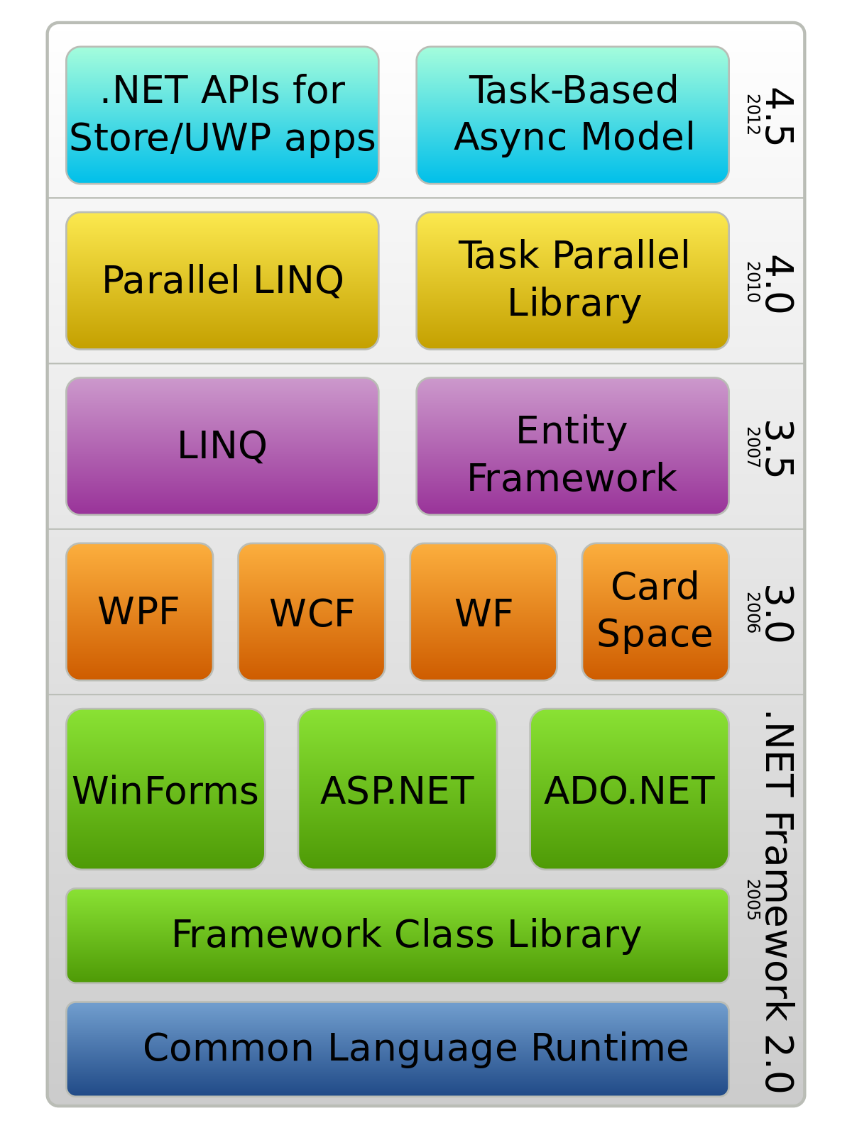
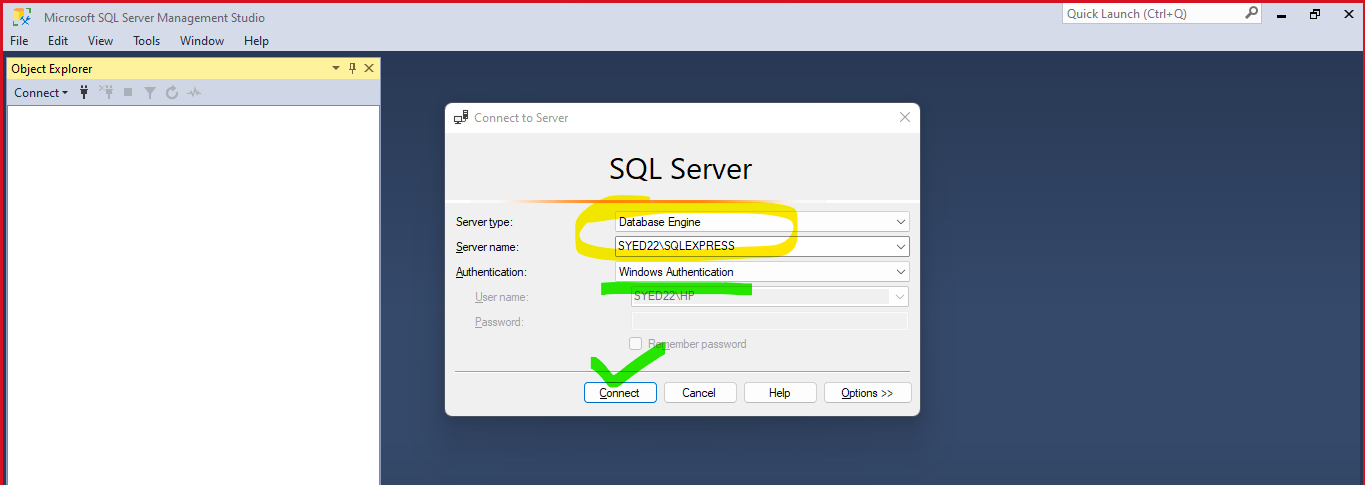
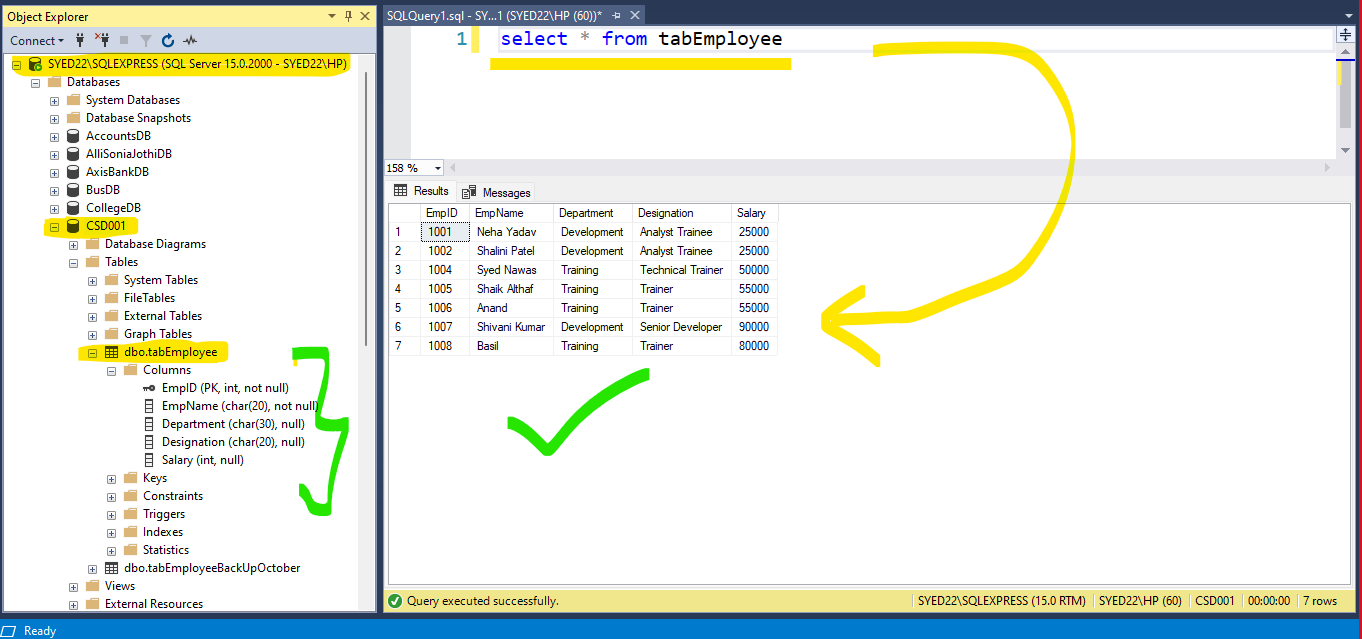
ADO.NET CRUD Example using Three Tier Architecture

