----All (Predicate)----

<<[Managers.All(p => p.Age < 30)]>>

----Any()----

<<[Managers.Any()]>>

----Any(Predicate)----

<<[Managers.Any(p => p.Name == “John Smith”)]>>

----Average(Selector)----

<<[Managers.Average(p => p.Age)]>>

----Concat(IEnumerable)----

<<foreach [m in Managers.Concat(Managers)]>>Name = <<[m.Name]>>, <</foreach>>

----Contains(Object)----

<<[Managers.Contains(Managers.First())]>>

----Count()----

<<[Managers.Count()]>>

----Count(Predicate)----

<<[Managers.Count(p => p.Age < 30)]>>

----Distinct()----

<<foreach [m in Managers.Distinct()]>><<[m.Name]>><</foreach>>

----First()----

<<[Managers.First().Name]>>

----First(Predicate)----

<<[Managers.First(p => p.Age < 40).Name]>>

----FirstOrDefault()----

<<[Managers.FirstOrDefault().Name]>>

----FirstOrDefault(Predicate)----

<<[Managers. FirstOrDefault(p => p.Age < 50).Name]>>

----GroupBy(Selector)----

<<foreach [m in Managers.GroupBy(p => new { p.Age, p.Name })]>><<foreach [c in m]>>Age = <<[c.Age]>>, Name = <<[c.Name]>>; <</foreach>><</foreach>>

----Last()----

<<[Managers.Last().Name]>>

----Last(Predicate)----

<<[Managers.Last(p => p.Age < 50).Name]>>

----LastOrDefault()----

<<[Managers.LastOrDefault().Name]>>

----LastOrDefault(Predicate)----

<<[Managers.LastOrDefault(p => p.Age < 50).Name]>>

----Max(ComparableSelector)----

<<[Managers.Max(p => p.Age)]>>

----Min(ComparableSelector)----

<<[Managers.Min(p => p.Age)]>>

----OrderBy(ComparableSelector)----

<<foreach [m in Managers.OrderBy(p => p.Age).ThenBy(p => p.Contracts.Sum(c => c.Price))]>>Name = <<[m.Name]>>, <</foreach>>

----OrderByDescending(ComparableSelector)----

<<foreach [m in Managers.OrderByDescending(p => p.Age)]>>Name = <<[m.Name]>>, <</foreach>>

----Single()----

<<[Managers.Where(p => p.Name == “John Smith”).Single().Name]>>

----Single(Predicate)----

<<[Managers. Where(p => p.Name == “John Smith”).Single(p => p.Name == “John Smith”).Name]>>

----SingleOrDefault()----

<<[Managers.Where(p => p.Name == “John Smith”).SingleOrDefault().Name]>>

----SingleOrDefault(Predicate)----

<<[Managers.Where(p => p.Name == “John Smith”).SingleOrDefault(p => p.Name == “John Smith”).Name]>>

----Skip(Int)----

<<foreach [m in Managers.Skip(2)]>><<[m.Name]>><</foreach>>

----SkipWhile(Predicate)----

<<foreach [m in Managers.SkipWhile(p => p.Age < 37)]>><<[m.Name]>><</foreach>>

----Sum(Selector)----

<<[Managers.Sum(p => p.Contracts.Count())]>>

----Take(int)----

<<foreach [m in Managers.Take(3)]>>Name = <<[m.Name]>>, <</foreach>>

----TakeWhile(Predicate)----

<<foreach [m in Managers.TakeWhile(p => p.Age < 40)]>>Name = <<[m.Name]>>, <</foreach>>

----Union(IEnumerable)----

<<foreach [m in Managers.Union(Managers)]>>Name = <<[m.Name]>>, <</foreach>>

----Where(Predicate)----

<<foreach [m in Managers.Where(p => p.Age < 45)]>>Name = <<[m.Name]>>, <</foreach>>