**Exercise 7: Packages**

CREATE TABLE Customers(CustomerID NUMBER PRIMARY KEY,Name VARCHAR2(100),DOB DATE,Balance NUMBER,LastModified DATE);

CREATE TABLE Accounts(AccountID NUMBER PRIMARY KEY,CustomerID NUMBER,AccountType VARCHAR2(20),Balance NUMBER,LastModified DATE,FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID));

CREATE TABLE Transactions(TransactionID NUMBER PRIMARY KEY,AccountID NUMBER,TransactionDate DATE,Amount NUMBER,TransactionType VARCHAR2(10),FOREIGN KEY (AccountID) REFERENCES Accounts(AccountID));

CREATE TABLE Loans(LoanID NUMBER PRIMARY KEY,CustomerID NUMBER,LoanAmount NUMBER,InterestRate NUMBER,StartDate DATE,EndDate DATE,FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID));

CREATE TABLE Employees(EmployeeID NUMBER PRIMARY KEY,Name VARCHAR2(100),Position VARCHAR2(50),Salary NUMBER,Department VARCHAR2(50),HireDate DATE);

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

VALUES(1,'John Doe',TO\_DATE('1985-05-15','YYYY-MM-DD'),1000,SYSDATE);

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

VALUES(2,'Jane Smith',TO\_DATE('1990-07-20','YYYY-MM-DD'),1500,SYSDATE);

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);

INSERT INTO Transactions(TransactionID,AccountID,TransactionDate,Amount,TransactionType)

VALUES(1,1,SYSDATE,200,'Deposit');

INSERT INTO Transactions(TransactionID,AccountID,TransactionDate,Amount,TransactionType)

VALUES(2,2,SYSDATE,300,'Withdrawal');

INSERT INTO Loans(LoanID,CustomerID,LoanAmount,InterestRate,StartDate,EndDate)

VALUES(1,1,5000,5,SYSDATE,ADD\_MONTHS(SYSDATE,60));

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

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

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

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

CREATE OR REPLACE PACKAGE CustomerManagement IS

    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 IS

    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);

    EXCEPTION

        WHEN DUP\_VAL\_ON\_INDEX THEN

            DBMS\_OUTPUT.PUT\_LINE('Customer already exists.');

    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;

    EXCEPTION

        WHEN NO\_DATA\_FOUND THEN

            RETURN NULL;

    END;

END CustomerManagement;

/

CREATE OR REPLACE PACKAGE EmployeeManagement IS

    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 IS

    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\*12 INTO v\_salary FROM Employees WHERE EmployeeID=p\_id;

        RETURN v\_salary;

    EXCEPTION

        WHEN NO\_DATA\_FOUND THEN

            RETURN NULL;

    END;

END EmployeeManagement;

/

CREATE OR REPLACE PACKAGE AccountOperations IS

    PROCEDURE OpenAccount(p\_accid NUMBER,p\_custid NUMBER,p\_type VARCHAR2,p\_balance NUMBER);

    PROCEDURE CloseAccount(p\_accid NUMBER);

    FUNCTION GetTotalBalance(p\_custid NUMBER) RETURN NUMBER;

END AccountOperations;

/

CREATE OR REPLACE PACKAGE BODY AccountOperations IS

    PROCEDURE OpenAccount(p\_accid NUMBER,p\_custid NUMBER,p\_type VARCHAR2,p\_balance NUMBER) IS

    BEGIN

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

        VALUES(p\_accid,p\_custid,p\_type,p\_balance,SYSDATE);

    END;

    PROCEDURE CloseAccount(p\_accid NUMBER) IS

    BEGIN

        DELETE FROM Accounts WHERE AccountID=p\_accid;

    END;

    FUNCTION GetTotalBalance(p\_custid NUMBER) RETURN NUMBER IS

        v\_total NUMBER:=0;

    BEGIN

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

        RETURN NVL(v\_total,0);

    END;

END AccountOperations;

