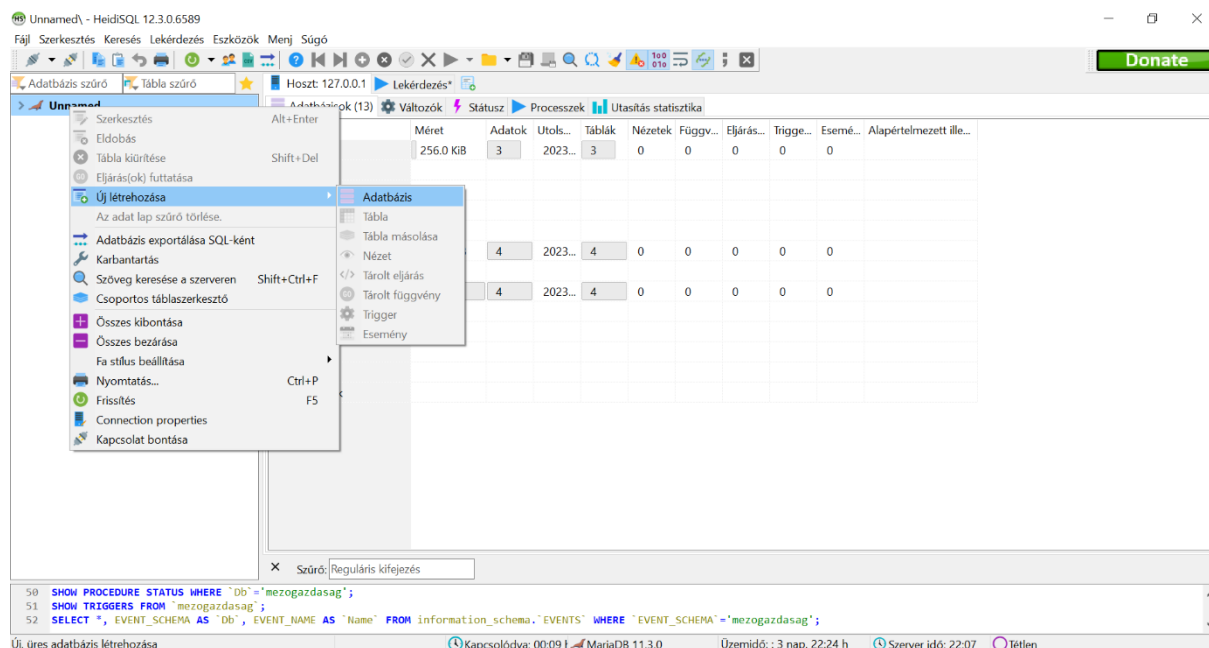


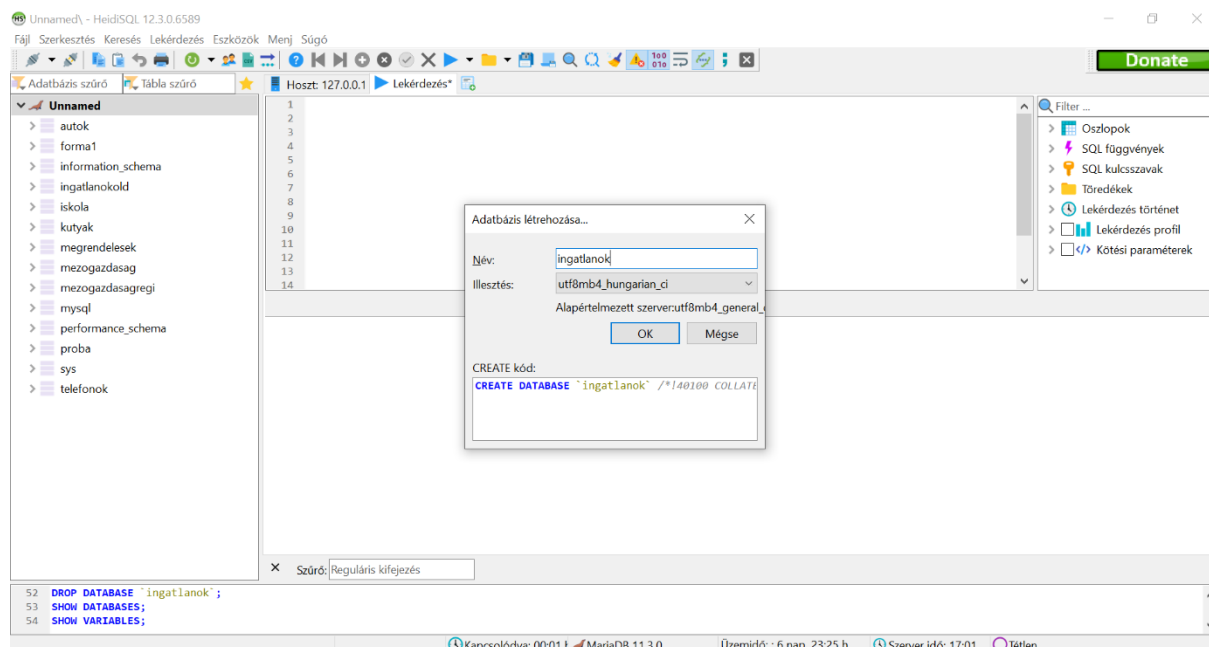
Adatbázis-kezelés D01 – Ingatlanok

Adatbázis létrehozása és az adatfájlok importálásának lépései a HeidiSQL felületen keresztül:

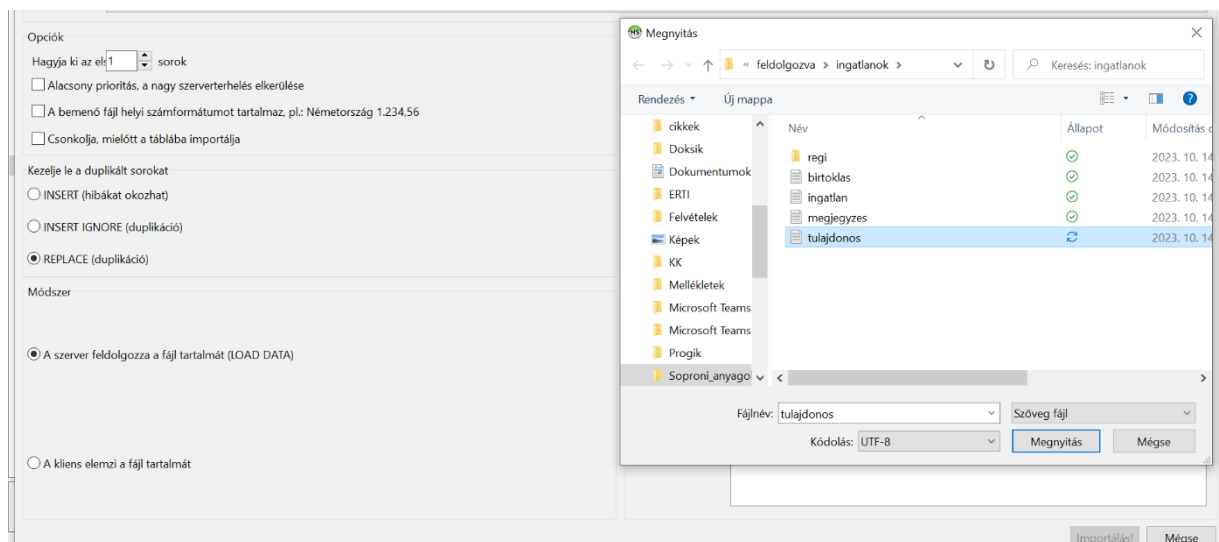
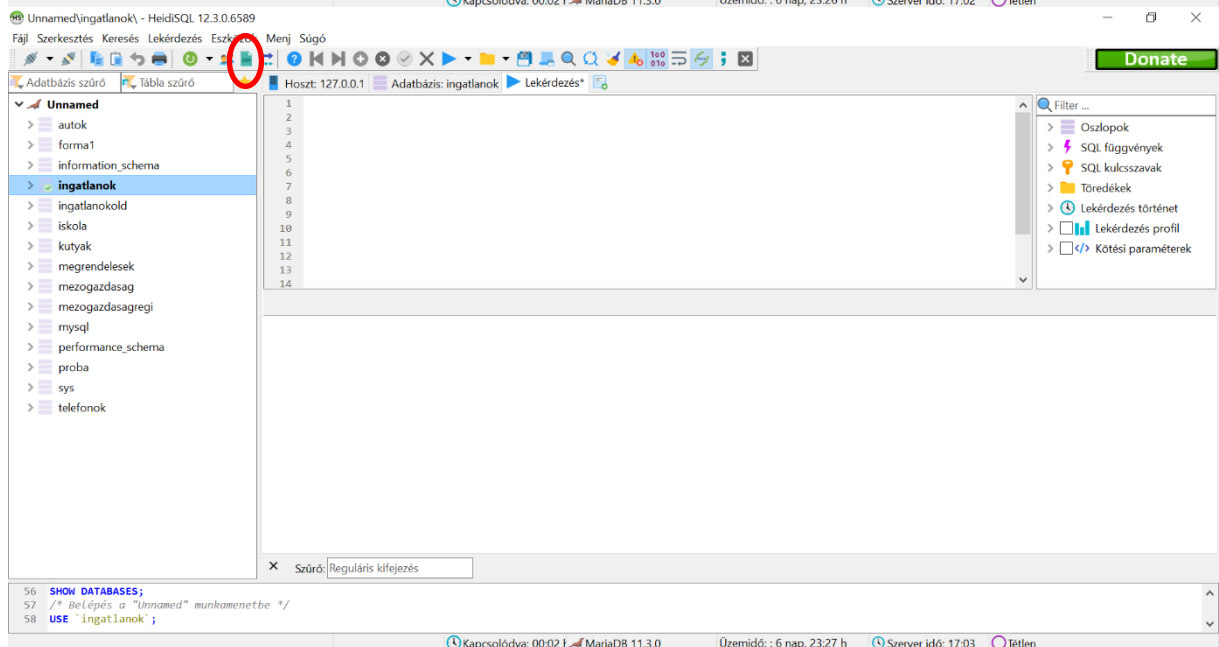
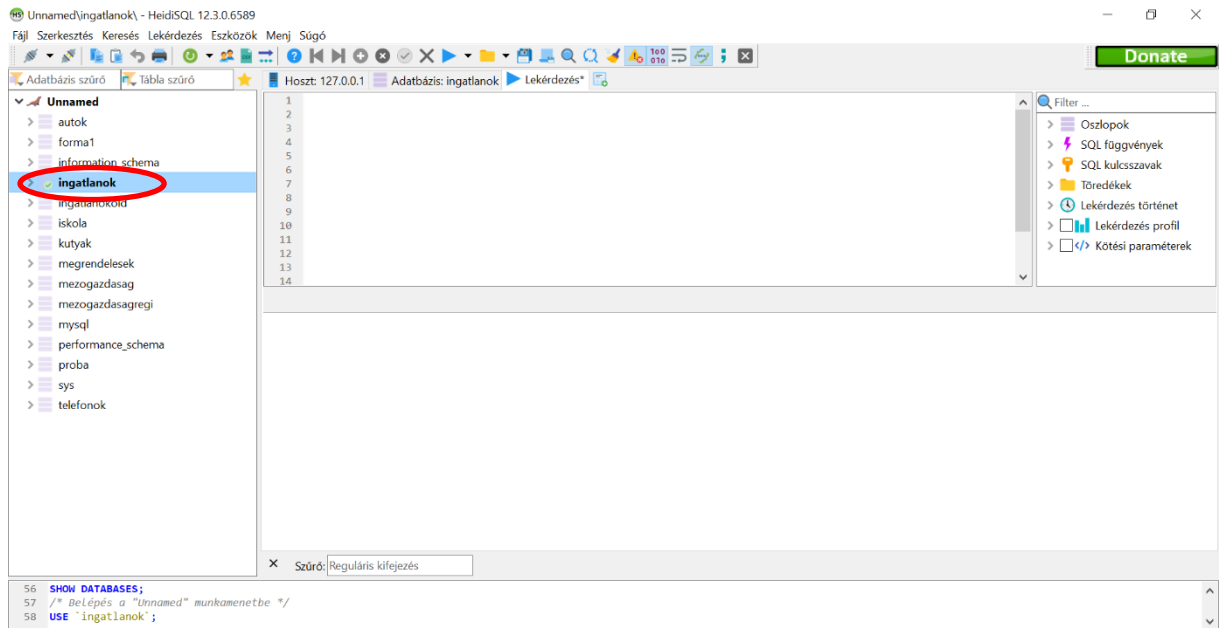
1. Új adatbázis létrehozása helyi menüvel.



