* **SSH —** это сетевой протокол. Он помогает создавать зашифрованное соединение между двумя компьютерами и безопасно передавать данные.
* Через SSH подключаются к удалённому серверу. Для этого нужно

использовать SSH-клиент — программу, которая позволяет устанавливать соединение с SSH-серверами. Windows SSH-клиент устанавливают отдельно. Например, вместе с **Cygwin**.

* В проекте использовалось СУБД – **PostgreSQL**
* **Нереляционная база данных —** это это база данных, в которой в отличие от большинства традиционных систем баз данных не используется табличная схема строк и столбцов. Например, данные могут храниться как простые пары "ключ — значение", документы JSON.
* **PostgreSQL -** это реляционная база данных с открытым кодом**.**
* **База данных —** это хранилище структурированной информации.
* **Сущности —** группы объектов с общими характеристиками.
* **Реляционные базы данных —** базы данных, в которых сущности — это

таблицы, а объекты — строки таблиц.

* **СУБД (система управления базами данных) —** это комплекс программ,который позволяет создать базу данных, наполнить её новыми таблицами,отобразить содержимое, редактировать существующие таблицы.
* **Таблица —** это совокупность строк и столбцов.
* **Поле —** столбец таблицы, который обозначает характеристику объекта и имеет уникальное имя и характерный тип данных.
* **Запись —** строка таблицы, которая содержит информацию об одном

объекте.

* **Ячейка —** место пересечения строки и столбца.
* **Первичный ключ —** поле или группа полей, которое применяют, чтобы

однозначно определить запись. Все значения первичного ключа уникальны.

* **SQL** — язык программирования; предназначен, чтобы управлять данными в реляционной базе.
* **Запрос —** это требование, которое сформулировали в соответствии с

синтаксисом SQL. В нём указывают, какие данные выбрать и как именно их обработать.

* **Операторы:**

**SELECT –** Выбрать столбцы из таблицы

**FROM –** Название таблицы (из какой таблицы выбрать)

**WHERE** - Определение условия, по которому отбираешь строки

* **Логические операторы :**

**AND -** позволяет выбрать строки, которые соответствуют всем условиям.

**OR -** позволяет выбрать строки, которые подходят любому условию в запросе.

**NOT-** позволяет выбрать строки, которые не соответствуют определённому условию**.**

* **Агрегирующие функции -** посчитать общее количество строк в результате запроса, сумму, среднее значение, максимум и минимум для определённого поля в ответе.

**COUNT() -** возвращает количество строк в таблице

**DISTINCT() -** возвращает уникальные строки в таблице

**SUM(column) -** возвращает сумму по столбцу column

**AVG (column) -** возвращает среднее значение по столбцу column.

**MIN() и MAX() -** Минимальное и максимальное значения вычисляют функциями

* **Конструкция** **CAST –** преобразование тип данных столбца в SQL-запросе

**Типы данных**

* **Числовые типы данных**

integer — целочисленный тип.

real — число с плавающей точкой.

* **Строковые типы данных**

**Пример: 'Практикум'** — значение строкового типа, в SQL-запросе его заключают в одинарные кавычки.

**varchar(n)** — строка переменной длины, где n — ограничение. Этот тип данных варьирует длину хранимой строки: в поле можно занести любую строку короче, чем n символов.

**T**

**ext** — строка любой длины.

* **Дата и время**

Любые значения даты или времени заключают в одинарные кавычки.

**timestamp** — дата и время. В формате timestamp чаще всего хранят события, для которых недостаточно указать только дату. Например, в логах указывают дату и точное время произошедшего события. Требуется точность до секунды.

Пример: '1990-01-01 00:00:00'

**date — дата**. Имеет точность до дня без поддержки часового пояса.

Пример: '1990-01-01'

* **Логический**

**boolean** — логический тип данных. В PostgreSQL есть три варианта значений:

**TRUE** — «истина»,

**FALSE** — «ложь»,

**NULL** — «неизвестно».

**Команда:**

* **GROUP BY -** применяют, когда данные нужно разделить на группы по значениям полей.
* **ORDER BY -** Чтобы сортировать данные по указанным полям

**Аргументы ORDER BY :**

**ASC -** сортирует данные в порядке возрастания.

**DESC-** сортирует данные по убыванию.

**LIMIT-** ограничивает количество строк в выводе.

**Менять содержимое таблиц в БД**

1. **INSERT** помогает добавить новую строку в таблицу.

**INSERT INTO** - указывает, в какую таблицу добавить данные.

**VALUES** - перечисляет данные, которые нужно добавить.

1. **Оператор UPDATE -** меняет текущую информацию в таблице

**Cинтаксис:**

**UPDATE -** указывает, в какой таблице изменить данные

**SET -** указывает столбец, в котором необходимо поменять данные.

**WHERE -** выбирает строки, опционально.

1. **Оператор DELETE -** удаляет записи из таблицы

**Cинтаксис:**

**DELETE FROM-** указывает, из какой таблицы удалить записи.

**WHERE -** выбирает строки, в которых нужно удалить данные.

**Дамп базы —** это файл с набором SQL-команд, который позволяет воссоздать базу данных и таблицы и восстановить информацию.

**Обьединение таблиц**

**Внутренний:**

**Оператор INNER JOIN -** формирует выборку только из тех данных, у которых выполнено условие присоединения.

**Внешний:**

**LEFT OUTER JOIN -** возвращаются все строки левой таблицы.

**RIGHT OUTER JOIN**- возвращаются все строки правой таблицы.

**FULL OUTER JOIN** - возвращаются строки и левой, и правой таблиц.