

# Task :

```
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  
);
```

10:42

## **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);
```

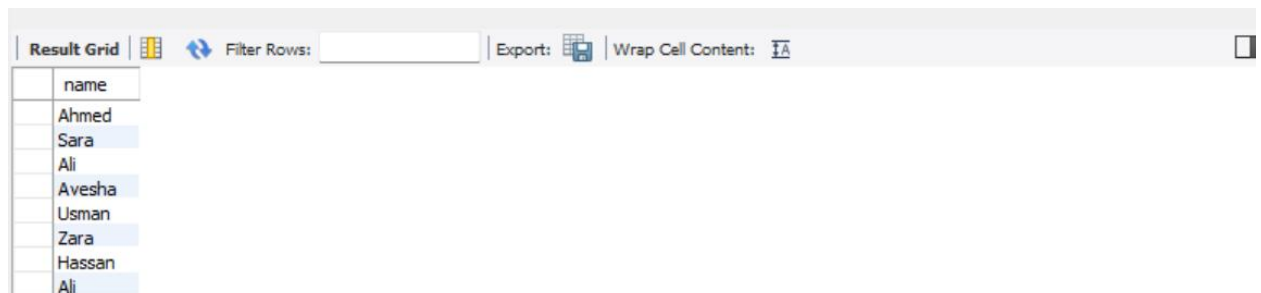
10:42

## QURIES TO IMPLEMENT

1. Write a query to get the names of all students.
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 lower.
20. Create a temp table and store Query #8 in it

## Solution:

- Write a query to get the names of all students.  
SELECT name FROM Students;

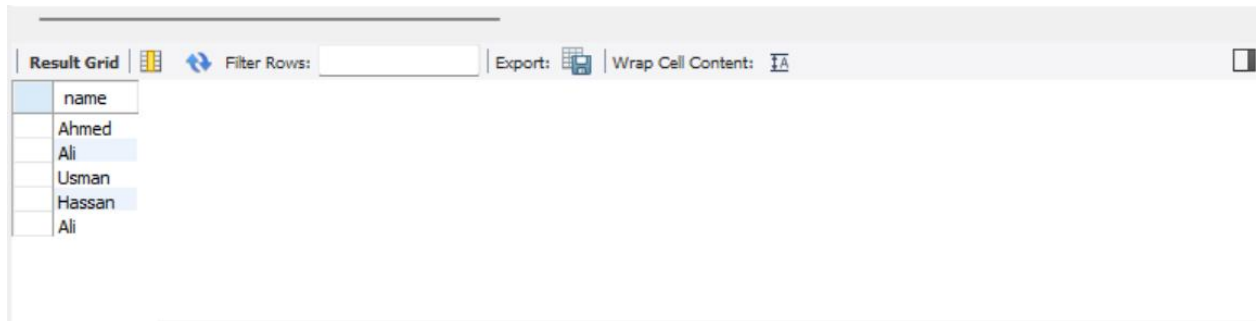


The screenshot shows a database query result grid. The grid has a single column labeled 'name'. The data rows contain the following names: Ahmed, Sara, Ali, Avesha, Usman, Zara, Hassan, and Ali. The grid is displayed in a window with a toolbar at the top containing options like 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'.

| name   |
|--------|
| Ahmed  |
| Sara   |
| Ali    |
| Avesha |
| Usman  |
| Zara   |
| Hassan |
| Ali    |

- Get the names of all male students.

