Experiment No.4
Apply DML commands for the specified system
Date of Performance:
Date of Submission:

Aim :- Write insert query to insefi rows for each table created of your database management system. Use update and delete commands to manipulate the inserted values in the table.

Objective :- To learn commands of Data Manipulation Language(DML) to insert, update or delete the values in the database system.

Theory:

Data Manipulation Language (DML) is a subset of SQL (Structured Query Language) used for managing data within relational database management systems (RDBMS). DML commands are used to perform operations such as inserting, updating, and deleting data from database tables.

1. Inserting Data

The INSERT statement is used to add new rows of data into a table. It specifies the table to insert data into and provides values or expressions for each column in the new row. If a column list is not specified, values must be provided for all columns in the table in the order they were defined.

Syntax:-

INSERT INTO table name (column 1, column2, column3) VALUES (valuel, value2, value3);

2. Updating Data

The UPDATE statement is used to modify existing data within a table. It allows you to change the values of one or more columns m one or more rows based on specified conditions. If no condition is specified, all rows in the table will be updated.

Syntax:-

UPDATE table name SET columnl = valuel, column2 = value2 WHERE condition;

3. Deleting Data

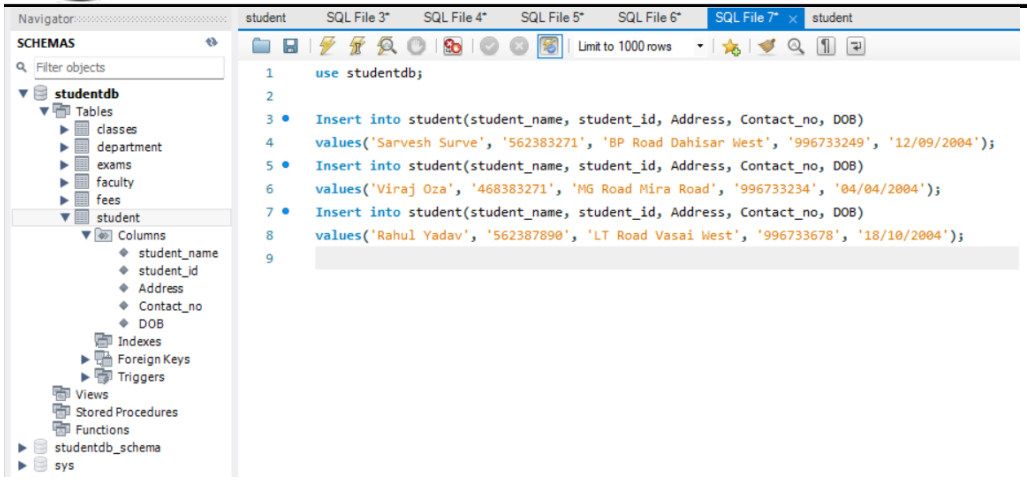
The DELETE statement is used to remove one or more rows from a table based on specified conditions. If no condition is specified, all rows in the table will be deleted.

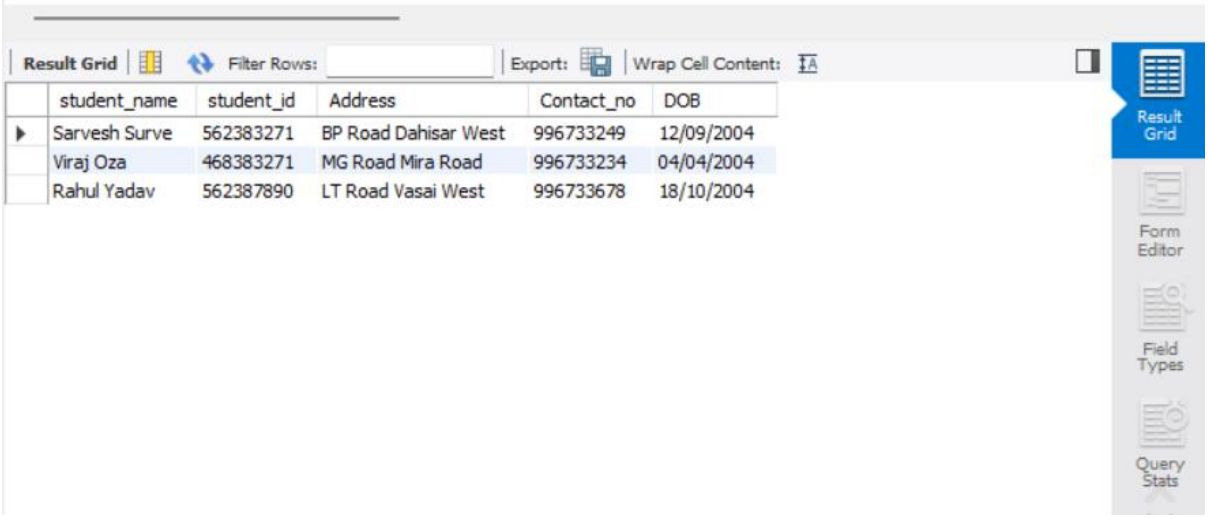
Syntax:

DELETE FROM table name WHERE condition;

Implementation:

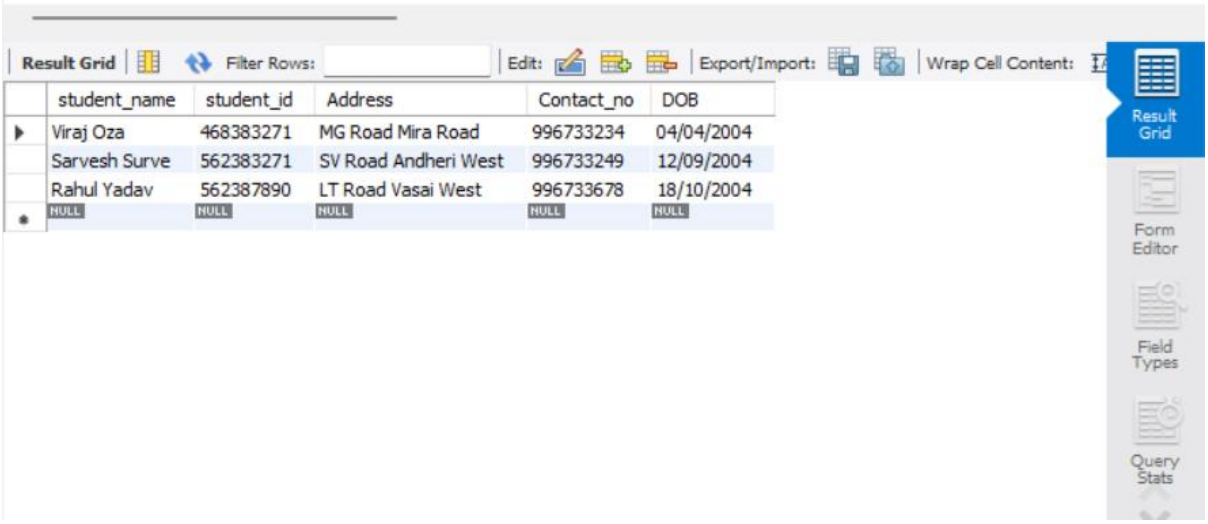
Inserting Data:





Updating Data:

```
9
10 •    UPDATE student
11      SET Address = 'SV Road Andheri West'
12      WHERE student_id = '562383271';
13
14
```



student_name	student_id	Address	Contact_no	DOB
Viraj Oza	468383271	MG Road Mira Road	996733234	04/04/2004
Sarvesh Surve	562383271	SV Road Andheri West	996733249	12/09/2004
Rahul Yadav	562387890	LT Road Vasai West	996733678	18/10/2004
NULL	NULL	NULL	NULL	NULL

Conclusion:

Database constraints play a crucial role in enforcing data integrity during Data Manipulation Language (DML) operations. Here's a brief explanation:

1. Primary Key Constraint: Ensures that each row in a table is uniquely identified by a primary key. This constraint prevents duplicate entries and ensures data integrity by guaranteeing the uniqueness of key values.

2. Foreign Key Constraint: Maintains referential integrity between two related tables by ensuring that values in a foreign key column match values in the corresponding primary key column of the referenced table. This constraint prevents orphaned records and maintains consistency in relational databases.

3.Unique Constraint: Enforces uniqueness on one or more columns, similar to a primary key constraint but without the requirement of being the primary key. It ensures that no two rows have the same combination of values in the specified columns.

4.Check Constraint: Validates the data entered into a column against a specific condition or set of conditions. It restricts the values that can be inserted or updated in a column, ensuring that only valid data is stored in the database.

5.Not Null Constraint: Specifies that a column cannot contain null values. It ensures that essential data is always present in the specified column, preventing the insertion of incomplete or missing information.

By implementing these constraints, databases maintain data consistency, accuracy, and reliability, thereby upholding data integrity during various DML operations such as insertion, updating, and deletion of records.

To update multiple columns in a table using a single UPDATE statement in SQL, you can specify each column you want to update followed by its new value within the SET clause. Here's the general syntax:

SYNTAX:-

```
UPDATE table_name
SET column1 = value1, column2 = value2, ..., columnN = valueN
WHERE condition;
```

For example, let's say we have a table named "employees" with columns "first_name", "last_name", and "age". If we want to update both the "first_name" and "last_name" columns for a specific employee with an employee_id of 123, we can do it like this:

SYNTAX:-

```
UPDATE employees
```

```
SET first_name = 'John', last_name = 'Doe'
WHERE employee_id = 123;
```

This single UPDATE statement will change the "first_name" to 'John' and the "last_name" to 'Doe' for the employee with the ID of 123.

You can include as many columns to update as needed, separated by commas, and you can use a WHERE clause to specify which rows to update based on certain conditions.