Module 3 - PL/SQL Programming

Exercise 1: Control Structures

Scenario 1: The bank wants to apply a discount to loan interest rates for customers above 60 years old.

- **Question:** Write a PL/SQL block that loops through all customers, checks their age, and if they are above 60, apply a 1% discount to their current loan interest rates.

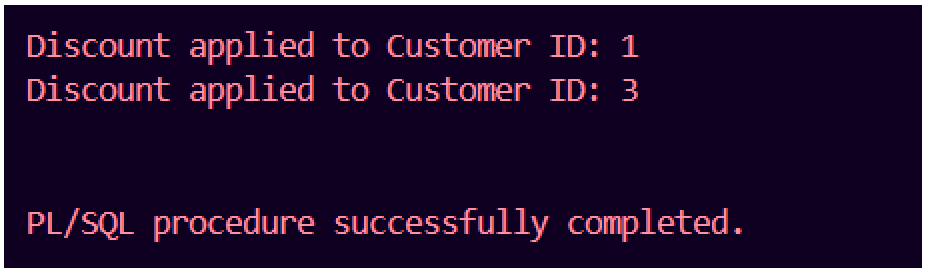
Scenario 2: A customer can be promoted to VIP status based on their balance.

- **Question:** Write a PL/SQL block that iterates through all customers and sets a flag IsVIP to TRUE for those with a balance over $10,000.

Scenario 3: The bank wants to send reminders to customers whose loans are due within the next 30 days.

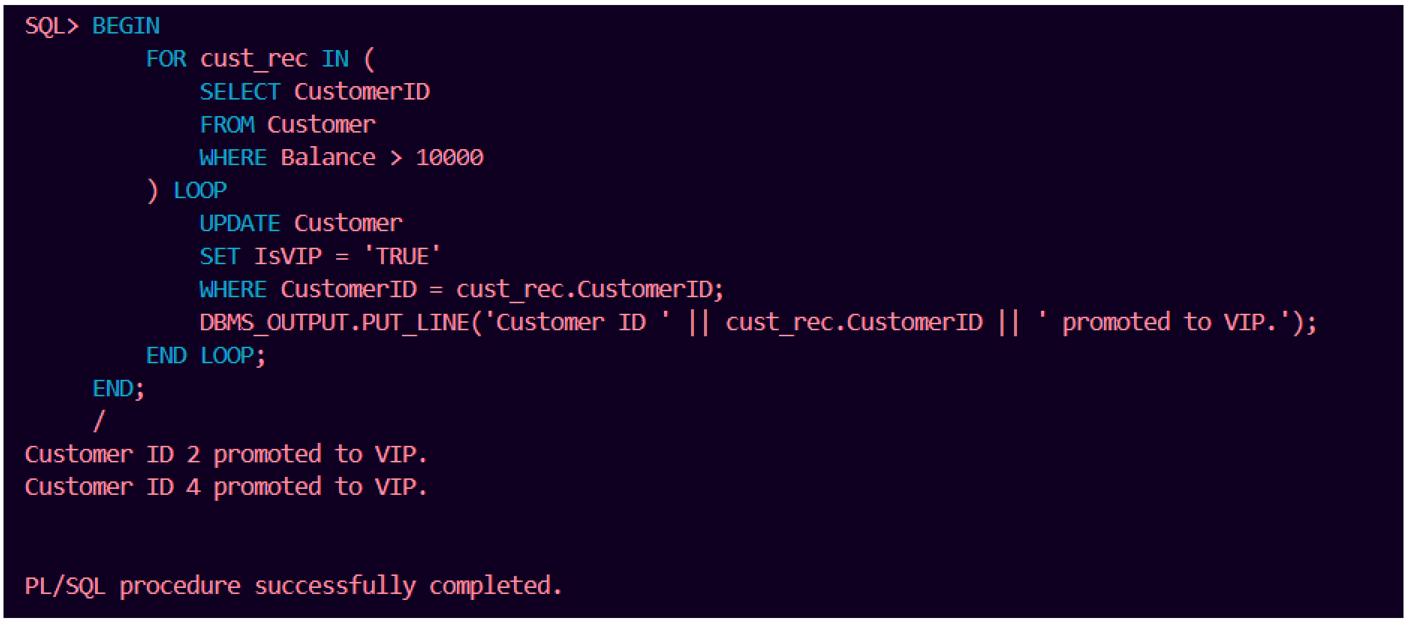
- **Question:** Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer.

