Examples of the LINQ *OrderBy*, *OrderByDescending*, *ThenBy*, *ThenByDescending* and *GroupBy* are shown below. In the examples below *_em1* is an EntityManager.

**C#**

```
[Description("Select all customers ordered by ContactName.")]
public void LinqToEntities53() {
  var query = from c in _em1.Customers
              orderby c.ContactName
              select c;
  Assert.IsTrue(query.First().ContactName == "Alejandra Camino");
}
[Description("Select all customers ordered by ContactName
descending.")]
public void LinqToEntities54() {
  var query = from c in _em1.Customers
              orderby c.CompanyName descending
              select c;
  Assert.IsTrue(query.First().CompanyName == "Wolski Zajazd");
}
[Description("Select an anonoymous type with all product IDs
ordered by UnitsInStock.")]
public void LinqToEntities55() {
  var query = from p in _em1.Products
              orderby p.UnitsInStock
              select new { p.ProductID, p.UnitsInStock };
  Assert.IsTrue(query.First().ProductID == 5);
}
[Description("Select an anonoymous type with all product IDs
ordered by UnitsInStock as a method query.")]
public void LinqToEntities56() {
  var query = _em1.Products.OrderBy(p => p.UnitsInStock)
      .Select(p2 => new { p2.ProductID, p2.UnitsInStock });
  Assert.IsTrue(query.First().ProductID == 5);
  Assert.IsTrue(query.First().UnitsInStock == 0);
}
[Description("Select all customers ordered by the descending
region.")]
public void LinqToEntities57() {
  var query = from c in _em1.Customers
              orderby c.Address.Region descending
              select c;
  Assert.IsTrue(query.First().CustomerID == "SPLIR");
  Assert.IsTrue(query.First().Address.Region == "WY");
}
[Description("Select all customers ordered by the descending region
as a method query.")]
public void LinqToEntities58() {
  var query = _em1.Customers.Select(c => c).OrderByDescending(c2
=> c2.Address.Region);
  Assert.IsTrue(query.First().CustomerID == "SPLIR");
}
[Description("Select all customers ordered by the region, then the
contact name.")]
public void LinqToEntities59() {
  var query = _em1.Customers.Select(c => c)
      .OrderBy(c => c.Address.Region).ThenBy(c => c.ContactName);
  Assert.IsTrue(query.First().CustomerID == "ROMEY");
  Assert.IsTrue(query.First().Address.City + "/" +
query.First().ContactName == "Madrid/Alejandra Camino");
}
[Description("Select all customers ordered by the region in
descending order, then the contact name.")]
public void LinqToEntities60() {
  var query = _em1.Customers.Select(c => c)
      .OrderByDescending(c => c.Address.Region).ThenBy(c =>
c.ContactName);
  Assert.IsTrue(query.First().CustomerID == "SPLIR");
}
[Description("Select all customers ordered by the region then the
contact name in descending order.")]
public void LinqToEntities61() {
```

```
  var query = _em1.Customers.Select(c => c).OrderBy(c =>
c.Address.Region).ThenByDescending(c => c.ContactName);
  Assert.IsTrue(query.First().CustomerID == "WOLZA");
 ////Alternate
 //var query = _Em1.Customers.OrderBy(c =>
c.Address.Region).ThenByDescending(c => c.ContactName).Select(c
=> c);
}
[Description("Select all products ordered by the descending unit
price.")]
public void LinqToEntities62() {
  var query = from p in _em1.Products
              orderby p.UnitPrice descending
              select p;
  Assert.IsTrue(query.First().ProductID == 38);
 //// Alternate
 //var query0 = _Em1.Products.OrderByDescending(p =>
p.UnitPrice).Select(p => p);
 //Assert.IsTrue(query0.First().ProductID == 38);}
}
[Description("Select all orders for a customer ordered by date that the
order was placed.")]
public void LinqToEntities63() {
 // Alternate:
 //var query0 = _Em1.Customers.Where(c=>c.CustomerID ==
"ALFKI")
 //    .SelectMany(c => c.Orders).OrderBy(o => o.OrderDate);
 //Assert.IsTrue(query0.First().OrderDate  == new
DateTime(1997,8,25));
  var query = _em1.Customers.Where(cust => cust.CustomerID ==
"ALFKI")
     .SelectMany(c => c.Orders.Select(o => o))
     .OrderBy(o2 => o2.OrderDate);
  Assert.IsTrue(query.First().OrderDate == new DateTime(1997, 8,
25));
}
[Description("Select all Regions with a customer.")]
public void LinqToEntities64() {
  var query = from c in _em1.Customers
              group c by c.Address.Region into regions
              select new { regions.Key };
  Assert.IsTrue(query.Count() == 19);
  var query2 = query.OrderByDescending(r => r.Key);
  Assert.IsTrue(query2.First().Key == "WY");
}
[Description("Select all dates with orders placed.")]
