

易百教程

- 所有教程
 - [Java技术](#)
 - [Web技术](#)
 - [脚本语言](#)
 - [数据库](#)
 - [高级语言](#)
 - [其它技术](#)
 - [查看所有教程](#)
- [Java技术](#)
 - [Java教程](#)
 - [Java设计模式](#)
 - [JSP教程](#)
 - [JDBC教程](#)
 - [Java XML教程](#)
 - [JUnit教程](#)
 - [Struts2教程](#)
 - [Maven教程](#)
 - [java实例教程](#)
 - [MyBatis教程](#)
 - [Spring教程](#)
 - [Spring MVC教程](#)
 - [Gradle教程](#)
- [数据库](#)
 - [MySQL教程](#)
 - [CouchDB教程](#)
 - [Cassandra教程](#)
 - [MariaDB教程](#)
 - [PL/SQL教程](#)

- [PostgreSQL教程](#)
 - [MongoDB教程](#)
 - [SQLite教程](#)
 - [DB2教程](#)
 - [Redis教程](#)
 - [Memcached教程](#)
 - [Access教程](#)
 - [SQL教程](#)
 - [SQL Server教程](#)
- [脚本语言](#)
 - [WxPython教程](#)
 - [Ruby教程](#)
 - [PHP教程](#)
 - [Python教程](#)
 - [Lua教程](#)
 - [Tcl教程](#)
 - [Tk教程](#)
 - [NumPy教程](#)
 - [Python3教程](#)
 - [PHP7教程](#)
- [极客文章](#)

[登录](#)

□

□

- [_LinQ教程](#)
- [LINQ环境安装设置](#)
- [LINQ查询运算符](#)
- [LINQ过滤运算符](#)
- [LINQ Join操作](#)
- [LINQ投影操作](#)
- [LINQ排序运算符](#)
- [LINQ分组操作](#)

- [LINQ转换操作](#)
- [LINQ级联](#)
- [LINQ聚合](#)
- [LINQ量词操作](#)
- [LINQ分区操作符](#)
- [LINQ生成操作](#)
- [LINQ Set操作](#)
- [LINQ相等](#)
- [LINQ元素操作符](#)
- **[LINQ SQL](#)**
- [LINQ对象](#)
- [LINQ Dataset \(数据集 \)](#)
- [LINQ XML](#)
- [LINQ实体](#)
- [LINQ Lambda表达式](#)
- [LINQ ASP.Net](#)

LINQ SQL

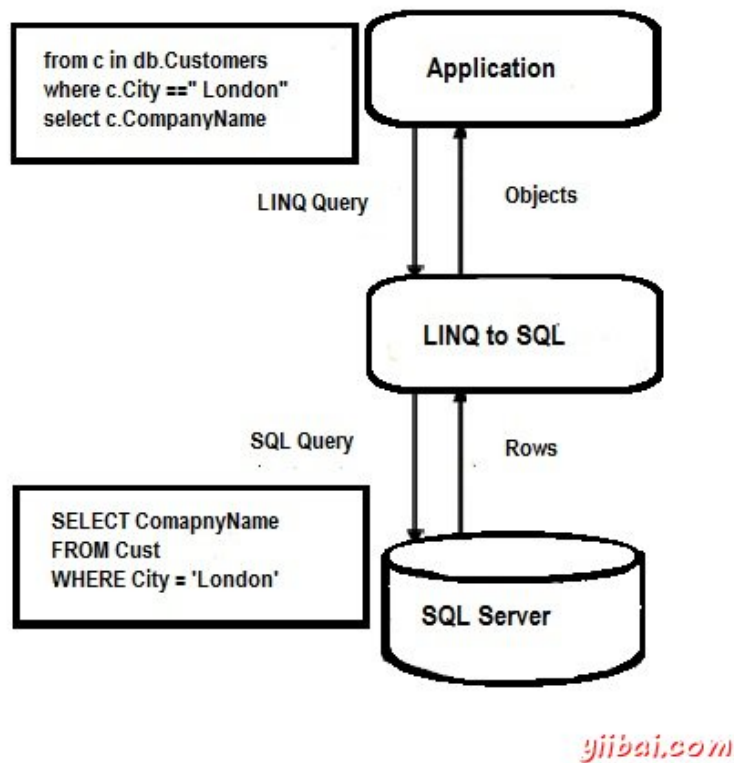
¥ 我要打赏 [LinQ教程](#) 作者：dhrtcv 评论：0 条 **Java技术QQ群：227270512**

