

Kapcsolatok – JOIN

Amikor több adattáblát kell használnunk, akkor ki kell alakítani a köztük lévő kapcsolatot. Tekintsük megint az iskolai adatbázisunkat.

osztaly	tanulo
ABC OSZTALY	ABC NEV
ABC OSZT_FONOK	ABC OSZTALY
ABC OSZT_TEREM	ABC SZUL_IDO
	123 FERFI
	123 TESTVER
	ABC LAKCIM

Ekkor ha lefuttatjuk a következő 3 parancsot,

```
1. SELECT COUNT(1) FROM TANULO;
2. SELECT COUNT(1) FROM OSZTALY;
3. SELECT COUNT(1) FROM TANULO, OSZTALY;
```

akkor rendre 496, 20, illetve 9920 kapunk. Érezhető, hogy valami nincs rendben az utolsó lekérdezésnél. Hiányzik a kapcsolat kialakítása, azaz a lekérdezés eredménye a Descartes-szorzatot adja. Ebből nekünk kell kiszűrni azon sorokat, amelyek összetartoznak.

```
1. SELECT COUNT(1)
2. FROM TANULO, OSZTALY
3. WHERE TANULO.OSZTALY=OSZTALY.OSZTALY;
```

Ennek eredményeképpen már 496-t kapunk, ami reális érték.

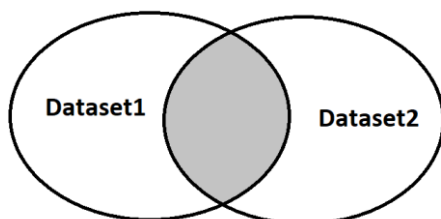
Általánosan

```
1. SELECT oszlop(ok)
2. FROM tablaNev1 T1, tablaNev2 T2
3. WHERE feltetel;
```

A kapcsolatot másféleképpen, és más módokon is kialakíthatjuk. Erre szolgál a JOIN: a JOIN záradékot két vagy több táblából álló sorok kombinálására használjuk az egymáshoz kapcsolódó oszlop alapján. Tekintsük végig a JOIN utasítások fajtáit.

INNERT JOIN

Ebben az esetben azon a sorok jelennek meg, ahol mindkét táblázatban egymásnak megfelelő oszlopbeli értékeket tartalmaznak.



Általános alak

```
1. SELECT oszlop(ok)
2. FROM tablaNev1
3. INNER JOIN tablaNev2
4. ON tablaNev1.oszlop = tablaNev2.oszlop;
```

Példa

```
1. SELECT *
2. FROM TANULO
3. INNER JOIN OSZTALY ON TANULO.OSZTALY=OSZTALY.OSZTALY;
```

NATURAL JOIN

Ez a kapcsolatot akkor használhatjuk, amikor a kapcsolatban résztvevő két oszlop neve megegyezik. Ekkor a lekérdezés eredményében csak egyszer jelenik meg ez az adathalmaz.

Példa

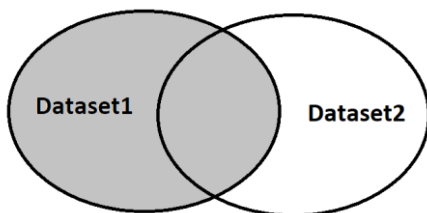
```
1. SELECT *
2. FROM TANULO NATURAL JOIN OSZTALY;
```

Eredmény

Grid	OSZTALY	NEV	SZUL_IDO	FERFI	TESTVER	LAKCIM	OSZT_FONOK	OSZT_TEREM
1	1/A	Szukits Adrián	1980-09-26	1	0	Ostromos, Csend u. 72.	Nagy Elemér	103
2	1/A	Gázser Jácint	1981-05-14	1	3	Ostromos, Drégely u. 33.	Nagy Elemér	103
3	1/A	Zsandár Gáspár	1980-12-05	1	3	Nyúlvaros, Hermina u. 68.	Nagy Elemér	103
4	1/A	Baummann Mirjam	1981-01-31	0	1	Végvár, Akó u. 19.	Nagy Elemér	103
5	1/A	Benkó Csaba	1981-08-24	1	2	Csókostó, Akadémia u. 45.	Nagy Elemér	103
6	1/A	Tanács Ivett	1980-08-06	0	0	Tiveletelek, Bunda u. 104.	Nagy Elemér	103
7	1/A	Kerekes Mária	1981-01-08	0	0	Szegényfő, Korom u. 28.	Nagy Elemér	103
8	1/A	Serfőző Csilla	1980-11-13	0	3	Szegényfő, Avar u. 99.	Nagy Elemér	103
9	1/A	Jász Szixtus	1980-10-25	1	1	Szegényfő, Emese u. 35.	Nagy Elemér	103
10	1/A	Sztanyik Rózsa	1980-11-07	0	0	Csónakkút, Határ u. 1.	Nagy Elemér	103

LEFT JOIN

Ebben az esetben a JOIN kapcsolat bal oldalán lévő adathalmaz minden értéke megjelenik, függetlenül, hogy van-e hozzá kapcsoló érték a másik adathalmazban. A „hiányzó” adatok helyén ebben az esetben NULL értékek jelennek meg.



Általános alak

```
1. SELECT oszlop(ok)
2. FROM tablaNev1
3. LEFT JOIN tablaNev2
4. ON tablaNev1.oszlop = tablaNev2.oszlop;
```

Példa

1. **INSERT INTO** tanulo **VALUES**
2. ("Szép Anna", null, "2010-02-23", false, 3, "Kerekerdő, Közép utca 12");

Új tanuló adatainak beszúrása osztály adatai nélkül (pl magántanuló).

