C# ASP.NET web forms For Beginners Summery By Mahmoud Badawy

-----------------------------------------------------

متطلبات الكورس انك تبقى فاهم .Net و C#

MB

################################################################

What is LINQ

-----------------

بص يا سيدى دى تقنيه جديده معموله للدوت نت بس بتقدر بيها تكلم الداتا بيز من اى نوع فمثلا تقدر تكلم xml او السكوال العادى او انك تكلم مثلا ال Entity بتاعت الانتيتى فريم ورك او حتى تكلم اى DataSet او اوبجكيت فى اى حته او اراى حتى …. اهه بقه بكود واحد لينك تقدر تكلم كل دول بدل ما تعد تكتب كود لكل واحده فيهم ههههههه



مثال بقه اكلم فيه مثلا اوبجيكت من كلاس عادى خالص

using System;

using System.Collections.Generic;

**using System.Linq;**

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

string[] Persons = { "Mahmoud", "Mostafa","Karim","Samir","Kalid","Salem","Rami","Samer" };

IEnumerable<string> MyPersonsList = Persons.**Where(Per => (Per.StartsWith("S")));**

foreach (string MyPerList in MyPersonsList)

{

Console.WriteLine("The Names that Starts With 'S' are : " + MyPerList);

}

int[] Numbers = { 17,58,32,11,13,2,5,8,7,10,1,3,4,5};

IEnumerable<int> MyNumbers = Numbers.Where( a => (a % 2 == 0));

foreach(int MyNum in MyNumbers)

{

Console.WriteLine("The Even Numbers are : " + MyNum);

}

}

}

}

بص ياعم دلوقتى انت عندك اراى و انا عاوز اجيب كل الاسامى الى بتبدا فيها بحرف S تجيبها اذاى دى ؟؟

لو هتجيبها بالطريقه العاديه هتعمل اتيريشن و تعد تلف على واحده واحده تعمل تشيك عليهم و تخزنهم و بعد ما تخزنهم هتعمل اتريشن على الى انت خزنته عشان تظهره ملحوظه اتريشن يعنى لوب طيب و على ايه كل ده ما احنا عندنا اللينك يحللنا الكلام ده فى لحظات

كل الى بتعمله انك بتشوف انت هترجع ايه انت مصلا فى الحاله بتعتنا دى هترجع سترينجات فبتقوله اعملى فاريبول من نوع

IEnumerable<string> MyPersonsList اينمرابول اوف سترينج عشان انت هترجع عدد مش هترجع واحد بس

بعد كده بتقوله

Persons.**Where(Per => (Per.StartsWith("S")));**

الى هى معناها ايه ؟؟ معناها ياعم انى بقوله روح على الاراى الى اسمه persons تمام نفذى عليه Extention Function اسمها Where و الى بتحتاج جواها Lambda Expression و الى هى فيها بقوله رجعلى الاسترينج فى ال Per لما تعمل تشيك على كل بير بير و تشوف كل البيرز الى بتبدا ب S  
عشان تفهم الحكايه بتمشى اذاى فى اللمبا اكسيبرشن بص هو الاول بيخش فى القوس ده **(Per.StartsWith("S")**

بيعتبر ان ال Per الى جوه القوس الى بعد السهم دى يعنى من الاخر كل الى موجود فى الاراى و هينفذ عليها الفلتر بتاع الوير فى السكوال الى هو يرجع ترو لما يلاقى اول حرف ب S و فولس لو اول حرف مش ب S و بدل بقه ما هياخد كله لأ هياخد الى رجع ترو بس طب و ده بيتم اذاى بص يا سيدى شايف ال Per الى قبل السهم ؟؟

اهو ال Per الى قبل السهم دى كل اما الكونديشن يرجع ترو تروح ال per الى قبل السهم مسجله بياناتها بس كده و بعد ما يخلص فلتره و تسجيل البيانات بيروح واحد كل الكلام ده و حطه فى الفاريبول بقه الى هى MyPersonsList

و عشان يشوف الكلام الى رجع ده بيروح عامل اتريشن على الى رجع

foreach (string MyPerList in MyPersonsList)

{

Console.WriteLine("The Names that Starts With 'S' are : " + MyPerList);

}

الفايده من الكلام ده ان داخل وير لازم يبقى فى اوت بوت احط فيه النواتج و ده بيبقى على الشمال و ان بوت و ده عباره عن بولين الى هو بيرجع ترو او فولس و ده بينتج من كونديشن من الاخر

* مثال تانى المره دى عاوز اجيب بيانات من كلاس او اوبجيكت خارجى

using System;

using System.Collections.Generic;

using System.Linq;

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

student StudentObj = new student();

List<student> AllStudents = StudentObj.GetAllStudents();

IEnumerable<student> MaleStudents = AllStudents.Where(Stud => (Stud.Gender == "Male"));

foreach (student item in MaleStudents)

{

Console.WriteLine("Student : {0} and his ID : {1} and It's Gender : {2} "

, item.Name , item.ID,item.Gender );

}

}

}

public class student {

public int ID { get; set; }

public string Name { get; set; }

public string Gender { get; set; }

public List<student> GetAllStudents () {

List<student> Studentlist = new List<student>();

student student\_1 = new student { ID = 1, Name = "Mahmoud Badawy", Gender = "Male" };

student student\_2 = new student { ID = 2, Name = "Mostafa Mahmoud", Gender = "Male" };

student student\_3 = new student { ID = 3, Name = "Noha Saber", Gender = "Female" };

student student\_4 = new student { ID = 4, Name = "Mahmoud Elkerdasy", Gender = "Male" };

student student\_5 = new student { ID = 5, Name = "Nadia Badawy", Gender = "Female" };

Studentlist.Add(student\_1);

Studentlist.Add(student\_2);

Studentlist.Add(student\_3);

Studentlist.Add(student\_4);

Studentlist.Add(student\_5);

return Studentlist;

}

}

}

ذى ما انت شايف كلاس الاستودينت جواه الاى دى و الاسم و النوع كل الى عملته انى عملت لسته جوه الكلاس من نوع ستيودينت عشان اقدر املها ب Objects يكون فيها الاسم و الاى دى و النوع و بعد كده ندهت على الفانكشن دى فى ال main فانكشن و رجعت لسته الطلاب و بعد كده مسكت لسته الطلاب دى اعلم عليها شغل اللينك و الى هو انا هرجع بس الطلبه الى من نوع ذكر

MB

################################################################

Extention Methods

---------------------------

شايف الفانكشنز الخاصه ب Linq دى و الفانكشنز الى بتكون خاصه باى نوع معين من الداتا و من الاخر الى بتتكتب على شكل سلسله او chain الى هى كده Str.toString() او Stud.Whrer() اهه الى بتتكتب قبلها دوت دى ذى اى Extention بتتسمى ال Extention Method

طب بعملها اذاى بص ياحج كل الى بتعمله انك بتعمل Class معين ده عشان تعين فيه كل ال Extention func بتعتك و بيبقى من نوع static و بتخلى الفانكشن برضوا static و اى براميتر الفانكشن بتخده بقه بتكتب قبل النوع بتاعه this من الاخر ذى الى فى الكود ده كده بالظبط

using System;

using System.Collections.Generic;

using System.Linq;

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

string MyName = "Mahmoud Badawy";

int x = MyName.MyNameLenght();

// OR

int Y = HelperClass.MyNameLenght(MyName);

Console.WriteLine("My name Lenght is : " + x + " Char");

Console.WriteLine("My name Lenght is : " + Y + " Char");

}

}

public static class HelperClass

{

public static int MyNameLenght(this string Param )

{

return Param.Length;

}

}

}

عشان تفهم ال LINQ اتعمل اذاى بقه ابقى خش اتفرج على الكلاس بقه و شوف معمول اذاى هتفهم فشخ

MB

################################################################

Aggregate Functions

---------------------------

دى الى هى من الاخر Max , Min , Average , Count , Sum  
طبعا مش محتاجين شرح بص انا كل الى عاوزه منك بقه هاتلى

اصغر عدد زوجى

و اكبر عدد فردى

و الافريدج بتاع الاعداد الزوجيه

و عدد الاعداد الفرديه

و مجموع الاعداد الزوجيه و اول رقم فى الاعداد الفرديه

using System;

using System.Collections.Generic;

using System.Linq;

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

int[] Numbers = { 2, 14, 54, 47, 25, 30, 11, 23, 25, 24, 228, 77, 33, 10, 1, 3 };

int Min\_Even\_num = Numbers.Where(x => (x % 2 == 0)).Min(); //2

Console.WriteLine(" Min Even Num is : "+ Min\_Even\_num);

int Max\_odd\_numbers = Numbers.Where(x => (x % 2 != 0)).Max(); //77

Console.WriteLine(" Max odd numbers is : " + Max\_odd\_numbers);

double Avg\_Even\_Num = Numbers.Where(x => (x % 2 == 0)).Average();

Console.WriteLine(" Avg Even numbers is : " + Avg\_Even\_Num); //51.71

int Count\_Odd\_Num = Numbers.Where(x => (x % 2 != 0)).Count();

Console.WriteLine(" Count Odd numbers is : " + Count\_Odd\_Num); //9

int Sum\_Even\_Num\_FirstOdd =

Numbers.Where(x => (x % 2 == 0)).Sum() + Numbers.Where(x => (x % 2 != 0)).First();

Console.WriteLine(" Sum Even Num and First Odd is : " + Sum\_Even\_Num\_FirstOdd); //9

}

}

}

طيب عاوز بقه اجيب من وسط البلد اكبر اسم بلد و اصغر اسم بلد برضوا هيبقى عدد حروفهم كام

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

string[] Countries = { "Egypt", "USA", "France", "Saudi Arabia" };

int MinCountryLength = Countries.Min(x=>x.Length);

// Console.WriteLine("Min Country Name Length : " + MinCountryLength);

IEnumerable<string> MinCountryName = Countries.Where(x => (x.Length == MinCountryLength));

foreach (var item in MinCountryName)

{

Console.WriteLine("Min Country Name : " + item + " and it's Lenght is : " + item.Length);

}

int MaxCountryLength = Countries.Max(x => x.Length);

// Console.WriteLine("Max Country Name Length: " + MaxCountryLength);

IEnumerable<string> MaxnCountryName = Countries.Where(x => (x.Length == MaxCountryLength));

foreach (var item in MaxnCountryName)

{

Console.WriteLine("Max Country Name : " + item + " and it's Lenght is : "+ item.Length);

}

}

}

}

MB

################################################################

Aggregate

---------------------------

بص يا سيدى دى فانكشن كده معموله عشان لو عندك حاجات عاوز تجمعها على بعض و تطلعها فى حاجه واحده يعنى و ليكن انت عندك Array فيها اسامى دول و انت عاوز تطلع الاسامى دى كلها جنب بعض تعملها اذاى ؟؟

public static void Main(string[] args)

