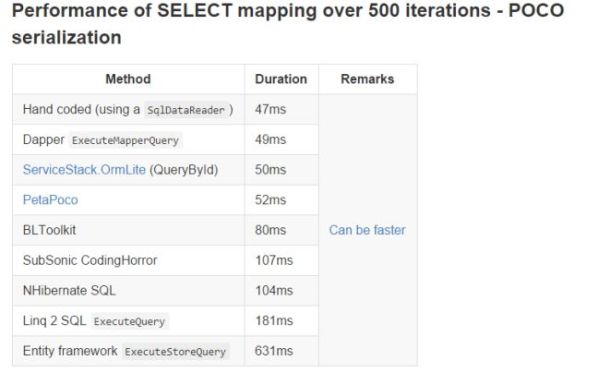
[第一篇：Dapper快速学习](http://www.cnblogs.com/huangxincheng/p/5828470.html)

一：为什么选择Dapper

1. 性能优越：

    其实在各大网站上，我们大概都会看到这样的一个对比效果图，在超过500次poco serialization的过程中所表现的性能，我们发现dapper是第二名，

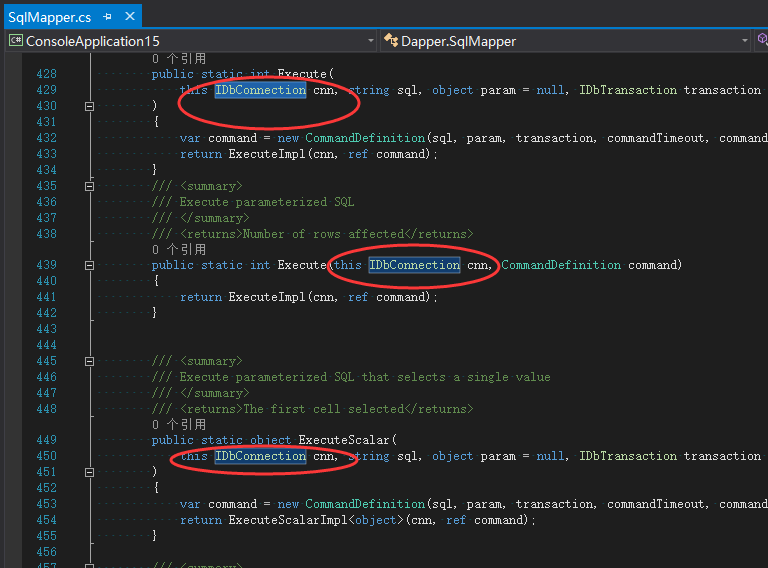
当然第一名谁也无法超越，越底层的当然久越快，同时也就越麻烦。就好像谁能超过“01代码”呢？？？



2. 支持多数据库

      支持多数据库的本质是因为Dapper是对IDBConnection接口进行了方法扩展，比如你看到的SqlMapper.cs，一旦你这样做了，我们也知道，

SqlConnection,MysqlConnection,OracleConnection都是继承于DBConnection，而DBConnection又是实现了IDBConnection的接口，对吧。。。

