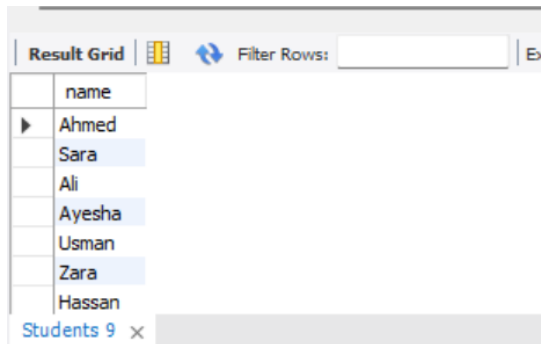


DATABASE QUERIES TO SOLVE

1. Write a query to get the names of all students.

```
SELECT name FROM Students;
```



The screenshot shows a database interface with a 'Result Grid' tab. It displays a table with one column named 'name'. The table contains nine rows of student names: Ahmed, Sara, Ali, Ayesha, Usman, Zara, and Hassan. The first row is highlighted with a blue background. At the bottom of the grid, it says 'Students 9' followed by a close button (x).

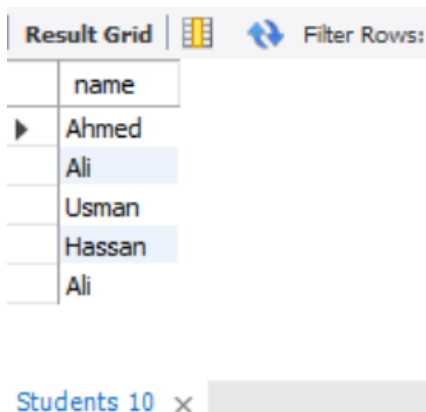
name
Ahmed
Sara
Ali
Ayesha
Usman
Zara
Hassan

Students 9 x

2. 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. It displays a table with one column named 'name'. The table contains five rows of male student names: Ahmed, Ali, Usman, Hassan, and Ali. The first row is highlighted with a blue background. At the bottom of the grid, it says 'Students 10' followed by a close button (x).

name
Ahmed
Ali
Usman
Hassan
Ali

Students 10 x

3. Find all students older than 18 years.

```
SELECT * FROM Students
```

```
WHERE age > 18;
```

DATABASE QUERIES TO SOLVE

Result Grid | Filter Rows: | Edit:

	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

Students 11 x

4. Get details of students who are in class_id = 2.

```
SELECT * FROM Students
```

```
WHERE class_id = 2;
```

Result Grid | Filter Rows: | Edit:

	student_id	name	age	gender	class_id
▶	2	Sara	18	Female	2
	5	Usman	21	Male	2
*	NULL	NULL	NULL	NULL	NULL

Students 12 x

5. List all students ordered by age, youngest first.

```
SELECT * FROM Students
```

```
ORDER BY age ASC;
```

Result Grid | Filter Rows: | Edit:

	student_id	name	age	gender	class_id
▶	1	Ahmed	17	Male	1
	4	Ayesha	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. Show top 5 students with the highest marks in "Math".

```
SELECT student_id, marks
```

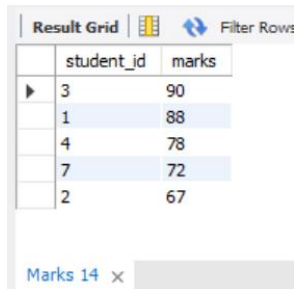
```
FROM Marks
```

DATABASE QUERIES TO SOLVE

WHERE subject = 'Math'

ORDER BY marks DESC

LIMIT 5;



The screenshot shows a database interface with a 'Result Grid' tab. It displays a table with two columns: 'student_id' and 'marks'. The data is sorted by marks in descending order, and only the top 5 results are shown. Below the table, there is a status bar that says 'Marks 14' with a close button.

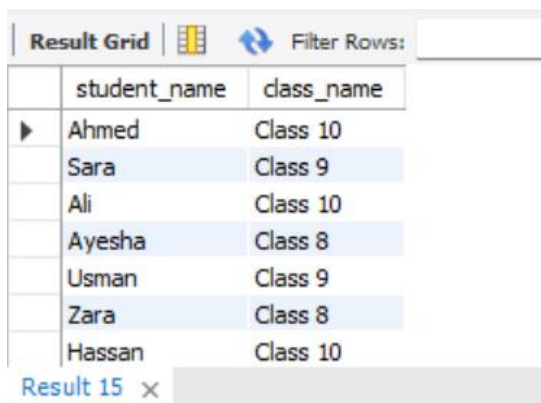
student_id	marks
3	90
1	88
4	78
7	72
2	67

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;



The screenshot shows a database interface with a 'Result Grid' tab. It displays a table with two columns: 'student_name' and 'class_name'. The data is sorted by student name in ascending order. Below the table, there is a status bar that says 'Result 15' with a close button.

student_name	class_name
Ahmed	Class 10
Sara	Class 9
Ali	Class 10
Ayesha	Class 8
Usman	Class 9
Zara	Class 8
Hassan	Class 10

8. 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;

DATABASE QUERIES TO SOLVE

Result Grid	Filter Rows:
student_name	teacher_name
Ahmed	Mr. Khan
Sara	Ms. Fatima
Ali	Mr. Khan
Ayesha	Mr. Bilal
Usman	Ms. Fatima
Zara	Mr. Bilal
Hassan	Mr. Khan

