# **Вебинар 4**

## **Знакомство с СУБД**

### ***Данные, информация, знания***

**Данные** – структурированная совокупность символов, принадлежащих определенным множествам

**Информация** – данные, обработанные некоторым образом (программой или человеком)

**Знания** – данные, содержащие в себе свою интерпретацию или способ интерпретации

### ***БД, СУБД, их виды***

**Определение:**

**БД** - база данных. Содержит табличное представление данных.

Базы данных - термин база данных используется в самых разных областях, но мы будем считать базу данных набором сведений, хранящихся неким упорядоченным способом. Проще всего рассматривать базу данных как шкаф для хранения документов.

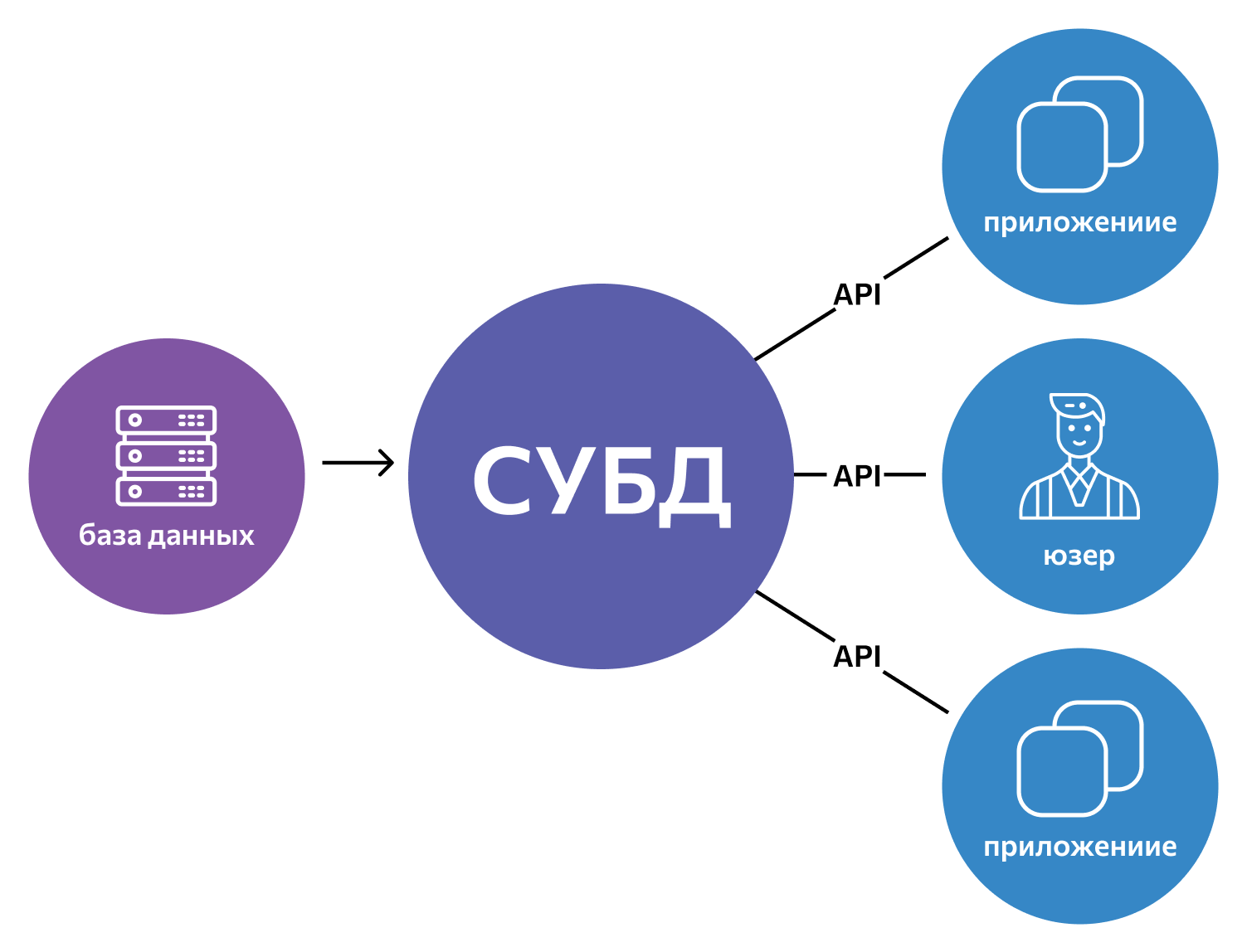
**Определение**:

База данных - Контейнер (обычно файл или группа файлов) для хранения упорядоченных данных.

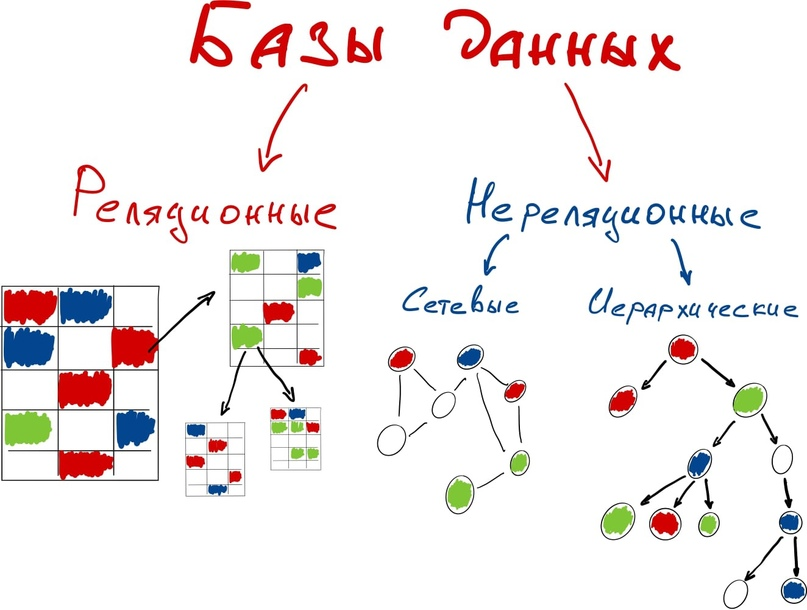
**Пример из реальной жизни:**

Шкаф — это просто физический объект для хранения данных, независимо от того, что это за данные и как они упорядочены.

**СУБД** - система управления базой данных. Представляет собой комплекс программ, которые позволяют управлять (манипулировать) данными.