SCENARIO 1 OUTPUT



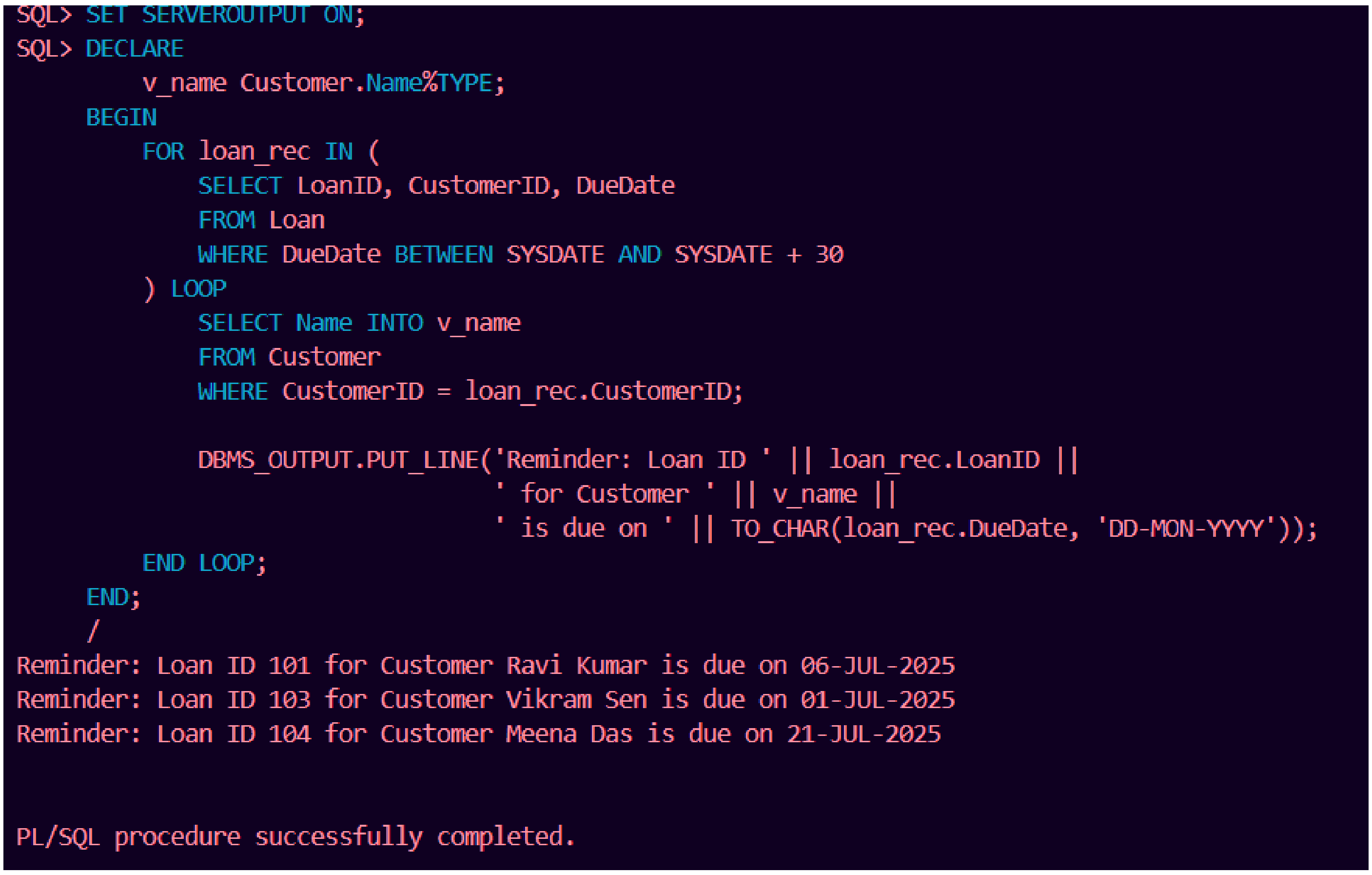
```
Discount applied to Customer ID: 1
Discount applied to Customer ID: 3


PL/SQL procedure successfully completed.
```

SCENARIO 2 OUTPUT



```
SQL> BEGIN
         FOR cust_rec IN (
             SELECT CustomerID
             FROM Customer
             WHERE Balance > 10000
         ) LOOP
             UPDATE Customer
             SET IsVIP = 'TRUE'
             WHERE CustomerID = cust_rec.CustomerID;
             DBMS_OUTPUT.PUT_LINE('Customer ID ' || cust_rec.CustomerID || ' promoted to VIP.');
         END LOOP;
     END;
     /
Customer ID 2 promoted to VIP.
Customer ID 4 promoted to VIP.


PL/SQL procedure successfully completed.
```

SCENARIO 3 OUTPUT



```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
         v_name Customer.Name%TYPE;
     BEGIN
         FOR loan_rec IN (
             SELECT LoanID, CustomerID, DueDate
             FROM Loan
             WHERE DueDate BETWEEN SYSDATE AND SYSDATE + 30
         ) LOOP
             SELECT Name INTO v_name
             FROM Customer
             WHERE CustomerID = loan_rec.CustomerID;

             DBMS_OUTPUT.PUT_LINE('Reminder: Loan ID ' || loan_rec.LoanID ||
                                  ' for Customer ' || v_name ||
                                  ' is due on ' || TO_CHAR(loan_rec.DueDate, 'DD-MON-YYYY'));
         END LOOP;
     END;
     /
Reminder: Loan ID 101 for Customer Ravi Kumar is due on 06-JUL-2025
Reminder: Loan ID 103 for Customer Vikram Sen is due on 01-JUL-2025
Reminder: Loan ID 104 for Customer Meena Das is due on 21-JUL-2025


PL/SQL procedure successfully completed.
```

Exercise 3: Stored Procedures

Scenario 1: The bank needs to process monthly interest for all savings accounts.

- **Question:** 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.

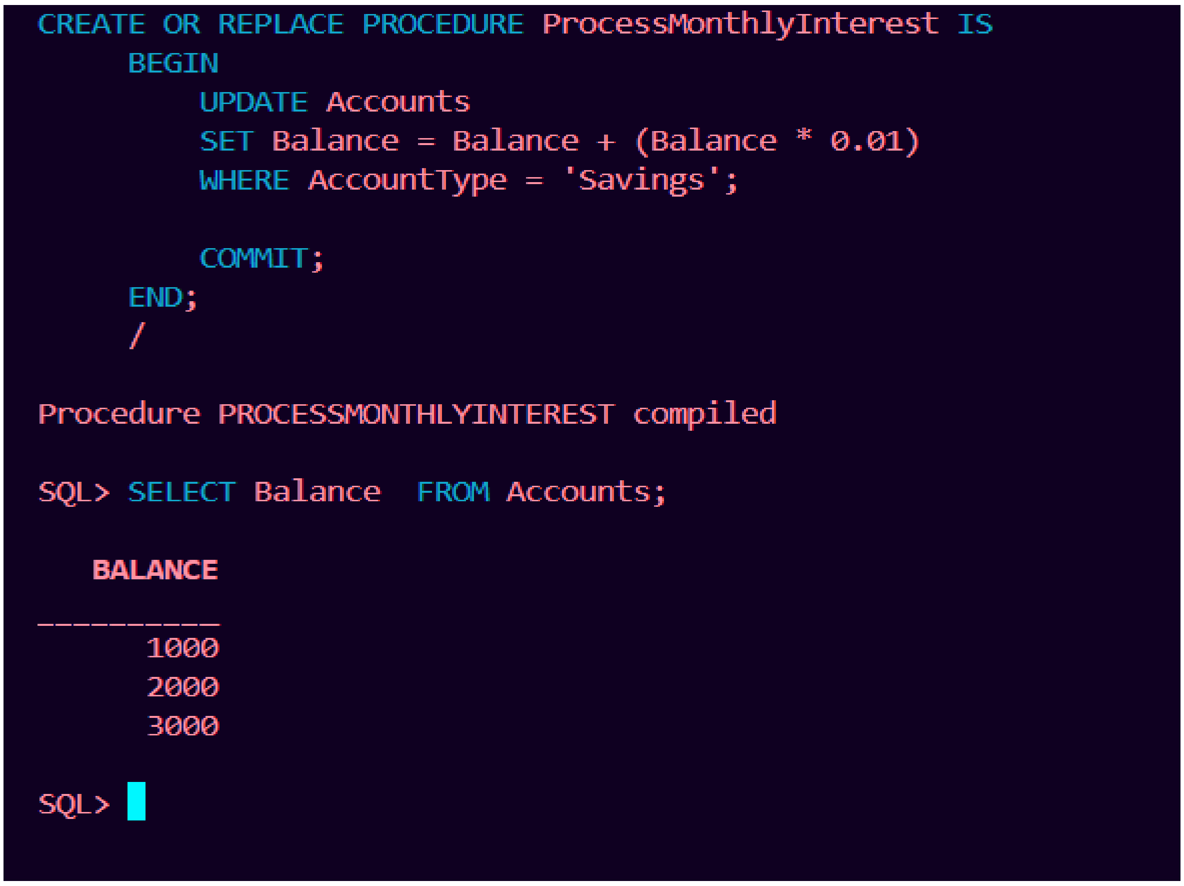
Scenario 2: The bank wants to implement a bonus scheme for employees based on their performance.

- **Question:** Write a stored procedure **UpdateEmployeeBonus** that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter.

Scenario 3: Customers should be able to transfer funds between their accounts.

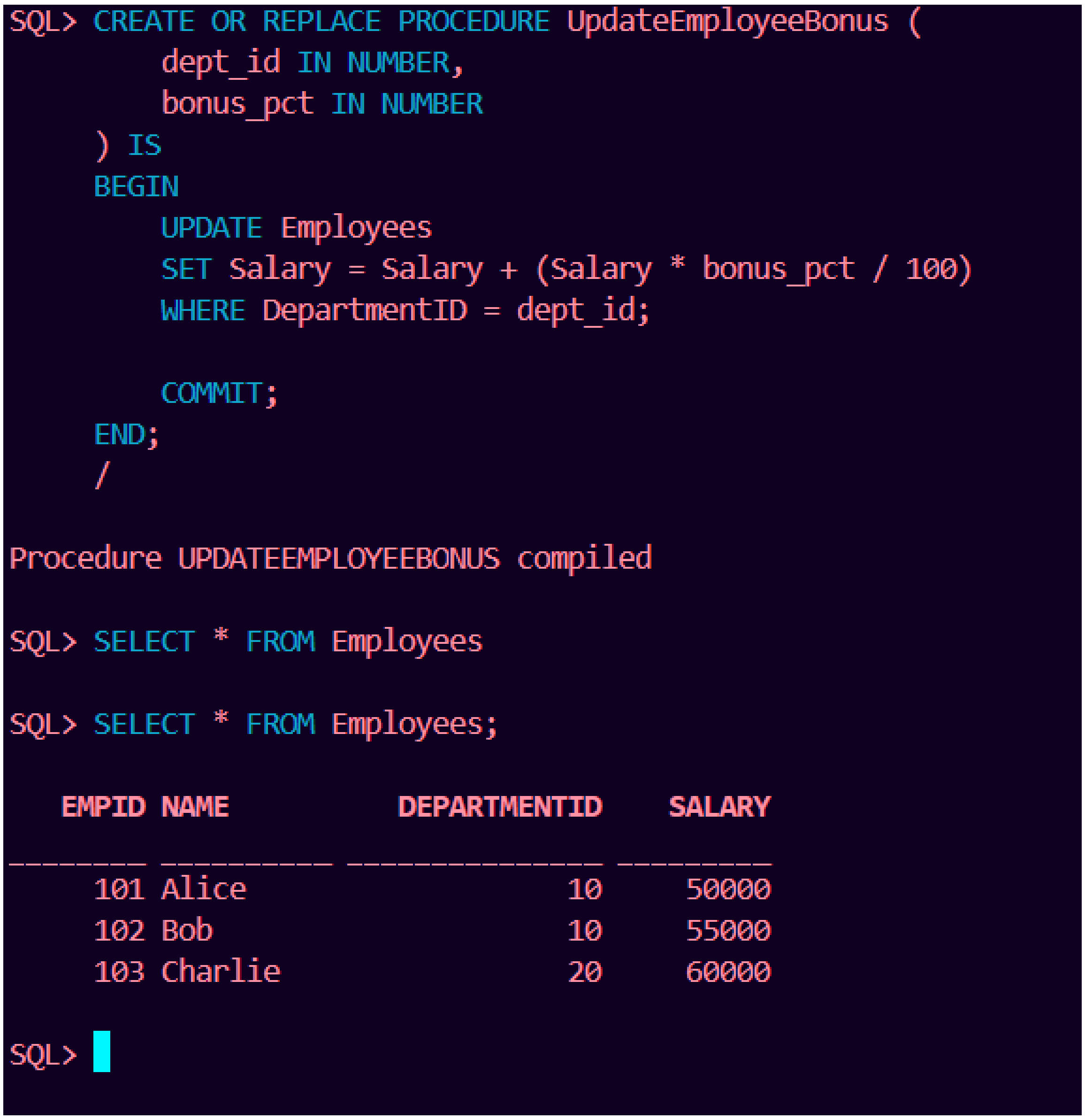
- **Question:** Write a stored procedure **TransferFunds** that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.

SCENARIO 1 OUTPUT



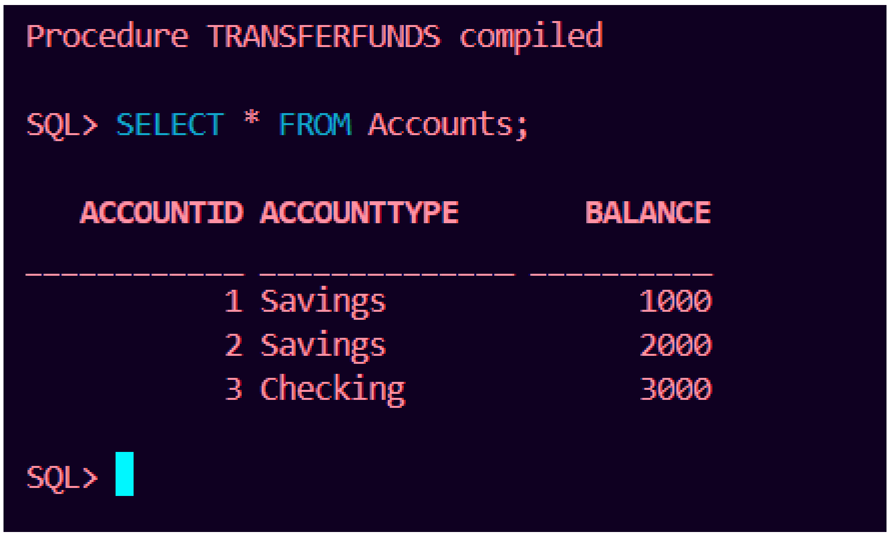
```
CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS
     BEGIN
         UPDATE Accounts
         SET Balance = Balance + (Balance * 0.01)
         WHERE AccountType = 'Savings';

         COMMIT;
     END;
     /

Procedure PROCESSMONTHLYINTEREST compiled

SQL> SELECT Balance  FROM Accounts;

   BALANCE
----------
      1000
      2000
      3000

SQL>
```

<u>**SCENARIO 2 OUTPUT**</u>



```
SQL> CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (
         dept_id IN NUMBER,
         bonus_pct IN NUMBER
     ) IS
     BEGIN
         UPDATE Employees
         SET Salary = Salary + (Salary * bonus_pct / 100)
         WHERE DepartmentID = dept_id;

         COMMIT;
     END;
     /

Procedure UPDATEEMPLOYEEBONUS compiled

SQL> SELECT * FROM Employees

SQL> SELECT * FROM Employees;

   EMPID NAME          DEPARTMENTID     SALARY
________ __________ _______________ _________
     101 Alice                   10      50000
     102 Bob                     10      55000
     103 Charlie                 20      60000

SQL>
```

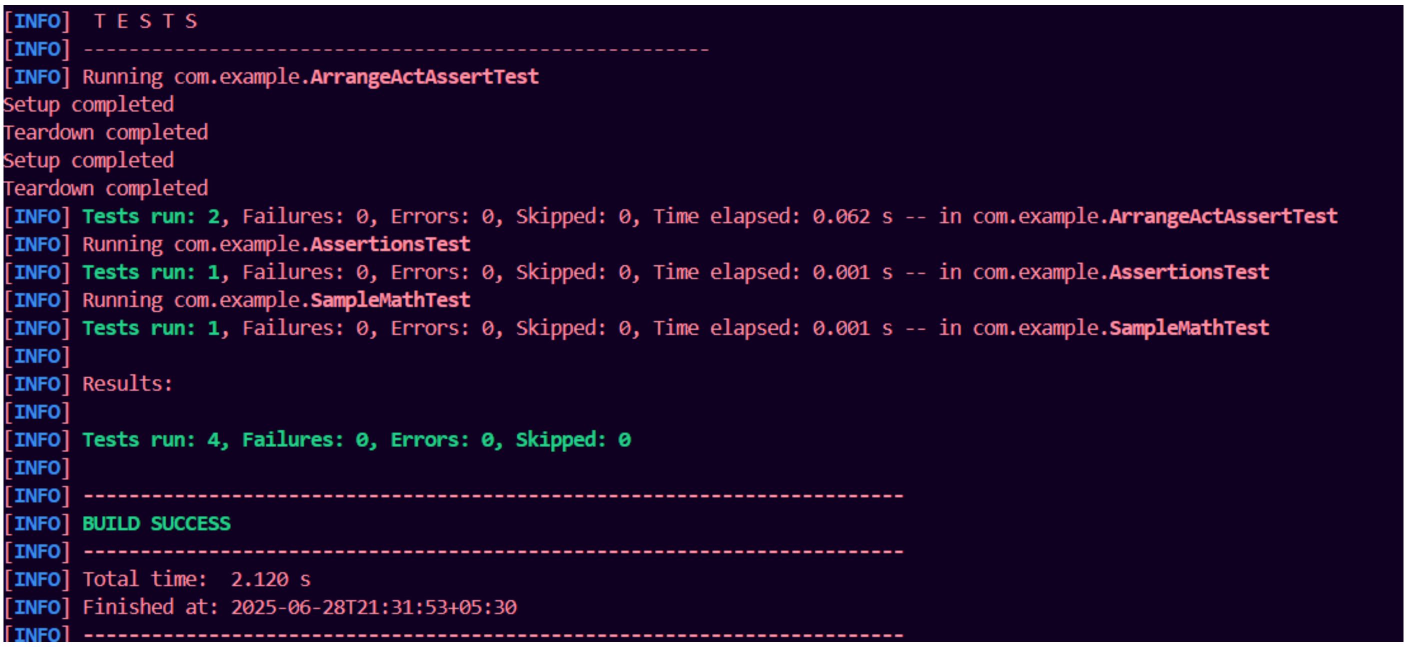
SCENARIO 3 OUTPUT



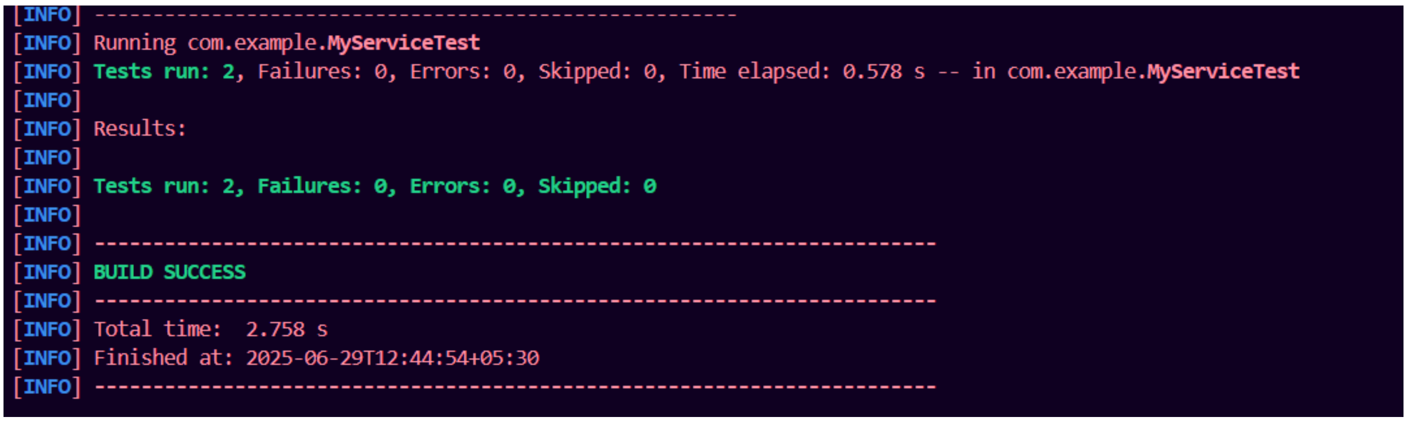
```
Procedure TRANSFERFUNDS compiled

SQL> SELECT * FROM Accounts;

   ACCOUNTID ACCOUNTTYPE          BALANCE
------------ --------------- ----------
           1 Savings               1000
           2 Savings               2000
           3 Checking              3000

SQL> 
```

Module 4 – Test driven development and Logging framework

OUTPUT



```
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running com.example.ArrangeActAssertTest
Setup completed
Teardown completed
Setup completed
Teardown completed
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.062 s -- in com.example.ArrangeActAssertTest
[INFO] Running com.example.AssertionsTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.001 s -- in com.example.AssertionsTest
[INFO] Running com.example.SampleMathTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.001 s -- in com.example.SampleMathTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.120 s
[INFO] Finished at: 2025-06-28T21:31:53+05:30
[INFO] ------------------------------------------------------------------------
```



```
[INFO] -------------------------------------------------------
[INFO] Running com.example.MyServiceTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.578 s -- in com.example.MyServiceTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.758 s
[INFO] Finished at: 2025-06-29T12:44:54+05:30
[INFO] ------------------------------------------------------------------------
```