1. Основой реляционной модели является математическая теория множеств, которая требует от вас мыслить категориями множеств данных

* **каждый столбец представляет какой-то конкретный атрибут (свойство) того объектного типа, который отображается этой таблицей**
* данные, вводимые в любую позицию столбца, являются однозначными (ато- марными, неделимыми
* **каждая строка** описывает какой-нибудь объект (реального мира), являющийся уникальной реализацией объектного типа (например, какой-нибудь студент или определенная бизнес-встреча);

1. **Первичный ключ -** никогда не принимает значение null.

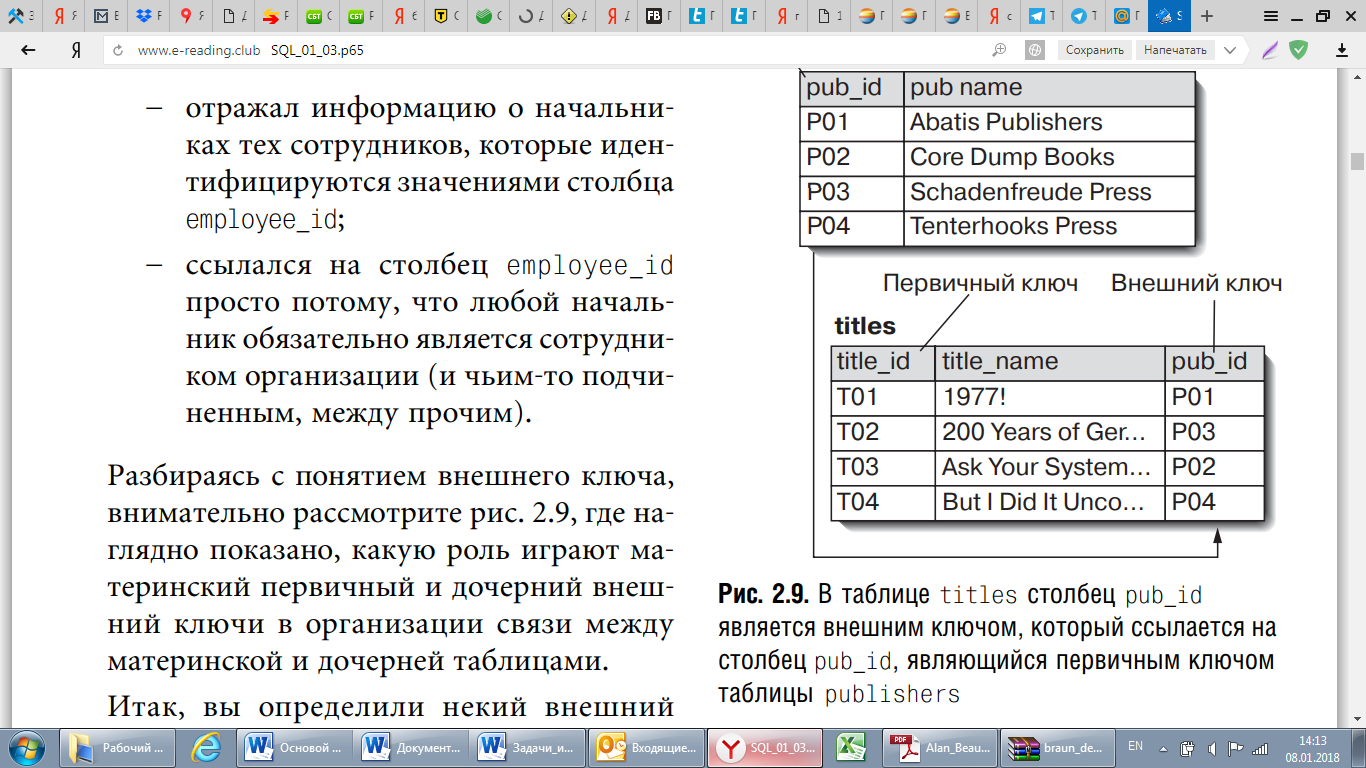
механизм их од- нозначной идентификации

* **простой или составной**. Итак, в любой таблице ее единственный первичный ключ может включать один или более столбцов этой таблицы. Если первичному ключу соответствует ровно один столбец, то такой ключ **называют простым**
* Любой первичный ключ, не яв- ляющийся простым и, следовательно, включающий два или более столбцов, **называют составным**
* Чтобы назначать первичные ключи, приме- няйте ограничение ***PRIMARY KEY***

