**PACKAGES.**

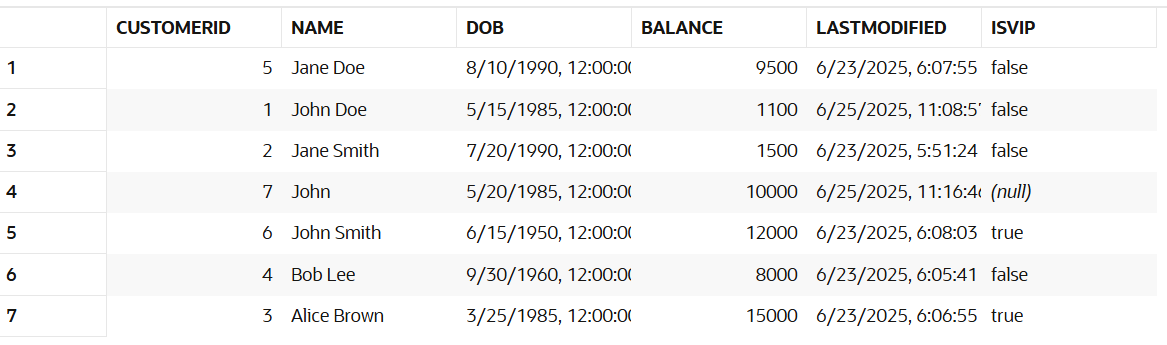
**Scenario 1:** Group all customer-related procedures and functions into a package.

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

**SOLUTION:**

**Query:** select \* from customers;

**Output:**



**Query:**

CREATE OR REPLACE PACKAGE CustomerManagement AS

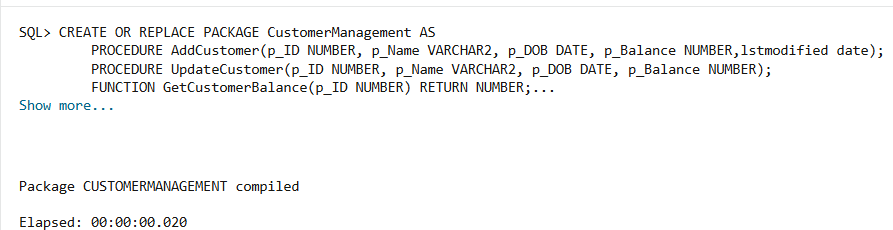
    PROCEDURE AddCustomer(p\_ID NUMBER, p\_Name VARCHAR2, p\_DOB DATE, p\_Balance NUMBER,lstmodified date);

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

    FUNCTION GetCustomerBalance(p\_ID NUMBER) RETURN NUMBER;

END CustomerManagement;

**Output:**



**Query:**

CREATE OR REPLACE PACKAGE BODY CustomerManagement AS

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

    BEGIN

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

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

    END;

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

    BEGIN

        UPDATE Customers

        SET Name = p\_Name,

            DOB = p\_DOB,

            Balance = p\_Balance,

            LastModified = SYSDATE

        WHERE CustomerID = p\_ID;

    END;

    FUNCTION GetCustomerBalance(p\_ID NUMBER) RETURN NUMBER IS

        v\_balance NUMBER;

    BEGIN

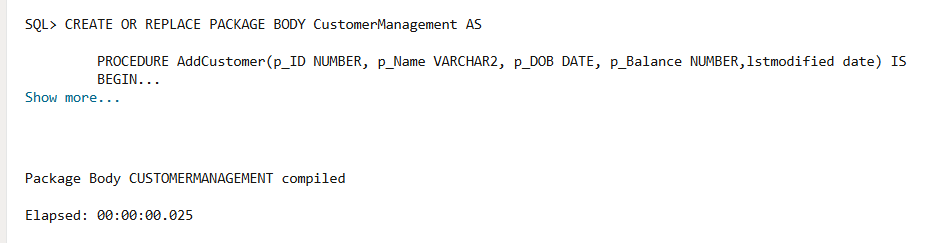
        SELECT Balance INTO v\_balance FROM Customers WHERE CustomerID = p\_ID;