2. Név adunk az adatbázisnak (*ingatlanok*) és figyelünk, hogy UTF8-magyar kódolást használjunk.



3. A jobboldali menüben a létrehozott *ingatlanok* adatbázist, majd a menüpontokból a CSV importot választjuk, ami után a kapott ablakban kijelölhetjük az importálandó fájlt (szöveg fájlt importálunk!). Ügyelünk, hogy az adatbázis nevéénél a megfelelő név (*ingatlanok*) látszódjon, illetve a tábla mezőben az új táblát kell választani.



Opciók

Hagyja ki az első 1 sorok

☐ Alacsony prioritás, a nagy szerverteljesítmény érdekében

☐ A bemenő fájl helyi számformátumot tartalmaz, pl.: Németország 1.234,56

☐ Csonkolja, mielőtt a táblába importálja

Kezelje le a duplikált sorokat

☐ INSERT (hibákat okozhat)

☐ INSERT IGNORE (duplikáció)

☒ REPLACE (duplikáció)

Módszer

☒ A szerverteljesítmény érdekében a fájl tartalmát (LOAD DATA)

☐ A kliens elemzi a fájl tartalmát

Vezérlő karakterek

Mező elválasztó: \t

Mező határoló: * ☒ opcionális

Mező escape sztring: *

Sor elválasztás: \r\n

Cél

Adatbázis: ingatlanok

Tábla: <Új tábla>

Oszlopok:

▲ Fel

▼ Le

☒ Mind

Importálás! Mégse

4. Új táblát definiáló SQL kód: a rendszer által felajánlott kódot a feladatban leírtaknak megfelelően javítsuk! Az 'ingatlanok' prefix használata nem kötelező, törölhető a kódból.

