MySQL Assignment: Comprehensive CRUD and Query Operations

Objective:

This assignment focuses on practicing MySQL CRUD operations, filtering data using `WHERE`, `LIMIT`, `ORDER BY`, and aggregate functions, all within a single table: `students`.

Section 1: Database & Table Creation

Task 1: Create a Database

- Create a database named `student_management` and set it as the active database.

Task 2: Create the `students` Table

- Create a table `students` with the following structure:

Column Name	Data Type	Constraints
`student_id`	INT	PRIMARY KEY, AUTO_INCREMENT
`name`	VARCHAR(50)	NOT NULL
`age`	INT	NOT NULL
`gender`	ENUM('Male', 'Female', 'Other')	NOT NULL
`course`	VARCHAR(50)	NOT NULL
`admission_date`	DATE	NOT NULL
`fees_paid`	DECIMAL(10,2)	NOT NULL

Section 2: Data Insertion

Task 3: Insert Sample Records into `students`

- Insert at least **10 student records** into the table using the following sample data:

```
INSERT INTO students (name, age, gender, course, admission_date, fees_paid)
VALUES
('Amit Sharma', 22, 'Male', 'B.Tech', '2024-01-10', 50000.00),
('Priya Singh', 21, 'Female', 'MCA', '2024-02-15', 60000.00),
('Rahul Verma', 23, 'Male', 'MBA', '2023-12-20', 55000.00),
('Neha Patil', 20, 'Female', 'B.Sc', '2024-03-05', 45000.00),
('Suresh Reddy', 24, 'Male', 'B.Tech', '2023-11-10', 52000.00),
('Divya Nair', 22, 'Female', 'MCA', '2024-04-12', 65000.00),
('Vikas Mehta', 25, 'Male', 'BBA', '2023-10-05', 40000.00),
('Ananya Rao', 21, 'Female', 'B.Tech', '2024-01-25', 53000.00),
('Rohan Das', 23, 'Male', 'MBA', '2023-09-15', 57000.00),
('Megha Jain', 22, 'Female', 'B.Sc', '2024-02-20', 48000.00);
```

Section 3: Data Retrieval (SELECT Queries)

Task 4: Basic Data Retrieval

- Retrieve all records from the `students` table.
- Retrieve only the `name`, `course`, and `fees_paid` columns.

Task 5: Conditional Data Retrieval (`WHERE` Clause)

- Retrieve students who are **older than 21 years**.
- Retrieve students who are **enrolled in the "B.Tech" course**.
- Retrieve students who paid **more than ₹50,000** in fees.
- Retrieve students who joined **in the year 2024**.

Task 6: Range-Based Queries (`BETWEEN` & `AND` Clauses)

- Retrieve students whose **fees are between ₹45,000 and ₹55,000**.
- Retrieve students aged **between 21 and 23 years**.

Section 4: Sorting and Limiting Data

Task 7: Sorting (`ORDER BY` Clause)

- Display students in **ascending order of fees paid**.
- Display students in **descending order of admission date**.

Task 8: Limiting Results (`LIMIT` Clause)

- Retrieve the **top 3 students who paid the highest fees**.
- Retrieve the **first 2 students who joined the earliest**.

Section 5: Data Modification (UPDATE & DELETE Queries)

Task 9: Updating Data (`UPDATE` Statement)

- Update the course of a student whose name is `"Amit Sharma"` to `"M.Tech"`.
- Increase the fees of all `"B.Tech"` students by **₹5,000**.

Task 10: Deleting Data (`DELETE` Statement)

- Delete the student record where `student_id = 3`.
- Delete all students who **paid less than ₹50,000**.

Section 6: Aggregate Functions & Data Grouping

Task 11: Data Aggregation (`GROUP BY` Clause)

- Count the total number of students in each course.
- Calculate the **total fees collected per course**.
- Find the **average fees paid** by students.

Submission Guidelines:

- Ensure all queries are well-formatted and executed successfully.
- Submit a `.sql` file or a document containing all executed queries along with their results.

Evaluation Criteria:
✅ Correctness of Queries
✅ Proper Use of SQL Clauses
✅ Optimization and Efficiency
✅ Presentation & Readability of Output

This structured assignment ensures students gain hands-on experience with SQL in a professional and practical manner. 🚀