{

string[] Countries = { "Egypt", "USA", "France", "Saudi Arabia" };

//Aggregate Function Take in it function That take 2 param and Return 1 Value

string CountriesList = Countries.Aggregate((a, b) =>( a + " , "+ b ));

Console.WriteLine(CountriesList);

}

ذى ما انت شايف الاجرجيت بتتعمل على ايه على المدن فالى بيحصل ان جوه الاجرجيت فانكشن بتجمع بين المدينه الحاليه و المدن السابقه   
المدينه الحاليه هى ال B و المدن السابقه تخزينها هى A فالى بيحصل بالظبط ان اولا بتعدى على مصر تحطها فى B بعد كده تاخدها تخزنها فى A ثم تروح على أمريكا تاخدها تحطها فى B بعد كده تضيفها على A فتبقى A جواها مصر و امريكا و هكذا مع باقى الاراى بقه

تعالى مثال تانى انا عاوز اجمع كل الارقام الزوجيه مع بعض بقه

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

int[] Numbers = { 1, 5, 4, 7, 12, 15, 5, 9, 10, 8, 2, 11, 13, 3 };

int EvenNumSum = Numbers.Where((z)=>(z % 2 == 0)).Aggregate((a, b) => (a + b));

Console.WriteLine(EvenNumSum); // 36

}

}

}

لو عاوز اعمل sead او بدايه للاجرجيت اعمله اذاى ده ؟ مجرد تغيير بسيط قبل قوس (a,b)

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

int[] Numbers = { 1, 5, 4, 7, 12, 15, 5, 9, 10, 8, 2, 11, 13, 3 };

int EvenNumSum = Numbers.Where((z)=>(z % 2== 0)).Aggregate(10,(a, b) => (a+b));

Console.WriteLine(EvenNumSum); // 46

}

}

}

MB

################################################################

Where

---------------------------

بيتم استخدام الوير عشان نعمل filter على النتايج يعنى مثلا معايا الارقام ذى ما انت شايف و انا عاوز اجيب الارقام الزوجيه بس من وسط كل الارقام ساعتها بتجيبها كده ذى ما انت شايف فى الكود

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

int[] Numbers = { 1, 5, 4, 7, 12, 15, 5, 9, 10, 8, 2, 11, 13, 3 };

IEnumerable<int> EvenNums = Numbers.Where((z)=>(z % 2 == 0));

foreach (var EvenNum in EvenNums)

{

Console.WriteLine(EvenNum);

}

}

}

}

ذى ما انت شايف داخل الوير موجود فانكشن بتاخد الرقم و تشوف نتيجه القسمه على 2 لو بصفر ترجع true و ترو هنا معناها حط العنصر الى اتعمل عليه الاتشيك ده جوه ال z و لو ب false تجاهله و فكك منه فبالتالى هيمشى على رقم رقم و هيشوف لو زوجى يخزنه و لو فردى هيتجاهله

MB

################################################################

Select

-----------------

select برجع النتايج بالشكل الى انا عاوزه بالظبط عن طريق انى بوريه اوبجيكت يجيب شبهه :D :D

فى الحاله دى انا عاوز اجيب الرقم و الانديكس بتاعه فى الاراى

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

int[] Numbers = { 1, 5, 4, 7, 12, 15, 5, 9, 10, 8, 2, 11, 13, 3 };

var EvenNums = Numbers.Select((Num, index) => new { Number = Num, Index =index });

foreach (var EvenNum in EvenNums)

{

Console.WriteLine( "Number : "+EvenNum.Number + " --> Index : " + EvenNum.Index);

}

}

}

}

مثال متقدم على ال Select اولا هعمل كلاس للموظفين و عاوز بقه الموظفين و مرتباتهم و كل بياناتهم

هنا و ليكن عاوز ارجع اساميهم بس

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

Employee EmpObj = new Employee();

List<Employee> Employees = EmpObj.getAllEmployee();

IEnumerable<string> Emps= Employees.Select(Emp => Emp.FName + " " + Emp.LName );

foreach (var EMP in Emps)

{

Console.WriteLine(EMP);

}

}

}

class Employee

{

public int EmplyeeID { get; set; }

public string FName { get; set; }

public string LName { get; set; }

public string Gender { get; set; }

public int AnnualSalary { get; set; }

public List<Employee> getAllEmployee()

{

List<Employee> ListEmployee = new List<Employee>

{

new Employee { EmplyeeID=1,FName="Mahmoud",LName="Badawy",Gender="Male",AnnualSalary= 100000},

new Employee { EmplyeeID=2,FName="Mostafa",LName="Badawy",Gender="Male",AnnualSalary= 110000},

new Employee { EmplyeeID=3,FName="Ahmed",LName="Badawy",Gender="Male",AnnualSalary= 650000},

new Employee { EmplyeeID=4,FName="Nouran",LName="Badawy",Gender="Female",AnnualSalary= 750000},

new Employee { EmplyeeID=5,FName="Samy",LName="Badawy",Gender="Male",AnnualSalary= 150000},

new Employee { EmplyeeID=6,FName="Nadia",LName="Badawy",Gender="Female",AnnualSalary= 300000},

};

return ListEmployee;

}

}

}

طيب هاتلى اساميهم و مرتباتهم و بونص 10% على مرتباتهم لو مرتبهم الشهرى اكتر من 50 الف عارف تعملها اذاى دى ؟

ايوه اعمل شكلهم بمزاجى عن طريق انى بعمل الشكل بشكل اوبجيكت انا بحدله شكله اهه الكود عشان تفهم

من الاخر هنستخدم ال Anonymous Object

public static void Main(string[] args)

{

Employee EmpObj = new Employee();

List<Employee> Employees = EmpObj.getAllEmployee();

var Emps = Employees.Select(Emp => (

new {

FullName = Emp.FName + " " + Emp.LName,

salary = Emp.AnnualSalary / 12,

Bonus = Emp.AnnualSalary / 12 \* 0.1

}

)).Where(A=>A.salary>50000);

foreach (var EMP in Emps)

{

Console.WriteLine("Name : " + EMP.FullName + " --> Salary : " + EMP.salary +

" --> Bonus :" + EMP.Bonus );

}

}

}

اظن كده انت فهمت ان ال select مهمتها فى الحياه شكل الحاجه الى راجعه العب فيها ذى ما انا عاوز بينما ال where بتعمل فلتر للنتايج و الاجرجيت عشان اعمل جمع ضرب او كونكاتيونيشن

ده مثال اخر للسته جواه اوبجيكت من نوع تانى

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

Employee EmpObj = new Employee();

List<Employee> Employees = EmpObj.getAllEmployee();

var Emps = Employees.Select(Emp => (

new {

ID = Emp.EmplyeeID,

FullName = Emp.FName + " " + Emp.LName,

DeptName = Emp.Department.DeptName

}

)).Where(A=>A.DeptName == "IT");

foreach (var EMP in Emps)

{

Console.WriteLine("Name : " + EMP.FullName + " --> Depart : " + EMP.DeptName );

}

}

}

class Department

{

public int DeptID { get; set; }

public string DeptName { get; set; }

}

class Employee

{

public int EmplyeeID { get; set; }

public string FName { get; set; }

public string LName { get; set; }

public Department Department { get; set; }

public List<Employee> getAllEmployee()

{

List<Employee> ListEmployee = new List<Employee>

{

new Employee { EmplyeeID=1,FName="Mahmoud",LName="Badawy",Department = new Department { DeptID=1 , DeptName = "IT" } },

new Employee { EmplyeeID=2,FName="Mostafa",LName="Badawy" ,Department = new Department { DeptID=2 , DeptName = "SWE" }},

new Employee { EmplyeeID=3,FName="Ahmed",LName="Badawy" ,Department = new Department { DeptID=2 , DeptName = "SWE" }},

new Employee { EmplyeeID=4,FName="Nouran",LName="Badawy" ,Department = new Department { DeptID=1 , DeptName = "IT" }},

new Employee { EmplyeeID=5,FName="Samy",LName="Badawy" ,Department = new Department { DeptID=3 , DeptName = "IS" }},

new Employee { EmplyeeID=6,FName="Nadia",LName="Badawy" ,Department = new Department { DeptID=4 , DeptName = "DS" }},

};

return ListEmployee;

}

}

}

MB

################################################################

SelectMany

-----------------

دى بقه بيتم استخدمها لو لسته جواها لسته فبتخش تجيب الى جوه بسهوله و و توريهولنا بسهوله من الاخر ال SelectMany بتتعامل فقط مع الاراى او الليست او اى كوليكشن يكون ابن للسته الاصليه اى يتعامل فقط مع اللسته التى تكون داخل اللسته الاصلىه فقط لا غير   
 ادى مثال هوريك كود السليكت و كود السليكت many هتفهم ايه اهميه ال many ساعتها يعنى مثلا هاتلى الاقسام الى كل الناس هنا تابعه ليها

أولا ده الكلاس المشترك بينهم

class Employee

{

public int EmplyeeID { get; set; }

public string FName { get; set; }

public string LName { get; set; }

public List<string> Department { get; set; }

public List<Employee> getAllEmployee()

{

List<Employee> ListEmployee = new List<Employee>

{

new Employee { EmplyeeID=1,FName="Mahmoud",LName="Badawy",Department = new List<string> { "IT","IS" } },

new Employee { EmplyeeID=2,FName="Mostafa",LName="Badawy" ,Department = new List<string> { "DS","IS" }},

new Employee { EmplyeeID=3,FName="Ahmed",LName="Badawy" ,Department = new List<string> { "CS","DS" }},

new Employee { EmplyeeID=4,FName="Nouran",LName="Badawy" ,Department = new List<string> { "OR","IT" }},

new Employee { EmplyeeID=5,FName="Samy",LName="Badawy" ,Department = new List<string> { "DS","IS" }},

new Employee { EmplyeeID=6,FName="Nadia",LName="Badawy" ,Department = new List<string> { "CS","IS" }},

};

return ListEmployee;

}

}

هرجع الاقسام كده بال Select و طبعا ذى ما انت شايف 2 فور ايتش عشان اعرف اجيب الاقسام

class Program

{

public static void Main(string[] args)

{

Employee EmpObj = new Employee();

List<Employee> Employees = EmpObj.getAllEmployee();

var Emps = Employees.Select(Emp =>Emp.Department);

foreach (var item in Emps)

{

foreach (var MyItem in item)

{

Console.WriteLine(MyItem);

}

}

}

}

الكود ده بال SelectMany مفيش الا هى حته فور ايتش واحده

class Program

{

public static void Main(string[] args)

{

Employee EmpObj = new Employee();

List<Employee> Employees = EmpObj.getAllEmployee();

var Emps = Employees.SelectMany(Emp =>Emp.Department);

foreach (var item in Emps)

{

Console.WriteLine(item);

}

}

}

**نفس الكلام لو هتعامل مع اوبجيكت** بقه بص كده المثال ده هتلاحظ ان الموضوع هينفع مع السليكت و مش هينفع تماما مع ال SelectMany

