

convert this LINQ expression into Lambda

Asked 10 years, 2 months ago Active 6 years, 4 months ago Viewed 30k times

Guys, I have a hard time converting this below linq expression(left join implementation) to lambda expression (for learning).

14

```
var result = from g in grocery
              join f in fruit on g.fruitId equals f.fruitId into tempFruit
              join v in veggie on g.vegid equals v.vegid into tempVegg
              from joinedFruit in tempFruit.DefaultIfEmpty()
              from joinedVegg in tempVegg.DefaultIfEmpty()
              select new { g.fruitId, g.vegid, fname = ((joinedFruit == null) ? string.Empty :
              joinedFruit.fname), vname = ((joinedVegg == null) ? string.Empty : joinedVegg.vname) };
```

Can some one suggest me how to do this.

And i really appreciate if someone give me the excellent tutorial links for "**C# Lambdas & Linqs**"

c#

linq

lambda

edited Oct 6 '09 at 10:57

asked Oct 6 '09 at 10:36



RameshVel

55.4k

24

155

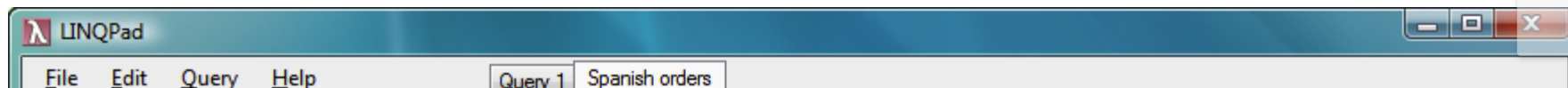
204

7 Answers

To convert a Linq query to it's Lambda equivalent:

27

1. Download [Linqpad](#) and run your query.
2. In the results window, click on the "λ" button in the toolbar. It's right above the Results window
3. Your query will be converted to a Lambda expression equivalent!



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Server Explorer' pane displays the 'NORTHWND' database schema, including tables like 'Categories', 'Contacts', 'CustomerCustomerDemos', 'CustomerDemographics', 'Customers', 'Employees', 'EmployeeTerritories', 'OrderDetails', and 'Orders'. The 'OrderDetails' table is selected. In the center, the 'Query Designer' pane shows a LINQ query for the 'Products' table, filtered by 'ShipCountry == "Spain"'. The query is written in C# Expression syntax. On the right, the 'Results' pane shows the output of the query, which is a grouped list of products by category. A red box highlights the 'Convert to Lambda' button, with an arrow pointing to it from a text box that says 'This is the button you're looking for (Convert to Lambda)'.

Query:

```
from p in Products
let spanishOrders = p.OrderDetails.Where (o => o.Order.ShipCountry == "Spain")
where spanishOrders.Any()
orderby p.Category.CategoryName, p.ProductName
group new
{
    p.ProductName,
    Orders = spanishOrders.Count(),
    TotalValue = spanishOrders.Sum (o => o.UnitPrice * o.Quantity)
}
by p.Category.CategoryName
```

Results:

Key=Beverages

ProductName	Orders	TotalValue
Chai	1	180.000000
Chang	1	380.000000
Guaraná Fantástica	2	75.600000
Lakkalikööri	1	90.000000
Laughing Lumberjack Lager	1	210.000000
Outback Lager	1	60.000000
Rhönbräu Klosterbier	1	310.000000
Steeleye Stout	1	57.600000

Key=Condiments

ProductName	Orders	TotalValue
Chef Anton's Cajun Seasoning	1	422.400000
Louisiana Fiery Hot Pepper Sauce	3	1030.050000
Original Frankfurter grüne Soße	1	52.000000
Sirop d'érable	1	285.000000

Query successful 00:00:00

edited Jul 19 '13 at 15:25

answered Jul 7 '11 at 14:40



Brad Parks

44.2k

44

191

285

1 I have Linqpad but without creating a 'connection' to a service that I can query, I can not see the button for the lambda. I do not know how to just paste in a LINQ query and convert it to a lambda. Is this actually possible? – [atconway](#) Jul 19 '13 at 15:15

I just added a pic for where the button is in the UI. I don't have Linqpad on hand, but I think you need to have a runnable query there before it can be converted to a Lambda. In another post, I detailed how you can test data in Linqpad without needing a DB, here: stackoverflow.com/questions/4611031/... – [Brad Parks](#) Jul 19 '13 at 15:31

can it convert SQL to lambda or LINQ? Doesn't seem to be working – [Toolkit](#) Nov 7 '18 at 11:39

I haven't used it in a long time, but what does clicking the SQL button beside the λ do? – [Brad Parks](#) Nov 7 '18 at 11:52

You can take a look at [101 LINQ Samples](#) and [C# 3.0 QUERY EXPRESSION TRANSLATION CHEAT SHEET](#)

answered Oct 6 '09 at 10:39



Dzmitry Huba

4,399

17

19

1 thanks for the links Dzmitry... the cheat sheet is useful... i already have a look at 101 samples.. – [RameshVel](#) Oct 6 '09 at 10:46

hey where would i get the source code for the 101 samples.. any idea.. i couldnt find there... – [RameshVel](#) Oct 6 '09 at 11:49

Here's the heuristic that I follow:

10

Favor LINQ expressions over lambdas when you have joins.

