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);
```

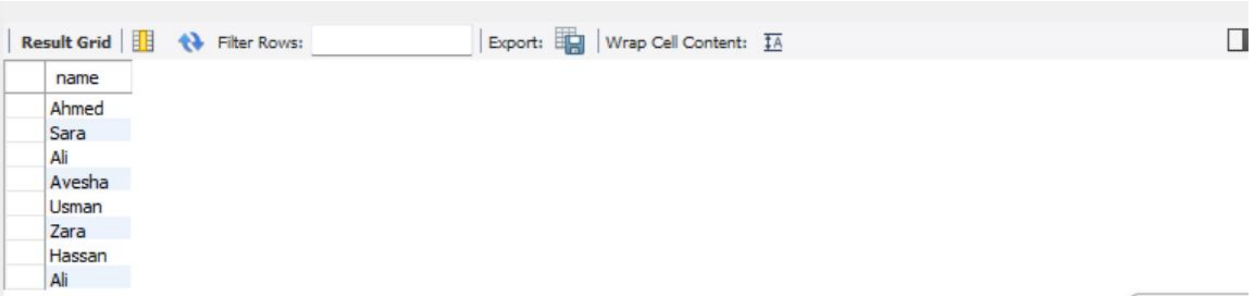
10:42

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:

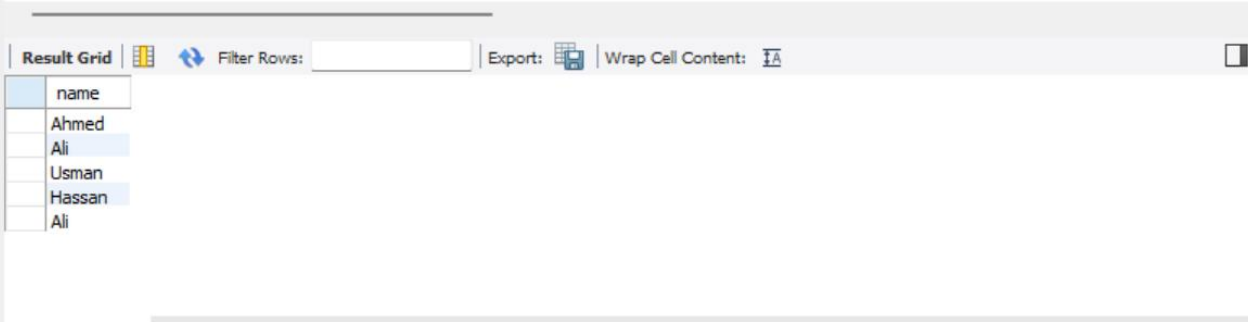
➢ Write a query to get the names of all students.
SELECT name FROM Students;



name
Ahmed
Sara
Ali
Ayesha
Usman
Zara
Hassan
Ali

- Get the names of all male students.

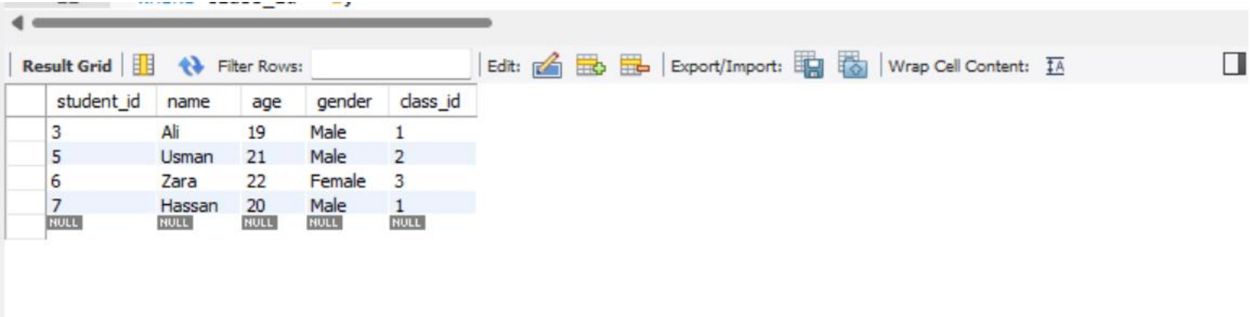
```
SELECT name
FROM Students
WHERE gender = 'Male';
```



name
Ahmed
Ali
Usman
Hassan
Ali

- Find all students older than 18 years.

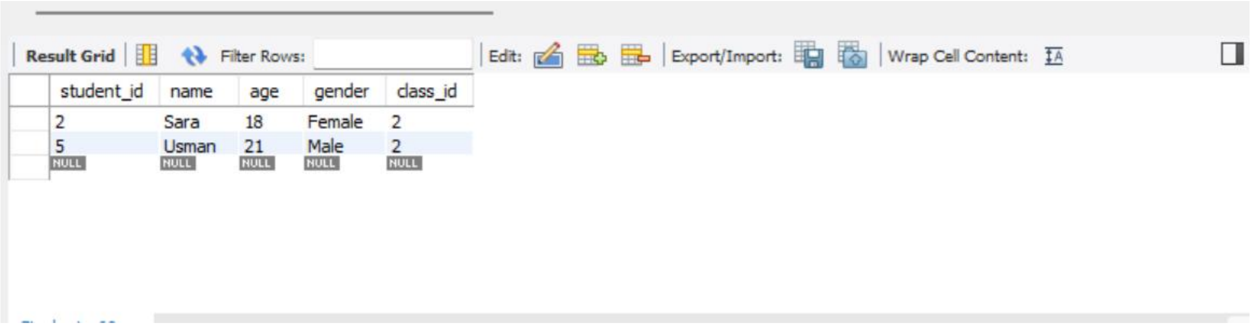
```
SELECT *

