

Různá spojení v SQL

Spojení (**JOIN**) slouží ke sloučení dvou a více relací na (**ON**) nějakém atributu v relační databázi.

Existuje více způsobů, jak tabulky spojit:

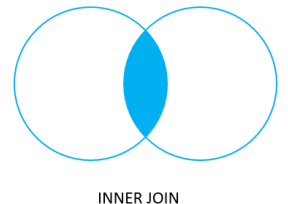
1. INNER JOIN (nebo také jen JOIN)
2. LEFT, RIGHT, FULL (OUTER) JOIN
3. CROSS JOIN, SELF JOIN

Dotazy mají syntaxi: **> SELECT * FROM A typ_joinu B ON A.id = B.id;**

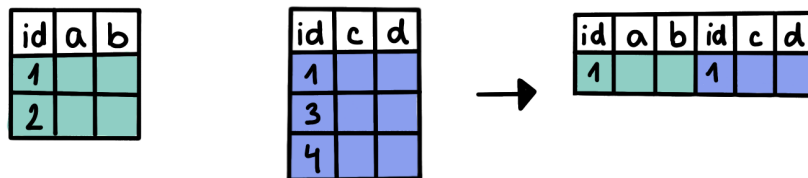
Inner join

INNER JOIN prochází každý řádek první tabulky a porovnává ho s řádky tabulky druhé.

- Pokud jsou hodnoty spojovací podmínky stejné, vytvoří se nový řádek, který obsahuje n-tici vytvořenou z obou vstupních tabulek a přidá ji do výsledné relace.
- Jinak řádek ignoruje.



> SELECT * FROM table1 JOIN table2 ON table1.id = table2.id;



> SELECT *

FROM zakaznik INNER JOIN objednavka

ON zakaznik.id_zakaznika =objednavka.id_zakaznika

id_zakaznika	jméno
1	Martin
2	Milan
3	Eva
4	Ladislav
5	Monika

id_objednavky	id_zakaznika	zakoupeny_predmet	cena
1003	1	reproduktor	2000
1004	3	notebook	28 000
999	10	monitor	9020
1075	24	klávesnice	1700
1005	5	sluchátka	750

id_zakaznika	jméno	id_objednavky	id_zakaznika	zakoupeny_predmet	cena
1	Martin	1003	1	reproduktor	2000
3	Eva	1004	3	notebook	28000
5	Monika	1005	5	sluchátka	750

Poznámka: JOIN vs. NATURAL JOIN

- NATURAL JOIN 'přirozeně' relace spojí podle stejně pojmenovaného parametru. (není potřeba specifikovat, na kterém atributu dojde ke spojení (**SELECT * FROM A NATURAL JOIN B;**)
- rozdílný počet sloupců ve výsledné relaci:

-- NATURAL JOIN vs INNER JOIN - rozdíl: počet sloupců ve výsledné relaci

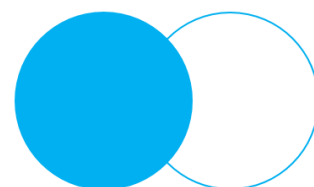
> **SELECT * FROM** zakaznik **NATURAL JOIN** objednavka;

> **SELECT * FROM** zakaznik **INNER JOIN** objednavka **ON** zakaznik.id_zakaznika =objednavka.id_zakaznika

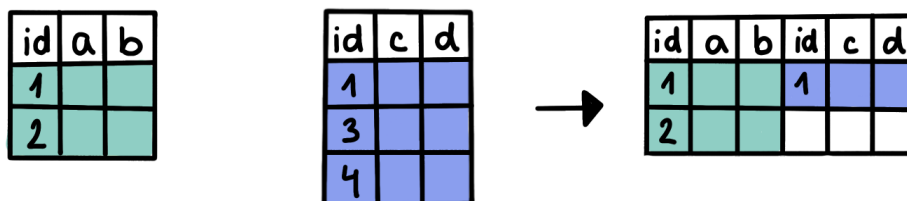
Left join

LEFT JOIN prochází řádky první (levé) tabulky, porovnává hodnoty spojovací podmínky v druhé (pravé) tabulce.

- Pokud narazí na řádek druhé tabulky, který splňuje spojovací podmínku, vytvoří se nový řádek, který obsahuje n-tici vytvořenou z obou vstupních tabulek a přidá ji do výsledné relace.
- Pokud na takový řádek nenarazí, LEFT JOIN **také** vytvoří nový řádek obsahující n-tici vytvořenou z obou vstupních tabulek do výsledné tabulky, **ale** hodnoty sloupců druhé (pravé) tabulky jsou vyplněny hodnotou NULL



> **SELECT * FROM** table1 **LEFT JOIN** table2 **ON** table1.id = table2.id;



> **SELECT ***

FROM zakaznik **LEFT JOIN** objednavka

ON zakaznik.id_zakaznika = objednavka.id_zakaznika;

id_zakaznika	jmeno	id_objednavky	id_zakaznika	zakoupeny_predmet	cena
1	Martin	1003	1	reproduktor	2000
3	Eva	1004	3	notebook	28000
5	Monika	1005	5	sluchátka	750
2	Milan	null	null	null	null
4	Ladislav	null	null	null	null

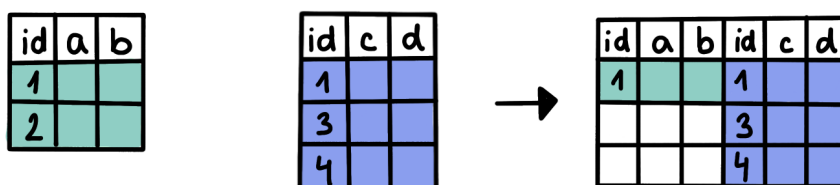
Right join

RIGHT JOIN zobrazí všechny n-tice vyskytující se v druhé tabulce (pravé) a k nim připojí n-tice z první tabulky, na základě spojovací podmínky. Pokud hodnoty neexistují, jsou doplněny NULL hodnotami.



