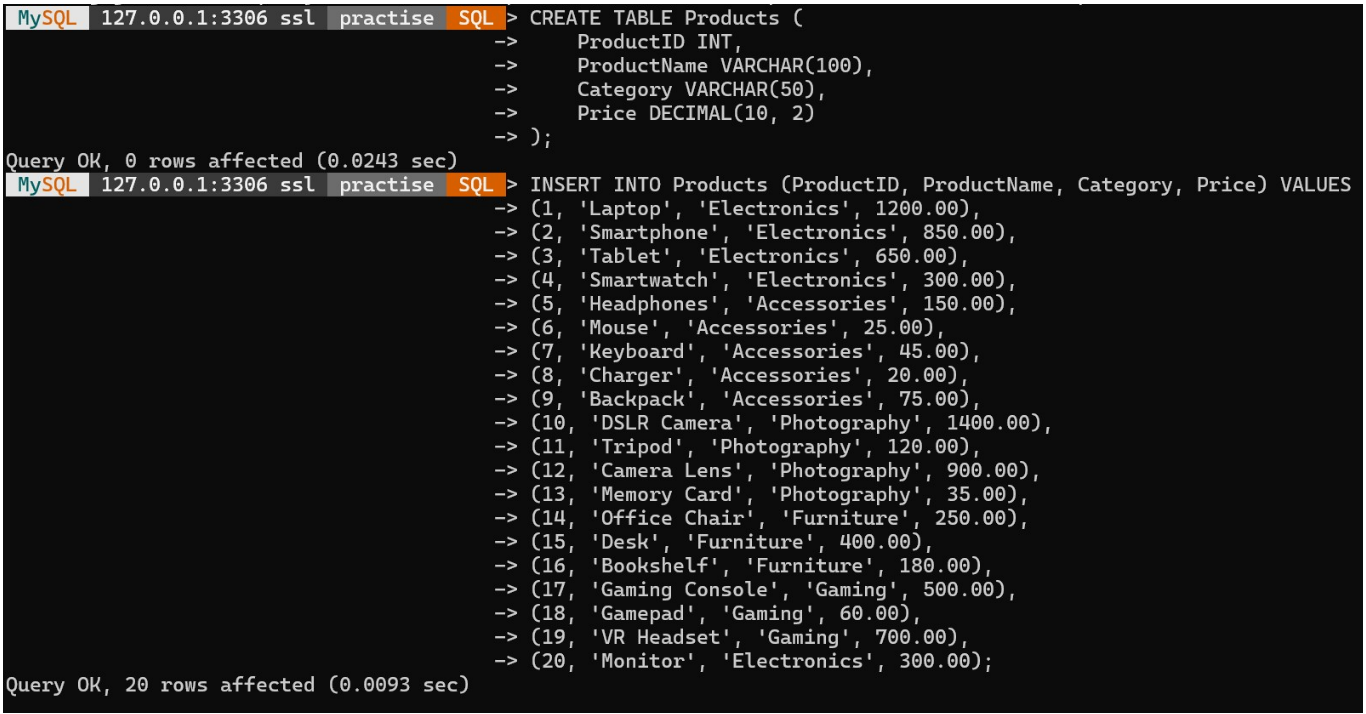
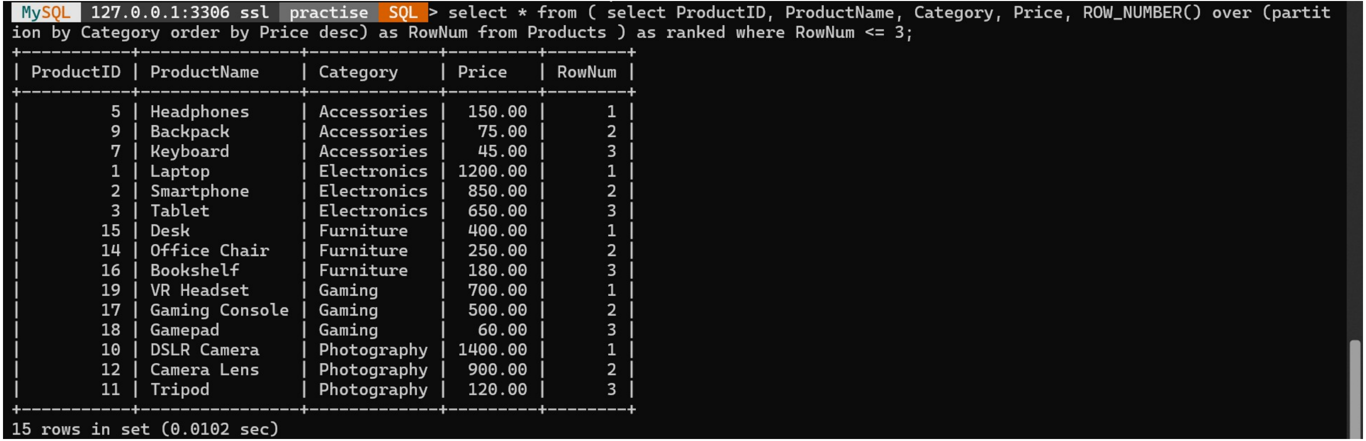
Week 2 Assignment Solutions (Advanced SQL & NUnit and Moq)

Advanced SQL

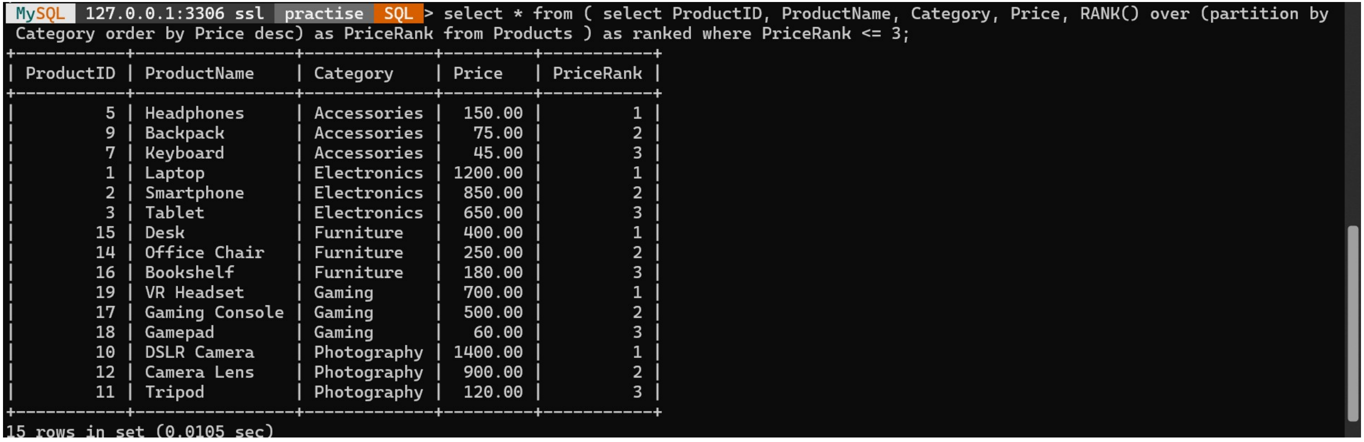
1. SQL Exercise - Advanced concepts (Exercise 1: Ranking and Window Functions)