FROM Students

WHERE age >
18;
```



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.

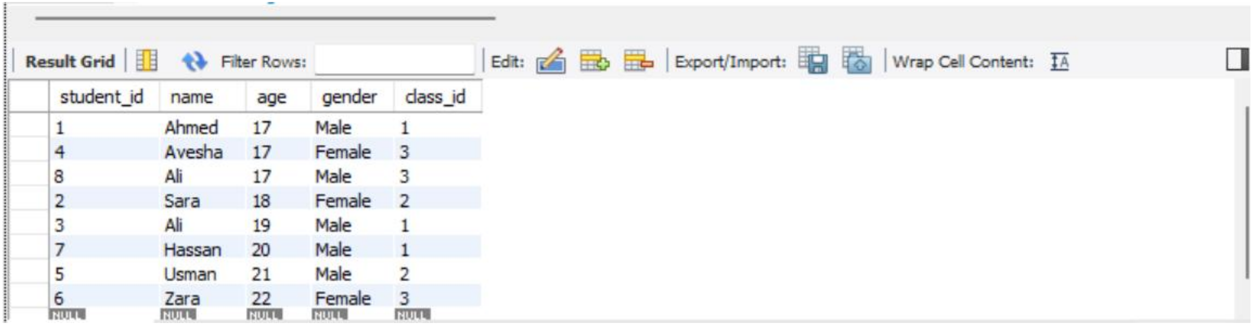
```
SELECT *

FROM Students

WHERE class_id = 2;
```



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;
```

