**Exercise 4: Functions**

**Scenario 1: Calculate the age of customers for eligibility checks.**

**o 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 (

dob IN DATE

) RETURN NUMBER IS

age NUMBER;

BEGIN

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

RETURN age;

END;

/

SELECT Name, DOB, CalculateAge(DOB) AS Age

FROM Customers;

**OUTPUT:**

A white background with black text

AI-generated content may be incorrect.

**Scenario 2: The bank needs to compute the monthly installment for a loan.**

**o 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 (

loan\_amount IN NUMBER,

annual\_rate IN NUMBER,

duration\_years IN NUMBER

) RETURN NUMBER IS

monthly\_rate NUMBER;

total\_months NUMBER;

emi NUMBER;

BEGIN

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

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

emi := loan\_amount \* monthly\_rate / (1 - POWER(1 + monthly\_rate, -total\_months));

RETURN ROUND(emi, 2);

END;

SELECT CalculateMonthlyInstallment(100000, 12, 5) AS EMI FROM dual;

**OUPUT:**

A white background with black dots

AI-generated content may be incorrect.

**Scenario 3: Check if a customer has sufficient balance before making a transaction.**

**o 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 (

accid IN NUMBER,

amt IN NUMBER

) RETURN BOOLEAN IS

balanc NUMBER;

BEGIN

SELECT Balance INTO balanc

FROM Accounts

WHERE AccountID = accid;

IF balanc >= amt THEN

RETURN TRUE;

ELSE

RETURN FALSE;

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN FALSE;

WHEN OTHERS THEN

RETURN FALSE;

END;

/

SELECT \* from Accounts;

DECLARE

sufficient BOOLEAN;

BEGIN

sufficient := HasSufficientBalance(1, 1600);

IF sufficient THEN

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

ELSE

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

END IF;

END;

/

**OUPTUT:**

A white paper with black text

AI-generated content may be incorrect.