Console With EF core DBFirst approach:

Create database in SQL Server

I have given Databasename as EFcoreDB1;

Create table tblDepartment

(

ID int primary key,

DepartmentName nvarchar(50),

Location nvarchar(50),

DepartmentHead nvarchar(50)

)

Go

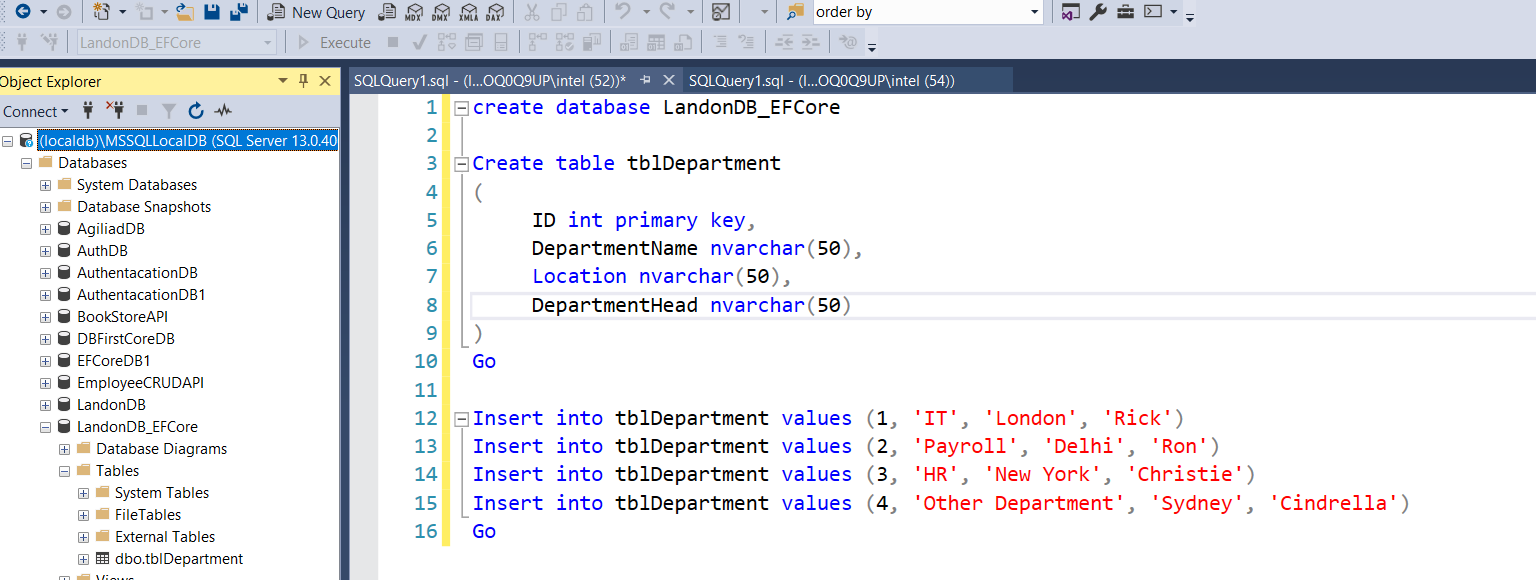
Insert into tblDepartment values (1, 'IT', 'London', 'Rick')

Insert into tblDepartment values (2, 'Payroll', 'Delhi', 'Ron')

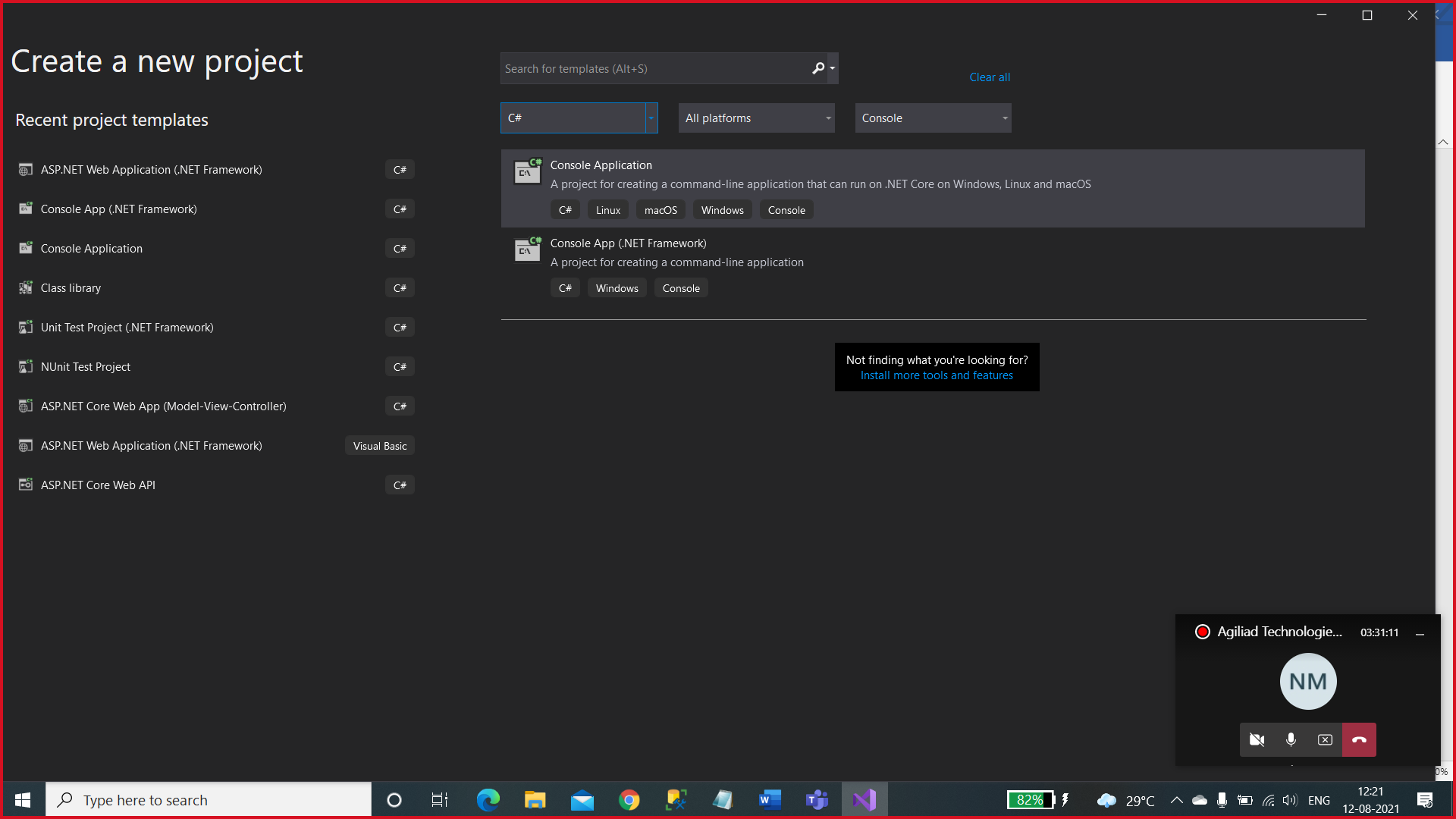
Insert into tblDepartment values (3, 'HR', 'New York', 'Christie')