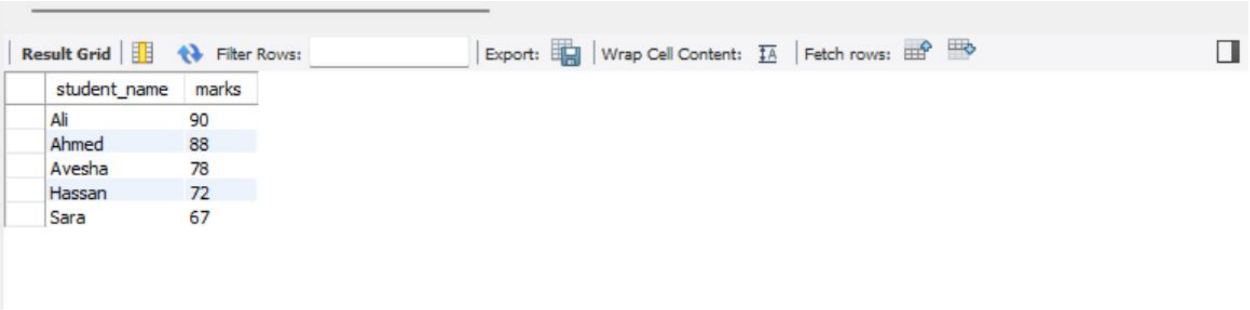


student_id	name	age	gender	class_id
1	Ahmed	17	Male	1
4	Avesha	17	Female	3
8	Ali	17	Male	3
2	Sara	18	Female	2
3	Ali	19	Male	1
7	Hassan	20	Male	1
5	Usman	21	Male	2
6	Zara	22	Female	3
NULL	NULL	NULL	NULL	NULL

- ➢ 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
```

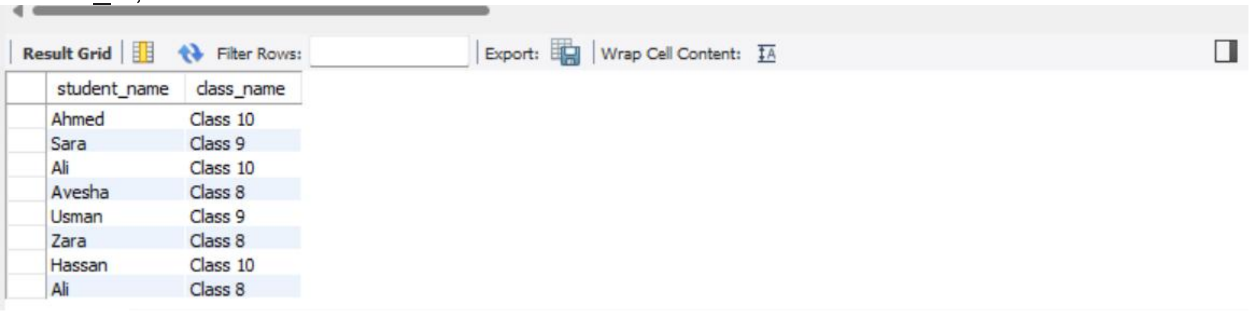
```
LIMIT
5;
```



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

➢ List student names along with their class names.

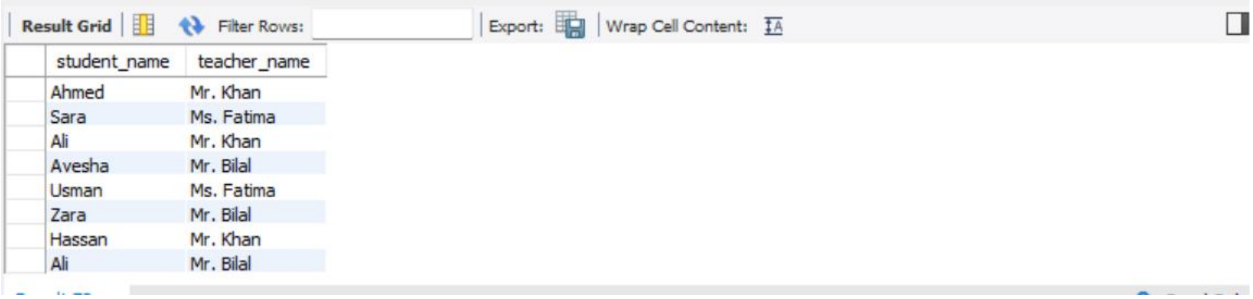
```
SELECT s.name AS student_name, c.class_name