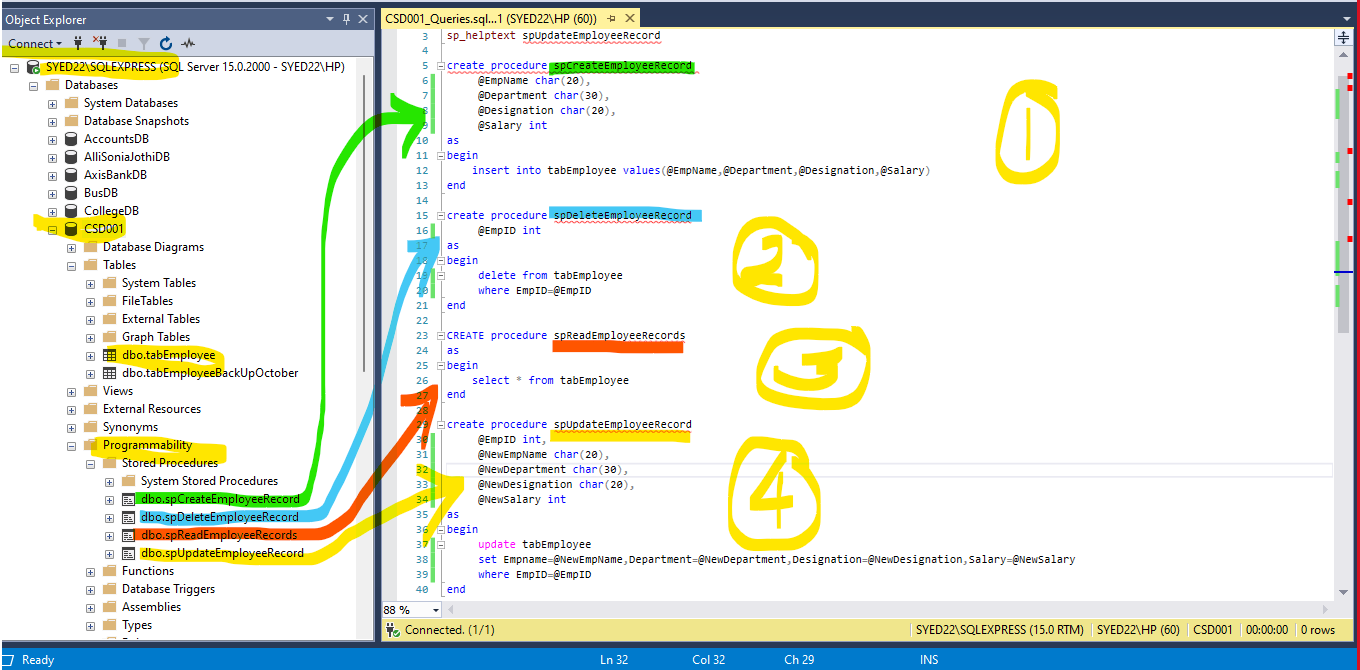


ADO.NET is a data access technology from the Microsoft .NET Framework that provides communication between relational and non-relational systems through a common set of components. ADO.NET is a set of computer software components that programmers can use to access data and data services from a database.

Step 0: Have a Database, Table







select \* from tabEmployee

sp\_helptext spUpdateEmployeeRecord

create procedure spCreateEmployeeRecord

@EmpName char(20),

@Department char(30),

@Designation char(20),

@Salary int

as

begin

insert into tabEmployee values(@EmpName,@Department,@Designation,@Salary)

end

create procedure spDeleteEmployeeRecord

@EmpID int

as

begin

delete from tabEmployee

where EmpID=@EmpID

end

CREATE procedure spReadEmployeeRecords

as

begin

select \* from tabEmployee

end

create procedure spUpdateEmployeeRecord

@EmpID int,

@NewEmpName char(20),

@NewDepartment char(30),

@NewDesignation char(20),

@NewSalary int

as

begin

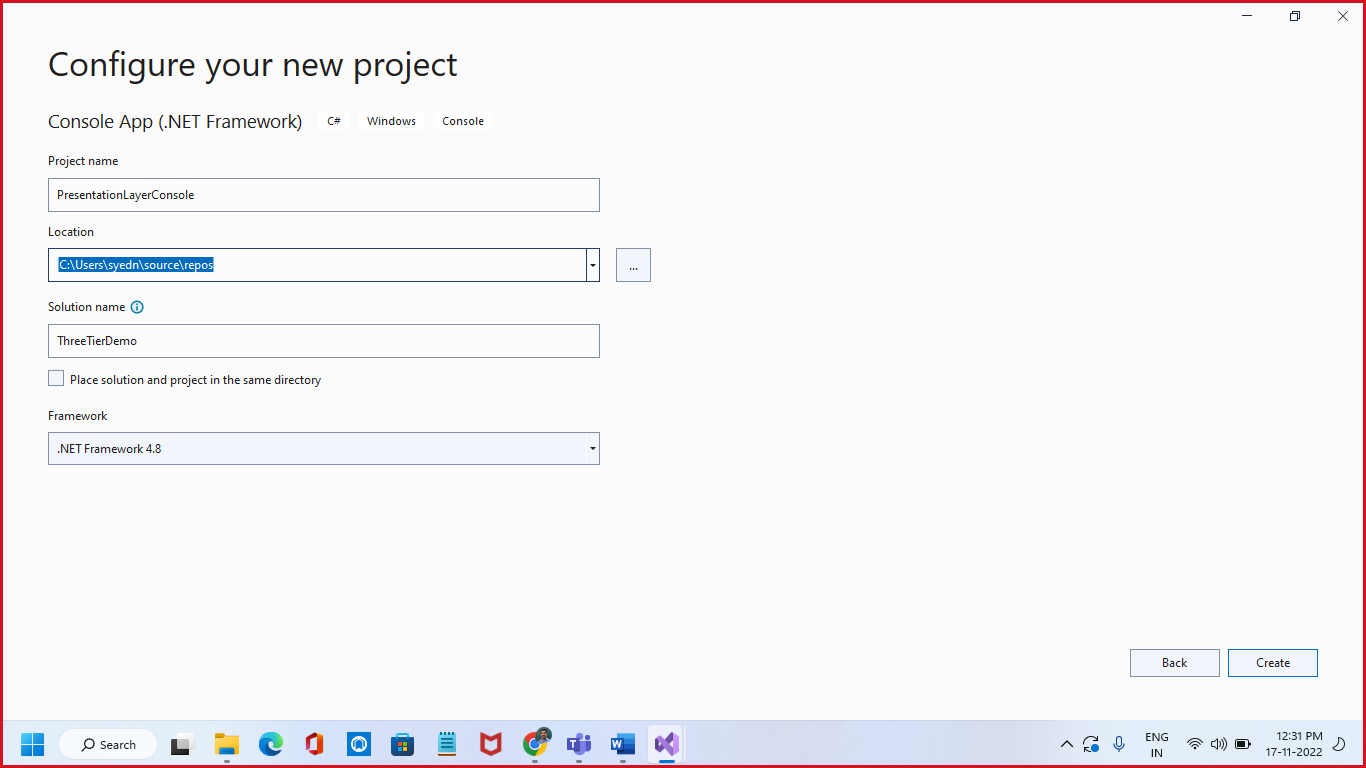
update tabEmployee

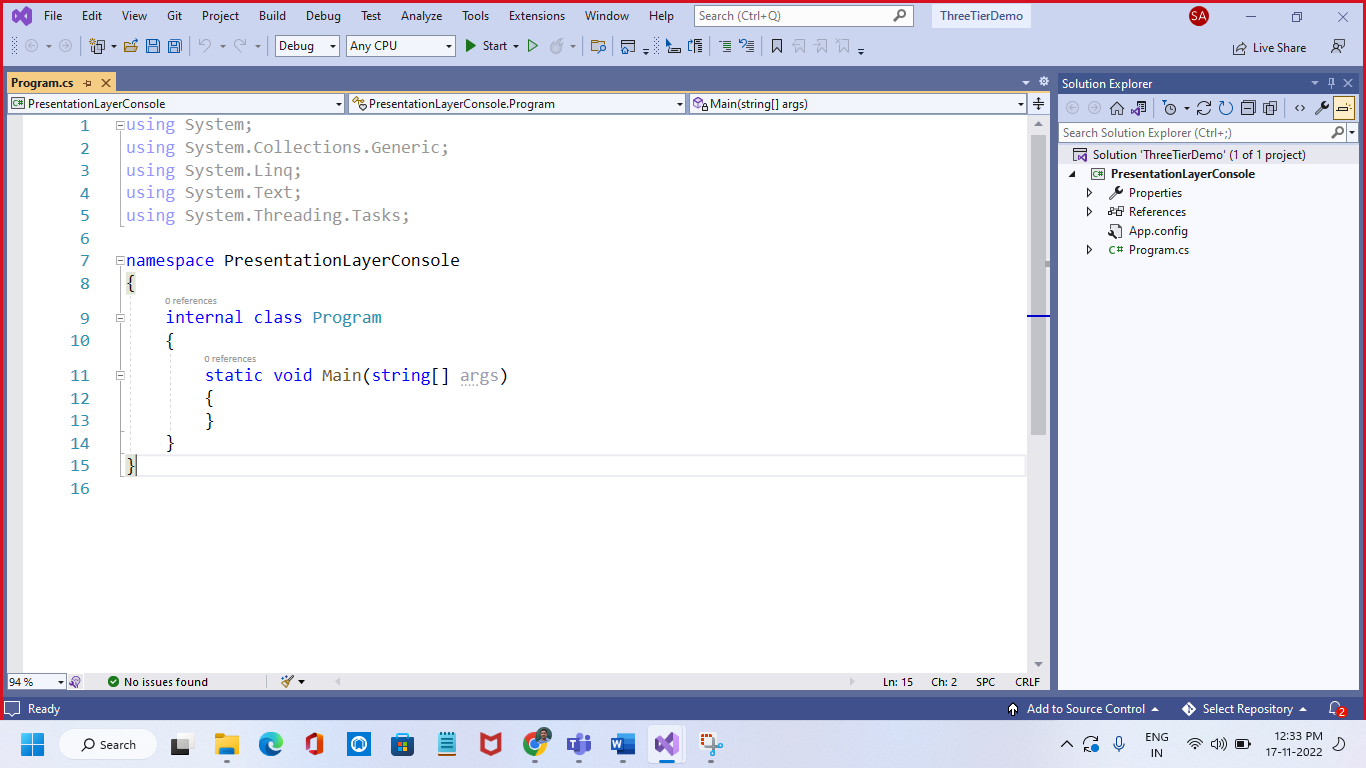
set Empname=@NewEmpName,Department=@NewDepartment,Designation=@NewDesignation,Salary=@NewSalary

