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:

Result Grid | Filter Rows:

student_name	gpa	school_lunch
Abby Johnson	3.1	Yes
Bob Smith	3.1	No
Catherine Davis	3.6	Yes
Daniel Brown	3.5	Yes
Eva Martinez	2.7	No
Frank Wilson	3.2	No
Grace Lee	3.0	Yes
Henry Taylor	3.0	Yes
Isabella Moore	2.8	Yes

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:

Result Grid | Filter Rows:

student_name	gpa	school_lunch
Catherine Davis	3.6	Yes
Daniel Brown	3.5	Yes
Liam Green	4.0	Yes
Olivia Adams	3.7	Yes

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:

Result Grid | Filter Rows:

student_name	gpa	school_lunch
Liam Green	4.0	Yes
Olivia Adams	3.7	Yes
Catherine Davis	3.6	Yes
Daniel Brown	3.5	Yes

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:

Result Grid | Filter Rows:

grade_level	avg(gpa)
9	3.40000
10	3.15000
11	3.05000
12	3.16667

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:

Result Grid | Filter R

grade_level	avg_gps
10	3.15000
11	3.05000
12	3.16667

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:

Result Grid | Filter Rows:

student_name	gpa	school_lunch
Abby Johnson	3.1	Yes
Bob Smith	3.1	No
Catherine Davis	3.6	Yes
Daniel Brown	3.5	Yes
Eva Martinez	2.7	No

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:

Result Grid

	COUNT(*)
▶	4

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:

Result Grid

	gpa
▶	4.0
	3.7
	3.6
	3.5
	3.4
	3.2
	3.1
	3.0
	2.9

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:

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

semester_id	class_id	department	class_name	student_id	final_grade
SPR_2024	101	Math	Algebra	4	85
SPR_2024	101	Math	Algebra	8	76
SPR_2024	101	Math	Algebra	11	90
SPR_2024	101	Math	Algebra	15	97
SPR_2024	102	Math	Geometry	1	93
SPR_2024	102	Math	Geometry	5	80
SPR_2024	102	Math	Geometry	9	72
SPR_2024	103	Math	Statistics	2	88
SPR_2024	103	Math	Statistics	6	90

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.

Result Grid | Filter Rows: | Export:

id	student_name	class_name	final_grade
1	Abby Johnson	Physical Education	85
1	Abby Johnson	English	82
1	Abby Johnson	Chemistry	94
1	Abby Johnson	Geometry	93
2	Bob Smith	Physical Education	80
2	Bob Smith	World History	75
2	Bob Smith	Chemistry	87
2	Bob Smith	Statistics	88
3	Catherine Davis	Senior Seminar	100

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.