FROM Students s

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



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

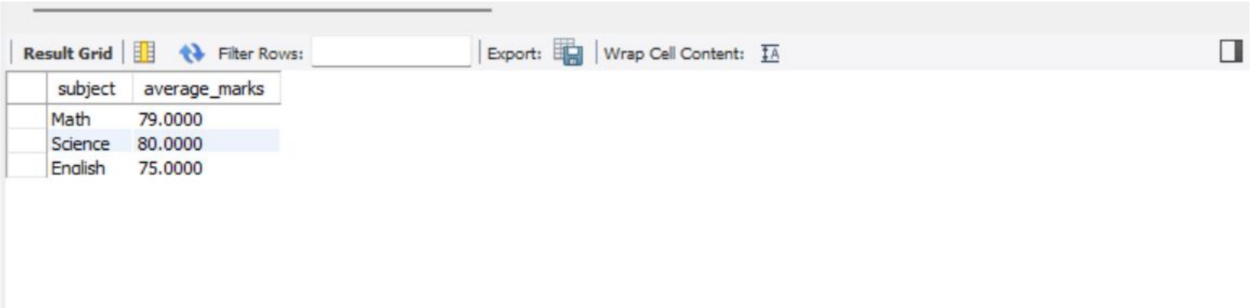
➢ Show student names with their teacher’s name for each class

.



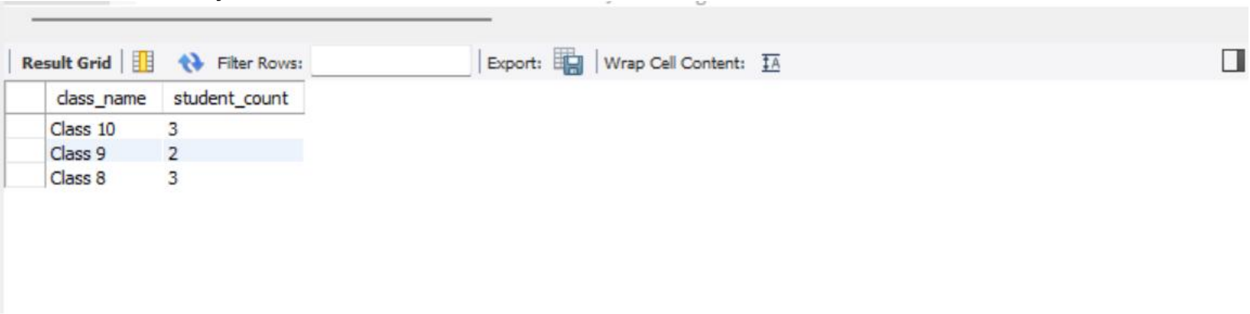
student_name	teacher_name
Ahmed	Mr. Khan
Sara	Ms. Fatima
Ali	Mr. Khan
Avesha	Mr. Bilal
Usman	Ms. Fatima
Zara	Mr. Bilal
Hassan	Mr. Khan
Ali	Mr. Bilal

➢ Find the average marks for each subject.



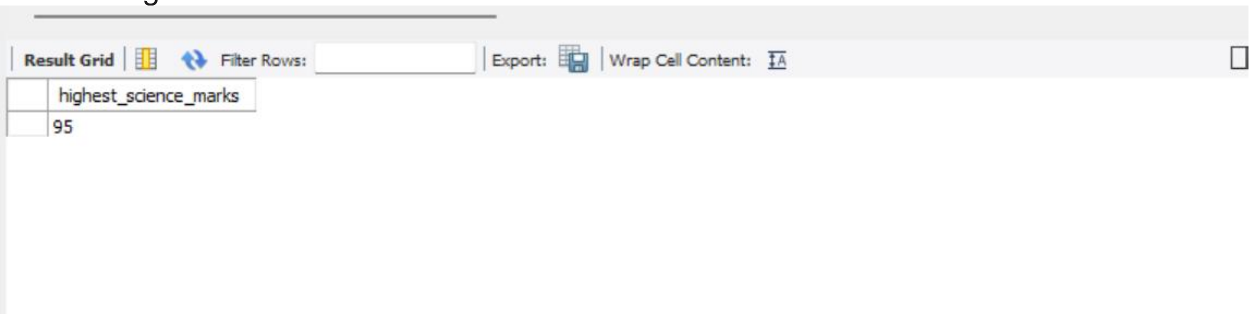
subject	average_marks
Math	79.0000
Science	80.0000
English	75.0000

- Count how many students are in each class.



