# Para crear el modelo AdventureWorks Sales y hacer referencia al mismo

1. En el menú **Proyecto**, haga clic en **Agregar nuevo elemento**.
2. En el panel Plantillas, seleccione ADO.NET Entity Data Model.
3. Escriba **AdvWorksModel.edmx** en el cuadro **Nombre** y, a continuación, haga clic en **Agregar**.
4. Seleccione **Generar desde la base de datos** y, a continuación, haga clic en **Siguiente**.
5. En la ventana **Elegir la conexión de datos**, seleccione una conexión de AdventureWorks existente en la lista, o cree una conexión nueva a una instancia de SQL Server que tenga la base de datos de ejemplo AdventureWorks.
6. Guarde la configuración de la conexión de entidad en el archivo App.Config como **AdventureWorksEntities** y haga clic en **Siguiente**.
7. En el cuadro de diálogo **Elija los objetos de base de datos**, borre todos los objetos, expanda **Tablas** y seleccione los objetos de tabla siguientes:
   * Address (Person)
   * Contact (Person)
   * Product (Production)
   * SalesOrderDetail (Sales)
   * SalesOrderHeader (Sales)
8. Asigne el nombre **AdventureWorksModel** al espacio de nombres del modelo y haga clic en **Finalizar**.
9. En un proyecto de C#, agregue una instrucción **using** para AdventureWorksEntities al archivo del código fuente.

En un proyecto de Visual Basic, agregue una instrucción **Imports** para AdventureWorksEntities al archivo del código fuente.

Select

Ejemplo

En el ejemplo siguiente se utiliza el método [Select](https://msdn.microsoft.com/es-es/library/system.linq.enumerable.select(v=vs.90).aspx) para devolver todas las filas de la tabla *Product* y mostrar los nombres de producto.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb738447(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-2)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Product> products = AWEntities.Product;

IQueryable<Product> productsQuery = **from product in products**

**select product**;

Console.WriteLine("Product Names:");

foreach (var prod in productsQuery)

{

Console.WriteLine(prod.Name);

}

}

El ejemplo siguiente utiliza **Select** para devolver una secuencia de nombres de producto solamente.

C#

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Product> products = AWEntities.Product;

IQueryable<string> productNames =

**from p in products**

**select p.Name;**

Console.WriteLine("Product Names:");

foreach (String productName in productNames)

{

Console.WriteLine(productName);

}

}

Ejemplo

En el ejemplo siguiente se utiliza el método [Select](https://msdn.microsoft.com/es-es/library/bb548743(v=vs.90).aspx) para proyectar las propiedades **Product.Name** y **Product.ProductID** en una secuencia de tipos anónimos.

C#

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Product> products = AWEntities.Product;

**var query =**

**from product in products**

**select new**

**{**

**ProductId = product.ProductID,**

**ProductName = product.Name**

**};**

Console.WriteLine("Product Info:");

foreach (var productInfo in query)

{

Console.WriteLine("Product Id: {0} Product name: {1} ",

productInfo.ProductId, productInfo.ProductName);

}

}

Ejemplos de sintaxis de expresiones de consultas: proyección (LINQ to Entities)

**.NET Framework 3.5**

[Otras versiones](javascript:void(0))

https://i-msdn.sec.s-msft.com/Areas/Epx/Content/Images/ImageSprite.png?v=636311636105440369

Los ejemplos de este tema muestran cómo se utilizan el método **Select** y las palabras clave **From … From …**para consultar el [modelo AdventureWorks Sales](https://msdn.microsoft.com/es-es/library/bb387147(v=vs.90).aspx) con la sintaxis de las expresiones de consulta. **From … From …** es el equivalente basado en las consultas del método **SelectMany**. El modelo AdventureWorks Sales que se usa en estos ejemplos se crea a partir de las tablas Contact, Address, Product, SalesOrderHeader y SalesOrderDetail en la base de datos de ejemplo AdventureWorks.

Los ejemplos de este tema usan las siguientes instrucciones **using**/**Imports**:

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb738447(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-1)

using System;

using System.Data;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Data.Objects;

using AdventureWorksModel;

using System.Globalization;

Para obtener más información, vea [Cómo crear un proyecto de LINQ to Entities en Visual Studio](https://msdn.microsoft.com/es-es/library/bb738557(v=vs.90).aspx).