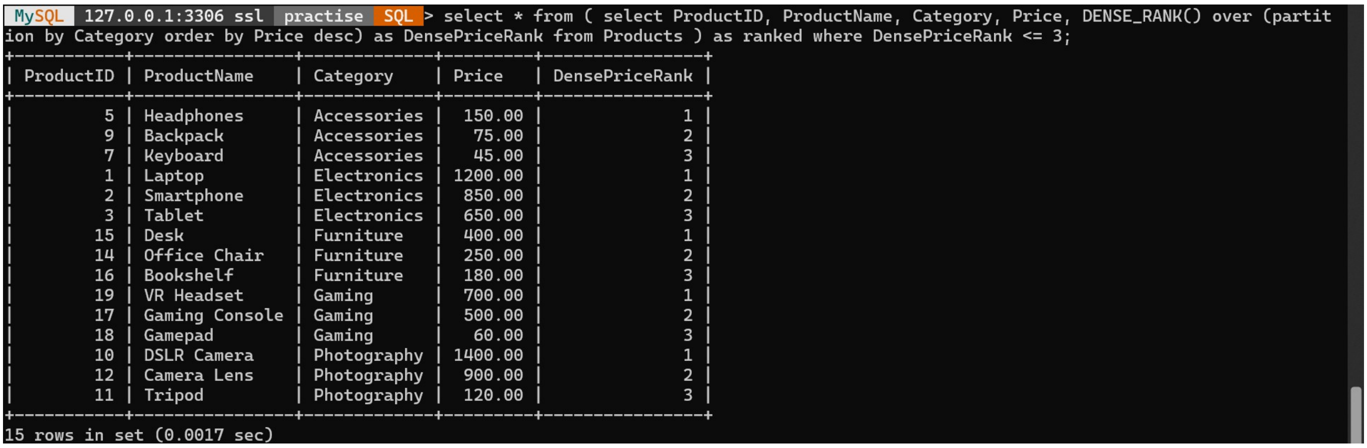
```
 MySQL  127.0.0.1:3306 ssl  practise  SQL > CREATE TABLE Products (
                                        ->     ProductID INT,
                                        ->     ProductName VARCHAR(100),
                                        ->     Category VARCHAR(50),
                                        ->     Price DECIMAL(10, 2)
                                        -> );
Query OK, 0 rows affected (0.0243 sec)
 MySQL  127.0.0.1:3306 ssl  practise  SQL > INSERT INTO Products (ProductID, ProductName, Category, Price) VALUES
                                        -> (1, 'Laptop', 'Electronics', 1200.00),
                                        -> (2, 'Smartphone', 'Electronics', 850.00),
                                        -> (3, 'Tablet', 'Electronics', 650.00),
                                        -> (4, 'Smartwatch', 'Electronics', 300.00),
                                        -> (5, 'Headphones', 'Accessories', 150.00),
                                        -> (6, 'Mouse', 'Accessories', 25.00),
                                        -> (7, 'Keyboard', 'Accessories', 45.00),
                                        -> (8, 'Charger', 'Accessories', 20.00),
                                        -> (9, 'Backpack', 'Accessories', 75.00),
                                        -> (10, 'DSLR Camera', 'Photography', 1400.00),
                                        -> (11, 'Tripod', 'Photography', 120.00),
                                        -> (12, 'Camera Lens', 'Photography', 900.00),
                                        -> (13, 'Memory Card', 'Photography', 35.00),
                                        -> (14, 'Office Chair', 'Furniture', 250.00),
                                        -> (15, 'Desk', 'Furniture', 400.00),
                                        -> (16, 'Bookshelf', 'Furniture', 180.00),
                                        -> (17, 'Gaming Console', 'Gaming', 500.00),
                                        -> (18, 'Gamepad', 'Gaming', 60.00),
                                        -> (19, 'VR Headset', 'Gaming', 700.00),
                                        -> (20, 'Monitor', 'Electronics', 300.00);
Query OK, 20 rows affected (0.0093 sec)
```



```
 MySQL  127.0.0.1:3306 ssl  practise  SQL > select * from ( select ProductID, ProductName, Category, Price, ROW_NUMBER() over (partit
ion by Category order by Price desc) as RowNum from Products ) as ranked where RowNum <= 3;
+-----------+----------------+-------------+---------+--------+
| ProductID | ProductName    | Category    | Price   | RowNum |
+-----------+----------------+-------------+---------+--------+
|         5 | Headphones     | Accessories |  150.00 |      1 |
|         9 | Backpack       | Accessories |   75.00 |      2 |
|         7 | Keyboard       | Accessories |   45.00 |      3 |
|         1 | Laptop         | Electronics | 1200.00 |      1 |
|         2 | Smartphone     | Electronics |  850.00 |      2 |
|         3 | Tablet         | Electronics |  650.00 |      3 |
|        15 | Desk           | Furniture   |  400.00 |      1 |
|        14 | Office Chair   | Furniture   |  250.00 |      2 |
|        16 | Bookshelf      | Furniture   |  180.00 |      3 |
|        19 | VR Headset     | Gaming      |  700.00 |      1 |
|        17 | Gaming Console | Gaming      |  500.00 |      2 |
|        18 | Gamepad        | Gaming      |   60.00 |      3 |
|        10 | DSLR Camera    | Photography | 1400.00 |      1 |
|        12 | Camera Lens    | Photography |  900.00 |      2 |
|        11 | Tripod         | Photography |  120.00 |      3 |
+-----------+----------------+-------------+---------+--------+
15 rows in set (0.0102 sec)
```

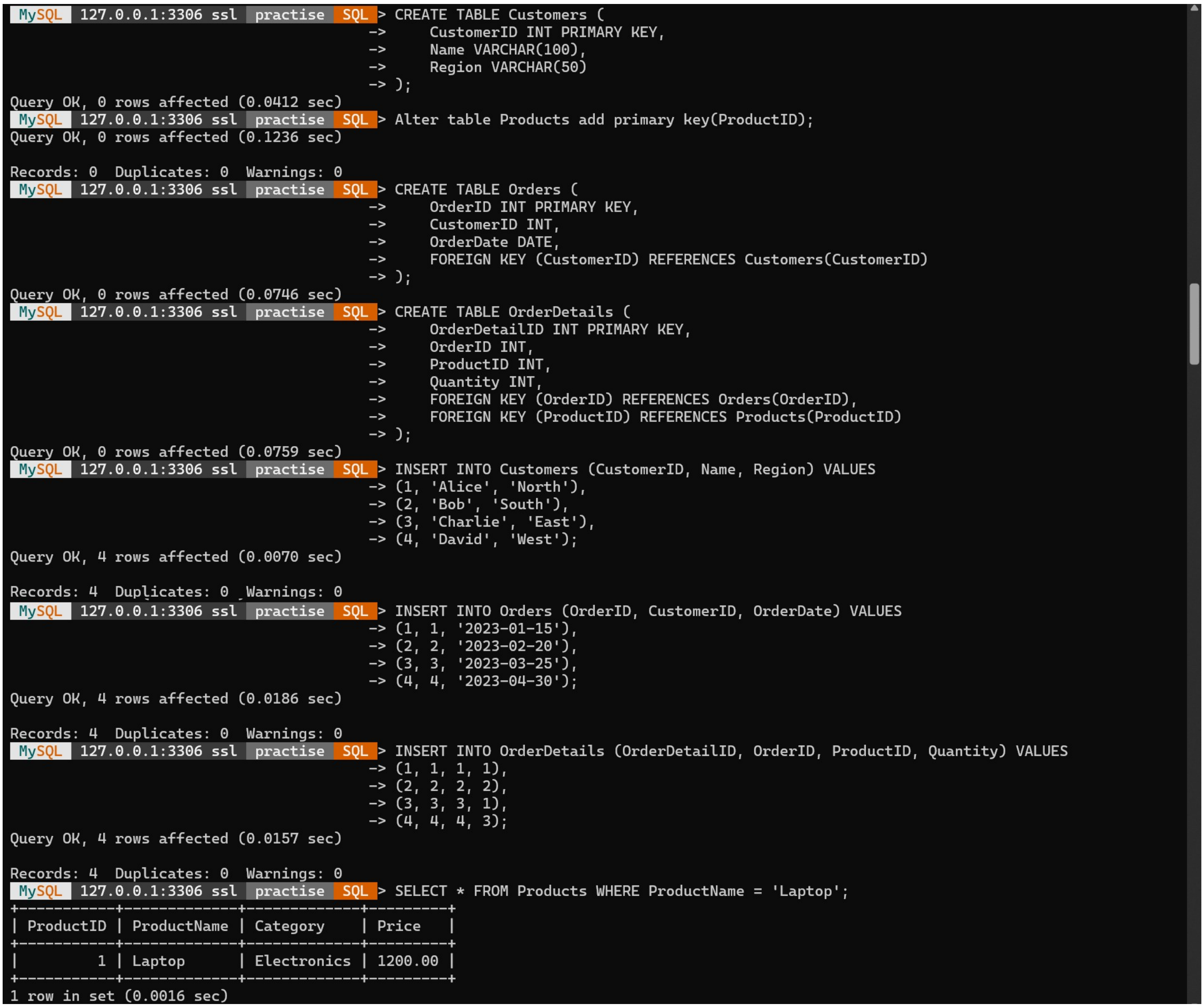


```
 MySQL  127.0.0.1:3306 ssl  practise  SQL > select * from ( select ProductID, ProductName, Category, Price, RANK() over (partition by
 Category order by Price desc) as PriceRank from Products ) as ranked where PriceRank <= 3;
+-----------+----------------+-------------+---------+-----------+
| ProductID | ProductName    | Category    | Price   | PriceRank |
+-----------+----------------+-------------+---------+-----------+
|         5 | Headphones     | Accessories |  150.00 |         1 |
|         9 | Backpack       | Accessories |   75.00 |         2 |
|         7 | Keyboard       | Accessories |   45.00 |         3 |
|         1 | Laptop         | Electronics | 1200.00 |         1 |
|         2 | Smartphone     | Electronics |  850.00 |         2 |
|         3 | Tablet         | Electronics |  650.00 |         3 |
|        15 | Desk           | Furniture   |  400.00 |         1 |
|        14 | Office Chair   | Furniture   |  250.00 |         2 |
|        16 | Bookshelf      | Furniture   |  180.00 |         3 |
|        19 | VR Headset     | Gaming      |  700.00 |         1 |
|        17 | Gaming Console | Gaming      |  500.00 |         2 |
|        18 | Gamepad        | Gaming      |   60.00 |         3 |
|        10 | DSLR Camera    | Photography | 1400.00 |         1 |
|        12 | Camera Lens    | Photography |  900.00 |         2 |
|        11 | Tripod         | Photography |  120.00 |         3 |
+-----------+----------------+-------------+---------+-----------+
15 rows in set (0.0105 sec)
```

