Exercise 7: Packages

**Scenario 1:** Group all customer-related procedures and functions into a package.

* + **Question:** Create a package **CustomerManagement** with procedures for adding a new customer, updating customer details, and a function to get customer balance.

CREATE OR REPLACE PACKAGE CustomerManagement AS

  PROCEDURE AddCustomer(p\_ID NUMBER, p\_Name VARCHAR2, p\_DOB DATE, p\_Balance NUMBER);

  PROCEDURE UpdateCustomer(p\_ID NUMBER, p\_Name VARCHAR2, p\_Balance NUMBER);

  FUNCTION GetCustomerBalance(p\_ID NUMBER) RETURN NUMBER;

END CustomerManagement;

/

CREATE OR REPLACE PACKAGE BODY CustomerManagement AS

  PROCEDURE AddCustomer(p\_ID NUMBER, p\_Name VARCHAR2, p\_DOB DATE, p\_Balance NUMBER) IS

  BEGIN

    INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

    VALUES (p\_ID, p\_Name, p\_DOB, p\_Balance, SYSDATE);

  END;

  PROCEDURE UpdateCustomer(p\_ID NUMBER, p\_Name VARCHAR2, p\_Balance NUMBER) IS

  BEGIN

    UPDATE Customers

    SET Name = p\_Name, Balance = p\_Balance, LastModified = SYSDATE

    WHERE CustomerID = p\_ID;

  END;

  FUNCTION GetCustomerBalance(p\_ID NUMBER) RETURN NUMBER IS

    v\_balance NUMBER;

  BEGIN

    SELECT Balance INTO v\_balance FROM Customers WHERE CustomerID = p\_ID;

    RETURN v\_balance;

  END;

END CustomerManagement;

