Q1. Create a table named students with fields:

- stdid INT PRIMARY KEY
- stdname VARCHAR(50)
- age INT
- city VARCHAR(50)

Ans:

```
mysql> desc students;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| stdid   | int         | NO   | PRI | NULL    |       |
| stdname | varchar(50) | YES  |     | NULL    |       |
| age     | int         | YES  |     | NULL    |       |
| city    | varchar(50) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

Q2. Insert the following records into the students table:

stdid	stdname	age	city
1	Rohan	20	Pune
2	Meera	22	Mumbai
3	Arjun	21	Delhi
4	Kavya	23	Pune
5	Neha	22	Kolkata

Ans:

```
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 (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

Q3. Display all student records.

Ans:

```
+-------+---------+-----+---------+
| stdid | stdname | age | city    |
+-------+---------+-----+---------+
|     1 | Rohan   |  20 | Pune    |
|     2 | Meera   |  22 | Mumbai  |
|     3 | Arjun   |  21 | Delhi   |
|     4 | Kavya   |  23 | Pune    |
|     5 | Neha    |  22 | kolkata |
+-------+---------+-----+---------+
```

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

Ans:

```
mysql> select stdname,age from students;
+---------+------+
| stdname | age  |
+---------+------+
| Rohan   |   20 |
| Meera   |   22 |
| Arjun   |   21 |
| Kavya   |   23 |
| Neha    |   22 |
+---------+------+
5 rows in set (0.00 sec)
```

Q5. Display students who are from Pune.

Ans:

```
+-------+---------+-----+------+
| stdid | stdname | age | city |
+-------+---------+-----+------+
|     1 | Rohan   |  20 | Pune |
|     4 | Kavya   |  23 | Pune |
+-------+---------+-----+------+
2 rows in set (0.00 sec)
```

Q6. Display students whose age is greater than 21.

Ans:

```
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 (0.00 sec)
```

Q7. Display students in descending order of age.

Ans:

```
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 (0.00 sec)
```

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

Ans:

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

Ans:

```
mysql>  select * from students where stdname  like 'k%';
+--------+----------+------+------+
| stdid | stdname | age  | city |
+--------+----------+------+------+
|     4 | Kavya   |   23 | Pune |
+--------+----------+------+------+
1 row in set (0.00 sec)
```

Q10. Delete student whose stdid = 5.

Ans :

```
mysql> select * from students;
+--------+----------+------+---------+
| stdid | stdname | age  | city    |
+--------+----------+------+---------+
|     1 | Rohan   |   20 | Pune    |
|     2 | Meera   |   22 | Mumbai  |
|     3 | Arjun   |   21 | Delhi   |
|     4 | Kavya   |   23 | Pune    |
+--------+----------+------+---------+
4 rows in set (0.00 sec)
```

Q11. Display student name and marks of only those students who have matching IDs in both tables.

Ans:

```
mysql> select students.stdname , marks.marks from students inner join marks on students.stdid=marks.stdid;
+----------+--------+
| stdname | marks |
+----------+--------+
| Rohan   |    88 |
| Meera   |    76 |
| Arjun   |    92 |
+----------+--------+
3 rows in set (0.00 sec)
```

Q12. Display all students and their marks.

Ans:

```
mysql> select students.stdname , marks.marks from students left join marks on students.stdid=marks.stdid;
+----------+--------+
| stdname | marks |
+----------+--------+
| Rohan   |    88 |
| Meera   |    76 |
| Arjun   |    92 |
| Kavya   |  NULL |
+----------+--------+
4 rows in set (0.00 sec)
```

Q13.Display all marks records along with student names.

```
mysql> select students.stdname , marks.marks from students right join marks on students.stdid=marks.stdid;
+----------+--------+
| stdname | marks |
+----------+--------+
| Rohan   |    88 |
| Meera   |    76 |
| Arjun   |    92 |
| NULL    |    67 |
+----------+--------+
4 rows in set (0.00 sec)
```

Q14. Display all possible combinations of students and subjects.

Ans:

```
mysql> select students.stdname , marks.subject from students cross join marks ;
+---------+---------+
| stdname | 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 (0.00 sec)
```

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

Ans:

```
mysql> SELECT students.stdname FROM students INNER JOIN marks ON students.stdId = marks.stdId WHERE marks.marks > 80;
+----------+
| stdname |
+----------+
| Rohan   |
| Arjun   |
+----------+
2 rows in set (0.00 sec)
```