**📘 My Stored Programs – PL/ Tutorial (My Style)**

**🔹 1. Introduction to Stored Procedures**

A **stored procedure** is a group of statements that perform a specific task and are stored in the database.

**✅ Syntax:**

DELIMITER $$

CREATE PROCEDURE ProcedureName(IN param1 INT, OUT param2 VARCHAR(50))

BEGIN

-- Logic

END$$

DELIMITER ;

**✅ Example – Basic Procedure:**

DELIMITER $$

CREATE PROCEDURE GetCustomerPlan(IN customerId VARCHAR(20))

BEGIN

SELECT name, plan\_type FROM customers WHERE customer\_id = customerId;

END$$

DELIMITER ;

**🔹 2. Using Variables**

**✅ Declaring & Using Variables:**

DECLARE v\_name VARCHAR(100);

SET v\_name = 'John Doe';

**✅ Input, Output, and InOut Parameters:**

IN → Input only

OUT → Output only

INOUT → Input and Output

**🔹 3. Using Control Structures (IF, CASE, LOOP)**

**✅ IF-ELSE:**

IF balance > 100 THEN

SET status = 'Active';

ELSE

SET status = 'Inactive';

END IF;

**✅ WHILE Loop:**

WHILE counter < 10 DO

SET counter = counter + 1;

END WHILE;

**🔹 4. Cursors in My**

Cursors allow row-by-row processing of query results inside stored procedures.

**✅ Syntax:**

DECLARE done INT DEFAULT FALSE;

DECLARE v\_id VARCHAR(20);

DECLARE cur CURSOR FOR SELECT customer\_id FROM customers;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN cur;

read\_loop: LOOP

FETCH cur INTO v\_id;

IF done THEN

LEAVE read\_loop;

END IF;

-- Do something with v\_id

END LOOP;

CLOSE cur;

**🔹 5. Example – Deduct Monthly Fee for Silver Customers**

DELIMITER $$

CREATE PROCEDURE DeductMonthlyFee()

BEGIN

DECLARE done INT DEFAULT FALSE;

DECLARE v\_id VARCHAR(20);

DECLARE v\_balance DECIMAL(10,2);

DECLARE cur CURSOR FOR

SELECT customer\_id, balance FROM customers WHERE plan\_type = 'Silver';

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN cur;

read\_loop: LOOP

FETCH cur INTO v\_id, v\_balance;

IF done THEN

LEAVE read\_loop;

END IF;

IF v\_balance >= 50 THEN

UPDATE customers SET balance = balance - 50 WHERE customer\_id = v\_id;

END IF;

END LOOP;

CLOSE cur;

END$$

DELIMITER ;

**🔹 6. Exception Handling in My**

My supports basic error handling using DECLARE ... HANDLER.

**✅ Example – Handling Division by Zero:**

DECLARE CONTINUE HANDLER FOR EXCEPTION

BEGIN

SET @msg = 'Error occurred';

END;

**✅ Named Condition Example:**

DECLARE EXIT HANDLER FOR EXCEPTION

BEGIN

ROLLBACK;

END;

**🔹 7. Combine All – Procedure with Cursor + Exception**

DELIMITER $$

CREATE PROCEDURE SafeDeduction()

BEGIN

DECLARE done INT DEFAULT FALSE;

DECLARE v\_id VARCHAR(20);

DECLARE v\_balance DECIMAL(10,2);

DECLARE v\_error VARCHAR(100);

DECLARE cur CURSOR FOR

SELECT customer\_id, balance FROM customers WHERE plan\_type = 'Silver';

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

DECLARE CONTINUE HANDLER FOR EXCEPTION

BEGIN

SET v\_error = 'Error in deduction.';

END;

OPEN cur;

read\_loop: LOOP

FETCH cur INTO v\_id, v\_balance;

IF done THEN

LEAVE read\_loop;

END IF;

IF v\_balance >= 100 THEN

UPDATE customers SET balance = balance - 100 WHERE customer\_id = v\_id;

END IF;

END LOOP;

CLOSE cur;

END$$

DELIMITER ;

**🔹 8. Calling Procedures**

CALL DeductMonthlyFee();