

## **How to Install MySQL Workbench on Windows**

### **Step 1: Go to the Official MySQL Workbench Download Page**

- Visit: <https://dev.mysql.com/downloads/workbench/>
- [MySQL :: Begin Your Download](#)

### **Step 2: Select the Windows Platform**

- Under **Select Operating System**, choose Microsoft Windows.
- Click the **Download** button next to the **Windows (x86, 64-bit), MSI Installer**.

### **Step 3: Skip Oracle Login (Optional)**

- You'll be prompted to log in or sign up for an Oracle account.
- Click "**No thanks, just start my download**" to continue directly.

### **Step 4: Run the Installer**

- After download completes, double-click the **.msi file**.
- Follow the **installation wizard**:
  - Click **Next** to accept defaults.
  - Choose **Complete Setup** (recommended for all features).

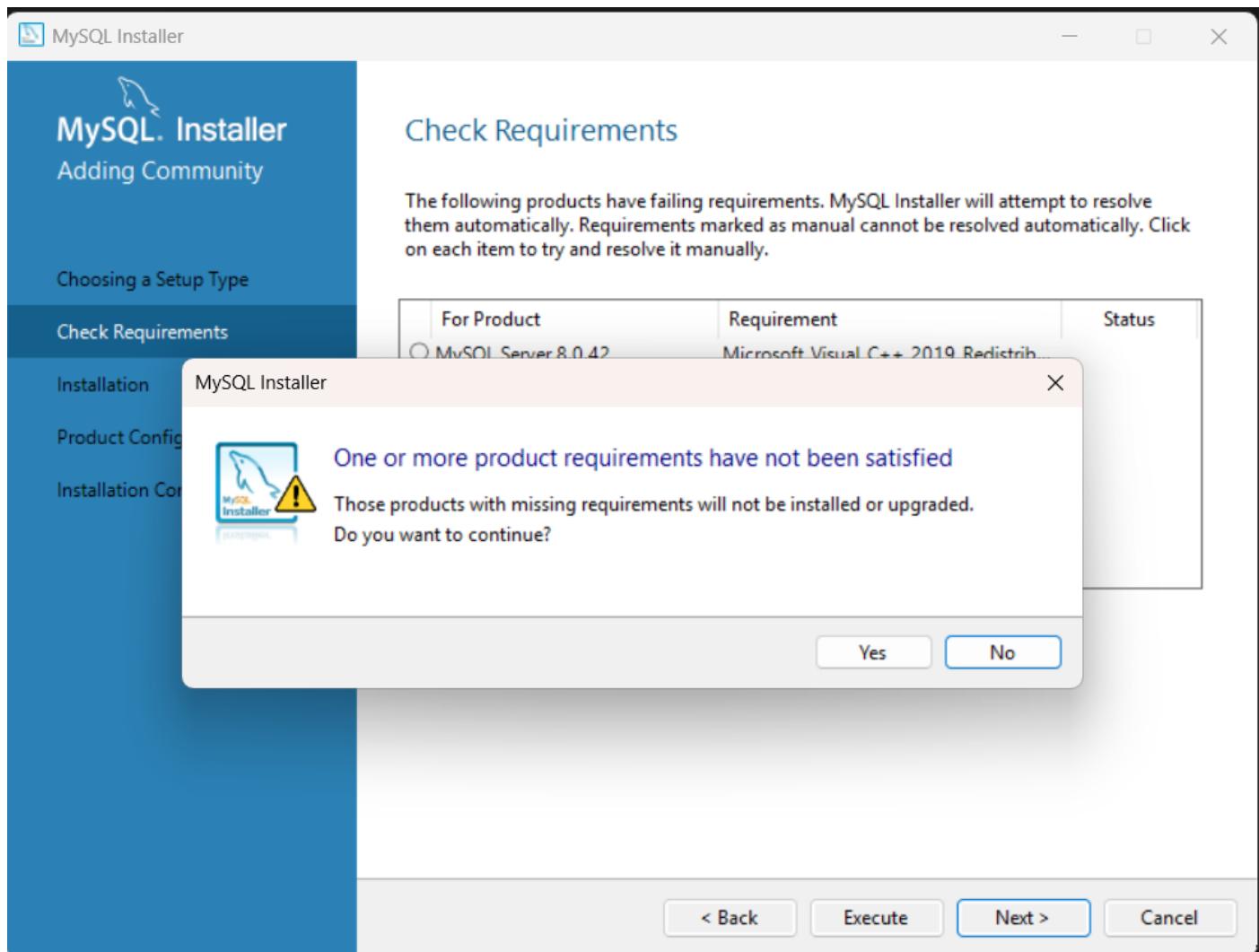
### **Step 5: Launch MySQL Workbench**

- Once installed, open **MySQL Workbench** from the Start menu.
- You'll see the **home screen** where you can add a new MySQL connection.

