**Changing from main**

**Creating student table**

CREATE TABLE student (

student\_id INT PRIMARY KEY AUTO\_INCREMENT,

student\_name VARCHAR(100),

gender VARCHAR(10),

dob DATE,

class VARCHAR(20)

);

**Creating student marks table**

**CREATE TABLE student\_marks (**

**mark\_id INT PRIMARY KEY AUTO\_INCREMENT,**

**student\_id INT,**

**subject VARCHAR(50),**

**marks INT,**

**exam\_date DATE,**

**FOREIGN KEY (student\_id) REFERENCES student(student\_id)**

**);**

**1.Write a query to list all students with their marks (INNER JOIN).**

**select s.student\_name,s1.subject,s1.marks**

**from student s**

**join student\_marks s1**

**on s.student\_id=s1.student\_id;**

**2.Show all students, including those who don’t have marks yet (LEFT JOIN).**

**select s.student\_name,s1.subject,s1.marks**

**from student s**

**left join student\_marks s1**

**on s.student\_id=s1.student\_id;**

**3. Find students who appeared in exams after ‘2025-01-01’.**

**select s1.exam\_date,s.student\_name**

**from student s**

**join student\_marks s1 on s.student\_id=s1.student\_id**

**where s1.exam\_date>2025-01-01;**

**4.** **List students with their total marks across all subjects.**

**select s.student\_name,sum(s1.marks)as total\_marks**

**from student s**

**join student\_marks s1 on s.student\_id=s1.student\_id**

**group by s1.student\_id;**

**5.** **Find the highest mark in each subject with student details.**

**select s.student\_name,s.gender,s.class,sm.subject,sm.marks,sm.exam\_date**

**from student\_marks sm**

**join student s**

**on sm.student\_id = s.student\_id**

**join (**

**select subject, max(marks) as max\_marks**

**from student\_marks**

**group by subject**

**) as max\_per\_subject**

**on sm.subject = max\_per\_subject.subject**

**and sm.marks = max\_per\_subject.max\_marks;**

**6.** **Use AVG() to find the average marks per subject.**

**select subject,avg(marks)**

**from student\_marks**

**group by subject order by avg(marks) asc;**

**7.Use COUNT() to find how many subjects each student appeared in.**

**select s.student\_name,COUNT(sm.subject) AS num\_of\_subjects**

**from student s**

**left join student\_marks sm on s.student\_id = sm.student\_id**

**group by s.student\_name;**

**8.Use ROUND() to display percentage (marks/total \* 100) rounded to 2 decimals.**

**select s.student\_name,sm.subject,sm.marks,ROUND((sm.marks / 100.0) \* 100, 2) as percentage**

**from student s**

**join student\_marks sm**

**on s.student\_id = sm.student\_id;**

**9.Use UPPER() to display all student names in uppercase.**

**select upper(student\_name) as student\_name\_upper from student;**

**10.Use YEAR(dob) to find students born after 2005.**

**select student\_name,dob from student where year(dob)>2005;**

**11.Use LENGTH(student\_name) to find the longest name student.**

**select student\_name,length(student\_name)as name\_length from student order by length(student\_name) desc;**

**12.Write a query to categorize marks as:**

**>=90 → "Outstanding",75–89 → "Excellent",60–74 → "Good",40–59 → "Pass",<40 → "Fail"**

**select**

**s.student\_name,sm.subject,sm.marks,**

**case**

**when marks>=90 then 'Outstanding'**

**when marks between 75 and 89 then 'Excellent'**

**when marks>=60 and marks<=74 then 'Good'**

**when marks between 40 and 59 then 'pass'**

**else 'fail'**

**end as performance**

**FROM student s**

**JOIN student\_marks sm**

**ON s.student\_id = sm.student\_id;**

**13.Create a query to display:**

**Gender as "M → Male", "F → Female", "Other → Unknown" using CASE.**

**select student\_name,gender,**

**case**

**when gender='M' then 'Male'**

**when gender='F' then 'Female'**

**end as gender**

**from student ;**

**14.Write a query to display scholarship eligibility:**

**If average ≥85 → "Eligible",Else → "Not Eligible"**

**select s.student\_name,avg(sm.marks),**

**case**

**when avg(sm.marks)>=85 then 'Eligible'**

**else 'Not Eligible'**

**end as Schlorship\_status**

**from student s join student\_marks sm on s.student\_id=sm.student\_id**

**group by student\_name;**