Adatbázis beadandó

Készítette:

Agócs Dániel

Krausz Márton Alajos

Tartalom

[Adatbázis 2](#_Toc135738548)

[Alter Table-ek 3](#_Toc135738549)

[Feladatok 4](#_Toc135738550)

[Feladatok A-C ig 5](#_Toc135738551)

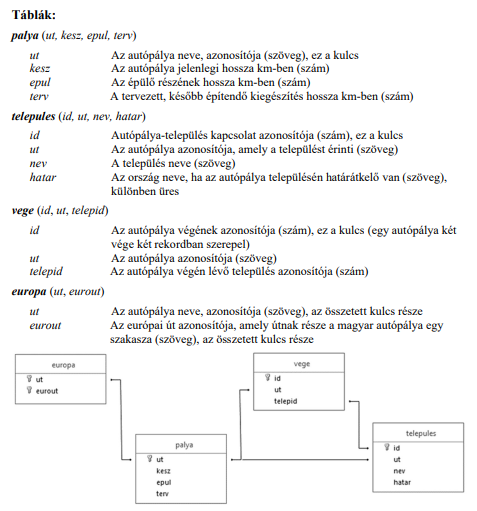
[Feladatok D-F ig 6](#_Toc135738552)

[Feladatok G-I ig 7](#_Toc135738553)

[Feladatok J-L ig 8](#_Toc135738554)

[Feladatok M-O ig 9](#_Toc135738555)

# Adatbázis



# Alter Table-ek

alter table europa add

    constraint fk\_europa\_palya FOREIGN KEY (ut)

    REFERENCES palya(ut)

alter table vege add

    constraint fk\_vege\_palya FOREIGN KEY (ut)

    REFERENCES palya(ut)

alter table telepules add

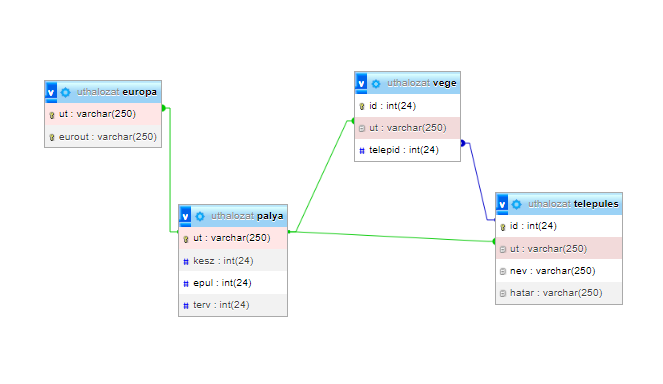
    constraint fk\_telepules\_palya FOREIGN KEY (ut)

    REFERENCES palya(ut)

alter table vege add

    constraint fk\_vege\_telepules FOREIGN KEY (telepid)

    REFERENCES telepules(id)



# Feladatok

**Beadandó követelmények**

A beadandónál, mindent dokumentálj, minden lépésedhez szülessen egy rövid feladatleírás is, mely után tartalmazza a megoldás lépéseit, vagy épp az SQL kódját.

1. Keress egy olyan adatbázis, ami legalább 4 táblából áll (vagy olyat, ami 3 táblából áll, amit 4-re bővítesz) és még nem használtuk az órán, nincs az órán kapott adatbázisok között. A forrást jelöld meg!
2. Az eddig tanult műveletekre találj ki egy-egy feladatot és oldd is meg.
   1. Új mező beszúrása egy olyan táblába, ahol nincs auto\_increment
   2. Összetett lekérdezés, ahol az allekérdezés, mint érték szerepeljen
   3. Összetett lekérdezés, ahol az allekérdezés, mint érték szerepeljen (többszörösen összetett – legalább két allekérdezés)
   4. Összetett lekérdezés, ahol az allekérdezés egy listával tér vissza: IN
   5. Összetett lekérdezés, ahol az allekérdezés egy listával tér vissza: NOT IN
   6. Összetett lekérdezés, ahol a „minden” feltételt után kerül az allekérdezés (ALL)
   7. Összetett lekérdezés, ahol a „bármelyik” feltételt után kerül az allekérdezés (ANY)
   8. Összetett lekérdezés, ahol az allekérdezés, mint tábla fog szerepelni.
   9. Egyszerű csoportosító lekérdezés + aggregátum függvények + INNER JOIN kapcsolattal, csoportosítás előtti feltétellel.
   10. Egyszerű csoportosító lekérdezés + aggregátum függvények + INNER JOIN kapcsolattal, csoportosítás utáni feltétellel.
   11. Választó lekérdezés LEFT JOIN kapcsolattal + csak az egyik tábla adatai
   12. Választó lekérdezés LEFT JOIN kapcsolattal + csoportsítás
   13. Választó lekérdezés LEFT JOIN kapcsolattal + több táblával
   14. Választó lekérdezés LEFT JOIN kapcsolattal + minden adat az egyik táblából
   15. Összetett lekérdezés UNION (Full Outer Join) segítségével.

