**Тема занятия №3: Многотабличные БД**

**Нормализация** – это процесс удаления избыточных данных.

**Нормализация** – это метод проектирования базы данных, который позволяет привести базу данных к минимальной избыточности.

Дело в том, что избыточность данных создает предпосылки для появления различных аномалий, снижает производительность, и делает управление данными не гибким и не очень удобным. Отсюда можно сделать вывод, что нормализация нужна для:

- Устранения аномалий

- Повышения производительности

- Повышения удобства управления данными

**Избыточность данных** – это когда одни и те же данные хранятся в базе в нескольких местах, именно это и приводит к аномалиям.

Существует 5 основных нормальных форм базы данных:

*- Первая нормальная форма (1NF)*

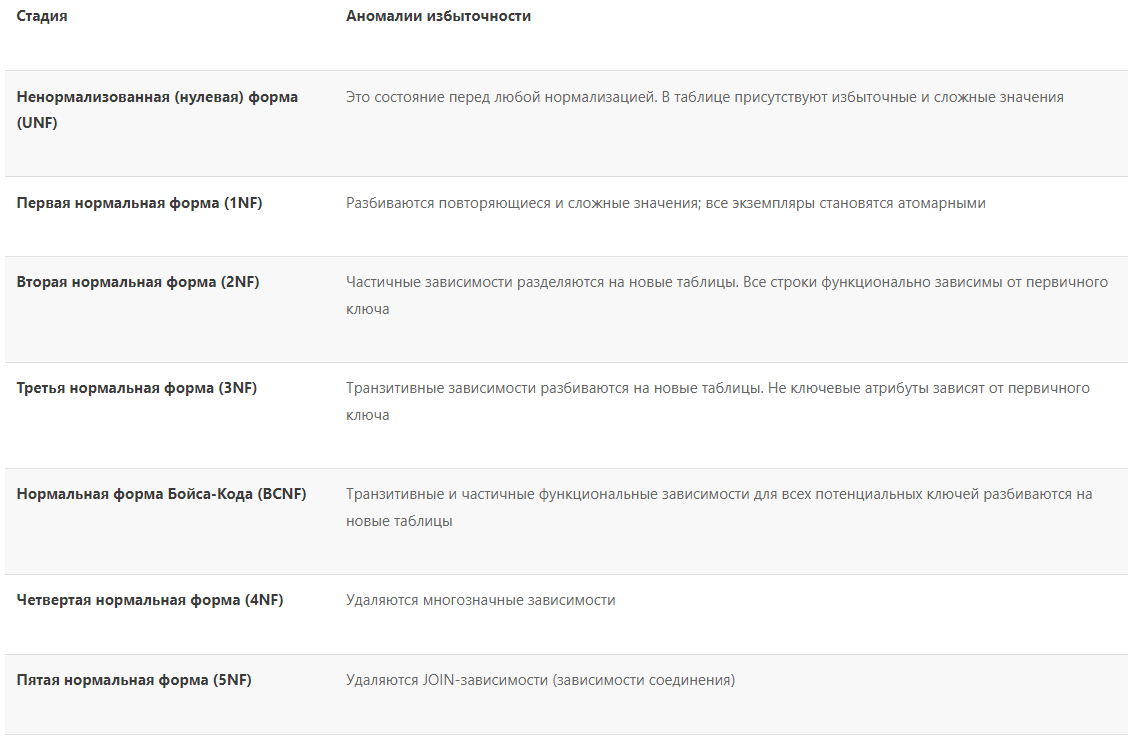
*- Вторая нормальная форма (2NF)*

*- Третья нормальная форма (3NF)*

*- Четвертая нормальная форма (4NF)*

*- Пятая нормальная форма (5NF)*

База данных считается нормализованной, если она находится как минимум в третьей нормальной форме (3NF).



Связи создаются с помощью внешних ключей (foreign key).

Внешний ключ — это атрибут или набор атрибутов, которые ссылаются на primary key или unique другой таблицы. Другими словами, это что-то вроде указателя на строку другой таблицы.

### **Виды связей**

Связи делятся на:

1. Многие ко многим.
2. Один ко многим.
   * с обязательной связью
   * с необязательной связью
3. Один к одному.
   * с обязательной связью
   * с необязательной связью

Рассмотрим подробно каждый из них.

### **Многие ко многим**

Представим, что нам нужно написать БД, которая будет хранить работником IT-компании. При этом существует некий стандартный набор должностей. При этом:

* Работник может иметь одну и более должностей. Например, некий работник может быть и админом, и программистом.
* Должность может «владеть» одним и более работников. Например, админами является определенный набор работников. Другими словами, к админам относятся некие работники.

Работников представляет таблица «Employee» (id, имя, возраст), должности представляет таблица «Position» (id и название должности). Как видно, обе эти таблицы связаны между собой по правилу многие ко многим: каждому работнику соответствует одна и больше должностей (многие должности), каждой должности соответствует один и больше работников (многие работники).

### **Один ко многим**

Эта самая распространенная связь между базами данных. Мы рассматриваем ее после связи многие ко многим для сравнения.

Предположим, нам нужно реализовать некую БД, которая ведет учет данных о пользователях. У пользователя есть: имя, фамилия, возраст, номера телефонов. При этом у каждого пользователя может быть от одного и больше номеров телефонов (многие номера телефонов).

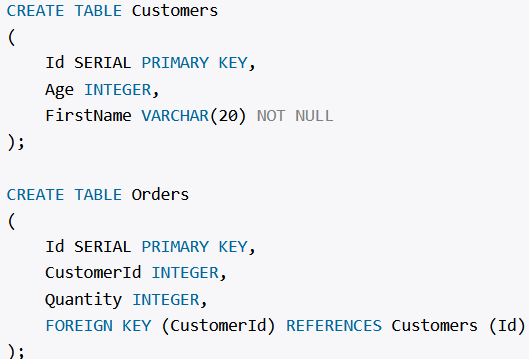
В этом случае мы наблюдаем следующее: пользователь может иметь многие номера телефонов, но нельзя сказать, что номеру телефона принадлежит определенный пользователь.

Другими словами, телефон принадлежит только одному пользователю. А пользователю могут принадлежать 1 и более телефонов (многие).

Как мы видим, это отношение один ко многим.

### **Один к одному**

Представим, что на работе вам дали задание написать БД для учета всех работников для HR. Начальник уверял, что компании нужно знать только об имени, возрасте и телефоне работника. Вы разработали такую БД и поместили в нее всю 1000 работников компании. И тут начальник говорит, что им зачем-то нужно знать о том, является ли работник инвалидом или нет. Наиболее простое, что приходит в голову — это добавить новый столбец типа bool в вашу таблицу. Но это слишком долго вписывать 1000 значений и ведь true вы будете вписывать намного реже, чем false (2% будут true, например).



Пример реализации связи.

Рассказать про UNION и все виды JOIN.