**Exercise 4 – Functions**

**Scenario 1: Calculate the Age of Customers:**

CREATE OR REPLACE FUNCTION CalculateAge (

p\_DOB DATE

) RETURN NUMBER

IS

v\_Age NUMBER;

BEGIN

-- Calculate age by subtracting the year of birth from the current year

v\_Age := TRUNC(MONTHS\_BETWEEN(SYSDATE, p\_DOB) / 12);

RETURN v\_Age;

END CalculateAge;

/

**Scenario 2: Compute the Monthly Installment for a Loan:**

CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment (

p\_LoanAmount NUMBER,

p\_InterestRate NUMBER,

p\_LoanDurationYears NUMBER

) RETURN NUMBER

IS

v\_MonthlyInstallment NUMBER;

v\_MonthlyInterestRate NUMBER;

v\_NumberOfPayments NUMBER;

BEGIN

-- Convert annual interest rate to a monthly rate

v\_MonthlyInterestRate := p\_InterestRate / 12 / 100;

-- Calculate the total number of payments

v\_NumberOfPayments := p\_LoanDurationYears \* 12;

-- Calculate the monthly installment using the formula for EMI

IF v\_MonthlyInterestRate > 0 THEN

v\_MonthlyInstallment := p\_LoanAmount \* v\_MonthlyInterestRate \*

POWER(1 + v\_MonthlyInterestRate, v\_NumberOfPayments) /

(POWER(1 + v\_MonthlyInterestRate, v\_NumberOfPayments) - 1);

ELSE

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

END IF;

RETURN v\_MonthlyInstallment;

END CalculateMonthlyInstallment;

/

**Scenario 3: Check If a Customer Has Sufficient Balance:**

CREATE OR REPLACE FUNCTION HasSufficientBalance (

p\_AccountID NUMBER,

p\_Amount NUMBER

) RETURN BOOLEAN

IS

v\_Balance NUMBER;

BEGIN

-- Retrieve the current balance of the account

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

-- Check if the balance is sufficient

IF v\_Balance >= p\_Amount THEN

RETURN TRUE;

ELSE

RETURN FALSE;

END IF;

END HasSufficientBalance;

/