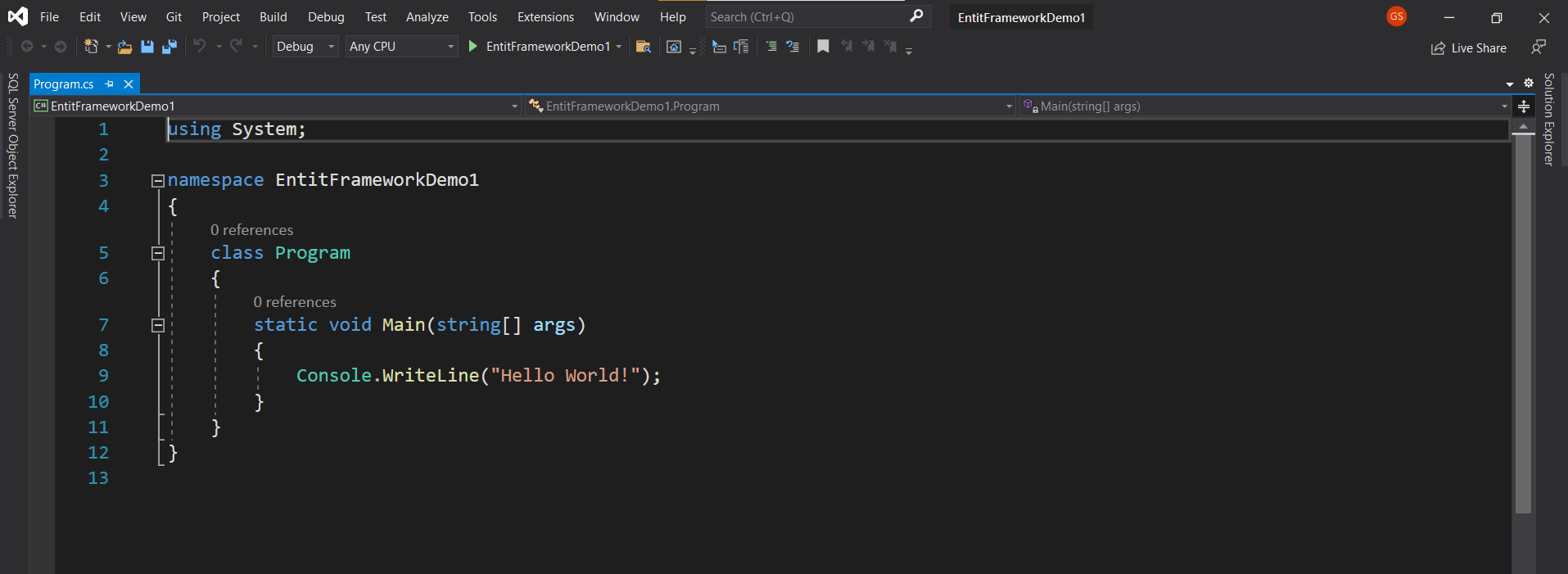
Insert into tblDepartment values (4, 'Other Department', 'Sydney', 'Cindrella')

Go



Create console app with .NET core





Install EF core in console app

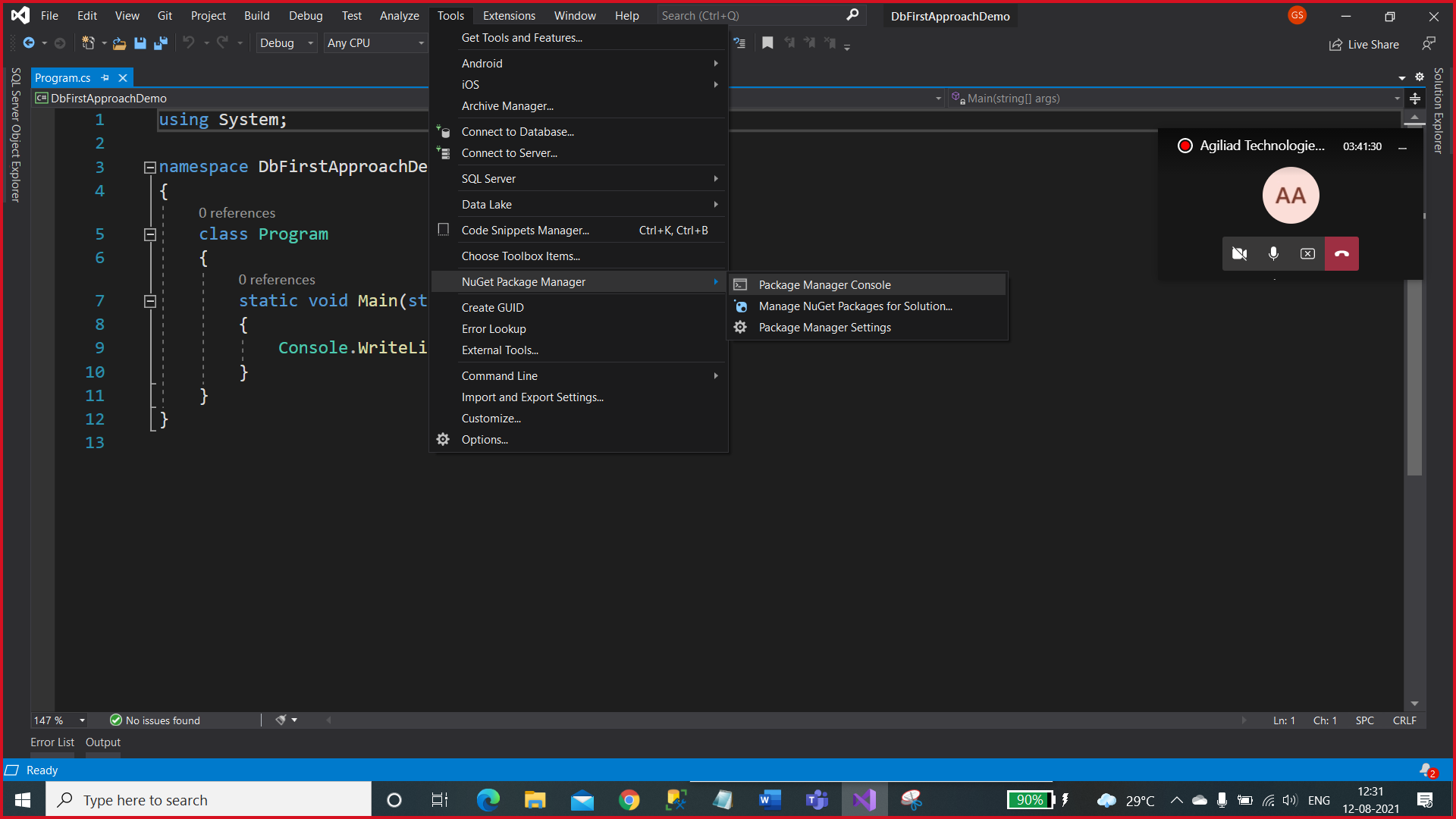
Goto 🡺 Tools🡺 Nuget Package Manager 🡺 Package Manage solution

Browse tab 🡺 Microsoft.EntityFrameworkcore.Sqlserver 🡺 select the stable version(5.0.9)

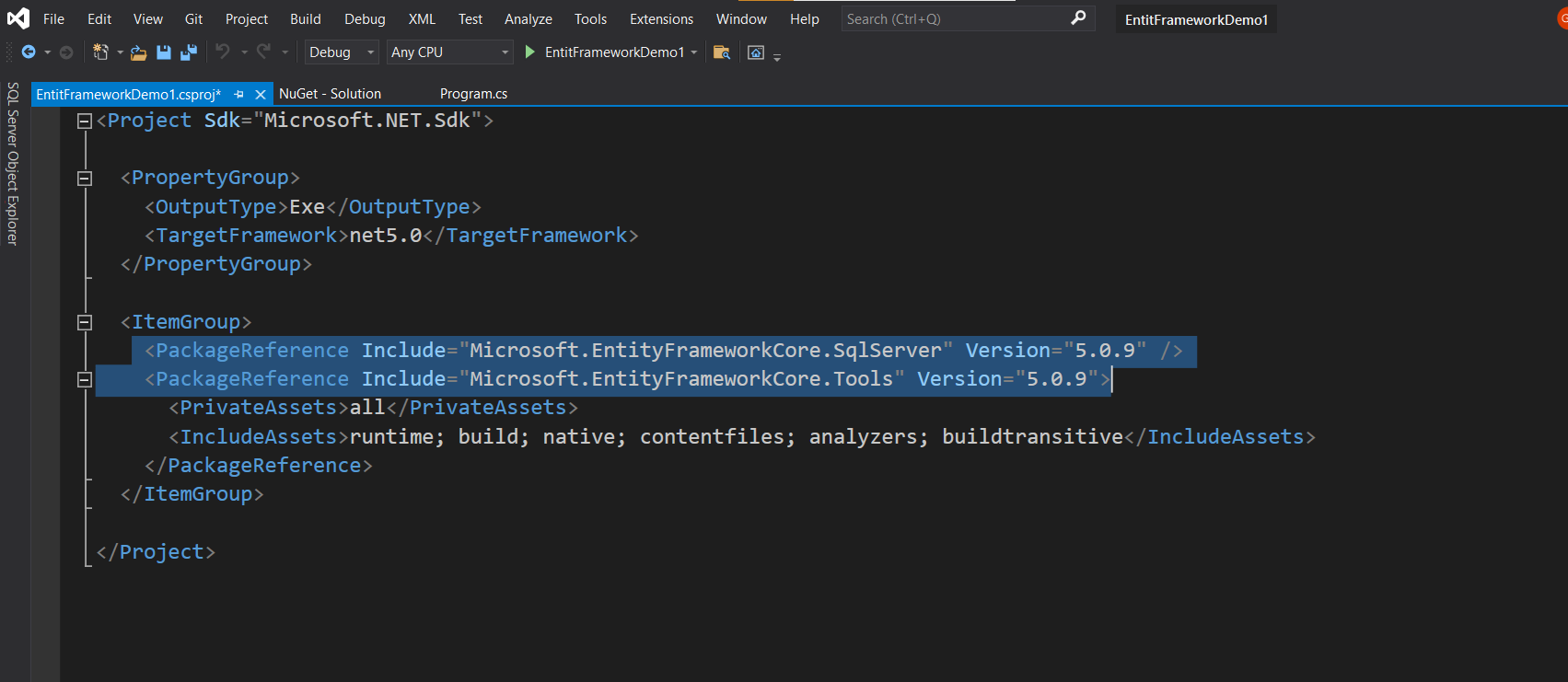
Accept the license agreement.

