**Exercise 4: Functions**

**Scenario 1: Calculate the Age of Customers**

**Function: CalculateAge**

CREATE OR REPLACE FUNCTION CalculateAge (

p\_dob DATE

) RETURN NUMBER IS

v\_age NUMBER;

BEGIN

-- Calculate the age in years

v\_age := TRUNC((SYSDATE - p\_dob) / 365.25);

RETURN v\_age;

END CalculateAge;

/

**Scenario 2: Compute the Monthly Installment for a Loan**

**Function: CalculateMonthlyInstallment**

CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment (

p\_loan\_amount NUMBER,

p\_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

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

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

-- Calculate the monthly installment using the formula for annuity

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

RETURN v\_monthly\_installment;

END CalculateMonthlyInstallment;

/

**Scenario 3: Check if a Customer Has Sufficient Balance**

**Function: HasSufficientBalance**

CREATE OR REPLACE FUNCTION HasSufficientBalance (

p\_account\_id NUMBER,

p\_amount NUMBER

) RETURN BOOLEAN IS

v\_balance accounts.balance%TYPE;

BEGIN

-- Get the balance of the account

SELECT balance INTO v\_balance

FROM accounts

WHERE accountid = p\_account\_id;

-- Check if the balance is sufficient

IF v\_balance >= p\_amount THEN

RETURN TRUE;

ELSE

RETURN FALSE;

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN FALSE; -- Account not found, consider it as insufficient balance

END HasSufficientBalance;

/