LINQ为SQL提供了一个基础结构(运行时间)，关系数据作为对象的管理。这是3.5版本的.NET框架的一个组成部分，并巧妙地做对象模型到SQL的语言集成查询的转换。这些查询被发送给数据库来执行。从数据库中获得的结果后，LINQ到SQL再次它们转换为对象。

LINQ到SQL简介

对于大多数ASP.NET开发，LINQ到SQL（也称为DLINQ）是Language集成查询通用部分，因为这允许在SQL服务器数据库查询数据通过使用常规的LINQ表达式。它还允许更新，删除和插入数据，但它受到的唯一缺点是它受SQL服务器数据库的限制。但是，也有LINQ的许多好处，SQL在ADO.NET一样降低了复杂性，编码等等。

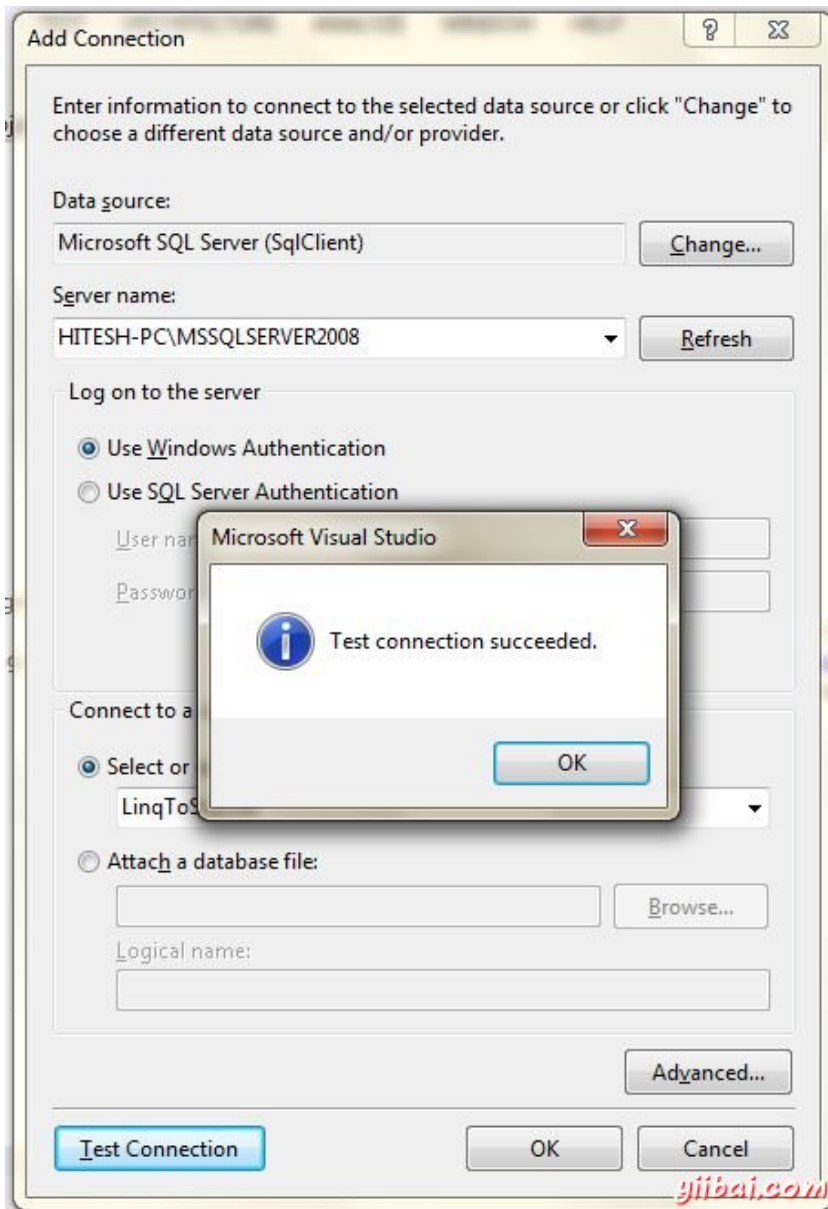
下面是表示LINQ执行架构到SQL的示意图。



LINQ SQL Architecture

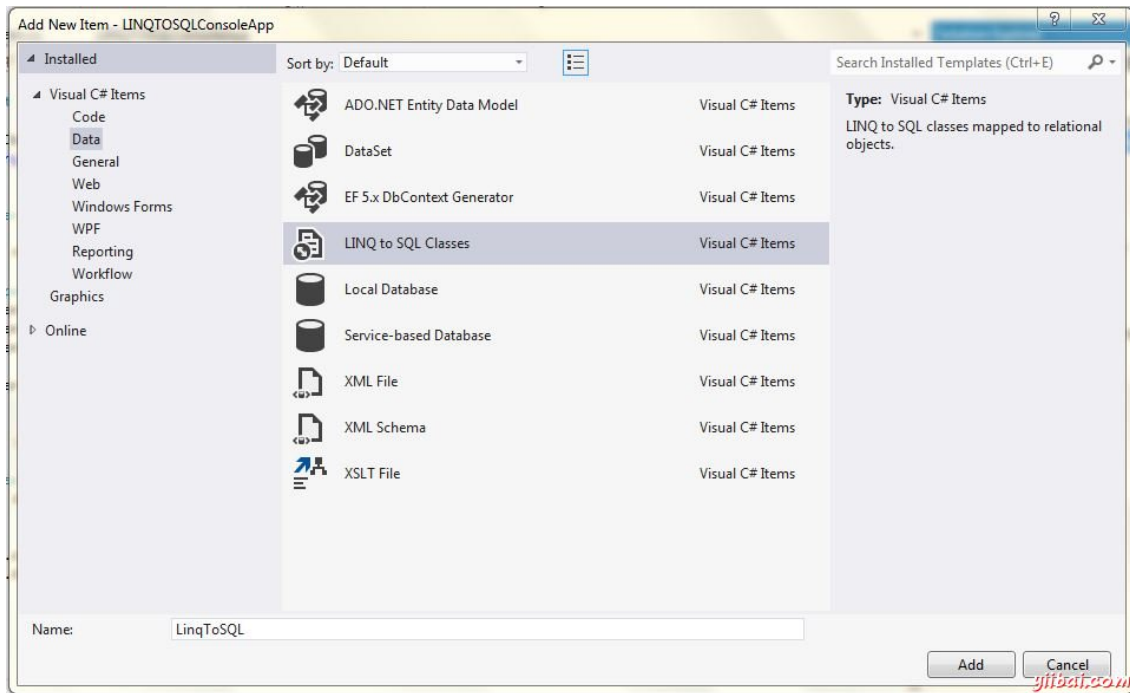
如何使用LINQ到SQL?

- 步骤 1: 创建一个新的“数据连接”到数据库服务器。View -> Server Explorer -> Data Connections -> Add Connection



