

- **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;  
/
```

@Usage Example

```
SELECT CalculateAge(TO_DATE('1990-07-20', 'YYYY-MM-DD')) AS Age FROM dual;
```

- **Scenario 2**

```
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 := p_annual_rate / 12 / 100;  
    v_months       NUMBER := p_years * 12;  
    v_installment  NUMBER;  
BEGIN  
    IF v_monthly_rate = 0 THEN  
        v_installment := p_loan_amount / v_months;  
    ELSE  
        v_installment := p_loan_amount * v_monthly_rate * POWER(1 + v_monthly_rate, v_months) /  
            (POWER(1 + v_monthly_rate, v_months) - 1);  
    END IF;  
  
    RETURN ROUND(v_installment, 2);  
END;  
/
```

@Usage Example

```
SELECT CalculateMonthlyInstallment(5000, 5, 5) AS EMI FROM dual;
```

- **Scenario 3**

```
CREATE OR REPLACE FUNCTION HasSufficientBalance (  
    p_account_id IN NUMBER,  
    p_amount    IN NUMBER  
) RETURN BOOLEAN  
IS  
    v_balance Accounts.Balance%TYPE;  
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;  
    WHEN OTHERS THEN  
        RETURN FALSE;  
END;  
/
```

@Usage Example

```
DECLARE  
    result BOOLEAN;  
BEGIN  
    result := HasSufficientBalance(1, 500);  
    IF result THEN  
        DBMS_OUTPUT.PUT_LINE('Sufficient balance');  
    ELSE  
        DBMS_OUTPUT.PUT_LINE('Insufficient balance');  
    END IF;  
END;  
/
```