

## Содержание:

image not found or type unknown



## Что такое связи между таблицами

В реляционной базе данных связи позволяют избежать избыточности данных. Например, в ходе создания базы данных, содержащей сведения о книгах, может появиться таблица под названием "Книги", в которой будут храниться параметры каждой книги, такие как ее название, дата публикации и издатель. Кроме того, есть и дополнительные сведения об издателе, которые может потребоваться сохранить, такие как его телефонный номер, адрес и почтовый индекс. Если хранить их в таблице с книгами, то телефонный номер издателя будет повторяться для каждой опубликованной им книги.

Более правильным вариантом является вынесение сведений об издателях в отдельную таблицу "Издатели". При этом таблица "Книги" будет содержать ссылки на записи таблицы "Издатели".

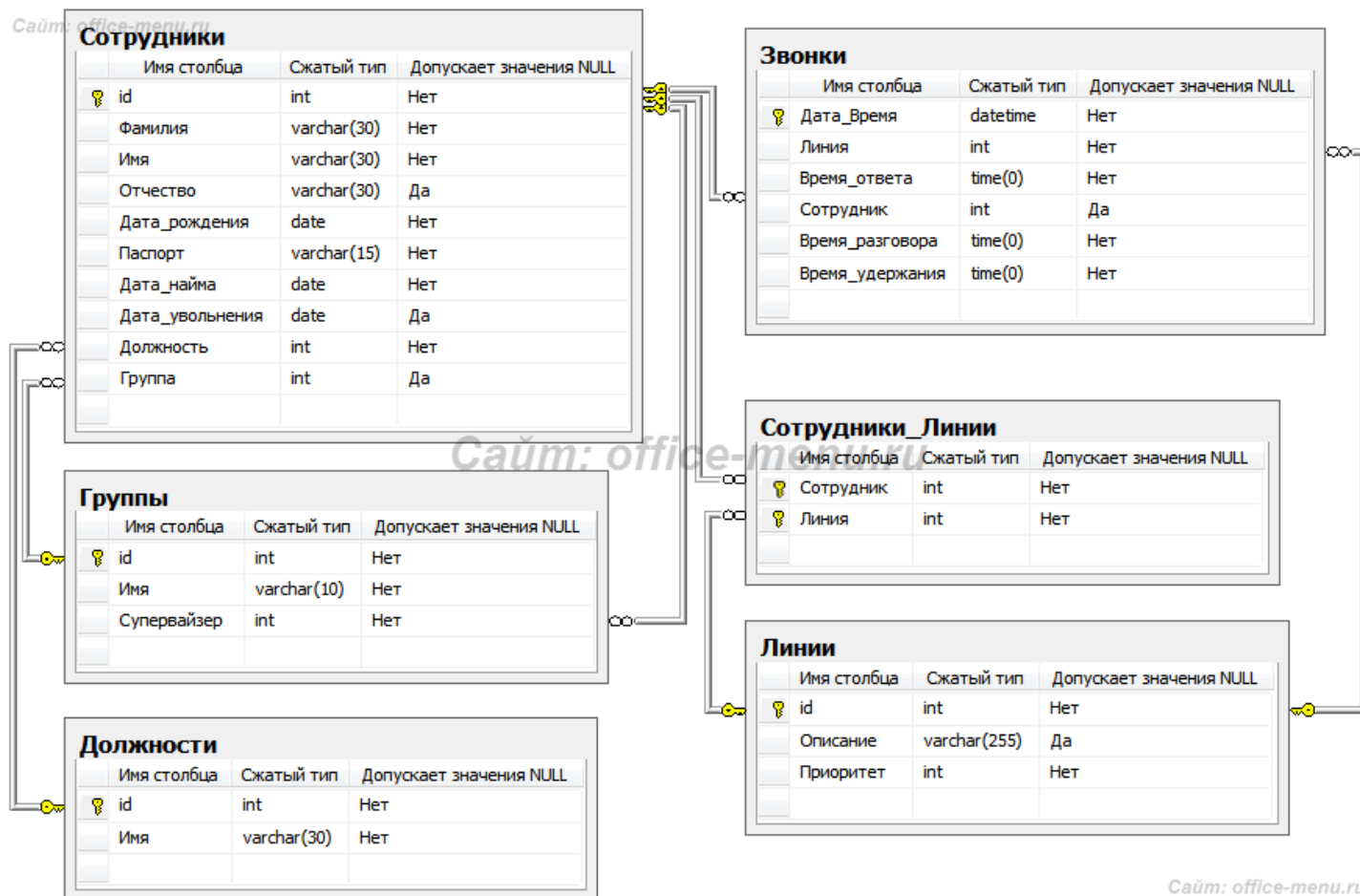
Чтобы сохранить синхронизацию, следует обеспечить целостность данных между таблицами "Книги" и "Издатели". Связи с обеспечением целостности данных позволяют следить за тем, чтобы данные в одной таблице соответствовали данным в другой. Например, каждая книга в таблице "Книги" связана с определенным издателем в таблице "Издатели". Добавить в таблицу книгу для издателя, отсутствующего в базе данных, невозможно.

