**Exercise 4: Functions**

**Scenario** 1: Calculate the age of customers for eligibility checks.

**Question**: Write a function CalculateAge that takes a customer's date of birth as input and returns their age in years

**PL/SQL Procedure:**

DELIMITER //

CREATE FUNCTION CalculateAge(p\_DOB DATE) RETURNS INT

DETERMINISTIC

BEGIN

DECLARE v\_Age INT;

SET v\_Age = TIMESTAMPDIFF(YEAR, p\_DOB, CURDATE());

RETURN v\_Age;

END;

//

DELIMITER ;

**To test this function:**

SELECT CalculateAge('1985-05-15') AS Age;

**Scenario 2:** The bank needs to compute the monthly installment for a loan.

**Question:** Write a function **CalculateMonthlyInstallment** that takes the loan amount, interest rate, and loan duration in years as input and returns the monthly installment amount.

**PL/SQL Procedure:**

DELIMITER //

CREATE FUNCTION CalculateMonthlyInstallment(p\_LoanAmount DECIMAL(10, 2), p\_InterestRate DECIMAL(5, 2), p\_DurationYears INT) RETURNS DECIMAL(10, 2)

DETERMINISTIC

BEGIN

DECLARE v\_MonthlyRate DECIMAL(5, 4);

DECLARE v\_NumberOfPayments INT;

DECLARE v\_MonthlyInstallment DECIMAL(10, 2);

SET v\_MonthlyRate = p\_InterestRate / 100 / 12;

SET v\_NumberOfPayments = p\_DurationYears \* 12;

IF v\_MonthlyRate = 0 THEN

SET v\_MonthlyInstallment = p\_LoanAmount / v\_NumberOfPayments;

ELSE

SET v\_MonthlyInstallment = p\_LoanAmount \* v\_MonthlyRate / (1 - POWER(1 + v\_MonthlyRate, -v\_NumberOfPayments));

END IF;

RETURN v\_MonthlyInstallment;

END;

//

DELIMITER ;

**To test this function:**

SELECT CalculateMonthlyInstallment(5000, 5, 5) AS MonthlyInstallment;

**Scenario 3:** Check if a customer has sufficient balance before making a transaction.

**Question:** Write a function **HasSufficientBalance** that takes an account ID and an amount as input and returns a boolean indicating whether the account has at least the specified amount

**PL/SQL Procedure:**

DELIMITER //

CREATE FUNCTION HasSufficientBalance(p\_AccountID INT, p\_Amount DECIMAL(10, 2)) RETURNS BOOLEAN

READS SQL DATA

BEGIN

DECLARE v\_Balance DECIMAL(10, 2);

SELECT Balance INTO v\_Balance FROM Accounts WHERE AccountID = p\_AccountID;

RETURN v\_Balance >= p\_Amount;

END;

//

DELIMITER ;

**To test this function:**

SELECT HasSufficientBalance(1, 200) AS Sufficient;

SELECT HasSufficientBalance(1, 2000) AS Sufficient;