****

**Green University of Bangladesh**

**Department of Computer Science and Engineering (CSE)**

**Faculty of Sciences and Engineering**

**Semester: (Summer, Year:2025), B.Sc. in CSE (Day)**

**LAB REPORT NO: 2**

**Course Title:** Database Lab

**Course Code:** CSE 210 **Section:** 232-D1

**Lab Experiment Name**: Querying and Filtering data in MySQL Table.

**Student Details**

|  |  |  |
| --- | --- | --- |
| **Name** | | **ID** |
| **1.** | Rukonuzzaman Topu | 232002280 |

**Submission Date : 07-17-2025**

**Course Teacher’s Name :** Farhana Akter Sunny

**[For Teachers use only: Don’t Write Anything inside this box]**

|  |
| --- |
| **Lab Report Status**  **Marks: ………………………………… Signature:.....................**  **Comments:.............................................. Date:..............................** |

1. **TITLE OF THE LAB EXPERIMENT:**

Querying and Filtering data in MySQL Table.

.

**OBJECTIVES:**

* To gain a clear understanding of how to query and filter data using logical operators such as AND, OR, and NOT.
* To explore how to restrict the number of records returned using the LIMIT clause.
* To learn how to use comparison operators like BETWEEN, NOT BETWEEN, IN, and NOT IN for data selection.
* To practice pattern matching in string data using the LIKE operator.
* To become skilled in sorting query results using ORDER BY with ASC (ascending) and DESC (descending) options.
* To build confidence in writing effective and accurate filtering conditions in MySQL queries.

**PROCEDURE:**

**1. Create the 'branch' table**

CREATE TABLE branch (

branch\_name VARCHAR(50) PRIMARY KEY,

branch\_city VARCHAR(50),

assets DECIMAL(15, 2)

);

**2. Create the 'customer' table**

CREATE TABLE customer (

customer\_id INT PRIMARY KEY,

customer\_name VARCHAR(100),

customer\_city VARCHAR(50)

);

**3. Create the 'account' table**

CREATE TABLE account (

account\_number INT PRIMARY KEY,

branch\_name VARCHAR(50),

balance DECIMAL(15, 2),

FOREIGN KEY (branch\_name) REFERENCES branch(branch\_name)

);

**4. Create the 'loan' table**

CREATE TABLE loan (

loan\_number INT PRIMARY KEY,

branch\_name VARCHAR(50),

amount DECIMAL(15, 2),

FOREIGN KEY (branch\_name) REFERENCES branch(branch\_name)

);

**5. Create the 'depositor' table**

CREATE TABLE depositor (

customer\_name VARCHAR(100),

account\_number INT,

FOREIGN KEY (account\_number) REFERENCES account(account\_number),

PRIMARY KEY (customer\_name, account\_number)

);

**6. Create the 'borrower' table**

CREATE TABLE borrower (

customer\_name VARCHAR(100),

loan\_number INT,

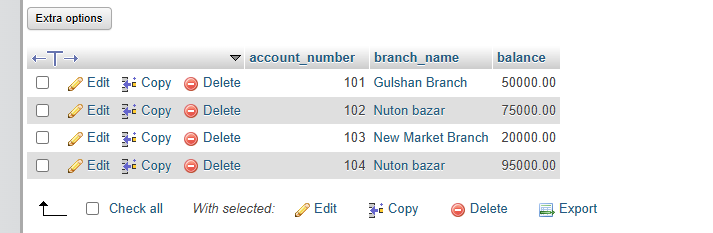
FOREIGN KEY (loan\_number) REFERENCES loan(loan\_number),

PRIMARY KEY (customer\_name, loan\_number)

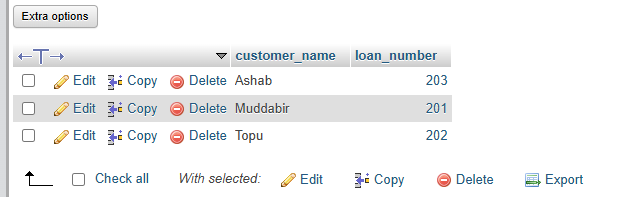
);