Select

Ejemplo

En el ejemplo siguiente se utiliza el método [Select](https://msdn.microsoft.com/es-es/library/system.linq.enumerable.select(v=vs.90).aspx) para devolver todas las filas de la tabla *Product* y mostrar los nombres de producto.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb738447(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-2)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Product> products = AWEntities.Product;

IQueryable<Product> productsQuery = from product in products

select product;

Console.WriteLine("Product Names:");

foreach (var prod in productsQuery)

{

Console.WriteLine(prod.Name);

}

}

Ejemplo

El ejemplo siguiente utiliza **Select** para devolver una secuencia de nombres de producto solamente.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb738447(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-3)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Product> products = AWEntities.Product;

IQueryable<string> productNames =

from p in products

select p.Name;

Console.WriteLine("Product Names:");

foreach (String productName in productNames)

{

Console.WriteLine(productName);

}

}

Ejemplo

En el ejemplo siguiente se utiliza el método [Select](https://msdn.microsoft.com/es-es/library/bb548743(v=vs.90).aspx) para proyectar las propiedades **Product.Name** y **Product.ProductID** en una secuencia de tipos anónimos.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb738447(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-4)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Product> products = AWEntities.Product;

var query =

from product in products

select new

{

ProductId = product.ProductID,

ProductName = product.Name

};

Console.WriteLine("Product Info:");

foreach (var productInfo in query)

{

Console.WriteLine("Product Id: {0} Product name: {1} ",

productInfo.ProductId, productInfo.ProductName);

}

}

From … From … (SelectMany)

Ejemplo

En el ejemplo siguiente se utiliza **From … From …** (el equivalente del método [SelectMany](https://msdn.microsoft.com/es-es/library/system.linq.enumerable.selectmany(v=vs.90).aspx)) para seleccionar todos los pedidos en los que *TotalDue* es inferior a 500,00.

C#

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Contact> contacts = AWEntities.Contact;

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

var query =

from contact in contacts

from order in orders

where contact.ContactID == order.Contact.ContactID

&& order.TotalDue < 500.00M

select new

{

ContactID = contact.ContactID,

LastName = contact.LastName,

FirstName = contact.FirstName,

OrderID = order.SalesOrderID,

Total = order.TotalDue

};

foreach (var smallOrder in query)

{

Console.WriteLine("Contact ID: {0} Name: {1}, {2} Order ID: {3} Total Due: ${4} ",

smallOrder.ContactID, smallOrder.LastName, smallOrder.FirstName,

smallOrder.OrderID, smallOrder.Total);

}

}

Ejemplo

En el ejemplo siguiente se utiliza **From … From …** (el equivalente del método **SelectMany**) para seleccionar todos los pedidos efectuados a partir del 1 de octubre de 2002.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb738447(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-6)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Contact> contacts = AWEntities.Contact;

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

var query =

from contact in contacts

from order in orders

where contact.ContactID == order.Contact.ContactID

&& order.OrderDate >= new DateTime(2002, 10, 1)

select new

{

ContactID = contact.ContactID,

LastName = contact.LastName,

FirstName = contact.FirstName,

OrderID = order.SalesOrderID,

OrderDate = order.OrderDate

};

foreach (var order in query)

{

Console.WriteLine("Contact ID: {0} Name: {1}, {2} Order ID: {3} Order date: {4:d} ",

order.ContactID, order.LastName, order.FirstName,

order.OrderID, order.OrderDate);

}

}

Ejemplo

En el ejemplo siguiente se utiliza **From … From …** (el equivalente al método **SelectMany**) para seleccionar todos los pedidos en los que el total del pedido es superior a 10000,00 y se utiliza la asignación **From** para evitar que se solicite dos veces el total.