public void LinqToEntities65() {
  var query = from o in _em1.Orders
              group o by o.OrderDate into dates
              select new { dates.Key };
  Assert.IsTrue(query.Count() == 480);
  var query2 = query.OrderBy(d => d.Key);
  Assert.IsTrue(query2.First().Key == new DateTime(1996, 7, 4));
}
[Description("Select all Regions and customer count for each
region.")]
public void LinqToEntities66() {
  var query = from c in _em1.Customers
              group c by c.Address.Region into regions
              orderby regions.Key
              select new { region = regions.Key, count =
regions.Count() };
  var r = query.ToList();
  Assert.IsTrue(r.First().count == 60); // the variable count
}
[Description("Select all Regions and customer count for each region
as a method query.")]
public void LinqToEntities67() {
  var query = _em1.Customers.GroupBy(c => c.Address.Region)
     .OrderByDescending(r => r.Key)
     .Select(r => new { region = r.Key, count = r.Count() });
```

```
  Assert.IsTrue(query.First().count == 1);
}
[Description("Select all Customer Regions with the total Freight on
all orders for Customers in that Region.")]
public void LinqToEntities68() {
  var query = from c in _em1.Customers
              group c by c.Address.Region into regions
              orderby regions.Key
              join c2 in _em1.Customers on regions.Key equals
c2.Address.Region
              select new { region = regions.Key, total = c2.Orders.Sum(o
=> o.Freight) };
  var r = query.ToList();
  Assert.IsTrue(query.First().region == null);
  Assert.IsTrue(query.First().total == 225.58M);
}
[Description("Select all Customer Regions with the total Freight on
all orders for Customers in that Region as a method query.")]
public void LinqToEntities69() {
  var query = _em1.Customers.GroupBy(c => c.Address.Region)
     .OrderBy(r => r.Key)
     .Select(g => new {
       Region = g.Key,
       FreightTotal = g
          .SelectMany(c2 => c2.Orders)
          .Sum(o => o.Freight)
     });
  Assert.IsTrue(query.First().Region == null);
  Assert.IsTrue(query.First().FreightTotal == 38063.31M);
  var query2 = query.OrderByDescending(r => r.Region);
  Assert.IsTrue(query2.First().Region == "WY");
  Assert.IsTrue(query2.First().FreightTotal == 558.67M);
}
```

**VB**

```
<Description("Select all customers ordered by ContactName.")>
Public Sub LinqToEntities53()
 Dim query = From c In _em1.Customers
             Order By c.ContactName
             Select c
  Assert.IsTrue(query.First().ContactName = "Alejandra Camino")
End Sub
<Description("Select all customers ordered by ContactName
descending.")>
Public Sub LinqToEntities54()
 Dim query = From c In _em1.Customers
             Order By c.CompanyName Descending
             Select c
  Assert.IsTrue(query.First().CompanyName = "Wolski Zajazd")
End Sub
<Description("Select an anonoymous type with all product IDs
ordered by UnitsInStock.")>
Public Sub LinqToEntities55()
 Dim query = From p In _em1.Products
             Order By p.UnitsInStock
             Select New With {Key p.ProductID, Key p.UnitsInStock}
  Assert.IsTrue(query.First().ProductID = 5)
End Sub
<Description("Select an anonoymous type with all product IDs
ordered by UnitsInStock as a method query.")>
Public Sub LinqToEntities56()
 Dim query = _em1.Products.OrderBy(Function(p)
p.UnitsInStock).Select(Function(p2) New With {Key p2.ProductID,
Key p2.UnitsInStock})
  Assert.IsTrue(query.First().ProductID = 5)
  Assert.IsTrue(query.First().UnitsInStock = 0)
End Sub
<Description("Select all customers ordered by the descending
region.")>
Public Sub LinqToEntities57()
 Dim query = From c In _em1.Customers
             Order By c.Address.Region Descending
             Select c
```

```
  Assert.IsTrue(query.First().CustomerID = "SPLIR")
  Assert.IsTrue(query.First().Address.Region = "WY")
End Sub
<Description("Select all customers ordered by the descending region
as a method query.")>
Public Sub LinqToEntities58()
 Dim query = _em1.Customers.Select(Function(c)
c).OrderByDescending(Function(c2) c2.Address.Region)
  Assert.IsTrue(query.First().CustomerID = "SPLIR")
End Sub
<Description("Select all customers ordered by the region, then the
contact name.")>
Public Sub LinqToEntities59()
 Dim query = _em1.Customers.Select(Function(c)
c).OrderBy(Function(c) c.Address.Region).ThenBy(Function(c)
c.ContactName)
  Assert.IsTrue(query.First().CustomerID = "ROMEY")
  Assert.IsTrue(query.First().Address.City & "/" &
query.First().ContactName = "Madrid/Alejandra Camino")
