

# Digging Deeper into LINQ to Entities

Exploring various ways to query with LINQ to Entities as well as taking entity relationships into account

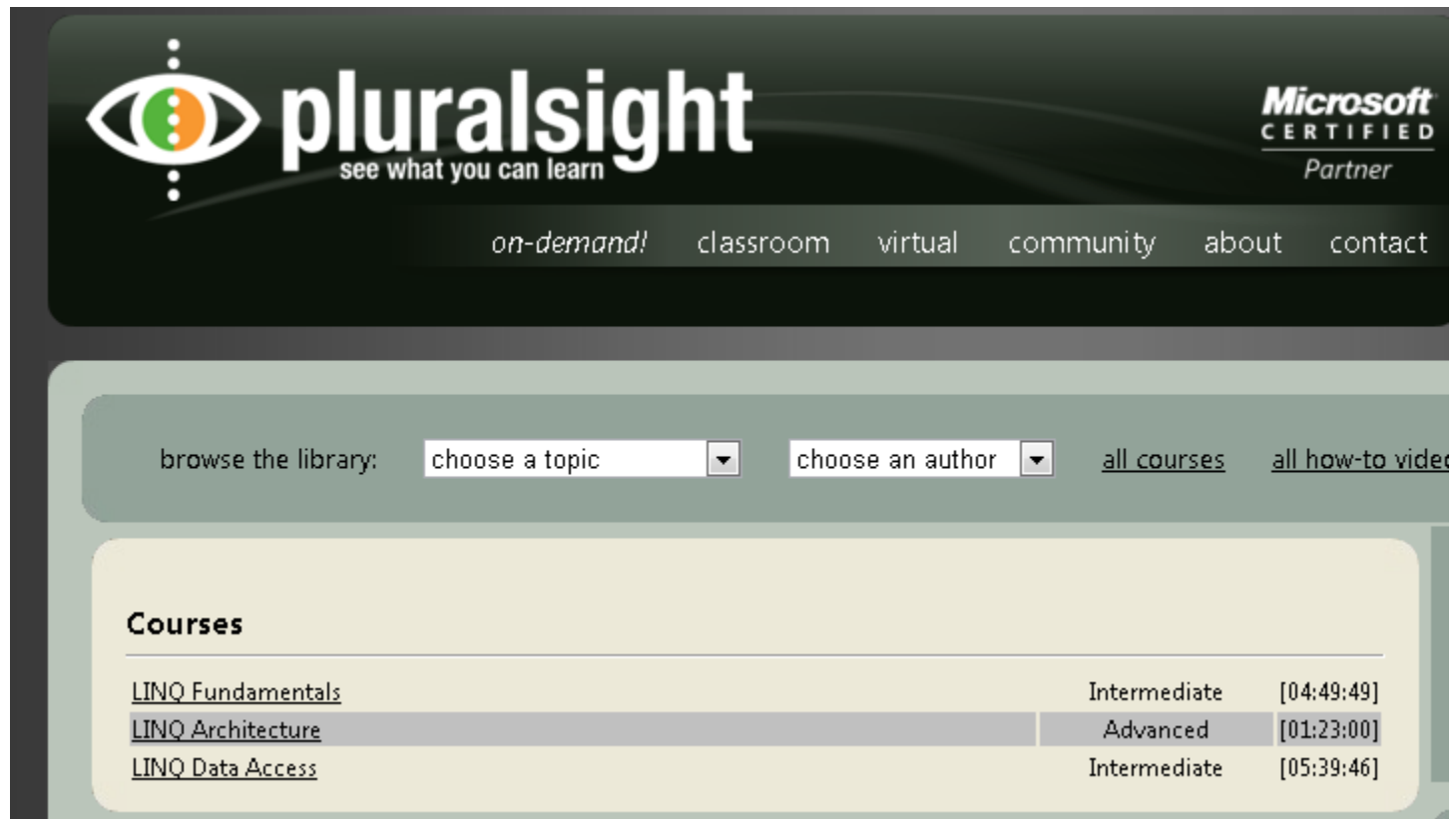


# WHY LINQ TO ENTITIES?

# Objectives

- Projecting properties in queries
- Querying with navigation properties
- Collection Logic & Aggregates
- Joins and Nested Queries
- Shaping results with Eager, Lazy & Explicit loading
- Grouping
- Retrieving a Single Entity
- Gotchas to watch out for

# Pluralsight On-Demand LINQ Courses



The image shows a screenshot of the Pluralsight website. The header features the Pluralsight logo (an eye icon with a green and orange dot) and the tagline "see what you can learn". To the right is the "Microsoft CERTIFIED Partner" logo. Below the header is a navigation bar with links: "on-demand!", "classroom", "virtual", "community", "about", and "contact".

