CGI:

Unity

Blender

Cinema 4d

Modo

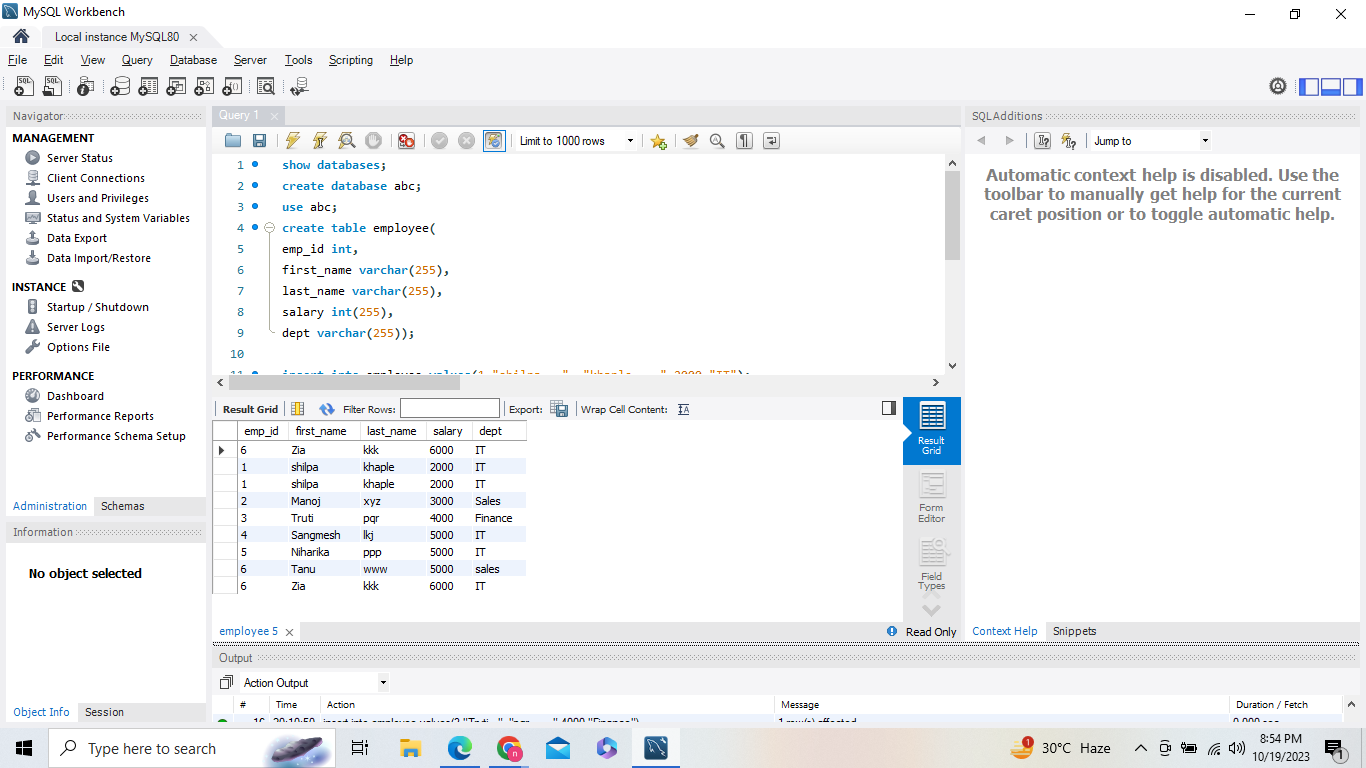
Light wave 3d

K short

Z brush

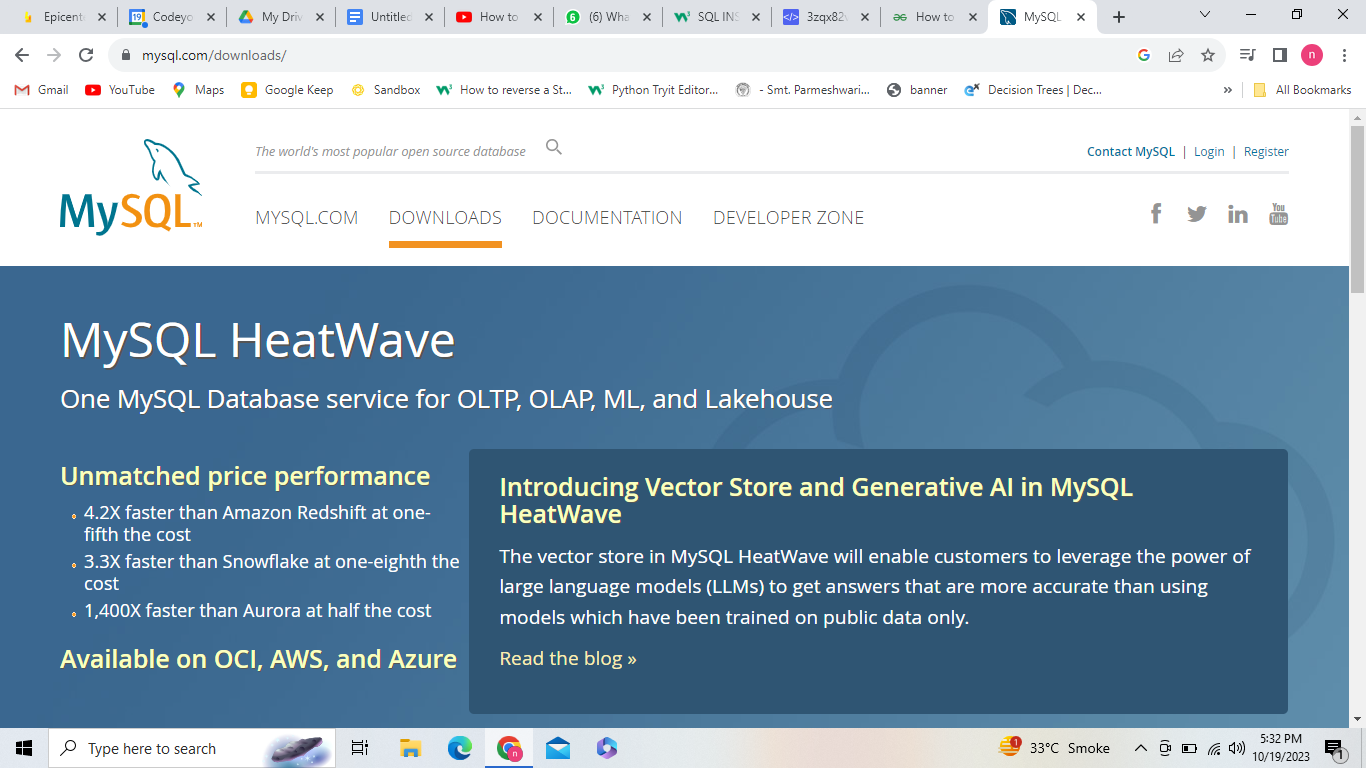
Adobe dimansentions

Important points after installation:before create table always create database like create database abc then write use abc