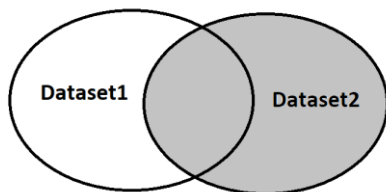
3. **SELECT** *
4. **FROM** TANULO
5. **LEFT JOIN** OSZTALY **ON** tanulo.OSZTALY=osztaly.OSZTALY;

Eredmény

Grid	NEV	OSZTALY	SZUL_IDO	FELE	TESTVER	LAKCIM	OSZTALY	OSZT_FONOK	OSZT_TE
417	Szukits Albert	3/A	1979-03-26	1	3	Végyvár, Góz u. 87.	3/A	Horváth Károly	308
418	Szunyogh Helén	2/D	1980-06-05	0	0	Nyúl város, Aradi u. 96.	2/D	Tomka Béla	117
419	Szász Kelemen	2/D	1980-05-19	1	0	Ostromos, Juhos u. 104.	2/D	Tomka Béla	117
420	Szekely Mirjam	3/B	1979-01-15	0	1	Radonvár, Föld u. 70.	3/B	Fergeteg Zsuzsanna	109
421	Szép Anna	[NULL]	2010-02-23	0	3	Kerekerdő, Közép utca 12	[NULL]	[NULL]	[NULL]
422	Szűcs Béla	3/A	1978-10-18	1	2	Nyúl város, Baka u. 92.	3/A	Horváth Károly	308

RIGHT JOIN

Hasonló a LEFT JOIN-hoz csak a JOIN jobb oldalán álló adattáblából jelenik meg az összes adat.



Általános alak

1. **SELECT** column_name(s)
2. **FROM** table1
3. **RIGHT JOIN** table2
4. **ON** table1.column_name = table2.column_name;

Példa

1. **INSERT INTO** osztaly **VALUES** ("5M", "Szigor János", 123);

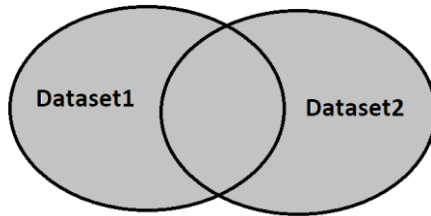
Egy újonnan induló osztály adatai

1. **SELECT** *
2. **FROM** TANULO
3. **RIGHT JOIN** OSZTALY **ON** tanulo.OSZTALY=osztaly.OSZTALY;

491	4/C	1978-01-31	0	2	Gazdaghely, Bontó u. 67.	4/C	Kovács Gábor	312
492	4/C	1977-12-20	1	3	Szegényfő, Juhos u. 81.	4/C	Kovács Gábor	312
493	4/C	1977-10-25	0	3	Tiveletelek, Gubó u. 62.	4/C	Kovács Gábor	312
494	4/C	1977-10-30	1	2	Végyvár, Kőbánya u. 15.	4/C	Kovács Gábor	312
495	4/C	1978-08-26	0	2	Radonvár, Akácfa u. 13.	4/C	Kovács Gábor	312
496	4/C	1978-06-30	1	3	Ostromos, Álom u. 74.	4/C	Kovács Gábor	312
497	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]	5M	Szigor János	123

FULL OUTER JOIN

Ebben az esetben mindkét táblából minden információ megjelenik.



Általánosan

```

1. SELECT column_name(s)
2. FROM table1
3. FULL OUTER JOIN table2
4. ON table1.column_name = table2.column_name
5. WHERE condition;

```

A MySQL ezt az SQL utasítást nem támogatja. Ha ezt szeretnénk elérni, akkor a baloldali, illetve a jobboldali kapcsolat unióját kell vennünk.

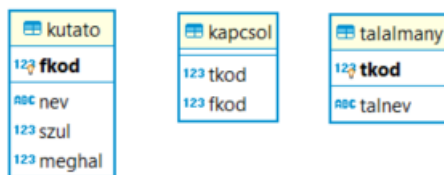
```

1. SELECT *
2. FROM TANULO
3. LEFT JOIN OSZTALY on tanulo.OSZTALY=osztaly.OSZTALY
4. UNION
5. SELECT *
6. FROM TANULO
7. RIGHT JOIN OSZTALY on tanulo.OSZTALY=osztaly.OSZTALY;

```

Példa

Vegyük a következő adatbázist, ahol a feltalálók és találmányaikat találhatjuk. Mivel ez egy több-több kapcsolat ezért szükséges egy kapcsoló tábla.



Írassuk ki azon kutatók nevét, akiknek nem szerepel találmánya az adatbázisban.

Megoldás

```

1. SELECT
2.   nev
3. FROM
4.   kutato
5. LEFT JOIN kapcsol ON
6.   kutato.fkod = kapcsol.fkod
7. WHERE
8.   tkod IS NULL

```

Eredmény

Grid	abc nev	
1	Hanaman Ferenc	

Írasd ki hány olyan találmány van az adatbázisban, akiknek nem szerepel feltalálója!

```
1. SELECT
2.     COUNT(talnev)
3. FROM
4.     kapcsol
5. RIGHT JOIN talalmany ON
6.     talalmany.tkod = kapcsol.tkod
7. WHERE
8.     fkod IS NULL;
```

Grid	123 COUNT(talnev) 7
1	7