**Task # 3**

**CREATE TABLES**CREATE TABLE Students (  
    student\_id INT PRIMARY KEY,  
    name VARCHAR(50),  
    age INT,  
    gender VARCHAR(10),  
    class\_id INT  
);CREATE TABLE Classes (  
    class\_id INT PRIMARY KEY,  
    class\_name VARCHAR(50),  
    teacher\_id INT  
);CREATE TABLE Teachers (  
    teacher\_id INT PRIMARY KEY,  
    name VARCHAR(50),  
    subject VARCHAR(50)  
);CREATE TABLE Marks (  
    mark\_id INT PRIMARY KEY,  
    student\_id INT,  
    subject VARCHAR(50),  
    marks INT  
);

**INSERT DATA**-- Students  
INSERT INTO Students (student\_id, name, age, gender, class\_id) VALUES  
(1, 'Ahmed', 17, 'Male', 1),  
(2, 'Sara', 18, 'Female', 2),  
(3, 'Ali', 19, 'Male', 1),  
(4, 'Ayesha', 17, 'Female', 3),  
(5, 'Usman', 21, 'Male', 2),  
(6, 'Zara', 22, 'Female', 3),  
(7, 'Hassan', 20, 'Male', 1);-- Classes  
INSERT INTO Classes (class\_id, class\_name, teacher\_id) VALUES  
(1, 'Class 10', 101),  
(2, 'Class 9', 102),  
(3, 'Class 8', 103);-- Teachers  
INSERT INTO Teachers (teacher\_id, name, subject) VALUES  
(101, 'Mr. Khan', 'Math'),  
(102, 'Ms. Fatima', 'Science'),

(103, 'Mr. Bilal', 'English');-- Marks

INSERT INTO Marks (mark\_id, student\_id, subject, marks) VALUES  
(1, 1, 'Math', 88),  
(2, 2, 'Science', 75),  
(3, 3, 'Math', 90),  
(4, 4, 'English', 65),  
(5, 5, 'Science', 95),  
(6, 6, 'English', 85),  
(7, 7, 'Math', 72),  
(8, 1, 'Science', 70),  
(9, 2, 'Math', 67),  
(10, 4, 'Math', 78);

**QURIES TO IMPLEMENT**

1. Write a query to get the names of all student
2. Get the names of all male students.
3. Find all students older than 18 years.
4. Get details of students who are in class\_id = 2.
5. List all students ordered by age, youngest first.
6. Show top 5 students with the highest marks in "Math".
7. List student names along with their class names.
8. Show student names with their teacher’s name for each class.
9. Find the average marks for each subject.
10. Count how many students are in each class.
11. Find the highest marks scored in "Science".
12. List names of students who scored more than the average marks.
13. Find the class name where the oldest student studies.
14. Write a query to insert a new student named "Ali", age 17, male, in class 3.
15. Update the subject of teacher with teacher\_id = 1 to "Computer Science".
16. Delete all students who have age > 25.
17. Get names of students who have not received marks in "English".
18. Display each class name with the total number of male and female students.
19. Get a list of students with total marks across all subjects, ordered from highest to lowest.
20. Create a temp table and store Query #8 in it

**Solutions**:

1. **Get the names of all students**

SELECT name FROM Students;



1. **Get the names of all male students**

SELECT name

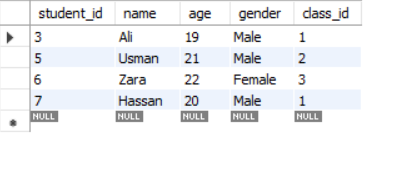
FROM Students

WHERE gender = 'Male';



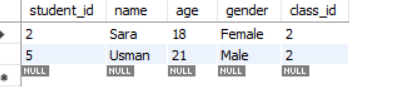
1. **Get details of students older than 18**

SELECT \* FROM Students WHERE age > 18;



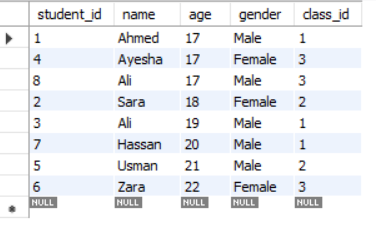
1. **Get details of students who are in class\_id = 2**

SELECT \* FROM Students WHERE class\_id = 2;



1. **List all students ordered by age (youngest first)**

SELECT \* FROM Students ORDER BY age ASC;



1. **Show top 5 students with the highest marks in "Math"**

SELECT s.name, m.marks

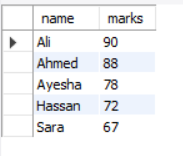
FROM Students s

JOIN Marks m ON s.student\_id = m.student\_id

WHERE m.subject = 'Math'

ORDER BY m.marks DESC

LIMIT 5;

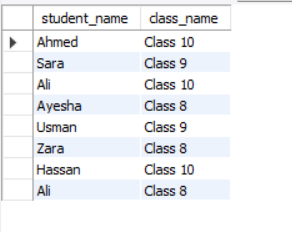


**7. List student names along with their class names**

SELECT s.name AS student\_name, c.class\_name

FROM Students s

JOIN Classes c ON s.class\_id = c.class\_id;



