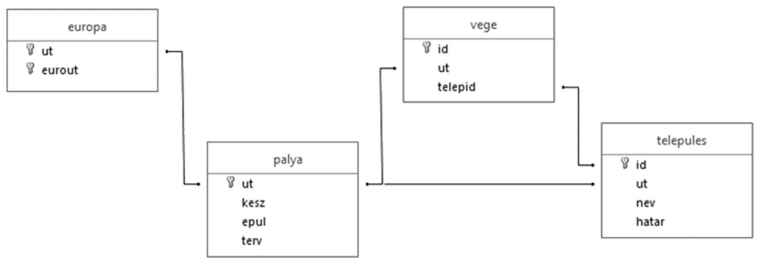
Adatbázis beadandó 2023

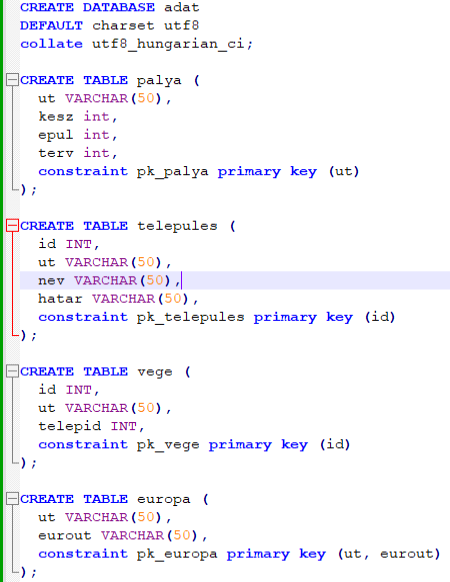
Forrás: <http://informatika.fazekas.hu/wp-content/uploads/2023/02/Magyarorsz%C3%A1g-aut%C3%B3p%C3%A1ly%C3%A1i.pdf>



0.Lépés – Kiválasztjuk az adatbázist.



1.Lépés – Létrehozzuk a táblákat:



CREATE DATABASE adat

DEFAULT charset utf8

collate utf8\_hungarian\_ci;

CREATE TABLE palya (

ut VARCHAR(50),

kesz int,

epul int,

terv int,

constraint pk\_palya primary key (ut)

);

CREATE TABLE telepules (

id INT,

ut VARCHAR(50),

nev VARCHAR(50),

hatar VARCHAR(50),

constraint pk\_telepules primary key (id)

);

CREATE TABLE vege (

id INT,

ut VARCHAR(50),

telepid INT,

constraint pk\_vege primary key (id)

);

CREATE TABLE europa (

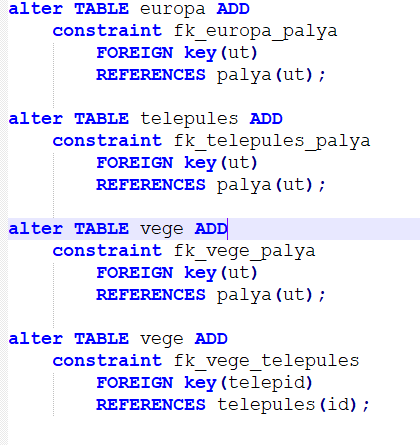
ut VARCHAR(50),

eurout VARCHAR(50),

constraint pk\_europa primary key (ut, eurout)

);

2.Lépés – Kapcsolások létrehozása:



alter TABLE europa ADD

constraint fk\_europa\_palya

FOREIGN key(ut)

REFERENCES palya(ut);

alter TABLE telepules ADD

constraint fk\_telepules\_palya

FOREIGN key(ut)

REFERENCES palya(ut);

alter TABLE vege ADD

constraint fk\_vege\_palya

FOREIGN key(ut)

REFERENCES palya(ut);

