**Exercise 4: Functions**

**Scenario 1:**

CREATE OR REPLACE FUNCTION CalculateAge(

p\_dob IN DATE

) RETURN NUMBER IS

v\_age NUMBER;

BEGIN

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

RETURN v\_age;

END;

/

**Scenario 2:**

CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment(

p\_loanAmount IN NUMBER,

p\_annualInterestRate IN NUMBER,

p\_durationYears IN NUMBER

) RETURN NUMBER IS

v\_monthlyRate NUMBER;

v\_months NUMBER;

v\_emi NUMBER;

BEGIN

v\_monthlyRate := p\_annualInterestRate / 12 / 100;

v\_months := p\_durationYears \* 12;

v\_emi := (p\_loanAmount \* v\_monthlyRate \* POWER(1 + v\_monthlyRate, v\_months)) /

(POWER(1 + v\_monthlyRate, v\_months) - 1);

RETURN ROUND(v\_emi, 2);

END;

/

**Scenario 3:**

CREATE OR REPLACE FUNCTION HasSufficientBalance(

p\_accountID IN NUMBER,

p\_amount IN NUMBER

) RETURN BOOLEAN IS

v\_balance NUMBER;

BEGIN

SELECT Balance INTO v\_balance

FROM Accounts

WHERE AccountID = p\_accountID;

RETURN v\_balance >= p\_amount;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN FALSE;

WHEN OTHERS THEN

RETURN FALSE;

END;

/