

# SQL SELECT – INNER JOIN

Найбільш важливим і часто використовуваним з'єднанням є INNER JOIN. Воно також називається EQUIJOIN.

INNER JOIN створює нову таблицю результатів, об'єднуючи значення стовпчиків з двох таблиць на основі предиката приєднання. Запит порівнює кожен рядок першої таблиці з кожним рядком другої таблиці, щоб знайти всі пари рядків, які задовольняють предикату приєднання. Коли предикат приєднання виконується, значення стовпців для кожної зіставленої пари рядків об'єднуються в рядок результату.

# SQL SELECT – INNER JOIN

The most important and frequently used of the joins is the INNER JOIN. They are also referred to as an EQUIJOIN.

The INNER JOIN creates a new result table by combining column values of two tables based upon the join-predicate. The query compares each row of the first table with each row of the second table to find all pairs of rows which satisfy the join-predicate. When the join-predicate is satisfied, column values for each matched pair of rows are combined into a result row.

# Query 02

SELECT

employee.employee\_id,  
employee.first\_name,  
employee.last\_name,  
position.position\_title,  
department.department\_title

FROM

(employee INNER JOIN position ON employee.position\_id  
= position.position\_id) INNER JOIN department ON  
employee.department\_id =  
department.department\_id;

Query1				
employee_id: ▾	first_name: ▾	last_name: ▾	position_tit ▾	department_title: ▾
1	Jim	Halpert	SW Eng.	Dev. Dept.
2	Pamela	Beesly	QA Eng.	QA Dept.
3	Dwight	Schrute	Sr. SW Eng.	Dev. Dept.
4	Kelly	Kapoor	Syst. Eng.	Maintenance Dept.
5	Michael	Scott	PM	Management Dept.

position	
position_id: ▾	position_tit ▾
1	QA Eng.
2	SW Eng.
3	Sr. SW Eng.
4	Syst. Eng.
5	PM

department	
department ▾	department_title: ▾
1	Dev. Dept.
2	QA Dept.
3	Maintenance Dept.
4	Management Dept.
5	Accounting Dept.
6	HR Dept.

Query1				
employee_id: ▾	first_name: ▾	last_name: ▾	position_tit ▾	department_title: ▾
1	Jim	Halpert	SW Eng.	Dev. Dept.
2	Pamela	Beesly	QA Eng.	QA Dept.
3	Dwight	Schrute	Sr. SW Eng.	Dev. Dept.
4	Kelly	Kapoor	Syst. Eng.	Maintenance Dept.
5	Michael	Scott	PM	Management Dept.

skill	
skill_id: ▾	skill_title: ▾
1	Java SE
2	Maven
3	Java EE
4	Servlets
5	JSP
6	Git
7	UML

skills_matrix		
employee_id: ▾	skill_id: ▾	gained_date: ▾
1	1	5/30/2012
1	2	5/30/2012
1	5	4/12/2013
1	6	5/30/2012
3	1	8/10/2011
3	2	8/10/2011
3	4	12/5/2011
3	5	1/20/2012
3	6	12/5/2011
5	7	5/21/2008

# Query 03

SELECT

skills\_matrix.employee\_id,  
employee.first\_name,  
employee.last\_name,  
skill.skill\_title

FROM

(skills\_matrix INNER JOIN employee ON  
skills\_matrix.employee\_id =  
employee.employee\_id) INNER JOIN skill ON  
skills\_matrix.skill\_id = skill.skill\_id;

Query1				
employee_id: ▾	first_name: ▾	last_name: ▾	skill_title: ▾	
1	Jim	Halpert	Java SE	
1	Jim	Halpert	Maven	
1	Jim	Halpert	Git	
1	Jim	Halpert	JSP	
3	Dwight	Schrute	Java SE	
3	Dwight	Schrute	Maven	
3	Dwight	Schrute	Git	
3	Dwight	Schrute	Servlets	
3	Dwight	Schrute	JSP	
5	Michael	Scott	UML	

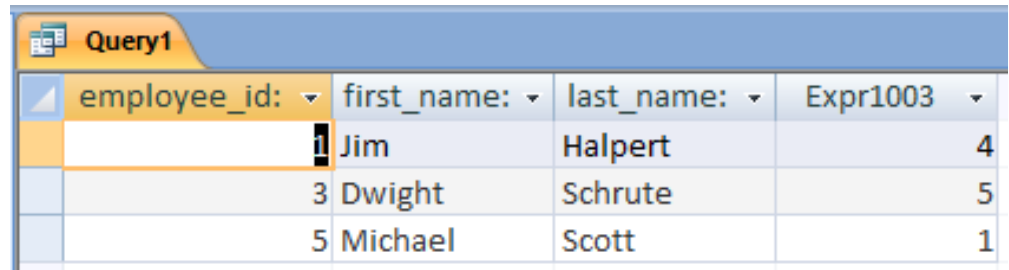
Query result:

skills have been assigned to 3 employees of 5

**HOW WE CAN SEE WHICH EMPLOYEES ARE NOT RELATED TO SKILLS?  
WHICH SKILLS ARE NOT RELATED TO EMPLOYEES?**

# Query 04a

```
SELECT
employee.employee_id,
employee.first_name,
employee.last_name,
COUNT(skills_matrix.skill_id)
FROM
employee INNER JOIN skills_matrix ON
    skills_matrix.employee_id = employee.employee_id
GROUP BY employee.employee_id, employee.first_name,
employee.last_name;
```

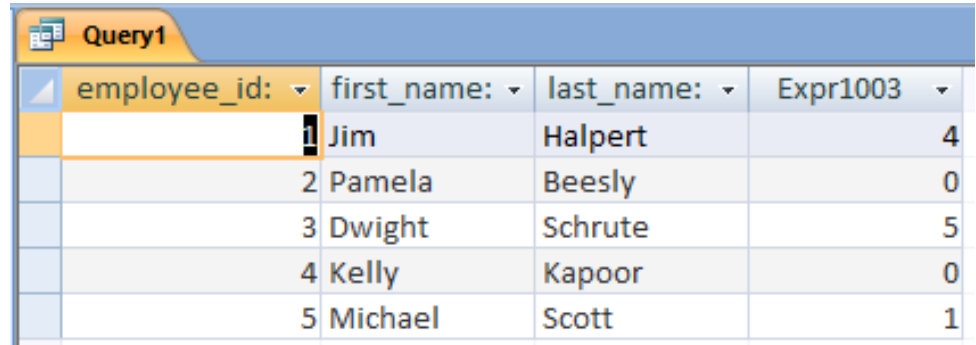


employee_id: ▾	first_name: ▾	last_name: ▾	Expr1003 ▾
1	Jim	Halpert	4
3	Dwight	Schrute	5
5	Michael	Scott	1



# Query 04b

```
SELECT
employee.employee_id,
employee.first_name,
employee.last_name,
COUNT(skills_matrix.skill_id)
FROM
employee LEFT JOIN skills_matrix ON
    skills_matrix.employee_id = employee.employee_id
GROUP BY employee.employee_id, employee.first_name,
employee.last_name;
```



employee_id: ▾	first_name: ▾	last_name: ▾	Expr1003 ▾
1	Jim	Halpert	4
2	Pamela	Beesly	0
3	Dwight	Schrute	5
4	Kelly	Kapoor	0
5	Michael	Scott	1

# Query 04c

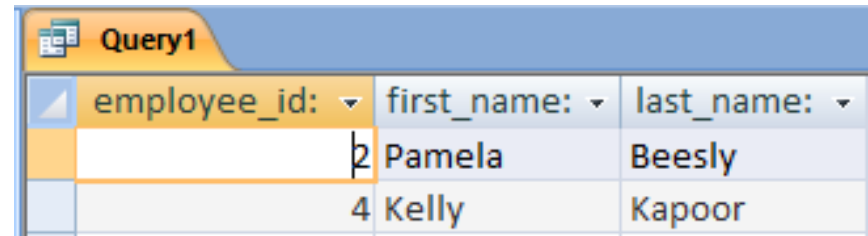
SELECT

employee.employee\_id,  
employee.first\_name,  
employee.last\_name

FROM

employee LEFT JOIN skills\_matrix ON  
skills\_matrix.employee\_id =  
employee.employee\_id

WHERE skills\_matrix.skill\_id IS NULL;



employee_id: ▾	first_name: ▾	last_name: ▾
2	Pamela	Beesly
4	Kelly	Kapoor

# SQL SELECT – LEFT JOIN

SQL LEFT JOIN повертає всі рядки лівої таблиці, навіть якщо у правій таблиці немає відповідностей. Це означає, що якщо умова ON відповідає 0 (нульовим) записам у правій таблиці; з'єднання все одно поверне рядок в результаті, але з NULL у кожному стовпці з правої таблиці.

Це означає, що LEFT JOIN повертає всі значення з лівої таблиці, а також відповідні значення з правої таблиці або NULL у випадку відсутності відповідності для предиката з'єднання.

