

Bài 11: Truy vấn và xử lý dữ liệu với Entity Framework Core (Tiếp)

1. Tổng quan về LINQ

2. ORM và EF

3. Entity Framework Core (EF Core)

4. Truy vấn dữ liệu sử dụng EF Core

4.1. Thêm các package cần thiết

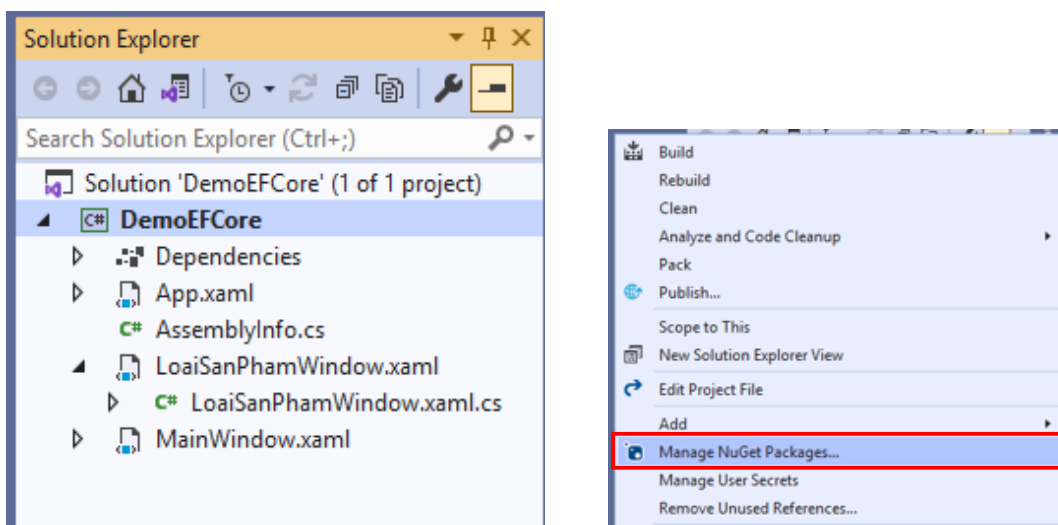
4.1.1. Cài đặt Data Provider

EF Core cho phép truy cập cơ sở dữ liệu qua Data Provider. Data Provider là tập hợp các lớp cho phép ta giao tiếp hiệu quả với cơ sở dữ liệu. EF Core có các Data Provider cho các cơ sở dữ liệu khác nhau. Các Data Provider này có sẵn dưới dạng các gói NuGet., như trong bảng sau:

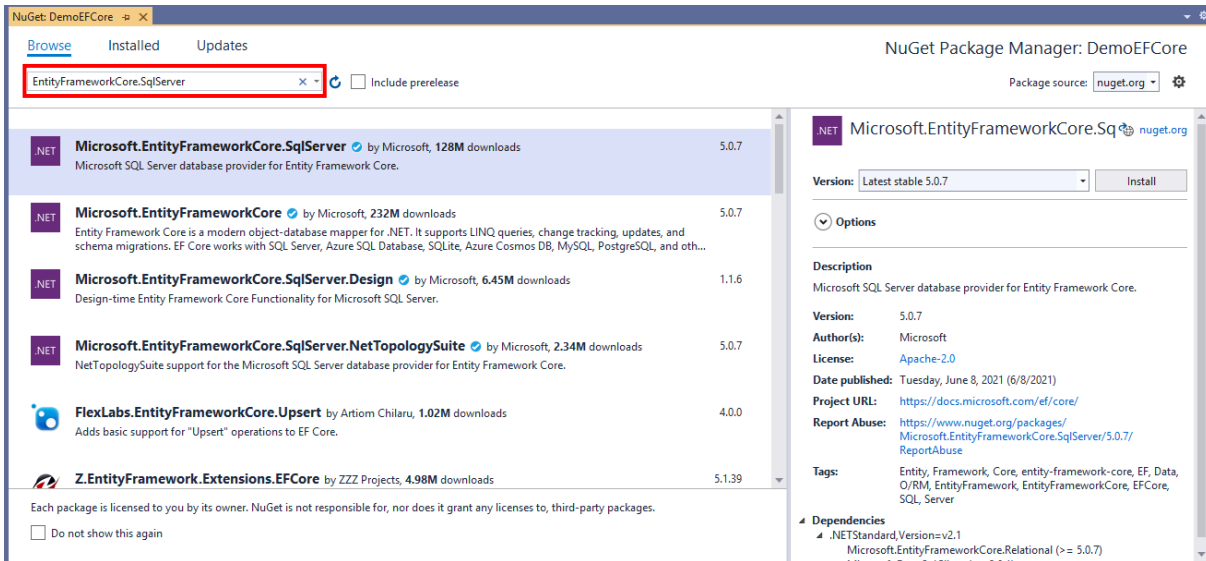
Cơ sở dữ liệu	NuGet packate
Microsoft SQL Server 2012 trở lên	Microsoft.EntityFrameworkCore.SqlServer
SQLite 3.7 trở lên	Microsoft.EntityFrameworkCore.Sqlite
MySQL	MySql.EntityFrameworkCore
Oracle DB 11.2	Oracle.EntityFrameworkCore
Azure Cosmos DB SQL API	Microsoft.EntityFrameworkCore.Cosmos

