Scenario 1: CalculateAge – Return Age from DOB

CREATE OR REPLACE FUNCTION CalculateAge(

p\_dob IN DATE

) RETURN NUMBER

IS

v\_age NUMBER;

BEGIN

v\_age := TRUNC(MONTHS\_BETWEEN(SYSDATE, p\_dob) / 12);

RETURN v\_age;

END;

Output:

**Example Execution:**

**SELECT CalculateAge(TO\_DATE('1985-05-15', 'YYYY-MM-DD')) AS Age FROM dual;**

**AGE**

**---**

**39**

Scenario 2: CalculateMonthlyInstallment – Loan EMI Formula

CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment(

p\_loan\_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\_loan\_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;

**Example Execution:**

SELECT CalculateMonthlyInstallment(50000, 7, 5) AS EMI FROM dual;

EMI

-----

990.06

Scenario 3: HasSufficientBalance – Boolean Check

CREATE OR REPLACE FUNCTION HasSufficientBalance(

p\_account\_id IN NUMBER,

p\_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\_amount;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN FALSE;

END;

Output:

Example Execution:

DECLARE

v\_result BOOLEAN;

BEGIN

v\_result := HasSufficientBalance(1, 1000);

IF v\_result THEN

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

ELSE

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

END IF;

END;

Output:

Sufficient Balance