**WEEK2-6362596(Superset id)**

**1st NUnit HandOn**

**Calculator Solution:-**

<Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<TargetFramework>net9.0</TargetFramework>

<LangVersion>latest</LangVersion>

<ImplicitUsings>enable</ImplicitUsings>

<Nullable>enable</Nullable>

<IsPackable>false</IsPackable>

</PropertyGroup>

<ItemGroup>

<PackageReference Include="coverlet.collector" Version="6.0.2" />

<PackageReference Include="Microsoft.NET.Test.Sdk" Version="17.14.1" />

<PackageReference Include="NUnit" Version="4.3.2" />

<PackageReference Include="NUnit.Analyzers" Version="4.4.0" />

<PackageReference Include="NUnit3TestAdapter" Version="5.0.0" />

</ItemGroup>

<ItemGroup>

<Using Include="NUnit.Framework" />

</ItemGroup>

<ItemGroup>

<ProjectReference Include="..\CalcLibrary\CalcLibrary.csproj" />

</ItemGroup>

</Project>

**Calculatortest.cs**

using NUnit.Framework;

using CalcLibrary;

namespace CalcLibrary.Tests

{

[TestFixture] // 1. TestFixture attribute

public class CalculatorTests

{

private ICalculator \_calculator;

[SetUp] // 2. Setup (initialize)

public void Setup() => \_calculator = new Calculator();

[TearDown] // 3. Teardown (cleanup)

public void Cleanup() => \_calculator = null;

// 4. Parameterized test with TestCase

[TestCase(5, 7, 12)]

[TestCase(-1, 1, 0)]

public void Add\_ValidInputs\_ReturnsSum(double a, double b, double expected)

{

var result = \_calculator.Add(a, b);

Assert.That(result, Is.EqualTo(expected)); // 5. Assert.That

}

[Test]

[Ignore("Pending implementation")] // 6. Ignore attribute

public void Subtract\_NotImplemented\_ShouldBeIgnored() { }

}

}

**CalcLibrary.csproj**

<Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<TargetFramework>net9.0</TargetFramework>

<ImplicitUsings>enable</ImplicitUsings>

<Nullable>enable</Nullable>

</PropertyGroup>

</Project>

**Calculator.cs**

namespace CalcLibrary

{

public interface ICalculator

{

double Add(double a, double b);

}

public class Calculator : ICalculator

{

public double Add(double a, double b) => a + b;

}

}

**Calculator SOL**

VisualStudioVersion = 17.0.31903.59

MinimumVisualStudioVersion = 10.0.40219.1

Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "CalcLibrary", "CalcLibrary\CalcLibrary.csproj", "{EFF92501-AAF4-469A-A77E-DA99E784EA1F}"

EndProject

Global

GlobalSection(SolutionConfigurationPlatforms) = preSolution

Debug|Any CPU = Debug|Any CPU

Debug|x64 = Debug|x64

Debug|x86 = Debug|x86

Release|Any CPU = Release|Any CPU

Release|x64 = Release|x64

Release|x86 = Release|x86

EndGlobalSection

GlobalSection(ProjectConfigurationPlatforms) = postSolution

{EFF92501-AAF4-469A-A77E-DA99E784EA1F}.Debug|Any CPU.ActiveCfg = Debug|Any CPU

{EFF92501-AAF4-469A-A77E-DA99E784EA1F}.Debug|Any CPU.Build.0 = Debug|Any CPU

{EFF92501-AAF4-469A-A77E-DA99E784EA1F}.Debug|x64.ActiveCfg = Debug|Any CPU

{EFF92501-AAF4-469A-A77E-DA99E784EA1F}.Debug|x64.Build.0 = Debug|Any CPU

{EFF92501-AAF4-469A-A77E-DA99E784EA1F}.Debug|x86.ActiveCfg = Debug|Any CPU

{EFF92501-AAF4-469A-A77E-DA99E784EA1F}.Debug|x86.Build.0 = Debug|Any CPU

{EFF92501-AAF4-469A-A77E-DA99E784EA1F}.Release|Any CPU.ActiveCfg = Release|Any CPU