Ta có thể cài đặt **NuGet package** theo hai cách sau:

Cách 1: Để cài đặt gói NuGet của DataProvider, ta nhấp phải chuột vào project trong cửa sổ **Solution Explorer** và chọn → **Manage NuGet Packages . . .** (hoặc chọn trên menu **Tools** → **NuGet Package Manager** → **Manage NuGet Packages for Solution . . .**)

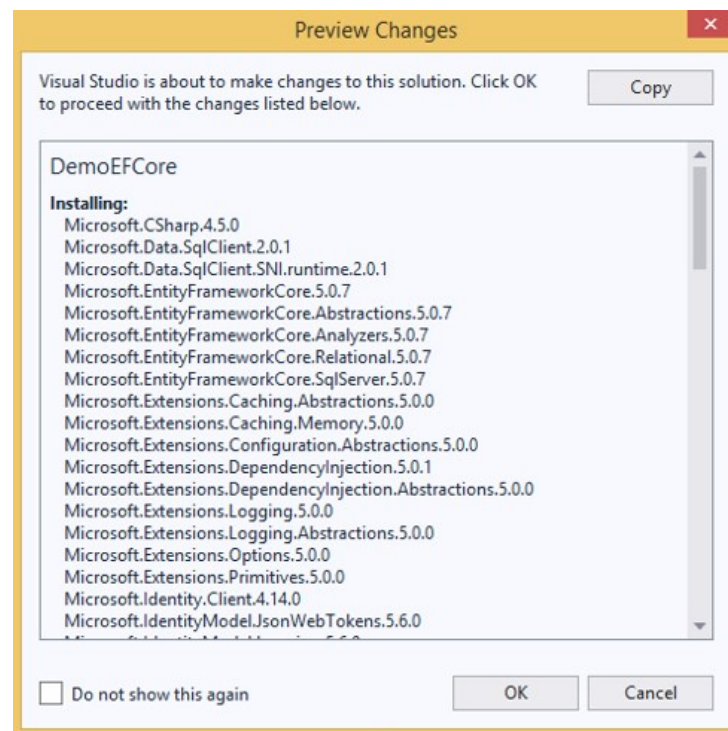


Giao diện trình quản lý gói NuGet được mở. Trong cửa sổ NuGet –Solution → chọn tab Browse và tìm kiếm gói muốn cài đặt như hình sau:

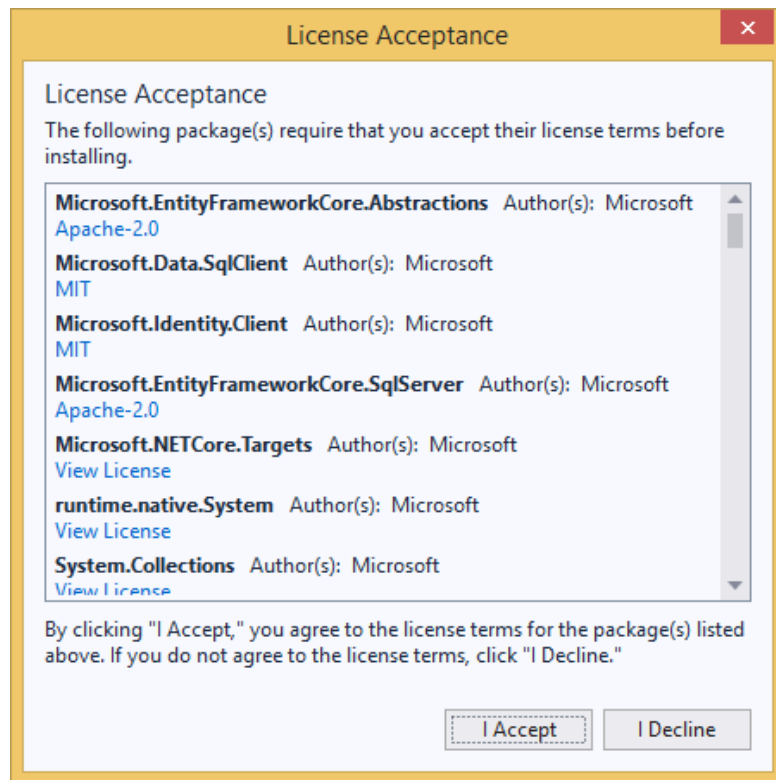


Chọn DataProvider muốn cài. Trong trường hợp này ta muốn truy cập cơ sở dữ liệu SQL Server vì vậy ta cần cài đặt **Microsoft.EntityFrameworkCore.SqlServer**, đảm bảo là nó có logo .NET và tác giả là Microsoft → nhấn Install để cài đặt.

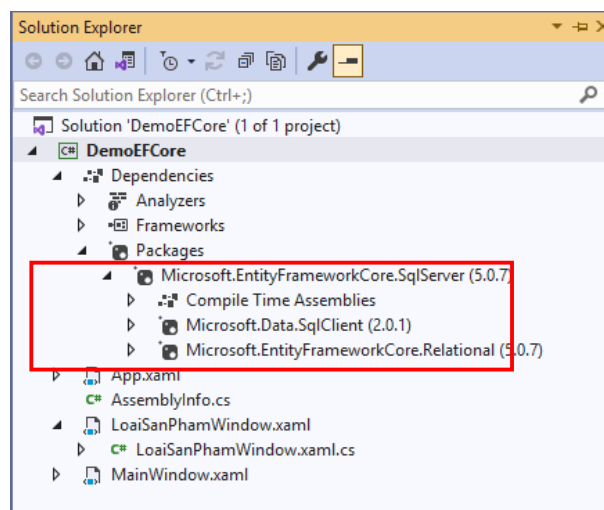
Cửa sổ xem trước hiển thị danh sách các gói sẽ được cài đặt trong ứng dụng, xem lại các thay đổi và nhấn OK



Cuối cùng chấp nhận các điều khoản cấp phép liên quan đến các gói được cài đặt



Sau đó kiểm tra xem gói **Microsoft.EntityFrameworkCore.SqlServer** đã được cài đặt trong Dependencies → NuGet như sau



Cách 2: Cài đặt package bằng **Package Manager Console**

- Chọn menu Tools → NuGet Package Manager → Package Manager Console
- Nhập lệnh theo cú pháp sau vào cửa sổ Package Manager Console

PM> Install-Package tên-package –Version tên-phiên-bản

+ Nếu không có tùy chọn –Version, mặc định cài phiên bản mới nhất

Ví dụ:

PM> Install-Package Microsoft.EntityFrameworkCore.SqlServer –Version 5.0.6

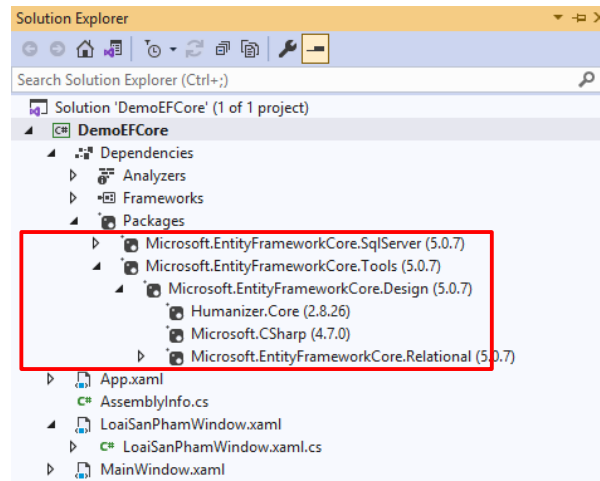
4.1.2. Cài đặt EF Core Tool

Để thực thi các lệnh EF Core ta cần cài đặt NuGet package

Microsoft.EntityFrameworkCore.Tools, điều này giúp ta thực hiện các nhiệm vụ liên quan đến EF Core trong dự án tại thời điểm thiết kế như scaffolding ...

Cài đặt EFCore Tool cũng thực hiện tương tự như cài đặt DataProvider.

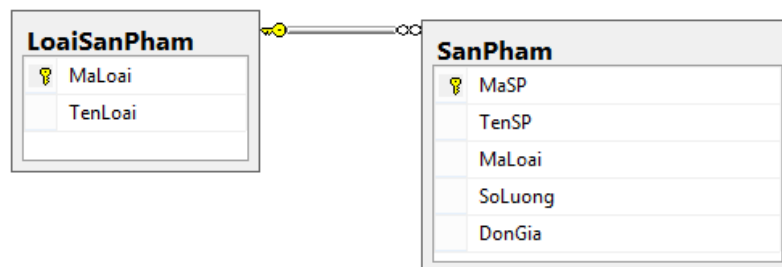
Kiểm tra xem gói **Microsoft.EntityFrameworkCore.Tools** đã được cài đặt thành công trong Dependencies → NuGet:



4.2. Tạo model

4.2.1. Chuẩn bị cơ sở dữ liệu

Tạo cơ sở dữ liệu **QLBanHang** gồm 2 bảng **LoaiSanPham** và **SanPham** có lược đồ cơ sở dữ liệu như sau:



Thiết kế của 2 bảng:

NHUNGNT.QLBanH...dbo.LoaiSanPham X			
	Column Name	Data Type	Allow Nulls
🔑	MaLoai	char(3)	<input type="checkbox"/>
	TenLoai	nvarchar(50)	<input type="checkbox"/>

NHUNGNT.QLBanHang - dbo.SanPham X			
	Column Name	Data Type	Allow Nulls
🔑	MaSP	char(4)	<input type="checkbox"/>
	TenSP	nvarchar(50)	<input type="checkbox"/>
	MaLoai	char(3)	<input checked="" type="checkbox"/>
	SoLuong	int	<input checked="" type="checkbox"/>
	DonGia	int	<input checked="" type="checkbox"/>

4.2.2. Tạo model

EF Core không hỗ trợ thiết kế trực quan cho mô hình cơ sở dữ liệu, tạo các lớp thực thể và lớp Context như các phiên bản trước. Thay vào đó ta sử dụng lệnh **Scaffold-DbContext**. Lệnh này tạo các lớp thực thể và Context (bằng cách dẫn xuất từ lớp DbContext) dựa trên lược đồ của cơ sở dữ liệu hiện có.

- Chọn menu Tools → NuGet Package Manager → Package Manager Console

Trong cửa sổ **Package Manager Console** thực hiện lệnh **Scaffold-DbContext** như sau:

Scaffold-DbContext “chuỗi-kết-nối” tên-DataProvider -OutputDir tên-thư-mục

Ví dụ:

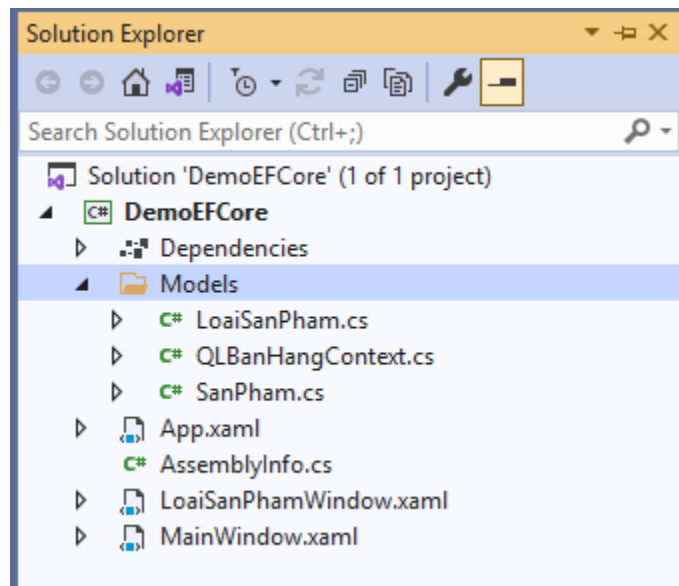
```
Scaffold-DbContext "Data Source= DESKTOP-MTPECJO\MAYAO;Initial
Catalog=QLBanHang;Integrated Security=True"
Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models
```

- Tham số chuỗi kết nối gồm ba phần: máy chủ cơ sở dữ liệu, tên cơ sở dữ liệu và thông tin bảo mật.

Ví dụ, trong chuỗi kết nối: "Data Source= DESKTOP-MTPECJO\MAYAO;Initial Catalog=QLBanHang;Integrated Security=True"

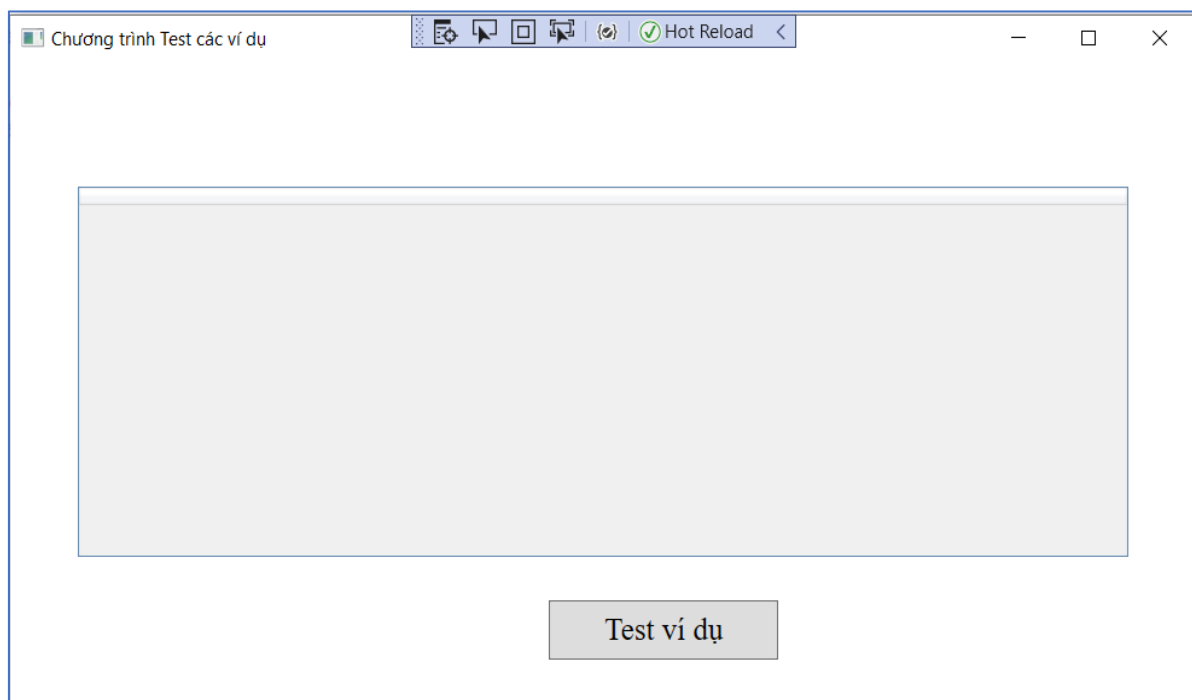
- + Máy chủ cơ sở dữ liệu là : DESKTOP-MTPECJO\MAYAO
- + Tên cơ sở dữ liệu là: QLBanHang
- + Kết nối với SQL Server sử dụng xác thực Windows Authentication
- Tham số thứ hai là tên DataProvider, vì ta sử dụng cơ sở dữ liệu SQL Server nên trong ví dụ tên DataProvider là Microsoft.EntityFrameworkCore.SqlServer
- Tham số thứ ba là tên thư mục, nơi mà ta muốn lưu tất cả các lớp thực thể, trong ví dụ là thư mục Models của project

Lệnh Scaffold-DbContext tạo các lớp thực thể cho mỗi bảng trong cơ sở dữ liệu QLBanHang và lớp Context với cấu hình cho tất cả các thực thể trong thư mục Models



4.2.3. Truy vấn dữ liệu sử dụng LINQ

Thiết kế giao diện:



❖ SELECT

- **Ví dụ 1:** Hiển thị dữ liệu bảng SanPham

```
using DemoEFCore.Models;
namespace DemoEFCore
{
    /// <summary>
    /// Interaction logic for Test_Vidu.xaml
    /// </summary>
    public partial class Test_Vidu : Window
    {
        public Test_Vidu()
```

```

{
    InitializeComponent();
}
//Tạo thể hiện của lớp Context
QLBanHangContext db = new QLBanHangContext();

private void btnTest_Click(object sender, RoutedEventArgs e)
{
    //Cách 1: Truy vấn LINQ theo biểu thức
    var sp = from p in db.SanPhams
            select p;
    //Hiển thị dữ liệu lên data grid
    dgvLoaiSanPham.ItemsSource = sp.ToList();

    /* Cách 2: Truy vấn LINQ theo phương thức mở rộng
    var sp = db.SanPhams
            .Select(x => x);
    dgvLoaiSanPham.ItemsSource = sp.ToList(); */
}
}

```

• **Ví dụ 2:** Hiển thị DL bảng SanPham (Lựa chọn cột hiển thị)

- + Cách 1: `var sp = from p in db.SanPhams
select new { p.MaSp, p.TenSp, p.MaLoai};`
- + Cách 2: `var sp = db.SanPhams
.Select(p => new {p.MaSp, p.TenSp, p.MaLoai });`

❖ **TAKE, SKIP**

• **Ví dụ 3:** Lấy mẫu tin đầu tiên trong bảng (Lấy sản phẩm đầu tiên trong bảng SanPham).

- + Cách 1: `var sp = (from p in db.SanPhams
select p).Skip(0).Take(1);` //Tương đương SELECT TOP 1 trong SQL
- + Cách 2: `var sp = db.SanPhams
.Select(x => x).Skip(0).Take(1);`

• **Ví dụ 4:** Tương tự trên, nhưng lấy mẫu tin cuối cùng (kết hợp 2 cách)

```

var sp = (from p in db.SanPhams
orderby p.MaSp descending
select p).Skip(0).Take(1);

```

• **Ví dụ 5:** Tương tự trên, nhưng lấy mẫu tin thứ 5 và 6

```

var sp = (from p in db.SanPhams
select p).Skip(4).Take(2);

```

❖ **ORDERBY**

• **Ví dụ 6:** Sắp xếp giảm dần theo cột đơn giá (mặc định tăng dần)

- + Cách 1: `var sp = from p in db.SanPhams
orderby p.DonGia descending
select new { p.MaSp, p.TenSp, p.DonGia, p.SoLuong,
ThanhTien= p.DonGia*p.SoLuong };`

+ Cách 2:

```
var sp = db.SanPhams
    .OrderByDescending(x=>x.DonGia)
    .Select(x=>new {x.MaSp, x.TenSp, x.DonGia,
        ThanhTien= x.DonGia*x.SoLuong });
```

❖ DISTINCT

- **Ví dụ 7:** Loại bỏ các phần tử trùng nhau

```
var sp = (from p in db.SanPhams
    select new { p.MaSp, p.TenSp, p.MaLoai}). Distinct();
```

❖ WHERE

- **Ví dụ 8:** Lấy điều kiện theo MaLoai

+ Cách 1:

```
var sp = from p in db.SanPhams
    where p.MaLoai=="L01"
    select new { p.MaSP, p.TenSP, p.MaLoai };
```

+ Cách 2:

```
var sp = db.SanPhams
    .Where (p=>p.MaLoai=="L01")
    .Select(x => new { x.MaSP, x.TenSP, x.DonGia });
```

❖ JOIN

- **Ví dụ 9:** Liên kết nhiều bảng

```
var sp = from p in db.SanPhams
    join k in db.LoaiSanPhams
    on p.MaLoai equals k.MaLoai
    select new { p.MaSp, p.TenSp, k.TenLoai };
```

❖ GROUP

- **Ví dụ 10:** Nhóm dữ liệu (Tính tổng đơn giá theo loại sản phẩm)

```
var sp = from p in db.SanPhams
    join k in db.LoaiSanPhams
    on p.MaLoai equals k.MaLoai
    group p by p.MaLoai into kq
    select new { MaLoai = kq.Key, TongTien = kq.Sum(t => t.DonGia*t.SoLuong)
};
```

Kết quả:

MaLoai	TongTien
01	126000
02	380000

4.2.4. Truy vấn, cập nhật dữ liệu trên bảng Loại sản phẩm

Giao diện

Mã loại	Tên loại sản phẩm
L01	Gia dụng
L02	Điện lạnh
L03	Điện tử
L04	Laptop

Yêu cầu:

- Ngay khi hiển thị cửa sổ, dữ liệu trong bảng Loại sản phẩm đã hiển thị trong data grid.
- Người dùng nhập mã loại, tên loại sản phẩm sau đó nhấn nút **Thêm**, loại sản phẩm mới được thêm vào bảng. Có kiểm tra không cho thêm loại sản phẩm trùng mã. Hiển thị lại thông tin sau khi thêm.
- Người dùng nhập mã và tên mới của loại sản phẩm muốn sửa, sau đó nhấn nút **Sửa**, thông tin của loại sản phẩm được lưu vào bảng. Có thông báo nếu không tìm thấy loại sản phẩm muốn sửa. Hiển thị lại thông tin sau khi sửa.
- Người dùng nhập mã của loại sản phẩm muốn xóa, sau đó nhấn nút **Xóa**, loại sản phẩm bị xóa khỏi bảng. Có thông báo nếu không tìm thấy loại sản phẩm muốn xóa. Hiển thị lại thông tin sau khi xóa.

*Code XAML để tạo giao diện:

```
<Window x:Class="DemoEFCore.CapNhat_1_Bang"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:DemoEFCore"
        mc:Ignorable="d"
        Title="Quản lý loại sản phẩm" Height="440" Width="645"
        FontFamily="Times New Roman" FontSize="20"
        Loaded="Window_Loaded">
<Grid Margin="0,22,0,0">
    <Label Content="Mã loại sản phẩm:" HorizontalAlignment="Left" Margin="88,30,0,0"
            VerticalAlignment="Top" Height="40"/>
    <Label Content="Tên loại sản phẩm" HorizontalAlignment="Left" Margin="88,80,0,0"
            VerticalAlignment="Top" Height="40"/>
</Grid>
```

```

<TextBox Name="txtMa" HorizontalAlignment="Left" Margin="287,30,0,0" Text=""
    TextWrapping="Wrap" VerticalAlignment="Top" Width="250" Height="40"/>
<TextBox Name="txtTen" HorizontalAlignment="Left" Margin="287,80,0,0" Text=""
    TextWrapping="Wrap" VerticalAlignment="Top" Width="250" Height="40"/>

<DataGrid Name="dgvLoaiSanPham" HorizontalAlignment="Left" Height="180"
    Margin="74,136,0,0" VerticalAlignment="Top" Width="510"/>

<Button Name="btnThem" Content="Thêm" HorizontalAlignment="Left" Height="40"
    Margin="100,332,0,0" VerticalAlignment="Top" Width="100" Click="btnThem_Click"/>
<Button Name="btnSua" Content="Sửa" HorizontalAlignment="Left" Height="40"
    Margin="220,332,0,0" VerticalAlignment="Top" Width="100" Click="btnSua_Click"/>
<Button Name="btnXoa" Content="Xóa" HorizontalAlignment="Left" Height="40"
    Margin="340,332,0,0" VerticalAlignment="Top" Width="100" Click="btnXoa_Click"/>
<Button Name="btnThoat" Content="Thoát" HorizontalAlignment="Left" Height="40"
    Margin="460,332,0,0" VerticalAlignment="Top" Width="100"/>
</Grid>
</Window>

```

*Code C# xử lý:

using DemoEFCore.Models;

```

namespace DemoEFCore
{
    public partial class CapNhat_LoaiSanPham : Window
    {
        public CapNhat_LoaiSanPham()
        {
            InitializeComponent();

            //Tạo thể hiện của lớp Context
            QLBanHangContext db = new QLBanHangContext();
        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            HienThiDuLieu();
        }
        private void HienThiDuLieu()
        {
            //Truy vấn dữ liệu sử dụng LINQ
            var query = from lsp in db.LoaiSanPhams
                        select lsp;
            //Hiển thị dữ liệu lên data grid
            dgvLoaiSanPham.ItemsSource = query.ToList();
        }
    }
}

```

```

private void btnThem_Click(object sender, RoutedEventArgs e)
{
    //Tạo đối tượng mới của lớp Loại sản phẩm
    LoaiSanPham lspMoi = new LoaiSanPham();
    //Thuộc tính của đối tượng loại sản phẩm mới do user nhập vào
    lspMoi.MaLoai = txtMa.Text;
    lspMoi.TenLoai = txtTen.Text;
    //nếu chưa có loại sản phẩm thì mới thêm, ngược lại thông báo lỗi
    if (!db.LoaiSanPhams.Contains(lspMoi))
    {
    }
}

```

```

//Thêm đối tượng vào tập hợp loại sản phẩm
db.LoaiSanPhams.Add(lspMoi);
//Lưu thay đổi vào cơ sở dữ liệu
db.SaveChanges();
//Hiển thị lại dữ liệu sau khi thêm
HienThiDuLieu();
}
else
{
    MessageBox.Show("Đã có sản phẩm " + txtMa.Text, "THÊM DỮ LIỆU",
        MessageBoxButton.OK, MessageBoxImage.Error);
}
}

private void btnSua_Click(object sender, RoutedEventArgs e)
{
    //Lấy ra đối tượng loại sản muốn sửa
    var lspSua = (from lsp in db.LoaiSanPhams
        where lsp.MaLoai == txtMa.Text
        select lsp).SingleOrDefault();
    //Nếu có sản phẩm muốn sửa thì thực hiện cập nhật thông tin,
    //ngược lại hiển thị thông báo
    if (lspSua != null)
    {
        //sửa thông tin sản phẩm -- không sửa mã
        lspSua.TenLoai = txtTen.Text;
        //lưu thay đổi vào cơ sở dữ liệu
        db.SaveChanges();
        //Hiển thị lại dữ liệu sau khi sửa
        HienThiDuLieu();
    }
    else
    {
        MessageBox.Show("Không có sản phẩm mã " + txtMa.Text, "SỬA DỮ LIỆU",
            MessageBoxButton.OK, MessageBoxImage.Information);
    }
}
}

```

```

private void btnXoa_Click(object sender, RoutedEventArgs e)
{
    //Lấy ra đối tượng loại sản muốn xóa
    var lspXoa = (from lsp in db.LoaiSanPhams
        where lsp.MaLoai == txtMa.Text
        select lsp).SingleOrDefault();
    //Nếu có sản phẩm muốn xóa thì thực hiện xóa, ngược lại hiển thị thông báo
    if (lspXoa != null)
    {
        //xóa sản phẩm
        db.LoaiSanPhams.Remove(lspXoa);
        //lưu thay đổi vào cơ sở dữ liệu
        db.SaveChanges();
        //Hiển thị lại dữ liệu sau khi xóa
        HienThiDuLieu();
    }
    else
    {
        MessageBox.Show("Không có sản phẩm mã " + txtMa.Text, "XÓA DỮ LIỆU",
            MessageBoxButton.OK, MessageBoxImage.Information);
    }
}
}
}

```

*Sửa lại định dạng dữ liệu cho DataGridView:

```
<DataGridView Name="dgvLoaiSanPham" HorizontalAlignment="Left" Height="180" Margin="74,136,0,0"
VerticalAlignment="Top" Width="510"
    AutoGenerateColumns="False"
    AlternatingRowBackground="LightBlue"
    CanUserAddRows="False" >
    <DataGridView.Columns>
        <DataGridViewTextColumn Header="Mã Loại" Binding="{Binding MaLoai}" />
        <DataGridViewTextColumn Header="Tên loại sản phẩm" Binding="{Binding TenLoai}" Width="*"/>
    </DataGridView.Columns>
</DataGridView>
```

4.2.5. Truy vấn, cập nhật dữ liệu trên bảng sản phẩm

Mã SP	Tên SP	Mã loại SP	Số lượng	Đơn giá	Thành tiền
01	Áo sơ mi nam	01	12	500	6000
02	Quần bò nữ	01	200	600	120000
04	Lò vi sóng	02	400	700	280000
03	Nồi cơm điện	02	100	1000	100000

Yêu cầu:

- Hiện thị các **SanPham** trong DataGridView. Thông tin gồm: mã sản phẩm, tên sản phẩm, mã loại sản phẩm, số lượng, đơn giá, thành tiền. Dữ liệu được sắp xếp theo đơn giá tăng dần.
- Sử dụng **combo box** hiển thị tên loại hàng từ bảng **LoaiSanPham** nhưng khi chọn lấy mã loại để cập nhật dữ liệu.
- Thêm 1 bản ghi vào bảng **SanPham** khi nhấn nút **Thêm**, dữ liệu do người dùng nhập qua các điều khiển nhập liệu.
 - Kiểm tra số lượng và đơn giá phải là số nguyên và > 0 .
 - Hiện thị lại dữ liệu sau khi thêm trong DataGridView.
- Sửa thông tin sản phẩm được chọn trong danh sách khi nhấn nút **Sửa**, dữ liệu do người dùng nhập qua các điều khiển nhập liệu, không sửa mã sản phẩm. Lưu thông tin đã sửa vào csdl và hiện thị lại dữ liệu trong DataGridView.

- e) Xóa sản phẩm được chọn trong danh sách khi nhấn nút **Xóa**. Hiện thị thông báo xác nhận trước khi xóa.(tự làm)
- f) Chọn 1 loại sản phẩm trong **ComboBox**, sau đó nhấn nút **Tìm**: Hiện thị thông tin của các sản phẩm tìm được lên DataGridView, gồm các cột: mã sản phẩm, tên sản phẩm, mã loại sản phẩm, số lượng, đơn giá, thành tiền. (tự làm)

*** Code XAML xây dựng giao diện:**

```

...
<DataGrid x:Name="dgvSanPham" AutoGenerateColumns="False" CanUserAddRows="False"
          AlternatingRowBackground="AntiqueWhite" >
    <DataGrid.Columns>
        <DataGridTextColumn Header="Mã SP" Binding="{Binding MaSp}" />
        <DataGridTextColumn Header="Tên SP" Binding="{Binding TenSp}" />
        <DataGridTextColumn Header="Mã loại SP" Binding="{Binding MaLoai}" />
        <DataGridTextColumn Header="Số lượng" Binding="{Binding SoLuong}" />
        <DataGridTextColumn Header="Đơn giá" Binding="{Binding DonGia}" />
        <DataGridTextColumn Header="Thành tiền" Binding="{Binding ThanhTien}" Width="*" />
    </DataGrid.Columns>
</DataGrid>
...

```

*** Code C# xử lý:**

using DemoEFCore.Models;

```

namespace DemoEFCore
{
    public partial class CapNhat_SanPham : Window
    {
        public CapNhat_SanPham()
        {
            InitializeComponent();
        }
        QLBanHangContext db = new QLBanHangContext();

        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            HienThiDuLieu();
            var querycombo = from lsp in db.LoaiSanPhams
                             select lsp;
            cboLoai.ItemsSource = querycombo.ToList();
            cboLoai.DisplayMemberPath="TenLoai";
            cboLoai.SelectedValue="MaLoai";
        }

        private void HienThiDuLieu()
        {
            var query = from sp in db.SanPhams
                        orderby sp.DonGia
                        select new
                        {
                            sp.MaSp,
                            sp.TenSp,
                            sp.MaLoai,
                            sp.SoLuong,
                            sp.DonGia,
                            ThanhTien = sp.SoLuong * sp.DonGia
                        };
            dgvSanPham.ItemsSource = query.ToList();
        }
    }
}

```

```

private void btnThem_Click(object sender, RoutedEventArgs e)
{
    try
    {
        int donGia = int.Parse(txtDonGia.Text);
        if (donGia < 0)
        {
            MessageBox.Show("Đơn giá là số nguyên> 0.");
            return;
        }
        try
        {
            int soLuong = int.Parse(txtSoLuong.Text);
            if (soLuong < 0)
            {
                MessageBox.Show("Số lượng là số nguyên> 0.");
                return;
            }
            SanPham spMoi = new SanPham();
            spMoi.MaSp = txtMa.Text;
            spMoi.TenSp = txtTen.Text;
            spMoi.DonGia = donGia;
            spMoi.SoLuong = soLuong;
            LoaiSanPham itemSelected = (LoaiSanPham)cboLoai.SelectedItem;
            spMoi.MaLoai = itemSelected.MaLoai;
            db.SanPhams.Add(spMoi);
            db.SaveChanges();
            HienThiDuLieu();
        }
        catch (FormatException)
        {
            MessageBox.Show("Số lượng phải là số nguyên> 0.");
        }
        catch (Exception ex)
        {
            throw;
        }
    }
    catch (FormatException)
    {
        MessageBox.Show("Đơn giá là số nguyên> 0.");
    }
    catch (Exception ex)
    {
        throw;
    }
}

```

```

private void btnSua_Click(object sender, RoutedEventArgs e)
{
    //Lấy ra đối tượng sản phẩm muốn sửa
    var spSua = (from sp in db.SanPhams
                 where sp.MaSp == txtMa.Text
                 select sp).SingleOrDefault();
    //Nếu có sản phẩm muốn sửa thì thực hiện cập nhật thông tin,
    //ngược lại hiển thị thông báo
    if (spSua != null)
    {
        //sửa thông tin sản phẩm -- không sửa mã
        spSua.TenSp = txtTen.Text;
        LoaiSanPham itemSelected = (LoaiSanPham)cboLoai.SelectedItem;
        spSua.MaLoai = itemSelected.MaLoai;
        spSua.DonGia = int.Parse(txtDonGia.Text);
        spSua.SoLuong = int.Parse(txtSoLuong.Text);

        //lưu thay đổi vào cơ sở dữ liệu
        db.SaveChanges();
        //Hiển thị lại dữ liệu sau khi sửa
        HienThiDuLieu();
    }
    else
    {
        MessageBox.Show("Không có sản phẩm mã " + txtMa.Text, "SỬA DỮ LIỆU",
            MessageBoxButton.OK, MessageBoxImage.Information);
    }
}

```

```

private void btnXoa_Click(object sender, RoutedEventArgs e)
{
    //Lấy ra đối tượng loại sản muốn xóa
    var spXoa = (from sp in db.SanPhams
                 where sp.MaSp == txtMa.Text
                 select sp).SingleOrDefault();
    //Nếu có sản phẩm muốn xóa thì thực hiện xóa, ngược lại hiển
    //thị thông báo
    if (spXoa != null)
    {
        //xóa sản phẩm
        db.SanPhams.Remove(spXoa);
        //lưu thay đổi vào cơ sở dữ liệu
        db.SaveChanges();
        //Hiển thị lại dữ liệu sau khi xóa
        HienThiDuLieu();
    }
    else
    {
        MessageBox.Show("Không có sản phẩm mã " + txtMa.Text, "XÓA DỮ LIỆU",
            MessageBoxButton.OK, MessageBoxImage.Information);
    }
}
}

```