

Examples of the LINQ *OfType* operator are shown below. In the examples below *\_em1* is an EntityManager.

C#	<pre> [Description("Select all products, both active and discontinued products, and show the type.")] public void LinqToEntities77() {     var query = _em1         .Products         .Select(p =&gt; p);     var query2 = query         // force local execution to show local type         .AsEnumerable()         .Select(p =&gt; new { type = p.GetType().ToString(), prod = p });     StringBuilder msg = new StringBuilder();     foreach (var aProduct in query2) {         msg.Append("(" + aProduct.type + ") " + aProduct.prod.ProductName + Environment.NewLine);     }     // set break point and inspect msg.ToString() as desired     var query3 = query2.OrderBy(p =&gt; p.prod.ProductName);     Assert.IsTrue(query3.First().prod.ProductName == "Alice Mutton"); } [Description("Select only discontinued products.")] public void LinqToEntities78() {     var query = _em1.Products.OfType&lt;DiscontinuedProduct&gt;().Select(p =&gt; p);     var r = query.ToList();     Assert.IsTrue(r.Count() == 8); } [Description("Select only products, which will retrn all Products and subtypes of Products (DiscontinuedProducts and ActiveProducts).")] public void LinqToEntities79() {     var query = _em1.Products.OfType&lt;Product&gt;().Select(p =&gt; p);     Assert.IsTrue(query.Count() == 77); } [Description("Select only discontinued products.")] public void LinqToEntities80() {     // Similar to LinqToEntities78; no .Select clause     var query = _em1.Products.OfType&lt;DiscontinuedProduct&gt;();     Assert.IsTrue(query.Count() == 8); } [TestMethod] [Description("Select only discontinued products.")] public void LinqToEntities81() {     var query = _em1.Products         .Where(p =&gt; p is DiscontinuedProduct);     Assert.IsTrue(query.Count() == 8); } [Description("Select all current employees.")] public void LinqToEntities87() {     var query = _em1.Employees.OfType&lt;CurrentEmployee&gt;()         .ToList().Select(p =&gt; new { type = p.GetType().ToString(), p });     Assert.IsTrue(query.Count() &gt;= 8);     int lastId = query.OrderBy(e =&gt; e.p.EmployeeID).Last().p.EmployeeID;     var query2 = query.OrderByDescending(e =&gt; e.p.EmployeeID);     Assert.IsTrue(query2.First().p.EmployeeID == lastId); } </pre>
VB	<pre> &lt;Description("Select all products, both active and discontinued products, and show the type.")&gt; Public Sub LinqToEntities77()     Dim query = _em1.Products.Select(Function(p) p)     ' force local execution to show local type     Dim query2 = query.AsEnumerable().Select(Function(p) New With { Key .type = p.GetType().ToString(), Key .prod = p })     Dim msg As New StringBuilder()     For Each aProduct In query2         msg.Append("(" &amp; aProduct.type &amp; ") " &amp; aProduct.prod.ProductName &amp; Environment.NewLine)     Next aProduct     ' set break point and inspect msg.ToString() as desired </pre>

```

Dim query3 = query2.OrderBy(Function(p) p.prod.ProductName)
Assert.IsTrue(query3.First().prod.ProductName = "Alice Mutton")
End Sub
<Description("Select only discontinued products.")>
Public Sub LinqToEntities78()
Dim query = _em1.Products.OfType(Of DiscontinuedProduct)
().Select(Function(p) p)
Dim r = query.ToList()
Assert.IsTrue(r.Count() = 8)
End Sub
<Description("Select only products, which will reutrn all
Products and subtypes of Products (DiscontinuedProducts and
ActiveProducts).")>
Public Sub LinqToEntities79()
Dim query = _em1.Products.OfType(Of Product)
().Select(Function(p) p)
Assert.IsTrue(query.Count() = 77)
End Sub
<Description("Select only discontinued products.")>
Public Sub LinqToEntities80()
' Similar to LinqToEntities78; no .Select clause
Dim query = _em1.Products.OfType(Of DiscontinuedProduct)()
Assert.IsTrue(query.Count() = 8)
End Sub
<TestMethod, Description("Select only discontinued products.")>
Public Sub LinqToEntities81()
Dim query = _em1.Products.Where(Function(p) TypeOf p Is
DiscontinuedProduct)
Assert.IsTrue(query.Count() = 8)
End Sub
<Description("Select all current employees.")>
Public Sub LinqToEntities87()
Dim query = _em1.Employees.OfType(Of CurrentEmployee)
().ToList().Select(Function(p) New With { Key .type =
p.GetType().ToString(), Key p })
Assert.IsTrue(query.Count() >= 8)
Dim lastId As Integer = query.OrderBy(Function(e)
e.p.EmployeeID).Last().p.EmployeeID
Dim query2 = query.OrderByDescending(Function(e)
e.p.EmployeeID)
Assert.IsTrue(query2.First().p.EmployeeID = lastId)
End Sub

```