

# KARPOV.COURSES >>> КОНСПЕКТ



## > Конспект > 3 урок > SQL

### > Оглавление

1. INNER JOIN
2. LEFT/RIGHT JOIN
3. FULL JOIN
4. Дополнительные типы соединений
5. CROSS JOIN

[Документация](#)

### > INNER JOIN

**INNER JOIN** – возвращаются только совпадающие строки. Без указания типа **JOIN** используется **INNER**. Пример запроса:

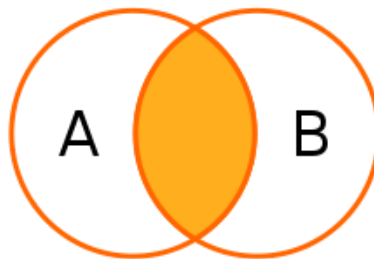
```
SELECT A.id as id, A.city as city, B.country as country  
FROM table_A as A  
JOIN table_B as B  
ON A.id = B.id
```

В виде диаграммы: Шаг 2Следующий шаг Преподаватель отключил комментарии для этого урока.

<b>A</b>		<b>B</b>	
id	city	id	country
1	London	2	Belarus
2	Brest	3	Russia

<b>C</b>		
id	city	country
2	Brest	Belarus



## > LEFT/RIGHT JOIN

**LEFT OUTER JOIN** – возвращаются строки из левой таблицы, и соответствующие строки из правой. Как можно заметить, ключевое слово **OUTER** можно опускать и использовать запись **LEFT JOIN**.

```
SELECT
    A.id as id,
    A.city as city,
    B.country as country
FROM
    table_A as A
LEFT JOIN
    table_B as B
ON
    A.id = B.id
```

**A**

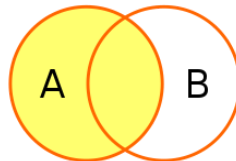
id	city
1	London
2	Brest

**B**

id	country
2	Belarus
3	Russia

**C**

id	city	country
1	London	
2	Brest	Belarus



**RIGHT OUTER JOIN** – возвращаются строки из правой таблицы, и соответствующие строки из левой.

```
SELECT
  A.id as id,
  A.city as city,
  B.country as country
FROM
  table_A as A
RIGHT JOIN
  table_B as B
ON
  A.id = B.id
```

**A**

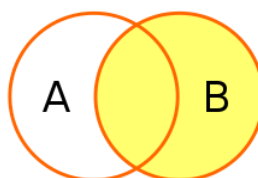
id	city
1	London
2	Brest

**B**

id	country
2	Belarus
3	Russia

**C**

id	city	country
2	Brest	Belarus
3		Russia



## > FULL JOIN

**FULL JOIN** – возвращаются все строки из обеих таблиц.

```
SELECT
  A.id as id,
  A.city as city,
  B.country as country
FROM
  table_A as A
FULL JOIN
  table_B as B
ON
  A.id = B.id
```

**A**

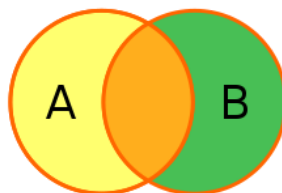
id	city
1	London
2	Brest

**B**

id	country
2	Belarus
3	Russia

**C**

id	city	country
1	London	
2	Brest	Belarus
3		Russia



## > Дополнительные типы соединений

В ClickHouse также присутствуют несколько других видов объединений.

1. **LEFT SEMI JOIN**
2. **RIGHT SEMI JOIN**
3. **LEFT ANTI JOIN**

#### 4. RIGHT ANTI JOIN

## LEFT/RIGHT SEMI JOIN

1. **LEFT SEMI JOIN** – возвращает данные только из левой таблички, которые имеют соответствие с правой табличкой. В отличие от **LEFT JOIN**, колонки из правой таблицы добавлены не будут.
2. **RIGHT SEMI JOIN** – аналогично, возвращает данные только из правой таблички, которые имеют соответствие с левой табличкой. Дополнительные столбцы из левой таблицы добавлены не будут.

Пример:

**A**

id	city
1	London
2	Brest

**B**

id	country
2	Belarus
3	Russia

**C**

id	city
2	Brest

## LEFT/RIGHT ANTI JOIN

3. **LEFT ANTI JOIN** – возвращает строки левой таблицы, для которых не нашлось соответствие в правой таблице.
4. **RIGHT ANTI JOIN** – возвращает строки правой таблицы, для которых не нашлось соответствие в левой таблице.

Пример:

**A**

id	city
1	London
2	Brest

**B**

id	country
2	Belarus
3	Russia

**C**

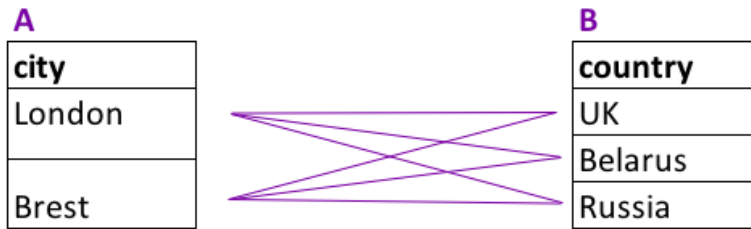
id	city
1	London

## > CROSS

**CROSS JOIN** – перекрёстное соединение. С помощью данного объединения мы получаем возможные комбинации значений из первой и второй таблицы. Например:

```
SELECT
  *
FROM
  A
CROSS JOIN
  B
```

Обратите внимание, при использовании **CROSS JOIN** ключи соединения указывать не нужно.



**C**

city	country
London	UK
London	Belarus
London	Russia
Brest	UK
Brest	Belarus
Brest	Russia