Szöveg fájl importálása

Bemeneti fájl

Fájlnev: C:\Users\Pödör Zoltán\OneDrive - Eotvos Lorand Tudományegyetem\Soproni anyagok\Infoerettseg\Mintafeladat_AB\feldolgozva\ingatlanok\tulajdonos.txt

Kódolás:

Opciók

Hagyja ki az első 1 sorok

☐ Alacsony prioritás, a nagy szerverteljesítmény érdekében

☐ A bemenő fájl helyi számformátumot tartalmaz, pl.: Németország 1.234,56

☐ Csonkolja, mielőtt a táblába importálja

Kezelje le a duplikált sorokat

☐ INSERT (hibákat okozhat)

☐ INSERT IGNORE (duplikáció)

☒ REPLACE (duplikáció)

Módszer

☒ A szerverteljesítmény érdekében a fájl tartalmát (LOAD DATA)

☐ A kliens elemzi a fájl tartalmát

Vezérlő karakterek

Mező elválasztó: \t

Mező határoló: * ☒ opcionális

Mező escape sztring: *

Sor elválasztás: \r\n

Cél

Adatbázis: ingatlanok

Tábla: <Új tábla>

Oszlopok:

▲ Fel

▼ Le

☒ Mind

Importálás! Mégse

CSV séma felismerése

Fájl beolvasása...

```

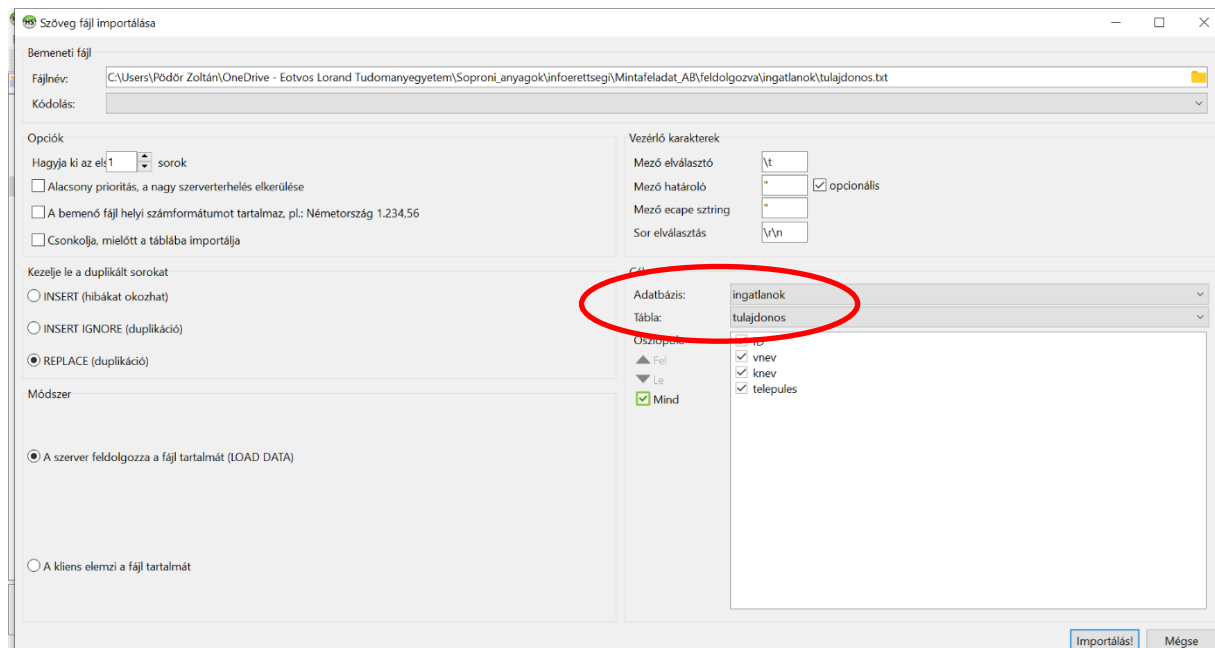
1 CREATE TABLE tulajdonos (
2   id INT PRIMARY KEY,
3   vnev VARCHAR(20) NOT NULL,
4   knev VARCHAR(20) NOT NULL,
5   telepules VARCHAR(30) NOT NULL
6 )
7

```

OK, hozzuk létre a táblát! Mégse

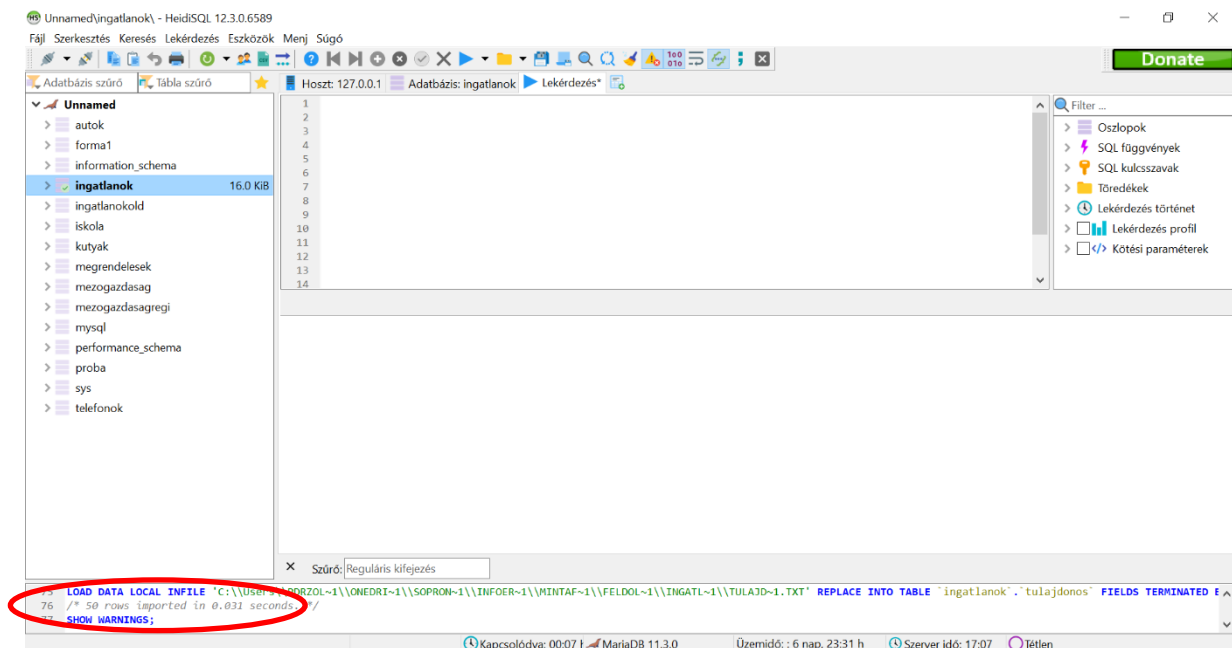
Hozzuk létre a táblát!