```
SELECT name  
FROM Students  
WHERE gender = 'Male';
```

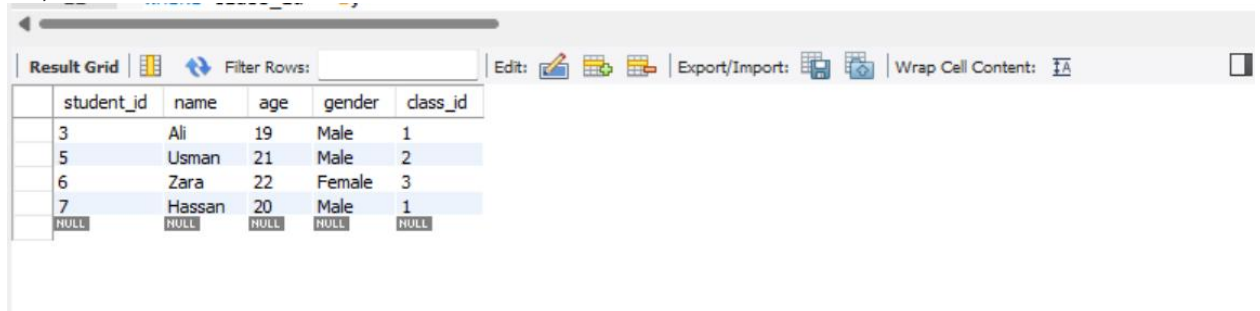


The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of a query that filters for male students. The first column is labeled 'name' and contains the names 'Ahmed', 'Ali', 'Usman', 'Hassan', and 'Ali'. The interface includes a 'Filter Rows' search bar, an 'Export' button, and a 'Wrap Cell Content' option.

| name   |
|--------|
| Ahmed  |
| Ali    |
| Usman  |
| Hassan |
| Ali    |

- Find all students older than 18 years.

```
SELECT *  
FROM Students  
WHERE age >  
18;
```



The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of a query that filters for students older than 18. The columns are 'student\_id', 'name', 'age', 'gender', and 'class\_id'. The results show four students: Ali (19, Male, class\_id 1), Usman (21, Male, class\_id 2), Zara (22, Female, class\_id 3), and Hassan (20, Male, class\_id 1). The interface includes a 'Filter Rows' search bar, an 'Edit' button, an 'Export/Import' button, and a 'Wrap Cell Content' option.

| student_id | name   | age  | gender | class_id |
|------------|--------|------|--------|----------|
| 3          | Ali    | 19   | Male   | 1        |
| 5          | Usman  | 21   | Male   | 2        |
| 6          | Zara   | 22   | Female | 3        |
| 7          | Hassan | 20   | Male   | 1        |
| NULL       | NULL   | NULL | NULL   | NULL     |

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

```
SELECT *  
FROM Students  
WHERE class_id = 2;
```

| Result Grid  |            |       |      |        |          |
|--------------|------------|-------|------|--------|----------|
| Filter Rows: |            |       |      |        |          |
|              | student_id | name  | age  | gender | class_id |
|              | 2          | Sara  | 18   | Female | 2        |
|              | 5          | Usman | 21   | Male   | 2        |
|              | NULL       | NULL  | NULL | NULL   | NULL     |

- List all students ordered by age, youngest first.

```
SELECT *
```

```
FROM Students
```

```
ORDER BY age
```

```
ASC;
```

| Result Grid  |            |        |     |        |          |
|--------------|------------|--------|-----|--------|----------|
| Filter Rows: |            |        |     |        |          |
|              | student_id | name   | age | gender | class_id |
|              | 1          | Ahmed  | 17  | Male   | 1        |
|              | 4          | Avesha | 17  | Female | 3        |
|              | 8          | Ali    | 17  | Male   | 3        |
|              | 2          | Sara   | 18  | Female | 2        |
|              | 3          | Ali    | 19  | Male   | 1        |
|              | 7          | Hassan | 20  | Male   | 1        |
|              | 5          | Usman  | 21  | Male   | 2        |
|              | 6          | Zara   | 22  | Female | 3        |

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