### ***Какие базы данных бывают?***

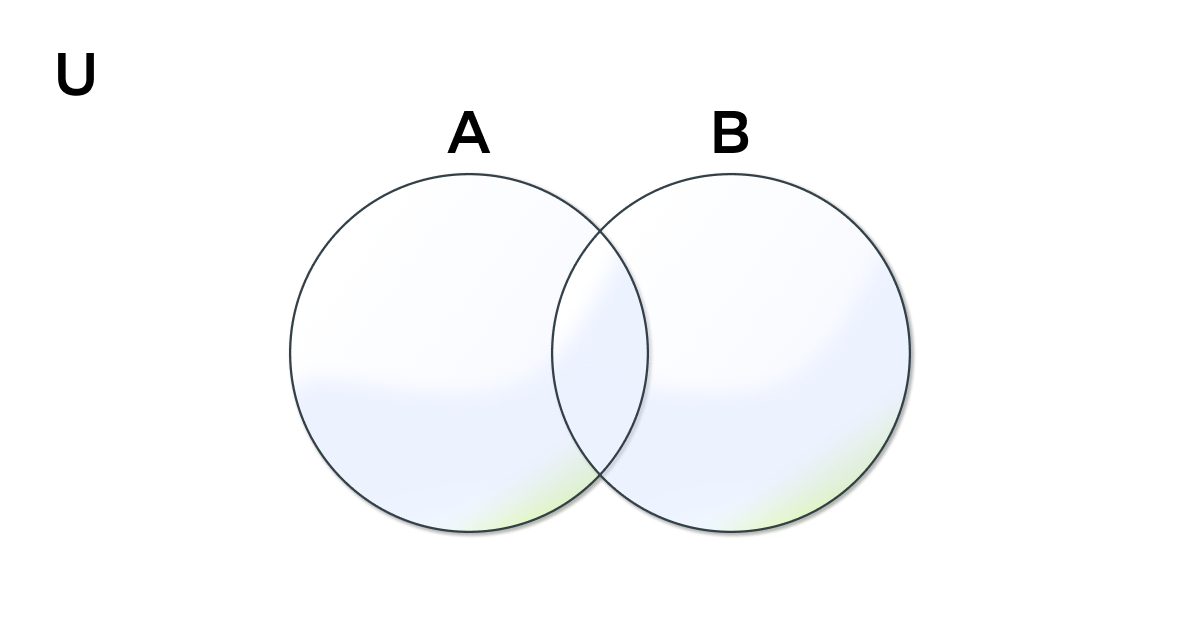


**Базы данных бывают:**

* Иерархическими. Данные представлены в виде древовидной структуры.
* Сетевыми.
* Функциональными.
* Объектными (объектно-ориентированными). Данные представлены в виде объектов с атрибутами, методами и классами.
* Объектно-реляционными. Реляционная БД с поддержкой ООП.
* Реляционными. Основаны на реляционной модели.

**Реляционные базы данных.** Данный тип БД является старейшим: здесь данные формируются в таблицы из строк и столбцов. В строках приводятся сведения об объектах (значения свойств), а в столбцах — сами свойства объектов (поля).

Суть данной модели БД заключается в том, что основанная на математической теории множеств она рассматривает таблицы как отдельные множества, объединенные по определенному признаку, как показано на рисунке



Для удобства представления и хранения система хранения данных в рамках реляционной модели обычно представляет собой совокупность таблиц, взаимосвязанных между собой определенным образом.

Таблица представляет собой плоскую двумерную сетку, которая содержит определенный тип или типы структурированных данных, например фильмы, группы крови, маршруты движения и т. д.

Если разбирать дальнейшее устройство любой таблицы, то она состоит из столбцов и строк.

Столбец таблицы представляет собой вертикальную область, выделенное свойство всех типологий, которые содержатся обычно в строках таблицы.

Строка таблицы представляет собой горизонтальную область, которая уникализирует типологию. Например, в строках таблицы могут содержаться фамилии, марки автомобилей, названия улиц и т. д.

Структурные элементы таблицы, из которых она состоит, взаимодействуют между собой, а также с другими структурными элементами других таблиц.

**Принципы работы реляционных БД:**

* Данные представлены в виде отношений (таблиц) между строками и столбцами.
* Всё наполнение БД представлено одним и только одним способом.
* В пересечение столбца и строки можно внести только одно значение.
* Все операции выполняются только над целыми отношениями (таблицами), результатом этих операций является отношение (таблица).

### ***4. Термины***

**Отношение** - это связанные столбцы и строки. Проще говоря, таблица.

**Строка** - отдельная запись в таблице.

**Кортеж** - запись в таблице (полная строка).

**Мощность** - количество кортежей в таблице. Может иметь значение от 0 до бесконечности.

**Столбец** - одиночное поле таблицы. Все таблицы состоят из одного или нескольких столбцов.

**Атрибут** - столбец или поле в таблице.

**Размерность** - количество столбцов в таблице. Может иметь значение от 1 до бесконечности.

**Ключевое поле** - поле, которое содержит уникальное значение, которое не повторяется в пределах таблицы. Используется для связи таблиц в БД.

