# WEEK 2: Exercise 3: Stored Procedures

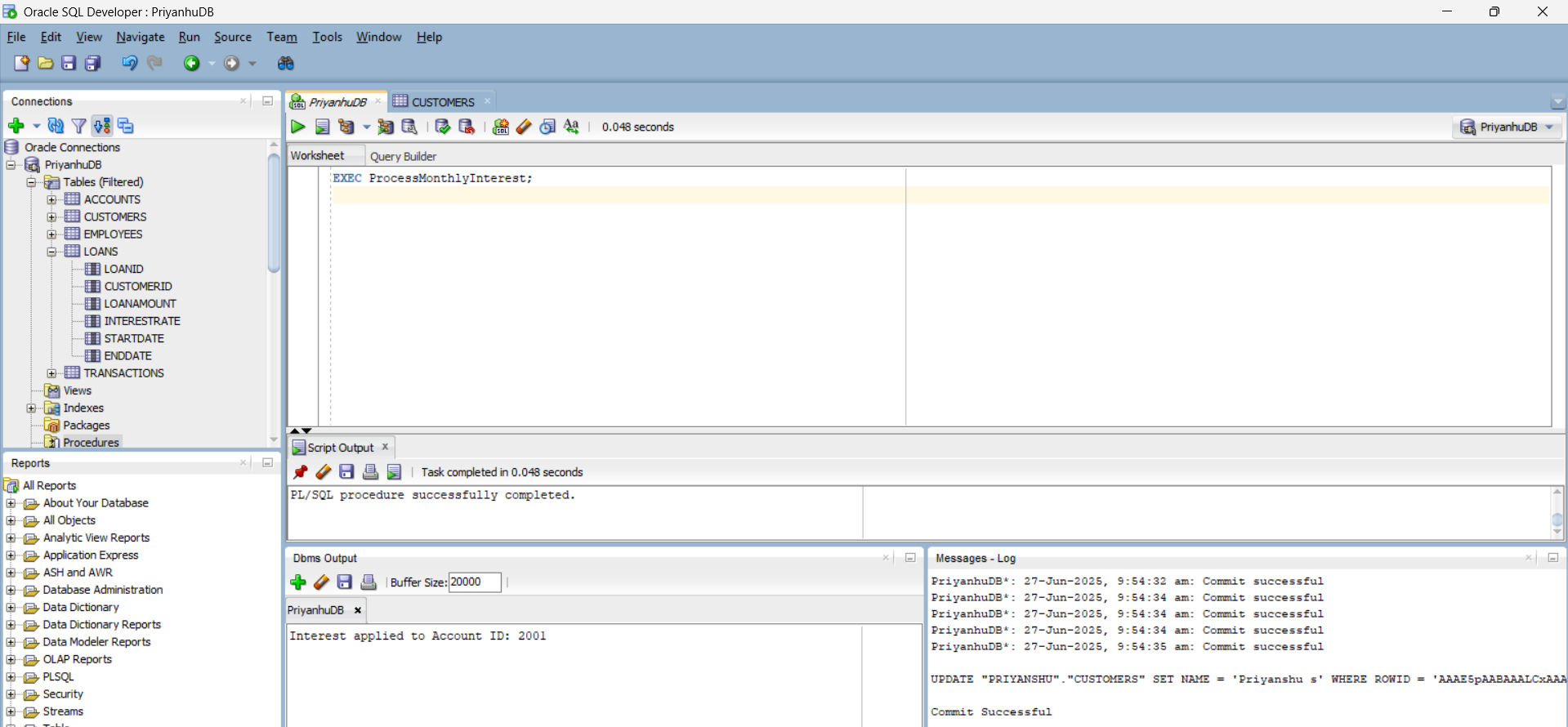
## Scenario 1: Process Monthly Interest

The bank needs to process monthly interest for all savings accounts.  
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.

**QUERY:**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS  
BEGIN  
 FOR acc IN (SELECT AccountID, Balance FROM Accounts WHERE AccountType = 'Savings') LOOP  
 UPDATE Accounts  
 SET Balance = Balance + (acc.Balance \* 0.01)  
 WHERE AccountID = acc.AccountID;  
   
 DBMS\_OUTPUT.PUT\_LINE('Interest applied to Account ID: ' || acc.AccountID);  
 END LOOP;  
   
