**Report on Student Data Analysis**

**1. Introduction**

This report provides an analysis of student data using SQL queries. The queries extract, filter, and organize student-related information from the students and student\_grades tables. The key objectives include identifying students receiving school lunch, sorting students by GPA, calculating average GPAs for different grade levels, limiting results, counting specific records, removing duplicates, and joining tables for combined insights.

**2. Query Methodology**

**2.1 Displaying Students by Name, GPA, and School Lunch Status**

SELECT student\_name, gpa, school\_lunch

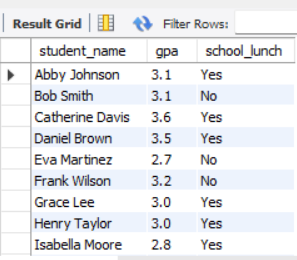
FROM students;

**Purpose:** Retrieves basic student details, including GPA and school lunch eligibility.

**Command Explanation:**

* **SELECT** is used to specify the columns to retrieve.
* **FROM** students determines the table from which data is extracted.

**OUTPUT:**

****

**2.2 Identifying Students Receiving School Lunch with a GPA Above 3.3**

SELECT student\_name, gpa, school\_lunch

FROM students

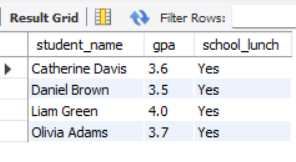
WHERE school\_lunch="Yes" AND gpa>3.3;

**Purpose:** Filters students who receive school lunch and have a GPA greater than 3.3.

**Command Explanation:**

* **WHERE** filters records based on conditions.
* **AND** combines multiple conditions that must be met.

**OUTPUT:**

****

**2.3 Sorting Students by GPA in Descending Order**

SELECT student\_name, gpa, school\_lunch

FROM students

WHERE school\_lunch="Yes" AND gpa>3.3

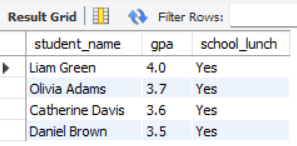
ORDER BY gpa DESC;

**Purpose:** Ranks students (who receive school lunch and have a GPA above 3.3) in descending order of GPA.

**Command Explanation:**

* **ORDER** **BY** sorts the results.
* **DESC** specifies descending order.

**OUTPUT:**

****

**2.4 Calculating the Average GPA for Each Grade Level**

SELECT grade\_level, AVG(gpa)

FROM students

GROUP BY grade\_level

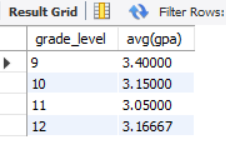
ORDER BY grade\_level;

**Purpose:** Computes the average GPA for each grade level and organizes results in ascending order of grade level.

**Command Explanation:**

* **AVG(gpa)** calculates the average GPA.
* **GROUP** **BY** groups rows with the same grade level.
* **ORDER** **BY** sorts the results by grade level.

**OUTPUT:**

****

**2.5 Identifying Grade Levels with an Average GPA Below 3.3**

SELECT grade\_level, AVG(gpa) AS avg\_gpa

FROM students

GROUP BY grade\_level

HAVING avg\_gpa < 3.3

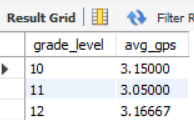
ORDER BY grade\_level;

**Purpose:** Identifies grade levels where the average GPA falls below 3.3.

**Command Explanation:**

* **HAVING** filters grouped results based on an aggregate function.

**OUTPUT:**

****

**2.6 Displaying a Limited Number of Rows**

SELECT student\_name, gpa, school\_lunch

FROM students

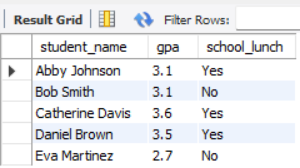
LIMIT 5;

**Purpose:** Limits the number of displayed student records to 5.

**Command Explanation:**

* **LIMIT** restricts the number of rows returned.