### **Step 6: Connect to MySQL Server**

- Click "**+**" (**New Connection**).
- Enter the following details:
  - **Connection Name:** Any name you like (e.g., "localhost")
  - **Hostname:** localhost
  - **Port:** 3306

- **Username:** root
- **Password:** (click **Store in Vault** and enter the one you set earlier)
- Click **Test Connection** → then OK.



## How to Fix This

### Step 1: Manually Download Visual C++ 2019 Redistributable

1. Go to the official Microsoft download page:  
<https://learn.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist>
2. Scroll down to **Visual Studio 2015, 2017, 2019, and 2022** section.
3. Download:
  - vc\_redist.x64.exe (for 64-bit Windows)
  - vc\_redist.x86.exe (for 32-bit compatibility)
4. **Install both** by running the .exe files and following the instructions.

### Step 2: Restart MySQL Installer

- Close the MySQL Installer.

- Reopen it.
- Click **Check Requirements** again.
- Now the missing requirement should be marked **Resolved**.

## What is SQL?

SQL stands for **Structured Query Language**.

It is a **standard programming language** used to **store, retrieve, manage, and manipulate data** in **relational databases** like MySQL, PostgreSQL, SQL Server, and SQLite.

## What is MySQL?

MySQL is an **open-source relational database management system (RDBMS)** that uses **SQL (Structured Query Language)** to manage data.

It stores data in **tables** (rows and columns) and allows you to perform operations like insert, update, delete, search, and organize data efficiently.

Feature	SQL	MySQL
abc Full Form	Structured Query Language	My Structured Query Language
❑ Type	Language	Software / RDBMS (Relational Database Management System)
🎯 Purpose	Used to write queries to interact with databases	A database system that uses SQL to manage data
☒ What It Does	Defines and manipulates data (e.g., SELECT, INSERT)	Stores, retrieves, and manages databases using SQL
🔧 Developer	Standard maintained by ISO/ANSI	Originally developed by MySQL AB (now owned by Oracle)
💼 Use Case	You use SQL to <b>talk</b> to a database	You use MySQL to <b>store and manage</b> the data
🔑 License	Not a software; it's a language standard	Open-source (Community Edition) and Paid (Enterprise)
💬 Usage Example	SELECT * FROM users;	Executing this SQL query <b>on</b> MySQL database

## CRUD OPERATION:

```
CREATE DATABASE abc;
```

```
use abc;
```

```
CREATE TABLE tbl2(id INT AUTO_INCREMENT PRIMARY KEY,name VARCHAR(50) ,age  
VARCHAR(10))
```

```
INSERT INTO tbl2(name,age) values ('animesh','27');
```

```
INSERT INTO tbl2(name,age) values ('animesh','27'),('abc','90');
```

```
select * from tbl2;
```

```
update tbl2 set age=47 where id=1;
```

```
delete from tbl2 where id=1;
```

```
ALTER table tbl2 ADD marks VARCHAR(50);
```

```
ALTER table tbl2 DROP marks;
```

```
drop table tbl2;
```

```
drop database abc;
```

## **Connection with Python**

**pip install mysql**

**pip install mysql-connector-python**

**pip install pymysql**

**# # # # add data base in mysql**

```
# import mysql.connector as myc  
  
# conn = myc.connect(host="localhost" ,user="root",password="animesh@1234567")  
  
# mycursor = conn.cursor()  
  
# mycursor.execute("CREATE DATABASE soubhagya")  
  
  
# print("done")
```

### **# # # # add table in database**

```
# import mysql.connector as myc  
  
# conn = myc.connect(host="localhost"  
,user="root",password="animesh@1234567",database="soubhagya")  
  
# mycursor = conn.cursor()  
  
# mycursor.execute("CREATE TABLE tbl2(id INT AUTO_INCREMENT PRIMARY KEY,name  
VARCHAR(50) ,age VARCHAR(10))")  
  
# print("done")
```

### **# # # # insertion in table**

```
# import mysql.connector as myc  
  
# conn = myc.connect(host="localhost"  
,user="root",password="animesh@1234567",database="soubhagya")  
  
# mycursor = conn.cursor()  
  
# sql= "INSERT INTO tbl2(name,age)VALUES(%s,%s)"  
  
# val=[('animesh','27'),  
#     ('abhishek','28'),  
#     ('aryan','19')  
# ]  
  
# mycursor.executemany(sql,val)  
  
# conn.commit()
```

```
# print("done")
```

