*Count of students in the database*

SELECT COUNT(\*) AS total\_students FROM Student;

1. *Distribution of students by gender*

SELECT gender, COUNT(\*) AS count

FROM Student

GROUP BY gender

ORDER BY count DESC;

1. *Find students with GPA above 3.0*

SELECT student\_id, gender, gpa, grade\_class

FROM Student

WHERE gpa > 3.0

ORDER BY gpa DESC;

1. *List students with their ethnicity names*

SELECT s.student\_id, s.gender, e.ethnicity\_name, s.gpa

FROM Student s

JOIN Ethnicity e ON s.ethnicity\_id = e.ethnicity\_id

ORDER BY s.student\_id;

1. *Count of students by grade class*

SELECT grade\_class, COUNT(\*) AS student\_count

FROM Student

GROUP BY grade\_class

ORDER BY grade\_class;

1. *Average GPA by parental education level*

SELECT p.education\_level, AVG(s.gpa) AS average\_gpa, COUNT(\*) AS student\_count

FROM Student s

JOIN ParentalEducationLevel p ON s.parental\_education\_id = p.education\_id

GROUP BY p.education\_level

ORDER BY average\_gpa DESC;

1. *Impact of study time on GPA*

SELECT

CASE

WHEN study\_time\_weekly < 5 THEN 'Low (< 5 hrs)'

WHEN study\_time\_weekly >= 5 AND study\_time\_weekly < 10 THEN 'Medium (5-10 hrs)'

WHEN study\_time\_weekly >= 10 AND study\_time\_weekly < 15 THEN 'High (10-15 hrs)'

ELSE 'Very High (15+ hrs)'

END AS study\_time\_category,

COUNT(\*) AS student\_count,

AVG(gpa) AS average\_gpa,

MIN(gpa) AS min\_gpa,

MAX(gpa) AS max\_gpa

FROM Student

GROUP BY study\_time\_category

ORDER BY AVG(study\_time\_weekly);

1. *Relationship between absences and GPA*

SELECT

CASE

WHEN absences = 0 THEN 'No Absences'

WHEN absences BETWEEN 1 AND 5 THEN '1-5'

WHEN absences BETWEEN 6 AND 10 THEN '6-10'

WHEN absences BETWEEN 11 AND 15 THEN '11-15'

WHEN absences BETWEEN 16 AND 20 THEN '16-20'

ELSE '20+'

END AS absence\_category,

COUNT(\*) AS student\_count,

AVG(gpa) AS average\_gpa

FROM Student

GROUP BY absence\_category

ORDER BY

CASE absence\_category

WHEN 'No Absences' THEN 1

WHEN '1-5' THEN 2

WHEN '6-10' THEN 3

WHEN '11-15' THEN 4

WHEN '16-20' THEN 5

ELSE 6

END;

1. *Effect of tutoring and parental support*

SELECT

CASE

WHEN tutoring = 1 THEN 'Receives Tutoring'

ELSE 'No Tutoring'

END AS tutoring\_status,

parental\_support,

COUNT(\*) AS student\_count,

AVG(gpa) AS average\_gpa

FROM Student

GROUP BY tutoring\_status, parental\_support

ORDER BY average\_gpa DESC;

9. *Course enrollment details with student and course information*

SELECT e.enrollment\_id, s.student\_id, CONCAT(s.first\_name, ' ', s.last\_name) AS student\_name,

c.course\_code, c.course\_name, e.semester, e.academic\_year, e.grade

FROM Enrollment e

JOIN Student s ON e.student\_id = s.student\_id

JOIN Course c ON e.course\_id = c.course\_id

ORDER BY e.academic\_year, e.semester, s.last\_name, s.first\_name;

*10.. Window Functions: Rank students by GPA within each department*

SELECT

s.student\_id,

CONCAT(s.first\_name, ' ', s.last\_name) AS student\_name,

d.department\_name,

s.gpa,

RANK() OVER (PARTITION BY d.department\_id ORDER BY s.gpa DESC) AS dept\_rank

FROM Student s

JOIN Enrollment e ON s.student\_id = e.student\_id

JOIN Course c ON e.course\_id = c.course\_id

JOIN Department d ON c.department\_id = d.department\_id

GROUP BY s.student\_id, student\_name, d.department\_id, d.department\_name, s.gpa