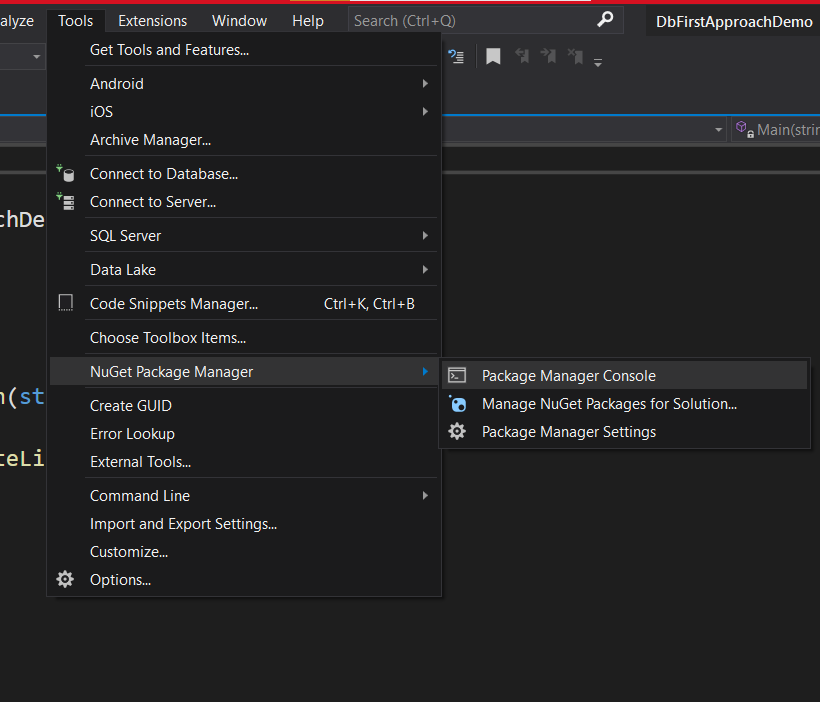
Microsoft.EntityFrameworkcore.Sqlserver

Microsoft.EntityFrameworkcore.tools



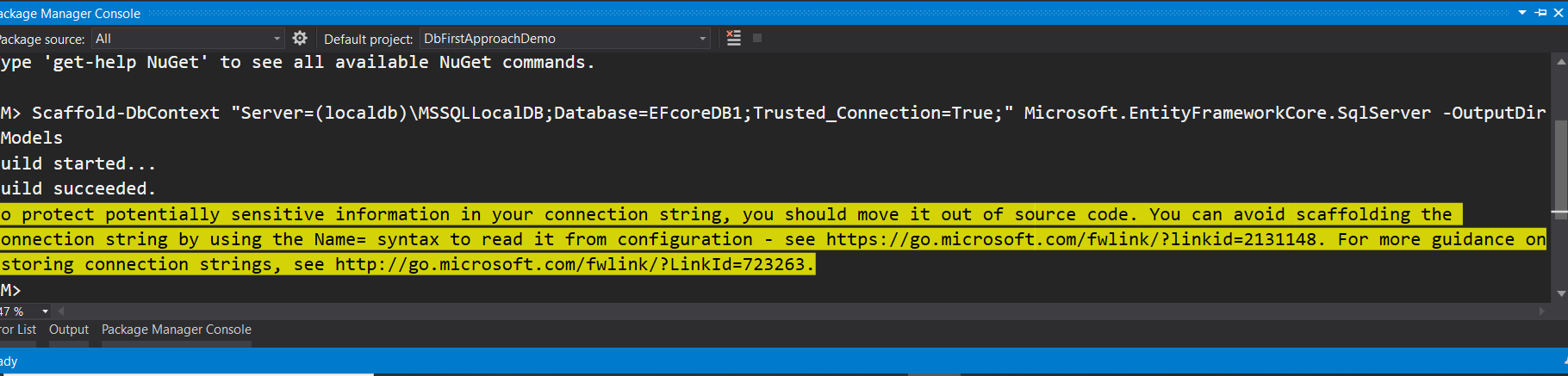
Right click on the Project in solution Explorer and Edit csproj file

