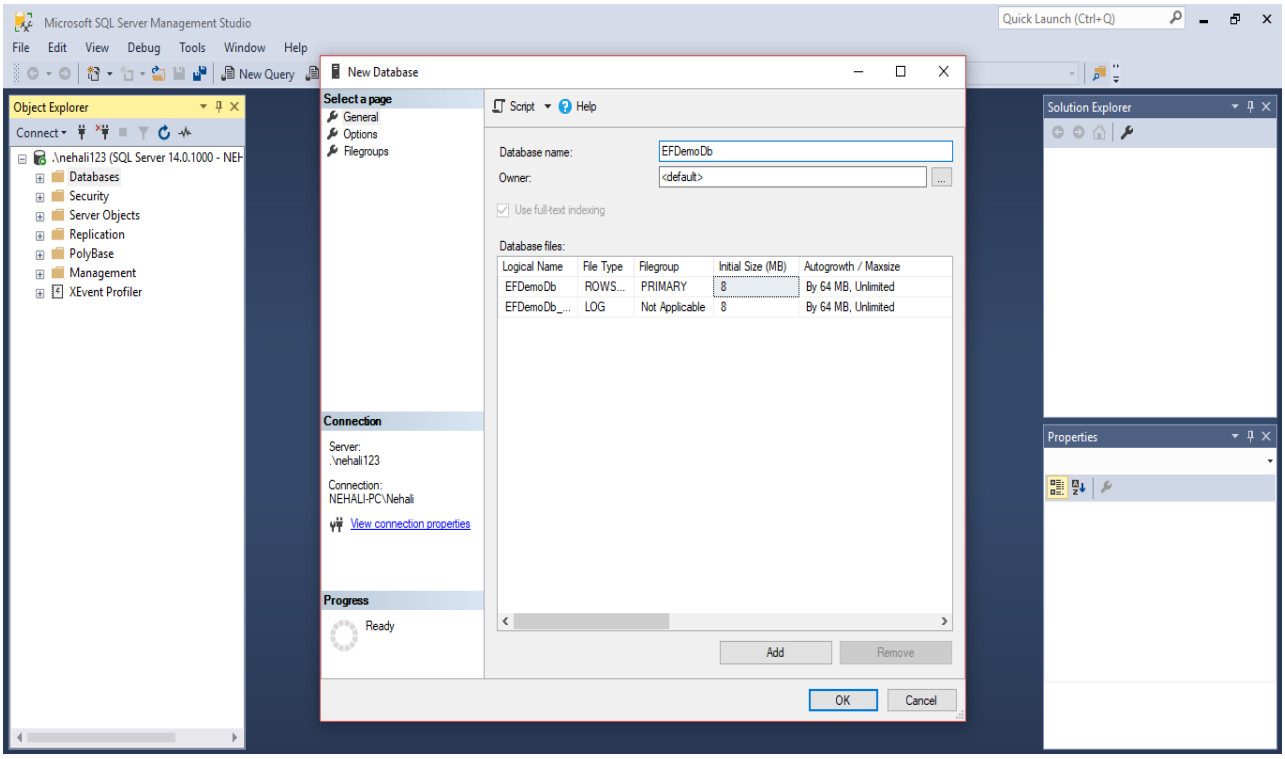
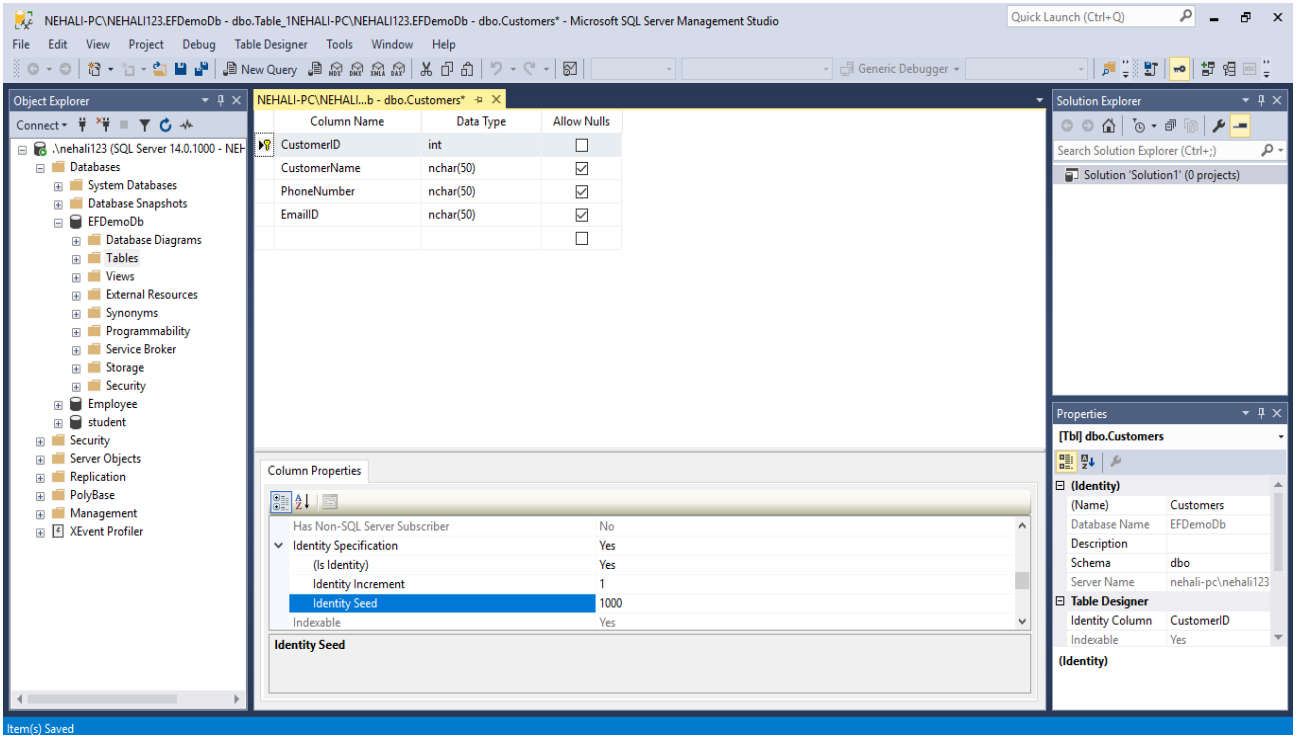
