Heaven's Light is Our Guide

Rajshahi University of Engineering & Technology

Course Title

Data Base Systems Sessional

Course No.

ECE 2216

Lab Report

01

Submitted By
Anirban Sarkar
Roll no: 2110060
Reg no: 1114/2021-22
Department of ECE, RUET

Submitted To
Oishi Jyoti
Assistant Professor,
Department of ECE, RUET

Experiment Number: 01

Experiment Name: DDL and DML commands with MySQL

Theory:

MySQL is a widely used open-source relational database management system (RDBMS) that allows for the storage, management, and retrieval of data. It relies on SQL (Structured Query Language) to interact with databases and is a popular choice for web applications due to its reliability, efficiency, and capacity to handle large datasets. Understanding the difference between DDL and DML commands is essential for anyone working with databases, as it affects how data is stored, accessed, and modified.

DDL commands are utilized to define and modify the structure of a database. They are crucial for creating and managing databases and tables. Some important DDL commands include:

- **CREATE:** Creates new databases or tables.
- **ALTER:** Modifies the structure of an existing table, such as adding or changing columns.
- **RENAME:** Renames a table or column.
- **DROP:** Removes an entire table or database.
- **TRUNCATE:** Clears all data from a table while preserving its structure.

DML commands are used to manipulate the data within tables, including adding, updating, or deleting records. Important DML commands include:

- **INSERT:** Inserts new data into a table.
- **UPDATE:** Updates existing records in a table.
- **DELETE:** Deletes specific records from a table based on a condition.
- **WHERE:** Filters records for use in SELECT, UPDATE, or DELETE operations.
- **MODIFY:** Changes the data type of a column.
- **SET:** Used in UPDATE statements to modify values under specific conditions.

1. Create a database and a table for storing information of 10 students.
2. Change a specific column name and data type.
3. Add a new column named “Log”. Set the value to “Applicable” or “Not Applicable” based on the condition (marks < 30).
4. Delete the student information for those whose marks are below 30.

Software Used:

1. XAMPP (used for running MySQL locally through the Apache server)

Problem 1: Create a database and a table for storing information of 10 students

SQL Commands:

```
CREATE DATABASE Student;

USE Student;

CREATE TABLE Students (
    Roll INT,
```

```
    Name VARCHAR(50),
    Semester INT,
    Major VARCHAR(50),
    Obtained_Marks FLOAT
);

INSERT INTO Students (Roll, Name, Semester, Major, Obtained_Marks) VALUES
(1, 'Aisha', 2, 'Mechanical Engineering', 28),
(2, 'Rahul', 3, 'Mathematics', 25),
(3, 'Zara', 1, 'Physics', 72),
(4, 'Adnan', 3, 'Mathematics', 29),
(5, 'Emily', 2, 'Civil Engineering', 60),
(6, 'Daniel', 2, 'Physics', 67),
(7, 'Fatima', 3, 'Mechanical Engineering', 40),
(8, 'Leo', 2, 'Civil Engineering', 30),
(9, 'Noah', 1, 'Physics', 50),
(10, 'Sophia', 2, 'Mathematics', 65);
```

Output:

Roll	Name	Semester	Major	Obtained_Marks
1	Aisha	2	Mechanical Engineering	28
2	Rahul	3	Mathematics	25
3	Zara	1	Physics	72
4	Adnan	3	Mathematics	29
5	Emily	2	Civil Engineering	60
6	Daniel	2	Physics	67
7	Fatima	3	Mechanical Engineering	40
8	Leo	2	Civil Engineering	30
9	Noah	1	Physics	50
10	Sophia	2	Mathematics	65

Figure 1: Database and table creation output

Problem 2: Change a specific column name and data type

SQL Commands:

```
ALTER TABLE Students CHANGE COLUMN Major Fav_Subject VARCHAR(50);

ALTER TABLE Students MODIFY Obtained_Marks DOUBLE;
```

Output:

Roll	Name	Semester	Fav_Subject	Obtained_Marks
1	Aisha	2	Mechanical Engineering	28
2	Rahul	3	Mathematics	25
3	Zara	1	Physics	72
4	Adnan	3	Mathematics	29
5	Emily	2	Civil Engineering	60
6	Daniel	2	Physics	67
7	Fatima	3	Mechanical Engineering	40
8	Leo	2	Civil Engineering	30
9	Noah	1	Physics	50
10	Sophia	2	Mathematics	65