where EmpID=@EmpID

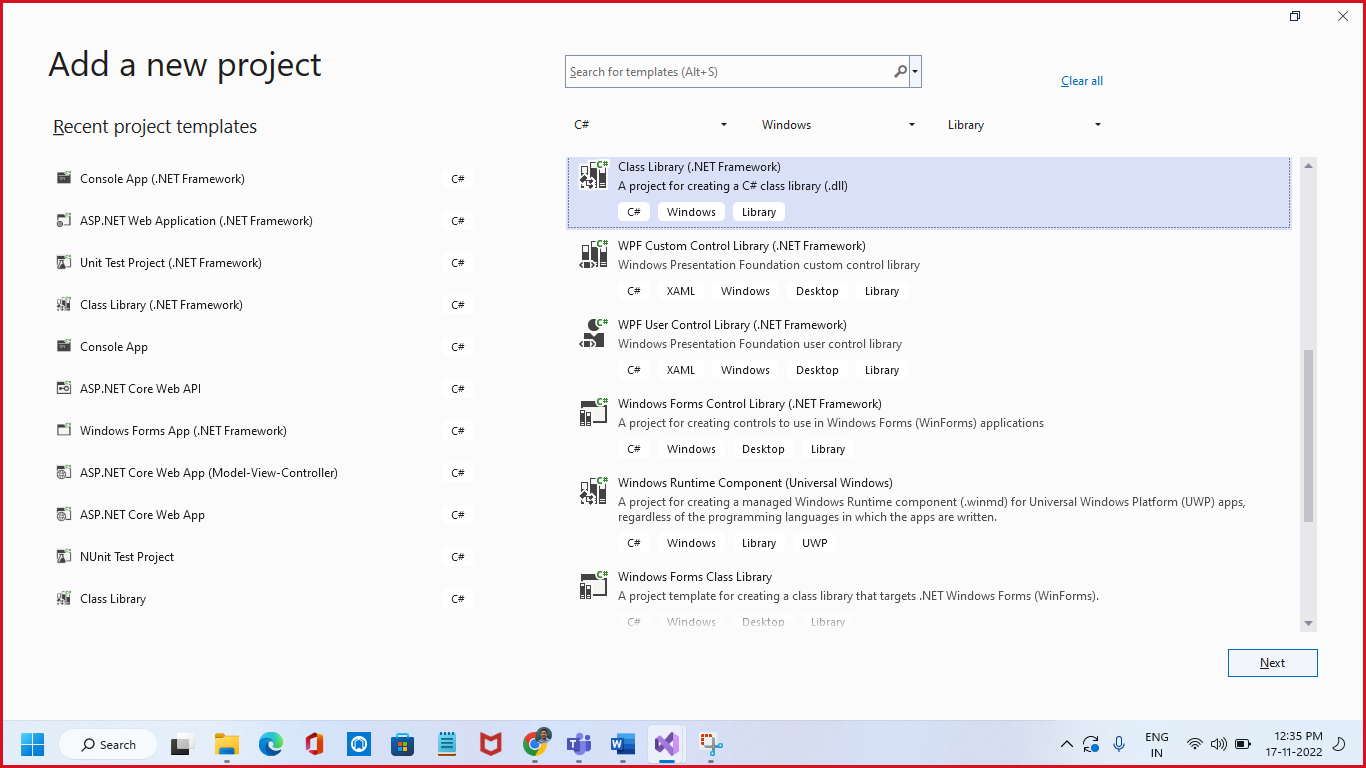
end

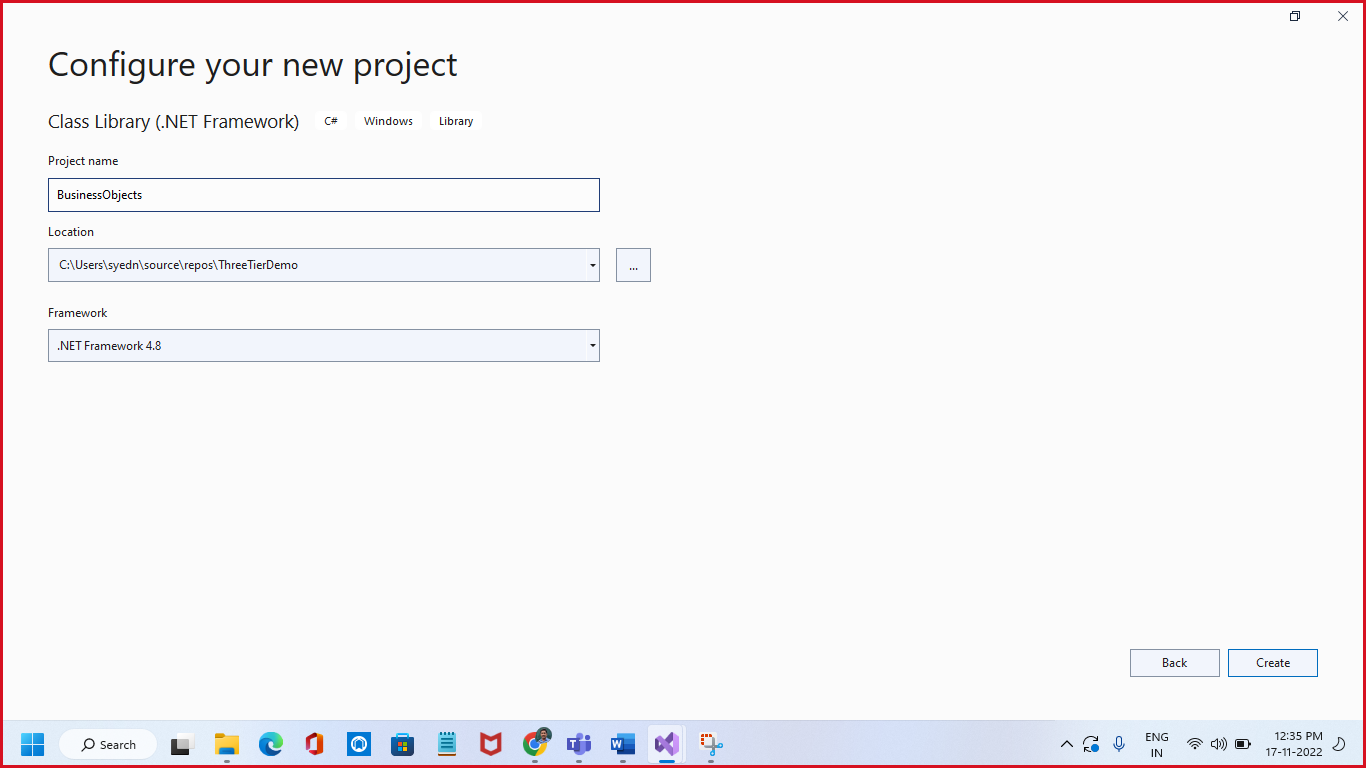
Step 1: Create Front End Application – Presentation Layer (Console / WebForms / Desktop / MVC)

