**✅ 1. CREATE TABLE Schema**

Let’s create a table named students to apply all the logical and ordering concepts:

CREATE TABLE students (

student\_id INT PRIMARY KEY,

full\_name VARCHAR(100),

age INT,

department VARCHAR(50),

gpa DECIMAL(3, 2),

city VARCHAR(50)

);

**✅ 2. INSERT Data**

INSERT INTO students (student\_id, full\_name, age, department, gpa, city) VALUES

(1, 'Ali Raza', 22, 'Computer Science', 3.4, 'Lahore'),

(2, 'Sana Tariq', 21, 'Mathematics', 3.7, 'Karachi'),

(3, 'Hamza Malik', 23, 'Physics', 3.1, 'Islamabad'),

(4, 'Ayesha Khan', 22, 'Computer Science', 3.8, 'Lahore'),

(5, 'Bilal Ahmed', 20, 'Mathematics', 2.9, 'Multan'),

(6, 'Fatima Noor', 24, 'Physics', 3.3, 'Lahore'),

(7, 'Zain Shah', 22, 'Computer Science', 3.6, 'Karachi'),

(8, 'Rabia Aslam', 21, 'Mathematics', 3.5, 'Faisalabad'),

(9, 'Tariq Mehmood', 23, 'Physics', 2.8, 'Lahore'),

(10, 'Hina Saeed', 22, 'Computer Science', 3.9, 'Karachi');

**✅ 3. Apply SQL Concepts with Queries and Explanations**

**🔹 Logical Conditions**

**Query:**

SELECT \* FROM students

WHERE department = 'Computer Science' AND gpa > 3.5;

**Explanation:**  
Returns students from the **Computer Science department** who have a **GPA greater than 3.5**.

* AND is a logical operator: **both conditions must be true**.

**🔹 Rules of Precedence using Parentheses**

**Query:**

SELECT \* FROM students

WHERE department = 'Mathematics' OR (city = 'Lahore' AND gpa >= 3.5);

**Explanation:**

* Returns students who are either:
  + In the **Mathematics department**, **or**
  + From **Lahore** **AND** have **GPA ≥ 3.5**
* **Parentheses override** normal operator precedence (AND evaluated before OR by default).

**🔹 The ORDER BY Clause (Default Ordering)**

**Query:**

SELECT \* FROM students

ORDER BY full\_name;

**Explanation:**  
By default, ORDER BY sorts **in ascending order** (A-Z). This query returns all students **sorted by name**.

**🔹 Reverse the Default Order (Descending)**

**Query:**

SELECT \* FROM students

ORDER BY gpa DESC;

**Explanation:**  
Sorts the records **by GPA from highest to lowest** using DESC.

**🔹 Sorting by Column Aliases**

**Query:**

SELECT full\_name AS Name, gpa AS Grade

FROM students

ORDER BY Grade DESC;

**Explanation:**

* Uses **aliases (AS)** to rename columns for display.
* Orders results using the **alias** Grade instead of original column gpa.

**🔹 Sorting by Multiple Columns**

**Query:**

SELECT \* FROM students

ORDER BY department ASC, gpa DESC;

**Explanation:**

* Sorts data **first by department (A–Z)**.
* Within each department, sorts by **GPA from highest to lowest**.
* Great for grouping + ranking scenarios.

**✅ Summary of What You Learned:**

| **Concept** | **Query Focus** |
| --- | --- |
| Logical Conditions | AND, OR to filter data |
| Precedence with Parentheses | Override default AND > OR rule |
| ORDER BY (default) | Ascending sort (A-Z / low to high) |
| DESC Order | Reverses order (Z-A / high to low) |
| Sorting by Aliases | Use renamed columns in sort clauses |
| Multi-column Sorting | Sort by multiple levels: primary, secondary, etc. |