5. Adatok importálása a létrehozott táblába



Bökjünk rá az importálás gombra, ügyelve, hogy az adatbázis és a tábla neve a megfelelő legyen! Jelen esetben az adatbázis neve *ingatlanok*, a tábla neve pedig *tulajdonos*.

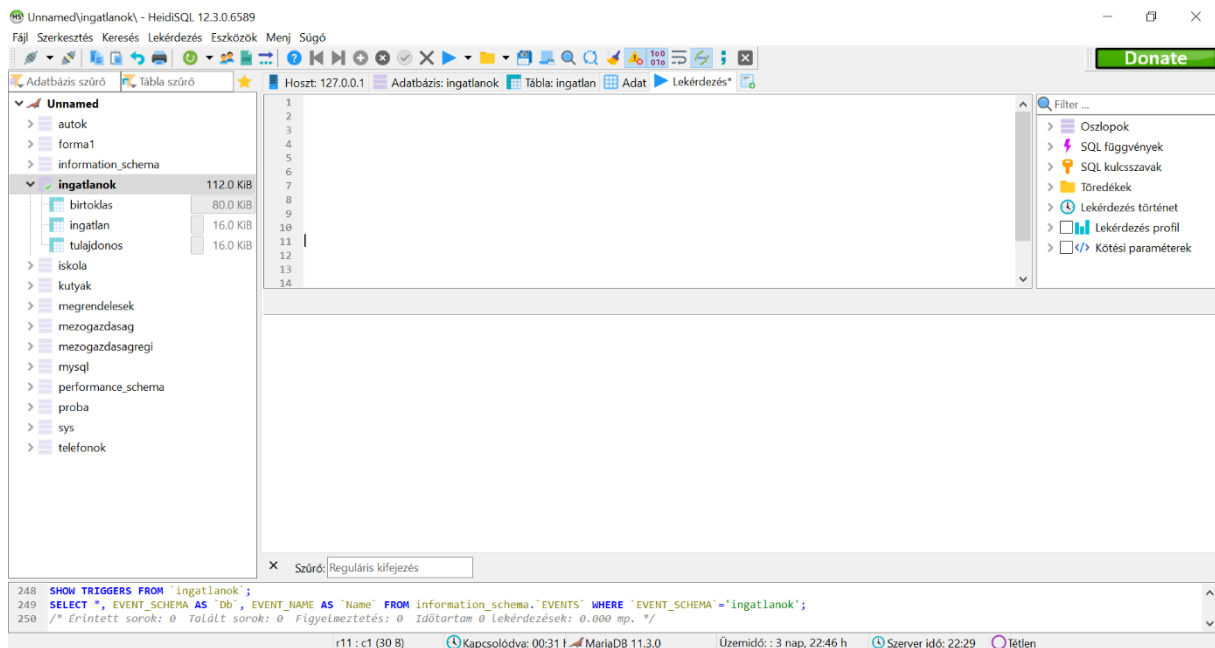
6. Tábla sikeresen létrehozva