C#

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Contact> contacts = AWEntities.Contact;

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

var query =

from contact in contacts

from order in orders

let total = order.TotalDue

where contact.ContactID == order.Contact.ContactID

&& total >= 10000.0M

select new

{

ContactID = contact.ContactID,

LastName = contact.LastName,

OrderID = order.SalesOrderID,

total

};

foreach (var order in query)

{

Console.WriteLine("Contact ID: {0} Last name: {1} Order ID: {2} Total: {3}",

order.ContactID, order.LastName, order.OrderID, order.total);

}

}

Where

Ejemplo

En el ejemplo siguiente se devuelven todos los pedidos en línea.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb738636(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-2)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

var onlineOrders =

from order in orders

where order.OnlineOrderFlag == true

select new

{

SalesOrderID = order.SalesOrderID,

OrderDate = order.OrderDate,

SalesOrderNumber = order.SalesOrderNumber

};

foreach (var onlineOrder in onlineOrders)

{

Console.WriteLine("Order ID: {0} Order date: {1:d} Order number: {2}",

onlineOrder.SalesOrderID,

onlineOrder.OrderDate,

onlineOrder.SalesOrderNumber);

}

}

Ejemplo

En el ejemplo siguiente se devuelven los pedidos en los que la cantidad de pedido es superior a 2 e inferior a 6.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb738636(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-3)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<SalesOrderDetail> orders = AWEntities.SalesOrderDetail;

var query =

from order in orders

where order.OrderQty > 2 && order.OrderQty < 6

select new

{

SalesOrderID = order.SalesOrderID,

OrderQty = order.OrderQty

};

foreach (var order in query)

{

Console.WriteLine("Order ID: {0} Order quantity: {1}",

order.SalesOrderID, order.OrderQty);

}

}

Ejemplo

En el ejemplo siguiente se devuelven todos los productos de color rojo.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb738636(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-4)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Product> products = AWEntities.Product;

var query =

from product in products

where product.Color == "Red"

select new

{

Name = product.Name,

ProductNumber = product.ProductNumber,

ListPrice = product.ListPrice

};

foreach (var product in query)

{

Console.WriteLine("Name: {0}", product.Name);

Console.WriteLine("Product number: {0}", product.ProductNumber);

Console.WriteLine("List price: ${0}", product.ListPrice);

Console.WriteLine("");

}

}

Ejemplo

En el ejemplo siguiente se utiliza el método **Where** para buscar los pedidos que se realizaron después del 1 de diciembre de 2003 y, a continuación, se utiliza la propiedad de navegación **order.SalesOrderDetail** para obtener los detalles de cada pedido.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb738636(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-5)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

IQueryable<SalesOrderHeader> query =

from order in orders

where order.OrderDate >= new DateTime(2003, 12, 1)

select order;

Console.WriteLine("Orders that were made after December 1, 2003:");

foreach (SalesOrderHeader order in query)

{

Console.WriteLine("OrderID {0} Order date: {1:d} ",

order.SalesOrderID, order.OrderDate);

foreach (SalesOrderDetail orderDetail in order.SalesOrderDetail)

{

Console.WriteLine(" Product ID: {0} Unit Price {1}",

orderDetail.ProductID, orderDetail.UnitPrice);

}

}

}

OrderBy

Ejemplo

En el ejemplo siguiente se utiliza [OrderBy](https://msdn.microsoft.com/es-es/library/system.linq.enumerable.orderby(v=vs.90).aspx) para devolver una lista de contactos organizados por apellido.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb738627(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-2)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Contact> contacts = AWEntities.Contact;

IQueryable<Contact> sortedNames =

from n in contacts

orderby n.LastName

select n;

Console.WriteLine("The sorted list of last names:");

foreach (Contact n in sortedNames)

{

Console.WriteLine(n.LastName);

}

}

Ejemplo

En el ejemplo siguiente se utiliza **OrderBy** para ordenar una lista de contactos por longitud del apellido.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb738627(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-3)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Contact> contacts = AWEntities.Contact;

IQueryable<Contact> sortedNames =

from n in contacts

orderby n.LastName.Length

select n;

Console.WriteLine("The sorted list of last names (by length):");

foreach (Contact n in sortedNames)

{

Console.WriteLine(n.LastName);