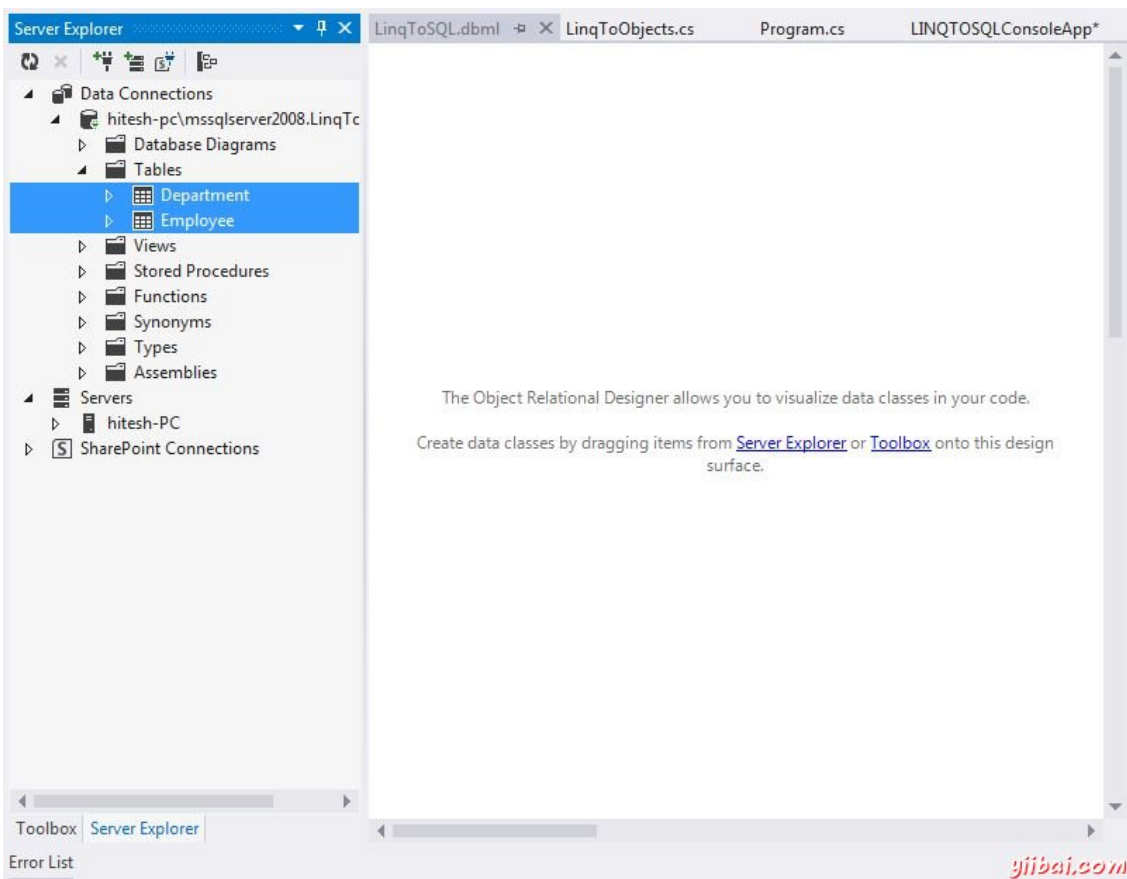
LINQ to SQL

- 步骤2: 添加LINQ 到 SQL类文件



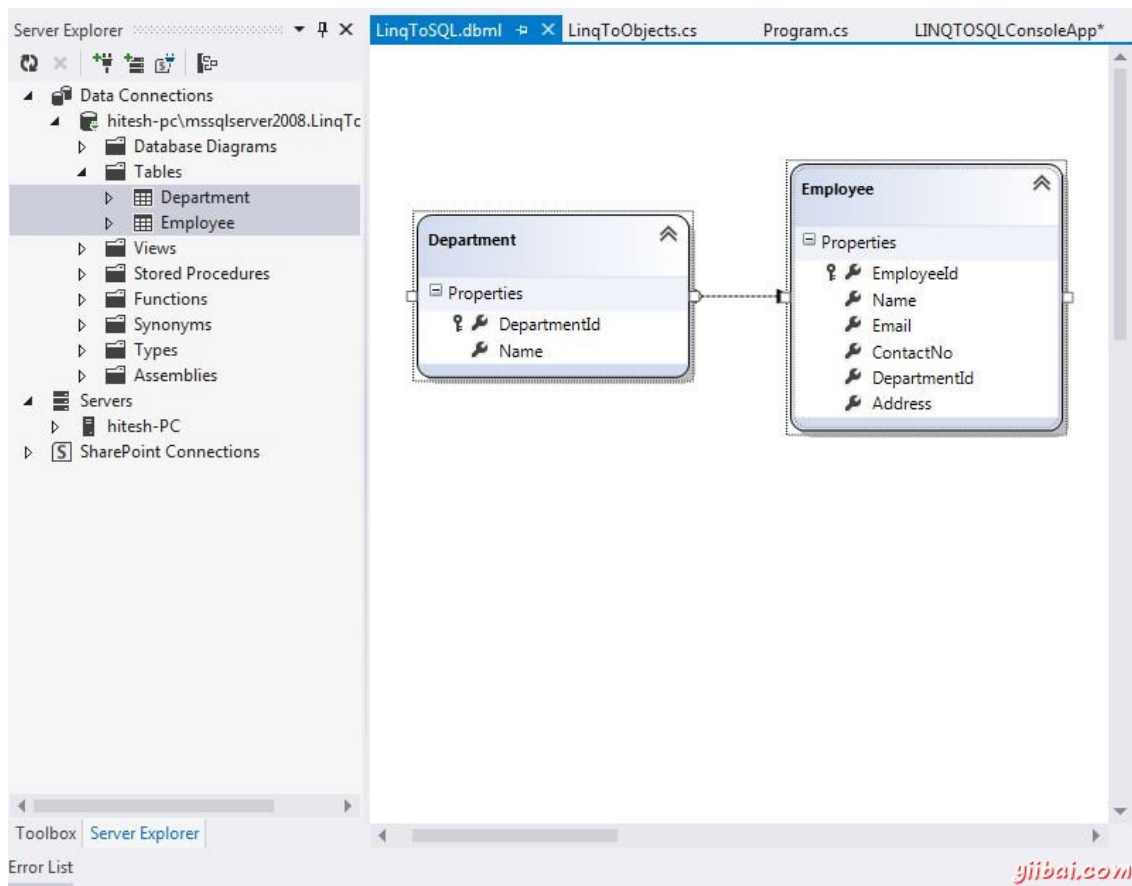
LINQ to SQL

- 步骤 3: 选择数据库并拖动表格拖放到新的LINQ 到 SQL类文件。



LINQ to SQL

- 步骤 4: 添加表类文件。



LINQ to SQL

使用LINQ到SQL查询

执行一个查询使用LINQ到SQL规则类似到一个标准的LINQ查询，即 执行查询或者延迟或马上执行。在执行使用LINQ到SQL查询的发挥作用，这些都是下述的组件。

- **LINQ到SQL API** – 请求查询执行代表一个应用程序，并把它交给LINQ到SQL提供程序
- **LINQ 到SQL提供程序** - 转换查询到Transact SQL (T-SQL)，并发送新的查询到ADO提供程序执行
- **ADO 提供程序** - 执行查询之后，发送结果在一个DataReader形式的LINQ 到 SQL，提供这反过来将其转换成用户的对象的形式

应当指出的是，在进行一个LINQ到SQL查询之前，重要的是要连接到经由DataContext类的数据源。

使用LINQ 到SQL插入，更新和删除

添加或插入

C#

```
using System;
using System.Linq;

namespace LINQtoSQL
{
    class LinqToSQLCRUD
```

```

{
    static void Main(string[] args)
    {
        string connectionString =
System.Configuration.ConfigurationManager.ConnectionStrings["LinqToSQLDBConnectionString"].ToString();

        LinqToSQLDataContext db = new LinqToSQLDataContext(connectionString);

        //Create new Employee
        Employee newEmployee = new Employee();
        newEmployee.Name = "Michael";
        newEmployee.Email = "yourname@companyname.com";
        newEmployee.ContactNo = "343434343";
        newEmployee.DepartmentId = 3;
        newEmployee.Address = "Michael - USA";

        //Add new Employee to database
        db.Employees.InsertOnSubmit(newEmployee);

        //Save changes to Database.
        db.SubmitChanges();

        //Get new Inserted Employee
        Employee insertedEmployee = db.Employees.FirstOrDefault(e => e.Name.Equals("Michael"));

        Console.WriteLine("Employee Id = {0} , Name = {1}, Email = {2}, ContactNo = {3}, Address = {4}",
            insertedEmployee.EmployeeId, insertedEmployee.Name, insertedEmployee.Email,
            insertedEmployee.ContactNo, insertedEmployee.Address);

        Console.WriteLine("\nPress any key to continue.");
        Console.ReadKey();
    }
}

