# The University of Haripur

Khyber Pakhtunkhwa

| | |
|---|---|
| **Name** | **Hammad Younis Abbasi** |
| **Roll Number** | **F23-0159** |
| **Subject** | **Database System** |
| **Department** | **Information Technology** |
| **Teacher** | **Wafa Habib** |

# Table of Contant

# LAB NO 01:

## INTRODUCTION:

**MYSQL:**

MySQL is a relational database management system (RDBMs) developed by Oracle that is based on structured query language (SQL). A database is structured collection of data. It maybe anything from a simple shopping list to a picture gallery or a place to hold the vast amounts of information in a cooperate network.

WHY IS IT USED?

MySQL is an open source relational database management , scalability, and ease of use make MySQL a popular choice for developers. In fact you will find it at the heart of demanding, high traffic applications such as Facebook, Netflix, Shopify and booking.com.

DATABASE MANAGENEMT SYSTEM:

A Database Management System (DBMs) is a software system for creating and managing database. A DBMs enables end users to create, protect, read, update, and delete data in a database. It also manages security, data integrity and concurrency for data bases.

# LAB NO 02:

## INSTALLATION:

Installing MySQL on your Windows PC is a straightforward process, but it requires ensuring that your system meets specific hardware and software prerequisites.

In this article, We will learn about How to Install MySQL on Windows by understanding each step in detail.

What is MySQL?

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

It is widely used for storing, retrieving, and managing data in applications ranging from small-scale projects to large enterprise systems.

MySQL is known for its reliability, scalability, and ease of use, making it a popular choice for web developers and organizations worldwide.

Note: Uninstall any previous MySQL versions to avoid potential conflicts during the new installation.

Hardware and Software Requirements to Install MySQL

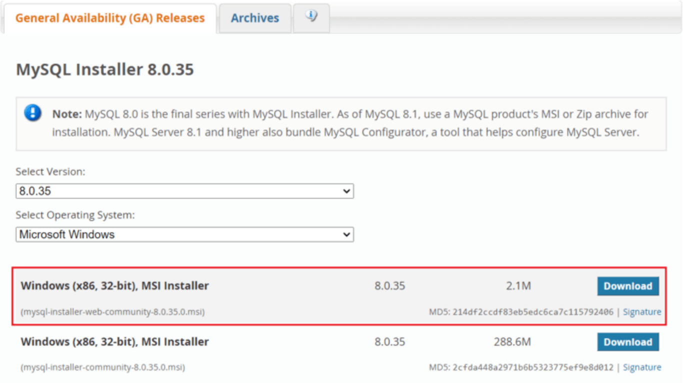
Before installing MySQL to your PC, ensure your system has a capable processor (like Intel Core), a minimum of 4 GB RAM (or 6 GB), a compatible graphics card, and a display with at least 1024×768 resolution.

Download and Install MySQL for Windows Steps

Now, Let' 's break down MySQL software downloading steps for a better understanding and see install MySQL on Windows 10 step by step.

## Step 1: Visit the Official MySQL Website

Open your preferred web browser and navigate to the official MySQL website. Now, Simple click on first download button.



MySQL Installer 8.0.35

**Note:** MySQL 8.0 is the final series with MySQL Installer. As of MySQL 8.1, use a MySQL product's MSI or Zip archive for installation. MySQL Server 8.1 and higher also bundle MySQL Configurator, a tool that helps configure MySQL Server.

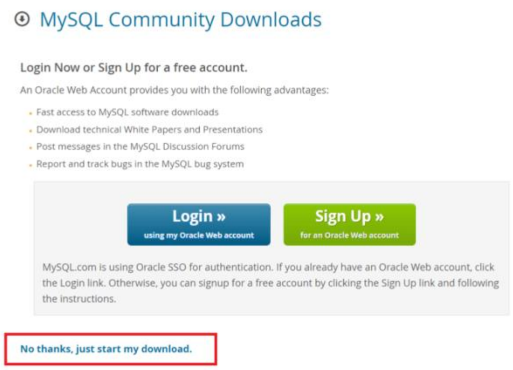
Select Version: 8.0.35

Select Operating System: Microsoft Windows

| | | | |
|---|---|---|---|
| **Windows (x86, 32-bit), MSI Installer**<br>(mysql-installer-web-community-8.0.35.0.msi) | 8.0.35 | 2.1M | Download<br>MD5: 214df2ccdf83eb5edc6ca7c115792406 \| Signature |
| **Windows (x86, 32-bit), MSI Installer**<br>(mysql-installer-community-8.0.35.0.msi) | 8.0.35 | 288.6M | Download<br>MD5: 2cfda448a2971b6b5323775ef9e8d012 \| Signature |

## Step 2: Go to the Downloads Section

On the MySQL homepage, Click on the " No thanks, just start my download" link to proceed MySql downloading.



MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

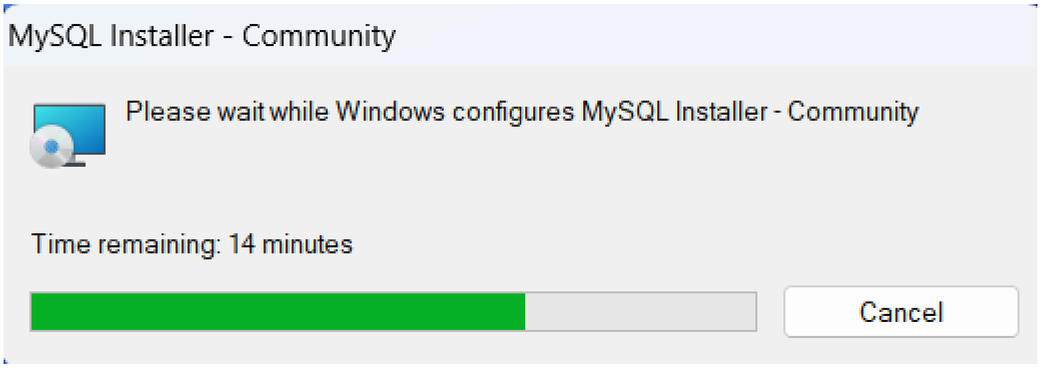
- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

No thanks, just start my download.

## Step 3: Run the Installer

After MySQL downloading MySQL.exe file , go to your Downloads folder, find the file, and double-click to run the installer.



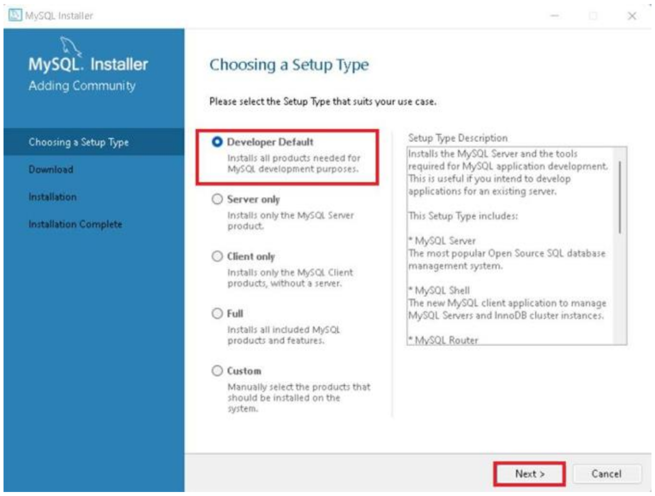
MySQL Installer - Community

Please wait while Windows configures MySQL Installer - Community

Time remaining: 14 minutes

## Step 4: Choose Setup Type

The installer will instruct you to choose the setup type. For most users, the “Developer Default” is suitable. Click “Next” to proceed.



MySQL Installer

MySQL. Installer
Adding Community

Choosing a Setup Type

Please select the Setup Type that suits your use case.

● Developer Default
Installs all products needed for MySQL development purposes.

○ Server only
Installs only the MySQL Server product.

○ Client only
Installs only the MySQL Client products, without a server.

○ Full
Installs all included MySQL products and features.

○ Custom
Manually select the products that should be installed on the system.

Setup Type Description

Installs the MySQL Server and the tools required for MySQL application development. This is useful if you intend to develop applications for an existing server.

This Setup Type includes:

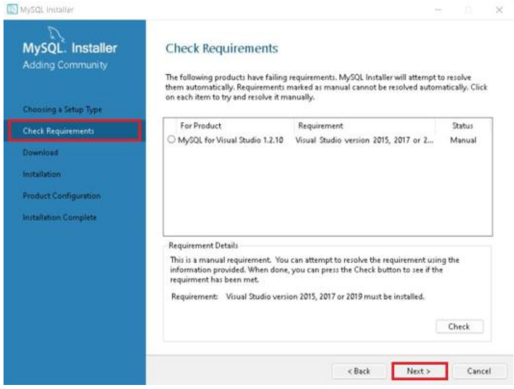
* MySQL Server
The most popular Open Source SQL database management system.

* MySQL Shell
The new MySQL client application to manage MySQL Servers and InnoDB cluster instances.

* MySQL Router

## Step 5: Check Requirements

You might be prompted to install necessary MySQL software, typically Visual Code. The installer can auto-resolve some issues, but not in this case.



