



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)

id_person	name	position_ref
1	Владимир	1
2	Татьяна	2
3	Александр	6
4	Борис	2

Столбец *position_ref* (от «position reference») это ссылка на следующую таблицу, где перечислены должности сотрудников.

Positions (должности)

id_pos	title
1	Дизайнер
2	Редактор
3	Программист

Т.е. чтобы узнать должность сотрудника в таблице *Persons*, нужно присоединить соответствующие данные из таблицы *Positions*, связывая их по значениям *position_ref* и *id_pos*.

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 (верно для MYSQL, в стандарте SQL INNER JOIN не эквивалентен синтаксически CROSS JOIN, т.к. используется с выражением ON).

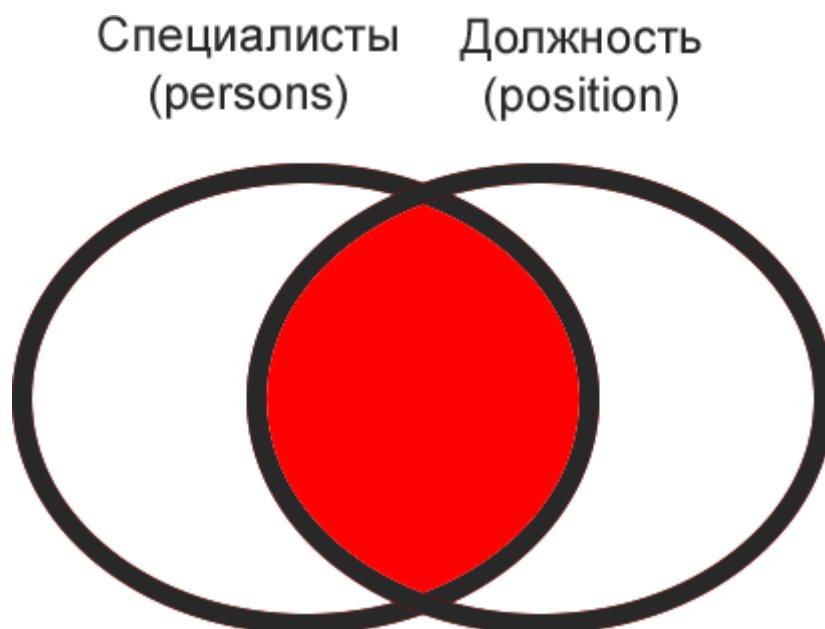
```
SELECT id_person, name, id_pos, title
FROM `persons`
INNER JOIN `positions` ON id_pos = position_ref
```

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

id_person	name	id_pos	title
1	Владимир	1	Дизайнер
2	Татьяна	2	Редактор
4	Борис	2	Редактор

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

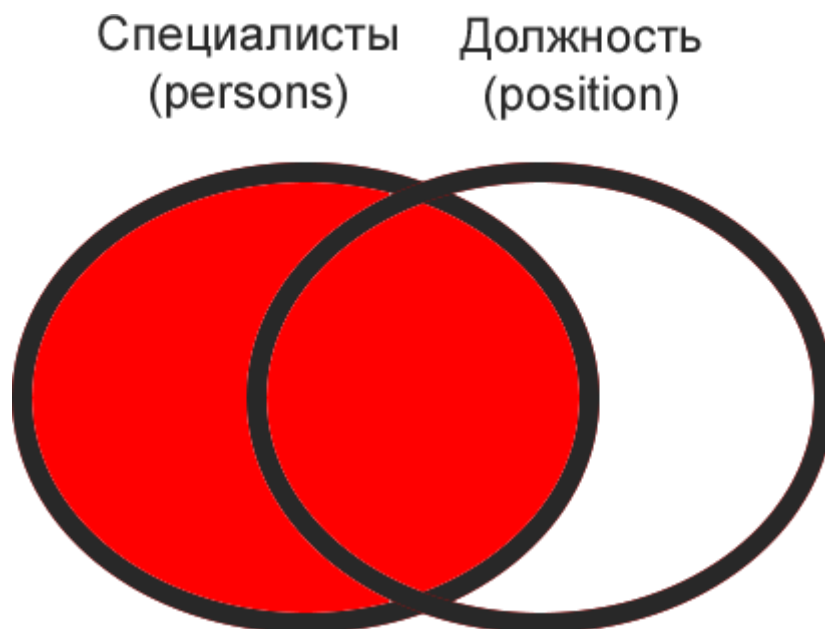
Внешнее присоединение «слева».

```
SELECT id_person, name, id_pos, title
FROM `persons`
LEFT OUTER JOIN `positions` ON id_pos = position_ref
```

id_person	name	id_pos	title
1	Владимир	1	Дизайнер
2	Татьяна	2	Редактор
4	Борис	2	Редактор
3	Александр	NULL	NULL

«Левая» таблица `persons`, содержит строку `id_person#3` — «Александр», где указан идентификатор должности, отсутствующей в словаре. Мы увидим все записи из «левой» таблицы, тогда как правая будет присоединена по возможности.

