



School of Computer Science, UPES, Dehradun.

A

LABORATORY FILE

On

DATABASE MANAGEMENT
SYSTEM (DBMS) LAB

B.TECH. -III Semester

AUG. – NOV.- 2024.

Submitted by:

Name: Harsh Vardhan Saini

SAP ID: 500120369

Roll No: R2142230056

Batch: 2

EXPERIMENT 03

MY SQL INSTALLATION AND COMMANDS

AIM:

To install MySQL and perform basic SQL commands.

Problem Statement:

- 1) Installation of MySQL Server, MySQL Shell and Workbench.
- 2) Building simple data table without key constraints.
- 3) To create a basic SQL program for a university database named "UPES," we need to consider the entities and
- 4) relationships that a university database might typically have. For a basic university database, we'll include
- 5) tables such as Students, Courses, Departments, Professors, and Enrollments.

THEORY:

Structured query language (SQL) is a programming language for storing and processing information in a relational database. A relational database stores information in tabular form, with rows and columns representing different data attributes and the various relationships between the data values.

Results:

- 1) If your computer does not have SQL Server installed: Go to web site below–
<https://www.mysql.com/downloads/>– And click on “MySQL Community (GPL) Downloads
=> MySQL Installer for Windows

MySQL Installer 8.0.39

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:
8.0.39

Select Operating System:
Microsoft Windows

Windows (x86, 32-bit), MSI Installer (mysql-installer-web-community-8.0.39.0.msi)	8.0.39	2.1M	Download
Windows (x86, 32-bit), MSI Installer (mysql-installer-community-8.0.39.0.msi)	8.0.39	303.6M	Download

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

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

Login »
using my Oracle Web account

Sign Up »
for an Oracle Web account

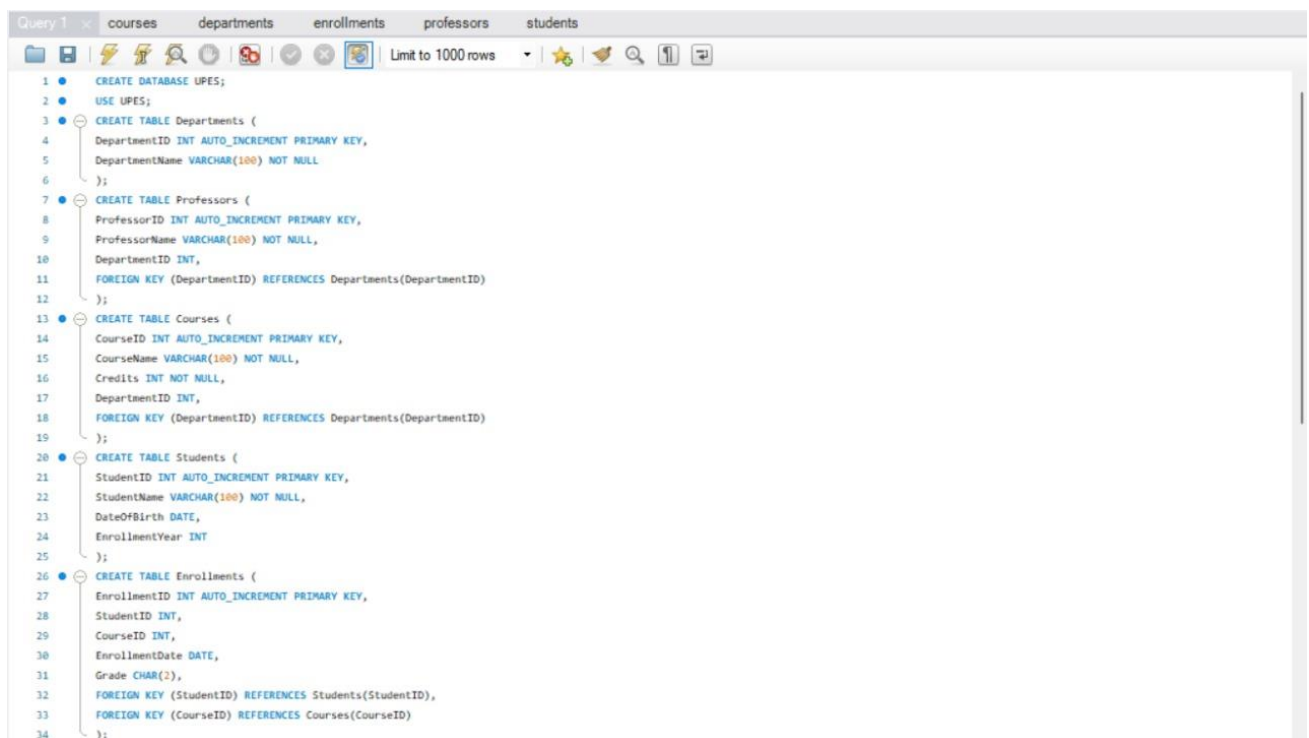
MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