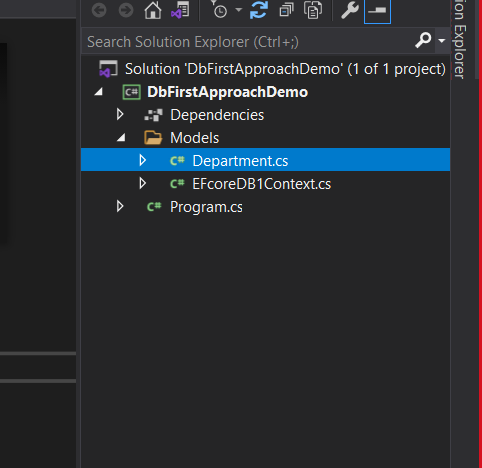




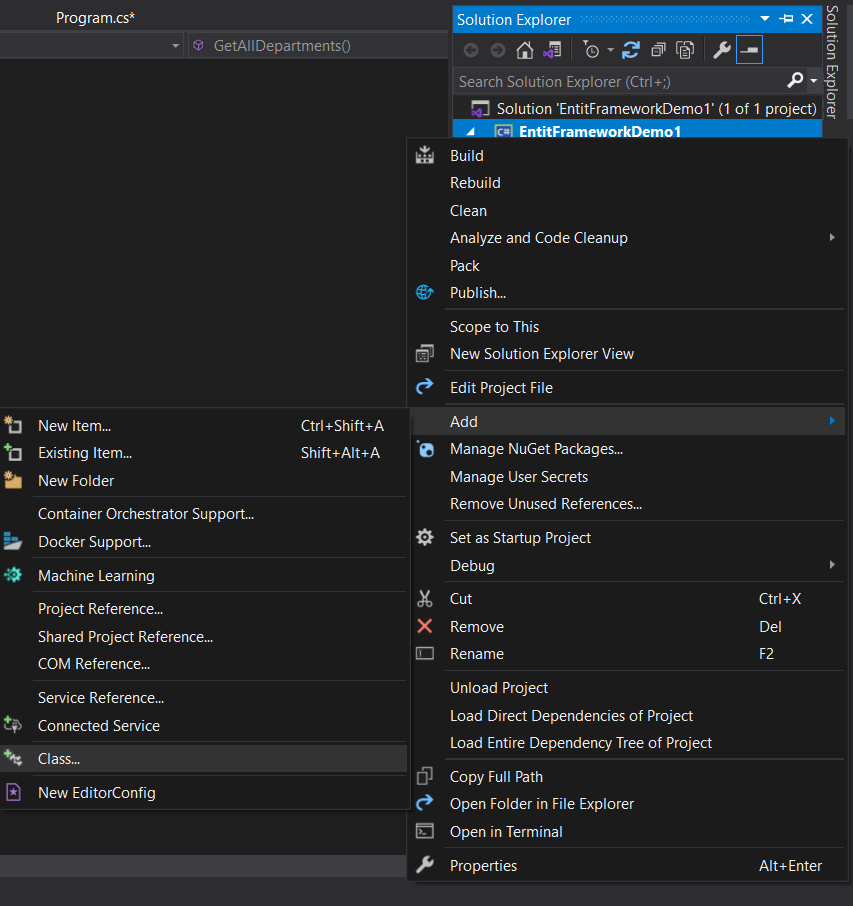
In Package Manager Console

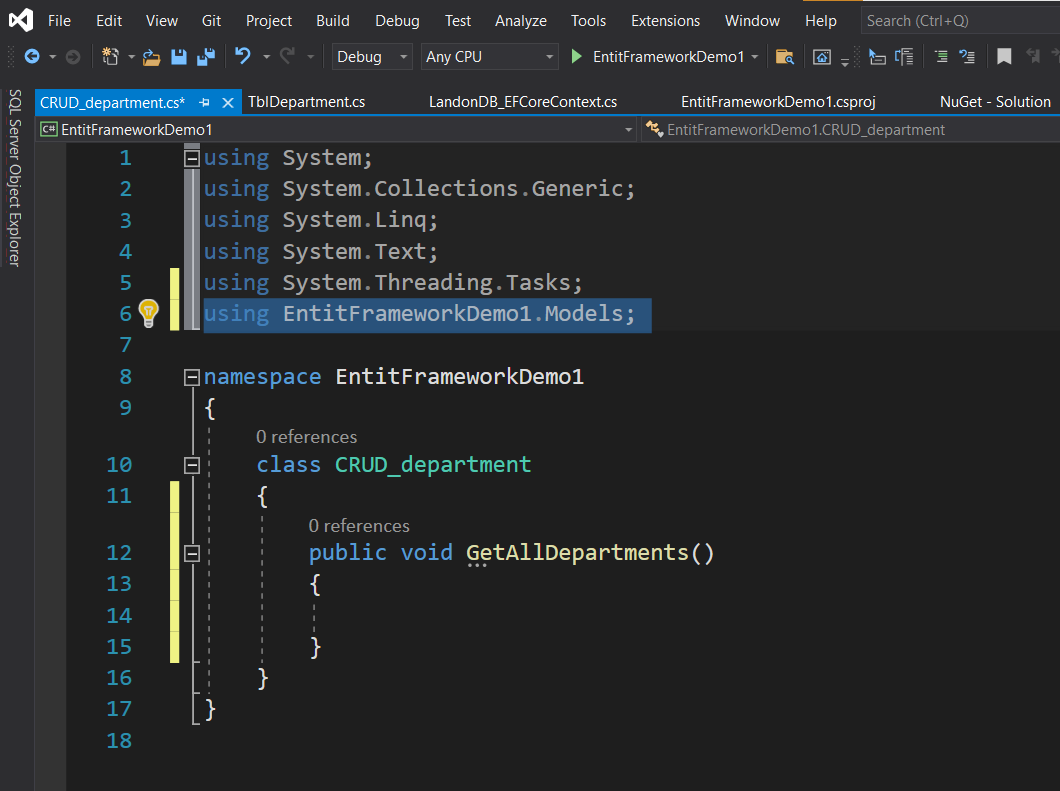
Scaffold-DbContext "Server=(localdb)\MSSQLLocalDB;Database=EFCorecrud;Trusted\_Connection=True;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models

