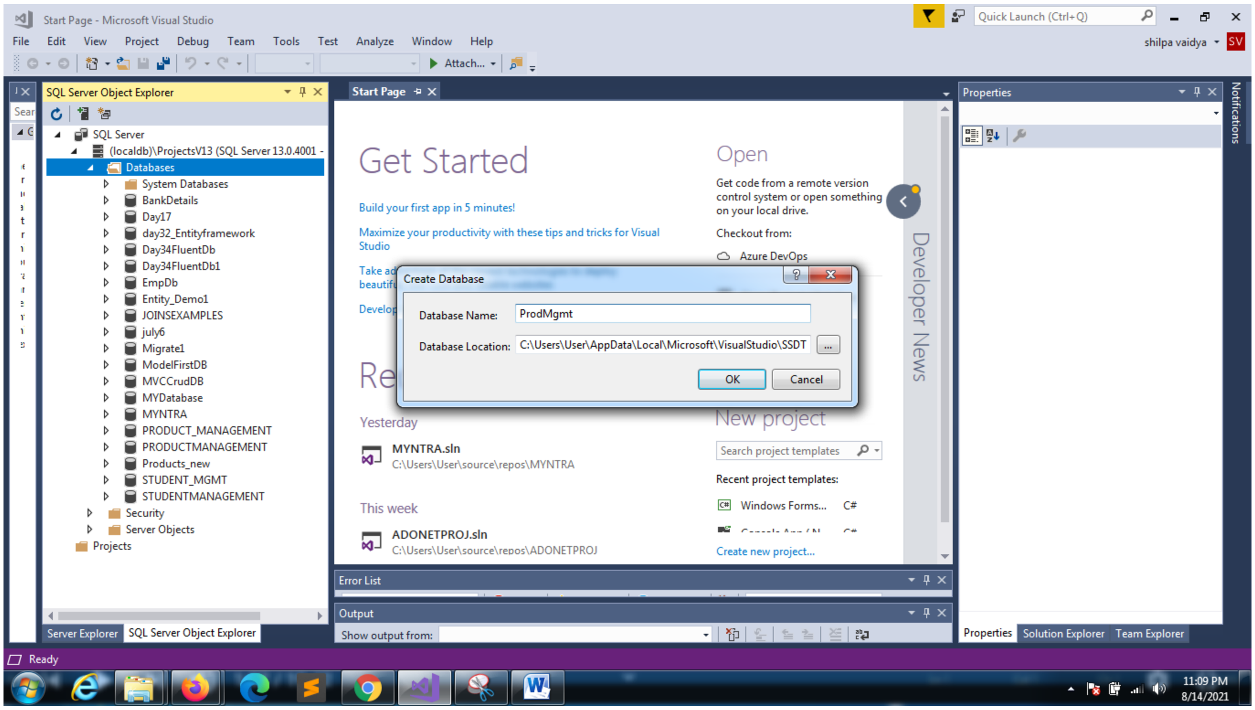
Create database

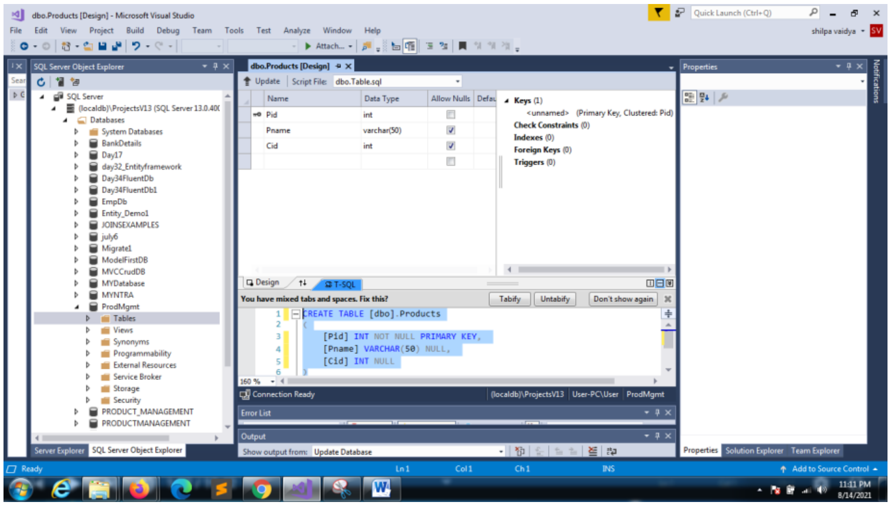


Create Database

Database Name: ProdMgmt

Database Location: C:\Users\User\AppData\Local\Microsoft\VisualStudio\SSDT

```
CREATE TABLE [dbo].Products
(
	 [Pid] INT NOT NULL PRIMARY KEY,
	[Pname] VARCHAR(50) NULL,
	[Cid] INT NULL
)
```



dbo.Products [Design]

Script File: dbo.Table.sql

Name	Data Type	Allow Nulls	Default
Pid	int	☐	
Pname	varchar(50)	☑	
Cid	int	☑	
		☐	

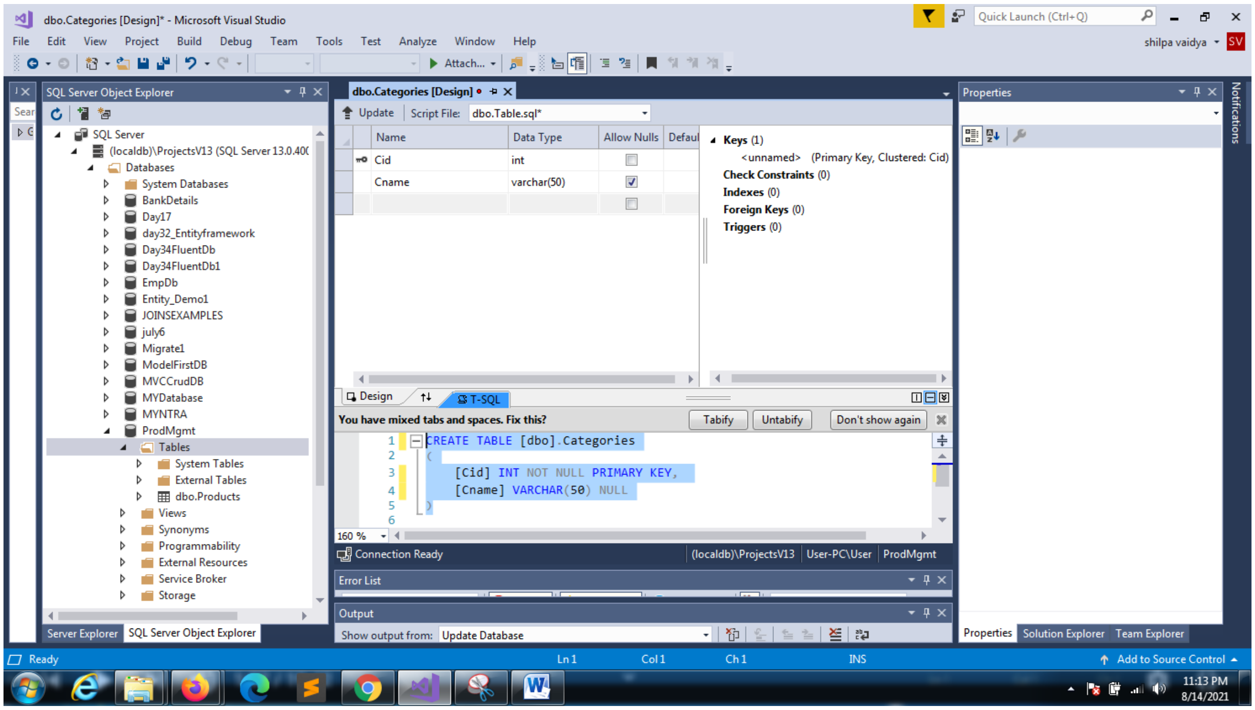
Keys (1)
<unnamed> (Primary Key, Clustered: Pid)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

```
CREATE TABLE [dbo].Products
(
	[Pid] INT NOT NULL PRIMARY KEY,
	[Pname] VARCHAR(50) NULL,
	[Cid] INT NULL
)
```

Connection Ready (localdb)\ProjectsV13 User-PC\User ProdMgmt

Show output from: Update Database

```
CREATE TABLE [dbo].Categories
(
        [Cid] INT NOT NULL PRIMARY KEY,
     [Cname] VARCHAR(50) NULL
)
```



- SQL Server
  - (localdb)\ProjectsV13 (SQL Server 13.0.400
    - Databases
      - System Databases
      - BankDetails
      - Day17
      - day32_Entityframework
      - Day34FluentDb
      - Day34FluentDb1
      - EmpDb
      - Entity_Demo1
      - JOINSEXAMPLES
      - july6
      - Migrate1
      - ModelFirstDB
      - MVCCrudDB
      - MYDatabase
      - MYNTRA
      - ProdMgmt
        - Tables
          - System Tables
          - External Tables
          - dbo.Products
        - Views
        - Synonyms
        - Programmability
        - External Resources
        - Service Broker
        - Storage

Name	Data Type	Allow Nulls	Defaul
Cid	int	□	
Cname	varchar(50)	☑	
		□	

Keys (1)
<unnamed> (Primary Key, Clustered: Cid)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

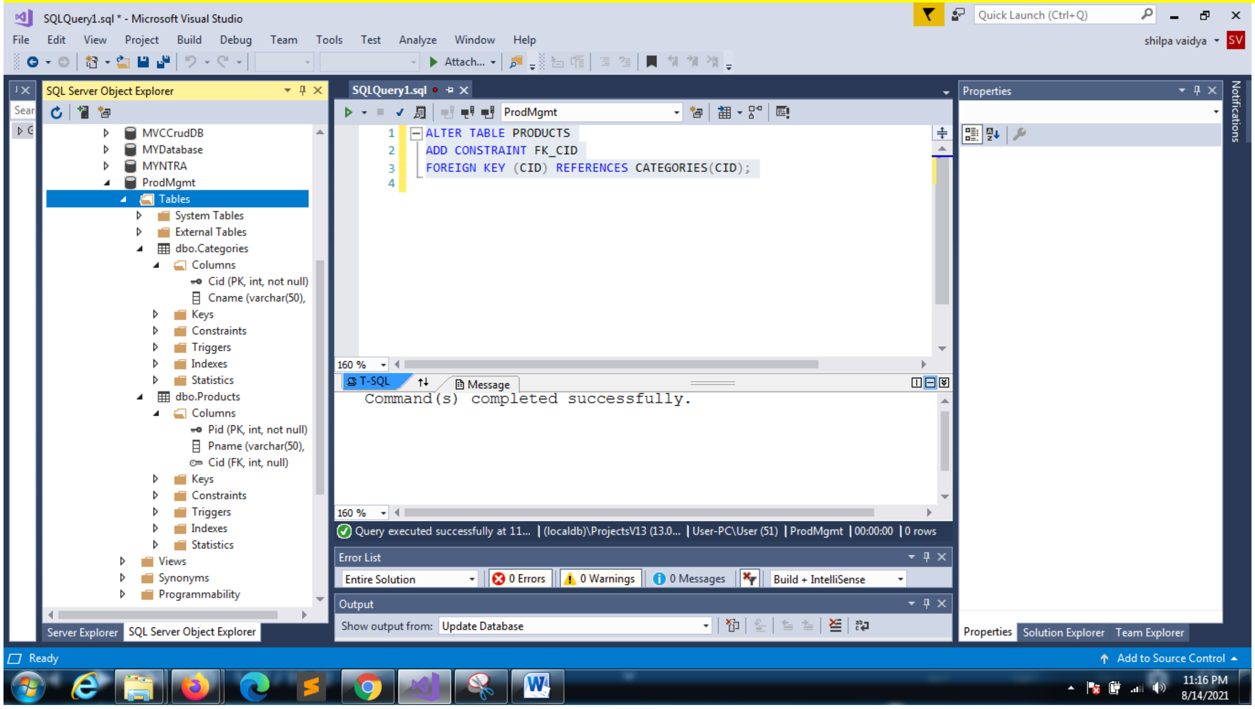
You have mixed tabs and spaces. Fix this?

```
CREATE TABLE [dbo].Categories
(
    [Cid] INT NOT NULL PRIMARY KEY,
    [Cname] VARCHAR(50) NULL
)
```