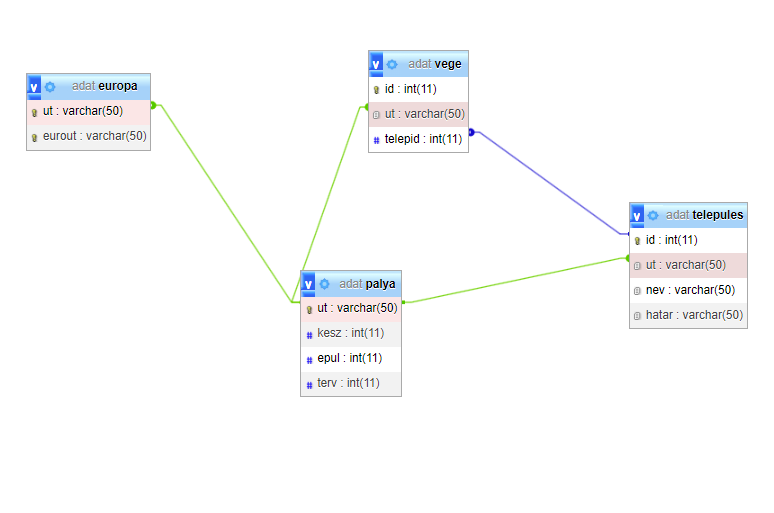
alter TABLE vege ADD

constraint