**Ключевое слово** - зарезервированное слово. Никогда не называйте таблицу или столбец таким словом.

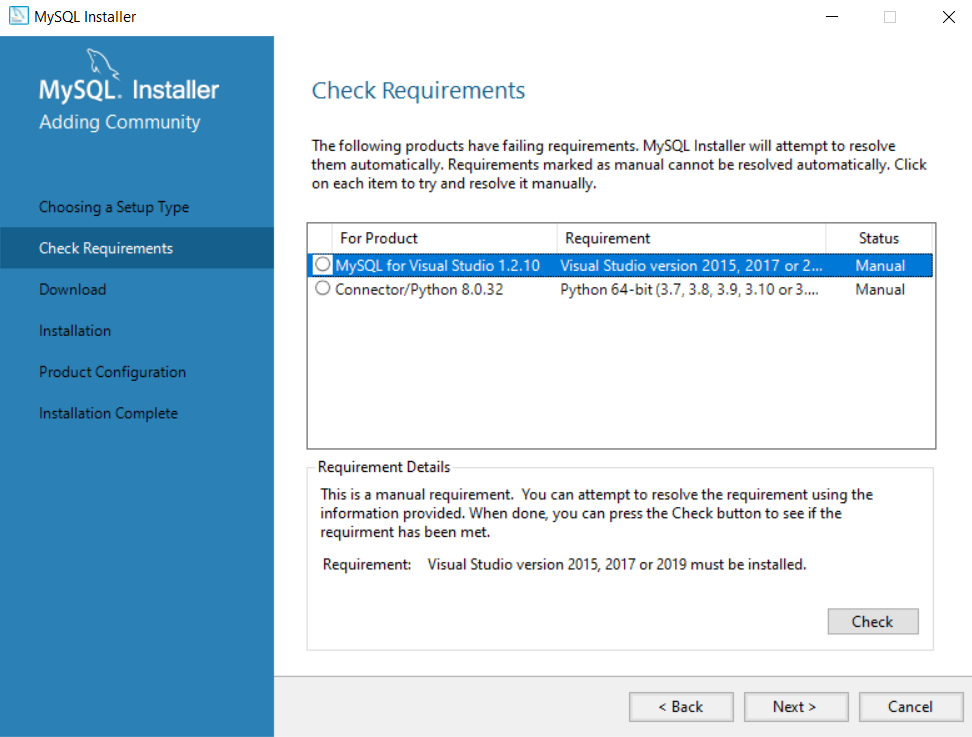
**Тип данных** - Тип разрешенных для хранения данных. Каждому столбцу таблицы присваивается тип, который разрешает хранить в нем только определенную информацию.

**SQL (Structured Query Language)** — это язык структурированных запросов, который был специально разработан для взаимодействия с базами данных.

### ***Установка MySQL:***

### Шаг 1: Перейти по ссылке, которую вам направили в телеграмм-канал, и выбрать пункт “Нет, спасибо, просто начните загрузку”.

