WEEK 2 : PL/SQL programming

**Exercise 1: stored procedure**

-- STEP 1: Create Tables

-- Table 1: SavingsAccounts

CREATE TABLE SavingsAccounts (

    AccountID NUMBER PRIMARY KEY,

    CustomerName VARCHAR2(50),

    Balance NUMBER

);

-- Table 2: Employees

CREATE TABLE Employees (

    EmployeeID NUMBER PRIMARY KEY,

    Name VARCHAR2(50),

    Department VARCHAR2(50),

    Salary NUMBER

);

-- Table 3: Accounts (for TransferFunds)

CREATE TABLE Accounts (

    AccountNumber NUMBER PRIMARY KEY,

    CustomerName VARCHAR2(50),

    Balance NUMBER

);

-- STEP 2: Insert Sample Data

-- SavingsAccounts data

INSERT INTO SavingsAccounts VALUES (1, 'Alice', 10000);

INSERT INTO SavingsAccounts VALUES (2, 'Bob', 20000);

-- Employees data

INSERT INTO Employees VALUES (1, 'John', 'Sales', 50000);

INSERT INTO Employees VALUES (2, 'Sara', 'HR', 45000);

INSERT INTO Employees VALUES (3, 'Mike', 'Sales', 55000);

-- Accounts data for transfer

INSERT INTO Accounts VALUES (101, 'Krishika', 15000);

INSERT INTO Accounts VALUES (102, 'Krishika', 10000);

COMMIT;

-- STEP 3: Stored Procedure – ProcessMonthlyInterest (1% interest to all savings)

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

    FOR acc IN (SELECT AccountID FROM SavingsAccounts)

    LOOP

        UPDATE SavingsAccounts

        SET Balance = Balance + (Balance \* 0.01)

        WHERE AccountID = acc.AccountID;

    END LOOP;

    COMMIT;

END;

/

-- Run Procedure

BEGIN

    ProcessMonthlyInterest;

END;

/

-- STEP 4: Stored Procedure – UpdateEmployeeBonus

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

    dept IN VARCHAR2,

    bonus\_percent IN NUMBER

) IS

BEGIN

    UPDATE Employees

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

    WHERE Department = dept;

    COMMIT;

END;

/

-- Run Procedure (Give 10% bonus to Sales department)

BEGIN

    UpdateEmployeeBonus('Sales', 10);

END;

/

-- STEP 5: Stored Procedure – TransferFunds

CREATE OR REPLACE PROCEDURE TransferFunds (

    from\_acc IN NUMBER,

    to\_acc IN NUMBER,

    amount IN NUMBER

) IS

    insufficient\_balance EXCEPTION;

    from\_balance NUMBER;

BEGIN

    -- Check balance

    SELECT Balance INTO from\_balance FROM Accounts WHERE AccountNumber = from\_acc;

    IF from\_balance < amount THEN

        RAISE insufficient\_balance;

    END IF;

    -- Transfer

    UPDATE Accounts

    SET Balance = Balance - amount

    WHERE AccountNumber = from\_acc;

    UPDATE Accounts

    SET Balance = Balance + amount

    WHERE AccountNumber = to\_acc;

    COMMIT;

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

EXCEPTION

    WHEN insufficient\_balance THEN

        DBMS\_OUTPUT.PUT\_LINE('Error: Insufficient balance for transfer.');

    WHEN OTHERS THEN

        DBMS\_OUTPUT.PUT\_LINE('Unexpected error: ' || SQLERRM);

END;

/

-- Run Procedure (Transfer ₹2000 from Account 101 to 102)

BEGIN

    TransferFunds(101, 102, 2000);

END;