MySQL Installer

MySQL. Installer
Adding Community

Check Requirements

The following products have failing requirements. MySQL Installer will attempt to resolve them automatically. Requirements marked as manual cannot be resolved automatically. Click on each item to try and resolve it manually.

| For Product | Requirement | Status |
|---|---|---|
| ○ MySQL for Visual Studio 1.2.10 | Visual Studio version 2015, 2017 or 2... | Manual |

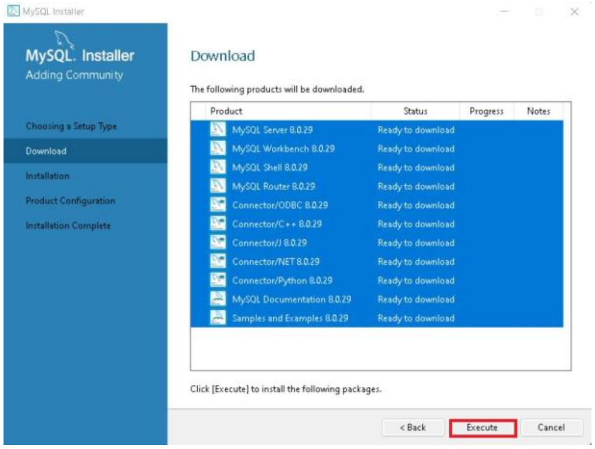
Requirement Details

This is a manual requirement. You can attempt to resolve the requirement using the information provided. When done, you can press the Check button to see if the requirment has been met.

Requirement: Visual Studio version 2015, 2017 or 2019 must be installed.

## Step 6: MySQL Downloading

Now that you’re in the download section, click “Execute” to start downloading the components you selected. Wait a few minutes until all items show tick marks, indicating completion, before moving forward.



Download

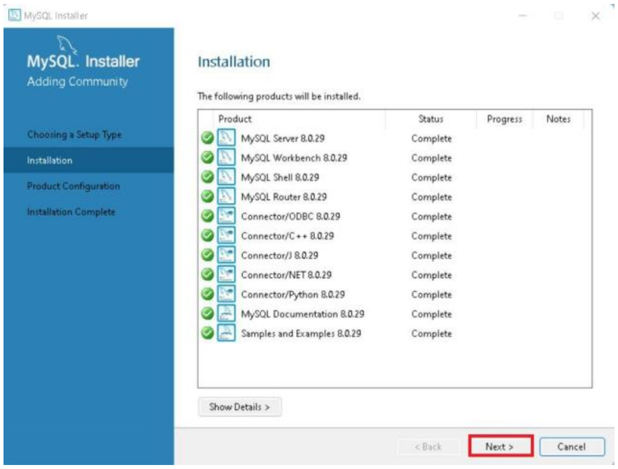
The following products will be downloaded.

| Product | Status | Progress | Notes |
|---|---|---|---|
| MySQL Server 8.0.29 | Ready to download | | |
| MySQL Workbench 8.0.29 | Ready to download | | |
| MySQL Shell 8.0.29 | Ready to download | | |
| MySQL Router 8.0.29 | Ready to download | | |
| Connector/ODBC 8.0.29 | Ready to download | | |
| Connector/C++ 8.0.29 | Ready to download | | |
| Connector/J 8.0.29 | Ready to download | | |
| Connector/NET 8.0.29 | Ready to download | | |
| Connector/Python 8.0.29 | Ready to download | | |
| MySQL Documentation 8.0.29 | Ready to download | | |
| Samples and Examples 8.0.29 | Ready to download | | |

Click [Execute] to install the following packages.

## Step 7: MySqL Installation

Now the downloaded components will be installed. Click “Execute” to start the installation process. MySQL will be installed on your Windows system. Then click Next to proceed



Installation

The following products will be installed.

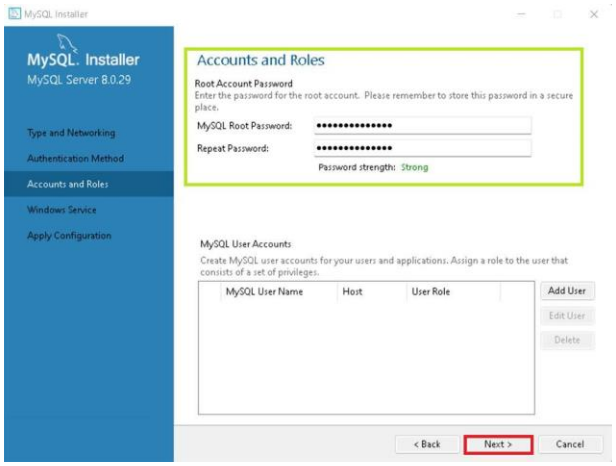
| Product | Status | Progress | Notes |
|---|---|---|---|
| MySQL Server 8.0.29 | Complete | | |
| MySQL Workbench 8.0.29 | Complete | | |
| MySQL Shell 8.0.29 | Complete | | |
| MySQL Router 8.0.29 | Complete | | |
| Connector/ODBC 8.0.29 | Complete | | |
| Connector/C++ 8.0.29 | Complete | | |
| Connector/J 8.0.29 | Complete | | |
| Connector/NET 8.0.29 | Complete | | |
| Connector/Python 8.0.29 | Complete | | |
| MySQL Documentation 8.0.29 | Complete | | |
| Samples and Examples 8.0.29 | Complete | | |

## Step 8: Navigate to Few Configuration Pages

Proceed to “Product Configuration” > “Type and Networking” > “Authentication Method” Pages by clicking the “Next” button.