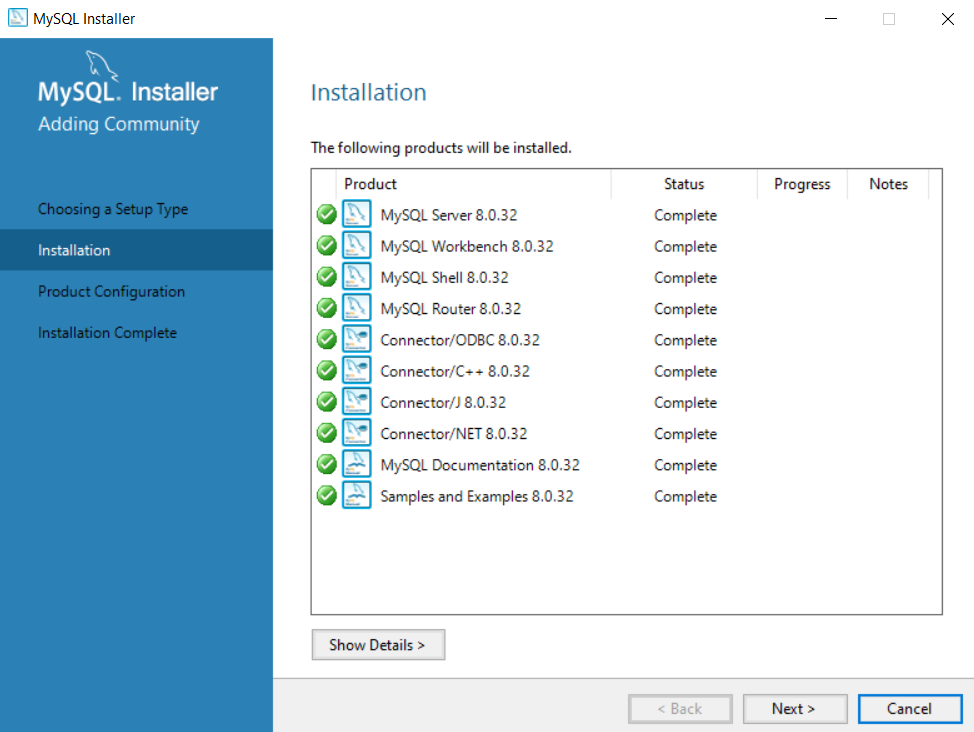
### Шаг 2: У вас начнется загрузка файла “mysql-installer-web-community-8.0.32.0.msi”. После его загрузки, нажмите на файл, и перейдите к дальнейшей установке. Шаг 3: В появившемся окне выберите вариант “Full”

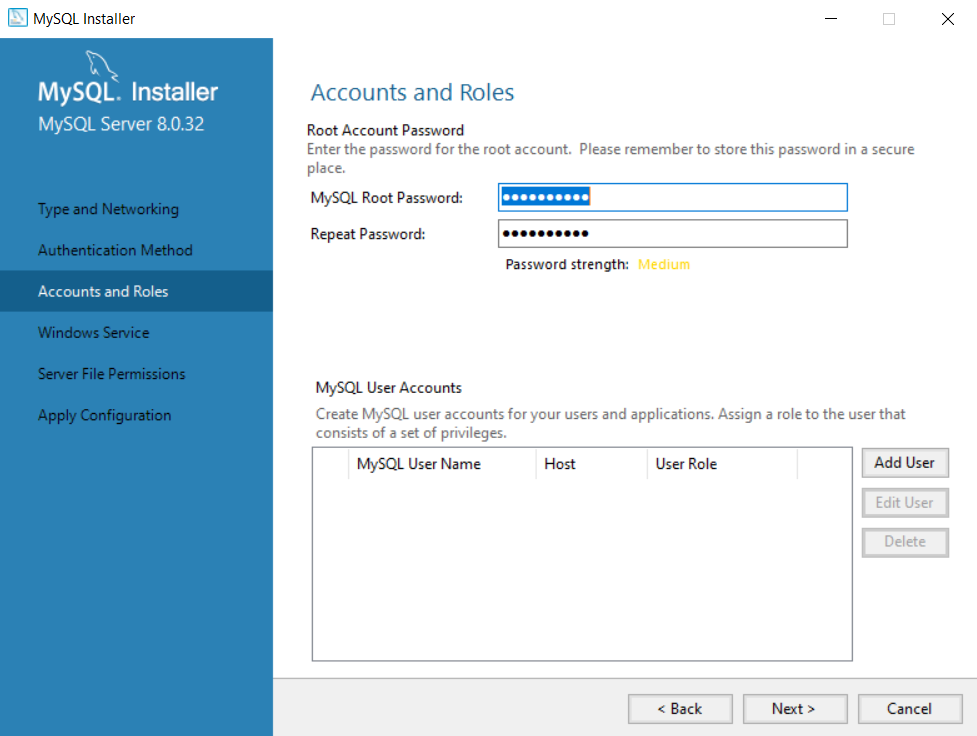
Шаг 4:  
Выберите пункт “MySQL for Visual Studio”  


