INT,  
    lakosság INT,  
    lakasokszama INT )  
CHARACTER SET latin2 COLLATE latin2_hungarian_ci;
```

### A település tábla feltöltése másolással ~ másolás adatbázison belül:

```
INSERT INTO telepules SELECT * FROM dudasn.telepules;           (sémanév.sémaelemneve)
```

### Tábla mentése fájlba:

```
SELECT * FROM táblanév INTO OUTFILE 'fájlnév';
```

(a fájlnévben \ helyett / használandó! Valamint megfelelő jogok szükségesek hozzá!)

Mint minden SQL utasítást, ezt is a szerver hajtja végre, így a működéshez a szerveren kellő jogok szükségesek. Ha ugyanaz a gép a szerver és a kliens, akkor nincs ilyen probléma.

Pl.:

```
SELECT * FROM település INTO OUTFILE 'f:/mentes';
```

### Tábla feltöltése a mentett fájlból:

```
LOAD DATA LOCAL INFILE 'fájlnév' INTO TABLE táblanév CHARACTER SET latin2;
```

Pl.:

```
LOAD DATA LOCAL INFILE 'f:/mentes'  
INTO TABLE telepules CHARACTER SET latin2  
FIELDS TERMINATED BY '\t';  
vagy  
LOAD DATA LOCAL INFILE 'd:/munka/hely.csv'  
INTO TABLE telepules CHARACTER SET latin2  
FIELDS TERMINATED BY ',';
```

## Több táblából álló adatbázis. Az SQL SELECT utasítása: oszlopválogatás, sorválogatás; SFW különböző lehetőségei.

1. hely.csv Moodle
2. telepules tábla létrehozása

```
CREATE TABLE telepules (  
    helysegnev VARCHAR(255),  
    tipus VARCHAR(255),  
    megye VARCHAR(255),  
    jaras VARCHAR(255),  
    terület INT,  
    lakosság INT,  
    lakasokszama INT )  
CHARACTER SET latin2 COLLATE latin2_hungarian_ci;
```

3. Tábla feltöltése a mentett fájlból

```
LOAD DATA LOCAL INFILE 'd:/munka/hely.csv'  
INTO TABLE telepules CHARACTER SET latin2  
FIELDS TERMINATED BY ',';
```

vagy

1. telepules tábla létrehozása

```
CREATE TABLE telepules (  
    helysegnev VARCHAR(255),  
    tipus VARCHAR(255),  
    megye VARCHAR(255),  
    jaras VARCHAR(255),  
    terület INT,  
    lakosság INT,  
    lakasokszama INT )  
CHARACTER SET latin2 COLLATE latin2_hungarian_ci;
```

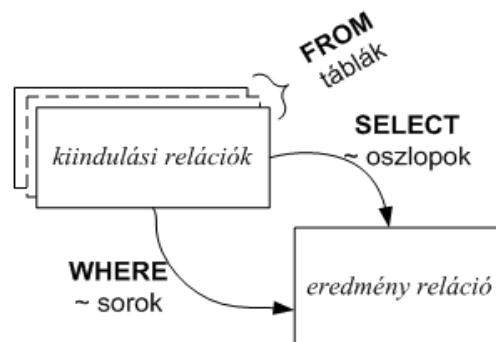
2. A telepules tábla feltöltése másolással

```
INSERT INTO telepules SELECT * FROM dudasn.telepules;
```

### SELECT

A **SELECT** utasítás egy vagy több kiindulási táblából hoz létre eredménytáblát.

#### SELECT általános bemutatása (SFW)



```
SELECT * FROM telepules;
```

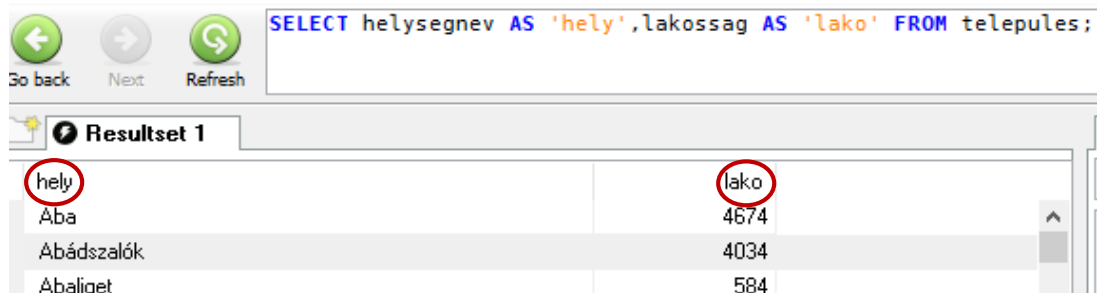
Az utasítás eredménytáblája megegyezik az eredeti táblával!

## Több táblából álló adatbázis. Az SQL SELECT utasítása: oszlopválogatás, sorválogatás; SFW különböző lehetőségei.

**Projekció (vetítés, itt oszlopválogatás):** `SELECT` oszlopnév(ek) `FROM` táblanév ;