1. **Внешний ключ**

**Любой внешний ключ обладает следующими свойствами:** 

* является или одним столбцом, или группой столбцов в некоторой таблице, но таким столбцом или группой таких столбцов**, что их значения или связаны со значениями столбцов другой какой- то таблицы, или содержат ссылки на значения столбцов этой таблицы;** 
* позволяет строкам из одной какой-ни- будь таблицы иметь соответствующие им строки в другой таблице; 
* таблица, которая содержит внешний ключ, называется **указывающей,** или дочерней, а таблица, на которую этот внешний ключ указывает, – **указываемо**й, или материнской



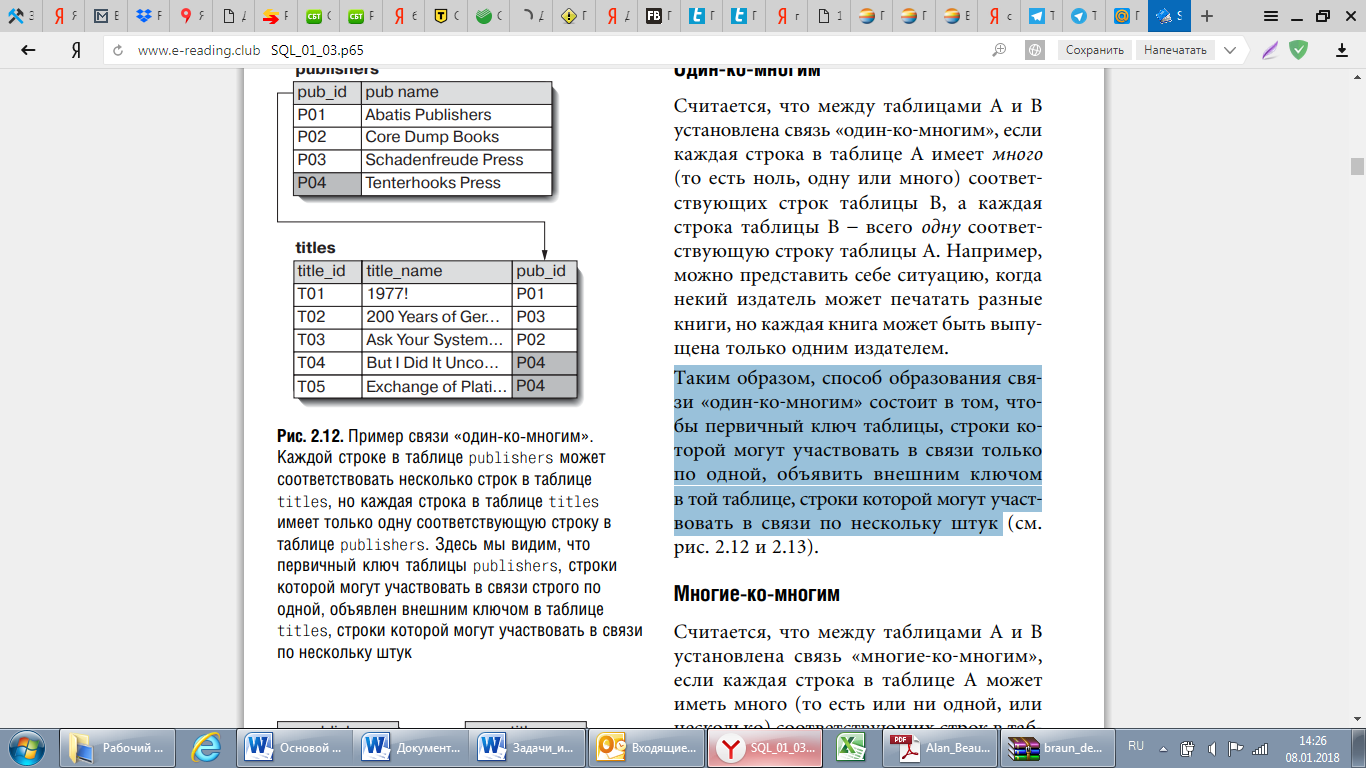
* **устанавливает прямую взаимосвязь с первичным (или альтернативным) ключом материнской таблицы.**
* в отличие от первичного ключа, значе- ниями внешнего ключа могут быть зна- чения NULL (то есть значения внешнего ключа могут быть пустыми или, иначе говоря, обнуляемыми; см. советы в кон- це этого раздела); 
* может иметь имя, отличное от имени родительского ключа
* любой внешний ключ может ссылать- ся на первичный ключ своей собствен- ной таблицы, а не только на первич- ный ключ какой-нибудь другой табли- цы. Всякая таблица, где имеет место такая ситуация, называется ссылаю- щейся на себя.

