

MEGOLDÁSOK (Szi Brigitta)

```
--Feladat:
--Készítsünk lekérdezést, amely kilistázza az összes hallgató adatait.

--Megoldás:
Select * --A select segítségével listázunk a * az összes adat listázására szolgál
from hallgatok; --megadjuk mely táblát használjuk

--Feladat:
--Készítsünk lekérdezést, amely kilistázza a győri hallgatók adatait.

--Megoldás:
Select *
from hallgatok
where telepules='Győr'; --itt már egy feltételt is megadtunk '' közé rakjuk, mivel ez
a mezőben szereplő érték

--Feladat:
--Készítsen lekérdezést, amely listázza a 2019. 01.12. dátumú vizsgákat.

--Megoldás:
Select *
from tantargyak
where vdatum='2019-01-12';

--Feladat:
--Készítsen lekérdezést, amely megjeleníti, hogy hány darab hallgatónk van összesen.

--Megoldás:
Select COUNT(*) AS 'Hallgatók száma' -- A COUNT() függvény segítségével megszámoljuk a
sorokat
from hallgatok;

--Feladat:
--Készítsen lekérdezést, amely megjeleníti hány darab tárgy van összesen.

--Megoldás:
Select COUNT(*) AS 'Tantárgyak száma'
from tantargyak;

--Feladat:
--Készítsen statisztikát, amely a tantárgyi követelmény típusonként megjeleníti a
tantárgyak számát (azaz, mennyi tantárgynak előfeltétele egy adott másik tárgy, és
melyik az a tárgy).

--Megoldás:
Select tkov AS 'Tantárgyi követelmény kódja', COUNT(tkov) AS 'rá épülő tárgy
darabszáma'
from tantargyak
GROUP BY tkov; -- A Group by segítségével csoportosítást végzünk.

--Feladat:
--Készítsen lekérdezést, amely listázza az adatbázisban lévő összes NGB kezdetű
tantárgyat.

--Megoldás:
Select *
from tantargyak
```

where tkod like 'NGB%'; --Mivel nem konkrét tárgykódot adunk meg csak egy részét ezért a like a használatos, illetve a % több karakterre utal a B után, tehát bármennyi karakter előfordulhat még.

--Feladat:

--Készítsen lekérdezést, amely listázza az adatbázisban lévő összes Ádám keresztnévű hallgatót.

--Megoldás:

```
Select *
from hallgatok
where knev like '%Ádám%';
```

--Feladat:

--Készítsen lekérdezést, amely megmutatja, hogy hányszor utóvizsgáztak a hallgatók összesen.

--Megoldás:

```
Select COUNT(uv) AS 'Hányszor utóvizsgáztak már a hallgatók'
from jegyek
where uv <> 0;
```

--Feladat:

--Készítsen statisztikát, amely megmutatja, hogy az egyes képzési formákon hány darab hallgató vesz részt. (BSC, MSC, PHD egyaránt)

--Megoldás:

```
Select CASE kforma
        When 'Bsc' then 'Alap képzés'
        When 'Msc' then 'Mester képzés'
        When 'Phd' then 'Doktori képzés'
        ELSE 'Kódhiba'
        END AS 'Képzési forma', --Átírtuk a képzési neveket
        COUNT(*) AS 'Hallgatók száma'
from hallgatok
GROUP BY kforma;
```

--Feladat:

--Kérdezze le a hallgatók nevét, hallgatói azonosítóját és jegyeinek egyszerű számtani átlagát.

--Megoldás:

```
Select vnev, knev, hallgatok.neptun, AVG(jegy) AS 'Átlag' --Az AVG() függvény segítségével számolunk átlagot (jelen esetben az eredményből is láthatjuk, hogy egész számot kapunk eredményül átlagnak).
FROM jegyek inner join hallgatok --kapcsolódó feltétel(jegyek és hallgatók közös része a nekünk fontos jelen esetben, a közös elem a két táblában pedig a neptun)
on jegyek.neptun = hallgatok.neptun
GROUP BY vnev, knev, hallgatok.neptun; --csoportosítást használunk, hogy csakis az adott személyhez tartozó érdemjegyeket számoljuk, ne pedig bárkiét.
```

--Amit még fontos megjegyezni, hogy ha csak neptun-t írnék akkor nem működik, hiszen kettő táblában is megtalálható, ezért kell megadni hogy jegyek.neptun vagy hallgatok.neptun-t szeretnénk.