## Step 9: Create MySQL Accounts

Create a password for the MySQL root user. Ensure it’s strong and memorable. Click “Next” to proceed.



Accounts and Roles

Root Account Password

Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password: ••••••••••••••

Repeat Password: ••••••••••••••

Password strength: Strong

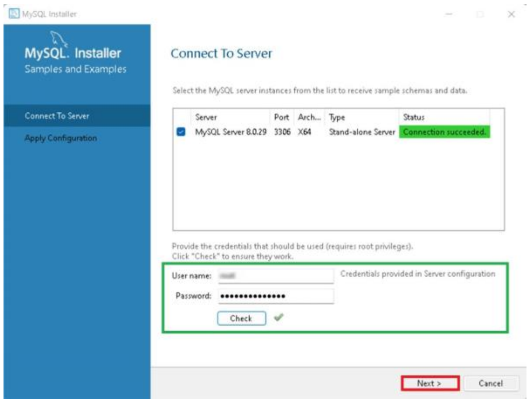
MySQL User Accounts

Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.

| MySQL User Name | Host | User Role |
|---|---|---|

## Step 10: Connect To Server

Enter the root password, click Check. If it says “Connection succeed,” you’ve successfully connected to the server.



Connect To Server

Select the MySQL server instances from the list to receive sample schemas and data.

| | Server | Port | Arch... | Type | Status |
|---|---|---|---|---|---|
| ☑ | MySQL Server 8.0.29 | 3306 | X64 | Stand-alone Server | Connection succeeded. |

Provide the credentials that should be used (requires root privileges).
Click "Check" to ensure they work.

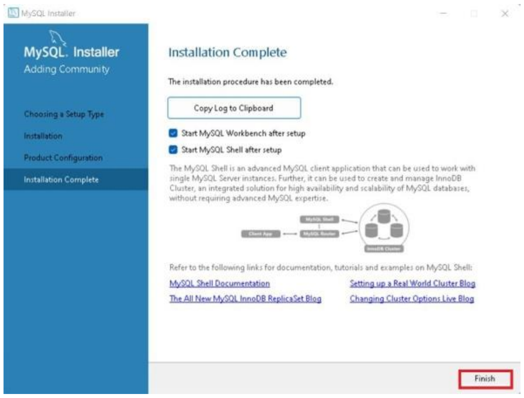
User name: [redacted]

Password: ••••••••••••••

Credentials provided in Server configuration

## Step 11: Complete Installation

Once the installation is complete, click “Finish.” Congratulations! MySQL is now installed on your Windows system.



Installation Complete

The installation procedure has been completed.

☑ Start MySQL Workbench after setup

☑ Start MySQL Shell after setup

The MySQL Shell is an advanced MySQL client application that can be used to work with single MySQL Server instances. Further, it can be used to create and manage InnoDB Cluster, an integrated solution for high availability and scalability of MySQL databases, without requiring advanced MySQL expertise.

Refer to the following links for documentation, tutorials and examples on MySQL Shell:

MySQL Shell Documentation
Setting up a Real World Cluster Blog
The All New MySQL InnoDB ReplicaSet Blog
Changing Cluster Options Live Blog

## Step 12: Verify Installation

To ensure a successful installation of MySQL, open the MySQL Command Line Client or MySQL Workbench, both available in your Start Menu. Log in using the root user credentials you set during installation.

MySQL Workbench Is Ready To Use

MySQL is an open-source relational database management system that is based on [SQL queries](). MySQL is used for data operations like querying, filtering, sorting, grouping, modifying, and joining the tables present in the database.

Read more: [MySQL and its working]()

Welcome to MySQL Workbench

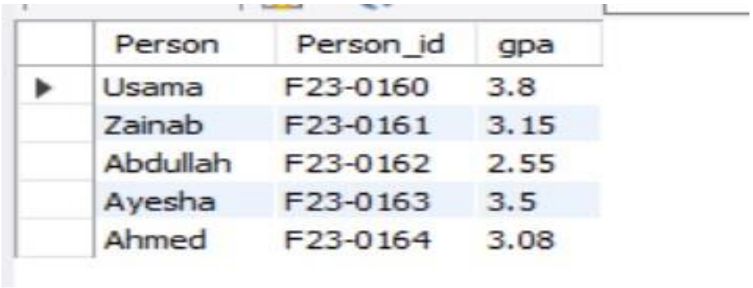
MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