/

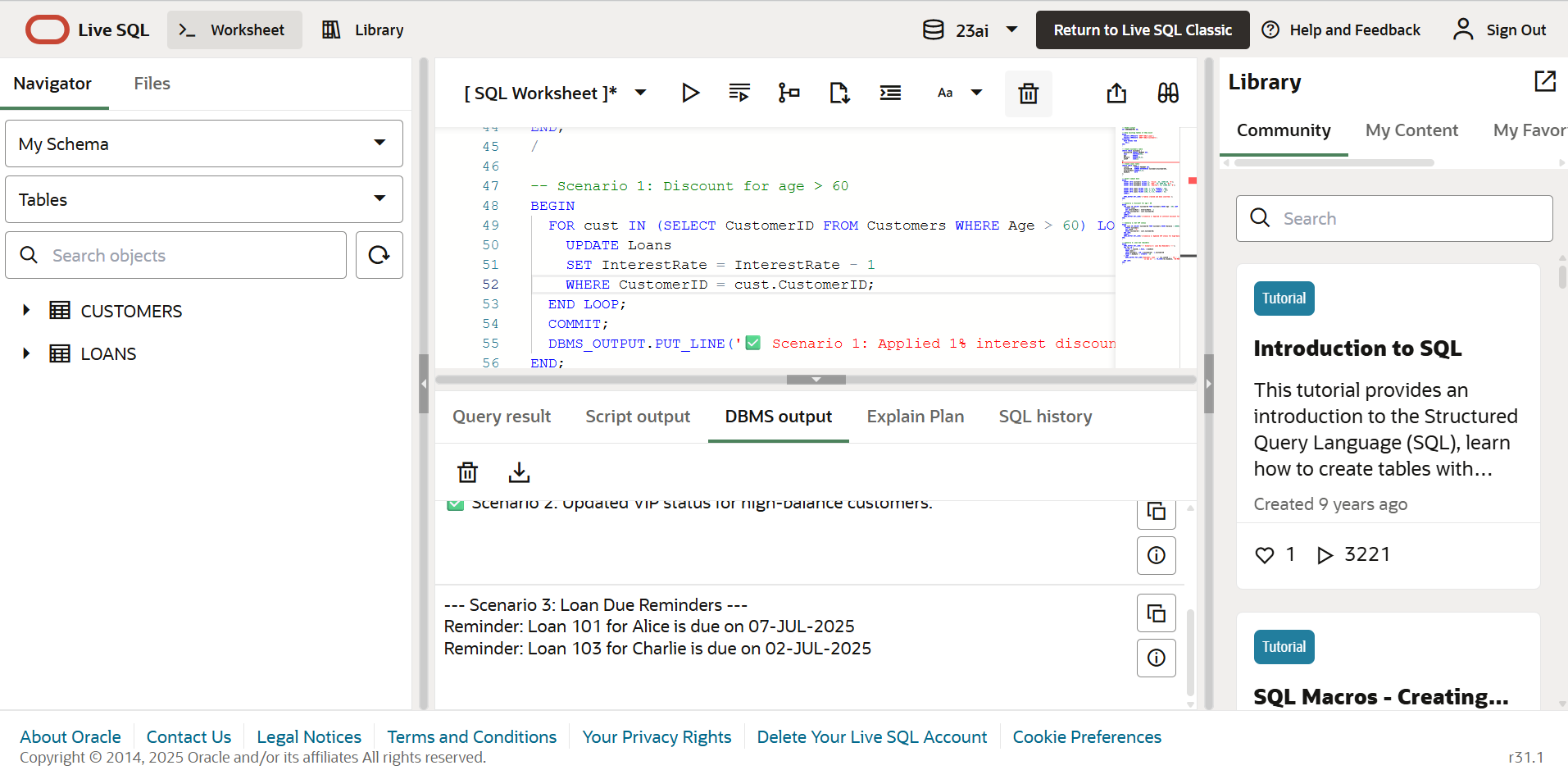
-- STEP 6: Verify the Final Data

SELECT \* FROM SavingsAccounts;

SELECT \* FROM Employees;

SELECT \* FROM Accounts;

**OUTPUT:**

****