{EFF92501-AAF4-469A-A77E-DA99E784EA1F}.Release|Any CPU.Build.0 = Release|Any CPU

{EFF92501-AAF4-469A-A77E-DA99E784EA1F}.Release|x64.ActiveCfg = Release|Any CPU

{EFF92501-AAF4-469A-A77E-DA99E784EA1F}.Release|x64.Build.0 = Release|Any CPU

{EFF92501-AAF4-469A-A77E-DA99E784EA1F}.Release|x86.ActiveCfg = Release|Any CPU

{EFF92501-AAF4-469A-A77E-DA99E784EA1F}.Release|x86.Build.0 = Release|Any CPU

EndGlobalSection

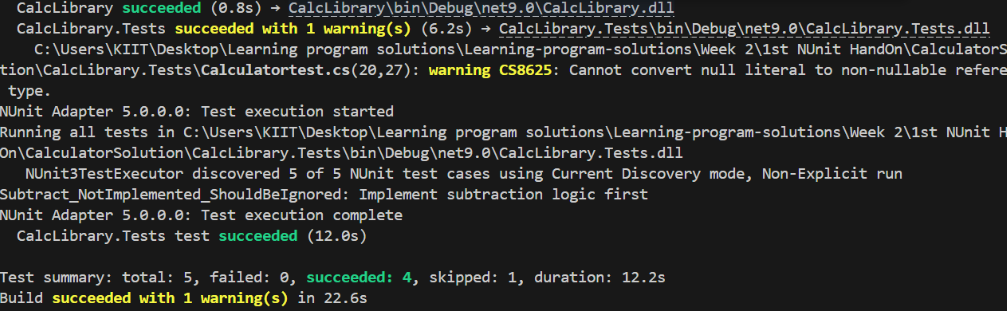
GlobalSection(SolutionProperties) = preSolution

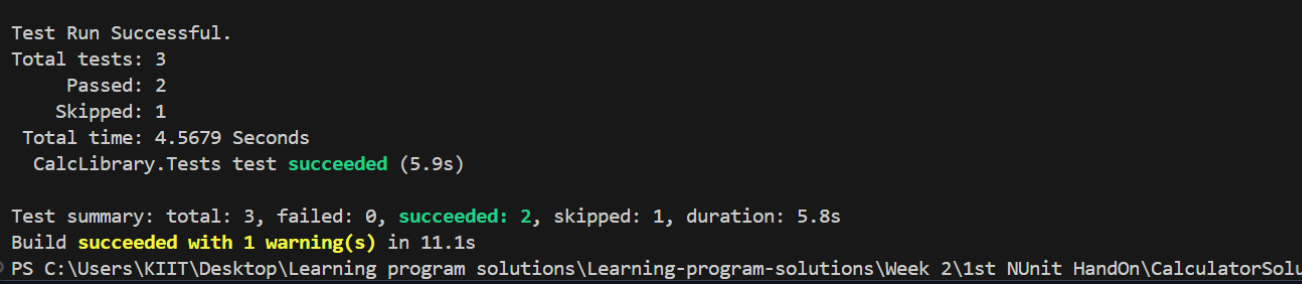
HideSolutionNode = FALSE

EndGlobalSection

EndGlobal

**OUTPUT:-**





**MOQ HandON:-**

<Project Sdk="Microsoft.NET.Sdk">

<ItemGroup>

<ProjectReference Include="..\CustomerCommLib\CustomerCommLib.csproj" />

</ItemGroup>

<ItemGroup>

<PackageReference Include="Microsoft.NET.Test.Sdk" Version="17.14.1" />

<PackageReference Include="Moq" Version="4.20.72" />

<PackageReference Include="NUnit" Version="4.3.2" />

<PackageReference Include="NUnit3TestAdapter" Version="5.0.0" />

</ItemGroup>

<PropertyGroup>

<TargetFramework>net9.0</TargetFramework>

<ImplicitUsings>enable</ImplicitUsings>

<Nullable>enable</Nullable>

</PropertyGroup>

</Project>

using System;

namespace CustomerCommLib

{

public class CustomerComm

{

private readonly IMailSender \_mailSender;

public CustomerComm(IMailSender mailSender)

{

\_mailSender = mailSender ?? throw new ArgumentNullException(nameof(mailSender));

}

public bool SendMailToCustomer()

{

return \_mailSender.SendMail("cust123@abc.com", "Some Message");

