RDIFramework.NET

平台代码生成器

使用教程

国 思 软 件

目录

1. 前言

RDIFramework.NET 代码生成器集代码生成、各数据库对象文档生成、数据库常用管理于一身，是软件开发者快速开发的神器。代码的生成支持基于数据库与 PowerDesign 设计文件生成，为企业及个人开发者能快速生成常用代码与文档以达到快速开发，快速应用的目的，节省开发成本。

基于数据库的代码生成不仅可直接在生成器上做数据库 Sql 级的操作，如常用的表、视图、存储过程、函数的创建、查询、修改、删除、生成数据脚本、生成数据库设计文档、表设计文档、生成业务逻辑代码、生成 Web、WinForm 界面代码等。便捷的数据库管理功能，轻松操作数据库。完全不用打开数据库企业管理工具即可完成相应的数据库层面的操作，支持 SqlServer、Oracle、MySql 版本，其他类型的数据库也在陆续支持中。

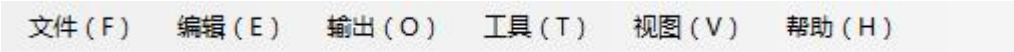
基于 PowerDesigner 设计文件的代码生成，是我们代码生成器独居的特色，开发人员不用关心是什么类型的数据库，即可完美的生成项目代码。建议在开发过程中使用 PowerDesigner 设计工具完成数据库的建模，PD 是非常优秀的数据库建模工具，熟练的使用该工具对系统的分析和设计有很大的帮助。可以用 PowerDesigner 做需求分析，各种模型之间的转换，设计完成还可自动把数据库生成出来。在设计阶段我们只需把精力集中在模型上。

系统主界面


RDIFramework.NET 平台代码生成器
文件 (F)　编辑 (E)　输出 (O)　工具 (T)　视图 (V)　帮助 (H)
RDIFramework61　BASE_MODULE
数据库视图
起始页　DEMO_PRODUCT
RDIFramework.NET 代码生成器
企业级.NET快速开发平台(十年专注、精益求精)
代码、文档一键生成...
常用操作
新增数据库服务器
对象资源管理器
数据库脚本生成器
数据库文档生成器
生成代码项目
代码生成器
代码批量生成器
项目属性设置
最新信息
国思RDIF-Wms仓储管理系统助力企业高效数字化（源码交付）
国思RDIF-Wms是一款基于国思RDIF低代码快速开发框架下的仓库管理系统，不仅涵盖了仓库/库区/货架管理，出入库管理，客户/供应商/承运商，库存看板、库存记录、库存预警与到期提醒、基础数据报表展示等功能，更整合了增值业务、多租户支持、用户权限管理以及各基础数据的维护。从而进一步提升业务执行效率，实现仓储的智能化管理 。同时还全源码提供，方便企业扩展，加速落地应用，方便满足不同特性仓库的管理需求。
PowerDesigner Objects
BASE_AREA
BASE_ATTACHMENT
BASE_CODE_RULE
BASE_CODE_RULE_SEED
BASE_COMPANY
BASE_DATASOURCE
BASE_DBLINK_DEFINE
BASE_EXCEPTION
BASE_FILE
BASE_FOLDER
BASE_FORM_MODULE_RELATION
BASE_FORM_TEMPLATE
BASE_FORM_TEMPLATE_INFO
BASE_ITEM_DETAILS
BASE_ITEMS
BASE_LOG
BASE_MESSAGE
BASE_MODULE
生成SQL语句到
浏览表数据
生成数据脚本
生成存储过程
导出文件
生成业务代码(Framework)
生成业务代码(Core)
生成敏捷MvcUI界面
生成敏捷CoreMvcUI界面
生成WinForm界面
生成业务代码(Framework)_兼容V6.0以下
生成业务代码(Core)_兼容V6.0以下
生成MvcEasyUI界面
SELECT(S)
UPDATE(U)
DELETE(D)
INSERT(I)
DROP(D)
BASE_STAFF_ORGANIZE
BASE_TABLE_COLUMNS
BASE_TABLE_PERMISSION_SCOPE
BASE_USER
BASE_USER_ITEM_DETAILS
BASE_USER_LOGON
BASE_USER_ORGANIZE
BASE_USER_ROLE
2024-06-28 10:02:27
2024-06-26 09:51:10
2024-06-25 10:25:41
2024-06-20 10:05:15
2024-06-17 10:25:38
2024-06-13 10:05:29
2024-04-13 10:51:02
2024-01-08 10:55:02
2023-08-25 14:11:43
2023-07-27 09:16:53
2023-07-25 11:03:16
2023-07-13 11:37:29
2023-06-26 11:35:40
2023-06-20 08:56:46
2023-05-09 14:25:22
2023-04-27 11:34:01
2023-04-25 09:29:38
2023-04-20 09:00:00
2023-04-19 08:30:00
2023-04-19 08:00:00
系统消息　就绪
http://www.rdiframework.net/　当前版本：6.1.9070.26340

在 RDIFramework.NET 代码生成器主界面中，共包括 6 个不同的工作区域：系统菜单、工具栏、数据库视图导航区、PowerDesigner 设计文档对象导航区，业务工作区、系统状态栏。