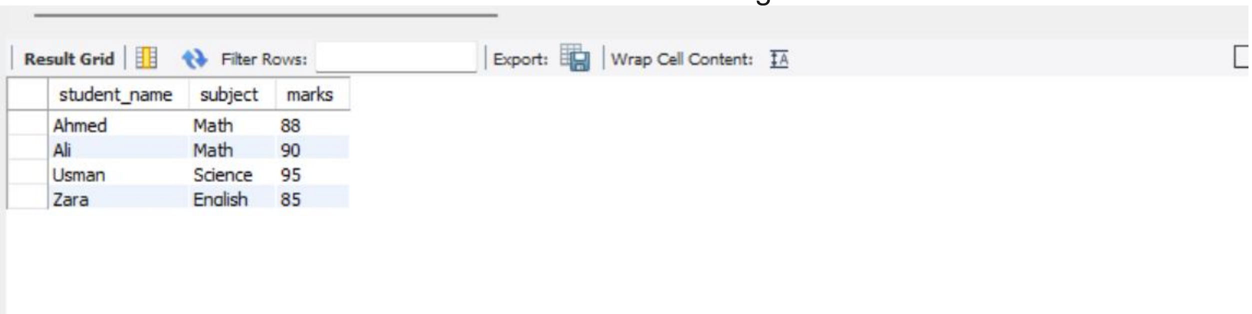
class_name	student_count
Class 10	3
Class 9	2
Class 8	3

- Find the highest marks scored in "Science".



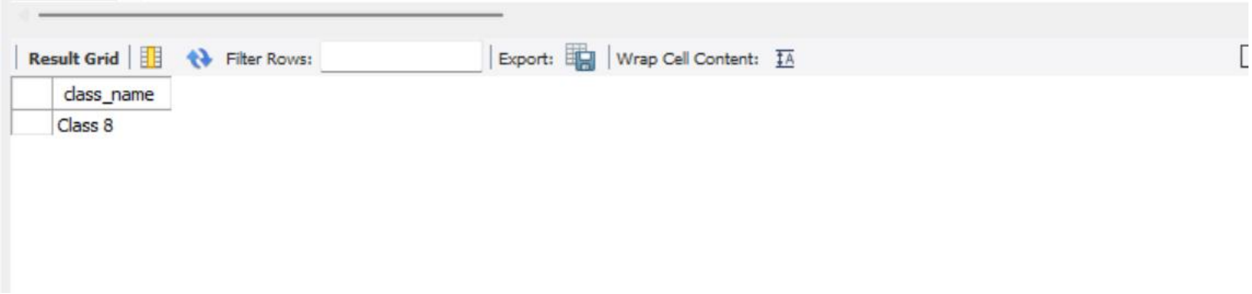
highest_science_marks
95

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



student_name	subject	marks
Ahmed	Math	88
Ali	Math	90
Usman	Science	95
Zara	English	85

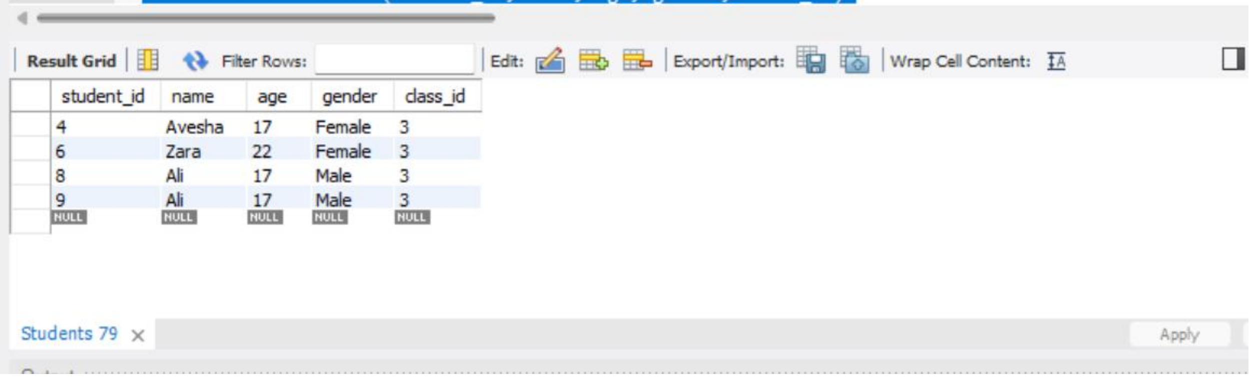
- Find the class name where the oldest student studies.