/

BEGIN

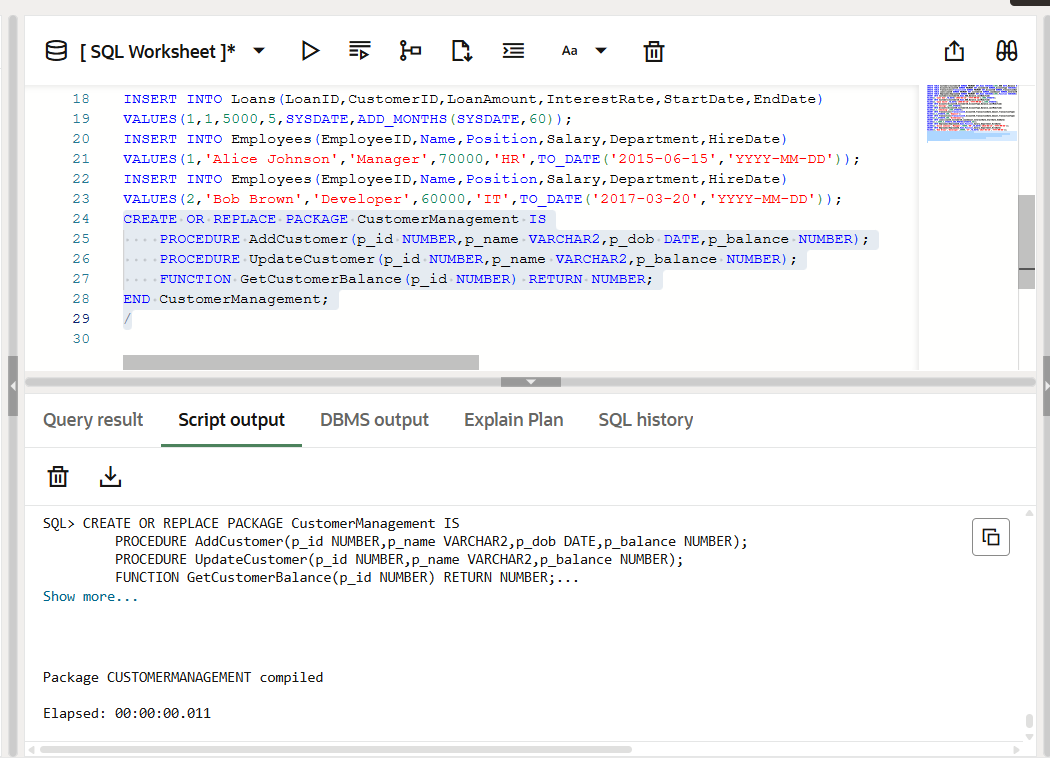
    CustomerManagement.AddCustomer(3,'Sam Wilson',TO\_DATE('1992-09-10','YYYY-MM-DD'),2500);

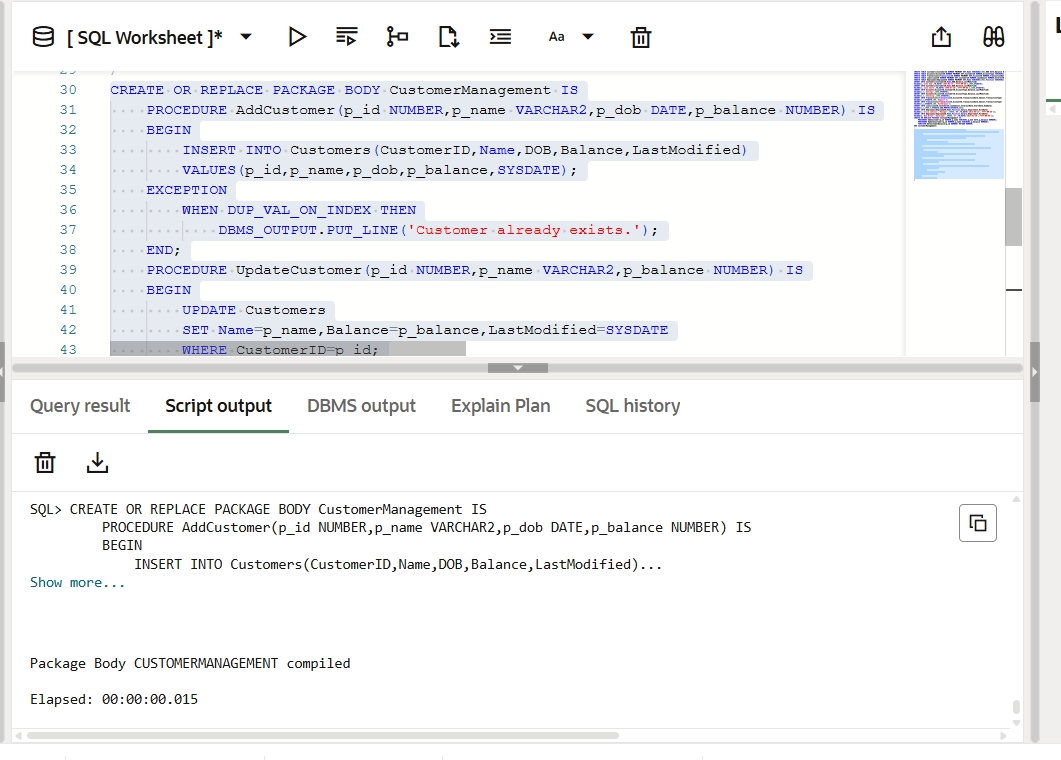
    DBMS\_OUTPUT.PUT\_LINE('Balance: '||CustomerManagement.GetCustomerBalance(3));