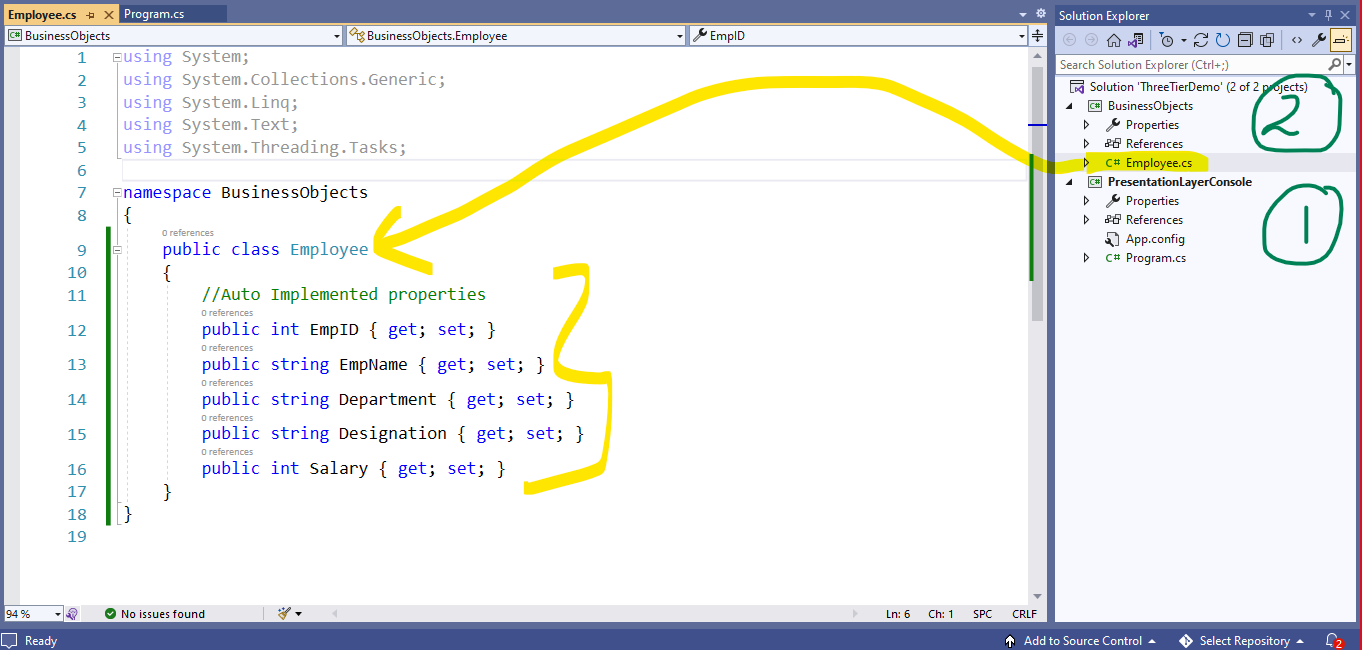




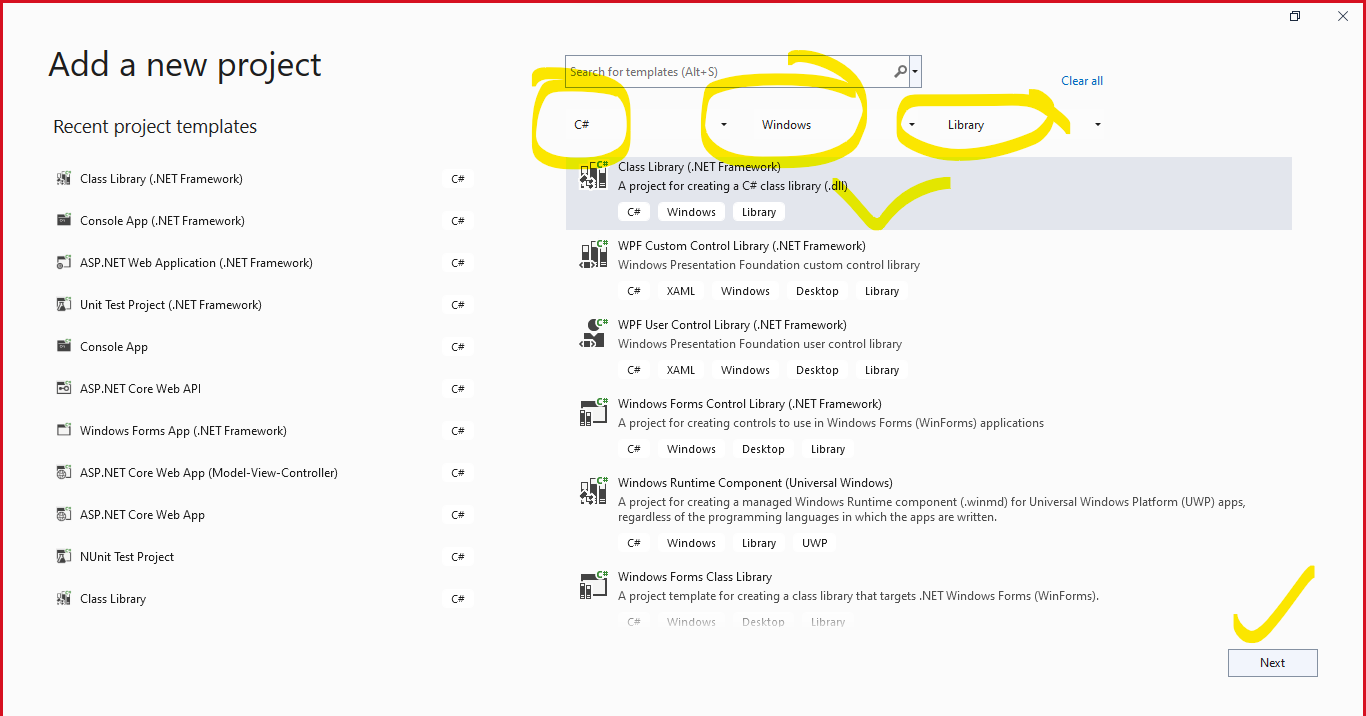
Step 2: Add another Library Project (Business Objects)