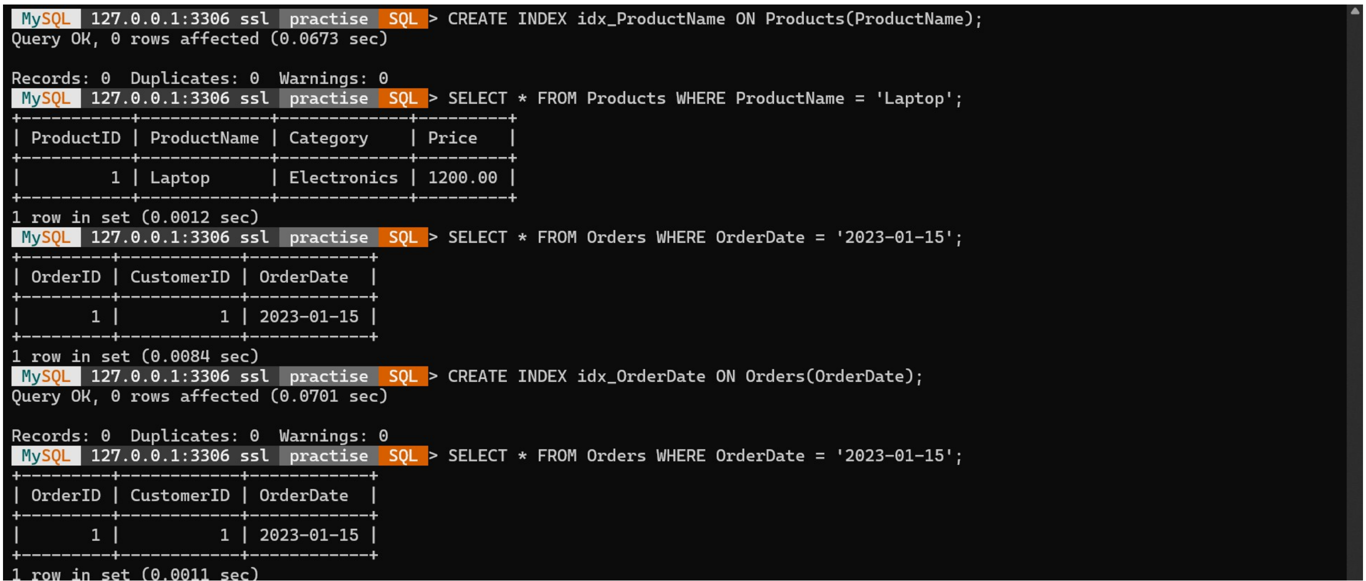


```
 MySQL  127.0.0.1:3306 ssl  practise  SQL > select * from ( select ProductID, ProductName, Category, Price, DENSE_RANK() over (partit
ion by Category order by Price desc) as DensePriceRank from Products ) as ranked where DensePriceRank <= 3;
+-----------+----------------+-------------+---------+----------------+
| ProductID | ProductName    | Category    | Price   | DensePriceRank |
+-----------+----------------+-------------+---------+----------------+
|         5 | Headphones     | Accessories |  150.00 |              1 |
|         9 | Backpack       | Accessories |   75.00 |              2 |
|         7 | Keyboard       | Accessories |   45.00 |              3 |
|         1 | Laptop         | Electronics | 1200.00 |              1 |
|         2 | Smartphone     | Electronics |  850.00 |              2 |
|         3 | Tablet         | Electronics |  650.00 |              3 |
|        15 | Desk           | Furniture   |  400.00 |              1 |
|        14 | Office Chair   | Furniture   |  250.00 |              2 |
|        16 | Bookshelf      | Furniture   |  180.00 |              3 |
|        19 | VR Headset     | Gaming      |  700.00 |              1 |
|        17 | Gaming Console | Gaming      |  500.00 |              2 |
|        18 | Gamepad        | Gaming      |   60.00 |              3 |
|        10 | DSLR Camera    | Photography | 1400.00 |              1 |
|        12 | Camera Lens    | Photography |  900.00 |              2 |
|        11 | Tripod         | Photography |  120.00 |              3 |
+-----------+----------------+-------------+---------+----------------+
15 rows in set (0.0017 sec)
```

