

## How This Test Helps Students

- **Basic Queries:** Practice retrieving data from tables.
- **WHERE Clause:** Filtering based on conditions.
- **ORDER BY:** Sorting results based on specific columns.
- **Nested Queries:** Introductory level to spark curiosity for more advanced SQL topics.

## Database: **school\_test**

---

### 1. Table: **students**

```
CREATE TABLE students (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    age INT,  
    grade VARCHAR(5),  
    city VARCHAR(50)  
);  
  
-- Insert sample data  
INSERT INTO students (name, age, grade, city)  
VALUES  
( 'Amit Sharma', 14, 'A', 'Delhi'),  
( 'Neha Verma', 15, 'B', 'Mumbai'),  
( 'Ravi Kumar', 13, 'A', 'Bangalore'),  
( 'Priya Singh', 16, 'C', 'Kolkata'),  
( 'Anjali Gupta', 14, 'B', 'Delhi'),  
( 'Rajesh Yadav', 15, 'A', 'Chennai'),  
( 'Kiran Das', 16, 'B', 'Pune'),  
( 'Pooja Iyer', 14, 'C', 'Hyderabad');
```

---

### 2. Table: **subjects**

```
CREATE TABLE subjects (  
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
        subject_name VARCHAR(100),
        teacher_name VARCHAR(100)
    );

-- Insert sample data
INSERT INTO subjects (subject_name, teacher_name)
VALUES
('Mathematics', 'Mr. Joshi'),
('Science', 'Ms. Mehra'),
('English', 'Mrs. Iyer'),
('History', 'Mr. Menon'),
('Art', 'Ms. Nair');
```

---

### 3. Table: scores

```
CREATE TABLE scores (
    student_id INT,
    subject_id INT,
    marks INT
);

-- Insert sample data
INSERT INTO scores (student_id, subject_id, marks)
VALUES
(1, 1, 85),
(1, 2, 78),
(1, 3, 92),
(2, 1, 72),
(2, 2, 65),
(2, 3, 80),
(3, 1, 90),
(3, 2, 85),
(3, 3, 88),
(4, 1, 60),
(4, 2, 55),
(4, 3, 70),
(5, 1, 95),
```

(5, 2, 90),  
(5, 3, 85),  
(6, 1, 88),  
(6, 2, 72),  
(6, 3, 78),  
(7, 1, 74),  
(7, 2, 66),  
(7, 3, 80),  
(8, 1, 59),  
(8, 2, 63),  
(8, 3, 69);

---

## Practice Problems

### Basic SELECT Queries

1. Retrieve all student details.
2. List all subjects and their teachers.
3. Find all students who live in Delhi.
4. Get the details of students who are older than 14.

### Queries with **WHERE** and Conditions

5. Find students who are in grade 'A'.
6. List students who are either in 'Delhi' or 'Mumbai'.
7. Find students aged between 14 and 16.

### Queries with **ORDER BY**

8. List students in ascending order of their age.
9. Retrieve student scores sorted by marks in descending order.
10. List students in 'Bangalore' sorted by their name.

### Nested Queries (Simple)

11. Find the names of students who scored above 90 in any subject.
12. Get the names of students who scored less than 60 in any subject.

### Advanced Practice Problems

1. Find students who scored the highest marks in Mathematics (subject\_id = 1).
2. List students who scored lower than the average marks in Science (subject\_id = 2).
3. Display students who did not score in English (subject\_id = 3).
4. Find the second-highest marks in History (subject\_id = 4).
5. List students whose total score across all subjects exceeds 250