}

}

}

<Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<TargetFramework>net9.0</TargetFramework>

<ImplicitUsings>enable</ImplicitUsings>

<Nullable>enable</Nullable>

</PropertyGroup>

</Project>

using System.Net;

using System.Net.Mail;

namespace CustomerCommLib

{

public interface IMailSender

{

bool SendMail(string toAddress, string message);

}

public class MailSender : IMailSender

{

public bool SendMail(string toAddress, string message)

{

MailMessage mail = new MailMessage();

SmtpClient SmtpServer = new SmtpClient("smtp.gmail.com");

mail.From = new MailAddress("your\_email\_address@gmail.com");

mail.To.Add(toAddress);

mail.Subject = "Test Mail";

mail.Body = message;

SmtpServer.Port = 587;

SmtpServer.Credentials = new NetworkCredential("username", "password");

SmtpServer.EnableSsl = true;

SmtpServer.Send(mail);

return true;

}

}

public class CustomerComm

{

private readonly IMailSender \_mailSender;

public CustomerComm(IMailSender mailSender)

{

\_mailSender = mailSender;

}

public bool SendMailToCustomer()

{

\_mailSender.SendMail("cust123@abc.com", "Some Message");

return true;

}

}

}

VisualStudioVersion = 17.0.31903.59

MinimumVisualStudioVersion = 10.0.40219.1

Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "CustomerCommLib", "CustomerCommLib\CustomerCommLib.csproj", "{699A75FF-CFB8-48F0-B333-136958CF6C23}"

EndProject

Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "CustomerComm.Tests", "CustomerComm.Tests\CustomerComm.Tests.csproj", "{F2F1AD6A-35B3-42D2-89F9-A9B986F040CC}"

EndProject

Global

GlobalSection(SolutionConfigurationPlatforms) = preSolution

Debug|Any CPU = Debug|Any CPU

Debug|x64 = Debug|x64

Debug|x86 = Debug|x86

Release|Any CPU = Release|Any CPU

Release|x64 = Release|x64

Release|x86 = Release|x86

EndGlobalSection

GlobalSection(ProjectConfigurationPlatforms) = postSolution

{699A75FF-CFB8-48F0-B333-136958CF6C23}.Debug|Any CPU.ActiveCfg = Debug|Any CPU

{699A75FF-CFB8-48F0-B333-136958CF6C23}.Debug|Any CPU.Build.0 = Debug|Any CPU

{699A75FF-CFB8-48F0-B333-136958CF6C23}.Debug|x64.ActiveCfg = Debug|Any CPU

{699A75FF-CFB8-48F0-B333-136958CF6C23}.Debug|x64.Build.0 = Debug|Any CPU

{699A75FF-CFB8-48F0-B333-136958CF6C23}.Debug|x86.ActiveCfg = Debug|Any CPU

{699A75FF-CFB8-48F0-B333-136958CF6C23}.Debug|x86.Build.0 = Debug|Any CPU

{699A75FF-CFB8-48F0-B333-136958CF6C23}.Release|Any CPU.ActiveCfg = Release|Any CPU

{699A75FF-CFB8-48F0-B333-136958CF6C23}.Release|Any CPU.Build.0 = Release|Any CPU

{699A75FF-CFB8-48F0-B333-136958CF6C23}.Release|x64.ActiveCfg = Release|Any CPU

{699A75FF-CFB8-48F0-B333-136958CF6C23}.Release|x64.Build.0 = Release|Any CPU

{699A75FF-CFB8-48F0-B333-136958CF6C23}.Release|x86.ActiveCfg = Release|Any CPU

{699A75FF-CFB8-48F0-B333-136958CF6C23}.Release|x86.Build.0 = Release|Any CPU

{F2F1AD6A-35B3-42D2-89F9-A9B986F040CC}.Debug|Any CPU.ActiveCfg = Debug|Any CPU

{F2F1AD6A-35B3-42D2-89F9-A9B986F040CC}.Debug|Any CPU.Build.0 = Debug|Any CPU

{F2F1AD6A-35B3-42D2-89F9-A9B986F040CC}.Debug|x64.ActiveCfg = Debug|Any CPU

{F2F1AD6A-35B3-42D2-89F9-A9B986F040CC}.Debug|x64.Build.0 = Debug|Any CPU

{F2F1AD6A-35B3-42D2-89F9-A9B986F040CC}.Debug|x86.ActiveCfg = Debug|Any CPU

{F2F1AD6A-35B3-42D2-89F9-A9B986F040CC}.Debug|x86.Build.0 = Debug|Any CPU

{F2F1AD6A-35B3-42D2-89F9-A9B986F040CC}.Release|Any CPU.ActiveCfg = Release|Any CPU