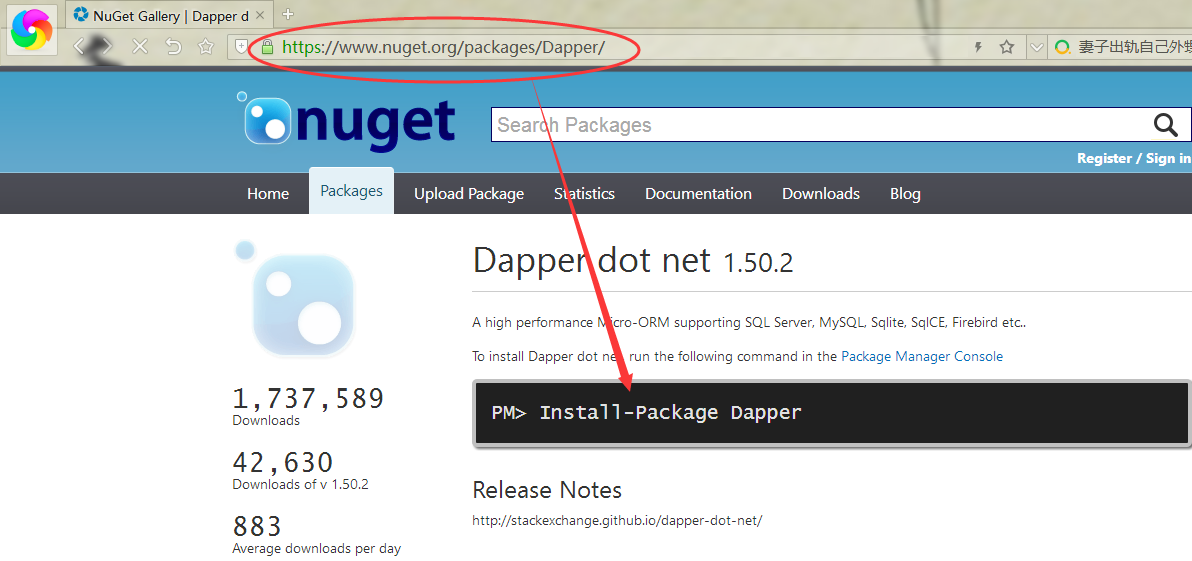


二：安装Dapper

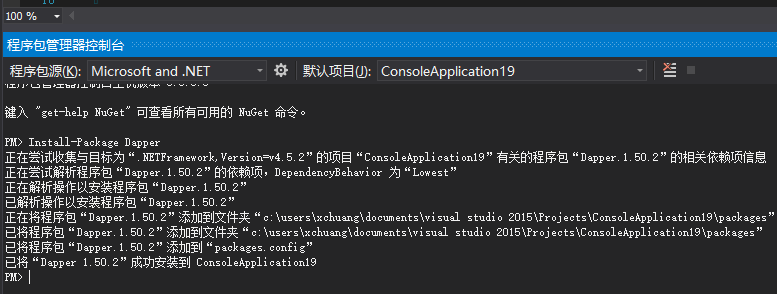
    install dapper的方式通常有两种：

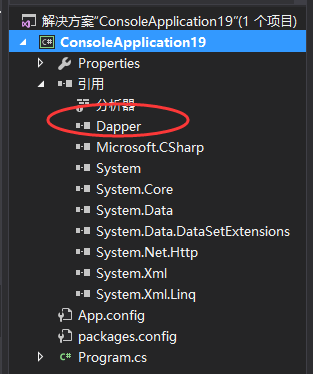
1. 通过nuget进行安装

　  如果你不知道怎么用nuget进行安装，或者不知道install-package是什么，可以在browser上找一下，比如下面这样：



然后我们copy到package console 试试看。

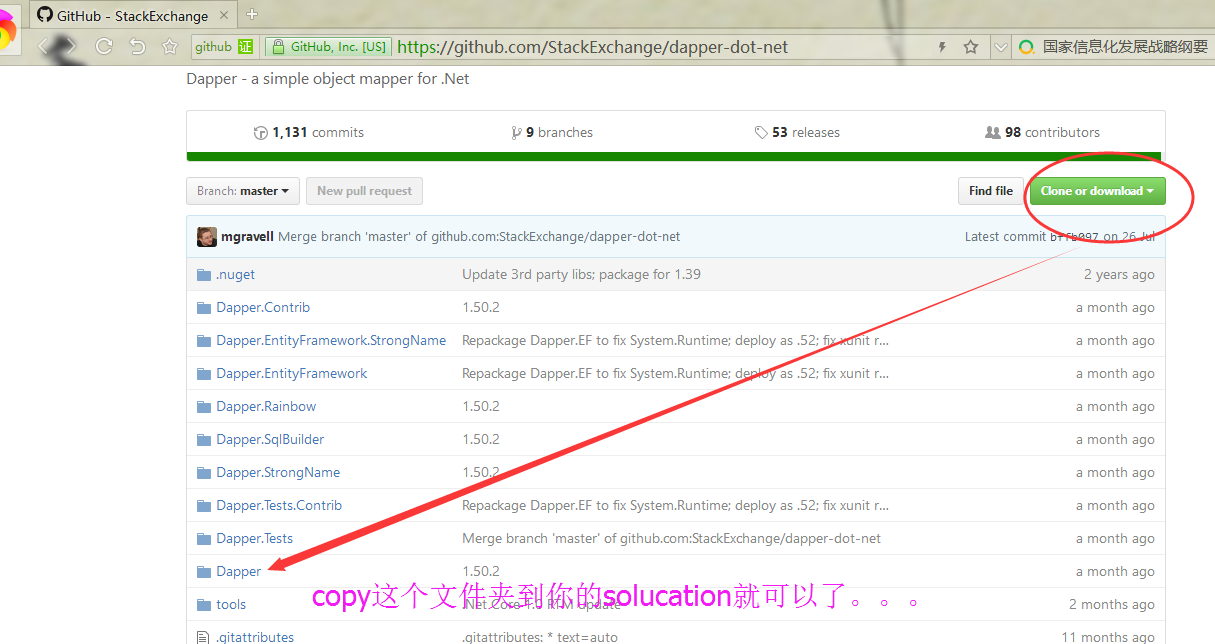


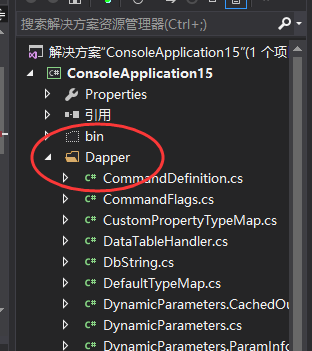


2. 在github上获取源码。

   为什么要获取源码，是因为用ilspy调试dapper的源码太费劲了，毕竟现在都是异步编程了，从ilspy中看都是匿名方法很多都无法渗透，废话不多

说，我们只要把Dapper文件夹拉出来然后copy到我们的solution就可以了，如下图：





三：快速CURD操作

   其实对数据库的操作莫过于CURD,在进行操作之前我们再配一个Users表。

1. 配置Users表

[复制代码](javascript:void(0);)

CREATE TABLE [dbo].[Users](

[UserID] [int] IDENTITY(**1**,**1**) NOT NULL,

[UserName] [varchar](**50**) NULL,

[Email] [varchar](**100**) NULL,

[Address] [varchar](**100**) NULL,

CONSTRAINT [PK\_Users] PRIMARY KEY CLUSTERED

(

[UserID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

[复制代码](javascript:void(0);)

2. Insert操作

  通常来说，有两种insert操作：

<1>单条insert操作

  这是一个简单的参数化insert，而且还可以塞入匿名类型，对吧，跟原始的SqlParameter相比，是不是简单的多？？？

[复制代码](javascript:void(0);)

static void Main(string[] args)

{

