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.

CREATE OR REPLACE FUNCTION CalculateAge(

p\_date\_of\_birth DATE

) RETURN NUMBER IS

v\_age NUMBER;

BEGIN

-- Calculate age based on the current date and date of birth

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

RETURN v\_age;

END;

/

**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.

CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment(

p\_loan\_amount NUMBER,

p\_interest\_rate NUMBER,

p\_loan\_duration\_years NUMBER

) RETURN NUMBER IS

v\_monthly\_interest\_rate NUMBER;

v\_total\_months NUMBER;

v\_monthly\_installment NUMBER;

BEGIN

-- Calculate monthly interest rate

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

-- Calculate total number of months

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

-- Calculate the monthly installment using the formula for an annuity

IF v\_monthly\_interest\_rate > 0 THEN

v\_monthly\_installment := p\_loan\_amount \* v\_monthly\_interest\_rate /

(1 - POWER(1 + v\_monthly\_interest\_rate, -v\_total\_months));

ELSE

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

END IF;

RETURN v\_monthly\_installment;

END;

/

**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.

CREATE OR REPLACE FUNCTION HasSufficientBalance(

p\_account\_id NUMBER,

p\_amount NUMBER

) RETURN BOOLEAN IS

v\_balance NUMBER;

BEGIN

-- Retrieve the current balance for the account

SELECT balance INTO v\_balance

FROM accounts

WHERE account\_id = 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 any other exceptions

RETURN FALSE;

END;

/