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.

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

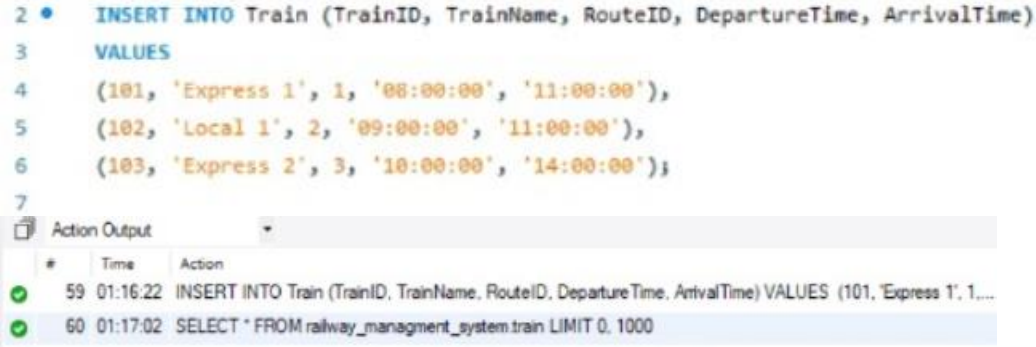
The DELETE statement is used to remove one or more rows from a table based on specified conditions. If no condition is specified, all rows in the table will be deleted.

Syntax:

```
DELETE FROM table_name WHERE condition;
```

Implementation:

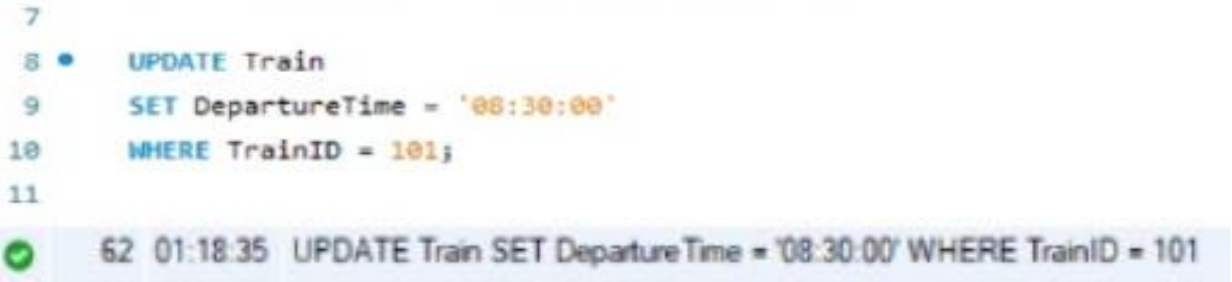
Inserting Data:



```
INSERT INTO Train (TrainID, TrainName, RouteID, DepartureTime, ArrivalTime)
VALUES
(101, 'Express 1', 1, '08:00:00', '11:00:00'),
(102, 'Local 1', 2, '09:00:00', '11:00:00'),
(103, 'Express 2', 3, '10:00:00', '14:00:00');
```

```
#   Time      Action
59  01:16:22  INSERT INTO Train (TrainID, TrainName, RouteID, DepartureTime, ArrivalTime) VALUES (101, 'Express 1', 1,...
60  01:17:02  SELECT * FROM railway_managment_system.train LIMIT 0, 1000
```

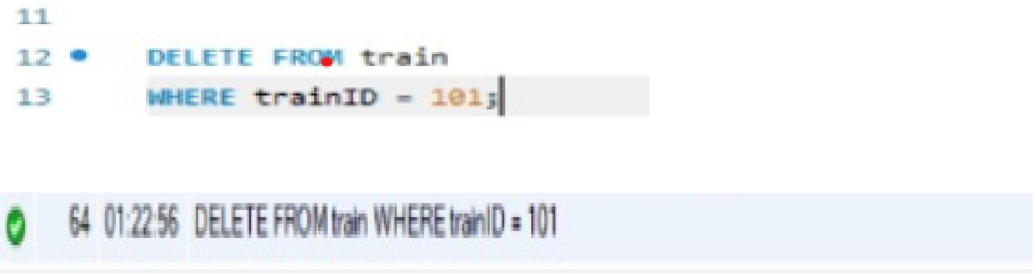
Updating Data:



```
UPDATE Train
SET DepartureTime = '08:30:00'
WHERE TrainID = 101;
```

```
62  01:18:35  UPDATE Train SET DepartureTime = '08:30:00' WHERE TrainID = 101
```

Deleting Data:



```
DELETE FROM train
WHERE trainID = 101;
```

```
64  01:22:56  DELETE FROM train WHERE trainID = 101
```

Conclusion:

1. Explain the 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:
   1. Entity Integrity: Primary key constraints ensure each row in a table is uniquely identified, preventing duplicates.
   2. Referential Integrity: Foreign key constraints establish relationships between tables, enforcing consistency across related data.
   3. Domain Integrity: Check constraints enforce rules on column values, such as data type or range restrictions, ensuring valid data entry.
   4. Null Integrity: Not null constraints mandate that a column must have a value, preventing null entries where inappropriate.
   5. User-defined Integrity: Custom constraints allow businesses to enforce specific rules tailored to their needs, further enhancing data integrity. By enforcing these constraints, the database ensures that only valid, consistent, and accurate data is stored, maintaining data integrity across all DML operations.

2. How do you update multiple columns in a table using a single UPDATE statement.
   To update multiple columns in a table with one SQL statement:
   - Use the UPDATE statement.
   - Specify the table name.
   - Set each column with its new value using SET.
   - Add a WHERE clause to specify the condition for which rows to update.
   - This allows updating specific columns in specific rows efficiently

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