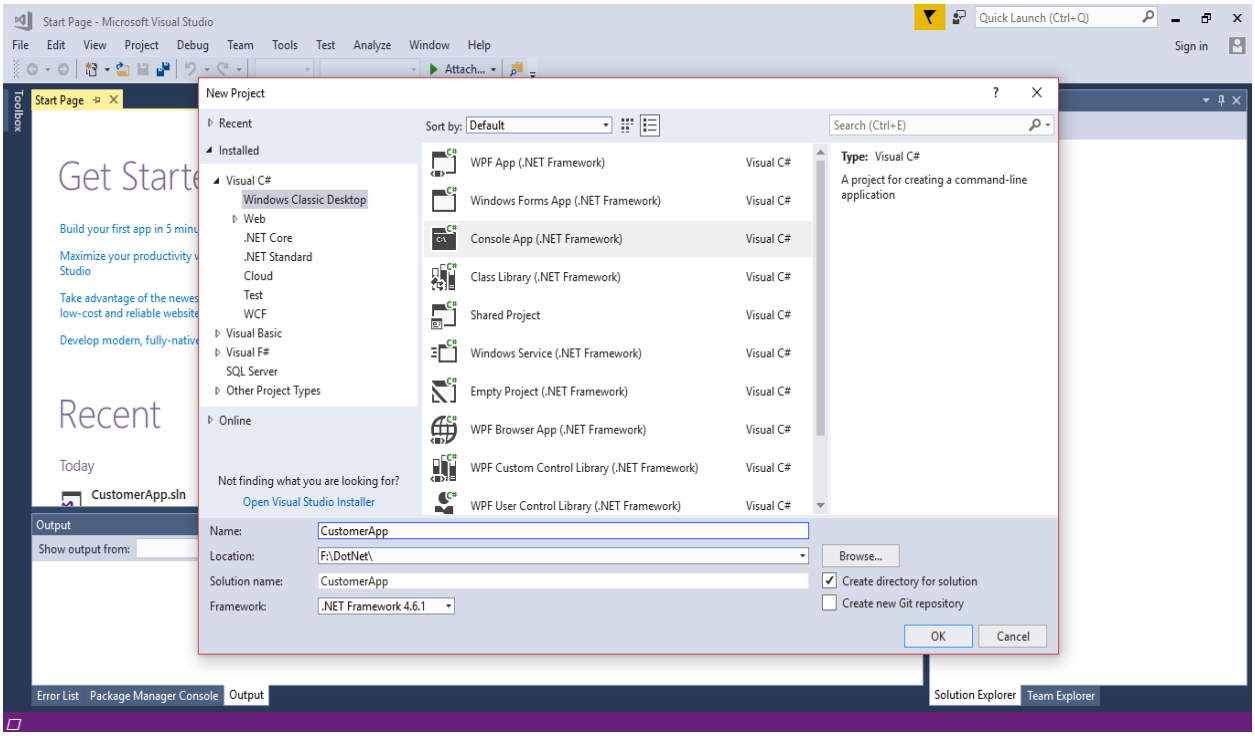
Entity Framework with Database