На картинке это можно показать вот так:



RIGHT JOIN

Присоединение «справа».

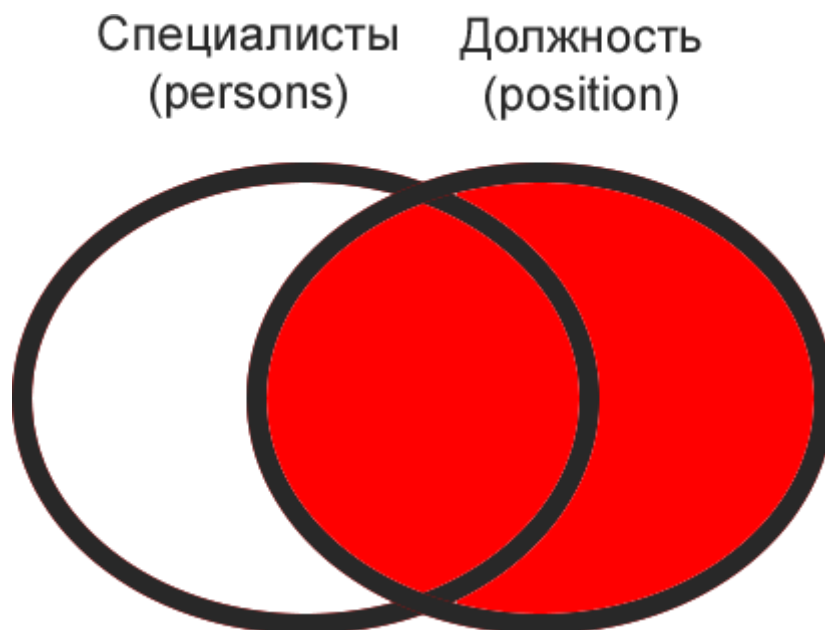
```
SELECT id_person, name, id_pos, title
FROM persons
RIGHT OUTER JOIN positions ON id_pos = position_ref
```

id_person	name	id_pos	title
1	Владимир	1	Дизайнер
2	Татьяна	2	Редактор
4	Борис	2	Редактор
NULL	NULL	3	Программист

Тут данные из «левой» таблицы присоединяются к «правой».

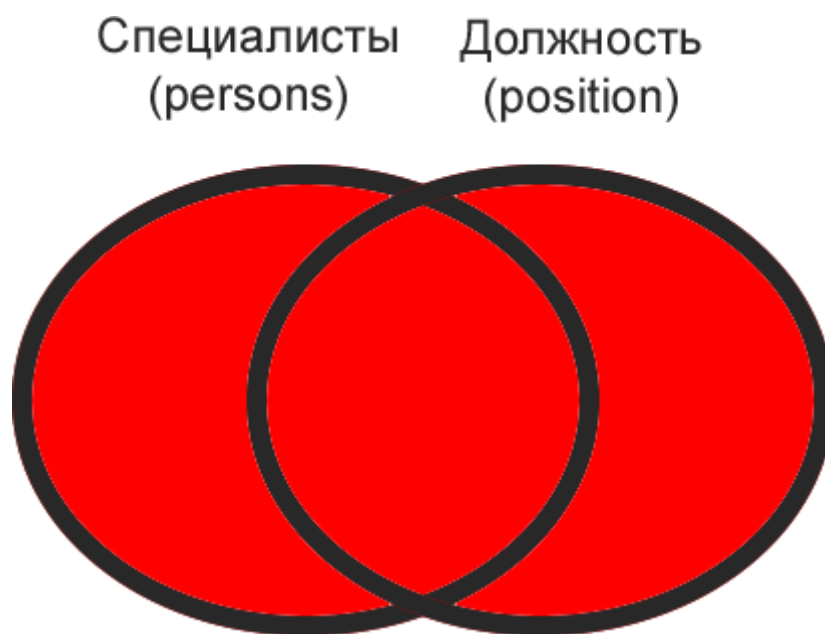
Словарь должностей (правая таблица) содержит неиспользуемую запись с `id_pos#3` — «программист». Теперь она попала в результат запроса.





