**Exercise 4: Functions**

**Scenario 1:**

**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

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

  RETURN v\_age;

END;

/DECLARE

  v\_age NUMBER;

BEGIN

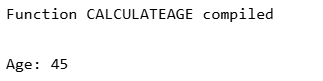
  v\_age := CalculateAge(TO\_DATE('1980-05-10', 'YYYY-MM-DD'));

  DBMS\_OUTPUT.PUT\_LINE('Age: ' || v\_age);

END;

/

**Output**



**Scenario 2:**

**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\_rate NUMBER,

  p\_years NUMBER

)

RETURN NUMBER

IS

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

  v\_months NUMBER := p\_years \* 12;

  v\_emi NUMBER;

BEGIN

  IF v\_monthly\_rate = 0 THEN

    v\_emi := p\_loan\_amount / v\_months;

  ELSE

    v\_emi := p\_loan\_amount \* v\_monthly\_rate \* POWER(1 + v\_monthly\_rate, v\_months) /

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

  END IF;

  RETURN ROUND(v\_emi, 2);

END;

/

DECLARE

  v\_emi NUMBER;

BEGIN

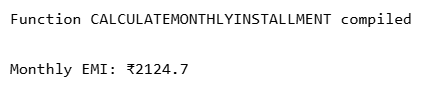
  v\_emi := CalculateMonthlyInstallment(100000, 10, 5); -- ₹100,000, 10% annual, 5 years

  DBMS\_OUTPUT.PUT\_LINE('Monthly EMI: ₹' || v\_emi);

END;

/

**Output**



**Scenario 3:**

**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;

END;

/

DECLARE

  v\_result BOOLEAN;

BEGIN

  v\_result := HasSufficientBalance(1, 5000);

  IF v\_result THEN

    DBMS\_OUTPUT.PUT\_LINE('Account 1 has sufficient balance.');

  ELSE

    DBMS\_OUTPUT.PUT\_LINE('Account 1 does NOT have sufficient balance.');

  END IF;

END;

/

**Output**

