**EXERCISE 4:**

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 in years

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

RETURN v\_age;

END;

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\_annual\_interest\_rate NUMBER,

p\_loan\_duration\_years NUMBER

) RETURN NUMBER IS

v\_monthly\_installment NUMBER;

v\_monthly\_interest\_rate NUMBER;

v\_total\_months NUMBER;

BEGIN

-- Convert annual interest rate to monthly and duration in months

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

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

-- Calculate monthly installment

IF v\_monthly\_interest\_rate > 0 THEN

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

ELSE

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

END IF;

RETURN v\_monthly\_installment;

END;

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

-- Get the account balance

SELECT balance INTO v\_balance FROM accounts WHERE account\_id = p\_account\_id;

-- Check if balance is sufficient

IF v\_balance >= p\_amount THEN

RETURN TRUE;

ELSE

RETURN FALSE;

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

-- Account ID not found

RETURN FALSE;

END;