}

}

OrderByDescending

Ejemplo

En el ejemplo siguiente se utiliza **orderby… descending** (**Order By … Descending**) en Visual Basic, que es equivalente al método [OrderByDescending](https://msdn.microsoft.com/es-es/library/system.linq.enumerable.orderbydescending(v=vs.90).aspx), para ordenar el precio de venta de mayor a menor.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb738627(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-4)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Product> products = AWEntities.Product;

IQueryable<Decimal> sortedPrices =

from p in products

orderby p.ListPrice descending

select p.ListPrice;

Console.WriteLine("The list price from highest to lowest:");

foreach (Decimal price in sortedPrices)

{

Console.WriteLine(price);

}

}

ThenBy

Ejemplo

En el ejemplo siguiente se utiliza [OrderBy](https://msdn.microsoft.com/es-es/library/system.linq.queryable.orderby(v=vs.90).aspx) y [ThenBy](https://msdn.microsoft.com/es-es/library/system.linq.queryable.thenby(v=vs.90).aspx) para devolver una lista de contactos ordenados por apellido y luego por nombre.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb738627(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-5)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Contact> contacts = AWEntities.Contact;

IQueryable<Contact> sortedContacts =

from contact in contacts

orderby contact.LastName, contact.FirstName

select contact;

Console.WriteLine("The list of contacts sorted by last name then by first name:");

foreach (Contact sortedContact in sortedContacts)

{

Console.WriteLine(sortedContact.LastName + ", " + sortedContact.FirstName);

}

}

ThenByDescending

Ejemplo

En el ejemplo siguiente se utiliza **OrderBy… Descending**, que es equivalente al método [ThenByDescending](https://msdn.microsoft.com/es-es/library/system.linq.enumerable.thenbydescending(v=vs.90).aspx), para ordenar una lista de productos, primero por nombre y después por precio de venta, de mayor a menor.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb738627(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-6)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Product> products = AWEntities.Product;

IQueryable<Product> query =

from product in products

orderby product.Name, product.ListPrice descending

select product;

foreach (Product product in query)

{

Console.WriteLine("Product ID: {0} Product Name: {1} List Price {2}",

product.ProductID,

product.Name,

product.ListPrice);

}

}

Average

Ejemplo

En el ejemplo siguiente se utiliza el método **Average** para encontrar el precio de venta promedio de los productos de cada estilo.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896340(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-2)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Product> products = AWEntities.Product;

var query = from product in products

group product by product.Style into g

select new

{

Style = g.Key,

AverageListPrice =

g.Average(product => product.ListPrice)

};

foreach (var product in query)

{

Console.WriteLine("Product style: {0} Average list price: {1}",

product.Style, product.AverageListPrice);

}

}

Ejemplo

En el ejemplo siguiente se utiliza **Average** para obtener el importe total a pagar promedio para cada identificador de contacto.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896340(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-3)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

var query =

from order in orders

group order by order.Contact.ContactID into g

select new

{

Category = g.Key,

averageTotalDue = g.Average(order => order.TotalDue)

};

foreach (var order in query)

{

Console.WriteLine("ContactID = {0} \t Average TotalDue = {1}",

order.Category, order.averageTotalDue);

}

}

Ejemplo

En el ejemplo siguiente se utiliza el método **Average** para obtener el pedido con el importe total promedio a pagar de cada contacto.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896340(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-4)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

var query =

from order in orders

group order by order.Contact.ContactID into g

let averageTotalDue = g.Average(order => order.TotalDue)

select new

{

Category = g.Key,

CheapestProducts =

g.Where(order => order.TotalDue == averageTotalDue)

};

foreach (var orderGroup in query)

{

Console.WriteLine("ContactID: {0}", orderGroup.Category);

foreach (var order in orderGroup.CheapestProducts)

{

Console.WriteLine("Average total due for SalesOrderID {1} is: {0}",

order.TotalDue, order.SalesOrderID);

}

Console.Write("\n");

}

}

Count

Ejemplo

En el ejemplo siguiente se utiliza el método **Count** para devolver una lista de identificadores de contactos y el número de pedidos que tiene cada uno.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896340(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-5)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Contact> contacts = AWEntities.Contact;