Полное множество

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



Первый способ — объединение запросов LEFT и RIGHT.

```
(SELECT id_person, name, id_pos, title
FROM persons
LEFT OUTER JOIN positions ON id_pos = position_ref)

UNION

(SELECT id_person, name, id_pos, title
FROM persons
RIGHT OUTER JOIN positions ON id_pos = position_ref)
```

id_person	name	id_pos	title
1	Владимир	1	Дизайнер
2	Татьяна	2	Редактор
4	Борис	2	Редактор
3	Александр	NULL	NULL
NULL	NULL	3	Программист

При таком вызове UNION, после слияния результатов, SQL отсекает дубли (как DISTINCT). Для отсеечения 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.

```
(SELECT id_person, name, id_pos, title
FROM persons
LEFT OUTER JOIN positions ON id_pos = position_ref)

UNION ALL

(SELECT id_person, name, id_pos, title FROM persons
RIGHT OUTER JOIN positions ON id_pos = position_ref
WHERE id_person is NULL)
```

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.

```
SELECT id_person, name, id_pos, title
FROM persons
LEFT OUTER JOIN positions ON id_pos = position_ref
WHERE id_pos is NULL
```

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

Правое подмножество

Аналогично выделяем правую часть.

```
SELECT id_person, name, id_pos, title
FROM persons
RIGHT OUTER JOIN positions ON id_pos = position_ref
WHERE id_person is NULL
```

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).

```
(SELECT id_person, name, id_pos, title
FROM persons
LEFT OUTER JOIN positions ON id_pos = position_ref
WHERE id_pos is NULL)

UNION ALL

(SELECT id_person, name, id_pos, title
FROM persons
RIGHT OUTER JOIN positions ON id_pos = position_ref
WHERE id_person is NULL)
```

Это запрос соберет все случаи, когда по какой то причине данные из таблиц не связаны.

А графически такое объединение выглядит следующим образом:

💬 Написать комментарий

Данная запись опубликована в 19.09.2017 20:19 и размещена в [mySQL](#). Вы можете перейти в конец страницы и [оставить ваш комментарий](#).

Мало букафф? Читайте есчо !

Создание бекапов базы mySQL из консоли unix

Февраль 13, 2018 г.

Ещё одна шпаргалка по администрированию UNIX серверов. На этот раз рассмотрим несколько полезных команд для создания бекапа базы данных из консоли. Нам понадобится команда `mysqldump`, которая позволяет создавать sql скрипт базы или её части. Создание ...

Читать

Развертывание бекапов mySQL из консоли unix

Февраль 6, 2018 г.

Как развернуть дамп базы данных mySQL, созданный ранее? Дамп представляет из себя `mysql` скрипт, в котором последовательно создаются таблицы (`CREATE TABLE ...`) и, возможно, другие объекты и производится их наполнение (`INSERT INTO ...`). Нужно лишь отправить ...

Читать

Комментарии к «SQL join в примерах с описанием»

Понравилась статья? Есть вопросы? - пишите в комментариях.



Имя (обязательно)

Е-майл (не будет опубликован на сайте, обязательно для заполнения)

Комментарий:

☒ Я принимаю [соглашение сайта](#) об обработке персональных данных.

[Добавить комментарий](#)

Много комментариев в “SQL join в примерах с описанием”

← [Ранее](#) [1](#) [2](#)

Анна:

[22 февраля 2021 в 18:15](#)

Откуда взялись сокращения р. ps. и pos.? Учитывая, что таблицы две Persons и Positions, не понятно что где и, главное, зачем сокращено.

[Ответить](#)

Manul74:

[17 марта 2021 в 11:32](#)

Сокращать можно и нужно когда таблиц много и у них длинные имена. Тогда получается много текста и взглядом трудно понять что где. Сокращения помогут избавиться от лишней писанины. Это придет с опытом. После написания пару десятков сложных запросов с кучей join. Сами начнете сокращать.

