**Name: Amogh Girish Nagarkar**

**Superset ID: 6403503**

**Exercise 1: Control Structures**

**Scenario 1:** The bank wants to apply a discount to loan interest rates for customers above 60 years old.

* + **Question:** Write a PL/SQL block that loops through all customers, checks their age, and if they are above 60, apply a 1% discount to their current loan interest rates.

**Scenario 2:** A customer can be promoted to VIP status based on their balance.

* + **Question:** Write a PL/SQL block that iterates through all customers and sets a flag IsVIP to TRUE for those with a balance over $10,000.

**Scenario 3:** The bank wants to send reminders to customers whose loans are due within the next 30 days.

* + **Question:** Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer.

**Scenario 1 Code:**

DELIMITER //

CREATE PROCEDURE ApplyInterestDiscount()

BEGIN

DECLARE i INT DEFAULT 0;

DECLARE total INT;

SELECT COUNT(\*) INTO total FROM Customers;

WHILE i < total DO

SET @cust\_id := NULL;

SET @dob := NULL;

SELECT CustomerID, DOB INTO @cust\_id, @dob

FROM Customers

LIMIT 1 OFFSET i;

SET @age := TIMESTAMPDIFF(YEAR, @dob, CURDATE());

SELECT CONCAT('Customer ID: ', @cust\_id, ', Age: ', @age) AS DebugOutput;

IF @age > 60 THEN

UPDATE Loans

SET InterestRate = InterestRate - 1.0

WHERE CustomerID = @cust\_id;

SELECT CONCAT('→ Discount applied to Customer ID: ', @cust\_id) AS UpdateStatus;

END IF;

SET i = i + 1;

END WHILE;

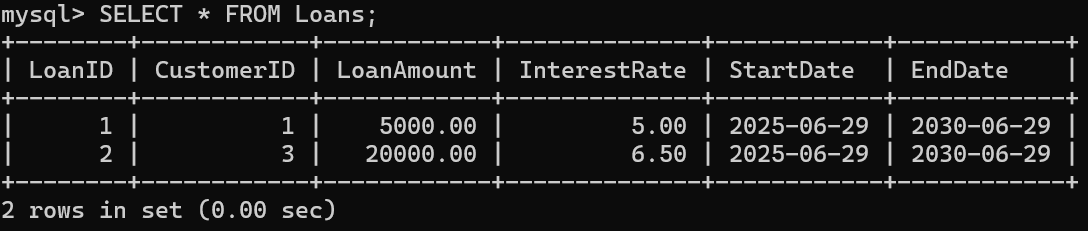
END;

//

DELIMITER ;

Output:





**Scenario 2 Code:**

DELIMITER //

CREATE PROCEDURE PromoteToVIP()

BEGIN

UPDATE Customers

SET IsVIP = TRUE

WHERE Balance > 10000;

SELECT CustomerID, Name, Balance, IsVIP

FROM Customers

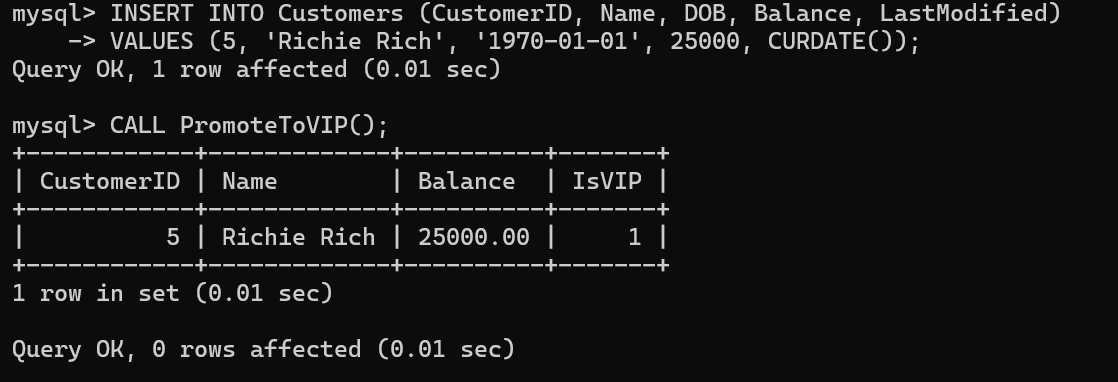
WHERE IsVIP = TRUE;

END;

//

DELIMITER ;

**Output:**

****

**Scenario 3 Code:**

DELIMITER //

CREATE PROCEDURE SendLoanReminders()

BEGIN

DECLARE done INT DEFAULT FALSE;

DECLARE cust\_id INT;

DECLARE name VARCHAR(100);

DECLARE end\_date DATE;

DECLARE cur CURSOR FOR

SELECT c.CustomerID, c.Name, l.EndDate

FROM Customers c

JOIN Loans l ON c.CustomerID = l.CustomerID

WHERE l.EndDate BETWEEN CURDATE() AND DATE\_ADD(CURDATE(), INTERVAL 30 DAY);

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN cur;

read\_loop: LOOP

FETCH cur INTO cust\_id, name, end\_date;

IF done THEN

LEAVE read\_loop;

END IF;

SELECT CONCAT('Reminder: Loan for customer ', name, ' (ID: ', cust\_id, ') is due on ', end\_date) AS ReminderMessage;

END LOOP;

CLOSE cur;

END;

//

DELIMITER ;

**Output:**