```

VB

Module Module1

```

Sub Main()
    Dim connectionString As String =
System.Configuration.ConfigurationManager.ConnectionStrings["LinqToSQLDBConnectionString"].ToString()

    Dim db As New LinqToSQLDataContext(connectionString)

    Dim newEmployee As New Employee()
    newEmployee.Name = "Michael"
    newEmployee.Email = "yourname@companyname.com"
    newEmployee.ContactNo = "343434343"
    newEmployee.DepartmentId = 3
    newEmployee.Address = "Michael - USA"

    db.Employees.InsertOnSubmit(newEmployee)

    db.SubmitChanges()

    Dim insertedEmployee As Employee = db.Employees.FirstOrDefault(Function(e)
e.Name.Equals("Michael"))

    Console.WriteLine("Employee Id = {0} , Name = {1}, Email = {2}, ContactNo = {3}, Address = {4}",

```



```
insertedEmployee.EmployeeId, insertedEmployee.Name, insertedEmployee.Email,  
insertedEmployee.ContactNo, insertedEmployee.Address)
```

```
    Console.WriteLine(vbLf & "Press any key to continue.")  
    Console.ReadKey()  
End Sub  
End Module
```

当C#或VB上面的代码被编译并运行，它会产生以下结果：

```
Employee ID = 4, Name = Michael, Email = yourname@companyname.com, ContactNo =  
343434343, Address = Michael - USA
```

Press any key to continue.

更新

C#

```
using System;  
using System.Linq;  
  
namespace LINQtoSQL  
{  
    class LinqToSQLCRUD  
    {  
        static void Main(string[] args)  
        {  
            string connectionString =  
System.Configuration.ConfigurationManager.ConnectionStrings["LinqToSQLDBConnectionString"].ToString();  
  
            LinqToSQLDataContext db = new LinqToSQLDataContext(connectionString);  
  
            //Get Employee for update  
            Employee employee = db.Employees.FirstOrDefault(e => e.Name.Equals("Michael"));  
  
            employee.Name = "George Michael";  
            employee.Email = "yourname@companyname.com";  
            employee.ContactNo = "99999999";  
            employee.DepartmentId = 2;  
            employee.Address = "Michael George - UK";  
  
            //Save changes to Database.  
            db.SubmitChanges();  
  
            //Get Updated Employee  
            Employee updatedEmployee = db.Employees.FirstOrDefault(e => e.Name.Equals("George Michael"));  
  
            Console.WriteLine("Employee Id = {0} , Name = {1}, Email = {2}, ContactNo = {3}, Address = {4}",  
                updatedEmployee.EmployeeId, updatedEmployee.Name, updatedEmployee.Email,  
                updatedEmployee.ContactNo, updatedEmployee.Address);  
  
            Console.WriteLine("\nPress any key to continue.");  
            Console.ReadKey();  
        }  
    }  
}
```

VB

Module Module1

```
Sub Main()  
    Dim connectionString As String =  
System.Configuration.ConfigurationManager.ConnectionStrings("LinqToSQLDBConnectionString").ToString()  
  
    Dim db As New LinqToSQLDataContext(connectionString)  
  
    Dim employee As Employee = db.Employees.FirstOrDefault(Function(e) e.Name.Equals("Michael"))  
  
    employee.Name = "George Michael"  
    employee.Email = "yourname@companyname.com"  
    employee.ContactNo = "999999999"  
    employee.DepartmentId = 2  
    employee.Address = "Michael George - UK"  
  
    db.SubmitChanges()  
  
    Dim updatedEmployee As Employee = db.Employees.FirstOrDefault(Function(e) e.Name.Equals("George  
Michael"))  
  
    Console.WriteLine("Employee Id = {0} , Name = {1}, Email = {2}, ContactNo = {3}, Address = {4}",  
updatedEmployee.EmployeeId, updatedEmployee.Name, updatedEmployee.Email,  
updatedEmployee.ContactNo, updatedEmployee.Address)  
  
    Console.WriteLine(vbLf & "Press any key to continue.")  
    Console.ReadKey()  
End Sub  
End Module
```

当C#或VB上面的代码被编译并运行，它会产生以下结果：

```
Employee ID = 4, Name = George Michael, Email = yourname@companyname.com, ContactNo =  
999999999, Address = Michael George - UK
```

Press any key to continue.