No thanks, just start my download.

2) Building simple data table without key constrains.

```
1  -- Ayush Vashishth
2  -- 500119331
3
4  • CREATE DATABASE Hostel;
5  • USE Hostel;
6  • CREATE TABLE Person
7  (
8    PersonID int,
9    FirstName varchar(255),
10   Address varchar(255)
11  );
12 • insert into Hostel.Person(PersonID,FirstName,Address) values(1,'Ayush','xyz123');
13 • insert into Hostel.Person(PersonID,FirstName,Address) values(2,'Aryan','xyz456');
14 • insert into Hostel.Person(PersonID,FirstName,Address) values(3,'Shaurya','xyz789');
15 • insert into Hostel.Person(PersonID,FirstName,Address) values(4,'Abhinav','xyz025');
16 • SELECT * FROM Hostel.person;
```

3) Basic SQL program for a university database named "UPES,"



```
Query 1  courses  departments  enrollments  professors  students
Limit to 1000 rows

1  • CREATE DATABASE UPES;
2  • USE UPES;
3  • CREATE TABLE Departments (
4    DepartmentID INT AUTO_INCREMENT PRIMARY KEY,
5    DepartmentName VARCHAR(100) NOT NULL
6  );
7  • CREATE TABLE Professors (
8    ProfessorID INT AUTO_INCREMENT PRIMARY KEY,
9    ProfessorName VARCHAR(100) NOT NULL,
10   DepartmentID INT,
11   FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)
12  );
13 • CREATE TABLE Courses (
14   CourseID INT AUTO_INCREMENT PRIMARY KEY,
15   CourseName VARCHAR(100) NOT NULL,
16   Credits INT NOT NULL,
17   DepartmentID INT,
18   FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)
19  );
20 • CREATE TABLE Students (
21   StudentID INT AUTO_INCREMENT PRIMARY KEY,
22   StudentName VARCHAR(100) NOT NULL,
23   DateOfBirth DATE,
24   EnrollmentYear INT
25  );
26 • CREATE TABLE Enrollments (
27   EnrollmentID INT AUTO_INCREMENT PRIMARY KEY,
28   StudentID INT,
29   CourseID INT,
30   EnrollmentDate DATE,
31   Grade CHAR(2),
32   FOREIGN KEY (StudentID) REFERENCES Students(StudentID),
33   FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)
34  );
```

