Exercise 4: Functions

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

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

CREATE OR REPLACE FUNCTION CalculateAge (p\_dob DATE)

RETURN NUMBER

IS

v\_age NUMBER;

BEGIN

v\_age := TRUNC(MONTHS\_BETWEEN(SYSDATE, p\_dob) / 12);

RETURN v\_age;

END;

/

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

**O 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.**

CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment (

p\_loan\_amount NUMBER,

p\_interest\_rate NUMBER,

p\_loan\_duration NUMBER

)

RETURN NUMBER

IS

v\_monthly\_interest NUMBER;

v\_installment NUMBER;

BEGIN

v\_monthly\_interest := p\_interest\_rate / 12 / 100;

v\_installment := (p\_loan\_amount \* v\_monthly\_interest) / (1 - POWER(1 + v\_monthly\_interest, -p\_loan\_duration \* 12));

RETURN v\_installment;

END;

/

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

**O 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.**

CREATE OR REPLACE FUNCTION HasSufficientBalance (

p\_account\_id NUMBER,

p\_amount NUMBER

)

RETURN BOOLEAN

IS

v\_balance NUMBER;

BEGIN

SELECT Balance INTO v\_balance FROM Accounts WHERE AccountID = p\_account\_id;

RETURN v\_balance >= p\_amount;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN FALSE;

END;

/