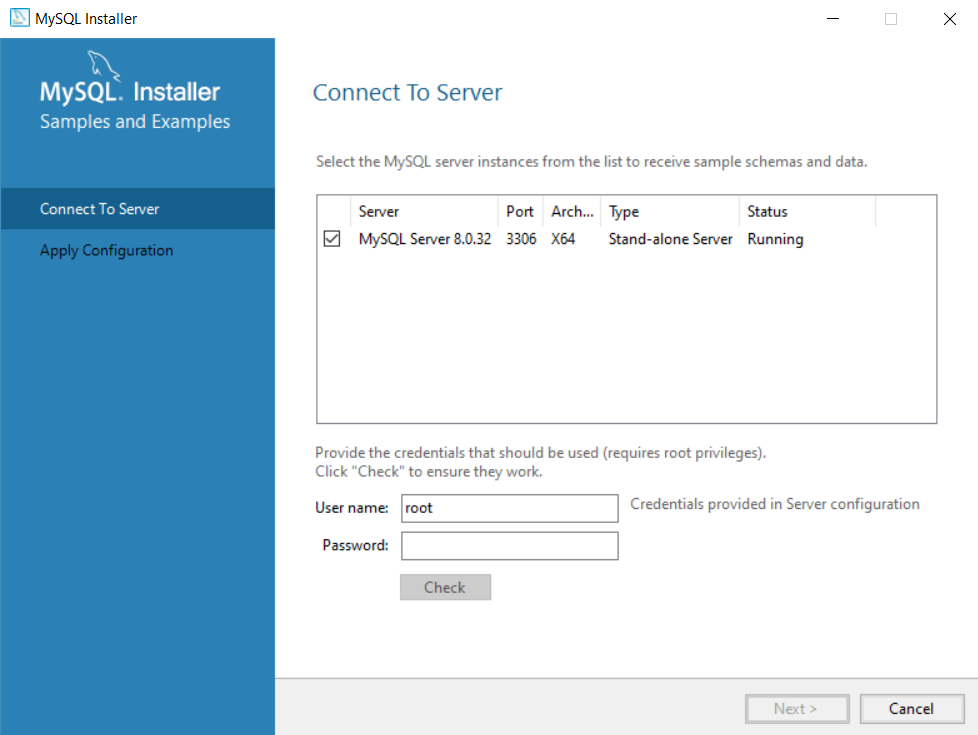
### Шаг 5: У вас будет представлен перечень компонентов, доступных для загрузки. Чтобы начать загрузку, нажмите “Execute”.

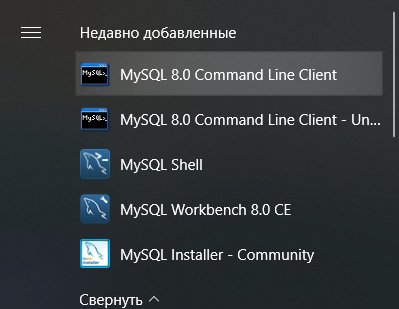
### Шаг 6: После загрузки, нажимайте на кнопку “next”. Если во время установки вы видите окна, которые не представлены в данной методичке в шагах установки - значит все необходимое выбрано автоматически, и вам нужно нажать на “next”.