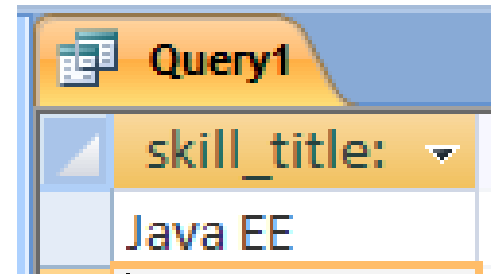
# SQL SELECT – LEFT JOIN

The SQL LEFT JOIN returns all rows from the left table, even if there are no matches in the right table. This means that if the ON clause matches 0 (zero) records in the right table; the join will still return a row in the result, but with NULL in each column from the right table.

This means that a left join returns all the values from the left table, plus matched values from the right table or NULL in case of no matching join predicate.

# Query 05

```
SELECT  
skill.skill_title  
FROM  
skills_matrix RIGHT JOIN skill ON  
    skills_matrix.skill_id = skill.skill_id  
WHERE skills_matrix.employee_id IS NULL;
```



# SQL SELECT – RIGHT JOIN

SQL RIGHT JOIN повертає всі рядки з правої таблиці, навіть якщо у лівій таблиці немає відповідностей. Це означає, що якщо умова ON відповідає 0 (нульовим) записам у лівій таблиці; з'єднання все-таки поверне рядок в результаті, але з NULL у кожному стовпці лівої таблиці.

Це означає, що RIGHT JOIN повертає всі значення з правої таблиці, а також відповідні значення з лівої таблиці або NULL у випадку відсутності відповідності для предиката з'єднання.

# SQL SELECT – RIGHT JOIN

The SQL RIGHT JOIN returns all rows from the right table, even if there are no matches in the left table. This means that if the ON clause matches 0 (zero) records in the left table; the join will still return a row in the result, but with NULL in each column from the left table.

This means that a right join returns all the values from the right table, plus matched values from the left table or NULL in case of no matching join predicate.

# SQL SELECT – FULL JOIN

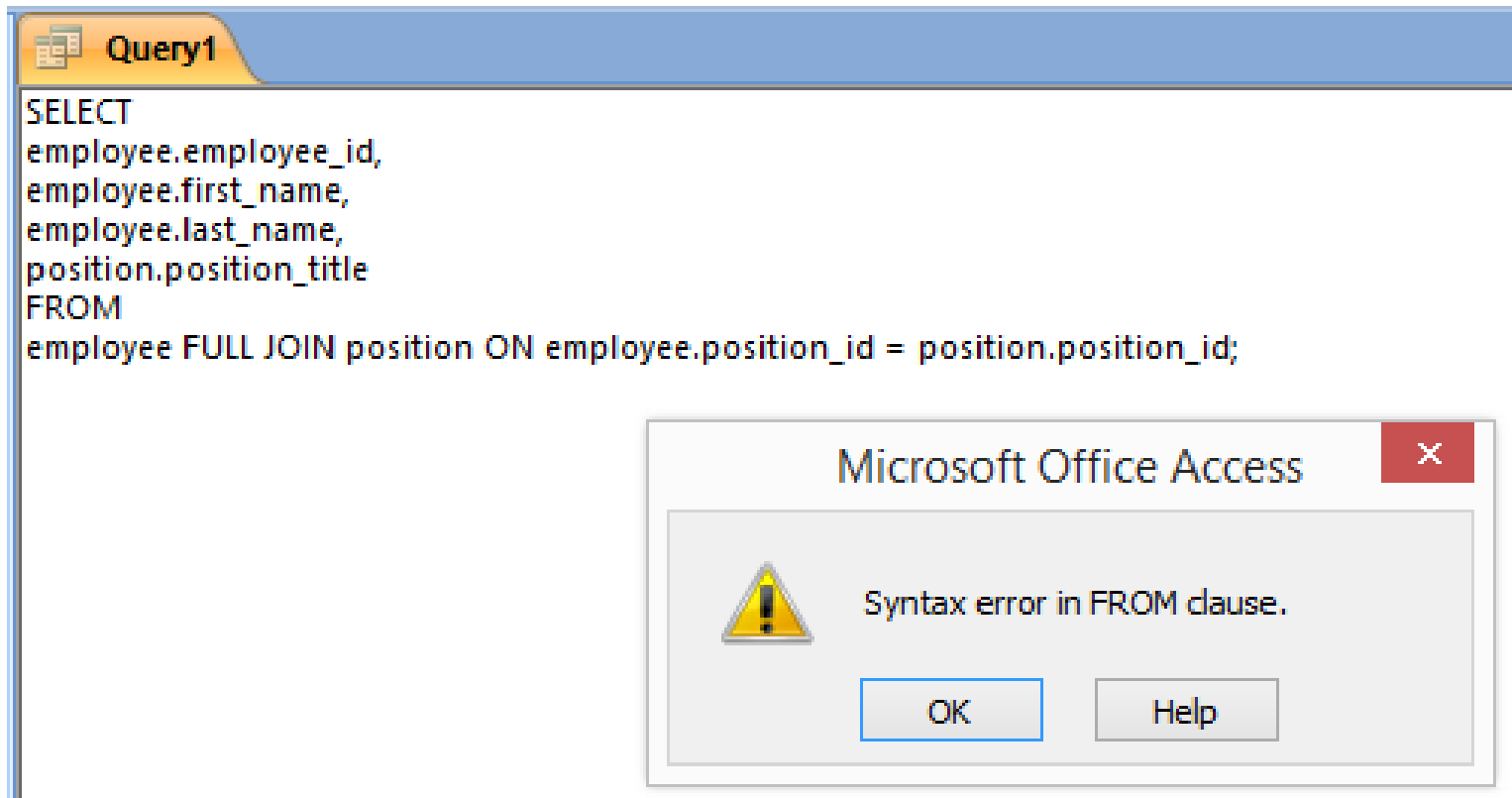
SQL FULL JOIN поєднує результати як з'єднання LEFT JOIN, так і з'єднання RIGHT JOIN.