/

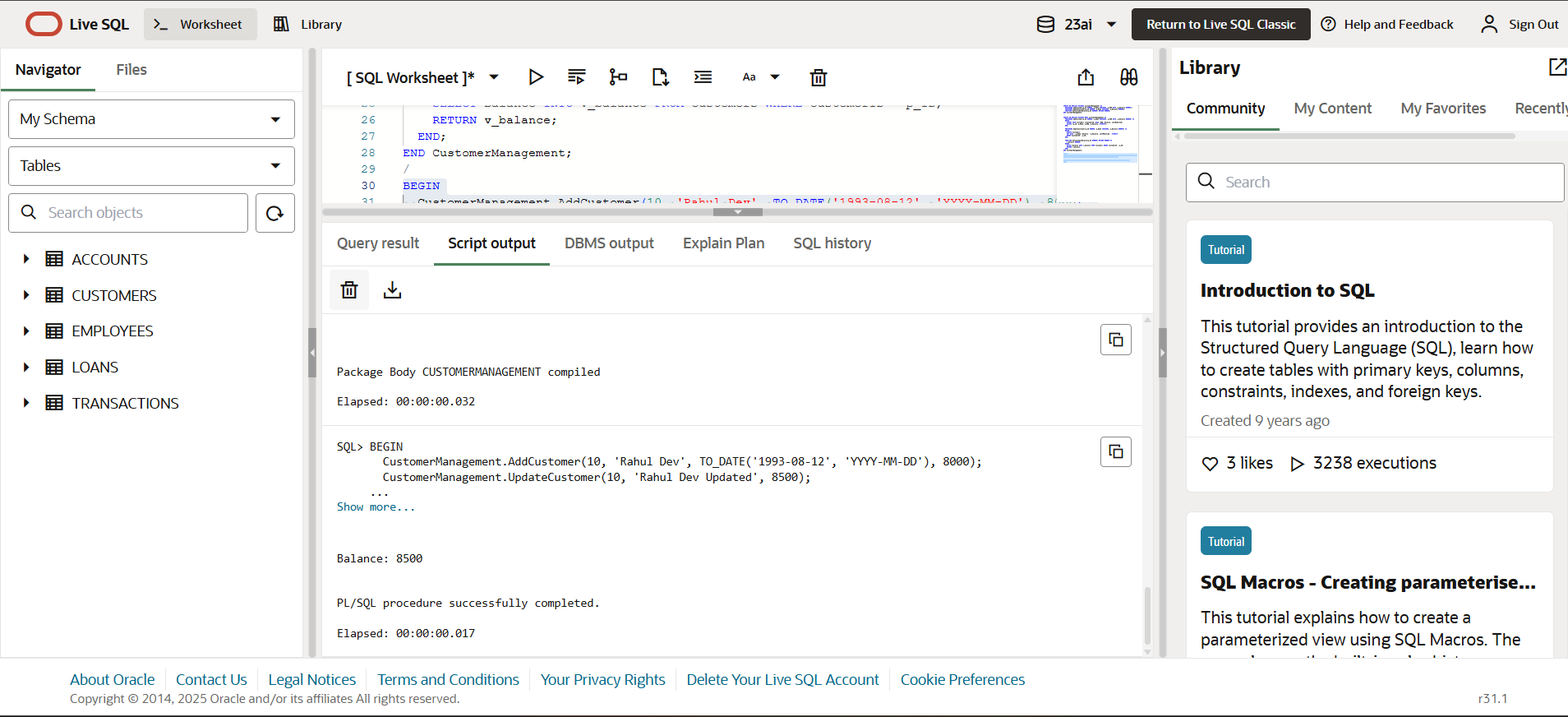
BEGIN

  CustomerManagement.AddCustomer(10, 'Rahul Dev', TO\_DATE('1993-08-12', 'YYYY-MM-DD'), 8000);

  CustomerManagement.UpdateCustomer(10, 'Rahul Dev Updated', 8500);

  DBMS\_OUTPUT.PUT\_LINE('Balance: ' || CustomerManagement.GetCustomerBalance(10));

END;



**Scenario 2:** Create a package to manage employee data.

* + **Question:** Write a package **EmployeeManagement** with procedures to hire new employees, update employee details, and a function to calculate annual salary.

CREATE OR REPLACE PACKAGE EmployeeManagement AS

  PROCEDURE HireEmployee(p\_ID NUMBER, p\_Name VARCHAR2, p\_Position VARCHAR2, p\_Salary NUMBER, p\_Dept VARCHAR2, p\_HireDate DATE);

  PROCEDURE UpdateEmployee(p\_ID NUMBER, p\_Salary NUMBER);

  FUNCTION GetAnnualSalary(p\_ID NUMBER) RETURN NUMBER;

END EmployeeManagement;

/

CREATE OR REPLACE PACKAGE BODY EmployeeManagement AS

  PROCEDURE HireEmployee(p\_ID NUMBER, p\_Name VARCHAR2, p\_Position VARCHAR2, p\_Salary NUMBER, p\_Dept VARCHAR2, p\_HireDate DATE) IS

  BEGIN

    INSERT INTO Employees (EmployeeID, Name, Position, Salary, Department, HireDate)

    VALUES (p\_ID, p\_Name, p\_Position, p\_Salary, p\_Dept, p\_HireDate);

  END;

  PROCEDURE UpdateEmployee(p\_ID NUMBER, p\_Salary NUMBER) IS

  BEGIN

    UPDATE Employees SET Salary = p\_Salary WHERE EmployeeID = p\_ID;

  END;

  FUNCTION GetAnnualSalary(p\_ID NUMBER) RETURN NUMBER IS

    v\_salary NUMBER;

  BEGIN

    SELECT Salary INTO v\_salary FROM Employees WHERE EmployeeID = p\_ID;

    RETURN v\_salary \* 12;

  END;

END EmployeeManagement;

