--Scenario 1

CREATE OR REPLACE FUNCTION CalculateAge (

p\_DOB DATE

) RETURN NUMBER

IS

v\_Age NUMBER;

BEGIN

v\_Age := TRUNC(MONTHS\_BETWEEN(SYSDATE, p\_DOB) / 12);

RETURN v\_Age;

END CalculateAge;

/

--Scenario 2

CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment (

p\_LoanAmount NUMBER,

p\_InterestRate NUMBER,

p\_LoanDurationYears NUMBER

) RETURN NUMBER

IS

v\_MonthlyRate NUMBER;

v\_NumPayments NUMBER;

v\_MonthlyInstallment NUMBER;

BEGIN

v\_MonthlyRate := p\_InterestRate / 1200;

v\_NumPayments := p\_LoanDurationYears \* 12;

IF v\_MonthlyRate = 0 THEN

v\_MonthlyInstallment := p\_LoanAmount / v\_NumPayments;

ELSE

v\_MonthlyInstallment := p\_LoanAmount \* v\_MonthlyRate / (1 - POWER(1 + v\_MonthlyRate, -v\_NumPayments));

END IF;

RETURN v\_MonthlyInstallment;

END CalculateMonthlyInstallment;

/

--Scenario 3

CREATE OR REPLACE FUNCTION HasSufficientBalance (

p\_AccountID NUMBER,

p\_Amount NUMBER

) RETURN BOOLEAN

IS

v\_Balance NUMBER;

BEGIN

SELECT Balance INTO v\_Balance FROM Accounts WHERE AccountID = p\_AccountID;

IF v\_Balance >= p\_Amount THEN

RETURN TRUE;

ELSE

RETURN FALSE;

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN FALSE;

END HasSufficientBalance;

/