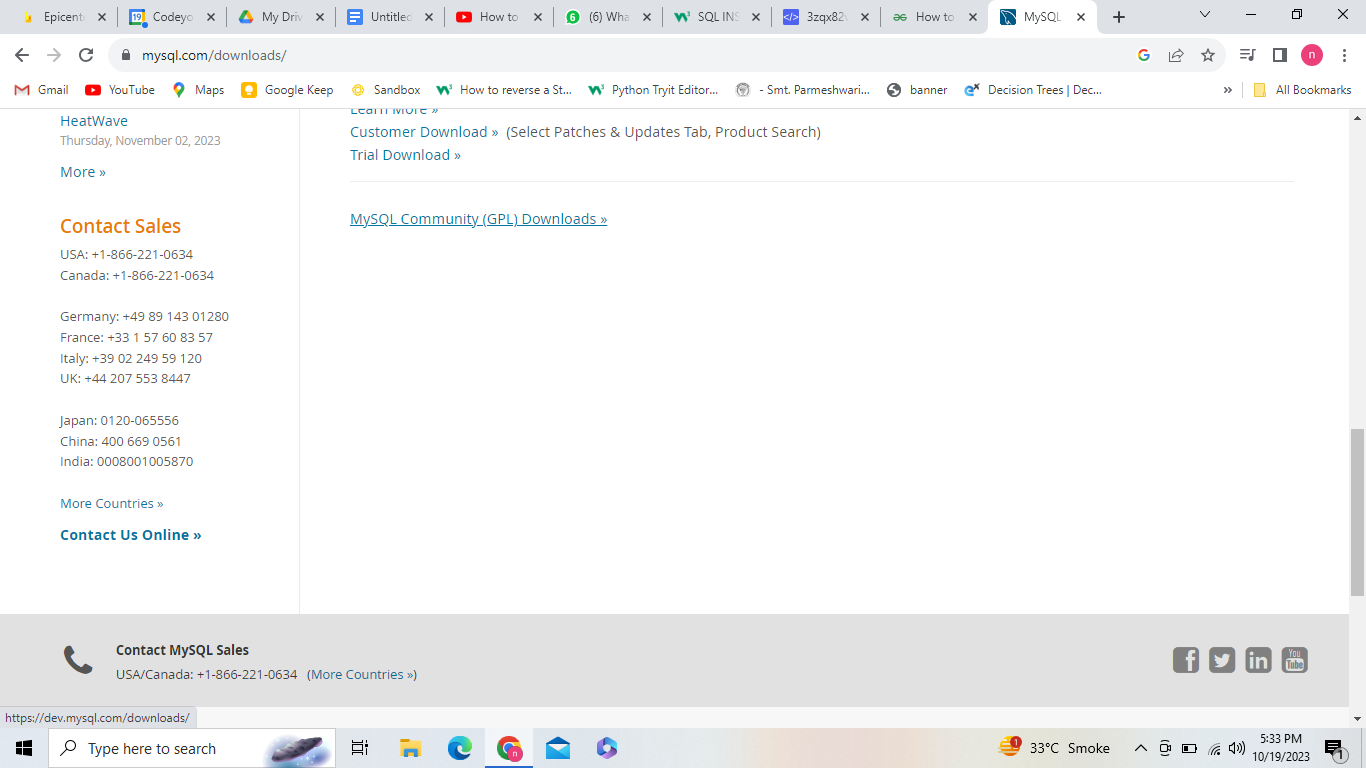


Sql:SQL is a standard language for storing, manipulating and retrieving data in databases.

