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.

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

  DBMS\_OUTPUT.PUT\_LINE('Starting monthly interest processing...');

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

    UPDATE Accounts

    SET Balance = Balance + (rec.Balance \* 0.01)

    WHERE AccountID = rec.AccountID;

    DBMS\_OUTPUT.PUT\_LINE('Interest applied to AccountID: ' ||rec.AccountID);

END LOOP;

  DBMS\_OUTPUT.PUT\_LINE('Monthly interest processing completed.');

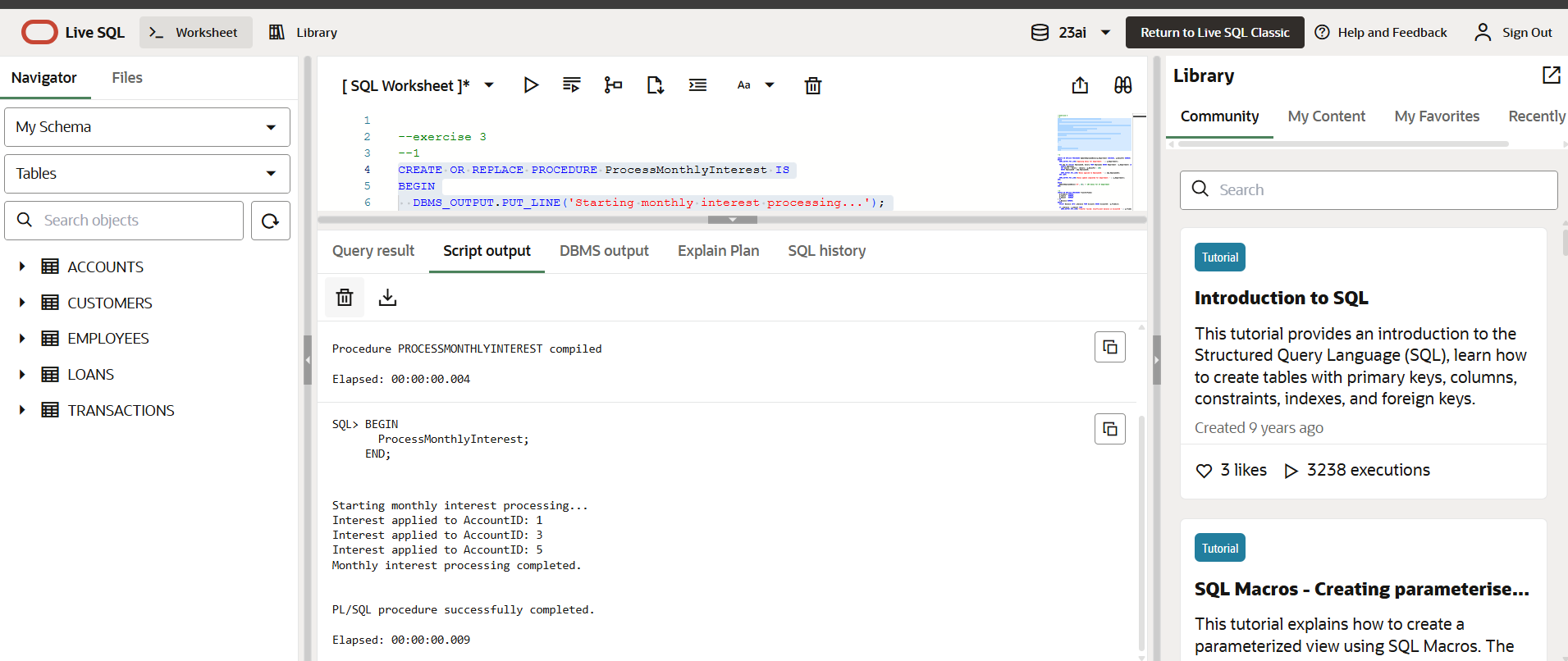
END;

/

BEGIN

  ProcessMonthlyInterest;

END;



**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.

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(p\_Department VARCHAR2, p\_BonusPct NUMBER) IS

BEGIN

  DBMS\_OUTPUT.PUT\_LINE('Applying bonus for department: ' || p\_Department);

  FOR emp IN (SELECT EmployeeID, Salary FROM Employees WHERE Department = p\_Department) LOOP

    UPDATE Employees

    SET Salary = Salary + (Salary \* p\_BonusPct / 100)

    WHERE EmployeeID = emp.EmployeeID;

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

  END LOOP;

  DBMS\_OUTPUT.PUT\_LINE('Bonus update completed for department: ' || p\_Department);