1.1. 系统菜单

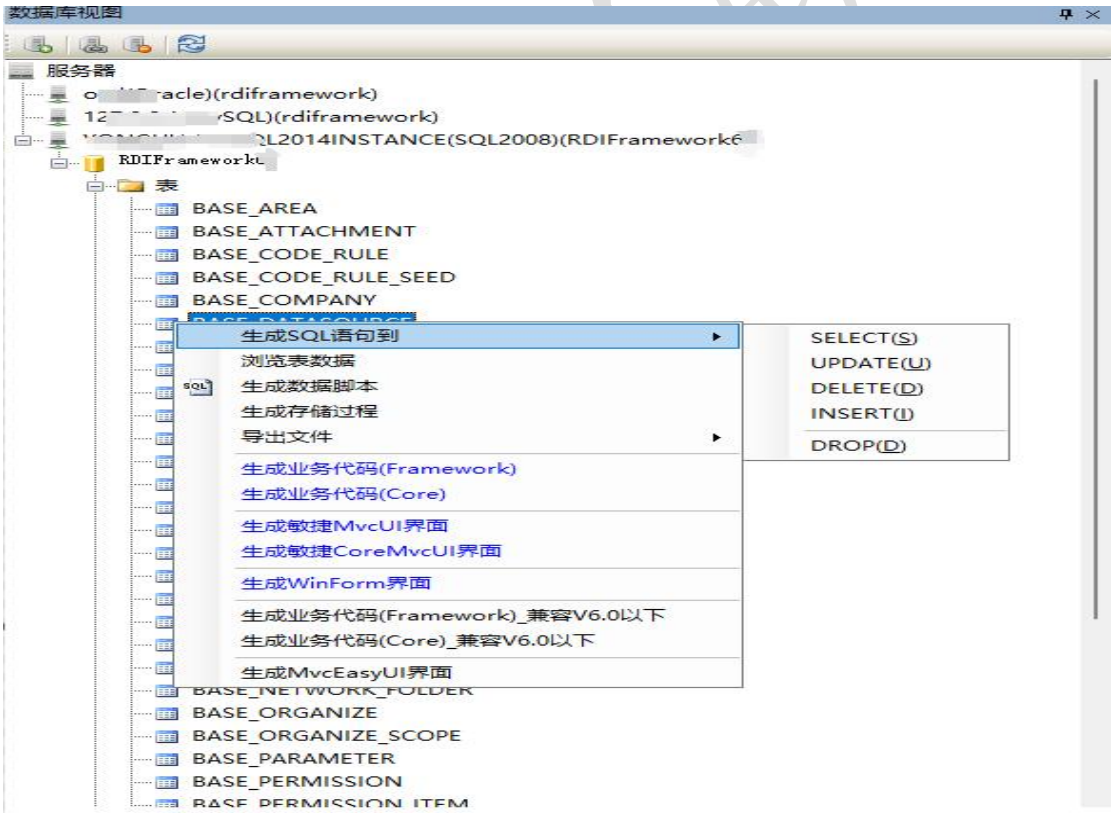


RDIFramework.NET 代码生成器系统菜单由文件、编辑、输出、工具、视图、帮助等菜单组成。选择相应的功能项，系统菜单会有相应的变化。

1.2. 工具栏

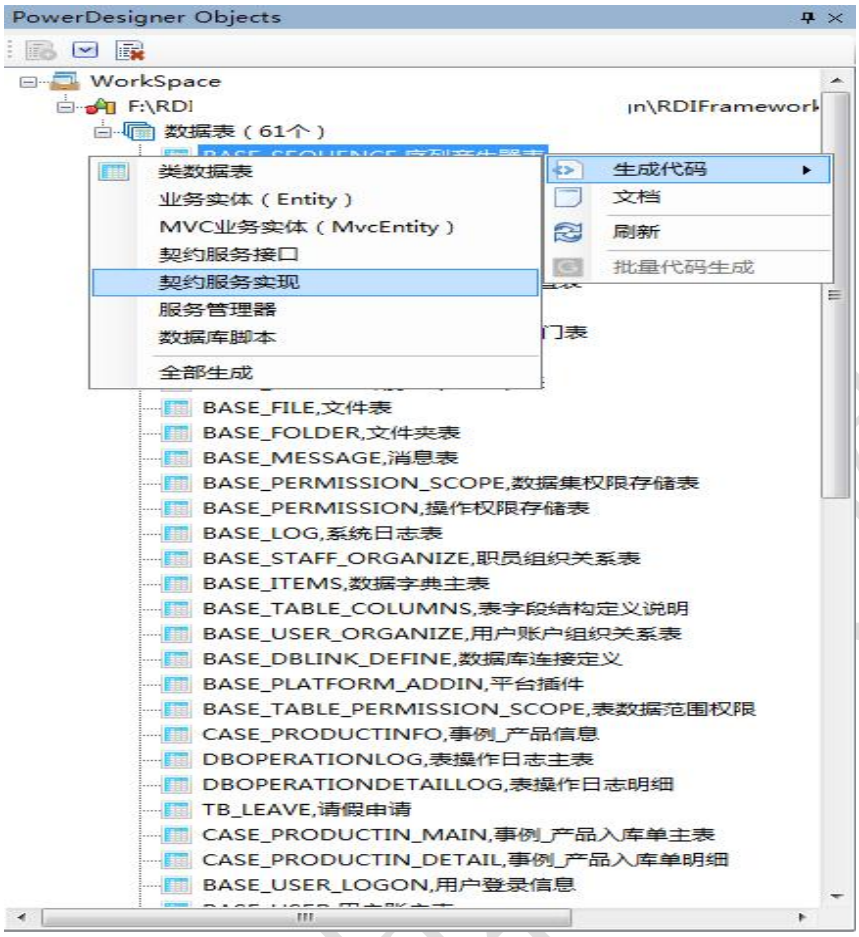
工具栏放置了与系统菜单相当的常用功能模块的快捷操作方式。

1.3. 数据库视图导航区



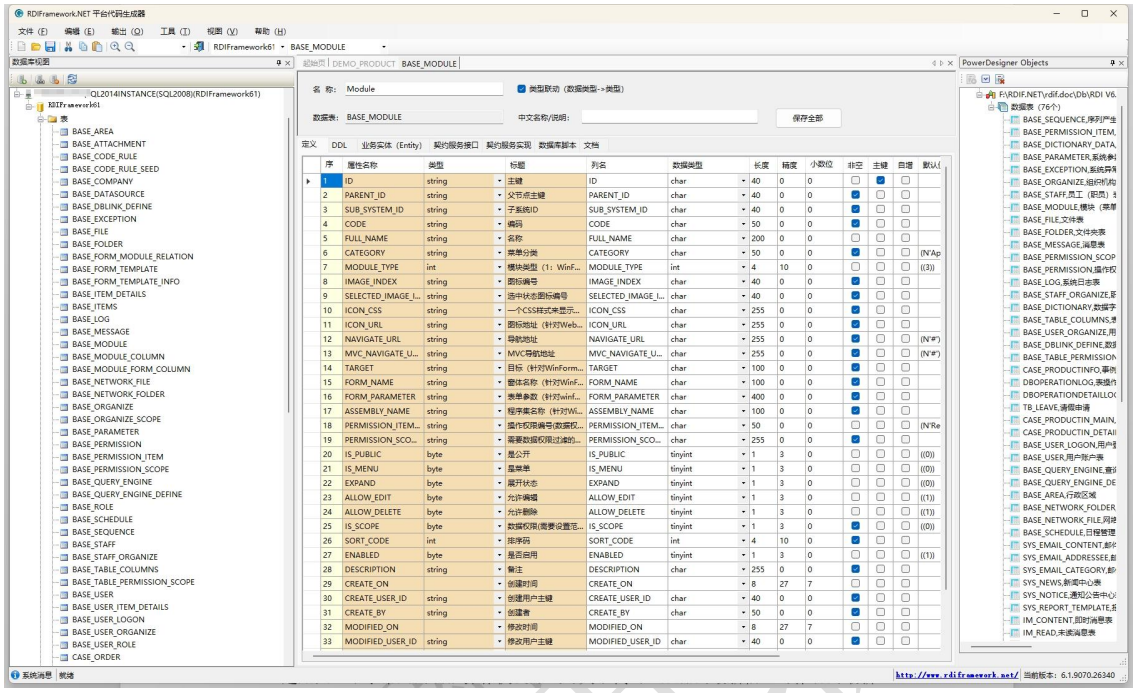
数据库视图导航区是整个代码生成器主要的工作导航区域，在这儿，我们可以通过右键快捷菜单来进行常用的操作，如：查询数据、修改数据、删除数据、删除表、浏览表的数据、生成表的数据脚本（表数据导出）、生成表的存储过程（主要用于业务代码的数据访问层采用存储过程的方式）、导出数据库对象为常用文件、以及非常强大的基于框架的代码生成主要操作界面（生成业务代码，支持 core 与 framework）、针对项目生成界面 UI 代码。

