Spajanje tabela

(JOIN familija operatora)

Možete pogledati video prezentaciju sa kojom su to lepo objasnili i to na našem jeziku na sledećem linku:

- VIDEO PREDAVANJE SA YOUTUBE-a

  (klikom na link treba da se otvori drugi tab sa video materijalom)

  - Na sledećim slajdovima je prezentacija sa prethodnog časa kao podsetnik onoga što smo radili

Dekartov (Kartezijanov) proizvod

Student

BrInd	Ime	Prezime
100/2010	Marko	Marković
200/2011	Petar	Petrović

Predmet

Sifra	Naziv
SPRI01	Informatika
SPRI02	Matematika
SPRI03	Engleski jezik 1

BrInd	Ime	Prezime	Sifra	Naziv
100/2010	Marko	Marković	SPRI01	Informatika
100/2010	Marko	Marković	SPRI02	Matematika
100/2010	Marko	Marković	SPRI03	Engleski jezik 1
200/2011	Petar	Petrović	SPRI01	Informatika
200/2011	Petar	Petrović	SPRI02	Matematika
200/2011	Petar	Petrović	SPRI03	Engleski jezik 1

```
SELECT *
        FROM Student, Predmet;
```

Ako se u SELECT naredbi navedu dve tabele obavezno se dešava Dekartov proizvod

```
SELECT * FROM Student, Predmet;
```

Drugačija sintaksa:

```
SELECT *
FROM Student CROSS JOIN Predmet;
```

Dekartov proizvod nad 3 relacije (r1, r2, r3):

```
SELECT *
FROM r1 CROSS JOIN r2 CROSS JOIN r3;
```

spajanje

Linija

Let	Udaljenost
AA251	1000
UX112	4000
CC222	11000

Avion

Tip	Dolet
B747	15000
A320	7000
DC9	3000

```
SELECT *
    FROM Linija, Avion
    WHERE Dolet ≥Udaljenost;
```

Let	Udaljenost	Tip	Dolet
AA251	1000	B747	15000
AA251	1000	A320	7000
AA251	1000	DC9	3000
UX112	4000	B747	15000
UX112	4000	A320	7000
CC222	11000	B747	15000

Drugačija sintaksa:

```
SELECT *
    FROM Linija JOIN Avion
        ON Dolet ≥Udaljenost;
```

spajanje nad 3 relacije (r1, r2, r3):

```
SELECT *
    FROM r1
        JOIN r2 ON JoinCondition
        JOIN r3 ON Join Condition;
```

Kako pronaći linije i avione koji na tim linijama mogu leteti, ali samo za one linije na kojima je udaljenost veća od 3000km

```
SELECT *
    FROM Linija, Avion
    WHERE Dolet ≥Udaljenost
    AND Udaljenost ≥ 3000;
```

ILI

```
SELECT *
    FROM Linija JOIN Avion ON Dolet ≥Udaljenost
    WHERE Udaljenost ≥ 3000;
```

Levo spoljašnje spajanje

- Svi zapisi relacije Student će se pojaviti u rezultatu

Student

BrInd	Ime
100/2009	Marko
101/2009	Petar
105/2008	Nikola

UpisaniPredmet

BrIndSt	Predmet
100/2009	Baze podataka
101/2009	Baze podataka
100/2009	Internet tehnologije

```
SELECT Student.*, UpisaniPredmet.*,
    FROM Student LEFT OUTER JOIN UpisaniPredmet
        ON BrInd = BrIndSt;
```

BrInd	Ime	BrIndSt	Predmet
100/2009	Marko	100/2009	Baze podataka
100/2009	Marko	100/2009	Internet tehnologije
101/2009	Petar	101/2009	Baze podataka
105/2008	Nikola	NULL	NULL

Desno spoljašnje spajanje

- Svi zapisi relacije ProfesorPredmet će se pojaviti u rezultatu

StudentPredmet

BrInd	Ime	Predmet
100/2009	Marko	Baze podatka
101/2009	Petar	Programiranje

ProfesorPredmet

IdProf	PredmetProf
MV	Baze podataka
DZ	Programiranje
AJ	Baze podataka
AJ	Internet tehnologije

```
SELECT StudentPredmet.*, ProfesorPredmet.*,
  FROM StudentPredmet RIGHT OUTER JOIN ProfesorPredmet
                ON Predmet = PredmetProf;
```

BrInd	Ime	Predmet	IdProf	PredmetProf
100/2009	Marko	Baze podataka	MV	Baze podataka
100/2009	Marko	Baze podataka	AJ	Baze podataka
101/2009	Petar	Programiranje	DZ	Programiranje
NULL	NULL	NULL	AJ	InternetTehnologije

Puno spoljašnje spajanje

- Svi zapisi iz obe relacije će se pojaviti u rezultatu

StudentPredmet

BrInd	Ime	Predmet
100/2009	Marko	Baze podatka
101/2009	Petar	Programiranje
105/2008	Nikola	Zaštita IS

ProfesorPredmet

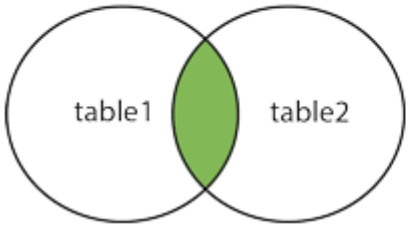
IdProf	PredmetProf
MV	Baze podataka
DZ	Programiranje
AJ	Baze podataka
AJ	Internet tehnologije