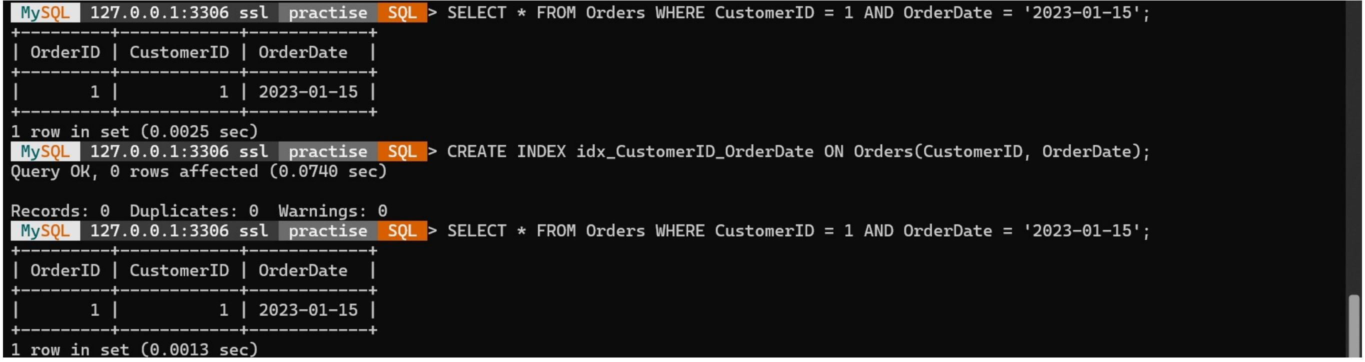
2. SQL Exercise - Index (Hands-on in this document)



```
 MySQL  127.0.0.1:3306 ssl  practise  SQL > CREATE TABLE Customers (
                                         ->     CustomerID INT PRIMARY KEY,
                                         ->     Name VARCHAR(100),
                                         ->     Region VARCHAR(50)
                                         -> );
Query OK, 0 rows affected (0.0412 sec)
 MySQL  127.0.0.1:3306 ssl  practise  SQL > Alter table Products add primary key(ProductID);
Query OK, 0 rows affected (0.1236 sec)

Records: 0  Duplicates: 0  Warnings: 0
 MySQL  127.0.0.1:3306 ssl  practise  SQL > CREATE TABLE Orders (
                                         ->     OrderID INT PRIMARY KEY,
                                         ->     CustomerID INT,
                                         ->     OrderDate DATE,
                                         ->     FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
                                         -> );
Query OK, 0 rows affected (0.0746 sec)
 MySQL  127.0.0.1:3306 ssl  practise  SQL > CREATE TABLE OrderDetails (
                                         ->     OrderDetailID INT PRIMARY KEY,
                                         ->     OrderID INT,
                                         ->     ProductID INT,
                                         ->     Quantity INT,
                                         ->     FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
                                         ->     FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
                                         -> );
Query OK, 0 rows affected (0.0759 sec)
 MySQL  127.0.0.1:3306 ssl  practise  SQL > INSERT INTO Customers (CustomerID, Name, Region) VALUES
                                         -> (1, 'Alice', 'North'),
                                         -> (2, 'Bob', 'South'),
                                         -> (3, 'Charlie', 'East'),
                                         -> (4, 'David', 'West');
Query OK, 4 rows affected (0.0070 sec)

Records: 4  Duplicates: 0  Warnings: 0
 MySQL  127.0.0.1:3306 ssl  practise  SQL > INSERT INTO Orders (OrderID, CustomerID, OrderDate) VALUES
                                         -> (1, 1, '2023-01-15'),
                                         -> (2, 2, '2023-02-20'),
                                         -> (3, 3, '2023-03-25'),
                                         -> (4, 4, '2023-04-30');
Query OK, 4 rows affected (0.0186 sec)

Records: 4  Duplicates: 0  Warnings: 0
 MySQL  127.0.0.1:3306 ssl  practise  SQL > INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity) VALUES
                                         -> (1, 1, 1, 1),
                                         -> (2, 2, 2, 2),
                                         -> (3, 3, 3, 1),
                                         -> (4, 4, 4, 3);
Query OK, 4 rows affected (0.0157 sec)

Records: 4  Duplicates: 0  Warnings: 0
 MySQL  127.0.0.1:3306 ssl  practise  SQL > SELECT * FROM Products WHERE ProductName = 'Laptop';
+-----------+-------------+-------------+---------+
| ProductID | ProductName | Category    | Price   |
+-----------+-------------+-------------+---------+
|         1 | Laptop      | Electronics | 1200.00 |
+-----------+-------------+-------------+---------+
1 row in set (0.0016 sec)
```

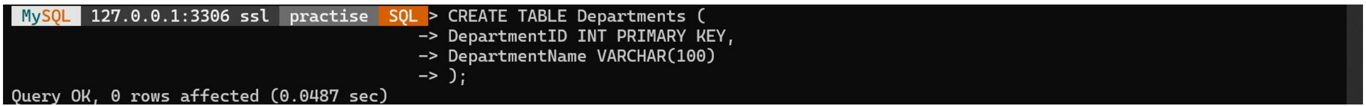


```
 MySQL  127.0.0.1:3306 ssl  practise  SQL > CREATE INDEX idx_ProductName ON Products(ProductName);
Query OK, 0 rows affected (0.0673 sec)

Records: 0  Duplicates: 0  Warnings: 0
 MySQL  127.0.0.1:3306 ssl  practise  SQL > SELECT * FROM Products WHERE ProductName = 'Laptop';
+-----------+-------------+-------------+---------+
| ProductID | ProductName | Category    | Price   |
+-----------+-------------+-------------+---------+
|         1 | Laptop      | Electronics | 1200.00 |
+-----------+-------------+-------------+---------+
1 row in set (0.0012 sec)
 MySQL  127.0.0.1:3306 ssl  practise  SQL > SELECT * FROM Orders WHERE OrderDate = '2023-01-15';
+---------+------------+------------+
| OrderID | CustomerID | OrderDate  |
+---------+------------+------------+
|       1 |          1 | 2023-01-15 |
+---------+------------+------------+
1 row in set (0.0084 sec)
 MySQL  127.0.0.1:3306 ssl  practise  SQL > CREATE INDEX idx_OrderDate ON Orders(OrderDate);
Query OK, 0 rows affected (0.0701 sec)

Records: 0  Duplicates: 0  Warnings: 0
 MySQL  127.0.0.1:3306 ssl  practise  SQL > SELECT * FROM Orders WHERE OrderDate = '2023-01-15';
+---------+------------+------------+
| OrderID | CustomerID | OrderDate  |
+---------+------------+------------+
|       1 |          1 | 2023-01-15 |
+---------+------------+------------+
1 row in set (0.0011 sec)
```

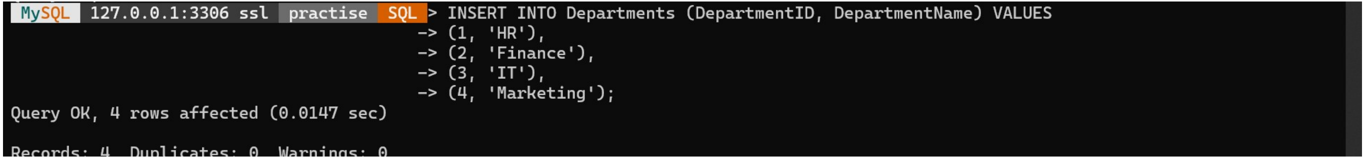


```
 MySQL  127.0.0.1:3306 ssl  practise  SQL > SELECT * FROM Orders WHERE CustomerID = 1 AND OrderDate = '2023-01-15';
+---------+------------+------------+
| OrderID | CustomerID | OrderDate  |
+---------+------------+------------+
|       1 |          1 | 2023-01-15 |
+---------+------------+------------+
1 row in set (0.0025 sec)
 MySQL  127.0.0.1:3306 ssl  practise  SQL > CREATE INDEX idx_CustomerID_OrderDate ON Orders(CustomerID, OrderDate);
Query OK, 0 rows affected (0.0740 sec)

Records: 0  Duplicates: 0  Warnings: 0
 MySQL  127.0.0.1:3306 ssl  practise  SQL > SELECT * FROM Orders WHERE CustomerID = 1 AND OrderDate = '2023-01-15';
+---------+------------+------------+
| OrderID | CustomerID | OrderDate  |
+---------+------------+------------+
|       1 |          1 | 2023-01-15 |
+---------+------------+------------+
1 row in set (0.0013 sec)
```