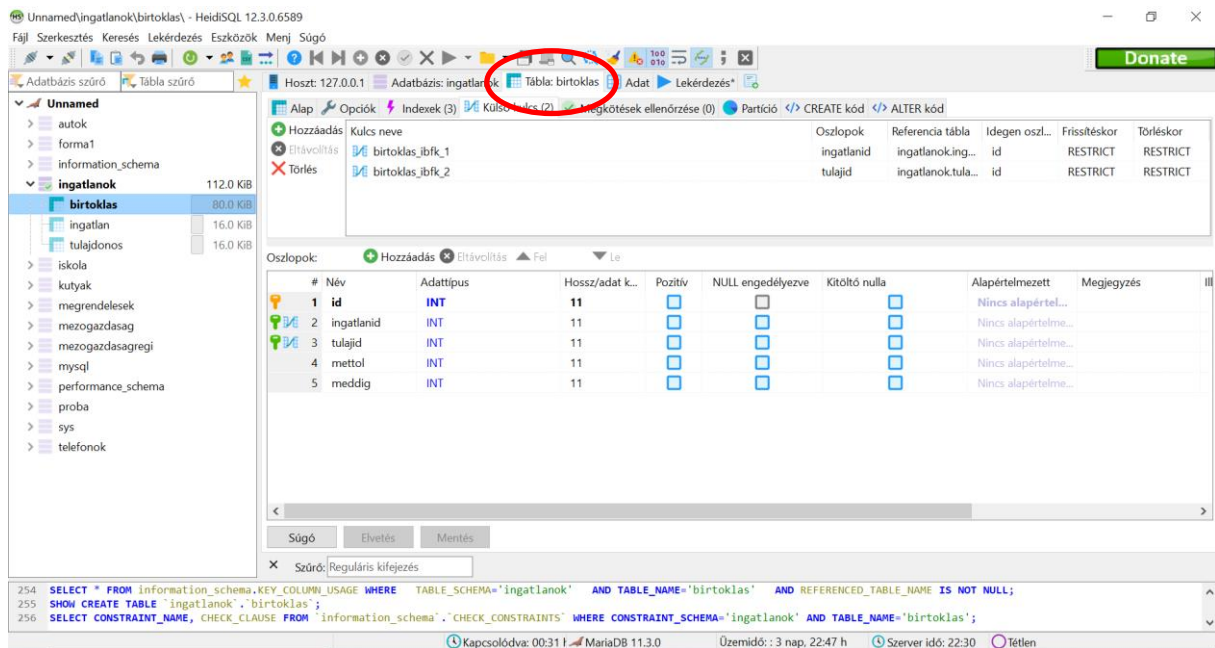
Alul láthatjuk, hogy 50 rekord beszúrása sikeresen megtörtént.

Ezt követően hasonlóan járunk el a másik két tábla esetében is a 3-6 pontok megfelelő ismétlésével.

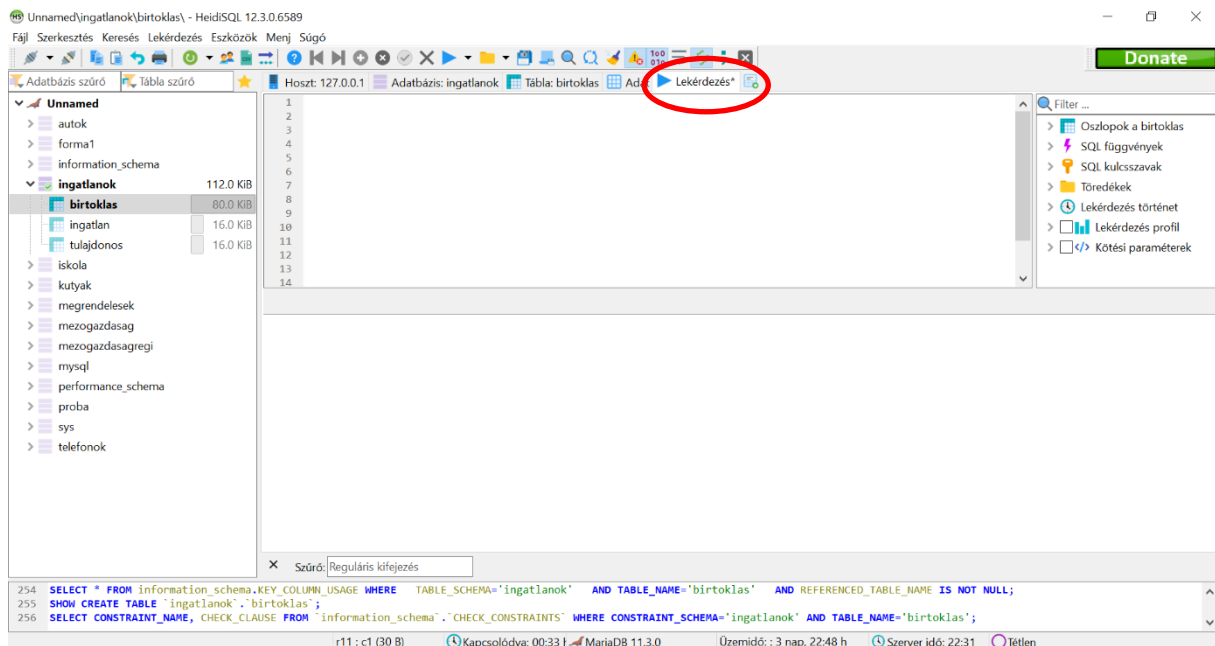
Ezzel az ingatlan adatbázist létrehoztuk és a szükséges adatokat importáltuk.



