Final Exam: Database Management Systems – SQL Solutions

Part A – Basic SQL (Short Questions – 5 Marks)

1. Create table Students with constraints

```
CREATE TABLE Students (
    student_id INT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE,
    dob DATE,
    department_id INT,
    FOREIGN KEY (department_id) REFERENCES Departments(department_id)
);
```

2. Insert two new rows

```
INSERT INTO Students (student_id, name, email, dob, department_id)
VALUES
(101, 'John Doe', 'john@example.com', '2002-06-15', 2),
(102, 'Jane Smith', 'jane@example.com', '2001-10-30', 1);
```

3. Retrieve all students born after 2002, sorted by name descending

```
SELECT *
FROM Students
WHERE dob > '2002-01-01'
ORDER BY name DESC;
```

4. Display total number of students in each department

```
SELECT department_id, COUNT(*) AS total_students
FROM Students
GROUP BY department_id;
```

5. Delete all students whose dob is before 2000-01-01

```
DELETE FROM Students
WHERE dob < '2000-01-01';
```

Part B – Intermediate SQL (JOIN, GROUP BY, Subquery – 15 Marks)

6. Retrieve name and department_name of all students

```
SELECT s.name, d.department_name
FROM Students s
JOIN Departments d ON s.department_id = d.department_id;
```

7. Find average grade of each course (only > 3.0)

```
SELECT c.course_name, AVG(e.grade) AS avg_grade
FROM Enrollments e
JOIN Courses c ON e.course_id = c.course_id
GROUP BY c.course_name
HAVING AVG(e.grade) > 3.0;
```

8. Display all students not enrolled in any course

```
SELECT s.name
FROM Students s
LEFT JOIN Enrollments e ON s.student_id = e.student_id
WHERE e.student_id IS NULL;
```

9. Display course_name and number of students enrolled (highest → lowest)

```
SELECT c.course_name, COUNT(e.student_id) AS total_students
FROM Courses c
LEFT JOIN Enrollments e ON c.course_id = e.course_id
GROUP BY c.course_name
```

```
ORDER BY total_students DESC;
```

10. List all students whose grade in any course is above the average grade of that course

```
SELECT DISTINCT s.name
FROM Students s
JOIN Enrollments e ON s.student_id = e.student_id
WHERE e.grade > (
    SELECT AVG(e2.grade)
    FROM Enrollments e2
    WHERE e2.course_id = e.course_id
);
```