Exercise 3: Stored Procedures

Scenario 1: The bank needs to process monthly interest for all savings accounts.

o Question: Write a stored procedure ProcessMonthlyInterest that calculates and updates the balance of all savings accounts by applying an interest rate of 1% to the current balance.

Scenario 2: The bank wants to implement a bonus scheme for employees based on their performance.

o Question: Write a stored procedure UpdateEmployeeBonus that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter.

Scenario 3: Customers should be able to transfer funds between their accounts.

o Question: Write a stored procedure TransferFunds that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.

**SCENARIO 1:**

we need a table in which we have data about all saving accounts present in that bank

CREATE TABLE SAVINGS\_ACCOUNTS (

ACCT\_ID NUMBER,

BALANCE NUMBER

);

Then procedure:

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

FOR acc IN (SELECT \* FROM SAVINGS\_ACCOUNTS) LOOP

UPDATE SAVINGS\_ACCOUNTS

SET BALANCE = BALANCE + (BALANCE \* 0.01)

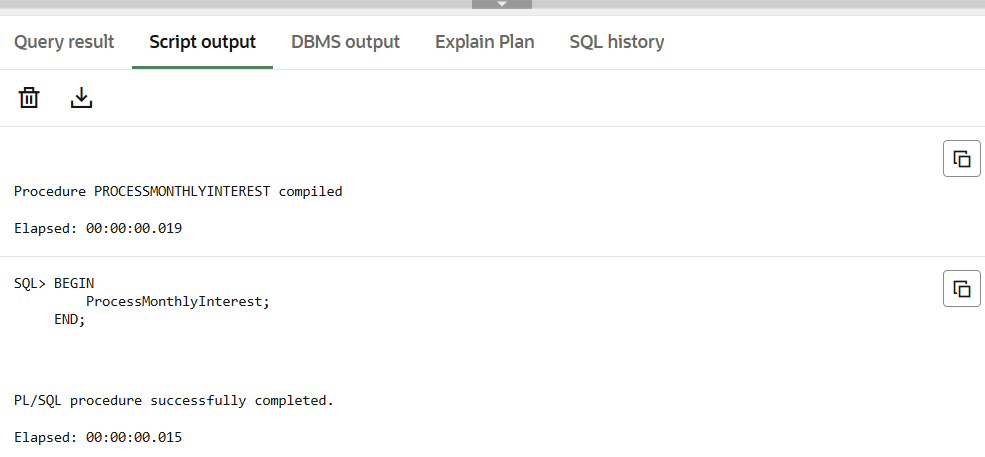
WHERE ACCT\_ID = acc.ACCT\_ID;

END LOOP;

COMMIT;

END;

**Output:**



**Scenario 2:**

We need to create a table called employees

CREATE TABLE EMPLOYEES (

    EMP\_ID NUMBER,

    NAME VARCHAR2(100),

    DEPT\_ID NUMBER,

    SALARY NUMBER

);

INSERT INTO EMPLOYEES VALUES (1, 'John', 101, 30000);

INSERT INTO EMPLOYEES VALUES (2, 'Priya', 101, 35000);

INSERT INTO EMPLOYEES VALUES (3, 'Ravi', 102, 28000);

INSERT INTO EMPLOYEES VALUES (4, 'Neha', 102, 32000);

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

    dept\_id IN NUMBER,

    bonus\_percent IN NUMBER

) IS

BEGIN

    FOR emp IN (SELECT \* FROM EMPLOYEES WHERE DEPT\_ID = dept\_id) LOOP

        UPDATE EMPLOYEES

        SET SALARY = SALARY + (SALARY \* bonus\_percent / 100)

        WHERE EMP\_ID = emp.EMP\_ID;

        DBMS\_OUTPUT.PUT\_LINE('Bonus applied to employee: ' || emp.NAME);

    END LOOP;

    COMMIT;

END;

