Basic SQL Operations

1. Database Creation and Table Setup

- Create a database named University.
- Inside the University database, create the following tables:
  1. Students:
     - StudentID (Primary Key, INT, Auto Increment)
     - Name (VARCHAR, 50)
     - Age (INT)
     - Gender (CHAR(1))
     - DepartmentID (INT, Foreign Key references Departments table)
  2. Departments:
     - DepartmentID (Primary Key, INT)
     - DepartmentName (VARCHAR, 50)

Task: Write SQL queries to:

1. Create the University database.
2. Create the Students and Departments tables.

2. Inserting and Retrieving Data

Insert the following data into the Departments and Students tables:

Departments

DepartmentID	DepartmentName
1	Computer Science
2	AI
3	Business

Students

StudentID	Name	Age	Gender	DepartmentID
1	Ali Ahmed	21	M	1
2	Ayesha Khan	20	F	2
3	Bilal Saeed	22	M	3
4	Maria Tanveer	23	F	1

Task: Write SQL queries to:

1. Insert the above data into the tables.
2. Retrieve all students from the Computer Science department.
3. Retrieve all female students.

3. Updating and Deleting Data

Task:

1. Update Ali Ahmed's department to AI.
2. Delete students older than 22 years.

Medium Level: Intermediate SQL Concepts

4. Joins

Add a Courses table:

- CourseID (Primary Key, INT)
- CourseName (VARCHAR, 50)
- DepartmentID (Foreign Key references Departments)

Courses

CourseID	CourseName	DepartmentID
1	Database Systems	1
2	Machine Learning	2
3	Marketing Strategies	3

Task: Write SQL queries to:

1. Retrieve the names of all students along with their respective department names.
2. List all courses offered by the AI department.

5. Aggregate Functions

Task:

1. Count the total number of students in each department.

2. Find the average age of students in the Business department.
3. Find the youngest student across all departments.

6. Subqueries

Task:

1. Retrieve the names of students who are enrolled in the same department as Ayesha Khan.
2. Find departments with more than 1 student enrolled.

Complex SQL Concepts

7. Stored Procedures and Triggers

Task:

1. Write a stored procedure that inserts a new student into the Students table and automatically assigns the default department as Computer Science if not specified.
2. Create a trigger that updates the department count in a new table DepartmentStatistics whenever a new student is added.

8. Indexes

Task:

1. Add an index to the DepartmentName column in the Departments table. 2. Explain how indexing improves query performance for a SELECT statement.

9. Transactions

Task:

1. Create a transaction to update a student's department and roll back the change if the department doesn't exist.

10. Normalization

Given the following unnormalized table:

StudentID	StudentName	CourseName	DepartmentName
1	Ali Ahmed	Database Systems	Computer Science
2	Ayesha Khan	Machine Learning	AI
3	Bilal Saeed	Marketing Strategies	Business
4	Maria Tanveer	Database Systems	Computer Science

Task:

1. Normalize the table into 3NF and create SQL statements to implement the normalized schema.

11. Optimization

Task:

1. Analyze the execution plan of a complex SELECT query involving multiple joins.
2. Suggest optimizations (e.g., using indexes, reducing joins).

12. Real-World Project

Case Study: University Management System

1. Create tables for Faculty, Exams, and Results.
2. Design relationships between Students, Courses, and Exams.
3. Write queries to:
    - List all students who scored above 80% in Database Systems.
    - Find students who haven't taken any exams.
    - Calculate the average marks per course.