

```
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 CalculateAge;
```

```
CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment (
```

```
    p_loan_amount IN NUMBER,
```

```
    p_interest_rate IN NUMBER,
```

```
    p_duration_years IN NUMBER
```

```
) RETURN NUMBER IS
```

```
    v_monthly_rate NUMBER;
```

```
    v_num_payments NUMBER;
```

```
    v_monthly_installment NUMBER;
```

```
BEGIN
```

```
    v_monthly_rate := p_interest_rate / 12 / 100;
```

```
    v_num_payments := p_duration_years * 12;
```

```
    IF v_monthly_rate = 0 THEN
```

```
        v_monthly_installment := p_loan_amount / v_num_payments;
```

```
    ELSE
```

```
        v_monthly_installment := p_loan_amount * v_monthly_rate / (1 - POWER(1 + v_monthly_rate, -  
v_num_payments));
```

```
    END IF;
```

```
    RETURN v_monthly_installment;
```

```
END CalculateMonthlyInstallment;
```

```
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;

    IF v_balance >= p_amount THEN

        RETURN TRUE;

    ELSE

        RETURN FALSE;

    END IF;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        RETURN FALSE;

    WHEN OTHERS THEN

        RETURN FALSE;

END HasSufficientBalance;
```