fk\_vege\_telepules

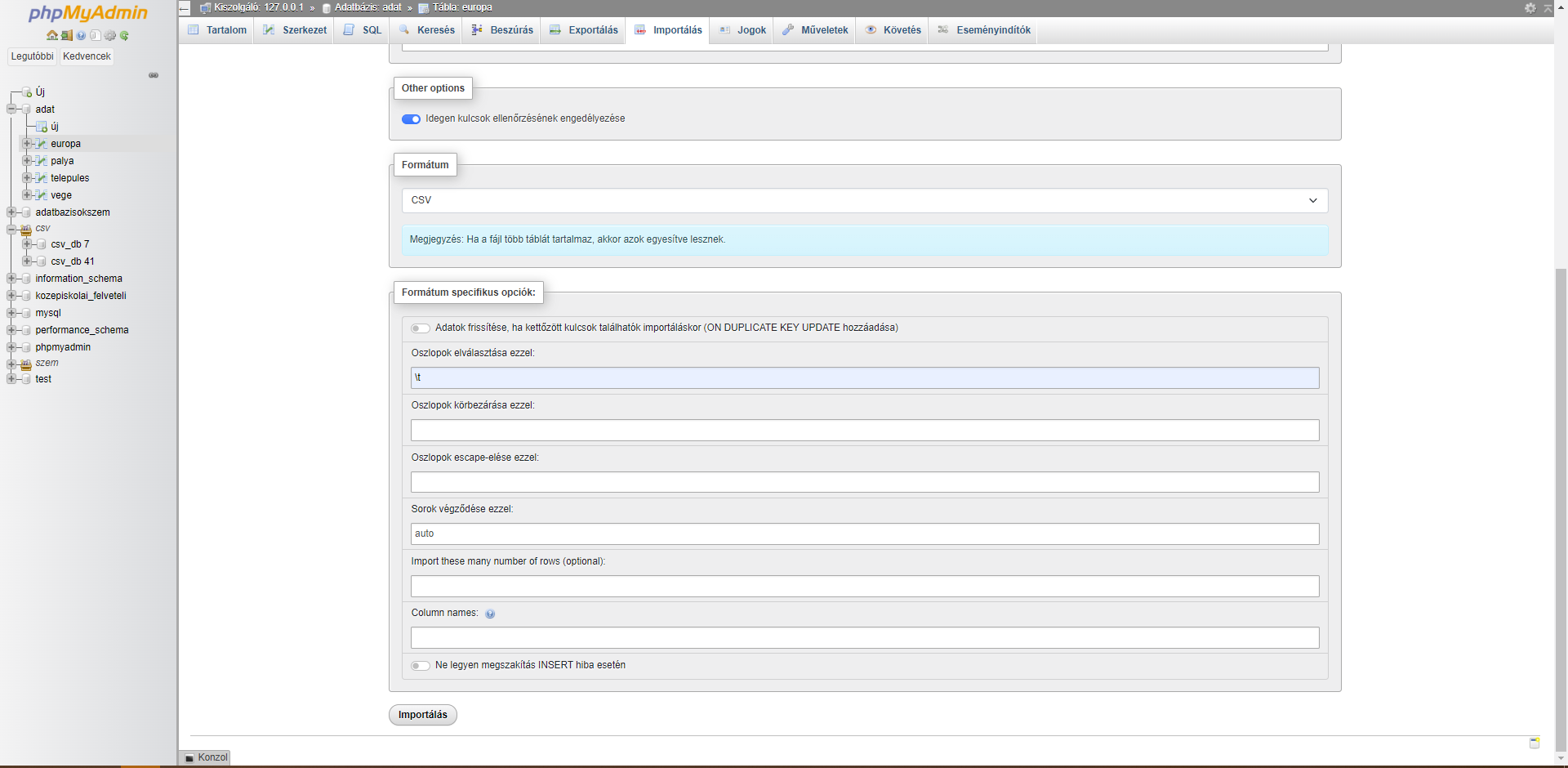
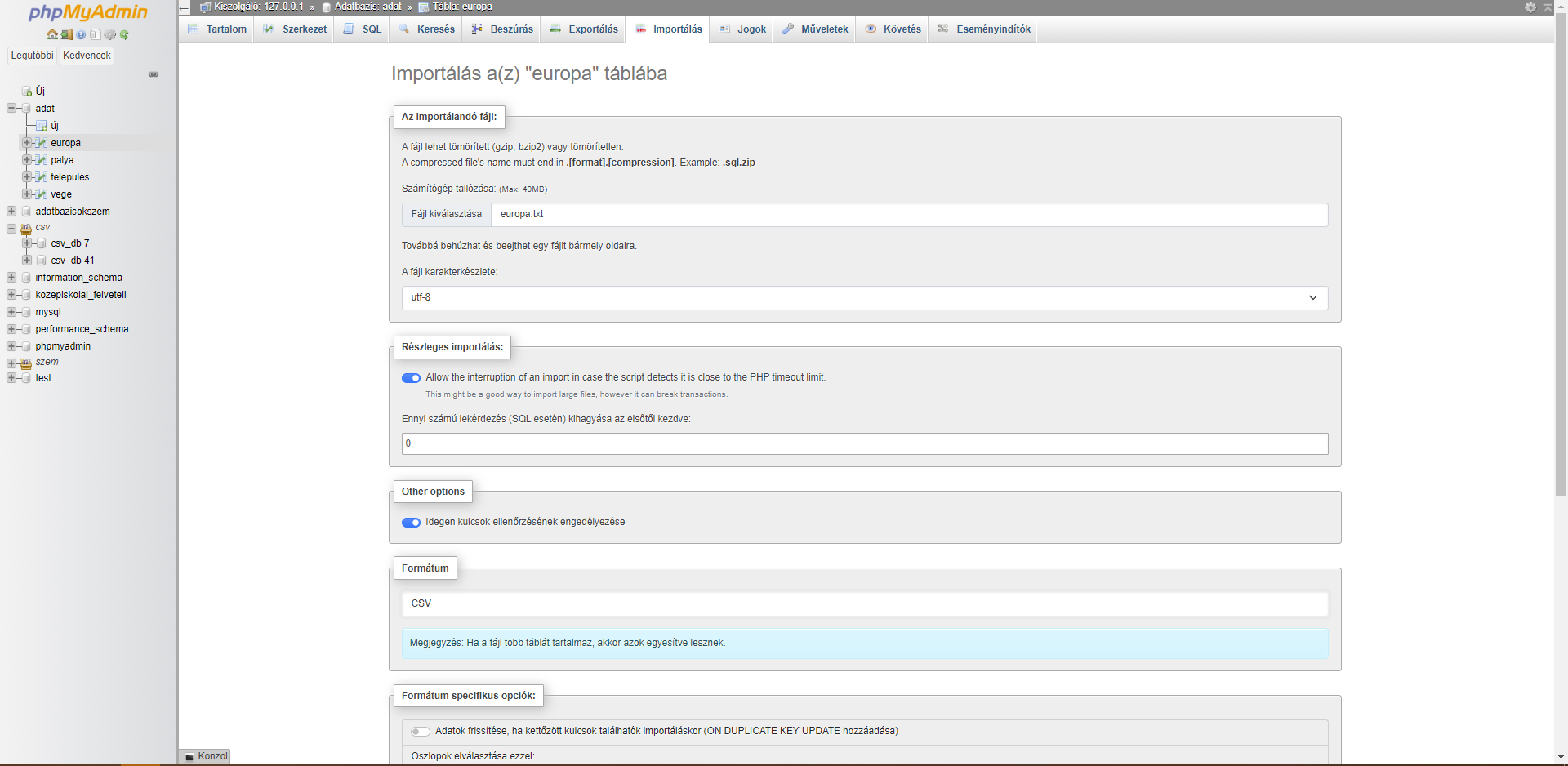