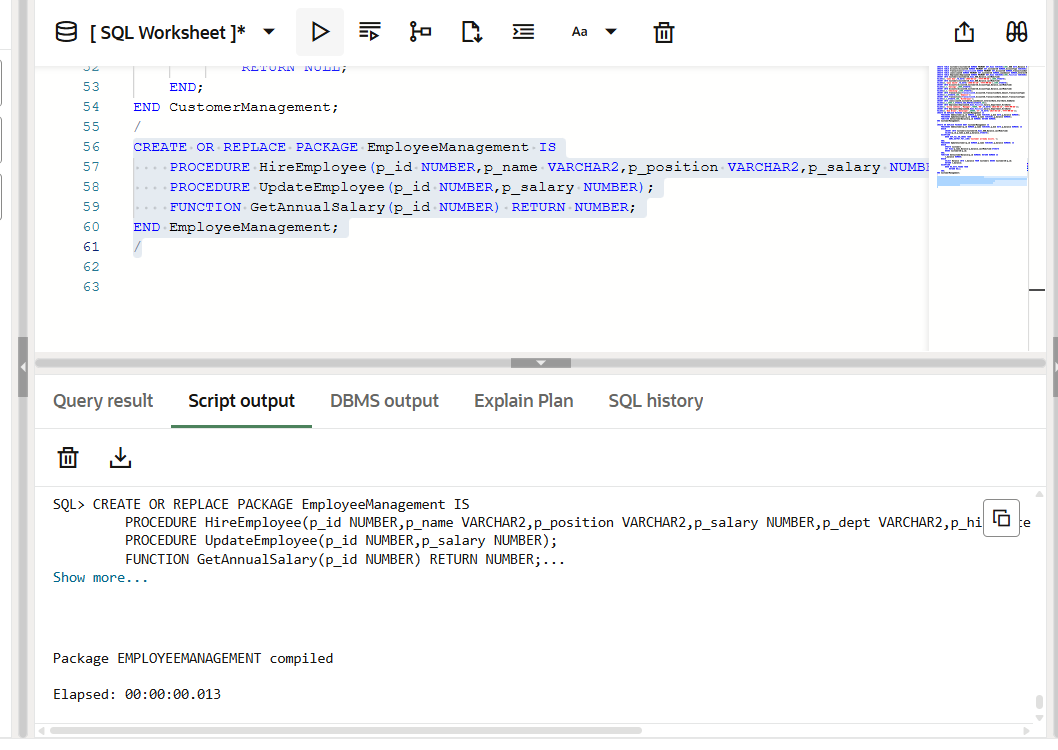
END;

