

Az IN használata többoszlopos tábla esetén; Az SQL alapértelmezett működése halmaz-multihalmaz szempontból; Több tábla használata a FROM-ban (keresztszorzat); Többszörösen beágyazott SELECT

Az IN használata többoszlopos tábla esetén

- *Állítsuk elő megyénként a legkisebb lakosságú települések (megye, településnév, lakosság) listáját!
Rendezés: lakosság, megye, településnév szerint!*

Lépésenként:

- Legkisebb lakosság megyénként:

```
SELECT megye,MIN(lakosság) FROM telepules GROUP BY megye;
```

```
SELECT megye,helysegnev,lakosság FROM telepules WHERE (megye,lakosság) IN  
(SELECT megye,MIN(lakosság) FROM telepules GROUP BY megye) ORDER BY 3,1,2;
```

Vagy:

```
SELECT megye,helysegnev,lakosság FROM telepules AS t1  
WHERE lakosság = (SELECT MIN(lakosság)FROM telepules AS t2  
WHERE t1.megye=t2.megye)  
ORDER BY 3,1,2;
```

Megjegyzés: E konkrét példában kb. 1/3 idő alatt állítható elő ugyanaz az eredmény a második megoldással!!!

Az SQL alapértelmezett működése halmaz-multihalmaz szempontból:

Az SQL a relációkat alapvetően multihalmazként tekinti. Ez azt jelenti, hogy ugyanaz a sor többször is megjelenhet egy relációban.

Ha halmazos viselkedést szeretnénk elérni, akkor a **DISTINCT** kifejezést használjuk.

- *Állítsuk elő a megyék neveit!*

```
SELECT megye FROM település;
```

```
SELECT DISTINCT megye FROM telepules ORDER BY 1;
```

Esetleg:

```
SELECT megye FROM telepules GROUP BY megye ORDER BY 1;
```

**Az IN használata többoszlopos tábla esetén; Az SQL alapértelmezett működése halmaz-
multihalmaz szempontból; Több tábla használata a FROM-ban (keresztszorzat);
Többszörösen beágyazott SELECT**

Az SQL a halmazműveletek eredményét halmaznak tekinti:

```
SELECT helysegnev,jaras,megye FROM telepules WHERE helysegnev LIKE '%lászló%'; (9)
```

helysegnev	jaras	megye
Bakonyszentlászló	Pannonhalmi	Győr-Moson-Sopron
Búcsúszentlászló	Pacsi	Zala
Jászsztentlászló	Kiskunmajsai	Bács-Kiskun
Pilisszentlászló	Szentendre	Pest
Pusztaszentlászló	Zalaegerszegi	Zala
Sajólászlófalva	Miskolci	Borsod-Abaúj-Zemplén
Szentlászló	Szigetvári	Baranya
Vácszentlászló	Gödöllői	Pest
Zalaszentlászló	Zalaszentgróti	Zala

```
SELECT helysegnev,jaras,megye FROM telepules WHERE helysegnev LIKE '%szent%'; (148)
```

```
(SELECT helysegnev,jaras,megye FROM telepules WHERE helysegnev LIKE '%lászló%')  
UNION  
(SELECT helysegnev,jaras,megye FROM telepules WHERE helysegnev LIKE '%szent%'); (149)
```

Ha multihalmazt szeretnénk elérni, akkor az **UNION ALL** kifejezést alkalmazzuk:

```
(SELECT helysegnev,jaras,megye FROM telepules WHERE helysegnev LIKE '%lászló%')  
UNION ALL  
(SELECT helysegnev,jaras,megye FROM telepules WHERE helysegnev LIKE '%szent%'); (157)
```

Településnév szerint rendezve:

```
SELECT helysegnev, jaras, megye FROM (  
  (SELECT helysegnev, jaras, megye FROM telepules WHERE helysegnev LIKE '%lászló%')  
  union all  
  (SELECT helysegnev, jaras, megye FROM telepules WHERE helysegnev LIKE '%szent%')) AS x  
ORDER BY helysegnev;
```

**Az IN használata többszlopós tábla esetén; Az SQL alapértelmezett működése halmaz-
multihalmaz szempontból; Több tábla használata a FROM-ban (keresztszorzat);
Többszörösen beágyazott SELECT**

Több tábla használata a FROM-ban = keresztszorzat

HALLGATO (Hkod, Nev, Irszam, Varos, Utca, Telefon, Szdatum)

TANUL (Hkod, Tkod, Jegy)

TANTARGY (Tkod, Tnev)

Hallgato

Hkod	Nev	Irszam	Varos	Utca_hsz	Telefon	Szdatum
1	Török Anna	7620	Pécs	Pipacs u. 12.	(75) 111-222	1982.01.20
2	Németh Ábel	2513	Ercsi	Fa u.123.	(75) 111-223	1980.01.21
3	Nagy Elvira	7400	Paks	Dózsa Gy. 23.	(75) 111-224	1980.11.22
4	Kiss Jutka	8500	Pápa	Sóház u.120.	(75) 111-225	1982.01.22
5	Szabadi Imre	8500	Pápa	Szőlő u. 16.	(75) 111-226	1980.01.20
6	Vendég Anna	2514	Adony	Fő tér 12/B.	(75) 111-227	1980.11.25
7	Haragos Lajos	2580	Tata	Kossuth u. 16.	(75) 111-228	1982.01.25

Tantargy

Tkod	Tnev
10	Angol
20	Algebra
30	Német
40	Matematika
50	Logika
60	Történelem
70	Fizika

Tanul

Hkod	Tkod	Jegy
1	10	4
1	20	1
2	20	5
2	30	1
3	60	2
4	70	5

`SELECT * FROM Hallgato, Tanul, Tantargy;` Descartes-szorzat

Csak azok a tanulók adatai jelenjenek meg, akiknek van jegyük valamilyen tantárgyból:

`SELECT * FROM Hallgato, Tanul, Tantargy
WHERE Hallgato.Hkod = Tanul.Hkod AND Tantargy.Tkod=Tanul.Tkod;`

Az IN használata többszlopos tábla esetén; Az SQL alapértelmezett működése halmaz-multihalmaz szempontból; Több tábla használata a FROM-ban (keresztszorzat); Többszörösen beágyazott SELECT

Akinek van jegye valamilyen tantárgyból, azoknak csak a nevét, a tantárgy nevét és a kapott érdemjegyet írassuk ki az eredménytáblába:

```
SELECT Hallgato.Nev, Tantargy.Tnev, Tanul.Jegy
FROM Hallgato, Tanul, Tantargy
WHERE Hallgato.Hkod=Tanul.Hkod AND Tanul.Tkod=Tantargy.Tkod;
```

Vagy

```
SELECT Nev, Tnev, Jegy FROM Hallgato, Tanul, Tantargy
WHERE Hallgato.Hkod = Tanul.Hkod AND Tantargy.Tkod=Tanul.Tkod;
```

A mezőnevek egyediek (mindegyik csak egy adott táblában szerepel csak mezőnévként), ezért a táblára való hivatkozás elhagyható SELECT és FROM között!

Eredmény:

Nev	Tnev	Jegy
Török Anna	Angol	4
Török Anna	Álgebra	1
Németh Ábel	Álgebra	5
Németh Ábel	Német	1
Nagy Elvira	Történelem	2
Kiss Jutka	Fizika	5

Mezők átnevezése:

```
SELECT Nev AS Név, Tnev AS Tantárgy, Jegy FROM Hallgato, Tanul, Tantargy
WHERE Hallgato.Hkod = Tanul.Hkod AND Tantargy.Tkod=Tanul.Tkod;
```

Ha van „saját” válogatás is, akkor azt AND-del adjuk hozzá:

Pl.: Csak az „A” betűvel kezdődő tantárgyakból szerzett érdemjegyeket írassuk ki hallgatónként:

```
SELECT Nev AS Név, Tnev AS Tantárgy, Jegy FROM Hallgato, Tanul, Tantargy
WHERE Hallgato.Hkod = Tanul.Hkod AND Tantargy.Tkod=Tanul.Tkod
AND Tnev like 'A%';
```

Név	Tantárgy	Jegy
Török Anna	Angol	4
Török Anna	Álgebra	1
Németh Ábel	Álgebra	5

Pl.: Csak az „A” betűvel kezdődő vagy a német tantárgyból szerzett érdemjegyeket írassuk ki hallgatónként:

```
SELECT Nev AS Név, Tnev AS Tantárgy, Jegy FROM Hallgato, Tanul, Tantargy
WHERE Hallgato.Hkod = Tanul.Hkod AND Tantargy.Tkod=Tanul.Tkod
AND (Tnev like 'A%' OR Tnev='Német');
```

Zárójelzés!

Az IN használata többszlopos tábla esetén; Az SQL alapértelmezett működése halmaz-multihalmaz szempontból; Több tábla használata a FROM-ban (keresztszorzat); Többszörösen beágyazott SELECT

Név	Tantárgy	Jegy
Török Anna	Angol	4
Török Anna	Algebra	1
Németh Ábel	Algebra	5
Németh Ábel	Német	1

Ugyanez JOIN-nal: gyorsabb a működésben és még kevesebbet is kell írni!!!

```
SELECT Nev AS Név, Tnev AS Tantárgy, Jegy  
FROM Hallgato NATURAL JOIN Tanul NATURAL JOIN Tantargy;
```

```
SELECT Nev AS Név, Tnev AS Tantárgy, Jegy AS Jegy  
FROM Hallgato NATURAL JOIN Tanul NATURAL JOIN Tantargy  
WHERE Tnev LIKE 'A%' OR Tnev='Német';
```

Mindig figyelni kell arra, hogy a NATURAL JOIN feltételei teljesüljenek:

- A táblákban az (összes!) azonos nevű oszlopok tartalma logikailag és formailag is megfeleljen egymásnak!
- Ha az azonos nevű oszlopok adattípusa eltérő, akkor hibával tér vissza az utasítás.

Gyakorló feladatok:

1. Írassuk ki az egyes hallgatók jegyeinek átlagát!
2. Írassuk ki „Török Anna” jegyeit!
3. Kinek van jeles érdemjegye?
4. Kinek van jeles érdemjegye, és hány darab?
5. Mely tantárgyaktól van jeles érdemjegy?
6. Hány darab jeles érdemjegy van tantárgyanként?

Az IN használata többszörös tábla esetén; Az SQL alapértelmezett működése halmaz-multihalmaz szempontból; Több tábla használata a FROM-ban (keresztszorzat); Többszörösen beágyazott SELECT

Gyakorló feladatok:

Fel.: Állítsuk elő az azonos lakosságú települések darabszámát!

- Lakosság szerint növekvő sorrendben:

lakosság	db
13	3
26	3
27	2
33	2
40	3
44	2
45	3
48	3
49	3
50	2
51	2
52	2
54	2

```
SELECT lakosság, COUNT(*) AS db FROM telepules  
GROUP BY lakosság HAVING count(*)>1;
```

- Darabszám szerint csökkenő sorrendben:

lakosság	count(*)
312	7
324	7
270	7
198	7
184	7
127	6
200	6
374	6
204	6

```
SELECT lakosság, COUNT(*) AS db FROM telepules  
GROUP BY lakosság HAVING count(*)>1 ORDER BY 2 DESC;
```

Fel.: Állítsuk elő a legnagyobb számú települések darabszámát!

max(db)
7

```
SELECT MAX(db) FROM(  
SELECT lakosság, COUNT(*) AS db FROM telepules GROUP BY lakosság ORDER BY 2 DESC) AS x;
```

Az IN használata többszörös tábla esetén; Az SQL alapértelmezett működése halmaz-multihalmaz szempontból; Több tábla használata a FROM-ban (keresztszorzat); Többszörösen beágyazott SELECT

Többszörösen beágyazott SELECT

- *Állítsuk elő a legnagyobb számú, azonos lakosságú települések listáját!*

Lépésenként:

- Azonos lakosságú települések darabszáma:

```
SELECT lakosság, COUNT(*) AS db FROM telepules GROUP BY lakosság;
```

- Legnagyobb darabszám:

```
SELECT MAX(db) FROM(
  SELECT lakosság, COUNT(*) AS db FROM telepules GROUP BY lakosság) AS x;
```

MAX(db)

7

- Lakosság száma, ahol a darabszám 7:

```
SELECT lakosság FROM(
  SELECT lakosság, COUNT(*) AS db FROM telepules GROUP BY lakosság) AS x1 WHERE db=(
  SELECT MAX(db) FROM(
    SELECT lakosság, COUNT(*) AS db FROM telepules GROUP BY lakosság) AS x);
```

lakosság
184
198
270
312
324

- Legnagyobb számú, azonos lakosságú települések listája:

```
SELECT helysegnev, lakosság FROM telepules WHERE lakosság IN(
  SELECT lakosság FROM(
    SELECT lakosság, COUNT(*) AS db FROM telepules GROUP BY lakosság) AS x1 WHERE db=(
    SELECT MAX(db) FROM(
      SELECT lakosság, COUNT(*) AS db FROM telepules GROUP BY lakosság) AS x)) ORDER BY 2;
```

**Az IN használata többszlopos tábla esetén; Az SQL alapértelmezett működése halmaz-
multihalmaz szempontból; Több tábla használata a FROM-ban (kereszt szorzat);
Többszörösen beágyazott SELECT**

helysegnev	lakosság
Abaújszolnok	184
Pusztáederics	184
Monyoród	184
Kisgyalán	184
Kéked	184
Karakó	184
Bakháza	184
Szilvásszentmárton	198
Somogyegres	198
Terpes	198
Tárnokréti	198
Baranyahídvég	198
Kercaszomor	198
Lickóvadas	198
Magyarszentmiklós	270
Mánd	270

(Összesen 35 település)