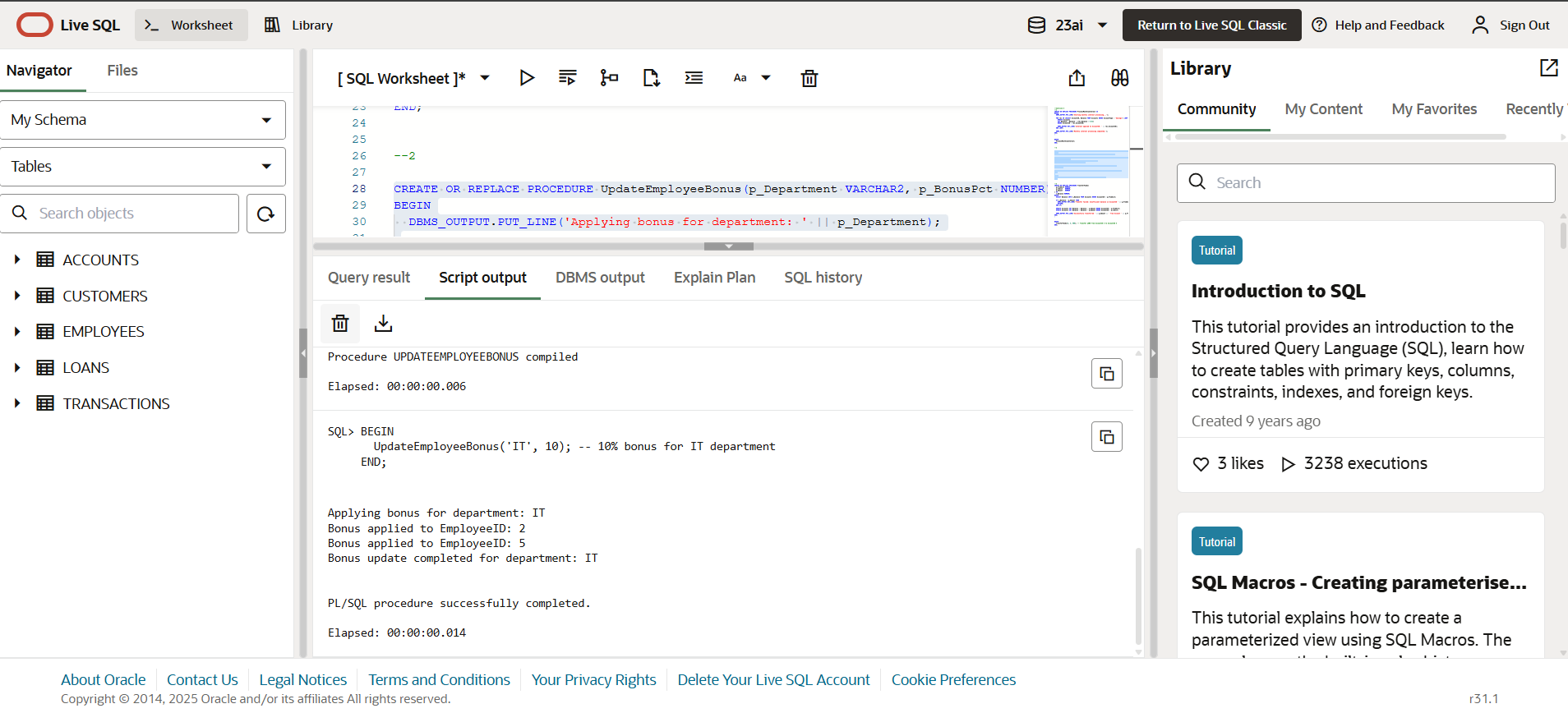
END;

/

BEGIN

  UpdateEmployeeBonus('IT', 10); -- 10% bonus for IT department

END;



**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.

CREATE OR REPLACE PROCEDURE TransferFunds(

  p\_FromAcct NUMBER,

  p\_ToAcct   NUMBER,

  p\_Amount   NUMBER

) IS

  v\_Balance NUMBER;

BEGIN

  SELECT Balance INTO v\_Balance FROM Accounts WHERE AccountID = p\_FromAcct;

  IF v\_Balance < p\_Amount THEN

    DBMS\_OUTPUT.PUT\_LINE('Transfer failed: Insufficient balance in AccountID ' || p\_FromAcct);

    RETURN;

  END IF;

  UPDATE Accounts SET Balance = Balance - p\_Amount WHERE AccountID p\_FromAcct;

  UPDATE Accounts SET Balance = Balance + p\_Amount WHERE AccountID p\_ToAcct;

  DBMS\_OUTPUT.PUT\_LINE('Successfully transferred ' || p\_Amount || ' from Account ' || p\_FromAcct || ' to Account ' || p\_ToAcct);

END;

/

BEGIN

  TransferFunds(3, 4, 500); -- Transfer ₹500 from AccountID 3 to AccountID 4

END;

