Q1. Create a table named students with fields:
• stdid INT PRIMARY KEY
• stdname VARCHAR(50)
• age INT
• city VARCHAR(50)

```
mysql> CREATE TABLE students (
    ->     stdid INT PRIMARY KEY,
    ->     stdname VARCHAR(50),
    ->     age INT,
    ->     city VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.078 sec)
```

Q2. Insert the following records into the students table:

```
mysql> INSERT INTO students (stdid, stdname, age, city) VALUES
    -> (1, 'Rohan', 20, 'Pune'),
    -> (2, 'Meera', 22, 'Mumbai'),
    -> (3, 'Arjun', 21, 'Delhi'),
    -> (4, 'Kavya', 23, 'Pune'),
    -> (5, 'Neha', 22, 'Kolkata');
Query OK, 5 rows affected (0.031 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

Q3. Display all student records.

```
mysql> SELECT * FROM students;
+-------+---------+------+---------+
| stdid | stdname | age  | city    |
+-------+---------+------+---------+
|     1 | Rohan   |   20 | Pune    |
|     2 | Meera   |   22 | Mumbai  |
|     3 | Arjun   |   21 | Delhi   |
|     4 | Kavya   |   23 | Pune    |
|     5 | Neha    |   22 | Kolkata |
+-------+---------+------+---------+
5 rows in set (0.001 sec)
```

Q4. Display only the name and age of all students.

```
mysql> SELECT stdname, age FROM students;
+---------+------+
| stdname | age  |
+---------+------+
| Rohan   |   20 |
| Meera   |   22 |
| Arjun   |   21 |
| Kavya   |   23 |
| Neha    |   22 |
+---------+------+
5 rows in set (0.000 sec)
```

Q5. Display students who are from Pune.

```
mysql> SELECT * FROM students
    -> WHERE city = 'Pune';
+-------+---------+------+------+
| stdid | stdname | age  | city |
+-------+---------+------+------+
|     1 | Rohan   |   20 | Pune |
|     4 | Kavya   |   23 | Pune |
+-------+---------+------+------+
2 rows in set (0.001 sec)
```

Q6. Display students whose age is greater than 21.

```
mysql> SELECT * FROM students
    -> WHERE age > 21;
+-------+---------+------+---------+
| stdid | stdname | age  | city    |
+-------+---------+------+---------+
|     2 | Meera   |   22 | Mumbai  |
|     4 | Kavya   |   23 | Pune    |
|     5 | Neha    |   22 | Kolkata |
+-------+---------+------+---------+
3 rows in set (0.000 sec)
```

Q7. Display students in descending order of age.

```
mysql> SELECT * FROM students
    -> ORDER BY age DESC;
+-------+---------+------+---------+
| stdid | stdname | age  | city    |
+-------+---------+------+---------+
|     4 | Kavya   |   23 | Pune    |
|     2 | Meera   |   22 | Mumbai  |
|     5 | Neha    |   22 | Kolkata |
|     3 | Arjun   |   21 | Delhi   |
|     1 | Rohan   |   20 | Pune    |
+-------+---------+------+---------+
5 rows in set (0.001 sec)
```

Q8. Count how many students belong to each city. (Use GROUP BY)

```
mysql> SELECT city, COUNT(*) AS total_students
    -> FROM students
    -> GROUP BY city;
+---------+----------------+
| city    | total_students |
+---------+----------------+
| Pune    |              2 |
| Mumbai  |              1 |
| Delhi   |              1 |
| Kolkata |              1 |
+---------+----------------+
4 rows in set (0.002 sec)
```

Q9. Display students whose name starts with 'K'. (Use LIKE)

```
mysql> SELECT * FROM students
    -> WHERE stdname LIKE 'K%';
+-------+---------+------+------+
| stdid | stdname | age  | city |
+-------+---------+------+------+
|     4 | Kavya   |   23 | Pune |
+-------+---------+------+------+
1 row in set (0.000 sec)
```

Q10. Delete student whose stdid = 5.

```
mysql> DELETE FROM students
    -> WHERE stdid = 5;
Query OK, 1 row affected (0.009 sec)
```

PART 2 — JOIN PRACTICE

Tables:

Table: students

stdid	student_name	city
1	Rohan	Pune
2	Meera	Mumbai
3	Arjun	Delhi
4	Kavya	Pune

Table: marks

stdid	subject	marks
1	Maths	88

2	Maths	76
3	Maths	92
5	Maths	67

Q11. Display student name and marks of only those students who have matching IDs in both tables. (Students without marks should not appear.) LEFT JOIN

```
mysql> SELECT s.student_name, m.marks
    -> FROM students s
    -> INNER JOIN marks m
    -> ON s.stdid = m.stdid;
+--------------+-------+
| student_name | marks |
+--------------+-------+
| Rohan        |    88 |
| Meera        |    76 |
| Arjun        |    92 |
+--------------+-------+
3 rows in set (0.000 sec)
```

Q12. Display all students and their marks. (If marks not available, show NULL.) RIGHT JOIN

```
mysql> SELECT s.student_name, m.marks
    -> FROM students s
    -> LEFT JOIN marks m
    -> ON s.stdid = m.stdid;
+--------------+-------+
| student_name | marks |
+--------------+-------+
| Rohan        |    88 |
| Meera        |    76 |
| Arjun        |    92 |
| Kavya        |  NULL |
+--------------+-------+
4 rows in set (0.000 sec)
```

Q13. Display all marks records along with student names. (If student doesn’t exist in students table, show NULL.) CROSS JOIN

```
mysql> SELECT s.student_name, m.marks
    -> FROM students s
    -> RIGHT JOIN marks m
    -> ON s.stdid = m.stdid;
+--------------+-------+
| student_name | marks |
+--------------+-------+
| Rohan        |    88 |
| Meera        |    76 |
| Arjun        |    92 |
| NULL         |    67 |
+--------------+-------+
4 rows in set (0.000 sec)
```

Q14. Display all possible combinations of students and subjects. (Use CROSS JOIN between students and marks table to show every pair.) JOIN with Filtering

```
mysql> SELECT s.student_name, m.subject
    -> FROM students s
    -> CROSS JOIN marks m;
+--------------+---------+
| student_name | subject |
+--------------+---------+
| Kavya        | Maths   |
| Arjun        | Maths   |
| Meera        | Maths   |
| Rohan        | Maths   |
| Kavya        | Maths   |
| Arjun        | Maths   |
| Meera        | Maths   |
| Rohan        | Maths   |
| Kavya        | Maths   |
| Arjun        | Maths   |
| Meera        | Maths   |
| Rohan        | Maths   |
| Kavya        | Maths   |
| Arjun        | Maths   |
| Meera        | Maths   |
| Rohan        | Maths   |
+--------------+---------+
16 rows in set (0.000 sec)
```

Q15. Using INNER JOIN, display students who scored more than 80

```
mysql> SELECT s.student_name, m.marks
    -> FROM students s
    -> INNER JOIN marks m
    -> ON s.stdid = m.stdid
    -> WHERE m.marks > 80;
+--------------+-------+
| student_name | marks |
+--------------+-------+
| Rohan        |    88 |
| Arjun        |    92 |
+--------------+-------+
2 rows in set (0.000 sec)
```