```
-- Create Database
CREATE DATABASE Student;
USE Student;

-- Create Tables
CREATE TABLE Stud (
    Roll_no INT PRIMARY KEY,
    Name VARCHAR(50),
    Marks INT
);

CREATE TABLE Result (
    Roll_no INT,
    Name VARCHAR(50),
    Class VARCHAR(50)
);

-- Insert Sample Data
INSERT INTO Stud VALUES (1, 'Ravi', 1200);
INSERT INTO Stud VALUES (2, 'Sneha', 950);
INSERT INTO Stud VALUES (3, 'Harshad', 880);
INSERT INTO Stud VALUES (4, 'Priya', 840);
INSERT INTO Stud VALUES (5, 'Kiran', 700);
INSERT INTO Stud VALUES (6, 'Invalid', -50);   -- test negative marks

-- ------------------------------------------------------
```

```
-- ✅ FUNCTION fn_Grade() WITH SIGNAL FOR NEGATIVE MARKS
-- ------------------------------------------------------
DELIMITER $$

CREATE FUNCTION fn_Grade(marks INT)
RETURNS VARCHAR(50)
DETERMINISTIC
BEGIN
    DECLARE grade VARCHAR(50);

    -- ✅ Throw error if marks < 0
    IF marks < 0 THEN
        SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Marks cannot be negative!';
    END IF;

    -- ✅ Throw error if NULL
    IF marks IS NULL THEN
        SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Marks cannot be NULL!';
    END IF;

    -- ✅ Normal grade calculation
    IF marks BETWEEN 990 AND 1500 THEN
        SET grade = 'Distinction';
```

```
    ELSEIF marks BETWEEN 900 AND 989 THEN
        SET grade = 'First Class';

    ELSEIF marks BETWEEN 825 AND 899 THEN
        SET grade = 'Second Class';

    ELSE
        SET grade = 'Fail';

    END IF;

    RETURN grade;
END $$

DELIMITER ;



-- ------------------------------------------------------
-- ✅ PROCEDURE WITH CONTINUE + EXIT HANDLERS
-- ------------------------------------------------------
DELIMITER $$

CREATE PROCEDURE proc_Grade(IN In_roll INT)
BEGIN
    DECLARE v_name VARCHAR(50);
    DECLARE v_marks INT;
```

```
DECLARE v_class VARCHAR(50);

DECLARE error_occurred INT DEFAULT 0;
DECLARE exit_message VARCHAR(100);



-- ✅ CONTINUE HANDLER for SQL Exceptions
DECLARE CONTINUE HANDLER FOR SQLEXCEPTION
BEGIN
   SET error_occurred = 1;
   SET v_class = 'Error';
END;

-- ✅ CONTINUE HANDLER for NOT FOUND
DECLARE CONTINUE HANDLER FOR NOT FOUND
BEGIN
   SET error_occurred = 1;
   SET v_class = 'Roll Not Found';
END;

-- ✅ EXIT HANDLER for SQL Warning
DECLARE EXIT HANDLER FOR SQLWARNING
BEGIN
   SET exit_message = 'Warning occurred!';
   SIGNAL SQLSTATE '45000'
      SET MESSAGE_TEXT = exit_message;
```

```
    END;


    -- ✅ STEP 1: Fetch student record
    SELECT Name, Marks INTO v_name, v_marks
    FROM Stud
    WHERE Roll_no = In_roll;


    -- ✅ STEP 2: If no error, call function
    IF error_occurred = 0 THEN
        SET v_class = fn_Grade(v_marks);
    END IF;


    -- ✅ STEP 3: Insert into result table
    INSERT INTO Result(Roll_no, Name, Class)
    VALUES (In_roll, v_name, v_class);

END $$

DELIMITER ;


-- ------------------------------------------------
-- ✅ CALL PROCEDURE FOR ALL STUDENTS
```

```
-- -----------------------------------------------------

CALL proc_Grade(1);

CALL proc_Grade(2);

CALL proc_Grade(3);

CALL proc_Grade(4);

CALL proc_Grade(5);


-- invalid marks will trigger SIGNAL SQLSTATE

CALL proc_Grade(6);


-- invalid roll triggers NOT FOUND handler

CALL proc_Grade(99);



-- -----------------------------------------------------

-- ✅ VIEW RESULT

-- -----------------------------------------------------

SELECT * FROM Result;
```