**Exercise 4: Functions**

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

CREATE OR REPLACE FUNCTION CalculateAge(

p\_dob IN DATE

) RETURN NUMBER IS

v\_age NUMBER;

BEGIN

-- Calculate age in years

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

RETURN v\_age;

END CalculateAge;

/

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

CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment(

p\_loan\_amount IN NUMBER,

p\_annual\_interest\_rate IN NUMBER,

p\_loan\_duration\_years IN NUMBER

) RETURN NUMBER IS

v\_monthly\_rate NUMBER;

v\_total\_months NUMBER;

v\_monthly\_installment NUMBER;

BEGIN

-- Calculate monthly interest rate

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

-- Calculate total number of payments

v\_total\_months := p\_loan\_duration\_years \* 12;

-- Calculate monthly installment using the formula for an annuity

IF v\_monthly\_rate > 0 THEN

v\_monthly\_installment := p\_loan\_amount \* (v\_monthly\_rate \* POWER(1 + v\_monthly\_rate, v\_total\_months)) / (POWER(1 + v\_monthly\_rate, v\_total\_months) - 1);

ELSE

-- If interest rate is 0%, the formula is simply the loan amount divided by the number of months

v\_monthly\_installment := p\_loan\_amount / v\_total\_months;

END IF;

RETURN v\_monthly\_installment;

END CalculateMonthlyInstallment;

/

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

CREATE OR REPLACE FUNCTION HasSufficientBalance(

p\_account\_id IN NUMBER,

p\_amount IN NUMBER

) RETURN BOOLEAN IS

v\_balance NUMBER;

BEGIN

-- Fetch the current balance of the account

SELECT Balance INTO v\_balance

FROM Accounts

WHERE AccountID = p\_account\_id;

-- Check if the balance is sufficient

IF v\_balance >= p\_amount THEN

RETURN TRUE;

ELSE

RETURN FALSE;

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

-- Handle case where the account ID does not exist

RETURN FALSE;

WHEN OTHERS THEN

-- Handle other exceptions

DBMS\_OUTPUT.PUT\_LINE('Error occurred: ' || SQLERRM);

RETURN FALSE;

END HasSufficientBalance;

/