

Assignment 2 (Creation of Tables and Implementation of WHERE, ORDER BY and LIKE Clauses using MySQL)

Name: Dinisha Jain

Domain: AI/ML

Assignment

Aim of the Assignment

The aim of this assignment is to create two tables in MySQL and perform conditional operations using WHERE clause with AND, OR, NOT, along with ORDER BY and LIKE operators.

Database Creation

Command:

```
CREATE DATABASE college_db;
```

```
CREATE DATABASE college_db;
```

Output:

```
MySQL localhost:33060+ ssl SQL > CREATE DATABASE college_db;
Query OK, 1 row affected (0.0291 sec)
```

Use Database Command

Command:

```
USE college_db;
```

```
USE college_db;
```

Output:

```
Default schema set to 'college_db'.
```

Assignment 2 (Creation of Tables and Implementation of WHERE, ORDER BY and LIKE Clauses using MySQL)

Table Creation

Table 1: Students

Command:

```
CREATE TABLE Students(
```

```
    student_id INT PRIMARY KEY,
```

```
    name VARCHAR(50),
```

```
    AGE INT,
```

```
    city VARCHAR(50)
```

```
);
```

```
MySQL localhost:33060+ ssl college_db SQL > CREATE TABLE Students(
->     student_id INT PRIMARY KEY,
->     name VARCHAR(50),
->     AGE INT,
->     city VARCHAR(50)
-> );
```

Output:

```
Query OK, 0 rows affected (0.0466 sec)
```

Assignment 2 (Creation of Tables and Implementation of WHERE, ORDER BY and LIKE Clauses using MySQL)

Table 2: Courses

Command:

```
CREATE TABLE Courses(
```

```
    course_id INT PRIMARY KEY,
```

```
    course_name VARCHAR(50),
```

```
    duration INT
```

```
);
```

```
MySQL localhost:33060+ ssl college_db SQL > CREATE TABLE Courses (
    ->     course_id INT PRIMARY KEY,
    ->     course_name VARCHAR(50),
    ->     duration INT
    -> );
```

Output:

```
Query OK, 0 rows affected (0.0464 sec)
```

Assignment 2 (Creation of Tables and Implementation of WHERE, ORDER BY and LIKE Clauses using MySQL)

Inserting Data into Students Tables

Command:

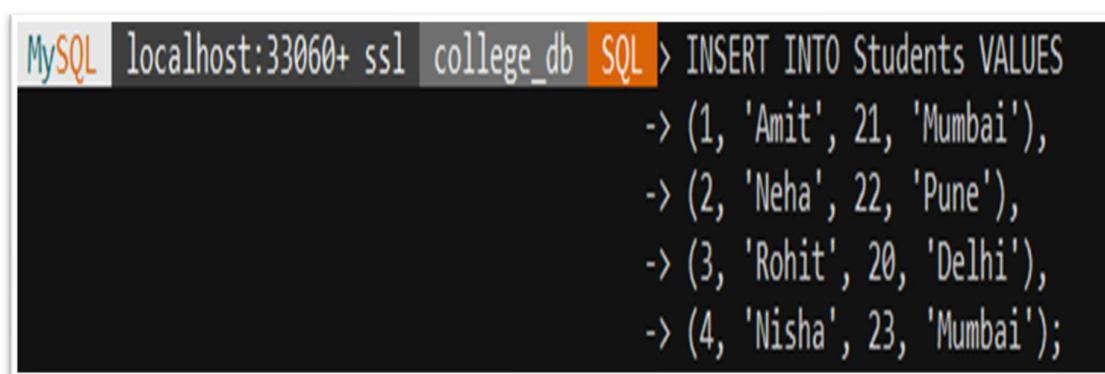
```
INSERT INTO Students VALUES
```

```
(1, 'Amit', 21, 'Mumbai'),
```

```
(2, 'Neha', 22, 'Pune'),
```

```
(3, 'Rohit', 20, 'Delhi'),
```

```
(4, 'Nisha', 23, 'Mumbai');
```



The screenshot shows a terminal window with the MySQL command-line client. The prompt indicates the connection is to 'localhost:33060+ssl' and the database is 'college_db'. The user has entered an 'INSERT INTO Students VALUES' statement followed by four data tuples. The command is shown in orange, and the input text is in white.

```
MySQL [localhost:33060+ssl] college_db SQL > INSERT INTO Students VALUES  
        -> (1, 'Amit', 21, 'Mumbai'),  
        -> (2, 'Neha', 22, 'Pune'),  
        -> (3, 'Rohit', 20, 'Delhi'),  
        -> (4, 'Nisha', 23, 'Mumbai');
```