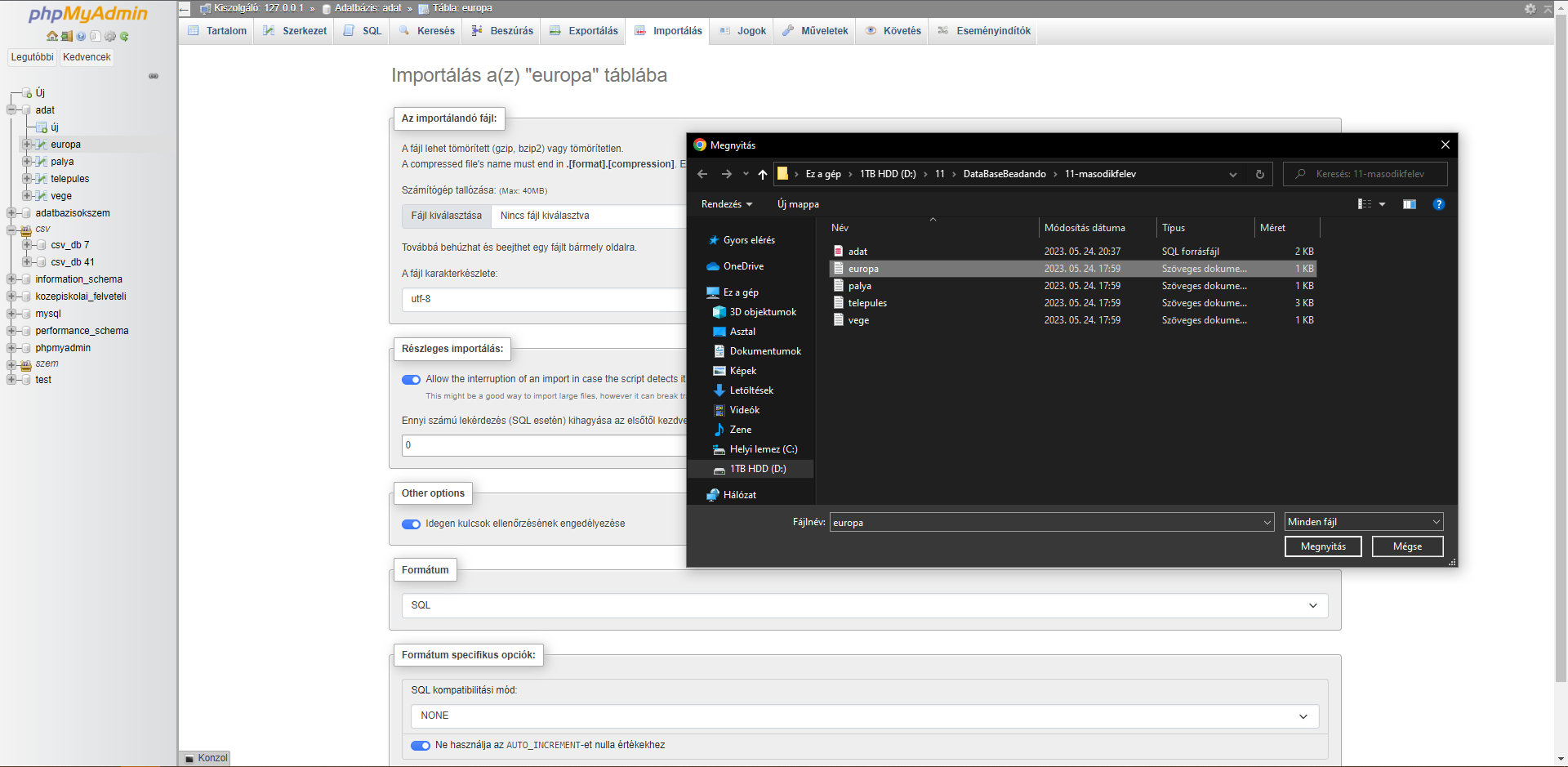
FOREIGN key(telepid)

