## **https://github.com/enhorse/java-interview/blob/master/db.md#%D0%A7%D1%82%D0%BE-%D1%82%D0%B0%D0%BA%D0%BE%D0%B5-%D0%B1%D0%B0%D0%B7%D0%B0-%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85**

## **Что такое «база данных»?**

****База данных**** — организованный и адаптированный для обработки вычислительной системой набор информации.

[к оглавлению](https://github.com/enhorse/java-interview/blob/master/db.md" \l "%D0%91%D0%B0%D0%B7%D1%8B-%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)

## **Что такое «система управления базами данных»?**

****Система управления базами данных (СУБД)**** - набор средств общего или специального назначения, обеспечивающий создание, доступ к материалам и управление базой данных.

Основные функции СУБД:

* управление данными
* журнализация изменений данных
* резервное копирование и восстановление данных;
* поддержка языка определения данных и манипулирования ими.

## **Что такое «реляционная модель данных»?**

****Реляционная модель данных**** — это логическая модель данных и прикладная теория построения реляционных баз данных.

Реляционная модель данных включает в себя следующие компоненты:

* Структурный аспект — данные представляют собой набор отношений.
* Аспект целостности — отношения отвечают определенным условиям целостности: уровня домена (типа данных), уровня отношения и уровня базы данных.
* Аспект обработки (манипулирования) — поддержка операторов манипулирования отношениями (реляционная алгебра, реляционное исчисление).
* Нормальная форма - свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности и определённое как совокупность требований, которым должно удовлетворять отношение.

## **Дайте определение терминам «простой», «составной» (composite), «потенциальный» (candidate) и «альтернативный» (alternate) ключ.**

****Простой ключ**** состоит из одного атрибута (поля). ****Составной**** - из двух и более.

****Потенциальный ключ**** - простой или составной ключ, который уникально идентифицирует каждую запись набора данных. При этом потенциальный ключ должен обладать критерием неизбыточности: при удалении любого из полей набор полей перестает уникально идентифицировать запись.

Из множества всех потенциальных ключей набора данных выбирают первичный ключ, все остальные ключи называют ****альтернативными****.

## **Что такое «первичный ключ» (primary key)? Каковы критерии его выбора?**

****Первичный ключ (primary key)**** в реляционной модели данных один из потенциальных ключей отношения, выбранный в качестве основного ключа (ключа по умолчанию).

Если в отношении имеется единственный потенциальный ключ, он является и первичным ключом. Если потенциальных ключей несколько, один из них выбирается в качестве первичного, а другие называют «альтернативными».

В качестве первичного обычно выбирается тот из потенциальных ключей, который наиболее удобен. Поэтому в качестве первичного ключа, как правило, выбирают тот, который имеет наименьший размер (физического хранения) и/или включает наименьшее количество атрибутов. Другой критерий выбора первичного ключа — сохранение его уникальности со временем. Поэтому в качестве первичного ключа стараются выбирать такой потенциальный ключ, который с наибольшей вероятностью никогда не утратит уникальность.

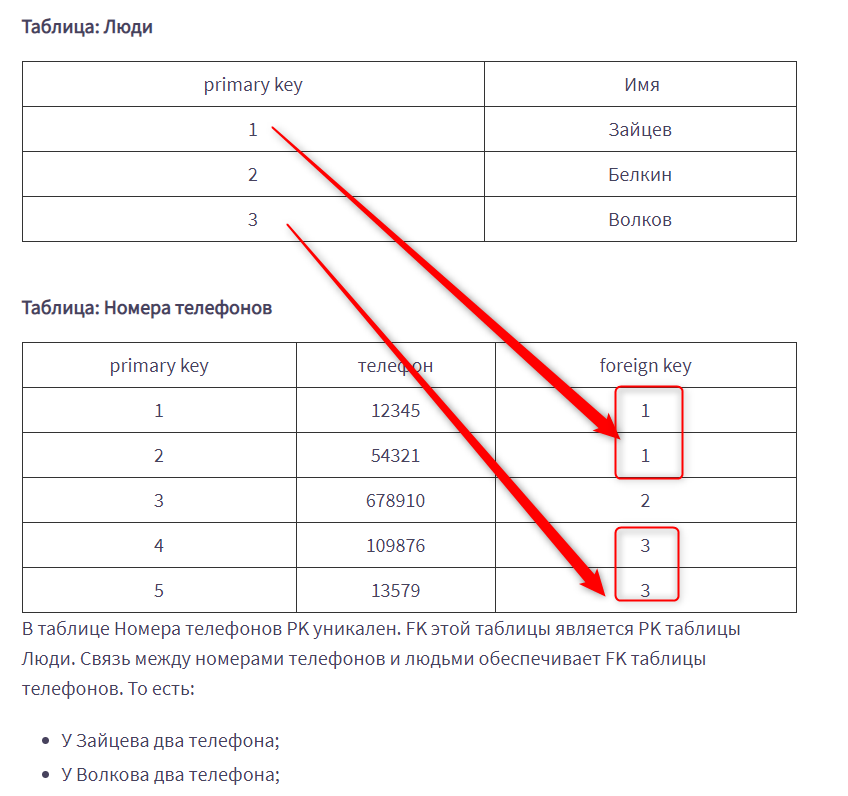
* Primary key не позволяет создавать одинаковых записей (строк) в таблице;
* PK обеспечивают логическую связь между таблицами одной базы данных

Первичный ключ – это одно или несколько полей (столбцов), комбинация значений которых однозначно определяет каждую запись в таблице. Первичный ключ не допускает значений ****Null**** и всегда должен иметь уникальный индекс. Первичный ключ используется для связывания таблицы с внешними ключами в других таблицах.