Ili SELECT *

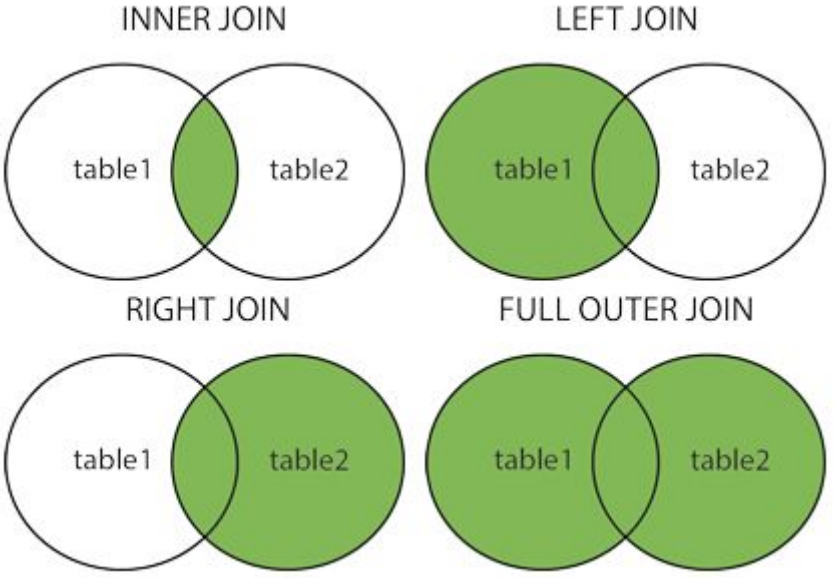
```
SELECT StudentPredmet.*, ProfesorPredmet.*,
FROM StudentPredmet FULL OUTER JOIN ProfesorPredmet
        ON Predmet = PredmetProf;
```

BrInd	Ime	Predmet	IdProf	PredmetProf
100/2009	Marko	Baze podataka	MV	Baze podataka
100/2009	Marko	Baze podataka	AJ	Baze podataka
101/2009	Petar	Programiranje	DZ	Programiranje
NULL	NULL	NULL	AJ	InternetTehnologije
105/2008	Nikola	Zaštita IS	NULL	NULL

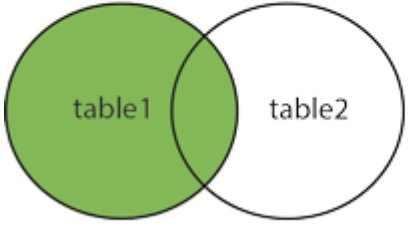
INNER JOIN selects records that have matching values in both tables.




```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```



LEFT JOIN returns all records from the left table (table1), and the matched records from the right table (table2). The result is NULL from the right side, if there is no match.



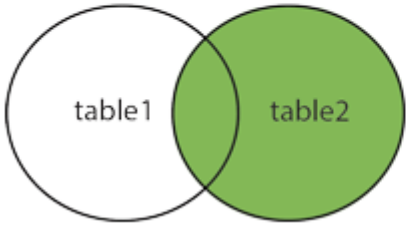

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

Note: In some databases LEFT JOIN is called LEFT OUTER JOIN.

RIGHT JOIN

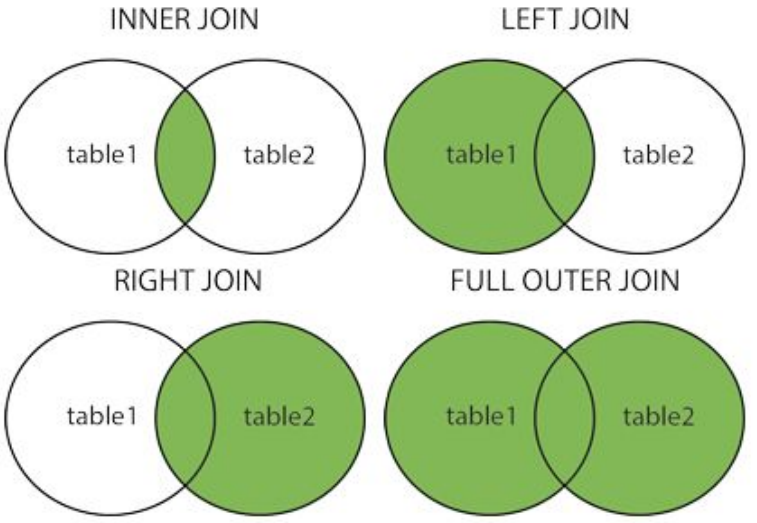
returns all records from the right table (table2), and the matched records from the left table (table1). The result is NULL from the left side, when there is no match.

RIGHT JOIN




```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

Note: In some databases RIGHT JOIN is called RIGHT OUTER JOIN.

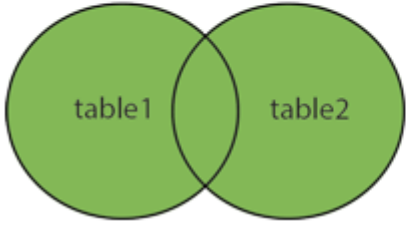


FULL OUTER JOIN

returns all records when there is a match in left (table1) or right (table2) table records.

Note: FULL OUTER JOIN can potentially return very large result-sets!

Tip: FULL OUTER JOIN and FULL JOIN are the same.




```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;
```