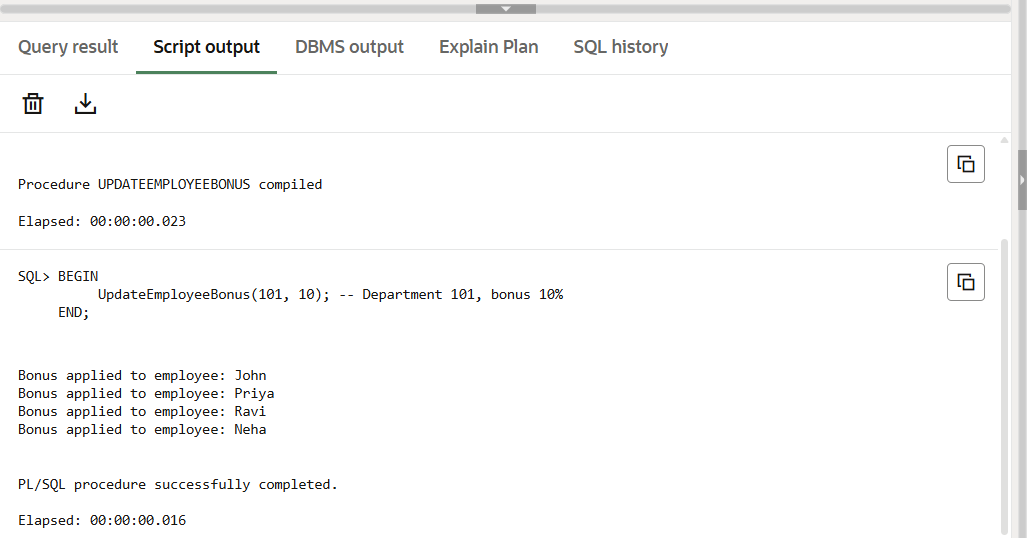
/

BEGIN

     UpdateEmployeeBonus(101, 10); -- Department 101, bonus 10%

END;

/



**Scenario 3:**

we need to create a table called bank\_accounts

CREATE TABLE BANK\_ACCOUNTS (

ACCT\_ID NUMBER,

CUST\_ID NUMBER,

BALANCE NUMBER

);

-- Insert 3 accounts with sample balances

INSERT INTO BANK\_ACCOUNTS VALUES (1, 101, 5000);

INSERT INTO BANK\_ACCOUNTS VALUES (2, 102, 3000);

INSERT INTO BANK\_ACCOUNTS VALUES (3, 103, 1000);

CREATE OR REPLACE PROCEDURE TransferFunds (

    from\_acct IN NUMBER,

    to\_acct IN NUMBER,

    amount IN NUMBER

) IS

    from\_balance NUMBER;

BEGIN

    -- Get balance of the source account

    SELECT BALANCE INTO from\_balance

    FROM BANK\_ACCOUNTS

    WHERE ACCT\_ID = from\_acct;

    -- Check if enough balance is available

    IF from\_balance >= amount THEN

        -- Deduct from source account

        UPDATE BANK\_ACCOUNTS

        SET BALANCE = BALANCE - amount

        WHERE ACCT\_ID = from\_acct;

        -- Add to destination account

        UPDATE BANK\_ACCOUNTS

        SET BALANCE = BALANCE + amount

        WHERE ACCT\_ID = to\_acct;

        DBMS\_OUTPUT.PUT\_LINE('Transfer successful. Amount: ' || amount);

        COMMIT;

    ELSE

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

    END IF;

END;

/

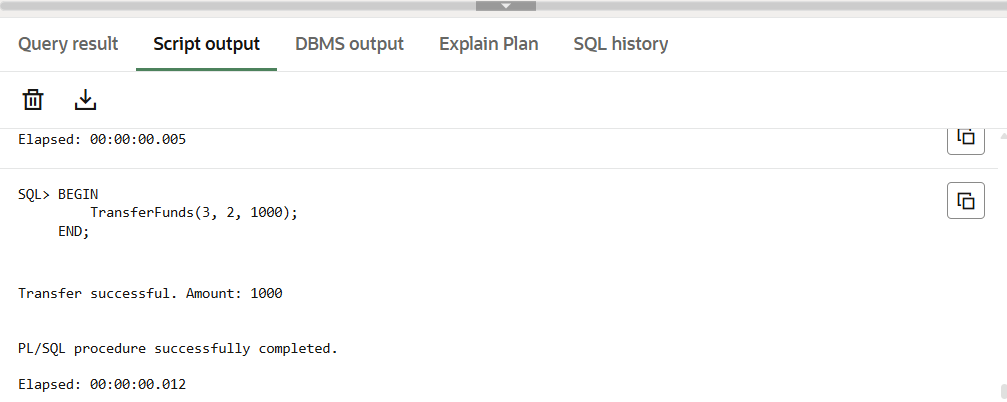
BEGIN

    TransferFunds(3, 2, 1000);

END;

/

**Output:**

****