```
SELECT s.name AS student_name, m.marks
```

```
FROM Students s
```

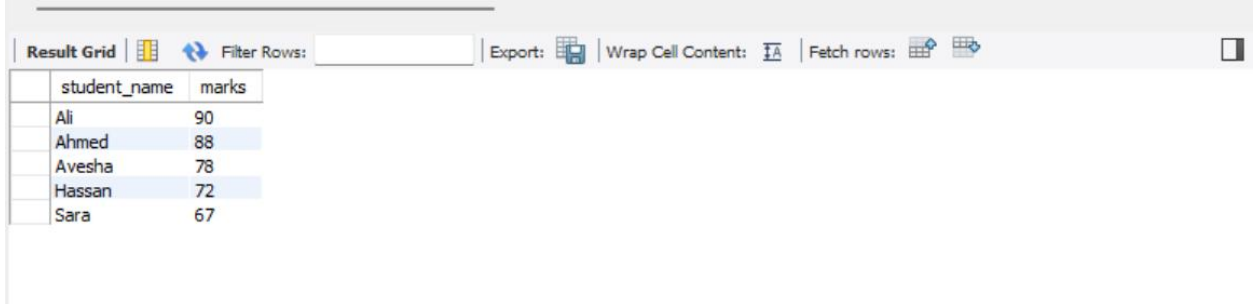
```
JOIN Marks m ON s.student_id = m.student_id
```

```
WHERE m.subject = 'Math'
```

```
ORDER BY m.marks DESC
```

## LIMIT

5;



The screenshot shows a database query result grid with the following data:

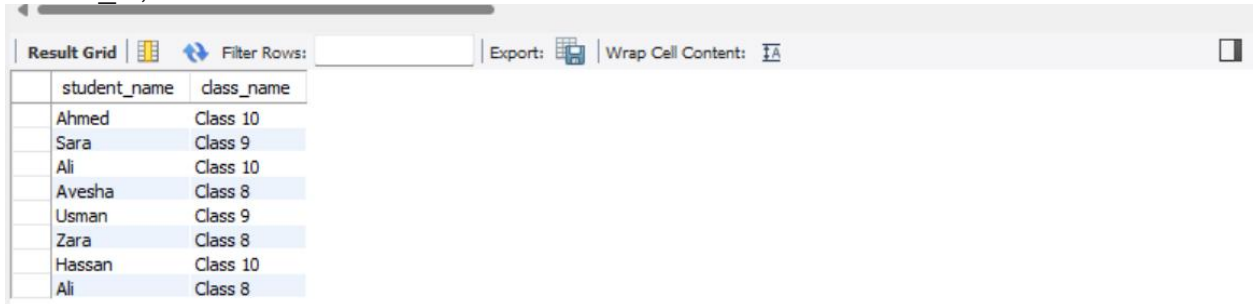
| student_name | marks |
|--------------|-------|
| Ali          | 90    |
| Ahmed        | 88    |
| Avesha       | 78    |
| Hassan       | 72    |
| Sara         | 67    |

- List student names along with their class names.

