**Exercise 3: Stored Procedures**

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

* + **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.**

**PL/SQL CODE:**  
  
CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

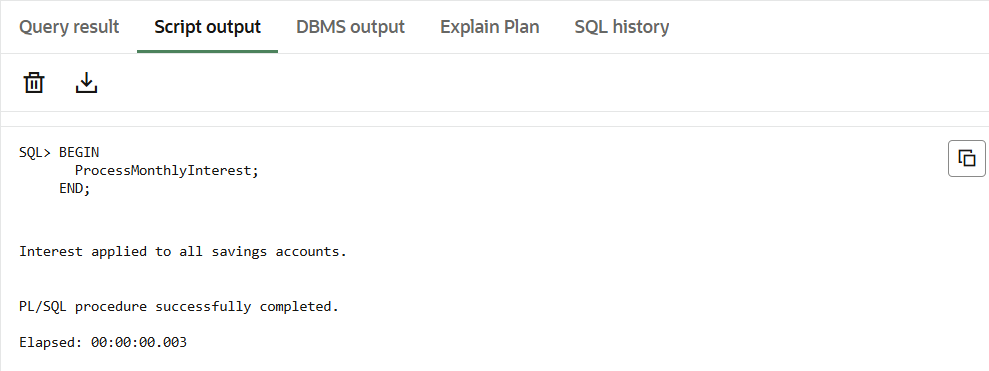
UPDATE Accounts

SET Balance = Balance + (Balance \* 0.01)

WHERE AccountType = 'Savings';

DBMS\_OUTPUT.PUT\_LINE('Interest applied to all savings accounts.');

END;

OUTPUT:  


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

* + **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.**

**PL/SQL CODE:**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

deptName IN VARCHAR2,

bonusPercent IN NUMBER

) IS

BEGIN

UPDATE Employees

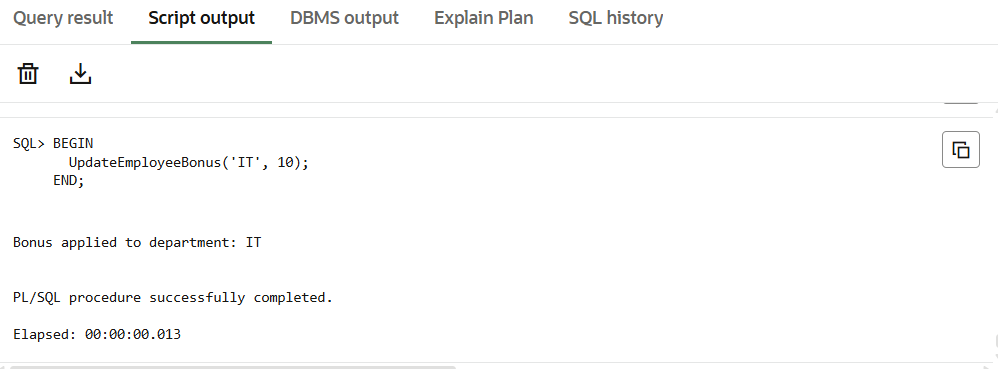
SET Salary = Salary + (Salary \* bonusPercent / 100)

WHERE Department = deptName;

DBMS\_OUTPUT.PUT\_LINE('Bonus applied to department: ' || deptName);

END;

/**OUTPUT:**



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

**PL/SQL CODE:**

CREATE OR REPLACE PROCEDURE TransferFunds (

fromAccountId IN NUMBER,

toAccountId IN NUMBER,

amount IN NUMBER

) IS

insufficient\_balance EXCEPTION;

BEGIN

-- Check balance first

DECLARE

source\_balance NUMBER;

BEGIN

SELECT Balance INTO source\_balance

FROM Accounts

WHERE AccountID = fromAccountId;

IF source\_balance < amount THEN

RAISE insufficient\_balance;

END IF;

END;

-- Debit from source

UPDATE Accounts

SET Balance = Balance - amount

WHERE AccountID = fromAccountId;

-- Credit to target

UPDATE Accounts

SET Balance = Balance + amount

WHERE AccountID = toAccountId;

DBMS\_OUTPUT.PUT\_LINE('Transferred ' || amount ||

' from account ' || fromAccountId ||

' to account ' || toAccountId);

EXCEPTION

WHEN insufficient\_balance THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Insufficient balance in source account.');

END;

**OUTPUT**