Практически всегда БД не ограничивается одной таблицей. Сложно представить себе какой-либо бизнес-процесс на предприятии, который мог бы сконцентрироваться только на одном предмете в плане информации.

Рассмотрим пример учебной базы данных. Имеется отдел, который занимается обработкой звонков, поступающих на различные линии. Линии обслуживаются конкретными операторами. Операторы состоят в разных группах под присмотром супервайзеров.

Только из данного краткого описания можно выделить несколько самостоятельных объектов:

- Телефонные линии обслуживания;
- Сотрудники отдела;
- Должности сотрудников;
- Группы, по которым распределены сотрудники;
- Звонки.



Ознакомившись с диаграммой базы данных, можно обратить внимание на то, что некоторая информация из одних таблиц присутствует в других, т.е. между ними имеются связи.

В нашем конкретном случае, все таблицы можно соединить между собой. Чтобы понять, как это правильно сделать, необходимо рассмотреть типы связей.

Логика соединения таблиц в БД важно понять с самого начала изучения SQL, так как наверняка Вы не будете писать запросы только к одной таблице.

Всего существует 3 типа связей:

- Один к одному;
- Один ко многим;
- Многие ко многим.

Примечание:

В данном материале обозначения связей приводятся на примере MS SQL Server. В иных СУБД они могут обозначаться по-разному, но у Вас не должно возникнуть проблем с определением их типа, т.к. они либо очень похожи, либо интуитивно понятны.

## **Связь «Один к одному»**

Связь один к одному образуется, когда ключевой столбец (идентификатор) присутствует в другой таблице, в которой тоже является ключом либо свойствами столбца задана его уникальность (одно и тоже значение не может повторяться в разных строках).

На практике связь «один к одному» наблюдается не часто. Например, она может возникнуть, когда требуется разделить данных одной таблицы на несколько отдельных таблиц с целью безопасности.

В учебной базе данных нет подходящего примера, но гипотетически могла бы существовать необходимость разделения таблицы сотрудников.

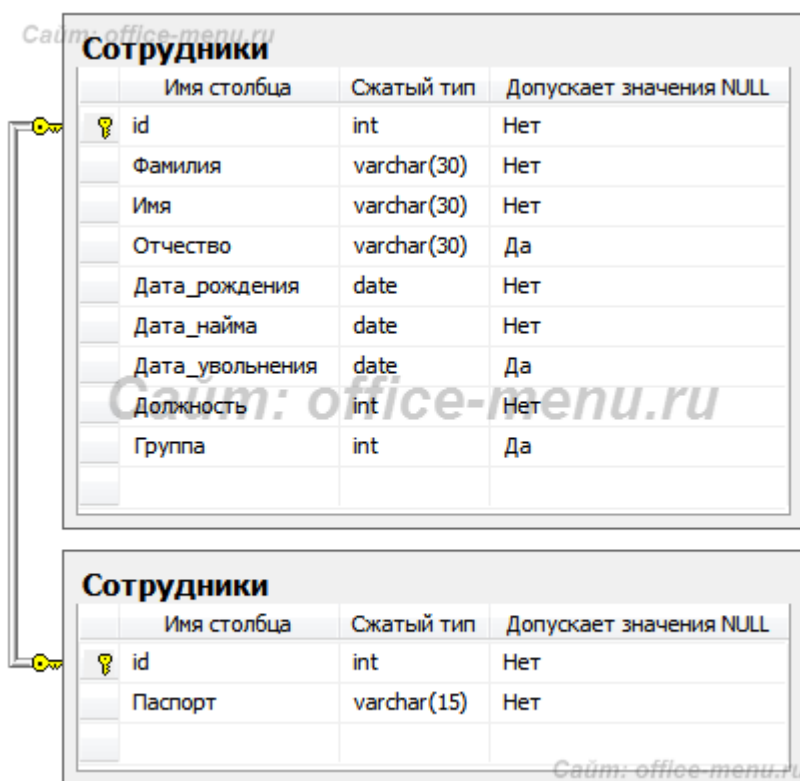
Пример:

Представьте, что базой данных пользуются несколько менеджеров и аналитиков, а таблица «Сотрудники» содержит те же столбцы, что и учебная база.

