



Bài 5

Thao tác CSDL với ADO.NET & Entity Framework



Giới thiệu về ADO.NET và Entity Framework

Kết nối và truy vấn CSDL với ADO.NET

Kết nối và thao tác CSDL với Entity Framework

Sử dụng Service tại các client khác nhau

ADO.NET là tập hợp các thư viện được cung cấp bởi Microsoft nhằm giúp cho ứng dụng có thể thao tác (select, insert, update, delete) tới cơ sở dữ liệu (CSDL) bất kỳ như Sql Server, Oracle, MySql, SQLite...

Entity Framework là một ORM (Object-Relational Mapping) xây dựng dựa trên ADO.NET để tiếp tục đơn giản hóa lập trình với cơ sở dữ liệu.

Entity Framework được sử dụng đặc biệt phổ biến trong phát triển ứng dụng .NET thay cho lập trình trực tiếp với ADO.NET.

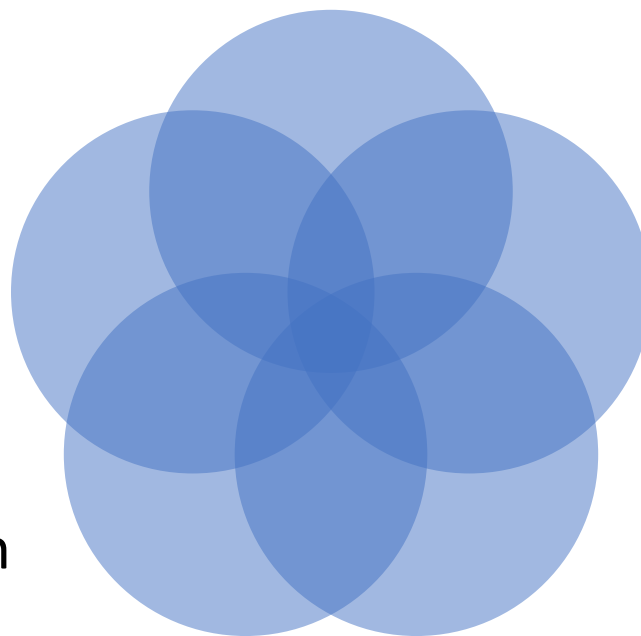
ADO.NET và Entity Framework không phải là hai công nghệ độc lập. Entity Framework được xây dựng bên trên ADO.NET, sử dụng một bộ phận (Data Provider) của ADO.NET.

Tuy nhiên, chúng ta lại có thể học hai công nghệ này hoàn toàn độc lập

ADO.NET

Demo CRUD
cơ bản

Các truy vấn
cơ bản với
ADO.NET



Gọi thủ tục
trong
ADP.NET 1/5

Cấu hình kết
nối CSDL với
ADO.NET

- Cấu hình kết nối trong file **Web.config**
- Chuỗi kết nối đặt trong thành phần ConnectionString, cần cung cấp đầy đủ thông tin kết nối CSDL như
- Tên chuỗi kết nối: sẽ được gọi khi thực hiện kết nối
- Tên máy chủ, tên CSDL cần kết nối, thông tin user và pass quản lý CSDL

```
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />
  </startup>
  <connectionStrings>
    <add name="DefaultConnection" connectionString="Data Source=Server_name|Server IP; Initial Catalog=DbName; User ID=sa; Password=123a@" />
  </connectionStrings>
</configuration>
```

Chuỗi kết nối

Thông tin Máy chủ Sql Server

Tên CSDL

User + Pass

Ví dụ

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />
  </startup>
  <connectionStrings>
    <add name="DefaultConnection" connectionString="Data Source=DESKTOP-HQE4MR1\SQL2017; Initial Catalog=QL_SV; User ID=sa; Password=123a@" />
  </connectionStrings>
</configuration>
```

Có thể tạo một C# class để đảm nhận các nghiệp vụ thao tác CSDL hoặc tại nơi cần kết nối CSDL thì sử dụng code tương tự sau đây



```
// lấy chuỗi kết nối đã cấu hình tại file Web.config
string constring = ConfigurationManager.ConnectionStrings["constr"].ToString();
// thực hiện kết nối
SqlConnection conn = new SqlConnection(constring);
conn.Open(); // mở kết nối
// code truy vấn dữ liệu tại đây
conn.Close(); // đóng kết nối
```

Ví dụ

