

Правило нормализаций Баз данных

1 первичный ключ

2 сурогатный ключ

3 внешние ключи

4 нормализация баз данных

Первичные ключи (Primary key - PK)

Используются для уникальной идентификации строки и ускорения поиска.

Первичный ключ должен соответствовать следующим требованиям:

1. содержать значение
2. иметь уникальное значение
3. значение должно оставаться неизменным

Если первичный ключ состоит из одного столбца, его называют **простым ключом**.

Если первичный ключ состоит из двух и более столбцов, его называют **составным ключом**.

Таблица заказов

<i>order_id</i>	<i>item</i>
1	Misfortune Box
2	Insanity Fountain
3	Karma Key
4	Mac

Таблица граждан

<i>series</i>	<i>number</i>	<i>name</i>
45	1234567	Cortez Hair
MA	2341256	Jonell Tomer
DI	0038110	Rosie Hash
AC	0909099	Tosha Hair

Суррогатный ключ - автоматически сгенерированное поле, никак не связанное с информационным содержанием записи таблицы.

Естественный ключ - набор атрибутов сущности, уникально ее идентифицирующих.

Таблица студентов

<u><i>id</i></u>	<i>name</i>
1	Bridget Kaminski
2	Jeffie Radel
3	Jonell Tomer
4	Ignacio Suitt

Таблица пользователей

<u><i>e-mail</i></u>	<i>name</i>
mthurn@live.com	Cortez Hair
jpguyer@aol.com	Tosha Lentini
firstpr@att.net	Jonell Tomer
miyop@icloud.com	Rosie Hash

Внешний ключи (Foreign key - FK)

Внешний ключ - столбец таблицы, который взаимосвязан со столбцом из другой таблицы, обычно с первичным ключом.

Таблица users

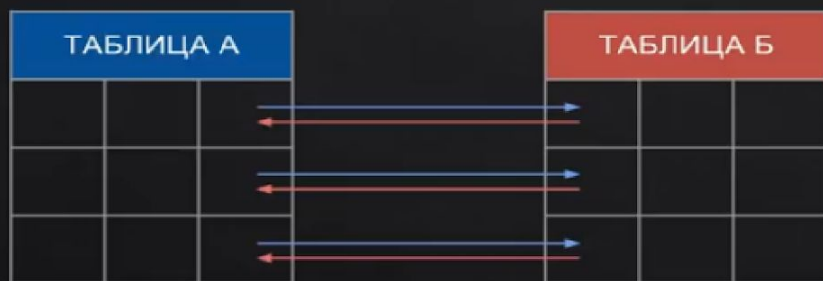
<i>id</i>	<i>username</i>
1	oma123
2	mega11
3	tapkismerti
4	supchik

Таблица comments

<i>id</i>	<i>user_id</i>	<i>message</i>
1	2	Первый
2	1	Супер!
3	4	начало с 0:00
4	3	repeat it

Один-к-одному

Запись из таблицы А может быть связана не более чем с одной записью в таблице Б.



Пример: Человек и его паспорт.

Один-ко-многим

Запись в таблице А может быть связана с многими записями в таблице Б, но запись в таблице Б может быть связана только с одной записью в таблице А.



Пример: Мать может иметь множество детей, но каждый ребенок может иметь только одну мать.

Многие-ко-многим

Запись в таблице А может быть связана **с многими** записями в таблице Б, а запись в таблице Б может быть связана **с многими** записями в таблице А.



Обычно состоят из двух связей «один-ко-многим», объединенных при помощи соединительной таблицы.

Нормализация

Нормализация — процесс приведение структуры базы данных в порядок в соответствии с определенными правилами.

Цель нормализации сводится к тому, чтобы каждый факт появлялся лишь один раз, т.е. исключить избыточность информации, а также устранить противоречивость информации.

