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

CREATE OR REPLACE FUNCTION CalculateAge (

p\_date\_of\_birth IN DATE

) RETURN NUMBER IS

v\_age NUMBER;

BEGIN

-- Calculate age based on the difference between current date and date of birth

v\_age := FLOOR(MONTHS\_BETWEEN(SYSDATE, p\_date\_of\_birth) / 12);

RETURN v\_age;

END CalculateAge;

/

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

CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment (

p\_loan\_amount IN NUMBER,

p\_annual\_interest\_rate IN NUMBER,

p\_loan\_duration\_years IN NUMBER

) RETURN NUMBER IS

v\_monthly\_interest\_rate NUMBER;

v\_number\_of\_payments NUMBER;

v\_monthly\_installment NUMBER;

BEGIN

-- Convert annual interest rate to monthly and calculate number of payments

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

v\_number\_of\_payments := p\_loan\_duration\_years \* 12;

-- Calculate the monthly installment using the formula

IF v\_monthly\_interest\_rate > 0 THEN

v\_monthly\_installment := p\_loan\_amount \* (v\_monthly\_interest\_rate \* POWER(1 + v\_monthly\_interest\_rate, v\_number\_of\_payments)) /

(POWER(1 + v\_monthly\_interest\_rate, v\_number\_of\_payments) - 1);

ELSE

v\_monthly\_installment := p\_loan\_amount / v\_number\_of\_payments;

END IF;

RETURN v\_monthly\_installment;

END CalculateMonthlyInstallment;

/

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

CREATE OR REPLACE FUNCTION HasSufficientBalance (

p\_account\_id IN NUMBER,

p\_amount IN NUMBER

) RETURN BOOLEAN IS

v\_balance NUMBER;

BEGIN

-- Retrieve the current balance for the given account ID

SELECT balance INTO v\_balance

FROM accounts

WHERE account\_id = p\_account\_id;

-- Return TRUE if balance is sufficient, FALSE otherwise

RETURN v\_balance >= p\_amount;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

-- Handle the case where the account ID does not exist

RETURN FALSE;

END HasSufficientBalance;

/