**对 Join 子句的结果进行排序（C# 编程指南）**

此示例演示如何对联接运算的结果进行排序。请注意，排序是在联接之后执行的。尽管可以在联接之前将 **orderby** 子句用于一个或多个源序列，但通常我们不建议这样做。某些 LINQ 提供程序可能不会在联接之后保留排序。

此查询将创建一个分组联接，然后基于类别元素（仍然在范围中）对组进行排序。在匿名类型初始值设定项中，子查询将对产品序列中的所有匹配元素进行排序。

class HowToOrderJoins

{

#region Data

class Product

{

public string Name { get; set; }

public int CategoryID { get; set; }

}

class Category

{

public string Name { get; set; }

public int ID { get; set; }

}

// Specify the first data source.

List<Category> categories = new List<Category>()

{

new Category(){Name="Beverages", ID=001},

new Category(){ Name="Condiments", ID=002},

new Category(){ Name="Vegetables", ID=003},

new Category() { Name="Grains", ID=004},

new Category() { Name="Fruit", ID=005}

};

// Specify the second data source.

List<Product> products = new List<Product>()

{

new Product{Name="Cola", CategoryID=001},

new Product{Name="Tea", CategoryID=001},

new Product{Name="Mustard", CategoryID=002},

new Product{Name="Pickles", CategoryID=002},

new Product{Name="Carrots", CategoryID=003},

new Product{Name="Bok Choy", CategoryID=003},

new Product{Name="Peaches", CategoryID=005},

new Product{Name="Melons", CategoryID=005},

};

#endregion

static void Main()

{

HowToOrderJoins app = new HowToOrderJoins();

app.OrderJoin1();

// Keep console window open in debug mode.

Console.WriteLine("Press any key to exit.");

Console.ReadKey();

}

void OrderJoin1()

{

var groupJoinQuery2 =

from category in categories

join prod in products on category.ID equals prod.CategoryID into prodGroup

orderby category.Name

select new

{

Category = category.Name,

Products = from prod2 in prodGroup

orderby prod2.Name

select prod2

};

foreach (var productGroup in groupJoinQuery2)

{

Console.WriteLine(productGroup.Category);

foreach (var prodItem in productGroup.Products)

{

Console.WriteLine(" {0,-10} {1}", prodItem.Name, prodItem.CategoryID);

}

}

}

/\* Output:

Beverages

Cola 1

Tea 1

Condiments

Mustard 2

Pickles 2

Fruit

Melons 5

Peaches 5

Grains

Vegetables

Bok Choy 3

Carrots 3

\*/

}