Scenario 1: Calculate the Age of Customers

CREATE OR REPLACE FUNCTION CalculateAge(

p\_date\_of\_birth DATE

) RETURN NUMBER IS

v\_age NUMBER;

BEGIN

-- Calculate the age based on the date of birth

SELECT FLOOR(MONTHS\_BETWEEN(SYSDATE, p\_date\_of\_birth) / 12) INTO v\_age FROM dual;

RETURN v\_age;

END CalculateAge;

/

Scenario 2: Compute the Monthly Installment for a Loan

CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment(

p\_loan\_amount NUMBER,

p\_annual\_interest\_rate NUMBER,

p\_loan\_duration\_years NUMBER

) RETURN NUMBER IS

v\_monthly\_installment NUMBER;

v\_monthly\_interest\_rate NUMBER;

v\_total\_payments NUMBER;

BEGIN

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

v\_monthly\_interest\_rate := p\_annual\_interest\_rate / 1200;

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

-- Calculate the monthly installment using the formula for an amortizing loan

IF v\_monthly\_interest\_rate > 0 THEN

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

ELSE

v\_monthly\_installment := p\_loan\_amount / v\_total\_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 NUMBER,

p\_amount NUMBER

) RETURN BOOLEAN IS

v\_balance NUMBER;

BEGIN

-- Get the current balance of the specified account

SELECT balance INTO v\_balance

FROM accounts

WHERE account\_id = 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

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

RETURN FALSE;

END HasSufficientBalance;

/