Local instance wampmysqld64
root
localhost:3306

# Lab Report # 3

## Creating Database, Creating Table, Delete Database & Table and Using Constraints

```
show databases;

create database clsss;

use clsss;

create table student (

   Person varchar (230),

   Person_id varchar(255),

   gpa float);

select * from student;

insert into student (Person, Person_id,gpa) value ('Usama','F23-0160',3.80);

insert into student (Person, Person_id,gpa) value ('Zainab','F23-0161',3.15);

insert into student (Person, Person_id,gpa) value ('Abdullah','F23-0162',2.55);

insert into student (Person, Person_id,gpa) value ('Ayesha','F23-0163',3.50);

insert into student (Person, Person_id,gpa) value ('Ahmed','F23-0164',3.08);

select * from student;
```

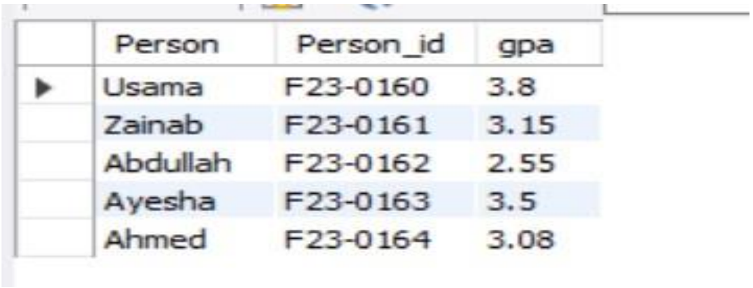


|  | Person | Person_id | gpa |
|---|---|---|---|
| ▶ | Usama | F23-0160 | 3.8 |
|  | Zainab | F23-0161 | 3.15 |
|  | Abdullah | F23-0162 | 2.55 |
|  | Ayesha | F23-0163 | 3.5 |
|  | Ahmed | F23-0164 | 3.08 |

# Lab Report # 4

## Creating Student Table

```
show databases;

create database clsss;

use clsss;

create table student (

   Person varchar (230),

   Person_id varchar(255),

   gpa float);

select * from student;

insert into student (Person, Person_id,gpa) value ('Usama','F23-0160',3.80);

insert into student (Person, Person_id,gpa) value ('Zainab','F23-0161',3.15);

insert into student (Person, Person_id,gpa) value ('Abdullah','F23-0162',2.55);

insert into student (Person, Person_id,gpa) value ('Ayesha','F23-0163',3.50);

insert into student (Person, Person_id,gpa) value ('Ahmed','F23-0164',3.08);

select * from student;
```

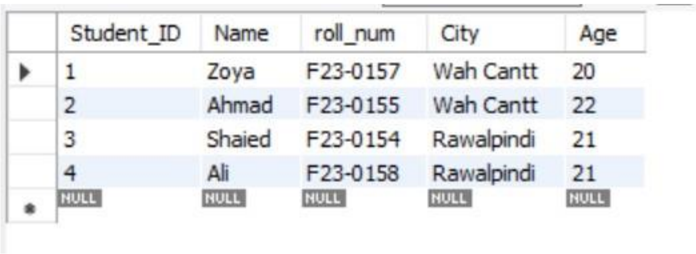


|  | Person | Person_id | gpa |
|---|---|---|---|
| ▶ | Usama | F23-0160 | 3.8 |
|  | Zainab | F23-0161 | 3.15 |
|  | Abdullah | F23-0162 | 2.55 |
|  | Ayesha | F23-0163 | 3.5 |
|  | Ahmed | F23-0164 | 3.08 |

## Concept of Primary and Foreigen Key

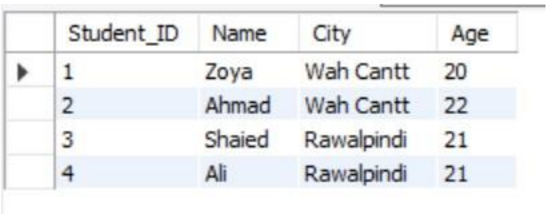
```
show databases;
create database clsssss;
use clsssss;
CREATE TABLE Students (
  Student_ID int AUTO_INCREMENT,
  Name varchar(50),
  roll_num varchar(230),
  City varchar(100),
  Age int,
  PRIMARY KEY (Student_ID)
);
CREATE TABLE Boys (
  Student_ID int AUTO_INCREMENT,
  Name varchar(50),
  City varchar(100),
  Age int,
  foreign key (Student_ID) references Students(Student_ID)
);

INSERT INTO Students (Name,roll_num, City, Age) VALUES ('Zoya','F23-0157', 'Wah Cantt', 20);
INSERT INTO Students (Name,roll_num, City, Age) VALUES('Ahmad','F23-0155', 'Wah Cantt', 22);
INSERT INTO Students (Name,roll_num, City, Age) VALUES ('Shaied','F23-0154', 'Rawalpindi', 21);
INSERT INTO Students (Name,roll_num, City, Age) VALUES ('Ali','F23-0158', 'Rawalpindi', 21);
SELECT * FROM Students;

INSERT INTO boys (Name, City, Age) VALUES ('Zoya', 'Wah Cantt', 20);
INSERT INTO boys (Name, City, Age) VALUES('Ahmad', 'Wah Cantt', 22);
INSERT INTO boys (Name, City, Age) VALUES ('Shaied', 'Rawalpindi', 21);
INSERT INTO boys (Name, City, Age) VALUES ('Ali', 'Rawalpindi', 21);

SELECT * FROM boys;
```