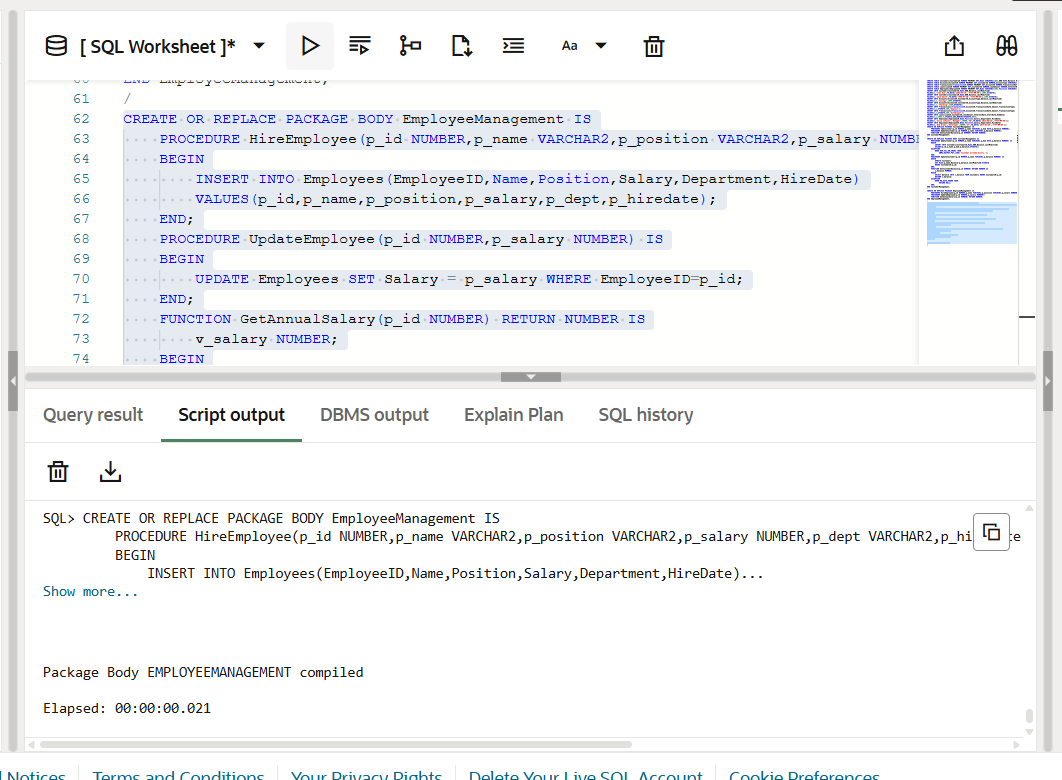
/

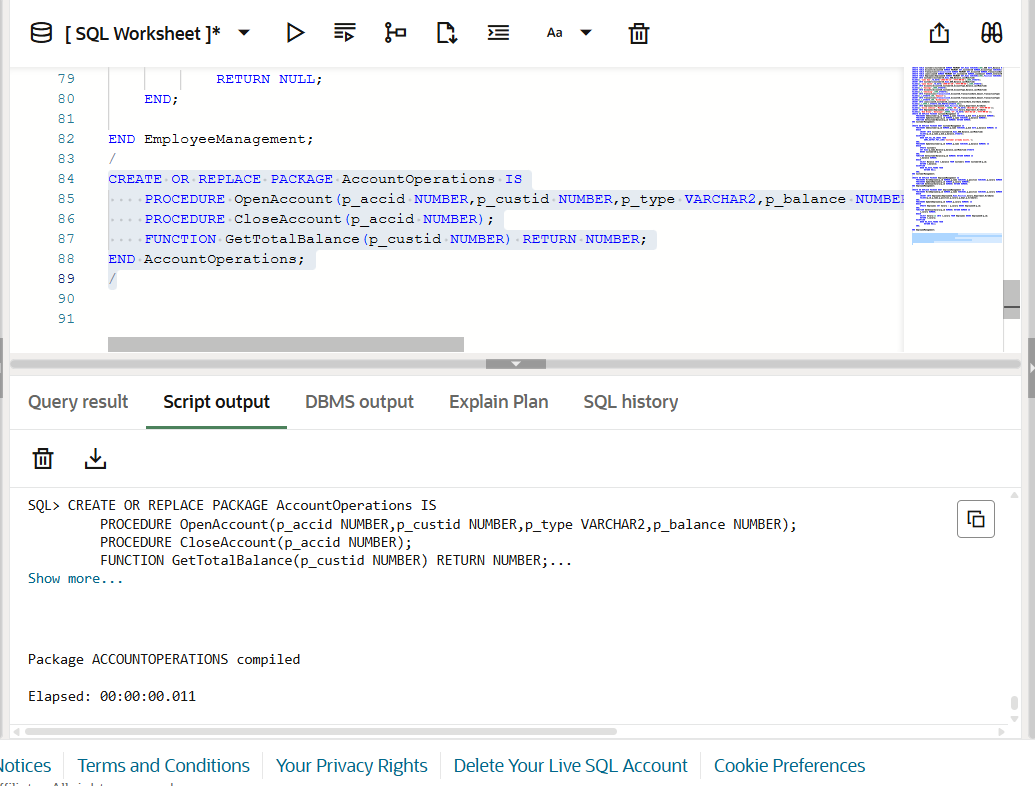
**Output:**