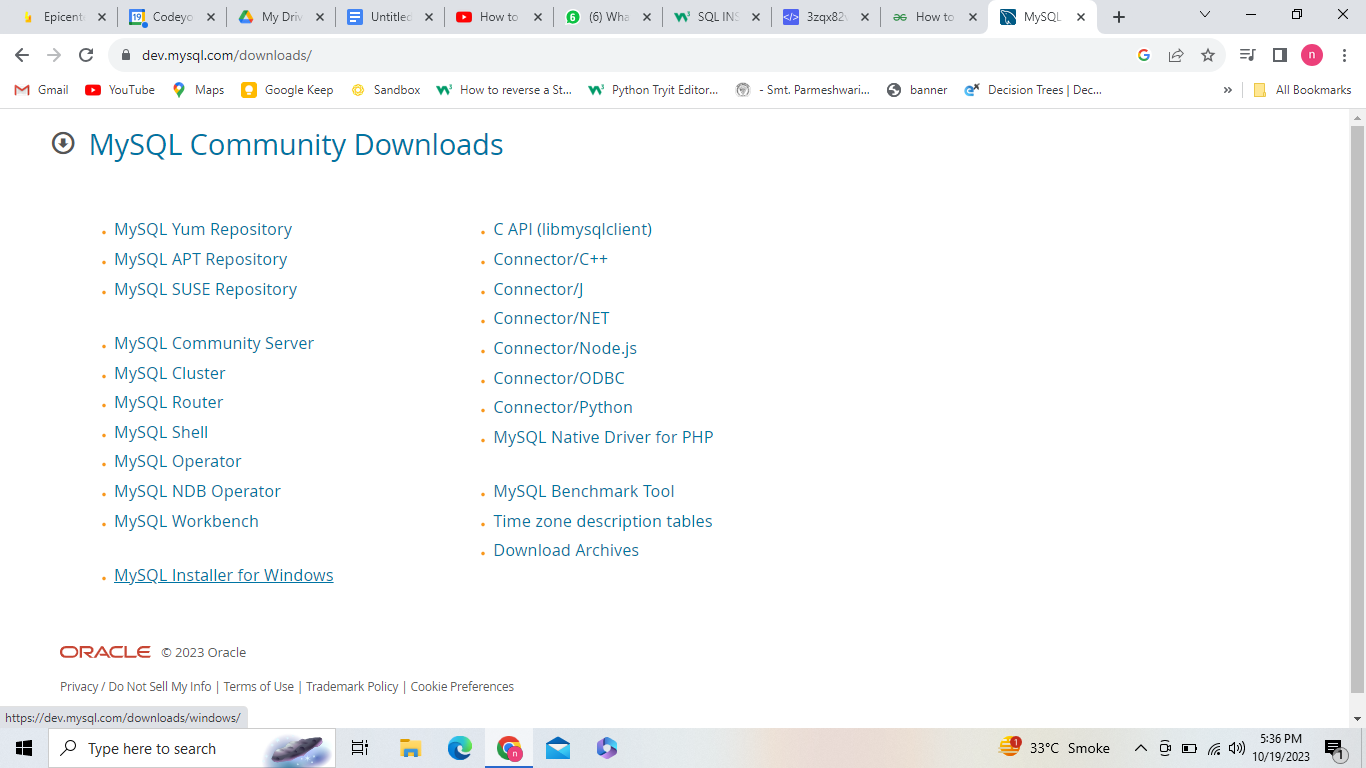
How to download MYSQl



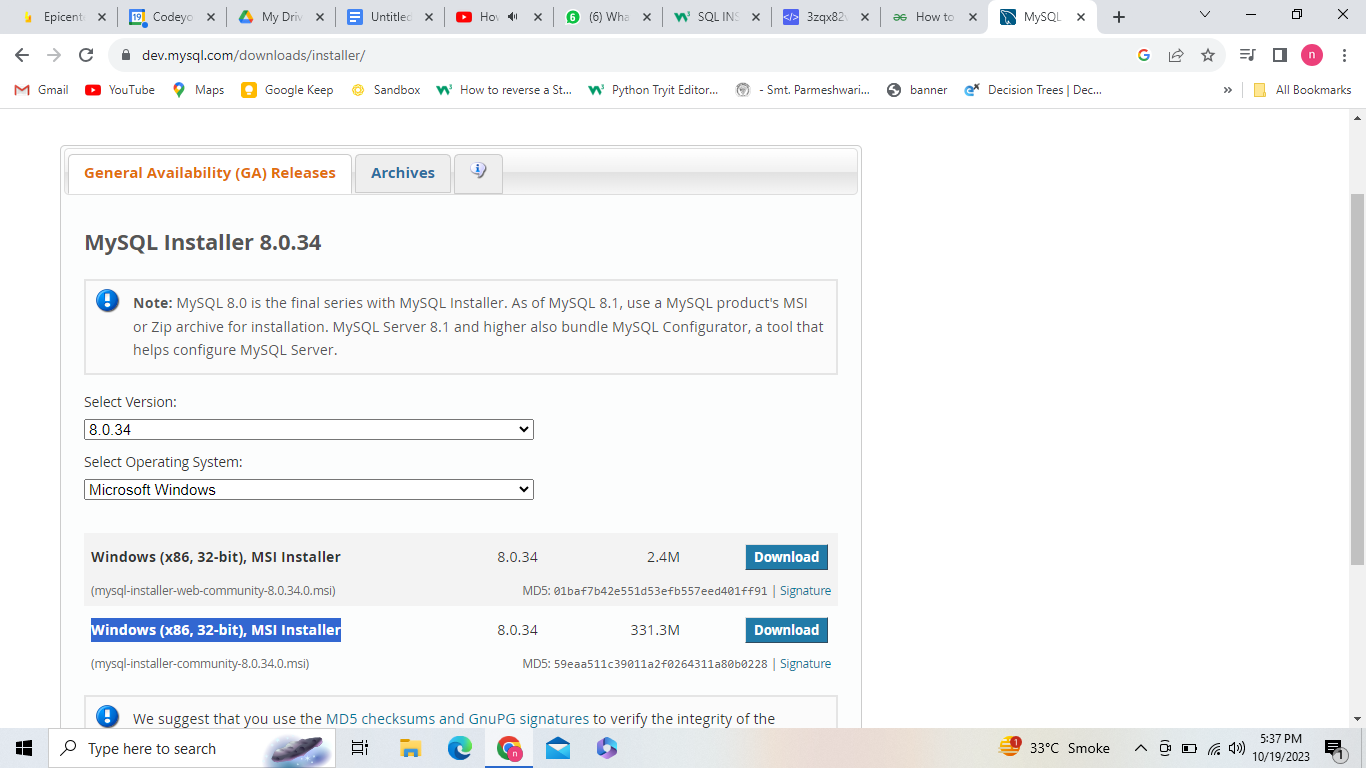
2)go down and click on : [MySQL Community (GPL) Downloads »](https://dev.mysql.com/downloads/)



3)click on to [MySQL Installer for Windows](https://dev.mysql.com/downloads/windows/)

