

Объединение таблиц

Синтаксис операции объединения таблиц

```
SELECT список_выбора  
FROM таблица_1, таблица_2[, ..., таблица_N]  
WHERE [таблица_1].столбец  
      оператор_объединения  
      [таблица_2].столбец  
AND ...другие_условия...
```

Если столбцы, по которым выполняется объединение, имеют одинаковые имена, то в списке_выбора и в предложении WHERE нужно указать соответствующие им таблицы

Типы объединения таблиц

EQUIJOIN

NONEQUIJOIN

OUTER JOIN

SELF JOIN

CROSS JOINS

NATURAL JOINS

USING CLAUSE

Full or two sided OUTER JOINS

Arbitrary JOIN conditions for OUTER JOINS

EQUIJOIN – объединение по равенству (простое объединение)

Синтаксис этого объединения:

```
SELECT TABLE1.COLUMN, TABLE2.COLUMN  
FROM TABLE1, TABLE2  
WHERE TABLE1.COLUMN = TABLE2.COLUMN
```

Для объединения N-таблиц необходимо как минимум (N-1) условие.

Основные моменты объединения:

1. Желательно возле имя столбца в разделе SELECT указывать также имя (аллиас - синоним)
2. Если в разных таблицах используется одно и тоже имя столбца, то необходимо в качестве префикса указать имя используемой таблицы или её аллиас

Св-ва аллиаса:

1. Длина названия не должна превышать 30 символов
2. Аллиасы должны быть подобраны со смысловой нагрузкой
3. Аллиасы допустимы лишь для текущего запроса

Пример объединения по равенству:

```
SELECT T1.COLUMN, T2.COLUMN  
FROM TABLE1 T1, TABLE2 T2
```

NONEQUIJOIN – объединение по неравенству

Образуется объединением столбцов при помощи любого оператора(LIKE, BETWEEN, IN, >, <, >=, <=) соединения отличным от знака равенства

OUTER JOINS – внешнее соединение

Можно получить строки, которые нельзя получить из-за отсутствия соответствующих строк в другой таблице
Этот механизм реализован при помощи операнда (+), который ставится после имени столбца таблицы, в которой есть недостаток строк.

(+) – можно ставить только в одной части выражения

```
SELECT table1.column, table2.column FROM table1, table2  
WHERE table1.column(+) = table2.column
```

SELF JOIN – самообъединение

```
SELECT alias1.column, alias2.column  
FROM table1 alias1, table1 alias2  
WHERE alias1.column = alias2.column
```

Объединения таблиц при помощи SQL стандарта ANSI

SELECT table1.column, table2.column FROM table1

[CROSS JOIN table2] |

[NATURAL JOIN table2] |

[JOIN table2 USING(column_name)] |

[JOIN table2 ON

(table1.column_name = table2.column_name)] |

[INNER | {LEFT | RIGHT | FULL} JOIN table2 ON

(table1.column_name = table2.column_name)];

Таблица CITY (Города)

(Люди)

ID	Name
1	Москва
2	Санкт-Петербург
3	Казань

Таблица Person

Name	CityID
Андрей	1
Леонид	2
Сергей	1
Григорий	4

INNER JOIN – оператор внутреннего соединения

SELECT * FROM Person INNER JOIN City ON Person.CityID = City.ID

Результат:

Person.Name	Person.CityID	City.ID	City.Name
Андрей	1	1	Москва
Леонид	2	2	Санкт-Петербург
Сергей	1	1	Москва

LEFT OUTER JOIN – оператор левого внешнего соединения, соединяет две таблицы.

Порядок таблиц важен, т.к. оператор не является симметричным.

```
SELECT * FROM Person  
LEFT OUTER JOIN City ON Person.CityID = City.ID
```

Результат:

Person.Name	Person.CityID	City.ID	City.Name
Андрей	1	1	Москва
Леонид	2	2	Санкт-Петербург
Сергей	1	1	Москва
Григорий	4	NULL	NULL

RIGHT OUTER JOIN – оператор правого внешнего соединения

соединяет две таблицы

Порядок таблиц – важен.

```
SELECT * FROM Person
```

```
RIGHT OUTER JOIN City ON Person.CityID = City.ID
```

Результат:

Person.Name	Person.CityID	City.ID	City.Name
Андрей	1	1	Москва
Леонид	2	2	Санкт-Петербург
Сергей	1	1	Москва
NULL	NULL	3	Казань

FULL OUTER JOIN – оператор полного внешнего соединения

соединяет две таблицы

Порядок таблиц – не важен.

```
SELECT * FROM Person
```

```
    FULL OUTER JOIN City ON Person.CityID = City.ID
```

Результат:

Person.Name	Person.CityID	City.ID	City.Name
Андрей	1	1	Москва
Сергей	1	1	Москва
Леонид	2	2	Санкт-Петербург
NULL	NULL	3	Казань
Григорий	4	NULL	NULL

<http://apex.oracle.com>

WORKSPACE = YPPETELINA

USERNAME =

Y.P.PETELINA@GMAIL.COM

PASSWORD = 112233

SQL Workshop -> SQL Commands