        RETURN v\_balance;

    END;

END CustomerManagement;

**Output:**



**Query:**

declare

c\_bal number;

begin

    CustomerManagement.AddCustomer(8, 'Doe', TO\_DATE('1985-07-15', 'YYYY-MM-DD'), 10000, SYSDATE);

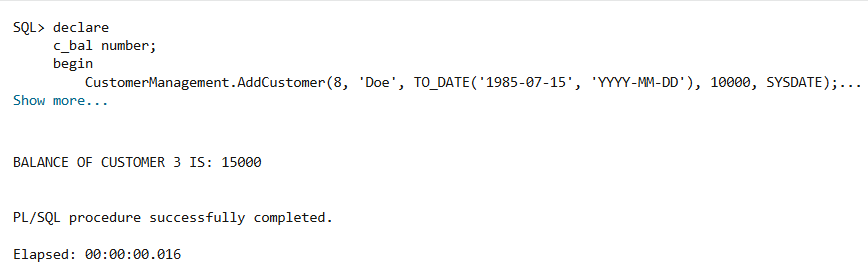
    CustomerManagement.UpdateCustomer(1, 'John Doe', TO\_DATE('1985-05-15', 'YYYY-MM-DD'), 5000);

    c\_bal:=CustomerManagement.GetCustomerBalance(3);

    dbms\_output.put\_line('BALANCE OF CUSTOMER 3 IS: '||c\_bal);

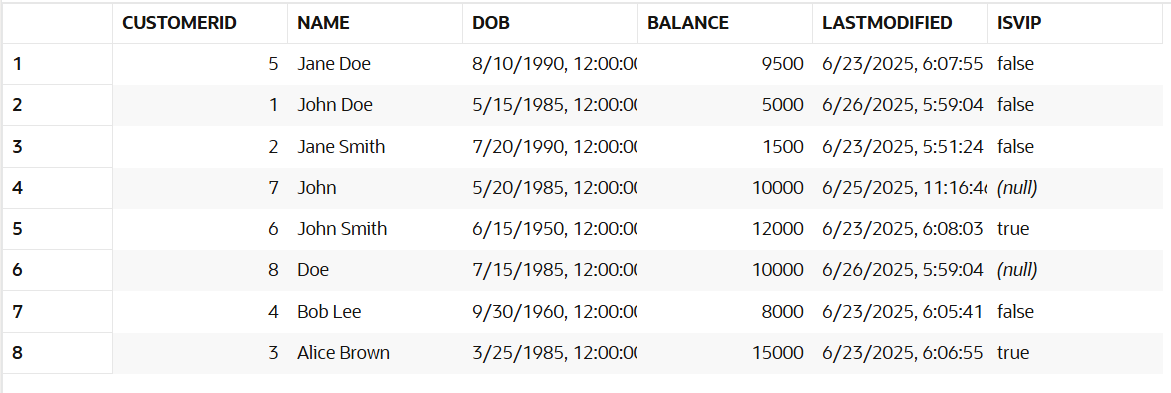
end;

**Output:**



**Query:** select \* from customers;

**Output:**



**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.

**SOLUTION:**

**Query:** select \* from employee;

**Output:**



**Query:**

CREATE OR REPLACE PACKAGE EmployeeManagement AS

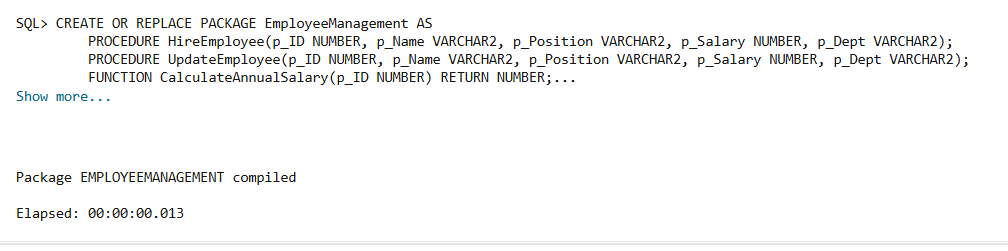
    PROCEDURE HireEmployee(p\_ID NUMBER, p\_Name VARCHAR2, p\_Position VARCHAR2, p\_Salary NUMBER, p\_Dept VARCHAR2);

