# **--- Database ---**

**Пользователь -> СУБД -> База Данных -> Жесткий Диск (Облако)**

**Можно делать с базами -> Create / Read / Update / Delete**

**Реляционные (Realitons - Отношения) Базы Данных** **-** берет информацию от пользователя -> структурирует -> записывает в хранилище. Упорядоченый набор данных, которые сохраняться по типу таблици и размещаються по строкам и столбцам. Одни таблицы могут ссылатся на другие таблици.

**Не Реляционые NoSQL Базы Данных** - упорядоченый набор данных, который может храниться по типу ключ - значение / документация / графы. Ето MongoDB - Cassandra

**СУБД (Система Управления Базами Данных)** - програма , которая отдает команды базе данных. Базы данных понимають чего хочет пользователь, а пользователь понимает что ответила база данных.Также отвечает за оптимизацию процеса извлечения данных из базы. Ето PostgreSQL - MySQL - Microsoft SQL Server

**SQL (Structured Query Language)**  - языки запросов для получения информации из базы данных.

**Отношения -** связи между несколькими таблицами

**Транзакции Базы Данных** - операция или операции над базой данных. Добавление , Чтение , Изменение , Удаление записей в базе

**Атрибут** - столбец в таблице

**Запись (Кортеж)** - строки в таблице

**Результирующий Набор** - результат запроса SQL из базы данных. Резульзат запроса всегда таблица

**Значение** - елемент таблици, который находиться на пересечении столбцов и строк

**Ключ** - столбец в таблице , который связывает разные таблици между собой

**Первичный Ключ (Primary Key)** - столбец, который индетифицирует каждую запись в таблице. Только уникальные значения

**Потенциальный** **Ключ** стобец, который который индетифицырует каждую запись в таблице , но с названием которое отличается от первичного ключа

**Внешний Ключ (Foreign Key)** - столбец, который ссылается на записи в других таблицах.

**SQL Операторы** - ключевые слова, символы которые используються для конкретных операций над базами данных

**Схема SQL (Schema SQL)** - опередяет стурктура базы данных. Указывает на размеры таблиц, значения строк и столбоц, также связь между таблицами

**Запрос SQL (Query SQL)** - запрос данных из базы данных (из таблицы). Запрос стоит на трех китах -> **Что делаем ? - Где Делаем? - Как делаем ?**

**Выборка -** когда пользователь берет данные из одной или нескольких таблиц одновременно

**Отношение Один ко Одному -** когда каждая запись в каждой таблице используеться только один раз. Создаеться третья таблица для етого

**Отношение Один ко Многим -** когда каждая запись в первой таблице используеться один раз , а записи из второй таблици - id могут повторяються. Вторая табличка какбы пренадлежит первой(самая популярная)

**Отношение Многие ко Многим** - каждая запись в каждой таблице может повторяться. Такое отношение только возможно в третьей(другой) таблице. В первой колонке третье таблици числа удит от меньше к большему, а во второй колонке как попало

# **Соединения**

**Left Таблица -** таблица к которой будем добавлять таблицу

**Right Таблица** - таблица, которую будем добавлять

**Таблици соединяться по внешним ключам**

**INNER JOIN (Внутрешнее Соединение) -** к левой таблице добавляються значения по ключу из правой таблици. В левой таблице все значения, в правой только найденные

**LEFT JOIN (Левое Соединение)** - беруться все значения из текущей (левой) таблици и значения, которые будут найденые из правой таблици, если найденыне по ключу. Пустые значения заполняются NULL

**RIGHT JOIN (Правое Соединение)** - беруться все данные из правой таблички и данные которые будут найдение из левой таблички, если найденные по ключу. Пустые значения заполняются NULL

**FULL JOIN (Полное Соединение) -** беруться абсолютно все данные из двох таблиц, даже если нет ключей. Пустые значения заполняются NULL

**Cross Join (Декартовое Соеднение)** - соединение основаное на соединении каждого ключа из левой таблици со всеми ключами из правой таблици