1.4. PowerDesigner 设计文档对象导航区



在 PowerDesigner 设计文档对象导航区，我们可以导入多个数据库物理设计模型，以通过数据库的设计模型来完成代码及相应文档的生成，通过 PowerDesigner 设计文档，生成的代码与目标数据库的类型无关。

1.5. 业务工作区



业务工作区是整个代码生成器的核心工作区域，该区域会根据用户相应的操作来加载不同的功能模块或窗体界面以完成用户对应的操作。

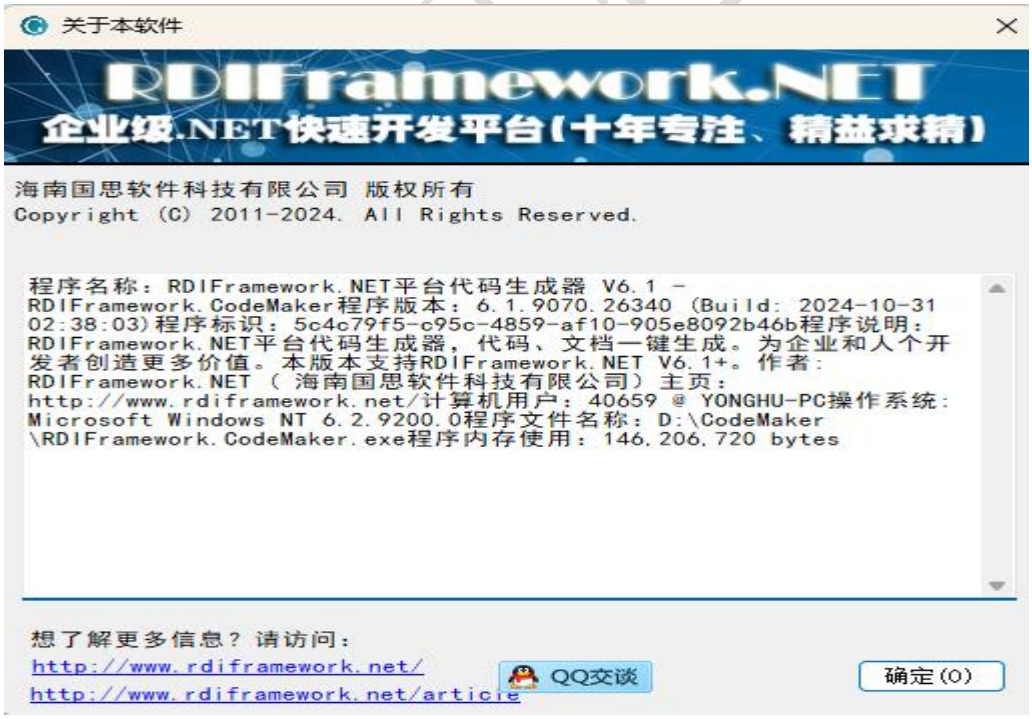
1.6. 系统状态栏

系统状态栏上显示了系统当前的处理任务及任务的处理状态，用于通用用户，以便及时了解系统的运行情况。

1.7. 起始页

起始页显示了常用的功能操作快捷方式与关于代码生成器的最新信息与作者的最新信息，通过这儿，可以了解到关于框架或代码生成器的最新情况，如下图所示。

1.8. 关于



提供对当前程序相关的描述信息，如程序的名称、版本、说明、作者、操作系统的相关系统等。同时，你还可以通过下面的网址了解最新的信息，或直接点击“QQ 交流”按钮与我们 QQ 进行沟通。

2. 数据库视图导航区

数据库视图导航区主要完成对数据库（包括 MSSQLSERVER、ORACLE、ACCESS、MYSQL 等）及数据库对象的管理，包括：

1) 、数据库服务器管理。
2) 、数据库管理。
3) 、数据库对象的管理。

2.1. 数据库服务器管理

数据库服务器管理主要是对建立、移除、备份、导入数据库服务的连接，如下图所示。

选择“服务器”根节点，点击鼠标右键可以添加服务器、备份服务器配置、导入已存在的服务器配置以及刷新。

2.1.1. 添加服务器

在上图中，我们选择“添加服务器”，弹出“数据库类型选择”窗口，如下图所示，可以选择“SQLSERVER”、“ORACLE”、“MYSQL”、“OLEDB”、“SQLLITE”几种数据库类型，代码生成器对数据库当前仅支持 SQLSERVER、ORACLE，在后期的版本中会陆续对其他数据库提供支持，对于其他类型数据库的代码生成，可以通过 PowerDesigner 设计源文件进行生成。

一、连接到 SQLServer。

点击下一步，打开“连接到服务器”对话框，如下图所示。

在上图中，输入相关连接信息，即可连接到 SQLSERVER 数据库。

其中：

服务器名称：SQLSERVER 实例所在的服务名称（可为名称或 IP 地址）。

服务器类型：SQLSERVER 实例的数据库类型（SQLSERVER2000、SQLSERVER2005、SQLSERVER2008、SQLSERVER2012、SQLSERVER2014 等）。

身份验证：连接到 SQLSERVER 的身份认证方式，分为“SQL SERVER 身份认证”和“WINDOWS 身份认证”两种方式。

登录名：登录到 SQLSERVER 的用户名称。

密码：登录到 SQLSERVER 的用户密码。

全部输入后，单击“连接/测试”，可以测试是否能正确连接到 SQLSERVER 数据库，如果测试成功，即可在数据库列表中列出当前 SQLSERVER 实例的所有数据库列表。如果选择“全部数据库”，则会加载所有数据库到“数据库视图”，如果当前实例，数据库过多，我们可以选择一个特定的需操作的数据库加载即可，这样可以提高加载的速度。