**DATA INSERT**

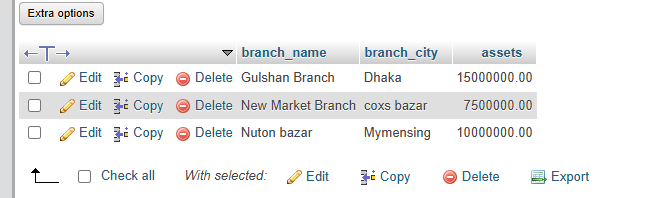
**1. Account table**



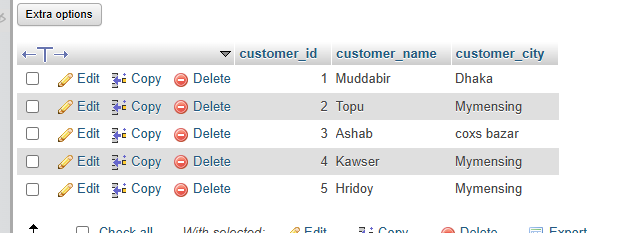
**2. Borrower table**

****

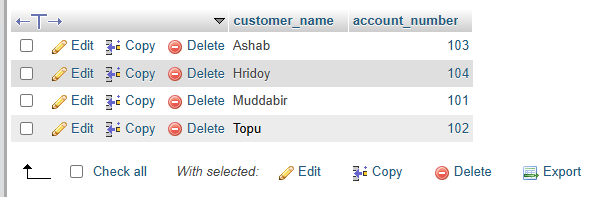
**3. Branch table**



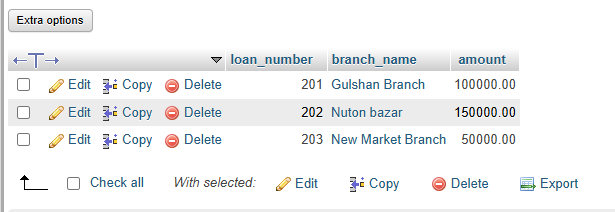
**4. Customer table**

****

**5. Depositor table**

****

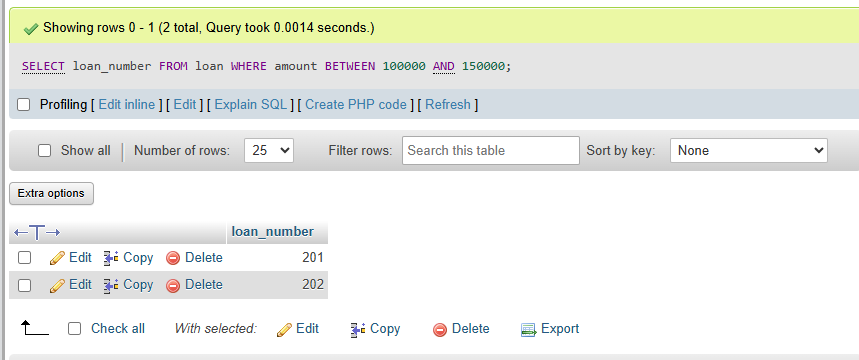
**6. loan table**

****

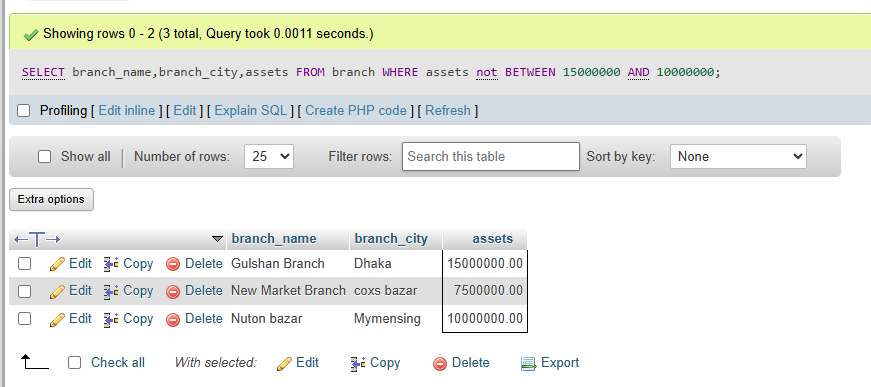