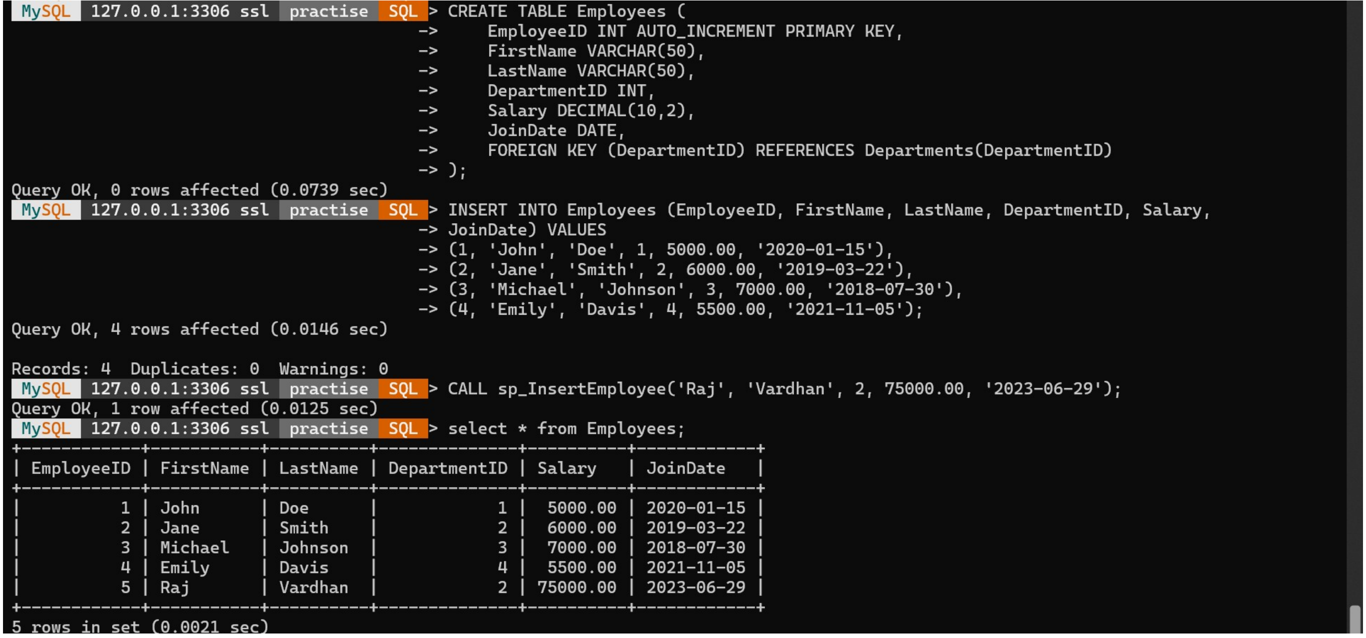
4. SQL Exercise - Stored procedure (Exercise 1: Create a Stored Procedure)



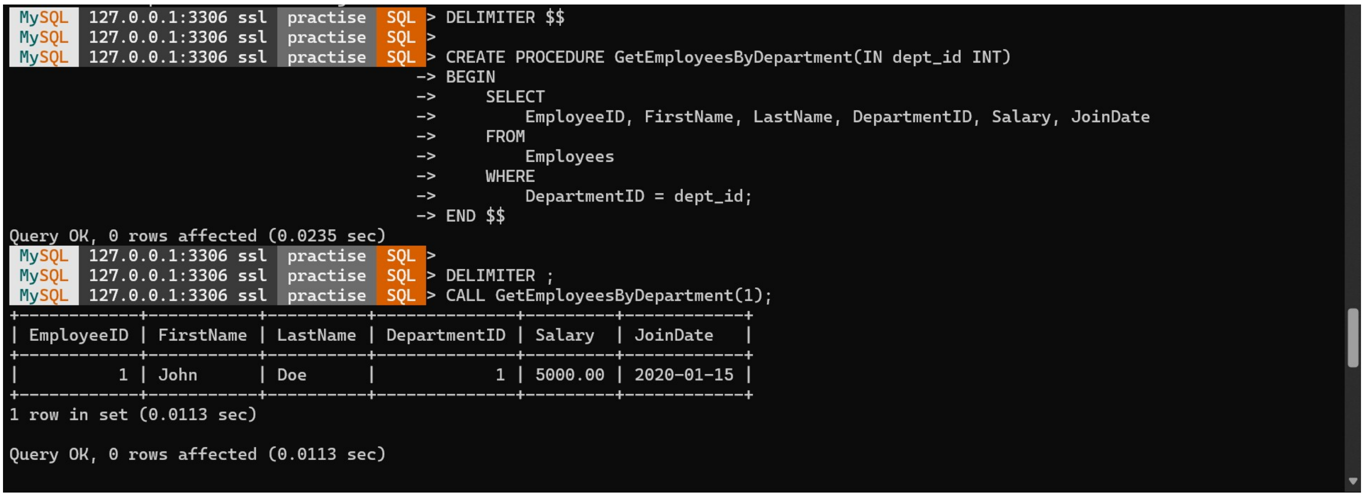
```
 MySQL  127.0.0.1:3306 ssl  practise  SQL > CREATE TABLE Departments (
                                         -> DepartmentID INT PRIMARY KEY,
                                         -> DepartmentName VARCHAR(100)
                                         -> );
Query OK, 0 rows affected (0.0487 sec)
```



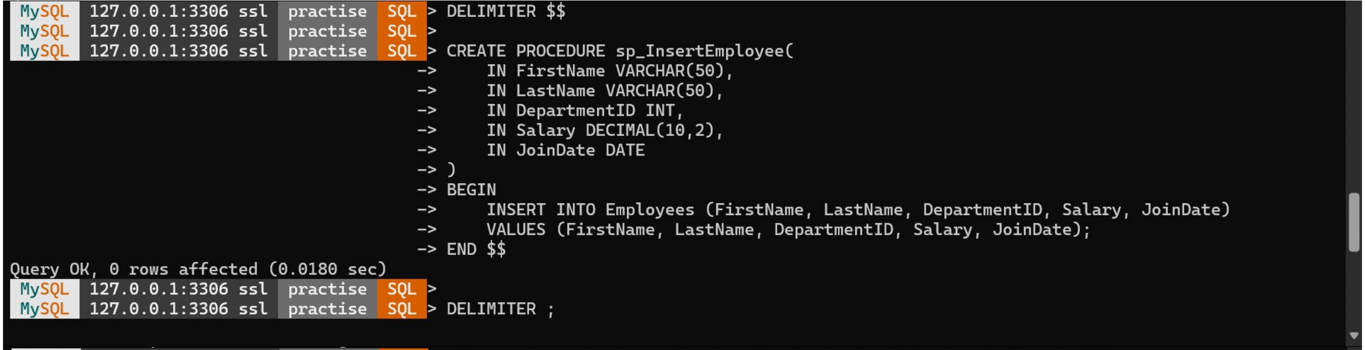
```
 MySQL  127.0.0.1:3306 ssl  practise  SQL > INSERT INTO Departments (DepartmentID, DepartmentName) VALUES
                                         -> (1, 'HR'),
                                         -> (2, 'Finance'),
                                         -> (3, 'IT'),
                                         -> (4, 'Marketing');
Query OK, 4 rows affected (0.0147 sec)

Records: 4  Duplicates: 0  Warnings: 0
```