二、连接到 ORACLE。

点击下一步，打开“登录”到 ORACLE 对话框，如下图所示。

输入正确的连接到 ORACLE 的相关信息后，即可成功连接到 ORACLE。

三、连接到 MySql。

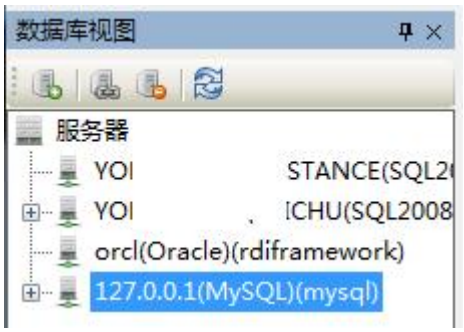
点击下一步，打开“登录”到 MySql 对话框，如下图所示。

输入正确的连接到 MySql 的相关信息后，即可成功连接到 MySql。

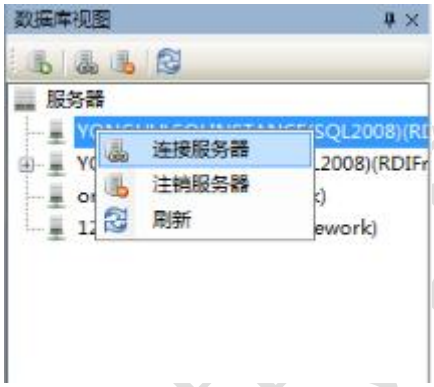
2.1.2. 连接服务器

新增服务器注册成功后，我们就可以连接到新增的服务器上，有两种方式连接到成功注册的服务器上。

方法一：通过数据库视图的工具栏中的“连接到服务器”功能按钮进行连接。



方法二：通过右键功能菜单。



通过上面的两种方式可以建立与注册服务器的连接。

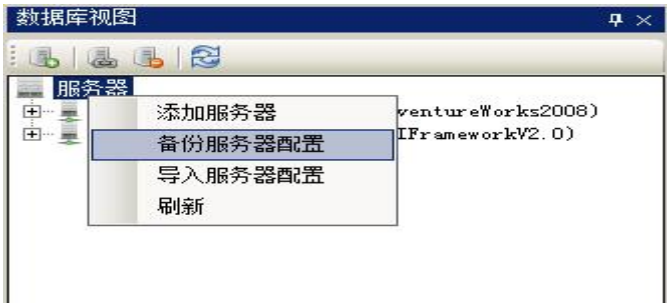
2.1.3. 断开与注销服务器连接

如果已注册的服务器我们不再需要，可以对其进行注销，即删除其注册连接。对于已经成功连接的服务器，我们也可以断开其连接，断开服务器连接如下图所示。

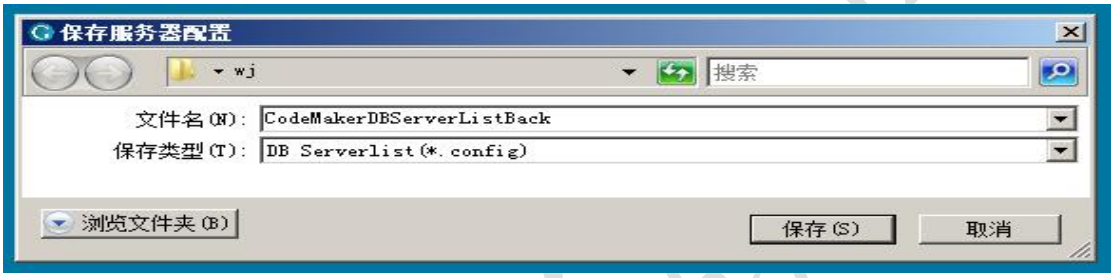
注销服务器连接，如下图所示。

2.1.4. 备份服务器配置

我们可以把当前所有的服务器配置备份到我的电脑，以作他用，要“备份服务器配置”，需要用鼠标右击数据库树视图的“服务器”根节点，如下图所示。



选择“备份服务器配置”，弹出“保存服务器配置”对话框，输入保存的文件名称与保存的目录后，即可对当前所有的服务器连接进行备份。



2.1.5. 导入服务器配置

“导入服务器配置”是“备份服务配置”的逆操作，操作方式与“备份服务器配置”相当，如下图所示。

2.2. 数据库管理

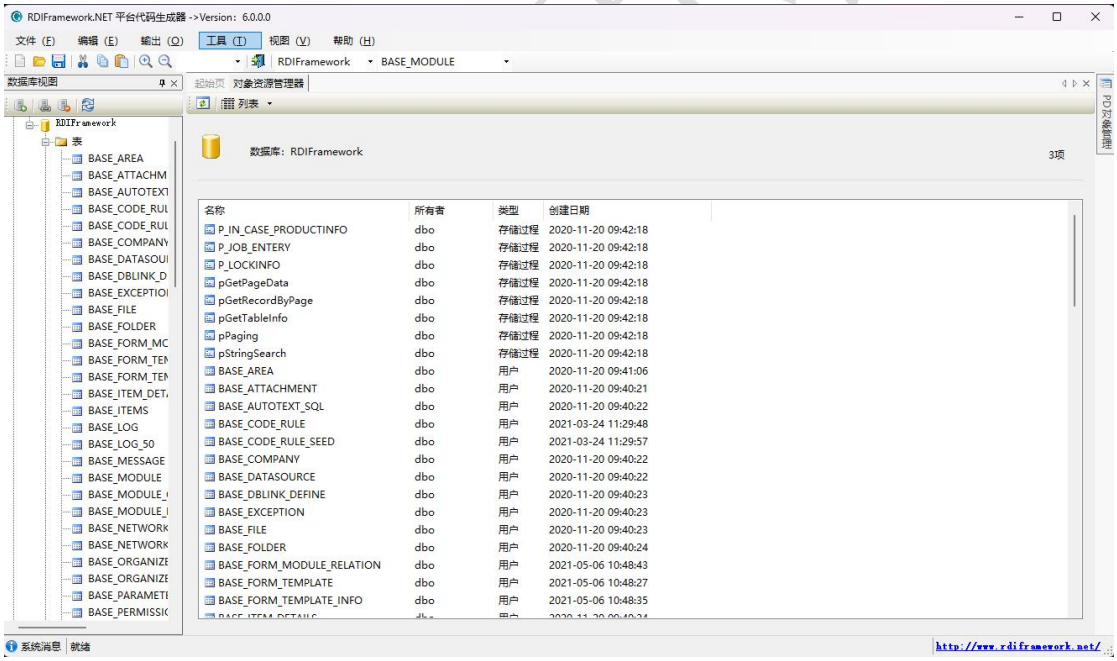
连接到一个已注册的服务器，我们可以对当前服务器所选“数据库”进行相应的操作，主要包括：浏览数据库、新建查询、生成存储过程、生成数据脚本、导入文件（存储过程与数据脚本）等。