Об'єднана таблиця буде містити всі записи з обох таблиць, а також значення NULL для відсутніх відповідностей з обох боків.

The SQL FULL JOIN combines the results of both left and right outer joins.

The joined table will contain all records from both the tables and fill in NULLs for missing matches on either side.





Some database management systems do not support FULL JOINS.

# Query 06

SELECT

employee.employee\_id,  
employee.first\_name,  
employee.last\_name,  
position.position\_title

FROM

employee LEFT JOIN position ON employee.position\_id = position.position\_id

UNION

SELECT

employee.employee\_id,  
employee.first\_name,  
employee.last\_name,  
position.position\_title

FROM

employee RIGHT JOIN position ON employee.position\_id = position.position\_id;

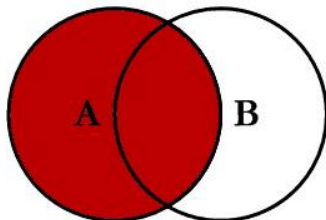
employee							
	employee_id: ▾	first_name: ▾	last_name: ▾	position_id: ▾	birth_date: ▾	onboarding ▾	department ▾
+	1	Jim	Halpert	2	5/12/1990	3/2/2014	1
+	2	Pamela	Beesly	1	11/2/1992	9/23/2015	2
+	3	Dwight	Schrute	3	2/11/1991	12/3/2013	1
+	4	Kelly	Kapoor	4	6/4/1993	3/2/2014	3
+	5	Michael	Scott	5	12/9/1989	4/28/2014	4

position	
position_id: ▾	position_tit ▾
1	QA Eng.
2	SW Eng.
3	Sr. SW Eng.
4	Syst. Eng.
5	PM

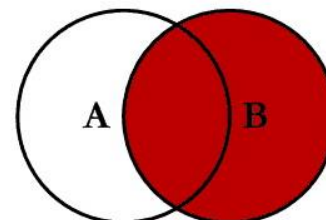
Query1			
employee_id: ▾	first_name: ▾	last_name: ▾	position_tit ▾
			CEO
			Sales Manager
1	Jim	Halpert	SW Eng.
2	Pamela	Beesly	QA Eng.
3	Dwight	Schrute	Sr. SW Eng.
4	Kelly	Kapoor	Syst. Eng.
5	Michael	Scott	PM

# SQL JOINS

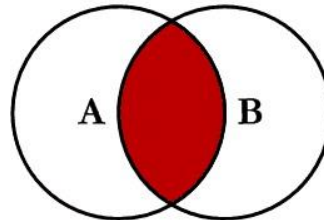
## SQL JOINS



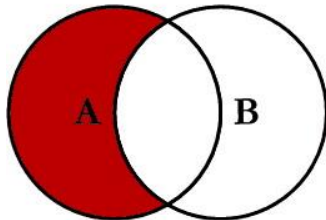
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



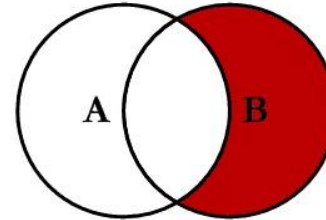
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



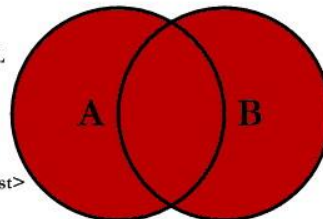
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



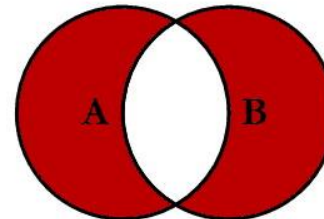
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

© C.L. Moffatt, 2008