اولا لان SelectMany تتعامل فقط مع الكوليكشنز فقط لا غير يعنى اروح اجيب بيها LIST او array انما مجبش بيها سترينج عشان هيجبهولك مفرط :D

و ثانيا لان ال SelectMany لا تقبل ال Anonymous Object بداخلها

ثالثا و انا برجع بيانات الاقسام هنا او بيانات الاب لو مكنتش كوليكشن هيروح داخل جايبها ذى ما هى كده و محولها لحروف وراب بعض

و ده مثال يا جماعه

اتنين كلاس مشتركين

class Department

{

public int DeptID { get; set; }

public string DeptName { get; set; }

}

class Employee

{

public int EmplyeeID { get; set; }

public string FName { get; set; }

public string LName { get; set; }

public Department Department { get; set; }

public List<Employee> getAllEmployee()

{

List<Employee> ListEmployee = new List<Employee>

{

new Employee { EmplyeeID=1,FName="Mahmoud",LName="Badawy",Department = new Department { DeptID=1 , DeptName = "IT" } },

new Employee { EmplyeeID=2,FName="Mostafa",LName="Badawy" ,Department = new Department { DeptID=2 , DeptName = "SWE" }},

new Employee { EmplyeeID=3,FName="Ahmed",LName="Badawy" ,Department = new Department { DeptID=2 , DeptName = "SWE" }},

new Employee { EmplyeeID=4,FName="Nouran",LName="Badawy" ,Department = new Department { DeptID=1 , DeptName = "IT" }},

new Employee { EmplyeeID=5,FName="Samy",LName="Badawy" ,Department = new Department { DeptID=3 , DeptName = "IS" }},

new Employee { EmplyeeID=6,FName="Nadia",LName="Badawy" ,Department = new Department { DeptID=4 , DeptName = "DS" }},

};

return ListEmployee;

}

}

و ادى اذاى اجيب بيانات الاتنين تيبول بقه مره بال Select

class Program

{

public static void Main(string[] args)

{

Employee EmpObj = new Employee();

List<Employee> Employees = EmpObj.getAllEmployee();

var Emps = Employees.Select(Emp => (

new {

ID = Emp.EmplyeeID,

FullName = Emp.FName + " " + Emp.LName,

DeptName = Emp.Department.DeptName

}

)).Where(A=>A.DeptName == "IT");

foreach (var EMP in Emps)

{

Console.WriteLine("Name : " + EMP.FullName + " --> Depart : " + EMP.DeptName );

}

}

}

**عرفت اجيبها بال Select** بس لما جيت اجرب اشغل على حته ان اوبجيكت يبقى جوه اوبجيكت مش ليست جوه اوبجيكت منفعش الشغل بال SelectMany خالص   
 نفس الكود بال SelectMany منفعش لان مينفعش اعمل جواها anonymous Object و لا ينفع اجيب حاجه من الابناء لو كانوا اوبجيكتات مش بداخلها لسته و بالتالى مش هقدر اجيب بيانات الموظفين و الاقسام و هيكون الى راجع لو مش كوليكشن مينفعش فمثلا الى راجع لو سترينج هتلاقى الاسترينج راجعلك حروف شوف الكود ده هتفهمنى

class Program

{

public static void Main(string[] args)

{

Employee EmpObj = new Employee();

List<Employee> Employees = EmpObj.getAllEmployee();

var Emps = Employees.SelectMany(Emp => Emp.Department.DeptName);

foreach (var item in Emps)

{

Console.WriteLine(item);

}

}

}

الفرق بين ال Select و SelectMany مبيبنش الفرق الا لو كان معانا لسته جوه لسته

ال Select بترجع اللسته الاب و اللسته الابنه عشان كده بتحتاج اتنين اتيريشن عشان اعرف اجيب اللسته الابنه

ال SelectMany بترجع السلكتايه الابنه على طول و خلاص :D و بالتالى بتحتاج اتيريشن واحد و شكرا

MB

################################################################

OrderBy – OrderByDescending – ThenBy – ThenByDescending -- reverse

-------------------------------------------------------------------------------------------------

ترتيب على حسب الحاجه الى انت هتحطها فيها من الصغير للكبير فمثلا انا عاوز ارتب الموظفين برقم القسم بتعهم هعمله كده و طبعا انا مستخدم الكلاسين المشتركين فمش هكتبهم تانى

class Program

{

public static void Main(string[] args)

{

Employee EmpObj = new Employee();

List<Employee> Employees = EmpObj.getAllEmployee();

var Emps = Employees.OrderBy(Em=>Em.Department.DeptID);

foreach (var item in Emps)

{

Console.WriteLine("FullName : " + item.FName +" "+ item.LName + " DeptID : " + item.Department.DeptID + " deptName : " + item.Department.DeptName);

}

}

}

طيب لو عاوز ارتبهم من الكبير للصغير ساعتها هتسخدم   
**OrderByDescending**

طيب

لو عاوز ارتب برقم القسم و بعد كده بالاسم الاول بتاع الموظف   
ساعتها هستخدم   
OrderBy للترتيب الاول و اى ترتيب أخر عاوز تضيفه على الترتيب ده بتحطه داخل فانكشن ال ThenBy  
لانك لو استخدمت OderBy مرتين هو بيعتبر ان المره التانيه هى يرتب عليها و خلاص و عشان يحلوا المشكله دى و يخلوا فى ترتيب بكذا حاجه عملوا ThenBy  
و هنكتب الى انت طالبه كده

var Emps = Employees.OrderBy(Em=>Em.Department.DeptID).ThenBy(Em=>Em.FName);  
و طبعا thenBy دى تقدر تعمل منها خامسين واحده مش هتقولك لأ و هتشتغل بترتيبهم كلهم

فى بقه حاجه برضوا للترتيب الى من الكبير للصغيره فيها هى كمان و بيسمى

**ThenByDescending**

اما بقه reverse دى يا كبير بتعكس الى راجع تخليه عكسه و خلاص

MB

################################################################

Take – TakeWhile – Skip - SkipWhile

-------------------------------------------------------------------------------------------------

Take : دى يا سيدى بتجيب من اللسته الى هى شغاله عليها عدد عناصر ما انا بحدد و ليكن فى المثال ده انا بقوله هاتلى أول 3 مدن فى الأراى ده

class Program

{

public static void Main(string[] args)

{

string[] Countries = { "Egypt", "France", "Sudan", "Emirates", "USA", "Iraq", "Lebia", "KSA", "Germany" };

IEnumerable<string> TakenCountries = Countries.Take(3);

foreach (var item in TakenCountries)

{

Console.WriteLine(item);

}

}

}

TakeWhile : دى بقه بتجيب العناصر الى بتقابلها طول ما العناصر دى بتحقق الشرط الى جواها

فمثلا لو قولتله الشرط ان حروف الدوله يبقى اكبر من 3 حروف ساعتها هيروح يفحص العناصر فهيلاقى مصر و فرنسا و السودان و الامارات اكبر من 3 حروف فهياخدهم و اول ما يخش على امريكا هيلاقيها بتساوى 3 حروف يعنى مش اكبر من 3 و بالتالى الشرط لا ينطبق عليها ساعتها هيطلع false اول ما هيشوف ال false هيقف و مش هيرجع حاجه تانى خلاص

public static void Main(string[] args)

{

string[] Countries = { "Egypt", "France", "Sudan", "Emirates", "USA", "Iraq", "Lebia", "KSA", "Germany" };

IEnumerable<string> TakenCountries = Countries.TakeWhile(Country=>Country.Length > 3);

foreach (var item in TakenCountries)

{

Console.WriteLine(item);

}

}

Skip : عكس take يعنى هى بتتجاهل اول عناصر انا محددله عددها مثال انا هقوله هنا تجاهل اول 3 بلاد و جيبلى باقى البلاد بقه

public static void Main(string[] args)

{

string[] Countries = { "Egypt", "France", "Sudan", "Emirates", "USA", "Iraq", "Lebia", "KSA", "Germany" };

IEnumerable<string> TakenCountries = Countries.Skip(3);

foreach (var item in TakenCountries)

{

Console.WriteLine(item);

}

}

SkipWhile : عكس TakeWhile و بالتالى هى طول ما الشرط بيتحقق و يرجع true هى بتتجاهل الى بيرجعلها true و أول ما تشوف false لعنصر ما ساعتها بتخده هو و العناصر الى بعده على سبيل المثال برضوا انا هنا هقوله اعمل Skip لاى حاجه اكبر من 3 حروف و بالتالى هو هيعد يعمل Skip لحد ما يوصل ل USA ساعتها هيلاقيها محققتش الشرط هيروح واخدها هى و العراق و ليبيا و السعوديه و ألمانيا

public static void Main(string[] args)

{

string[] Countries = { "Egypt", "France", "Sudan", "Emirates", "USA", "Iraq", "Lebia", "KSA", "Germany" };

IEnumerable<string> TakenCountries = Countries.SkipWhile(Country=>Country.Length > 3);

foreach (var item in TakenCountries)

{

Console.WriteLine(item);

}

}

MB

################################################################

Implementing Paging

-------------------------------------

هنا بقه بتتعلم اذاى تعمل Paging بالحاجات الى لسه شرحنها فوق دى   
بص يا سيدى ان اهم حاجه يبقى معاك 1- رقم الصفحه الى انت رايحها 2- حجم الصفحات بمعنى الصفحه هيكون فيها كام عنصر   
كل الى هتعمله انك هتديله و ليكن الصفحه التانيه هيروح هو عامل ايه ؟؟

المفروض هيتجاهل اول صفحه و هياخد تانى صفحه و يتجاهل الباقى بقه طيب تتعامل اذاى دى بقه كود ؟؟

public static void Main(string[] args)

{

string[] Countries = { "Egypt", "France", "Sudan", "Emirates", "USA", "Iraq", "Lebia", "KSA", "Germany" };

int pageSize = 0;

int PageNumber = 0;

bool Exit = false;

int TotalElements = Countries.Length;

do

{

Console.WriteLine("You Have : {0} Elements ", TotalElements);

Console.WriteLine("########################");

Console.Write("Write Page Size : ");

pageSize = Convert.ToInt16(Console.ReadLine());

Console.Write("Write Page Number : ");

PageNumber = Convert.ToInt16(Console.ReadLine());

Console.WriteLine("You Have those Elements in Page Number {0} With page size {1}", PageNumber, pageSize);

IEnumerable<string> TakenCountries = Countries.Skip((PageNumber - 1) \* pageSize).Take(pageSize);

foreach (var item in TakenCountries)

{

Console.WriteLine(item);

}

Console.Write("To exit Y/N: ");

String Key = Console.ReadLine();

if (Key.ToUpper() == "Y")

{

Exit = true;

}

} while (Exit == false);