IDbConnection connection = new SqlConnection("Data Source=.;Initial Catalog=DataMip;Integrated Security=True;MultipleActiveResultSets=True");

var result = connection.Execute("Insert into Users values (@UserName, @Email, @Address)",

new { UserName = "jack", Email = "380234234@qq.com", Address = "上海" });

}

[复制代码](javascript:void(0);)

<2> InsertBulk操作

     既然是Bulk操作，那肯定就是批量插入了，我们要做的就是将上面这个 ”匿名对象" 变成 ”匿名对象集合“ 就可以了。。。为了方便操作，这里定义

一个Users类，比如下面这样。。。

[复制代码](javascript:void(0);)

static void Main(string[] args)

{

IDbConnection connection = new SqlConnection("Data Source=.;Initial Catalog=DataMip;Integrated Security=True;MultipleActiveResultSets=True");

//var result = connection.Execute("Insert into Users values (@UserName, @Email, @Address)",

// new { UserName = "jack", Email = "380234234@qq.com", Address = "上海" });

var usersList = Enumerable.Range(0, 10).Select(i => new Users()

{

Email = i + "qq.com",

Address = "安徽",

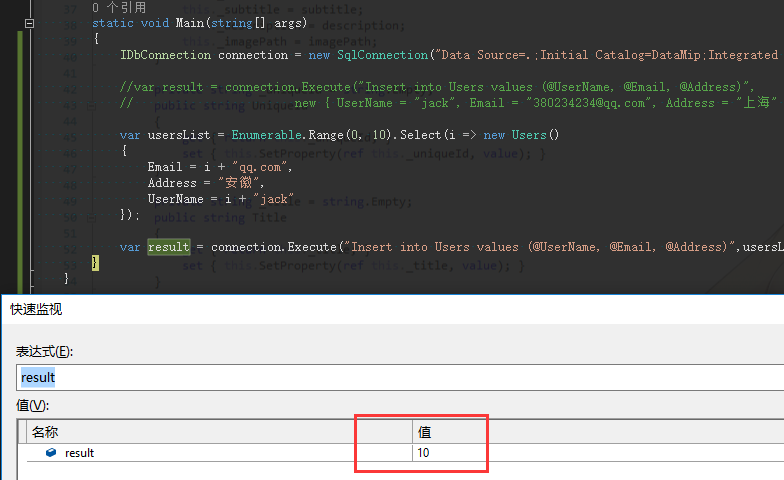
UserName = i + "jack"

});

var result = connection.Execute("Insert into Users values (@UserName, @Email, @Address)",usersList);

}

[复制代码](javascript:void(0);)