```
string constr = ConfigurationManager.ConnectionStrings["constr"].ToString();
SqlConnection conn = new SqlConnection(constr);
conn.Open();

string sql = "INSERT INTO Category VALUE(@name, @status)";
SqlCommand cmd = new SqlCommand(sql, conn);
cmd.CommandType = CommandType.Text;
cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = obj.Name;
cmd.Parameters.Add("@status", SqlDbType.Int).Value = obj.Status;
cmd.ExecuteNonQuery();

conn.Close();
return obj;
```

Truy vấn cơ bản trong ADO.NET

```
public List<Category> GetAll()
{
    List<Category> cats = new List<Category>();

    string constr = ConfigurationManager.ConnectionStrings["constr"].ToString();
    SqlConnection conn = new SqlConnection(constr);
    conn.Open();
    SqlCommand cmd = new SqlCommand("SELECT * FROM Category", conn);
    cmd.CommandType = CommandType.Text;
    SqlDataReader dr = cmd.ExecuteReader();
    if (dr.HasRows)
    {
        while (dr.Read())
        {
            cats.Add(new Category
            {
                Id = Convert.ToInt32(dr["Id"].ToString()),
                Name = dr["Name"].ToString(),
                Status = Convert.ToInt32(dr["Status"].ToString())
            });
        }
    }
    conn.Close();
    return cats;
}
```

SELECT * FROM

Đọc và gán vào List

```
public Category GetById(int Id)
{
    Category cat = new Category();
    string constr = ConfigurationManager.ConnectionStrings["constr"].ToString();
    SqlConnection conn = new SqlConnection(constr);
    conn.Open();
    string sql = "SELECT * FROM Category WHERE Id = @id";
    SqlCommand cmd = new SqlCommand(sql, conn);
    cmd.CommandType = CommandType.Text;
    cmd.Parameters.Add("@id", SqlDbType.Int).Value = Id;
    SqlDataReader dr = cmd.ExecuteReader();
    if (dr.HasRows && dr.Read())
    {
        cat.Id = Convert.ToInt32(dr["Id"].ToString());
        cat.Name = dr["Name"].ToString();
        cat.Status = Convert.ToInt32(dr["Status"].ToString());
    }
    conn.Close();
    return cat;
}
```

SELECT... WHERE...

Đọc bản ghi, gán cho đối tượng

Truy vấn cơ bản trong ADO.NET

```
public Category AddNew(Category obj)
{
    string constr = ConfigurationManager.ConnectionStrings["constr"].ToString();
    SqlConnection conn = new SqlConnection(constr);
    conn.Open();
    string sql = "INSERT INTO Category VALUE(@name, @status)";
    SqlCommand cmd = new SqlCommand(sql, conn);
    cmd.CommandType = CommandType.Text;
    cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = obj.Name;
    cmd.Parameters.Add("@status", SqlDbType.Int).Value = obj.Status;
    cmd.ExecuteNonQuery();
    conn.Close();
    return obj;
}
```

INSERT INTO

CÁC Parameter

```
public int Remove(int Id)
{
    string constr = ConfigurationManager.ConnectionStrings["constr"].ToString();
    SqlConnection conn = new SqlConnection(constr);
    conn.Open();
    string sql = "DELETE FROM Category WHERE Id = @id";
    SqlCommand cmd = new SqlCommand(sql, conn);
    cmd.CommandType = CommandType.Text;
    cmd.Parameters.Add("@id", SqlDbType.Int).Value = Id;
    int check = cmd.ExecuteNonQuery();
    conn.Close();
    return check;
}
```

DELETE FROM

```
public int Update(Category obj)
{
    string constr = ConfigurationManager.ConnectionStrings["constr"].ToString();
    SqlConnection conn = new SqlConnection(constr);
    conn.Open();
    string sql = "UPDATE Category SET Name = @name, Status = @status WHERE Id = @id";
    SqlCommand cmd = new SqlCommand(sql, conn);
    cmd.CommandType = CommandType.Text;
    cmd.Parameters.Add("@id", SqlDbType.Int).Value = obj.Id;
    cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = obj.Name;
    cmd.Parameters.Add("@status", SqlDbType.Int).Value = obj.Status;
    int check = cmd.ExecuteNonQuery();
    conn.Close();
    return check;
}
```

UPDATE ... SET ...

- Tạo thủ tục lấy ra danh sách sinh viên như ví dụ sau đây