Student Table



| Student_ID | Name | roll_num | City | Age |
|---|---|---|---|---|
| 1 | Zoya | F23-0157 | Wah Cantt | 20 |
| 2 | Ahmad | F23-0155 | Wah Cantt | 22 |
| 3 | Shaied | F23-0154 | Rawalpindi | 21 |
| 4 | Ali | F23-0158 | Rawalpindi | 21 |
| NULL | NULL | NULL | NULL | NULL |

Boys Table:



| Student_ID | Name | City | Age |
|---|---|---|---|
| 1 | Zoya | Wah Cantt | 20 |
| 2 | Ahmad | Wah Cantt | 22 |
| 3 | Shaied | Rawalpindi | 21 |
| 4 | Ali | Rawalpindi | 21 |

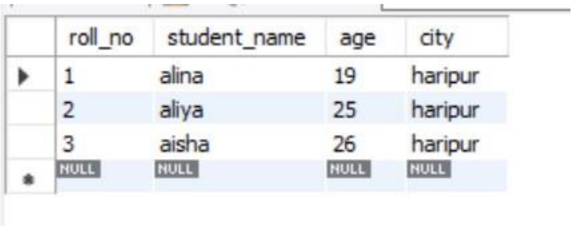
# Lab number #5

- **Select**
- **Select Data with Where Clause**
- **Where Comparison Operators**
- **Between Operator**

```
drop database if exists department;

create database department;

use department;

create table student(

roll_no int primary key,

student_name varchar (30),

age int,

city varchar (50)

);

insert into student (roll_no, student_name, age, city)

values (1, "alina", 19, "haripur");
```

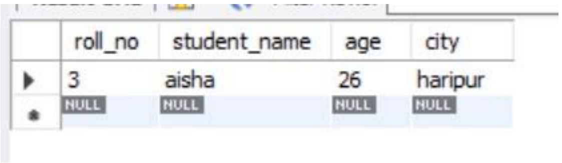
```
insert into student (roll_no, student_name, age, city)

values  (2, "aliya", 25, "haripur");

insert into student (roll_no, student_name, age, city)

values  (3, "aisha", 26, "haripur");

select * from student;

select roll_no as Roll_no, student_name as "Student Name " ,age as Age from student;

select * from student where age = '26';

select * from student where age<26;

select * from student where age<=26;

select * from student where age>26;

select * from student where city!= 'Haripur';

select * from student where city= 'Haripur';
```

```
select roll_no as Roll_no, student_name as "Student Name " ,age as Age from student;
```



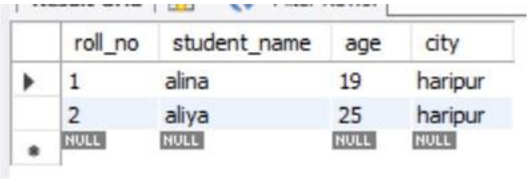
| roll_no | student_name | age | city |
|---|---|---|---|
| 1 | alina | 19 | haripur |
| 2 | aliya | 25 | haripur |
| 3 | aisha | 26 | haripur |
| NULL | NULL | NULL | NULL |

```
select * from student where age = '26';
```



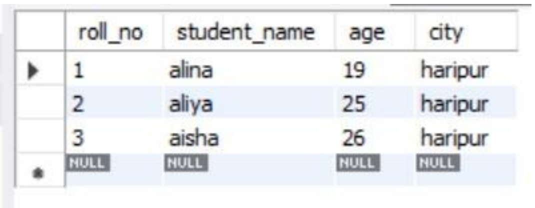
| roll_no | student_name | age | city |
|---|---|---|---|
| 3 | aisha | 26 | haripur |
| NULL | NULL | NULL | NULL |

```
select * from student where age<26;
```



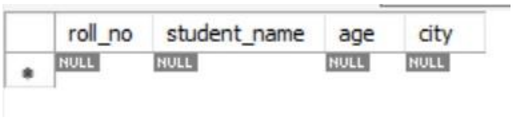
| roll_no | student_name | age | city |
|---|---|---|---|
| 1 | alina | 19 | haripur |
| 2 | aliya | 25 | haripur |
| NULL | NULL | NULL | NULL |

```
select * from student where age<=26;
```