2. Query操作

   其实在Dapper在query上提供的的文章太多了。。。这篇我们就按照最简单的参数化查询就好了。。。比如我要找到username=jack的记录，如下：

[复制代码](javascript:void(0);)

1 static void Main(string[] args)

2 {

3 IDbConnection connection = new SqlConnection("Data Source=.;Initial Catalog=DataMip;Integrated Security=True;MultipleActiveResultSets=True");

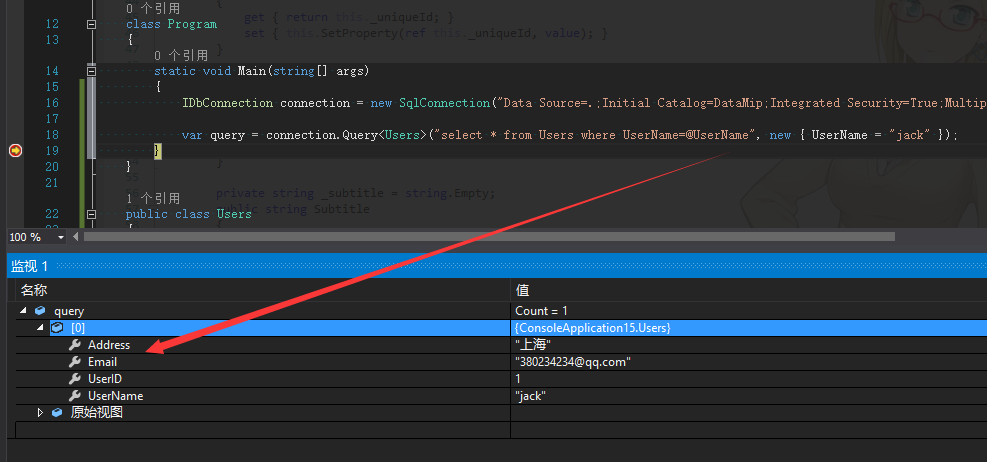
4

5 var query = connection.Query<Users>("select \* from Users where UserName=@UserName", new { UserName = "jack" });

6

7 }

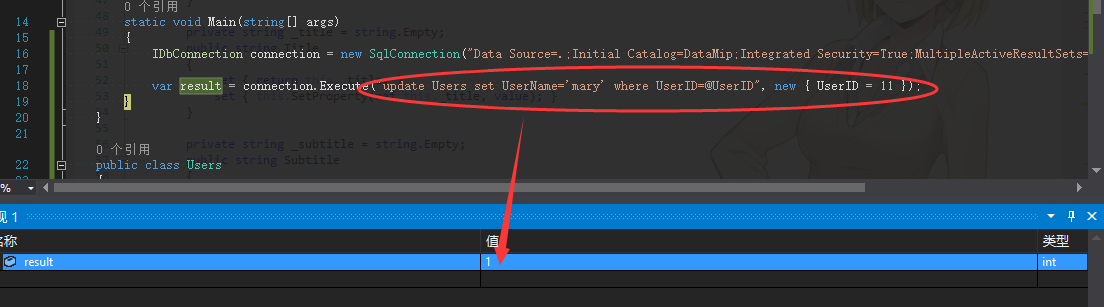
[复制代码](javascript:void(0);)