ORDER BY d.department\_name, dept\_rank;

*11. Identifying At-Risk Students: Find students with declining performance*

SELECT

s.student\_id,

CONCAT(s.first\_name, ' ', s.last\_name) AS student\_name,

c.course\_code,

COUNT(a.assignment\_id) AS num\_assignments,

ROUND(

(SUM(CASE WHEN g.points\_earned/a.max\_points \* 100 < 70 THEN 1 ELSE 0 END) \* 100.0) /

COUNT(a.assignment\_id),

2) AS percent\_low\_grades

FROM Student s

JOIN Grade g ON s.student\_id = g.student

10. List all students who have received tutoring.

SELECT \* FROM StudyHabit WHERE Tutoring = 1;

11. Find the average GPA of male students.

SELECT AVG(GPA) AS Average\_GPA FROM AcademicPerformance JOIN Student ON AcademicPerformance.StudentID = Student.StudentID WHERE Gender = 'Male';

12. Find the average study time per week for students grouped by parental education level.

SELECT AVG(StudyTimeWeekly) AS Avg\_StudyTime\_Weekly, ParentalEducation FROM StudyHabit JOIN ParentalInvolvement ON StudyHabit.StudentID = ParentalInvolvement.StudentID GROUP BY ParentalEducation;

13. Find all pairs of students who have the same ParentalEducation level and whose GPA differs by less than 0.5. List their IDs and GPAs.

SELECT s1.StudentID AS StudentID1, ap1.GPA AS GPA1, s2.StudentID AS StudentID2, ap2.GPA AS GPA2, p1.ParentalEducation FROM AcademicPerformance ap1 JOIN ParentalInvolvement p1 ON ap1.StudentID = p1.StudentID JOIN AcademicPerformance ap2 ON ap1.StudentID < ap2.StudentID JOIN ParentalInvolvement p2 ON ap2.StudentID = p2.StudentID JOIN Student s1 ON ap1.StudentID = s1.StudentID JOIN Student s2 ON ap2.StudentID = s2.StudentID WHERE p1.ParentalEducation = p2.ParentalEducation AND ABS(ap1.GPA - ap2.GPA) < 0.5;

14. Create a trigger to log a warning into a **separate AbsenceWarnings table** whenever a student has more than 15 absences.

CREATE TABLE AbsenceWarnings (

StudentID INT,

Absences INT,

WarningDate DATETIME DEFAULT CURRENT\_TIMESTAMP

);

DELIMITER $$ CREATE TRIGGER log\_absence\_warning AFTER INSERT ON StudyHabit FOR EACH ROW BEGIN IF NEW.Absences > 15 THEN INSERT INTO AbsenceWarnings (StudentID, Absences) VALUES (NEW.StudentID, NEW.Absences); END IF; END$$ DELIMITER ;

15. Find students who are involved in two or more extracurricular activities

SELECT StudentID,

(Extracurricular + Sports + Music + Volunteering) AS TotalActivities

FROM ExtracurricularActivities

WHERE (Extracurricular + Sports + Music + Volunteering) >= 2;

16.Which gender has a higher proportion of students receiving tutoring?

SELECT Gender, COUNT(\*) AS Tutored\_Count

FROM Student s

JOIN StudyHabit sh ON s.StudentID = sh.StudentID

WHERE sh.Tutoring = 1

GROUP BY Gender

ORDER BY Tutored\_Count DESC;

17. Which parental support level is associated with the highest average GPA?

SELECT ParentalSupport, AVG(GPA) AS Avg\_GPA

FROM ParentalInvolvement pi

JOIN AcademicPerformance ap ON pi.StudentID = ap.StudentID

GROUP BY ParentalSupport

ORDER BY Avg\_GPA DESC;

18. Categorize students by GPA into performance tiers

SELECT StudentID, GPA,

CASE

WHEN GPA >= 3.5 THEN 'Excellent'

WHEN GPA >= 3.0 THEN 'Good'

WHEN GPA >= 2.0 THEN 'Average'

ELSE 'Poor'

END AS PerformanceCategory

FROM AcademicPerformance;