1. **Связи**

**4.1 1 – 1**

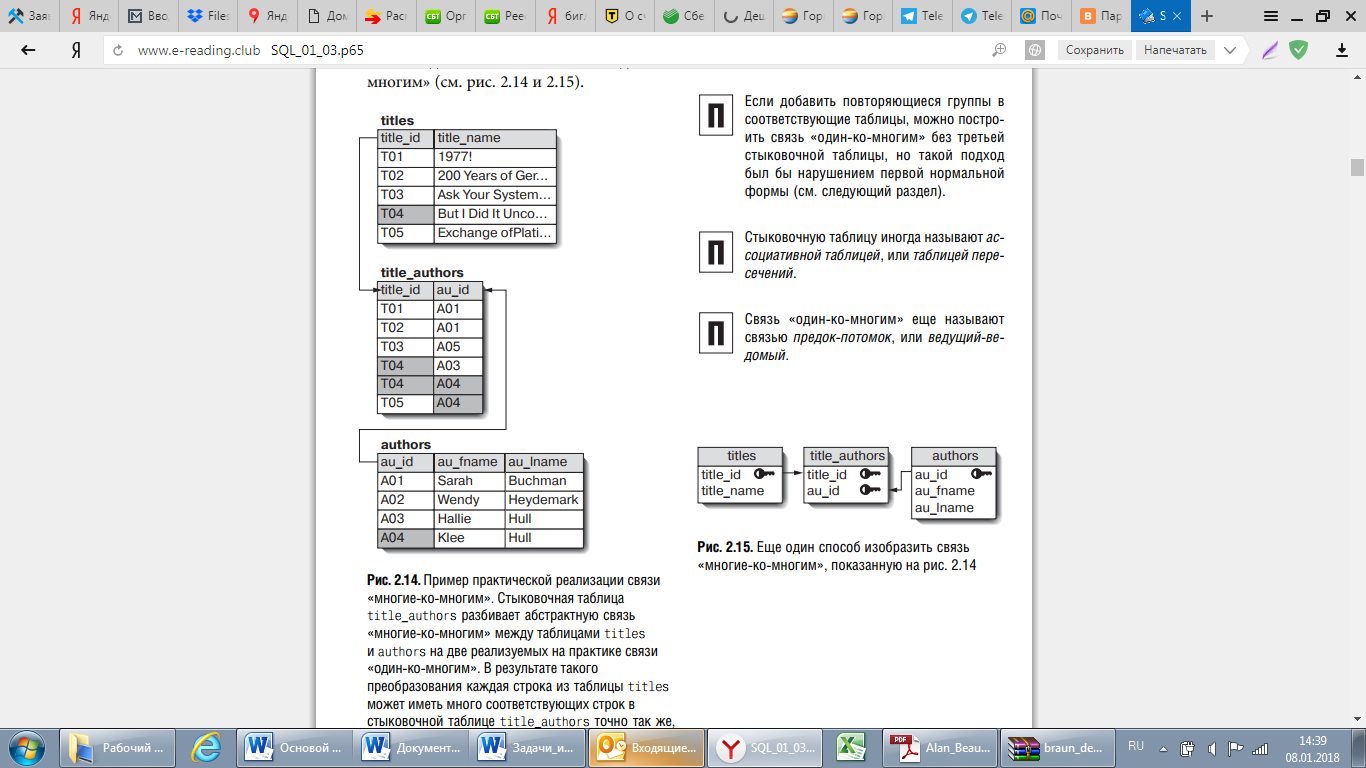
**4.2 1 – М**

Таким образом, способ образования св зи «один-ко-многим» состоит в том, чтобы первичный ключ таблицы, строки ко- торой могут участвовать в связи только по одной, объявить внешним ключом в той таблице, строки которой могут участ- вовать в связи по нескольку штук



**4.3 М – М**

Единственный способ установить связь «многие-ко-многим» между двумя произ- вольными таблицами заключается в том, чтобы создать третью таблицу (называе- мую стыковочной), в которой ее составной первичный ключ является комбинацией первичных ключей обоих таблиц со связью «один-ко-многим». При этом каждый из столбцов составного ключа по отдельнос- ти является внешним ключом



1. **Выражение явного преобразования типов CAST**

CAST(<выражение> AS <тип данных>)

1. **LIKE используется для сравнения значения для аналогичных значений с использованием подстановочных знаков.**

SELECT FROM table\_name

WHERE column LIKE 'XXXX%'

1. **SQL – Ограничения**

Обычно используются ограничения, доступные в SQL:

* [Ограничение NOT NULL](https://andreyex.ru/bazy-dannyx/uchebnoe-posobie-po-sql/sql-konstanta-not-null/): Обеспечивает , что столбец не может иметь значение NULL.
* [Ограничение DEFAULT](https://andreyex.ru/bazy-dannyx/uchebnoe-posobie-po-sql/sql-konstanta-default/): Предоставляет значение по умолчанию для столбца, если он не указан.
* [Ограничение UNIQUE](https://andreyex.ru/bazy-dannyx/uchebnoe-posobie-po-sql/sql-konstanta-unique/): Обязывает всем значениям в столбце быть уникальными.
* [PRIMARY Key](https://andreyex.ru/bazy-dannyx/uchebnoe-posobie-po-sql/sql-primary-key-pervichnyj-klyuch/): Уникальный идентификатор для каждой строки / записей в таблице базы данных.
* [FOREIGN Key](https://andreyex.ru/bazy-dannyx/uchebnoe-posobie-po-sql/sql-foreign-key-vneshnij-klyuch/): Уникальный идентификатор строк / записей в любой таблице базы данных.
* [Ограничение СНЕСК](https://andreyex.ru/bazy-dannyx/uchebnoe-posobie-po-sql/sql-ogranichenie-check/): проверочное ограничение гарантирует, что все значения в столбце удовлетворяют определенным условиям.
* [INDEX](https://andreyex.ru/bazy-dannyx/uchebnoe-posobie-po-sql/sql-konstanta-index/): Используется для создания и очень быстрого извлечения данных из базы данных.

**FOREIGN KEY(model) REFERENCES Product;**

1. ALTER TABLE PC
2. ADD CONSTRAINT fk\_pc\_product
3. FOREIGN KEY(model) REFERENCES Product(model);

Для обеспечения категорной целостности в языке [SQL](http://www.sql-tutorial.ru/ru/termins/sql.html) существуют спецификации **PRIMARY KEY** (первичный ключ) и **UNIQUE** (уникальный ключ). Первичный ключ может быть только один в таблице, уникальных же ключей может быть несколько. Т.е. у нас есть возможность для одного из потенциальных ключей задать спецификацию PRIMARY KEY, а для остальных – UNIQUE.

**Нормальные формы.**

**Нормализация** представляет процесс разделения данных по отдельным связанным таблицам. Нормализация устраняет избыточность данных (data redundancy) и тем самым избежать нарушения целостности данных при их изменении, то есть избежать аномалий изменения (update anomaly).

**1НФ**

Считается, что произвольная таблица на- ходится в первой нормальной форме, если одновременно выполняются следующие два условия: 

* любое значение любого столбца этой таблицы является элементарным;
* таблица не включает в себя повторяю- щихся групп
* произвольное значение любой табли- цы называется элементарным (иногда эти значения называют скалярами), если оно представляет собой единое целое, которое нельзя разделить на ча- сти без потери смысла (см. рис. 2.16); 
* любое множество из двух и более логи- чески связанных столбцов одной табли- цы называют повторяющейся группой

**Первая нормальная форма**(1НФ) говорит, что каждый атрибут отношения должен хранить атомарное значение, каждое отношение (строка в таблице) должно содержать одинаковое количество атрибутов (столбцов), т.е.

* запрещает повторяющиеся столбцы (содержащие одинаковую по смыслу информацию)
* запрещает множественные столбцы (содержащие значения типа списка и т.п.)
* требует определить первичный ключ для таблицы, то есть тот столбец или комбинацию столбцов, которые однозначно определяют каждую строку

**2НФ** хотя бы одно

*  первичный ключ этой таблицы не явля- ется составным (то есть состоит из од- ного столбца); 
* любой столбец этой таблицы входит в состав ее простого или составного первичного ключа.

**Значение любого столбца однозначно определяется только по совокупности первичного ключа**

«Могу ли я определить значение не- ключевого столбца, если знаю только часть значений первичного ключа?

отношение находится во второй нормальной форме, если оно находится в 1НФ, и при этом все неключевые атрибуты зависят только от первичного ключа, т.е.

* **Вторая нормальная форма требует, чтобы неключевые столбцы таблиц зависили от первичного ключа в целом, но не от его части.**
* Если таблица находится в первой нормальной форме и первичный ключ у нее состоит из одного столбца, то она автоматически находится и во второй нормальной форме.

**3НФ**

Считается, что произвольная таблица на- ходится в третьей нормальной форме, если **одновременно** выполнены два следующих условия:

*  она находится во второй нормальной форме;
*  она не содержит транзитивных зависи- мостей.

Отношение находится в **третьей нормальной форме**(3НФ), если оно находится во второй нормальной форме и каждый неключевой атрибут зависиттолько от первичного ключа и не зависят друг от друга.

Итак, некая таблица содержит по крайней мере одну транзи- тивную зависимость, если в этой таблице есть пара неключевых столбцов, в которых значение одного неключевого столбца оп- ределяет значение другого неключевого столбца.

«Могу я выяс- нить значение неключевого столбца, если буду знать только значение какого-нибудь другого неключевого столбца?