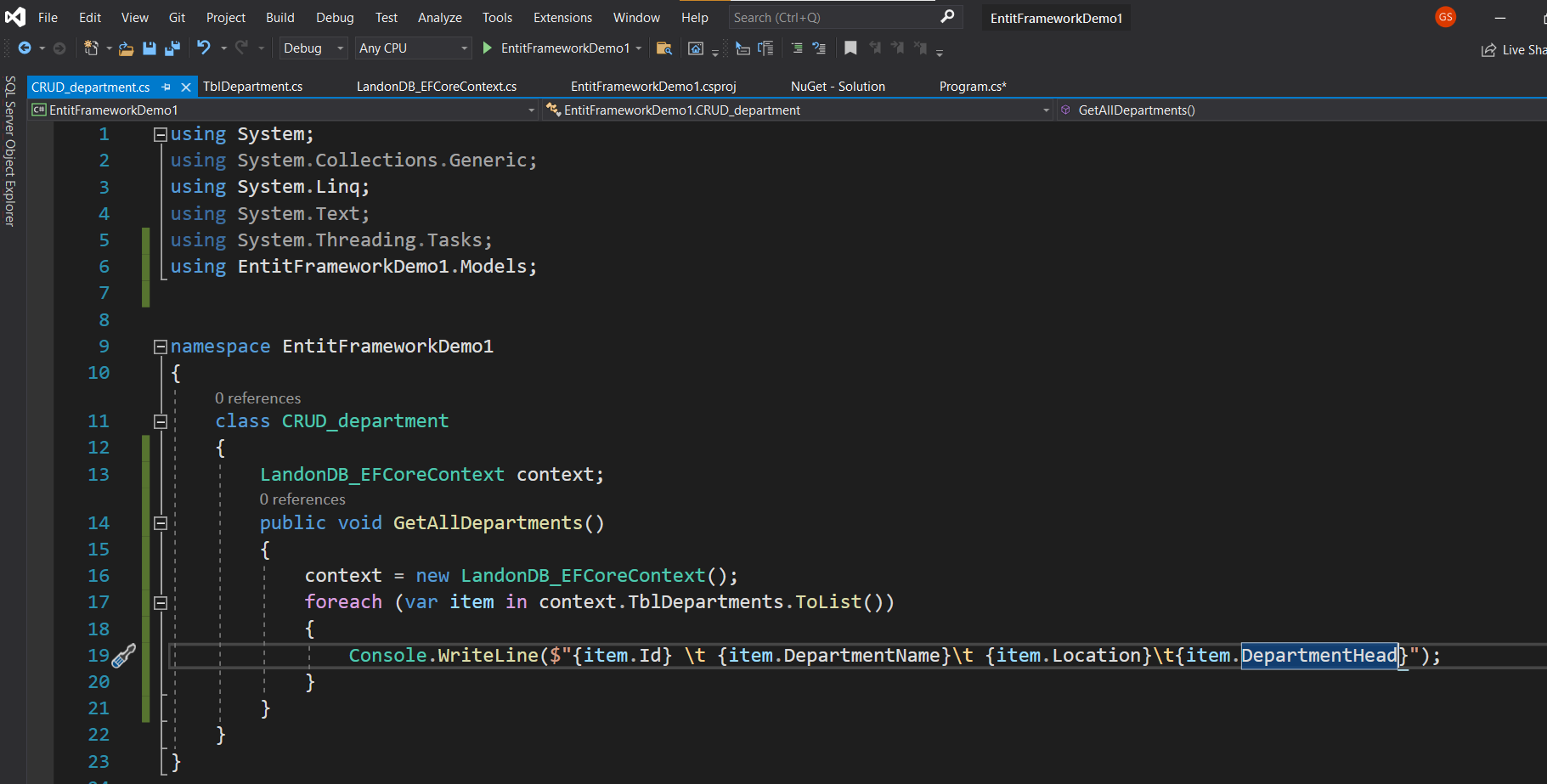




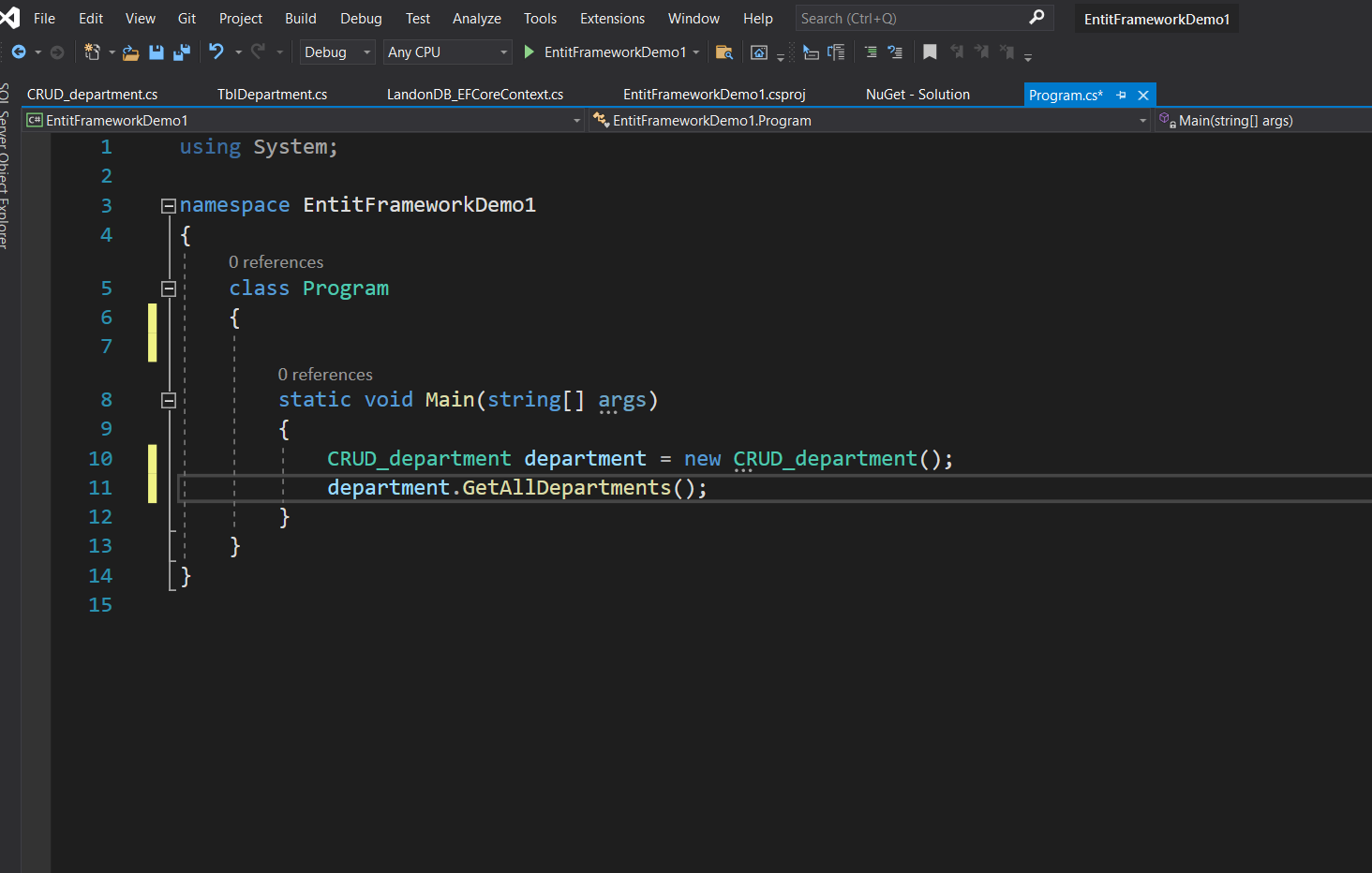
Right click on project and add new class with CRUd\_department (follow the below steps if any assistance)

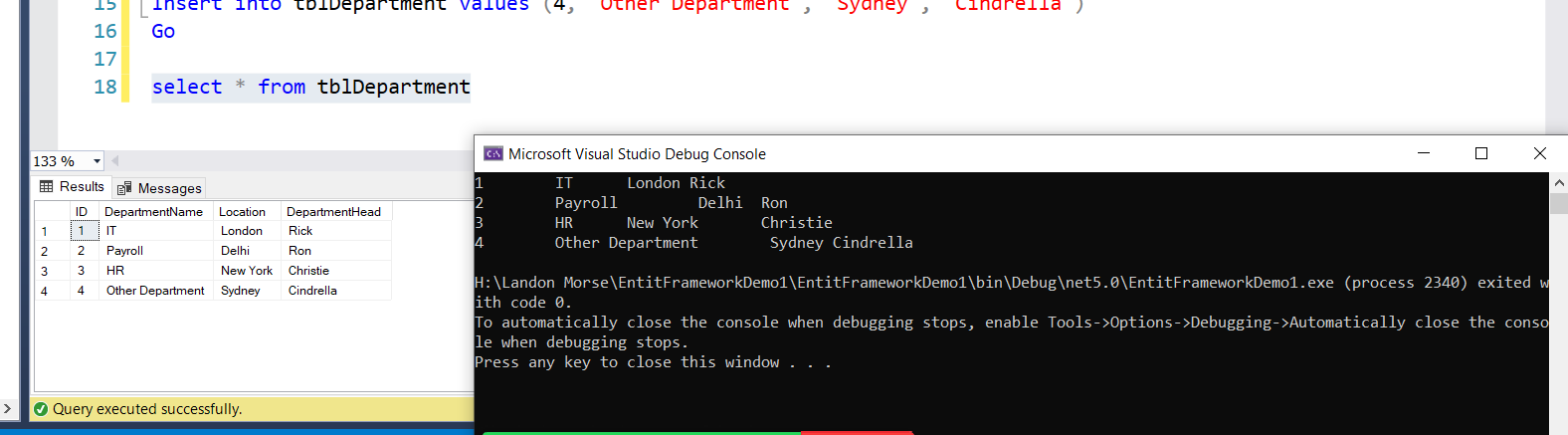


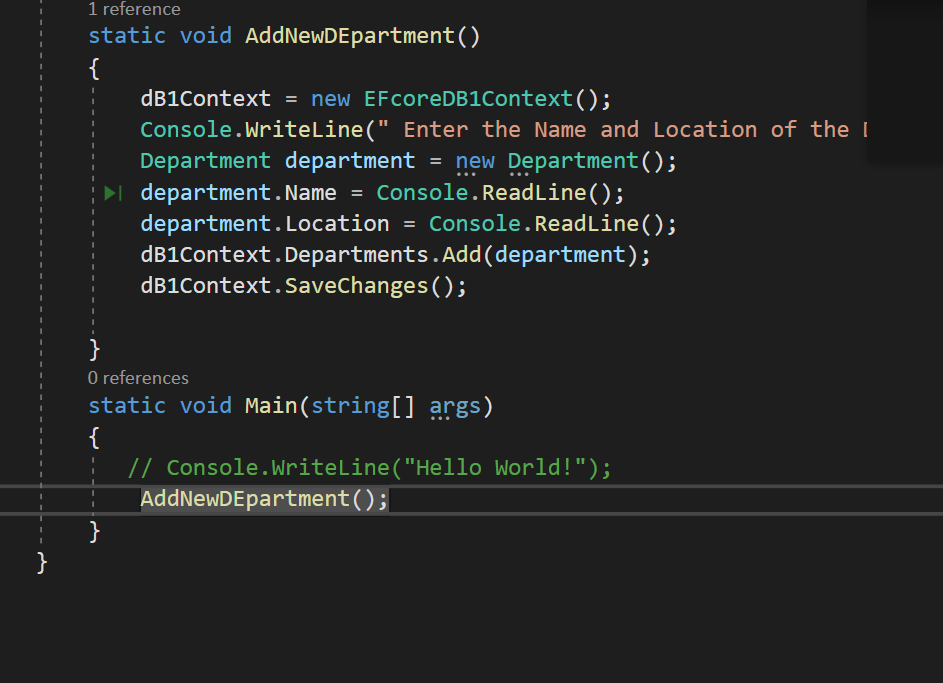




In Program.cs we need to GetAllDepartment method and Execute







static void AddNewDEpartment()

{

dB1Context = new EFcoreDB1Context();

Console.WriteLine(" Enter the Name and Location of the Department");

Department department = new Department();

department.Name = Console.ReadLine();

department.Location = Console.ReadLine();

dB1Context.Departments.Add(department);

dB1Context.SaveChanges();

}

static void Main(string[] args)

{

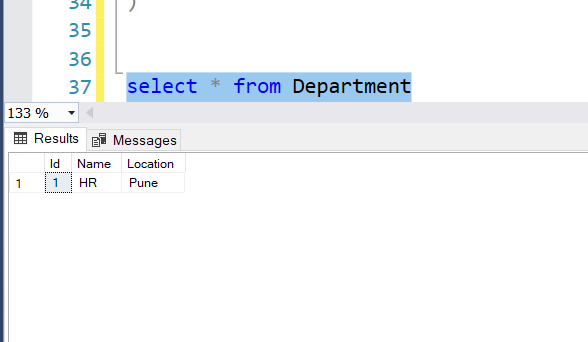
// Console.WriteLine("Hello World!");