//Can't find field SalesOrderContact

var query =

from contact in contacts

select new

{

CustomerID = contact.ContactID,

OrderCount = contact.SalesOrderHeader.Count()

};

foreach (var contact in query)

{

Console.WriteLine("CustomerID = {0} \t OrderCount = {1}",

contact.CustomerID,

contact.OrderCount);

}

}

Ejemplo

En el ejemplo siguiente se agrupan los productos por colores y se utiliza **Count** para devolver el número de productos de cada grupo de color.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896340(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-6)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Product> products = AWEntities.Product;

var query =

from product in products

group product by product.Color into g

select new { Color = g.Key, ProductCount = g.Count() };

foreach (var product in query)

{

Console.WriteLine("Color = {0} \t ProductCount = {1}",

product.Color,

product.ProductCount);

}

}

Max

Ejemplo

En el ejemplo siguiente se utiliza el método **Max** para obtener el mayor importe total a pagar de cada identificador de contacto.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896340(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-7)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

var query =

from order in orders

group order by order.Contact.ContactID into g

select new

{

Category = g.Key,

maxTotalDue =

g.Max(order => order.TotalDue)

};

foreach (var order in query)

{

Console.WriteLine("ContactID = {0} \t Maximum TotalDue = {1}",

order.Category, order.maxTotalDue);

}

}

Ejemplo

En el ejemplo siguiente se utiliza el método **Max** para obtener el pedido con el mayor importe total a pagar de cada identificador de contacto.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896340(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-8)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

var query =

from order in orders

group order by order.Contact.ContactID into g

let maxTotalDue = g.Max(order => order.TotalDue)

select new

{

Category = g.Key,

CheapestProducts =

g.Where(order => order.TotalDue == maxTotalDue)

};

foreach (var orderGroup in query)

{

Console.WriteLine("ContactID: {0}", orderGroup.Category);

foreach (var order in orderGroup.CheapestProducts)

{

Console.WriteLine("MaxTotalDue {0} for SalesOrderID {1}: ",

order.TotalDue,

order.SalesOrderID);

}

Console.Write("\n");

}

}

Min

Ejemplo

En el ejemplo siguiente se utiliza el método **Min** para obtener el menor importe total a pagar de cada identificador de contacto.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896340(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-9)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

var query =

from order in orders

group order by order.Contact.ContactID into g

select new

{

Category = g.Key,

smallestTotalDue =

g.Min(order => order.TotalDue)

};

foreach (var order in query)

{

Console.WriteLine("ContactID = {0} \t Minimum TotalDue = {1}",

order.Category, order.smallestTotalDue);

}

}

Ejemplo

En el ejemplo siguiente se utiliza el método **Min** para obtener el pedido con el menor importe total a pagar de cada contacto.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896340(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-10)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

var query =

from order in orders

group order by order.Contact.ContactID into g

let minTotalDue = g.Min(order => order.TotalDue)

select new

{

Category = g.Key,

smallestTotalDue =

g.Where(order => order.TotalDue == minTotalDue)

};

foreach (var orderGroup in query)

{

Console.WriteLine("ContactID: {0}", orderGroup.Category);

foreach (var order in orderGroup.smallestTotalDue)

{

Console.WriteLine("Mininum TotalDue {0} for SalesOrderID {1}: ",

order.TotalDue,

order.SalesOrderID);

}

Console.Write("\n");

}

}

Sum

Ejemplo

En el ejemplo siguiente se utiliza el método **Sum** para obtener el menor importe total a pagar de cada identificador de contacto.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896340(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-11)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

var query =

from order in orders

group order by order.Contact.ContactID into g

select new

{

Category = g.Key,

TotalDue = g.Sum(order => order.TotalDue)

};

foreach (var order in query)

{

Console.WriteLine("ContactID = {0} \t TotalDue sum = {1}",

order.Category, order.TotalDue);

}

}

# Ejemplos de sintaxis de expresiones de consultas: crear particiones

Skip

Ejemplo

En el ejemplo siguiente se utiliza el método **Skip** para obtener las dos primeras direcciones de Seattle.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896296(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-2)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Address> addresses = AWEntities.Address;