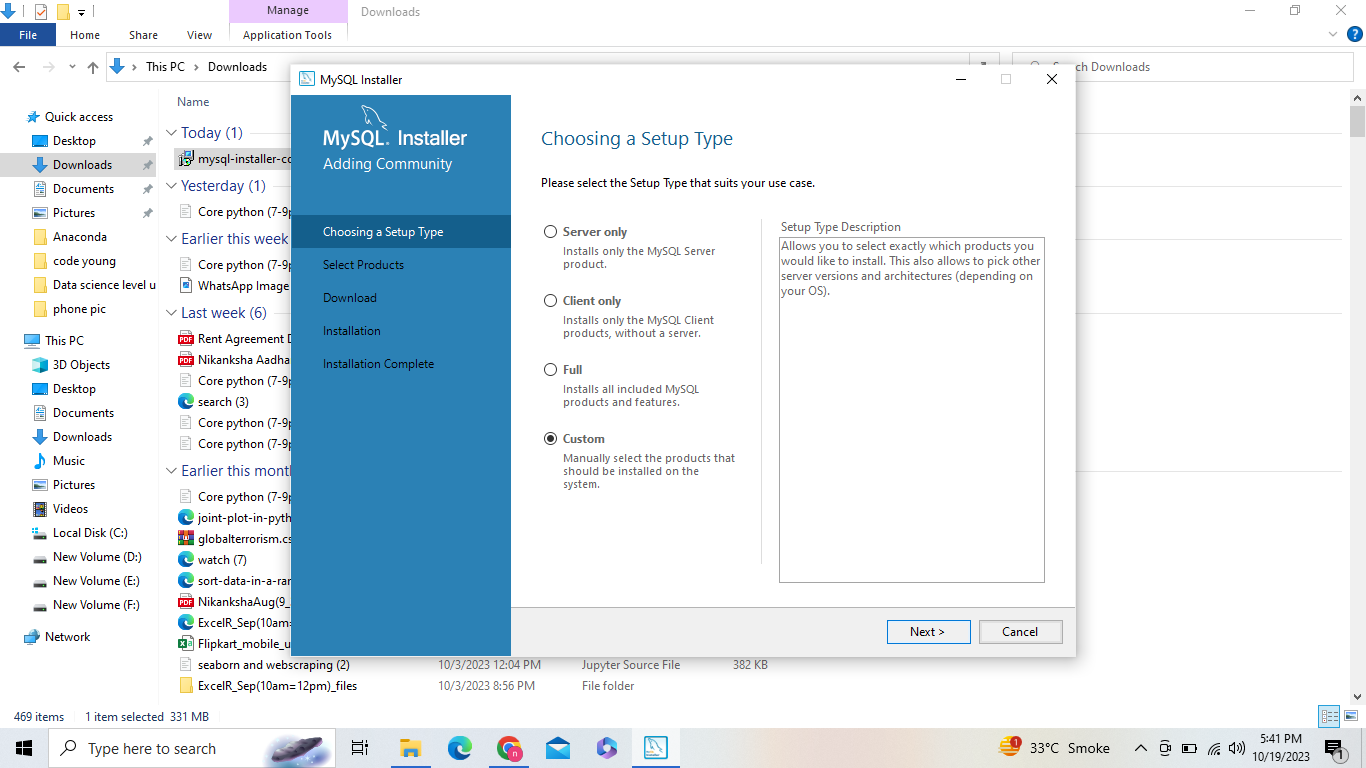


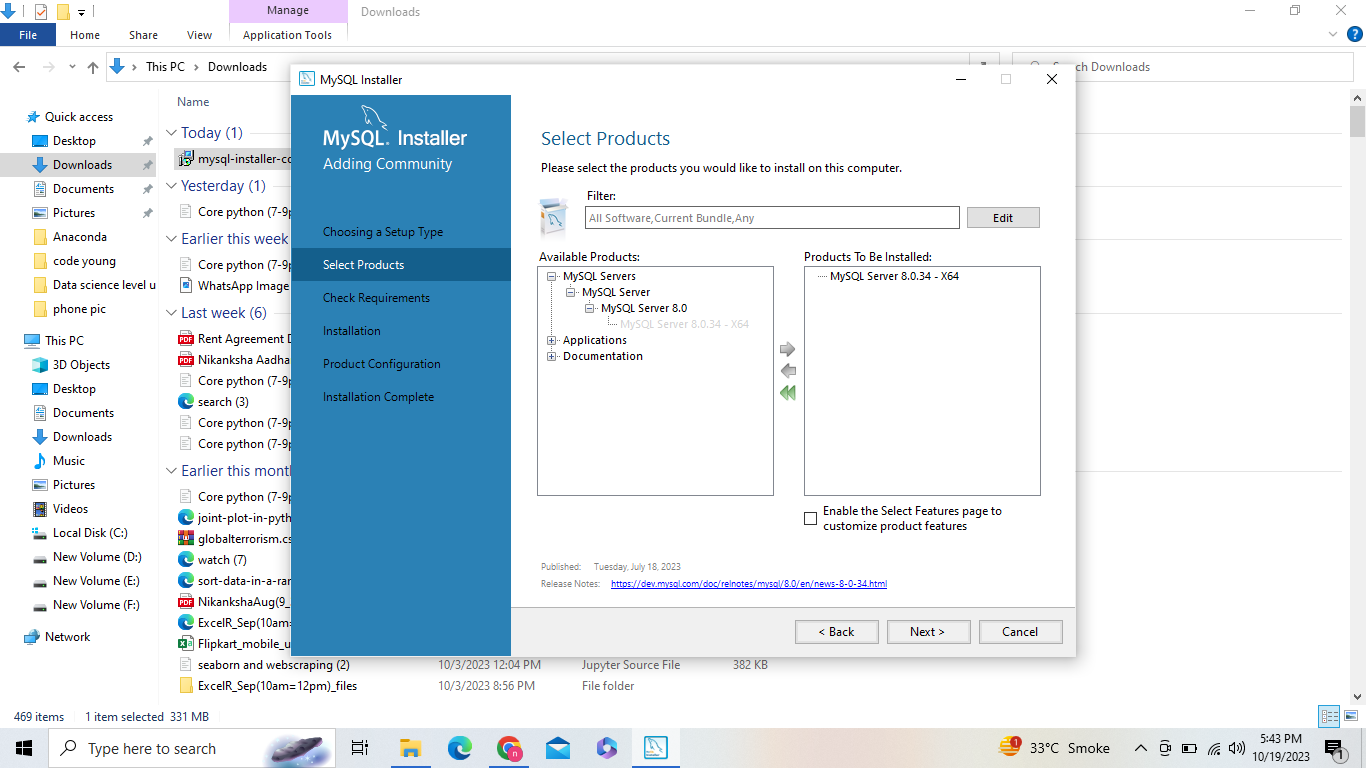
Select second option



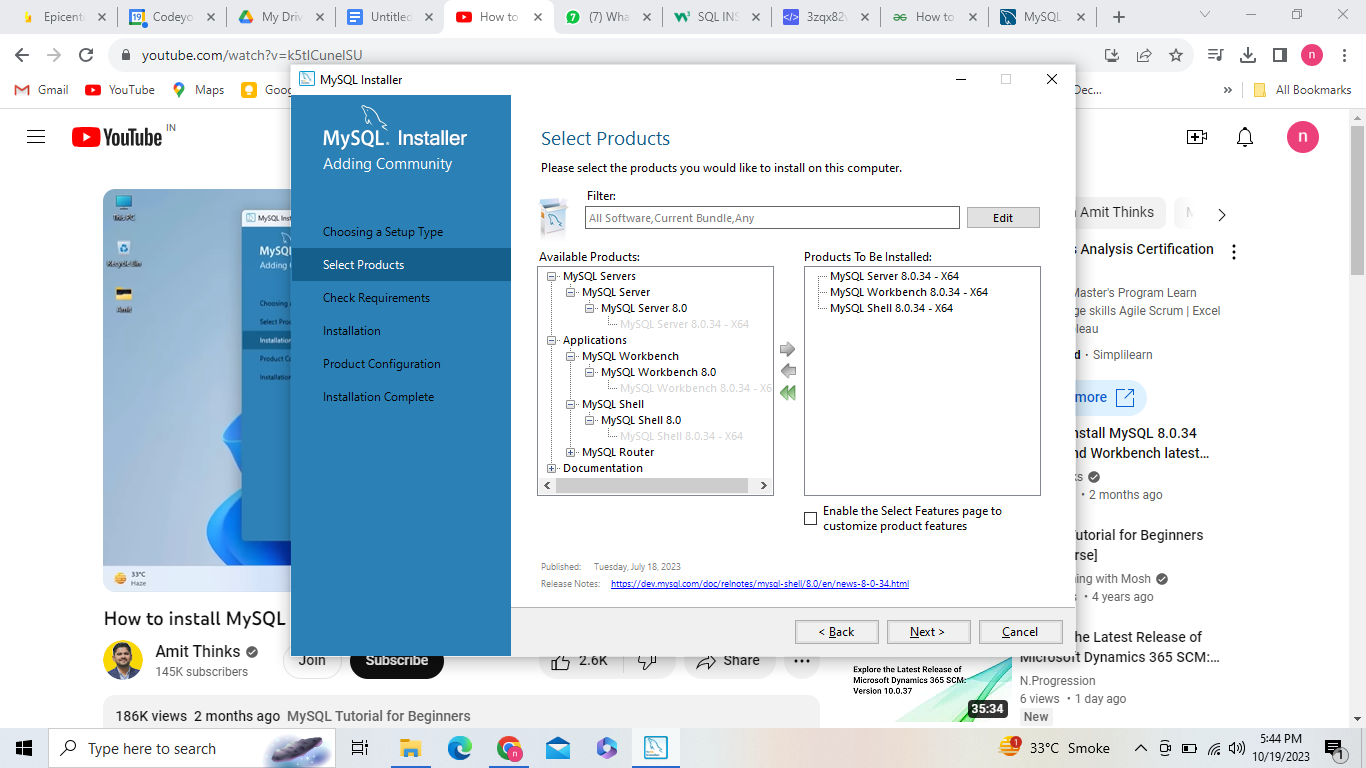
5) select No thanks Juststart download

6) Go to downloads and start installing it

7) select custom

8)use right arrow to drag 3 things 

Select only my sql server and work banch and use right arrow thn select next or execute.Will show some other window to add visual c++.we have to allow them.



## What is SQL?

* SQL stands for Structured Query Language
* SQL lets you access and manipulate databases
* SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

## What Can SQL do?

* SQL can execute queries against a database
* SQL can retrieve data from a database
* SQL can insert records in a database
* SQL can update records in a database
* SQL can delete records from a database
* SQL can create new databases
* SQL can create new tables in a database
* SQL can create stored procedures in a database
* SQL can create views in a database
* SQL can set permissions on tables, procedures, and views

## SQL is a Standard - BUT....

Although SQL is an ANSI/ISO standard, there are different versions of the SQL language.

