Question 1: Fetch all students

```
SELECT * FROM students;
```

Question 2: Show only name and department

```
SELECT name, department FROM students;
```

Question 3: Show distinct departments

```
SELECT DISTINCT department FROM students;
```

Question 4: Students older than 20

```
SELECT * FROM students WHERE age > 20;
```

Question 5: Students from Mumbai

```
SELECT * FROM students WHERE city = 'Mumbai';
```

Question 6: Students with marks above 80

```
SELECT * FROM students WHERE marks > 80;
```

Question 7: Students whose name starts with A

```
SELECT * FROM students WHERE name LIKE 'A%';
```

Question 8: Students with age between 18 and 22

```
SELECT * FROM students WHERE age BETWEEN 18 AND 22;
```

Question 9: Students from IT or CS department

```
SELECT * FROM students WHERE department IN ('IT','CS');
```

Question 10: Sort students by marks descending

```
SELECT * FROM students ORDER BY marks DESC;
```

Question 11: Top 10 highest scoring students

```
SELECT * FROM students ORDER BY marks DESC LIMIT 10;
```

Question 12: Students whose attendance is below 60

```sql
SELECT * FROM students WHERE attendance < 60;
```

Question 13: Students not from Delhi

```sql
SELECT * FROM students WHERE city <> 'Delhi';
```

Question 14: Students admitted after 2023

```sql
SELECT * FROM students WHERE admission_date > '2023-01-01';
```

Question 15: Students with marks between 70 and 90

```sql
SELECT * FROM students WHERE marks BETWEEN 70 AND 90;
```

Question 16: Students whose name ends with a

```sql
SELECT * FROM students WHERE name LIKE '%a';
```

Question 17: Students whose fees_paid is NULL

```sql
SELECT * FROM students WHERE fees_paid IS NULL;
```

Question 18: Students with attendance above 90 and marks above 75

```sql
SELECT * FROM students WHERE attendance > 90 AND marks > 75;
```

Question 19: Show students ordered by department then marks

```sql
SELECT * FROM students ORDER BY department, marks DESC;
```

Question 20: Fetch only first 5 records

```sql
SELECT * FROM students LIMIT 5;
```

Question 21: Count total students

```sql
SELECT COUNT(*) FROM students;
```

Question 22: Average marks of all students

```sql
SELECT AVG(marks) FROM students;
```

Question 23: Maximum marks scored

```sql
SELECT MAX(marks) FROM students;
```

Question 24: Minimum attendance

```
SELECT MIN(attendance) FROM students;
```

Question 25: Total fees collected

```
SELECT SUM(fees_paid) FROM students;
```

Question 26: Number of students in each department

```
SELECT department, COUNT(*) FROM students GROUP BY department;
```

Question 27: Average marks per department

```
SELECT department, AVG(marks) FROM students GROUP BY department;
```

Question 28: Departments having more than 50 students

```
SELECT department, COUNT(*) FROM students GROUP BY department HAVING
COUNT(*) > 50;
```

Question 29: Cities where average marks > 75

```
SELECT city, AVG(marks) FROM students GROUP BY city HAVING AVG(marks)
> 75;
```

Question 30: Highest marks in each department

```
SELECT department, MAX(marks) FROM students GROUP BY department;
```

Question 31: Add grade column using CASE

```
SELECT name, CASE WHEN marks >= 90 THEN 'A' WHEN marks >= 75 THEN 'B'
WHEN marks >= 60 THEN 'C' ELSE 'D' END AS grade FROM students;
```

Question 32: Students who paid more than average fees

```
SELECT * FROM students WHERE fees_paid > (SELECT AVG(fees_paid) FROM
students);
```

Question 33: Students having marks above department average

```
SELECT * FROM students s WHERE marks > (SELECT AVG(marks) FROM
students WHERE department = s.department);
```

Question 34: Second highest marks

```
SELECT MAX(marks) FROM students WHERE marks < (SELECT MAX(marks) FROM students);
```

Question 35: Top 3 students by marks

```
SELECT * FROM students ORDER BY marks DESC LIMIT 3;
```

Question 36: Students with duplicate names

```
SELECT name, COUNT(*) FROM students GROUP BY name HAVING COUNT(*) > 1;
```

Question 37: Students admitted in last 30 days

```
SELECT * FROM students WHERE admission_date >= CURDATE() - INTERVAL 30 DAY;
```

Question 38: Show marks category

```
SELECT name, CASE WHEN marks >= 80 THEN 'High' WHEN marks >= 50 THEN 'Medium' ELSE 'Low' END AS category FROM students;
```

Question 39: Students whose attendance is above overall average

```
SELECT * FROM students WHERE attendance > (SELECT AVG(attendance) FROM students);
```

Question 40: Department with highest average marks

```
SELECT department, AVG(marks) avg_marks FROM students GROUP BY department ORDER BY avg_marks DESC LIMIT 1;
```

Question 41: Fetch latest admitted student

```
SELECT * FROM students ORDER BY admission_date DESC LIMIT 1;
```

Question 42: Students whose name length > 5

```
SELECT * FROM students WHERE LENGTH(name) > 5;
```

Question 43: Replace NULL fees with 0

```
SELECT name, IFNULL(fees_paid,0) FROM students;
```

Question 44: Students whose city starts with M

```
SELECT * FROM students WHERE city LIKE 'M%';
```

Question 45: Students whose marks are even numbers

```
SELECT * FROM students WHERE marks % 2 = 0;
```

Question 46: Students with highest attendance

```
SELECT * FROM students WHERE attendance = (SELECT MAX(attendance) FROM
students);
```

Question 47: Students scoring above overall average marks

```
SELECT * FROM students WHERE marks > (SELECT AVG(marks) FROM
students);
```

Question 48: Count male and female students separately

```
SELECT gender, COUNT(*) FROM students GROUP BY gender;
```

Question 49: Departments sorted by student count descending

```
SELECT department, COUNT(*) cnt FROM students GROUP BY department
ORDER BY cnt DESC;
```

Question 50: Students who joined this year

```
SELECT * FROM students WHERE YEAR(admission_date) = YEAR(CURDATE());
```