**Типы отношений**

Один к одному (One-to-One): Каждая запись в одной таблице соответствует одной и только одной записи в другой таблице, и наоборот.

Один ко многим (One-to-Many): Каждая запись в одной таблице может иметь несколько соответствующих записей в другой таблице, но каждая запись во второй таблице соответствует только одной записи в первой таблице.

Многие к одному (Many-to-One): Обратное отношение к “Один ко многим”. Каждая запись в одной таблице имеет только одну соответствующую запись в другой таблице, но каждая запись во второй таблице может иметь несколько связанных записей в первой таблице.

Многие ко многим (Many-to-Many): Каждая запись в одной таблице может соответствовать нескольким записям в другой таблице, и наоборот. Для реализации таких отношений требуется использование дополнительной таблицы – связи, для такой таблицы primary key будет выражен двумя этими атрибутами, которые и выражают это отношение, так как их комбинации всегда будут уникальны.

Самоотносящее отношение (Self-Referencing): Таблица может иметь отношение к самой себе.

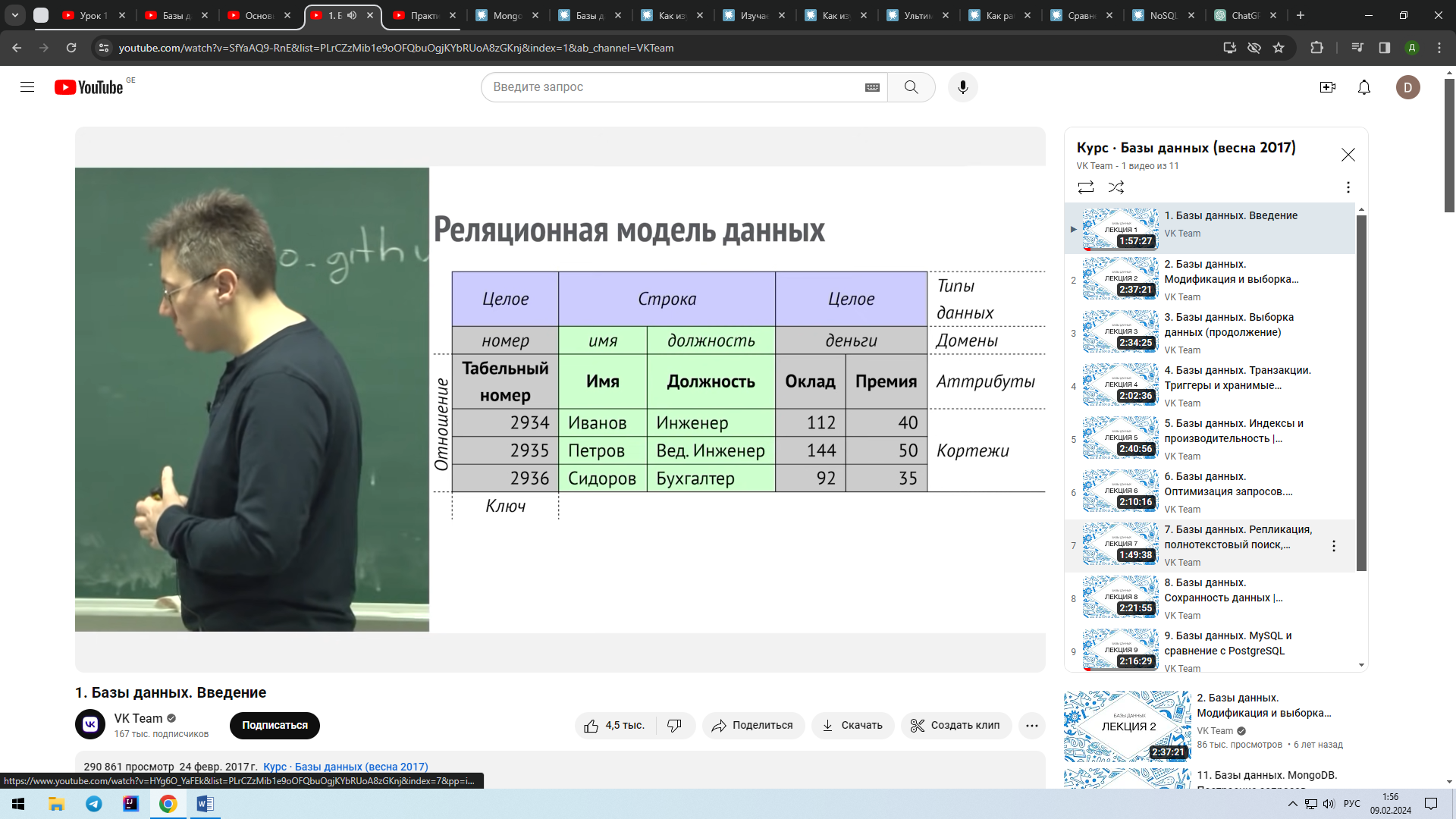
Любое отношение может быть представлено в виде отношения один к одному посредством создания промежуточной таблицы.

**Constraints** – правило, установленное для данных в таблице, с целью обеспечения их корректности и соответствия с определенными условиями.

*Not null, Uniqe, Check* (Задает диапазон значений), *Default* (Добавляет значение по умолчанию, в случае отсутствия данных), *Primary Key, Foreign Key (ограничение ссылочной целостности)*.

**Подробнее про реляционные СУБД**

*Реляционная модель данных:*



*Атрибут (столбец) –* Имеет Тип данных, Домен, Имя

*Домен –* некое условие для типа данных (ограничение)

*Строка/кортеж* – одна запись в таблице, т.е. набор значений каждого атрибута для определенного ключа

*Отношение -* таблица

*Сущность* – Объект предметной области, который может быть описан

Результирующий набор – результат запроса на языке SQL

**Primary key –** ограничениена аргумент или набор аргументов, по которому (-ым) можно гарантированно идентифицировать строку. Это достигается за счет свойства уникальности и наличия значения отличного от null.

В качестве первичного ключа может служить либо естественный ключ (т.е. уже существующий атрибут). Минусы – проблемы, в случае его изменения, так как нужно будет поменять его во всех таблицах, может занимать больше памяти. Либо суррогатный – к отношению добавляется новый атрибут никак его не характеризующий, например, порядковый номер или id, в качестве значения используется число с автоинкрементом.

**Foreign key –** ограничение, заданное аргументом или набором аргументов, обеспечивающее связь между таблицами посредством ссылок, значения которых соответствуют значениям первичного ключа в другой таблице. Так же позволяют определить каскадные операции, управлять целостностью данных и устанавливать правила отношений между таблицами:

Запрет на удаление исходной строки, если есть живые ссылки.

Каскадное обновление/удаление значений – обновление/удаление данных в исходной таблице будет применено ко всем строкам в дочерней.

Запрет на вставку данных – невозможно вставить данные в дочернюю таблицу, если такой записи нет в исходной.

Аргументы могут соответствовать критериям для первичного или внешнего ключа, но они не дают гарантий, что их свойства будет исполняться при любых условиях. Ограничения Primary key и Foreign key гарантируют это.