--Feladat:

--Kérdezze le a hallgatók nevét, hallgatói azonosítóját és jegyeinek egyszerű számtani átlagát. Az előző feladatban egész számot kaptunk átlagként, ami nem éppen valós, kérem jelenítse meg a tizedes értékeket is.

--Megoldás:

Select vnev, knev, h.neptun, AVG(CAST(jegy as float)) AS 'Átlag' --Az AVG() függvény segítségével számolunk átlagot (jelen esetben már konvertáltunk is, így látható a pontos átlag).

FROM jegyek j inner join hallgatok h --kapcsolódó feltétel (jegyek és hallgatók közös része a nekünk fontos jelen esetben, a közös elem a két táblában pedig a neptun)
on j.neptun = h.neptun

GROUP BY vnev, knev, h.neptun; --csoportosítást használunk, hogy csakis az adott személyhez tartozó érdemjegyeket számoljuk, ne pedig bárkiét.

--itt szeretném megjegyezni, hogy már nem hallgatok.neptun hanem csak h.neptun-t használtam, hiszen a FROM jegyek j inner join hallgatok h részben megadtam hogy rövidítettem a táblák nevét j illetve h betűkkel.

--Feladat:

--Kérdezze le a hallgatók nevét, hallgatói azonosítóját és jegyeinek egyszerű számtani átlagát, jelenítse meg átlag szerint csökkenő sorrendben az eredményt.

--Megoldás:

Select vnev, knev, h.neptun, AVG(CAST(jegy as float)) AS 'Átlag' --Az AVG() függvény segítségével számolunk átlagot (jelen esetben már konvertáltunk is, így látható a pontos átlag).

FROM jegyek j inner join hallgatok h --kapcsolódó feltétel (jegyek és hallgatók közös része a nekünk fontos jelen esetben, a közös elem a két táblában pedig a neptun)
on j.neptun = h.neptun

GROUP BY vnev, knev, h.neptun --csoportosítást használunk, hogy csakis az adott személyhez tartozó érdemjegyeket számoljuk, ne pedig bárkiét.

ORDER BY 4 DESC; --Az order by segítségével rendezünk, a 4 szám azt jelöli, hogy hányadik oszlop alapján rendezzük az egész eredmény táblánkat, megadhattam volna így is : ORDER BY AVG(CAST(jegy as float)) DESC; A DESC pedig a csökkenő sorrend = Descending. ASC = Ascending azaz növekvő sorrend.

--Feladat:

--Kérdezzük le a hallgató neptunkódját és tanulmányi átlagát.

--Megoldás:

Select h.neptun, AVG(CAST(jegy as float)) AS 'Átlag' --Az AVG() függvény segítségével számolunk átlagot (jelen esetben már konvertáltunk is, így látható a pontos átlag).

FROM jegyek j inner join hallgatok h --kapcsolódó feltétel (jegyek és hallgatók közös része a nekünk fontos jelen esetben, a közös elem a két táblában pedig a neptun)
on j.neptun = h.neptun

GROUP BY h.neptun --csoportosítást használunk, hogy csakis az adott személyhez tartozó érdemjegyeket számoljuk, ne pedig bárkiét.

--Feladat:

--Kérdezzük le a hallgatók nevét, hallgatói azonosítóját (neptunkód), majd jegyeik egyszerű számtani átlagát, átlag szerint csökkenő sorrendbe rendezzük a végeredményt, de jelen esetben csak a legutoljára szerzett érdemjegyüket vegyük figyelembe az átlag számítás során, ha többször is voltak vizsgázni egy tárgyból.