REFERENCES telepules(id);

2.Lépés eredménye:



3.Lépés – Adatok beszúrás:

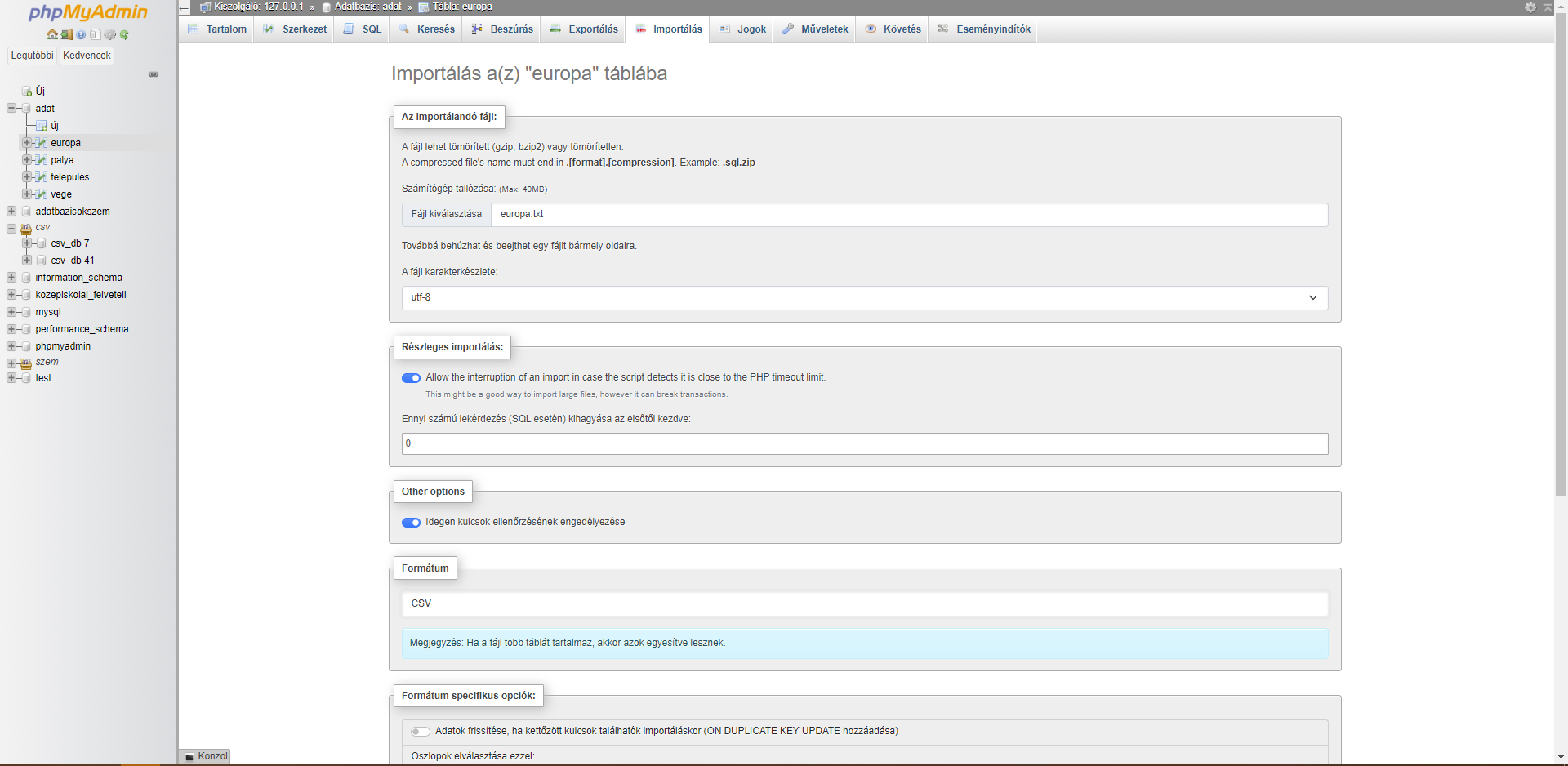
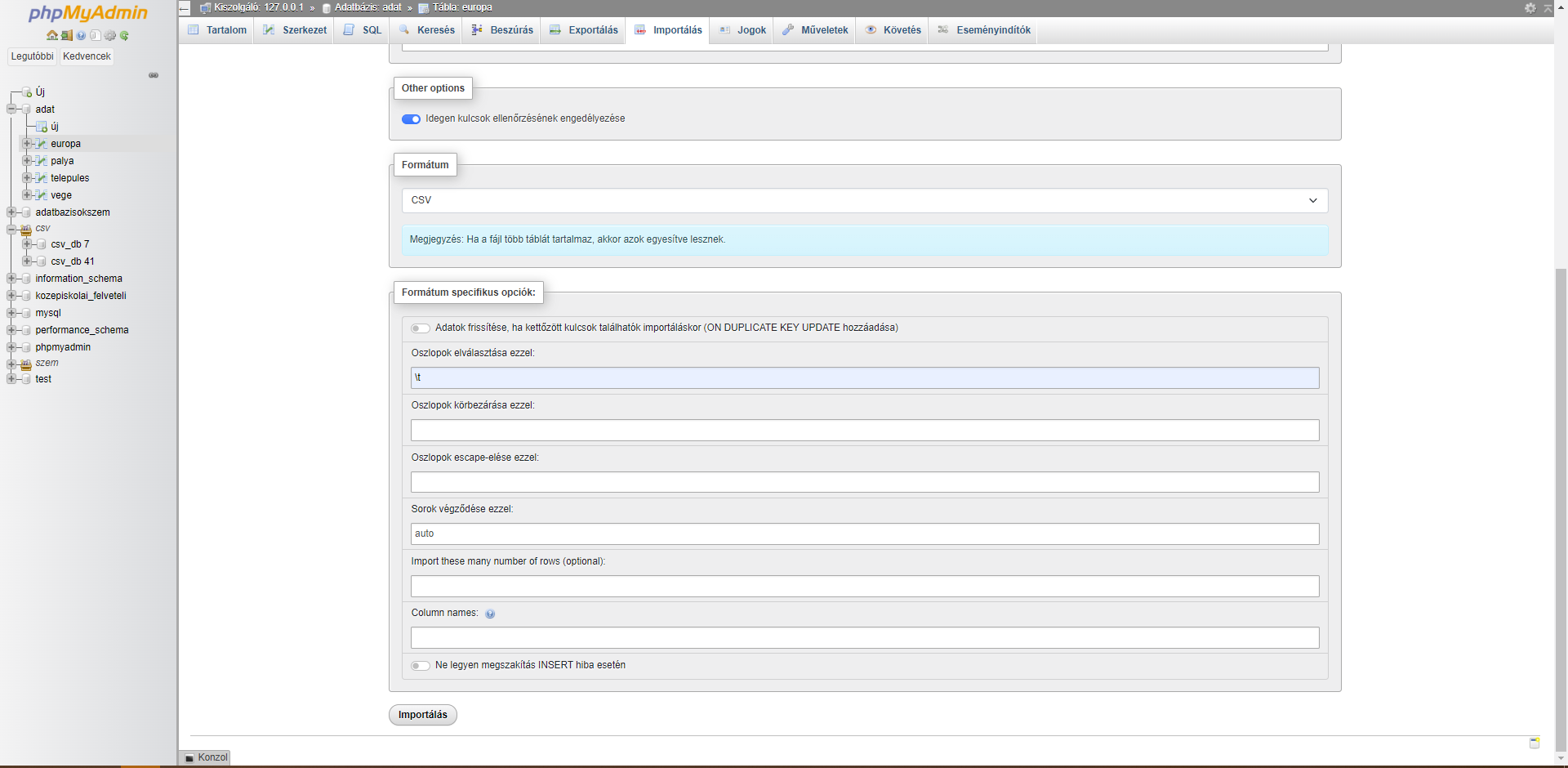


