Рассмотрим таблицу book



В этой таблице фамилии авторов повторяются для нескольких книг. А что, если придется вместо инициалов для каждого автора хранить его полное имя и отчество? Тогда, если в таблице содержится информация о 50 книгах Достоевского, придется 50 раз исправлять «Ф.М.» на «Федор Михайлович». При этом, если в некоторых записях использовать «Фёдор Михайлович» (c буквой ё), то мы вообще получим двух разных авторов...

Чтобы устранить эту проблему в реляционных базах данных создается новая таблица author, в которой перечисляются все различные (уникальные) авторы, а затем эта таблица связывается с таблицей book. При этом такая связь называется «**один ко многим**»,

таблица author называется главной, таблица book – связанной или подчиненной.

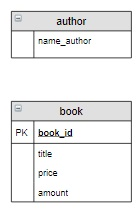
Связь «один ко многим» имеет место, когда одной записи главной таблицы соответствует несколько записей связанной таблицы, а каждой записи связанной таблицы соответствует только одна запись из главной таблицы. Обозначается это так:



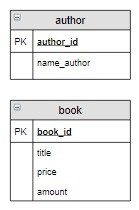
Этапы реализации связи «один ко многим» на следующем примере:

Один автор может написать несколько книг, а одна книга написана только одним автором. Для каждой книги известны ее количество и цена.

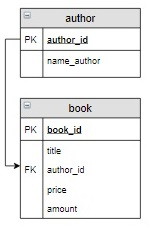
1. Создать таблицу author, в которую включить уникальных авторов книг, хранящихся на складе:



2. Обе таблицы должны содержать первичный ключ, в таблице book он уже есть, в таблицу author добавим ключ author\_id:



3. Включить в таблицу book **связанный столбец** (**внешний ключ, FOREIGN KEY**), соответствующий по имени и типу ключевому столбцу главной таблицы (в нашем случае это столбец author\_id). Для наглядности связь на схеме обозначается стрелкой от ключевого столбца главной таблицы к внешнему ключу связанной таблицы:



Технически получается, что в каждой таблице есть primary key, и для того чтобы создать связь один ко многим или многие к многим надо primary key одной таблицы поместить как самостоятельный столбец в другой таблице.