    PROCEDURE UpdateEmployee(p\_ID NUMBER, p\_Name VARCHAR2, p\_Position VARCHAR2, p\_Salary NUMBER, p\_Dept VARCHAR2);

    FUNCTION CalculateAnnualSalary(p\_ID NUMBER) RETURN NUMBER;

END EmployeeManagement;

**Output:**



**Query:**

CREATE OR REPLACE PACKAGE BODY EmployeeManagement AS

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

    BEGIN

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

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

    END;

    PROCEDURE UpdateEmployee(p\_ID NUMBER, p\_Name VARCHAR2, p\_Position VARCHAR2, p\_Salary NUMBER, p\_Dept VARCHAR2) IS

    BEGIN

        UPDATE Employee

        SET Name = p\_Name,

            Position = p\_Position,

            Salary = p\_Salary,

            Department = p\_Dept

        WHERE EmployeeID = p\_ID;

    END;

    FUNCTION CalculateAnnualSalary(p\_ID NUMBER) RETURN NUMBER IS

        v\_salary NUMBER;

    BEGIN

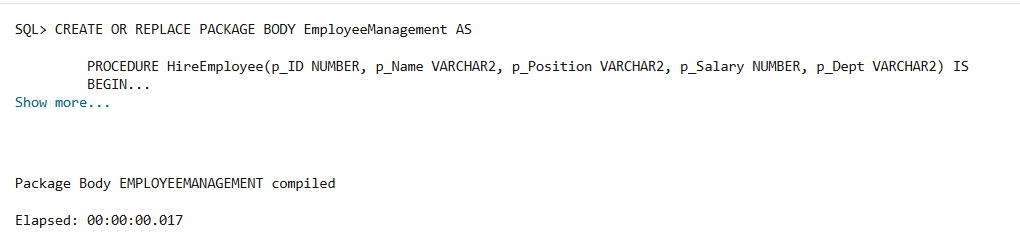
        SELECT Salary \* 12 INTO v\_salary FROM Employee WHERE EmployeeID = p\_ID;

        RETURN v\_salary;

    END;

END EmployeeManagement;

**Output:**



**Query:**

declare

annual\_salary number;

begin

    EmployeeManagement.HireEmployee(3,'John','Manager',50000,'IT');

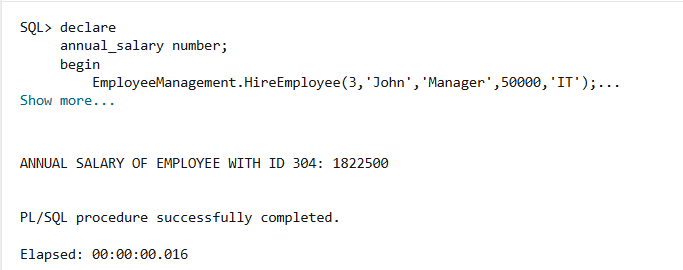
    EmployeeManagement.UpdateEmployee(301,'Michael Scott','Senior Manager',300000,'Sales');

    annual\_salary:=EmployeeManagement.CalculateAnnualSalary(304);

    dbms\_output.put\_line('ANNUAL SALARY OF EMPLOYEE WITH ID 304: '||annual\_salary);

end;