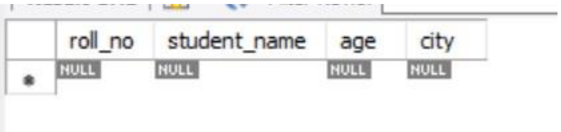
| roll_no | student_name | age | city |
|---|---|---|---|
| 1 | alina | 19 | haripur |
| 2 | aliya | 25 | haripur |
| 3 | aisha | 26 | haripur |
| NULL | NULL | NULL | NULL |

select * from student where age>26;



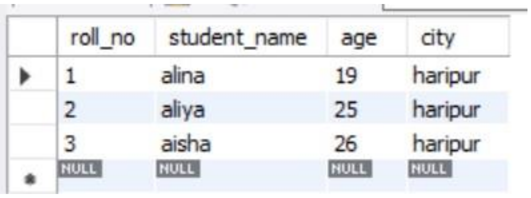
| roll_no | student_name | age | city |
|---|---|---|---|
| NULL | NULL | NULL | NULL |

select * from student where city!= 'Haripur';



| roll_no | student_name | age | city |
|---|---|---|---|
| NULL | NULL | NULL | NULL |

select * from student where city= 'Haripur';

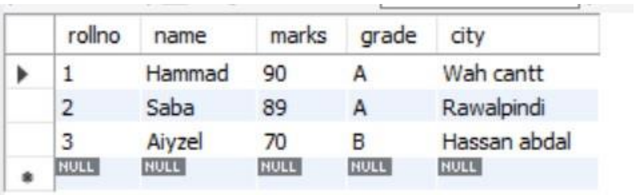


| roll_no | student_name | age | city |
|---|---|---|---|
| 1 | alina | 19 | haripur |
| 2 | aliya | 25 | haripur |
| 3 | aisha | 26 | haripur |
| NULL | NULL | NULL | NULL |

## Between Operator

```
drop database if exists college;

create database college;

use college;



create table student(

   rollno int primary key,

   name varchar(100),

   marks int not null,

   grade varchar(20),

   city varchar(70)

);

insert into student

(rollno,name,marks,grade,city)
```

```
values

(1,"Hammad",90,"A","Wah cantt"),

(2,"Saba",89,"A","Rawalpindi"),

(3,"Aiyzel",70,"B","Hassan abdal"),

(4,"Ubaid",65,"C","Haripur");

select * from student where marks between 70 and 90;
```



| | rollno | name | marks | grade | city |
|---|---|---|---|---|---|
| ▶ | 1 | Hammad | 90 | A | Wah cantt |
| | 2 | Saba | 89 | A | Rawalpindi |
| | 3 | Aiyzel | 70 | B | Hassan abdal |
| * | NULL | NULL | NULL | NULL | NULL |

# Lab Number # 6

# Distinct Function & Limit

```
drop database if exists departmentss;
create database departmentss;
use departmentss;
create table student(
roll_no int ,
student_name varchar (30),
age int,
city varchar (50)
);
insert into student (roll_no, student_name, age, city)
values (1, "alina", 19, "haripur");
insert into student (roll_no, student_name, age, city)
values  (2, "aliya", 25, "haripur");
insert into student (roll_no, student_name, age, city)
values  (3, "aisha", 26, "haripur");
select distinct age from student;
select roll_no ,age , city from student limit 3;
select roll_no ,age ,city from student where age=19 limit 3;
select roll_no ,age ,city from student where age>19 limit 1,3;
select roll_no ,age ,city from student limit 2,3;
select roll_no ,age ,city from student limit 0,4;
```

# Lab Number #07

## Project Report:

## Class GPA Table Management System

---

### 1. Project Title

**Class GPA Table Management System Using SQL**

---

### 2. Objective

The primary goal of this project is to design and implement a structured database system for managing and analyzing academic records of students, focusing on GPA calculation and performance evaluation.

---

### 3. Tools and Technology Used

- **SQL** (Structured Query Language)
- **MySQL / MariaDB** (or any compatible RDBMS)
- SQL queries executed via CLI or GUI tool (e.g., MySQL Workbench)
- Minimum 4GB RAM , 1 GB storage, Desktop or Laptop computer.