图上的亮点就在于能够自动化mapper到我们object上面来，这是我们DataReader所不能办到的，对吧~~

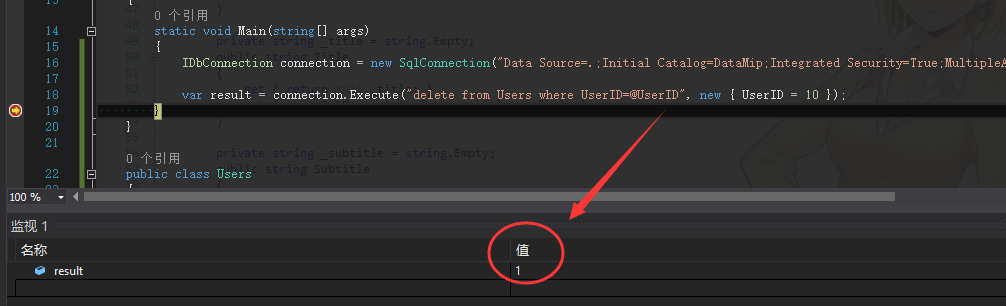
3.update操作

  这种操作方式，我们还是使用Execute方法来实现，和insert是一种套路的哦。



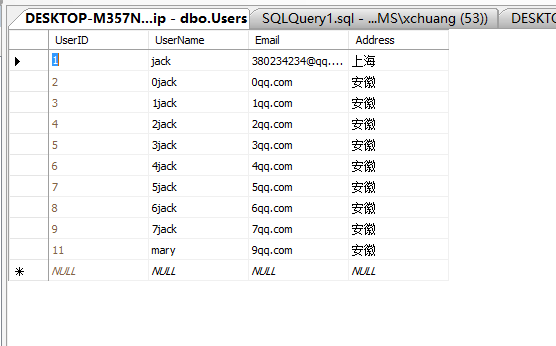
4. delete操作

    这里我还是采用参数化的形式来删除UserID=10这条记录，方式如下：



最终sql的table展示如下，可以看到已经正确的修改了UserID=11的记录，删除了UserID=10的record。。。。当然Dapper好玩的地方多着呢，

这篇只是一个入门而已。。。希望本篇对大家有帮助~~~



[第二篇：Dapper中的一些复杂操作和inner join应该注意的坑](http://www.cnblogs.com/huangxincheng/p/5832281.html)

一：table sql

    为了方便，这里我们生成两个表，一个Users，一个Product，sql如下：

<1> Users table

[复制代码](javascript:void(0);)

CREATE TABLE [dbo].[Users](

[UserID] [int] IDENTITY(**1**,**1**) NOT NULL,

[UserName] [varchar](**50**) NULL,

[Email] [varchar](**100**) NULL,

[Address] [varchar](**100**) NULL,

CONSTRAINT [PK\_Users] PRIMARY KEY CLUSTERED

(

[UserID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

[复制代码](javascript:void(0);)

<2> Product table

[复制代码](javascript:void(0);)

CREATE TABLE [dbo].[Product](

[ProductID] [int] IDENTITY(**1**,**1**) NOT NULL,

[ProductName] [varchar](**220**) NULL,

[ProductDesc] [varchar](**220**) NULL,

[UserID] [int] NULL,

[CreateTime] [datetime] NULL,

CONSTRAINT [PK\_Product] PRIMARY KEY CLUSTERED

(

[ProductID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

[复制代码](javascript:void(0);)

二：in操作

      很多时候我们在manipulate table的时候，或多或少的都会用到 ”in关键字”，比如:我要找到User表中Email in ('5qq.com','8qq.com')的

Users record。。。

[复制代码](javascript:void(0);)

static void Main(string[] args)

{

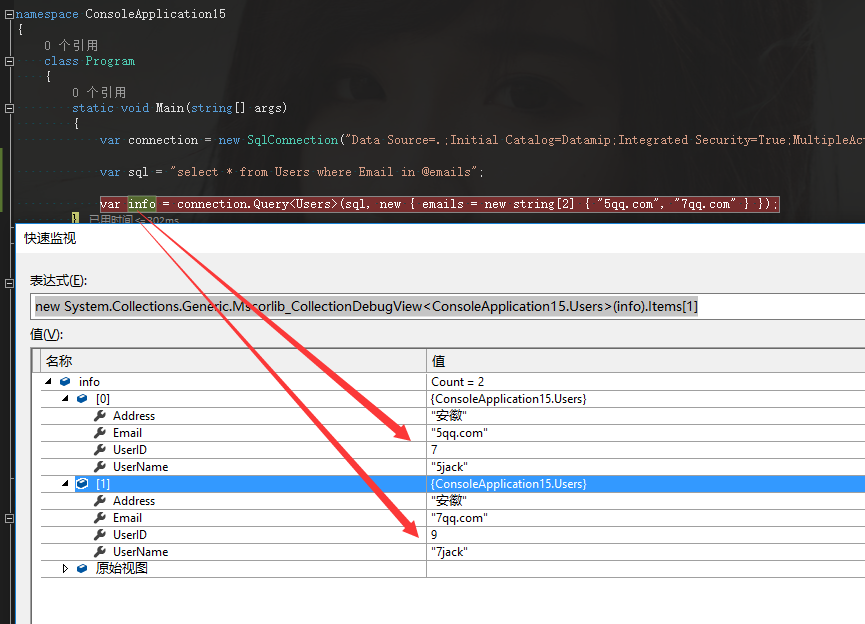
var connection = new SqlConnection("Data Source=.;Initial Catalog=Datamip;Integrated Security=True;MultipleActiveResultSets=True");

var sql = "select \* from Users where Email in @emails";

var info = connection.Query<Users>(sql, new { emails = new string[2] { "5qq.com", "7qq.com" } });

}

[复制代码](javascript:void(0);)



