**WEEK-2**

**MODULE -3**

**PL/SQL PROGRAMMING**

**SUPERSET ID:6407550**

**Exercise:4-Functions**

--Insert an account with specific balance for test

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)

VALUES (30, 3, 'Savings', 5000, SYSDATE);

COMMIT;

**Scenario:1**

CREATE OR REPLACE FUNCTION CalculateAge(p\_DOB IN DATE)

RETURN NUMBER IS

v\_Age NUMBER;

BEGIN

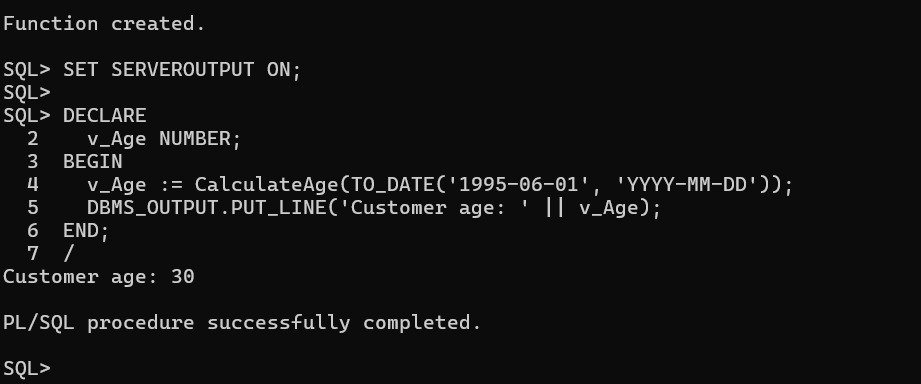
v\_Age := FLOOR(MONTHS\_BETWEEN(SYSDATE, p\_DOB) / 12);

RETURN v\_Age;

END;

/

**OUTPUT:**

****

**Scenario:2**

CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment(

p\_LoanAmount IN NUMBER,

p\_InterestRate IN NUMBER,

p\_Years IN NUMBER

) RETURN NUMBER IS

r NUMBER := p\_InterestRate / 12 / 100;

n NUMBER := p\_Years \* 12;

EMI NUMBER;

BEGIN

IF r = 0 THEN

EMI := p\_LoanAmount / n;

ELSE

EMI := p\_LoanAmount \* r \* POWER(1 + r, n) / (POWER(1 + r, n) - 1);

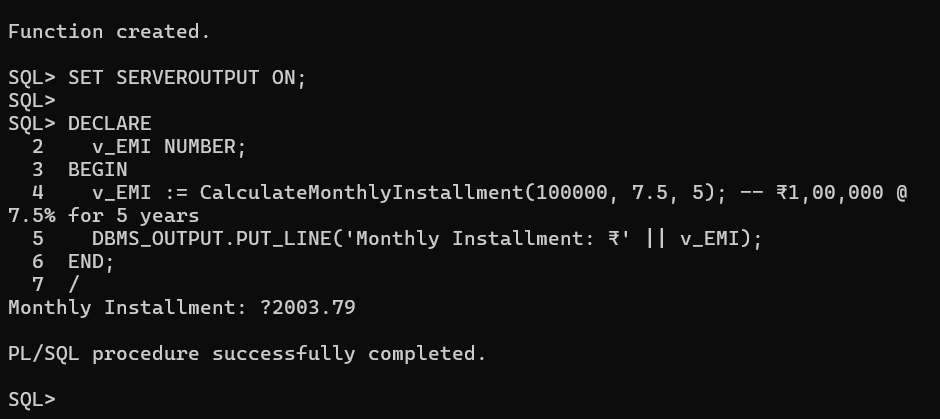
END IF;

RETURN ROUND(EMI, 2);

END;

/

**OUTPUT:**

****

**Scenario:3**

CREATE OR REPLACE FUNCTION HasSufficientBalance(

p\_AccountID IN NUMBER,

p\_Amount IN NUMBER

) RETURN BOOLEAN IS

v\_Balance NUMBER;

BEGIN

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

RETURN v\_Balance >= p\_Amount;

EXCEPTION

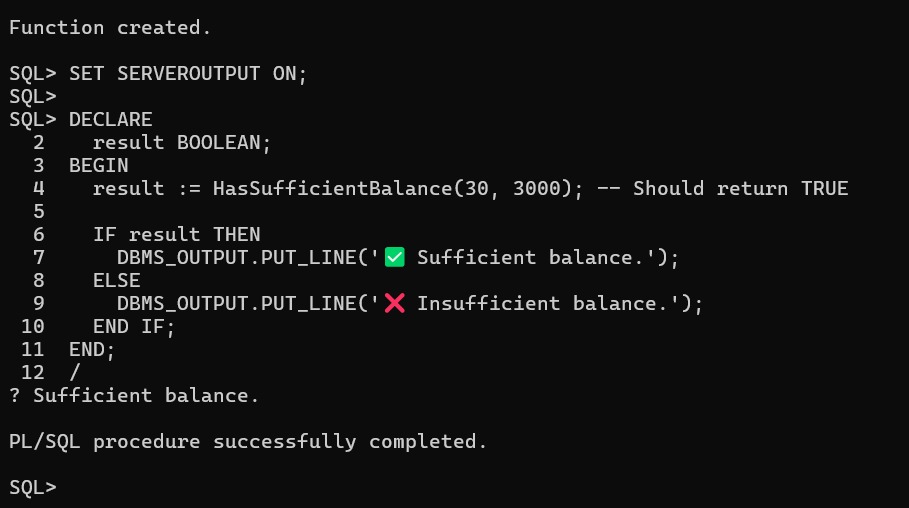
WHEN NO\_DATA\_FOUND THEN

RETURN FALSE;

END;

**/**

**OUTPUT:**

****