Scenario 1: CalculateAge

CREATE OR REPLACE FUNCTION CalculateAge(

p\_DOB DATE

) RETURN NUMBER IS

v\_Age NUMBER;

BEGIN

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

RETURN v\_Age;

END CalculateAge;

/

Scenario 2: CalculateMonthlyInstallment

CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment(

p\_LoanAmount NUMBER,

p\_InterestRate NUMBER,

p\_LoanDurationYears NUMBER

) RETURN NUMBER IS

v\_MonthlyInstallment NUMBER;

v\_MonthlyInterestRate NUMBER;

v\_TotalMonths NUMBER;

BEGIN

v\_MonthlyInterestRate := p\_InterestRate / 12 / 100;

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

v\_MonthlyInstallment := p\_LoanAmount \* v\_MonthlyInterestRate /

(1 - POWER(1 + v\_MonthlyInterestRate, -v\_TotalMonths));

RETURN v\_MonthlyInstallment;

END CalculateMonthlyInstallment;

/

Scenario 3: HasSufficientBalance

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;

END HasSufficientBalance;

/