看了上面的操作，是不是很简单，只要我们的参数类型是Array的时候，dappper会自动将其转化。。。

三：多条sql一起执行

     有时候我们会想在一条sql中灌入很多的snippet sql，然后让其一起执行，此时让我想起了一个操作，我会在db中load data的时候会写到

select ... from marketing where  id in (....); select .... from eventmarketing where in (...)类似这样的语句，然后进行结果合并，这篇

为了方便演示，在User上做一个\*操作，在Product上做一个\* 操作，比如下面这样：

[复制代码](javascript:void(0);)

static void Main(string[] args)

{

var connection = new SqlConnection("Data Source=.;Initial Catalog=Datamip;Integrated Security=True;MultipleActiveResultSets=True");

var sql = "select \* from Product; select \* from Users";

var multiReader = connection.QueryMultiple(sql);

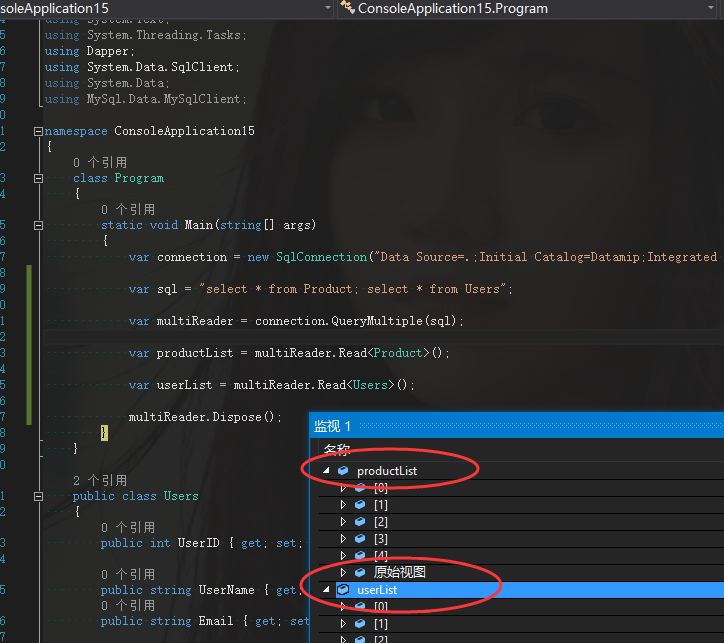
var productList = multiReader.Read<Product>();

var userList = multiReader.Read<Users>();

multiReader.Dispose();

}

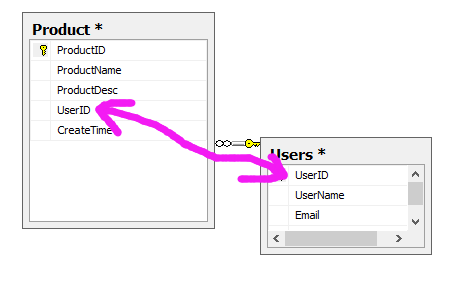
[复制代码](javascript:void(0);)



四：多表join操作

     不管sql写的多么好或者多么烂，接触一个月还是接触到十年，都必然跑不了多表查询，那么在多表查询上dapper该如何使用呢？？？比如

说我要找到2015-12-12之后的商品信息和个人信息，很显然这是一个多表查询，可以先来看一下users和product的关系。



可以发现其实他们有一个外键关系，然后我们在Product Entity上做一下小修改，将Users作为Product的一个entity property。。。

[复制代码](javascript:void(0);)

public class Product

{

public int ProductID { get; set; }

public string ProductName { get; set; }

public string ProductDesc { get; set; }

public Users UserOwner { get; set; }

public string CreateTime { get; set; }

}

[复制代码](javascript:void(0);)

有了这些储备，我们大概就可以写出如下的sql。

[复制代码](javascript:void(0);)

static void Main(string[] args)

{

var connection = new SqlConnection("Data Source=.;Initial Catalog=Datamip;Integrated Security=True;MultipleActiveResultSets=True");

var sql = @"select p.ProductName,p.CreateTime,u.UserName

from Product as p

join Users as u

on p.UserID = u.UserID

where p.CreateTime > '2015-12-12'; ";

