Department of
Computer Science and Engineering

Title: Querying and Filtering data in MySQL Table

Database System Lab

CSE 210

Green University of Bangladesh

1 Objective(s)

- To gather knowledge about Querying and filtering data in MySQL table.
- To implement distinct and filtering data commands in MySQL table.

2 Problem analysis

The SQL DISTINCT keyword is used with the SELECT statement to eliminate all the duplicate records and fetching only unique records.There may be a situation when you have multiple duplicate records in a table. While fetching such records, it makes more sense to fetch only those unique records instead of fetching duplicate records.
The SQL WHERE clause is used to specify a condition while fetching the data from a single table or by joining with multiple tables. If the given condition is satisfied, then only it returns a specific value from the table.

2.1 Table modification using alter table

The SQL SELECT statement returns a result set of records, from one or more tables. A SELECT statement retrieves zero or more rows from one or more database tables or database views The SELECT DISTINCT statement is used to return only distinct values. Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

- **SELECT Syntax:**

```
SELECT column1, column2, ... FROM table_name;
```

- **SELECT DISTINCT Syntax:**

```
SELECT DISTINCT column1, column2, ... FROM table_name;
```

- **The SQL WHERE Clause syntax:**

```
SELECT column1, column2, ... FROM table_name WHERE condition;
```

3 Procedure (Implementation in MySQL)

1. **Using MySQL SELECT statement to query data(Create a employees table):**

```
CREATE TABLE 'employees'(
Emp_id          int(11)          NOT NULL,
First_Name      varchar(255)     NOT NULL,
Last_name       varchar(55)      NOT NULL,
DOB             date             NOT NULL,
Gender          enum('M','F')    DEFAULT NULL,
Salary          int              NOT NULL,
Entry_date      datetime         NOT NULL,        DEFAULT current_timestamp(),
PRIMARY KEY(Emp_id)
);
```

2. **Insert Multiple VALUES at a time:**

```
INSERT INTO employees (Emp_id, First_Name, Last_name,DOB, Gender, Salary)
VALUES          (1, 'Sabbir', 'Rahman','1998-08-02', 'Male', 30000),
                (2, 'Sakib', 'Hasan','1998-08-02', 'Male', 20000),
                (3, 'Ananna', 'Rahman','1998-08-02', 'Female', 40000),
                (4, 'Jannat', 'Hasan','1998-08-02', 'Female', 45000),
                5, 'Sabbir ', 'Rahman','1998-08-02', 'Male', 30000);
```

3. **View data from table employees:** SELECT * FROM employees;

Emp_id	First_Name	Last_name	DOB	Gender	Salary	Entry_date
1	Sabbir	Rahman	1998-08-02	Male	30000	2021-07-25 16:48:56
2	Sakib	Hasan	1998-08-02	Male	20000	2021-07-25 16:48:56
3	Ananna	Rahman	1998-08-02	Female	40000	2021-07-25 16:48:56
4	Jannat	Hasan	1998-08-02	Female	45000	2021-07-25 16:48:56
5	Sabbir	Rahman	1998-08-02	Male	30000	2021-07-25 16:48:56

Figure 1: Employees Table Information

4. **Eliminating duplicate rows with DISTINCT Operator:**

```
SELECT DISTINCT First_Name, Last_name FROM 'employees';
```

5. **Filtering rows using MySQL WHERE:**

- MySQL WHERE Clause for INETEGER type value :

```
SELECT First_Name, Last_name, Salary FROM 'employees' WHERE Emp_id=2 ;
```

- MySQL WHERE Clause for String type value:

```
SELECT Emp_id, Last_name, DOB, Salary,Entry_date
FROM 'employees'
WHERE First_Name='Sabbir';
```

6. **Using comparison operators (<,>, <=,>=, <>):**

Example:1

```
SELECT Emp_id, First_Name, Last_name
FROM 'employees'
WHERE Salary>=40000;
```

Example 2:

```
SELECT Emp_id, First_Name, Last_name
FROM 'employees'
WHERE Salary <> 30000;                    (<> Means Not Equal to);
```

4 Discussion & Conclusion

Based on the focused objective(s) to understand about the knowledge of SELECT,WHERE and DISTINCT commands. The additional lab exercise made me more confident towards the fulfilment of the objectives(s)

5 Lab Task (Please implement yourself and show the output to the instructor)

1. Insert multiple values at a time for your existing database table (Do it for all table).

2. Remove duplicate values from all table.
3. View the data from any two tables.
4. Implement WHERE clause and search the specific information from existing tables.
5. Implement comparison operators using WHERE clause.

6 Lab Exercise (Submit as a report)

1. Input multiple data in any existing database table from previous lab report.
2. Query with primary key, query with condition, query with comparison operation.
   (Note: Implement All Query which have completed in this experiment)
3. Attach with query codes and with output screenshots in the report.

7 Policy

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.