```
CREATE PROC sp_student_list AS  
SELECT Id, Name, Email, Phone FROM Student
```

- Trở lại phương thức load_data(), load danh sách, thay vì sử dụng lệnh SELECT bây giờ sẽ sử dụng thủ tục vừa tạo tên là sp_student_list

```
private void Load_data()  
{  
    DB.OpenConection();  
    DB.cmd.CommandText = "sp_student_list"; // tên thủ tục  
    DB.cmd.CommandType = CommandType.StoredProcedure;  
    SqlDataAdapter da = new SqlDataAdapter(DB.cmd);  
    DataTable dt = new DataTable();  
    da.Fill(dt);  
    // đổ dữ liệu lên DataGridView đã tạo bên mã XAML  
    StdGrid.ItemsSource = dt.DefaultView;  
    DB.CloseConnection();  
}
```

- Tạo thủ tục sp_student_by_id có tham số @id lấy ra thông tin sinh viên theo id

```
CREATE PROC sp_student_by_id  
    @id int  
as  
SELECT Id, Name, EMail, Phone FROM Student  
Where Id = @id;
```

- Sự kiện BtnEdit_click

```
var Tag = ((Button)sender).Tag;  
  
if (Tag != null)  
{  
    var id = Tag.ToString();  
    DB.OpenConection();  
    DB.cmd.CommandText = "sp_student_by_id"; // tên thủ tục  
    DB.cmd.CommandType = CommandType.StoredProcedure;  
    SqlParameter param;  
    param = DB.cmd.Parameters.Add("@id", SqlDbType.Int); // add param @id kiểu dữ liệu int  
    param.Value = id; // gán giá trị cho param  
    SqlDataReader rd = DB.cmd.ExecuteReader();  
    // đổ dữ liệu lên DataGridView đã tạo bên mã XAML  
    if (rd.HasRows) ...  
    DB.CloseConnection();  
}
```

- Tạo thủ tục **sp_add_student** có nhiều tham số

```
CREATE PROC sp_add_student
    @name nvarchar(10),
    @email varchar(100),
    @phone varchar(50)
as
INSERT INTO Student (Name, Email, Phone)
VALUES(@name, @email, @phone)
```

```
private void BtnAdd_Click(object sender, RoutedEventArgs e)
{
    DB.OpenConection();
    DB.cmd.CommandText = "sp_add_student";
    DB.cmd.CommandType = CommandType.StoredProcedure;

    SqlParameter param;
    // add param @name
    param = DB.cmd.Parameters.Add("@name", SqlDbType.NVarChar, 100);
    param.Value = Name.Text; // gán giá trị cho param @name

    // add param @email
    param = DB.cmd.Parameters.Add("@email", SqlDbType.VarChar, 100);
    param.Value = Email.Text; // gán giá trị cho param @email

    // add param @phone
    param = DB.cmd.Parameters.Add("@phone", SqlDbType.VarChar, 50);
    param.Value = Phone.Text; // gán giá trị cho param @phone

    DB.cmd.ExecuteNonQuery();
    DB.CloseConnection();
    Load_data(); // gọi lại phương thức load lại dữ liệu trên DataGrid
}
```

Gọi thủ tục có nhiều hơn 1 tham số 4/5

- Sử dụng phương thức `AddWithValue` để thêm tham số và giá trị

```
private void BtnAdd_Click(object sender, RoutedEventArgs e)
{
    DB.OpenConection();
    DB.cmd.CommandText = "sp_add_student";
    DB.cmd.CommandType = CommandType.StoredProcedure;
    // add param @name
    DB.cmd.Parameters.AddWithValue("@name", Name.Text);
    // add param @email
    DB.cmd.Parameters.AddWithValue("@email", Email.Text);
    // add param @phone
    DB.cmd.Parameters.AddWithValue("@phone", Phone.Text);

    DB.cmd.ExecuteNonQuery();
    DB.CloseConnection();
    Load_data(); // gọi lại phương thức load lại dữ liệu trên DataGridView
}
```

AddWithValue

- Có thể truyền tham số theo thứ tự của nó trong thủ tục

```
Create PROC sp_add_student
    @name nvarchar(10), 1
    @email varchar(100), 2
    @phone varchar(50) 3
as
INSERT INTO Student (Name, Email, Phone)
VALUES(@name, @email, @phone)
```