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

//LINQ to Entities only supports Skip on ordered collections.

var query = (

from address in addresses

from order in orders

where address.AddressID == order.Address.AddressID

&& address.City == "Seattle"

orderby order.SalesOrderID

select new

{

City = address.City,

OrderID = order.SalesOrderID,

OrderDate = order.OrderDate

}).Skip(2);

Console.WriteLine("All but first 2 orders in Seattle:");

foreach (var order in query)

{

Console.WriteLine("City: {0} Order ID: {1} Total Due: {2:d}",

order.City, order.OrderID, order.OrderDate);

}

Take

Ejemplo

En el ejemplo siguiente se utiliza el método **Take** para obtener las tres primeras direcciones de Seattle.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896296(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-3)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Address> addresses = AWEntities.Address;

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

var query = (

from address in addresses

from order in orders

where address.AddressID == order.Address.AddressID

&& address.City == "Seattle"

select new

{

City = address.City,

OrderID = order.SalesOrderID,

OrderDate = order.OrderDate

}).Take(3);

Console.WriteLine("First 3 orders in Seattle:");

foreach (var order in query)

{

Console.WriteLine("City: {0} Order ID: {1} Total Due: {2:d}",

order.City, order.OrderID, order.OrderDate);

}

}

Combinaciones

GroupJoin

Ejemplo

El ejemplo siguiente realiza una **GroupJoin** en las tablas SalesOrderHeader y SalesOrderDetail para buscar el número de pedidos por cliente. Una combinación de grupo es el equivalente a una combinación externa izquierda, que devuelve cada elemento del primer origen de datos (izquierdo), incluso si no hay elementos correlacionados en el otro origen de datos.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896266(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-2)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

ObjectQuery<SalesOrderDetail> details = AWEntities.SalesOrderDetail;

var query =

from order in orders

join detail in details

on order.SalesOrderID

equals detail.SalesOrderID into orderGroup

select new

{

CustomerID = order.SalesOrderID,

OrderCount = orderGroup.Count()

};

foreach (var order in query)

{

Console.WriteLine("CustomerID: {0} Orders Count: {1}",

order.CustomerID,

order.OrderCount);

}

}

Ejemplo

En el ejemplo siguiente se realiza una **GroupJoin** en las tablas Contact y SalesOrderHeader para buscar el número de pedidos por contacto. Se muestran el recuento de pedidos y los identificadores para cada contacto.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896266(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-3)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Contact> contacts = AWEntities.Contact;

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

var query =

from contact in contacts

join order in orders

on contact.ContactID

equals order.Contact.ContactID into contactGroup

select new

{

ContactID = contact.ContactID,

OrderCount = contactGroup.Count(),

Orders = contactGroup.Select(order => order)

};

foreach (var group in query)

{

Console.WriteLine("ContactID: {0}", group.ContactID);

Console.WriteLine("Order count: {0}", group.OrderCount);

foreach (var orderInfo in group.Orders)

{

Console.WriteLine(" Sale ID: {0}", orderInfo.SalesOrderID);

}

Console.WriteLine("");

}

}

Ejemplo

En el ejemplo siguiente se realiza una **GroupJoin** en las tablas Contact y SalesOrderHeader. Una combinación de grupo es el equivalente a una combinación externa izquierda, que devuelve cada elemento del primer origen de datos (izquierdo), incluso si no hay elementos correlacionados en el otro origen de datos.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896266(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-4)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Contact> contacts = AWEntities.Contact;

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

var query =

from contact in contacts

join order in orders

on contact.ContactID

equals order.Contact.ContactID into contactGroup

select new

{

ContactID = contact.ContactID,

OrderCount = contactGroup.Count(),

Orders = contactGroup.Select(order => order)

};

foreach (var group in query)

{

Console.WriteLine("ContactID: {0}", group.ContactID);

Console.WriteLine("Order count: {0}", group.OrderCount);

foreach (var orderInfo in group.Orders)

{

Console.WriteLine(" Sale ID: {0}", orderInfo.SalesOrderID);

}

Console.WriteLine("");