```
SELECT helysegnev,lakosság FROM telepules;
```

Az eredménytáblában megjelenő oszlopnevek megadásával:



hely	lako
Aba	4674
Abádszalók	4034
Abaliget	584

A `SELECT` kulcsszót követő attribútum listában konstansok és kiszámítási formulák is szerepelhetnek:

*Pl.:*

```
SELECT fizetes*1.15 AS 'uj_fizetes' FROM Dolgozok;
```

```
SELECT *,fizetes*1.15 AS 'uj_fizetes' FROM Dolgozok;
```

**Szelekció (sorválogatás):** `SELECT` oszlopnév `FROM` táblanév `WHERE` feltétel;

A feltételben szerepelhetnek:

- összehasonlító operátorok: =, <>, <, >, <=, >=, != oszlopnév operátor kifejezés
- zárójelek: (,) zárójelek
- logikai műveletek: AND, OR, NOT
- matematikai alpműveletek: +, -, \*, stb.
- az SQL standard összehasonlító függvényei: MIN, MAX, SUM, AVG, COUNT
- Összehasonlítás NULL értékkel: oszlopnév IS NULL

*Pl.:* `SELECT * FROM hallgatok WHERE varos IS NULL;`

*Fel.:*

```
|SELECT * FROM telepules WHERE lakosság>100000;
```

```
SELECT * FROM telepules WHERE helysegnev LIKE '%tisza%';
```

## Több táblából álló adatbázis. Az SQL SELECT utasítása: oszlopválogatás, sorválogatás; SFW különböző lehetőségei.

---

### Projekció és szelekció együtt:

Pl.:

```
SELECT helysegnev,lakosság,lakasokszama  
FROM telepules  
WHERE lakosság>100000;
```

vagy

```
SELECT helysegnev AS 'hely',lakosság AS 'lako',  
lakasokszama AS 'lakas'  
FROM telepules  
WHERE lakosság>100000;
```

### **Feladat (sor és oszlopválogatás):**

1. Szűrjük ki azokat a településeket (minden adatával együtt), amelyek területe 5000 és 10 000 közötti.
2. Keressük meg azoknak a településeknek a nevét, amelyek lakossága 150 000 feletti.
3. Gyűjtsük ki azoknak a településeknek a nevét, amelyek neve D betűvel kezdődik.
4. Gyűjtsük ki azoknak a településeknek a nevét, és a lakosok számát, melyek neve D betűvel kezdődik és 18 000 feletti a lakosok száma.
5. Melyek a Fejér megyei települések?
6. Gyűjtsük ki a Tolna és Fejér megyei településeket!
7. Gyűjtsük ki az összes települést, kivéve a Tolna és a Fejér megyeiket!
8. Jelenítsük meg a 100 000 fő feletti Fejér vagy Baranya megyei településeket!



## Több táblából álló adatbázis. Az SQL SELECT utasítása: oszlopválogatás, sorválogatás; SFW különböző lehetőségei.

### Projekció (oszlopválogatás) kibővítése

- konstans beírása (új oszlopok megjelenése az eredményben):

```
SELECT helysegnev,terulet,'hektar'  
FROM telepules  
WHERE lakosság>100000;
```

helysegnev	terulet	hektar
Budapest III. kerület	3970	hektar
Budapest XI. kerület	3349	hektar
Budapest XIII. kerület	1344	hektar
Budapest XIV. kerület	1813	hektar
Debrecen	46166	hektar
Győr	17462	hektar
Kecskemét	32257	hektar
Miskolc	23666	hektar
Nyíregyháza	27454	hektar
Pécs	16277	hektar
Szeged	28100	hektar
Székesfehérvár	17089	hektar

- Karakteres művelet -összefűzés:

```
SELECT CONCAT(megye,',',helysegnev),lakosság  
FROM telepules  
WHERE lakosság>150000;
```

CONCAT(megye,',',helysegnev)	lakosság
Hajdú-Bihar,Debrecen	208016
Borsod-Abaúj-Zemplén,Miskolc	168075
Baranya,Pécs	157721
Csongrád,Szeged	170285

- indokoltabb lehet az eredménytáblában megjelenő oszlopnevek megadása:

```
SELECT CONCAT(megye,',',helysegnev) AS 'hely', lakosság  
FROM telepules  
WHERE lakosság>150000;
```

hely	lakosság
Hajdú-Bihar,Debrecen	208016
Borsod-Abaúj-Zemplén,Miskolc	168075
Baranya,Pécs	157721
Csongrád,Szeged	170285

- Egyéb karakteres műveletek:

**LOWER** ( oszlopnév )

Kisbetűre konvertál

Pl.: SELECT LOWER (nev) FROM hallgatok ;

**UPPER** ( oszlopnév )

Nagybetűre konvertál

**SUBSTR**( oszlopnév, m)

Az m-ik karaktertől kezdődően n db karaktert ad vissza.

Pl.: SELECT SUBSTR (nev,2) FROM hallgatok ;

## Több táblából álló adatbázis. Az SQL SELECT utasítása: oszlopválogatás, sorválogatás; SFW különböző lehetőségei.

### Feladat (Projekció kibővítése):

1. Szűrjük ki a *telepules* táblából azokat a városokat és a területüket, amelyek lakossága 150 000 feletti:

helysegnev	terulet
Debrecen	46166 hektar
Miskolc	23666 hektar
Pécs	16277 hektar
Szeged	28100 hektar

2. Keressük ki a 145 000 feletti településeket lakosságukkal együtt:

helysegnev	lakosság	fő
Debrecen	208016	fő
Miskolc	168075	fő
Pécs	157721	fő
Szeged	170285	fő

- *számítással előállított oszlop – Egy lakásra jutó lakók száma.*

```
SELECT helysegnev, lakosság/lakasokszama  
FROM telepules WHERE lakosság>200000;
```

helysegnev	lakosság/lakasokszama
Debrecen	2.3025

- *itt is indokoltabb lehet az eredménytáblában megjelenő oszlopnevek megadása:*

```
SELECT helysegnev, lakosság/lakasokszama AS 'Egy lakásra jutó lakók száma'  
FROM telepules WHERE lakosság>200000;
```

```
SELECT helysegnev, CONCAT(lakosság/lakasokszama, ' ', 'fő') AS 'Egy lakásra jutó lakók száma'  
FROM telepules WHERE lakosság>200000;
```

vagy

```
select helysegnev, concat(lakosság/lakasokszama, ' fő') as 'Egy lakásra jutó lakók száma'  
from telepules where lakosság>200000;
```

**Megjegyzés:** zérusosztás eredménye: **nullérték** – nem hiba (szemben a programok működésekor, ahol általában hibaállapothoz vezet)

### Feladat:

Adjuk meg az egy főre jutó lakások számát!

## Több táblából álló adatbázis. Az SQL SELECT utasítása: oszlopválogatás, sorválogatás; SFW különböző lehetőségei.

### Számformázás:

Pl.: Ezres csoport – formázás ezres elválasztó használatával.

```
SELECT helysegnev,FORMAT(lakosság,0) FROM telepules;
```

helysegnev	FORMAT(lakosság,0)
Aba	4,674
Abádszalók	4,034
Abaliget	584

- itt is indokoltabb lehet az eredménytáblában megjelenő oszlopnevek megadása:

```
SELECT helysegnev,FORMAT(lakosság,0) AS 'lakosság' FROM telepules;
```

helysegnev	lakosság
Aba	4,674
Abádszalók	4,034
Abaliget	584

### Érdekességek:

```
SELECT format(250000,0) AS 'Eredmeny';
```

Eredmeny

250,000

```
SELECT date_format(now(),'%H:%i %p') AS 'Pontos ido';
```

Pontos ido

14:31 PM

```
SELECT date_format(now(),'%h:%i %p') AS 'Pontos ido';
```

Pontos ido

02:37 PM

## Több táblából álló adatbázis. Az SQL SELECT utasítása: oszlopválogatás, sorválogatás; SFW különböző lehetőségei.

---

**Pl.:**

```
CREATE TABLE ido(nev varchar(25), ..., mikor datetime);
```

```
INSERT INTO ido VALUES('Kis Ede', ... now());
```

```
SELECT nev, ..., date_format(mikor,'%H:%i %p') FROM ido;
```

<b>Format</b>	<b>Description</b>
%a	Abbreviated weekday name (Sun-Sat)
%b	Abbreviated month name (Jan-Dec)
%c	Month, numeric (0-12)
%D	Day of month with English suffix (0th, 1st, 2nd, 3rd,)
%d	Day of month, numeric (00-31)
%e	Day of month, numeric (0-31)
%f	Microseconds (000000-999999)
%H	Hour (00-23)
%h	Hour (01-12)
%I	Hour (01-12)
%i	Minutes, numeric (00-59)
%j	Day of year (001-366)
%k	Hour (0-23)
%l	Hour (1-12)
%M	Month name (January-December)
%m	Month, numeric (00-12)
%p	AM or PM
%r	Time, 12-hour (hh:mm:ss followed by AM or PM)
%S	Seconds (00-59)
%s	Seconds (00-59)
%T	Time, 24-hour (hh:mm:ss)
%U	Week (00-53) where Sunday is the first day of week
%u	Week (00-53) where Monday is the first day of week
%V	Week (01-53) where Sunday is the first day of week, used with %X
%v	Week (01-53) where Monday is the first day of week, used with %x
%W	Weekday name (Sunday-Saturday)
%w	Day of the week (0=Sunday, 6=Saturday)
%X	Year for the week where Sunday is the first day of week, four digits, used with %V
%x	Year for the week where Monday is the first day of week, four digits, used with %v
%Y	Year, numeric, four digits
%y	Year, numeric, two digits