```
private void BtnAdd_Click(object sender, RoutedEventArgs e)
{
    DB.OpenConection();
    DB.cmd.CommandText = "sp_add_student";
    DB.cmd.CommandType = CommandType.StoredProcedure;

    SqlCommandBuilder.DeriveParameters(DB.cmd);

    // add param 1 => @name
    DB.cmd.Parameters[1].Value = Name.Text;
    // add param 2 => @email
    DB.cmd.Parameters[2].Value = Email.Text;
    // add param 3 => @phone
    DB.cmd.Parameters[3].Value = Phone.Text;

    DB.cmd.ExecuteNonQuery();
    DB.CloseConnection();
    Load_data(); // gọi lại phương thức laod lại dữ liệu trên DataGrid
}
```

Tham số theo thứ tự



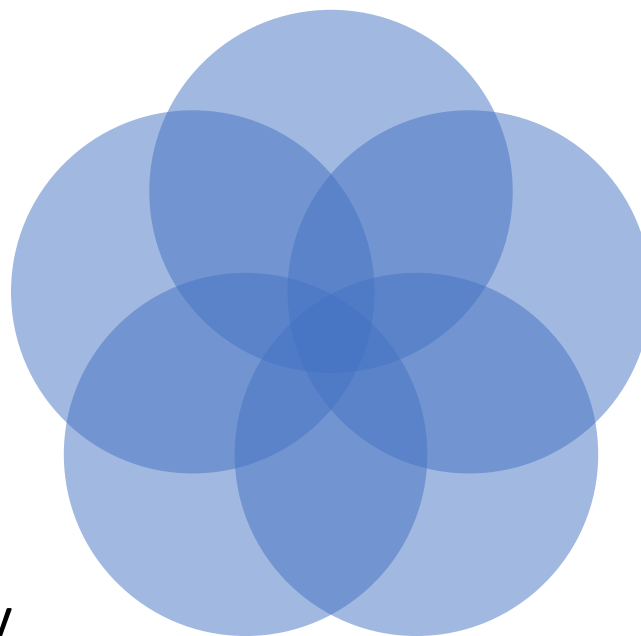
LIVE DEMO



Entity Framework

Demo CRUD
cơ bản