Connection Ready | (localdb)\ProjectsV13 | User-PC\User | ProdMgmt

Ready | Ln 1 | Col 1 | Ch 1 | INS

```
ALTER TABLE PRODUCTS
ADD CONSTRAINT FK_CID
FOREIGN KEY (CID) REFERENCES CATEGORIES(CID);
```

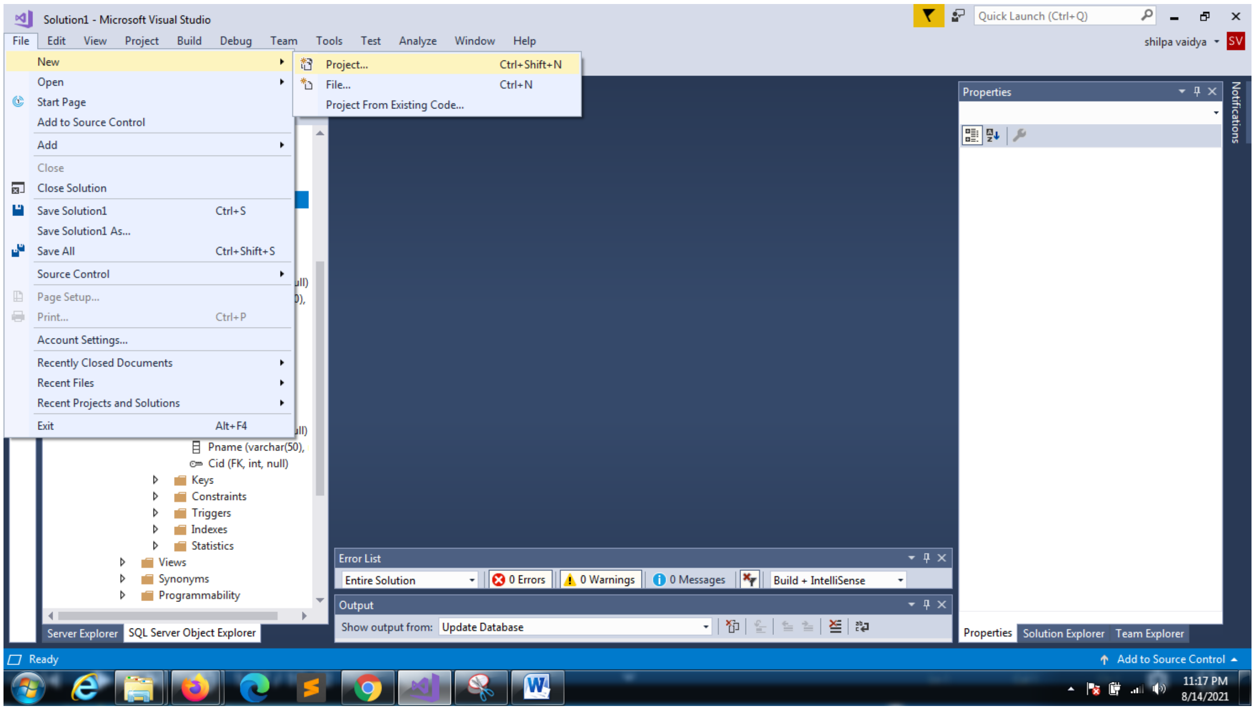


```
ALTER TABLE PRODUCTS
ADD CONSTRAINT FK_CID
FOREIGN KEY (CID) REFERENCES CATEGORIES(CID);
```

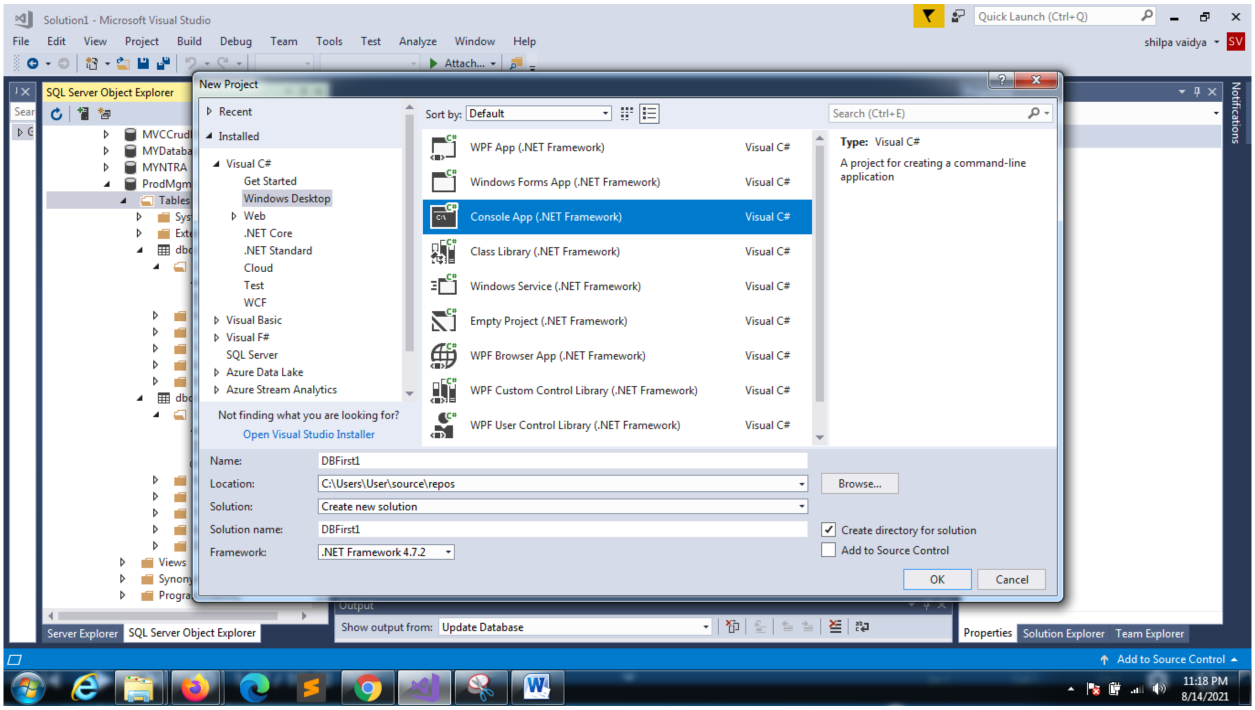
```
Command(s) completed successfully.
```

Query executed successfully at 11... | (localdb)\ProjectsV13 (13.0... | User-PC\User (51) | ProdMgmt | 00:00:00 | 0 rows

CREATE NEW PROJ



CONSOLE APPLICATION



Solution1 - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Quick Launch (Ctrl+Q)
shilpa vaidya SV
SQL Server Object Explorer
MVCCrudl
MYDataba
MYNTRA
ProdMgm
Tables
New Project
Recent
Installed
Visual C#
Get Started
Windows Desktop
Web
.NET Core
.NET Standard
Cloud
Test
WCF
Visual Basic
Visual F#
SQL Server
Azure Data Lake
Azure Stream Analytics
Not finding what you are looking for?
Open Visual Studio Installer
Sort by: Default
Search (Ctrl+E)
WPF App (.NET Framework) Visual C#
Windows Forms App (.NET Framework) Visual C#
Console App (.NET Framework) Visual C#
Class Library (.NET Framework) Visual C#
Windows Service (.NET Framework) Visual C#
Empty Project (.NET Framework) Visual C#
WPF Browser App (.NET Framework) Visual C#
WPF Custom Control Library (.NET Framework) Visual C#
WPF User Control Library (.NET Framework) Visual C#
Browse...
OK
Cancel
Views
Synony
Progra
Output
Show output from: Update Database
Server Explorer
SQL Server Object Explorer
Properties
Solution Explorer
Team Explorer
Add to Source Control
Notifications
11:18 PM
8/14/2021