--Megoldás:

--Az elején meg kell vizsgáljuk, hogy az utoljára szerzett érdemjegyekhez már nem elég a 2 tábla, hiszen meg kell nézzük a tárgyakat külön és az azokból szerzett legutolsó érdemjegyekkel számolunk csak.

Select vnev, knev, h.neptun, AVG(CAST(jegy as float)) AS 'Átlag' --Az AVG() függvény segítségével számolunk átlagot (jelen esetben már konvertáltunk is, így látható a pontos átlag).

FROM jegyek j, hallgatok h, tantargyak t --kapcsolódó feltétel, ha jobban megnézik most nem joint használunk csak a WHERE szócskát

WHERE j.neptun = h.neptun and j.tkod = t.tkod --a közös elemek a táblákban

```

and
uv=(Select MAX(uv) --ebben a részben egy al-select létrehozására van szükségünk, ahol
kiválasztjuk az utolsó vizsga jegyeket és kizárólag azzal számolunk.
from jegyek j2 --mivel fentebb a külső selectben már használtunk a jegyek táblát így
más rövidítést adok neki, és megmondom a következő sorban, hogy a két táblám ugyanaz
where j2.neptun=j.neptun and j2.tkod=j.tkod)
GROUP BY vnev, knev, h.neptun --csoportosítást használunk, hogy csakis az adott
személyhez tartozó érdemjegyeket számoljuk, ne pedig bárkiét.
ORDER BY 4 DESC; --Az order by segítségével rendezünk, a 4 szám azt jelöli, hogy
hányadik oszlop alapján rendezzük az egész eredmény táblánkat, megadhattam volna ígyis
: ORDER BY AVG(CAST(jegy as float)) DESC; A DESC pedig a csökkenő sorrend =
Descending. ASC = Ascending azaz növekvő sorrend.

```

```

--Mivel NULL értékek szerepelnek az uv értékeknél, ezért az előbbi eredmény nem
pontos, azonban ha átírjuk 0 értékekre őket, akkor pontos eredményt kapunk.
--A következő Update segítségével módosíthatjuk ezeket az értékeket:

```

```

Update jegyek
SET uv='0' from jegyek
Where uv is NULL

```

```

--Feladat:

```

```

--Kérdezzük le a hallgatók nevét, hallgatói azonosítóját (neptunkód), majd a hallgatók
súlyozott tanulmányi átlagát, átlag szerint csökkenő sorrendbe rendezzük a
végeredményt, de jelen esetben csak a legutoljára szerzett érdemjegyüket vegyük
figyelembe az átlag számítás során, ha többször is voltak vizsgázni egy tárgyból.

```

```

--Megoldás:

```

```

--Tudnunk kell, hogy a súlyozott átlag kiszámítása a következő képlet alapján történik
SUM(jegy*kredit)/Sum(kredit).

```

```

Select vnev, knev, h.neptun, SUM(CAST(jegy as float)*kredit)/SUM(kredit) AS 'Súlyozott
átlag'

```

```

FROM jegyek j, hallgatók h, tantargyak t --kapcsolódó feltétel, ha jobban megnézik
most nem joint használunk csak a WHERE szócskát

```

```

WHERE j.neptun = h.neptun and j.tkod = t.tkod --a közös elemek a táblákban

```

```

and

```

```

uv=(Select MAX(uv) --ebben a részben egy al-select létrehozására van szükségünk, ahol
kiválasztjuk az utolsó vizsga jegyeket és kizárólag azzal számolunk.

```

```

from jegyek j2 --mivel fentebb a külső selectben már használtunk a jegyek táblát így
más rövidítést adok neki, és megmondom a következő sorban, hogy a két táblám ugyanaz
where j2.neptun=j.neptun and j2.tkod=j.tkod)

```

```

GROUP BY vnev, knev, h.neptun --csoportosítást használunk, hogy csakis az adott
személyhez tartozó érdemjegyeket számoljuk, ne pedig bárkiét.