**OUTPUT:**

****

**2.7 Counting Students Who Receive School Lunch and Have a GPA Above 3.3**

SELECT COUNT(\*)

FROM students

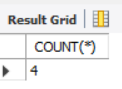
WHERE school\_lunch="Yes" AND gpa>3.3;

**Purpose:** Counts the number of students meeting the criteria.

**Command Explanation:**

* **COUNT**(\*) counts all rows satisfying the condition.

**OUTPUT:**

****

**2.8 Removing Duplicate GPA Values**

SELECT DISTINCT(gpa)

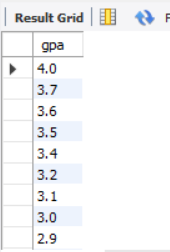
FROM students

ORDER BY gpa DESC;

**Purpose:** Retrieves unique GPA values in descending order.

**Command Explanation:**

* **DISTINCT** ensures only unique values are selected.

**OUTPUT:  
**

**2.9 Displaying All Data from the student\_grades Table**

SELECT \*

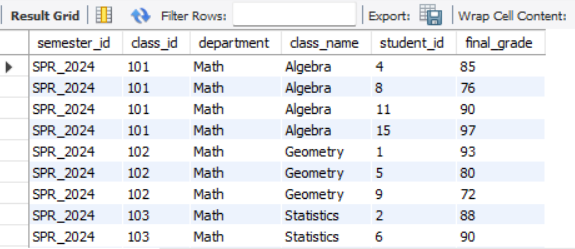
FROM student\_grades;

**Purpose:** Retrieves all available data from the student\_grades table for further analysis.

**Command Explanation:**

* **SELECT** \* selects all columns from the table.

**OUTPUT:**

****

**2.10 Combining Student and Grade Information Using a Left Join**

SELECT students.id, students.student\_name,

student\_grades.class\_name, student\_grades.final\_grade

FROM students

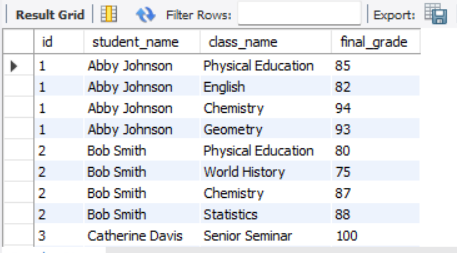
LEFT JOIN student\_grades

ON students.id = student\_grades.student\_id;

**Purpose:** Merges student data with their grades using a left join.

**Command Explanation:**

* **LEFT** **JOIN** retrieves all records from the left table (students) and matching records from the right table (student\_grades).
* ON specifies the condition for joining tables.



**3. Results and Analysis**

**3.1 Student Information**

* The first query retrieved a comprehensive list of students with their GPA and school lunch status.
* The second query identified students receiving school lunch with a high GPA (>3.3).
* The third query ranked these students by GPA in descending order.

**3.2 GPA Analysis**

* The fourth query showed the average GPA per grade level.
* The fifth query highlighted grade levels where the average GPA was below 3.3, helping identify areas where academic performance might need improvement.
* The eighth query extracted distinct GPA values, eliminating duplicates for better data integrity.

**3.3 Student Grades Data**

* The sixth query limited student results to the top five.
* The seventh query counted students matching a certain condition.
* The ninth query extracted all records from student\_grades for potential further insights.
* The tenth query combined student and grade data, providing a holistic view of student performance.

**4. Conclusion & Recommendations**

**Key Findings**

* Several students with a high GPA (>3.3) receive school lunch, indicating a potential correlation between lunch program participation and academic performance.
* Some grade levels have an average GPA below 3.3, requiring further investigation into possible causes.
* The student\_grades table contains additional data that could be leveraged for deeper analysis.
* Removing duplicate GPAs provides cleaner data for evaluation.
* Using joins enhances insights by linking student data with their academic performance.

**Recommendations**

* Focus on supporting lower-performing grade levels through targeted interventions.
* Investigate additional factors influencing GPA trends, such as attendance or extracurricular participation.
* Utilize the student\_grades table to analyze individual subject performances.
* Implement data cleaning techniques to ensure accurate reporting.

**5. Appendix**

* Additional queries can be executed for more detailed insights.
* Further exploration of other student attributes could be beneficial for a holistic analysis.