PL/SQL Programming

SuperSet ID:6412063

Exercise 7: Packages

Scenario 1:

Package Specification:

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);

FUNCTION GetBalance(p\_id NUMBER) RETURN NUMBER;

END CustomerManagement;

/

Package Body:

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);

DBMS\_OUTPUT.PUT\_LINE('Customer Added: ' || p\_name);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

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

END;

PROCEDURE UpdateCustomer(p\_id NUMBER, p\_name VARCHAR2) IS

BEGIN

UPDATE Customers

SET Name = p\_name

WHERE CustomerID = p\_id;

DBMS\_OUTPUT.PUT\_LINE('Customer Updated: ' || p\_id);

END;

FUNCTION GetBalance(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 -1;

END;

END CustomerManagement;

/

Test Calls:

SET SERVEROUTPUT ON;

BEGIN

CustomerManagement.AddCustomer(201, 'Ravi Kumar', TO\_DATE('1990-06-15', 'YYYY-MM-DD'), 8000);

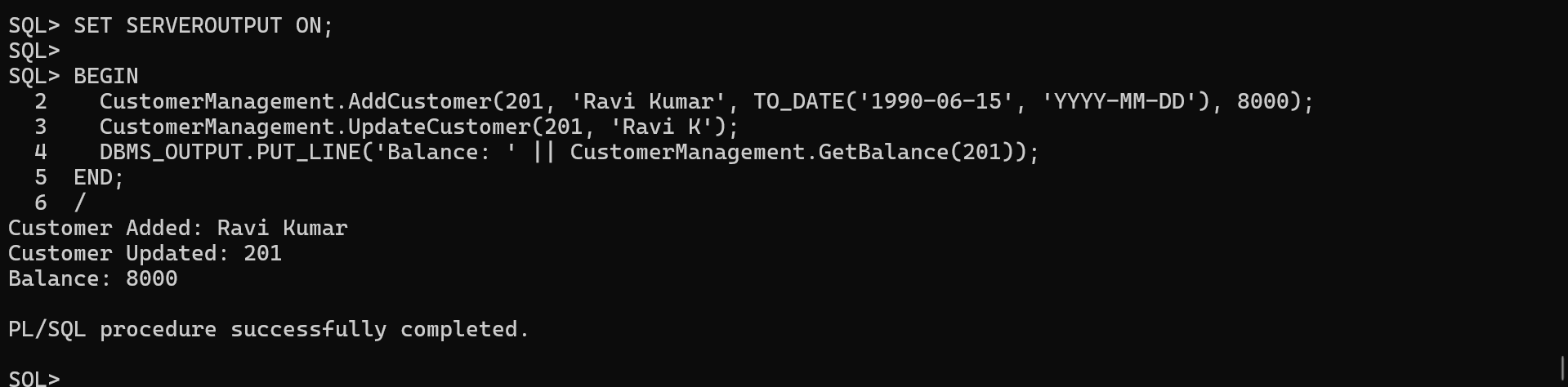
CustomerManagement.UpdateCustomer(201, 'Ravi K');

DBMS\_OUTPUT.PUT\_LINE('Balance: ' || CustomerManagement.GetBalance(201));

END;

/

Output:



Scenario 2:

Package Specification:

CREATE OR REPLACE PACKAGE EmployeeManagement IS

PROCEDURE HireEmployee(p\_id NUMBER, p\_name VARCHAR2, p\_pos VARCHAR2, p\_salary NUMBER, p\_dept VARCHAR2, p\_hire DATE);

PROCEDURE UpdateEmployee(p\_id NUMBER, p\_salary NUMBER);

FUNCTION GetAnnualSalary(p\_id NUMBER) RETURN NUMBER;

END EmployeeManagement;

/

Package Body:

CREATE OR REPLACE PACKAGE BODY EmployeeManagement IS

PROCEDURE HireEmployee(p\_id NUMBER, p\_name VARCHAR2, p\_pos VARCHAR2, p\_salary NUMBER, p\_dept VARCHAR2, p\_hire DATE) IS

BEGIN

INSERT INTO Employees VALUES (p\_id, p\_name, p\_pos, p\_salary, p\_dept, p\_hire);

DBMS\_OUTPUT.PUT\_LINE('Employee Hired: ' || p\_name);

END;

PROCEDURE UpdateEmployee(p\_id NUMBER, p\_salary NUMBER) IS

BEGIN

UPDATE Employees SET Salary = p\_salary WHERE EmployeeID = p\_id;