}

طبعا الكود الى فوق ده هيقضى الغرض بس هو حرفيا كود مش نضيف و بالتالى انا محتاج انضفه و ده لان انا لو كتبت اى مدخل استرينج و جيت اعمله كونفيرت لانتيجر الدنيا هتضرب و ليكن هدخل كلمه محمود فى ال pageSize هيروح ضارب طب اذاى احل المشكله دى ؟؟

اغستخدم tryPares لو نفع فعلا اروح ساعتها حاطط الى انا كاتبه فى OutPut عشان احل المشاكل دى يا رايق

using System;

using System.Collections.Generic;

using System.Linq;

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

string[] Countries = { "Egypt", "France", "Sudan", "Emirates", "USA", "Iraq", "Lebia", "KSA", "Germany" };

int pageSize = 0;

int PageNumber = 0;

bool Exit = false;

int TotalElements = Countries.Length;

do

{

Console.WriteLine("You Have : {0} Elements ", TotalElements);

Console.WriteLine("########################");

Console.Write("Write Page Size : ");

if (int.TryParse(Console.ReadLine(), out pageSize))

{

if(pageSize != 0)

{

int pages = 0;

if (TotalElements % pageSize != 0)

{

pages = (TotalElements / pageSize) + 1 ;

}

else

{

pages = (TotalElements / pageSize) ;

}

Console.WriteLine("You Have to Chosse Pages Between 1 & {0} ", pages);

Console.Write("Write Page Number : ");

if (int.TryParse(Console.ReadLine(), out PageNumber))

{

Console.WriteLine("You Have those Elements in Page Number {0} With page size {1}", PageNumber, pageSize);

IEnumerable<string> TakenCountries = Countries.Skip((PageNumber - 1) \* pageSize).Take(pageSize);

foreach (var item in TakenCountries)

{

Console.WriteLine(item);

}

Console.Write("To exit Y/N: ");

String Key = Console.ReadLine();

if (Key.ToUpper() == "Y")

{

Exit = true;

}

}

}

}

} while (Exit == false);

}

}

}

MB

################################################################

Deferred Vs Immediate Execution

-------------------------------------

Immediate Execution : ده من الاخر بتتم فى نفس ذات السطر بتستناش الاتريشن و امثله عليها اهه

هنجيب لست و نجيب الى جواها كله و بعد مده هنضيف عليها اى حاجه قبل الاتريشن هتلاقيها مطلعتش فى الاتريشن و ده لان تم تنفيذها فى نفس السطر بشكل فورى

و للامثله عليها فانكشن   
ToList() -Count() – Max() – Min() – Average()

و ادى كود عملى

public static void Main(string[] args)

{

List<string> Countries = new List<string>() ;

Countries.Add("Egypt");

Countries.Add("Lebia");

Countries.Add("France");

List<string> My\_Countiries = Countries.ToList();

Countries.Add("USA");

foreach (var item in My\_Countiries)

{

Console.WriteLine(item);

}

}

فى حين انى لو استخدمت اى حاجه تانيه ذى ال Select - where - Take - Skip

بيتنفذ فى اثناء الاتريشن الى هو لفه اللوب يعنى :D

public static void Main(string[] args)

{

List<string> Countries = new List<string>() ;

Countries.Add("Egypt");

Countries.Add("Lebia");

Countries.Add("France");

var My\_Countiries = Countries.Where(X=>X.Length>3);

Countries.Add("USA");

Countries.Add("Sudan");

Countries.Add("Germany");

foreach (var item in My\_Countiries)

{

Console.WriteLine(item);

}

}

على الرغم انى ضايف 3 بلاد بعد فانكشن الوير الا ان لما النتايج طلعت طلع ان ال where حطت المدن الجديده معاها عادى و بالتالى هى بتتنفذ وقت اللوب مش فى ساعتها من الاخر

MB

################################################################

Tolist – ToArray

-------------------------------------

ToList : بتحول البيانات الى راجعالك الى List و بتخليها تترن فى نفس السطر مش بتستنى الاتريشن

ToArray : بتحول البيانات الى راجعالك الى array و بتخليها تترن فى نفس السطر مش بتستنى الاتريشن

MB

################################################################

ToDictionary

-------------------------------------

بتحول البيانات الى جايه و فى الغالب بتبقى جايه لسته من الاوبجيكتات الى Dictionary مكون من Key & Value   
بس خلى بالك و انت بتعمل ال Key لازم يبقى عمود Unique و الا هيضرب من الاخر يتعمل على ال ID احسن حاجه

namespace CsharpLearning

{

class Student

{

public int StudentID { get; set; }

public string StudentName { get; set; }

public int TotalMarks { get; set; }

}

class Program

{

public static void Main(string[] args)

{

List<Student> Students = new List<Student> {

new Student {StudentID=101,StudentName="Mahmoud" , TotalMarks=880 },

new Student {StudentID=101,StudentName="Mostafa" , TotalMarks=1100 },

new Student {StudentID=101,StudentName="Karim" , TotalMarks=900 },

} ;

Dictionary<string,int> studentDic = Students.ToDictionary(Stud => Stud.StudentName, stud => stud.TotalMarks);

foreach (KeyValuePair<string, int> item in studentDic)

{

Console.WriteLine("Key : " + item.Key +" --> value : " + item.Value);

}

}

}

}

MB

################################################################

ToLookUp

-------------------------------------

بص يا سيدى هما لقوا ان ال dictionary بيقبلش ان Key يبقى متكرر فعملوا ال Lookup بحيث انه ينفع ياخد Key متكرر عادى بس ساعتها بيخلى كل الكيز المتكرر على هيئه جروب و بالتالى اصبحت كده برضوا حاجه واحده فريده من نوعها بداخلها بقه معلومات متكرر او مش متكرره مش مهم الى هى ال values

على فكره صحيح اللوك اب بيقبل ال Key بس او ال Key & Value  
ادى المثالين اهه و انت استخدم الطريقه الى تعجبك

using System;

using System.Collections.Generic;

using System.Linq;

namespace CsharpLearning

{

class Student

{

public int StudentID { get; set; }

public string StudentName { get; set; }

public int TotalMarks { get; set; }

public string StudentDept { get; set; }

}

class Program

{

public static void Main(string[] args)

{

List<Student> Students = new List<Student> {

new Student {StudentID=101,StudentName="Mahmoud" , TotalMarks=1159 ,StudentDept="IS" },

new Student {StudentID=102,StudentName="Noha" , TotalMarks=1200 ,StudentDept="IT"},

new Student {StudentID=103,StudentName="Mostafa" , TotalMarks=1400 ,StudentDept="IS"},

new Student {StudentID=104,StudentName="Karim" , TotalMarks=900 ,StudentDept="CS"},

new Student {StudentID=105,StudentName="Rami" , TotalMarks=687 ,StudentDept="DS"},

new Student {StudentID=106,StudentName="Saed" , TotalMarks=681 ,StudentDept="IT"},

new Student {StudentID=107,StudentName="Mariam" , TotalMarks=1000 ,StudentDept="CS"},

new Student {StudentID=108,StudentName="Khaled" , TotalMarks=954 ,StudentDept="CS"},

};

ILookup<string, string> studentDic = Students.ToLookup(stud=>stud.StudentDept, Stud=>Stud.StudentName);

foreach (IGrouping<string, string> item in studentDic)

{

Console.WriteLine("Group : " + item.Key);

Console.WriteLine("------------------------");

// foreach (var Elem in studentDic[item.Key])

foreach (var Elem in item)

{

Console.WriteLine("Names : " + Elem);

}

Console.WriteLine("########################");

}

}

}

}

**الطريقه التانيه الى بال Key بس و طبعا الطريقه دى الاحسن لانها بتيح ليك انك ترجع اى بيان مش مجرد حاجتين بس**

var studentDic = Students.ToLookup(stud=>stud.StudentDept);

foreach (var item in studentDic)

{

Console.WriteLine("Group : " + item.Key);

Console.WriteLine("------------------------");

// foreach (var Elem in studentDic[item.Key])

foreach (var Elem in item)

{

Console.WriteLine("Names : " + Elem.StudentName);

}

Console.WriteLine("########################");

}

MB

################################################################

Cast

-----------------

دى بتحول الى راجع من نوع اوبجيكت الى نوع معين انا عاوزه و لازم يبقى كل العناصر الموجوده و هحولها دى تبقى ينفع فعلا تتحول الى النوع ده يعنى مثلا مجبش ارقام كلها انتيجرز و اقوله انا عاوزها فلوت فمثلا انا عندى اراى ليست فيها شويه عناصر من نوع Object و عاوز احولها ل Int فهو بيحولها كل لانتيجر دون استثناء و لو فى فى النص عنصر مش هيعرف يحوله بيروح الكود ضارب   
**بص عشان انجزلك الحته دى اعتبر انه مش يبعمل كاست اصلا اعتبر انك بتقوله هاتلى العناصر الانتيجر بس فلو هو دخل و لقى انتيجرز بس بيجيب الدنيا سليمه لو لقى فى عنصر فى النص نوعه مختلف و ليكن دابل مثلا بيروح ضارب**

ده الكود السليم الى مش هيضرب عشان مفيش حاجه فى النص تبوظ الكاستينج

public static void Main(string[] args)

{

ArrayList Numbers = new ArrayList() { 1, 2, 4, 5, 7, 8 };

IEnumerable<int> DNumbers = Numbers.Cast<int>();

foreach (int item in DNumbers)

{

Console.WriteLine(item);

}

}

**اما ده بقه الكود الى هيضرب بسبب انه معرفش يعمل لكل العناصر كاستينج**

public static void Main(string[] args)

{

ArrayList Numbers = new ArrayList() { 1, 2, 4, "5", 7, 8 };

IEnumerable<int> DNumbers = Numbers.Cast<int>();

foreach (int item in DNumbers)

{

Console.WriteLine(item);

}

}

مثال اخر ضارب عشان الارقام كلها دابلز و انا عوزها دابل و فى النص رقم انتيجر

public static void Main(string[] args)

{

ArrayList Numbers = new ArrayList() { 1.5, 2.5, **4**, 7.8, 8.5 };

IEnumerable<double> DNumbers = Numbers.Cast<double>();

foreach (double item in DNumbers)

{

Console.WriteLine(item);

}

}

################################################################

OfType

-----------------

طيب يحلوا المشكله دى اذاى ؟؟؟؟

عملوا فانكشن اسمها OfType و دى بقه بترجع فقط العناصر الى هى نفس النوع الى انا قايله عليه فمثلا لو جربتها على المثال الى كان بيضرب ده هتلاقيها رجعتلك 1 و 2 و 4 و 7 و 8 و هتتجاهل تماما 5 عشان ملقتهوش انه انتيجر اصلا فتجاهلته

public static void Main(string[] args)

{

ArrayList Numbers = new ArrayList() { 1, 2, 4, "5", 7, 8 };

IEnumerable<int> DNumbers = Numbers.OfType<int>();

foreach (int item in DNumbers)