2.2.1. 浏览数据库

浏览数据库主要包括浏览当前所选数据库的所有对象（表、视图、存储过程等），当前所选表、视图、存储过程的详细信息。

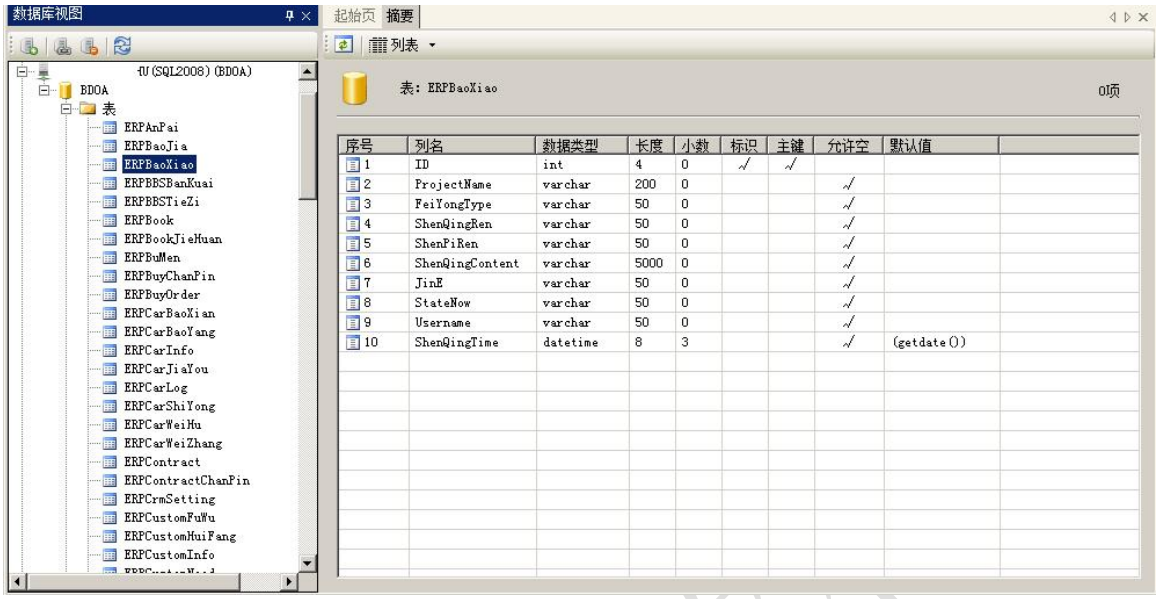
◆ 浏览数据库的详细信息

用鼠标选择数据库树节点，如选择“RDIFRAMEWOKR”数据库，可以显示“RDIFRAMEWOKR”数据库（表、视图、存储过程等）的详细信息，如下图所示。



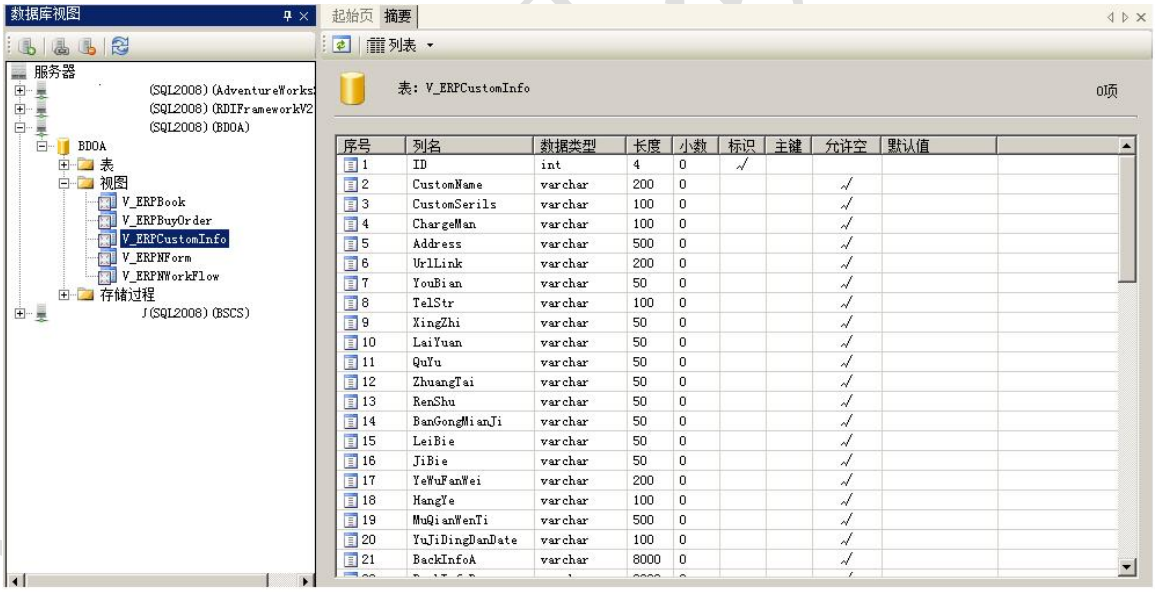
◆ 浏览表的详细信息

用鼠标选择“表”树节点中的任意一个表，我们可以查看该表各字段的详细信息（表名称、数据类型、长度、小数位数、是否是标识列、是否是主键列、是否允许为空、列的默认值等），如下图所示。