kiválasztom a fájlt

formátumot kiválasztom hogy csv legyen

A formátum specifikus opcióknál az oszlopok elválasztását átírtam \t-re és kitöröltem az oszlopok körbezárását és escape-elését

4.Lépés – Exportálás

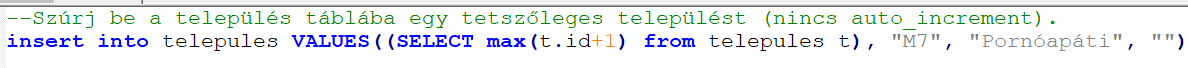


Az exportálásnál elneveztük az exportálni kívánt fájlt, a formátumát SQL-re állítottuk és az exportálás módját pedig egyedire állítjuk

Ezután az objektum-létrehozási beállításokban engedélyezzük a "CREATE DATABASE"-t, vagyis az adatbázis létrehozását, végül pedig az oldal alján az exportálásra nyomunk.

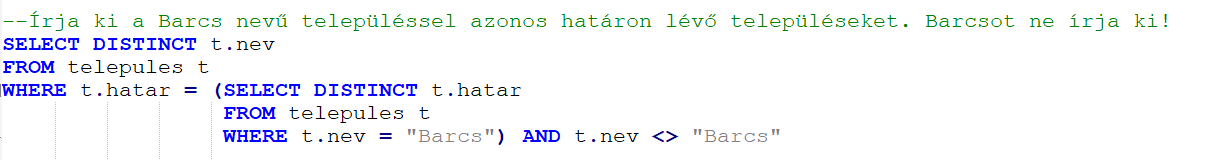
Feladatok és megoldásaik

1.Feladat:



--Szúrj be a település táblába egy tetszőleges települést (nincs auto\_increment).

2.Feladat:



--Írja ki a Barcs nevű településsel azonos határon lévő településeket. Barcsot ne írja ki!

SELECT DISTINCT t.nev

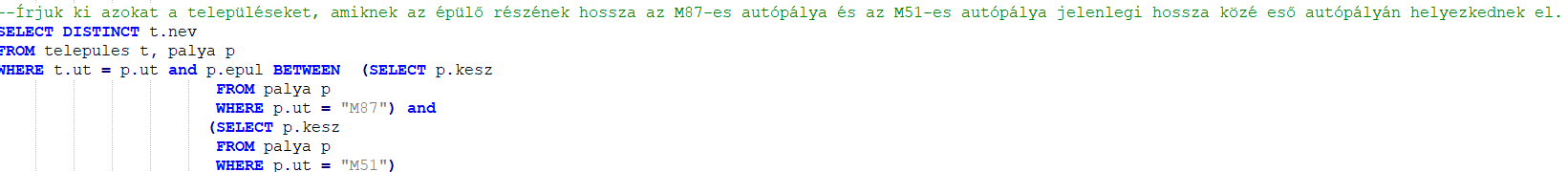
FROM telepules t

WHERE t.hatar = (SELECT DISTINCT t.hatar

FROM telepules t

WHERE t.nev = "Barcs") AND t.nev <> "Barcs"

3.Feladat:



--Írjuk ki azokat a településeket, amiknek az épülő részének hossza az M87-es autópálya és az M51-es autópálya jelenlegi hossza közé eső autópályán helyezkednek el.