var result = connection.Query<Product, Users, Product>(sql,

(product, users) =>

{

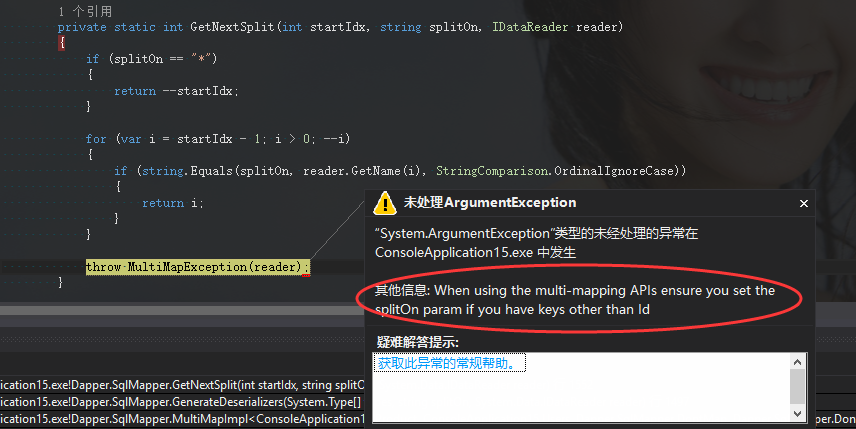
product.UserOwner = users; return product;

});

}

[复制代码](javascript:void(0);)

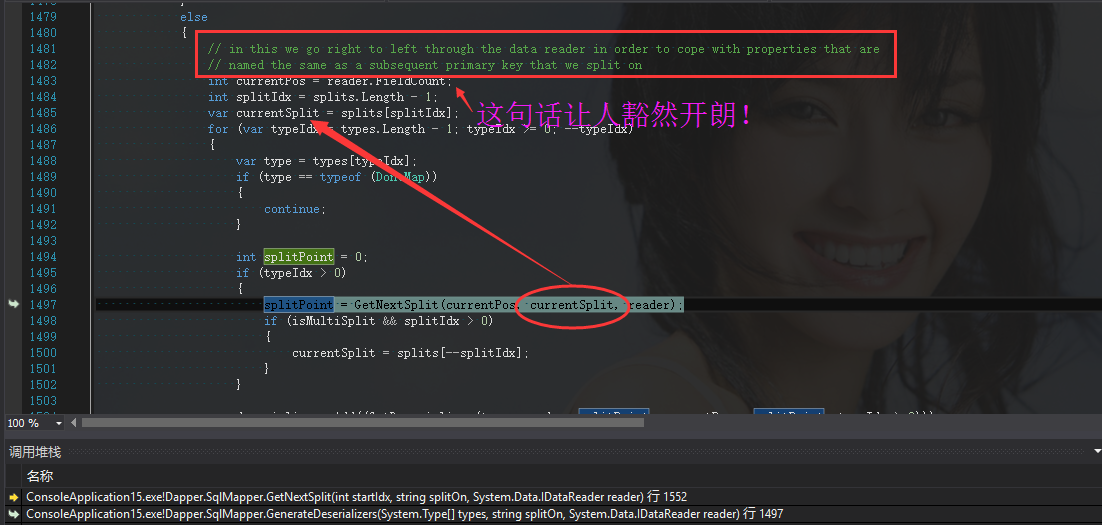
结果就是”操！！！！“。。。。。。。。。。。。



从错误信息中可以看到：当你使用multi-mapping的时候要确保设置了splitOn参数，除了Id。。。从这句话中好像也看不出什么名堂，也就是说

除了Id，你都需要设置SplitOn参数，好吧，这是逼着哥哥看源代码。。。。看看SplitOn到底是个什么样的鸟玩法。。。然后我从Call Stack往上

面找，发现了非常”至关重要“的一段话。



    然来splitOn就是Dapper对DataReader进行”从右到左“的扫描，这样就可以从sequent中获取到一个subsequent，然后遇到设置的splitOn

就停止。。。然来是这样，哈哈。。。这回我就知道了，将splitOn设置为”userName“就好了。。。比如下面这样。。。

[复制代码](javascript:void(0);)

static void Main(string[] args)

{

var connection = new SqlConnection("Data Source=.;Initial Catalog=Datamip;Integrated Security=True;MultipleActiveResultSets=True");

var sql = @"select p.ProductName,p.CreateTime,u.UserName

from Product as p

join Users as u

on p.UserID = u.UserID

where p.CreateTime > '2015-12-12'; ";

var result = connection.Query<Product, Users, Product>(sql,

(product, users) =>

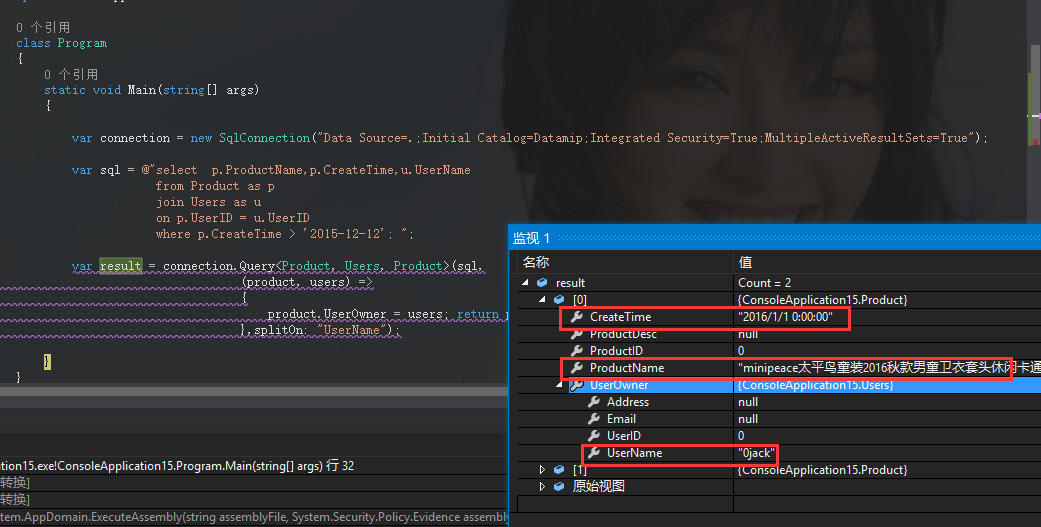
{

product.UserOwner = users; return product;

},splitOn: "UserName");

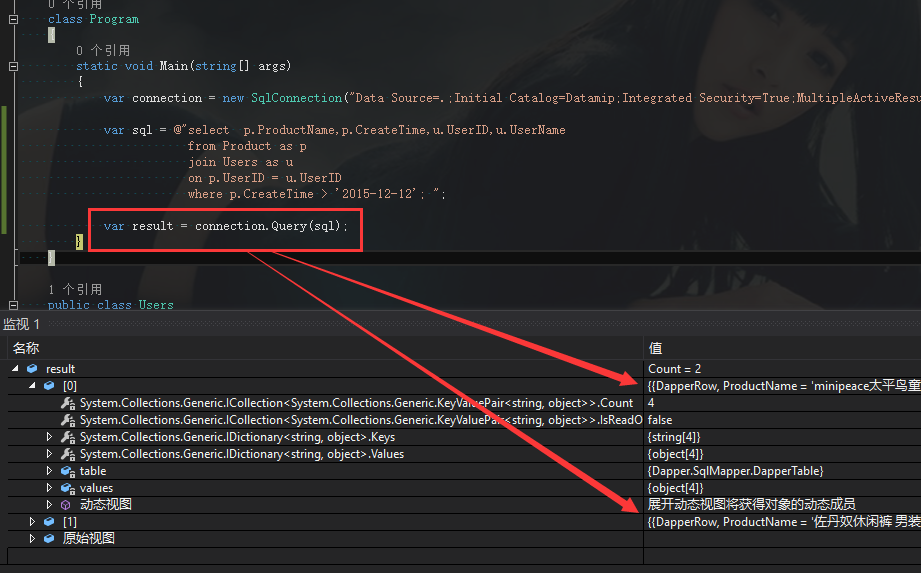
}

[复制代码](javascript:void(0);)



当然如果你觉得我上面说的太啰嗦了，注意事项还tmd的多，又是泛型，又是Lambda的。。。你也可以不指定这些具体Type，而默认使用

dynamic也是可以的，比如下面这样：



五：支持存储过程

   对于存储过程，也是一个不得不说的话题，我们的dapper同样也是可以执行的，只需要在Query中的CommandType中标记一下当前就是一个

StoredProcedure就八九不离十了，比如现在在Users表上创建一个简单的StoredProcedure。

[复制代码](javascript:void(0);)

USE [Datamip]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[sp\_GetUsers] Script Date: 09/02/2016 09:14:04 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

Create proc [dbo].[sp\_GetUsers]

@id int

as

begin

select \* from Users where UserID = @id ;

end

[复制代码](javascript:void(0);)

在这里，我们需要向存储过程塞入一个@id参数，返回具体的Users EntityList，好了，下面再看一下Query如何构造。

[复制代码](javascript:void(0);)

static void Main(string[] args)

{

var connection = new SqlConnection("Data Source=.;Initial Catalog=Datamip;Integrated Security=True;MultipleActiveResultSets=True");

var info = connection.Query<Users>("sp\_GetUsers", new { id = 5 },

commandType: CommandType.StoredProcedure);

}

[复制代码](javascript:void(0);)