**OUTPUT:**

**SQL QUERIES:**

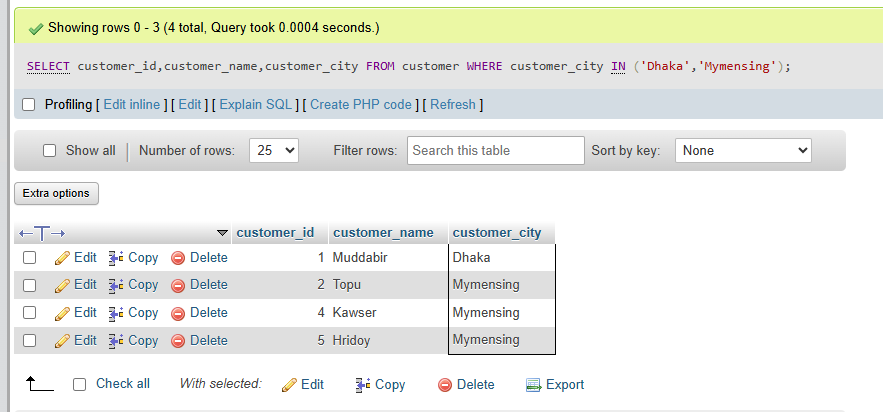
**1. BETWEEN Operator**

****

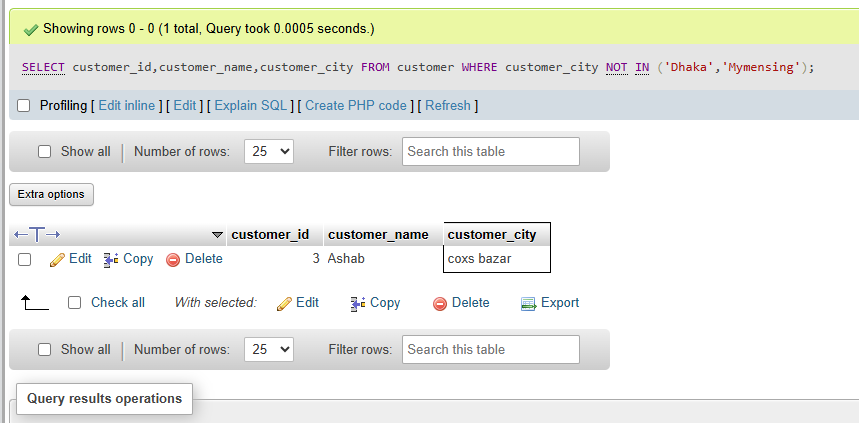
**2. Not Between**



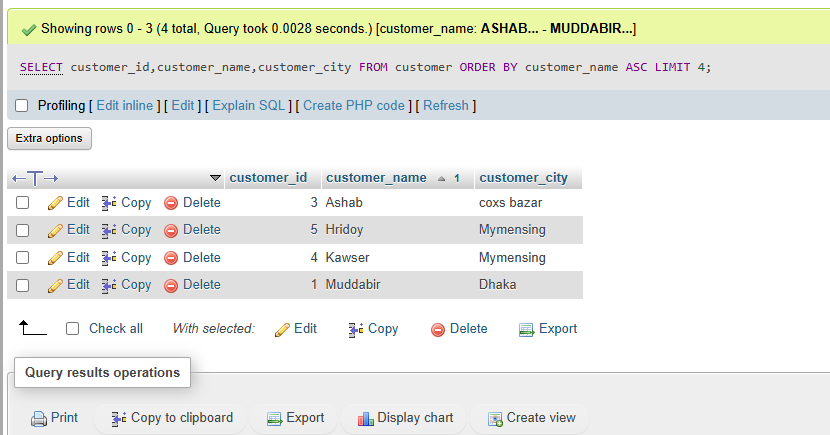
**3. IN Operator**

****

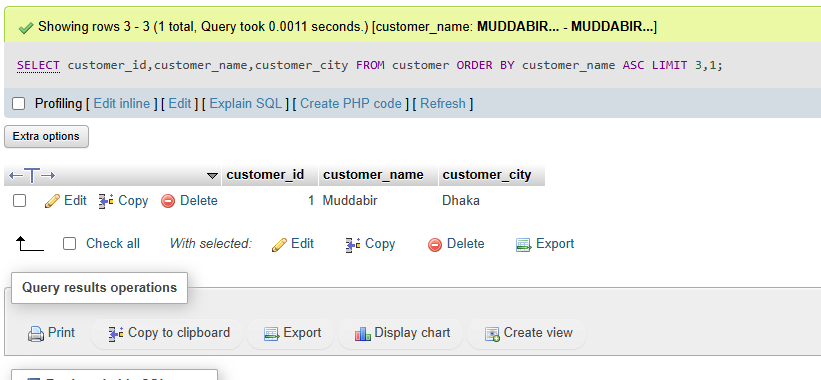
**4. NOT IN Operator**

****

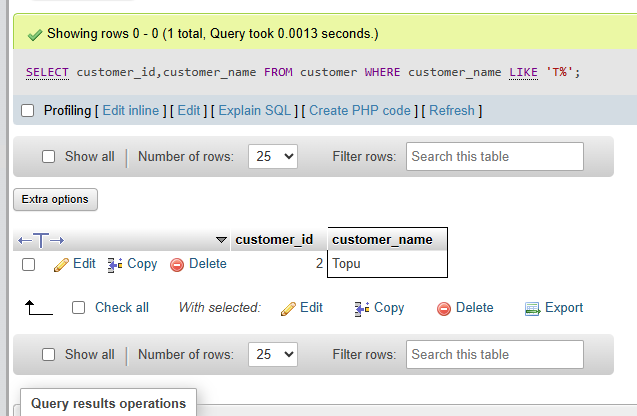
**5. Order By (First 4 customers alphabetically)**