---

### 4. Database Design

**Database Name:** `Result`

**Table:** `student`

| Column Name | Data Type | Description |
|---|---|---|
| id | INT (Primary Key) | Unique identifier for each student |
| name | VARCHAR(200) | Student’s full name |
| Total_marks | INT | Total marks (default: 500) |
| Obtained_marks | INT | Marks obtained by the student |
| GPA | FLOAT | GPA corresponding to obtained marks |

---

### 5. Key Operations Performed

- **Database creation and selection**
- **Table creation with default values and primary key**
- **Insertion of 50 student records**
- **Selection queries for individual and group analysis**
- **Aggregate functions for max, min, average GPA**
- **Conditional filtering (e.g., GPA > 3.0)**
- **Record deletion (e.g., GPA < 1.5)**
- **Column modification using ALTER**
- **Subqueries for nested selection**

---

## 6. Sample Queries and Outcomes

- **List of all students:**
  `SELECT * FROM student;`
- **Top GPA Holder:**
  `SELECT id, name, GPA FROM student WHERE GPA = (SELECT MAX(GPA) FROM student);`
  ➤ *Abu Bakar with a GPA of 3.91*
- **Students with GPA > 3.0:**
  `SELECT * FROM student WHERE GPA > 3.0;`
  ➤ *Majority of students fall within this range.*
- **Average GPA of class:**
  `SELECT AVG(GPA) FROM student;`
  ➤ *Average GPA ≈ 2.97*
- **Deletion of poor performers:**
  `DELETE FROM student WHERE GPA < 1.5;`
  ➤ *Low-performing students removed for better insights.*

---

## 7. Results and Analysis

- The structured approach allowed for easy manipulation and analysis of student academic performance.
- Filtering based on GPA helped identify top and low-performing students.
- The average GPA gives a good indicator of overall class performance.
- Data integrity was maintained through primary keys and structured inputs.

---

## 8. Conclusion

This project demonstrates the practical use of SQL in managing student academic data efficiently. With structured queries, various analyses such as identifying top scorers, calculating averages, and managing student records become straightforward. The system can be scaled or integrated into a larger educational management system for broader use.

---

## 9. Future Enhancements

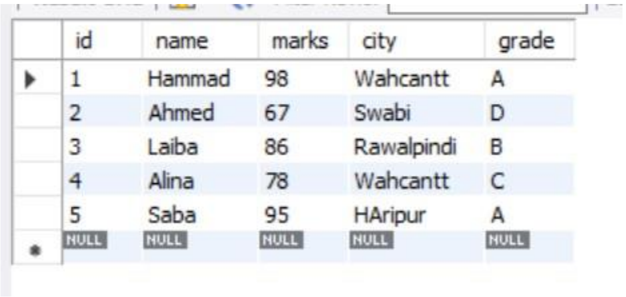
- Introduce **semester-wise breakdown** for more detailed analysis.
- Add tables for **courses**, **attendance**, or **extracurricular performance**.
- Build a **front-end interface** for real-time interaction using PHP/JavaScript.
- Generate **automated reports and visual dashboards**.

# Lab Number #8

**Commit & Rollback works for:**

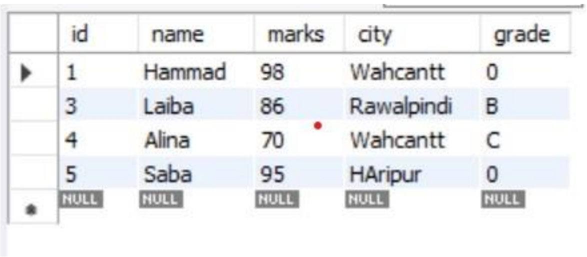
- **UPDATE**
- **INSERT**
- **DELETE**

```
drop database if exists functions;
create database functions;
use functions;
create table funrules(
 id int primary key,
 name varchar(100),
 marks int,
 city varchar(200),
 grade char(200)
);
insert into funrules
(id,name,marks,city,grade)
values
(1,"Hammad",98,"Wahcantt","A"),
(2,"Ahmed",67,"Swabi","D"),
(3,"Laiba",86,"Rawalpindi","B"),
(4,"Alina",78,"Wahcantt","C"),
(5,"Saba",95,"HAripur","A");
select * from funrules;
set sql_safe_updates =0;
update funrules
set grade="0"
where grade ="A";
update funrules set marks=70 where id=4;
delete from funrules where marks<70;
select * from funrules;


select * from funrules;
```



| | id | name | marks | city | grade |
|---|---|---|---|---|---|
| ▶ | 1 | Hammad | 98 | Wahcantt | A |
| | 2 | Ahmed | 67 | Swabi | D |
| | 3 | Laiba | 86 | Rawalpindi | B |
| | 4 | Alina | 78 | Wahcantt | C |
| | 5 | Saba | 95 | HAripur | A |
| * | NULL | NULL | NULL | NULL | NULL |

```
update funrules
set grade="0"
where grade ="A";
update funrules set marks=70 where id=4;
delete from funrules where marks<70;
select * from funrules;
```



| | id | name | marks | city | grade |
|---|---|---|---|---|---|
| ▶ | 1 | Hammad | 98 | Wahcantt | 0 |
| | 3 | Laiba | 86 | Rawalpindi | B |
| | 4 | Alina | 70 | Wahcantt | C |
| | 5 | Saba | 95 | HAripur | 0 |
| * | NULL | NULL | NULL | NULL | NULL |