SELECT DISTINCT t.nev

FROM telepules t, palya p

WHERE t.ut = p.ut and p.epul BETWEEN (SELECT p.kesz

FROM palya p

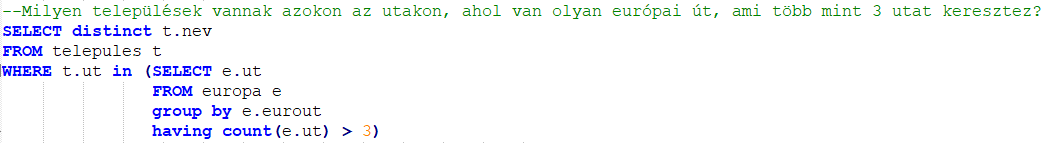
WHERE p.ut = "M87") and

(SELECT p.kesz

FROM palya p

WHERE p.ut = "M51")

4.Feladat:



SELECT distinct t.nev

FROM telepules t

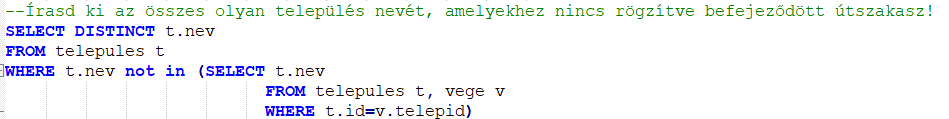
WHERE t.ut in (SELECT e.ut

FROM europa e

group by e.eurout

having count(e.ut) > 3)

5.Feladat:



--Írasd ki az összes olyan település nevét, amelyekhez nincs rögzítve befejeződött útszakasz!

SELECT DISTINCT t.nev

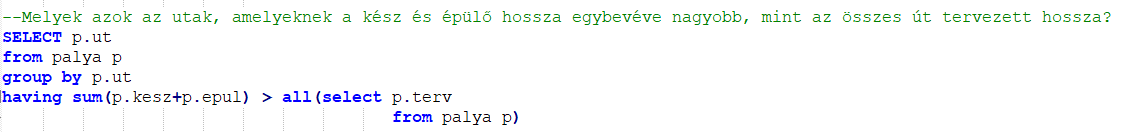
FROM telepules t

WHERE t.nev not in (SELECT t.nev

FROM telepules t, vege v

WHERE t.id=v.telepid)

6.Feladat:



--Melyek azok az utak, amelyeknek a kész és épülő hossza egybevéve nagyobb, mint az összes út tervezett hossza?

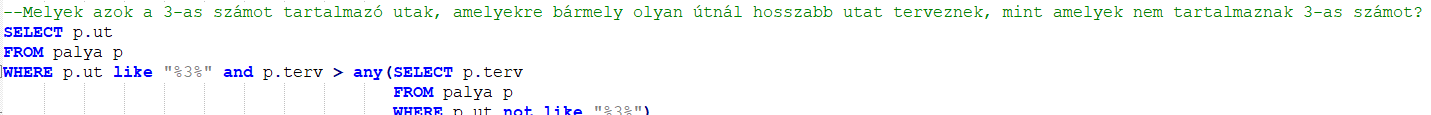
SELECT p.ut

from palya p

group by p.ut

having sum(p.kesz+p.epul) > all(select p.terv from palya p)

7.Feladat:



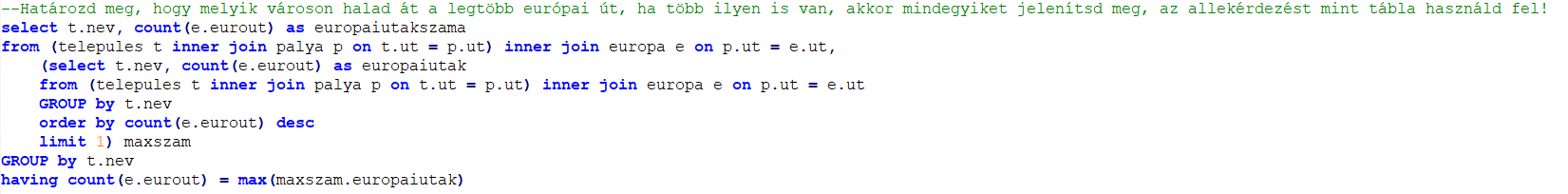
--Melyek azok a 3-as számot tartalmazó utak, amelyekre bármely olyan útnál hosszabb utat terveznek, mint amelyek nem tartalmaznak 3-as számot?

SELECT p.ut

FROM palya p

WHERE p.ut like "%3%" and p.terv > any(SELECT p.terv FROM palya p WHERE p.ut not like "%3%")

8.Feladat:



--Határozd meg, hogy melyik városon halad át a legtöbb európai út, ha több ilyen is van, akkor mindegyiket jelenítsd meg, az allekérdezést mint tábla használd fel!

select t.nev, count(e.eurout) as europaiutakszama

from (telepules t inner join palya p on t.ut = p.ut) inner join europa e on p.ut = e.ut,

(select t.nev, count(e.eurout) as europaiutak

from (telepules t inner join palya p on t.ut = p.ut) inner join europa e on p.ut = e.ut

