

Database Systems Laboratory

B.Tech. 5th Semester



Name : K Srikanth

Roll Number : 17ETCS002124

Department : Computer Science and Engineering

Faculty of Engineering & Technology

Ramaiah University of Applied Sciences



Ramaiah University of Applied Sciences

Private University Established in Karnataka State by Act No. 15 of 2013

Faculty	Engineering & Technology
Programme	B. Tech. in Computer Science and Engineering
Year/Semester	3 rd Year / 5 th Semester
Name of the Laboratory	Database Systems Laboratory
Laboratory Code	19CSL306A

List of Experiments

1. DDL and DML commands
2. Requirement analysis and data modelling
3. Data model to relational model
4. Data constraints and built in functions
5. Java database programming
6. Interface to the system
7. Nested queries and Join queries
8. Procedure in MySQL
9. Multidimensional data modelling

Index Sheet

No.	Lab Experiment	(a) Performing the experiment (20)	(b) Document (10)	(c) Viva (10)
1	DDL and DML commands			
2	Requirement analysis and data modelling			
3	Data model to relational model			
4	Data constraints and built in functions			
5	Java database programming			
6	Interface to the system			
7	Nested queries and Join queries			
8	Procedure in MySQL			
9	Multidimensional data modelling			
	Total Marks (Average of 9 labs)			
	(d) Lab Internal Test conducted along the lines of SEE, valued for 50 Marks and reduced to 10 Marks			
	Lab Internal Marks (50) (a+b+c+d)			
	Lab Internal Marks (25)			

Signature of the Staff In-charge

Laboratory 1

Title of the Laboratory Exercise: DDL and DML commands

1. Introduction and Purpose of Experiment

Structured Query Language (SQL) is used to pass the query to retrieve and manipulate the information from database. Depending upon the nature of query, SQL is divided into different components such as Data Definition Language (DDL) and Data Manipulation Language (DML). DDL statements create the database, maintain the structure of the database and remove database objects such as tables, indexes, and users. DML statements are used for managing data in database such as insert tuples into, delete tuples from, and modify tuples in the database. By doing this lab, students will be able to execute DDL and DML commands

2. Aim and Objectives

Aim

- To execute Data Definition Language (DDL) and Data Management Language (DML) commands

Objectives

At the end of this lab, the student will be able to

- Create a database and populate it with data using SQL commands
- Execute DDL and DML commands for the given database

3. Experimental Procedure

- Analyse the problem statement
- Execute DDL and DML commands
- Create a database for the given schema
- Design SQL commands using DDL and DML commands
- Test the executed commands
- Analyse and discuss the outcomes of your experiment
- Document the work

4. Questions

- Practice DDL and DML commands
- Consider the following relational schema that keeps track of the students in a college. Enter at least five tuples for the relation. Assume appropriate domain and data type for each field.

STUDENT (StudId, StudName, StudAddress)

Execute the following queries based on the above schema

- Display the details of all the students
- Display the name and address of the student with id=101

- iii. Insert a new student <105, 'John', 'Bangalore'>
- iv. Change the address of the student John to 'Delhi'
- v. Delete the details of a student with student id=105
- vi. Add a column to the schema Student with appropriate data type

5. Presentation of Results

Q-A) Practice DDL and DML commands

DDL Commands

- Creating a database

```
mysql> create database lab1;
Query OK, 1 row affected (0.01 sec)

mysql> use lab1;
Database changed
```

Figure 1 MySQL Command to Creating a database

We are creating a new database "lab1" and then we are using it.

- Describing a database

```
mysql> create table student(
-> Student_ID INT auto_increment,
-> Student_Name varchar(20),
-> Student_Address varchar(20),
-> primary key(Student_ID));
Query OK, 0 rows affected (0.01 sec)

mysql> describe student;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| Student_ID     | int           | NO   | PRI | NULL    | auto_increment |
| Student_Name   | varchar(20)   | YES  |     | NULL    |                |
| Student_Address | varchar(20)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Figure 2 MySQL Command to Describing Database

Firstly we need to create a table to describe a database, so we create table "student" with a schema then we can say describe student.

- **Renaming a table inside database**

```
[mysql> rename table student to class;
Query OK, 0 rows affected (0.03 sec)
```

Figure 3 MySQL Command to Renaming a table inside database

We are renaming our current table from student to class.

- **Deleting a table inside database**

```
[mysql> drop table class;
Query OK, 0 rows affected (0.01 sec)
```

Figure 4 MySQL Command to delete a table inside database

We are deleting our table "class".

- **Modifying a table inside database**

1. **Adding New Columns**

```
[mysql> alter table student add Student_Subject varchar(20);
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0

[mysql> describe student;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| Student_ID     | int           | NO   | PRI | NULL    | auto_increment |
| Student_Name   | varchar(20)   | YES  |     | NULL    |                |
| Student_Address | varchar(20)   | YES  |     | NULL    |                |
| Student_Subject | varchar(20)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

Figure 5 MySQL Command to add new column in table.