```
SELECT s.name AS student_name, c.class_name
```

```
FROM Students s
```

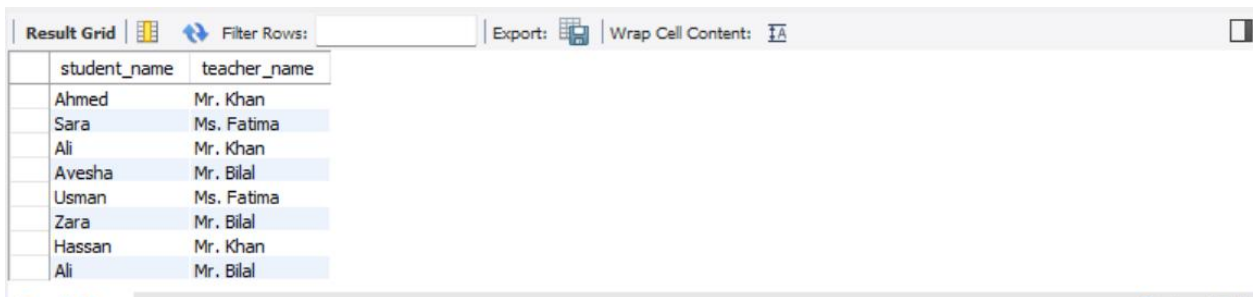
```
JOIN Classes c ON s.class_id =  
c.class_id;
```



The screenshot shows a database query result grid with the following data:

| student_name | class_name |
|--------------|------------|
| Ahmed        | Class 10   |
| Sara         | Class 9    |
| Ali          | Class 10   |
| Avesha       | Class 8    |
| Usman        | Class 9    |
| Zara         | Class 8    |
| Hassan       | Class 10   |
| Ali          | Class 8    |

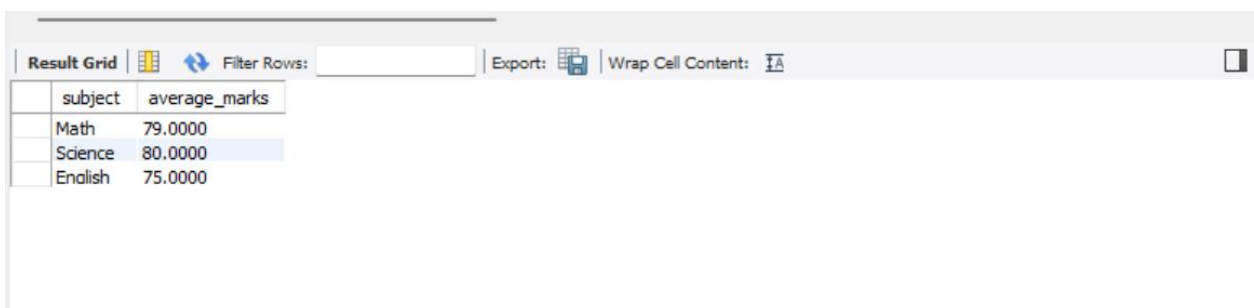
- Show student names with their teacher's name for each class



The screenshot shows a database query result grid with the following data:

| student_name | teacher_name |
|--------------|--------------|
| Ahmed        | Mr. Khan     |
| Sara         | Ms. Fatima   |
| Ali          | Mr. Khan     |
| Avesha       | Mr. Bilal    |
| Usman        | Ms. Fatima   |
| Zara         | Mr. Bilal    |
| Hassan       | Mr. Khan     |
| Ali          | Mr. Bilal    |

- Find the average marks for each subject.



The screenshot shows a database query result grid with the following data:

| subject | average_marks |
|---------|---------------|
| Math    | 79.0000       |
| Science | 80.0000       |
| English | 75.0000       |

- Count how many students are in each class.

| Result Grid |               | Filter Rows: | Export: | Wrap Cell Content: |
|-------------|---------------|--------------|---------|--------------------|
| class_name  | student_count |              |         |                    |
| Class 10    | 3             |              |         |                    |
| Class 9     | 2             |              |         |                    |
| Class 8     | 3             |              |         |                    |

- Find the highest marks scored in "Science".

| Result Grid           |  | Filter Rows: | Export: | Wrap Cell Content: |
|-----------------------|--|--------------|---------|--------------------|
| highest_science_marks |  |              |         |                    |
| 95                    |  |              |         |                    |

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

| Result Grid  |         | Filter Rows: | Export: | Wrap Cell Content: |
|--------------|---------|--------------|---------|--------------------|
| student_name | subject | marks        |         |                    |
| Ahmed        | Math    | 88           |         |                    |
| Ali          | Math    | 90           |         |                    |
| Usman        | Science | 95           |         |                    |
| Zara         | English | 85           |         |                    |

- Find the class name where the oldest student studies.

| Result Grid |  | Filter Rows: | Export: | Wrap Cell Content: |
|-------------|--|--------------|---------|--------------------|
| class_name  |  |              |         |                    |
| Class 8     |  |              |         |                    |

- Write a query to insert a new student named "Ali", age 17, male, in class 3.

```
INSERT INTO Students (student_id, name, age, gender, class_id)
```

```
VALUES (9, 'Ali', 17, 'Male', 3);
```

```
SELECT * FROM Students
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

|  | student_id | name   | age  | gender | class_id |
|--|------------|--------|------|--------|----------|
|  | 4          | Avesha | 17   | Female | 3        |
|  | 6          | Zara   | 22   | Female | 3        |
|  | 8          | Ali    | 17   | Male   | 3        |
|  | 9          | Ali    | 17   | Male   | 3        |
|  | NULL       | NULL   | NULL | NULL   | NULL     |

Students 79

Apply

WHERE class\_id = 3;

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

UPDATE Teachers

SET subject = 'Computer Science'

WHERE teacher\_id = 101;

-- Check if teacher\_id = 1 exists

SELECT \* FROM Teachers WHERE teacher\_id = 101;

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

| teacher_id | name     | subject          |
|------------|----------|------------------|
| 101        | Mr. Khan | Computer Science |
| NULL       | NULL     | NULL             |

- Delete all students who have age > 25.