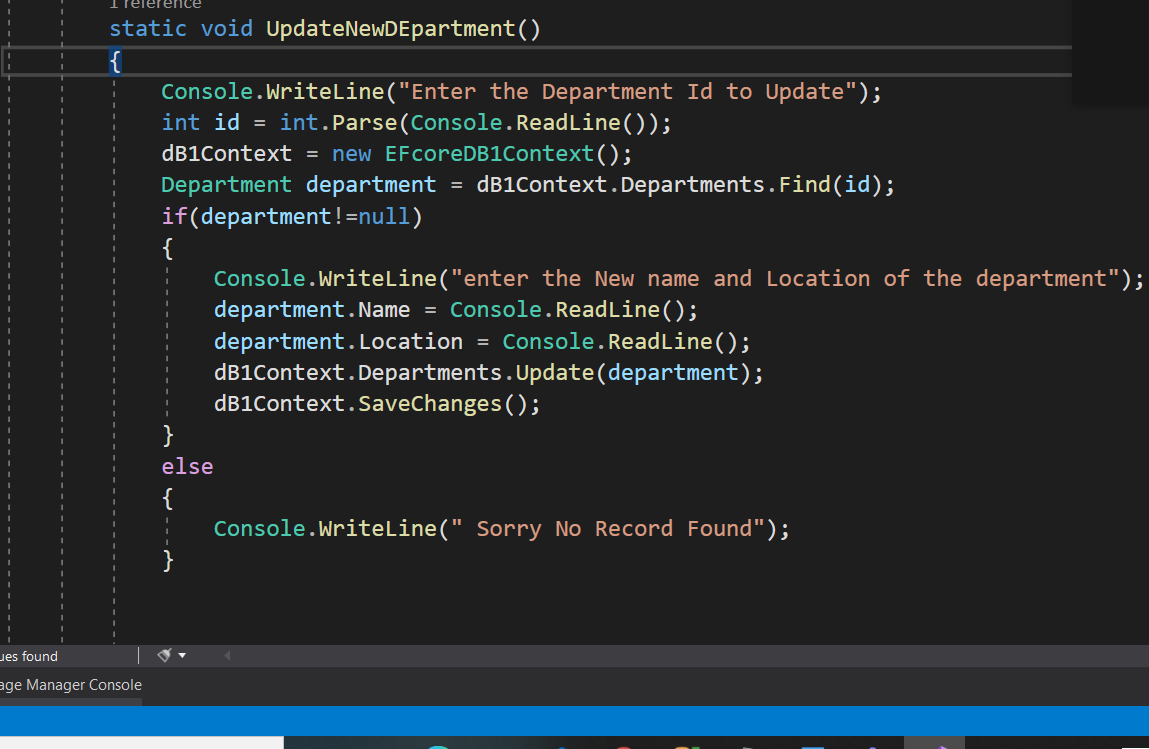
AddNewDEpartment();

