Code

Create Database

```
CREATE DATABASE school_db;
USE school_db;
SHOW DATABASES;
```

Show Tables

```
SHOW TABLES;
```

1. DATA DEFINITION LANGUAGE (DDL)

A) Students

```
CREATE TABLE students (
    student_id INT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    age INT,
    contact BIGINT UNIQUE,
    class_id INT
);
DESC students;
```

-- B) Teacher

```
CREATE TABLE teacher (
    teacher_id INT PRIMARY KEY,
    name VARCHAR(100),
    subject VARCHAR(50),
    salary DECIMAL(10, 2),
    joining_date DATE
);
DESC teacher;
```

C) Classes

```
CREATE TABLE classes (
    class_id INT PRIMARY KEY,
    class_name VARCHAR(50),
    teacher_id INT,
    FOREIGN KEY (teacher_id) REFERENCES teachers(teacher_id)
);
DESC classes;
```

D) Non-teaching staff

```
CREATE TABLE non_teaching_staff (
    staff_id INT PRIMARY KEY,
```

```
    name VARCHAR(100) NOT NULL,
    salary DECIMAL(10, 2),
    joining_date DATE
);
DESC non_teaching_staff;
```

E) Subject

```
CREATE TABLE subjects (
    subject_id INT PRIMARY KEY,
    name VARCHAR(100) NOT NULL
);
DESC subjects;
```

F) Marks

```
CREATE TABLE marks (
    mark_id INT PRIMARY KEY,
    student_id INT,
    subject_id INT,
    marks_obtained DECIMAL(5,2),
    exam_date DATE,
    FOREIGN KEY (student_id) REFERENCES student(student_id),
    FOREIGN KEY (subject_id) REFERENCES subjects(subject_id)
);
DESC marks;
```

Alter Table

```
ALTER TABLE students ADD gender VARCHAR(10);
ALTER TABLE teacher MODIFY name VARCHAR(150);
ALTER TABLE students CHANGE student_id st_id VARCHAR(100);
ALTER TABLE non_teaching_staff DROP COLUMN salary;
RENAME TABLE non_teaching_staff TO support_staff;
```

Truncate & Drop

```
TRUNCATE TABLE support_staff;
DROP TABLE support_staff;
```

2. DATA MANIPULATION LANGUAGE (DML)

Insert

```
INSERT INTO students (st_id, name, age, contact, class_id, gender) VALUES
(1, 'Anjali Sharma', 15, 9876543211, 101, 'Female');
SELECT * FROM students;
```

Update

```
UPDATE students SET contact = 9876500000, age = 17 WHERE st_id = 1;
SELECT * FROM students;
```

Delete

```
DELETE FROM subjects WHERE subject_id= 3;
SELECT * FROM subjects;
```

3. DATA QUERY LANGUAGE (DQL)

Select Queries

```
SELECT * FROM students;
SELECT st_id, name FROM students;
SELECT name AS fullname FROM students;
```

Order By

```
SELECT * FROM teacher ORDER BY salary;
SELECT * FROM subjects ORDER BY subject_id DESC;
```

Limit

```
SELECT * FROM marks ORDER BY marks_obtained DESC LIMIT 5;
```

Distinct

```
SELECT DISTINCT gender FROM students;
```

WHERE Clause

```
SELECT * FROM students WHERE age > 15;
SELECT * FROM students WHERE age < 16 AND gender='Female';
SELECT * FROM teacher WHERE (subject = 'English' OR subject = 'Mathematics') AND
salary > 45000;
SELECT * FROM teacher WHERE NOT subject = 'Mathematics';
SELECT * FROM teacher WHERE subject IS NOT NULL;
SELECT * FROM marks WHERE marks_obtained BETWEEN 70 AND 90;
SELECT * FROM students WHERE class_id IN (101, 103, 104);
SELECT * FROM students WHERE age > ANY (SELECT age FROM students WHERE
class_id = 101);
SELECT * FROM students WHERE age < ALL (SELECT age FROM students WHERE
class_id = 104);
```

Aggregate Functions

```
SELECT COUNT(*) AS total_number_students FROM students;
SELECT ROUND(AVG(marks_obtained), 2) AS average_students_marks FROM marks;
SELECT SUM(salary) AS total_teacher_salary FROM teacher;
```

```
SELECT MAX(salary) AS highest_salary, MIN(salary) AS lowest_salary FROM teacher;
```

Group By

```
SELECT class_id, COUNT(*) AS student_count FROM students GROUP BY class_id;
SELECT class_id, gender, COUNT(*) AS count FROM students GROUP BY class_id,
gender;
SELECT class_id, AVG(age) AS average_age FROM students GROUP BY class_id;
```

Having

```
SELECT class_id, COUNT(*) AS student_count FROM students GROUP BY class_id
HAVING COUNT(*) > 2;
```

LIKE

```
SELECT * FROM teacher WHERE name LIKE 'S%';
SELECT * FROM teacher WHERE name LIKE '%dy';
```

Union

```
SELECT teacher_id FROM teacher
UNION
SELECT teacher_id FROM classes;
```

Joins

```
SELECT t.teacher_id, t.name AS teacher_name, t.subject, s.subject_id
FROM teacher t
INNER JOIN subjects s ON t.subject = s.name;

SELECT c.class_id, c.class_name, c.teacher_id, t.name AS teacher_name, t.subject
FROM classes c
LEFT JOIN teacher t ON c.teacher_id = t.teacher_id;
```

Subqueries

```
SELECT name FROM students WHERE st_id = (
   SELECT student_id FROM marks ORDER BY marks_obtained DESC LIMIT 1
);
SELECT name FROM subjects WHERE subject_id IN (
   SELECT DISTINCT subject_id FROM marks
);
SELECT name, class_id FROM students WHERE st_id IN (
   SELECT student_id FROM marks WHERE marks_obtained > 85);
```

View

```
CREATE VIEW Student_information AS
SELECT st_id, name, age, contact, class_id, gender FROM students;
SELECT * FROM Student_information;
```