

SQL Query Outputs – SchoolDB Database

This document contains the **output screenshots** for SQL queries executed on the `SchoolDB` database in MySQL Workbench. The database includes four tables — `Students`, `Classes`, `Teachers`, and `Marks` — populated with sample data.

-- Create a new database named `Student_db`

```
CREATE DATABASE Student_db;
```

-- Select the database to use for the following operations

```
USE Student_db;
```

-- =====

-- CREATE TABLES

-- =====

-- Create `Students` table: stores basic info about students

```
CREATE TABLE Students (  
    student_id INT PRIMARY KEY,  
    name VARCHAR(50),  
    age INT,  
    gender VARCHAR(10),  
    class_id INT  
);
```

-- Create `Classes` table: stores each class info and the teacher assigned

```
CREATE TABLE Classes (  
    class_id INT PRIMARY KEY,  
    class_name VARCHAR(50),
```

```
teacher_id INT  
);
```

-- Create Teachers table: stores teacher details and subjects taught

```
CREATE TABLE Teachers (  
    teacher_id INT PRIMARY KEY,  
    name VARCHAR(50),  
    subject VARCHAR(50)  
);
```

-- Create Marks table: stores marks for each student in various subjects

```
CREATE TABLE Marks (  
    mark_id INT PRIMARY KEY,  
    student_id INT,  
    subject VARCHAR(50),  
    marks INT  
);
```

```
-- =====
```

-- INSERT SAMPLE DATA

```
-- =====
```

-- Insert sample student data

```
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);
```

```
-- Insert class records
```

```
INSERT INTO Classes (class_id, class_name, teacher_id) VALUES  
(1, 'Class 10', 101),  
(2, 'Class 9', 102),  
(3, 'Class 8', 103);
```

```
-- Insert teacher records
```

```
INSERT INTO Teachers (teacher_id, name, subject) VALUES  
(101, 'Mr. Khan', 'Math'),  
(102, 'Ms. Fatima', 'Science'),  
(103, 'Mr. Bilal', 'English');
```

```
-- Insert marks for students
```

```
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);
```

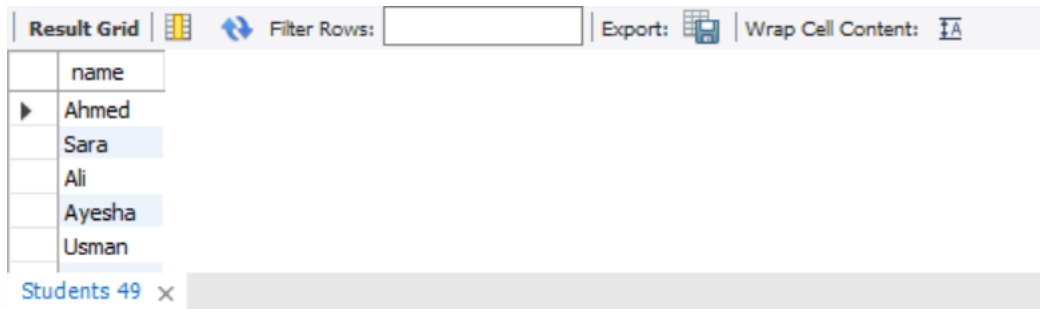
```
-- =====
```

-- QUERIES

-- =====

-- 1. Get the names of all students

SELECT name FROM Students;



The screenshot shows a database interface with a toolbar at the top containing options like 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. Below the toolbar is a table with a single column labeled 'name'. The table contains six rows with the following names: Ahmed, Sara, Ali, Ayesha, and Usman. The last row is highlighted in blue. At the bottom of the table, there is a status bar that reads 'Students 49' followed by a close button 'x'.

name
Ahmed
Sara
Ali
Ayesha
Usman

Students 49 x

-- 2. Get the names of all male students

SELECT name FROM Students WHERE gender = 'Male';

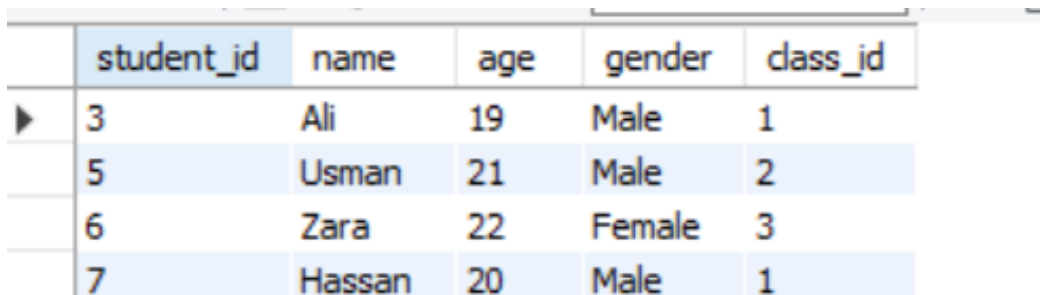


The screenshot shows a database interface with a table containing a single column labeled 'name'. The table has six rows with the following names: Ahmed, Ali, Usman, Hassan, and Ali. The second and fourth rows are highlighted in blue.

name
Ahmed
Ali
Usman
Hassan
Ali

-- 3. Get details of students older than 18

SELECT * FROM Students WHERE age > 18;



The screenshot shows a database interface with a table containing six columns: student_id, name, age, gender, and class_id. The table has four rows of data. The first and third rows are highlighted in blue.

