СОЕДИНЕНИЕ ТАБЛИЦ

ТАБЛИЦЫ "ТОВАРЫ" И "ОПИСАНИЯ"

- Таблица с наименованием товаров хранит номер товара (id) и краткое название (name) и таблица с описанием товаров

id	name
1	Книга
2	Табуретка
3	Карандаш

id	description
1	Замечательная книга
3	Красный карандаш
5	Зелёная машинка

Таблица nomenclature содержит перечень всех товаров, которые есть в базе. Таблица описаний description, напротив, содержит лишь неполный перечень описаний для товаров, которые необязательно присутсвуют в базе. Чтобы однозначно привязать описание к товару, в таблицах присутвует столбец id, который содержит уникальный номер товара.

INNER JOIN (2 СПОСОБА)

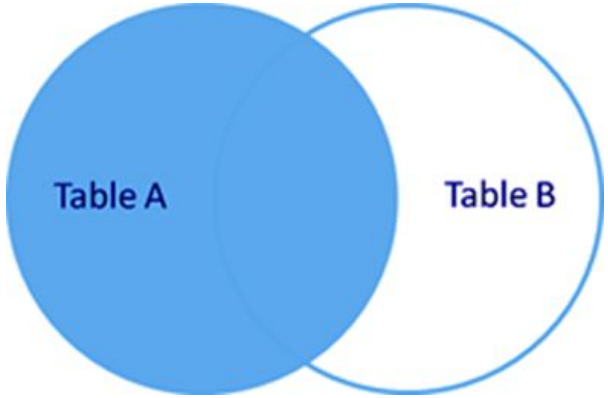
Код - способы объявления внутреннего объединения таблиц

SELECT * FROM Таблица1, Таблица2[, Таблица3, ...] [WHERE Условие1 [Условие2 ...]

SELECT * FROM Таблица1 [INNER | CROSS] JOIN Таблица2 [(ON Условие1 [Условие2 ...]) | (USING(Поле))]

LEFT JOIN

Левосторонние объединения позволяют извлекать данные из таблицы, дополняя их по возможности данными из другой таблицы.



LEFT JOIN

К примеру, чтобы получить полный список наименований товаров вместе с их описанием, нужно выполнить следующий запрос:

```
SELECT * FROM nomenclature LEFT JOIN description ON
nomenclature.id= description.id;
```

id	name	description
1	Книга	Замечательная книга
2	Табуретка	NULL
3	Карандаш	Красный карандаш

Поскольку для наименования Табуретка в таблице описаний нет подходящей записи, то в поле description подставился NULL. Это справедливо для всех записей, у которых нет подходящей пары.

LEFT JOIN

Если дополнить предыдущий запрос условием на проверку несуществования описания, то можно получить список записей, которые не имеют пары в таблице описаний:

SELECT id, name FROM nomenclature LEFT JOIN description ON nomenclature.id= description.id WHERE description IS NULL;

```
+----+-----------+
| id | name      |
+----+-----------+
|  2 | Табуретка |
+----+-----------+
```

- По сути это и есть основное назначение внешних запросов - показывать расхождение данных двух таблиц.
- Кроме того, при таком объединении обязательным является условие, которое задаётся через ON или USING. Без него запрос будет выдавать ошибку.

RIGHT JOIN

Этот вид объединений практически ничем не отличается от левостороннего объединения, за тем исключением, что данные берутся из второй таблицы, которая находится справа от констркуции JOIN, и сравниваются с данными, которые находятся в таблице, указанной перед конструкцией.

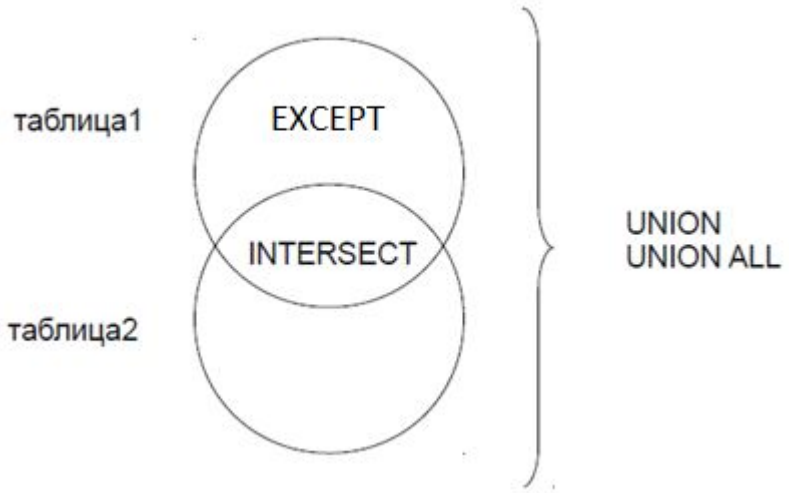
SELECT * FROM nomenclature RIGHT JOIN description ON nomenclature.id= description.id ;

id	description	name
1	Замечательная книга	Книга
3	Красный карандаш	Карандаш
5	Зелёная машинка	NULL

Как видно, теперь уже поле name содержит нулевые значения. Также поменялся и порядок расположения столбцов.
Однако, во всех случаях использования правосторонних объединений, запрос можно переписать, используя левостороннее объединение, просто поменяв таблицы местами, и наоборот. Следующие два запроса равнозначны:

SELECT * FROM nomenclature LEFT JOIN description ON nomenclature.id= description.id;
SELECT * FROM description RIGHT JOIN nomenclature ON nomenclature.id= description.id;

ОПЕРАТОРЫ СОЕДИНЕНИЯ



UNION возвращает все строки из обоих операторов SELECT; повторяющиеся значения удаляются.
UNION ALL возвращает все строки из обоих операторов SELECT; повторяющиеся значения показываются.

ОПЕРАТОРЫ СОЕДИНЕНИЯ

```
SELECT product_name
FROM purchase
ORDER BY product_name
```

product_name
Chrome Phoobar
Medium Widget
Medium Widget
Round Chrome Snaphoo
Small Widget
Small Widget

```
SELECT product_name
FROM purchase_archive
ORDER BY product_name
```

product_name
Chrome Phoobar
Large Harf linger
Medium Wodget
Round Snaphoo
Small Widget
Small Widget

```
SELECT product_name
FROM purchase
UNION
SELECT product_name
FROM purchase_archive
ORDER BY product_name
```

product_name
Chrome Phoobar
Large Harf linger
Medium Widget
Medium Wodget
Round Chrome Snaphoo
Round Snaphoo
Small Widget