End Sub
<Description("Select all customers ordered by the region in
descending order, then the contact name.")>
Public Sub LinqToEntities60()
 Dim query = _em1.Customers.Select(Function(c)
c).OrderByDescending(Function(c)
c.Address.Region).ThenBy(Function(c) c.ContactName)
  Assert.IsTrue(query.First().CustomerID = "SPLIR")
End Sub
<Description("Select all customers ordered by the region then the
contact name in descending order.")>
Public Sub LinqToEntities61()
 Dim query = _em1.Customers.Select(Function(c)
c).OrderBy(Function(c)
c.Address.Region).ThenByDescending(Function(c) c.ContactName)
  Assert.IsTrue(query.First().CustomerID = "WOLZA")
 '//Alternate
 'var query = _Em1.Customers.OrderBy(c =>
c.Address.Region).ThenByDescending(c => c.ContactName).Select(c
=> c);
End Sub
<Description("Select all products ordered by the descending unit
price.")>
Public Sub LinqToEntities62()
 Dim query = From p In _em1.Products
             Order By p.UnitPrice Descending
            Select p
  Assert.IsTrue(query.First().ProductID = 38)
 '// Alternate
 'var query0 = _Em1.Products.OrderByDescending(p =>
p.UnitPrice).Select(p => p);
 'Assert.IsTrue(query0.First().ProductID == 38);}
End Sub
<Description("Select all orders for a customer ordered by date that
the order was placed.")>
Public Sub LinqToEntities63()
 ' Alternate:
 'var query0 = _Em1.Customers.Where(c=>c.CustomerID ==
"ALFKI")
 '    .SelectMany(c => c.Orders).OrderBy(o => o.OrderDate);
 'Assert.IsTrue(query0.First().OrderDate  == new
DateTime(1997,8,25));
 Dim query = _em1.Customers.Where(Function(cust)
cust.CustomerID = "ALFKI").SelectMany(Function(c)
c.Orders.Select(Function(o) o)).OrderBy(Function(o2)
o2.OrderDate)
  Assert.IsTrue(query.First().OrderDate = New Date(1997, 8, 25))
End Sub
<Description("Select all Regions with a customer.")>
Public Sub LinqToEntities64()
 Dim query = From c In _em1.Customers
             Group c By c.Address.Region Into regions = Group
```

```
            Select New With {Key Region}
  Assert.IsTrue(query.Count() = 19)
 Dim query2 = query.OrderByDescending(Function(r) r.Key)
  Assert.IsTrue(query2.First().Key = "WY")
End Sub
<Description("Select all dates with orders placed.")>
Public Sub LinqToEntities65()
 Dim query = From o In _em1.Orders
             Group o By o.OrderDate Into dates = Group
            Select New With {Key OrderDate}
  Assert.IsTrue(query.Count() = 480)
 Dim query2 = query.OrderBy(Function(d) d.Key)
  Assert.IsTrue(query2.First().Key = New Date(1996, 7, 4))
End Sub
<Description("Select all Regions and customer count for each
region.")>
Public Sub LinqToEntities66()
 Dim query = From c In _em1.Customers
             Group c By c.Address.Region Into regions = Group
             Order By Region
            Select New With {Key .region = Region, Key .count =
regions.Count()}
 Dim r = query.ToList()
  Assert.IsTrue(r.First().count = 60) ' the variable count
End Sub
<Description("Select all Regions and customer count for each region
as a method query.")>
Public Sub LinqToEntities67()
 Dim query = _em1.Customers.GroupBy(Function(c)
c.Address.Region).OrderByDescending(Function(r)
r.Key).Select(Function(r) New With {Key .region = r.Key, Key .count
= r.Count()})
  Assert.IsTrue(query.First().count = 1)
End Sub
<Description("Select all Customer Regions with the total Freight on
all orders for Customers in that Region.")>
Public Sub LinqToEntities68()
 Dim query = From c In _em1.Customers
             Group c By c.Address.Region Into regions = Group
             Order By Region
             Join c2 In _em1.Customers On Region Equals
c2.Address.Region
            Select New With {Key .region = Region, Key .total =
c2.Orders.Sum(Function(o) o.Freight)}
 Dim r = query.ToList()
  Assert.IsTrue(query.First().region Is Nothing)
  Assert.IsTrue(query.First().total = 225.58D)
End Sub
<Description("Select all Customer Regions with the total Freight on
all orders for Customers in that Region as a method query.")>
Public Sub LinqToEntities69()
 Dim query = _em1.Customers.GroupBy(Function(c)
c.Address.Region).OrderBy(Function(r) r.Key).Select(Function(g)
New With {Key .Region = g.Key, Key .FreightTotal =
g.SelectMany(Function(c2) c2.Orders).Sum(Function(o) o.Freight)})
  Assert.IsTrue(query.First().Region Is Nothing)
  Assert.IsTrue(query.First().FreightTotal = 38063.31D)
 Dim query2 = query.OrderByDescending(Function(r) r.Region)
  Assert.IsTrue(query2.First().Region = "WY")
  Assert.IsTrue(query2.First().FreightTotal = 558.67D)
End Sub
```