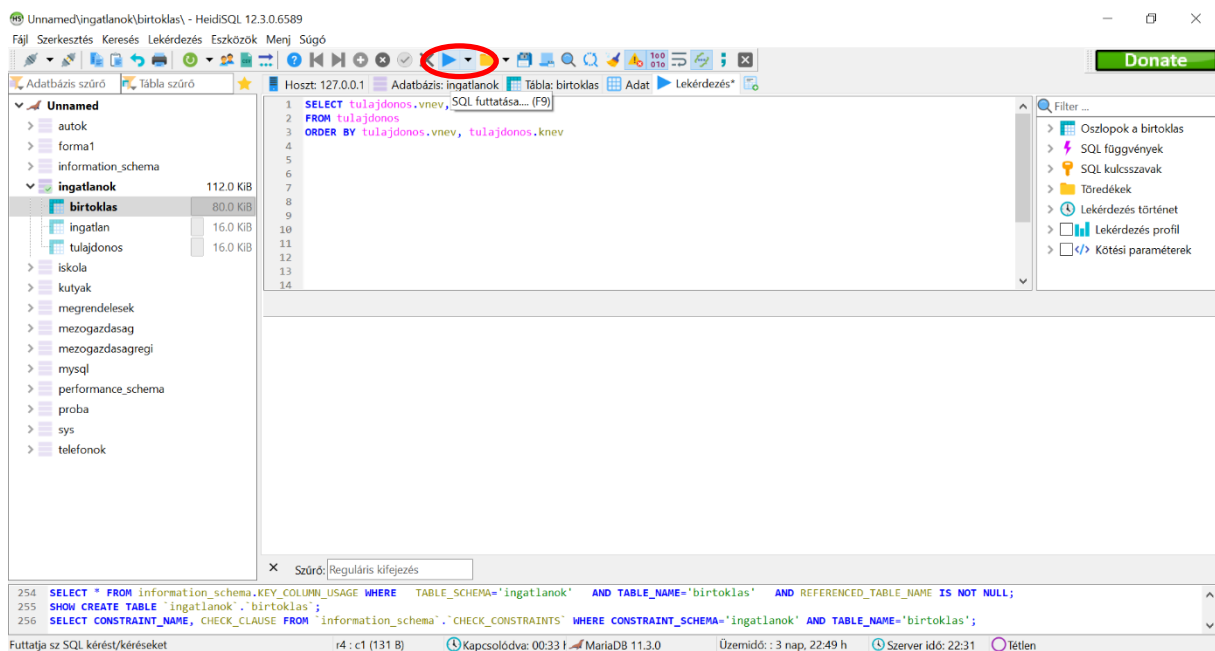
A létrehozott táblák ezt követően az adatbázis és a megfelelő tábla kiválasztását követően módosíthatóak a tábla fülön.



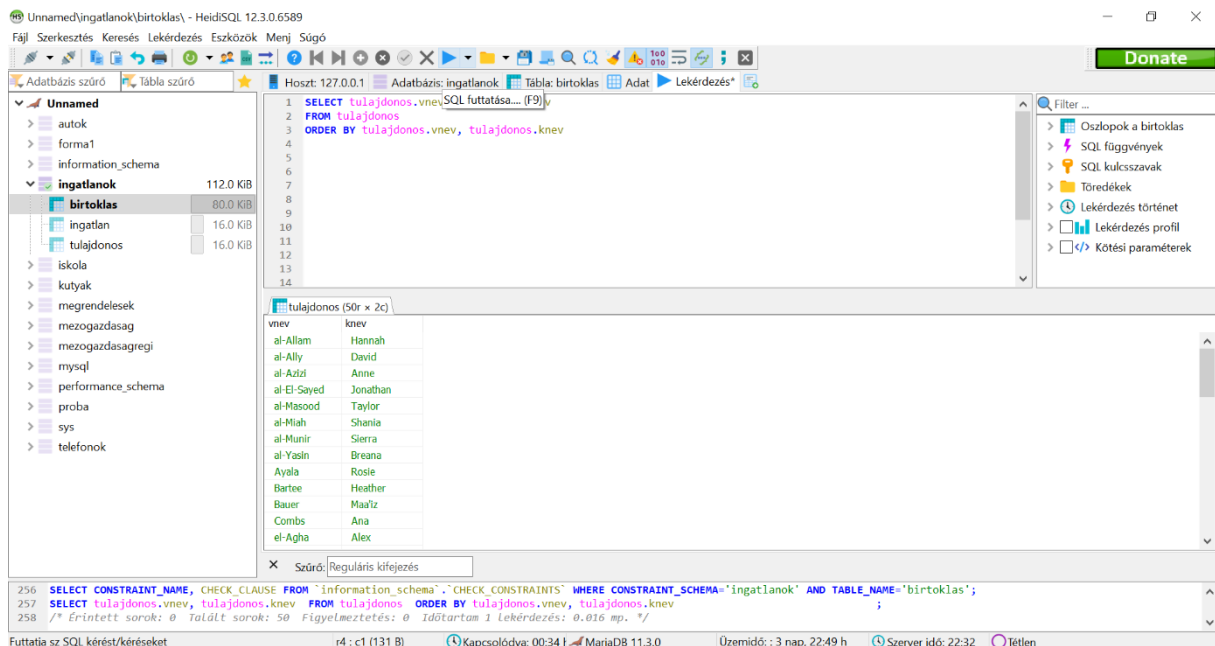
Lekérdezéseket létrehozni Az adott adatbázis kiválasztását követően a lekérdezés fülön tudunk:



Futtatni pedig a menüsor futtatás parancsával tudjuk:



Az eredmény megtekintése:



1.

