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

Aim :- Write insert query to insert 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.

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 (column1, column2, column3) VALUES (value1, value2, value3);

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 in 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 column1 = value1, column2 = value2 WHERE condition;

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

INSERT

```
1  CREATE DATABASE social_media_db;
2  USE social_media_db;
3  CREATE TABLE users (
4      user_id INT PRIMARY KEY AUTO_INCREMENT,
5      username VARCHAR(50) UNIQUE NOT NULL,
6      email VARCHAR(100) UNIQUE NOT NULL,
7      password VARCHAR(100) NOT NULL
8  );
9  INSERT INTO users (user_id, username, email, password)
10 VALUES ('107', 'Aaru', 'aradhya@gmail.com', 'wewy2vduqw');
11 INSERT INTO users (user_id, username, email, password)
12 VALUES ('108', 'Ansh', 'ansha@gmail.com', 'wewqwy2vduqw');
13
14  select * from users;
15
```

Result Grid

user_id	username	email	password	created_at
107	Aaru	aradhya@gmail.com	wewy2vduqw	2024-04-22 19:11:25
108	Ansh	ansha@gmail.com	wewqwy2vduqw	2024-04-22 19:13:09
NULL	NULL	NULL	NULL	NULL

UPDATE

```
3  CREATE TABLE users (
4      user_id INT PRIMARY KEY AUTO_INCREMENT,
5      username VARCHAR(50) UNIQUE NOT NULL,
6      email VARCHAR(100) UNIQUE NOT NULL,
7      password VARCHAR(100) NOT NULL
8  );
9  INSERT INTO users (user_id, username, email, password)
10 VALUES ('107', 'Aaru', 'aradhya@gmail.com', 'wewy2vduqw');
11 INSERT INTO users (user_id, username, email, password)
12 VALUES ('108', 'Ansh', 'ansha@gmail.com', 'wewqwy2vduqw');
13
14  select * from users;
15   update users set username='Ekansh' where user_id=108;
16 select * from users;
17
```

Result Grid

user_id	username	email	password	created_at
107	Aaru	aradhya@gmail.com	wewy2vduqw	2024-04-22 19:11:25
108	Ekansh	ansha@gmail.com	wewqwy2vduqw	2024-04-22 19:13:09
NULL	NULL	NULL	NULL	NULL

DELETE

```
CREATE TABLE users (
    user_id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(100) NOT NULL
);
INSERT INTO users (user_id, username, email, password)
VALUES ('107', 'Aaru', 'aradhya@gmail.com', 'wewy2vduqw');
INSERT INTO users (user_id, username, email, password)
VALUES ('108', 'Ansh', 'ansha@gmail.com', 'wewqwy2vduqw');

 select * from users;
  update users set username='Ekansh' where user_id=108;
select * from users;
```

Result Grid

user_id	username	email	password	created_at
107	Aaru	aradhya@gmail.com	wewy2vduqw	2024-04-22 19:11:25
108	Ekansh	ansha@gmail.com	wewqwy2vduqw	2024-04-22 19:13:09
NULL	NULL	NULL	NULL	NULL

Conclusion

1. **Explain the role of database constraints in enforcing data integrity during DML operations.**

   Database constraints play a crucial role in enforcing data integrity during Data Manipulation Language (DML) operations by imposing rules and conditions on the data being inserted, updated, or deleted. Here's how they ensure data integrity:

a) Preventing Invalid Data: Constraints such as primary key, unique, and check constraints ensure that only valid and permissible data is inserted into the database. They restrict the insertion of duplicate values, enforce data format and range constraints, and prevent the insertion of data that does not meet specified criteria.

b) Maintaining Relationships: Foreign key constraints maintain referential integrity by ensuring that relationships between tables are upheld. They prevent the insertion of data that references non-existent records in related tables, thus maintaining consistency and preventing orphaned records.

c) Enforcing Business Rules: Constraints enforce business rules and requirements defined for the database schema. They ensure that data adheres to predefined business logic, preventing data inconsistencies and ensuring that the database accurately represents the real-world domain it models.

d) Automatic Validation: Constraints automatically validate data during DML operations, providing immediate feedback if data violates integrity rules. This proactive validation prevents the insertion of erroneous data, reducing the likelihood of data corruption or inconsistencies.

2. **How do you update multiple columns in a table using a single UPDATE statement?**
   To update multiple columns in a table using a single UPDATE statement, you specify the column names and their new values within the UPDATE command's SET clause. Here's a short example:

   UPDATE table_name
   SET column1 = value1, column2 = value2, column3 = value3
   WHERE condition;

   In this syntax:
   table_name is the name of the table you want to update.
   column1, column2, column3, etc., are the names of the columns you want to update.
   value1, value2, value3, etc., are the new values you want to assign to the respective columns.
   condition is an optional condition that specifies which rows should be updated. If omitted, all rows in the table will be updated.
   This single UPDATE statement will modify the specified columns in the table according to the provided values, subject to the specified condition.