|  |  |  |
| --- | --- | --- |
| **Academic Year:** 2024-25 | **Year:** Second Year | **Semester:** II |
| **PRN No.:** | **Name:** | |
| **Subject:** Database Management System | | |
| **Assignment No.**: 3 |  | |
| **Date:** |  | |

**Lab Assignment: 03**

**Title: DML Queries**: Write DML (Data Manipulation Language) queries for a suitable database application. The queries should cover:

**Basic Queries**: Insert, update, delete and select records in the database.

* 1. **Search and Filtering**: Retrieve specific data using WHERE clauses with conditions such as equality, range.

**Sorting and Limiting**: Use ORDER BY and LIMIT clauses to arrange and restrict the result set, and pattern matching (LIKE).

**Theory:**

**What is DML?**

DML is a subset of SQL (Structured Query Language) that deals with modifying and managing data within a database. The key DML commands are:

* INSERT – Adds new records to a table.
* UPDATE – Modifies existing records.
* DELETE – Removes records from a table.
* SELECT – Retrieves data from a database (though primarily part of DQL, it is often considered in DML).

**Arithmetic Operators**

|  |  |  |
| --- | --- | --- |
| Operator | Description | Example |
| + | Addition | SELECT 10 + 5; → 15 |
| - | Subtraction | SELECT 10 - 5; → 5 |
| \* | Multiplication | SELECT 10 \* 5; → 50 |
| / | Division | SELECT 10 / 5; → 2 |
| % | Modulus (Remainder) | SELECT 10 % 3; → 1 |

**Logical Operators**

|  |  |  |
| --- | --- | --- |
| Operator | Description | Example |
| AND | Returns true if both conditions are true | WHERE age > 18 AND city = 'Pune' |
| OR | Returns true if at least one condition is true | WHERE age > 18 OR city = 'Pune' |
| NOT | Returns true if the condition is false | WHERE NOT city = 'Pune' |

**Comparison Operators**

|  |  |  |
| --- | --- | --- |
| Operator | Description | Example |
| = | Equal to | WHERE age = 25 |
| != or <> | Not equal to | WHERE age != 25 |
| > | Greater than | WHERE age > 25 |
| < | Less than | WHERE age < 25 |
| >= | Greater than or equal to | WHERE age >= 25 |
| <= | Less than or equal to | WHERE age <= 25 |

**Order By**

The ORDER BY clause is used to sort query results in ascending (ASC, default) or descending (DESC) order.

1. SELECT name, age FROM students ORDER BY age ASC;

2. SELECT name, age FROM students ORDER BY age DESC;

3.

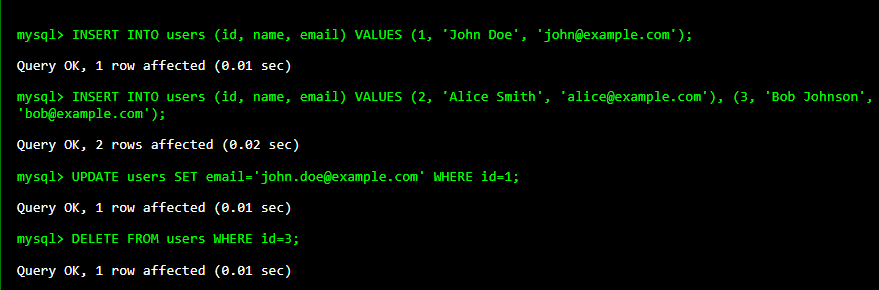
**Like**

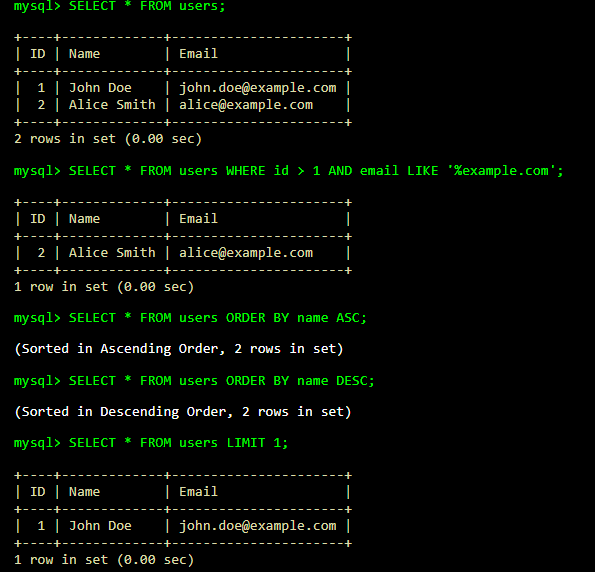
The LIKE operator is used for pattern matching in text columns. It works with wildcards:

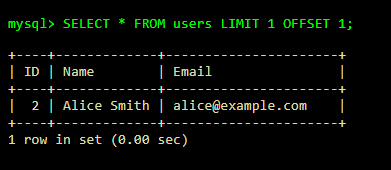
|  |  |  |
| --- | --- | --- |
| Wildcard | Description | Example |
| % | Matches zero or more characters | WHERE name LIKE 'A%' (Names starting with A) |
| \_ | Matches a single character | WHERE name LIKE '\_o%' (Second letter is 'o') |

**Show Query Execution Screenshots for:**

* Insert – Single Record
* Insert – Multiple Records
* Update
* Delete
* Select
* Where clause with AND, OR, NOT, ALL, ANY, NOT ANY, EXISTS, NOT EXISTS, BETWEEN, NOT BETWEEN, LESS THAN, GREATER THAN, EQUAL TO, NULL, NOT NULL, UNIQUE, DISTINCT
* Order By – Ascending
* Order By – Descending
* Limit
* With Offset, and two parameters
* Like with % and \_







**FAQs:**

1. Which commands come under DML?

DML (Data Manipulation Language) includes commands used to manipulate data in tables:

* **INSERT** – Adds new records to a table.
* **UPDATE** – Modifies existing records in a table.
* **DELETE** – Removes records from a table.
* **SELECT** – Retrieves records from a table.

1. What is the difference between DML and DDL (Data Definition Language)?

|  |  |  |
| --- | --- | --- |
| **Feature** | **DML (Data Manipulation Language)** | **DDL (Data Definition Language)** |
| Purpose | Modifies data within tables | Modifies database structure (schema) |
| Commands | INSERT, UPDATE, DELETE, SELECT | CREATE, ALTER, DROP, TRUNCATE |
| Effects | Affects rows in a table | Affects entire table or database |
| Rollback | Can be rolled back (COMMIT, ROLLBACK) | Mostly permanent changes |
| Example | INSERT INTO users VALUES (1, 'Alice'); | CREATE TABLE users (id INT, name VARCHAR(50)); |

1. Explain Different between ANY and ALL

**ANY**:

Compares a value with **any** value in a subquery (at least one condition must be true).

1. SELECT \* FROM employees WHERE salary > ANY (SELECT salary FROM interns);

* This retrieves employees whose salary is greater than the lowest intern salary.

**ALL**:

Compares a value with **all** values in a subquery (must satisfy all conditions).

1. SELECT \* FROM employees WHERE salary > ALL (SELECT salary FROM interns);

* This retrieves employees whose salary is higher than the highest intern salary.