Az adatbázist a könyvben, az ingatlanok feladatban (lásd fent) részletesen bemutatott megoldásnak megfelelően helyi menük alkalmazásával VAGY a CREATE DATABASE INGATLANOK paranccsal hozzuk létre.

A txt fájlok importálását HeidiSQL kliens esetében a könyvben részletesen bemutatott mintának megfelelően végezzük el.

Felhívjuk a figyelmet, hogy a megoldásban található kód nem egyben futtatható, a feladatkiírásnak megfelelő részletekben kell lefuttatni (pl. külön-külön létrehozva a táblákat, futtatva a lekérdezéseket).

```
CREATE TABLE tulajdonos (
    id INT PRIMARY KEY,
    vnev VARCHAR(20) NOT NULL,
    knev VARCHAR(20) NOT NULL,
    telepules VARCHAR(30) NOT NULL
)
```

```
CREATE TABLE ingatlan (
    id INT PRIMARY KEY,
    telepules VARCHAR(30) NOT NULL,
    utca VARCHAR(20) NOT NULL,
    hazszam INT NOT NULL,
    alapterulet FLOAT NOT NULL DEFAULT(60)
)
```

```
CREATE TABLE birtoklas (
    id INT PRIMARY KEY,
    ingatlanid INT NOT NULL,
    tulajid INT NOT NULL,
    mettol INT NOT NULL,
```

```

        meddig INT NOT NULL,
        FOREIGN KEY (ingatlanid) REFERENCES ingatlan(id),
        FOREIGN KEY (tulajid) REFERENCES tulajdonos(id)
    )

```

2.

```

SELECT tulajdonos.vnev, tulajdonos.knev
FROM tulajdonos
ORDER BY tulajdonos.vnev, tulajdonos.knev

```

3.

```

SELECT tulajdonos.vnev, tulajdonos.knev
FROM tulajdonos
JOIN birtoklas ON tulajdonos.ID = birtoklas.tulajID
WHERE birtoklas.mettol>=2008 AND birtoklas.mettol<=2010

```

4.

```

SELECT ingatlan.alapterulet, ingatlan.telepules, tulajdonos.vnev, tulajdonos.knev
FROM tulajdonos
JOIN birtoklas ON tulajdonos.ID = birtoklas.tulajID
JOIN ingatlan ON ingatlan.ID = birtoklas.ingatlanID
WHERE ingatlan.alapterulet>160 AND ingatlan.telepules <>'Budapest'

```

5.

```

SELECT ingatlan.telepules, Count(ingatlan.ID) AS darab
FROM ingatlan
GROUP BY ingatlan.telepules

```

6.

```

SELECT tulajdonos.vnev, tulajdonos.knev, Max(meddig-mettol) AS tartam
FROM tulajdonos
JOIN birtoklas ON birtoklas.tulajid=tulajdonos.id
JOIN ingatlan ON ingatlanid=birtoklas.ingatlanid
WHERE tulajdonos.telepules IN ('Szombathely', 'Sopron')
GROUP BY tulajdonos.vnev, tulajdonos.knev

```

7.

```

SELECT tulajdonos.vnev, tulajdonos.knev
FROM tulajdonos
JOIN birtoklas ON birtoklas.tulajid=tulajdonos.id
JOIN ingatlan ON ingatlanid=birtoklas.ingatlanid
WHERE      ingatlan.telepules='Budapest'      And      birtoklas.mettol<=2015      And
birtoklas.meddig>=2015
INTERSECT
SELECT tulajdonos.vnev, tulajdonos.knev
FROM tulajdonos
JOIN birtoklas ON birtoklas.tulajid=tulajdonos.id
JOIN ingatlan ON ingatlanid=birtoklas.ingatlanid
WHERE      ingatlan.telepules!='Budapest'      And      birtoklas.mettol<=2015      And
birtoklas.meddig>=2015

```


8.

```
SELECT ingatlan.telepules, ingatlan.alapterulet, tulajdonos.vnev, tulajdonos.knev  
FROM tulajdonos  
JOIN birtoklas ON birtoklas.tulajid=tulajdonos.id  
JOIN ingatlan ON ingatlanid=birtoklas.ingatlanid  
WHERE ingatlan.alapterulet=(SELECT MAX(ingatlan.alapterulet) FROM ingatlan)
```

9.

```
SELECT tulajdonos.vnev, tulajdonos.knev, SUM(ingatlan.alapterulet) AS 'összesített terület'  
FROM tulajdonos  
JOIN birtoklas ON birtoklas.tulajid=tulajdonos.id  
JOIN ingatlan ON ingatlan.id=birtoklas.ingatlanid  
WHERE ingatlan.telepules<>'Budapest'  
GROUP BY tulajdonos.vnev, tulajdonos.knev  
HAVING SUM(ingatlan.alapterulet)>1400
```