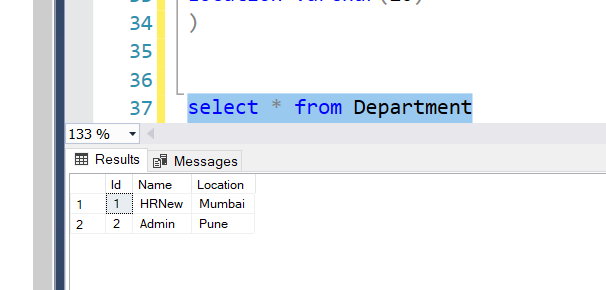
}

}

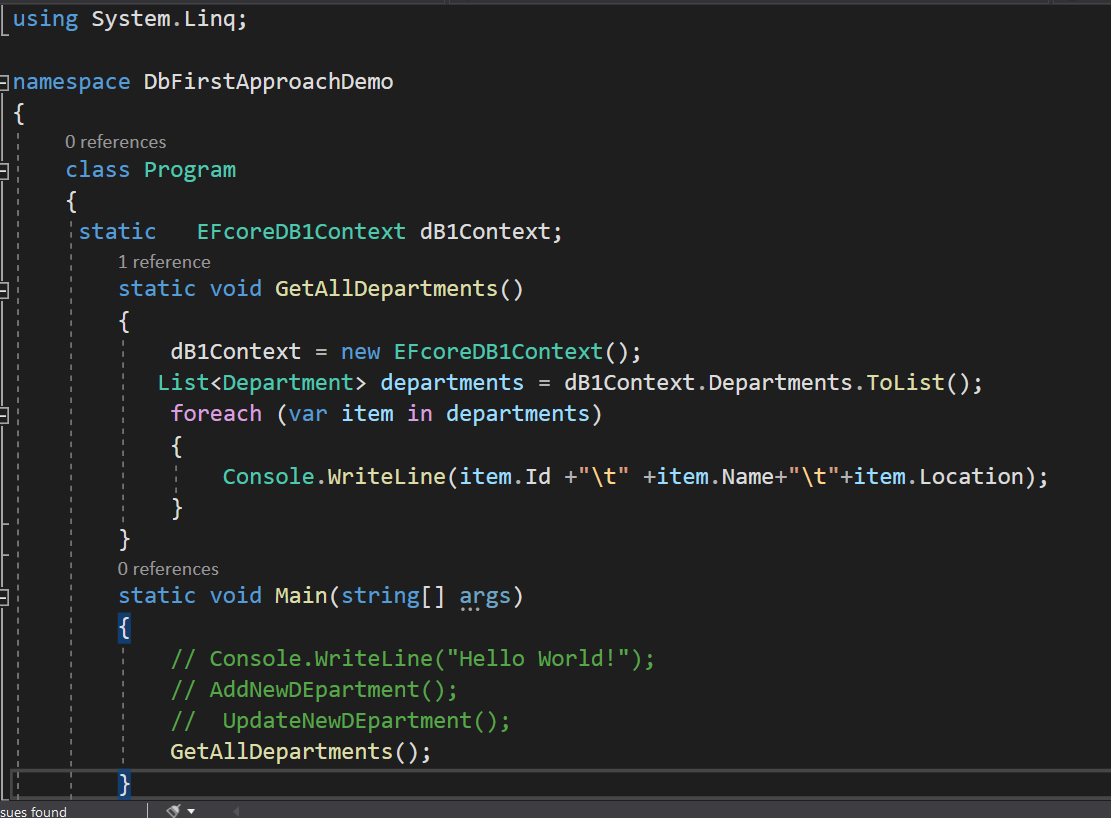
}

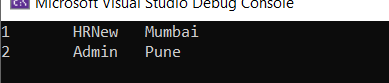






To read Data from SQL Server

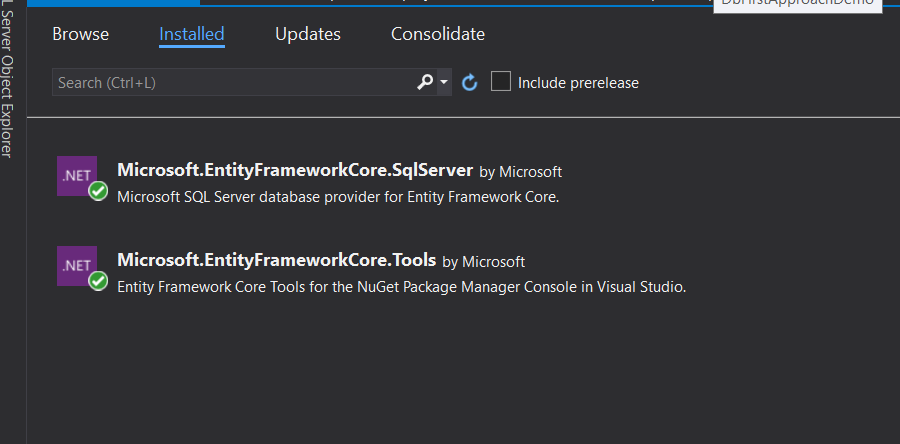


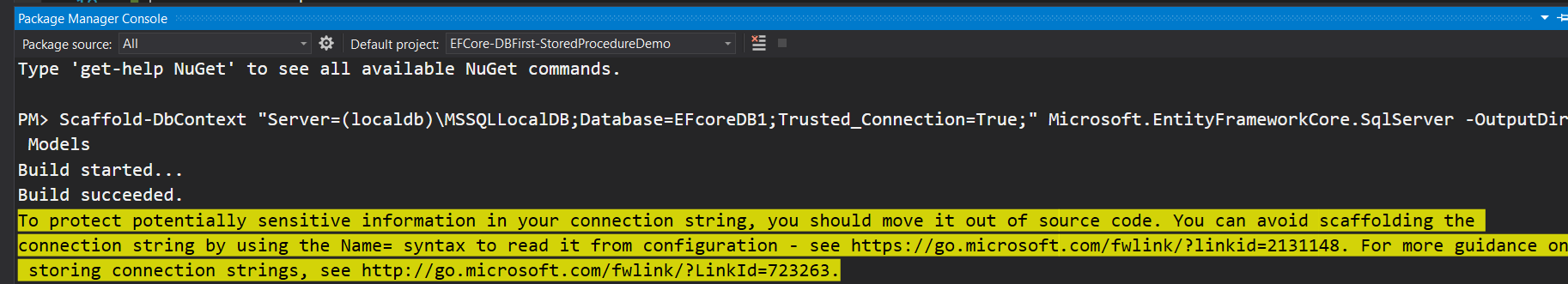


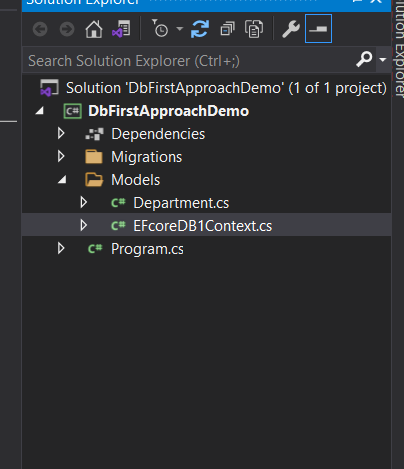
DBFirst Approach with Stored Procedure

Create New Project

Install 2 package from Nuget Package

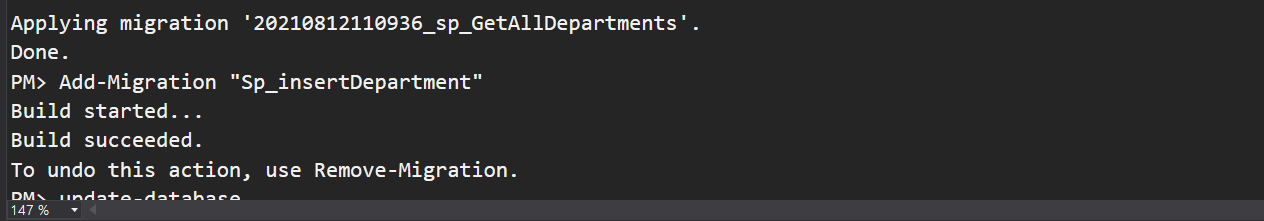


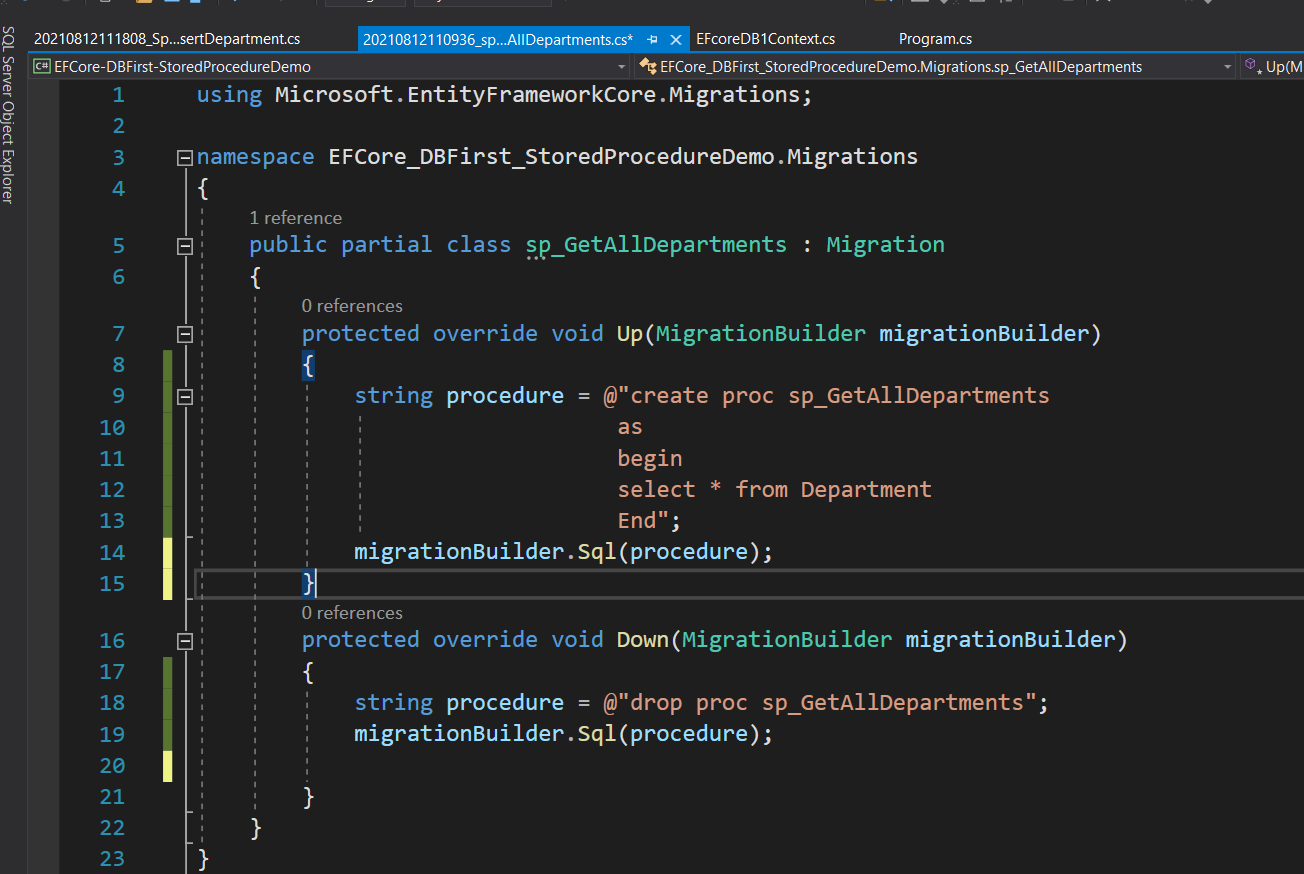


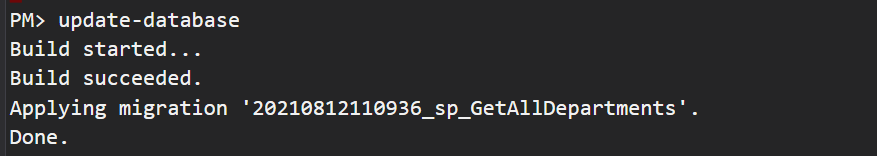


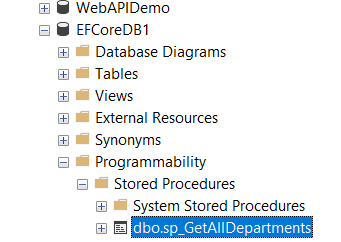
To create stored procedure from console(or) any app

Add-Migration “sp\_getAllDepartments”





