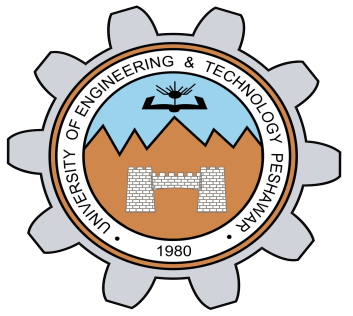
LAB REPORT NO 04

DATABASE DEVELOPMENT USING MYSQL



**CSE-403L**

**Data Base Management System Lab**

**Submitted By: Naveed Ahmad**

**Registration No: 22PWCSE2165**

**Section: B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work”

Signature: \_\_\_\_\_\_\_\_

March 10, 2025

**Submitted To: Engr. Sumayyea Salahuddin**

*Department of Computer Systems Engineering University of Engineering and Technology*

**DATABASE DEVELOPMENT USING MYSQL:**

**OBJECTIVE**

In this lab, we focus on understanding:

* What MySQL is and how it works as an example of an SQL Database Management System.
* How to use the MySQL Command Line on Windows.
* The different categories of MySQL commands.
* How to apply these commands to solve real-world problems.

**TASK 4.1: LAB PERFORMANCE**

**What are DDL, DML, TCL, and DCL? Explain in your own words. Also, list a few commands in each category.**

When working with databases in SQL, there are different types of commands we use depending on what we need to do. Here’s a simple breakdown:

**DDL (Data Definition Language):**

* These commands help you define and change the structure of your database—things like creating tables or deleting them.
* **Common DDL commands:**
* CREATE: Makes a new table or database.
* ALTER: Modifies an existing table (like adding a column).
* DROP: Deletes a table or a database entirely.

**DML (Data Manipulation Language):**

* DML is all about the data itself. These commands let you add, edit, or delete the actual records inside your tables.
* **Common DML commands:**
* SELECT: Pulls data from the database.
* INSERT: Adds new data (records) into a table.
* UPDATE: Changes existing data in a table.
* DELETE: Removes records from a table.

**TCL (Transaction Control Language):**

* These commands manage transactions—basically groups of operations that are treated as a single unit.
* **Common TCL commands:**
* COMMIT: Saves changes made by a transaction permanently.
* ROLLBACK: Undoes changes made in a transaction that haven’t been saved yet.
* SAVEPOINT: Lets you set a point in a transaction to which you can later roll back.
* **DCL (Data Control Language):**
* DCL is about controlling access to the database. You can allow or restrict users from doing certain things.
* **Common DCL commands:**
* GRANT: Gives users permission to do things (like read or update data).
* REVOKE: Removes a user’s access permissions.

**Hands-On Work**

* All the commands we covered in the lab have already been done and saved in the .rar file. Shown to the lab instructor

**TASK 4.2: DIFFERENCE BETWEEN SQL AND MYSQL + WHY MYSQL?**

**SQL vs MySQL**

* **SQL (Structured Query Language)** is the standard language we use to interact with databases. You can think of it like the “language” used to ask questions or give instructions to any database.
* **MySQL** is a database management system (DBMS). It uses SQL as its language to store, manage, and retrieve data. It’s kind of like a specific app that understands SQL and lets you manage your data efficiently.

**Why Use MySQL?**

* It’s **open-source**, which means it’s free and customizable.
* It works on **multiple platforms** (Windows, Linux, etc.).
* It’s **reliable**, **secure**, and handles **large databases** really well.
* It’s **fast** and supports **complex queries**.
* It also offers **full transactional support**, meaning you can roll back changes if something goes wrong.

**TASK 4.3: DATABASE ENGINE IN MYSQL**

* **What’s a Database Engine?**
  + A database engine is like the core part of the DBMS. It’s what actually stores, processes, and secures your data behind the scenes.
  + Without the engine, the database is just an empty shell.
* **How Many Engines Does MySQL Support?**
* MySQL supports several storage engines, each designed for different needs:
  + **InnoDB**
  + **MyISAM**
  + **Memory**
  + **CSV**
  + **Archive**

**Which is Most Common & Why?**

* + **InnoDB** is the most widely used engine today. Why?
  + It supports **ACID** properties (ensuring reliable transactions).
  + It allows **foreign keys**, which help maintain **referential integrity** between tables.
  + It offers **row-level locking**, improving performance when many users are accessing the data at once.

**TASK 4.4: MYSQL DATA TYPES (15+ EXAMPLES WITH EXPLANATIONS)**

Here’s a list of different data types MySQL supports, along with what they’re used for and an example for each:

* **1. INT**: Stores whole numbers. *Example:* age INT;
* 2. **BIGINT**: Stores very large integers. *Example:* population BIGINT;
* 3. **TINYINT**: Stores very small integers. *Example:* level TINYINT;
* 4. **SMALLINT**: Stores small integers. *Example:* steps SMALLINT;
* 5. **MEDIUMINT**: Stores medium-range integers. *Example:* followers\_count MEDIUMINT;
* 6. **FLOAT**: Stores numbers with decimals (less precise). *Example:* score FLOAT;
* 7. **DOUBLE**: Stores numbers with decimals (more precise than FLOAT) *Example:* balance DOUBLE;
* 8. **DECIMAL**: Stores exact numeric values (great for money). *Example:* price DECIMAL(10,2);
* 9. **BIT**: Stores binary values. *Example:* permissions BIT(8);
* 10. **CHAR**: Stores fixed-length text. *Example:* gender CHAR(1);
* 11. **VARCHAR**: Stores variable-length text. *Example:* name VARCHAR(255);
* 12. **TEXT**: Stores long blocks of text. *Example:* description TEXT;
* 13. **DATE**: Stores dates. *Example:* birthdate DATE;
* 14. **DATETIME**: Stores date and time. *Example:* created\_at DATETIME;
* 15. **TIMESTAMP**: Stores date and time, auto-updated on changes. *Example:* last\_updated TIMESTAMP;
* 16. **BOOLEAN**: Stores TRUE/FALSE (really just tiny integers). *Example:* is\_active BOOLEAN;
* 17. **ENUM**: Stores a single value from a predefined list. *Example:* status ENUM('new', 'pending', 'complete');
* 18. **SET**: Stores zero or more values from a predefined list. *Example:* features SET('GPS', 'Bluetooth');
* 19. **BLOB**: Stores binary data like images or files. *Example:* file BLOB;