

# ВИДЫ СОЕДИНЕНИЯ ТАБЛИЦ: ДЖОИНЫ

## | ВВЕДЕНИЕ

На этом уроке вы

- Познакомитесь со способами объединения таблиц
- Научитесь создавать запросы, использующие данные одновременно из нескольких таблиц базы данных
- Научитесь создавать различные способы пересечения и объединения таблиц

## | ОПЕРАТОРЫ СОЕДИНЕНИЯ ТАБЛИЦ

Существуют следующие операторы соединения таблиц

CROSS JOIN

INNER JOIN

LEFT JOIN

RIGHT JOIN

FULL JOIN

### CROSS JOIN

Оператор **CROSS JOIN** формирует таблицу перекрестным соединением (декартовым произведением) двух таблиц при использовании оператора **CROSS JOIN** каждая строка левой таблицы сцепляется с каждой строкой правой таблицы

В результате получается таблица со всеми возможными сочетаниями строк обеих таблиц

ooo

```
SELECT
column_names
FROM
Table_1 CROSS JOIN Table_ 2
```

## INNER JOIN

Оператор **INNER JOIN** формирует таблицу из записей двух или нескольких таблиц. Каждая строка из первой таблицы, сопоставляется с каждой строкой из второй таблицы, после чего происходит проверка условия. Если условие истинно, то строки попадают в результирующую таблицу. В результирующей таблице строки формируются конкатенацией строк первой и второй таблиц

```
SELECT
column_names
FROM
Table_1 INNER JOIN Table_ 2
ON
Condition
```

## LEFT JOIN

Оператор **LEFT JOIN** осуществляет формирование таблицы из записей двух или нескольких таблиц. В операторе **SQL LEFT JOIN** важен порядок следования таблиц, так как от этого будет зависеть полученный результат

**Алгоритм работы оператора следующий:**

Сначала происходит формирование таблицы внутренним соединением (оператор **INNER JOIN**) левой и правой таблиц

Затем, в результат добавляются записи левой таблицы не вошедшие в результат формирования таблицы внутренним соединением. Для них, соответствующие записи из правой таблицы заполняются значениями **NULL**

```
SELECT
column_names
FROM
Table_1 LEFT JOIN Table_2
ON
condition
```

## RIGHT JOIN

Оператор **SQL RIGHT JOIN** осуществляет формирование таблицы из записей двух или нескольких таблиц. В операторе **RIGHT JOIN** важен порядок следования таблиц, так как от этого будет зависеть полученный результат

**Алгоритм работы оператора следующий:**

- 01** Сначала происходит формирование таблицы внутренним соединением (оператор **INNER JOIN**) левой и правой таблиц
- 02** Затем, в результат добавляются записи левой таблицы не вошедшие в результат формирования таблицы внутренним соединением. Для них, соответствующие записи из левой таблицы заполняются значениями **NULL**

ooo

```
SELECT
column_names
FROM
Table_1 RIGHT JOIN Table_ 2
ON
condition
```

## FULL JOIN

Оператор **FULL JOIN** осуществляет формирование таблицы из записей двух или нескольких таблиц. В операторе **SQL FULL JOIN** не важен порядок следования таблиц, он никак не влияет на окончательный результат, так как оператор является симметричным

Оператор **FULL JOIN** можно воспринимать как сочетание операторов **INNER JOIN LEFT** и **JOIN RIGHT JOIN**

**Алгоритм его работы:**

- 01** Сначала формируется таблица на основе внутреннего соединения (оператор **INNER JOIN**)
- 02** Затем, в таблицу добавляются значения не вошедшие в результат формирования из правой таблицы (оператор **LEFT JOIN**). Для них, соответствующие записи из правой таблицы заполняются значениями **NULL**.

- 03** Наконец, в таблицу добавляются значения не вошедшие в результат формирования из левой таблицы (оператор RIGHT JOIN). Для них, соответствующие записи из левой таблицы заполняются значениями NULL

## ИТОГ

- ✓ Вы познакомились и научились работать со всеми видами соединений: CROSS JOIN, INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN и разобрали практические особенности и тонкости их применения