删除

C#

```
using System;  
using System.Linq;  
  
namespace LinqtoSQL  
{  
    class LinqToSQLCRUD  
    {  
        static void Main(string[] args)  
        {  
            string connectionString =  
System.Configuration.ConfigurationManager.ConnectionStrings["LinqToSQLDBConnectionString"].ToString();  
  
            LinqToSQLDataContext db = new LinqToSQLDataContext(connectionString);  
  
            //Get Employee to Delete  
            Employee deleteEmployee = db.Employees.FirstOrDefault(e => e.Name.Equals("George Michael"));  
  
            //Delete Employee  
            db.Employees.DeleteOnSubmit(deleteEmployee);  
        }  
    }  
}
```

```

//Save changes to Database.
db.SubmitChanges();

//Get All Employee from Database
var employeeList = db.Employees;
foreach (Employee employee in employeeList)
{
    Console.WriteLine("Employee Id = {0} , Name = {1}, Email = {2}, ContactNo = {3}",
        employee.EmployeeId, employee.Name, employee.Email, employee.ContactNo);
}

Console.WriteLine("\nPress any key to continue.");
Console.ReadKey();
}
}
}

```

VB

Module Module1

```

Sub Main()
    Dim connectionString As String =
System.Configuration.ConfigurationManager.ConnectionStrings("LinqToSQLDBConnectionString").ToString()

```

```

    Dim db As New LinqToSQLDataContext(connectionString)

```

```

    Dim deleteEmployee As Employee = db.Employees.FirstOrDefault(Function(e) e.Name.Equals("George Michael"))

```

```

    db.Employees.DeleteOnSubmit(deleteEmployee)

```

```

    db.SubmitChanges()

```

```

    Dim employeeList = db.Employees
    For Each employee As Employee In employeeList
        Console.WriteLine("Employee Id = {0} , Name = {1}, Email = {2}, ContactNo = {3}",
employee.EmployeeId, employee.Name, employee.Email, employee.ContactNo)
    Next

```

```

    Console.WriteLine(vbLf & "Press any key to continue.")
    Console.ReadKey()

```

```

End Sub

```

```

End Module

```

当C#或VB上面的代码被编译并运行，它会产生以下结果：

Employee ID = 1, Name = William, Email = abc@gy.co, ContactNo = 999999999

Employee ID = 2, Name = Miley, Email = amp@esds.sds, ContactNo = 999999999

Employee ID = 3, Name = Benjamin, Email = asdsad@asdsa.dsd, ContactNo =

Press any key to continue.

易百教程移动端：请扫描本页面底部(右侧)二维码并关注微信公众号，回复："教程" 选择相关教程阅读或直接访问：<http://m.yibai.com>。

上一篇：[LINQ元素操作符](#) 下一篇：[LINQ对象](#)

加QQ群啦，易百教程官方技术学习群

所有其它免费技术学习QQ群，请点击：<http://www.yiibai.com/siteinfo/qq-group.html>

QQ群名称	群号	人数/免费/等级	群介绍
JAVA技术	227270512	2000人/否/LV5	Java基础，JSP (Servlet) ，JAVA框架，Java高并发架构，Maven等等
MySQL/SQL	418407075	2000人/否/LV5	SQL基础，MySQL基础，MySQL存储过程，视图，触发器等等
大数据开发	655154550	2000人/否/LV5	Spark , zookeeper , kafka , CDH , hive , flume , hbase等Hadoop云计算生态圈技术
Python技术	287904175	2000人/否/LV5	Python编程，Python Web，Python大数据，Python爬虫，自然语言处理等
测试工程师(新群)	415553199	1000人/是/LV1	软件测试，硬件测试，测试平台开发，黑白盒测试，Labview等技术
前端技术群(新群)	410430016	1000人/是/LV1	CSS,HTML/HTML5,JS/JQuery/VueJS/A