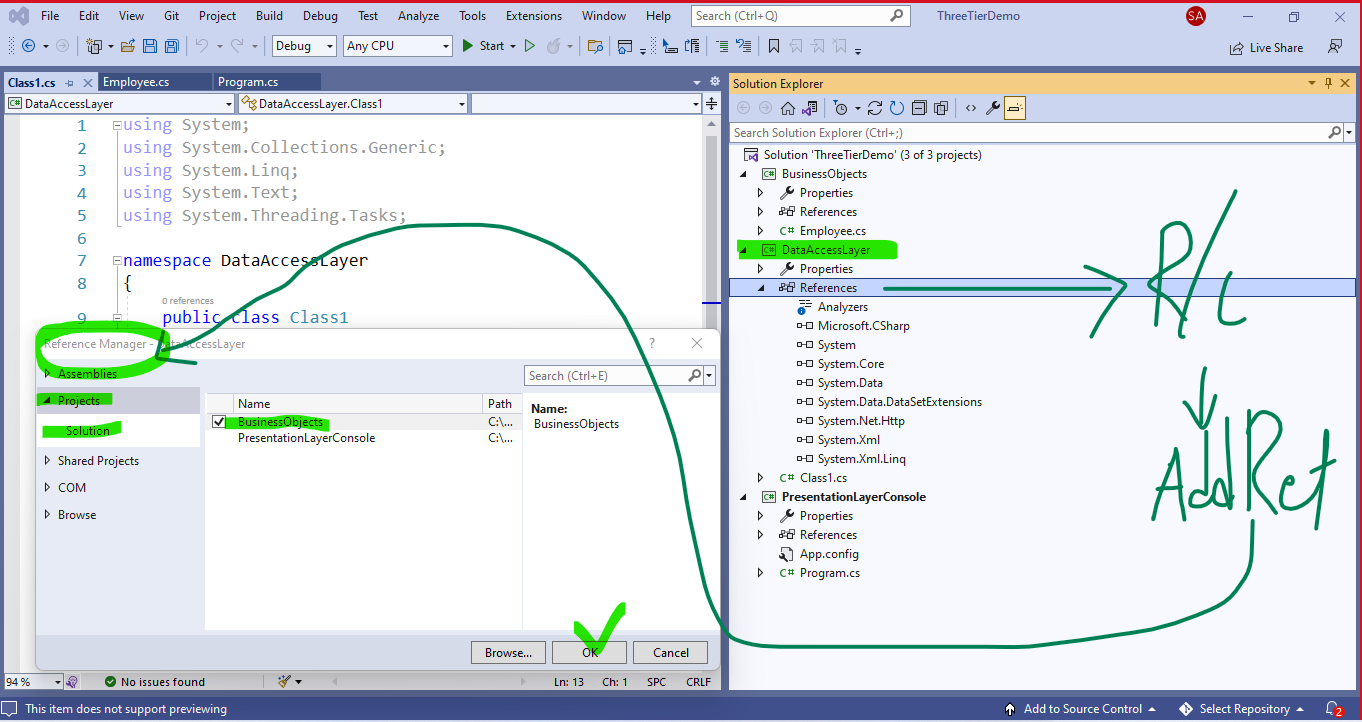




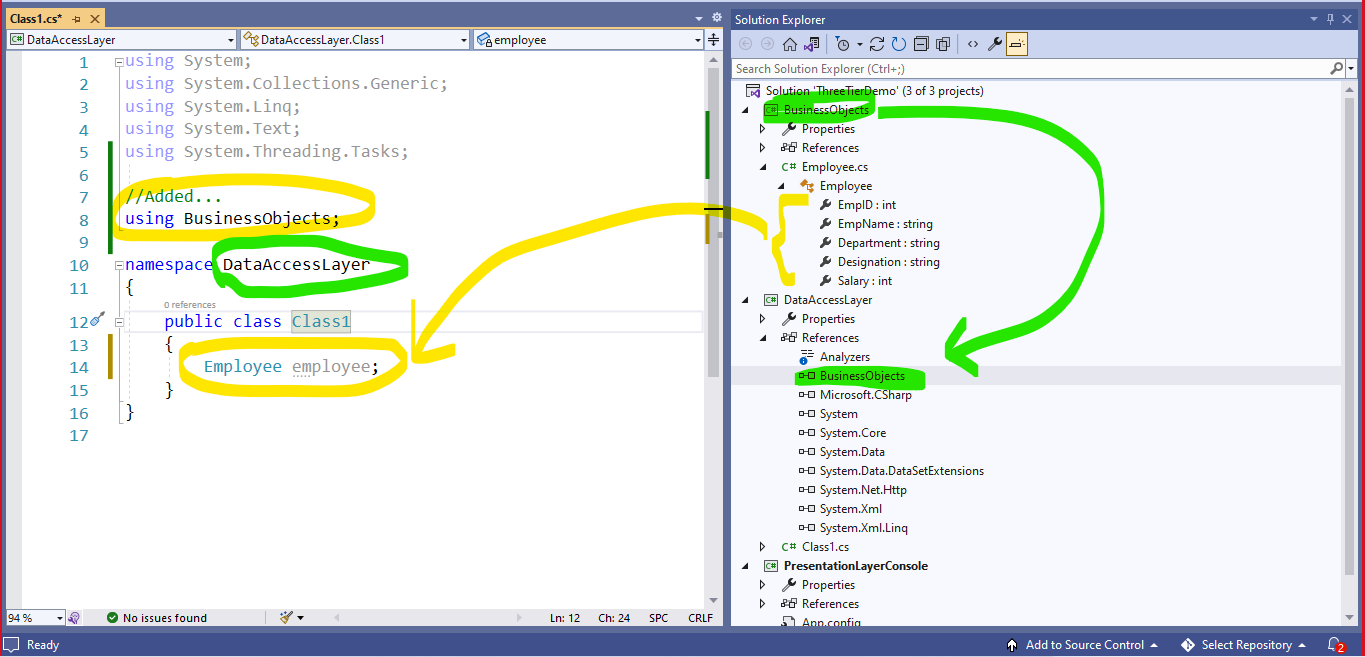
Step 3: Add Another Class Library Project (Give the Name - DataAccessLayer)



After project get created, add Project Reference as below



By referring BusinessObjects in DataAccessLayer, We can user Employee class in DAL as below



Rename Class1.cs as DALOperations.cs and Replace the entire content of above class with following Code

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

//Added Namespaces

using System.Data.SqlClient;

//Added Projects...

using BusinessObjects;

namespace DataAccessLayer

{

public class DALOperations //ADO.NET Code for CRUD

{

static string ConStr = @"Data Source=SYED22\SQLEXPRESS;Initial Catalog=CSD001;Integrated Security=True;";

public static List<Employee> ReadEmployeeRecords()

{

//Step1

SqlConnection con = new SqlConnection(ConStr);

con.Open();

//Step2

SqlCommand cmd = new SqlCommand("spReadEmployeeRecords", con);

cmd.CommandType = System.Data.CommandType.StoredProcedure;

SqlDataReader dr = cmd.ExecuteReader(); //Retreive data

List<Employee> employees = new List<Employee>();

while (dr.Read()) // (x<=10) or (x!=5) or (i<=n) true or false, dr.Read() - true or false

{

Employee emp = new Employee();

emp.EmpID = Convert.ToInt32(dr["EmpID"]); //1001

emp.EmpName = dr["EmpName"].ToString(); //Neha

emp.Department = dr["Department"].ToString(); //Deve

emp.Designation = dr["Designation"].ToString(); //Analyst

emp.Salary = Convert.ToInt32(dr["Salary"]); //25000

employees.Add(emp);

}

cmd.Dispose();

con.Close();

return employees;

}

public static int SaveEmployeeRecord(Employee emp)

{

//Step1

SqlConnection con = new SqlConnection(ConStr);

con.Open();

//Step2

SqlCommand cmd = new SqlCommand("spCreateEmployeeRecord", con);

cmd.CommandType = System.Data.CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@EmpName", emp.EmpName);

cmd.Parameters.AddWithValue("@Department", emp.Department);

cmd.Parameters.AddWithValue("@Designation", emp.Designation);

cmd.Parameters.AddWithValue("@Salary", emp.Salary);

//Query means Retreive data, NonQuery means all ops excluding retreival op

int RowsAffected = cmd.ExecuteNonQuery();

cmd.Dispose();

con.Close();

return RowsAffected;

}

public static int UpdateEmployeeRecord(int EmpID, Employee UpdatedEmpRec)

{

//Step1

SqlConnection con = new SqlConnection(ConStr);

con.Open();

//Step2

SqlCommand cmd = new SqlCommand("spUpdateEmployeeRecord", con);

cmd.CommandType = System.Data.CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@EmpID", EmpID);

cmd.Parameters.AddWithValue("@NewEmpName", UpdatedEmpRec.EmpName);

cmd.Parameters.AddWithValue("@NewDepartment", UpdatedEmpRec.Department);

cmd.Parameters.AddWithValue("@NewDesignation", UpdatedEmpRec.Designation);

cmd.Parameters.AddWithValue("@NewSalary", UpdatedEmpRec.Salary);

//Query means Retreive data, NonQuery means all ops excluding retreival op

int RowsAffected = cmd.ExecuteNonQuery();

cmd.Dispose();

con.Close();

return RowsAffected;

}

public static int DeleteEmployeeRecord(int EmpID)

{

//Step1

SqlConnection con = new SqlConnection(ConStr);

con.Open();

//Step2

SqlCommand cmd = new SqlCommand("spDeleteEmployeeRecord", con);

cmd.CommandType = System.Data.CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@EmpID", EmpID);

//Query means Retreive data, NonQuery means all ops excluding retreival op

int RowsAffected = cmd.ExecuteNonQuery();

cmd.Dispose();

con.Close();

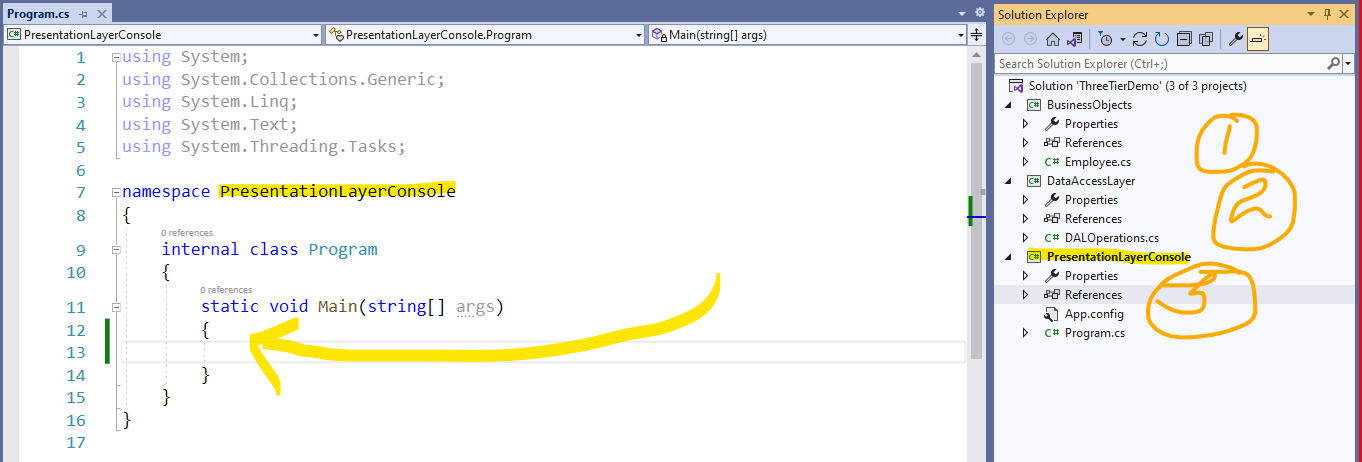
return RowsAffected;