9. Find the average marks for each subject.

```
SELECT subject, AVG(marks) AS average_marks
FROM Marks
GROUP BY subject;
```

Result Grid	Filter Rows:
subject	average_marks
Math	79.0000
Science	80.0000
English	75.0000

10. Count how many students are in each class.

```
SELECT class_id, COUNT(*) AS total_students
FROM Students
GROUP BY class_id;
```

Result Grid	Filter Rows:
class_id	total_students
1	3
2	2
3	3

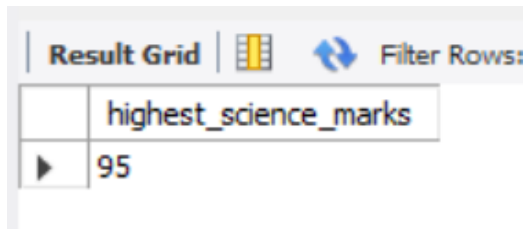
11. Find the highest marks scored in "Science".

```
SELECT MAX(marks) AS highest_science_marks
```

DATABASE QUERIES TO SOLVE

FROM Marks

WHERE subject = 'Science';



The screenshot shows a database interface with a 'Result Grid' tab. The grid has one column labeled 'highest_science_marks' and one row with the value '95'. There are icons for 'Filter Rows' and a 'Filter' button.

highest_science_marks
95

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

SELECT S.name, M.marks

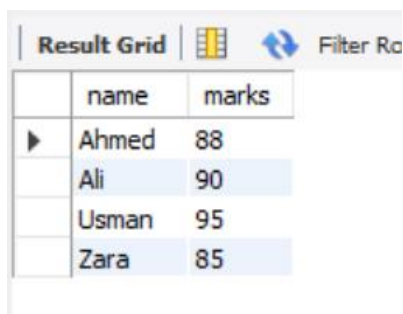
FROM Marks M

JOIN Students S ON M.student_id = S.student_id

WHERE M.marks > (

SELECT AVG(marks) FROM Marks

);



The screenshot shows a database interface with a 'Result Grid' tab. The grid has two columns: 'name' and 'marks'. There are four rows of data: Ahmed (88), Ali (90), Usman (95), and Zara (85). There are icons for 'Filter Rows' and a 'Filter' button.

name	marks
Ahmed	88
Ali	90
Usman	95
Zara	85

13. Find the class name where the oldest student studies.

SELECT C.class_name

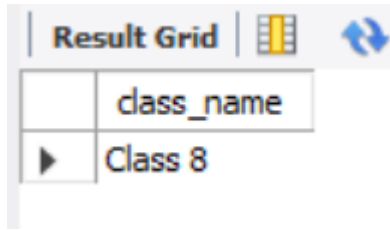
FROM Students S

JOIN Classes C ON S.class_id = C.class_id

ORDER BY S.age DESC

LIMIT 1;

DATABASE QUERIES TO SOLVE



	class_name
▶	Class 8

14. 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 (8, 'Ali', 17, 'Male', 3);
```

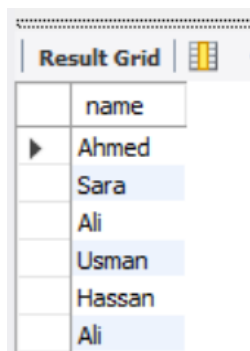
15. Update the subject of teacher with teacher_id = 1 to "Computer Science".

```
UPDATE Teachers
SET subject = 'Computer Science'
WHERE teacher_id = 1;
```

16. Delete all students who have age > 25.

17. 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'
);
```

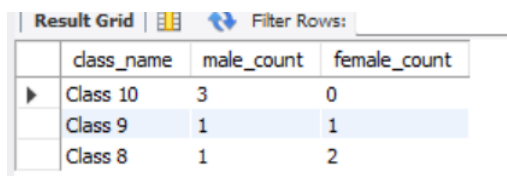


	name
▶	Ahmed
	Sara
	Ali
	Usman
	Hassan
	Ali

DATABASE QUERIES TO SOLVE

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

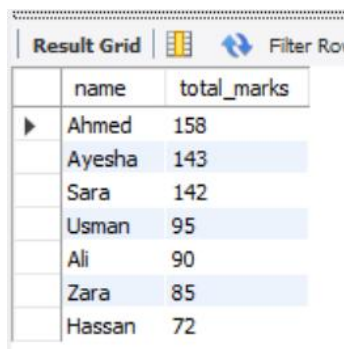
```
SELECT C.class_name,  
       SUM(CASE WHEN S.gender = 'Male' THEN 1 ELSE 0 END) AS male_count,  
       SUM(CASE WHEN S.gender = 'Female' THEN 1 ELSE 0 END) AS female_count  
FROM Students S  
JOIN Classes C ON S.class_id = C.class_id  
GROUP BY C.class_name;
```



	class_name	male_count	female_count
▶	Class 10	3	0
	Class 9	1	1
	Class 8	1	2

19. 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;
```



	name	total_marks
▶	Ahmed	158
	Ayesha	143
	Sara	142
	Usman	95
	Ali	90
	Zara	85
	Hassan	72

20. Create a temp table and store Query #8 in it

```
CREATE TEMPORARY TABLE StudentTeacherInfo AS  
SELECT S.name AS student_name, T.name AS teacher_name
```

DATABASE QUERIES TO SOLVE

FROM Students S

JOIN Classes C ON S.class_id = C.class_id

JOIN Teachers T ON C.teacher_id = T.teacher_id;