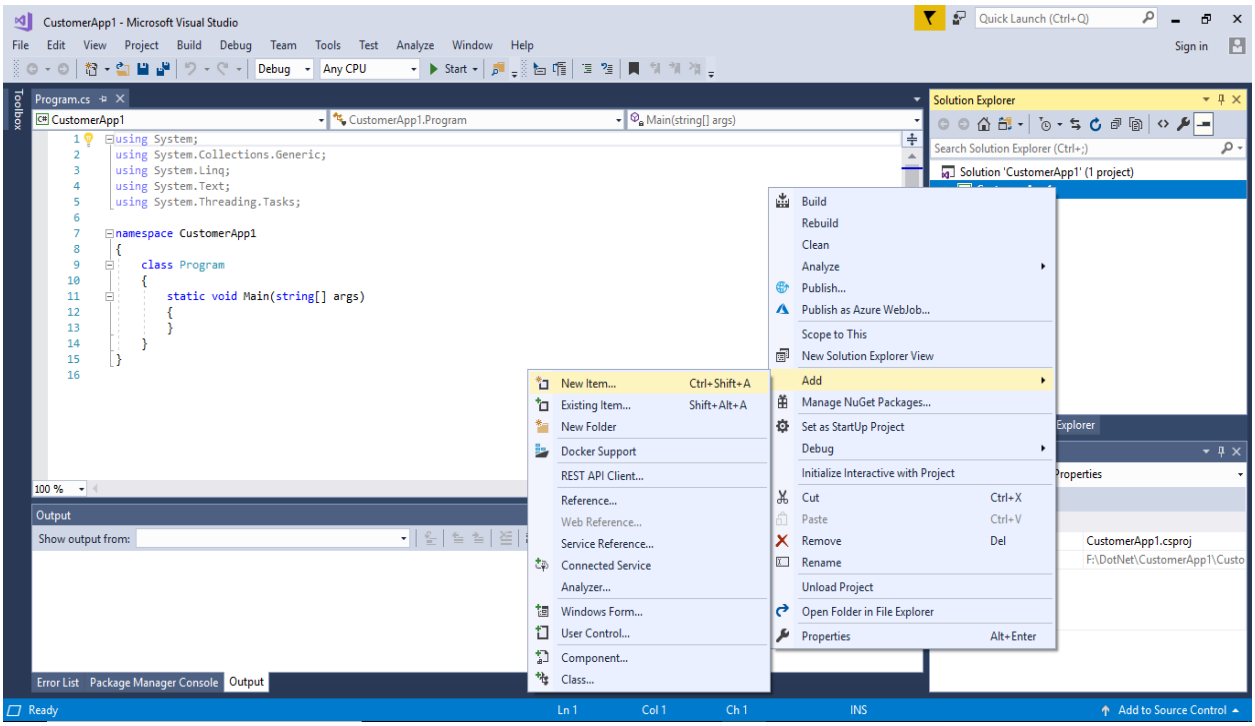
Step 1: Creation of Database and table

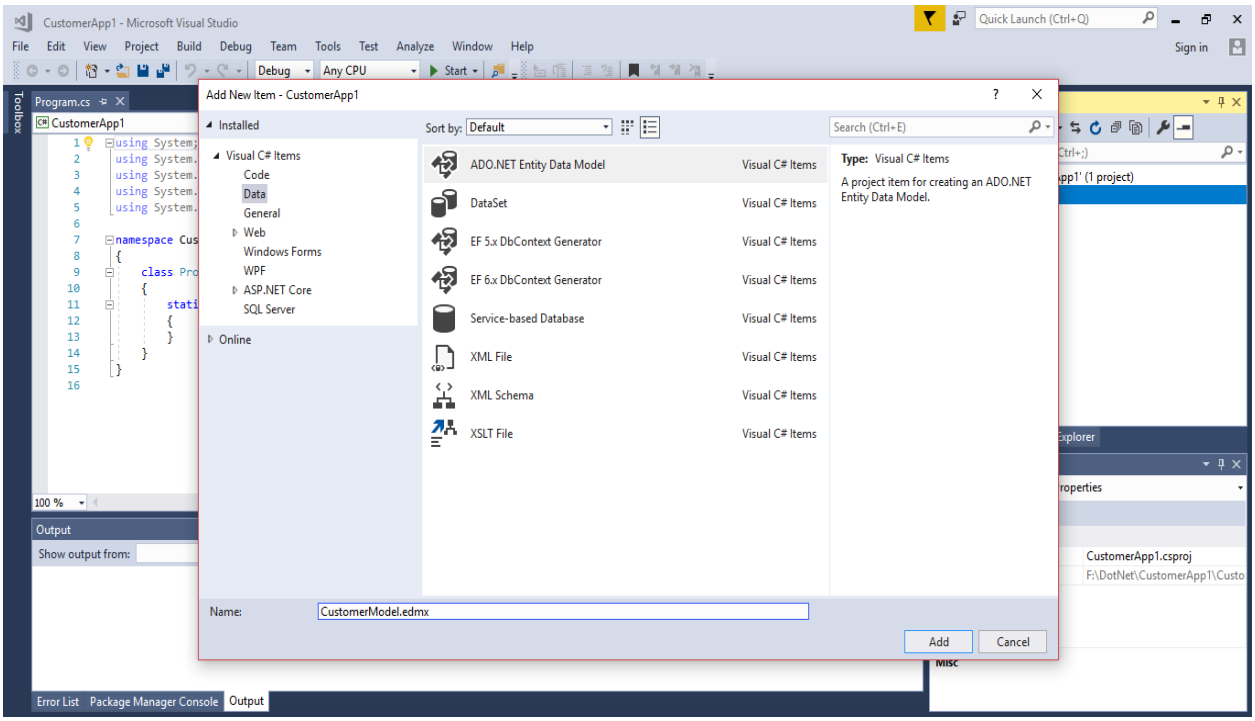