}

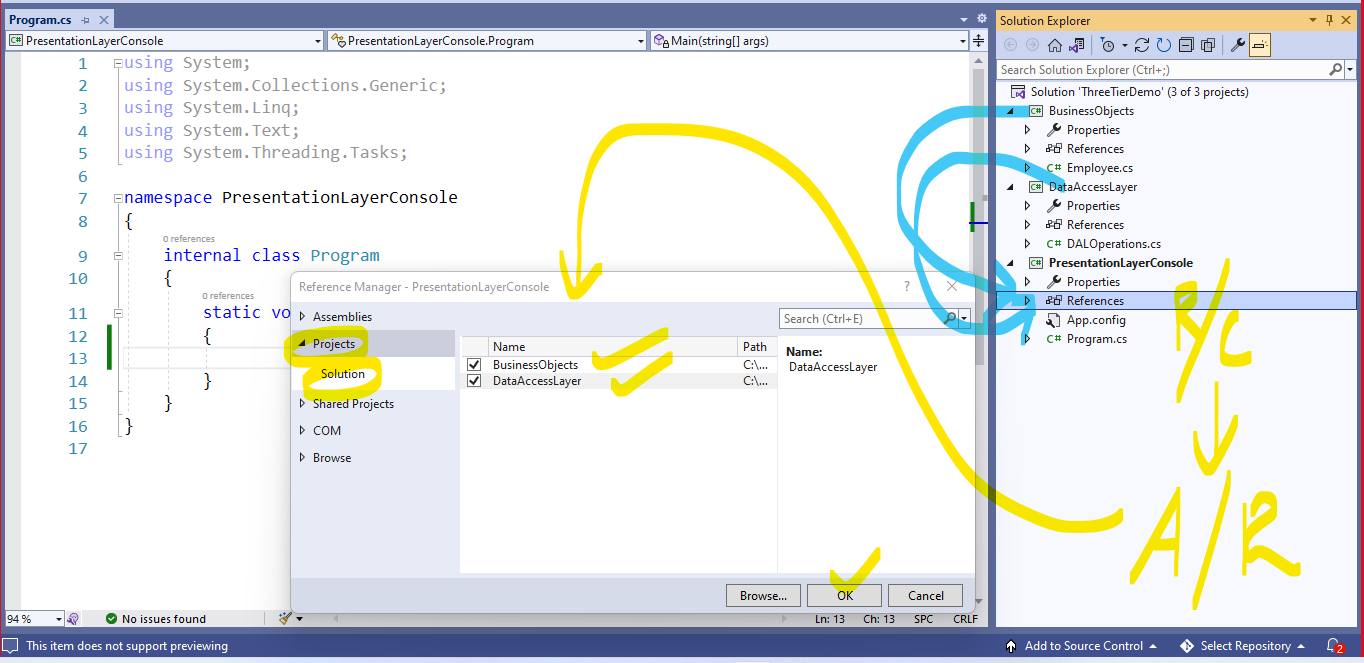
}

}

Step 4: Add **PresentationLayerConsole** (Console Application Project) in your solution as below



Add – Project Refences as below



Add the following front-End C# Code to Perform CRUD operation by Calling methods in DALOperations.cs class of DataAccessLayer Project.

do

{

Console.WriteLine("1.Read Employee Records");

Console.WriteLine("2.Create Employee Record");

Console.WriteLine("3.Update Employee Data");

Console.WriteLine("4.Delete Employee Data");

Console.WriteLine("5.Exit");

int choice = int.Parse(Console.ReadLine());

switch (choice)

{

default:

Console.WriteLine("Wrong input.. Please give 1->5 only");

break;

case 1:

List<Employee> employees = DALOperations.ReadEmployeeRecords();

foreach(Employee emp in employees)

{

Console.Write("{0,-6}{1}{2}{3}{4}\n", emp.EmpID,emp.EmpName,emp.Department,emp.Designation,emp.Salary);

}

break;

case 2: //Create

Employee emprec = new Employee();

Console.WriteLine("Enter Employee Name");

emprec.EmpName = Console.ReadLine();

Console.WriteLine("Enter Employee Department");

emprec.Department = Console.ReadLine();

Console.WriteLine("Enter Employee Designation");

emprec.Designation = Console.ReadLine();

Console.WriteLine("Enter Employee Salary");

emprec.Salary = int.Parse(Console.ReadLine());

int RowsInserted = DALOperations.SaveEmployeeRecord(emprec);

if(RowsInserted>0)

Console.WriteLine("Record insertion is successful...");

else

Console.WriteLine("Record insertion is unsuccessful..");

break;

case 3: //Update

Employee NewObj = new Employee();

Console.WriteLine("Enter Employee ID you want to update");

int EmpID = int.Parse(Console.ReadLine());

Console.WriteLine("Enter Employee Name");

NewObj.EmpName = Console.ReadLine();

Console.WriteLine("Enter Employee Department");

NewObj.Department = Console.ReadLine();

Console.WriteLine("Enter Employee Designation");

NewObj.Designation = Console.ReadLine();

Console.WriteLine("Enter Employee Salary");

NewObj.Salary = int.Parse(Console.ReadLine());

int RowsUpdated = DALOperations.UpdateEmployeeRecord(EmpID,NewObj);

if (RowsUpdated > 0)

Console.WriteLine("Record Updation is successful...");

else

Console.WriteLine("Record Updation is unsuccessful..");

break;

case 4: //Delete

Console.WriteLine("Enter Employee ID you want to delete");

int EmpIDToDelete = int.Parse(Console.ReadLine());

int RowsDeleted = DALOperations.DeleteEmployeeRecord(EmpIDToDelete);

if (RowsDeleted > 0)

Console.WriteLine("Record Deletion is successful...");

else

Console.WriteLine("Record Deletion is unsuccessful..");

break;

case 5:

Console.WriteLine("Thank you..Press any key");

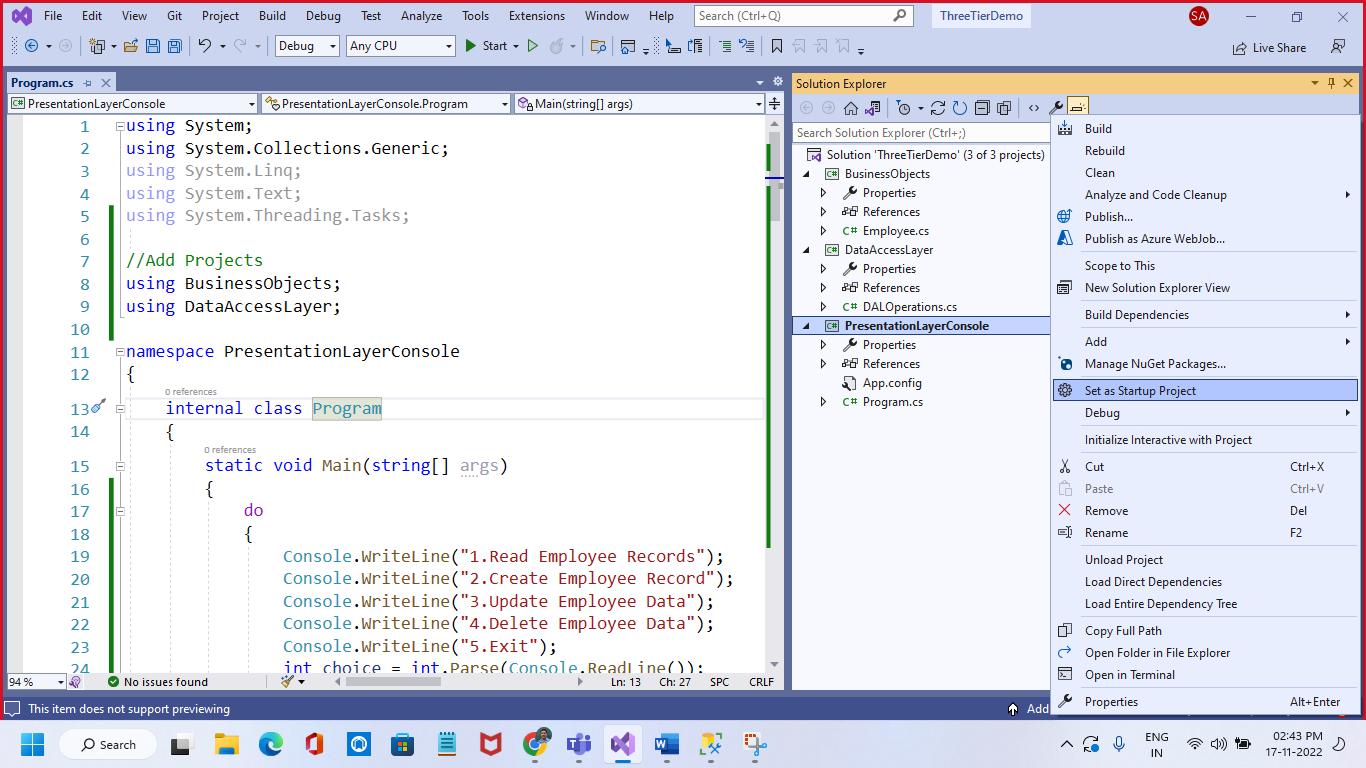
Console.ReadKey();

return;