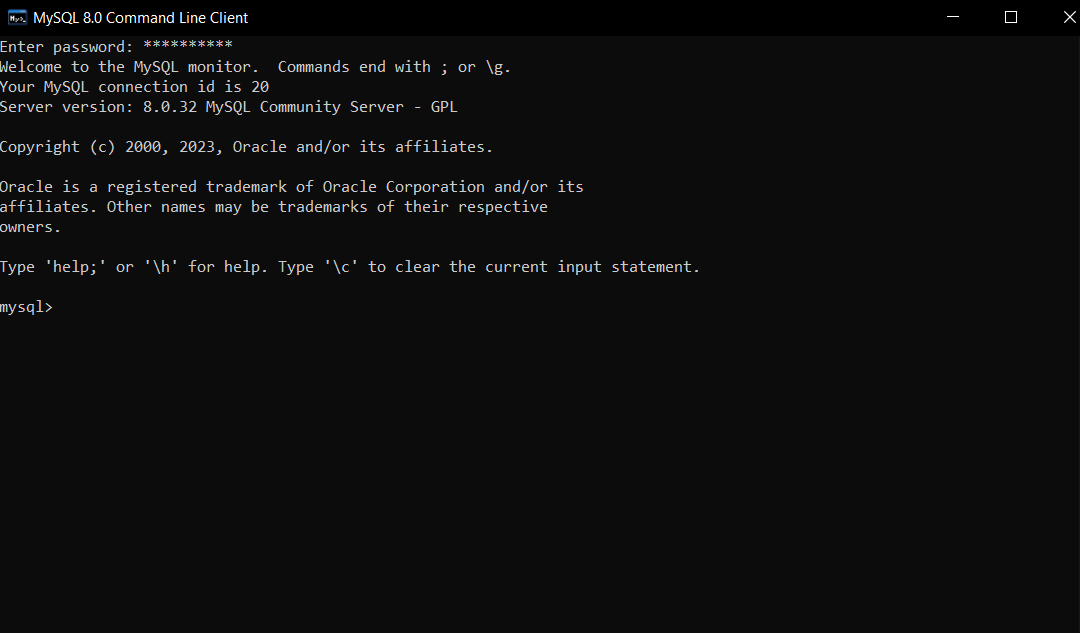
Шаг 7:

“Next”  


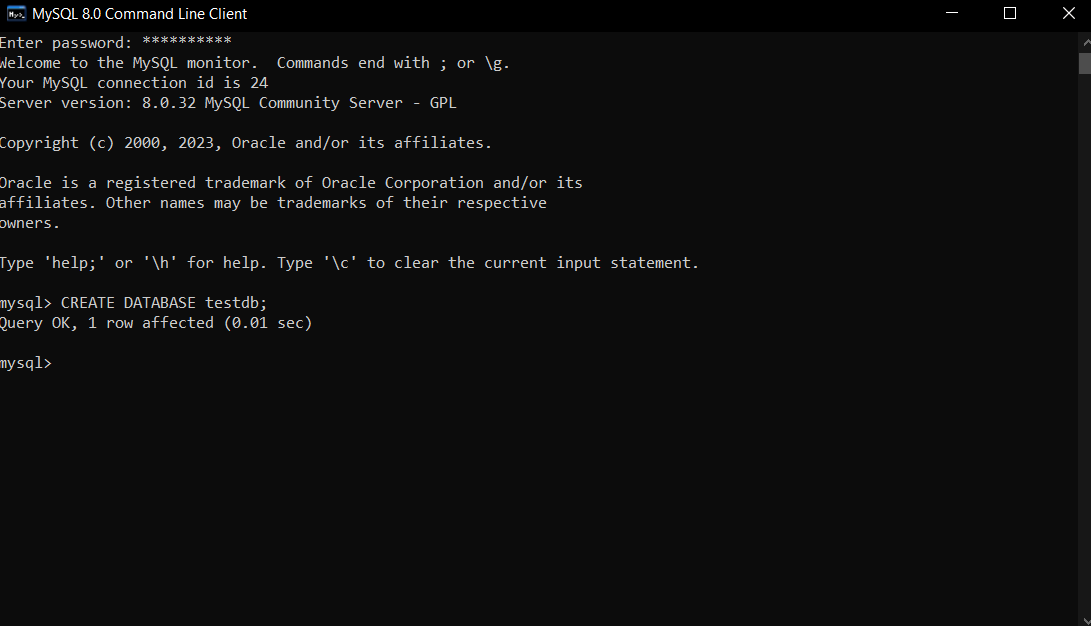
Шаг 8:  
Вам нужно придумать пароль, который в дальнейшем потребуется ввести при запуске программы.  
!Запишите себе этот пароль, чтобы не забыть!  