{

Console.WriteLine(item);

}

}

MB

################################################################

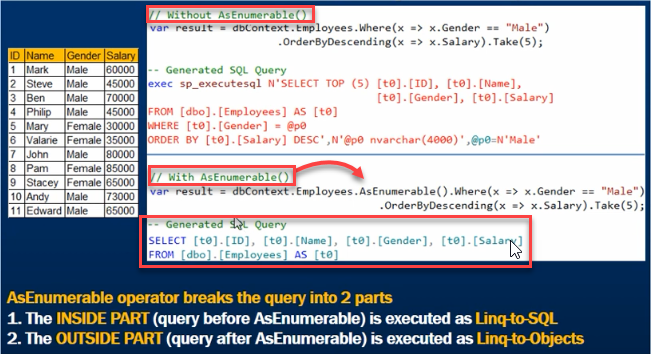
AsEnumerable Vs AsQuerable

-----------------------------------------------

AsQuerable : ده يا سيدى ان اللينك كلها تتحول ل SQL و دى عمتا بتتكتبش فى الكود

AsEnumerable : دى يا سيدى الى قبلها بيتحول ل SQL و الى بعدها بيتحول ل Object

بالنسبه للنتايج بقه فالتنين بيطلعوا نفس النتايج   
بس الفرق ان ال AsQuerable ده بيتنفذ كله فى ال Server اما ASEnumerable  
الى قبل الفانكشن بتاعت ال AsEnumerable بيتنفذ فى السيرفر اما الى بعديها بيتنفذ فى ال Client-Side  
بصراحه شايف انى افلتر فى السيرفر احسن الا بقه لو هجيب كل حاجه مره واحده و اقعد اقطع فيها لاماكن كتير عشان مكلمش السيرفر عشرين مره :D   
المثال فى الصوره اهه بص على الجينيرتيد SQL



MB

################################################################

GroupBy

-----------------------------------------------

دى من الاخر بترجع سلسله من الجروبات و بتبقى على شكل (Key & Value) تماما ذى ما كان اللوك اب بيرجع بس هنا بيختلف شويه

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

namespace CsharpLearning

{

class Student

{

public int StudentID { get; set; }

public string StudentName { get; set; }

public int TotalMarks { get; set; }

public string StudentDept { get; set; }

}

class Program

{

public static void Main(string[] args)

{

List<Student> Students = new List<Student> {

new Student {StudentID=101,StudentName="Mahmoud" , TotalMarks=1159 ,StudentDept="IS" },

new Student {StudentID=102,StudentName="Noha" , TotalMarks=1200 ,StudentDept="IT"},

new Student {StudentID=103,StudentName="Mostafa" , TotalMarks=1400 ,StudentDept="IS"},

new Student {StudentID=104,StudentName="Karim" , TotalMarks=900 ,StudentDept="CS"},

new Student {StudentID=105,StudentName="Rami" , TotalMarks=687 ,StudentDept="DS"},

new Student {StudentID=106,StudentName="Saed" , TotalMarks=681 ,StudentDept="IT"},

new Student {StudentID=107,StudentName="Mariam" , TotalMarks=1000 ,StudentDept="CS"},

new Student {StudentID=108,StudentName="Khaled" , TotalMarks=954 ,StudentDept="CS"},

};

var studentDic = Students.GroupBy(stud => stud.StudentDept);

foreach (var item in studentDic)

{

Console.WriteLine("Groups : " + item.Key + " -- Count of Students : " + item.Count());

Console.WriteLine("------------------------------------------");

foreach (var Elem in item)

{

Console.WriteLine("Name : " + Elem.StudentName + " --> Dept : " + Elem.StudentDept );

}

Console.WriteLine("##########################################");

}

}

}

طيب بعد ما عرفنا اذاى نعمل GroupBy باللينك دلوقتى احنا عاوزين نعرف نعمل اذاى GroupBy With OrderBy

انا عاوز اعمل جروب باى بالاقسام و ارتب الى راجع جوه كل قسم بالاسماء بقه   
- اولا بعمل جروب باى بالقسم عادى خالص فى انونيموس فانكشن ذى ما انت شايف و ده علشان لو عاوز استخدم الجروب باى بحاجه واحده او بتنين او بتلاته بمزاجك يعنى

- ثانيا بعد كده بقوله اعملى للحاجات الراجعه دى OrderBy و ليكن القسم عشان يرتب الى راجع على هيئه جروبات بالاقسام فيبقى الاقسام هتيجى بترتيب ابجدى سليم

- ثالثا انا عاوز ارتب بقه الى جوه كل قسم يعنى كل قسم اطلع منه الطلبه الى فيه مترتبه بالاسم   
فانا عشان اعمل كده بروح اعمل سليكت تانى عشان ارجع الداتا الى انا عاوزها بالشكل الى انا عاوزه و الى فيها برجعها بالشكل الاتى   
1- بحط فيها الحاجات الى انا مرتب بيها تمام الى هى ال keys من الاخر

2- بروح مرجع فيها بقه اوبجيكت كامل الاوبجيكت ده بحط فيه الى طلع من الجروب و اتعمله اوردر بالقسم و من ثم اتعمل جواه بقه ترتيب باسم الناس الى في كل جروب (برجع فيه الداتا الى داخل كل جروب على حدى و بعملها ترتيب باسماء الناس الى جواه )

الكود هيبقى هو هو بس التغيير فى الجزء ده بس و بالتالى مش محتاج اكتبه كله انا محتاج منه الجزء ده بس

var StudentGroups = Students.GroupBy(stud => new

{

StudDept = stud.StudentDept

}).OrderBy(Group => Group.Key.StudDept).Select(

Group => new

{

StudentDept = Group.Key.StudDept,

Students\_G\_Ordered = Group.OrderBy(X => X.StudentName)

}

);

foreach (var Groups in StudentGroups)

{

Console.WriteLine(" Groups : " + Groups.StudentDept + " -- Count Of Students : " + Groups.Students\_G\_Ordered.Count());

Console.WriteLine(" ---------------------------------------------- ");

foreach (var Ordered\_Groups in Groups.Students\_G\_Ordered)

{

Console.WriteLine(" Names : " + Ordered\_Groups.StudentName + " -- TotalMarks : " + Ordered\_Groups.TotalMarks);

}

Console.WriteLine(" ############################################## ");

}

الفورايتش بقه اولا بلف على الى راجع من اللينك كويرى فهيطلعلى الجروبات و فيها كام طالب طيب انا عاوز اجيب الطلبه الى جواه بقه

MB

################################################################

GroupBy Multi

-----------------------------------------------

دلوقتى فوق انا علمتك اذاى تعمل الجروب باى بحاجه واحده بس دلوقتى هعلمك اذاى تعملها بحجتين تلاته اربعه لان الموضوع بسيط خالص الفرق اننا الحاجه الى عاوزينه يزودها هنزودها فى الانونيموس فانكشن الى بحط فيها ال keys الى بعمل بيها جروب باى و انا معلم على شويه التغيرات الى تمت على الكود و هو هنا عمتا بيعمل جروب بالقسم و بنوع الشخص راجل او ست

namespace CsharpLearning

{

class Student

{

public int StudentID { get; set; }

public string StudentName { get; set; }

public int TotalMarks { get; set; }

public string StudentDept { get; set; }

public string Gender { get; set; }

}

class Program

{

public static void Main(string[] args)

{

List<Student> Students = new List<Student> {

new Student {StudentID=101,StudentName="Mahmoud" , TotalMarks=1159 ,StudentDept="IS" , Gender = "Male" },

new Student {StudentID=102,StudentName="Noha" , TotalMarks=1200 ,StudentDept="IT", Gender = "Female"},

new Student {StudentID=103,StudentName="Mostafa" , TotalMarks=1400 ,StudentDept="IS", Gender = "Male"},

new Student {StudentID=104,StudentName="Karim" , TotalMarks=900 ,StudentDept="CS", Gender = "Male"},

new Student {StudentID=105,StudentName="Rami" , TotalMarks=687 ,StudentDept="DS", Gender = "Male"},

new Student {StudentID=106,StudentName="Saed" , TotalMarks=681 ,StudentDept="IT", Gender = "Male"},

new Student {StudentID=107,StudentName="Mariam" , TotalMarks=1000 ,StudentDept="CS", Gender = "Female"},

new Student {StudentID=108,StudentName="Khaled" , TotalMarks=954 ,StudentDept="CS", Gender = "Male"},

};

var StudentGroups = Students.GroupBy(stud => new

{

StudDept = stud.StudentDept,

studGender= stud.Gender

}).**OrderBy**(

Group => Group.Key.StudDept

).**ThenBy**(Group => Group.Key.studGender).Select(

G => new

{

StudentDept = G.Key.StudDept,

studentGender=G.Key.studGender,

Students\_G\_Ordered = G.OrderBy(X => X.StudentName)

}

); foreach (var Groups in StudentGroups)

{

Console.WriteLine(" Groups : " + Groups.StudentDept + " Gender -- " + Groups.studentGender +" -- Count Of Students : " + Groups.Students\_G\_Ordered.Count());

Console.WriteLine(" ---------------------------------------------- ");

foreach (var Ordered\_Groups in Groups.Students\_G\_Ordered)

{

Console.WriteLine(" Names : " + Ordered\_Groups.StudentName + " -- TotalMarks : " + Ordered\_Groups.TotalMarks);

}

Console.WriteLine(" ############################################## ");

}

}

}

}

MB

################################################################

First Vs Last Vs FirstOrDefault Vs LastOrDefault

--------------------------------------------------------------------------

First : ده يا سيدى بيجيب أول عنصر فقط و ادى مثال

public static void Main(string[] args)

{

int[] Numbers = { 5, 3, 2, 45, 12, 4, 7, 8, 32 };

int MyNumber = Numbers.First();

Console.WriteLine("My Number is : " + MyNumber);

}

طيب نفترض بقه ان الاراى فاضيه اصلا ؟؟

ساعتها هيضرب لو الاراى كده

public static void Main(string[] args)

{

int[] Numbers = { };

int MyNumber = Numbers.First();

Console.WriteLine("My Number is : " + MyNumber);

}

فعشان يحلوا المشكله دى قالك نعمل   
**FirstOrDefault** بحيث انه يروح يشوف الاراى لو فيها عناصر فعلا يجيب اول واحده فيها طب لو لقاها فاضيه يروح مرجع قيمه افتراضيه و ليكن هنجرب نفس الامصله الى فوق بقه على الفانكشن دى

**طيب لو عاوز تجيب اخر رقم تستخدم ايه ؟؟**

Last أو الاحسن طبعا LastOrDefault

**معلومه خاصه بيهم كلهم بقه هما بيقبلوا ان لاندا اكسيبريشن جواهم و ده لو مثلا عاوز اقوله هاتلى أول رقم زوجى مثلا أو أخر رقم زوجى برضوا :D :D   
ده مثال انه يجيب اول رقم زوجى**

int MyNumber = Numbers.FirstOrDefault(x=> x % 2 == 0);