Следовательно, доступ к персональным данным может получить любой из упомянутых работников.

	id	Фамилия	Имя	Отчество	Дата_рождения	Паспорт	Дата_найма	Дата_увольнения	Должность	Группа
1	1	Мясников	Всеволод	Алексеевич	1987-01-02	6161 864808	2011-05-03	NULL	4	4
2	2	Беляев	Ярослав	Павлович	1990-01-02	8064 257116	2011-02-15	NULL	6	2
3	3	Степанов	Константин	Борисович	1981-02-05	9983 189856	2006-02-27	NULL	1	2
4	4	Калинина	Наталья	Анатольевна	1987-01-03	9453 359241	2013-03-08	NULL	4	1
5	5	Доронин	Олег	Леонидович	1992-04-04	1773 714978	2012-06-08	NULL	5	4
6	6	Никонов	Кирилл	Евгеньевич	1989-04-08	7428 261527	2011-10-31	NULL	4	1
7	7	Доронина	Дина	Вячеславовна	1993-01-04	9850 186129	2012-08-26	2014-08-16	5	3
8	8	Селезнёв	Владислав	Степанович	1992-02-12	5992 356004	2014-10-26	NULL	3	1
9	9	Дементьев	Иван	Степанович	1983-11-21	3543 037574	2010-11-23	NULL	5	3

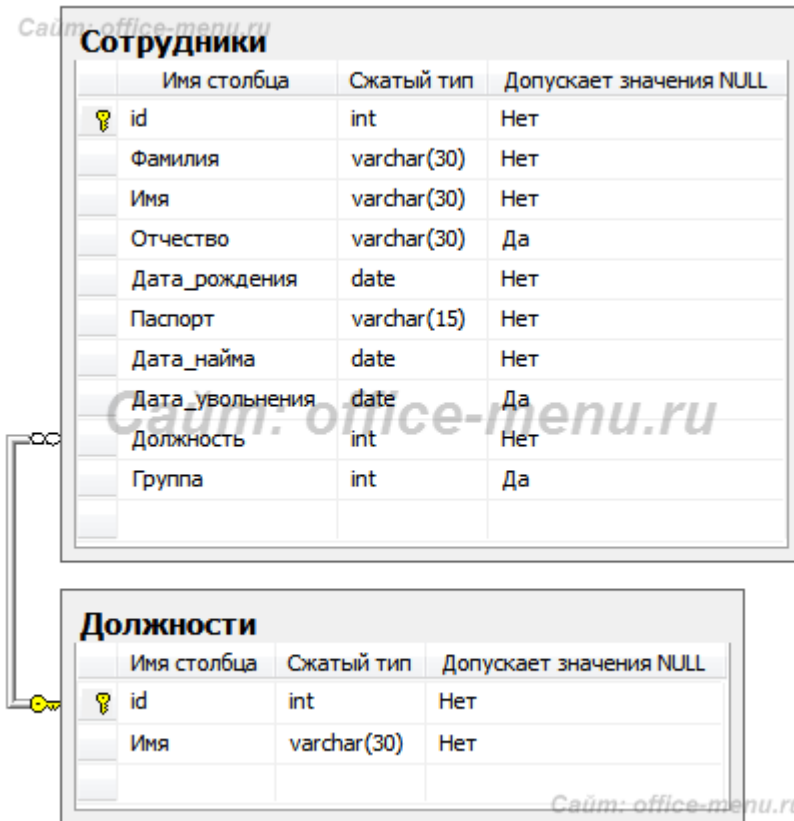
Чтобы устранить возможность утечки конфиденциальной информации, принимается решение о переносе информации паспортных данных в отдельную таблицу, доступ к которой предоставляется ограниченному кругу лиц.



## Связь «Один ко многим»

В типе связей один ко многим одной записи первой таблицы соответствует несколько записей в другой таблице.

Рассмотрим связь учебной базы данных между должностями и сотрудниками, которая относится к рассматриваемому типу.



