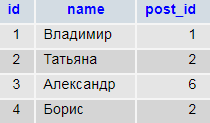
**SQL join в примерах с описанием**

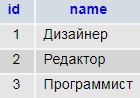
Присоединение таблиц в запросах — это базовый инструмент в работе с базами данных. Давайте рассмотрим какие присоединения (JOIN) бывают, и что от этого меняется в результатах запроса.

Для начала создадим две таблицы, над которыми будем проводить опыты. Это таблица с именами сотрудников и словарь с перечнем должностей.

Persons (Сотрудники)



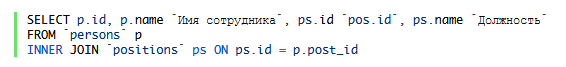
Positions (должности)



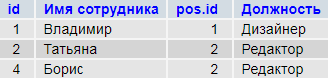
Т.е. чтобы узнать должность сотрудника, нужно присоединить соответствующие данные. Далее мы рассмотрим все варианты присоединений. Данные специально подобраны так, чтобы продемонстрировать отличия в результатах разных запросов.

**INNER JOIN**

Внутреннее присоединение. Равносильно просто JOIN или CROSS JOIN.

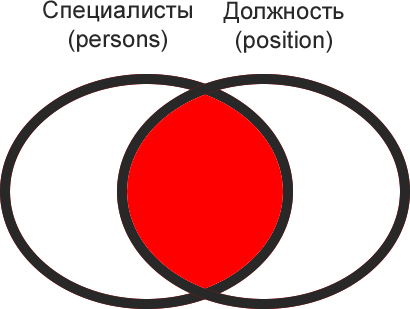


Такое присоединение покажет нам данные из таблиц только если условие связывания соблюдается — т.е. для сотрудника указан существующий в словаре идентификатор должности.



Если поменять порядок соединения таблиц — получим тот же результат.

Условно представим себе эти таблицы, как пересекающиеся множества, где пересечение — это наличие связи между таблицами. Получим картинку:



Далее проследим как получить разные части (подмножества) данного множества.

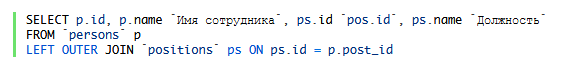
**OUTER JOIN**

Внешнее присоединение. Различают LEFT OUTER JOIN и RIGHT OUTER JOIN, и обычно опускают слово «OUTER».

Внешнее присоединение включает в себя результаты запроса INNER и добавляются «неиспользованные» строки из одной из таблиц. Какую таблицу использовать в качестве «добавки» — указывает токен LEFT или RIGHT.

**LEFT JOIN**

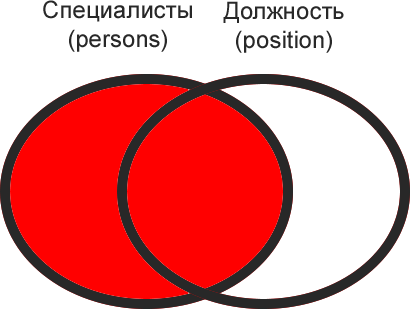
Внешнее присоединение «слева».





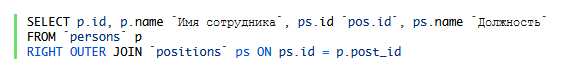
«Левая» таблица persons, содержит строку id#3 — «Александр», где указан идентификатор должности, отсутствующей в словаре.

На картинке это можно показать вот так:



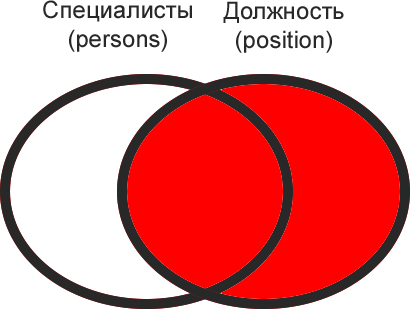
**RIGHT JOIN**

Присоединение «справа».





Словарь должностей (правая таблица) содержит неиспользуемую запись с id#3 — «программист». Теперь она попала в результат запроса.

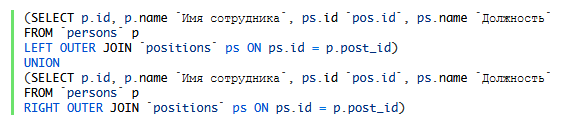


**Полное множество**

MySQL не знает соединения FULL OUTER JOIN. Что если нужно получить полное множество?



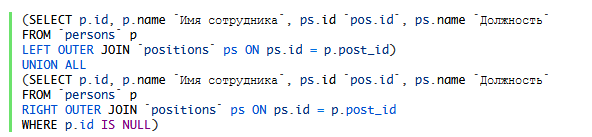
Первый способ — объединение запросов LEFT и RIGHT.





При таком вызове UNION, после слияния результатов, SQL отсечет дубли (как DISTINCT). Для отсечения дублей SQL прибегает к сортировке. Это может сказываться на быстродействии.

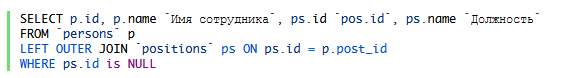
Второй способ — объединение LEFT и RIGHT, но в одном из запросов мы исключаем часть, соответствующую INNER. А объединение задаём как UNION ALL, что позволяет движку SQL обойтись без сортировки.

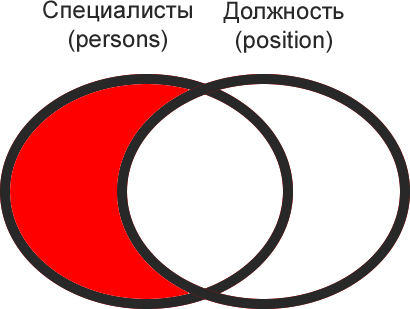


Этот пример показывает нам как исключить пересечение и получить только левую или правую часть множества.

**Левое подмножество**

LEFT JOIN ограничиваем проверкой, что данных из второй таблицы нет.



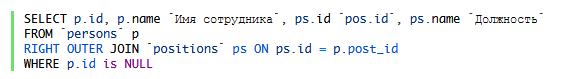


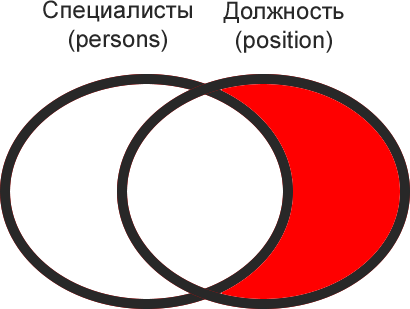
В нашем примере — это специалисты, у которых не задана должность или нет должности с указанным ключом.

https://shra.ru/wp-content/uploads/2017/09/most-left-join-res.png

**Правое подмножество**

Точно также выделяем правую часть.





https://shra.ru/wp-content/uploads/2017/09/most-right-join-res.png

В нашем случае получим должности, которые никому не назначены.

**Всё кроме пересечения**

Остался один вариант, тот когда исключено пересечение множеств. Его можно сложить из двух предыдущих запросов через UNION ALL (т.к. подмножества не пересекаются).