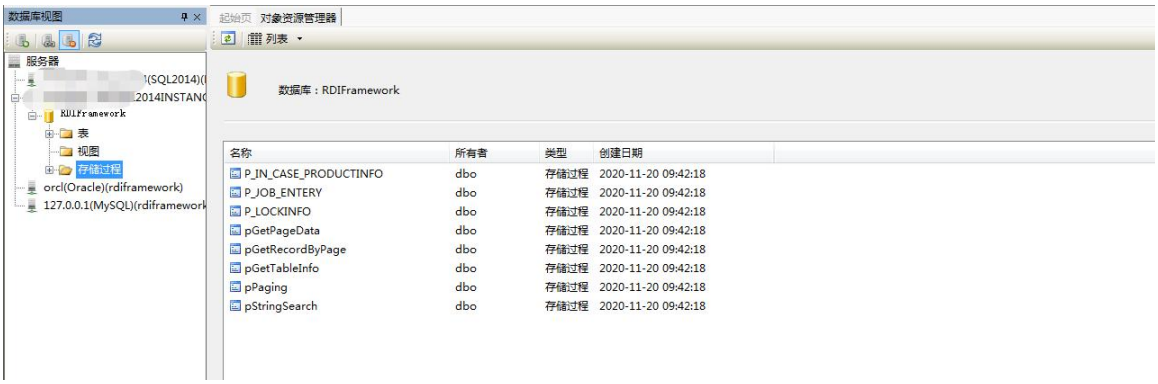
◆ 浏览视图的详细信息

浏览视图的详细信息与表的详细信息差不多，选中一个视图，如下图所示。



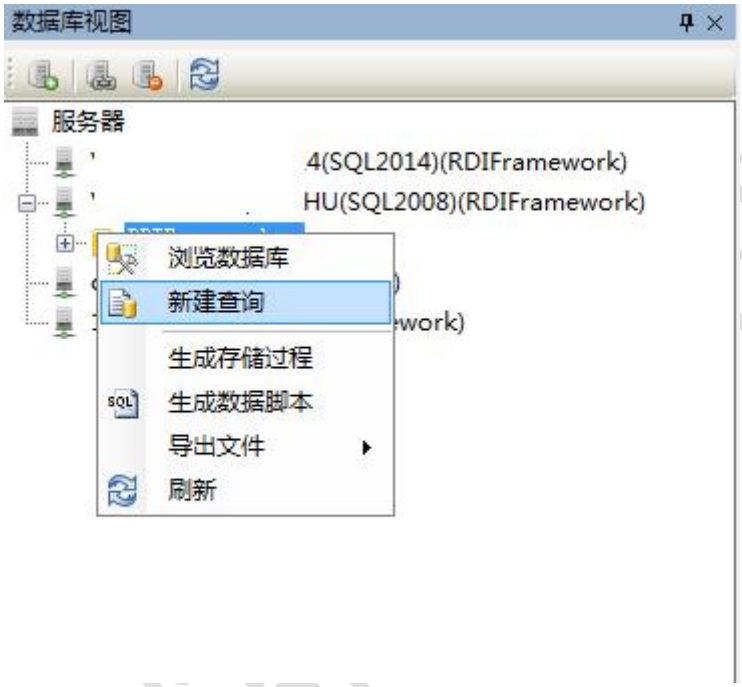
◆ 浏览存储过程的详细信息

选择“存储过程”对节点，显示当前库的所有存储过程信息，如下图所示。

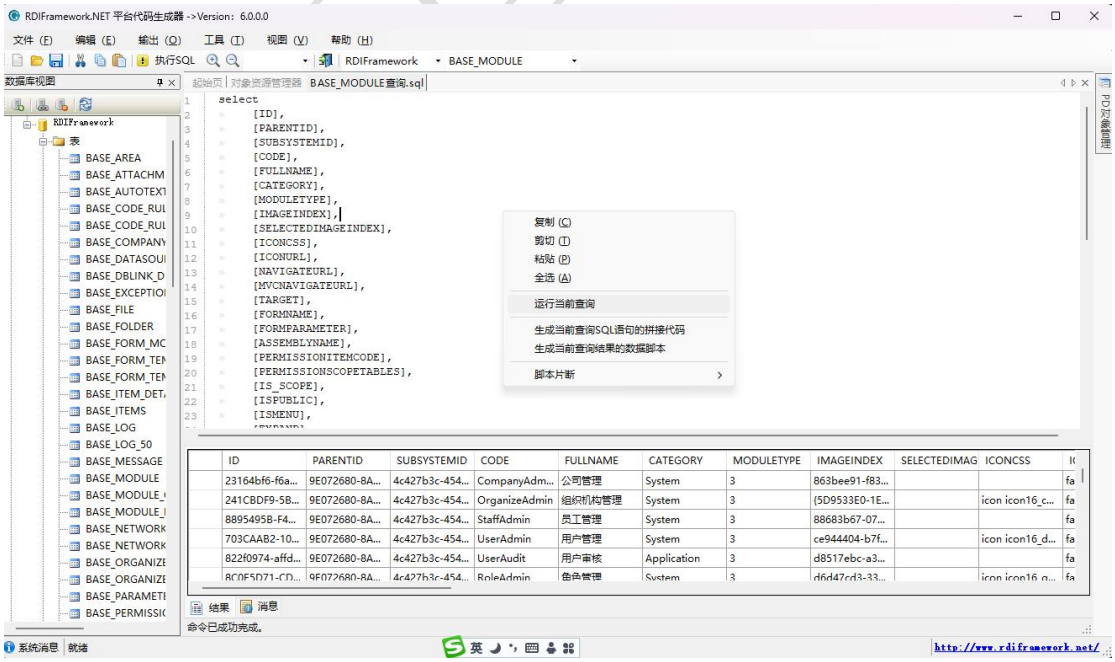


2.2.2. 新建查询

新建查询类似类数据库的查询分析器，我们可以在查询窗体输入正确的 SQL 语句直接进入 SQL 语句操作，需要说明的是：相关的 SQL 语句的执行权限与当前注册服务器的用户名有关。选择某个数据库后，单击鼠标右键，选择“新建查询”，即可打开查询窗口，如下图所示。



单击“新建查询”后，打开当前数据库的查询窗口，如下图所示。



在“查询”窗口，我们可以输入任意有效的 SQL 语句进行执行，在上图中，我们输入了一条查询数据的 SQL 语句，下窗口下方可以看到其执行结果。输入完 sql 语句后，我们可以通过三种方式进行执行：

a) 按“F5”功能键执行。

b) 通过右键菜单，选择“运行当前查询”来执行。

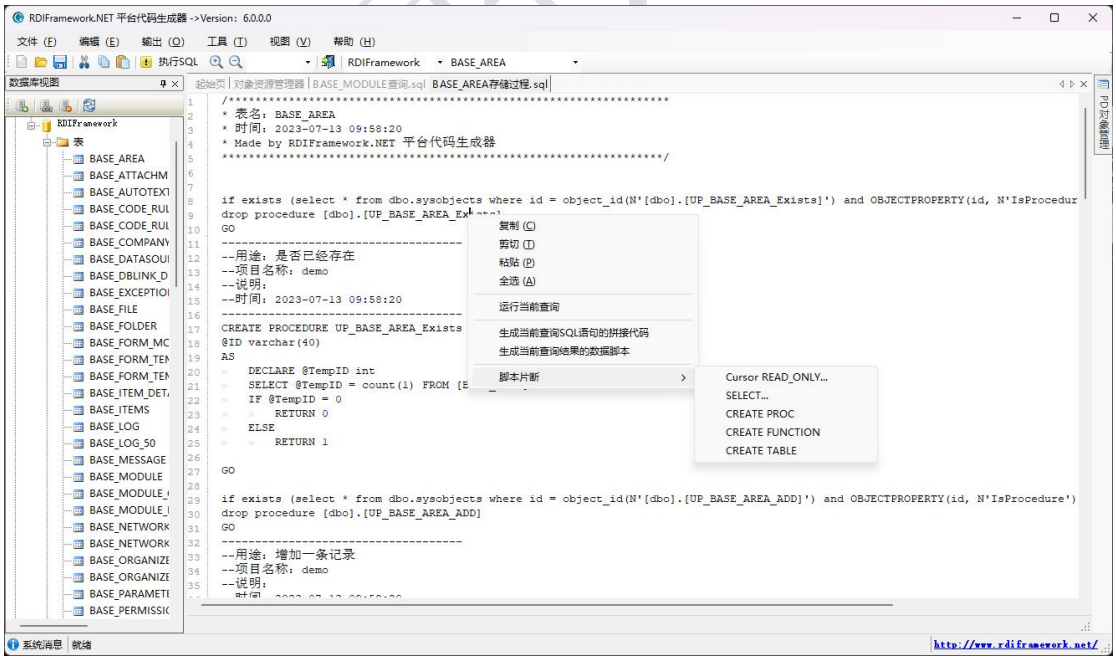
c) 通过工具栏的“执行 SQL”按钮来执行。

注：如果输入的是查询语句，建议不要一次查询过多数据，可以加入 **TOP n** 关键字，对输出的数据条数加以限制。

2.2.3. 生成存储过程

通过右键“数据库”节点，选择“生成存储过程”功能，即可快速当前数据库所有表的业务逻辑存储过程（生成后，用户可根据需要进行相应的修改），这对倾向于存储过程的开发者大大提高了其写存储过程的效率，缩短开发时间，节约开发成本。

