Príklad SQL dotazu

```
SELECT
     concat(e.firstname,’ ‘,e.lastname) AS ename,

     (CASE
          WHEN e.comm IS NULL THEN e.sal
          ELSE e.comm + e.sal
     ) AS ‘total_salary’
FROM emp
WHERE deptno>=20 AND lower(e.firstname)=‘john’
LIMIT 1 OFFSET 7
```

Multimnožiny

- Jazyk SQL uvažuje relácie ako multimnožiny, čiže môžu obsahovať duplikáty riadkov (na rozdiel od Datalogu).
- Ak duplikáty nechceme, musíme si to dodatočne vynútiť (obmedzením **UNIQUE** pri vytváraní tabuliek --- viac o tom neskôr pri DDL --- a pomocou **DISTINCT** v dotazoch).

JOINs

- *join* je *s*pojenie dvoch tabuliek; je to podmnožina karteziánskeho súčinu tabuliek špecifikovaná dodatočnými podmienkami na prepájanie (karteziánsky súčin – každý riadok s každým)
- Karteziánsky súčin
- INNER JOIN alebo iba JOIN
- LEFT OUTER JOIN
- RIGHT OUTER JOIN
- FULL OUTER JOIN

Karteziánsky súčin

Name	Deptno
John	10
Thomas	20
Joe	40

X

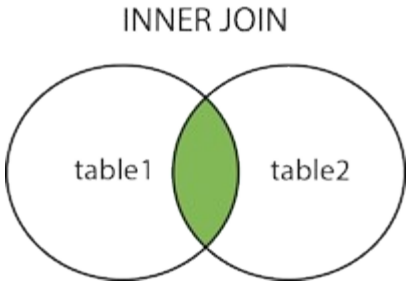
Deptno	Dept. name
10	Accounting
20	PR
30	Development

=

Name	Deptno	Deptno	Dept. name
John	10	10	Accounting
John	10	20	PR
John	10	30	Development
Thomas	20	10	Accounting
Thomas	20	20	PR
Thomas	20	30	Development
Joe	40	10	Accounting
Joe	40	20	PR
Joe	40	30	Development

SELECT * FROM emp, dept

INNER JOIN = JOIN:



Name	Deptno
John	10
Thomas	20
Joe	40

JOIN

Deptno	Dept. name
10	Accounting
20	PR
30	Development

=

Name	Deptno	Deptno	Dept. name
John	10	10	Accounting
~~John~~	~~10~~	~~20~~	~~PR~~
~~John~~	~~10~~	~~30~~	~~Development~~
~~Thomas~~	~~20~~	~~10~~	~~Accounting~~
Thomas	20	20	PR
~~Thomas~~	~~20~~	~~30~~	~~Development~~
~~Joe~~	~~40~~	~~10~~	~~Accounting~~
~~Joe~~	~~40~~	~~20~~	~~PR~~
~~Joe~~	~~40~~	~~30~~	~~Development~~

```
SELECT * FROM emp e, dept d
WHERE e.deptno = d.deptno
```

```
SELECT * FROM emp e
        JOIN dept d ON e.deptno = d.deptno
```

```
SELECT * FROM emp e NATURAL JOIN dept d
```

INNER JOIN = JOIN:

Name	Deptno
John	10
Thomas	20
Joe	40

JOIN

Deptno	Dept. name
10	Accounting
20	PR
30	Development
10	Human res.

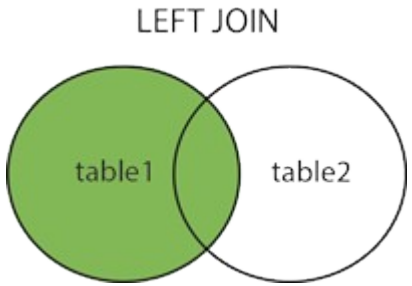
=

Name	Deptno	Deptno	Dept. name
John	10	10	Accounting
~~John~~	~~10~~	~~20~~	~~PR~~
~~John~~	~~10~~	~~30~~	~~Development~~
John	10	10	Human res.
~~Thomas~~	~~20~~	~~10~~	~~Accounting~~
Thomas	20	20	PR
~~Thomas~~	~~20~~	~~30~~	~~Development~~
~~Thomas~~	~~20~~	~~10~~	~~Human res.~~
~~Joe~~	~~40~~	~~10~~	~~Accounting~~
~~Joe~~	~~40~~	~~20~~	~~PR~~
~~Joe~~	~~40~~	~~30~~	~~Development~~
~~Joe~~	~~40~~	~~10~~	~~Human res.~~

```
SELECT *
FROM emp e
        JOIN dept d ON e.deptno = d.deptno
```

Ako by ste zapísali JOIN v relačnom kalkule či datalogu?

LEFT [OUTER] JOIN:



Name	Deptno
John	10
Thomas	20
Joe	40

LEFT JOIN

Deptno	Dept. name
10	Accounting
20	PR
30	Development
10	Human res.

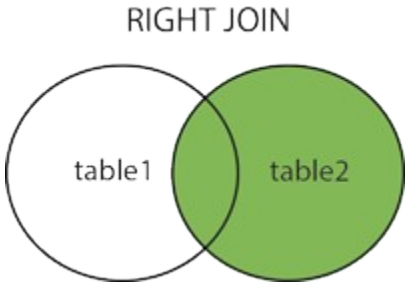
=

Name	Deptno	Deptno	Dept. name
John	10	10	Accounting
John	10	10	Human res.
Thomas	20	20	PR
Joe	40	null	null

```
SELECT *
FROM emp as e
        LEFT JOIN dept as d
                ON e.deptno = d.deptno
```

Ako by ste zapísali LEFT JOIN v relačnom kalkule či datalogu?

RIGHT [OUTER] JOIN:



Deptno	Dept. name
10	Accounting
20	PR
30	Development
10	Human res.

RIGHT JOIN

Name	Deptno
John	10
Thomas	20
Joe	40

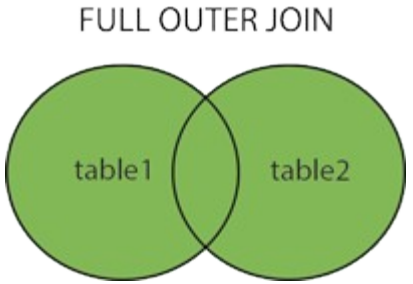
=

Name	Deptno	Deptno	Dept. name
John	10	10	Accounting
John	10	10	Human res.
Thomas	20	20	PR
Joe	40	null	null

To isté ako LEFT JOIN, akurát v obrátenom poradí

```
SELECT *
FROM dept AS d
        RIGHT JOIN emp AS e ON e.deptno = d.deptno
```

Kartézsky súčin (FULL JOIN):



Name	Deptno
John	10
Joe	40

X

Deptno	Dept. name
10	Accounting
30	Development

=

Name	Deptno	Deptno	Dept. name
John	10	10	Accounting
Joe	40	null	null
null	null	30	Development

```
SELECT *
FROM emp e FULL JOIN dept d ON e.deptno = d.deptno
```