Below the navigation bar is a search section with the text "browse the library:" followed by two dropdown menus labeled "choose a topic" and "choose an author". To the right of these are links for "all courses" and "all how-to videos".

Below the search section is a "Courses" section with a table listing LINQ courses. The table has three columns: course name, level, and duration. The course "LINQ Architecture" is highlighted with a grey background.

Courses		
<a href="#">LINQ Fundamentals</a>	Intermediate	[04:49:49]
<a href="#">LINQ Architecture</a>	Advanced	[01:23:00]
<a href="#">LINQ Data Access</a>	Intermediate	[05:39:46]

# Simple Query Projections

## C#

```
from c in context.Customers  
select new MiniCustomer {c.FirstName,c.LastName, c.ID}
```

## VB

```
From c In context.Customers  
Select New MiniCustomer With {c.FirstName, c.LastName, c.ID}
```

Project to:

- Anonymous type
- Class
- *NOT to struc*

# Using Navigation Properties in Queries

**C#**

```
from o in context.Orders  
where o.Customer.Country=="UK"  
select o
```

```
from o in context.Orders  
orderby o.Customer.CompanyName  
select new {o.Total,o.Customer.CompanyName}
```

```
from c in context.Customers  
select new {c.CompanyName, c.Orders}
```

# EntityCollection Logic in Queries

```
from c in context.Customers  
select new {c.CompanyName, c.Orders}
```

```
from c in context.Customers  
where c.Orders.Sum(o=>o.SubTotal)>10000  
select c;
```

```
from c in context.Customers  
where c.Orders.Any()  
select c;
```

# Nested Queries and Joins

## JOIN

```
FROM c IN context.Customers  
JOIN o IN context.Orders  
ON c.CustomerID EQUALS o.CustomerID  
SELECT .....
```



*No Order.Customer  
navigation property*

## NESTING

```
from c in context.Customers  
from o in c.Orders . . .
```

## NESTING

```
var universe=from c in context.Customers where [complex filter]  
var queryA=from c in universe ...  
var queryB=from o in universe.Orders . . .
```



# Getting Related Data

## Eager Loading:

```
from c in context.Customers.Include("Orders") select c  
from o in context.Orders.Include("Customer.Address) select o
```

## Explicit & Lazy Loading:

```
var cust=(from c in cc      ↙ lazy load      select c).First();  
cust.Orders.Load(); ↙ explicit load  
var orderCount=cust.Orders.Count();
```

## Projection:

```
from c in context.Customers  
select new {c, BigOrders=c.Orders.Where(o=>o.Total>1000)}
```

# Grouping

## C#

```
from o in context.Orders  
group o by o.[PROPERTY OR EXPRESSION]  
into GROUPVARIABLE  
select GROUPVARIABLE [or projection]
```

## VB

```
From o In context.Orders  
Group o By o.[PROPERTY OR EXPRESSION]  
Into GROUPVARIABLE = Group  
Select GROUPVARIABLE [or projection]
```

Group key

item

item

item

Group key

item

Group key

item

item

Group key

item

# Getting Single Entities

`[query].Single/SingleOrDefault` *Looking for a unique item*

`[query].First/FirstOrDefault` *First of possibly many*

`ObjectContext.GetObjectByKey(EntityKey)`

# A Few LINQ to Entities Gotchas

- **Client Side processing vs. Server Side Processing**
  - Is it a LINQ to Entities or LINQ to Objects method?
- **Unsupported .NET methods**
  - Caught at runtime, not compile time
- **Custom Properties can't be used in query**
  - Won't be caught until runtime
- **Careful about loading related data in disconnected scenarios**
- **Profile, profile, profile!**

# Resources

- **Pluralsight On Demand**
  - <http://www.pluralsight.com>
- **Programming Entity Framework, 2<sup>nd</sup> Edition, Julie Lerman**
  - O'Reilly Media, August 2010
  - <http://www.learnentityframework.com>
- **Query Profiling Options for Entity Framework**
  - December 2010 MSDN Magazine Data Points Column
    - [msdn.com/magazine](http://msdn.com/magazine)
- **Microsoft Data Developer Center**
  - <http://msdn.com/data>