

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 vizsgázní 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 alselect 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
```

```
--Feladat:
--A lányoknak vagy fiúknak (választhatnak) adjanak minden tárgyból egy jeggyel jobb
jegyet.
--(Tipikus DML utasításról van szó - nem lekérdezni, hanem módosítani kell.)
```

```
--Megoldás:
Update jegyek --mivel módosítani fogunk a táblában, ezért az update használat
SET jegy=jegy+1 --megadjuk mégis mit módosítunk és hogyan
From hallgatok inner join jegyek
on hallgatok.neptun=jegyek.neptun
where lany='false' and jegy<5; --feltétel, legyen fiú, vagy lány, amit szeretnénk,
illetve azért 5-nél nagyobb érdemjegy ne lehessen, hiszen nincs.
```

```
--Feladat:
--Az előző feladatot picit egészítsük ki: Adjanak a lányoknak vagy fiúknak egy jeggyel
jobb jegyet úgy, hogy csak a legutoljára szerzett érdemjegyüket módosítsuk az egyes
tárgyakból.
```

```
--Megoldás:
Update jegyek
set jegy=jegy+1
from hallgatok inner join jegyek
on hallgatok.neptun=jegyek.neptun
where lany='false' and jegy<5 and
uv=(Select max(uv) from jegyek j2
where j2.neptun=jegyek.neptun and j2.tkod=jegyek.tkod);
--régebbiről gondolom az utolsó jegy megállapítása már ismerős, így itt ezt külön nem
magyarázom már.
```

```
--Feladat:
--Töröljenek egy választott vizsgadátumot a rendszerből (pl. 2018.06.17.)
```

```
--Megoldás:
--Megjegyzés: 3 DML utasítást használunk Insert, Update és a Delete
Delete from jegyek
where vdatum='2018-06-17';
```

```
--Feladat:
-- Kérdezzék le az első oszlopba a hallgatók teljes nevét, a második oszlopba pedig
a teljes címüket.
```

```
--Megoldás:
--Összefűzést fogunk alkalmazni
Select vnev+ ' '+knev AS 'Név', utca+ ' '+telepules+ ' '+irsz AS 'Cím' --Az oszlopok
összefűzése a + segítségével történik, a + + jelek közé Apostrofok között egy üres
karaktert helyezünk el, máskülönben folyamatosan egybeírná csak az adatokat.
from hallgatok;
```

```
--Feladat:
```

-- Kérdezzék le a legtöbb testvérrel rendelkező hallgatók névsorát (a testvérek számát is jelenítsék meg).

--Megoldás:

```
Select vnev+' '+knev AS 'Név',tszam AS 'Testvérek száma'
from hallgatok
```

```
where tszam=(Select max(tszam) from hallgatok)
```

--ugyanazon az elven működik, mint a maximális uv szám kiválasztása az al-select segítségével.

--Feladat:

-- Listázzák ki a hallgatókat, a tantárgyakat és a megszerzett jegyeket is. Azok a hallgatók is jelenjenek meg, akiknek még nincs érdemjegyük, és azok a tárgyak is jelenjenek meg, melyekből még senki sem vizsgázott. (segítség - FULL JOIN)

--Megoldás:

--ha jobban megnézzük tulajdonképpen mindhárom táblánk adatait szeretnénk látni, emiatt a full joint használjuk.

```
Select *
from hallgatok h full join jegyek j
on h.neptun=j.neptun full join tantargyak t
on j.tkod=t.tkod;
```

--Feladat:

-- Próbálják ki az előző feladatra (de legalább 2 táblára) a LEFT és RIGHT joinokat is, nézzék meg milyen eredményt kapnak az inner join, right join, left join és full join esetében.

--Megoldás:

```
Select *
from hallgatok h right join jegyek j
on h.neptun=j.neptun right join tantargyak t
on j.tkod=t.tkod;
```