However, to be compliant with the ANSI standard, they all support at least the major commands (such as SELECT, UPDATE, DELETE, INSERT, WHERE) in a similar manner.

Note: Most of the SQL database programs also have their own proprietary extensions in addition to the SQL standard!

## Using SQL in Your Web Site

To build a web site that shows data from a database, you will need:

* An RDBMS database program (i.e. MS Access, SQL Server, MySQL)
* To use a server-side scripting language, like PHP or ASP
* To use SQL to get the data you want
* To use HTML / CSS to style the page

## RDBMS

RDBMS stands for Relational Database Management System.

RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

The data in RDBMS is stored in database objects called tables. A table is a collection of related data entries and it consists of columns and rows.

**Extra Example:**

CREATE TABLE EMPLOYEE (

empId INTEGER PRIMARY KEY,

name TEXT NOT NULL,

dept TEXT NOT NULL

);

-- insert

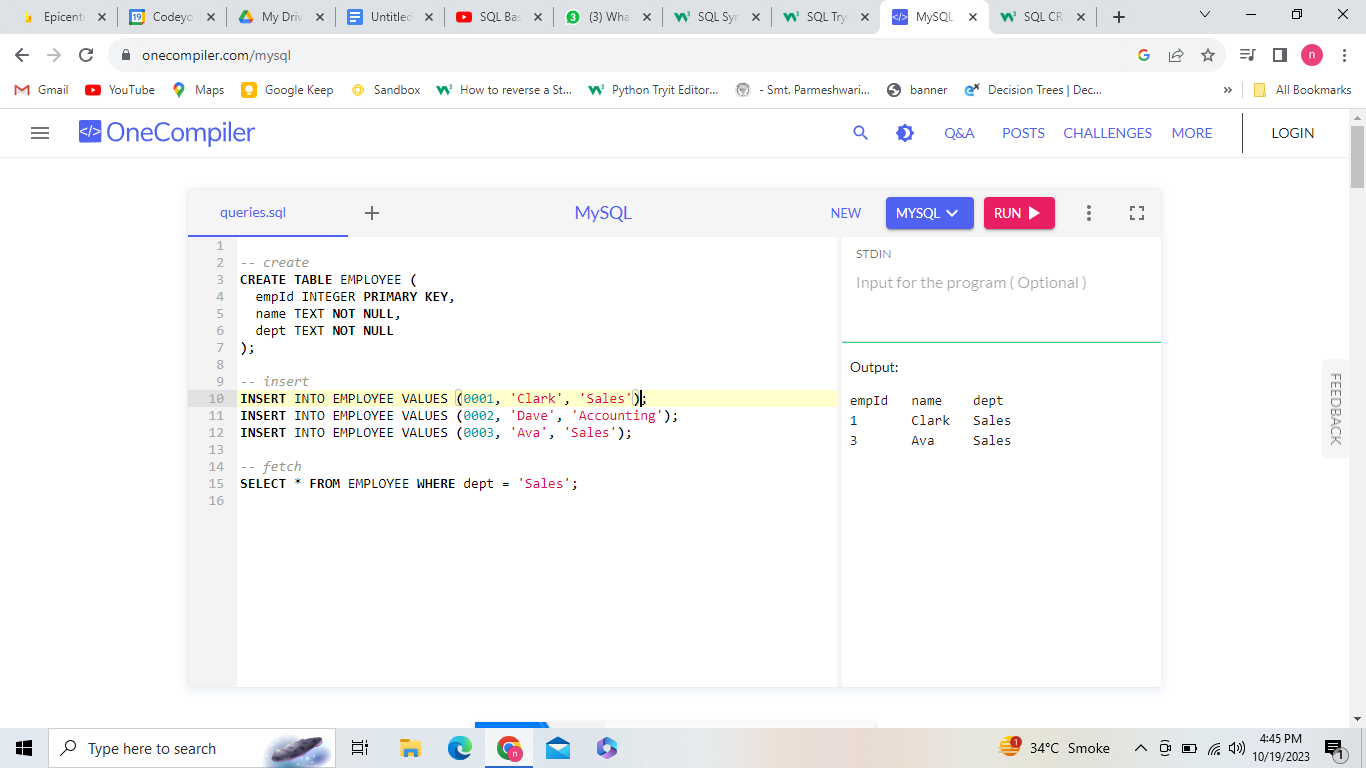
INSERT INTO EMPLOYEE VALUES (0001, 'Clark', 'Sales');