We are adding a new column "Student_Subject" type "varchar(40)" into our existing table.

2. **Dropping a columns from a table**

```
[mysql> alter table student drop Student_Subject;
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Figure 6 MySQL Command to delete a column.

We are deleting an existing column "Student_Subject" from our table.

3. Modifying Existing Columns

```
[mysql> alter table student modify Student_Address varchar(40);
Query OK, 0 rows affected (0.00 sec)
Records: 0 Duplicates: 0 Warnings: 0

[mysql> describe student;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| Student_ID     | int           | NO   | PRI | NULL    | auto_increment |
| Student_Name   | varchar(20)   | YES  |     | NULL    |                |
| Student_Address | varchar(40)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

Figure 7 MySQL Command to Modify an existing column.

We are increasing size of data type of existing column “Student_Address” to varchar(40).

- Deleting contents inside a database

```
[mysql> insert into student(Student_Name,Student_Address) values('K Srikanth','Bangalore');
Query OK, 1 row affected (0.01 sec)

[mysql> select * from student;
+-----+-----+-----+
| Student_ID | Student_Name | Student_Address |
+-----+-----+-----+
| 1 | K Srikanth | Bangalore |
+-----+-----+-----+
1 row in set (0.00 sec)

[mysql> truncate table student;
Query OK, 0 rows affected (0.01 sec)

[mysql> select * from student;
Empty set (0.00 sec)
```

Figure 8 MySQL Command to delete all the data inside a table.

We are deleting all the data (Rows) inside table “Student”.

DML Commands

- Inserting data into database

```
mysql> insert into student(Student_Name,Student_Address) values('K Srikanth','Bangalore');
Query OK, 1 row affected (0.01 sec)

mysql> select * from student;
+-----+-----+-----+
| Student_ID | Student_Name | Student_Address |
+-----+-----+-----+
|          1 | K Srikanth   | Bangalore       |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Figure 9 MySQL Command to insert data into table.

We are inserting Specific data using insert command into table “student ” with specific values.

- Updating data in database

```
mysql> update student set Student_Address='Peenya,Bangalore' where Student_ID =1 ;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from student;
+-----+-----+-----+
| Student_ID | Student_Name | Student_Address |
+-----+-----+-----+
|          1 | K Srikanth   | Peenya,Bangalore |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Figure 10 MySQL Command to update data in table.

We are updating the address of student where his Student ID is “1” in table “Student”.

- Deleting data from database

```
mysql> delete from student where Student_ID =1 ;
Query OK, 1 row affected (0.01 sec)
```

Figure 11 MySQL Command to delete data in table.

We are deleting a Student using his ID in table “Student”.

- **Displaying data from database**

```
mysql> select * from student;
+-----+-----+-----+
| Student_ID | Student_Name | Student_Address |
+-----+-----+-----+
|          1 | K Srikanth   | Bangalore        |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Figure 12 MySQL Command to print all the data from the table.

We are Displaying all the data inside table "Student".

```
mysql> select Student_Name from student;
+-----+
| Student_Name |
+-----+
| K Srikanth   |
+-----+
1 row in set (0.01 sec)
```

Figure 13 MySQL Command to print specific data from the table.

We are Displaying specific data "Student_Name" from table "Student".

Q-B)

Creating a Table “Students” with schema (Student ID , Student Name , Student Address)

```
mysql> create table student(
  -> Student_ID INT auto_increment,
  -> Student_Name varchar(20),
  -> Student_Address varchar(20),
  -> primary key(Student_ID));
Query OK, 0 rows affected (0.01 sec)

mysql> describe student;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| Student_ID     | int           | NO   | PRI | NULL    | auto_increment |
| Student_Name   | varchar(20)   | YES  |     | NULL    |                |
| Student_Address | varchar(20)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Figure 14 MySQL Command to Create a table with schema.

We are creating a table “Student ”with a schema (*as per question) with extra being auto increment where we don’t have to specify Student_ID every-time as it increases on its own also the primary key for table “Student” is Student_ID and we are describing it .

Inserting Data into the table

```
mysql> insert into student(Student_Name,Student_Address) values('K Srikanth','Bangalore');
Query OK, 1 row affected (0.00 sec)

mysql> insert into student(Student_Name,Student_Address) values('Naveen Kumar GN','Bangalore');
Query OK, 1 row affected (0.01 sec)

mysql> insert into student(Student_Name,Student_Address) values('P Supraja','Bangalore');
Query OK, 1 row affected (0.01 sec)

mysql> insert into student(Student_Name,Student_Address) values('N Sushanth','Bangalore');
Query OK, 1 row affected (0.01 sec)

mysql> insert into student(Student_Name,Student_Address) values('G Akansha','Bangalore');
Query OK, 1 row affected (0.01 sec)
```

Figure 15 MySQL Command to add data inside a table

We are adding data of 5 student with Specific data using insert command into table “student ” with specific values.