# Feladatok A-C ig

--a

--Egy adat beszúrása

insert into telepules values(

    (select max(t.id) + 1

    from telepules t), "M32", "Egy nagyon jó város név",null)

--b

--Melyik útnak van a leghosszabb kész része?

--Ha több mint egy akkor az összessel térjen vissza a lekérdezés!

select p.ut

from palya p

where p.kesz = (select max(p.kesz)

                from palya p)

--c

--Melyik autopalyaknak van hosszab tervezet resze mint a leghoszabb autopályának(a kész részét figyelembe véve) a tervezet része?

--Ha több mint egy akkor az összeset!

select p.ut

from palya p

where p.terv > (

    select p.terv

    from palya p

    where p.kesz = (

        select max(p.kesz)

        from palya p)

    )

# Feladatok D-F ig

--d

--Válaszuk ki azokat a magyar utakat amelyek az E60-s európai úthálózatba tartoznak!

select p.ut

from palya p

where p.ut in (select e.ut

               from europa e

               where e.eurout = "E60")

--e

--Válasszuk ki az európai úthálózatnak azokat a részeit amelyeknek nincs egyszerre tervezet és epülö része.

select e.eurout

from europa e

where e.ut not in (select p.ut

               from palya p

               where p.terv > 0 and p.epul > 0)

--f:

--Válassza ki az összes utat amiknek a kész hossza nagyobb mint az m1-nek, vagy az m0-nak a kész hossza.

select p.ut, p.kesz

from palya p

where kesz > ALL(

    select p.kesz

    from palya p

    where p.ut = "M0" or p.ut = "M1"

);

# Feladatok G-I ig

--g

--Mely két település található az m7-es autópálya két végén?

select t.nev

from telepules t

where t.id = ANY (

    select v.telepid

    from vege v

    where v.ut = 'M7'

);

--h

-- Válassza ki az összes olyan települést amelynek a hozzá tartozó autópályája hosszab mint 150km.

select t.nev

FROM (

   select p.ut

   from palya p

   where p.kesz > 150

) AS p

join telepules t on p.ut = t.ut

--i

--A magyar uthálózat utjaiként hány olyan európai úthálózathoz tartozó út van amelynek az épülő része kisebb mint a tervezett része?

--Csak az egy vagy annál nagyobb értékekkel térjen vissza a lekérdezés.

select p.ut, count(p.ut) as darab

from palya p inner join europa e on e.ut = p.ut

where p.epul < p.terv

group by p.ut

order by p.kesz desc

# Feladatok J-L ig

--j

--Melyek azok a magyar utak amelyek legfeljebb 3 európai úthoz kapcsolódnak és a tervezett része kisebb mint 35km és mennyi tartozik ezekhez?

--Csökkenő sorrendben jelenjenek meg!

select p.ut, count(p.ut) as darab

from palya p inner join europa e on e.ut = p.ut

where p.terv < 35

group by p.ut

having count(p.ut) <= 3

order by darab desc

--k

--Melyek azok a települések amelyek nem határosak semmivel vagy az autópálya végén helyezkednek el?

--Minden település maximum egyszer jelenjen meg!

select distinct t.nev

from telepules t

left join vege v on t.id = v.telepid

where t.hatar is null or v.id is not null;

--l

--Melyek azok a települések amelyek nem határosak semmivel, vagy a hozzájuk tartozó autopálya végén helyezkednek el ekkor az is térjen vissza hogy hány autopályának a vége az a település?

select t.nev, count(\*)

from telepules t

left join vege v on t.id = v.telepid

where t.hatar is null or v.id is not null

group by t.nev;

# Feladatok M-O ig

--m

--Melyek azok az európai utak amelyeknek a hozzájuk tartozó települése(i) tartalmaznak "t"-t?

--A lekérdezés térjen vissza az úttal és a hozzá tartotó, feltételnek megfelelő településsel.

select e.eurout, t.nev

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

where lower(t.nev) like "%t%"

--n

--Melyek azok az autópályák amelyek nem részei az európai úthálózatnak?

select distinct p.\*

from palya p left outer join europa e on e.ut = p.ut

where e.eurout is null

--o

--Írja ki az összes olyan T-vel kezdodo telepulést ami az m0-ás mellet található, illetve az összes olyan B-vel kezdodo telepulést ami az m3-ás mellet található

select t.nev

from telepules t, vege v

where t.ut = v.ut

and t.nev like "T%"

and v.ut = "M0"

Union

select t.nev

from telepules t, vege v

where t.ut = v.ut

and t.nev like "B%"

and v.ut = "M3"