Шаг 9:  
У вас появится такое окно, здесь вам нужно поставить галочку возле MySQL Server, а затем нажать на Cancel.  


Шаг 10:  
Продолжаете установку, нажимаете Finish. После этого у вас прошла установка.   
Для того, чтобы начать работать с MySQL, вам нужно зайти в пуск и выбрать MySQL 8.0 Command Line Client.  


Шаг 11:  
У вас откроется окно, в котором вы будете создавать свои базы данных. Вас попросят ввести пароль, который вы придумывали ранее.  
После ввода пароля у вас окно станет выглядеть как на приложенном ниже фото:  


Шаг 12:  
Теперь перейдем к настройке вашей самой первой БД.  
Повторите текст, написанный на картинке ниже:  
CREATE DATABASE testdb;

  
Вы создали тестовую БД.

### 

### ***Правила наименования полей и таблиц***

В SQL, также, как и в С++ есть ряд правил подбора имён:

* Регистр не имеет значения (Имя и имя - одно и то же)
* Имя может состоять из латинских букв, нижнего подчёркивания и цифр
* Имя должно начинаться с буквы
* Имя таблицы должно быть уникальным в пределах БД
* Имя поля должно быть уникальным в пределах отношения

### ***Типы данных***

В SQL существуют следующие типы данных:

* INT (INTEGER) может содержать целочисленное значение
* DECIMAL (NUMERIC) может содержать числа с плавающей точкой. В скобках указывается количество цифр и количество цифр после запятой. В качестве разделителя между дробной и целой частью выступает точка.
* DATE может содержать данные в формате гггг-мм-дд.
* VARCHAR может содержать до 255 символов. Значения указываются в кавычках, двойных или одинарных.

INT PRIMARY KEY AUTO\_INCREMENT - особая запись ключевого поля, которая позволяет автоматически проставлять уникальные целочисленные ключи с шагом 1. Работает не везде.

### ***Запросы:***

#### *Создание таблицы*

Ключевое слово CREATE TABLE -> далее указывается имя таблицы -> в круглых скобках указывают имена и типы полей.

Пример:

CREATE TABLE имя(

имя\_id INT PRIMARY KEY AUTO\_INCREMENT,

имя\_поля ТИП,

...

);

Обратите внимание, что в конце запроса ставится точка с запятой.

#### *Добавление записей*

Ключевое слово INSERT INTO (INTO можно опустить) -> далее указывается имя отношения -> в скобках указываются поля, в которые добавляем данные -> ключевое слово VALUES -> в скобках указываются значения этих полей (количество полей и значений должно совпадать, типы данных значений должны соответствовать столбцам, новые значения нельзя задавать в поля с описанием PRIMARY KEY AUTO\_INCREMENT)

Пример:

INSERT INTO таблица(поле1, поле2)

VALUES (значение1, значение2);

#### *Вывод таблицы на экран*

Ключевое слово SELECT -> звёздочка (собирает таблицу) -> ключевое слово FROM -> название таблицы

Пример:

SELECT \* FROM имя;

Можно выбрать конкретные столбцы. Для этого вместо звёздочки нужно указать названия столбцов:

SELECT имя\_столбца FROM имя;

В SQL можно добавлять столбцы при выводе таблицы на экран и задавать им имя. Чтобы это сделать нужно написать ключевое слово SELECT, далее написать формулу и добавить слово AS и название нового столбца:

SELECT имя\_столбца1, имя\_столбца2,

имя\_столбца1+имя\_столбца2 AS имя\_столбца3

FROM имя