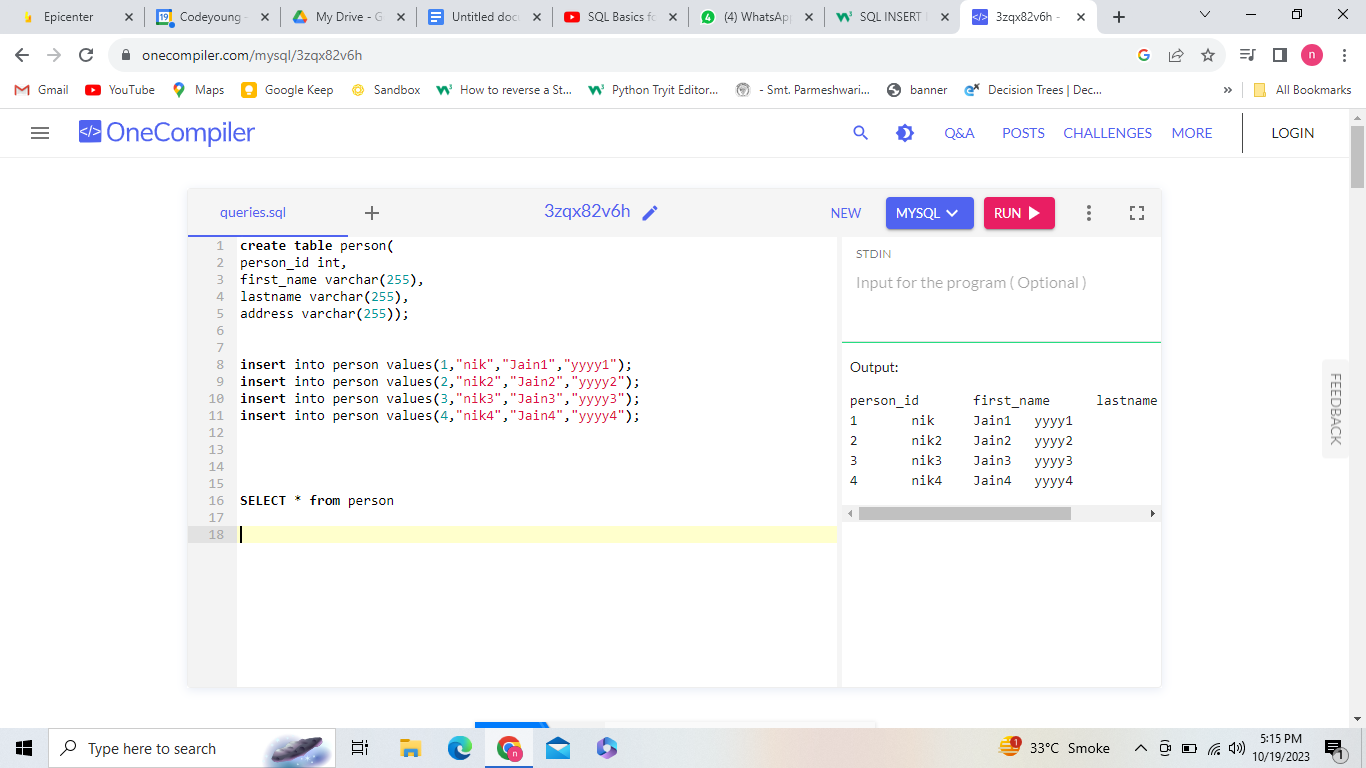
INSERT INTO EMPLOYEE VALUES (0002, 'Dave', 'Accounting');

INSERT INTO EMPLOYEE VALUES (0003, 'Ava', 'Sales');

-- fetch

SELECT \* FROM EMPLOYEE WHERE dept = 'Sales';



How to add multiple values in the table:

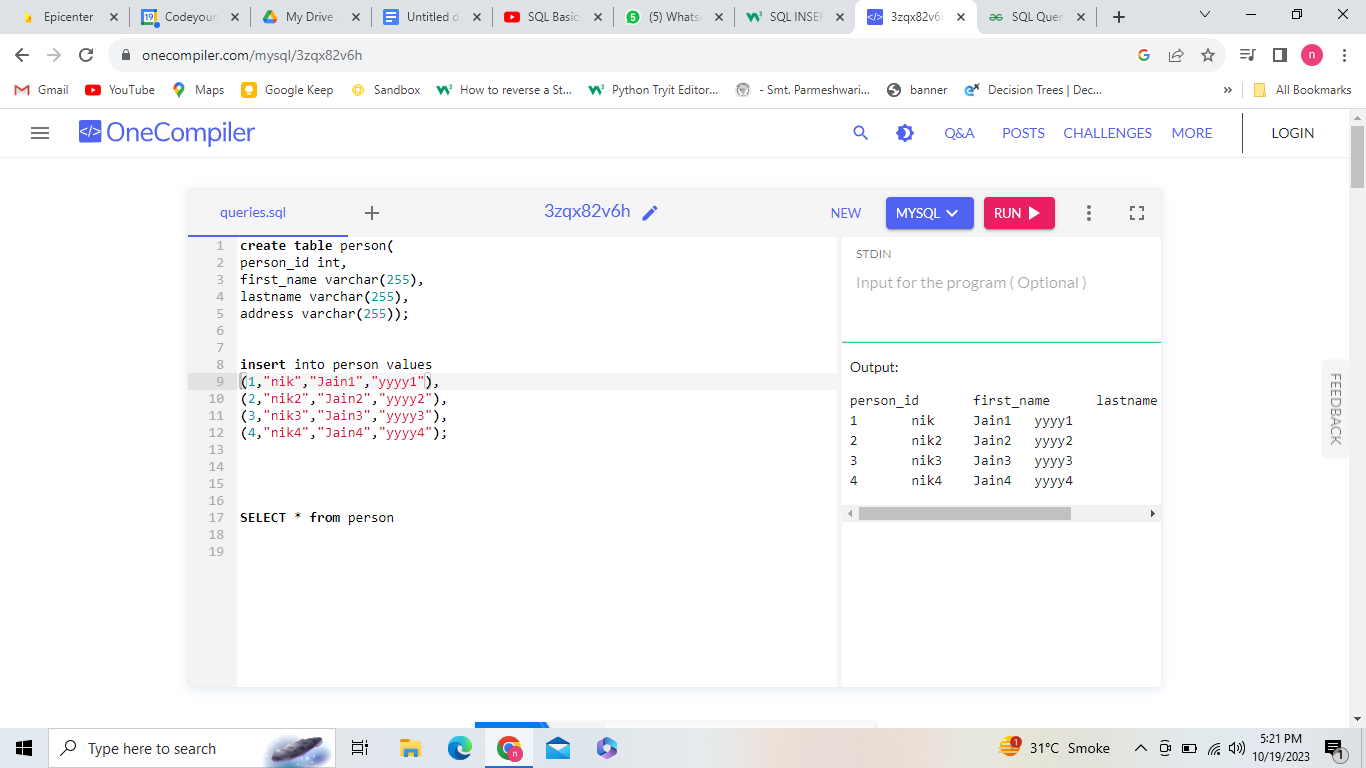
Another way to insert values:

insert into person values(1,"nik","Jain1","yyyy1"),

(2,"nik2","Jain2","yyyy2"),

(3,"nik3","Jain3","yyyy3"),

(4,"nik4","Jain4","yyyy4");



**Create table:**

create table person(

person\_id int,

first\_name varchar(255),

lastname varchar(255),

address varchar(255));

**Insert :**

insert into person (person\_id,first\_name,lastname,address)

values(1,"nik","Jain","yyyy");

**Select:**

select \* from person

**SELECT distinct values:**

SELECT person\_id,first\_name from person;

**Where condition:**

**SELECT \* from person**

**where person\_id = 1 ;**

**Example 2:**

**SELECT \* from person**

**where first\_name = "nik" ;**

**Example 3:**

**SELECT \* FROM Customers**

**WHERE CustomerID > 80;**

**Distinct : Select all the different countries from the "Customers" table:Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.**

**SELECT DISTINCT Country FROM Customers;**

**If you omit the DISTINCT keyword, the SQL statement returns the "Country" value from all the records of the "Customers" table:**

**select first\_name from student;**

**By using the DISTINCT keyword in a function called COUNT, we can return the number of different countries.**

**The following operators can be used in the WHERE clause:**

**= Equal**

**> Greater than**

**< Less than**

**>= Greater than or equal**

**<= Less than or equal**

**<> Not equal. Note: In some versions of SQL this operator may be written as !=**

**BETWEEN Between a certain range**

**LIKE Search for a pattern**

**IN To specify multiple possible values for a column**

**For more queries :**

[**https://www.w3schools.com/sql/sql\_insert.asp**](https://www.w3schools.com/sql/sql_insert.asp)

**SELECT \* FROM Customers**

**WHERE Country='Mexico';**

**SELECT \* FROM Customers**

**WHERE CustomerID > 80;**

**Order by:The ORDER BY keyword is used to sort the result-set in ascending or descending order.**

**SELECT \* FROM Products**

**ORDER BY Price;**

**SELECT \* FROM Products**

**ORDER BY Price DESC;**

## Order Alphabetically:**For string values the ORDER BY keyword will order alphabetically:**

**SELECT \* FROM Products**

**ORDER BY ProductName;**

**To sort the table reverse alphabetically, use the DESC keyword:**

**SELECT \* FROM Products**

**ORDER BY ProductName DESC;**

## **Using Both ASC and DESC:**

**SELECT \* FROM Customers**

**ORDER BY Country ASC, CustomerName DESC;**

## **The SQL AND Operator:**

**SELECT \*FROM Customers**

**WHERE Country = 'Spain' AND CustomerName LIKE 'G%';**

**SELECT \* FROM Customers**

**WHERE Country = 'Germany'**

**AND City = 'Berlin'**

**AND PostalCode > 12000;**

**SELECT \* FROM Customers**

**WHERE Country = 'Spain' AND (CustomerName LIKE 'G%' OR CustomerName LIKE 'R%');**

**Or operator:**

**SELECT \*FROM Customers**

**WHERE Country = 'Germany' OR Country = 'Spain';**

**Not operator:**

**SELECT \* FROM Customers**

**WHERE NOT Country = 'Spain';**

**Select customers that does not start with the letter 'A':**

**SELECT \* FROM Customers**

**WHERE CustomerName NOT LIKE 'A%';**

**SELECT \* FROM Customers**

**WHERE CustomerID NOT BETWEEN 10 AND 60;**

**Select customers that are not from Paris or London:**

**SELECT \* FROM Customers**

**WHERE City NOT IN ('Paris', 'London');**

**SELECT \* FROM Customers**

**WHERE NOT CustomerId < 50;**

**Update Table:**

**UPDATE Customers**

**SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'**

**WHERE CustomerID = 1;**

**UPDATE Customers**

**SET ContactName='Juan'**

**WHERE Country='Mexico';**

**Warning:Be careful when updating records. If you omit the WHERE clause, ALL records will be updated!**

**UPDATE Customers**

**SET ContactName='Juan';**

**Delete**

**DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';**

**Delete all the records:**

**DELETE FROM Customers;**

**Drop table:**

**DROP TABLE Customers;**

## **The SQL MIN() and MAX() Functions:**

SELECT MIN(Price)

