Experiment No.4

Apply DML commands for the specified system

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.

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

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:

DOCUMENT MANAGEMENT SYSTEM:

```
CREATE TABLE documents (
    doc_id INT PRIMARY KEY,
    title VARCHAR(255),
    author VARCHAR(100),
    content TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
INSERT INTO documents (doc_id, title, author, content)
VALUES
    (1, 'Introduction to DMS', 'John Doe', 'This document explains the basics of DMS.'),
    (2, 'Best Practices for Document Organization', 'Jane Smith', 'Tips on organizing your documents efficiently.');
SELECT doc_id, title, author FROM documents;
UPDATE documents
SET author = 'Alice Johnson'
WHERE doc_id = 1;
DELETE FROM documents WHERE doc_id = 2;
SELECT doc_id, title, author FROM documents;
```

Output:

```
+--------+-----------------------------------------+------------+
| doc_id | title                                   | author     |
+--------+-----------------------------------------+------------+
|      1 | Introduction to DMS                     | John Doe   |
|      2 | Best Practices for Document Organization | Jane Smith |
+--------+-----------------------------------------+------------+
+--------+---------------------+---------------+
| doc_id | title               | author        |
+--------+---------------------+---------------+
|      1 | Introduction to DMS | Alice Johnson |
+--------+---------------------+---------------+
```

Conclusion:

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

   ➔ **Role of Database Constraints in Enforcing Data Integrity during DML Operations**:

   - **Database constraints** play a crucial role in maintaining data integrity during Data Manipulation Language (DML) operations (such as `INSERT`, `UPDATE`, and `DELETE`). Here's how they contribute:
     - **Primary Key Constraint**: Ensures that each row in a table has a unique identifier (primary key). It prevents duplicate records and ensures data consistency.
     - **Unique Constraint**: Guarantees that a column (or a combination of columns) contains unique values across all rows. This prevents duplicate entries.
     - **Foreign Key Constraint**: Links data between related tables. It ensures referential integrity by enforcing that values in a child table's foreign key column match values in the parent table's primary key column.
     - **Check Constraint**: Defines rules for column values (e.g., age > 0, status in ('active', 'inactive')). It prevents invalid data from being inserted or updated.
     - **Not Null Constraint**: Ensures that a column cannot have null (empty) values. It enforces mandatory data entry.
   - By applying these constraints, the database system ensures that data adheres to predefined rules, preventing inconsistencies and maintaining data quality.

2. How do you update multiple columns in a table using a single UPDATE statement?

➔ **Updating Multiple Columns in a Table Using a Single UPDATE Statement**:

- To update multiple columns in a table using a single `UPDATE` statement, follow this syntax:

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

- Here's an example: Suppose we have a `students` table with columns `student_id`, `first_name`, `last_name`, and `age`. We want to update the first name and age of a student with `student_id = 101`.

```
UPDATE students
SET first_name = 'Alice', age = 25
WHERE student_id = 101;
```

This query will update the specified columns for the student with ID 101.