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

-- 4. Get details of students enrolled 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

-- 5. 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	Ayesha	17	Female	3
8	Ali	17	Male	3
2	Sara	18	Female	2
3	Ali	19	Male	1

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

SELECT student_id, marks

FROM Marks

WHERE subject = 'Math'

ORDER BY marks DESC

LIMIT 5;

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

-- 7. Show student names with their respective class names

SELECT Students.name, Classes.class_name

FROM Students

JOIN Classes ON Students.class_id = Classes.class_id;

name	class_name
Ahmed	Class 10
Sara	Class 9
Ali	Class 10
Ayesha	Class 8
Usman	Class 9

-- 8. Show student names with their class teacher's name

```
SELECT Students.name AS student_name, Teachers.name AS teacher_name
```

```
FROM Students
```

```
JOIN Classes ON Students.class_id = Classes.class_id
```

```
JOIN Teachers ON Classes.teacher_id = Teachers.teacher_id;
```

student_name	teacher_name
Ahmed	Mr. Khan
Sara	Ms. Fatima
Ali	Mr. Khan
Ayesha	Mr. Bilal
Usman	Ms. Fatima

-- 9. Calculate average marks for each subject

```
SELECT subject, AVG(marks) AS average_marks
```

```
FROM Marks
```

```
GROUP BY subject;
```

subject	average_marks
Math	79.0000
Science	80.0000
English	75.0000

-- 10. Count students in each class

```
SELECT class_id, COUNT(*) AS total_students
```

```
FROM Students
```

GROUP BY class_id;

class_id	total_students
1	3
2	2
3	3

-- 11. Get the highest marks scored in Science

SELECT MAX(marks) AS highest_science_marks

FROM Marks

WHERE subject = 'Science';

highest_science_marks
95

-- 12. List students who scored above average marks (across all subjects)

SELECT name

FROM Students

WHERE student_id IN (

SELECT student_id

FROM Marks

GROUP BY student_id

HAVING AVG(marks) > (

SELECT AVG(marks) FROM Marks

)

);

name
Ahmed
Ali
Usman
Zara

-- 13. Find class name of the oldest student

```
SELECT class_name
FROM Classes
WHERE class_id = (
  SELECT class_id
  FROM Students
  ORDER BY age DESC
  LIMIT 1
);
```

class_name
Class 8

-- 14. Add a new student record: 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;
```

student_id	name	age	gender	class_id
5	Usman	21	Male	2
6	Zara	22	Female	3
7	Hassan	20	Male	1
8	Ali	17	Male	3

-- 15. Update the subject of the teacher with ID = 101 to Computer Science

```
UPDATE Teachers
```



```
SET subject = 'Computer Science'
```

```
WHERE teacher_id = 101;
```

```
select * FROM Teachers;
```

teacher_id	name	subject
101	Mr. Khan	Computer Science
102	Ms. Fatima	Science
103	Mr. Bilal	English

```
-- 16. Delete students older than 25 years
```

```
SET SQL_SAFE_UPDATES = 0;
```

```
DELETE FROM Students
```

```
WHERE age > 25;
```

```
SET SQL_SAFE_UPDATES = 1;
```

```
Select * FROM Students;
```

student_id	name	age	gender	class_id
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

```
-- 17. Get names of students without 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

-- 18. Display number of male & female students in each class

```
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 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	0	2

-- 19. Show each student with their total marks across all subjects, ordered by marks

```
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	143
Usman	95
Ali	90

-- 20. Create a temporary table storing result of Query #8 (Student + Teacher names)

DROP TEMPORARY TABLE IF EXISTS Temp_Student_Teachers;

CREATE TEMPORARY TABLE Temp_Student_Teachers AS

SELECT Students.name AS student_name, Teachers.name AS teacher_name

FROM Students

JOIN Classes ON Students.class_id = Classes.class_id

JOIN Teachers ON Classes.teacher_id = Teachers.teacher_id;

Select * FROM Temp_Student_Teachers;

student_name	teacher_name
Ahmed	Mr. Khan
Sara	Ms. Fatima
Ali	Mr. Khan
Ayesha	Mr. Bilal
Usman	Ms. Fatima