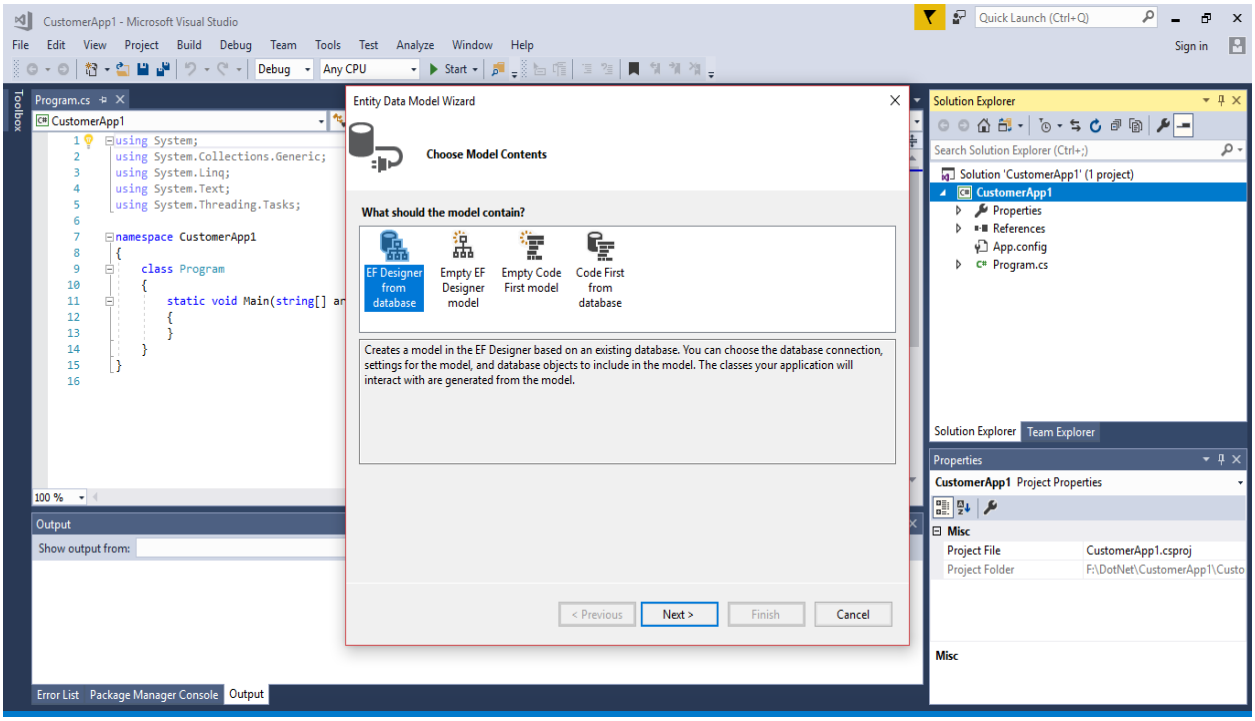


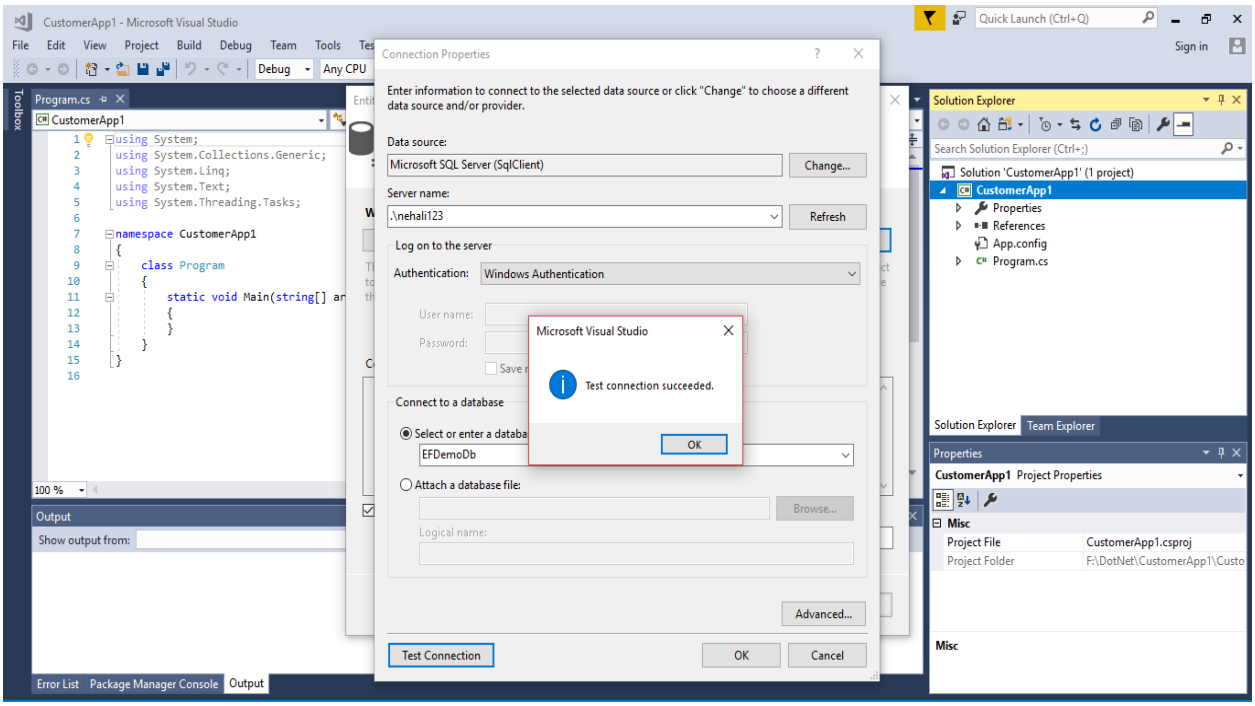
Step 2: Database Connection through Console Application.