}

}

Join

Ejemplo

El ejemplo siguiente realiza una combinación en las tablas SalesOrderHeader y SalesOrderDetail para obtener los pedidos en línea del mes de agosto.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896266(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-5)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

ObjectQuery<SalesOrderDetail> details = AWEntities.SalesOrderDetail;

var query =

from order in orders

join detail in details

on order.SalesOrderID equals detail.SalesOrderID

where order.OnlineOrderFlag == true

&& order.OrderDate.Month == 8

select new

{

SalesOrderID = order.SalesOrderID,

SalesOrderDetailID = detail.SalesOrderDetailID,

OrderDate = order.OrderDate,

ProductID = detail.ProductID

};

foreach (var order in query)

{

Console.WriteLine("{0}\t{1}\t{2:d}\t{3}",

order.SalesOrderID,

order.SalesOrderDetailID,

order.OrderDate,

order.ProductID);

}

}

First

Ejemplo

En el ejemplo siguiente se usa el método **First** para devolver el primer contacto cuyo nombre es "Brooke".

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896262(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-2)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Contact> contacts = AWEntities.Contact;

Contact query = (

from contact in contacts

where contact.FirstName == "Brooke"

select contact)

.First();

Console.WriteLine("ContactID: " + query.ContactID);

Console.WriteLine("FirstName: " + query.FirstName);

Console.WriteLine("LastName: " + query.LastName);

}

Agrupamiento

Ejemplo

El ejemplo siguiente devuelve los objetos **Address** agrupados por código postal. Los resultados se proyectan en un tipo anónimo.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896250(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-2)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Address> addresses = AWEntities.Address;

var query =

from address in addresses

group address by address.PostalCode into addressGroup

select new { PostalCode = addressGroup.Key,

AddressLine = addressGroup };

foreach (var addressGroup in query)

{

Console.WriteLine("Postal Code: {0}", addressGroup.PostalCode);

foreach (var address in addressGroup.AddressLine)

{

Console.WriteLine("\t" + address.AddressLine1 +

address.AddressLine2);

}

}

}

Ejemplo

El ejemplo siguiente devuelve los objetos **Contact** agrupados por la primera letra del apellido del contacto. Los resultados se ordenan también por la primera letra del apellido y se proyectan en un tipo anónimo.

C#

**[VB](https://msdn.microsoft.com/es-es/library/bb896250(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb" \l "code-snippet-3)**

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Contact> contacts = AWEntities.Contact;

var query = (

from contact in contacts

group contact by contact.LastName.Substring(0, 1) into contactGroup

select new { FirstLetter = contactGroup.Key, Names = contactGroup }).

OrderBy(letter => letter.FirstLetter);

foreach (var contact in query)

{

Console.WriteLine("Last names that start with the letter '{0}':",

contact.FirstLetter);

foreach (var name in contact.Names)

{

Console.WriteLine(name.LastName);

}

}

}

Ejemplo

En el ejemplo siguiente se devuelven objetos **SalesOrderHeader** agrupados por identificador de cliente. También se devuelve el número de ventas para cada cliente.

C#

**[VB](https://msdn.microsoft.com/es-es/library/bb896250(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb" \l "code-snippet-4)**

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<SalesOrderHeader> salesOrders = AWEntities.SalesOrderHeader;

var query = from order in salesOrders

group order by order.CustomerID into idGroup

select new {CustomerID = idGroup.Key,

OrderCount = idGroup.Count(),

Sales = idGroup};

foreach (var orderGroup in query)

{

Console.WriteLine("Customer ID: {0}", orderGroup.CustomerID);

Console.WriteLine("Order Count: {0}", orderGroup.OrderCount);

foreach (SalesOrderHeader sale in orderGroup.Sales)

{

Console.WriteLine(" Sale ID: {0}", sale.SalesOrderID);

}

Console.WriteLine("");