Type: Visual C#

A project for creating a command-line application

Name: DBFirst1

Location: C:\Users\User\source\repos

Solution: Create new solution

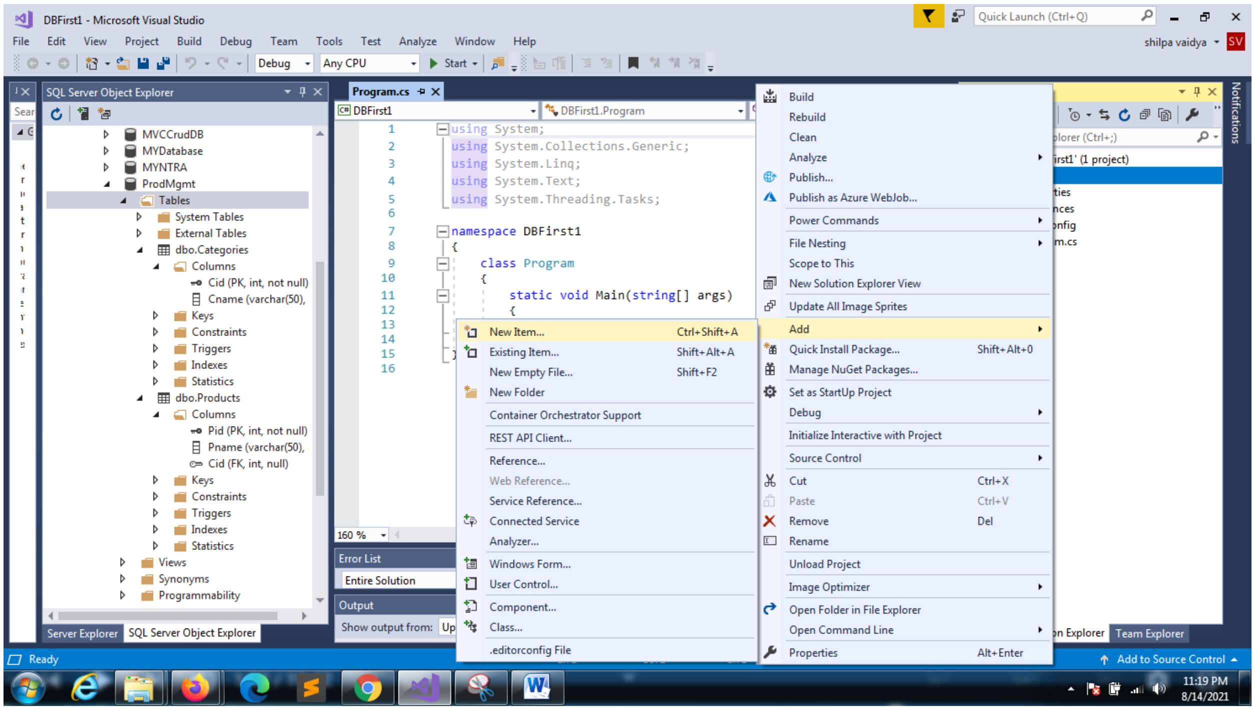
Solution name: DBFirst1

Framework: .NET Framework 4.7.2

☑ Create directory for solution

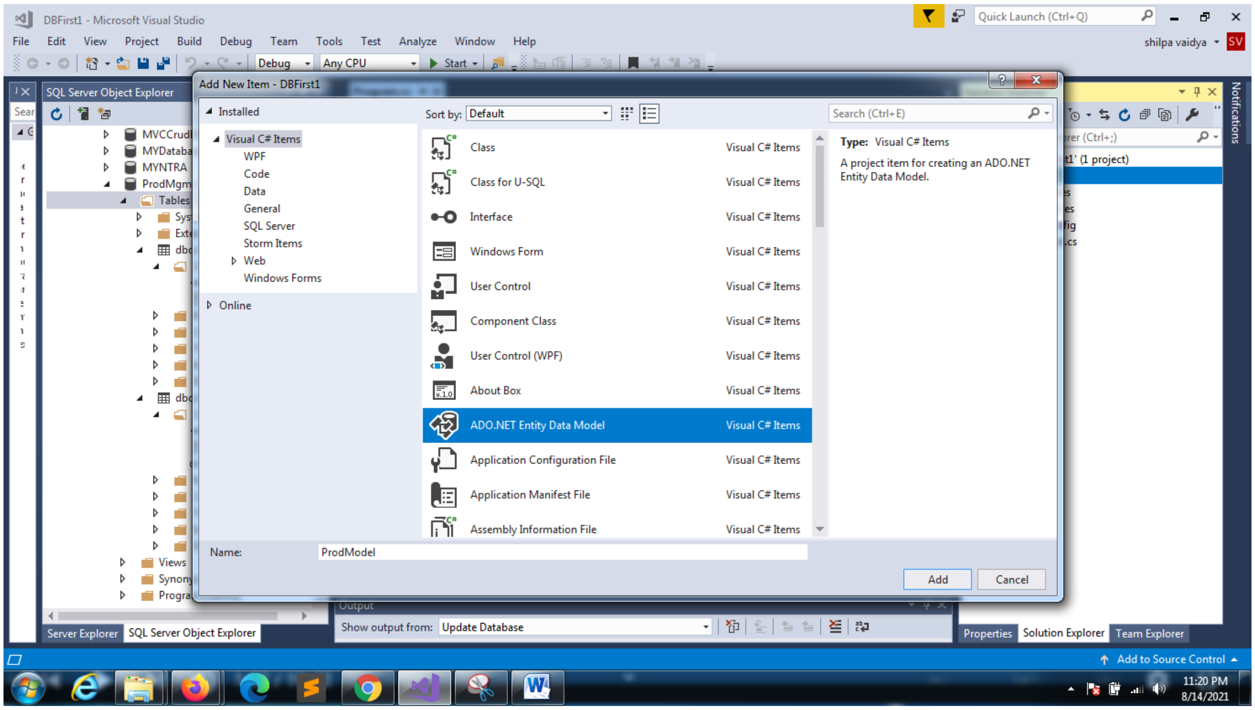
□ Add to Source Control

Add new project item



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DBFirst1
{
    class Program
    {
        static void Main(string[] args)
        {
```

SELECT ADO.NET



Add New Item - DBFirst1

Installed
- Visual C# Items
  - WPF
  - Code
  - Data
  - General
  - SQL Server
  - Storm Items
  - Web
  - Windows Forms
- Online

Sort by: Default

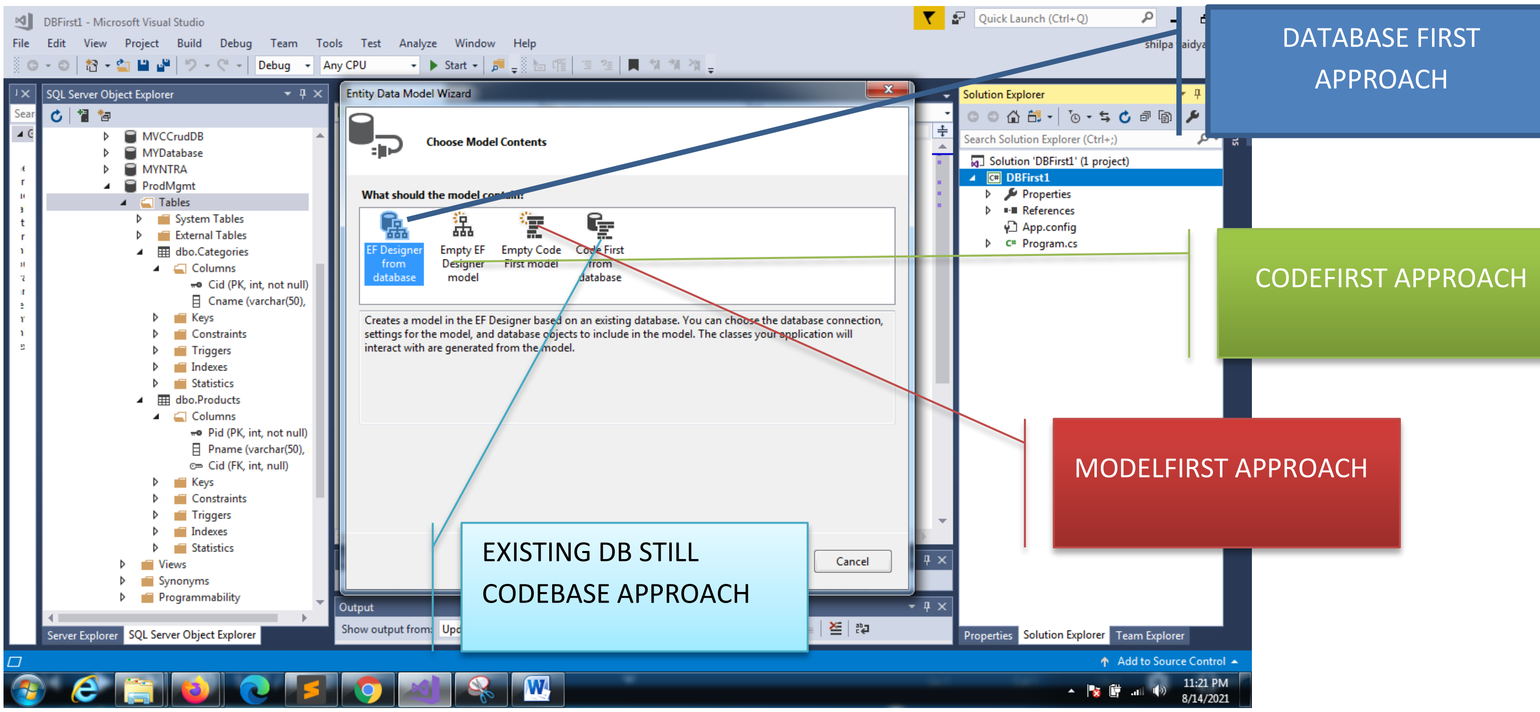
Class	Visual C# Items
Class for U-SQL	Visual C# Items
Interface	Visual C# Items
Windows Form	Visual C# Items
User Control	Visual C# Items
Component Class	Visual C# Items
User Control (WPF)	Visual C# Items
About Box	Visual C# Items
ADO.NET Entity Data Model	Visual C# Items
Application Configuration File	Visual C# Items
Application Manifest File	Visual C# Items
Assembly Information File	Visual C# Items

Type: Visual C# Items

A project item for creating an ADO.NET Entity Data Model.

Name: ProdModel

Add Cancel



Entity Data Model Wizard

Choose Model Contents

What should the model contain?

Creates a model in the EF Designer based on an existing database. You can choose the database connection, settings for the model, and database objects to include in the model. The classes your application will interact with are generated from the model.

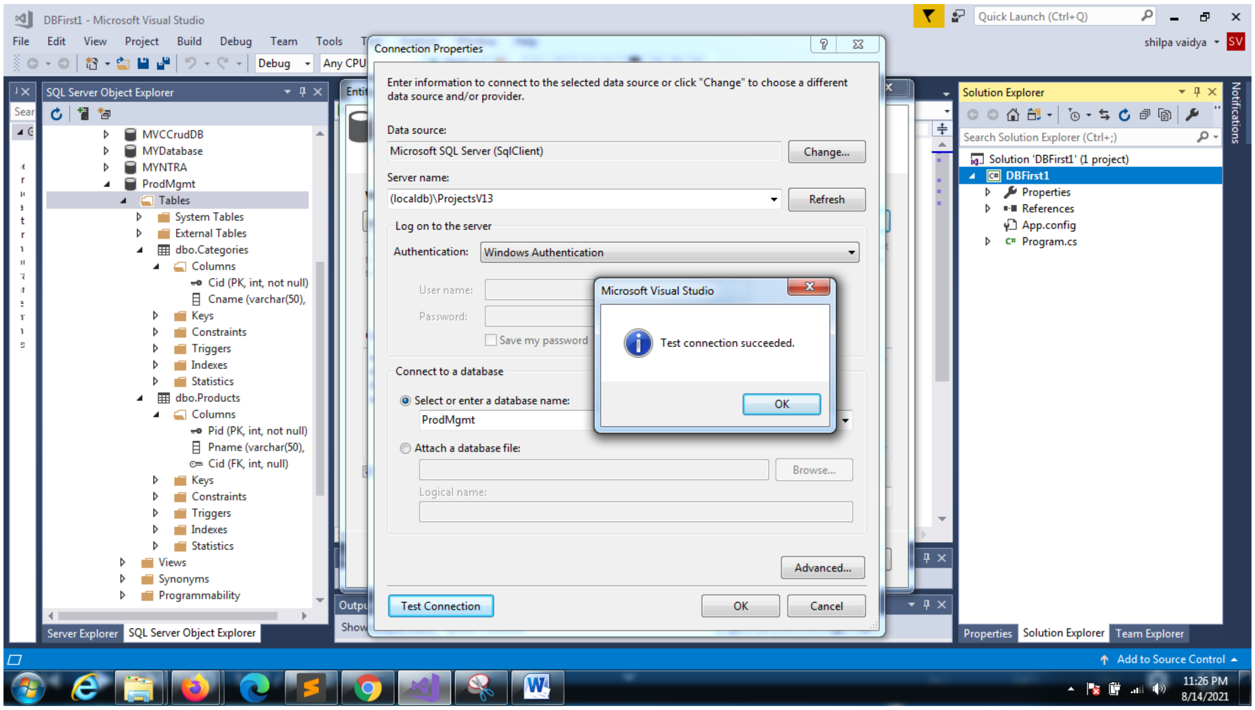
DATABASE FIRST APPROACH

CODEFIRST APPROACH

MODELFIRST APPROACH

EXISTING DB STILL CODEBASE APPROACH

NEW CONNECTION



Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:

Microsoft SQL Server (SqlClient) [Change...]

Server name:

(localdb)\ProjectsV13 [Refresh]

Log on to the server

Authentication: Windows Authentication

User name:

Password:

□ Save my password

Connect to a database

● Select or enter a database name:

ProdMgmt

○ Attach a database file:

[Browse...]

Logical name:

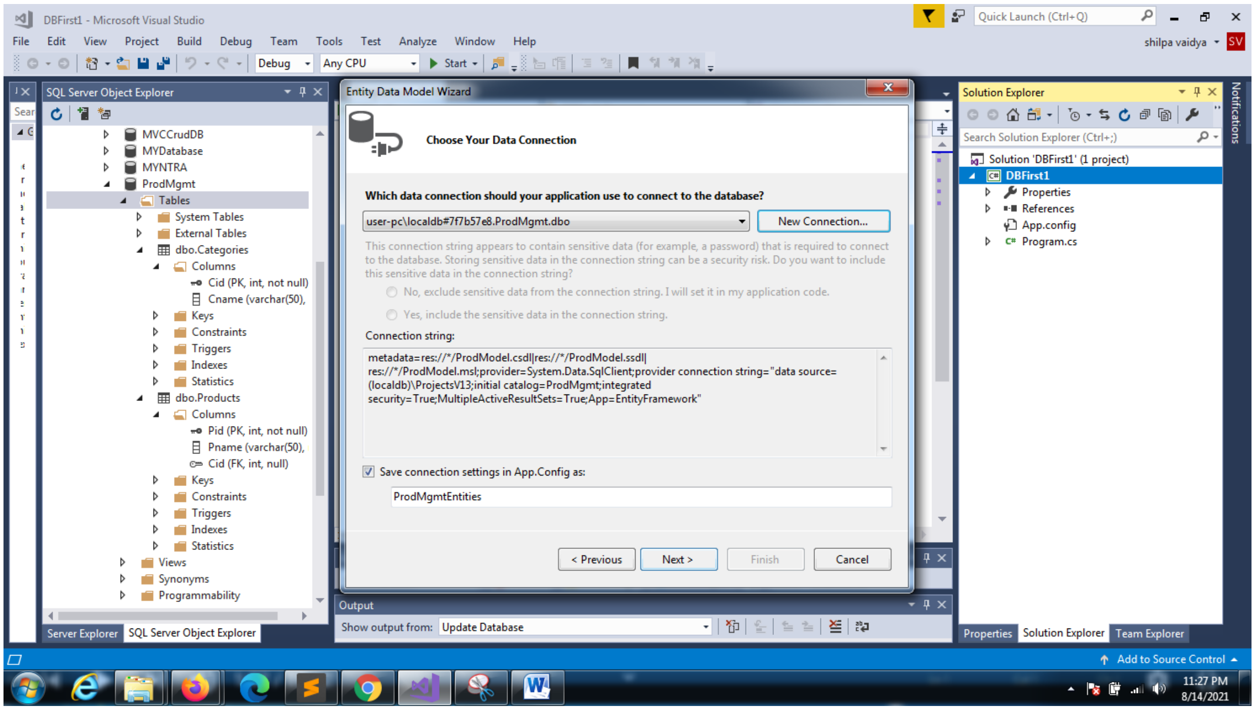
[Advanced...]

[Test Connection] [OK] [Cancel]

Microsoft Visual Studio

Test connection succeeded.

[OK]



Entity Data Model Wizard

Choose Your Data Connection

Which data connection should your application use to connect to the database?

user-pc\localdb#7f7b57e8.ProdMgmt.dbo [New Connection...]

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

○ No, exclude sensitive data from the connection string. I will set it in my application code.

○ Yes, include the sensitive data in the connection string.

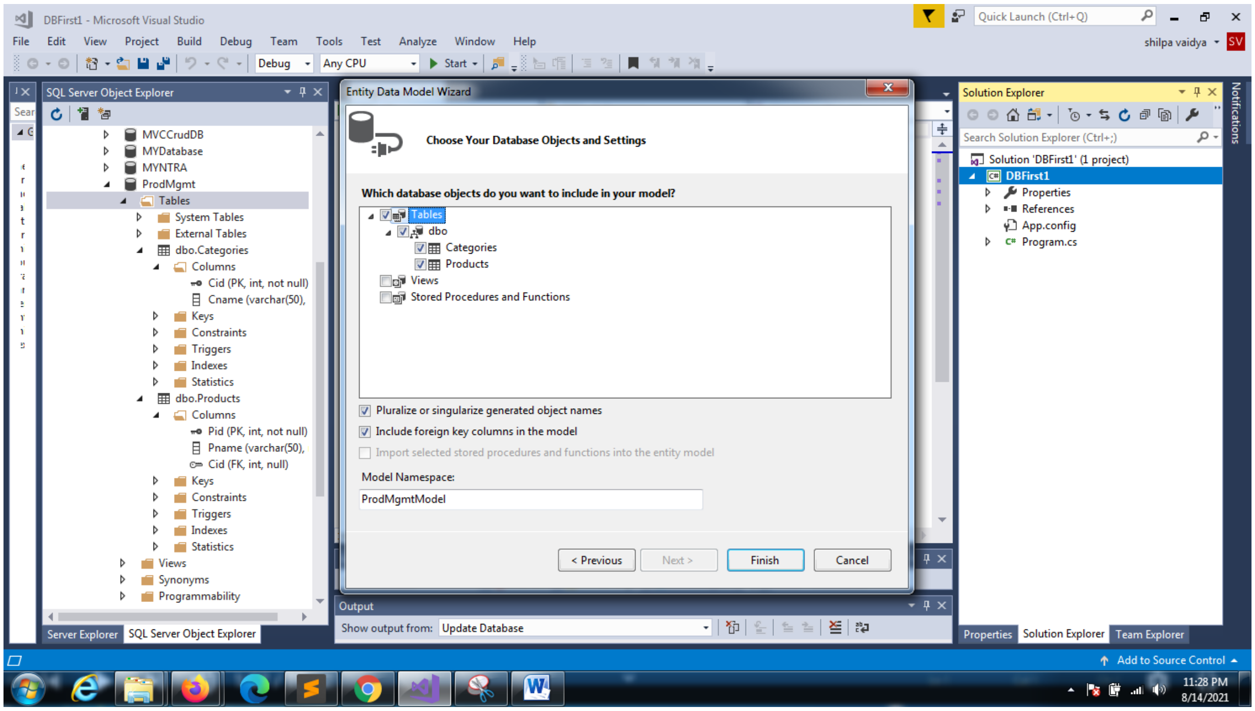
Connection string:

```
metadata=res://*/ProdModel.csdl|res://*/ProdModel.ssdl|
res://*/ProdModel.msl;provider=System.Data.SqlClient;provider connection string="data source=
(localdb)\ProjectsV13;initial catalog=ProdMgmt;integrated
security=True;MultipleActiveResultSets=True;App=EntityFramework"
```

☑ Save connection settings in App.Config as:

ProdMgmtEntities

[< Previous] [Next >] [Finish] [Cancel]



Entity Data Model Wizard

Choose Your Database Objects and Settings

Which database objects do you want to include in your model?

- ☑ Tables
  - ☑ dbo
    - ☑ Categories
    - ☑ Products
- □ Views
- □ Stored Procedures and Functions

☑ Pluralize or singularize generated object names

☑ Include foreign key columns in the model

□ Import selected stored procedures and functions into the entity model

Model Namespace:

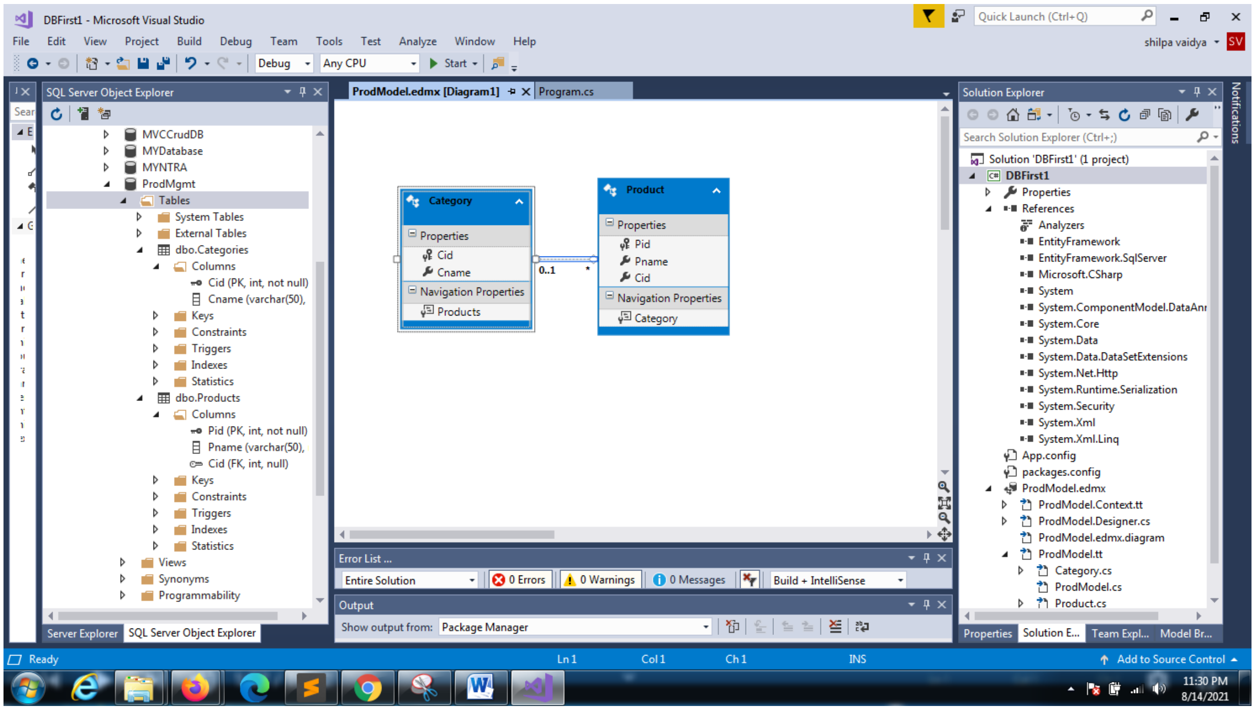
ProdMgmtModel

< Previous | Next > | Finish | Cancel

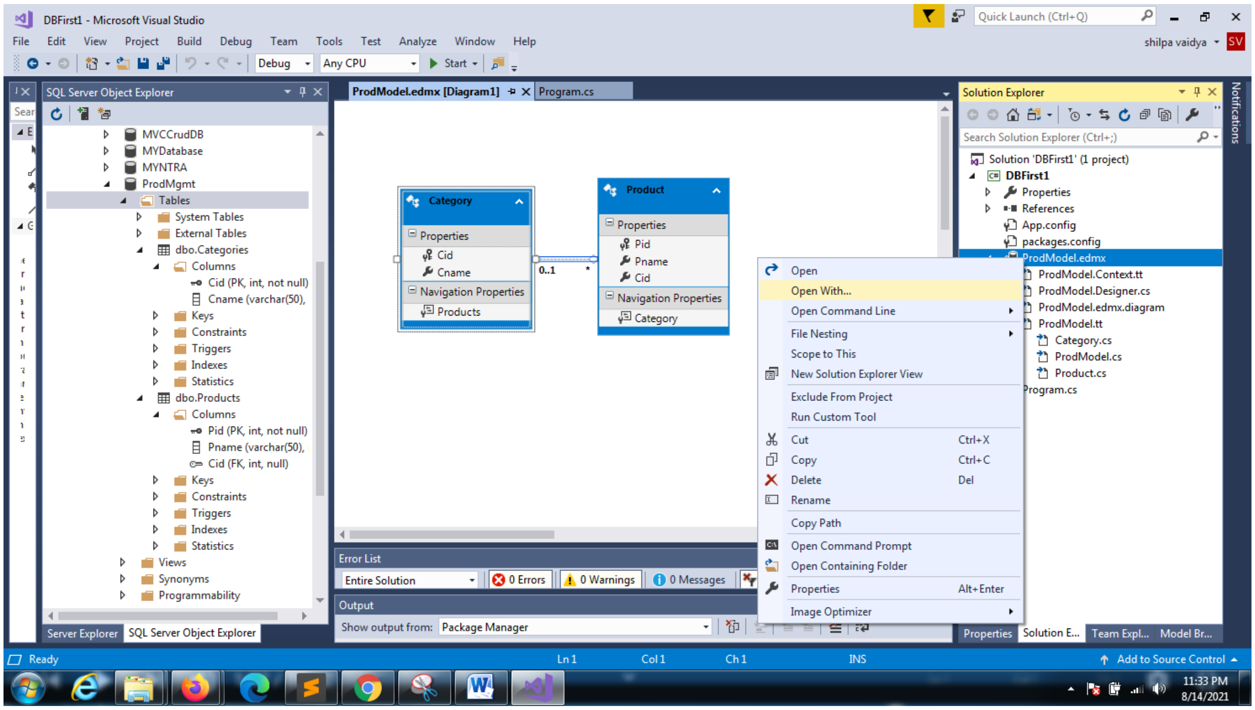
Output

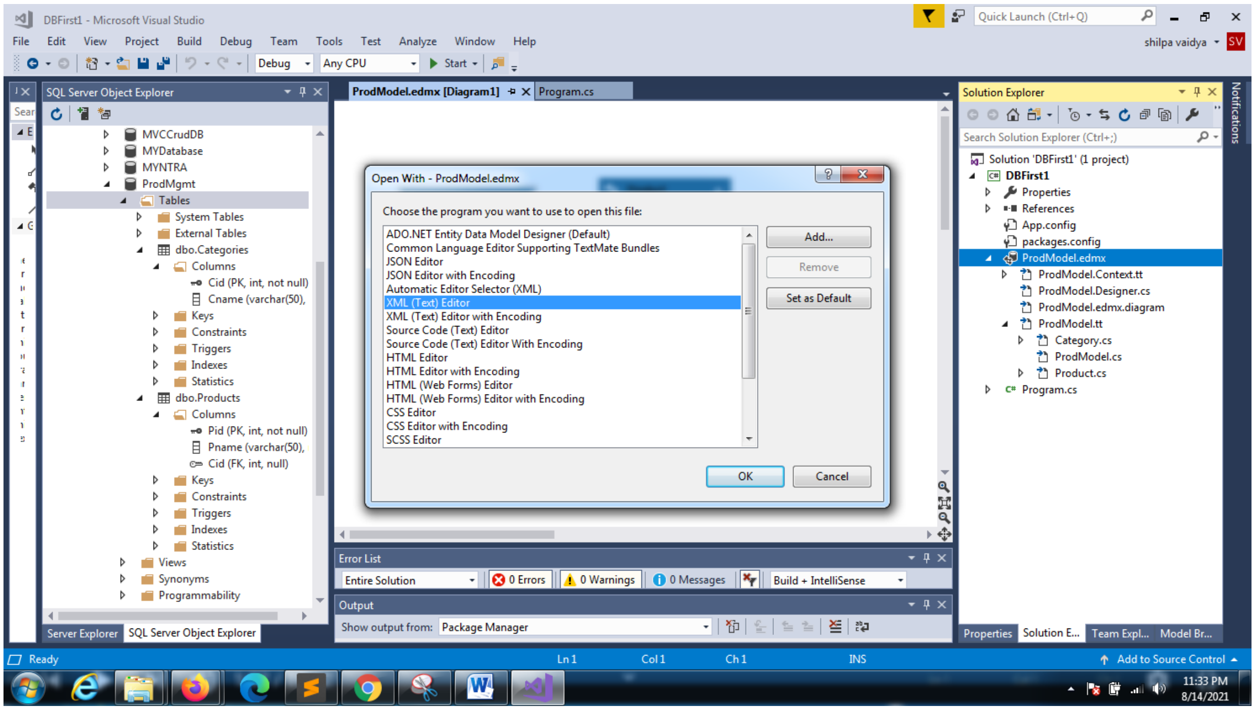
Show output from: Update Database

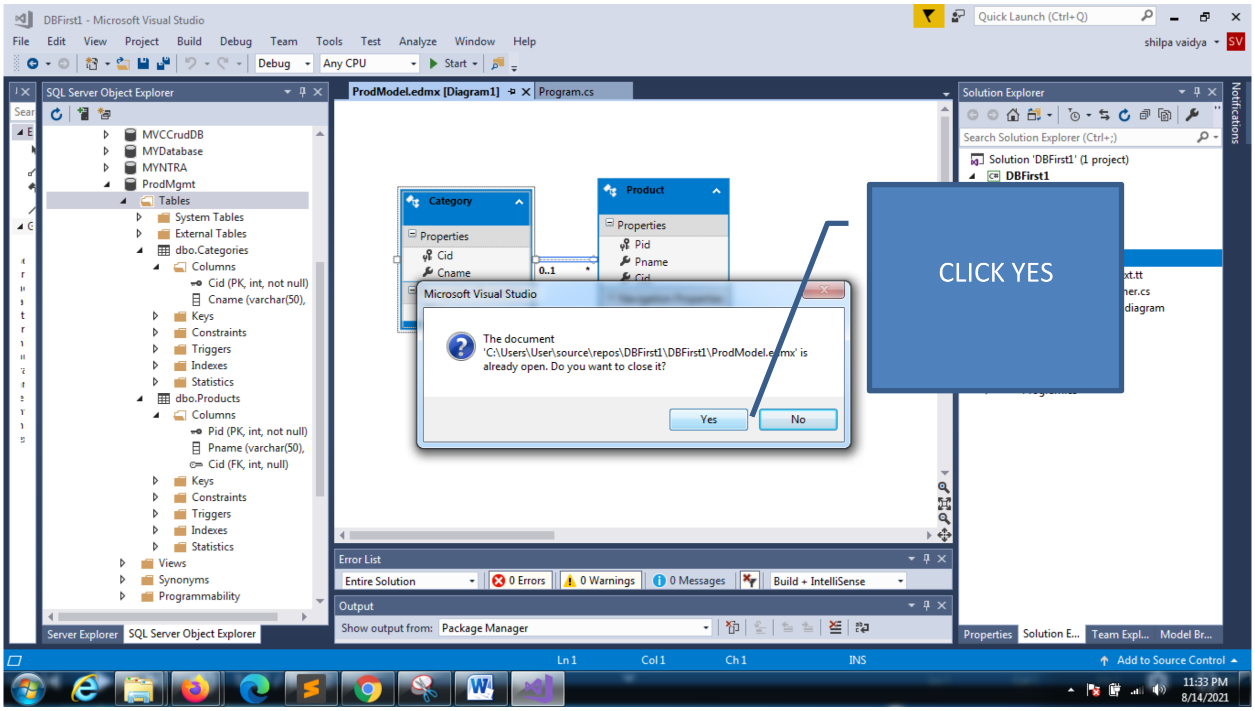
EDMX FILE WILL BE CREATED



IF YOU WANT TO SEE THE SAME IN XML FORMAT




Open With - ProdModel.edmx
Choose the program you want to use to open this file:
ADO.NET Entity Data Model Designer (Default)
Common Language Editor Supporting TextMate Bundles
JSON Editor
JSON Editor with Encoding
Automatic Editor Selector (XML)
XML (Text) Editor
XML (Text) Editor with Encoding
Source Code (Text) Editor
Source Code (Text) Editor With Encoding
HTML Editor
HTML Editor with Encoding
HTML (Web Forms) Editor
HTML (Web Forms) Editor with Encoding
CSS Editor
