Data Retrieval Using SQL (Read-Only Queries)

Objective: To understand and practice retrieving and analyzing data from existing database tables using read-only SQL queries, including filtering, sorting, aliasing, and computing derived information, without modifying the database structure or stored data.

<u>Short Notes</u>

1. Data Retrieval

Data retrieval is used to view data stored in a table without changing it.

Example code:

```
SELECT * FROM Student;
```

This displays all records from the Student table.

---

2. Column Alias (AS)

A column alias is a temporary name given to a column in the output to make it easier to understand. It does not change the actual column name in the table.

Example code:

```
SELECT StudentID AS Roll_No, Name AS Student_Name
FROM Student;
```

---

3. Filtering Data (WHERE Clause)

Filtering is used to display only those records that satisfy a given condition.

Example code:

```
SELECT * FROM Student
WHERE DepartmentID = 'D101';
```

This displays students belonging to a specific department.

---

4. Sorting Data (ORDER BY)

Sorting arranges the output in ascending or descending order.

Example code:

```
SELECT Name FROM Student
ORDER BY Name ASC;
```

This displays student names in alphabetical order.

---

5. Limiting Results

Limiting is used to restrict the number of rows shown in the output.

Example code:

```
SELECT * FROM Student
LIMIT 5;
```

This displays only the first five student records.

---

6. Derived or Computed Output

Derived output shows calculated values without changing the stored data.

Example code:

```
SELECT CourseName, Credits + 1 AS Updated_Credits
FROM Course;
```

Credits are increased only in the output, not in the table.

---

7. Working with Dates

Date values can be used to calculate information like age.

Example code:

```
SELECT Name, YEAR(CURDATE()) - YEAR(DOB) AS Age
FROM Student;
```

This calculates age from date of birth.

Lab Questions

Part A: Basic Data Display (Using Aliases)

1. Display all columns from the **Student** table using suitable aliases for every column.
2. Display only `StudentID`, `Name`, and `DepartmentID` from the **Student** table, renaming them as `Roll_No`, `Student_Name`, and `Dept_ID`.
3. Display `FacultyID`, `Name`, `Designation`, and `Email` from the **Faculty** table using readable column names.
4. Display all columns from the **Course** table with renamed column headers.
5. Display all columns from the **Enrollment** table with meaningful aliases.

Part B: Conditional Data Display (WHERE Clause)

6. Display all students whose `DepartmentID` is `'D101'`.
7. Display all students whose `Gender` is `'Female'`.
8. Display faculty members whose `Designation` is `'Assistant Professor'`.
9. Display faculty members whose `DepartmentID` is `'D102'`.
10. Display courses whose `Credits` are **greater than or equal to 4**.
11. Display students born **after `2003-01-01`** using the `DOB` column.
12. Display enrollment records for students enrolled in **Semester 4**.

Part C: Sorting and Limiting Results

13. Display student records sorted by `Name` in ascending order.

14. Display student records sorted by `DOB` in descending order.

15. Display faculty records sorted by `Designation` in ascending order.

16. Display courses sorted by `Credits` in descending order.

17. Display only the **first 3** student records.

18. Display only the **first 5** course records.

Part D: Derived / Computed Output (No Table Modification)

19. Display student `Name` along with their calculated **Age** using `DOB`.

20. Display course `CourseName` along with `Credits + 1` as `Updated_Credits`.

21. Display enrollment details with column `Grade` shown as `Final_Grade`.

22. Display student `Name` along with **year of birth** extracted from `DOB`.

23. Display faculty `Name` along with email domain extracted from `Email`.