****

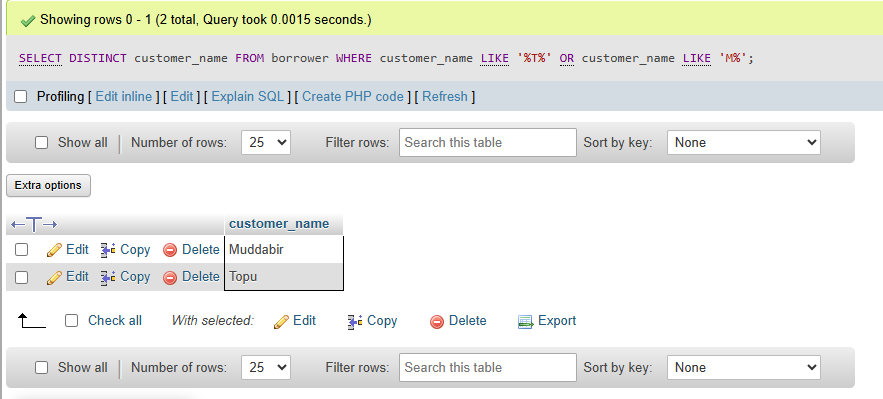
**6. Order by ASC (4th customer in alphabetical order)**

****

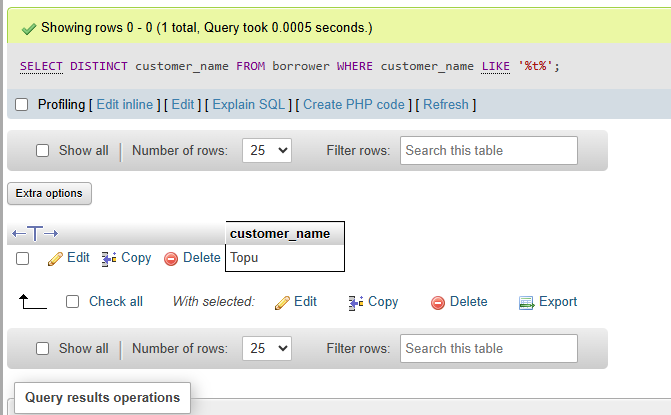
**7. String with ‘T’**

****

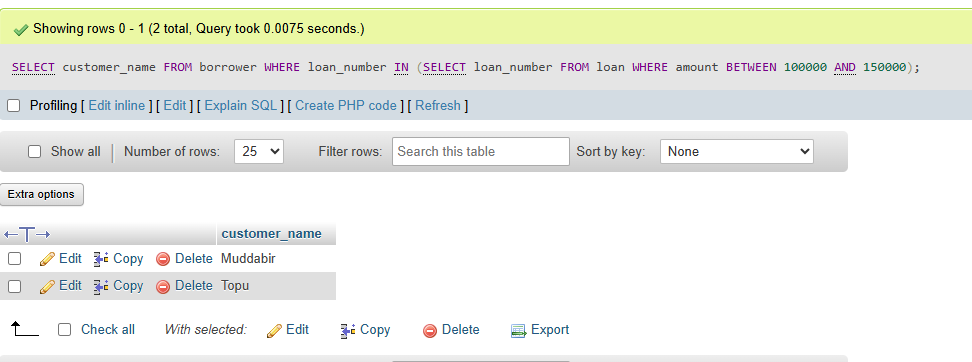
**8. String Operator and OR Operator**

****

**9. String Operator**

****

**10. Subqueries**

****

**Discussion:**  
In this lab session, we explored how to effectively work with a MySQL database by performing various operations on banking-related tables such as customer, account, branch, and loan. We practiced inserting multiple records into these tables to simulate real-world data. Through different SQL queries, we learned how to retrieve specific information based on given conditions. A key part of the lab involved understanding how to filter data using logical operators like AND, OR, and NOT to refine query results. Additionally, we applied operators such as BETWEEN, IN, and LIKE to search for values using different criteria and patterns. We also learned how to organize data by sorting the output with ORDER BY, and how to limit the number of displayed results using the LIMIT clause. Overall, this lab helped strengthen our understanding of data retrieval and filtering techniques in MySQL.valid departments. Check constraints ensure that salary values meet logical conditions. Successful data insertion verifies the correctness of the schema design, and errors highlight potential violations, demonstrating the vital role of constraints in preventing invalid data entry.