

## 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_dob DATE)
RETURN NUMBER
IS
    v_age NUMBER;
BEGIN
    SELECT FLOOR(MONTHS_BETWEEN(SYSDATE, p_dob) / 12) INTO v_age FROM dual;
    RETURN v_age;
EXCEPTION
    WHEN OTHERS THEN
        RETURN NULL;
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_annual_interest_rate NUMBER,  
    p_loan_duration_years NUMBER  
)  
RETURN NUMBER  
IS  
    v_monthly_interest_rate NUMBER;  
    v_number_of_months NUMBER;  
    v_monthly_installment NUMBER;  
BEGIN  
    v_monthly_interest_rate := p_annual_interest_rate / 12 / 100;  
    v_number_of_months := p_loan_duration_years * 12;  
  
    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_number_of_months));  
    ELSE  
        v_monthly_installment := p_loan_amount / v_number_of_months;  
    END IF;  
  
    RETURN v_monthly_installment;  
EXCEPTION  
    WHEN OTHERS THEN  
        RETURN NULL;  
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  
    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;  
    WHEN OTHERS THEN  
        RETURN FALSE;  
END;
```