**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.

**CODE:**

CREATE OR REPLACE FUNCTION CalculateAge(dob DATE)

RETURN NUMBER

IS

age NUMBER;

BEGIN

age := FLOOR(MONTHS\_BETWEEN(SYSDATE, dob) / 12);

RETURN age;

END;

/

DECLARE

age NUMBER;

BEGIN

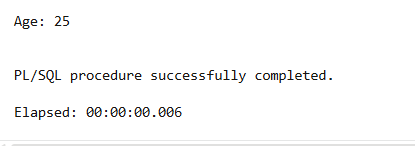
age := CalculateAge(DATE '2000-06-29');

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

END;

/

**OUTPUT:**

****

**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.

**CODE:**

CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment(

loan\_amount NUMBER,

annual\_rate NUMBER,

duration\_years NUMBER

)

RETURN NUMBER

IS

monthly\_rate NUMBER;

num\_months NUMBER;

installment NUMBER;

BEGIN

monthly\_rate := annual\_rate / 12 / 100;

num\_months := duration\_years \* 12;

installment := loan\_amount \* monthly\_rate /

(1 - POWER(1 + monthly\_rate, -num\_months));

RETURN ROUND(installment, 2);

END;

/

DECLARE

emi NUMBER;

BEGIN

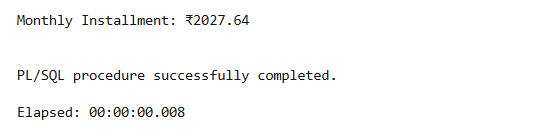
emi := CalculateMonthlyInstallment(100000, 8, 5);

DBMS\_OUTPUT.PUT\_LINE('Monthly Installment: ₹' || emi);

END;

/

**OUTPUT:**



**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.

**CODE:**

CREATE OR REPLACE FUNCTION HasSufficientBalance(

acc\_id NUMBER,

amount NUMBER

)

RETURN BOOLEAN

IS

bal NUMBER;

BEGIN

SELECT balance INTO bal FROM accounts WHERE accountid = acc\_id;

IF bal >= amount THEN

RETURN TRUE;

ELSE

RETURN FALSE;

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN FALSE;

END;

/

DECLARE

result BOOLEAN;

BEGIN

result := HasSufficientBalance(101, 500);

IF result THEN

DBMS\_OUTPUT.PUT\_LINE('Sufficient balance.');

ELSE

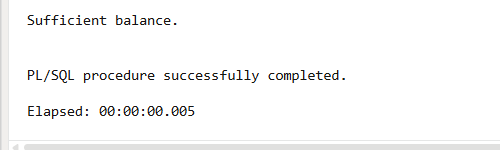
DBMS\_OUTPUT.PUT\_LINE('Insufficient balance.');

END IF;

END;

/

**OUTPUT:**

****