Output:

```
Query OK, 4 rows affected (0.0103 sec)
```

Assignment 2 (Creation of Tables and Implementation of WHERE, ORDER BY and LIKE Clauses using MySQL)

Insert Data into Courses Table

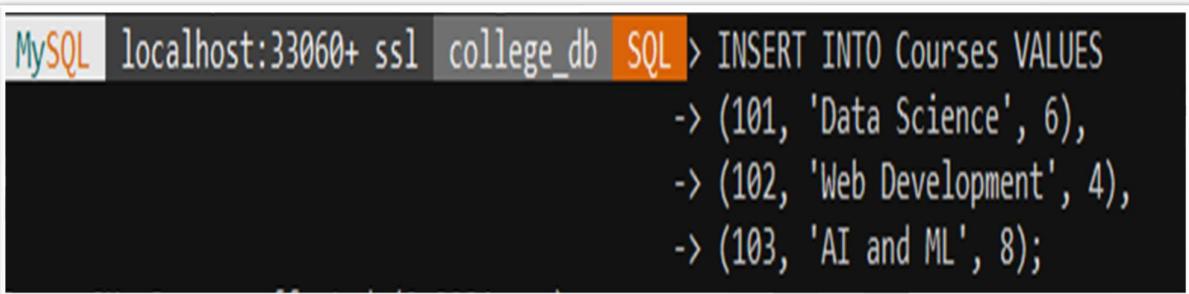
Command:

```
INSERT INTO Courses VALUES
```

```
(101, 'Data Science', 6),
```

```
(102, 'Web Development', 4),
```

```
(103, 'AI and ML', 8);
```



The screenshot shows a terminal window with the MySQL command-line client. The prompt indicates the connection is to 'localhost:33060+ ssl' database 'college_db'. The user has entered an 'INSERT INTO Courses VALUES' statement followed by three rows of data. The data rows are formatted with a leading dash and parentheses, indicating they are being inserted as multiple values.

```
MySQL | localhost:33060+ ssl | college_db | SQL > INSERT INTO Courses VALUES
      -> (101, 'Data Science', 6),
      -> (102, 'Web Development', 4),
      -> (103, 'AI and ML', 8);
```

Output:

```
Query OK, 3 rows affected (0.0084 sec)
```

Assignment 2 (Creation of Tables and Implementation of WHERE, ORDER BY and LIKE Clauses using MySQL)

Display Students Table

Command:

```
SELECT * FROM Students;
```



The image shows a screenshot of a MySQL command-line interface. The title bar includes the MySQL logo, the host (localhost:33060+ ssl), the database (college_db), and the SQL prompt (>). Below the title bar, the command `SELECT * FROM Students;` is entered and ready to be executed.

Output:

| student_id | name | AGE | city |
|------------|-------|-----|--------|
| 1 | Amit | 21 | Mumbai |
| 2 | Neha | 22 | Pune |
| 3 | Rohit | 20 | Delhi |
| 4 | Nisha | 23 | Mumbai |

4 rows in set (0.0028 sec)

Assignment 2 (Creation of Tables and Implementation of WHERE, ORDER BY and LIKE Clauses using MySQL)

SQL Operations

WHERE Clause using AND

Command:

```
SELECT * FROM Students
```

```
WHERE AGE > 21 AND city = 'Mumbai';
```

```
MySQL localhost:33060+ssl college_db SQL > SELECT * FROM Students
-> WHERE AGE > 21 AND city = 'Mumbai';
```

Output:

```
+-----+-----+-----+
| student_id | name   | AGE   | city   |
+-----+-----+-----+
|          4 | Nisha | 23   | Mumbai |
+-----+-----+-----+
1 row in set (0.0013 sec)
```

Assignment 2 (Creation of Tables and Implementation of WHERE, ORDER BY and LIKE Clauses using MySQL)

WHERE Clause using OR

Command:

```
SELECT * FROM Students
```

```
WHERE city = 'Mumbai' OR city = 'Pune';
```

```
MySQL [localhost:33060+ ssl] college_db SQL > SELECT * FROM Students
-> WHERE city = 'Mumbai' OR city = 'Pune';
```

Output:

```
+-----+-----+-----+
| student_id | name   | AGE   | city   |
+-----+-----+-----+
|          1 | Amit   |   21  | Mumbai |
|          2 | Neha   |   22  | Pune   |
|          4 | Nisha  |   23  | Mumbai |
+-----+-----+-----+
3 rows in set (0.0012 sec)
```