FROM Products;

SELECT MAX(Price)

FROM Products;

**Count:The COUNT() function returns the number of rows that matches a specified criterion.**

SELECT COUNT(\*)

FROM Products;

**SELECT COUNT(ProductID)**

**FROM Products**

**WHERE Price > 20;**

## **The SQL SUM() Function**

**SELECT SUM(Quantity)**

**FROM OrderDetails;**

**Avg:**

**SELECT AVG(Price)**

**FROM Products;**

## The SQL LIKE Operator:

**The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.**

**There are two wildcards often used in conjunction with the LIKE operator:**

* **The percent sign % represents zero, one, or multiple characters**
* **The underscore sign \_ represents one, single character**

**Select all customers that starts with the letter "a":**

**SELECT \* FROM Customers**

**WHERE CustomerName LIKE 'a%';**

**SELECT \* FROM Customers**

**WHERE city LIKE 'L\_nd\_\_';**

**SELECT \* FROM Customers**

**WHERE city LIKE '%L%';**

**SELECT \* FROM Customers**

**WHERE CustomerName LIKE 'La%';**

**SELECT \* FROM Customers**

**WHERE CustomerName LIKE 'a%' OR CustomerName LIKE 'b%';**

**SELECT \* FROM Customers**

**WHERE CustomerName LIKE '%a';**

**Return all customers that starts with "b" and ends with "s":**

**SELECT \* FROM Customers**

**WHERE CustomerName LIKE 'b%s';**

**Return all customers that starts with "a" and are at least 3 characters in length:**

**SELECT \* FROM Customers**

**WHERE CustomerName LIKE 'a\_\_%';**

**Return all customers from Spain:**

**SELECT \* FROM Customers**

**WHERE Country LIKE 'Spain';**

**Sql in:**

**Return all customers from 'Germany', 'France', or 'UK'**

**SELECT \* FROM Customers**

**WHERE Country IN ('Germany', 'France', 'UK');**

**SELECT \* FROM Customers**

**WHERE Country NOT IN ('Germany', 'France', 'UK');**

## **The SQL BETWEEN Operator**

**The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates.**

**The BETWEEN operator is inclusive: begin and end values are included.**

**Selects all products with a price between 10 and 20:**

**SELECT \* FROM Products**

**WHERE Price BETWEEN 10 AND 20;**

**How to add new column name:**

Use an Alias**Give the summarized column a name by using the AS keyword.**

### Example**Name the column "total":**

**SELECT SUM(Quantity) AS total**

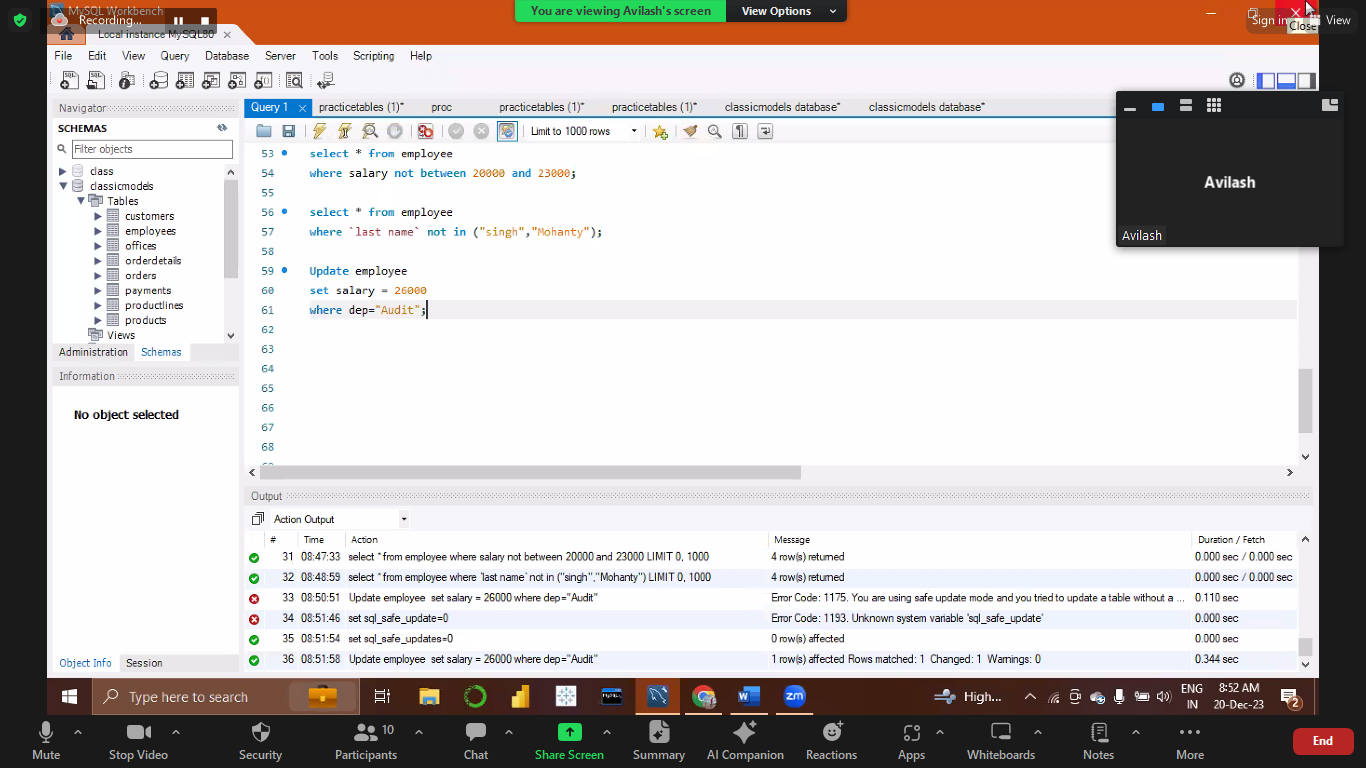
**FROM OrderDetails;**

**Return all products with a higher price than the average price:**

**SELECT \* FROM Products**

**WHERE price > (SELECT AVG(price) FROM Products);**

**We have to off set mode**

****