Внешний (вторичный) ключ - это одно или несколько полей (столбцов) в таблице, содержащих ссылку на поле или поля первичного ключа в другой таблице. Внешний ключ определяет способ объединения таблиц.

## **Что такое «внешний ключ» (foreign key)?**

****Внешний ключ (foreign key)**** — подмножество атрибутов некоторого отношения A, значения которых должны совпадать со значениями некоторого потенциального ключа некоторого отношения B.



[к оглавлению](https://github.com/enhorse/java-interview/blob/master/db.md" \l "%D0%91%D0%B0%D0%B7%D1%8B-%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)

## **Что такое «нормализация»?**

Нормализация - это процесс преобразования отношений базы данных к виду, отвечающему нормальным формам (пошаговый, обратимый процесс замены исходной схемы другой схемой, в которой наборы данных имеют более простую и логичную структуру).

Нормализация предназначена для приведения структуры базы данных к виду, обеспечивающему минимальную логическую избыточность, и не имеет целью уменьшение или увеличение производительности работы или же уменьшение или увеличение физического объёма базы данных. Конечной целью нормализации является уменьшение потенциальной противоречивости хранимой в базе данных информации.

[к оглавлению](https://github.com/enhorse/java-interview/blob/master/db.md" \l "%D0%91%D0%B0%D0%B7%D1%8B-%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)

## **Какие существуют нормальные формы?**

****Первая нормальная форма (1NF)**** - Отношение находится в 1NF, если значения всех его атрибутов атомарны (неделимы).

****Вторая нормальная форма (2NF)**** - Отношение находится в 2NF, если оно находится в 1NF, и при этом все неключевые атрибуты зависят только от ключа целиком, а не от какой-то его части.

****Третья нормальная форма (3NF)**** - Отношение находится в 3NF, если оно находится в 2NF и все неключевые атрибуты не зависят друг от друга.

****Четвёртая нормальная форма (4NF)**** - Отношение находится в 4NF , если оно находится в 3NF и если в нем не содержатся независимые группы атрибутов, между которыми существует отношение «многие-ко-многим».

****Пятая нормальная форма (5NF)**** - Отношение находится в 5NF, когда каждая нетривиальная зависимость соединения в ней определяется потенциальным ключом (ключами) этого отношения.

****Шестая нормальная форма (6NF)**** - Отношение находится в 6NF, когда она удовлетворяет всем нетривиальным зависимостям соединения, т.е. когда она неприводима, то есть не может быть подвергнута дальнейшей декомпозиции без потерь. Каждая переменная отношения, которая находится в 6NF, также находится и в 5NF. Введена как обобщение пятой нормальной формы для хронологической базы данных.

****Нормальная форма Бойса-Кодда, усиленная 3 нормальная форма (BCNF)**** - Отношение находится в BCNF, когда каждая её нетривиальная и неприводимая слева функциональная зависимость имеет в качестве своего детерминанта некоторый потенциальный ключ.

****Доменно-ключевая нормальная форма (DKNF)**** - Отношение находится в DKNF, когда каждое наложенное на неё ограничение является логическим следствием ограничений доменов и ограничений ключей, наложенных на данное отношение.

[к оглавлению](https://github.com/enhorse/java-interview/blob/master/db.md" \l "%D0%91%D0%B0%D0%B7%D1%8B-%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)

## **Что такое «денормализация»? Для чего она применяется?**

****Денормализация базы данных**** — это процесс осознанного приведения базы данных к виду, в котором она не будет соответствовать правилам нормализации. Обычно это необходимо для повышения производительности и скорости извлечения данных, за счет увеличения избыточности данных.

[к оглавлению](https://github.com/enhorse/java-interview/blob/master/db.md" \l "%D0%91%D0%B0%D0%B7%D1%8B-%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)

## **Какие существуют типы связей в базе данных? Приведите примеры.**

* ****Один к одному**** - любому значению атрибута А соответствует только одно значение атрибута В, и наоборот.

Каждый университет гарантированно имеет 1-го ректора: 1 университет → 1 ректор.

* ****Один ко многим**** - любому значению атрибута А соответствует 0, 1 или несколько значений атрибута В.

В каждом университете есть несколько факультетов: 1 университет → много факультетов.

* ****Многие ко многим**** - любому значению атрибута А соответствует 0, 1 или несколько значений атрибута В, и любому значению атрибута В соответствует 0, 1 или несколько значение атрибута А.

1 профессор может преподавать на нескольких факультетах, в то же время на 1-ом факультете может преподавать несколько профессоров: Несколько профессоров ↔ Несколько факультетов.