**Exercise 4: Functions**

CREATE OR REPLACE FUNCTION CalculateAge(p\_dob DATE)

RETURN NUMBER

IS

    v\_age NUMBER;

BEGIN

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

    RETURN v\_age;

END;

BEGIN

    DBMS\_OUTPUT.PUT\_LINE('Age: '||CalculateAge(DATE '1995-06-29'));

END;

CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment(p\_amount IN NUMBER,p\_annual\_rate IN NUMBER,p\_duration\_years IN NUMBER) RETURN NUMBER

IS

    monthly\_rate NUMBER:=p\_annual\_rate/12/100;

    months NUMBER:=p\_duration\_years\*12;

    emi NUMBER;

BEGIN

    IF monthly\_rate=0 THEN

        emi:=p\_amount/months;

    ELSE

        emi:=(p\_amount\*monthly\_rate\*POWER(1+monthly\_rate,months))/(POWER(1+monthly\_rate,months)-1);

    END IF;

    RETURN ROUND(emi,2);

END;

BEGIN

    DBMS\_OUTPUT.PUT\_LINE('Monthly EMI: ₹'||CalculateMonthlyInstallment(100000,10,2));

END;

CREATE TABLE Accounts(AccountID NUMBER PRIMARY KEY,HolderName VARCHAR2(100),Balance NUMBER);

INSERT INTO Accounts VALUES(1,'John',8000);

INSERT INTO Accounts VALUES(2,'Jane',4000);

COMMIT;

CREATE OR REPLACE FUNCTION HasSufficientBalance(p\_account\_id IN NUMBER,p\_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\_amount THEN

        RETURN 1;

    ELSE

        RETURN 0;

    END IF;

EXCEPTION

    WHEN NO\_DATA\_FOUND THEN

        RETURN 0;

    WHEN OTHERS THEN

        RETURN 0;

END;

DECLARE

    result NUMBER;

BEGIN

    result:=HasSufficientBalance(1,5000);

