SQL,LINQ,Lambda 语法对照图

SQL	LINQ	Lambda
SELECT *FROM HumanResources.Employee	from e in Employees select e	Employees .Select (e => e)

SELECT e.LoginID, e.JobTitle
 	
FROM HumanResources.Employee AS e
	from e in Employees
	
	select new {e.LoginID, e.JobTitle}
		Employees.Select (
		e => new {
		LoginID = e.LoginID,
		JobTitle = e.JobTitle
		}
		)

SELECT e.LoginID AS ID, e.JobTitle AS Title
 	
FROM HumanResources.Employee AS e
	from e in Employees
	
	select new {ID = e.LoginID, Title = e.JobTitle}
		Employees.Select (
		e => new
		{
		ID = e.LoginID,
		Title = e.JobTitle
		}
		)

SELECT DISTINCT e.JobTitle **FROM HumanResources.Employee AS e**	(from e in Employees select e.JobTitle).Distinct()	Employees .Select (e => e.JobTitle) .Distinct ()
SELECT e.* **FROM HumanResources.Employee AS e** **WHERE e.LoginID = 'test'**	from e in Employees where e.LoginID == "test" select e	Employees .Where (e => (e.LoginID == "test"))
SELECT e.* **FROM HumanResources.Employee AS e** **WHERE e.LoginID = 'test' AND e.SalariedFlag = 1**	from e in Employees where e.LoginID == "test" && e.SalariedFlag select e	Employees .Where (e => ((e.LoginID == "test") && e.SalariedFlag))

SELECT e.* FROM HumanResources.Employee AS e **WHERE e.VacationHours >= 2 AND e.VacationHours <= 10**	from e in Employees where e.VacationHours >= 2 && e.VacationHours <= 10 select e	Employees .Where (e => (((Int32)(e.VacationHours) >= 2) && ((Int32)(e.VacationHours) <= 10)))
SELECT e.* FROM HumanResources.Employee AS e **ORDER BY e.NationalIDNumber**	from e in Employees orderby e.NationalIDNumber select e	Employees .OrderBy (e => e.NationalIDNumber)
SELECT e.* FROM HumanResources.Employee AS e **ORDER BY e.HireDate DESC, e.NationalIDNumber**	from e in Employees orderby e.HireDate descending, e.NationalIDNumber select e	Employees .OrderByDescending (e => e.HireDate) .ThenBy (e => e.NationalIDNumber)
SELECT e.* FROM HumanResources.Employee AS e **WHERE e.JobTitle LIKE 'Vice%' OR SUBSTRING(e.JobTitle, 0, 3) = 'Pro'**	from e in Employees where e.JobTitle.StartsWith("Vice") || e.JobTitle.Substring(0, 3) == "Pro" select e	Employees .Where (e => (e.JobTitle.StartsWith ("Vice") || (e.JobTitle.Substring (0, 3) == "Pro")))
SELECT SUM(e.VacationHours) **FROM HumanResources.Employee AS e**		Employees.Sum(e => e.VacationHours);
SELECT COUNT(*) FROM HumanResources.Employee AS e		Employees.Count();

SELECT SUM(e.VacationHours) AS TotalVacations, e.JobTitle
 	
FROM HumanResources.Employee AS e
 	
GROUP BY e.JobTitle
	from e in Employees
	
	group e by e.JobTitle into g
	
	select new {JobTitle = g.Key, TotalVacations = g.Sum(e => e.VacationHours)}
		Employees
		.GroupBy (e => e.JobTitle)
		.Select (
		g => new
		{
		JobTitle = g.Key,
		TotalVacations = g.Sum (e => (Int32)(e.VacationHours))
		}
		)

SELECT e.JobTitle, SUM(e.VacationHours) AS	from e in Employees	Employees .GroupBy (e => e.JobTitle)

TotalVacations
 	
FROM HumanResources.Employee AS e
 	
GROUP BY e.JobTitle
 	
HAVING e.COUNT(*) > 2
	group e by e.JobTitle into g
	
	where g.Count() > 2
	
	select new {JobTitle = g.Key, TotalVacations = g.Sum(e => e.VacationHours)}
		.Where (g => (g.Count () > 2))
		.Select (
		g =>
		new
		{
		JobTitle = g.Key,
		TotalVacations = g.Sum (e => (Int32)(e.VacationHours))
		}
		)

SELECT *
 	
FROM Production.Product AS p, Production.ProductReview AS pr
	from p in Products
	
	from pr in ProductReviews
	
	select new {p, pr}
		Products
		.SelectMany (
		p => ProductReviews,
		(p, pr) =>
		new
		{
		p = p,
		pr = pr
		}
		)

SELECT *
 	
FROM Production.Product AS p
 	
INNER JOIN Production.ProductReview AS pr ON p.ProductID = pr.ProductID
	from p in Products
	
	join pr in ProductReviews on p.ProductID equals pr.ProductID
	
	select new {p, pr}
		Products
		.Join (
		ProductReviews,
		p => p.ProductID,
		pr => pr.ProductID,
		(p, pr) =>
		new
		{
		p = p,
		pr = pr
		}
		)

SELECT * **FROM Production.Product AS p** **INNER JOIN Production.ProductCostHistory AS pch ON p.ProductID = pch.ProductID AND p.SellStartDate = pch.StartDate**	from p in Products join pch in ProductCostHistories on new {p.ProductID, StartDate = p.SellStartDate} equals new {pch.ProductID, StartDate = pch.StartDate} select new {p, pch}	Products .Join (ProductCostHistories, p => new { ProductID = p.ProductID, StartDate = p.SellStartDate },

		pch =>
		new
		{
		ProductID = pch.ProductID,
		StartDate = pch.StartDate
		},
		(p, pch) =>
		new
		{
		p = p,
		pch = pch
		}
		)

SELECT *
 	
FROM Production.Product AS p
 	
LEFT OUTER JOIN Production.ProductReview AS pr ON p.ProductID = pr.ProductID
	from p in Products
	
	join pr in ProductReviews on p.ProductID equals pr.ProductID
	
	into prodrev
	
	select new {p, prodrev}
		Products
		.GroupJoin (
		ProductReviews,
		p => p.ProductID,
		pr => pr.ProductID,
		(p, prodrev) =>
		new
		{
		p = p,
		prodrev = prodrev
		}
		)

SELECT p.ProductID AS ID
 	
FROM Production.Product AS p
 	
UNION
 	
SELECT pr.ProductReviewID
 	
FROM Production.ProductReview AS pr
	(from p in Products
	
	select new {ID = p.ProductID}).Union(
	
	from pr in ProductReviews
	
	select new {ID = pr.ProductReviewID})
		Products
		.Select (
		p =>
		new
		{
		ID = p.ProductID
		}
		)
		.Union (
		ProductReviews
		.Select (
		pr =>
		new
		{
		ID = pr.ProductReviewID
		}

		))
SELECT TOP (10) * **FROM Production.Product AS p** **WHERE p.StandardCost < 100**	(from p in Products where p.StandardCost < 100 select p).Take(10)	Products .Where (p => (p.StandardCost < 100)) .Take (10)
SELECT * **FROM [Production].[Product] AS p** **WHERE p.ProductID IN(** **SELECT pr.ProductID** **FROM [Production].[ProductReview] AS [pr]** **WHERE pr.[Rating] = 5** **)**	from p in Products where (from pr in ProductReviews where pr.Rating == 5 select pr.ProductID).Contains(p.ProductID) select p	Products .Where (p => ProductReviews .Where (pr => (pr.Rating == 5)) .Select (pr => pr.ProductID) .Contains (p.ProductID))