```

```

ORDER BY 4 DESC;

```

```

--Feladat:

```

```

--Kérdezzük le a 3 legjobb tanuló nevét, azonosítóját és tanulmányi átlagát.

```

```

--Megoldás:

```

```

--Az előző megoldásból induljunk ki, az első 3 sor tulajdonképpen meg is adja a
legjobb 3 tanulót.

```

```

--Ha valaki benne van a legjobbak között azt mondják, hogy benne van a TOP-ban. -->
legjobb 3 --> TOP 3 (Fontos, hogy van szóköz és mindjárt a SELECT után írjuk!)

```

```

Select TOP 3 vnev, knev, h.neptun, SUM(CAST(jegy as float)*kredit)/SUM(kredit) AS
'Súlyozott átlag'

```

```

FROM jegyek j, hallgatók h, tantargyak t

```

```

WHERE j.neptun = h.neptun and j.tkod = t.tkod

```

```

and

```

```

uv=(Select MAX(uv)

```

```

from jegyek j2

```



```
where j2.neptun=j.neptun and j2.tkod=j.tkod)
GROUP BY vnev, knev, h.neptun
ORDER BY 4 DESC;
```

--Feladat:

--Kérdezzük le a győri hallgatók nevét, neptunkódját és tanulmányi átlagát.

--Megoldás:

```
Select vnev, knev, h.neptun, SUM(CAST(jegy as float)*kredit)/SUM(kredit) AS 'Súlyozott
átlag'
FROM jegyek j, hallgatók h, tantárgyak t
WHERE j.neptun = h.neptun and j.tkod = t.tkod
and telepules='Győr' and --ide bekerült az újabb feltétel, hogy a hallgatónk legyen
győri
uv=(Select MAX(uv)
from jegyek j2
where j2.neptun=j.neptun and j2.tkod=j.tkod)
GROUP BY vnev, knev, h.neptun
ORDER BY 4 DESC;
```

--Feladat:

--Kérdezzük le a győri hallgatók nevét, neptunkódját és tanulmányi átlagát, akik
átlaga eléri a 3-t.

--Megoldás:

```
Select vnev, knev, h.neptun, SUM(CAST(jegy as float)*kredit)/SUM(kredit) AS 'Súlyozott
átlag'
FROM jegyek j, hallgatók h, tantárgyak t
WHERE j.neptun = h.neptun and j.tkod = t.tkod
and
uv=(Select MAX(uv)
from jegyek j2
where j2.neptun=j.neptun and j2.tkod=j.tkod)
GROUP BY vnev, knev, h.neptun
Having SUM(CAST(jegy as float)*kredit)/SUM(kredit)>=3 --ide kerül az átlagra érvényes
feltételünk, hogy a számítások után az átlag érje el a minimum a 3-at.
ORDER BY 4 DESC;
```

--Feladat:

--Kérdezzük le azon hallgatók adatait, akik még nem szereztek érdemjegyet.

--Megoldás:

--Azok a hallgatók, akik még nem szereztek érdemjegyet sosem, azok nem szerepelnek
értelem szerűen a jegyek táblában sem.
--Tehát meg kell néznünk mely hallgatók szerepelnek a hallgatók táblában de a jegyek
táblában nem. Ezt egy újabb aselect segítségével tesszük.

```
Select *
from hallgatók
where neptun not in (select neptun from jegyek); --itt látható
```

--Feladat:

--Kérdezzük le, hogy eddig hány hallgató tett vizsgát.

--Megoldás:
--Az előző feladatból kiindulva ezek a hallgatók már szerepelnek a jegyek táblában.
--Viszont vigyáznunk kell, mert egy hallgató többször is, sőt több tárgyból is
vizsgálhat.
--Amiatt, hogy ne számoljunk egy hallgatót többször a DISTINCT segítségével szűrjük
őket.

```
Select Count (distinct neptun)  
from jegyek
```