public class Movie

{

public string Title { get; set; }

public string Director { get; set; }

public int Genre { get; set; }

public int RunTime { get; set; }

public DateTime ReleaseDate { get; set; }

}

public class Grouping

{

public int ID { get; set; }

public int MovieCount { get; set; }

}

public class Genre

{

public int Genre { get; set; }

public string Name { get; set; }

}

public List<Movie> GetMovies()

{

return new List<Movie> {

new Movie {Title = ‘Shrek’, Director=’Adam’, Genre=0, ReleaseDate=DateTime.Parse(‘5/16/2011’), Runtime=89 },

new Movie {Title = ‘Fetch’, Director=’Tom’, Genre=1, ReleaseDate=DateTime.Parse(‘1/16/2013’), Runtime=95 },

new Movie {Title = ‘Boat’, Director=’John’, Genre=0, ReleaseDate=DateTime.Parse(‘5/26/2015’), Runtime=90 },

new Movie {Title = ‘Storm’, Director=’James’, Genre=1, ReleaseDate=DateTime.Parse(‘7/16/2012’), Runtime=80 }

};

}

public List<Genre> GetGenres()

{

return new List<Genre> {

new Genre { ID= 0, Name = “Comedy” },

new Genre { ID= 1, Name = “Action” },

new Genre { ID= 2, Name = “Drama” }

};

}

<?xml version=”1.0” encoding=”utf-8” ?>

<Movies>

<Movie>

<Title>Shreck</Title>

<Director>Adam</Director>

<ReleaseDate>5/16/2011 </ReleaseDate>

<Runtime>80</Runtime>

</Movie>

<Movie>

<Title>Boat</Title>

<Director>John</Director>

<ReleaseDate>5/16/2011 </ReleaseDate>

<Runtime>80</Runtime>

</Movie>

</Movies>

ALL

var movies = GetMovies();

this.GridView1.DataSource = movies;

this.GridView1.DataBind();

FILTER

var movies = GetMovies();

var query = new List<Movie>();

foreach (var m in movies)

{

if (m.Genre ==0 ) query.Add(m);

}

this.GridView1.DataSource = query;

this.GridView1.DataBind();

MAS SIMPLE \*\*\*

movies = movies.Where(x => x.Genre == 0).ToList();

REMOVE

movies.RemoveAll(x => x.Director == "Bob");

GROUPING, HOW MANY MOVIES BY GENRE

var movies = GetMovies();

Dictionary<int, Grouping> groups = new Dictionary<int, Grouping>();

foreach (Movie m in movies)

{

If (!groups.ContainsKey(m.Genre))

{

groups(m.Genre) = new Grouping {Genre = m.Genre, MovieCount = 0 };

}

groups( m.Genre).MovieCount++;

}

List<Grouping> results = new List<Grouping>(groups.Values);

result.Sort(delegate(Grouping x, Grouping y)

{

x.MovieCount > y.MovieCount ? -1:

x.MovieCount < y.MovieCount ? 1:

0;

});

Mas simple

var results = from m in movies

group m by m.Genre into g

select new

{

MovieTitle = g.Key,

MovieGenre = g.Count()

};

this.GridView1.DataSource = results;

this.GridView1.DataBind();

GROUP AND ORDER DE VARIOS

var newSort = from row in dt.AsEnumerable()

group row by new { ID = row.Field<int>("client\_id"),

nombre = row.Field<string>("client\_descr")

} into grp

orderby grp.Key

select new

{

client\_id = grp.Key.ID,

client\_descr = grp.Key.nombre

Sum = grp.Sum(r => Convert.ToDecimal(r.ItemArray[2]))

};

SELECT, STEP BY STEP

var query = from m in movies

select m;

var query = from m in movies

select new

{

m.Title,

m.Genre

};

var query = from m in movies

select new

{

MovieTitle = m.Title,

MovieGenre = m.Genre

};

var query = from m in movies

orderby m.Title descending

select new

{

MovieTitle = m.Title,

MovieGenre = m.Genre

};

var query = from m in movies

where m.Genre ==0 && m.RunTime > 90

select m;

var query = from m in movies

group m by m.Genre into g

select new

{

MovieTitle = g.Key,

MovieGenre = g.Count()

};

OPERATORS

TotalMovies.Text = movies.Count.ToString();

LongestRunTime.Text = movies.Max( m => m.RunTime).ToString();

ShortestRunTime.Text = movies.Min( m => m.RunTime).ToString();

AverageRunTime.Text = movies.Average( m => m.RunTime).ToString();

SEARCH IN LIST

var result = Movies.FirstOrDefault(a => a.Director == “John”);

MovieName.Text = result.Title.ToString();

Release.Text = result.ReleaseDate.ToString();

SEARCH DATATABLE

DataRow rw = dt2.AsEnumerable().FirstOrDefault(a => a.Field<string>("User Name") == UserID);

if (rw != null)

{

GridView1.Rows[1].Cells[5].Value = rw[3].ToString();

}

LINQ JOINS

var movies = GetMovies();

var genres = GetGenres();

var query = from m in movies

join g in genres on m.Genre equals g.ID

select new

{

m.Title,

g.Name

};

PAGING

var query = ( from m in movies

join g in genres on m.Genre equals g.ID

select new

{

m.Title,

g.Name

} ). Skip(5).Take(3);

XML

var query = from m in XElement.Load(MapPath(“Movies.xml”)).Elements(“Movie”)

select m;

var query = from m in XElement.Load(MapPath(“Movies.xml”)).Elements(“Movie”)

select new Movie

{

Title = (string)m.Element(“Title”),

Release = (DateTime)m.Element(“ReleaseDate”)

};

JOIN XML

var query = from m in XElement.Load(MapPath(“Movies.xml”)).Elements(“Movie”)

join g in XElement.Load(MapPath(“Genres.xml”)).Elements(“Genre”)

on (int)m.Element(“Genre”) equals (int)g.Element(“ID”)

select new

{

Title = (string)m.Element(“Title”),

Genre = (string)g.Element(“Name”)

};

DATA TABLE

DataSet ds = (DataSet)Cache["TeamLeaders"];

DataTable dt = ds.Tables[0];

var newSort2 = from row in dt.AsEnumerable()

where row.Field<object>("client\_id") != null

orderby row.Field<object>("client\_descr")

group row by new

{

ID = Convert.ToString(row.Field<object>("client\_id")),

nombre = row.Field<string>("client\_descr")

} into grp

select new

{

ids = Convert.ToString(grp.Key.ID),

nombres = grp.Key.nombre

};

//foreach (var movie in newSort2)

//{

// string i = movie.ids;

// string ss = movie.nombres;

//}

ddlClient.DataSource = newSort2;

ddlClient.DataValueField = "ids";

ddlClient.DataTextField = "nombres";

ddlClient.DataBind();

ListItem li = new ListItem("Select One", "-1");

ddlClient.Items.Insert(0, li);

DATATABLE JOIN WHERE

DataTable orders = ds.Tables["SalesOrderHeader"];

DataTable details = ds.Tables["SalesOrderDetail"];

var query =

from order in orders.AsEnumerable()

join detail in details.AsEnumerable()

on order.Field<int>("OrderID") equals detail.Field<int>("SalesOrderID")

where order.Field<bool>("OnlineOrderFlag") == true

&& order.Field<DateTime>("OrderDate").**Month** == 8

select new

{

OrderID = order.Field<int>("SalesOrderID"),

OrderDate = order.Field<DateTime>("OrderDate"),

ProductID = detail.Field<int>("ProductID")

};

DataTable orderTable = query.**CopyToDataTable**();