

Introduction to LINQ (4): Language-Integrated Query



Relational DB Queries with LINQ

//This is what we know already:

```
string cn = @"SERVER=140.161.112.145;DATABASE=classicmodels;UID=luca;PASSWORD='luca';";
```

```
MySqlConnection conn = null; //create the connection
```

```
try
```

```
{
```

```
    conn = new MySqlConnection(cn); //conn object
```

```
    conn.Open(); //open connection
```

```
    Console.WriteLine("Ping Test = {0}",pt);
```

```
    string query = "select customerName from customers;";//sql query
```

```
MySqlCommand cmd = new MySqlCommand(query, conn);
```

```
cmd.Prepare();
```

```
cmd.ExecuteNonQuery();
```

```
MySqlDataReader rdr = null; //Data Reader
```

```
rdr = cmd.ExecuteReader();
```

```
while (rdr.Read())
```

```
{
```

```
    Console.WriteLine(rdr.GetString(0));
```

```
}
```

```
rdr.Close();
```

```
cmd.Dispose();
```

```
.....
```

```
}
```

Relational DB Queries with LINQ

This example brings together many concepts we have encountered in the previous slides: connections, tables, LINQ queries.

```
//create a data adapter with an sql command and the connection
MySqlDataAdapter da = new MySqlDataAdapter("select * from customers",conn);

//create a dataset
DataSet ds = new DataSet();

//fill the data adapter and a table in the dataset
da.Fill(ds,"customers");
DataTable table = ds.Tables["customers"];

//now we can run our LINQ-style query:
var qr = from e in table.AsEnumerable()
        where e.Field<int>("customerNumber") > 1
        select e;

foreach (var x in qr)
    Console.WriteLine(x.Field<string>("customerName"));
```

An additional LINQ example: Strings

Besides IEnumerable-derived objects, LINQ works well also in arrays. It works also on strings (arrays of characters). Here some examples:

```
var mystring = "I really love LINQ!";
```

```
var query1 = from c in mystring
              select c;
```

```
var query2 = from c in mystring
              where c=='a' || c=='l'
              select c;
```

```
var query3 = from c in mystring.ToLower()
              where c=='a' || c=='l'
              select c;
```

```
var query4 = from c in mystring.ToLower()
              where c=='a' || c=='l'
              orderby c ascending //or descending
              select c;
```