```
Select *
from hallgatok h left join jegyek j
on h.neptun=j.neptun left join tantargyak t
on j.tkod=t.tkod;
```

--Feladat:

-- Kérdezzék le minden hallgató legutoljára szerzett érdemjegyét. (Miből, mikor szerezte és hány?)

--Megoldás:

```
Select vnev, knev, vdatum, tnev, jegy
from jegyek j, hallgatok h, tantargyak t
where j.neptun=h.neptun and j.tkod=t.tkod
and
```

```
vdatum=(select max(vdatum) from jegyek j2
where j2.neptun=j.neptun);
```

-- mivel a legutoljára szerzett érdemjegyet, emiatt kizárólag a neptun kód alapján elég megnézni a legutolsó vizsgát, nem kell tárgyanként is.

--Feladat:

-- Hozzanak létre nézetet, mely megjeleníti a hallgatók névsorát és adataikat.

--Megoldás:

Create view Hallgatói_névsor AS -- A create view segítségével hozzuk létre a nézetet, majd adunk neki nevet (jelen esetben a Hallgatói_névsor) majd az As után írjuk magát a lekérdezésünket

```
Select *
from hallgatok;
```


--Ha nézetet hoztunk létre a views mappában találjuk utána, illetve az alábbi módon pedig tudjuk megnézni az eredményt:

```
Select * from Hallgatói_névsor
```

--Feladat:

-- Hozzanak létre nézetet a tantárgylistára is.

--Megoldás:

```
Create view Tantárgylista AS
```

```
Select *
```

```
from tantargyak;
```

--Ha nézetet hoztunk létre a views mappában találjuk utána, illetve az alábbi módon pedig tudjuk megnézni az eredményt:

```
Select * from Tantárgylista
```

--Feladat:

-- Írjon egy T-SQL szkriptet, ami egy változóba tárolja a hallgatók számát, majd a változó értékét megjeleníti eredménytáblaként és üzenetként is.

--Megoldás:

```
declare @i int --változó deklarálása
```

```
set @i=(select count(*) from hallgatok) --művelet
```

```
select @i --megjelenítés
```

```
print @i -- ennek eredménye a message fülben található
```

--Feladat:

-- Írja meg a következő T-SQL szkriptet: Hozzon létre egy DB nevű adatbázist, abban egy Table1 nevű táblát három tinyint típusú oszloppal és töltse fel a táblát a következő rekorddal: 1,2,3.

--Megoldás:

```
create database DB
```

```
go
```

```
use DB
```

```
create table Table1
```

```
( col1 tinyint,
```

```
col2 tinyint,
```

```
col3 tinyint,)
```

```
insert into Table1 values(1,2,3);
```

--Feladat:

-- Írjon egy T-SQL szkriptet, amely kiszámolja, hogy összesen hány rekordja van a DB adatbázis Table1 táblájának és a Hallgatói nyilvántatár adatbázisuk hallgatók táblájának.

--Megoldás:

```
USE DB
```

```
declare @i int
```

```
set @i=(select count(*) from Table1) --először a DB adatbázisban lévő Table1
```

```
rekordjait számoljuk meg
```

```
Use HallgatoiNyilvanT
```

```
declare @j int
```

```
set @j=(select count(*) from hallgatok)--utána a HallgatoiNyilvanT adatbázisban lévő
```

```
Table1 rekordjait számoljuk meg
```

```
Select @i+@j --majd összeadjuk őket
```

--Feladat:

-- Adjunk minden hallgatónak pl. adatbáziskezelés eggyel jobb jegyet, amíg az utoljára szerzett jegyeik tantárgyi átlaga el nem éri a 4-est.

--Megoldás:

```
While (select avg(Cast(jegy as float)) AS 'Átlag'
      from jegyek j
      where tkod='NGB_IN008_1' and
            uv=(Select max(uv) from jegyek j2 where j2.neptun=j.neptun
and tkod='NGB_IN008_1'))<4
BEGIN
    Update jegyek
    set jegy=jegy+1
    from jegyek j
    where tkod='NGB_IN008_1' and jegy<5 and
          uv=(select max(uv) from jegyek j2 where j2.neptun=j.neptun and
tkod='NGB_IN008_1')
END
Print 'Kész'
```

--Feladat:

-- Írják meg az előző feladatot tárolt eljárásként.

--Megoldás:

--Ha tárolt eljárást hozunk létre, akkor azt a CREATE PROC segítségével tesszük.

Create proc Jobbjegy

AS

```
While (select avg(Cast(jegy as float)) AS 'Átlag'
      from jegyek j
      where tkod='NGB_IN008_1' and
            uv=(Select max(uv) from jegyek j2 where j2.neptun=j.neptun
and tkod='NGB_IN008_1'))<4
BEGIN
    Update jegyek
    set jegy=jegy+1
    from jegyek j
    where tkod='NGB_IN008_1' and jegy<5 and
          uv=(select max(uv) from jegyek j2 where j2.neptun=j.neptun and
tkod='NGB_IN008_1')
END
Print 'Kész'
-- a tárolt eljárásunk a következő képpen tudjuk lefuttatni:
EXEC Jobbjegy
```

--Feladat:

-- Készítsenek tárolt eljárást, amely a kapott paraméterekben szereplő értékeket szűri be új rekordként a "Tantárgyak" táblába. Az eljárás természetesen ellenőrizze, hogy a felvinni kívánt tantárgy létezik-e már.

--Megoldás:

--először létrehozzuk a tárolt eljárást

Create procedure Tantargy_Insert

@tkod nvarchar(50), @tnev nvarchar(80), @kredit tinyint, @tkov nvarchar(40),

@visszater int --deklarálom mit is szeretnék, azok az adatok ugyanazzal az adattípussal, mint a táblánkban + a visszatérési érték az ellenőrzés miatt

OUTPUT

AS

IF EXISTS (Select tkod from tantargyak where tkod=@tkod) --megnézzük létezik-e már

SET @visszater=-1

ELSE

BEGIN

```

Insert into tantargyak values (@tkod, @tnev, @kredit, @tkov)
SET @visszater=0
end

-----
--itt pedig a már létrehozottal dolgozunk
Declare @retval int
EXECUTE Tantargy_Insert 'NGB_IN010_2', 'Vállalati információs rendszerek 2', 3, '',
@retval OUTPUT
IF @retval<>0
Print 'Ilyen tantárgykód már létezik!'
Else
Print 'A felvitel rendben megtörtént!'

--Feladat:
--      Készítsen egy táblaértékű függvényt, amely visszaadja a paraméterként kapott
településen lakó hallgatók nevét és neptunkódját.

--Megoldás:
Create function Gyori_Hallgatok(@telepules nvarchar(50))
returns table as
return (select vnev, knev, neptun
          from hallgatok
          where telepules=@telepules)
-----Futtatás:
Select * from Gyori_Hallgatok('Győr')

--Feladat:
--      Készítsen egy olyan táblaértékű függvényt, amely egy paraméterként kapott
dátumot kap és visszaadja egy táblában, hogy az adott hónap összes napján a hallgatók
által szerzett érdemjegyeknek mik voltak az átlagaik.

--Megoldás:
Create function HaviAtlag(@datum date)
returns @eredmeny table (nap datetime, napiatlag decimal(3,2)) as
begin
declare @aktualisdatum date
set @aktualisdatum = DATEADD(DAY, 1-DAY(@datum), @datum)
While @aktualisdatum<= EOMONTH(@datum)
begin
insert into @eredmeny select @aktualisdatum,
AVG(Convert(decimal(3,2),jegy))
from jegyek
where vdatum=@aktualisdatum
set @aktualisdatum=DATEADD(DAY, 1, @aktualisdatum)
END
return
end
-----Futtatás:
Select * from HaviAtlag('2018-06-18')

```