1. Display details of all students

```
mysql> select * from student;
+-----+-----+-----+
| Student_ID | Student_Name | Student_Address |
+-----+-----+-----+
| 1 | K Srikanth | Bangalore |
| 2 | Naveen Kumar GN | Bangalore |
| 3 | P Supraja | Bangalore |
| 4 | N Sushanth | Bangalore |
| 5 | G Akansha | Bangalore |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

Figure 16 MySQL Command to print all data from table.

We are displaying all the data inside table "Student" where we have data of all the 5 students.

2.Display the name and address of the student with id= 101

```
mysql> select * from student
-> where Student_ID = 1;
+-----+-----+-----+
| Student_ID | Student_Name | Student_Address |
+-----+-----+-----+
| 1 | K Srikanth | Bangalore |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Figure 17 MySQL Command to print specific data using an attribute from schema

We are displaying details of specific student using his Student_ID = 1.

3. Insert a new student <105, 'John', 'Bangalore'>

```
mysql> insert into student(Student_ID,Student_Name,Student_Address) values(10,'Parthu Reddy','Bangalore');
Query OK, 1 row affected (0.00 sec)

mysql> select * from student where Student_ID = 10;
+-----+-----+-----+
| Student_ID | Student_Name | Student_Address |
+-----+-----+-----+
| 10 | Parthu Reddy | Bangalore |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Figure 18 MySQL Command to add specific data into table.

We are inserting a new student with attributes (Student_No = 10 , Student_Name = Parthu Reddy , City = Bangalore).

4. Change the address of the student John to 'Delhi'

```
[mysql> update student
[    -> set Student_Address = 'Delhi'
[    -> where Student_ID = 10;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

[mysql> select * from student where Student_ID = 10;
+-----+-----+-----+
| Student_ID | Student_Name | Student_Address |
+-----+-----+-----+
|          10 | Parthu Reddy | Delhi           |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Figure 19 MySQL Command to update student address to Delhi.

We are updating the student address of Parthu Reddy from Bangalore to Delhi.

5. Delete the details of a student with student id=105

```
[mysql> delete from student
[    -> where Student_ID = 10;
Query OK, 1 row affected (0.01 sec)

[mysql> select * from student where Student_ID = 10;
Empty set (0.01 sec)
```

Figure 20 MySQL Command to delete student with ID.

We are deleting a student from our database with his ID = 10;

6. Add a column to the schema Student with appropriate data type

```
[mysql> alter table student add Student_PhoneNumber BIGINT;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Figure 21 MySQL Command to add a new column.

We are adding a new Column "Student_PhoneNumber" into our table "Student".

```
mysql> update student set Student_PhoneNumber=9493364308 where Student_ID =1;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 Warnings: 0

mysql> update student set Student_PhoneNumber=7019462108 where Student_ID =2;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 Warnings: 0

mysql> update student set Student_PhoneNumber=1234567890 where Student_ID =3;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 Warnings: 0

mysql> update student set Student_PhoneNumber=8096225056 where Student_ID =4;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 Warnings: 0

mysql> update student set Student_PhoneNumber=1234567890 where Student_ID =5;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 Warnings: 0
```

Figure 22 MySQL Command to add specific data to a column.

We are inserting new data into our new column "Student_PhoneNumber" for table "Student".

```
mysql> select * from student ;
+-----+-----+-----+-----+
| Student_ID | Student_Name | Student_Address | Student_PhoneNumber |
+-----+-----+-----+-----+
| 1 | K Srikanth | Bangalore | 9493364308 |
| 2 | Naveen Kumar GN | Bangalore | 7019462108 |
| 3 | P Supraja | Bangalore | 1234567890 |
| 4 | N Sushanth | Bangalore | 8096225056 |
| 5 | G Akansha | Bangalore | 1234567890 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Figure 23 MySQL Command to display all the data from student table.

Finally, the data has been added and we are displaying the table "Student"

6. Conclusions

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database. SQL can be used to insert, search, update and delete database records. In a relational database, digital information about a specific customer is organised into rows, columns and tables which are indexed to make it easier to find relevant information through SQL.

7. Comments

1. Limitations of Experiments

The limitations of this experiment would be

In MySQL we cannot inset multiple data to rows with different values in a single line and MySQL is a relational database where as Mongo DB is Dynamic.

2. Limitations of Results

Should have used IDE for a Perfect Screenshot Result.

3. Learning happened

Got to learn about DDL AND DML Commands and made a simple relational database

4. Recommendation

Blank