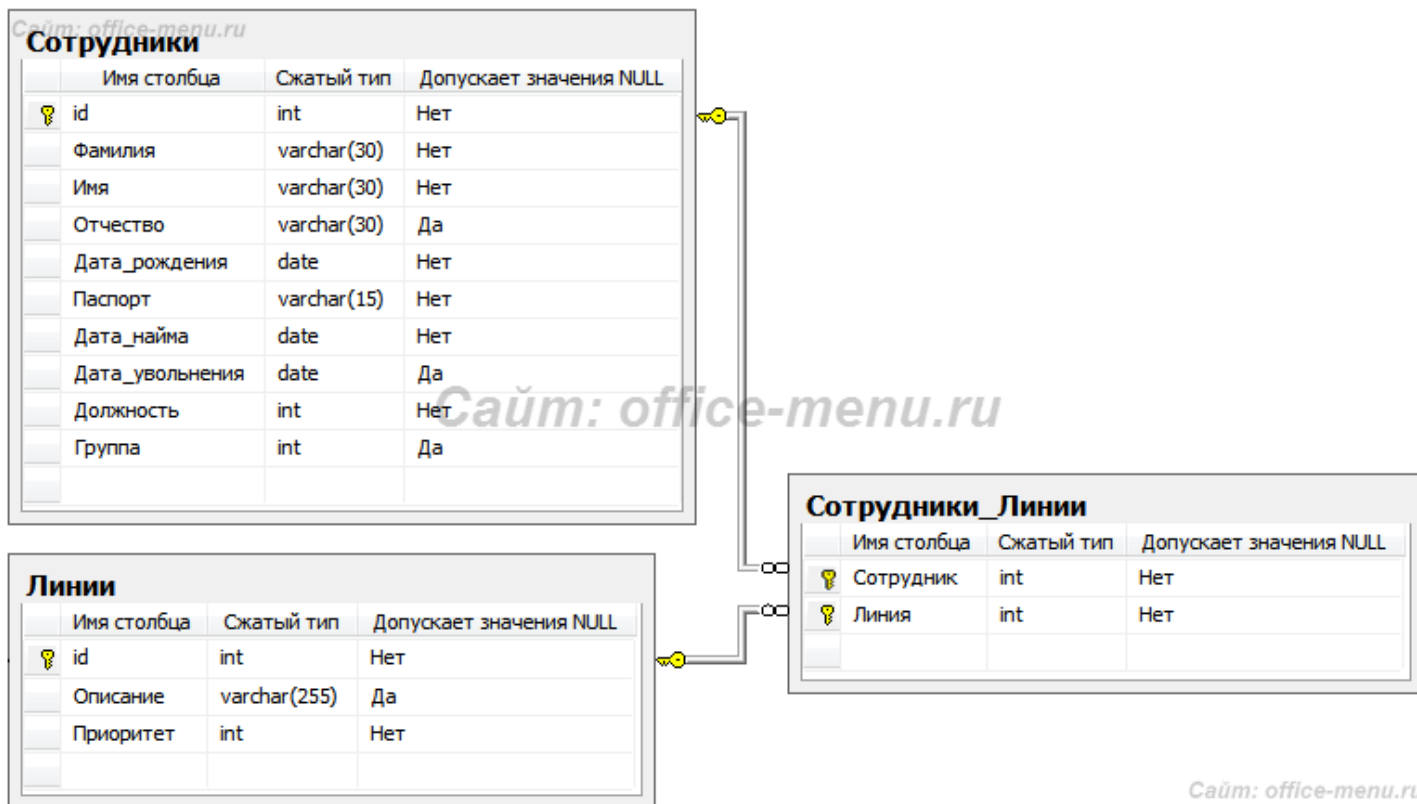
Записи должностей в таблице «Должность» уникальны, так как нет смысла повторно создавать имеющуюся запись. Записи в таблице «Сотрудники» также уникальны, но несколько различных сотрудников могут находиться на одинаковой должностной позиции.

Символ ключа на конце связи указывает, что таблица, к которой этой конец прилегает, находится на стороне «один» (связанный столбец является первичным ключом), а символ бесконечности находится на стороне «многие» (такой столбец является внешним ключом).

## Связь «Многие ко многим»

Если нескольким записям из одной таблицы соответствует несколько записей из другой таблицы, то такая связь называется «многие ко многим» и организовывается посредством связывающей таблицы.

В нашей базе подобное наблюдается только между таблицами с сотрудниками и линиями.



Из диаграммы видно, что имеются две связи «один ко многим» (один сотрудник может обрабатывать несколько телефонных линий, и одну линию могут обрабатывать несколько сотрудников), но в совокупности они образуют связь «многие ко многим».

## Для чего все это нужно?

Связи выполняют более важную роль, чем просто информация размещения данных по таблицам. Прежде всего они требуются разработчикам для поддержания целостности баз данных.

Правильно настроив связи, можно быть уверенным, что ничего не потеряется.

Представьте, что Вы решили удалить одну из групп в таблице учебной базы данных. Если бы связи не было, то для тех сотрудников, которые к ней были определены, остался идентификатор несуществующей группы. Связь не позволит удалить группу, пока она имеется во внешних ключах других таблиц. Для начала следовало определить сотрудников в другие имеющиеся или новые группы, а только затем удалить ненужную запись. Поэтому связи называют еще ограничениями.

## **СПИСОК ИСТОЧНИКОВ**

- <https://support.microsoft.com/ru-ru/help/304466/how-to-define-relationships-between-tables-in-an-access-database>
- <https://zametkinapolyah.ru/zametki-o-mysql/chast-3-2-vidy-svyazej-mezhdu-tablicami-v-baze-dannyx-svyazi-v-relyacionnyx-bazax-dannyx-otnosheniya-kortezhi-atributy.html>
- <http://office-menu.ru/uroki-sql/41-tipy-svyazej-v-relyatsionnykh-bazakh-dannykh>