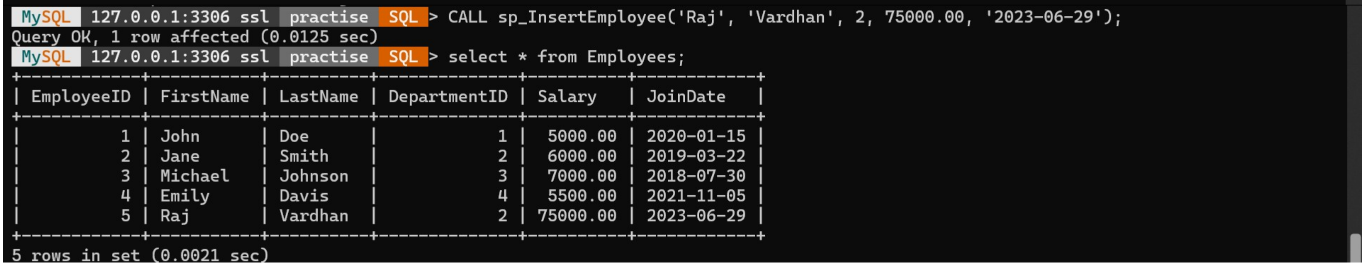
```
 MySQL  127.0.0.1:3306 ssl  practise  SQL > CREATE TABLE Employees (
                                         ->     EmployeeID INT AUTO_INCREMENT PRIMARY KEY,
                                         ->     FirstName VARCHAR(50),
                                         ->     LastName VARCHAR(50),
                                         ->     DepartmentID INT,
                                         ->     Salary DECIMAL(10,2),
                                         ->     JoinDate DATE,
                                         ->     FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)
                                         -> );
Query OK, 0 rows affected (0.0739 sec)
 MySQL  127.0.0.1:3306 ssl  practise  SQL > INSERT INTO Employees (EmployeeID, FirstName, LastName, DepartmentID, Salary,
                                         -> JoinDate) VALUES
                                         -> (1, 'John', 'Doe', 1, 5000.00, '2020-01-15'),
                                         -> (2, 'Jane', 'Smith', 2, 6000.00, '2019-03-22'),
                                         -> (3, 'Michael', 'Johnson', 3, 7000.00, '2018-07-30'),
                                         -> (4, 'Emily', 'Davis', 4, 5500.00, '2021-11-05');
Query OK, 4 rows affected (0.0146 sec)

Records: 4  Duplicates: 0  Warnings: 0
 MySQL  127.0.0.1:3306 ssl  practise  SQL > CALL sp_InsertEmployee('Raj', 'Vardhan', 2, 75000.00, '2023-06-29');
Query OK, 1 row affected (0.0125 sec)
 MySQL  127.0.0.1:3306 ssl  practise  SQL > select * from Employees;
+------------+-----------+----------+--------------+----------+------------+
| EmployeeID | FirstName | LastName | DepartmentID | Salary   | JoinDate   |
+------------+-----------+----------+--------------+----------+------------+
|          1 | John      | Doe      |            1 |  5000.00 | 2020-01-15 |
|          2 | Jane      | Smith    |            2 |  6000.00 | 2019-03-22 |
|          3 | Michael   | Johnson  |            3 |  7000.00 | 2018-07-30 |
|          4 | Emily     | Davis    |            4 |  5500.00 | 2021-11-05 |
|          5 | Raj       | Vardhan  |            2 | 75000.00 | 2023-06-29 |
+------------+-----------+----------+--------------+----------+------------+
5 rows in set (0.0021 sec)
```



```
 MySQL  127.0.0.1:3306 ssl  practise  SQL > DELIMITER $$
 MySQL  127.0.0.1:3306 ssl  practise  SQL >
 MySQL  127.0.0.1:3306 ssl  practise  SQL > CREATE PROCEDURE GetEmployeesByDepartment(IN dept_id INT)
                                         -> BEGIN
                                         ->     SELECT
                                         ->         EmployeeID, FirstName, LastName, DepartmentID, Salary, JoinDate
                                         ->     FROM
                                         ->         Employees
                                         ->     WHERE
                                         ->         DepartmentID = dept_id;
                                         -> END $$
Query OK, 0 rows affected (0.0235 sec)
 MySQL  127.0.0.1:3306 ssl  practise  SQL >
 MySQL  127.0.0.1:3306 ssl  practise  SQL > DELIMITER ;
 MySQL  127.0.0.1:3306 ssl  practise  SQL > CALL GetEmployeesByDepartment(1);
+------------+-----------+----------+--------------+---------+------------+
| EmployeeID | FirstName | LastName | DepartmentID | Salary  | JoinDate   |
+------------+-----------+----------+--------------+---------+------------+
|          1 | John      | Doe      |            1 | 5000.00 | 2020-01-15 |
+------------+-----------+----------+--------------+---------+------------+
1 row in set (0.0113 sec)

Query OK, 0 rows affected (0.0113 sec)
```

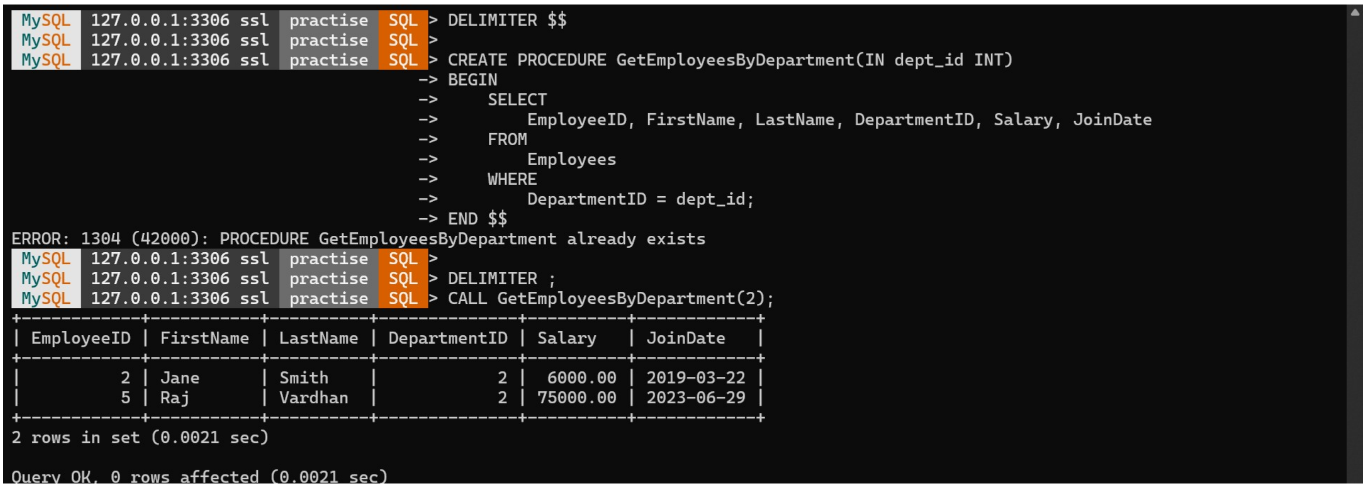


```
 MySQL  127.0.0.1:3306 ssl  practise  SQL > DELIMITER $$
 MySQL  127.0.0.1:3306 ssl  practise  SQL >
 MySQL  127.0.0.1:3306 ssl  practise  SQL > CREATE PROCEDURE sp_InsertEmployee(
                                         ->     IN FirstName VARCHAR(50),
                                         ->     IN LastName VARCHAR(50),
                                         ->     IN DepartmentID INT,
                                         ->     IN Salary DECIMAL(10,2),
                                         ->     IN JoinDate DATE
                                         -> )
                                         -> BEGIN
                                         ->     INSERT INTO Employees (FirstName, LastName, DepartmentID, Salary, JoinDate)
                                         ->     VALUES (FirstName, LastName, DepartmentID, Salary, JoinDate);
                                         -> END $$
Query OK, 0 rows affected (0.0180 sec)
 MySQL  127.0.0.1:3306 ssl  practise  SQL >
 MySQL  127.0.0.1:3306 ssl  practise  SQL > DELIMITER ;
```