I think that lambdas with joins look messy and are difficult to read.

answered Oct 6 '09 at 10:42



Jim G.

13.1k

17

84

158

thanks jim for the reply. But i wanted to do this in lambda to get familiar with.. i mentioned that already its for learning... – [RameshVel](#) Oct 6 '09 at

10:44

I usually use ReSharper to help me convert things to method chains and lambda's, which helps me go back and forth fairly easy.

8

```
var result = from g in grocery
              join f in fruit on g.fruitId equals f.fruitId into tempFruit
              join v in veggie on g.vegid equals v.vegid into tempVegg
              from joinedFruit in tempFruit.DefaultIfEmpty()
              from joinedVegg in tempVegg.DefaultIfEmpty()
              select new { g.fruitId, g.vegid, fname = ((joinedFruit == null) ?
string.Empty : joinedFruit.fname), vname = ((joinedVegg == null) ? string.Empty :
joinedVegg.vname) };
```

And then using ReSharper's option of convert LINQ to method chain equals the following:

```
var result = grocery .GroupJoin(fruit, g => g.fruitId, f => f.fruitId, (g,
tempFruit) => new {g, tempFruit})
                    .GroupJoin(veggie, @t => @t.g.vegid, v => v.vegid, (@t,
tempVegg) => new {@t, tempVegg})
                    .SelectMany(@t => @t.@t.tempFruit.DefaultIfEmpty(), (@t,
joinedFruit) => new {@t, joinedFruit})
                    .SelectMany(@t => @t.@t.tempVegg.DefaultIfEmpty(), (@t,
joinedVegg) =>
    new
    {
        @t.@t.@t.g.fruitId,
        @t.@t.@t.g.vegid,
        fname = ((@t.joinedFruit == null) ? string.Empty
: @t.joinedFruit.fname),
        vname = ((joinedVegg == null) ? string.Empty :
joinedVegg.vname)
    });
```

Granted the output is less then desirable, but It at least helps in starting somewhere on understanding the syntax.

answered Oct 6 '09 at 12:39



Mark Coleman

37.6k 8 77 102

thanks Mark for your response.. i will run it over here and will let you know if that works.. – [RameshVel](#) Oct 6 '09 at 13:24

Here's how you might write this query in lambda:

4

```
var customers = new List {  
    new Customer { CompanyId = "AC", CustomerId = "Customer1" },  
    new Customer { CompanyId = "not-AC", CustomerId = "Customer2" },  
};  
  
var userCustomers = new List {  
    new UserCustomer { CompanyId = "AC", CustomerId = "Customer1", User = "not-admin" },  
    new UserCustomer { CompanyId = "AC", CustomerId = "Customer1", User = "admin" },  
    new UserCustomer { CompanyId = "AC", CustomerId = "Customer2", User = "not-admin" },  
    new UserCustomer { CompanyId = "AC", CustomerId = "Customer2", User = "admin" },  
    new UserCustomer { CompanyId = "not-AC", CustomerId = "Customer1", User = "not-admin" },  
    new UserCustomer { CompanyId = "not-AC", CustomerId = "Customer1", User = "admin" },  
    new UserCustomer { CompanyId = "not-AC", CustomerId = "Customer2", User = "not-admin" },  
    new UserCustomer { CompanyId = "not-AC", CustomerId = "Customer2", User = "admin" },  
};
```

Using query expression

```
var query =  
from c in customers  
join uc in userCustomers on  
new { c.CompanyId, c.CustomerId } equals new { uc.CompanyId, uc.CustomerId }  
where c.CompanyId == "AC" && uc.User == "admin"  
select c;
```

Using lambda expressions

```
var lambda = customers.Where(c => c.CompanyId == "AC") // inner sequence  
    .Join(userCustomers.Where(uc => uc.User == "admin"), // outer sequence  
c => new { c.CompanyId, c.CustomerId }, // inner key selector  
uc => new { uc.CompanyId, uc.CustomerId }, // outer key selector  
(c, uc) => c);
```

Both approach yields the same result (customer with company Id “AC” and customer Id “Customer1”), but as you can see, lambda expression is much harder to write and read!

Hope this helps!

edited Apr 13 '12 at 5:27



menjaraz

7,339 4 34 76

answered Apr 13 '12 at 5:04



Sunandan Dutt

47 3

Download [LINQPad](#); it comes with built-in samples for learning LINQ.

3

answered Oct 6 '09 at 10:40



Mitch Wheat

266k 37 421 507

i thought that was only the paid version that had loads of samples etc – [Andrew](#) Oct 6 '09 at 10:53

The free version comes with samples too. – [Mitch Wheat](#) Oct 6 '09 at 11:18

Use Reflector .NET :)

3

answered Oct 6 '09 at 10:41



leppie

103k 16 178 287