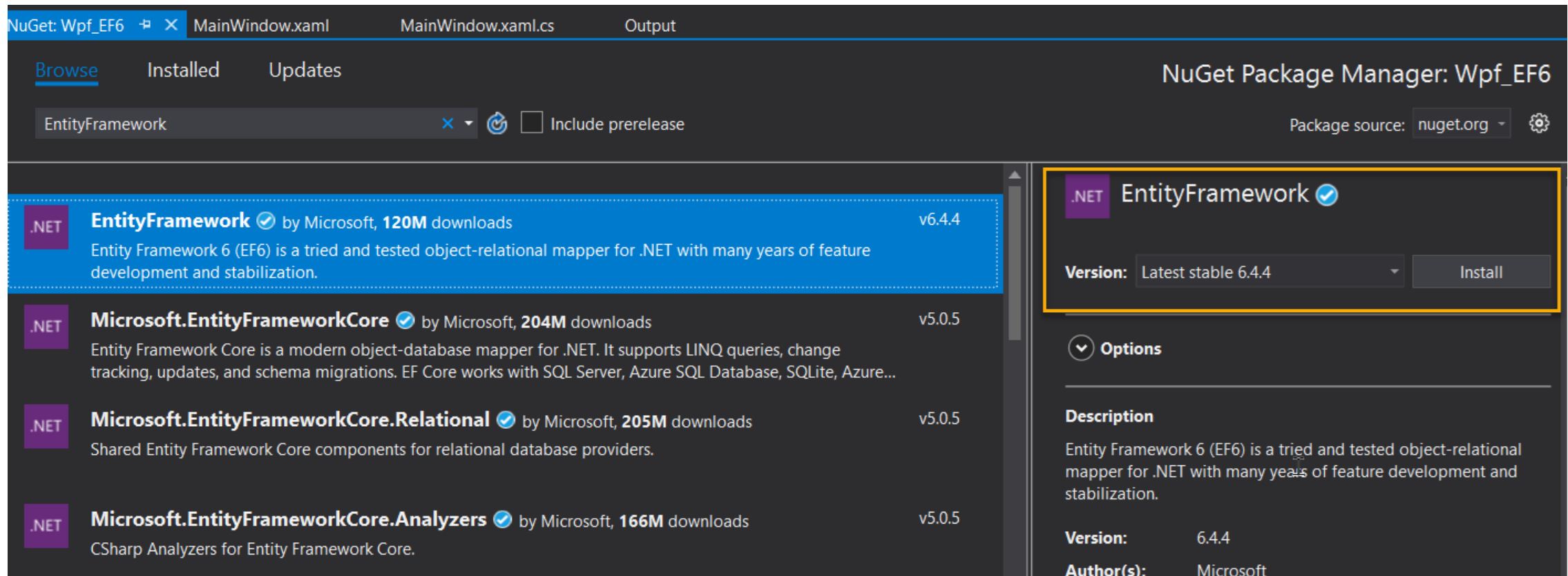
Truy vấn dữ
liệu với Entity
Framework



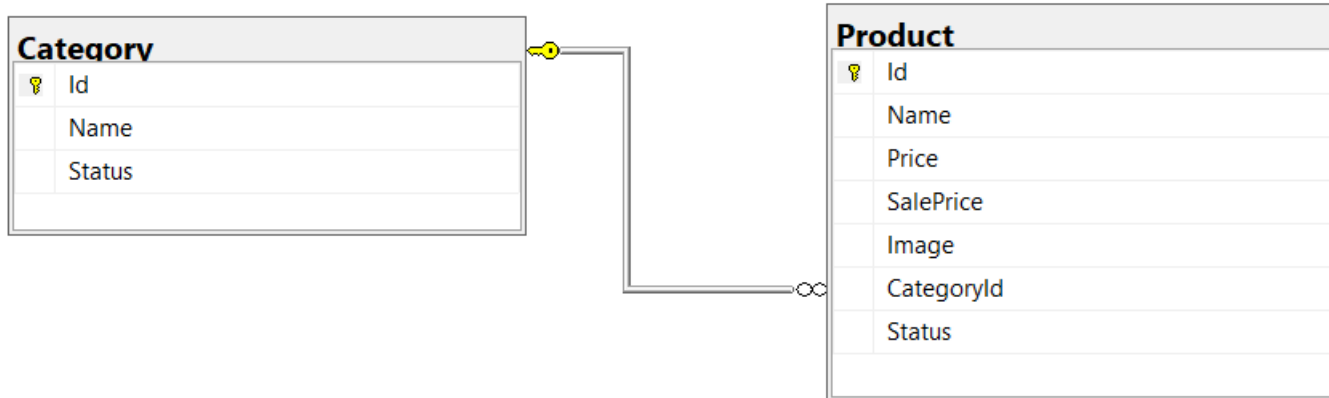
Cài đặt Entity
Framework

Cấu hình kết
nối CSDL với
Entity
Framework

- Tạo WCF Service Project và Mở NuGet Packages Manager và tìm EntityFramework để cài đặt vào dự án



- Có 2 phương pháp Code First và Database First
- Hướng dẫn sau đây sử dụng Database First
- Trước hết cần có một Database có các bảng, view, thủ tục ...
- VD Db là StoreManager có các bảng sau



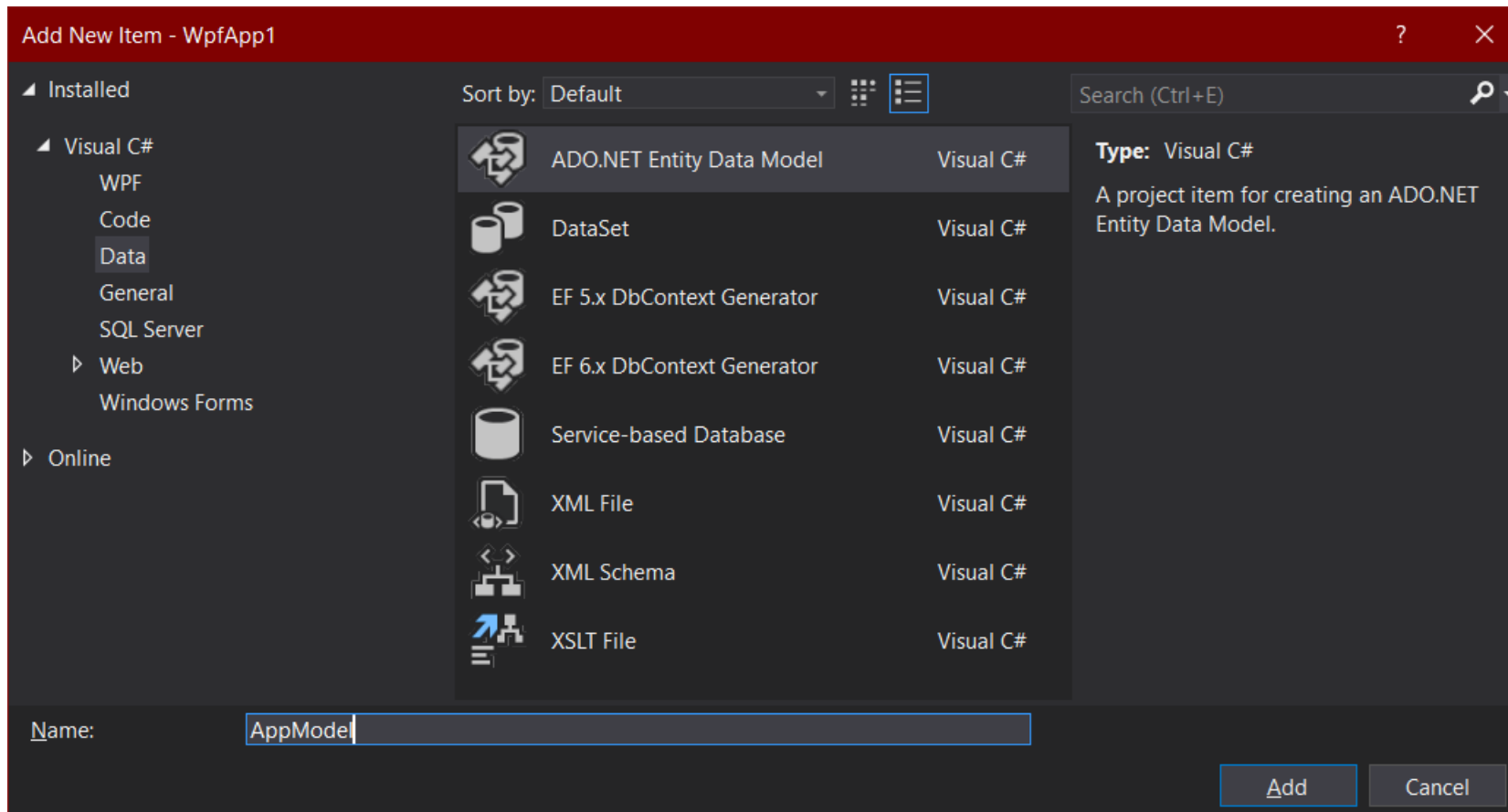
```
INSERT INTO Category(Name, Status) VALUES
(N'Đồng Hồ',1),
(N'Điện thoại',0),
(N'Phụ kiện điện thoại',1),
(N'Sạc dự phòng',0),
(N'Laptop',1),
(N'Loa cao cấp',1)
```

