



JOIN

Данная статья уточняет правило <joined table> из статьи [SELECT](#)

Версии сервера

	0.9	1.0	1.5.3	1.5.4	1.5.5	2.0	2.0.3	2.0.4	2.1	2.5	3.0
INNER, OUTER	+	+	+	+	+	+	+	+	+	+	+
CROSS	-	-	-	-	-	+	+	+	+	+	+
NATURAL, USING	-	-	-	-	-	-	-	-	+	+	+

Доступно в

[DSQL](#), [ESQL](#), [ISQL](#), [PSQL](#)

Формат

```
<joined table> ::=
  { <cross_join> | <qualified_join> | <named_columns_join> | <natural_join> }
```

```
<cross_join> ::=
  <reference_expression_list> CROSS JOIN <reference_expression_list>
```

```
<qualified_join> ::=
  <reference_expression_list> <join type> JOIN <reference_expression_list>
  ON <join condition>
```

```
<named_columns_join> ::=
  <reference_expression_list> <join type> JOIN <reference_expression_list>
  USING ( <column list> )
```

```
<natural_join> ::=
  <reference_expression_list> NATURAL <join type> JOIN <table primary>
```

```
<join type> ::=
  [{INNER | {LEFT | RIGHT | FULL} [OUTER]}]
```

Описание

Далее во всех примерах в качестве <reference_expression_list> будут использоваться следующие две таблицы:

```
CREATE TABLE A( A INT, B INT, C INT);  
CREATE TABLE B( A INT, B INT, D INT);
```

A

A	B	C
1	1	1
2	2	2

B

A	B	D
1	0	3
2	2	4

<qualified join>

«Обычное» соединение. Подразделяется на 4 типа (<join type>):

1. Внутреннее (INNER JOIN)
2. Внешнее Левое (LEFT OUTER JOIN)
3. Внешнее Правое (RIGHT OUTER JOIN)
4. Внешнее Полное (FULL OUTER JOIN)

Внутреннее

Из таблиц выделяются все возможные пары, для которых условие соединения <join condition> истинно

```
SELECT A.A AS AA, A.B AS AB, A.C AS AC, B.A AS BA, B.B AS BB, B.D AS BD  
FROM A INNER JOIN B ON A.B <= B.B
```

Эквивалентно неявному соединению:

```
SELECT A.A AS AA, A.B AS AB, A.C AS AC, B.A AS BA, B.B AS BB, B.D AS BD  
FROM A, B  
WHERE A.B <= B.B
```

Результат:

AA	AB	AC	BA	BB	BD
1	1	1	2	2	4
2	2	2	2	2	4

Внешнее Левое

Из левой таблицы выбираются все записи. Из правой выбираются только те, которые удовлетворяют условию соединения <join condition>. Если для какой-либо записи левой таблицы не нашлась ни одна пара, то столбцы правой таблицы заменяются NULL-ами.

```
SELECT A.A AS AA, A.B AS AB, A.C AS AC, B.A AS BA, B.B AS BB, B.D AS BD
FROM A LEFT JOIN B ON A.B = B.B
```

Результат:

AA	AB	AC	BA	BB	BD
1	1	1	NULL	NULL	NULL
2	2	2	2	2	4

Внешнее Правое

Эквивалентно левому, если поменять таблицы местами.

```
SELECT A.A AS AA, A.B AS AB, A.C AS AC, B.A AS BA, B.B AS BB, B.D AS BD
FROM A RIGHT JOIN B ON A.B = B.B
```

Эквивалентно:

```
SELECT A.A AS AA, A.B AS AB, A.C AS AC, B.A AS BA, B.B AS BB, B.D AS BD
FROM B LEFT JOIN A ON A.B = B.B
```

Результат:

AA	AB	AC	BA	BB	BD
NULL	NULL	NULL	1	0	3
2	2	2	2	2	4

Внешнее Полное

Представляет собой результат объединения левого и правого соединения

```
SELECT A.A AS AA, A.B AS AB, A.C AS AC, B.A AS BA, B.B AS BB, B.D AS BD
FROM A FULL JOIN B ON A.B = B.B
```

Результат:

AA	AB	AC	BA	BB	BD
1	1	1	NULL	NULL	NULL
2	2	2	2	2	4
NULL	NULL	NULL	1	0	3

<cross join>

Перекрестное соединение, или декартово произведение. Каждая строка левой таблицы соединяется с каждой строкой правой таблицы.

⚠ Появилось начиная с версии 2.0. В более ранних версиях необходимо использовать INNER JOIN с <join condition> тождественно равным истине.

```
SELECT A.A AS AA, A.B AS AB, A.C AS AC, B.A AS BA, B.B AS BB, B.D AS BD
FROM A CROSS JOIN B
```

Эквивалентно неявному соединению:

```
SELECT A.A AS AA, A.B AS AB, A.C AS AC, B.A AS BA, B.B AS BB, B.D AS BD
FROM A, B
```

В ранних версиях:

```
SELECT A.A AS AA, A.B AS AB, A.C AS AC, B.A AS BA, B.B AS BB, B.D AS BD
FROM A INNER JOIN B ON 1 = 1
```

Результат:

AA	AB	AC	BA	BB	BD
1	1	1	1	0	3
1	1	1	2	2	4
2	2	2	1	0	3
2	2	2	2	2	4

<named_columns join>

Является упрощенной записью <qualified join> для случая когда <join condition> содержит только сравнения на равенство и только одноименных столбцов. Часто встречается при соединению по **внешнему ключу**.

⚠ Появилось начиная с версии 2.1

⚠ Столбцы, которые участвуют в USING в результирующем наборе встречаются по одному разу.

Также как <qualified join> разделяется на 4 типа.

```
SELECT *
```

```
FROM A RIGHT JOIN B USING(A, B)
```

Эквивалентно:

```
SELECT COALESCE(A.A, B.A) AS A, COALESCE(A.B, B.B) AS B, A.C, B.D
FROM A RIGHT JOIN B ON A.A = B.A AND A.B = B.B
```

Результат:

A	B	C	D
1	0	NULL	3
2	2	2	4

<natural_join>

Является упрощенной записью <named_columns_join> для случая когда <column list> содержит все одноименные столбцы. Удобно при соединению по [внешнему ключу](#).

⚠ Появилось начиная с версии 2.1

⚠ В отличии от других СУБД в Firebird <natural_join> соединяет именно по **одноименным** столбцам, а не по [внешнему ключу](#).

```
SELECT *
FROM A NATURAL RIGHT JOIN B
```

Результат:

A	B	C	D
1	0	NULL	3
2	2	2	4

См. также

[SELECT](#)

<http://www.ibase.ru/devinfo/joins.htm>

Источник

<http://www.firebirdsql.org/rlsnotes/rlsnotes20.html#dml-dsql-cross-join>

<http://www.firebirdsql.org/rlsnotes/rlsnotes210.html#rnf210-joins-sntx>

From:

<http://firebirdsql.su/> - **Словарик по FireBird**

Permanent link:

<http://firebirdsql.su/doku.php?id=join>

Last update: **2009/12/09 14:00**