[Ответить](#)

Андрей:

[2 апреля 2021 в 04:07](#)

Хорошая была бы заметка. Вот только вместе с борьбой за JOIN еще приходится распутывать загадки уважаемого автора с этими сокращениями. Две таблицы, для начинающих можно обойтись без умствований и загадочных сокращений.

[Ответить](#)

Виталий:

[21 мая 2021 в 12:56](#)

Поддерживаю ;), пока пытаешься разобраться с основами забываешь зачем начал их смотреть перекидывая силы на расшифровку, которая к сожалению в предисловия не выписана.

admin:

[5 сентября 2021 в 13:56](#)

Специально для «лиги лени» переписал статью. Если вы действительно хотите разобраться с языком запросов, то надо посвятить этому какое то время. Лучше всего попробовать реализовать какие то практические задачи.

[Ответить](#)

Данил:

[20 апреля 2021 в 16:14](#)

Спасибо, все четко и понятно. Освежил знания!

[Ответить](#)

Дмитрий:

1 июля 2021 в 18:31

«INNER JOIN

Внутреннее присоединение. Равносильно просто JOIN или CROSS JOIN.»

Мне кажется тут нужно поправить. CROSS JOIN, оно же декартово произведение, это совсем не INNER JOIN.

[Ответить](#)

admin:

4 июля 2021 в 14:40

Спасибо за замечание! Здесь действительно необходимо уточнить, что эти три варианта синтаксически эквивалентны в MYSQL. Тогда как в SQL стандарте — INNER JOIN используется с выражением ON, а CROSS в прочих случаях.

[Ответить](#)

Михаил:

24 сентября 2021 в 18:32

Вообще половину не понял. Откуда взялась например левая таблица, правая таблица, что за муть ?? Было две таблицы зачем их рядом ставить на картинке и делать запрос из левой таблицы (вообще в голове вот конкретно это не укладывается.). Что значит Внешнее присоединение «слева», присоединение чего и куда ??

[Ответить](#)

Дмитрий:

28 сентября 2021 в 21:15

Это самая подробная статья с объяснением и картинками, которую я видел), если тебе не понятно, что это и зачем, может это не твое, может оно тебе и не надо?)

[Ответить](#)

admin:

28 сентября 2021 в 21:27

Михаил, видимо, находится на одной из начальных фаз изучения вопроса. И тут можно сравнить это со стадиями принятия неизбежного. Т.е. он в «отрицании» или «гневе». Ребята из лиги лени дошли то «торга». Впереди еще возможна «депрессия», но я думаю он справится.

Владимир:

7 октября 2021 в 09:40

Автор, спасибо, конечно, за статью, но что за косяки такие с «<code>»? Оно же прямо в тексте примеров. Вы за собой не проверяете результат редактирования?

Вот что я вижу на странице в примерах:

```
...FROM persons
LEFT OUTER JOIN positions ON...
```

Можете устранить пожалуйста? Выглядит очень неряшливо, а для непосвящённых так вообще дико.

[Ответить](#)

admin:

7 октября 2021 в 12:28

Спасибо, Владимир, что заметили и маякнули мне! Поправил.

[Ответить](#)

Steel:

10 июня 2022 в 22:54

Спасибо, познавательно! :)

P.S.: дизайн поплыл что-то, а фон классный

[Ответить](#)

← [Ранее](#) 1 2

Поделиться ссылкой

Выбрать язык ▼

Технологии  Переводчик

Войти

Имя пользователя

Пароль

☒ Запомнить меня

Войти →

[Забыли пароль?](#)

Последние комментарии

Ран к записи Ветвления и циклы чертёжника

admin к записи Обработка шоткодов в Contact Form 7

Alexxx к записи Добавляем админа wordpress, имея доступ к базе данных

Jacob Lewis к записи Обработка шоткодов в Contact Form 7

Гость к записи Морфологический словарь русского языка в виде SQL скрипта

admin к записи Морфологический словарь русского языка в виде SQL скрипта

Оксана к записи Морфологический словарь русского языка в виде SQL скрипта

Александр к записи Обновление пакетов в linux/ubuntu

Свежие записи

Как проверить занят ли в port в unix

Классическая задача о размене монет

Как выполнить mysql запрос из командной строки

Composer в системном окружении windows

Как переписать /// referencetypes=react-scripts

Сделано на wordpress. В сети с 1999 г.

Копирование, полная или частичная перепечатка материалов сайта допускается лишь с прямой ссылкой на страницу-первоисточник.

[Пользовательское соглашение](#)