{F2F1AD6A-35B3-42D2-89F9-A9B986F040CC}.Release|Any CPU.Build.0 = Release|Any CPU

{F2F1AD6A-35B3-42D2-89F9-A9B986F040CC}.Release|x64.ActiveCfg = Release|Any CPU

{F2F1AD6A-35B3-42D2-89F9-A9B986F040CC}.Release|x64.Build.0 = Release|Any CPU

{F2F1AD6A-35B3-42D2-89F9-A9B986F040CC}.Release|x86.ActiveCfg = Release|Any CPU

{F2F1AD6A-35B3-42D2-89F9-A9B986F040CC}.Release|x86.Build.0 = Release|Any CPU

EndGlobalSection

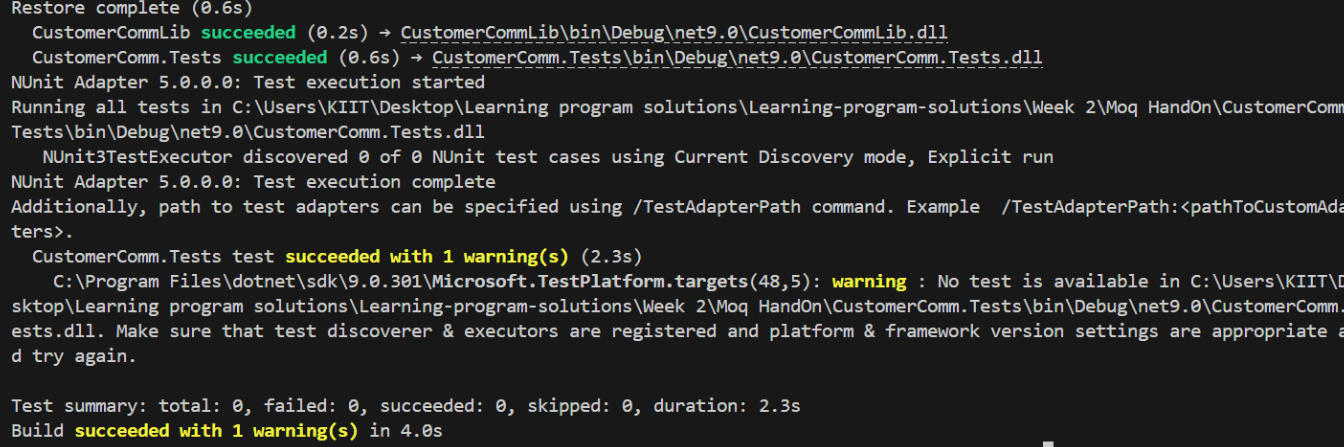
GlobalSection(SolutionProperties) = preSolution

HideSolutionNode = FALSE

EndGlobalSection

EndGlobal

**OUTPUT:-**



**Ranking and Window**

CREATE DATABASE ShopDB;

Use ShopDB;

CREATE TABLE Products (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(100),

Category VARCHAR(50),

Price DECIMAL(10, 2)

);

INSERT INTO Products (ProductID, ProductName, Category, Price) VALUES

(1, 'Laptop', 'Electronics', 1200.00),

(2, 'Smartphone', 'Electronics', 800.00),

(3, 'Tablet', 'Electronics', 600.00),

(4, 'Headphones', 'Accessories', 150.00);

-- Using ROW\_NUMBER()

SELECT \*

FROM (

SELECT \*,

ROW\_NUMBER() OVER (PARTITION BY Category ORDER BY Price DESC) AS row\_num

FROM Products

) AS ranked

WHERE row\_num <= 3;

INSERT INTO Products (ProductID, ProductName, Category, Price) VALUES

(5, 'Smartwatch', 'Electronics', 800.00), -- same price as Smartphone

(6, 'Earbuds', 'Accessories', 150.00);