و ده مثال انه يجيب اخر رقم زوجى

int MyNumber = Numbers.LastOrDefault(x=> x % 2 == 0);

MB

################################################################

ElementAt Vs ElementAtOrDefault

--------------------------------------------------------------------------

ElementAt : دى يا سيدى بتروح تجيبلك عنصر بدلاله الانديكس الى انت محددهلها يعنى مثلا انا عاوز هنا اجيب العنصر الى فى الانديكس رقم 3 فهيدينى القيمه 45

int[] Numbers = { 5, 3, 2, 45, 12, 4, 7, 8, 32 };

int MyNumber = Numbers.ElementAt(3);

Console.WriteLine("My Number is : " + MyNumber);

هنقول مثلا نفترض ان مفيش انديكس بالرقم ده ايه الى هيحصل ؟؟

هيعمل ايرور على طول

طيب حل المشكله ايه ؟؟

ElementAtOrDefault عشان لو ملقاش الانديكس موجود يديك قيمه افتراضيه

Single Vs SingleOrDefault : دول من الاخر لازم يكون الاراى فيها يا اما عنصر واحد ياما عنصر واحد فقط الى بيحقق الشرط الى جواهم غير كده الاتنين بيطلعوا ايرورز و الفرق بينهم ان واحد بتقبل ان الاراى تكون فاضيه عادى و التانيه لأ

لو قولنا مثلا هنطبق اى واحده فيهم على الى فوق دى و نقوله هتلنا رقم زوجى بالسنجل هتضرب لان فى ارقام زوجى كتير و هو شرطه عشان متضربش لازم يكون رقم وحيد شريد عماد و سعيد ههههههههه هو الى راجع

على سبيل المثال الكود ده هيضرب عشان مرجع ارقام زوجيه كتير و السنجل عاوز واحد بس

int[] Numbers = { 5, 3, 2, 45, 12, 4, 7, 8, 32 };

int MyNumber = Numbers.SingleOrDefault(X=>X % 2 == 0);

Console.WriteLine("My Number is : " + MyNumber);

**اما لو عملناه كده هيشتغل و ده لان مفيش غير رقم واحد بس هو الى بيساوى 3**

int MyNumber = Numbers.SingleOrDefault(X=>X == 3);

MB

################################################################

DefaultIfEmpty

---------------------------------

DefaultIfEmpty : بص يا سيدى لو الاراى او الكوليشكن او الليست فاضى فده هيرجع القيمه الافتراضيه الى هى ممكن يا ما بتاعت الكومبيلر او انك تحطهاله بنفسك و ليكن انا هعملها ب 10

اما لو مليان هيرجعه ذى ما هو

int[] Numbers = { 5, 3, 2, 45, 12, 4, 7, 8, 32 };

var MyNumber = Numbers.DefaultIfEmpty(10);

foreach (var item in MyNumber)

{

Console.WriteLine(item);

}

جرب بقه تديله اراى فاضيه هتلاقيه مرجعلك القيمه الافترضيه الى هى هنا بالنسبه للانتيجر بتساوى صفر و لكن انا قايله خليها 10 ياباشا

int[] Numbers = { };

var MyNumber = Numbers.DefaultIfEmpty(10);

foreach (var item in MyNumber)

{

Console.WriteLine(item);

}

MB

################################################################

Group Join

------------------------------------

بص دى نفس فكره ال outer join فى السكوال

هنا دى بتربط بين 2 تيبول يعنى جدولين و بتعملهم هيراريكال بمعنى ان و ليكن انا عندى الموظفين فى جدول الموظفين و عندى الاقسام فى جدول الاقسام كل موظف بيشتغل فى قسم ما كل قسم ليه برايمرى كى و البرايمرى كى الى فى جدول الاقسام ده بعمل انا بقه منه فورن كى داخل جدول الموظقين و هما الاتنيين دول الفورن كى و البريمرى كى هما الى هربط بيهم الجدولين ببعض

لما انت بقه بتعمل الجروب جوين بيروح هو عامل ايه بقه مرجعلك بيانات الاقسام و الموظفين مع بعض و انت بتنقى منهم بقه الاسامى و بتفكك من الارقام و بترجع بقه بمزاجك بقه الدنيا على شكل هيراريكال فهتلاحظ ان الى هيرجع

كل اسماء الاقسام و تحت كل قسم اسماء الموظفين الى فيه حتى لو القسم مفيهوش موظفين بيروح حاططهولك و يقولك انه مفهوش موظفين

و ادى مثال

**ملاحيظ هامه جدا قبل ما تعمل الكود لازم تسأل نفسك انت هتعمل ايه الى هيبقى على شكل جروبات ؟؟**

**الاقسام**

**اذا يا كبير لازم الاقسام تبقى هى الجدول الاول الى ناحيه الشمال من الجروب جوين**   
بالنسبه للكود اهه و هفهمهولك قبل الكتابه

**اولا انت بتنادى على الجدول بتاع الاقسام لان انت عاوز تعمل الاقسام على هيئه جروبات بعد كده بتعمل الجروب جوين و ناحيه يمينها بقه بتنادى على الجدول بتاع الموظفين الى هيبقى عباره عن اسماء الناس الى جوه كل قسم من الاقسام**

**طيب دلوقتى معاك الجدولين المفروض بقه هو هيفهم اذاى ان كل موظف تابع لقسم معين ؟؟**

**عن طريق البرايمرى كى و الفورن كى و دول برضوا هنحطهم بالترتيب عشان دول الى هيربطوا الدنيا**

**طيب بعد ام قولتله على الجدولين و البرايمرى كى و الفورن كى المفروض هتعمل ايه ؟**

**المفروض هعمل فانكشن اكون فيها شكل الداتا الى راجعه و بالتالى هعمل فانكشن اديها نوع الجدولين بحيث اخد الاوبجيكتات او الجدوال و احط كل واحد منهم فى الاوبجيكت بتاعه**

**كده خلاص مش باقى غير انى اروح انادى بقه للنتايج فى اتنين فور ايتش افتكر ان قولتلك هنا لازم 2 فورايتش**

الكود اهه يا كبير على فكره الاتنين كلاس بتاعت الموظفين و الاقسام دول مشتركين بقه فى كل الاكواد الى جايه الى هنعملها ليها علاقه بالجويبن بعده كده

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

var EmoployeeByDepartment = Department.GetAllDepartment().GroupJoin(Employee.GetAllEmployee(),

D\_DeptIDKey => D\_DeptIDKey.DepartmentID,

E\_DeptIDKey => E\_DeptIDKey.DeptID,

(Dept, Emp) => new

{

Departments = Dept,

Employees = Emp

});

foreach (var item in EmoployeeByDepartment)

{

Console.WriteLine(item.Departments.DepartmentName);

Console.WriteLine("#########################");

foreach (var Elem in item.Employees)

{

Console.WriteLine("Name : "+ Elem.EmployeeName + " --> Gender : " + Elem.EmployeeGender);

}

Console.WriteLine("--------------------------");

}

}

}

class Employee

{

public int EmployeeID { get; set; }

public string EmployeeName { get; set; }

public string EmployeeGender { get; set; }

public int DeptID { get; set; }

public static List<Employee> GetAllEmployee()

{

List<Employee> EmployeeList = new List<Employee>

{

new Employee { EmployeeID=1,EmployeeName="Mahmoud Badawy",EmployeeGender="Male",DeptID=1 },

new Employee { EmployeeID=1,EmployeeName="Mostafa Badawy",EmployeeGender="Male",DeptID=2 },

new Employee { EmployeeID=1,EmployeeName="Ahmed Sadek",EmployeeGender="Male",DeptID=1 },

new Employee { EmployeeID=1,EmployeeName="Noha Saber",EmployeeGender="Female",DeptID=1 },

new Employee { EmployeeID=1,EmployeeName="Kero Atef",EmployeeGender="Male",DeptID=2 },

new Employee { EmployeeID=1,EmployeeName="Sara Mahmoud",EmployeeGender="Female",DeptID=1 },

new Employee { EmployeeID=1,EmployeeName="Karim Elkerdasy",EmployeeGender="Male",DeptID=2 },

new Employee { EmployeeID=1,EmployeeName="Sayed Sanad",EmployeeGender="Male"} };

return EmployeeList;

}

}

class Department

{

public int DepartmentID { get; set; }

public string DepartmentName { get; set; }

public static List<Department> GetAllDepartment()

{

List<Department> DepartmentList = new List<Department>

{

new Department {DepartmentID=1,DepartmentName="CS" },

new Department {DepartmentID=2,DepartmentName="IT" },

new Department {DepartmentID=3,DepartmentName="DS" }

};

return DepartmentList;

}

}

}

MB

################################################################

Join

------------------------------------

دى نفس فكره ال inner Join الى فى السكوال بحيث انه بيجيب بيانات الجدولين الى بيربط بينهم بس بيرجع فقط البيانات المشتركه بين الجدولين   
يعنى فى حاله جدول الموظفين و الاقسام

هيشوف الموظفين لو فى موظف محطوطله رقم بتاع القسم برقم موجود فعلا فى الاقسام هيروح مرجع البيانات المشتركه من الجدولين للموظف ده

اما فى حاله ان الموظف ده مش موجود فى اى فسم فمش هيرجعه فى النتايج اصلا ذى الموظف سيد سند مش موجود فيه قسم فمش هيرجعوا اساسا فى النتايج اما باقى الموظفين هيرجعهم عادى كل واحد و القسم التابع ليه

و برضوا نفس الكلام على الاقسام يعنى قسم ال DS ده محدش عبر أمه و دخله من الموظفين و بالتالى برضوا مش هيرجع فى النتايج بتعتنا

و هو ده الفرق بين الجوين و الجروب جوين

ان الجوين مش هيرجع ال DS عشان ملقاش موظف فيها اصلا بينما فى الجروب جوين هرجعها و يقولك القسم ده مفيهوش حد

كذلك الجوين بترجع كل بياناتها باتريشن واحد انما الجروب جوين بيرجع البيانات ب 2 اتريشن الاتريشن الاول للجروبات و الاتريشن التانى للبيانات الى داخل كل جروب

ادى كود ال Join مش هتلاقى اختلاف كبير بينه و بين الى قبله غير فى الاتريشنز و غيرت الفانكشن و خلاص

public static void Main(string[] args)

{

var EmoployeeByDepartment = Department.GetAllDepartment().Join(Employee.GetAllEmployee(),

D\_DeptIDKey => D\_DeptIDKey.DepartmentID,

E\_DeptIDKey => E\_DeptIDKey.DeptID,

(Dept, Emp) => new

{

Departments = Dept,

Employees = Emp

});

foreach (var item in EmoployeeByDepartment)

{

Console.WriteLine("Dept : " + item.Departments.DepartmentName +

" Name : " + item.Employees.EmployeeName + " Gender : " +

item.Employees.EmployeeGender);

