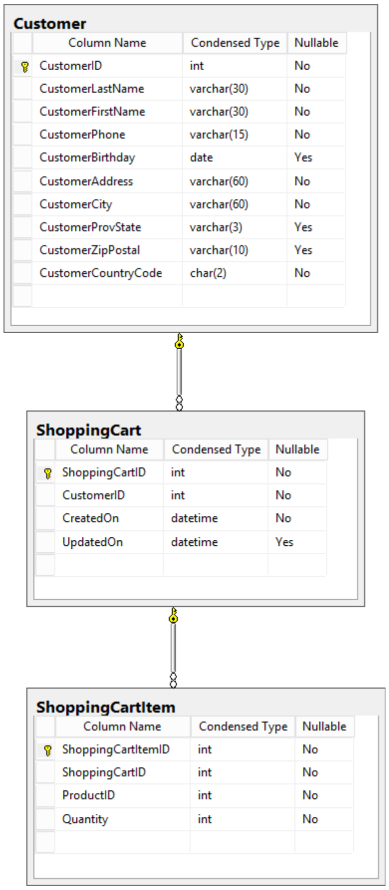
1.3.0 – LINQ Advanced

Introduction

In lesson 1.2.0 you were introduced to creating LINQ queries in LINQPad. In this lesson, you will create some advance LINQ queries in LINQPad. These queries will return data that contains a collection of data, and other advanced LINQ functions.

Example #1

Consider the following portion of the eStore ERD:



Customer

Column Name	Condensed Type	Nullable
CustomerID	int	No
CustomerLastName	varchar(30)	No
CustomerFirstName	varchar(30)	No
CustomerPhone	varchar(15)	No
CustomerBirthday	date	Yes
CustomerAddress	varchar(60)	No
CustomerCity	varchar(60)	No
CustomerProvState	varchar(3)	Yes
CustomerZipPostal	varchar(10)	Yes
CustomerCountryCode	char(2)	No

ShoppingCart

Column Name	Condensed Type	Nullable
ShoppingCartID	int	No
CustomerID	int	No
CreatedOn	datetime	No
UpdatedOn	datetime	Yes

ShoppingCartItem

Column Name	Condensed Type	Nullable
ShoppingCartItemID	int	No
ShoppingCartID	int	No
ProductID	int	No
Quantity	int	No

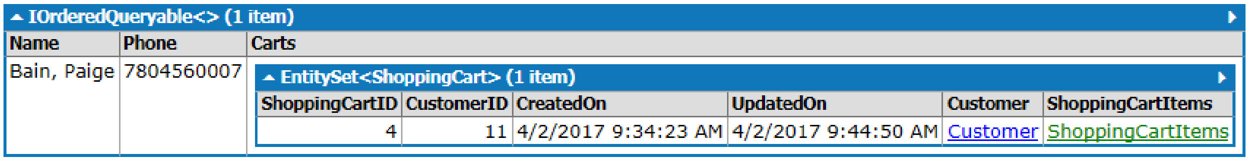
Figure 1: Customer - ShoppingCart - ShoppingCartItem

If we wanted to return the ShoppingCarts that belong to a selected Customer we could write the following query:

Listing 1: Customer Shopping Carts (1)

```
var results = from x in Customers
              where x.CustomerID == 11
              orderby x.LastName, x.FirstName
                      select new
                      {
                            Name = x.LastName + ", " + x.FirstName,
                            Phone = x.Phone,
                            Carts = x.ShoppingCarts
                     };
results.Dump();
```

The result of this query is:



IOrderedQueryable<> (1 item)

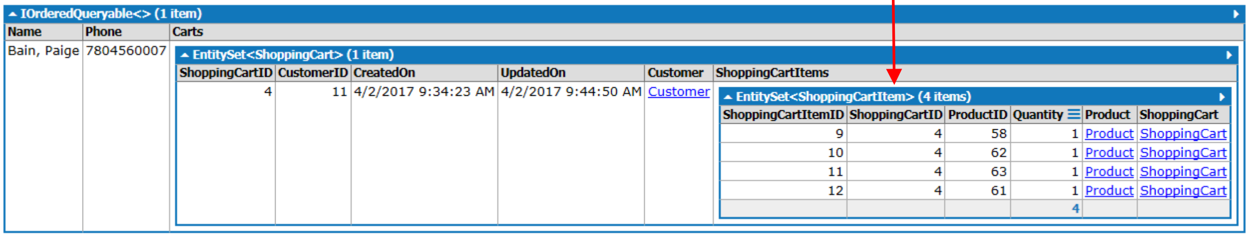
Name	Phone	Carts
Bain, Paige	7804560007	EntitySet<ShoppingCart> (1 item)

EntitySet<ShoppingCart> (1 item)