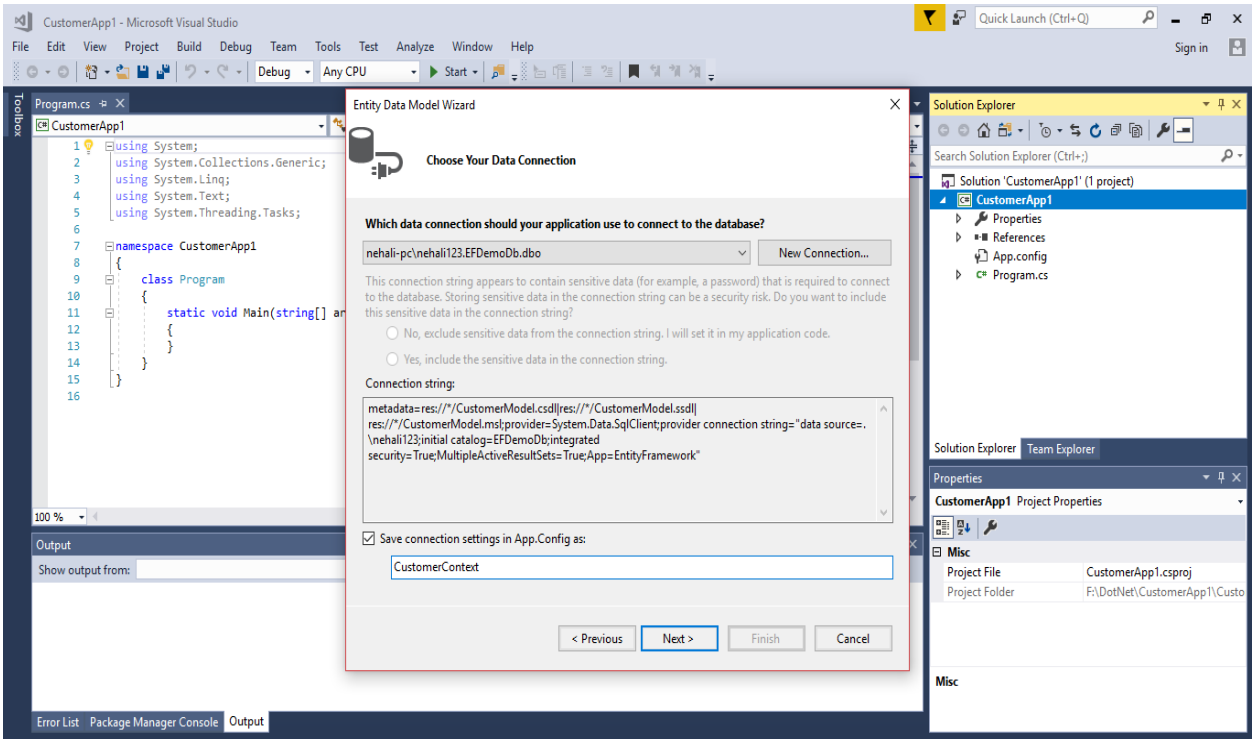


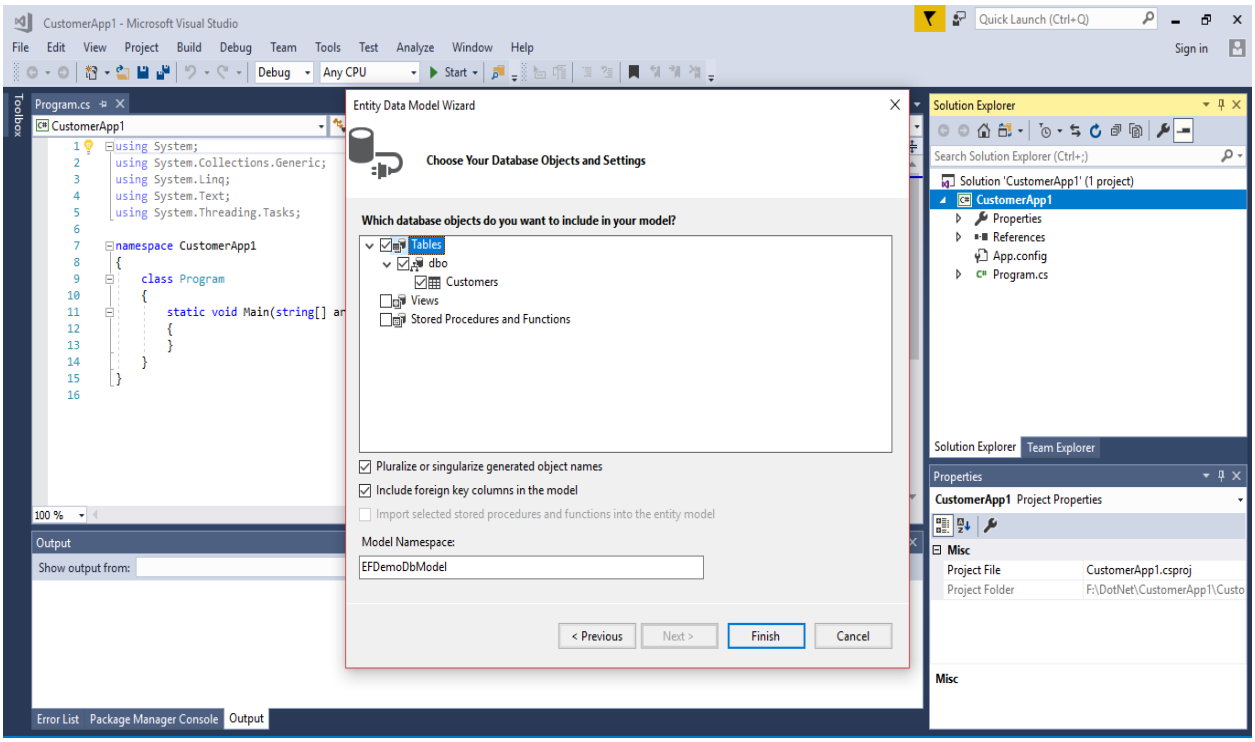


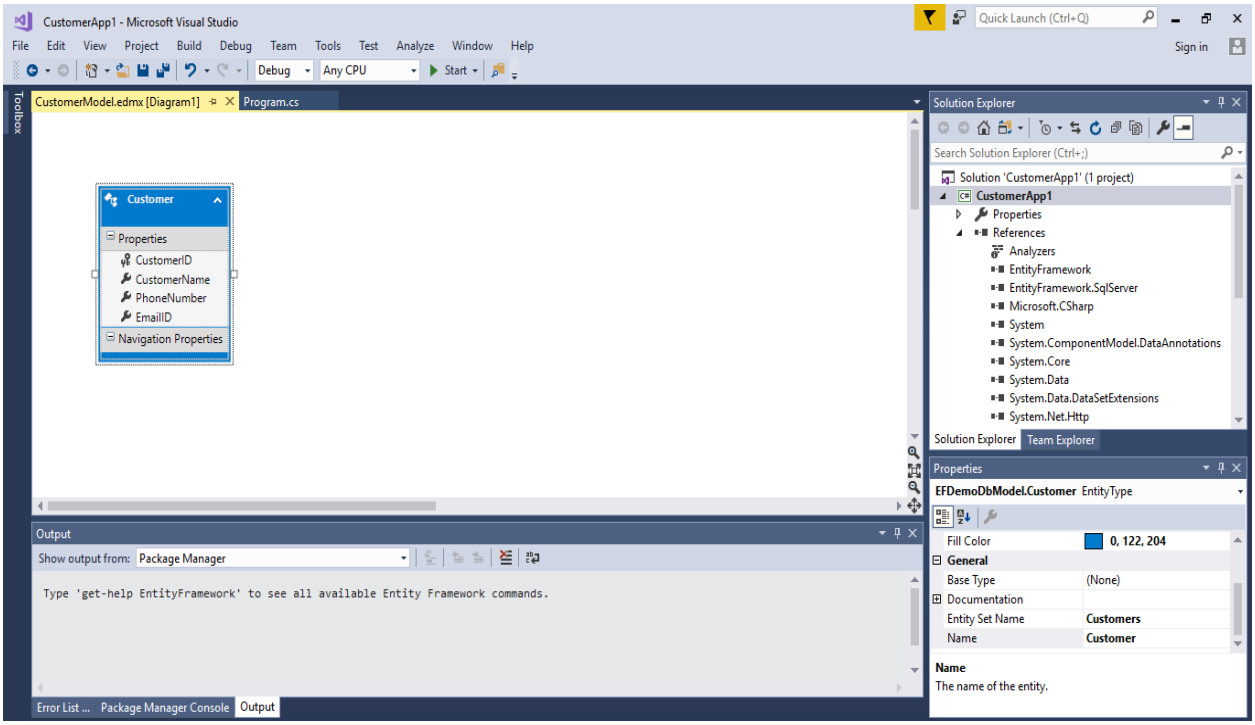

CustomerApp1 - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Add New Item - CustomerApp1
Installed
Visual C# Items
Code
Data
General
Web
Windows Forms
WPF
ASP.NET Core
SQL Server
Online
Sort by: Default
ADO.NET Entity Data Model
DataSet
EF 5.x DbContext Generator
EF 6.x DbContext Generator
Service-based Database
XML File
XML Schema
XSLT File
Visual C# Items
Type: Visual C# Items
A project item for creating an ADO.NET Entity Data Model.
Name: CustomerModel.edmx
Add
Cancel


CustomerApp1 - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Entity Data Model Wizard
Choose Model Contents
What should the model contain?
EF Designer from database
Empty EF Designer model
Empty Code First model
Code First from database
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.
< Previous
Next >
Finish
Cancel
Solution Explorer
Solution 'CustomerApp1' (1 project)