## # # # # # read data in table

```
# import mysql.connector as myc

# conn = myc.connect(host="localhost" ,user="root"
",password="animesh@1234567",database="soubhagya")

# mycursor = conn.cursor()

# mycursor.execute("select * from tbl2")

# myconn= mycursor.fetchall()

# for i in myconn:

#     print(i)

# print("done")
```

## # # # # update the table

```
# import mysql.connector as myc

# conn = myc.connect(host="localhost"
,user="root",password="animesh@1234567",database="soubhagya")

# mycursor = conn.cursor()

# mycursor.execute("update tbl2 set age=47 where id=1")

# conn.commit()

# print("done")
```

## # # # # delete the table

```
# import mysql.connector as myc
```

```
# conn = myc.connect(host="localhost"
, user="root",password="animesh@1234567",database="soubhagya")

# mycursor = conn.cursor()

# mycursor.execute("delete from tbl2 where id=1")

# conn.commit()

# print("done")
```

### **# # # add column in table with the help of alter**

```
# import mysql.connector as myc

# conn = myc.connect(host="localhost"
, user="root",password="animesh@1234567",database="soubhagya")

# mycursor = conn.cursor()

# mycursor.execute("ALTER table tbl2 ADD marks VARCHAR(50)")

# conn.commit()

# print("done")
```

### **# # # # delete column in table with the help of alter**

```
# import mysql.connector as myc

# conn = myc.connect(host="localhost"
, user="root",password="animesh@1234567",database="soubhagya")

# mycursor = conn.cursor()

# mycursor.execute("ALTER table tbl2 DROP marks")

# conn.commit()

# print("done")
```

### **# # # delete the table**

```
# import mysql.connector as myc
```

```
# conn = myc.connect(host="localhost"
, user="root",password="animesh@1234567",database="soubhagya")

# mycursor = conn.cursor()

# mycursor.execute("drop table tbl2")

# conn.commit()

# print("done")
```

#### **# # # # delete database**

```
# import mysql.connector as myc

# conn = myc.connect(host="localhost"
, user="root",password="animesh@1234567",database="soubhagya")

# mycursor = conn.cursor()

# mycursor.execute("drop database soubhagya")

# conn.commit()

# print("done")
```

### **MYSQL WITH TKINTER**

```
from tkinter import *

import mysql.connector as mysql

from tkinter import messagebox

root=Tk()

def conn():

    name=e.get()

    age=e1.get()

    if name == "" or age == "":

        messagebox.showerror("error","fill all fields")

    else:
```

```
con=mysql.connect(host="localhost",user="root",password="animesh@1234567",database  
="abc")  
  
mycursor=con.cursor()  
  
s="insert into tbl(name,age) values (%s,%s)"  
  
v=(name,age)  
  
mycursor.execute(s,v)  
  
con.commit()  
  
mycursor.close()  
  
con.close()  
  
messagebox.showinfo("sucess","sucessfully done")
```

```
e11=Label(root,text="name")  
e11.place(x=100,y=100)  
  
e=Entry(root,width=30)  
e.place(x=150,y=100)  
  
e12=Label(root,text="age")  
e12.place(x=100,y=150)  
  
e1=Entry(root,width=30)  
e1.place(x=150,y=150)  
  
btn=Button(root,text="submit",padx=20,pady=30,command=conn)  
btn.place(x=200,y=200)  
  
mainloop()
```

## **PRACTICE QUESTIONS:**

### **1. Create a Database and Table**

*Write a Python script to create a MySQL database named school and a table named students with columns: id, name, and age.*

### **2. Insert Single Record into Table**

*Write a Python program to insert a student's name and age into the students table.*

### **3. Insert Multiple Records**

*Insert a list of multiple students into the students table using Python.*

### **4. Retrieve and Display Data**

*Write a Python program to retrieve all records from the students table and print them.*

### **5. Update a Student's Age**

*Update the age of a student based on their ID using a Python MySQL connection.*

### **6. Delete a Record by ID**

*Write a Python script to delete a student record from the students table using their ID.*

### **7. Alter Table to Add New Column**

*Add a new column email to the students table using a Python script.*

### **8. Drop Table or Database**

*Write a Python program to delete the entire students table and then the school database.*

### **9. Create a GUI Form to Add Student Record**

*Design a Tkinter form to input name and age, and insert this data into the MySQL students table.*

### **10. Display All Student Records in GUI**

*Use Tkinter to display all student records in a multi-line Text widget or a label.*

### **11. Search Student by Name (GUI)**

*Create a GUI where users can enter a student's name and display their details from MySQL.*

## **12. Delete Student Record from GUI**

*Create a Tkinter app with a field to input a student ID and delete that student from the MySQL database.*

## **13. Update Student Age via GUI**

*Design a GUI to update a student's age by entering their name.*