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
```

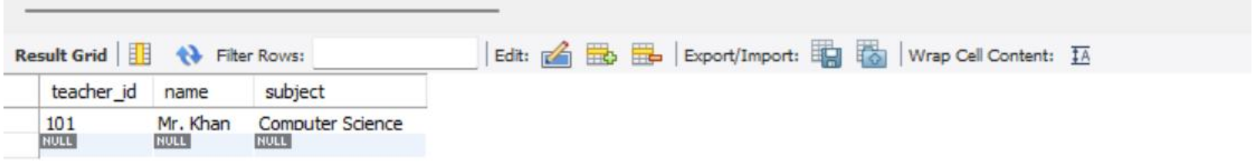


student_id	name	age	gender	class_id
4	Avesha	17	Female	3
6	Zara	22	Female	3
8	Ali	17	Male	3
9	Ali	17	Male	3
NULL	NULL	NULL	NULL	NULL

WHERE class_id = 3;

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

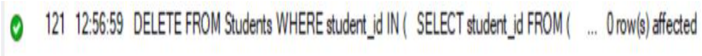
```
UPDATE Teachers

SET subject = 'Computer Science'

WHERE teacher_id = 101;



-- Check if teacher_id = 1 exists

SELECT * FROM Teachers WHERE teacher_id =
101;
```



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
```

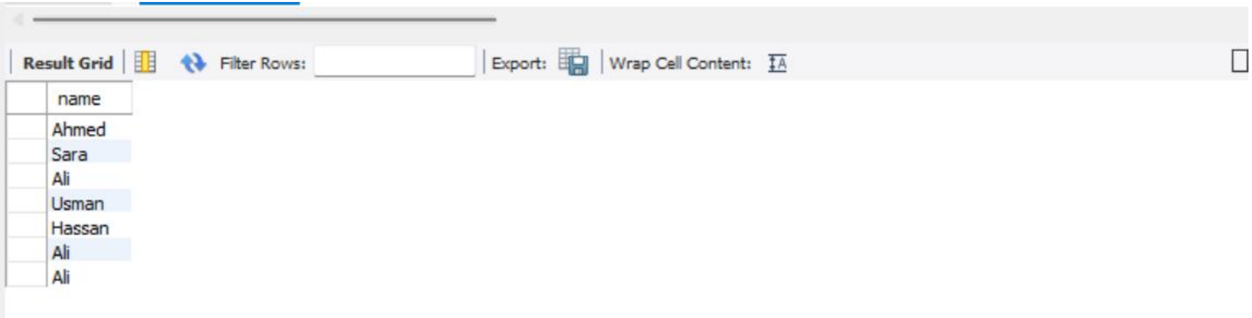


```
121 12:56:59 DELETE FROM Students WHERE student_id IN (  SELECT student_id FROM (  ... 0 row(s) affected
```

```
);
```

➢ 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'

