Exercise 4:

-- Create Customers Table

CREATE TABLE Customers (

CustomerID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

DOB DATE,

Balance NUMBER,

LastModified DATE

);

-- Create Accounts Table

CREATE TABLE Accounts (

AccountID NUMBER PRIMARY KEY,

CustomerID NUMBER,

AccountType VARCHAR2(20),

Balance NUMBER,

LastModified DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

-- Insert Sample Data

INSERT INTO Customers VALUES (1, 'John Doe', TO\_DATE('1980-01-01','YYYY-MM-DD'), 1000, SYSDATE);

INSERT INTO Customers VALUES (2, 'Jane Smith', TO\_DATE('1960-05-10','YYYY-MM-DD'), 15000, SYSDATE);

INSERT INTO Accounts VALUES (101, 1, 'Savings', 1000, SYSDATE);

INSERT INTO Accounts VALUES (102, 2, 'Checking', 15000, SYSDATE);

-- Scenario 1: CalculateAge (DOB → Age in Years)

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;

/

-- Scenario 2 : CalculateMonthlyInstallment (Loan EMI Formula)

CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment (

p\_loan\_amount NUMBER,

p\_interest\_rate NUMBER, -- Annual interest in %

p\_duration\_years NUMBER

) RETURN NUMBER IS

v\_monthly\_rate NUMBER;

v\_months NUMBER;

v\_emi NUMBER;

BEGIN

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

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

IF v\_monthly\_rate = 0 THEN

v\_emi := p\_loan\_amount / v\_months;

ELSE

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

END IF;

RETURN ROUND(v\_emi, 2);

END;

/

-- Scenario3 : HasSufficientBalance (Returns TRUE/FALSE by Account)

CREATE OR REPLACE FUNCTION HasSufficientBalance (

p\_account\_id IN NUMBER,

p\_required\_amount IN NUMBER

) RETURN BOOLEAN IS

v\_balance NUMBER;

BEGIN

SELECT Balance INTO v\_balance

FROM Accounts

WHERE AccountID = p\_account\_id;

RETURN v\_balance >= p\_required\_amount;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN FALSE;

END;

/

SELECT Name, CalculateAge(DOB) AS Age

FROM Customers;

SELECT CalculateMonthlyInstallment(100000, 7.5, 5) AS MonthlyEMI

FROM dual;

DECLARE

result BOOLEAN;

BEGIN

result := HasSufficientBalance(101, 800);

IF result THEN

DBMS\_OUTPUT.PUT\_LINE('Sufficient balance.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Insufficient balance.');

END IF;

END;

/