19. Rank Students by GPA and Study Time Combined

SELECT

s.StudentID,

ap.GPA,

sh.StudyTimeWeekly,

ROUND(ap.GPA + (sh.StudyTimeWeekly \* 0.1), 2) AS composite\_score

FROM Student s

JOIN AcademicPerformance ap ON s.StudentID = ap.StudentID

JOIN StudyHabit sh ON s.StudentID = sh.StudentID

ORDER BY composite\_score DESC

LIMIT 10;

20. Compare GPA Distribution Between Tutored and Non-Tutored Students

SELECT

CASE

WHEN sh.Tutoring = 1 THEN 'Tutored'

ELSE 'Not Tutored'

END AS tutoring\_group,

COUNT(\*) AS student\_count,

ROUND(AVG(ap.GPA), 2) AS avg\_gpa,

ROUND(STDDEV(ap.GPA), 2) AS gpa\_stddev,

MIN(ap.GPA) AS min\_gpa,

MAX(ap.GPA) AS max\_gpa

FROM StudyHabit sh

JOIN AcademicPerformance ap ON sh.StudentID = ap.StudentID

GROUP BY tutoring\_group

ORDER BY avg\_gpa DESC;

21.)Which students are putting in a lot of study time but not getting good grades?  
SELECT

s.StudentID,

sh.StudyTimeWeekly,

ap.GPA,

ROUND(ap.GPA / NULLIF(sh.StudyTimeWeekly, 0), 2) AS gpa\_efficiency

FROM Student s

JOIN StudyHabit sh ON s.StudentID = sh.StudentID

JOIN AcademicPerformance ap ON s.StudentID = ap.StudentID

WHERE sh.StudyTimeWeekly >= 10 AND ap.GPA < 2.5

ORDER BY gpa\_efficiency ASC;

22.)Within students with low parental support, who exceeds the group average GPA?

SELECT

s.StudentID,

ap.GPA,

pi.ParentalSupport,

group\_stats.avg\_gpa AS group\_avg\_gpa

FROM Student s

JOIN ParentalInvolvement pi ON s.StudentID = pi.StudentID

JOIN AcademicPerformance ap ON s.StudentID = ap.StudentID

JOIN (

SELECT pi.ParentalSupport, AVG(ap.GPA) AS avg\_gpa

FROM ParentalInvolvement pi

JOIN AcademicPerformance ap ON pi.StudentID = ap.StudentID

GROUP BY pi.ParentalSupport

) group\_stats ON pi.ParentalSupport = group\_stats.ParentalSupport

WHERE pi.ParentalSupport = 'Low' AND ap.GPA > group\_stats.avg\_gpa

ORDER BY ap.GPA DESC;

23. Which pair of students have the same parental education level, very similar GPAs (within 0.2), and completely different activity profiles?  
SELECT

s1.StudentID AS Student1,

s2.StudentID AS Student2,

pi1.ParentalEducation,

ap1.GPA AS GPA1,

ap2.GPA AS GPA2,

ROUND(ABS(ap1.GPA - ap2.GPA), 2) AS GPA\_Difference,

CONCAT(

'S1: E', ea1.Extracurricular, ' S', ea1.Sports,

' M', ea1.Music, ' V', ea1.Volunteering,

' | S2: E', ea2.Extracurricular, ' S', ea2.Sports,

' M', ea2.Music, ' V', ea2.Volunteering

) AS ActivityComparison

FROM Student s1

JOIN Student s2

ON s1.StudentID < s2.StudentID

JOIN ParentalInvolvement pi1 ON s1.StudentID = pi1.StudentID

JOIN ParentalInvolvement pi2 ON s2.StudentID = pi2.StudentID

AND pi1.ParentalEducation = pi2.ParentalEducation

JOIN AcademicPerformance ap1 ON s1.StudentID = ap1.StudentID

JOIN AcademicPerformance ap2 ON s2.StudentID = ap2.StudentID

AND ABS(ap1.GPA - ap2.GPA) <= 0.2

JOIN ExtracurricularActivities ea1 ON s1.StudentID = ea1.StudentID

JOIN ExtracurricularActivities ea2 ON s2.StudentID = ea2.StudentID

WHERE

(ea1.Extracurricular <> ea2.Extracurricular OR

ea1.Sports <> ea2.Sports OR

ea1.Music <> ea2.Music OR

ea1.Volunteering <> ea2.Volunteering)

ORDER BY GPA\_Difference;