CustomerApp1
Properties
References
App.config
Program.cs
Properties
CustomerApp1 Project Properties
Misc
Project File
CustomerApp1.csproj
Project Folder
F:\DotNet\CustomerApp1\Custo

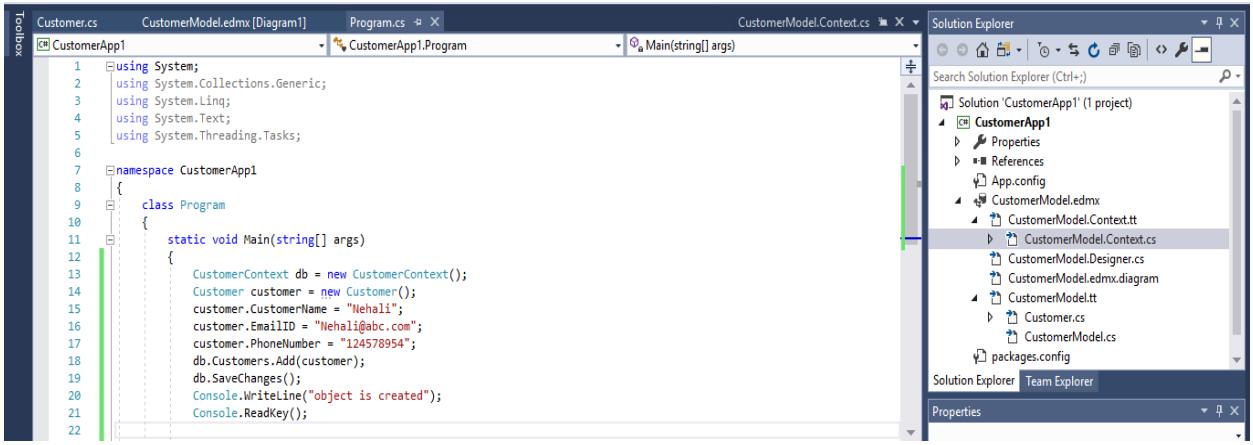

CustomerApp1 - Microsoft Visual Studio
Connection Properties
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.
Data source:
Microsoft SQL Server (SqlClient)
Change...
Server name:
.\nehali123
Refresh
Log on to the server
Authentication:
Windows Authentication
User name:
Password:
Save
Connect to a database
Select or enter a database
EFDemoDb
Attach a database file:
Browse...
Logical name:
Advanced...
Test Connection
OK
Cancel
Microsoft Visual Studio
Test connection succeeded.
OK
Solution Explorer
Solution 'CustomerApp1' (1 project)
CustomerApp1
Properties
References
App.config
Program.cs
Properties
CustomerApp1 Project Properties
Misc
Project File
CustomerApp1.csproj
Project Folder
Output
Error List
Package Manager Console
Output