/

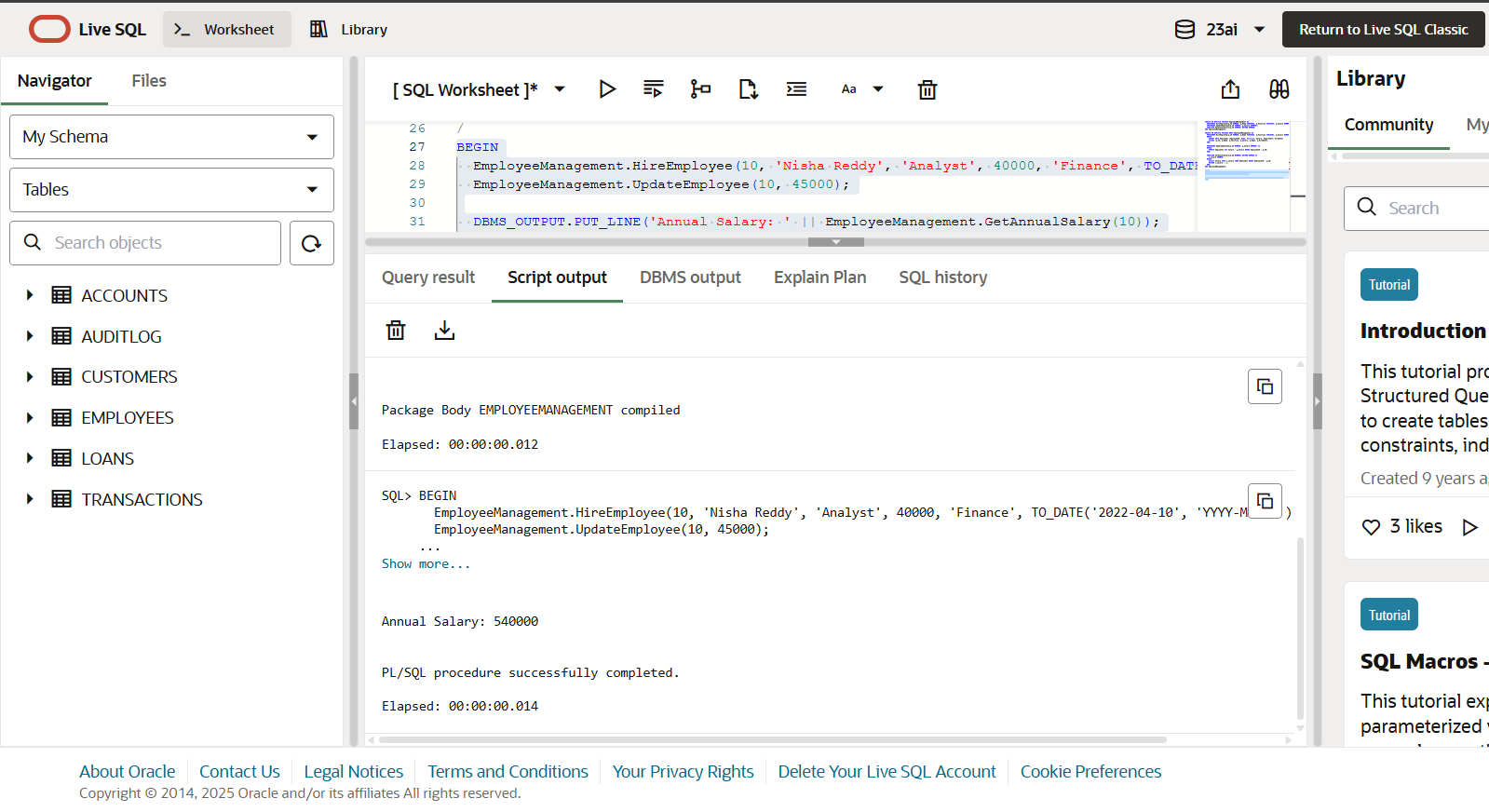
BEGIN

  EmployeeManagement.HireEmployee(10, 'Nisha Reddy', 'Analyst', 40000, 'Finance', TO\_DATE('2022-04-10', 'YYYY-MM-DD'));

  EmployeeManagement.UpdateEmployee(10, 45000);

  DBMS\_OUTPUT.PUT\_LINE('Annual Salary: ' || EmployeeManagement.GetAnnualSalary(10));

END;



**Scenario 3:** Group all account-related operations into a package.

* + **Question:** Create a package **AccountOperations** with procedures for opening a new account, closing an account, and a function to get the total balance of a customer across all accounts.

CREATE OR REPLACE PACKAGE AccountOperations AS

  PROCEDURE OpenAccount(p\_AccountID NUMBER, p\_CustID NUMBER, p\_Type VARCHAR2, p\_Balance NUMBER);

  PROCEDURE CloseAccount(p\_AccountID NUMBER);

  FUNCTION TotalCustomerBalance(p\_CustID NUMBER) RETURN NUMBER;

END AccountOperations;

/

CREATE OR REPLACE PACKAGE BODY AccountOperations AS

  PROCEDURE OpenAccount(p\_AccountID NUMBER, p\_CustID NUMBER, p\_Type VARCHAR2, p\_Balance NUMBER) IS

  BEGIN

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

    VALUES (p\_AccountID, p\_CustID, p\_Type, p\_Balance, SYSDATE);

  END;

  PROCEDURE CloseAccount(p\_AccountID NUMBER) IS

  BEGIN

    DELETE FROM Accounts WHERE AccountID = p\_AccountID;

  END;

  FUNCTION TotalCustomerBalance(p\_CustID NUMBER) RETURN NUMBER IS

    v\_total NUMBER;

  BEGIN

    SELECT SUM(Balance) INTO v\_total FROM Accounts WHERE CustomerID = p\_CustID;

    RETURN NVL(v\_total, 0);

  END;

END AccountOperations;

/

BEGIN

  AccountOperations.OpenAccount(20, 10, 'Savings', 5000);

  AccountOperations.OpenAccount(21, 10, 'Current', 3000);

  DBMS\_OUTPUT.PUT\_LINE('Total Balance: ' || AccountOperations.TotalCustomerBalance(10));

  AccountOperations.CloseAccount(21);

END;