Query 1 x courses departments enrollments professors students

Limit to 1000 rows

```

33 FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)
34 );
35 -- Insert data into Departments
36 INSERT INTO Departments (DepartmentName) VALUES
37 ('Computer Science'),
38 ('Electrical Engineering'),
39 ('Mechanical Engineering');
40 -- Insert data into Professors
41 INSERT INTO Professors (ProfessorName, DepartmentID) VALUES
42 ('Dr. John Doe', 1),
43 ('Dr. Jane Smith', 2),
44 ('Dr. Alan Turing', 1),
45 ('Dr. Nikola Tesla', 2),
46 ('Dr. James Watt', 3);
47 -- Insert data into Courses
48 INSERT INTO Courses (CourseName, Credits, DepartmentID) VALUES
49 ('Database Systems', 4, 1),
50 ('Computer Networks', 3, 1),
51 ('Circuit Analysis', 3, 2),
52 ('Thermodynamics', 3, 3),
53 ('Electromagnetics', 3, 2);
54 -- Insert data into Students
55 INSERT INTO Students (StudentName, DateOfBirth, EnrollmentYear) VALUES
56 ('Alice Johnson', '2000-05-15', 2018),
57 ('Bob Brown', '1999-08-22', 2017),
58 ('Charlie Davis', '2001-02-10', 2019),
59 ('Daisy Clark', '2000-11-30', 2018);
60 -- Insert data into Enrollments
61 INSERT INTO Enrollments (StudentID, CourseID, EnrollmentDate, Grade) VALUES
62 (1, 1, '2018-09-01', 'A'),
63 (1, 2, '2018-09-01', 'B'),
64 (2, 3, '2017-09-01', 'A'),
65 (3, 4, '2019-09-01', 'C'),
66 (4, 5, '2018-09-01', 'B');

```

Query 1 courses x

Limit to 1000 rows

```

1 SELECT * FROM upes.courses;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	CourseID	CourseName	Credits	DepartmentID
▶	1	Database Systems	4	1
	2	Computer Networks	3	1
	3	Circuit Analysis	3	2
	4	Thermodynamics	3	3
	5	Electromagnetics	3	2
•	NULL	NULL	NULL	NULL

Result Grid
Form Editor
Field Types

Query 1 courses departments

Limit to 1000 rows

```
1 • SELECT * FROM upes.departments;
```

Result Grid

	DepartmentID	DepartmentName
▶	1	Computer Science
	2	Electrical Engineering
	3	Mechanical Engineering
*	NULL	NULL

Result Grid
Form Editor

Query 1 courses departments enrollments

Limit to 1000 rows

```
1 • SELECT * FROM upes.enrollments;
```

Result Grid

	EnrollmentID	StudentID	CourseID	EnrollmentDate	Grade
▶	1	1	1	2018-09-01	A
	2	1	2	2018-09-01	B
	3	2	3	2017-09-01	A
	4	3	4	2019-09-01	C
	5	4	5	2018-09-01	B
*	NULL	NULL	NULL	NULL	NULL

Result Grid
Form Editor
Field Types

Query 1 courses departments enrollments **professors** x

Limit to 1000 rows

1 • `SELECT * FROM upes.professors;`

Result Grid

	ProfessorID	ProfessorName	DepartmentID
▶	1	Dr. John Doe	1
	2	Dr. Jane Smith	2
	3	Dr. Alan Turing	1
	4	Dr. Nikola Tesla	2
	5	Dr. James Watt	3
*	NULL	NULL	NULL

Result Grid
Form Editor

4)

Query 1 courses departments enrollments professors **students** x

Limit to 1000 rows

1 • `SELECT * FROM upes.students;`

Result Grid

	StudentID	StudentName	DateOfBirth	EnrollmentYear
▶	1	Alice Johnson	2000-05-15	2018
	2	Bob Brown	1999-08-22	2017
	3	Charlie Davis	2001-02-10	2019
	4	Daisy Clark	2000-11-30	2018
*	NULL	NULL	NULL	NULL

Result Grid
Form Editor

Conclusion:

In this exercise, we installed MySQL Server, MySQL Shell, and Workbench, essential tools for managing relational databases. We created a simple data table without key constraints and designed a basic university database with entities like Students, Courses, and Professors. This practical experience provided foundational knowledge of SQL and relational databases, highlighting their importance in data management and real-world applications.