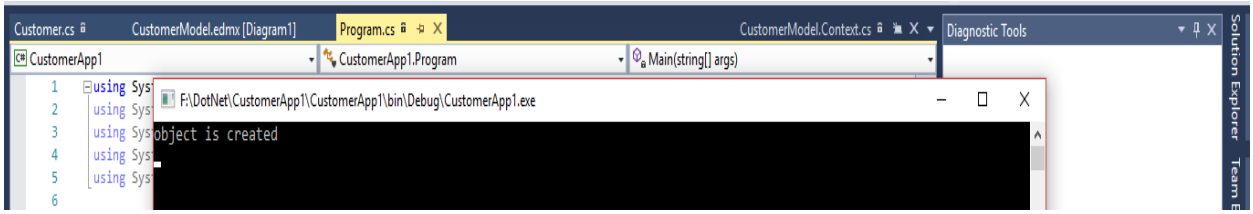

CustomerApp1 - Microsoft Visual Studio
Entity Data Model Wizard
Choose Your Data Connection
Which data connection should your application use to connect to the database?
nehali-pc\nehali123.EFDemoDb.dbo
New Connection...
This connection string appears to contain sensitive data (for example, a password) 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://*/CustomerModel.csdl|res://*/CustomerModel.ssdl|
res://*/CustomerModel.msl;provider=System.Data.SqlClient;provider connection string="data source=.
\nehali123;initial catalog=EFDemoDb;integrated
security=True;MultipleActiveResultSets=True;App=EntityFramework"
Save connection settings in App.Config as:
CustomerContext
< Previous
Next >
Finish
Cancel
Solution Explorer
Solution 'CustomerApp1' (1 project)
CustomerApp1
Properties
References
App.config
Program.cs
Properties
CustomerApp1 Project Properties
Misc
Project File
CustomerApp1.csproj
Project Folder
Output
Error List
Package Manager Console
Output


CustomerApp1 - Microsoft Visual Studio
Entity Data Model Wizard
Choose Your Database Objects and Settings
Which database objects do you want to include in your model?
Tables
dbo
Customers
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:
EFDemoDbModel
< Previous
Next >
Finish
Cancel
Solution Explorer
Properties

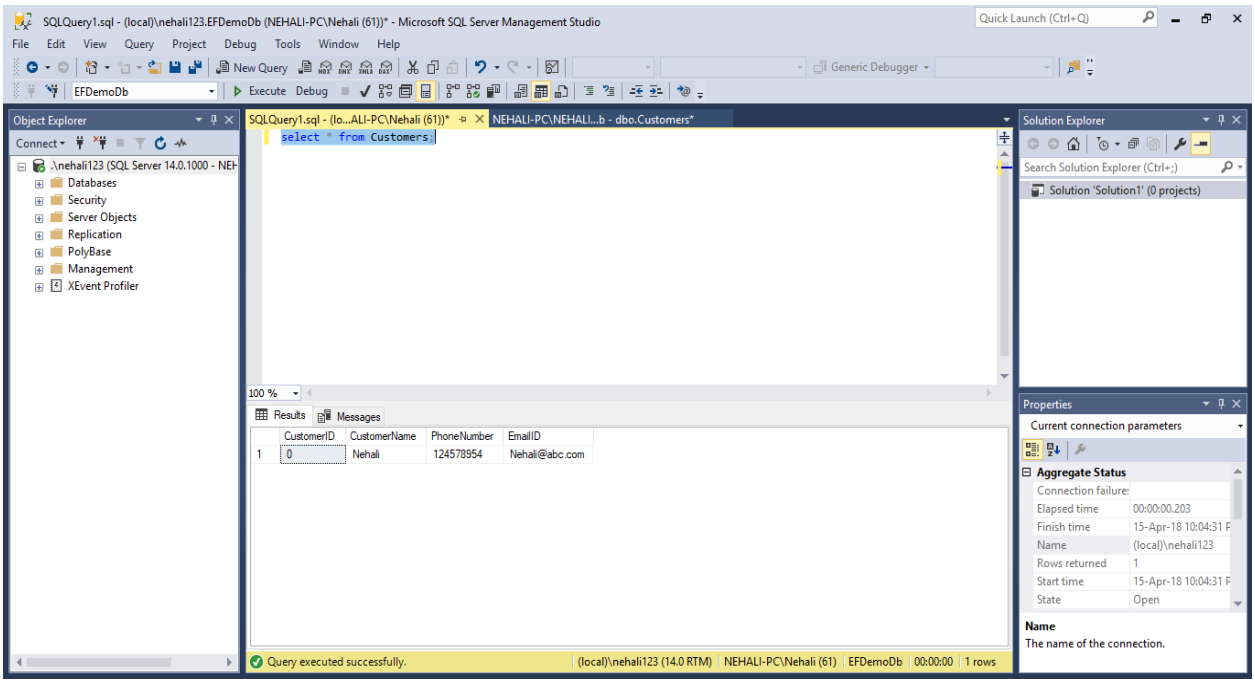

CustomerApp1 - Microsoft Visual Studio
CustomerModel.edmx [Diagram1]
Program.cs
Customer
Properties
CustomerID
CustomerName
PhoneNumber
EmailID
Navigation Properties
Solution Explorer
Properties
Output
Show output from: Package Manager
Type 'get-help EntityFramework' to see all available Entity Framework commands.

