Experiment Number: 01

Experiment Name: Database Management with MySQL: DDL and DML Operations

Objectives:

To create a database and table for student information, modify table structures, add and update columns, and delete records based on criteria.

Theory:

MySQL is a widely used open-source relational database management system (RDBMS) known for its reliability, efficiency, and ability to manage large datasets. It uses Structured Query Language (SQL) to interact with databases, making it a popular choice for web applications. A fundamental aspect of working with MySQL, or any database, is understanding the distinction between Data Definition Language (DDL) and Data Manipulation Language (DML) commands. This distinction is crucial as it affects how data is stored, accessed, and modified within the database.

Data Definition Language (DDL)

DDL commands are used to define and modify the structure of a database. They are essential for creating and managing databases and tables. Key DDL commands include:

- CREATE: Used to create new databases or tables.
- ALTER: Used to modify the structure of an existing table, such as adding or modifying columns.
- RENAME: Changes the name of a table or column.
- DROP: Deletes an entire table or database.
- TRUNCATE: Removes all data from a table but keeps the table structure.

Data Manipulation Language (DML)

DML commands manipulate the data within tables, such as adding, modifying, or deleting records.

Key DML commands include:

- INSERT: Adds new data into a table.
- UPDATE: Modifies existing records in a table.
- DELETE: Removes specific records from a table based on a condition.
- WHERE: Filters records for SELECT, UPDATE, or DELETE operations.
- MODIFY: Alters the data type of a column.
- SET: Used in UPDATE statements to change values based on conditions.

Tables in MySQL are fundamental structures for storing data. Creating and manipulating tables involves

defining their structure and managing the data they hold through various SQL commands.

Problem Statements:

1. Create a database and a table for storing student information.
2. Modify the table by changing a column's name and data type.
3. Add a new column to the table and set values based on conditions.
4. Delete records where student marks are below a specified threshold.

Software Used:

1. XAMPP

Problem 1: Create Database and Table

SQL Commands:

```
CREATE TABLE students (
    Roll INT PRIMARY KEY,
    Name VARCHAR(100),
    Semester INT,
    Major_Subject VARCHAR(100),
    Obtained_Marks INT
);

INSERT INTO students (Roll, Name, Semester, Major_Subject, Obtained_Marks)
VALUES
(1, 'Olivia Parker', 1, 'English Literature', 88),
(2, 'Liam Johnson', 2, 'History', 74),
(3, 'Emma Martinez', 1, 'Mathematics', 95),
(4, 'Noah Davis', 3, 'Chemistry', 82),
(5, 'Ava Garcia', 2, 'Physics', 67),
(6, 'Sophia Anderson', 4, 'Computer Science', 90),
(7, 'Mason Lee', 1, 'Biology', 79),
(8, 'Isabella Thompson', 3, 'Economics', 91),
(9, 'Lucas Martinez', 4, 'History', 85),
(10, 'Mia Hernandez', 2, 'Philosophy', 58);

SELECT * FROM students;
```

Output:

	Roll	Name	Semester	Major_Subject	Obtained_Marks
Edit Copy Delete	1	Olivia Parker	1	English Literature	88
Edit Copy Delete	2	Liam Johnson	2	History	74
Edit Copy Delete	3	Emma Martinez	1	Mathematics	95
Edit Copy Delete	4	Noah Davis	3	Chemistry	82
Edit Copy Delete	5	Ava Garcia	2	Physics	67
Edit Copy Delete	6	Sophia Anderson	4	Computer Science	90
Edit Copy Delete	7	Mason Lee	1	Biology	79
Edit Copy Delete	8	Isabella Thompson	3	Economics	91
Edit Copy Delete	9	Lucas Martinez	4	History	85
Edit Copy Delete	10	Mia Hernandez	2	Philosophy	58

Problem 2: Change Column Name and Data Type

SQL Command:

```sql
-- Rename 'Name' column to 'Student_Name'
ALTER TABLE students CHANGE COLUMN Name Student_Name VARCHAR(120);

-- Rename 'Semester' column to 'Academic_Year' and change data type to SMALLINT
ALTER TABLE students CHANGE COLUMN Semester Academic_Year SMALLINT;
```

Output:

	Roll	Student_Name	Academic_Year	Major_Subject	Obtained_Marks
Edit Copy Delete	1	Olivia Parker	1	English Literature	88
Edit Copy Delete	2	Liam Johnson	2	History	74
Edit Copy Delete	3	Emma Martinez	1	Mathematics	95
Edit Copy Delete	4	Noah Davis	3	Chemistry	82
Edit Copy Delete	5	Ava Garcia	2	Physics	67
Edit Copy Delete	6	Sophia Anderson	4	Computer Science	90
Edit Copy Delete	7	Mason Lee	1	Biology	79
Edit Copy Delete	8	Isabella Thompson	3	Economics	91
Edit Copy Delete	9	Lucas Martinez	4	History	85
Edit Copy Delete	10	Mia Hernandez	2	Philosophy	58

Problem 3: Add a New Column and Set Values Based on Condition

SQL Commands:

```sql
-- Add a new column named 'Status'
ALTER TABLE students ADD Status VARCHAR(25);

-- Update 'Status' column based on the condition that marks are 90 or above
UPDATE students SET Status = 'High Achiever' WHERE Obtained_Marks >= 90;

-- Update 'Status' column for marks between 60 and 89
UPDATE students SET Status = 'Satisfactory' WHERE Obtained_Marks >= 60 AND Obtained_Marks < 90;

-- Update 'Status' column for marks below 60
UPDATE students SET Status = 'Needs Improvement' WHERE Obtained_Marks < 60;
```

Output:

	Roll	Student_Name	Academic_Year	Major_Subject	Obtained_Marks	Status
Edit Copy Delete	1	Olivia Parker	1	English Literature	88	Satisfactory
Edit Copy Delete	2	Liam Johnson	2	History	74	Satisfactory
Edit Copy Delete	3	Emma Martinez	1	Mathematics	95	High Achiever
Edit Copy Delete	4	Noah Davis	3	Chemistry	82	Satisfactory
Edit Copy Delete	5	Ava Garcia	2	Physics	67	Satisfactory
Edit Copy Delete	6	Sophia Anderson	4	Computer Science	90	High Achiever
Edit Copy Delete	7	Mason Lee	1	Biology	79	Satisfactory
Edit Copy Delete	8	Isabella Thompson	3	Economics	91	High Achiever
Edit Copy Delete	9	Lucas Martinez	4	History	85	Satisfactory
Edit Copy Delete	10	Mia Hernandez	2	Philosophy	58	Needs Improvement

Problem 4: Delete Student Records Where Marks Are Below 30

SQL Commands:

```sql
-- Delete student records where Obtained_Marks are below 30
DELETE FROM students WHERE Obtained_Marks < 30;
```

Output:

	Roll	Student_Name	Academic_Year	Major_Subject	Obtained_Marks	Status
Edit Copy Delete	1	Olivia Parker	1	English Literature	88	Satisfactory
Edit Copy Delete	2	Liam Johnson	2	History	74	Satisfactory
Edit Copy Delete	3	Emma Martinez	1	Mathematics	95	High Achiever
Edit Copy Delete	4	Noah Davis	3	Chemistry	82	Satisfactory
Edit Copy Delete	5	Ava Garcia	2	Physics	67	Satisfactory
Edit Copy Delete	6	Sophia Anderson	4	Computer Science	90	High Achiever
Edit Copy Delete	7	Mason Lee	1	Biology	79	Satisfactory
Edit Copy Delete	8	Isabella Thompson	3	Economics	91	High Achiever
Edit Copy Delete	9	Lucas Martinez	4	History	85	Satisfactory
Edit Copy Delete	10	Mia Hernandez	2	Philosophy	58	Needs Improvement

Discussion:

This lab covered fundamental MySQL operations, including creating and modifying tables and managing data. We practiced renaming columns, altering data types, and updating records. To remove all records from a table while preserving its structure, we would use the `TRUNCATE TABLE students` command. On the other hand, to completely delete the table along with all its data, we would use the `DROP TABLE students` command. The `TRUNCATE` command is efficient for quickly clearing out a table, whereas `DROP` removes the table and its structure entirely.

References

1.MySQL Documentation - MySQL 8.0 Reference Manual. Retrieved from https://dev.mysql.com/doc/refman/8.0/en/.

2.W3Schools - SQL Tutorial. Retrieved from https://www.w3schools.com/sql/.

3.GeeksforGeeks - SQL | Truncate vs Delete. Retrieved from https://www.geeksforgeeks.org/sql-truncate-vs-delete/.