DELETE FROM Students

WHERE student\_id IN (

SELECT student\_id FROM (

SELECT student\_id FROM Students WHERE age > 25

) AS temp

);

121 12:56:59 DELETE FROM Students WHERE student\_id IN ( SELECT student\_id FROM ( ... 0 row(s) affected

- Get names of students who have not received marks in "English". SELECT s.name

FROM Students s

WHERE NOT EXISTS (

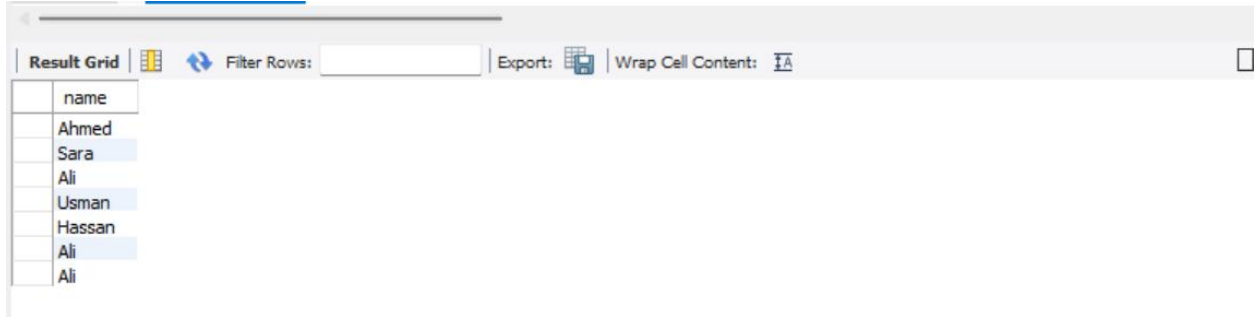
SELECT 1

FROM Marks m

WHERE m.student\_id = s.student\_id

AND m.subject = 'English'

);



The screenshot shows a database query result grid. The grid has a single column labeled 'name'. The rows contain the following names: Ahmed, Sara, Ali, Usman, Hassan, Ali, and Ali. The grid is displayed in a window with a toolbar at the top containing icons for 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'.

| name   |
|--------|
| Ahmed  |
| Sara   |
| Ali    |
| Usman  |
| Hassan |
| Ali    |
| Ali    |

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

SELECT

c.class\_name,

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

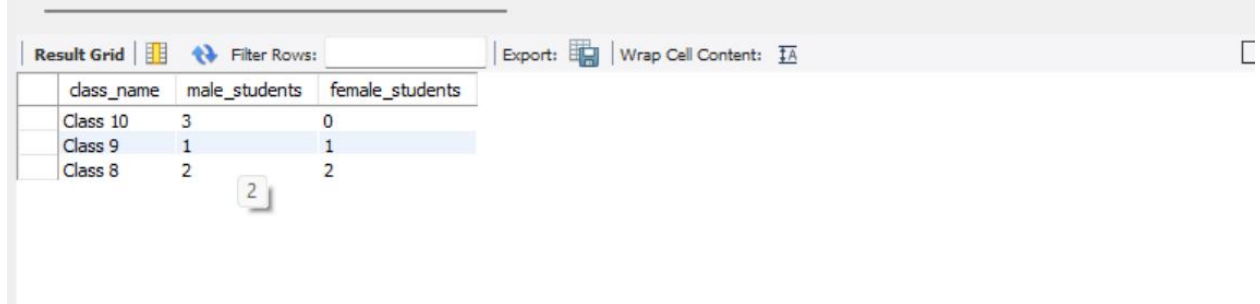
COUNT(CASE WHEN s.gender = 'Female' THEN 1 END) AS female\_students