Step 3: Insert a record.

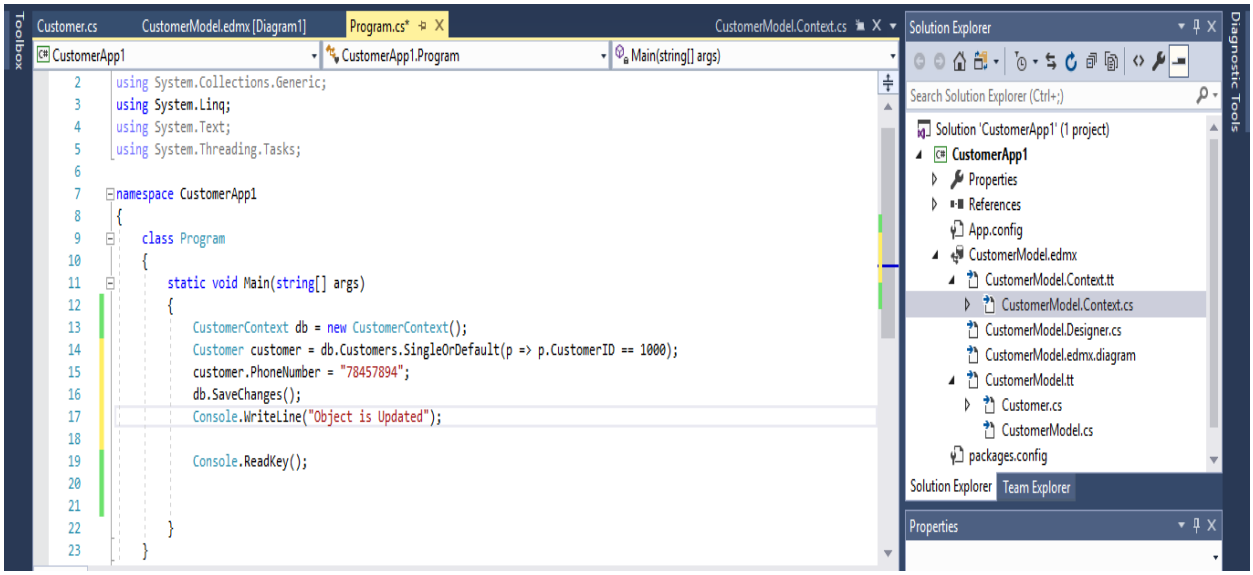




Step 4: Record is inserted.



Step 5: Code for Updating Record



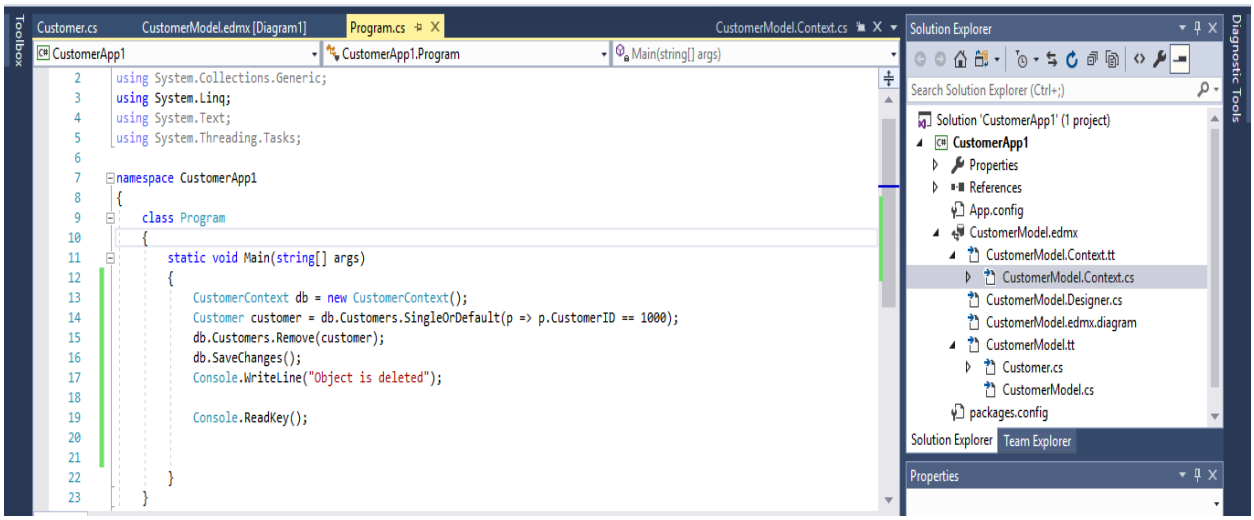
```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CustomerApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            CustomerContext db = new CustomerContext();
            Customer customer = db.Customers.SingleOrDefault(p => p.CustomerID == 1000);
            customer.PhoneNumber = "78457894";
            db.SaveChanges();
            Console.WriteLine("Object is Updated");

            Console.ReadKey();


        }
    }
```

Step 6: Code for Delete Record.



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

namespace CustomerApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            CustomerContext db = new CustomerContext();
            Customer customer = db.Customers.SingleOrDefault(p => p.CustomerID == 1000);
            db.Customers.Remove(customer);
            db.SaveChanges();
            Console.WriteLine("Object is deleted");

            Console.ReadKey();


        }
    }
```