```
CREATE TABLE Category
(
    Id int primary Key identity,
    Name nvarchar(100) NOT NULL,
    Status tinyint DEFAULT(1)
)

GO

CREATE TABLE Product
(
    Id int primary Key identity,
    Name nvarchar(100) NOT NULL,
    Price float NOT NULL,
    SalePrice float DEFAULT(0),
    Image varchar(200) NULL,
    CategoryId int NOT NULL REFERENCES Category(Id),
    Status tinyint DEFAULT(1)
)
```

- Click phải chuột lên thư mục Models (Hoặc nơi mà bạn muốn lưu các model trong project của bạn)
- Chọn Add Item và chọn đến template (ADO.NET Entity Data Model), đặt tên tùy chọn



Kết nối CSDL với EntityFramework

Entity Data Model Wizard

Choose Your Data Connection

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

desktop-hqe4mr1\sql2017.WPFDDB.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:/*/*Data.AppEntity.csdl|res:/*/*Data.AppEntity.ssdl|res:/*/*Data.AppEntity.msl;provider=System.Data.SqlClient;provider connection string="data source=DESKTOP-HQE4MR1\SQL2017;initial catalog=WPFDDB;persist security info=True;user id=sa;password=*****;MultipleActiveResultSets=True;App=EntityFramework"

☒ Save connection settings in App.Config as:

WPFDDBEntities

< Previous Next > Finish Cancel

Server name:

DESKTOP-HQE4MR1\SQL2017

Refresh

Log on to the server

Authentication: SQL Server Authentication

User name: sa

Password:

☐ Save my password

Connect to a database

☒ Select or enter a database name:

StoreManager

☐ Attach a database file:

Browse...

Logical name:

Máy chủ SQL

Đăng nhập

CSDL

Kế tiếp

Tạo kết nối mới

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

desktop-hqe4mr1\sql2017.StoreManager.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.
- ☒ Yes, include the sensitive data in the connection string.

Connection string:

```
metadata=res://*/Models.AppModel.csdl|res://*/Models.AppModel.ssdl|
res://*/Models.AppModel.msl;provider=System.Data.SqlClient;provider connection
string="data source=DESKTOP-HQE4MR1\SQL2017;initial catalog=StoreManager;user
id=sa;password=*****;MultipleActiveResultSets=
```

☒ Save connection settings in App.Config as:

StoreManagerEntities

Tạo chuỗi kết nối

Tên Entity tùy chọn

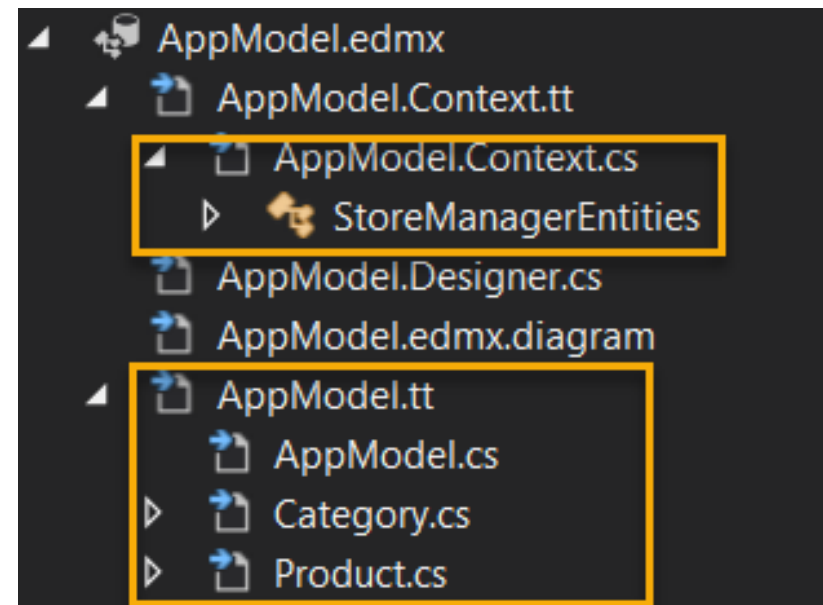
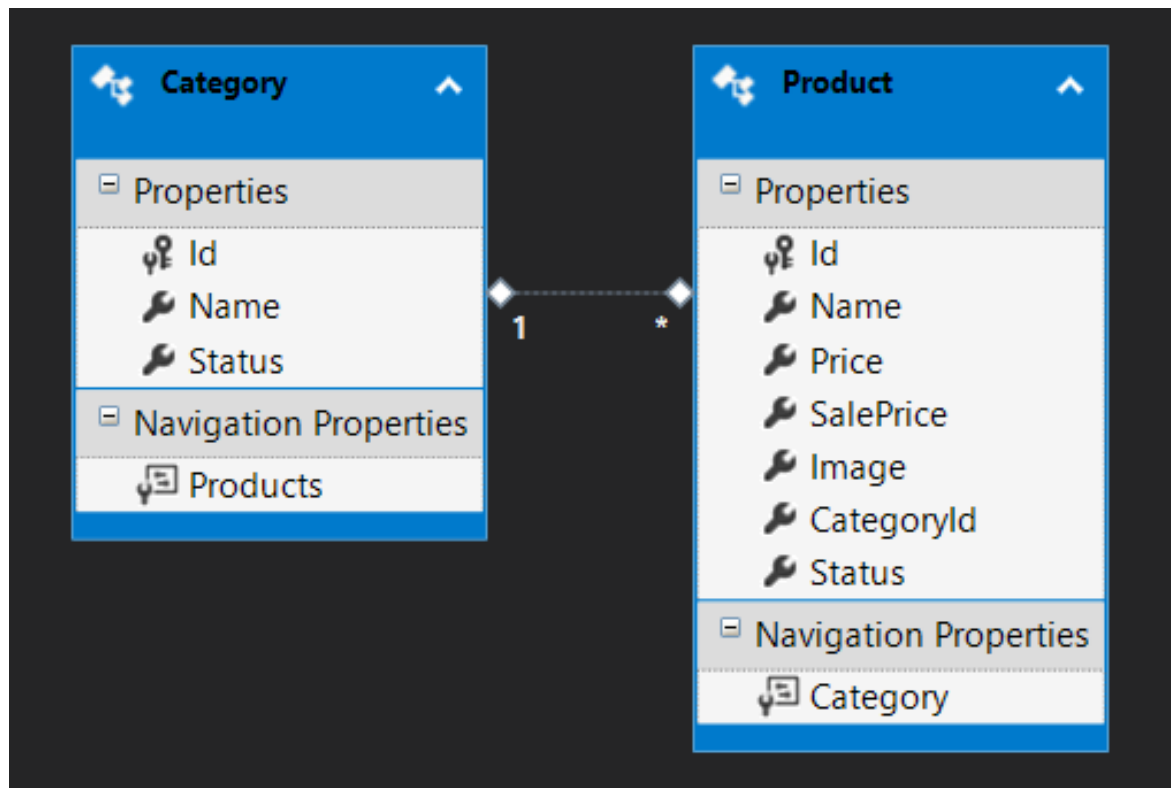
Which database objects do you want to include in your model?

- ☒ Tables
 - ☒ dbo
 - ☒ Category
 - ☒ Product
 - ☐ sysdiagrams
 - ☐ 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:

StoreManagerModel



