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.

**TABLES USED**

CREATE TABLE Accounts (

    AccountID NUMBER PRIMARY KEY,

    CustomerID NUMBER,

    AccountType VARCHAR2(20),

    Balance NUMBER,

    LastModified DATE

);

**INSERTING VALUES INTO THE TABLES**

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)

VALUES (1, 1, 'Savings', 1000, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)

VALUES (2, 2, 'Checking', 1500, SYSDATE);

**CREATE PROCEDURE**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

    FOR acc IN (SELECT AccountID, Balance FROM Accounts WHERE AccountType = 'Savings') LOOP

        UPDATE Accounts

        SET Balance = Balance + (acc.Balance \* 0.01),

            LastModified = SYSDATE

        WHERE AccountID = acc.AccountID;

    END LOOP;

    COMMIT;

END;

/

**RUN THE PROCEDURE**

BEGIN

    ProcessMonthlyInterest;

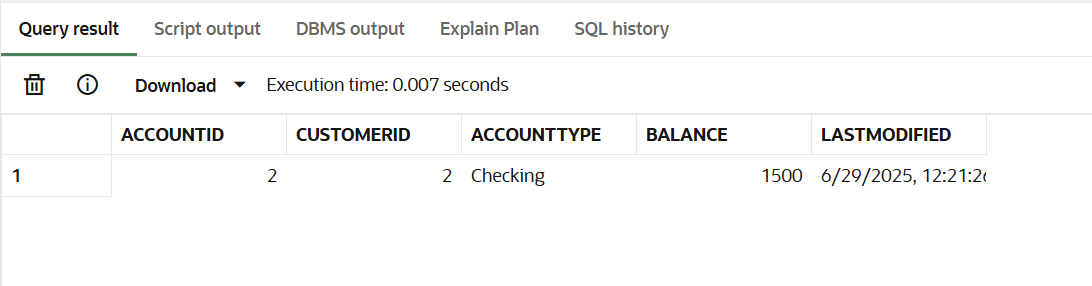
END;

/

**VIEW RESULT**

SELECT \* FROM Accounts;

**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.

**TABLES USED**

CREATE TABLE Employees (

    EmployeeID NUMBER PRIMARY KEY,

    Name VARCHAR2(100),

    Position VARCHAR2(50),

    Salary NUMBER,

    Department VARCHAR2(50),

    HireDate DATE

);

**INSERTING VALUES**

INSERT INTO Employees VALUES (1, 'Alice Johnson', 'Manager', 70000, 'HR', TO\_DATE('2015-06-15', 'YYYY-MM-DD'));

INSERT INTO Employees VALUES (2, 'Bob Brown', 'Developer', 60000, 'IT', TO\_DATE('2017-03-20', 'YYYY-MM-DD'));

**CREATE PROCEDURE**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

    dept\_name IN VARCHAR2,

    bonus\_percent IN NUMBER

) IS

BEGIN

    UPDATE Employees

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

    WHERE Department = dept\_name;

    COMMIT;

END;

/

**RUN THE PROCEDURE**

BEGIN

    UpdateEmployeeBonus('IT', 10);

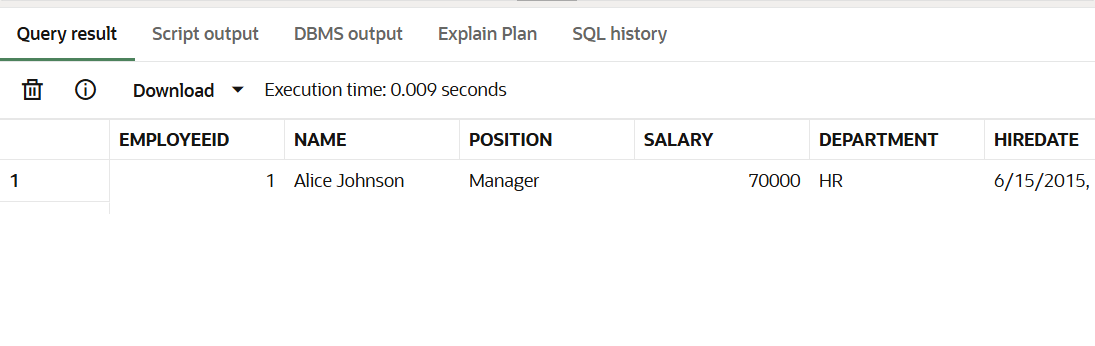
END;

/

**VIEW RESULT**

SELECT \* FROM Employees;

**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.

**TABLES USED**

CREATE TABLE Accounts (

    AccountID NUMBER PRIMARY KEY,

    CustomerID NUMBER,

    AccountType VARCHAR2(20),

    Balance NUMBER,

    LastModified DATE

);

CREATE TABLE Transactions (

    TransactionID NUMBER PRIMARY KEY,

    AccountID NUMBER,

    TransactionDate DATE,

    Amount NUMBER,

    TransactionType VARCHAR2(10)

);

**INSERTING VALUES**

-- Accounts

INSERT INTO Accounts VALUES (1, 1, 'Savings', 1000, SYSDATE);

INSERT INTO Accounts VALUES (2, 2, 'Checking', 1500, SYSDATE);

-- Transactions

INSERT INTO Transactions VALUES (1, 1, SYSDATE, 200, 'Deposit');

INSERT INTO Transactions VALUES (2, 2, SYSDATE, 300, 'Withdrawal');

**CREATE PROCEDURE**

CREATE OR REPLACE PROCEDURE TransferFunds (

    from\_account\_id IN NUMBER,

    to\_account\_id IN NUMBER,

    amount IN NUMBER

) IS

    insufficient\_balance EXCEPTION;

BEGIN

    DECLARE

        from\_balance NUMBER;

    BEGIN

        SELECT Balance INTO from\_balance FROM Accounts WHERE AccountID = from\_account\_id FOR UPDATE;

        IF from\_balance < amount THEN

            RAISE insufficient\_balance;

        END IF;

        -- Debit from source

        UPDATE Accounts

        SET Balance = Balance - amount, LastModified = SYSDATE

        WHERE AccountID = from\_account\_id;

        -- Credit to destination

        UPDATE Accounts

        SET Balance = Balance + amount, LastModified = SYSDATE

        WHERE AccountID = to\_account\_id;

        -- Record transactions

        INSERT INTO Transactions VALUES (

            Transactions\_seq.NEXTVAL, from\_account\_id, SYSDATE, amount, 'Debit'

        );

        INSERT INTO Transactions VALUES (

            Transactions\_seq.NEXTVAL, to\_account\_id, SYSDATE, amount, 'Credit'

        );

        COMMIT;

    EXCEPTION

        WHEN insufficient\_balance THEN

            DBMS\_OUTPUT.PUT\_LINE('Transfer failed: insufficient funds.');

    END;

END;

/

CREATE SEQUENCE Transactions\_seq START WITH 3 INCREMENT BY 1;

**RUN THE PROCEDURE**

BEGIN

    TransferFunds(1, 2, 300); -- Transfer $300 from account 1 to 2

END;

/

**VIEW RESULT**

SELECT \* FROM Accounts;

SELECT \* FROM Transactions;

**OUTPUT**