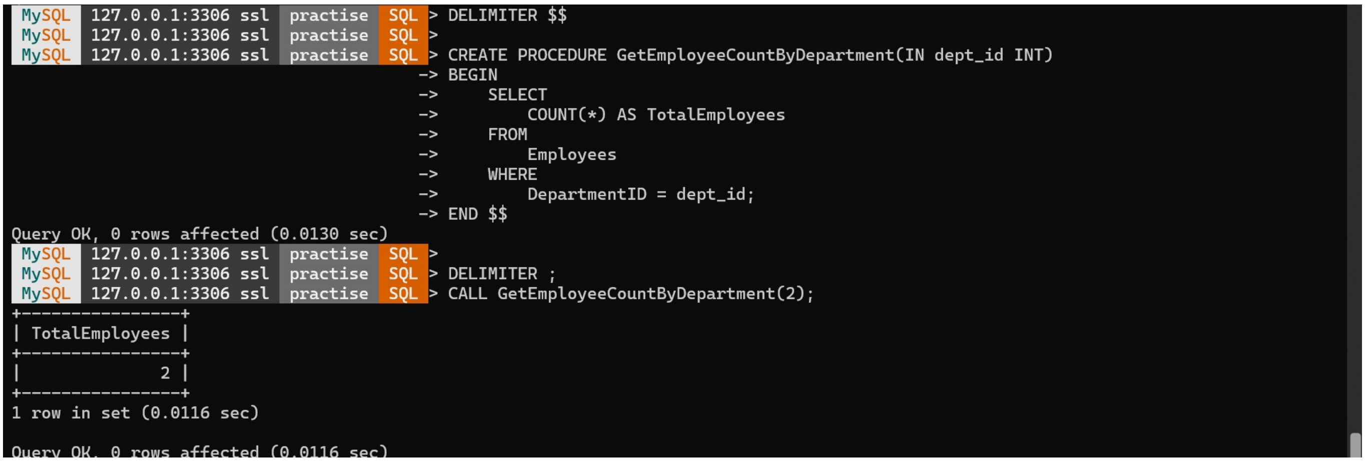
```
 MySQL  127.0.0.1:3306 ssl  practise  SQL > CALL sp_InsertEmployee('Raj', 'Vardhan', 2, 75000.00, '2023-06-29');
Query OK, 1 row affected (0.0125 sec)
 MySQL  127.0.0.1:3306 ssl  practise  SQL > select * from Employees;
+------------+-----------+----------+--------------+----------+------------+
| EmployeeID | FirstName | LastName | DepartmentID | Salary   | JoinDate   |
+------------+-----------+----------+--------------+----------+------------+
|          1 | John      | Doe      |            1 |  5000.00 | 2020-01-15 |
|          2 | Jane      | Smith    |            2 |  6000.00 | 2019-03-22 |
|          3 | Michael   | Johnson  |            3 |  7000.00 | 2018-07-30 |
|          4 | Emily     | Davis    |            4 |  5500.00 | 2021-11-05 |
|          5 | Raj       | Vardhan  |            2 | 75000.00 | 2023-06-29 |
+------------+-----------+----------+--------------+----------+------------+
5 rows in set (0.0021 sec)
```

4. SQL Exercise - Stored procedure (Exercise 4: Execute a Stored Procedure)



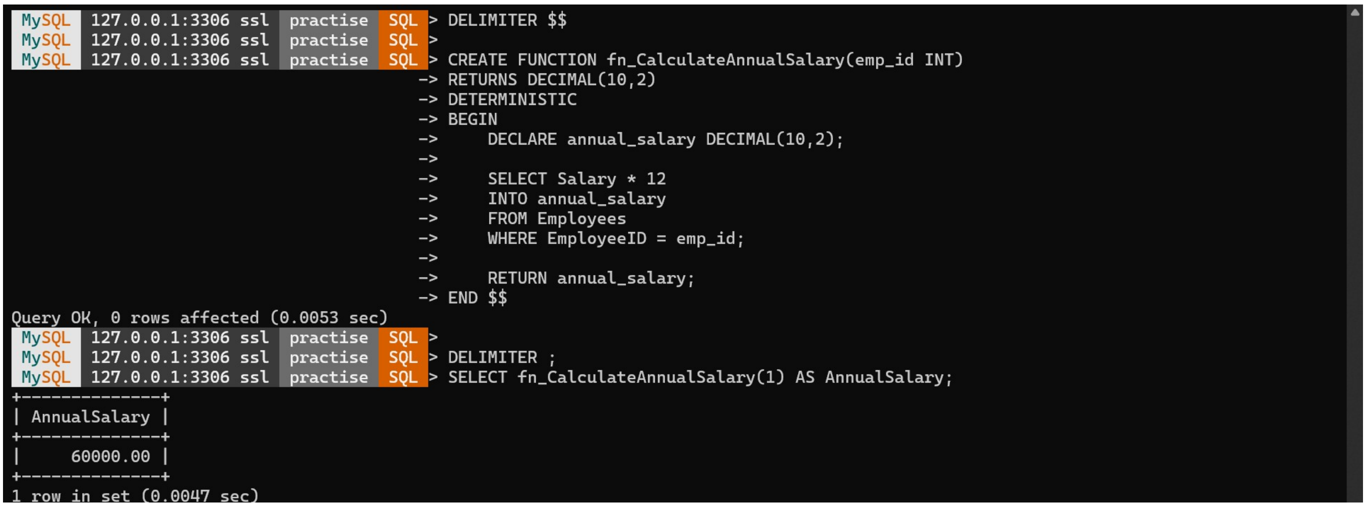
```
 MySQL  127.0.0.1:3306 ssl  practise  SQL > DELIMITER $$
 MySQL  127.0.0.1:3306 ssl  practise  SQL >
 MySQL  127.0.0.1:3306 ssl  practise  SQL > CREATE PROCEDURE GetEmployeesByDepartment(IN dept_id INT)
                                         -> BEGIN
                                         ->     SELECT
                                         ->         EmployeeID, FirstName, LastName, DepartmentID, Salary, JoinDate
                                         ->     FROM
                                         ->         Employees
                                         ->     WHERE
                                         ->         DepartmentID = dept_id;
                                         -> END $$
ERROR: 1304 (42000): PROCEDURE GetEmployeesByDepartment already exists
 MySQL  127.0.0.1:3306 ssl  practise  SQL >
 MySQL  127.0.0.1:3306 ssl  practise  SQL > DELIMITER ;
 MySQL  127.0.0.1:3306 ssl  practise  SQL > CALL GetEmployeesByDepartment(2);
+------------+-----------+----------+--------------+----------+------------+
| EmployeeID | FirstName | LastName | DepartmentID | Salary   | JoinDate   |
+------------+-----------+----------+--------------+----------+------------+
|          2 | Jane      | Smith    |            2 |  6000.00 | 2019-03-22 |
|          5 | Raj       | Vardhan  |            2 | 75000.00 | 2023-06-29 |
+------------+-----------+----------+--------------+----------+------------+
2 rows in set (0.0021 sec)

Query OK, 0 rows affected (0.0021 sec)
```

4. SQL Exercise - Stored procedure (Exercise 5: Return Data from a Stored Procedure)



```
 MySQL  127.0.0.1:3306 ssl  practise  SQL > DELIMITER $$
 MySQL  127.0.0.1:3306 ssl  practise  SQL >
 MySQL  127.0.0.1:3306 ssl  practise  SQL > CREATE PROCEDURE GetEmployeeCountByDepartment(IN dept_id INT)
                                         -> BEGIN
                                         ->     SELECT
                                         ->         COUNT(*) AS TotalEmployees
                                         ->     FROM
                                         ->         Employees
                                         ->     WHERE
                                         ->         DepartmentID = dept_id;
                                         -> END $$
Query OK, 0 rows affected (0.0130 sec)
 MySQL  127.0.0.1:3306 ssl  practise  SQL >
 MySQL  127.0.0.1:3306 ssl  practise  SQL > DELIMITER ;
 MySQL  127.0.0.1:3306 ssl  practise  SQL > CALL GetEmployeeCountByDepartment(2);
+----------------+
| TotalEmployees |
+----------------+
|              2 |
+----------------+
1 row in set (0.0116 sec)

Query OK, 0 rows affected (0.0116 sec)
```

5. SQL Exercise - Functions (Exercise 7: Return Data from a Scalar Function)



```
 MySQL  127.0.0.1:3306 ssl  practise  SQL > DELIMITER $$
 MySQL  127.0.0.1:3306 ssl  practise  SQL >
 MySQL  127.0.0.1:3306 ssl  practise  SQL > CREATE FUNCTION fn_CalculateAnnualSalary(emp_id INT)
                                         -> RETURNS DECIMAL(10,2)
                                         -> DETERMINISTIC
                                         -> BEGIN
                                         ->     DECLARE annual_salary DECIMAL(10,2);
                                         ->
                                         ->     SELECT Salary * 12
                                         ->     INTO annual_salary
                                         ->     FROM Employees
                                         ->     WHERE EmployeeID = emp_id;
                                         ->
                                         ->     RETURN annual_salary;
                                         -> END $$
Query OK, 0 rows affected (0.0053 sec)
 MySQL  127.0.0.1:3306 ssl  practise  SQL >
 MySQL  127.0.0.1:3306 ssl  practise  SQL > DELIMITER ;
 MySQL  127.0.0.1:3306 ssl  practise  SQL > SELECT fn_CalculateAnnualSalary(1) AS AnnualSalary;
+--------------+
| AnnualSalary |
+--------------+
|     60000.00 |
+--------------+
1 row in set (0.0047 sec)
```