DBMS\_OUTPUT.PUT\_LINE('Employee Updated: ' || 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;

END;

END EmployeeManagement;

/

Test calls:

BEGIN

EmployeeManagement.HireEmployee(301, 'Sneha Rao', 'Analyst', 40000, 'Finance', TO\_DATE('2023-01-01', 'YYYY-MM-DD'));

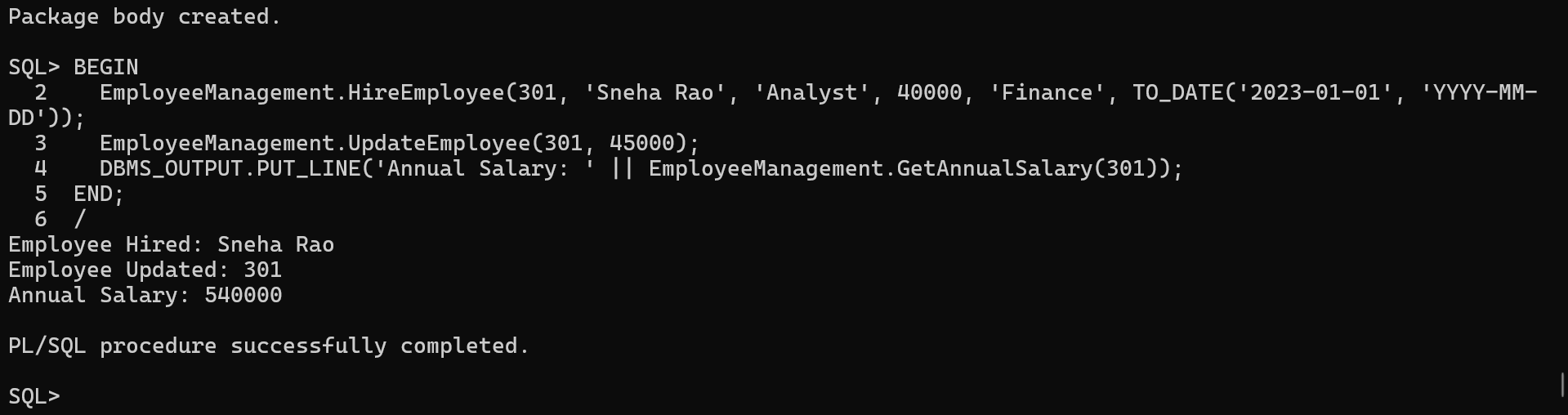
EmployeeManagement.UpdateEmployee(301, 45000);

DBMS\_OUTPUT.PUT\_LINE('Annual Salary: ' || EmployeeManagement.GetAnnualSalary(301));

END;

/

Output:



Scenario 3:

Package Specification:

CREATE OR REPLACE PACKAGE AccountOperations IS

PROCEDURE OpenAccount(p\_acc\_id NUMBER, p\_cust\_id NUMBER, p\_type VARCHAR2, p\_balance NUMBER);

PROCEDURE CloseAccount(p\_acc\_id NUMBER);

FUNCTION GetTotalBalance(p\_cust\_id NUMBER) RETURN NUMBER;

END AccountOperations;

/

Package Body:

CREATE OR REPLACE PACKAGE BODY AccountOperations IS

PROCEDURE OpenAccount(p\_acc\_id NUMBER, p\_cust\_id NUMBER, p\_type VARCHAR2, p\_balance NUMBER) IS

BEGIN

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

VALUES (p\_acc\_id, p\_cust\_id, p\_type, p\_balance, SYSDATE);

DBMS\_OUTPUT.PUT\_LINE('Account Opened: ' || p\_acc\_id);

END;

PROCEDURE CloseAccount(p\_acc\_id NUMBER) IS

BEGIN

DELETE FROM Accounts WHERE AccountID = p\_acc\_id;

DBMS\_OUTPUT.PUT\_LINE('Account Closed: ' || p\_acc\_id);

END;

FUNCTION GetTotalBalance(p\_cust\_id NUMBER) RETURN NUMBER IS

v\_total NUMBER;

BEGIN

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

RETURN NVL(v\_total, 0);

END;

END AccountOperations;

/

Test calls:

BEGIN

AccountOperations.OpenAccount(401, 201, 'Savings', 5000);

DBMS\_OUTPUT.PUT\_LINE('Total Balance: ' || AccountOperations.GetTotalBalance(201));

AccountOperations.CloseAccount(401);

END;

/

Output:

