

Lab 3: LINQ-ADO.NET-Entity Framework Core

Phần 1: LINQ

Bài 1: Cho mảng chuỗi tên các nước như sau:

```
string[] countries = { "Korea", "Japan", "USA", "Jordan", "Jamaica", "China" };
```

Cho chương trình sau:

```
using System;
using System.Collections.Generic;
using System.Linq;

namespace Lab3
{
    class Program
    {
        static void Main()
        {
            string[] countries = { "Korea", "Japan", "USA", "Jordan", "Jamaica", "China" };
            IEnumerable<string> result = from x in countries
                                       where x.StartsWith("J")
                                       select x;

            foreach (var country in result)
            {
                Console.WriteLine(country);
            }
        }
    }
}
```

Tham khảo chương trình trên, viết chương trình thực hiện yêu cầu xuất ra danh sách các nước bắt đầu bằng ký tự "J" và có ký tự kết thúc là ký tự "n".

Bài 2: LINQ Group By

Ví dụ: LINQ Group By

```
IList<Student> studentList = new List<Student>() {
    new Student() { StudentID = 1, StudentName = "John", Age = 18 },
    new Student() { StudentID = 2, StudentName = "Steve", Age = 21 },
    new Student() { StudentID = 3, StudentName = "Bill", Age = 18 },
    new Student() { StudentID = 4, StudentName = "Ram", Age = 20 },
    new Student() { StudentID = 5, StudentName = "Abram", Age = 21 }
};

var groupedResult = from s in studentList
                    group s by s.Age;

//iterate each group
foreach (var ageGroup in groupedResult)
{
    Console.WriteLine("Age Group: {0}", ageGroup.Key); //Each group has a key

    foreach(Student s in ageGroup) // Each group has inner collection
        Console.WriteLine("Student Name: {0}", s.StudentName);
}
```

Kết quả:

Output:

```
AgeGroup: 18
StudentName: John
StudentName: Bill
AgeGroup: 21
StudentName: Steve
StudentName: Abram
AgeGroup: 20
StudentName: Ram
```

Bài tập :

Hoàn thành đoạn chương trình sau:

```
namespace Lab3
{
    class Program
    {
        static void Main()
        {
            List<Student> objStudent = new List<Student>()
            {
                new Student() { Name = "Tom", Gender = "Male", Location = "NewYork" },
                new Student() { Name = "Alice", Gender = "Female", Location = "Hawai" },
                new Student() { Name = "Bob", Gender = "Male", Location = "NewYork" },
                new Student() { Name = "Onana", Gender = "Male", Location = "California"},
                new Student() { Name = "David", Gender = "Male", Location = "Arizona"},
                new Student() { Name = "Andrew", Gender = "Male", Location = "Hawai" },
            };
            //Write your code here
        }
    }
    class Student
    {
        public string Name { get; set; }
        public string Gender { get; set; }
        public string Location { get; set; }
    }
}
```

Sử dụng LINQ Group By nhóm dữ liệu theo các "Location" để được kết quả như sau:

```


Select Microsoft Visual Studio Debug Console
Tom      NewYork
Bob      NewYork
Alice    Hawai
Andrew   Hawai
Onana    California
David    Arizona
```

Phần 2: ADO.NET

| Objects | Công dụng |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Connection | Thiết lập kết nối tới CSDL |
| Command | Thực hiện các câu lệnh SQL |
| DataReader | Nhận kết quả trả về khi thực hiện command có chứa câu lệnh Select. Cho phép duyệt qua các mẫu tin nhanh(duyet một chiều, không cho phép cập nhật dữ liệu) |

Tạo database và thêm vào dữ liệu như sau:

CUSTOMER DATABASE

| 603-GV\SQLEXPRESS...er - dbo.Customer | | | |
|-----------------------------------------------------------------------------------|--------------|---------------|--------------------------|
| | Column Name | Data Type | Allow Nulls |
|  | CustomerID | int | <input type="checkbox"/> |
| | CustomerName | nvarchar(100) | <input type="checkbox"/> |
| | Address | nvarchar(200) | <input type="checkbox"/> |
| | Amount | int | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |

Thêm các dữ liệu sau vào database vừa tạo:

| 603-GV\SQLEXPRESS...er - dbo.Customer | | | | |
|---------------------------------------|------------|--------------|------------|--------|
| | CustomerID | CustomerName | Address | Amount |
| | 1 | David | Texas | 1500 |
| | 2 | Tony | NewYork | 3000 |
| | 3 | Alice | LA | 5000 |
| | 4 | Tom | California | 2500 |

Sử dụng ADO.NET xây dựng chương trình kết nối và hiển thị dữ liệu khách hàng lên DataGridView trên ứng dụng Winform.

Hướng dẫn tạo kết nối

Chuỗi kết nối:

```
string conectionString = @"Data Source=YourMachineName; Integrated Security=true;
Database=Customer";
```

Thêm Class Customer có các property tương ứng với thiết kế của Database.

```
public class Customer
{
    public int CustomerID {get; set;}

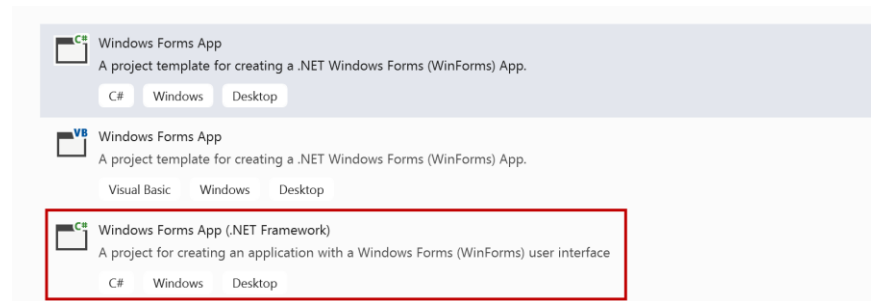
    .....
}
```

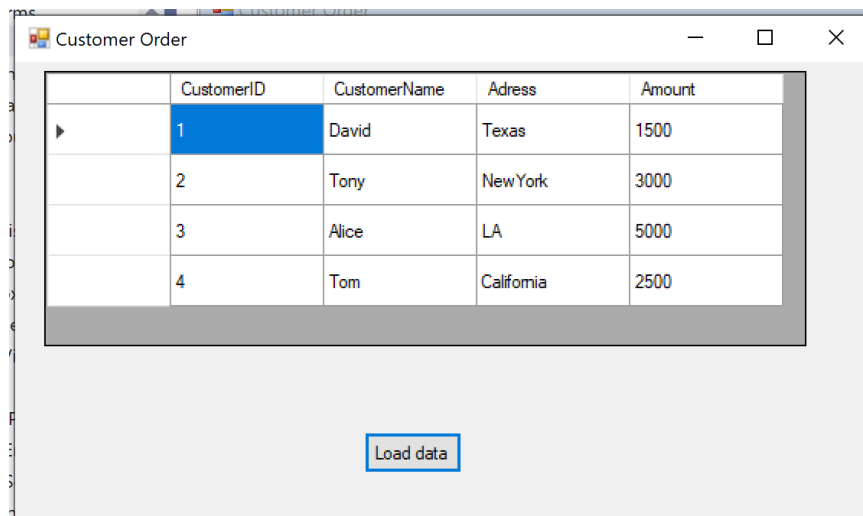
Ví dụ lệnh Select dữ liệu sử dụng ADO.NET:

```
public List<Customer> GetAllCustomer()
{
    List<Customer> result = new List<Customer>();
    using (SqlConnection connection = new SqlConnection(conectionString))
    {
        SqlCommand sqlCommand = new SqlCommand("Select * from [Customer]", connection);
        connection.Open();
        SqlDataReader reader = sqlCommand.ExecuteReader();
        while (reader.Read())
        {
            Customer customer = new Customer();
            customer.CustomerID = reader.GetInt32(0);
            customer.CustomerName = reader.GetString(1);
            customer.Address = reader.GetString(2);
            customer.Amount = reader.GetInt32(3);

            result.Add(customer);
        }
        return result;
    }
}
```

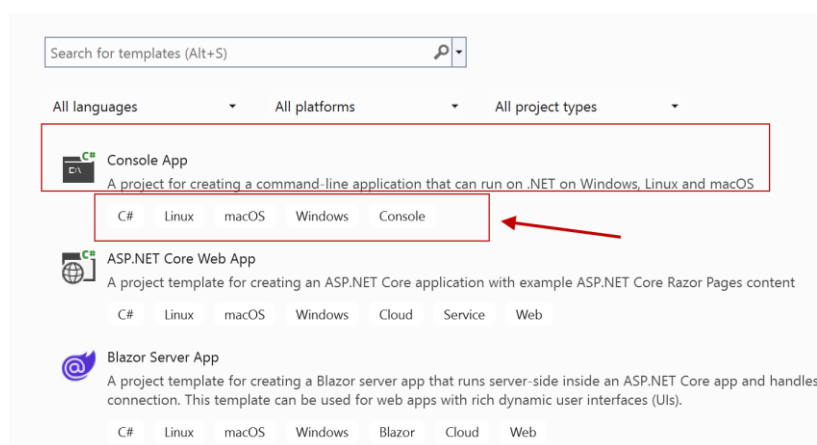
Tạo project **Winform**. Khi người dùng bấm vào nút “Load data” sử dụng đối tượng **DataGridView** để hiển thị dữ liệu đọc được từ database.





Phần 3: Entity Framework Core

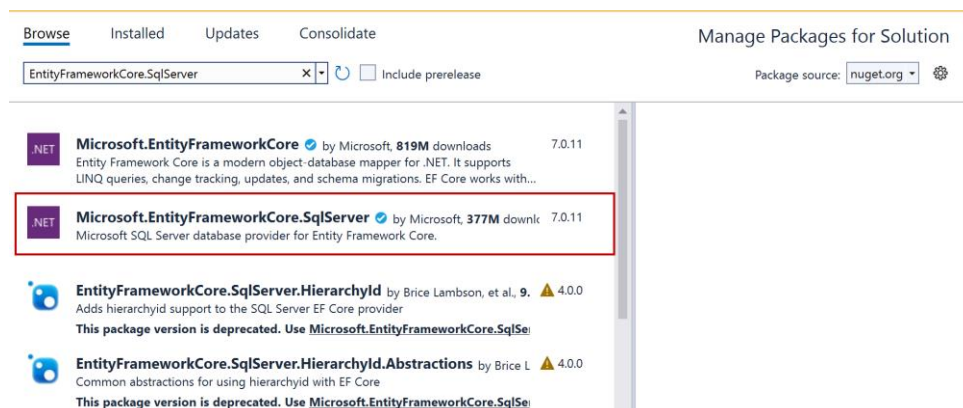
Tạo project **Console App (NET Core)** trên Visual Studio:

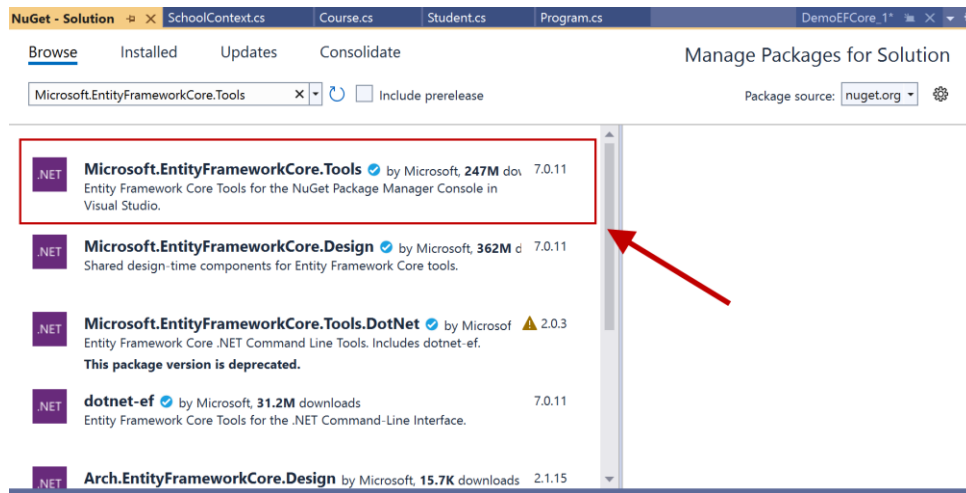


Tiến hành cài đặt các package sau:

Microsoft.EntityFrameworkCore.SqlServer

Microsoft.EntityFrameworkCore.Tools





Tạo mới hai Class sau:

Class Book

```
public class Book
{
    public string BookId { get; set; }
    public string Title { get; set; }
    public string Author { get; set; }
    public string Language { get; set; }
    public int Pages { get; set; }
}
```

Class Publisher

```
public class Publisher
{
    public int ID { get; set; }
    public string Name { get; set; }
}
```

Sử dụng Entity Framework Core – Code First để tạo hai table trong Database từ hai Class trên.

Gợi ý:

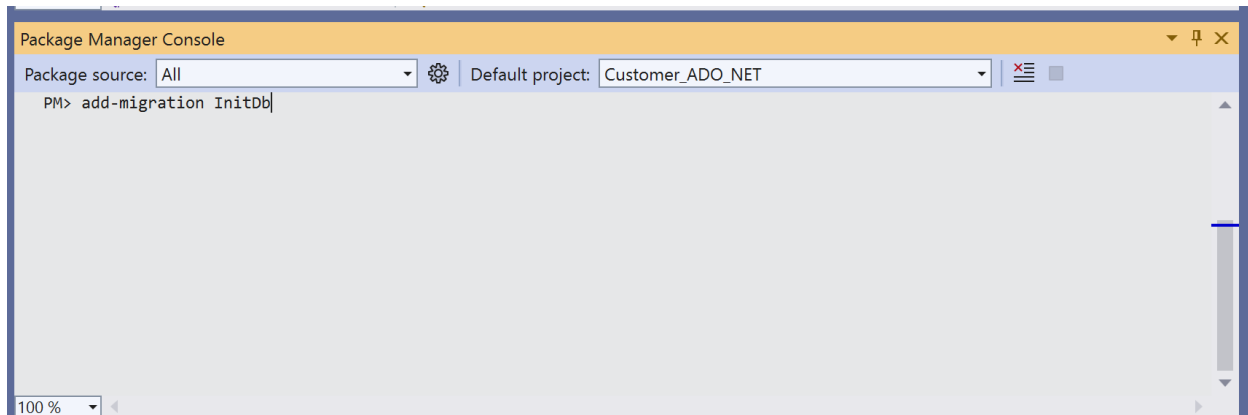
Sau khi tạo các class Book và Publisher, tạo Class YourDbContext như sau:

```
public class YourDbContext : DbContext
{
    public DbSet<Book> Book { get; set; }

    public DbSet<Publisher> Publisher { get; set; }

    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
        optionsBuilder.UseSqlServer(@"Server=.; Database=YourDB; Trusted_Connection=True; TrustServerCertificate=True");
    }
}
```

Mở Package Manager Console chạy các lệnh sau:



add-migration InitDb

update-database

- 1) Kiểm tra Database đã tạo thành công đúng các Table như thiết lập. Chụp màn hình Design của các table vừa mới tạo.
- 2) Thay đổi property Pages của Class Books thành NumberofPages. Sau đó chạy lại **add-migration** và **update-database**. Xem và chụp lại màn hình Design của Table Books.

Lưu các hình chụp và nộp kèm Source Code các câu trên.