}

}

Ejemplo

En el ejemplo siguiente se utiliza el método [Select](https://msdn.microsoft.com/es-es/library/system.linq.queryable.select(v=vs.90).aspx) para obtener todos los identificadores de contactos y la suma del importe total a pagar de cada contacto cuyo nombre sea "Zhou". La propiedad de navegación **Contact.SalesOrderHeader** se utiliza para obtener la colección de objetos **SalesOrderHeader** para cada contacto. El método **Sum** usa la propiedad de navegación **Contact.SalesOrderHeader** para sumar el importe total a pagar de todos los pedidos de cada contacto.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896243(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-2)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Contact> contacts = AWEntities.Contact;

var ordersQuery = from contact in contacts

where contact.LastName == "Zhou"

select new

{

ContactID = contact.ContactID,

Total = contact.SalesOrderHeader.Sum(o => o.TotalDue)

};

foreach (var contact in ordersQuery)

{

Console.WriteLine("Contact ID: {0} Orders total: {1}", contact.ContactID, contact.Total);

}

}

Ejemplo

El ejemplo siguiente obtiene todos los pedidos de los contactos cuyo apellido sea "Zhou". La propiedad de navegación **Contact.SalesOrderHeader** se utiliza para obtener la colección de objetos **SalesOrderHeader** para cada contacto. El nombre y los pedidos del contacto se devuelven en un tipo anónimo.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896243(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-3)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<Contact> contacts = AWEntities.Contact;

var ordersQuery = from contact in contacts

where contact.LastName == "Zhou"

select new { LastName = contact.LastName, Orders = contact.SalesOrderHeader };

foreach (var order in ordersQuery)

{

Console.WriteLine("Name: {0}", order.LastName);

foreach (SalesOrderHeader orderInfo in order.Orders)

{

Console.WriteLine("Order ID: {0}, Order date: {1}, Total Due: {2}",

orderInfo.SalesOrderID, orderInfo.OrderDate, orderInfo.TotalDue);

}

Console.WriteLine("");

}

}

Ejemplo

En el ejemplo siguiente se usan las propiedades de navegación **SalesOrderHeader.Address** y **SalesOrderHeader.Contact** para obtener la colección de los objetos **Address** y **Contact** asociados a cada pedido. El apellido del contacto, la dirección postal, el número de pedidos de venta y el importe total a pagar de cada pedido para la ciudad de Seattle se devuelven en un tipo anónimo.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/bb896243(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-4)

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

var ordersQuery = from order in orders

where order.Address.City == "Seattle"

select new

{

ContactLastName = order.Contact.LastName,

ContactFirstName = order.Contact.FirstName,

StreetAddress = order.Address.AddressLine1,

OrderNumber = order.SalesOrderNumber,

TotalDue = order.TotalDue

};

foreach (var orderInfo in ordersQuery)

{

Console.WriteLine("Name: {0}, {1}", orderInfo.ContactLastName, orderInfo.ContactFirstName);

Console.WriteLine("Street address: {0}", orderInfo.StreetAddress);

Console.WriteLine("Order number: {0}", orderInfo.OrderNumber);

Console.WriteLine("Total Due: {0}", orderInfo.TotalDue);

Console.WriteLine("");

}

}

Ejemplo

En el ejemplo siguiente se utiliza el método **Where** para buscar los pedidos que se realizaron después del 1 de diciembre de 2003 y, a continuación, se utiliza la propiedad de navegación **order.SalesOrderDetail** para obtener los detalles de cada pedido.

C#

**[VB](https://msdn.microsoft.com/es-es/library/bb896243(v=vs.90).aspx?cs-save-lang=1&cs-lang=vb" \l "code-snippet-5)**

using (AdventureWorksEntities AWEntities = new AdventureWorksEntities())

{

ObjectQuery<SalesOrderHeader> orders = AWEntities.SalesOrderHeader;

IQueryable<SalesOrderHeader> query =

from order in orders

where order.OrderDate >= new DateTime(2003, 12, 1)

select order;

Console.WriteLine("Orders that were made after December 1, 2003:");

foreach (SalesOrderHeader order in query)

{

Console.WriteLine("OrderID {0} Order date: {1:d} ",

order.SalesOrderID, order.OrderDate);

foreach (SalesOrderDetail orderDetail in order.SalesOrderDetail)

{

Console.WriteLine(" Product ID: {0} Unit Price {1}",

orderDetail.ProductID, orderDetail.UnitPrice);

}

}

}