}

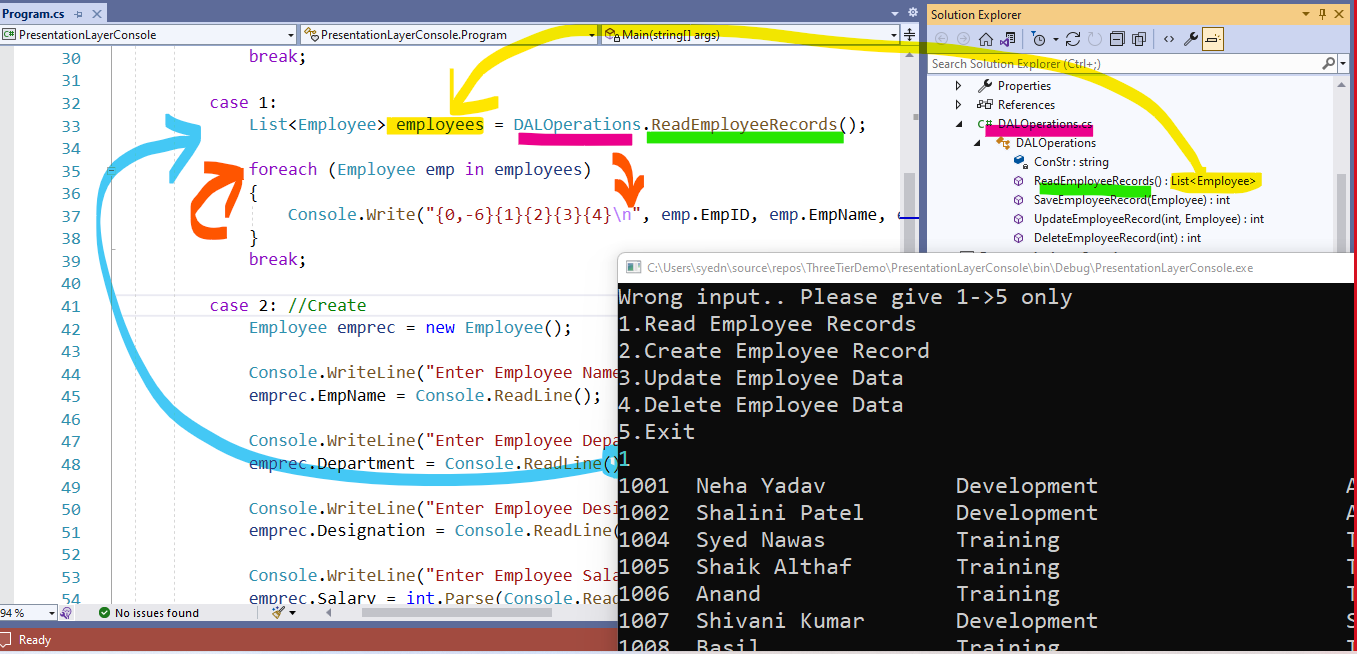
} while (true);

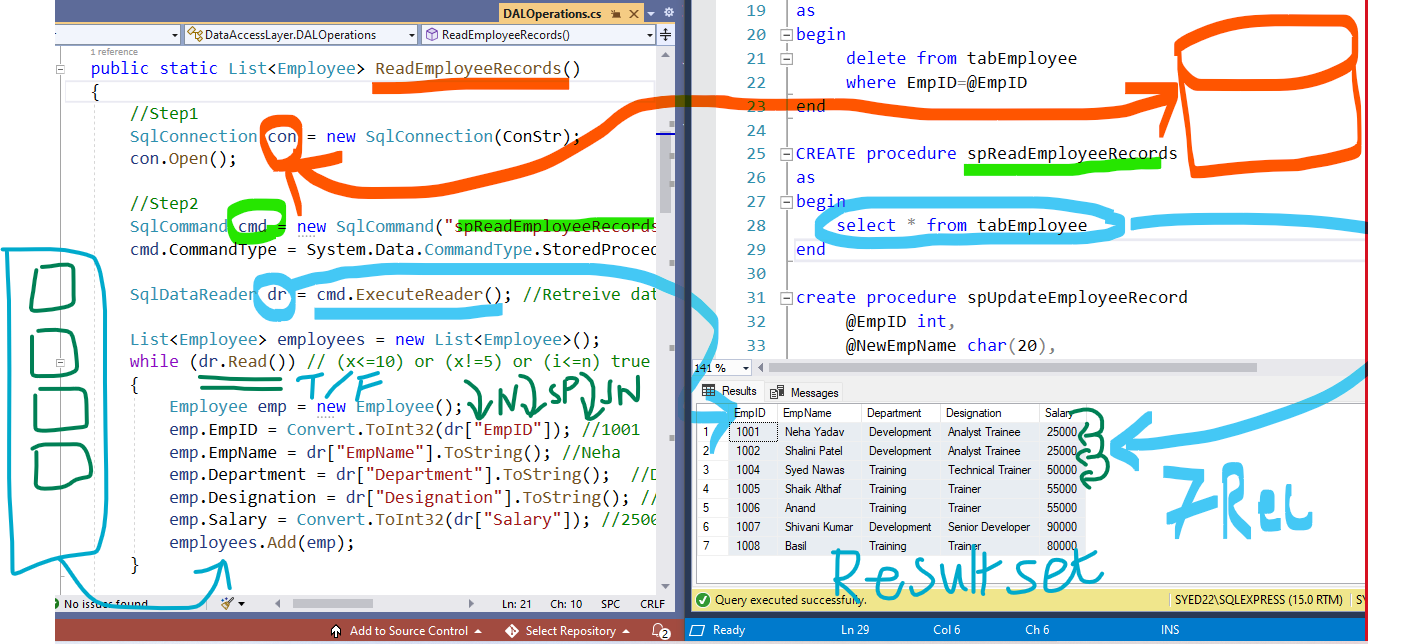
Note : Make as PresentationLayerConsole Project as Startup object