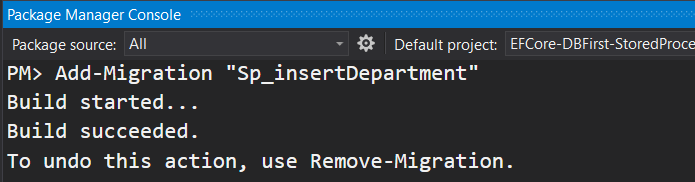


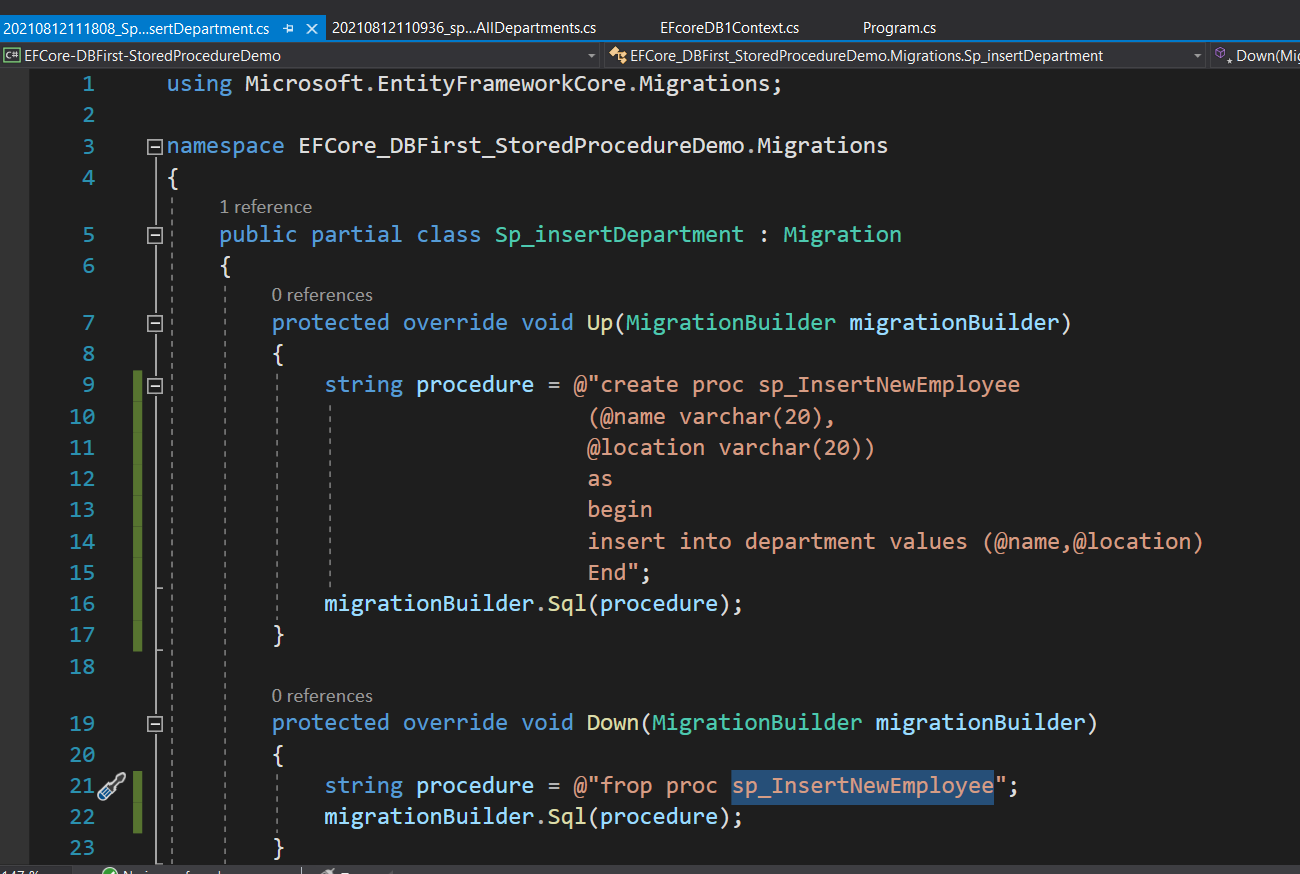


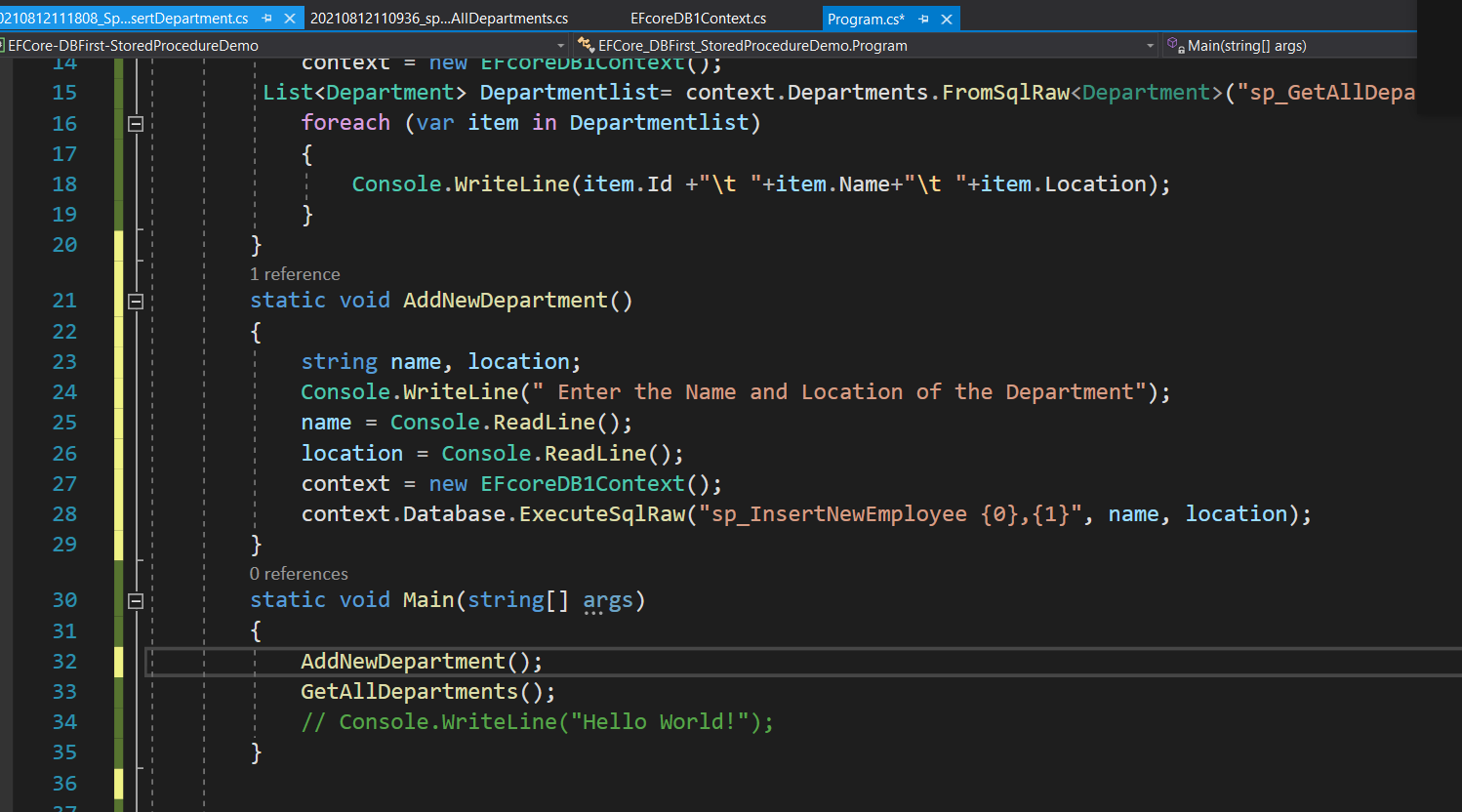


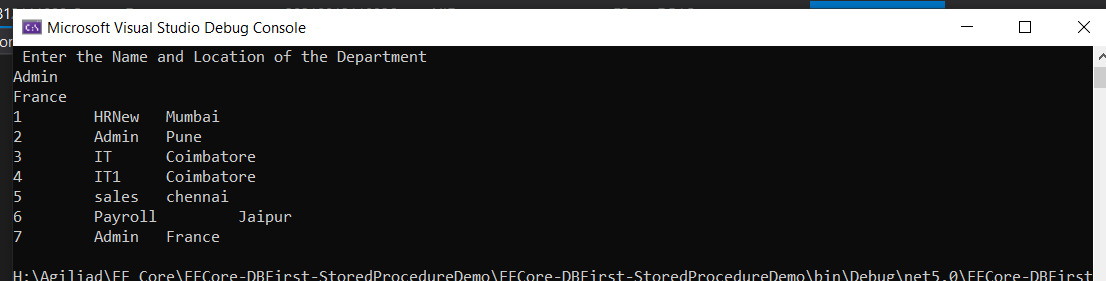


Store procedure for insertrecord









Same as above we can do for other procedures

using System;

using Microsoft.EntityFrameworkCore;

using System.Linq;

using EFCore\_DBFirst\_StoredProcedureDemo.Models;

using System.Collections.Generic;

namespace EFCore\_DBFirst\_StoredProcedureDemo

{

class Program

{

static EFcoreDB1Context context;

static void GetAllDepartments()

{

context = new EFcoreDB1Context();

List<Department> Departmentlist= context.Departments.FromSqlRaw<Department>("sp\_GetAllDepartments").ToList();

foreach (var item in Departmentlist)

{

Console.WriteLine(item.Id +"\t "+item.Name+"\t "+item.Location);

}

}

static void AddNewDepartment()

{

string name, location;

Console.WriteLine(" Enter the Name and Location of the Department");

name = Console.ReadLine();

location = Console.ReadLine();

context = new EFcoreDB1Context();

context.Database.ExecuteSqlRaw("sp\_InsertNewEmployee {0},{1}", name, location);

}

static void Main(string[] args)

{

AddNewDepartment();

GetAllDepartments();

// Console.WriteLine("Hello World!");

}

}

}