Console.WriteLine("#########################");

}

}

MB

################################################################

Left join

-----------------

دى يا سيدى بتربط بين جدولين بتجيب المشترك بينهم + بيزيد عليه بقه انه بيجيب باقى ال rows الى مش مشتركه من الجدول الى ناحيه الشمال   
من الاخر هو بيربط بين جدولين

جدول يمين و جدول شمال

الجدول اليمين ده فيه قسم IT1 و CS 2 و DS 3

الجدول الشمال فيه الموظفين و كل موظف في قسم مثلا و ليكن بقه كل الموظفين متوزعين على الاقسام ماعدا موظف واحد هو الى ملوش اى قسم

فى الليفت جوين بيجى يجيبلك كل موظف و فى انهى فسم و فى الموظف الى ملوش قسم بيروح حاطط جنبه بيانات فاضيه و بيجيبه عادى الكلام ده لو فى الجوين العادى مكنش هيرجع الموظف ده اصلا لانه مش تابع لاى قسم اساسا

بالمعنى العادى فى الجوين العادى كانك بتقوله هاتلى الموظفين الى تابعين للاقسام اما فى الليفت جوين بتقولهم هاتولى الموظفين التابعين و الغير تابعين لاقسام على ان يكون جدول الموظفين هو الى ناحيه الشمال لان الى ناحيه الشمال هو الى هيجى كله انما الى ناحيه اليمين هيجى المشترك بينه و بين الشمال بس

ادى كود بقه و هفهمهولك برضوا

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

var EmployeeByDeptWithLeft = Employee.GetAllEmployee().GroupJoin(Department.GetAllDepartment(),

E => E.DeptID,

D => D.DepartmentID,

(EMP, Dept) => new

{

EMP,

Dept

}

).SelectMany(z => z.Dept.DefaultIfEmpty(),

(Empl, Dep) => new

{

EmpObj = Empl.EMP,

DeptName = Dep == null ? " Noooooooooo Dept " : Dep.DepartmentName

}

);

foreach (var item in EmployeeByDeptWithLeft)

{

Console.WriteLine(" EmpName : "+ item.EmpObj.EmployeeName +

" EmpGender : " + item.EmpObj.EmployeeGender + " --> Dept : " + item.DeptName);

}

}

}

class Employee

{

public int EmployeeID { get; set; }

public string EmployeeName { get; set; }

public string EmployeeGender { get; set; }

public int DeptID { get; set; }

public static List<Employee> GetAllEmployee()

{

List<Employee> EmployeeList = new List<Employee>

{

new Employee { EmployeeID=1,EmployeeName="Mahmoud Badawy",EmployeeGender="Male",DeptID=1 },

new Employee { EmployeeID=1,EmployeeName="Mostafa Badawy",EmployeeGender="Male",DeptID=2 },

new Employee { EmployeeID=1,EmployeeName="Ahmed Sadek",EmployeeGender="Male",DeptID=1 },

new Employee { EmployeeID=1,EmployeeName="Noha Saber",EmployeeGender="Female",DeptID=1 },

new Employee { EmployeeID=1,EmployeeName="Kero Atef",EmployeeGender="Male",DeptID=2 },

new Employee { EmployeeID=1,EmployeeName="Sara Mahmoud",EmployeeGender="Female",DeptID=1 },

new Employee { EmployeeID=1,EmployeeName="Karim Elkerdasy",EmployeeGender="Male",DeptID=2 },

new Employee { EmployeeID=1,EmployeeName="Sayed Sanad",EmployeeGender="Male"}

};

return EmployeeList;

}

}

class Department

{

public int DepartmentID { get; set; }

public string DepartmentName { get; set; }

public static List<Department> GetAllDepartment()

{

List<Department> DepartmentList = new List<Department>

{

new Department {DepartmentID=1,DepartmentName="CS" },

new Department {DepartmentID=2,DepartmentName="IT" },

new Department {DepartmentID=3,DepartmentName="DS" }

};

return DepartmentList;

}

}

}

فى الكود ذى ما انت شايف سيد سند هو الوحيد الى مش موجود فى اى قسم و ده معناه انه مش مشترك مع الجدول بتاع الاقسام فى اى قسم و مع ذلك طلع فى النتايج بس طلع انه مش موجود فى اى قسم خالص

اذاى عملت الليفت جوين باللينك ؟؟

بص عملت جروب جوين و خليت الشمال هما الموظفين عشان يجيبهم كلهم و اليمين هما الاقسام عشان يجيب الى انا استخدمته بس و عشان كده مجبش القسم ال 3 الى انا مستخدمتوش اساسا و ربط بين الجدولين ب الفورن كى و البرايمرى كى و بعد كده قولتله انا عاوز شكل الحاجه الى هترجع تبقى على الشكل الفلانى الى هو عملتلها انونيموس فانكشن مخصوص بترجع اوبجيكتات الاقسام و الموظفين

و عشان اجيب حاجه ديفولت للاقسام فى حاله ان الموظف مش تبع اى قسم فعمنا حته السليكت مينى عشان معملش اتريشن تانى على الاقسام و عشان و برضوا عشان اعرفه ان ممكن الديبارت مينت يبقى ب Null و على فكره ممكن نعمل الكلام ده من غير ما نعمل

SelectMany بس هتحتاج ساعتها تعمل 2 اتريشنز ذى كده

class Program

{

public static void Main(string[] args)

{

var EmployeeByDeptWithLeft = Employee.GetAllEmployee().GroupJoin(Department.GetAllDepartment(),

E => E.DeptID,

D => D.DepartmentID,

(EMP, Dept) => new

{

EMP,

Dept

}

);

foreach (var item in EmployeeByDeptWithLeft)

{

Console.WriteLine(" EmpName : "+ item.EMP.EmployeeName +

" EmpGender : " + item.EMP.EmployeeGender);

foreach (var Elem in item.Dept)

{

Console.WriteLine(" Dept : " + Elem.DepartmentName);

}

}

}

}



MB

################################################################

Cross join

-----------------

هيجيب كل البيانات الى فى الجدولين سواء المشترك او الغير مشترك من الجدول اليمين و الشمال يعنى هيخليلك الموظف الواحد فى كل الاقسام عادى ههههههه و الى مش فى اى قسم هيبقى فى كل الاقسام برضوا عادى :D و بيرشق كله فى كله و طبعا استحاله تعمل العك ده فى الحقيقه

الكود اهه يا كبير

public static void Main(string[] args)

{

var EmployeeByDeptWithLeft = Employee.GetAllEmployee().Join(Department.GetAllDepartment(),

E => true,

D => true,

(EMP, Dept) => new

{

EMP,

Dept

});

foreach (var item in EmployeeByDeptWithLeft)

{

Console.WriteLine(" EmpName : "+ item.EMP.EmployeeName +

" EmpGender : " + item.EMP.EmployeeGender + " DeptName : " + item.Dept.DepartmentName );

}

}

و لو هتعملها بالسلكيت مينى هتبقى بالشكل ده

var EmployeeByDeptWithLeft = Employee.GetAllEmployee().SelectMany(E=>Department.GetAllDepartment(),

(EMP, Dept) => new

{

EMP,

Dept

});

MB

################################################################

Distinct and How to Implement Equality with Complex DataType

-------------------------------------------------------------------------------------------------

دى عشان العناصر المتشابهه متتكررش

هنا مصر متكرره 3 مرات شبه بعض بالحته فهياخد منهم واحده و مش هياخد الباقى اما بالنسبه لامريكا فالحروف مره كابتل و مره سمول فاعتبرهم مختلفين و جاب ال 3 أمريكا

public static void Main(string[] args)

{

String[] Countries = { "USA", "uSa", "usa", "Egypt", "Egypt", "Egypt" };

var DistinctCountries = Countries.Distinct();

foreach (var item in DistinctCountries)

{

Console.WriteLine(item);

}

}

طيب هنا ايه الحل ؟؟

الحل انه عشان يشوف امريكا كلهم ذى بعض انى استخدم الاسترينج كومبيرار داخل الديستينكت

var DistinctCountries = Countries.Distinct(StringComparer.InvariantCultureIgnoreCase);

طيب هنا الكلام ده نفع عشان دى سترينج مع سترينج طب المفروض ان الطبيعى بقه هيبعتولى اوبجيكت مع اوبجيكت دول بقه احل مشكلتهم اذاى ؟؟

بص يا سيدى فى الحاله دى الاول هيبقى الكود بالشكل ده و طبعا هيطلع كله مع بعضه من غير ما يشيل المتكرر لانه هو بالنسباله حاليا هو بيقارن الريفرنسيس بالريفرنس فبيلاقيه مختلف فبيجيب الاوبجيكت بالرغم من ان القيم متكرره فعلا

public static void Main(string[] args)

{

List<Person> Per = new List<Person>

{

new Person { ID = 1,Name = "Mahmoud"},

new Person { ID = 1,Name = "Mahmoud"},

new Person { ID = 1,Name = "Mahmoud"},

};

var perDistinct = Per.Distinct();

foreach (var item in perDistinct)

{

Console.WriteLine("ID : "+ item.ID + " --> Name : " + item.Name);

}

}

طيب عشان احل المشكله دى اعلم ايه ؟؟

اول حل اعمل انونيموس تايب داخل الديستينكت يبقى بنفس الاستراكتشار بتاع الاوبجيكت الى هتعامل معاه و ده الحل الاول بيبقى كده

public static void Main(string[] args)

{

List<Person> Per = new List<Person>

{

new Person { ID = 1,Name = "Mahmoud"},

new Person { ID = 1,Name = "Mahmoud"},

new Person { ID = 1,Name = "Mahmoud"},

};

var perDistinct = Per.Select(PerObj=>new{PerObj.ID, PerObj.Name}).Distinct();

foreach (var item in perDistinct)

{

Console.WriteLine("ID : "+ item.ID + " --> Name : " + item.Name);

}

}

الحل التانى انى اعمل اوفر رايد للفانكشن Equal و GetHashCode داخل الكلاس صاحب الاوبجكيت الى بتعامل معاه و هيبقى كده

class Person

{

public int ID { get; set; }

public string Name { get; set; }

public override bool Equals(object obj)

{

return this.ID == ((Person)obj).ID && this.Name == ((Person)obj).Name;

}

public override int GetHashCode()

{

return this.ID.GetHashCode() ^ this.Name.GetHashCode();

}

}

الحل التالت بقه انى اعمل كلاس تانى خالص و اعمله انه بيشتغل على الانترفيس بتاع ال IEqualityComparer و اعمل جواه امبليمنتيشن للفانكشنز Equal , GethashCode و من ثم استخدم اوبجيكت من الكلاس ده داخل الديستينكت فالكود هيبقى كده