CSS Editor with Encoding
SCSS Editor
Add...
Remove
Set as Default
OK
Cancel



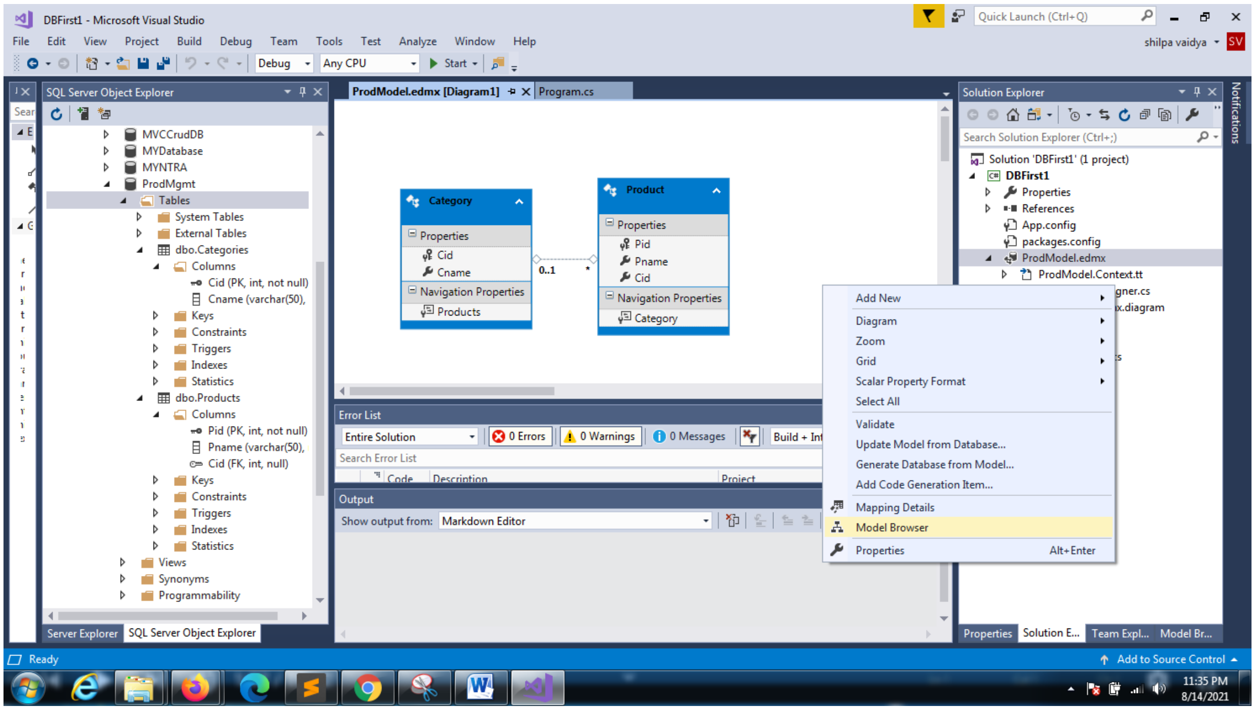
Microsoft Visual Studio

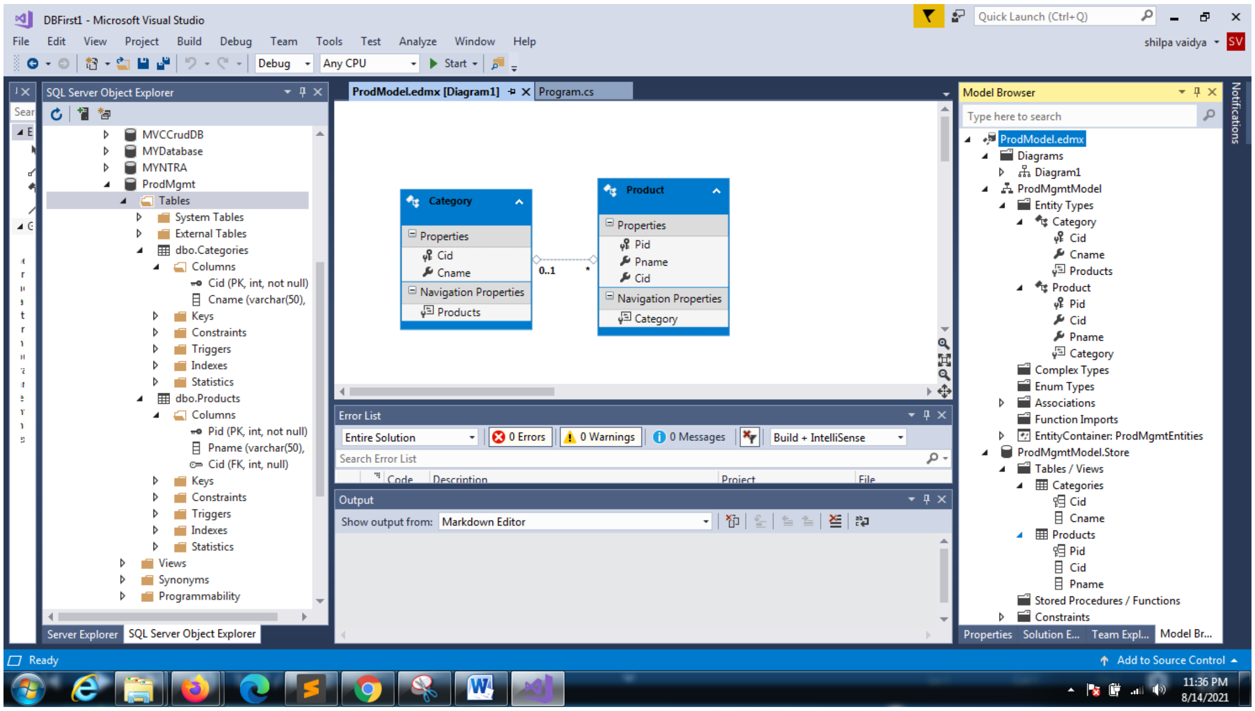
The document 'C:\Users\User\source\repos\DBFirst1\DBFirst1\ProdModel.edmx' is already open. Do you want to close it?

Yes No

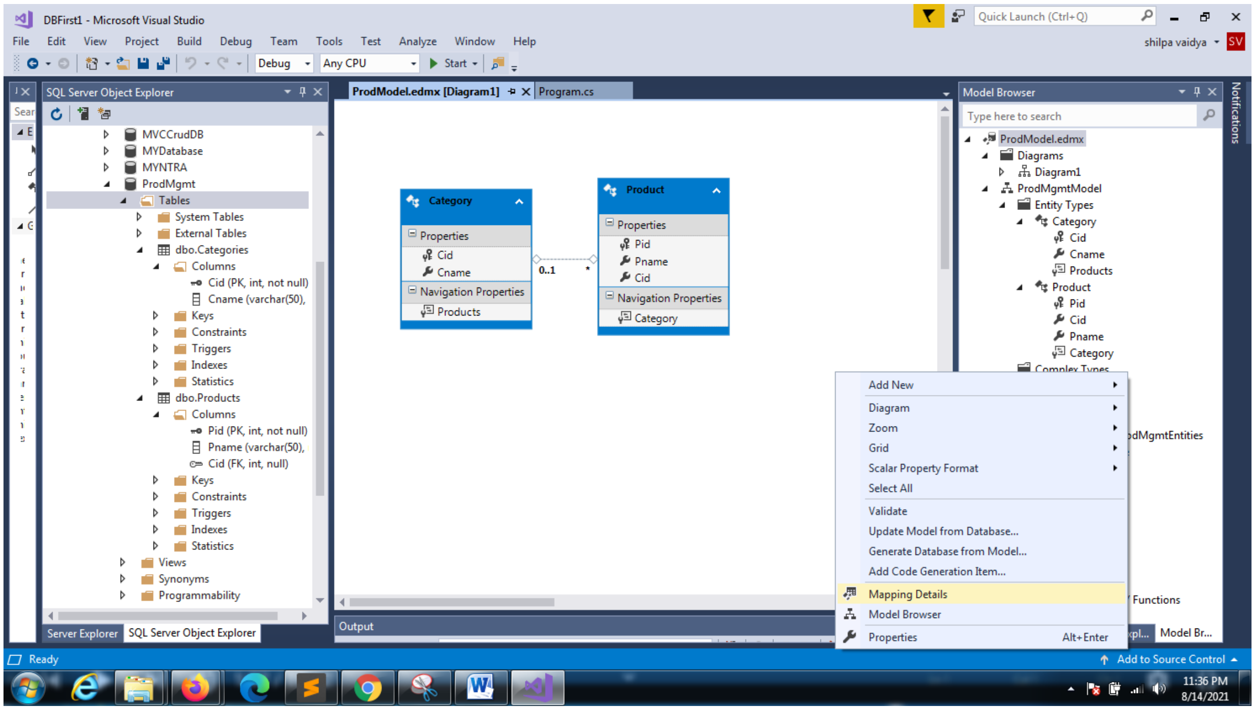
CLICK YES

MODEL BROWSER

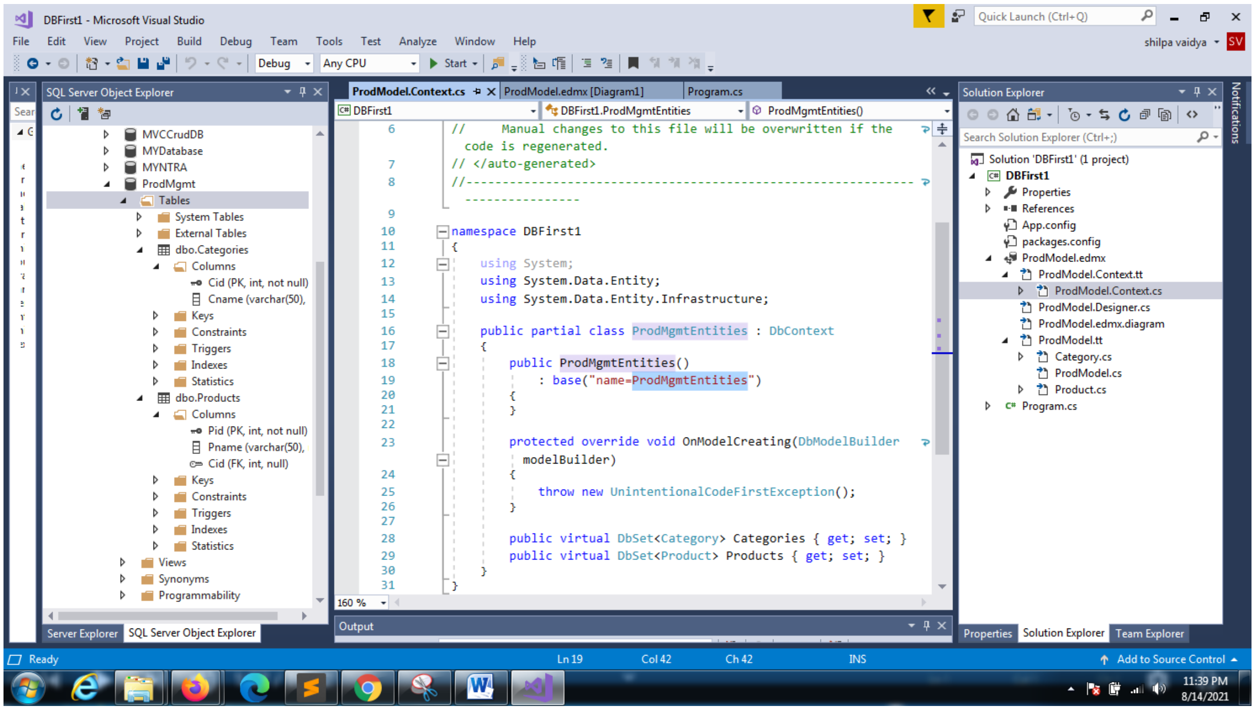




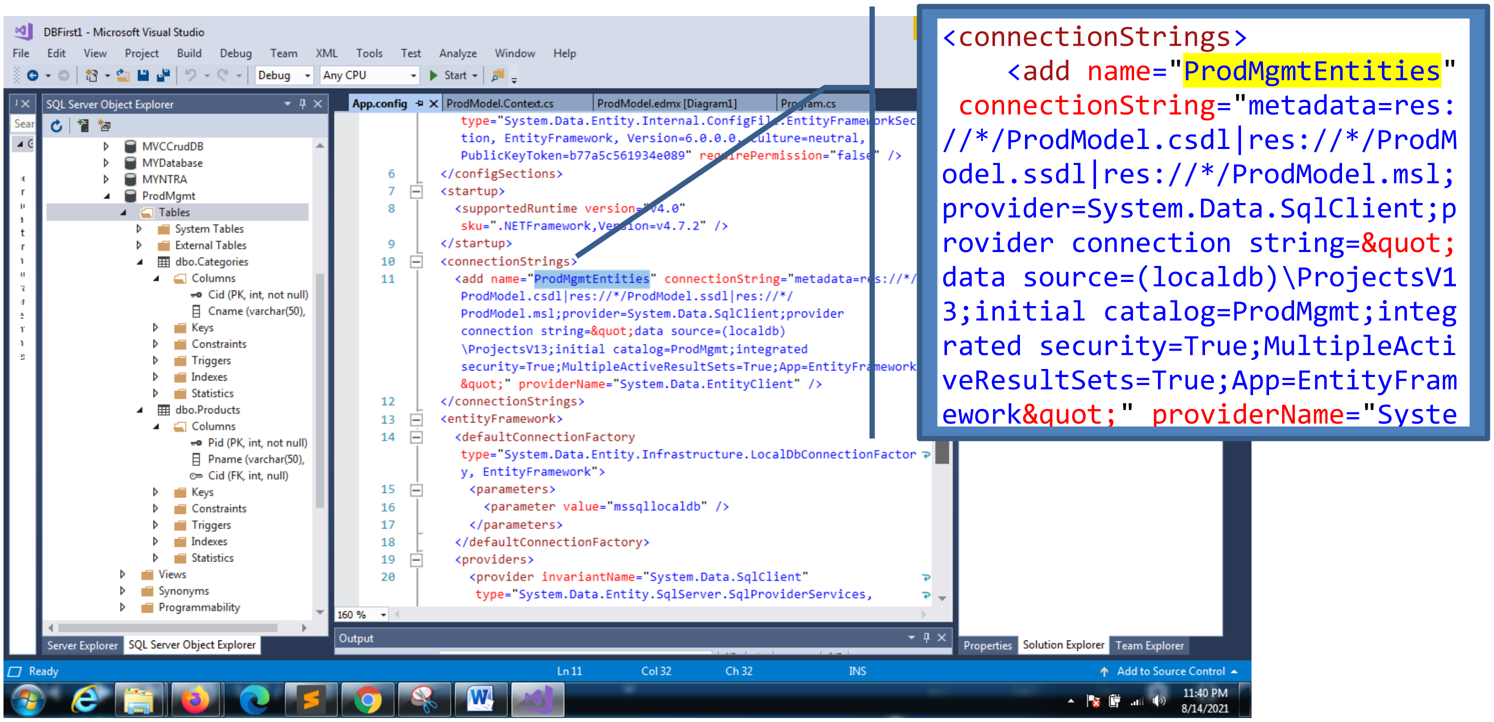
MAPPING DETAILS(CLICK ON THE ENTITY TO SEE MAPPING DETAILS)



DBCONTEXT CLASS THE NAME SPECIFIED IS IN APPCONFIG



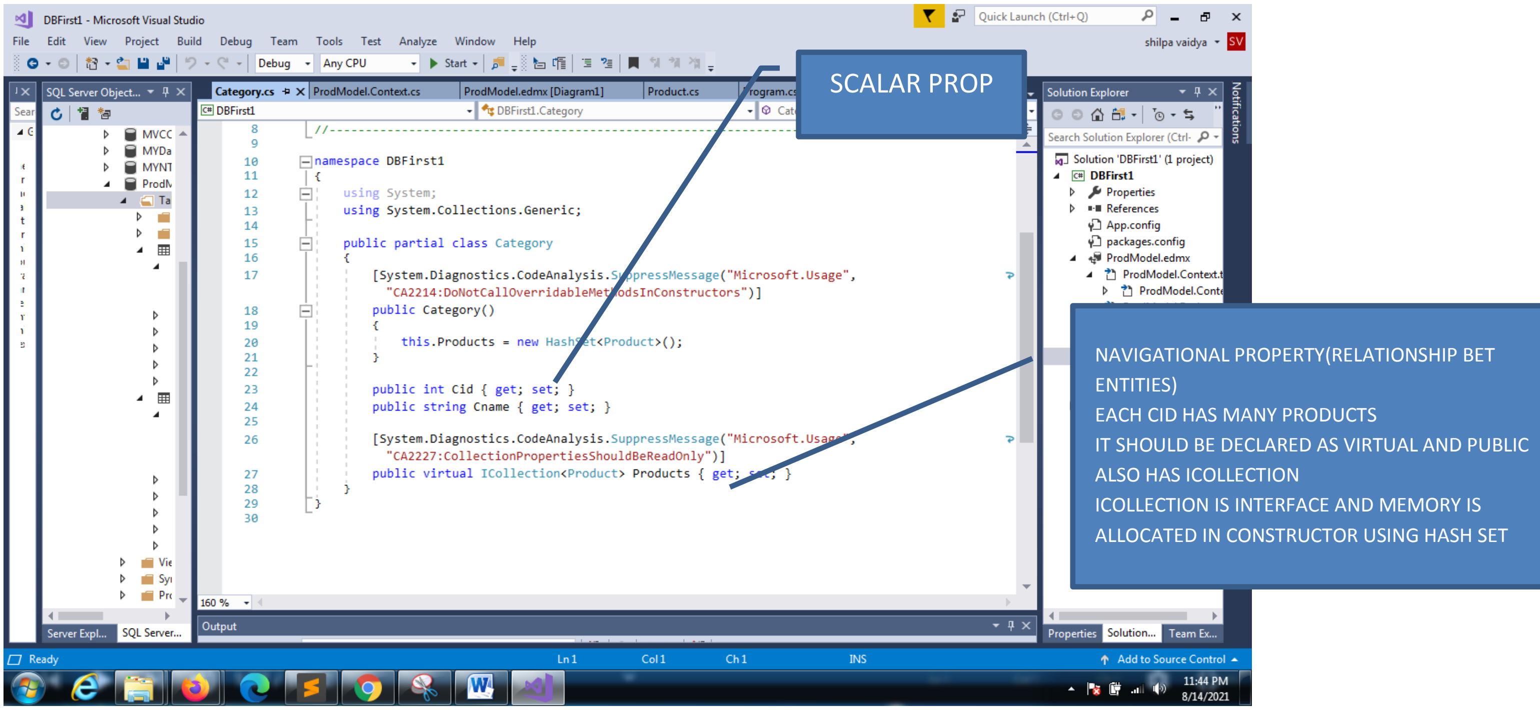
```
//     Manual changes to this file will be overwritten if the
  code is regenerated.
// </auto-generated>
//------------------------------------------------------------------
  ----------------