RIGHT OUTER JOIN

> **SELECT * FROM** table1 **RIGHT JOIN** table2 **ON** table1.id = table2.id;



> **SELECT ***

FROM zakaznik **RIGHT JOIN** objednavka

ON zakaznik.id_zakaznika = objednavka.id_zakaznika;

id_zakaznika	jmeno	id_objednavky	id_zakaznika	zakoupeny_predmet	cena
1	Martin	1003	1	reproduktor	2000
3	Eva	1004	3	notebook	28000
null	null	999	10	monitor	9020
null	null	1075	24	klávesnice	1700
5	Monika	1005	5	sluchátka	750

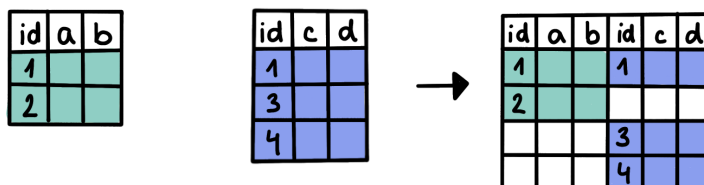
Full join

Operace, která vrátí všechny n-tice z obou tabulek. Ty, které splňují spojovací podmínku se spojí. Ostatní jsou doplněné hodnotami NULL.



FULL OUTER JOIN

> **SELECT * FROM** table1 **FULL JOIN** table2 **ON** table1.id =table2.id;



> **SELECT ***

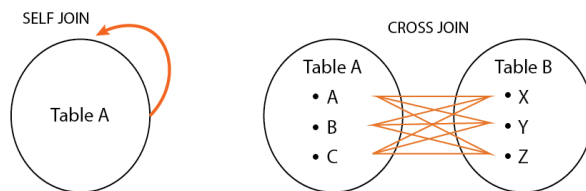
FROM zakaznik **FULL JOIN** objednavka

ON zakaznik.id_zakaznika = objednavka.id_zakaznika;

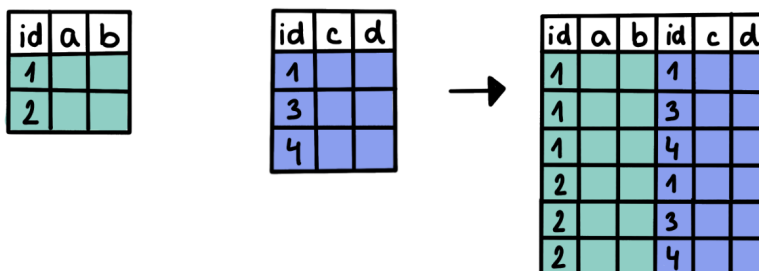
id_zakaznika	jmeno	id_objednavky	id_zakaznika	zakoupeny_predmet	cena
1	Martin	1003	1	reproduktor	2000
3	Eva	1004	3	notebook	28000
null	null	999	10	monitor	9020
null	null	1075	24	klávesnice	1700
5	Monika	1005	5	sluchátka	750
2	Milan	null	null	null	null
4	Ladislav	null	null	null	null

Cross join

CROSS JOIN umožňuje vytvořit kartézský součin řádků dvou tabulek. Výsledná tabulka bude mít $n*m$ řádků, kde m a n je počet řádků spojených tabulek.



> **SELECT * FROM table1 CROSS JOIN table2;**



> **SELECT * FROM T1 CROSS JOIN T2;**

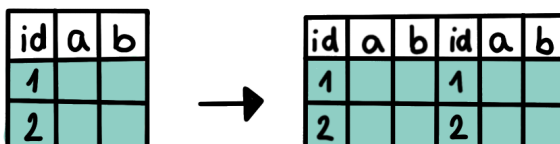
Výraz FROM T1 CROSS JOIN T2 lze zapsat jinak například jako FROM T1 INNER JOIN T2 ON TRUE, nebo FROM T1, T2.

Přirozené spojení nad tabulkami, které nemají společná jména sloupců je také ekvivalentní s CROSS JOIN.

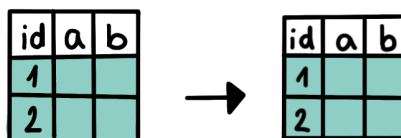
Self join

SELF JOIN nemá speciální syntax. Jedná se o spojení tabulky se sebou.

např.: > **SELECT * FROM table1 t1 INNER JOIN table1 t2 ON t1.id = t2.id;**



> **SELECT * FROM table1 t1 NATURAL INNER JOIN table2 t2;**



Zdroje

1. Dokumentace SQL - joins: <https://docs.oracle.com/database/121/SQLRF/queries006.htm#SQLRF30046>
2. Microsoft documentation - joins: <https://docs.microsoft.com/en-us/sql/relational-databases/performance/joins?view=sql-server-ver15>
3. Klauzule WHERE vs. ON: <https://www.geeksforgeeks.org/sql-on-clause/>
4. Obecné informace: <https://zooom.cz/sql-dotazy-join-vysvetleni-a-graficke-znazorneni>
5. Joiny v PostgreSQL: <https://www.postgresqltutorial.com/postgresql-joins/>
6. Vizualizace joinů: <https://joins.spathon.com/>