ShoppingCartID	CustomerID	CreatedOn	UpdatedOn	Customer	ShoppingCartItems
4	11	4/2/2017 9:34:23 AM	4/2/2017 9:44:50 AM	Customer	ShoppingCartItems

Figure 2: Customer Shopping Carts Output (1)

The output displays an EntitySet<ShoppingCart>, which has ShoppingCartItems ShoppingCartItems. If we expand this we see:



IOrderedQueryable<> (1 item)

Name	Phone	Carts
Bain, Paige	7804560007	EntitySet<ShoppingCart> (1 item)

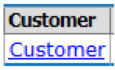
EntitySet<ShoppingCart> (1 item)

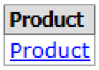
ShoppingCartID	CustomerID	CreatedOn	UpdatedOn	Customer	ShoppingCartItems
4	11	4/2/2017 9:34:23 AM	4/2/2017 9:44:50 AM	Customer	EntitySet<ShoppingCartItem> (4 items)

EntitySet<ShoppingCartItem> (4 items)

ShoppingCartItemID	ShoppingCartID	ProductID	Quantity	Product	ShoppingCart
9	4	58	1	Product	ShoppingCart
10	4	62	1	Product	ShoppingCart
11	4	63	1	Product	ShoppingCart
12	4	61	1	Product	ShoppingCart
			4		

Figure 3: Expanded ShoppingCartItems



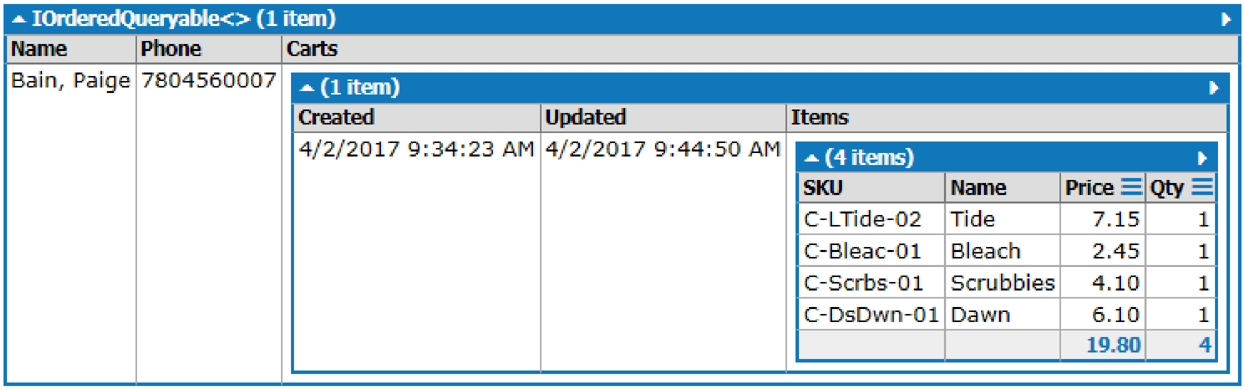


Also included in the output is Customer Customer, which we do not need. We will need Product Product in some part of the query.

The results shown are a result of LINQPad using the navigation properties, which are created from the Primary Key – Foreign Key relationships between the database tables.

Example #2

Let us refine the query from the previous example to show just the relevant information. Looking at the output below, what would be the query?



```
IOrderedQueryable<> (1 item)
Name         Phone       Carts
Bain, Paige  7804560007  (1 item)
                         Created               Updated               Items
                         4/2/2017 9:34:23 AM   4/2/2017 9:44:50 AM   (4 items)
                                                                     SKU         Name       Price  Qty
                                                                     C-LTide-02  Tide        7.15    1
                                                                     C-Bleac-01  Bleach      2.45    1
                                                                     C-Scrbs-01  Scrubbies   4.10    1
                                                                     C-DsDwn-01  Dawn        6.10    1
                                                                                            19.80    4
```

Figure 4: Example #2 Output

The query has two nested queries; one query for the ShoppingCart for the Customer, and the second for the ShoppingCartItems that are in the ShoppingCart:

Listing 2: Customer Shopping Carts (2)

```
var results = from x in Customers
              where x.CustomerID == 11
              orderby x.LastName, x.FirstName
              select new
              {
                  Name = x.LastName + ", " + x.FirstName,
                  Phone = x.Phone,
                  Carts = from y in x.ShoppingCarts
                          orderby y.UpdatedOn
                          select new
                          {
                              Created = y.CreatedOn,
                              Updated = y.UpdatedOn,
                              Items = from z in y.ShoppingCartItems
                                      select new
                                      {
                                          SKU = z.Product.SKU,
                                          Name = z.Product.ProductName,
                                          Price = z.Product.SellingPrice,
                                          Qty = z.Quantity
                                      }
                          }
              };
results.Dump();
```

Nested LINQ Query