namespace DBFirst1
{
    using System;
    using System.Data.Entity;
    using System.Data.Entity.Infrastructure;

    public partial class ProdMgmtEntities : DbContext
    {
        public ProdMgmtEntities()
            : base("name=ProdMgmtEntities")
        {
        }

        protected override void OnModelCreating(DbModelBuilder
         modelBuilder)
        {
            throw new UnintentionalCodeFirstException();
        }

        public virtual DbSet<Category> Categories { get; set; }
        public virtual DbSet<Product> Products { get; set; }
    }
}
```



```
     type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSec
     tion, EntityFramework, Version=6.0.0.0, Culture=neutral,
     PublicKeyToken=b77a5c561934e089" requirePermission="false" />
  </configSections>
  <startup>
    <supportedRuntime version="v4.0"
     sku=".NETFramework,Version=v4.7.2" />
  </startup>
  <connectionStrings>
    <add name="ProdMgmtEntities" connectionString="metadata=res://*/
     ProdModel.csdl|res://*/ProdModel.ssdl|res://*/
     ProdModel.msl;provider=System.Data.SqlClient;provider
     connection string=&quot;data source=(localdb)
     \ProjectsV13;initial catalog=ProdMgmt;integrated
     security=True;MultipleActiveResultSets=True;App=EntityFramework
     &quot;" providerName="System.Data.EntityClient" />
  </connectionStrings>
  <entityFramework>
    <defaultConnectionFactory
     type="System.Data.Entity.Infrastructure.LocalDbConnectionFactor
     y, EntityFramework">
      <parameters>
        <parameter value="mssqllocaldb" />
      </parameters>
    </defaultConnectionFactory>
    <providers>
      <provider invariantName="System.Data.SqlClient"
       type="System.Data.Entity.SqlServer.SqlProviderServices,
```

```
<connectionStrings>
    <add name="ProdMgmtEntities"
 connectionString="metadata=res:
//*/ProdModel.csdl|res://*/ProdM
odel.ssdl|res://*/ProdModel.msl;
provider=System.Data.SqlClient;p
rovider connection string=&quot;
data source=(localdb)\ProjectsV1
3;initial catalog=ProdMgmt;integ
rated security=True;MultipleActi
veResultSets=True;App=EntityFram
ework&quot;"  providerName="Syste
```

CATEGORY.CS

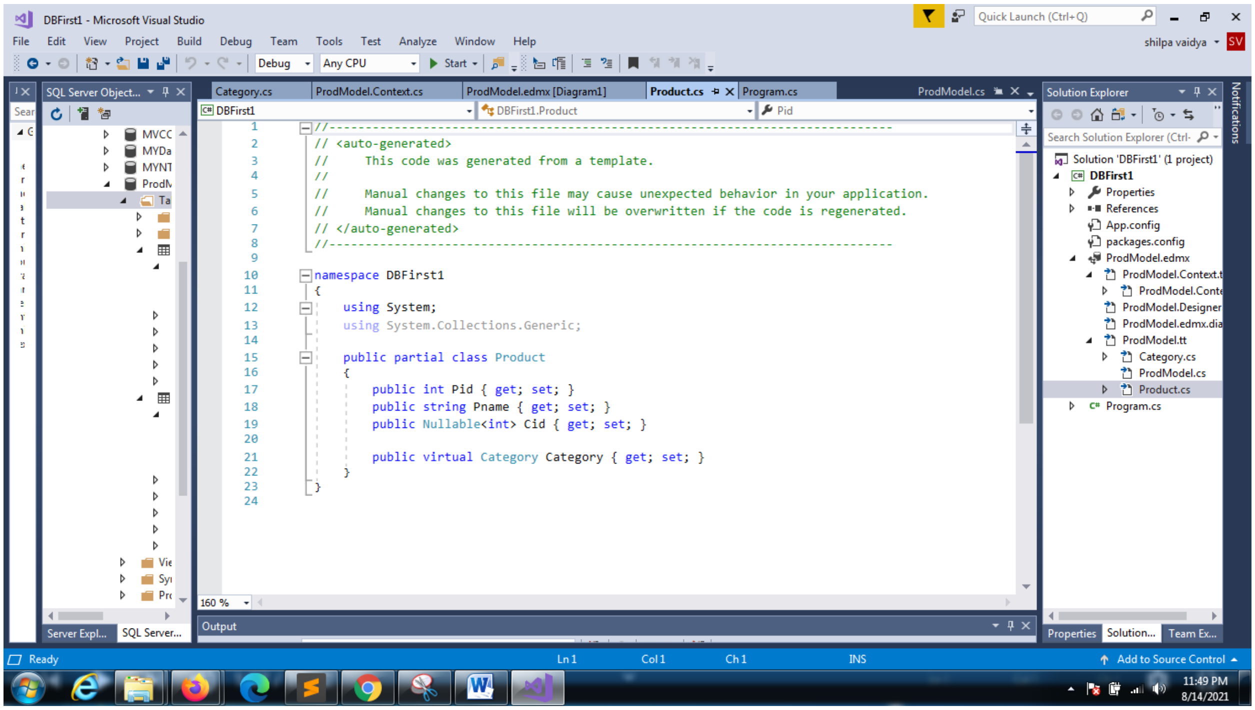


SCALAR PROP

NAVIGATIONAL PROPERTY(RELATIONSHIP BET ENTITIES)
EACH CID HAS MANY PRODUCTS
IT SHOULD BE DECLARED AS VIRTUAL AND PUBLIC
ALSO HAS ICOLLECTION
ICOLLECTION IS INTERFACE AND MEMORY IS ALLOCATED IN CONSTRUCTOR USING HASH SET

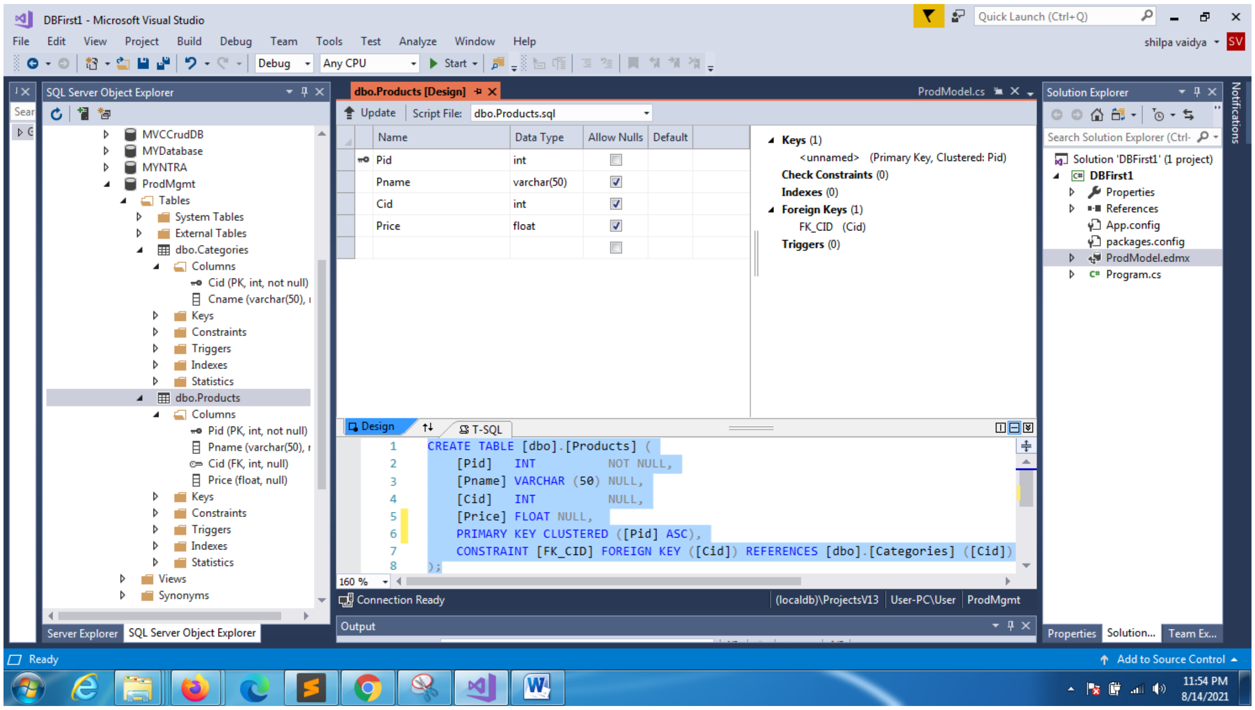
```
//------------------------------------------------------------------------------

namespace DBFirst1
{
    using System;
    using System.Collections.Generic;

    public partial class Category
    {
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",
          "CA2214:DoNotCallOverridableMethodsInConstructors")]
        public Category()
        {
            this.Products = new HashSet<Product>();
        }

        public int Cid { get; set; }
        public string Cname { get; set; }

        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",
          "CA2227:CollectionPropertiesShouldBeReadOnly")]
        public virtual ICollection<Product> Products { get; set; }
    }
}
```

PRODUCTS.CS



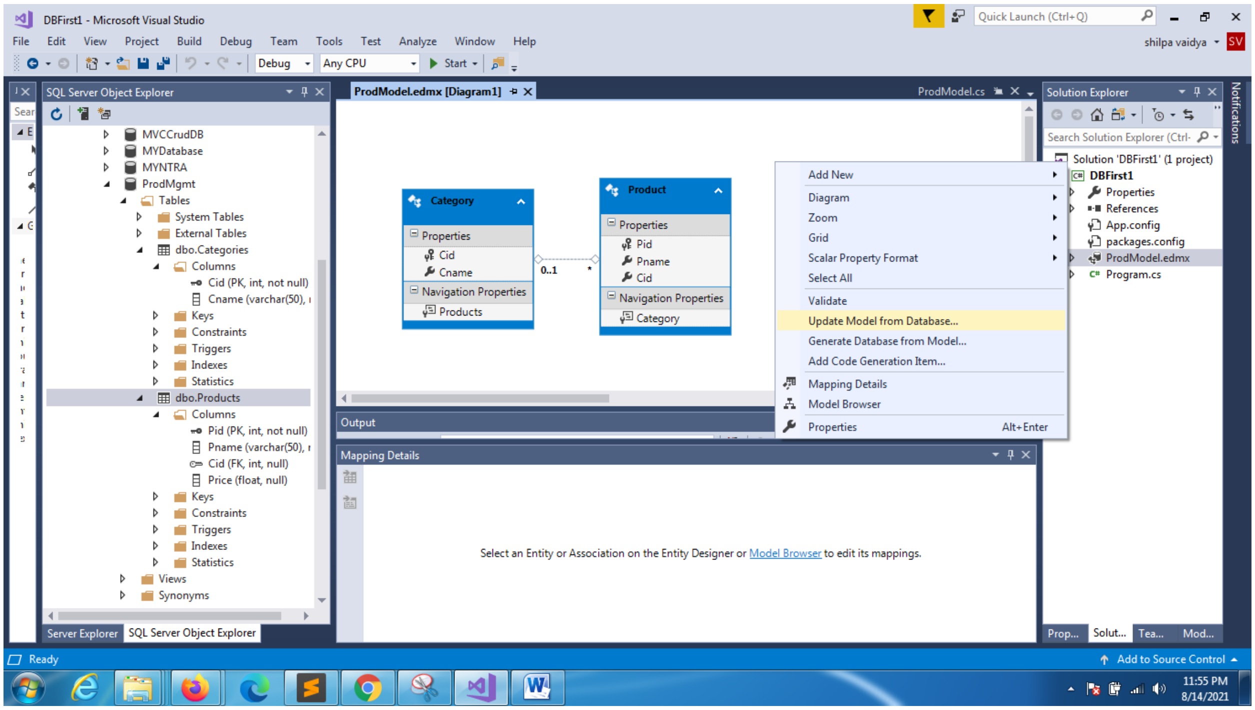
```
//------------------------------------------------------------------------------
// <auto-generated>
//     This code was generated from a template.
//
//     Manual changes to this file may cause unexpected behavior in your application.
//     Manual changes to this file will be overwritten if the code is regenerated.
// </auto-generated>
//------------------------------------------------------------------------------