**Output:**



**Query:** select \* from employee;

**Output:**



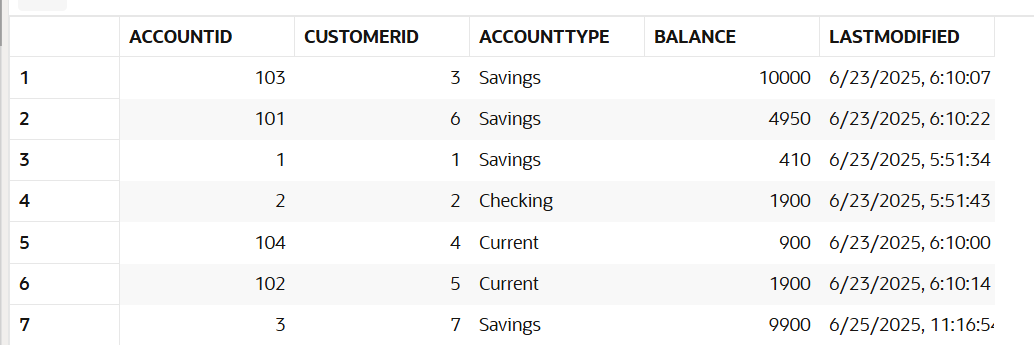
**Scenario 3:** Group all account-related operations into a package.

* + **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.

**SOLUTION:**

**Query:** select \* from accounts;

**Output:**



**Query:**

CREATE OR REPLACE PACKAGE AccountOperations AS

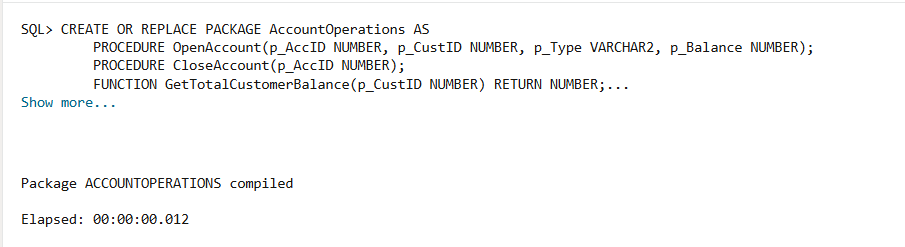
    PROCEDURE OpenAccount(p\_AccID NUMBER, p\_CustID NUMBER, p\_Type VARCHAR2, p\_Balance NUMBER);

    PROCEDURE CloseAccount(p\_AccID NUMBER);

    FUNCTION GetTotalCustomerBalance(p\_CustID NUMBER) RETURN NUMBER;

END AccountOperations;

**Output:**



**Query:**

CREATE OR REPLACE PACKAGE BODY AccountOperations AS

    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 GetTotalCustomerBalance(p\_CustID NUMBER) RETURN NUMBER IS

        v\_total NUMBER := 0;

    BEGIN

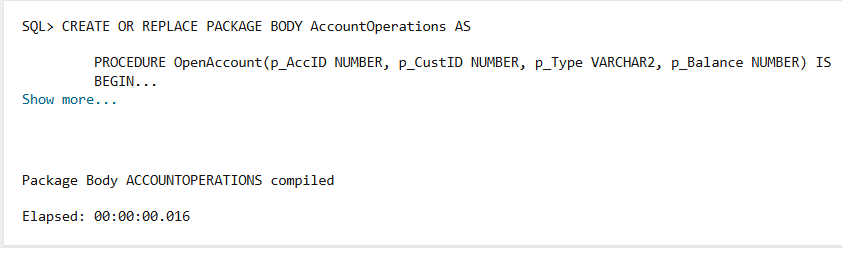
        SELECT NVL(SUM(Balance), 0) INTO v\_total FROM Accounts WHERE CustomerID = p\_CustID;

        RETURN v\_total;

    END;

END AccountOperations;

**Output:**



**Query:**

declare

c\_balance number;

begin

    AccountOperations.OpenAccount(105,8,'checking',2500);

    AccountOperations.CloseAccount(105);

    c\_balance:=AccountOperations.GetTotalCustomerBalance(3);

    dbms\_output.put\_line('BALANCE OF CUSTOMER 3 IS: '||c\_balance);

end;

**Output:**