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

List<Person> Per = new List<Person>

{

new Person { ID = 1,Name = "Mahmoud"},

new Person { ID = 1,Name = "Mahmoud"},

new Person { ID = 1,Name = "Mahmoud"},

};

var perDistinct = Per.Distinct(new personEquality());

foreach (var item in perDistinct)

{

Console.WriteLine("ID : "+ item.ID + " --> Name : " + item.Name);

}

}

}

class Person

{

public int ID { get; set; }

public string Name { get; set; }

}

class personEquality : IEqualityComparer<Person>

{

public bool Equals(Person x , Person y)

{

return x.ID == y.ID && x.Name == y.Name;

}

public int GetHashCode(Person z)

{

return z.ID.GetHashCode() ^z.Name.GetHashCode();

}

}

}

MB

################################################################

union

-----------------

ده بياخد اتنين من الكوليكشنز و يدمجهم مع بعض و يشيل التكرار و كذلك بيدمج بين جدولين و بيشيل التكرار بس لازم طبعا اظبط ساعتها الاكوال و الجيت هاش كود او اعمله انونيموس سليكت عشان يعرف المتساوى و يشسل المتكرر

اولا المثال العادى

public static void Main(string[] args)

{

string[] First = { "Egypt", "KSA", "France", "USA" };

string[] sec = { "Sudan", "Germany", "Egypt", "Egypt", "Egypt", "Egypt", "ksa", "France", "usa" };

var UnionCol = First.Union(sec);

foreach (var item in UnionCol)

{

Console.WriteLine(item);

}

}

من غير الاسترينج كومبيرار هيشيل المتكرر المتطابق فقط انما لو حطينا الاسترينج كومبيرار هيشيل حتى المتكرر و هو كابتل و سمول و الكلام ده و هكتب الكومبيرار فى اليونيون كده

var UnionCol = First.Union(sec,StringComparer.OrdinalIgnoreCase);

كده مستحيل يتكرر

طيب فى حاله الكومبليكس داتا تايب بقه هكتب الكود اذاى ؟؟ عشان المتكرر يتحذف لازم اعمله حلول الاكوالتى و هناانا اخترت اعمله الانونيموس فانكشن الى فى السليكت عشان اريح دماغى و ممكن برضوا تعمل اوفر رايد و الكلام ده هينفع

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

List<Person> Per1 = new List<Person>

{

new Person { ID = 1,Name = "Mahmoud"},

new Person { ID = 2,Name = "Mostafa"},

new Person { ID = 3,Name = "Sayed"},

};

List<Person> Per2= new List<Person>

{

new Person { ID = 1,Name = "Mahmoud"},

new Person { ID = 2,Name = "Mostafa"},

new Person { ID = 3,Name = "Mesho"},

};

var per1AfterEquality = Per1.Select(Per1Obj => new { ID = Per1Obj.ID, Name = Per1Obj.Name });

var per2AfterEquality = Per2.Select(Per2Obj => new { ID = Per2Obj.ID, Name = Per2Obj.Name });

var perUnion = per1AfterEquality.Union(per2AfterEquality);

foreach (var item in perUnion)

{

Console.WriteLine("ID : "+ item.ID + " --> Name : " + item.Name);

}

}

}

class Person

{

public int ID { get; set; }

public string Name { get; set; }

}

}

################################################################

Intersect

-----------------

بتجيب المشترك بين الحاجات بس --< المتشابه يعنى و الغير متشابه بترميه مبتجبهوش و بتتكتب بنفس طريقه اليونيون

################################################################

Except

-----------------

عكس الانترسيكت يعنى من الاخر بتجيب المختلف و بترمى المتشابه بين ال2 كوليكشن او الجدولين او الاوبجكتين :D

################################################################

Range

-----------------

دى بتديها رينج معين و هى بتعمل جينيريت للارقام دى يعنى انت هتقوله اعملى جينيريت للارقام من 1 لحد 100 مثلا هيتكتب كده

IEnumerable<int> num = Enumerable.Range(1, 100);

foreach (var item in num)

{

Console.WriteLine(item);

}

طيب لو عاوز الارقام الزوجيه بس من الرينج ده تتكتب اذاى ؟

IEnumerable<int> num = Enumerable.Range(1, 100).Where(x=>x%2==0);

MB

################################################################

Repeat

------------------------

بتخلى الحاجه تتكرر بالعدد الى انا عاوزه

IEnumerable<string> num = Enumerable.Repeat("Hello",5);

foreach (var item in num)

{

Console.WriteLine(item);

}

MB

################################################################

Empty

------------------------

دى بعملها لما اكون شاكك ان فى حاجه هترجع نول و هتتسبب فى ان الكود يبوظ فبقوله لو هيجرع null حطلها الامبتى احسن عشان متضربش

public static void Main(string[] args)

{

IEnumerable<string> Data = GetData() ?? Enumerable.Empty<string>();

foreach (var item in Data)

{

Console.WriteLine(item);

}

}

public static IEnumerable<string> GetData()

{

//return new List<string> { "mesho","Mezo"};

return null;

}

MB

################################################################

Concat Vs Union

------------------------

ألكونكاتيونيشن بيضيف 2 كوليكشن على بعض من غير ما يشيل المشترك

اليونيون بيضيف 2 كوليكشن على بعض بعد ما بيشيل المشترك بينهم و ادى مثال اهه الكونكاتيونيشن هتطلع النتيجه كده

1 2 3 4 1 2 3 4 5 6 بينما اليونييون هيطلع النتيجه كده 1 2 3 4 5 6 لانه هيشيل المتكرر

public static void Main(string[] args)

{

int[] num1 = { 1, 2, 3, 4 };

int[] num2 = { 1, 2, 3, 4 ,5,6};

var Data = num1.Concat(num2);

foreach (var item in Data)

{

Console.WriteLine(item);

}

}

الكود باليونيون بقه هو هواه بس التغيير فى الكلمه بس

var Data = num1.Union(num2);

MB

################################################################

SequenceEqual

------------------------

ده بيشوف هل الاتنين كوليكشن متطابقين فى الترتيب و الشكل و عدد العناصر الى جواهم و هل العناصر هى هى ولا لأ من الاخر لازم الاتنين يبقوا حاجه واحده :D

ذى كده مثلا فى الحاله دى هترجع **TRUE**

public static void Main(string[] args)

{

int[] num1 = { 1, 2, 3, 4 };

int[] num2 = { 1, 2, 3, 4 };

bool ISSequenceEqual = num1.SequenceEqual(num2);

Console.WriteLine(ISSequenceEqual); //TRUE

}

اما فى الحاله التانيه دى هرتجع **FALSE**

مع ان نفس العناصر بس ترتيبهم مختلف

public static void Main(string[] args)

{

int[] num1 = { 1, 2, 3, 4 };

int[] num2 = { 1, 2, 4, 3 };

bool ISSequenceEqual = num1.SequenceEqual(num2);

Console.WriteLine(ISSequenceEqual); //FALSE

}

طيب عشان احل المشكله بتاعت الترتيب دى اعمل ايه ؟؟ اعمل للاتنين اورد باى عشان يبقوا نفس الترتيب

public static void Main(string[] args)

{

int[] num1 = { 1, 2, 3, 4 };

int[] num2 = { 1, 2, 4, 3 };

bool ISSequenceEqual = num1.OrderBy(X=>X).SequenceEqual(num2.OrderBy(X => X));

Console.WriteLine(ISSequenceEqual);

}

طيب لو الاتنين كوليكشن دول سترينج و فى منهم كابتل و سمول هيبقى المشكله هنا انهم مش متطابقني فى الكتابه هتحل المشكله اذاى ؟

public static void Main(string[] args)

{

string[] con1 = { "EGYPT", "RUSSIA", "FRANCE" };

string[] con2 = { "russia", "france", "egypt" };

bool ISSequenceEqual = con1.OrderBy(X=>X).SequenceEqual(con2.OrderBy(X => X),StringComparer.OrdinalIgnoreCase);

Console.WriteLine(ISSequenceEqual);

}

لما يكون بقه فى complex datatype انت عارف هتعمل فيها ايه بقه ليها 3 حلول

MB

################################################################

All

------------------------

دى بتشوف هل كل العناصر الى موجود فى الكوليكشن بتحقق الشرط الى جواها و لا لأ لو كل العناصر بتحقق الشرط بترجع true غير كده هترجع false

هل كل العناصر اصغر من 20 فى المثال ده ؟؟ ايوه فكده هترجع true

public static void Main(string[] args)

{

int[] num = { 1, 2, 3 ,4,5,6,7,8,9,10};

bool IS\_OK = num.All((x) => (x < 20));

Console.WriteLine(IS\_OK);

}

طب لو فى عنصر واحد ازيد من عشرين هترجع false

int[] num = { 50,1, 2, 3 ,4,5,6,7,8,9,10};

bool IS\_OK = num.All((x) => (x < 20));

Console.WriteLine(IS\_OK);

MB

################################################################

any

------------------------

ده بقه مجرد ما عنصر واحد حقق الشرط بيروح مرجع true

على سبيل المثال ده مثلا

int[] num = { 1,50,51,52};

bool IS\_OK = num.Any((x) => (x < 20)); //true

Console.WriteLine(IS\_OK);

ذى ما انت شايف عنصر واحد حقق الشرط راح مرجعلك ture

عارف برضوا لو مفيش شروط خالص هيرجع true طالما ان الكوليكشن فيها عناصر و فى حاله انها مفيهاش عناصر خالص هيرجع false

int[] num = {};

bool IS\_OK = num.Any(); //false

Console.WriteLine(IS\_OK);

لو حطينا بقه عنصر واحد معفن من غير اى شروط اهه هيرجع ture

int[] num = {1};

bool IS\_OK = num.Any(); //true

Console.WriteLine(IS\_OK);

MB

################################################################

Contains

------------------------

بيشوف هل الكوليكشن بيحتوى على العنصر الى انا محددهوله فى الكونتينس ولا لأ يعنى كانى بعمل سيرش بالظبط كده على حاجه معينه بقوله مثلا هل مصر موجوده فى الكوليكشن ده ولا لأ ؟؟

public static void Main(string[] args)

{

string[] con = { "EGYPT", "RUSSIA", "FRANCE" };

bool IS\_OK = con.Contains("Egypt",StringComparer.OrdinalIgnoreCase); //true

Console.WriteLine(IS\_OK);

}

MB

################################################################

LinqPad

------------------------

دى يا سيدى اداه بجرب فيها اللينك و بتسهل عليه كتابه اللينك و ممكن كمان اوصلها بداتا بيز بشكل مباشر و اعيش معاها و فيها كمان اختصارات اكواد كتير هتسهل عليك التعامل مع اللينك بشكل حلو فشخ و ادى لينك الاداه لو حبيت تحملها

<https://www.linqpad.net/Download.aspx>

و اللينك ده تبصلك بصه على الموقع هتفهم لازمه الاداه هى الصراحه اداه جباره

<https://www.linqpad.net/>

MB

################################################################

Mahmoud badawy

------------------------