Первая нормальная форма

Таблица находится в первой нормальной форме, если

- 1) каждая ячейка столбца содержит только одно значение
- 2) не содержит повторяющуюся информацию
- 3) используется первичный ключ для идентификации строки

<i>id</i>	<i>username</i>	<i>friends</i>
1	oma123	mega11
2	mega11	oma123, supchik, tapkismerti
3	tapkismerti	mega11
4	supchik	mega11



Такая запись является нарушением первой нормальной формы.

Вторая нормальная форма

Таблица находится во второй нормальной форме, если:

- 1) находится в первой нормальной форме
- 2) каждый столбец полностью зависит от первичного ключа.

Следовательно, вторая нормальная форма может быть нарушена лишь в том случае, если ключ является составным, т.е. ключ состоит из нескольких столбцов.

<i>teacher</i>	<i>subject</i>	<i>birthday</i>
1	Алгебра	25
2	Биология	27
1	Геометрия	25
3	Спорт	23

Третья нормальная форма

Таблица находится в третьей нормальной форме, если:

- 1) находится во второй нормальной форме
- 2) отсутствуют транзитивные зависимости: любой столбец таблицы должен зависеть только от ключевого столбца.

<i>id</i>	<i>username</i>	<i>birthday</i>	<i>age</i>
1	oma123	03.12.90	26
2	mega11	09.10.91	27
3	tapkismerti	20.06.83	34
4	supchik	14.08.88	29

```
CREATE TABLE movies_genres(  
  movie_id INT,  
  genre_id INT,  
  FOREIGN KEY(movie_id) REFERENCES movie(id) ON DELETE CASCADE,  
  FOREIGN KEY(genre_id) REFERENCES genres(id) ON DELETE CASCADE,  
  PRIMARY KEY(movie_id, genre_id)  
);
```



Если указано выражение **ON DELETE CASCADE** и строка в родительской таблице удалена (**movies** или **genres**), то все взаимосвязанные строки из дочерней таблицы (**movies_genres**) также будут удалены.

Таблица **movies**

<i>id</i>	<i>title</i>	<i>duration</i>
1	Логан	141
2	Остров Проклятых	138
3	Престиж	125
4	Прибытие	116

Таблица **movies_genres**

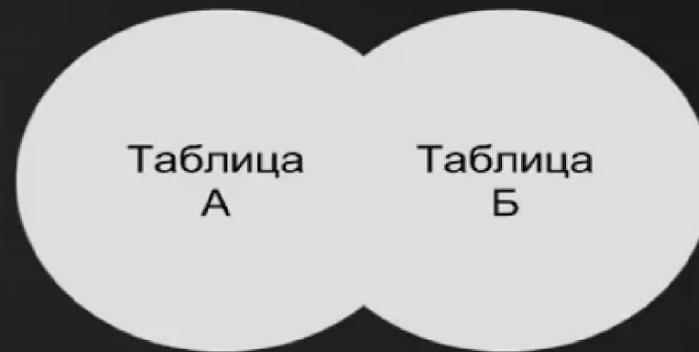
<i>movie_id</i>	<i>genre_id</i>
-----------------	-----------------

Таблица **genres**

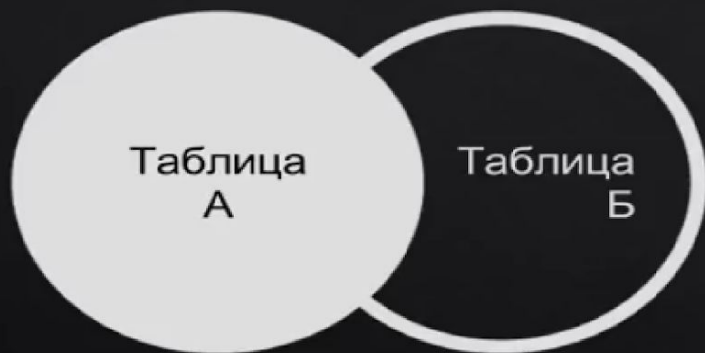
<i>id</i>	<i>genre</i>
1	Драма
2	Детектив
3	Триллер
4	Фантастика



[INNER] JOIN



FULL [OUTER] JOIN



LEFT [OUTER] JOIN



RIGHT [OUTER] JOIN