 COMMIT;  
END;  
/

**OUTPUT:**



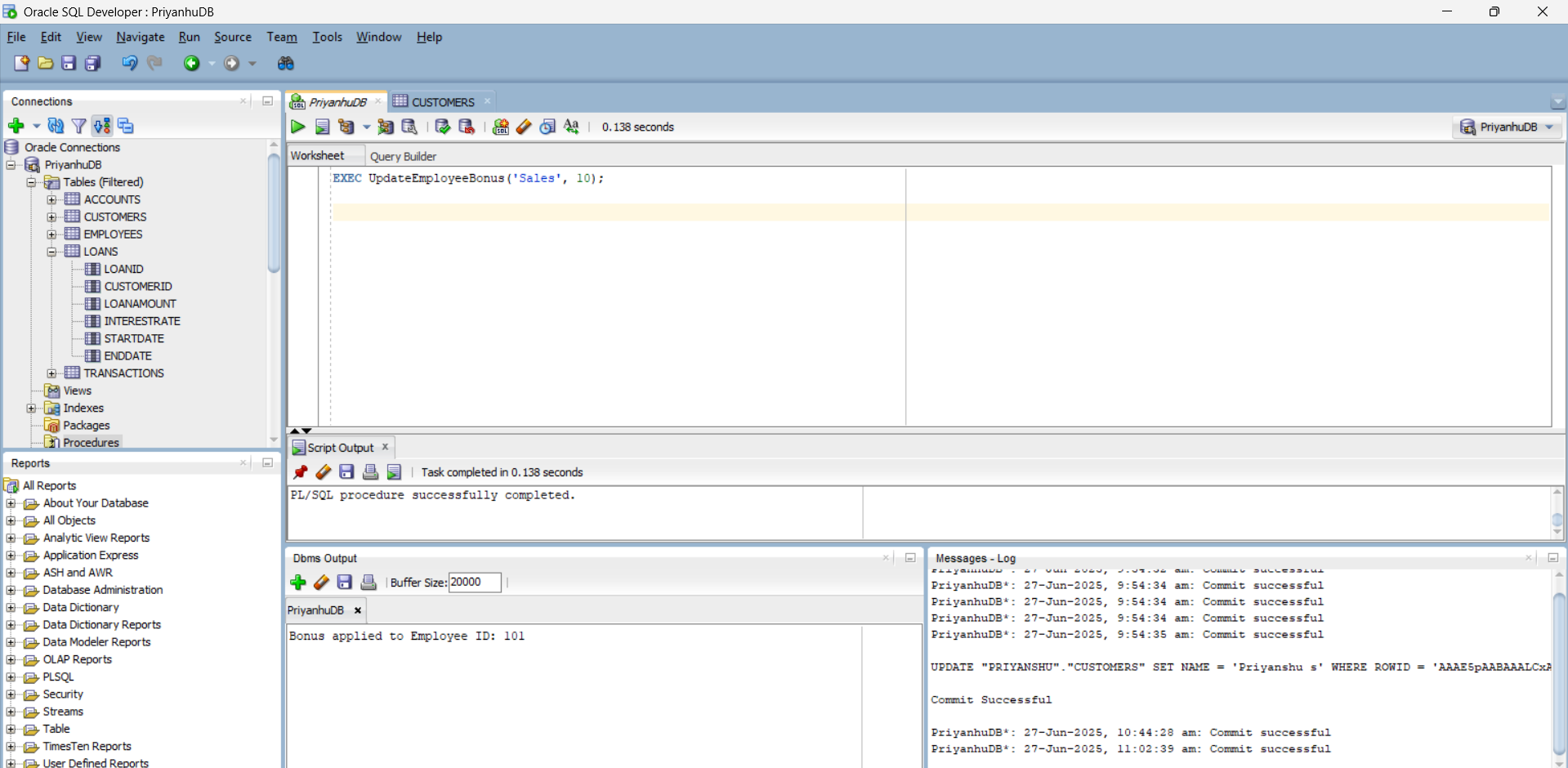
## Scenario 2: Update Employee Bonus

The bank wants to implement a bonus scheme for employees based on their performance.  
Write a stored procedure UpdateEmployeeBonus that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter.

**QUERY:**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(  
 dept\_name IN VARCHAR2,  
 bonus\_percent IN NUMBER  
) IS  
BEGIN  
 FOR emp IN (SELECT EmployeeID, Salary FROM Employees WHERE Department = dept\_name) LOOP  
 UPDATE Employees  
 SET Salary = Salary + (emp.Salary \* bonus\_percent / 100)  
 WHERE EmployeeID = emp.EmployeeID;  
   
 DBMS\_OUTPUT.PUT\_LINE('Bonus applied to Employee ID: ' || emp.EmployeeID);  
 END LOOP;  
   
 COMMIT;  
END;  
/

**OUTPUT:**



## Scenario 3: Transfer Funds Between Accounts

Customers should be able to transfer funds between their accounts.  
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(  
 source\_acc\_id IN NUMBER,  
 dest\_acc\_id IN NUMBER,  
 amount IN NUMBER  
) IS  
 insufficient\_balance EXCEPTION;  
BEGIN  
 DECLARE  
 src\_balance NUMBER;  
 BEGIN  
 SELECT Balance INTO src\_balance FROM Accounts WHERE AccountID = source\_acc\_id FOR UPDATE;  
   
 IF src\_balance < amount THEN  
 RAISE insufficient\_balance;  
 END IF;  
  
 UPDATE Accounts SET Balance = Balance - amount WHERE AccountID = source\_acc\_id;  
 UPDATE Accounts SET Balance = Balance + amount WHERE AccountID = dest\_acc\_id;  
  
 DBMS\_OUTPUT.PUT\_LINE( amount || ' transferred from Account ' || source\_acc\_id || ' to Account ' || dest\_acc\_id);  
 COMMIT;  
 END;  
  
EXCEPTION  
 WHEN insufficient\_balance THEN  
 DBMS\_OUTPUT.PUT\_LINE('Transfer failed: Insufficient balance in Account ' || source\_acc\_id);  
 ROLLBACK;  
END;  
/

**OUTPUT:**