    IF result=1 THEN

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

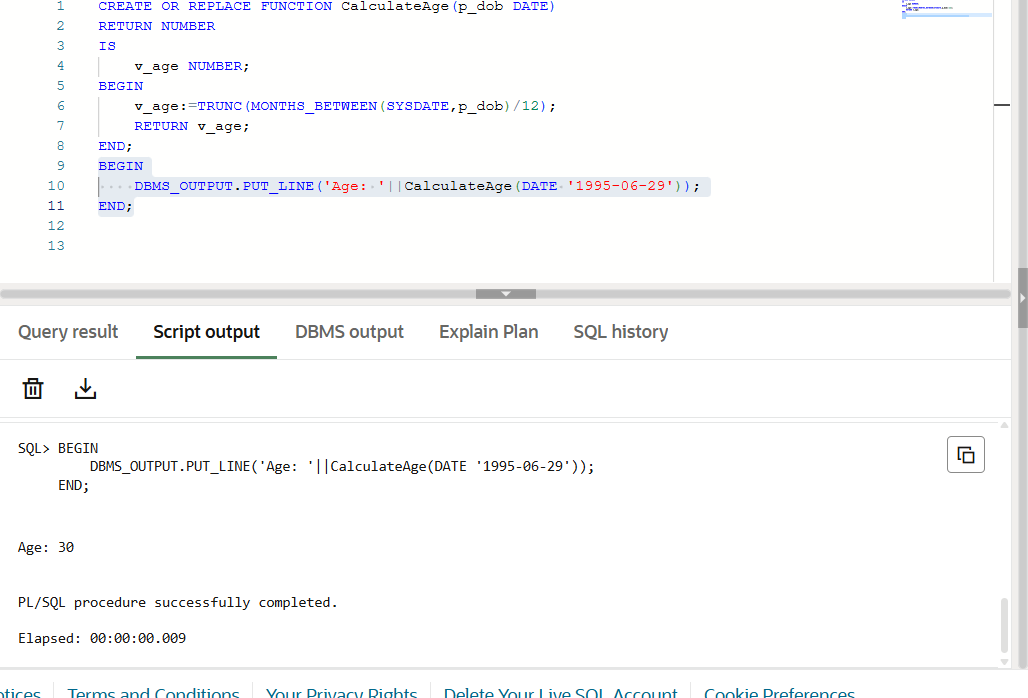
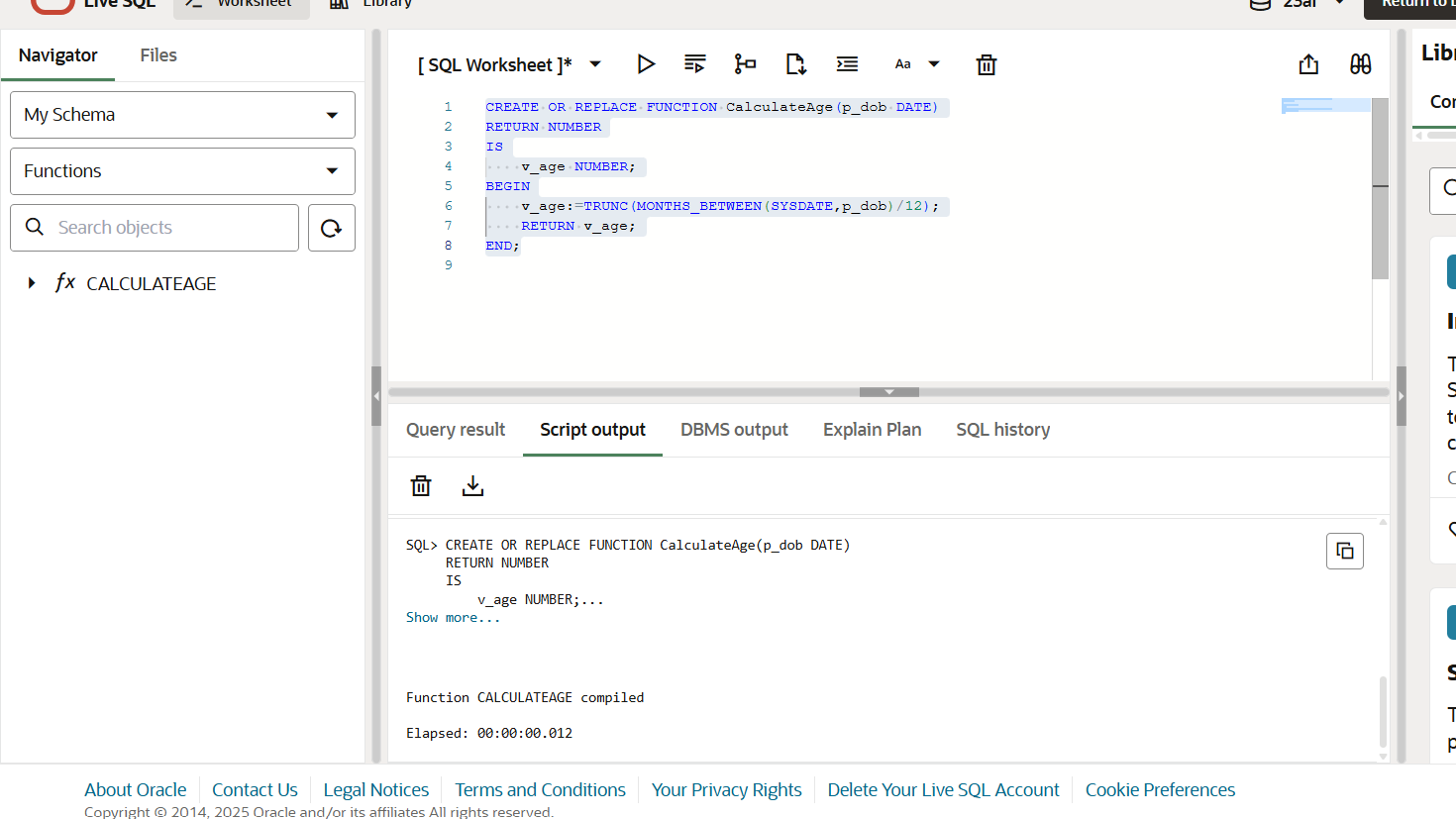
    ELSE

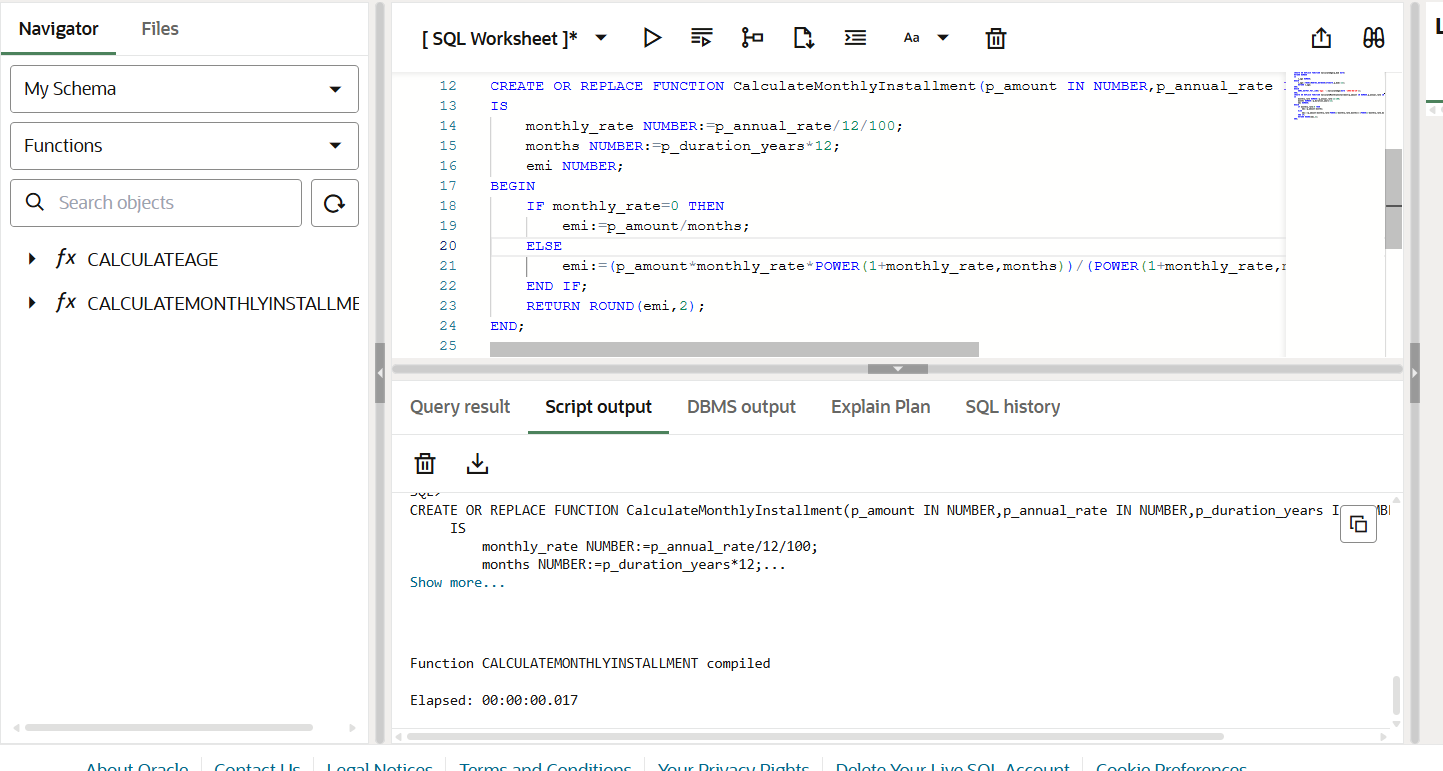
        DBMS\_OUTPUT.PUT\_LINE('Insufficient balance.');

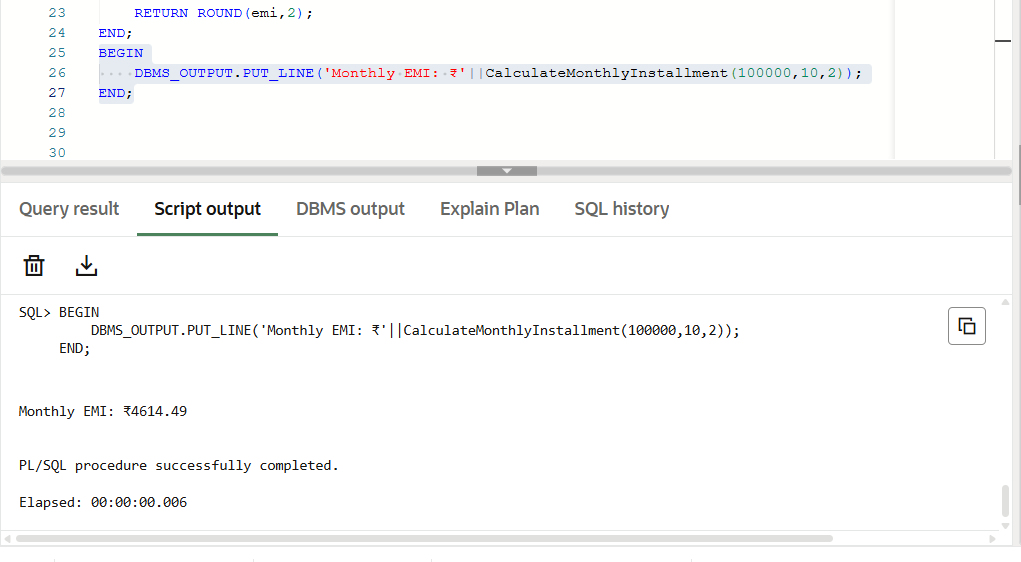
    END IF;

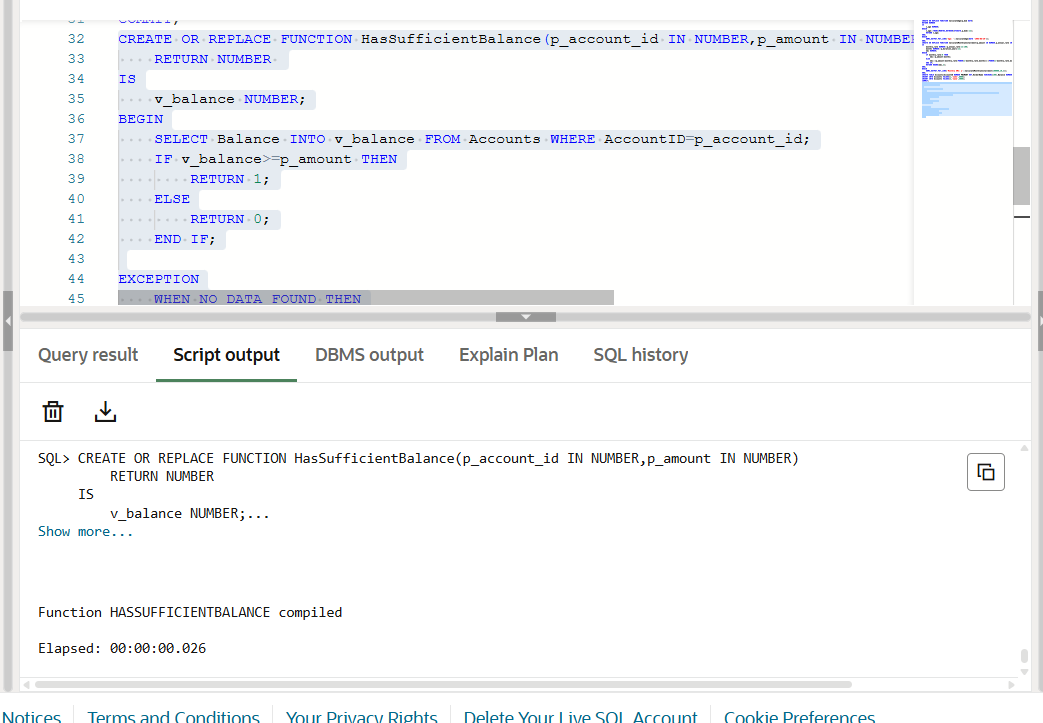
END;

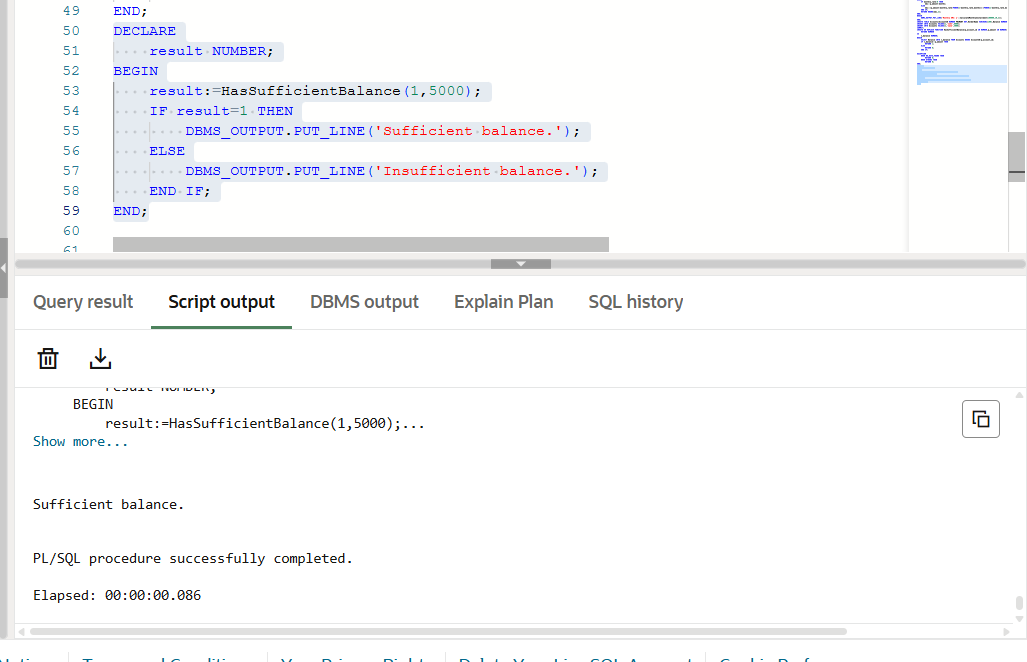
**Output:**

****

****

****

****

****