Assignment 2 (Creation of Tables and Implementation of WHERE, ORDER BY and LIKE Clauses using MySQL)

WHERE Clause using NOT

Command:

```
SELECT * FROM Students
```

```
WHERE NOT city = 'Delhi';
```

```
MySQL localhost:33060+ ssl college_db SQL > SELECT * FROM Students
-> WHERE NOT city = 'Delhi';
```

Output:

```
+-----+-----+-----+-----+
| student_id | name   | AGE  | city   |
+-----+-----+-----+-----+
|          1 | Amit   | 21   | Mumbai |
|          2 | Neha   | 22   | Pune   |
|          4 | Nisha  | 23   | Mumbai |
+-----+-----+-----+-----+
3 rows in set (0.0014 sec)
```

Assignment 2 (Creation of Tables and Implementation of WHERE, ORDER BY and LIKE Clauses using MySQL)

ORDER BY Clause

Command:

```
SELECT * FROM Students
```

```
ORDER BY AGE ASC;
```

```
MySQL localhost:33060+ ssl college_db SQL > SELECT * FROM Students
-> ORDER BY AGE ASC;
```

Output:

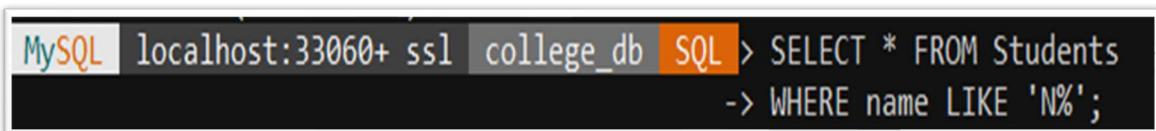
```
+-----+-----+-----+-----+
| student_id | name   | AGE   | city   |
+-----+-----+-----+-----+
|         3  | Rohit | 20    | Delhi  |
|         1  | Amit   | 21    | Mumbai |
|         2  | Neha   | 22    | Pune   |
|         4  | Nisha  | 23    | Mumbai |
+-----+-----+-----+-----+
4 rows in set (0.0013 sec)
```

Assignment 2 (Creation of Tables and Implementation of WHERE, ORDER BY and LIKE Clauses using MySQL)

LIKE Operator

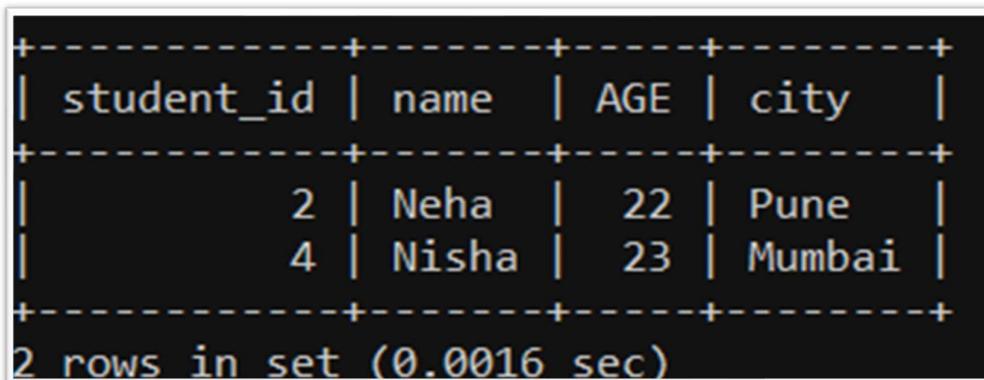
Command:

```
SELECT * FROM Students  
WHERE name LIKE 'N%';
```



The screenshot shows a MySQL command-line interface window. The title bar says "MySQL localhost:33060+ ssl college_db SQL". The main area contains the SQL command: "SELECT * FROM Students -> WHERE name LIKE 'N%'".

Output:



The screenshot shows the output of the MySQL query. It displays a table with four columns: student_id, name, AGE, and city. There are two rows returned: one for student_id 2 (name Neha, AGE 22, city Pune) and one for student_id 4 (name Nisha, AGE 23, city Mumbai). The message "2 rows in set (0.0016 sec)" is at the bottom.

| student_id | name | AGE | city |
|------------|-------|-----|--------|
| 2 | Neha | 22 | Pune |
| 4 | Nisha | 23 | Mumbai |

2 rows in set (0.0016 sec)

Conclusion

Thus, the MySQL database was successfully created and various conditional operations such as WHERE, AND, OR, NOT, ORDER BY, and LIKE were executed successfully using SQL queries.