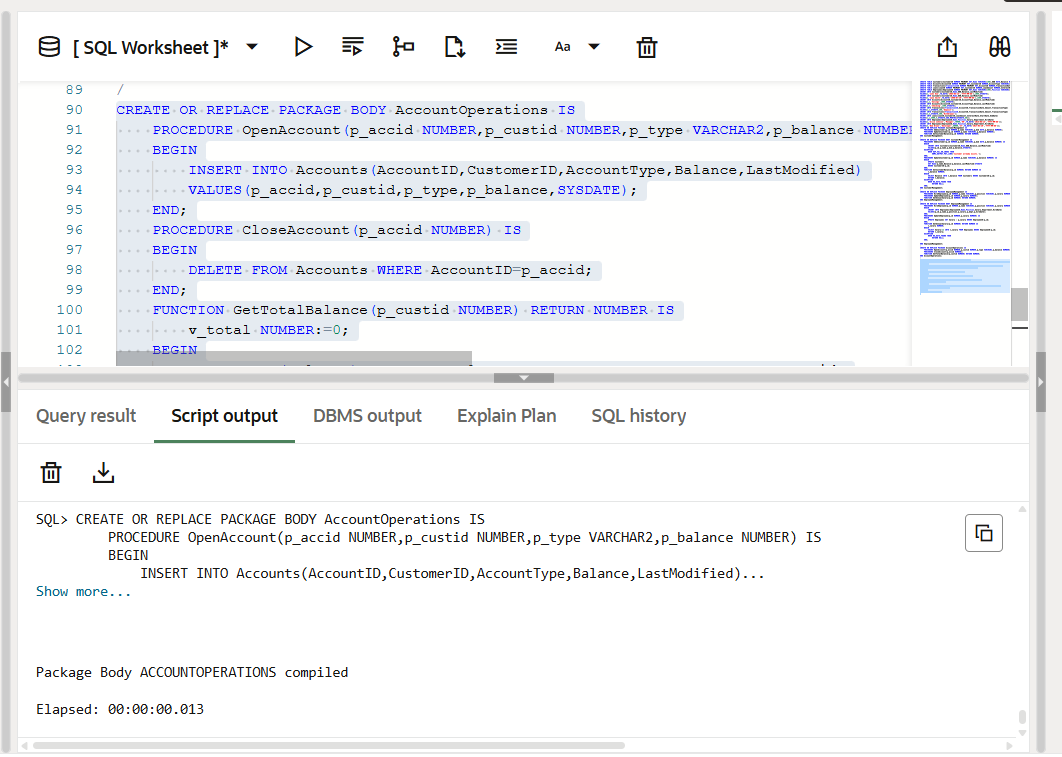
****

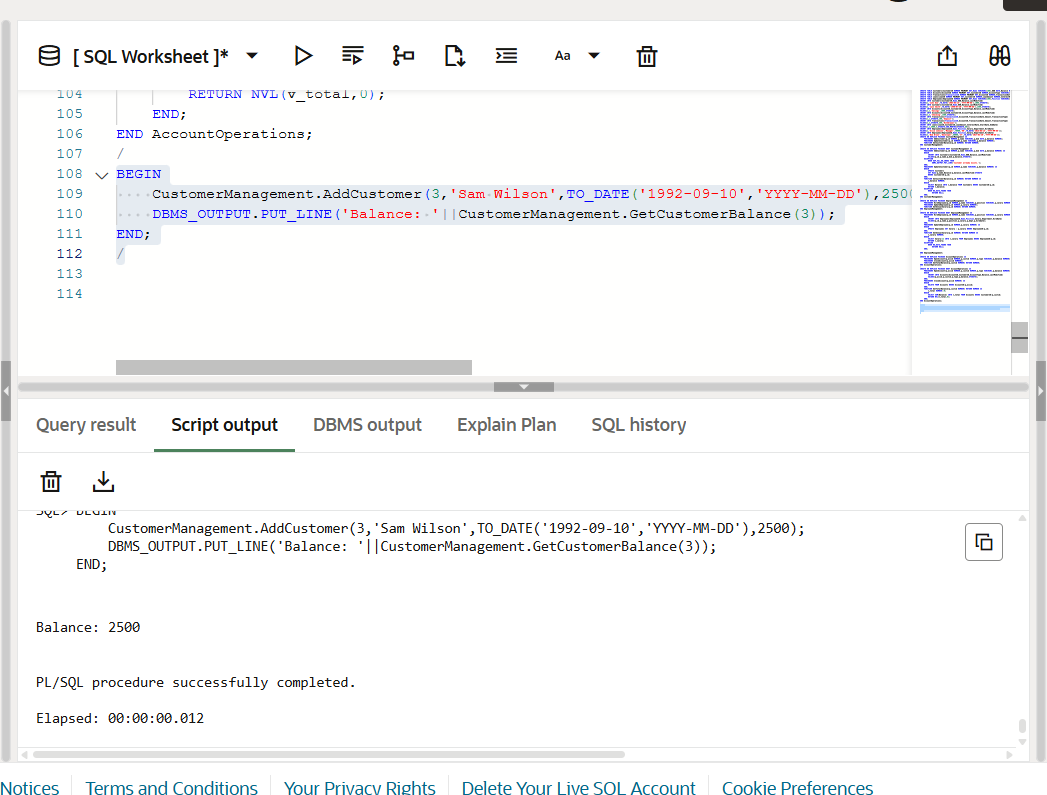
****

****

****

****

****

****