PL/SQL Programming

SuperSet ID:6412063

Exercise 4: Functions

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;

/

SET SERVEROUTPUT ON;

DECLARE

age NUMBER;

BEGIN

age := CalculateAge(TO\_DATE('1995-04-12', 'YYYY-MM-DD'));

DBMS\_OUTPUT.PUT\_LINE('Age: ' || age);

END;

/

Output:

A black screen with white text

Description automatically generated

Scenario 2**:**

CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment (

p\_amount IN NUMBER,

p\_annual\_rate IN NUMBER,

p\_years IN NUMBER

) RETURN NUMBER IS

v\_monthly\_rate NUMBER;

v\_months NUMBER;

v\_emi NUMBER;

BEGIN

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

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

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

(POWER(1 + v\_monthly\_rate, v\_months) - 1);

RETURN ROUND(v\_emi, 2);

END;

/

Output:

**A black screen with white text

Description automatically generated**

Scenario 3**:**

CREATE OR REPLACE FUNCTION HasSufficientBalance (

p\_account\_id IN NUMBER,

p\_required\_amount IN NUMBER

) RETURN NUMBER IS

v\_balance NUMBER;

BEGIN

SELECT Balance INTO v\_balance

FROM Accounts

WHERE AccountID = p\_account\_id;

IF v\_balance >= p\_required\_amount THEN

RETURN 1; -- TRUE

ELSE

RETURN 0; -- FALSE

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 0; -- Account not found, treat as insufficient

END;

/

SET SERVEROUTPUT ON;

DECLARE

result NUMBER;

BEGIN

result := HasSufficientBalance(101, 5000);

IF result = 1 THEN

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

ELSE

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

END IF;

END;

/

Output:

A black screen with white text

Description automatically generated