The University of Faisalabad

LAB: 06

Topic: SQL Data Filtering and Aggregation

Objective:

In this lab, we will learn how to filter data using multiple conditions and perform aggregation operations to analyze datasets efficiently. They will also explore grouping data using the GROUP BY clause.

Step 1: Use the University Database

Before performing operations, ensure that you are working with the correct database:

USE university;

Step 2: Filtering Data Using Multiple Conditions

You can filter records using WHERE with multiple conditions (AND, OR, NOT).

- Retrieve students from New York who have an 'A' grade:

SELECT * FROM students WHERE city = 'New York' AND grade = 'A';

- Retrieve students who are either from Los Angeles or have a grade of 'B':

SELECT * FROM students WHERE city = 'Los Angeles' OR grade = 'B';

- Retrieve students who are NOT from Chicago:

SELECT * FROM students WHERE NOT city = 'Chicago';

Step 3: Using Aggregate Functions

Aggregate functions allow you to perform calculations on data.

- Find the total number of students:

```
SELECT COUNT(*) AS total_students FROM students;
```

- Find the average age of students:

```
SELECT AVG(age) AS average_age FROM students;
```

- Find the youngest and oldest students:

```
SELECT MIN(age) AS youngest, MAX(age) AS oldest FROM students;
```

- Find the number of students in each city:

```
SELECT city, COUNT(*) AS student_count FROM students GROUP BY city;
```

Step 4: Grouping and Filtering Aggregated Data

The GROUP BY clause groups rows with the same values, and the HAVING clause filters aggregated results.

- Retrieve the count of students in each city, but only for cities with more than one student:

```
SELECT city, COUNT(*) AS student_count
FROM students
GROUP BY city
HAVING student_count > 1;
```

- Find the average age of students for each grade:

```
SELECT grade, AVG(age) AS average_age
FROM students
GROUP BY grade;
```

Step 5: Ordering Results with Aggregations

- Retrieve the count of students in each city, sorted in descending order:

```
SELECT city, COUNT(*) AS student_count
FROM students
GROUP BY city
ORDER BY student_count DESC;
```

Step 6: Truncating the Table (Optional)

- If you want to remove all records but keep the table structure:

```
TRUNCATE TABLE students;
```