namespace DBFirst1
{
    using System;
    using System.Collections.Generic;

    public partial class Product
    {
        public int Pid { get; set; }
        public string Pname { get; set; }
        public Nullable<int> Cid { get; set; }

        public virtual Category Category { get; set; }
    }
}
```

IF YOU ADD ANY COLUMN IN DB EX YOU ADDED PRICE IN PRODUCTS

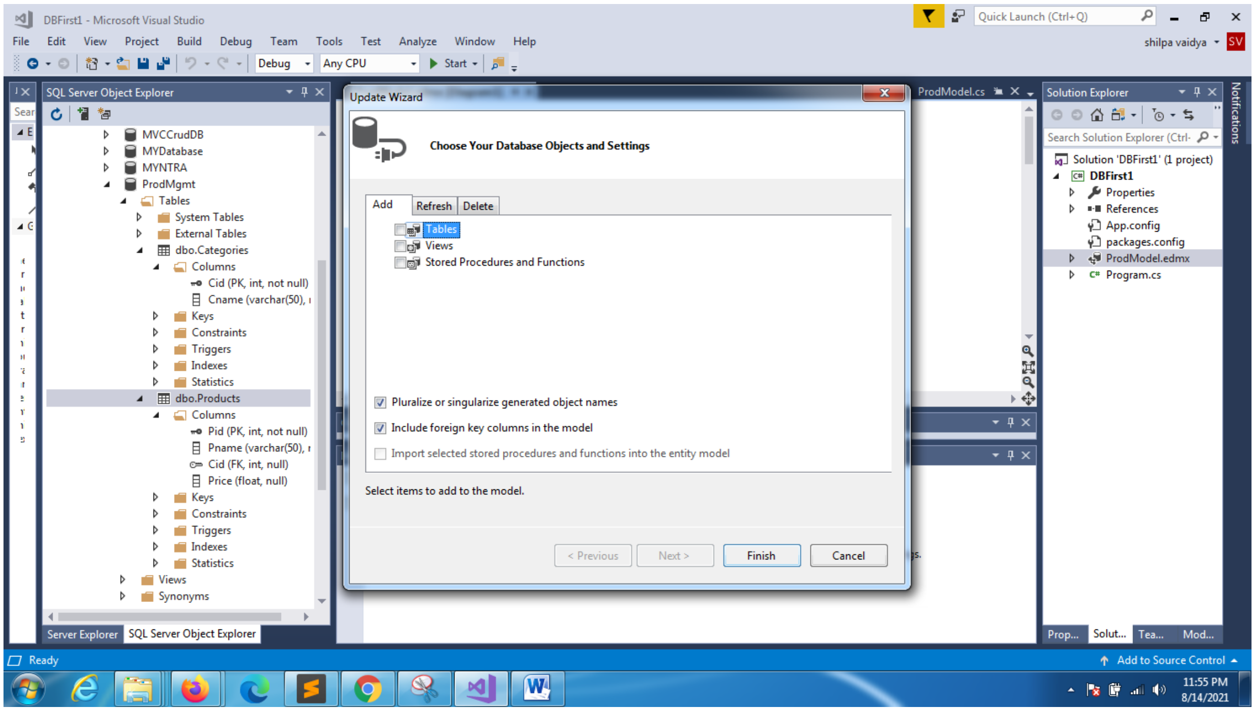


Name	Data Type	Allow Nulls	Default
Pid	int	☐	
Pname	varchar(50)	☑	
Cid	int	☑	
Price	float	☑	
		☐	

```
CREATE TABLE [dbo].[Products] (
    [Pid]   INT          NOT NULL,
    [Pname] VARCHAR (50) NULL,
    [Cid]   INT          NULL,
    [Price] FLOAT NULL,
    PRIMARY KEY CLUSTERED ([Pid] ASC),
    CONSTRAINT [FK_CID] FOREIGN KEY ([Cid]) REFERENCES [dbo].[Categories] ([Cid])
);
```



Select an Entity or Association on the Entity Designer or Model Browser to edit its mappings.



Update Wizard

Choose Your Database Objects and Settings

Add | Refresh | Delete

□ Tables
□ Views
□ Stored Procedures and Functions

☑ Pluralize or singularize generated object names

☑ Include foreign key columns in the model

□ Import selected stored procedures and functions into the entity model

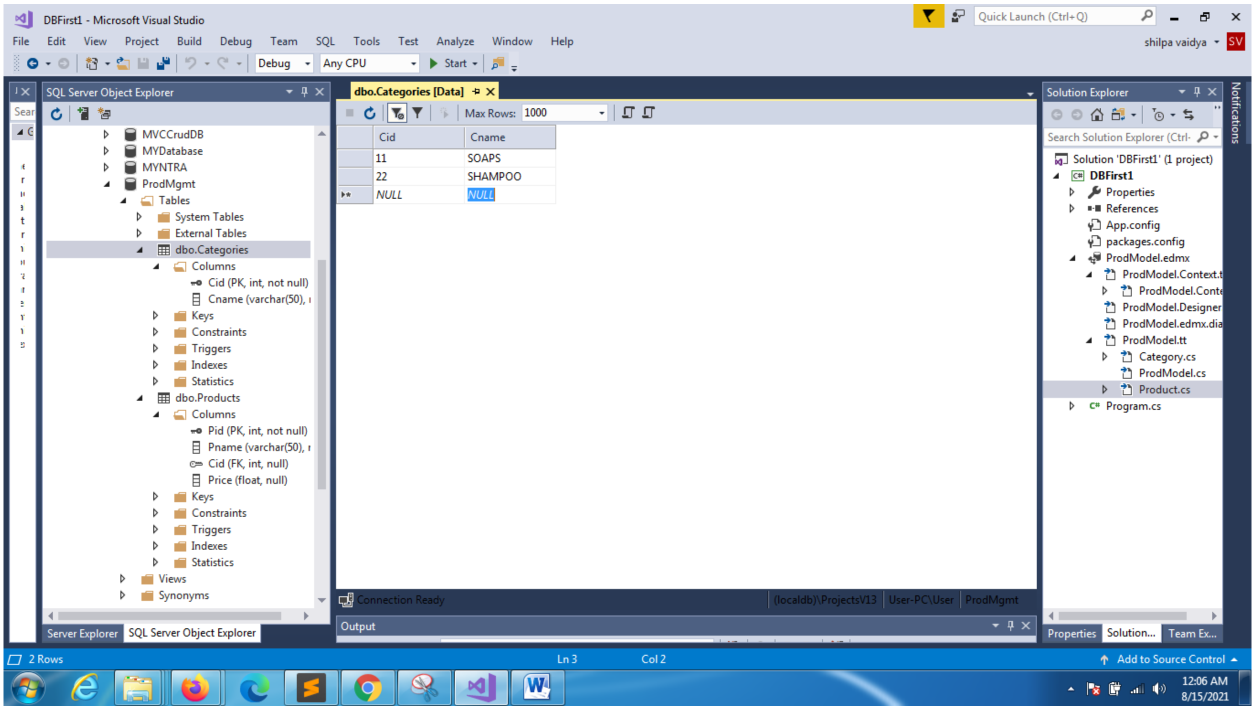
Select items to add to the model.

< Previous | Next > | Finish | Cancel

After this save your edmx file only then the changes will be reflected in entity classes.

We will be performing the crud operations here

Will first need to populate the Categories as Products has the CID as the foreign key and its dependent on the Categories.



Cid	Cname
11	SOAPS
22	SHAMPOO
NULL	NULL

WILL KEEP THE STRUCTURES READY IN THE START UP PROGRAM WITH OPERATIONS ON

INSERT

RETRIEVE

UPDATE

DELETE

```
using System;

namespace DBFirst1
{
    class Program
    {

        public static void display(Product p)
        {
            Console.WriteLine("{0}\t{1}\t{2}\t{3}\t{4}",p.Pid,p.Pname,p.Cid,p.Price,p.Category.Cname);
        }
        public static Product getProddata()
        {
            Product prod = new Product();
            prod.Pid = Convert.ToInt32(Console.ReadLine());
            prod.Pname = Console.ReadLine();
            prod.Cid = Convert.ToInt32(Console.ReadLine());
            prod.Price = float.Parse(Console.ReadLine());
            return prod;
        }
        static void Main(string[] args)
        {
            //CREATE OBJ OF CONTEXT CLASS
            ProdMgmtEntities db = new ProdMgmtEntities();

            int choice,id;
            do
            {
                Console.WriteLine("1 INSERT 2 DISPLAY 3 UPDATE 4 DELETE 5 EXIT");
                choice = Convert.ToInt32(Console.ReadLine());
                Product prod = new Product();
                switch (choice)
                {
                    case 1:
                        Console.WriteLine("Enter PID,PNAME.CID,PRICE");
                        prod=getProddata();
                        db.Products.Add(prod);
                        db.SaveChanges();
                        break;
                    case 2:
                        Console.WriteLine("PID\tPNAME\tCID\tPRICE\tCATEGORYNAME\t");
                        Console.WriteLine("----------------------------------------------");
                        foreach (Product p in db.Products)
                        {
                            display(p);
                        }
                        break;
                    case 3:
                        Console.WriteLine("UPDATE RECORD");
                        Console.WriteLine("Enter the PID");
                        id = Convert.ToInt32(Console.ReadLine());
                        prod = db.Products.Find(id);
                        if (prod == null)
                            Console.WriteLine("ID NOT FOUND");
                        else
                        {
                            Console.WriteLine("ENTER PNAME\tCID\tPRICE\t");
                            prod.Pname = Console.ReadLine();
                            prod.Cid = Convert.ToInt32(Console.ReadLine());
                            prod.Price = float.Parse(Console.ReadLine());
                            db.SaveChanges();
                        }
                        break;
                    case 4:
                        Console.WriteLine("UPDATE RECORD");
                        Console.WriteLine("Enter the PID");
                        id = Convert.ToInt32(Console.ReadLine());
                        prod = db.Products.Find(id);
                        if (prod == null)
                            Console.WriteLine("ID NOT FOUND");
                        else
                        {
                            db.Products.Remove(prod);
                            db.SaveChanges();
                        }
                        break;
                    case 5:break;
                    default: Console.WriteLine("Invalid choicce");break;
                }
            } while (choice!=5);
        }
    }
}
```