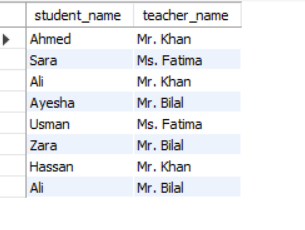
1. **Show student names with their teacher’s name for each class**

SELECT s.name AS student\_name, t.name AS teacher\_name

FROM Students s

JOIN Classes c ON s.class\_id = c.class\_id

JOIN Teachers t ON c.teacher\_id = t.teacher\_id;

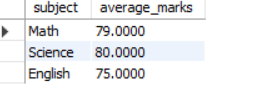


1. **Find the average marks for each subject**

SELECT subject, AVG(marks) AS average\_marks

FROM Marks

GROUP BY subject;

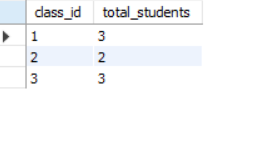


1. **Count how many students are in each class**

SELECT class\_id, COUNT(\*) AS total\_students

FROM Students

GROUP BY class\_id;

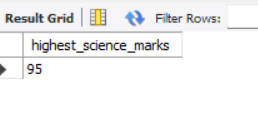


1. **Find the highest marks scored in "Science"**

SELECT MAX(marks) AS highest\_science\_marks

FROM Marks

WHERE subject = 'Science';



1. **List names of students who scored more than the average marks**

SELECT s.name, m.subject, m.marks

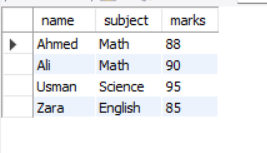
FROM Students s

JOIN Marks m ON s.student\_id = m.student\_id

WHERE m.marks > (

SELECT AVG(marks) FROM Marks

);



1. **Find the class name where the oldest student studies**

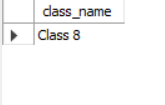
SELECT c.class\_name

FROM Students s

JOIN Classes c ON s.class\_id = c.class\_id

ORDER BY s.age DESC

LIMIT 1;

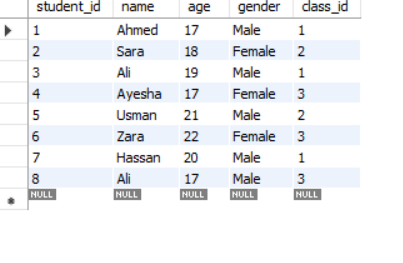


1. **Insert a new student named "Ali", age 17, male, in class 3**

INSERT INTO Students (student\_id, name, age, gender, class\_id)

VALUES (8, 'Ali', 17, 'Male', 3);

SELECT \* FROM Students;



1. **Update the subject of teacher with teacher\_id = 101 to "Computer Science"**

UPDATE Teachers

SET subject = 'Computer Science'

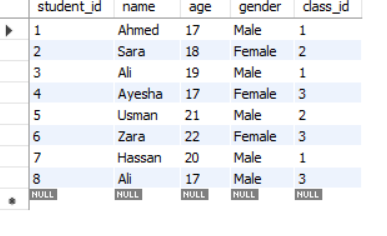
WHERE teacher\_id = 101;

SELECT \* FROM Teachers WHERE teacher\_id = 1;

1. **Delete all students who have age > 25**

DELETE FROM Students

WHERE age > 25;



1. **Get names of students who have not received marks in "English"**

SELECT name

FROM Students

WHERE student\_id NOT IN (

SELECT student\_id FROM Marks WHERE subject = 'English'

);



1. **Display each class name with the total number of male and female students**

SELECT

c.class\_name,

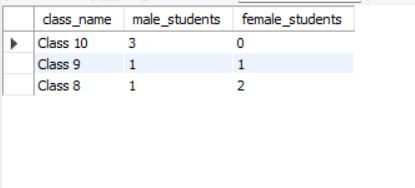
SUM(CASE WHEN s.gender = 'Male' THEN 1 ELSE 0 END) AS male\_students,

SUM(CASE WHEN s.gender = 'Female' THEN 1 ELSE 0 END) AS female\_students

FROM Classes c

JOIN Students s ON c.class\_id = s.class\_id

GROUP BY c.class\_name;



1. **Get a list of students with total marks across all subjects, ordered from highest to lowest**

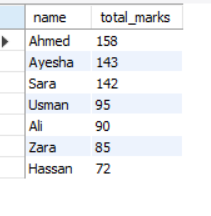
SELECT s.name, SUM(m.marks) AS total\_marks

FROM Students s

JOIN Marks m ON s.student\_id = m.student\_id

GROUP BY s.name

ORDER BY total\_marks DESC;



1. **Create a temporary table to store student names with their class names (Query #8)**

CREATE TEMPORARY TABLE StudentClassInfo AS

SELECT s.name AS student\_name, c.class\_name

FROM Students s

JOIN Classes c ON s.class\_id = c.class\_id;

-- View contents of the temporary table

SELECT \* FROM StudentClassInfo;