GROUP by t.nev

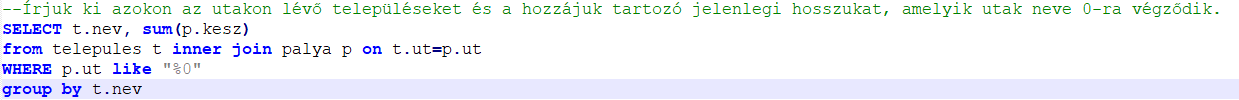
order by count(e.eurout) desc

limit 1) maxszam

GROUP by t.nev

having count(e.eurout) = max(maxszam.europaiutak)

9.Feladat:



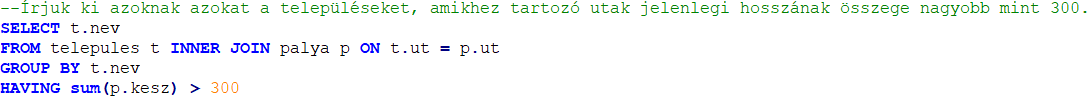
SELECT t.nev, sum(p.kesz)

from telepules t inner join palya p on t.ut=p.ut

WHERE p.ut like "%0"

group by t.nev

10.Feladat:



--Írjuk ki azoknak azokat a településeket, amikhez tartozó utak jelenlegi hosszának összege nagyobb mint 300.

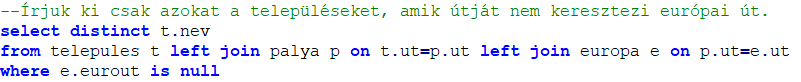
SELECT t.nev

FROM telepules t INNER JOIN palya p ON t.ut = p.ut

GROUP BY t.nev

HAVING sum(p.kesz) > 300

11.Feladat:



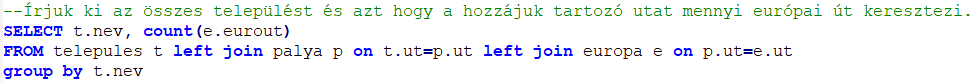
--Írjuk ki csak azokat a településeket, amik útját nem keresztezi európai út.

select distinct t.nev

from telepules t left join palya p on t.ut=p.ut left join europa e on p.ut=e.ut

where e.eurout is null

12.Feladat:



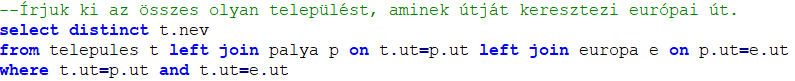
--Írjuk ki az összes települést és azt hogy a hozzájuk tartozó utat mennyi európai út keresztezi.

SELECT t.nev, count(e.eurout)

FROM telepules t left join palya p on t.ut=p.ut left join europa e on p.ut=e.ut

group by t.nev

13.Feladat:



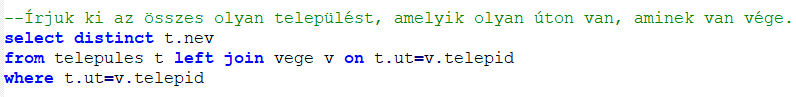
--Írjuk ki az összes olyan települést, aminek útját keresztezi európai út.

select distinct t.nev

from telepules t left join palya p on t.ut=p.ut left join europa e on p.ut=e.ut

where t.ut=p.ut and t.ut=e.ut

14.Feladat:



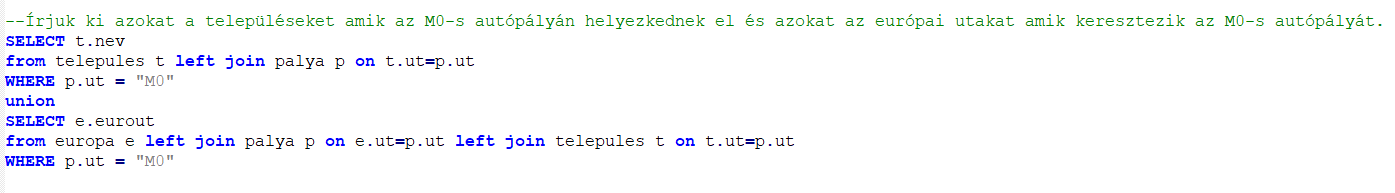
--Írjuk ki az összes olyan települést, amelyik olyan úton van, aminek van vége.

select distinct t.nev

from telepules t left join vege v on t.ut=v.telepid

where t.ut=v.telepid

15Feladat:



--Írjuk ki azokat a településeket amik az M0-s autópályán helyezkednek el és azokat az európai utakat amik keresztezik az M0-s autópályát.

SELECT t.nev

from telepules t left join palya p on t.ut=p.ut

WHERE p.ut = "M0"

union

SELECT e.eurout

from europa e left join palya p on e.ut=p.ut left join telepules t on t.ut=p.ut

WHERE p.ut = "M0"