```
CREATE TABLE table_name (
  column1 datatype CONSTRAINT,
  column2 datatype CONSTRAINT,
  ...
);
CREATE TABLE Student (
  RollNo INT PRIMARY KEY,
  Name VARCHAR(50),
  Dept VARCHAR(20),
  Year INT,
  Percentage FLO
AT
);
INSERT INTO table_name (column1, column2, ...) VALUES (value1, value2, ...);
INSERT INTO Student (RollNo, Name, Dept, Year, Percentage)
VALUES (1, 'Amit', 'IT', 2, 78.5;
SELECT * FROM Student;
SELECT Name, Dept FROM Student WHERE Percentage > 70;
SELECT * FROM Student ORDER BY Percentage DESC;
SELECT DISTINCT Dept FROM Student;
UPDATE table_name SET column1 = value1 WHERE condition;
UPDATE Student SET Percentage = 85 WHERE RollNo = 1;
DELETE FROM table_name WHERE condition;
DELETE FROM Student WHERE RollNo = 1;
ALTER TABLE Student ADD Email VARCHAR(100);
ALTER TABLE Student MODIFY Percentage DECIMAL(5,2);
ALTER TABLE Student DROP COLUMN Email;
```

```
DROP TABLE Student;

CREATE TABLE Dept (

  DeptID INT PRIMARY KEY,

  DeptName VARCHAR(50);

CREATE TABLE Student (

  RollNo INT PRIMARY KEY,

  Name VARCHAR(50),

  DeptID INT,

  FOREIGN KEY (DeptID) REFERENCES Dept(DeptID

)

);

SELECT COUNT(*) FROM Student;

SELECT MAX(Percentage) FROM Student;

SELECT MIN(Percentage) FROM Student;

SELECT AVG(Percentage) FROM Student;

SELECT SUM(Percentage) FROM Student;

SELECT Dept, AVG(Percentage)

FROM Student

GROUP BY Dept

HAVING AVG(Percentage) >

70;

SELECT Student.Name, Dept.DeptName

FROM Student

INNER JOIN Dept ON Student.DeptID = Dept.DeptID;




SELECT Student.Name, Dept.DeptName
```

```
FROM Student

LEFT JOIN Dept ON Student.DeptID = Dept.DeptID;



SELECT Student.Name, Dept.DeptName

FROM Student

RIGHT JOIN Dept ON Student.DeptID = Dept.DeptID;



SELECT Name FROM Student

WHERE DeptID IN (SELECT DeptID FROM Dept WHERE DeptName = 'IT')

;

SELECT Name FROM Student

WHERE Percentage = (SELECT MAX(Percentage) FROM Student)

;

CREATE VIEW HighScorers AS

SELECT Name, Percentage FROM Student WHERE Percentage > 75;



SELECT * FROM High scores;

CREATE INDEX idx_name ON Student(Name);

BEGIN;



UPDATE Student SET Percentage = 90 WHERE RollNo = 2;



SAVEPOINT before_discount;



UPDATE Student SET Percentage = Percentage - 5;



ROLLBACK TO before_discount;
```

```
CREATE VIEW view_name AS

SELECT columns

FROM table

WHERE conditi

on;

COMMIT;



CREATE VIEW Top_Students AS

SELECT Name, Percentage

FROM Student

WHERE Percentage >

85;



SELECT * FROM IT_Students;



CREATE INDEX index_name ON table(column);

CREATE INDEX idx_name ON Student(Name);

CREATE INDEX idx_dept_percentage ON Student(Dept, Percentage);

DROP INDEX idx_name ON Student;



CREATE TRIGGER trigger_name

AFTER INSERT ON table_name

FOR EACH ROW

BEGIN

  -- SQL statement

END;
```

```
CREATE TABLE Student_Log (
  RollNo INT,
  Action VARCHAR(20),
  LogTime TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);


CREATE TRIGGER after_insert_student
AFTER INSERT ON Student
FOR EACH ROW
BEGIN
  INSERT INTO Student_Log (RollNo, Action)
  VALUES (NEW.RollNo, 'INSER
TED');
END;



CREATE TRIGGER prevent_low_percentage
BEFORE INSERT ON Student
FOR EACH ROW
BEGIN
  IF NEW.Percentage < 35 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Percentage too low to insert';
  END I
F;
END;
```

```
DELIMITER $$



CREATE PROCEDURE procedure_name(IN param1 datatype, OUT param2 datatype)

BEGIN

  -- SQL statements

END $$

DELIMITER ;



DELIMITER $$



CREATE PROCEDURE GetAvgPercentage(IN deptName VARCHAR(20), OUT avgPercent FLOAT)

BEGIN

  SELECT AVG(Percentage) INTO avgPercent

  FROM Student

  WHERE Dept = deptName;

END $$

DELIMITER ;



CALL GetAvgPercentage('IT', @avg);

SELECT @avg;



DELIMITER $$



CREATE PROCEDURE AddStudent(

  IN rno INT,
```

```
    IN sname VARCHAR(50),
    IN dept VARCHAR(20),
    IN year INT,
    IN perc FLOAT
)
BEGIN
    INSERT INTO Student(RollNo, Name, Dept, Year, Percentage)
    VALUES (rno, sname, dept, year, perc);
END $$
DELIMITER ;
-- Calling it:
CALL AddStudent(4, 'Sneha'
, 'CS', 2, 89.5);
```