SELECT

ProductID,

ProductName,

Category,

Price,

RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS rank\_num

FROM Products;

SELECT

ProductID,

ProductName,

Category,

Price,

DENSE\_RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS dense\_rank\_num

FROM Products;

SELECT \*

FROM (

SELECT

ProductID,

ProductName,

Category,

Price,

DENSE\_RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS dense\_rank\_num

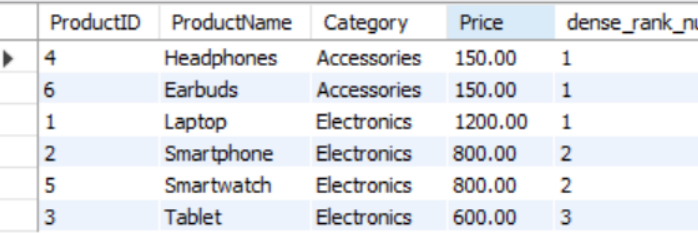
FROM Products

) AS ranked

WHERE dense\_rank\_num <= 3

ORDER BY Category, dense\_rank\_num;

**OUTPUT:-**



**Return Data from stored procedure**

DROP TABLE IF EXISTS Employees;

DROP TABLE IF EXISTS Departments;

CREATE TABLE Departments (

DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(100)

);

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

DepartmentID INT,

Salary DECIMAL(10,2),

JoinDate DATE,

FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)

);

-- Insert into Departments

INSERT INTO Departments (DepartmentID, DepartmentName) VALUES

(1, 'HR'),

(2, 'Finance'),

(3, 'IT'),

(4, 'Marketing');

-- Insert into Employees

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');

DELIMITER $$

CREATE PROCEDURE GetEmployeeCountByDepartment (

IN dept\_id INT

)

BEGIN

SELECT COUNT(\*) AS TotalEmployees

FROM Employees

WHERE DepartmentID = dept\_id;

END$$

DELIMITER ;

CALL GetEmployeeCountByDepartment(3);

**Stored Procedure**

CREATE TABLE Departments (

DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(100)

);

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

DepartmentID INT,

Salary DECIMAL(10,2),

JoinDate DATE,

FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)

);

INSERT INTO Departments (DepartmentID, DepartmentName) VALUES

(1, 'HR'),

(2, 'Finance'),

(3, 'IT'),

(4, 'Marketing');

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');

DROP PROCEDURE IF EXISTS sp\_GetEmployeesByDepartment;

DELIMITER //

CREATE PROCEDURE sp\_GetEmployeesByDepartment(IN dept\_id INT)

BEGIN

SELECT

EmployeeID,

FirstName,

LastName,

DepartmentID,

Salary,

JoinDate

FROM Employees

WHERE DepartmentID = dept\_id;

END //

DELIMITER ;

CALL sp\_GetEmployeesByDepartment(2);

DROP PROCEDURE IF EXISTS sp\_InsertEmployee;

DELIMITER //

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 (EmployeeID, FirstName, LastName, DepartmentID, Salary, JoinDate)

VALUES ((SELECT IFNULL(MAX(EmployeeID), 0) + 1 FROM Employees), FirstName, LastName, DepartmentID, Salary, JoinDate);

END //

DELIMITER ;

DROP TABLE IF EXISTS Employees;

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY AUTO\_INCREMENT,

FirstName VARCHAR(50),

LastName VARCHAR(50),

DepartmentID INT,

Salary DECIMAL(10,2),

JoinDate DATE,

FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)

);

DROP PROCEDURE IF EXISTS sp\_InsertEmployee;

DELIMITER //

CREATE PROCEDURE sp\_InsertEmployee(

IN p\_FirstName VARCHAR(50),

IN p\_LastName VARCHAR(50),

IN p\_DepartmentID INT,

IN p\_Salary DECIMAL(10,2),

IN p\_JoinDate DATE

)

BEGIN

INSERT INTO Employees (FirstName, LastName, DepartmentID, Salary, JoinDate)

VALUES (p\_FirstName, p\_LastName, p\_DepartmentID, p\_Salary, p\_JoinDate);

END //

DELIMITER ;

CALL sp\_InsertEmployee('Alice', 'Taylor', 3, 7200.00, '2022-10-01');

**OUTPUT:-**