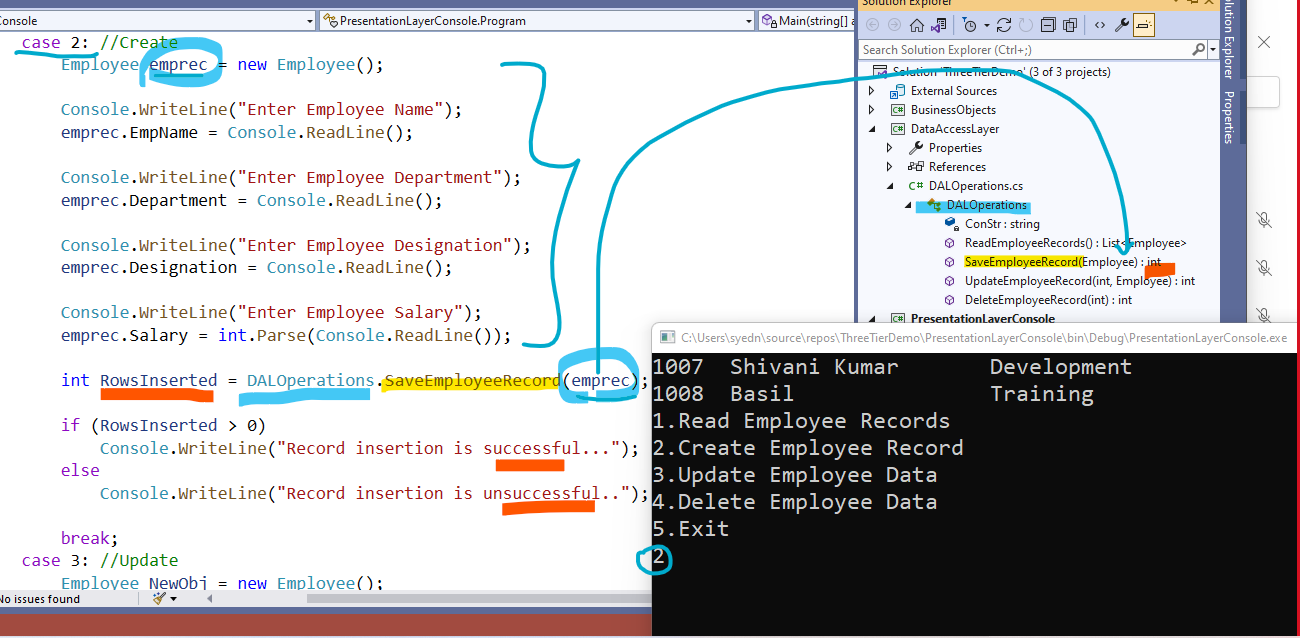
Output

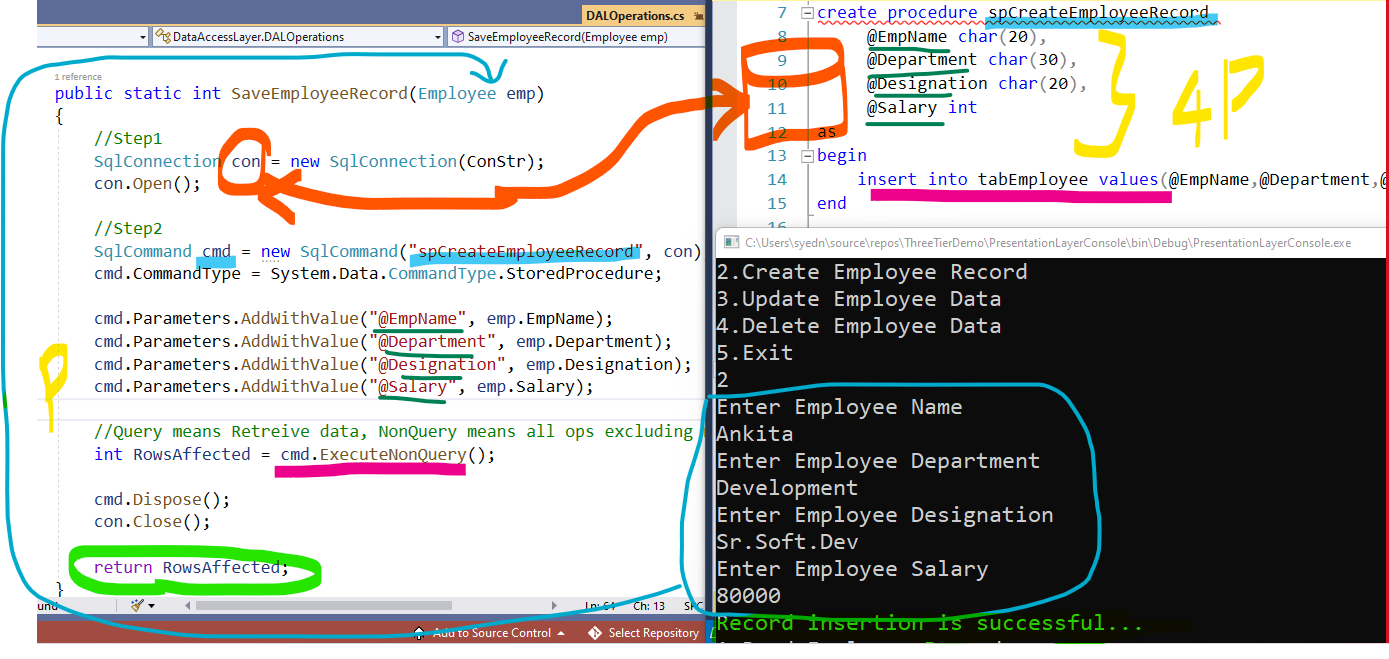
Read - Selection



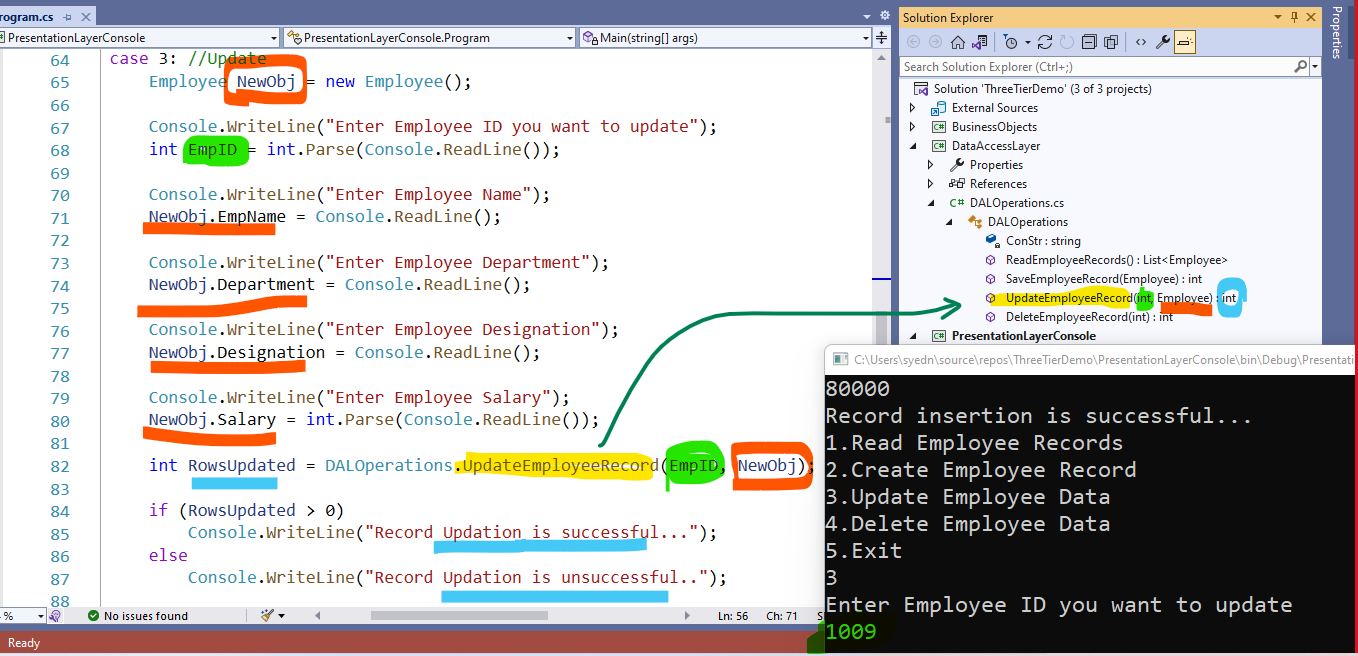


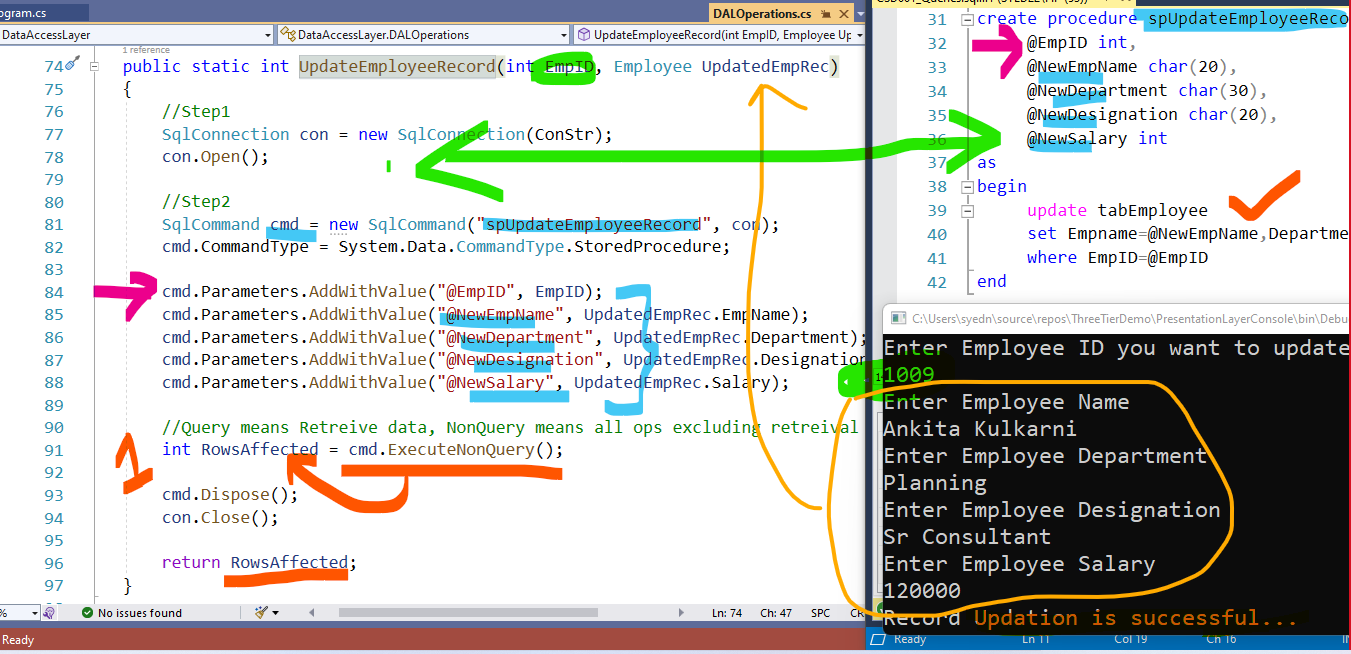
Create – By Insertion





Update Operation





Deletion – Delete