);
```

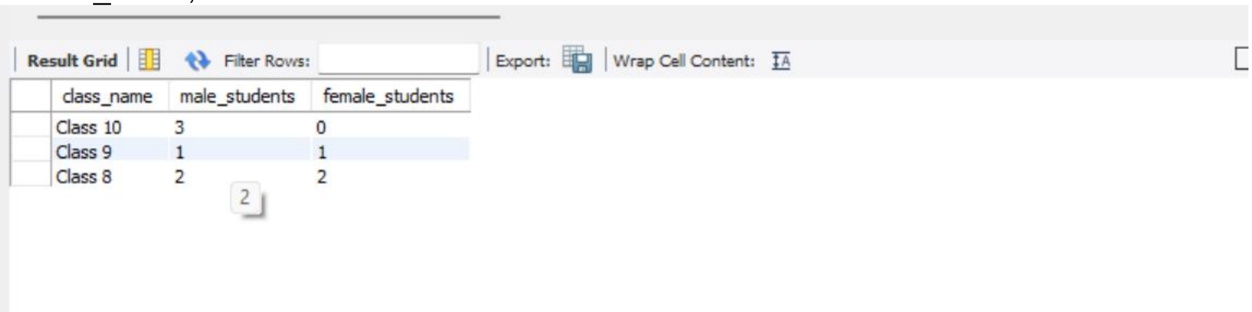


name
Ahmed
Sara
Ali
Usman
Hassan
Ali
Ali

➢ 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
```

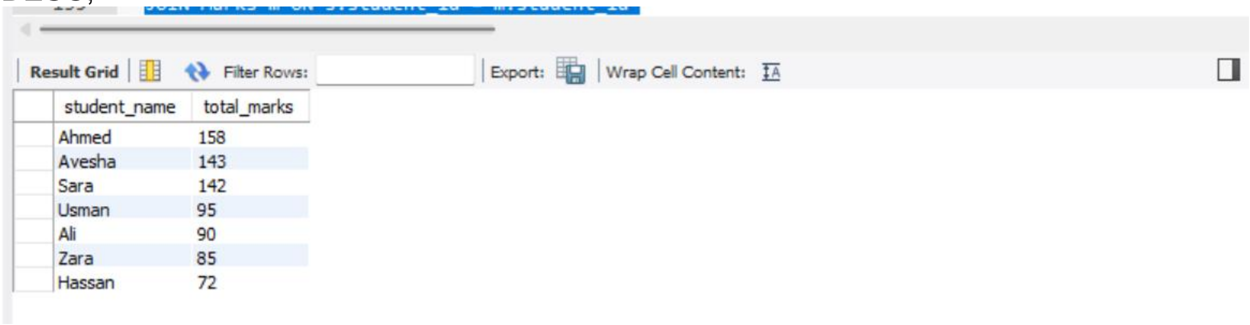
```
GROUP BY
c.class_name;
```



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;
```

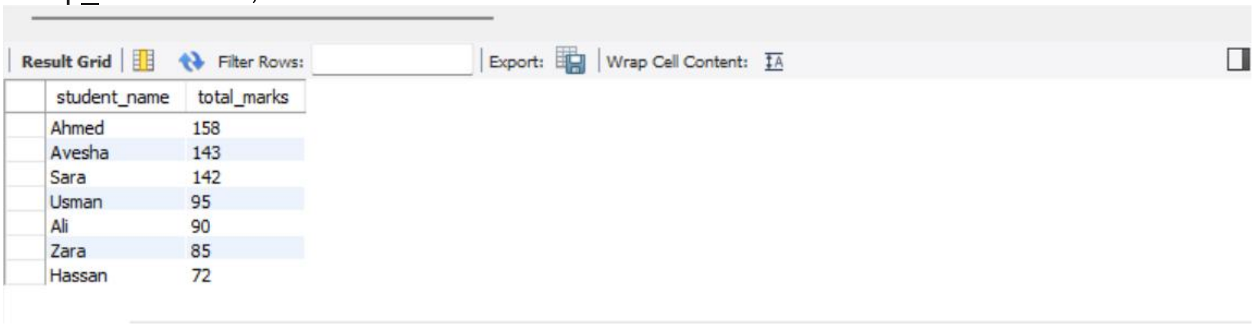


student_name	total_marks
Ahmed	158
Avesha	143
Sara	142
Usman	95
Ali	90
Zara	85
Hassan	72

- 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;
```



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