Figure 2: Column name change output

	#	Name	Type
☐	1	**Roll**	int(11)
☐	2	**Name**	varchar(50)
☐	3	**Semester**	int(11)
☐	4	**Fav_Subject**	varchar(50)
☐	5	**Obtained_Marks**	double

Figure 3: Data type change output

Problem 3: Add a new column named “Log”. Set the value to “Applicable” or “Not Applicable” based on the condition (marks $<$ 30)

SQL Commands:

```
ALTER TABLE Students ADD Log VARCHAR(20);

UPDATE Students SET Log = 'Applicable' WHERE Obtained_Marks < 30;

UPDATE Students SET Log = 'Not Applicable' WHERE Obtained_Marks >= 30;
```

Output:

Roll	Name	Semester	Fav_Subject	Obtained_Marks	Log
1	Aisha	2	Mechanical Engineering	28	Applicable
2	Rahul	3	Mathematics	25	Applicable
3	Zara	1	Physics	72	Not Applicable
4	Adnan	3	Mathematics	29	Applicable
5	Emily	2	Civil Engineering	60	Not Applicable
6	Daniel	2	Physics	67	Not Applicable
7	Fatima	3	Mechanical Engineering	40	Not Applicable
8	Leo	2	Civil Engineering	30	Not Applicable
9	Noah	1	Physics	50	Not Applicable
10	Sophia	2	Mathematics	65	Not Applicable

Figure 4: Output after adding and updating the 'Log' column

Problem 4: Delete the student information for those whose marks are below 30

SQL Commands:

```
DELETE FROM Students WHERE Obtained_Marks < 30;
```

Output:

Roll	Name	Semester	Fav_Subject	Obtained_Marks	Log
3	Zara	1	Physics	72	Not Applicable
5	Emily	2	Civil Engineering	60	Not Applicable
6	Daniel	2	Physics	67	Not Applicable
7	Fatima	3	Mechanical Engineering	40	Not Applicable
8	Leo	2	Civil Engineering	30	Not Applicable
9	Noah	1	Physics	50	Not Applicable
10	Sophia	2	Mathematics	65	Not Applicable

Figure 5: Output after deleting records with marks below 30

Discussion:

In this experiment, we explored both DDL (Data Definition Language) and DML (Data Manipulation Language) commands in MySQL to understand their roles in managing and manipulating databases. The theory outlined the significance of DDL commands in creating, modifying, and structuring databases and tables, whereas DML commands are used for inserting, updating, and deleting data within those tables. The first set of commands (Code 1) demonstrated how to create a new database (`Student`) and table (`Students`), followed by inserting multiple records. This highlights the use of DDL to define the structure of the database and DML to populate it with data. Furthermore, the table structure defined fields like `Roll`, `Name`, `Semester`, `Major`, and `Obtained Marks`, which are crucial for identifying students and their performance.

The subsequent codes focused on modifying the table's structure and data using a combination of DDL and DML commands. Code 2 used DDL commands to alter the column name `Major` to `Fav_Subject` and change the data type of `Obtained_Marks` from `FLOAT` to `DOUBLE`. This exemplifies how DDL commands help maintain and evolve the database structure. Code 3 and Code 4 applied DML commands to update data in the table and manage records based on conditions. For example, the `Log` column was added, and its values were updated based on the students' marks, demonstrating

how DML commands modify existing records. Finally, records of students with marks below 30 were deleted in Code 4, showing the power of DML in removing irrelevant or unwanted data. Together, these codes provide a comprehensive understanding of how DDL and DML work in tandem to manage database structure and data efficiently.

References:

[1] C. J. Date, H. Darwen, and N. Lorentzos, "A Guide to the SQL Standard," 4th ed. Reading, MA, USA: Addison-Wesley, 2001.

[2] S. Prasad, "Mastering MySQL for Web Application Development," *Journal of Web and Database Technologies*, vol. 15, no. 3, pp. 45–57, June 2020.

[3] T. Ramakrishnan, "MySQL Database Management: Performance Tuning and Security," in *Proceedings of the IEEE International Conference on Database Systems and Applications*, New York, NY, USA, 2019, pp. 142–150.

[4] M. Widenius and D. Axmark, "MySQL Reference Manual," MySQL AB, 2002. [Online]. Available: https://dev.mysql.com/doc/. [Accessed: Sept. 22, 2024].