```
<connectionStrings>  
  <add name="StoreManagerEntities" connectionString="metadata=res://*/Models.AppModel.csd1|res://*/Models.AppModel.ssd1|res://*/Models.AppModel.ssdl" provider="System.Data.EntityClient" providerInvariant="true" />  
</connectionStrings>
```


Tạo Các DataContract để Mapping với các Model

Tạo CategoryDataContract bằng cách click phải chuột vào thư mục dự án Add -> Class đặt tên là CategoryDataContruct, khai báo như sau

```
namespace WcfEntityDemo
{
    using System;

    public partial class Category
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public Nullable<byte> Status { get; set; }
    }
}
```

Model

```
using System;
using System.Runtime.Serialization;

namespace WcfEntityDemo
{
    [DataContract]
    public class CategoryDataContract
    {
        [DataMember]
        public int Id { get; set; }
        [DataMember]
        public string Name { get; set; }
        [DataMember]
        public Nullable<byte> Status { get; set; }
    }
}
```

DataContract

Tại các phương thức của DataService

```
public CategoryDataContract GetById(int Id)
{
    MyDataEntities _db = new MyDataEntities();
    CategoryDataContract item = new CategoryDataContract();
    Category cat = _db.Category.Find(Id);
    item.Id = cat.Id;
    item.Name = cat.Name;
    item.Status = cat.Status;
    return item;
}
```

Truy vấn theo Id

```
public int Remove(int Id)
{
    MyDataEntities _db = new MyDataEntities();
    Category obj = _db.Category.Find(Id);
    _db.Category.Remove(obj);
    return _db.SaveChanges();
}
```

DELETE

```
public List<CategoryDataContract> GetAll()
{
    MyDataEntities _db = new MyDataEntities();
    var data = _db.Category.ToList();
    List<CategoryDataContract> DataList = new List<CategoryDataContract>();
    foreach(Category cat in data)
    {
        var item = new CategoryDataContract
        {
            Id = cat.Id,
            Name = cat.Name,
            Status = cat.Status
        };
        DataList.Add(item);
    }
    return DataList;
}
```

Truy vấn danh sách

Duyệt dữ liệu từ Model gán cho DataContract

```
public int Update(CategoryDataContract obj)
{
    MyDataEntities _db = new MyDataEntities();
    Category cat = _db.Category.Find(obj.Id);
    cat.Name = obj.Name;
    cat.Status = obj.Status;
    _db.Entry(cat).State = System.Data.Entity.EntityState.Modified;
    return _db.SaveChanges();
}
```

UPDATE



LIVE DEMO



HỎI ĐÁP





TRẢI NGHIỆM THỰC HÀNH

HỆ THỐNG ĐÀO TẠO CNTT QUỐC TẾ BACHKHOA - APTECH



TRÂN TRỌNG CẢM ƠN!



238 Hoàng Quốc Việt, Bắc Từ Liêm, Hà Nội



0968.27.6996



tuyensinh@bachkhoa-aptech.edu.vn



www.bachkhoa-aptech.edu.vn