FROM Students s

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



GROUP BY  
c.class\_name;



The screenshot shows a 'Result Grid' window with a table containing student counts by class. The table has columns: class\_name, male\_students, and female\_students. The rows are Class 10 (3 male, 0 female), Class 9 (1 male, 1 female), and Class 8 (2 male, 2 female). A small tooltip with the number '2' is visible over the male\_students cell for Class 8.

| class_name | male_students | female_students |
|------------|---------------|-----------------|
| Class 10   | 3             | 0               |
| Class 9    | 1             | 1               |
| Class 8    | 2             | 2               |

- Get a list of students with total marks across all subjects, ordered from highest to lower.

```
SELECT s.name AS student_name, SUM(m.marks) AS total_marks
```

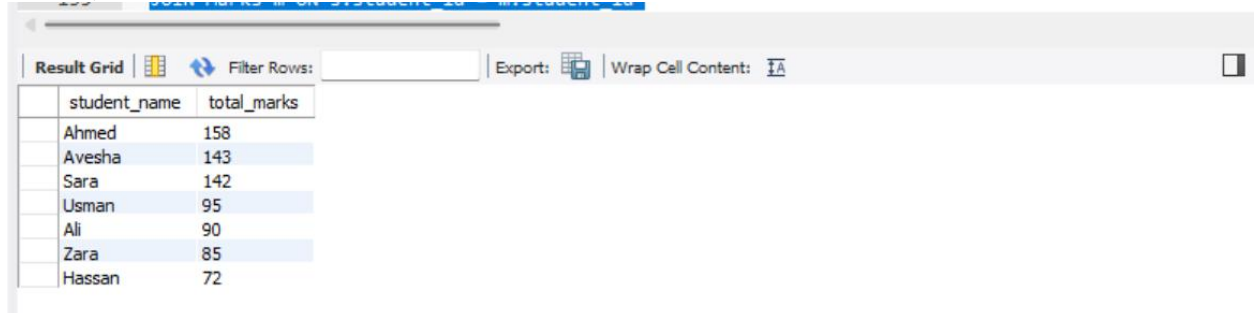
```
FROM Students s
```

```
JOIN Marks m ON s.student_id = m.student_id
```

```
GROUP BY s.student_id, s.name
```

```
ORDER BY total_marks
```

```
DESC;
```



The screenshot shows a 'Result Grid' window with a table containing student names and their total marks. The table has columns: student\_name and total\_marks. The rows are Ahmed (158), Avesha (143), Sara (142), Usman (95), Ali (90), Zara (85), and Hassan (72).

| student_name | total_marks |
|--------------|-------------|
| Ahmed        | 158         |
| Avesha       | 143         |
| Sara         | 142         |
| Usman        | 95          |
| Ali          | 90          |
| Zara         | 85          |
| Hassan       | 72          |

- Create a temp table and store Query #8 in it

```
CREATE TEMPORARY TABLE Temp_TotalMarks AS
```

```
SELECT s.student_id, s.name AS student_name, SUM(m.marks) AS total_marks
```

```
FROM Students s
```

```
JOIN Marks m ON s.student_id = m.student_id
```

```
GROUP BY s.student_id, s.name
```

```
ORDER BY total_marks DESC;
```

```
SELECT * FROM  
Temp_TotalMarks;
```

| Result Grid |              |             | Filter Rows: | Export: | Wrap Cell Content: |
|-------------|--------------|-------------|--------------|---------|--------------------|
|             | student_name | total_marks |              |         |                    |
|             | Ahmed        | 158         |              |         |                    |
|             | Avesha       | 143         |              |         |                    |
|             | Sara         | 142         |              |         |                    |
|             | Usman        | 95          |              |         |                    |
|             | Ali          | 90          |              |         |                    |
|             | Zara         | 85          |              |         |                    |
|             | Hassan       | 72          |              |         |                    |