1. Find the second youngest female student.

Select \* from students01  
where gender = ‘F’

Order by age

Limit 1 offset 1;

1. Retrieve the student(s) who have the same first and last names.

SELECT \*

FROM students

WHERE first\_name = last\_name;

1. List students whose enrollment\_date is the earliest among all.

SELECT \*

FROM students WHERE enrollment\_date = (SELECT MIN (enrollment\_date) FROM students);

1. Find the grade that has the maximum number of students.

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

FROM students

GROUP BY grade

ORDER BY student\_count DESC

LIMIT 1;

1. Display students whose email domain is 'example.com'.

SELECT \*

FROM students

WHERE email LIKE '%@example.com';

1. List students whose last name appears more than once.

SELECT \*

FROM students

WHERE last\_name IN (

SELECT last\_name

FROM students

GROUP BY last\_name

HAVING COUNT(\*) > 1

);

1. Find students who enrolled in the same month but different years.

SELECT s1.\*

FROM students s1

JOIN students s2

ON MONTH(s1.enrollment\_date) = MONTH(s2.enrollment\_date)

AND YEAR(s1.enrollment\_date) != YEAR(s2.enrollment\_date);

1. Retrieve students who have a 'C+' or 'B-' grade and age > 22.

SELECT \*

FROM students

WHERE grade IN ('C+', 'B-')

AND age > 22;

1. List students whose first name ends with 'a' or starts with 'H'.

SELECT \*

FROM students

WHERE first\_name LIKE '%a' OR first\_name LIKE 'H%';

1. Find the students who have duplicate emails ignoring case sensitivity.

SELECT email, COUNT(\*)

FROM students

GROUP BY LOWER(email)

HAVING COUNT(\*) > 1;

1. Retrieve top 3 grades with the highest average student age.

SELECT grade, AVG(age) AS avg\_age

FROM students

GROUP BY grade

ORDER BY avg\_age DESC

LIMIT 3;

1. Find students who enrolled exactly 10 days before today's date.

SELECT \*

FROM students

WHERE enrollment\_date = CURDATE() - INTERVAL 10 DAY;

1. List students where the combination of first name and last name is not unique.

SELECT first\_name, last\_name

FROM students

GROUP BY first\_name, last\_name

HAVING COUNT(\*) > 1;

1. Find how many students are there in each (first\_name, last\_name) pair.

SELECT first\_name, last\_name, COUNT(\*) AS student\_count

FROM students

GROUP BY first\_name, last\_name;

1. List students whose enrollment month is August and grade is either 'A' or 'A-'.

SELECT \* FROM students

WHERE MONTH(enrollment\_date) = 8

AND grade IN ('A', 'A-');

1. Find the youngest male student with grade 'A' or higher.

SELECT \*

FROM students

WHERE grade IN ('A', 'A+', 'A-') AND gender = 'Male'

ORDER BY age ASC

LIMIT 1;

17. Retrieve the list of distinct last names where more than two students share it.

SELECT last\_name

FROM students

GROUP BY last\_name

HAVING COUNT(\*) > 2;

1. Find students who have the exact same enrollment\_date and age.

SELECT \*

FROM students s1

JOIN students s2 ON s1.enrollment\_date = s2.enrollment\_date

AND s1.age = s2.age AND s1.student\_id != s2.student\_id;

1. Find students whose grade is one letter higher than 'B' alphabetically.

SELECT \*

FROM students

WHERE grade = 'B+';

1. List the third highest enrolled student based on enrollment\_date.

SELECT \*

FROM (

SELECT \*, ROW\_NUMBER() OVER (ORDER BY enrollment\_date DESC) AS rank FROM students) AS ranked\_students

WHERE rank = 3;