**Exercise 4: Functions**

**Scenario 1: Calculate Age**

**Function**:

CREATE OR REPLACE FUNCTION CalculateAge (p\_dob DATE) RETURN NUMBER AS

BEGIN

RETURN TRUNC((SYSDATE - p\_dob) / 365);

END;

/

**Verification Query**:

SELECT Name, CalculateAge(DOB) AS Age FROM Customers;

**Sample Output**:

NAME | AGE

----------------

John Doe | 39

Jane Smith | 33

**Scenario 2: Calculate Monthly Installment**

**Function**:

CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment (

p\_loan\_amount NUMBER,

p\_interest\_rate NUMBER,

p\_loan\_duration NUMBER

) RETURN NUMBER AS

v\_monthly\_rate NUMBER;

v\_installment NUMBER;

BEGIN

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

v\_installment := p\_loan\_amount \* v\_monthly\_rate / (1 - POWER(1 + v\_monthly\_rate, -p\_loan\_duration \* 12));

RETURN v\_installment;

END;

/

**Verification Query**:

SELECT CalculateMonthlyInstallment(5000, 5, 5) AS Installment FROM DUAL;

**Sample Output**:

INSTALLMENT

------------

94.26

**Scenario 3: Check Sufficient Balance**

**Function**:

CREATE OR REPLACE FUNCTION HasSufficientBalance (

p\_account\_id NUMBER,

p\_amount NUMBER

) RETURN NUMBER AS

v\_balance NUMBER;

BEGIN

SELECT Balance INTO v\_balance FROM Accounts WHERE AccountID = p\_account\_id;

IF v\_balance >= p\_amount THEN

RETURN 1; -- TRUE

ELSE

RETURN 0; -- FALSE

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 0; -- FALSE

END;

/

**Verification Query**:

SELECT HasSufficientBalance(1, 500) AS Sufficient FROM DUAL;

**Sample Output**:

SUFFICIENT

-----------

TRUE