			ngularJS 等技术
Linux技术	47942947 7	2000人/ 是/LV5	Redhat/C entos,Ub untu,Shell ,运维,监控 等技术
PHP开发 者	46015324 1	2000人/ 是/LV5	PHP基 础,PHP高 级,网站优 化/架 构,JS,HTM L,JQuery 等Web技 术
人工智能	45623608 2	2000人/ 是/LV5	人工智 能, 深度 学习, 算 法等技术
Oracle数 据库	17524814 6	2000人/ 是/LV5	SQL基 础, Oracle基 础, Oracle存 储过程, 视图, 触 发器等
Android 开发	15962918 5	2000人/ 是/LV1	Android 开发, Android Studio, Kotlin,Da gger等技 术
微软技术	57982170 6	2000人/ 是/LV1	C# , ASP.Net , VB.Net , ADO.Net , SQL Server , VBA , Excel等技 术
数据分析 师	39788399 6	2000人/ 是/LV1	R语言 , Matlab语 言 , Pandas等 技术

所有其它免费技术学习QQ群，请点击：<http://www.yiibai.com/siteinfo/qq-group.html>

0

收藏文章



登录

我的社区

来说两句吧...

表情删除后不可恢复，是否删除

取消

确定

图片正在上传，请稍后...

[取消上传](#)

评论内容为空！

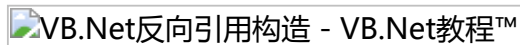
还没有评论，快来抢沙发吧！

该评论已关闭!

[易百教程正在使用畅言](#)

去社区看看吧

去热评看看吧



VB.Net反向引用构造 - VB.Net教程™

- [VB.Net反向引用构造 - VB.Net教程™](#)



Spring JavaConfig实例 - Spring教程™

- [Spring JavaConfig实例 - Spring教程™](#)



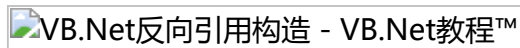
ASP.Net MVC Web API - ASP.Net MVC教程™

- [ASP.Net MVC Web API - ASP.Net MVC教程™](#)



VBA输入框 (InputBox) - VBA教程™

- [VBA输入框 \(InputBox \) - VBA教程™](#)



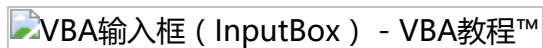
VB.Net反向引用构造 - VB.Net教程™



Spring JavaConfig实例 - Spring教程™



ASP.Net MVC Web API - ASP.Net MVC教程™



VBA输入框 (InputBox) - VBA教程™

热评话题

- [VBA输入框 \(InputBox \) - VBA教程™](#)
- [Python连接MongoDB操作 - MongoDB教程™](#)
- [jstl 标签 - JSP教程™](#)
- [Javascript Array.join\(\)方法 - Javascript教程™](#)
- [Git基础概念 - Git教程™](#)
- [R语言泊松回归 - R语言教程™](#)
- [ASP.Net MVC NuGet包管理 - ASP.Net MVC教程™](#)

推荐/最新教程

- [VueJS教程](#)
- [ASP.Net MVC教程](#)
- [.NET Core教程](#)
- [ASP.Net教程](#)
- [MariaDB教程](#)
- [ADO.Net教程](#)
- [VB.Net教程](#)
- [Pandas教程](#)

最新更新

- [Access创建关系](#)
- [Access关联数据](#)
- [Access备用条件](#)
- [Access参数查询](#)
- [Access创建查询](#)
- [Access操作查询](#)
- [Access查询标准\(条件\)](#)
- [Access查询数据](#)

站点信息

- [意见反馈](#)
- [免责声明](#)
- [关于我们](#)
- [关于捐助](#)
- [所有实例代码下载](#)

易百教程官方QQ群

JAVA技术：227270512

MySQL数据库：418407075

Python技术：287904175

大数据开发：655154550

Linux技术：479429477

PHP/Web技术：460153241

关注微信公众号



Copyright © 2012-2017 [易百教程](#) [yii2.com](#) All Rights Reserved. 备案号：琼ICP备13001417号-3 联系我们：769728683@qq.com [站长统计](#)