MySQL Practice Questions with Answers

1. Create the database and required tables (students, courses, enrollments).

```
CREATE DATABASE school;
USE school;

CREATE TABLE students (
    student_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(50),
    email VARCHAR(100) UNIQUE,
    age INT,
    joined_date DATE
);

CREATE TABLE courses (
    course_id INT PRIMARY KEY AUTO_INCREMENT,
    course_name VARCHAR(50),
    fee INT
);

CREATE TABLE enrollments (
    enroll_id INT PRIMARY KEY AUTO_INCREMENT,
    student_id INT,
    course_id INT,
    enrolled_on DATE,
    FOREIGN KEY(student_id) REFERENCES students(student_id),
    FOREIGN KEY(course_id) REFERENCES courses(course_id)
);
```

2. Insert at least 4 students, 3 courses, and some enrollments.

```
INSERT INTO students(name, email, age, joined_date) VALUES
('Mehul','mehul@gmail.com',21,'2024-01-01'),
('Riya','riya@gmail.com',22,'2024-01-10'),
('Aman','aman@gmail.com',20,'2024-02-05'),
('Nisha','nisha@gmail.com',23,'2024-02-15');

INSERT INTO courses(course_name, fee) VALUES
('MySQL',1000),
('Java',1500),
('React',2000);

INSERT INTO enrollments(student_id, course_id, enrolled_on) VALUES
(1,1,'2024-02-01'),
(1,2,'2024-02-03'),
(2,1,'2024-02-05'),
(3,3,'2024-02-07');
```

3. Update a student's email.

```
UPDATE students SET email='newmail@gmail.com' WHERE student_id=1;
```

4. Delete a course by ID.

```
DELETE FROM courses WHERE course_id = 3;
```

5. Find students aged between 20 and 22.

```
SELECT * FROM students WHERE age BETWEEN 20 AND 22;
```

6. Find students whose name contains 'a'.

```
SELECT * FROM students WHERE name LIKE '%a%';
```

7. Sort students by age (oldest first) and show only top 2.

```
SELECT * FROM students ORDER BY age DESC LIMIT 2;
```

8. Count total number of enrolled students.

```
SELECT COUNT(*) FROM students;
```

9. Show how many courses each student has enrolled in.

```
SELECT student_id, COUNT(*) AS total
FROM enrollments
GROUP BY student_id;
```

10. Show students who enrolled in more than 1 course.

```
SELECT student_id, COUNT(*) AS total
FROM enrollments
GROUP BY student_id
HAVING total > 1;
```

11. Join students with courses they enrolled in.

```
SELECT s.name, c.course_name, c.fee
FROM enrollments e
JOIN students s ON e.student_id=s.student_id
JOIN courses c ON e.course_id=c.course_id;
```

12. Create a view showing student name, course name, and enrollment date.

```
CREATE VIEW student_course_view AS
SELECT s.name, c.course_name, e.enrolled_on
FROM enrollments e
JOIN students s ON e.student_id=s.student_id
JOIN courses c ON e.course_id=c.course_id;
```

13. Find all students who enrolled in the MySQL course using a subquery.

```
SELECT name FROM students
WHERE student_id IN (
   SELECT student_id
   FROM enrollments
   WHERE course_id = (
      SELECT course_id FROM courses WHERE course_name='MySQL'
   )
);
```

14. Create a stored procedure to show all courses of a student by ID.

```
DELIMITER $$

CREATE PROCEDURE getStudentCourses(IN sid INT)
BEGIN
   SELECT s.name, c.course_name
   FROM enrollments e
   JOIN students s ON e.student_id=s.student_id
   JOIN courses c ON e.course_id=c.course_id
   WHERE s.student_id = sid;
END $$

DELIMITER ;
```

15. Rank students by age using a window function.

```
SELECT name, age,
      RANK() OVER (ORDER BY age DESC) AS age_rank
FROM students;
```