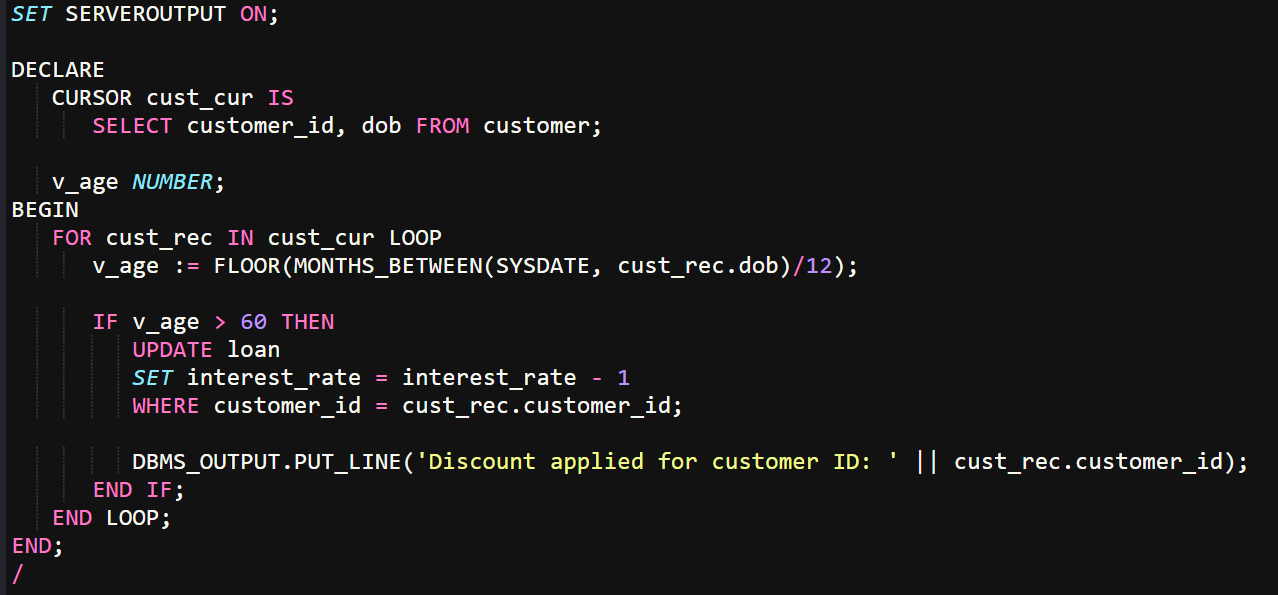
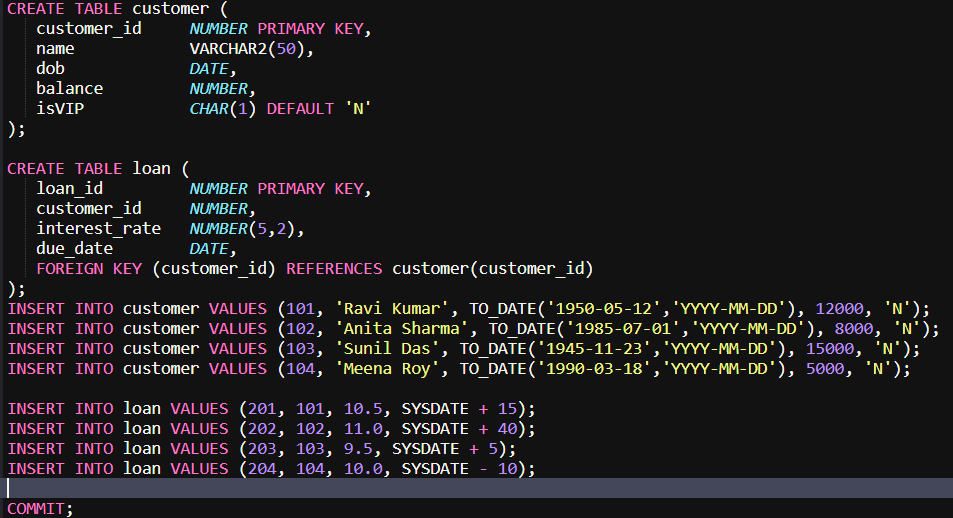
**Exercise 1: Control Structures:-**

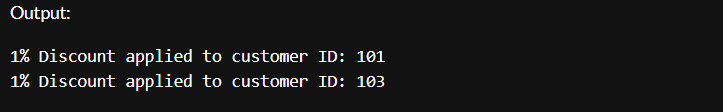
Code:-



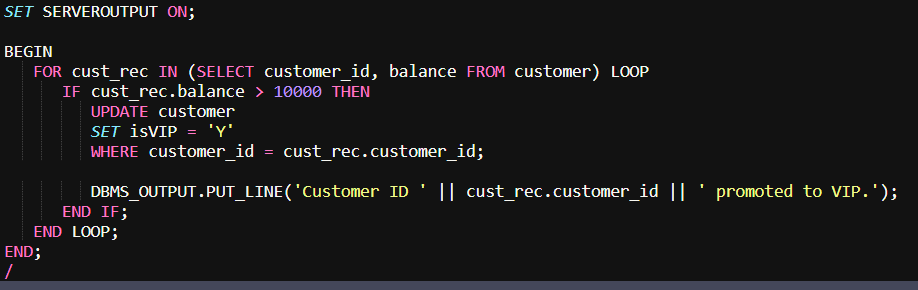
**Database Used to get the desired output:-**

****

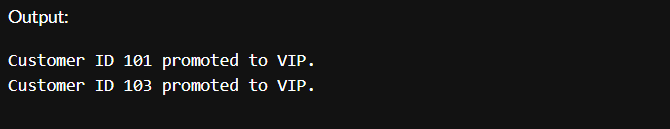
**Output:-**

****

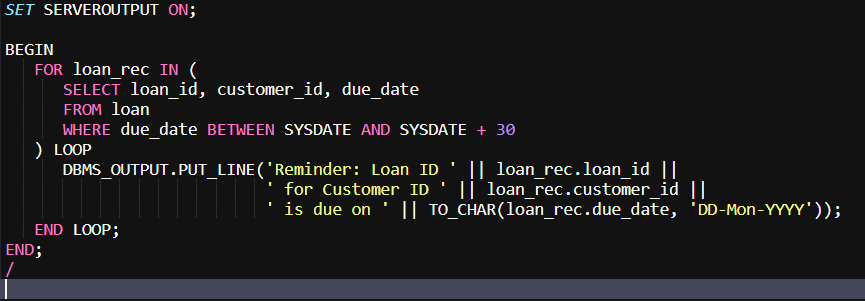
**Scenario 2: Promote Customers with Balance > 10000 to VIP:-**

****

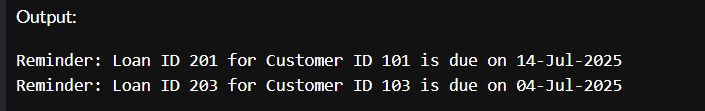
**Outpt:-**

****

**Sending reminders for loans due in next 30 days:-**

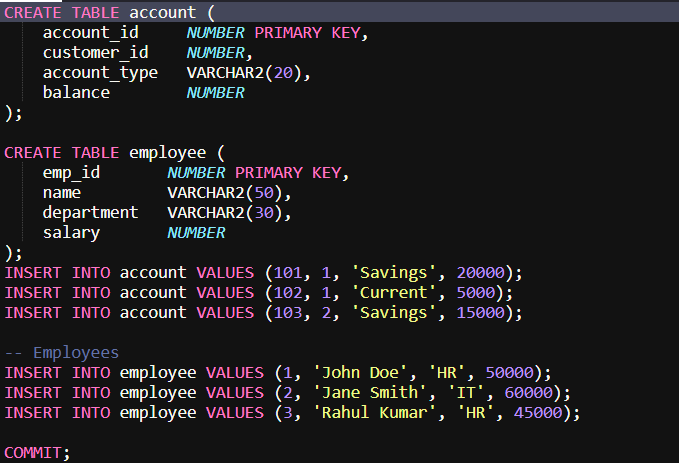
****

**Output:-**

****

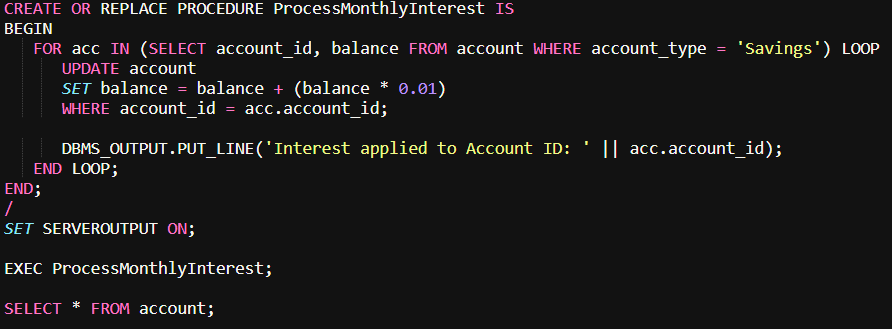
**Exercise 3: Stored Procedures:-**

Database used to check the working:-

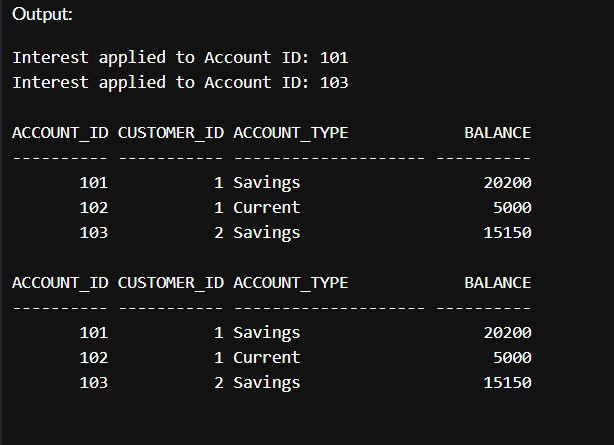


Code:-

Monthly Interest on Savings Accounts:-

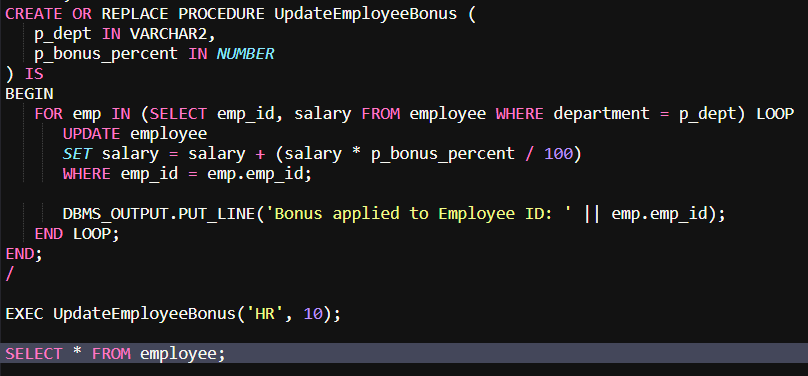


Output:-

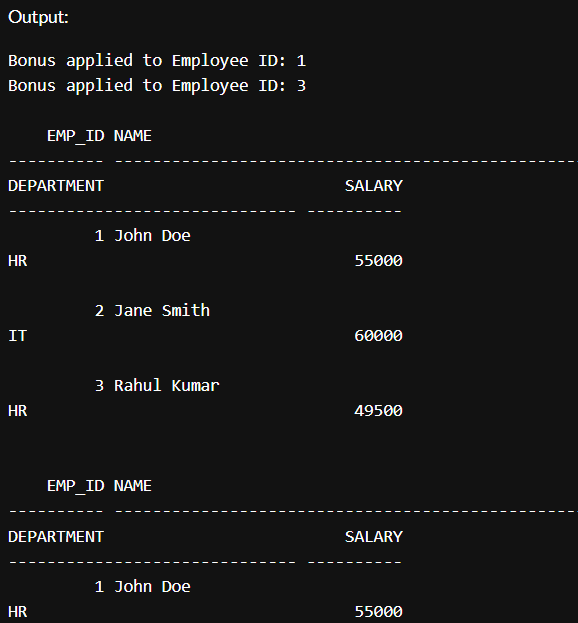


Bonus for Employees Based on Department:-

Code:-

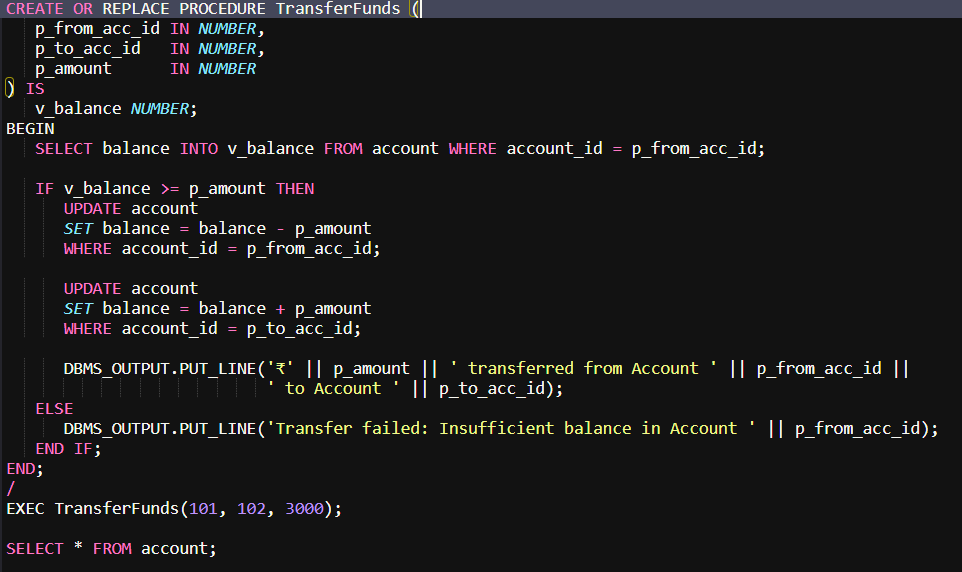


Output:-

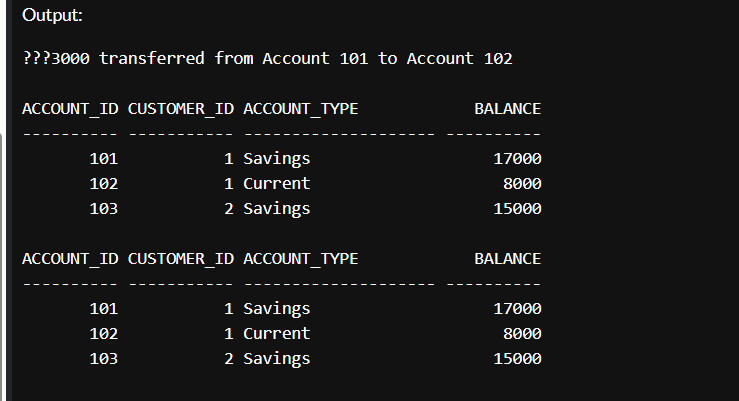


Fund Transfer Between Accounts:-

Code:-



Output:-



Exercise 1: Setting Up Junit:-

Calculator.java:-

package com.example;

public class Calculator {

    public int add(int a, int b) {

        return a + b;

    }

    public int multiply(int a, int b) {

        return a \* b;

    }

}

CalculatorTest.java:-

package com.example;

import org.junit.Test;

import static org.junit.Assert.assertEquals;

public class CalculatorTest {

    @Test

    public void testAdd() {

        Calculator calc = new Calculator();

        assertEquals(5, calc.add(2, 3));

    }

    @Test

    public void testMultiply() {

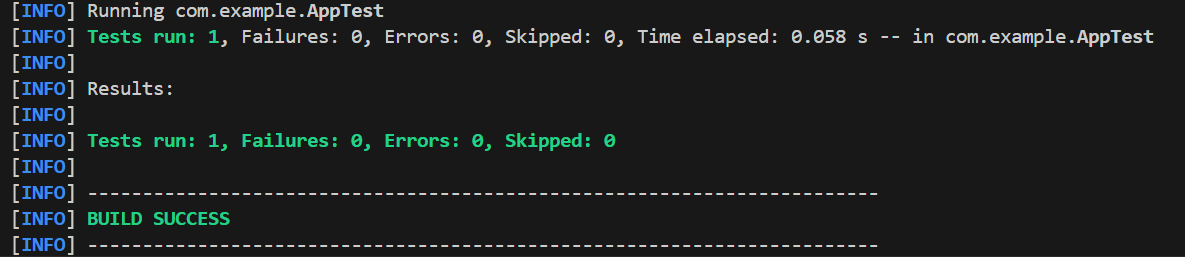
        Calculator calc = new Calculator();

        assertEquals(6, calc.multiply(2, 3));

    }

}

Output:-



Exercise 3: Assertions in Junit:-

Code:-

import org.junit.Test;

import static org.junit.Assert.\*;

public class AssertionsTest {

    @Test

    public void testAssertions() {

        assertEquals(5, 2 + 3);

        assertTrue(5 > 3);

        assertFalse(5 < 3);

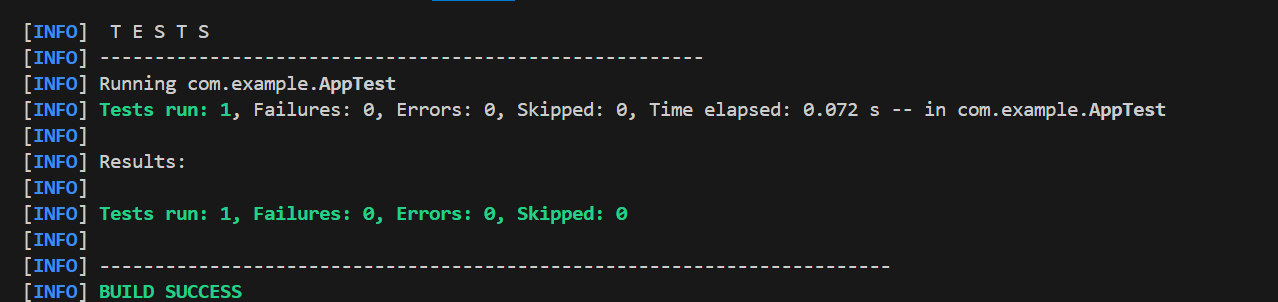
        assertNull(null);

        assertNotNull(new Object());

    }

}

Output:-



**Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and**

**Teardown Methods in Junit:-**

CalculatorTest.java:-

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

    private Calculator calculator;

    // Setup method (runs before each test)

    @Before

    public void setUp() {

        calculator = new Calculator();

        System.out.println("Setting up...");

    }

    // Teardown method (runs after each test)

    @After

    public void tearDown() {

        System.out.println("Cleaning up...");

    }

    @Test

    public void testAddition() {

        // Arrange

        int a = 5;

        int b = 3;

        // Act

        int result = calculator.add(a, b);

        // Assert

        assertEquals(8, result);

    }

    @Test

    public void testSubtraction() {

        // Arrange

        int a = 10;

        int b = 4;

        // Act

        int result = calculator.subtract(a, b);

        // Assert

        assertEquals(6, result);

    }

}

Calculator.java

public class Calculator {

    public int add(int a, int b) {

        return a + b;

    }

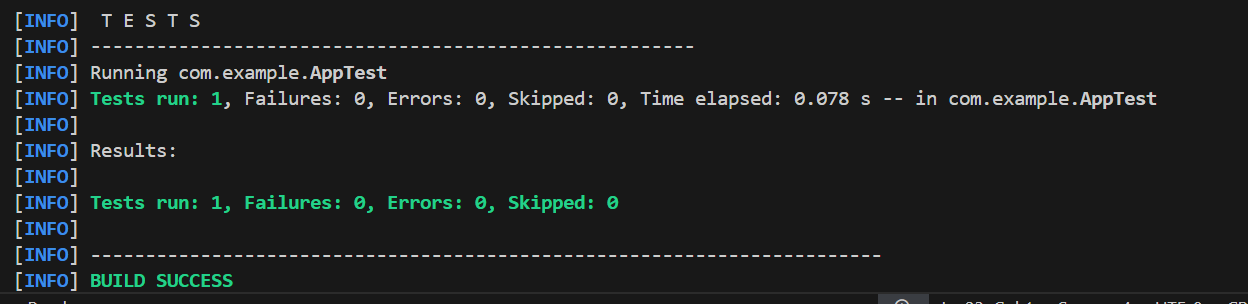
    public int subtract(int a, int b) {

        return a - b;

    }

}

Output:-



Exercise 1: Mocking and Stubbing:-

MyServiceTest.java:-

import static org.mockito.Mockito.\*;

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

interface ExternalApi {

    String getData();

}

class MyService {

    private ExternalApi api;

    public MyService(ExternalApi api) {

        this.api = api;

    }

    public String fetchData() {

        return api.getData();

    }

}

public class MyServiceTest {

    @Test

    public void testExternalApi() {

        ExternalApi mockApi = Mockito.mock(ExternalApi.class);

        when(mockApi.getData()).thenReturn("Mock Data");

        MyService service = new MyService(mockApi);

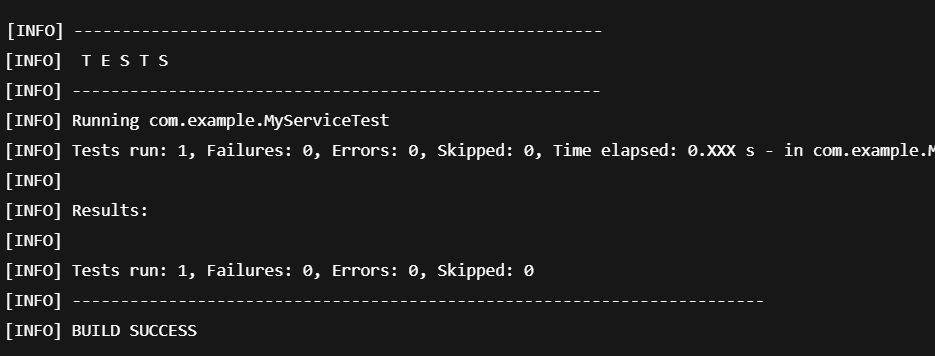
        String result = service.fetchData();

        assertEquals("Mock Data", result);

    }

}

Output:-



**Exercise 2: Verifying Interactions:-**

MyExternalApi.java

package com.example;

public interface ExternalApi {

    String getData();

}

MyService.java

public class MyService {

    private ExternalApi api;

    public MyService(ExternalApi api) {

        this.api = api;

    }

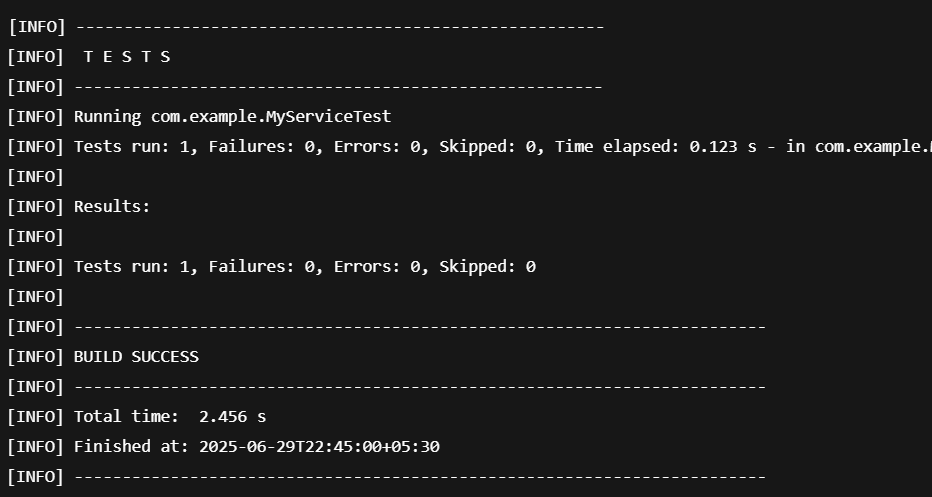
    public String fetchData() {

        return api.getData();

    }

}

Output:-



Exercise 1: Logging Error Messages and Warning Levels:-

Code:-

package com.example;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class LoggingExample {

    private static final Logger logger = LoggerFactory.getLogger(LoggingExample.class);

    public static void main(String[] args) {

        logger.error("This is an error message");

        logger.warn("This is a warning message");

    }

}

Output:-