NUnit and Moq

1. NUnit-Handson (Hands-on in this document)

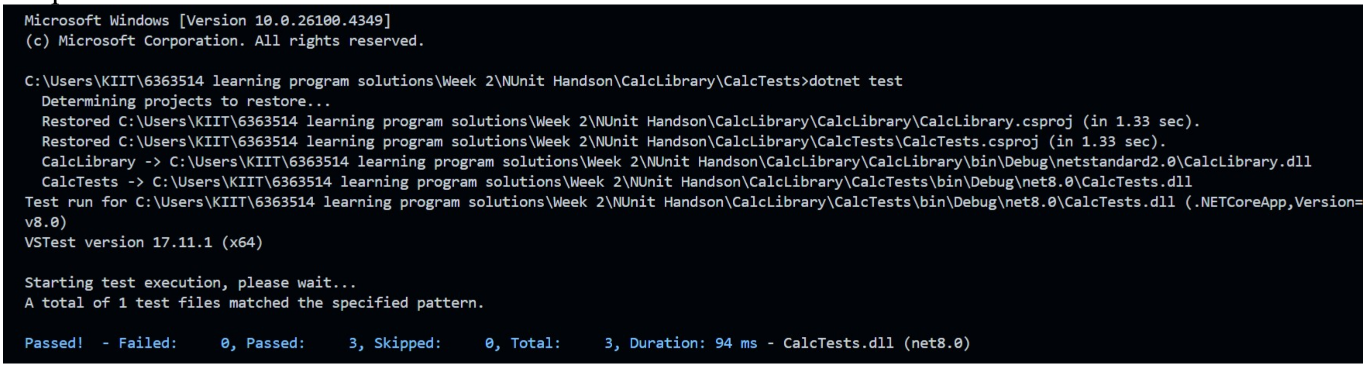
CalcLibrary/MathLibrary.cs

```
using System;

namespace CalcLibrary
{
    interface IMathLibrary
    {
        double Addition(double a, double b);
        double Subtraction(double a, double b);
        double Multiplication(double a, double b);
        double Division(double a, double b);
    }

    public class SimpleCalculator : IMathLibrary
    {
        double result = 0;
        public double Addition(double a, double b)
        {
            result = a + b;
            return result;
        }
        public double Subtraction(double a, double b)
        {
            result = a - b;
            return result;
        }

        public double Multiplication(double a, double b)
        {
            result = a * b;
            return result;
        }

        public double Division(double a, double b)
        {
            if (b == 0)
                throw new ArgumentException("Second Parameter Can't be Zero");
            result = a / b;
            return result;
        }

        public void AllClear()
        {
            result = 0;
        }

        public double GetResult
        {
            get { return result; }
        }
    }
}
```

CalcTests/CalculatorTests.cs

```
using NUnit.Framework;
using CalcLibrary;

namespace CalcTests
{
    [TestFixture]
    public class CalculatorTests
    {
        private SimpleCalculator calculator;

        [SetUp]
        public void Setup()
        {
            calculator = new SimpleCalculator();
        }

        [TearDown]
        public void TearDown()
        {
            // Optional cleanup
        }
        [Test]
        [TestCase(2, 3, 5)]
        [TestCase(-1, 1, 0)]
        public void Add_ShouldReturnCorrectSum(int a, int b, int expected)
        {
            Assert.That(calculator.Addition(a, b), Is.EqualTo(expected));
        }
    }
}
```

Output :

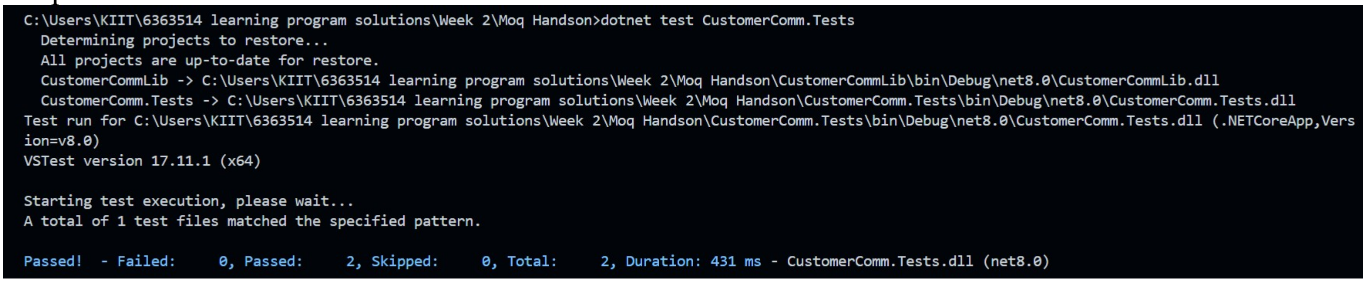


```
Microsoft Windows [Version 10.0.26100.4349]
(c) Microsoft Corporation. All rights reserved.

C:\Users\KIIT\6363514 learning program solutions\Week 2\NUnit Handson\CalcLibrary\CalcTests>dotnet test
  Determining projects to restore...
  Restored C:\Users\KIIT\6363514 learning program solutions\Week 2\NUnit Handson\CalcLibrary\CalcLibrary\CalcLibrary.csproj (in 1.33 sec).
  Restored C:\Users\KIIT\6363514 learning program solutions\Week 2\NUnit Handson\CalcLibrary\CalcTests\CalcTests.csproj (in 1.33 sec).
  CalcLibrary -> C:\Users\KIIT\6363514 learning program solutions\Week 2\NUnit Handson\CalcLibrary\CalcLibrary\bin\Debug\netstandard2.0\CalcLibrary.dll
  CalcTests -> C:\Users\KIIT\6363514 learning program solutions\Week 2\NUnit Handson\CalcLibrary\CalcTests\bin\Debug\net8.0\CalcTests.dll
Test run for C:\Users\KIIT\6363514 learning program solutions\Week 2\NUnit Handson\CalcLibrary\CalcTests\bin\Debug\net8.0\CalcTests.dll (.NETCoreApp,Version=
v8.0)
VSTest version 17.11.1 (x64)

Starting test execution, please wait...
A total of 1 test files matched the specified pattern.

Passed!  - Failed:     0, Passed:     3, Skipped:     0, Total:     3, Duration: 94 ms - CalcTests.dll (net8.0)
```

1. Moq-Handson (Write Testable Code with Moq)

Codes are available on GitHub in respective folder

Output :



```
C:\Users\KIIT\6363514 learning program solutions\Week 2\Moq Handson>dotnet test CustomerComm.Tests
  Determining projects to restore...
  All projects are up-to-date for restore.
  CustomerCommLib -> C:\Users\KIIT\6363514 learning program solutions\Week 2\Moq Handson\CustomerCommLib\bin\Debug\net8.0\CustomerCommLib.dll
  CustomerComm.Tests -> C:\Users\KIIT\6363514 learning program solutions\Week 2\Moq Handson\CustomerComm.Tests\bin\Debug\net8.0\CustomerComm.Tests.dll
Test run for C:\Users\KIIT\6363514 learning program solutions\Week 2\Moq Handson\CustomerComm.Tests\bin\Debug\net8.0\CustomerComm.Tests.dll (.NETCoreApp,Vers
ion=v8.0)
VSTest version 17.11.1 (x64)

Starting test execution, please wait...
A total of 1 test files matched the specified pattern.

Passed!  - Failed:     0, Passed:     2, Skipped:     0, Total:     2, Duration: 431 ms - CustomerComm.Tests.dll (net8.0)
```