根据数据库表的多少，会花一些时间来进行生成，生成后的存储过程如下图所示。



默认对每个表会生成以下几类存储过程：

a) 得到主键字段的最大值（过程命名方式为：表名+“_GetMaxId”）。

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[LinkMan_GetMaxId]') and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[LinkMan_GetMaxId]
GO
----------------------------------
--用途：得到主键字段最大值
--项目名称：
--说明：
--时间：2013-07-29 14:35:30
----------------------------------
CREATE PROCEDURE LinkMan_GetMaxId
AS
    DECLARE @TempID int
    SELECT @TempID = max([Id])+1 FROM [LinkMan]
    IF @TempID IS NULL
        RETURN 1
    ELSE
        RETURN @TempID

GO
```

b) 数据存在性判断（过程命名方式为：表名+“_Exists”）。

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[LinkMan_Exists]') and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[LinkMan_Exists]
GO
----------------------------------
--用途：是否已经存在
--项目名称：
--说明：
--时间：2013-07-29 14:35:30
----------------------------------
CREATE PROCEDURE LinkMan_Exists
@Id int
AS
    DECLARE @TempID int
    SELECT @TempID = count(1) FROM [LinkMan] WHERE Id=@Id
    IF @TempID = 0
        RETURN 0
    ELSE
        RETURN 1

GO
```

c) 新增数据（过程命名方式为：表名+“_ADD”）。

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[LinkMan_ADD]') and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[LinkMan_ADD]
GO
----------------------------------
--用途：增加一条记录
--项目名称：
--说明：
--时间：2013-07-29 14:35:30
----------------------------------
CREATE PROCEDURE LinkMan_ADD
@Id int output,@CustomerId int,@Name nvarchar(50),@Sex nvarchar(5),
@Postion nvarchar(50),@Department nvarchar(50),@MainLinkMan int,
@MobilePhone nvarchar(50),@Telephone nvarchar(50),@ShortNumber varchar(50),
@IDCard varchar(50),@OfficeAddress varchar(200),@OfficeFax varchar(50),
@HomePhone varchar(50),@Education varchar(50),@School varchar(50),
@Degree varchar(50),@HomeZipCode varchar(50),@HomeAddress varchar(200),
@HomeFax varchar(50),@NativePlace varchar(50),@Party varchar(50),
@Nation varchar(50),@Nationality varchar(50),@Major nvarchar(50),
@EducationalBackground nvarchar(50),@BirthdayType int,@Birthday date,
@BloodType nvarchar(10),@QQ nvarchar(50),@Email nvarchar(50),
@Interest nvarchar(200),@Description nvarchar(800),@DeleteMark int,
@Enabled int,@SortCode int,@CreateOn datetime,@CreateUserId nvarchar(50),
@CreateBy nvarchar(50),@ModifiedOn datetime,@ModifyUserId nvarchar(50),
@ModifiedBy nvarchar(50)
 AS
    INSERT INTO [LinkMan](
    [CustomerId],[Name],[Sex],[Postion],[Department],[MainLinkMan],[MobilePhone],[Telephone],[ShortNumber],[IDCard],[OfficeAddress],[OfficeF
    )VALUES(
    @CustomerId,@Name,@Sex,@Postion,@Department,@MainLinkMan,@MobilePhone,@Telephone,@ShortNumber,@IDCard,@OfficeAddress,@OfficeFax,@HomePho
    )
    SET @Id = @@IDENTITY

GO
```

d) 修改数据（过程命名方式为：表名+“_Update”）。

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[LinkMan_Update]') and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[LinkMan_Update]
GO
----------------------------------
--用途：修改一条记录
--项目名称：
--说明：
--时间：2013-07-29 14:35:30
----------------------------------
CREATE PROCEDURE LinkMan_Update
@Id int,@CustomerId int,@Name nvarchar(50),@Sex nvarchar(5),@Postion nvarchar(50),@Department nvarchar(50),
@MainLinkMan int,@MobilePhone nvarchar(50),@Telephone nvarchar(50),@ShortNumber varchar(50),@IDCard varchar(50),
@OfficeAddress varchar(200),@OfficeFax varchar(50),@HomePhone varchar(50),@Education varchar(50),@School varchar(50),
@Degree varchar(50),@HomeZipCode varchar(50),@HomeAddress varchar(200),@HomeFax varchar(50),@NativePlace varchar(50),
@Party varchar(50),@Nation varchar(50),@Nationality varchar(50),@Major nvarchar(50),@EducationalBackground nvarchar(50),
@BirthdayType int,@Birthday date,@BloodType nvarchar(10),@QQ nvarchar(50),@Email nvarchar(50),@Interest nvarchar(200),
@Description nvarchar(800),@DeleteMark int,@Enabled int,@SortCode int,@CreateOn datetime,@CreateUserId nvarchar(50),
@CreateBy nvarchar(50),@ModifiedOn datetime,@ModifyUserId nvarchar(50),@ModifiedBy nvarchar(50)
 AS
    UPDATE [LinkMan] SET
    [CustomerId] = @CustomerId,[Name] = @Name,[Sex] = @Sex,[Postion] = @Postion,[Department] = @Department,[MainLinkMan] = @MainLinkMan,[M
    WHERE Id=@Id

GO
```

e) 删除数据（过程命名方式为：表名+“_Delete”）。

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[LinkMan_Delete]') and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[LinkMan_Delete]
GO
-------------------------------------
--用途：删除一条记录
--项目名称：
--说明：
--时间：2013-07-29 14:35:30
-------------------------------------
CREATE PROCEDURE LinkMan_Delete
@Id int
AS
    DELETE [LinkMan]
     WHERE Id=@Id

GO
```

f) 得到实体模型（过程命名方式为：表名+“GetEntity”）。

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[LinkManGetEntity]') and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[LinkManGetEntity]
GO
-------------------------------------
--用途：得到实体对象的详细信息
--项目名称：
--说明：
--时间：2013-07-29 14:35:30
-------------------------------------
CREATE PROCEDURE LinkManGetEntity
@Id int
AS
    SELECT
    Id,CustomerId,Name,Sex,Postion,Department,MainLinkMan,MobilePhone,Telephone,ShortNumber,IDCard,OfficeAddress,OfficeFax,HomePhone,Educat
     FROM [LinkMan]
     WHERE Id=@Id

GO
```

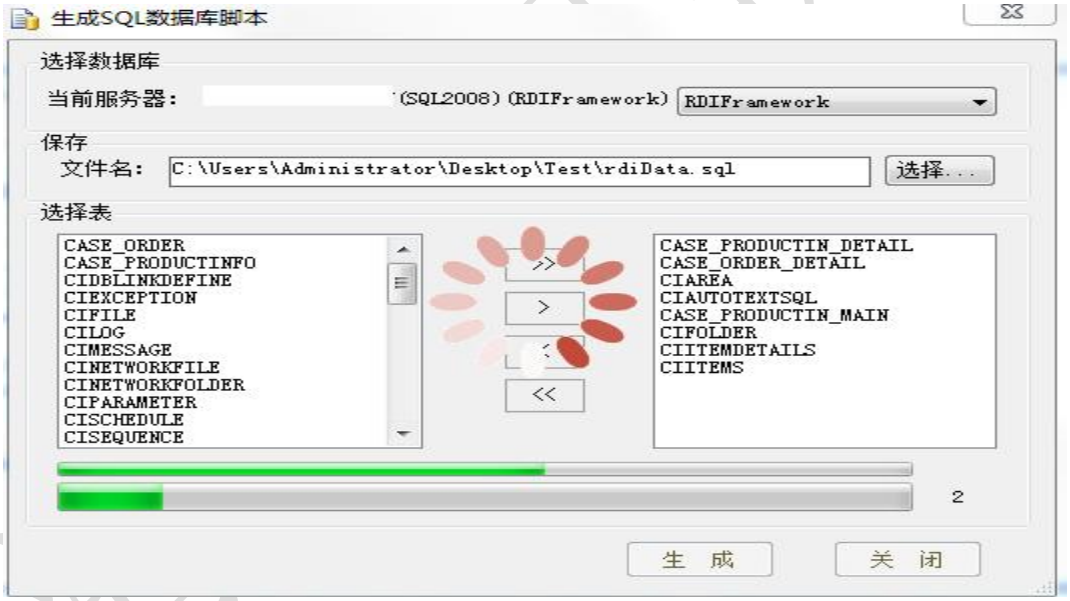
g) 得到数据列表（过程命名方式为：表名+“_GetList”）。

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[LinkMan_GetList]') and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[LinkMan_GetList]
GO
-------------------------------------
--用途：查询记录信息
--项目名称：
--说明：
--时间：2013-07-29 14:35:30
-------------------------------------
CREATE PROCEDURE LinkMan_GetList
AS
    SELECT
    Id,CustomerId,Name,Sex,Postion,Department,MainLinkMan,MobilePhone,Telephone,ShortNumber,IDCard,OfficeAddress,OfficeFax,HomePhone,Educa
     FROM [LinkMan]

GO
```

2.2.4. 生成数据脚本

“生成数据脚本”功能就是针对当前数据库，生成所有表的数据脚本，如下图所示，右键某个数据库节点，选择“生成数据脚本”。

打开“生成 SQL 数据库脚本”对话框，如下图所示。



在上图中，我们可以选择对应的数据库，默认选择的是当前所选数据库，选择需要生成数据脚本的表，设置好保存路径后就可以通过单击“生成”按钮，来生成所选数据库所表数据表的数据脚本（包括表的创建脚步与表的数据脚本）。部分脚本列表如下：

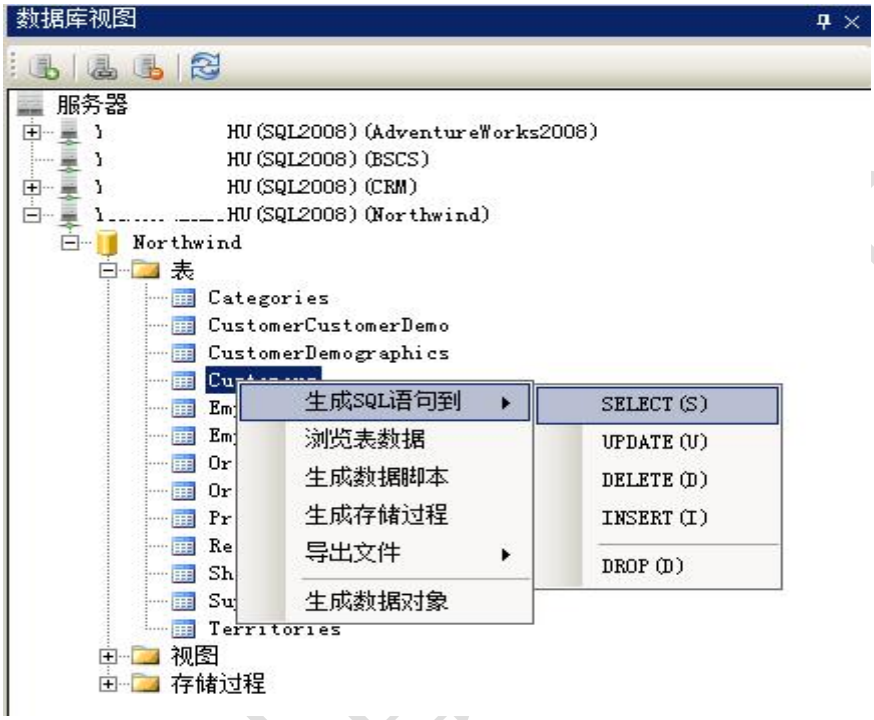
```
SET IDENTITY_INSERT [Customer] OFF
if exists (select * from sysobjects where id = OBJECT_ID('[CustomerClass]') and OBJECTPROPERTY(id, 'IsUserTable') = 1)
DROP TABLE [CustomerClass]

CREATE TABLE [CustomerClass] (
[Id] [int]  IDENTITY (1, 1)  NOT NULL,
[ParentId] [int]  NULL,
[ClassName] [nvarchar]  (50) NOT NULL,
[ClassCode] [nvarchar]  (30) NULL,
[Description] [nvarchar]  (800) NULL,
[SortCode] [int]  NULL,
[DeleteMark] [int]  NOT NULL DEFAULT ((0)),
[CreateOn] [datetime]  NOT NULL DEFAULT (getdate()),
[CreateUserId] [nvarchar]  (50) NULL,
[CreateBy] [nvarchar]  (50) NULL,
[ModifiedOn] [datetime]  NULL,
[ModifyUserId] [nvarchar]  (50) NULL,
[ModifiedBy] [nvarchar]  (50) NULL)

ALTER TABLE [CustomerClass] WITH NOCHECK ADD  CONSTRAINT [PK_CustomerClass] PRIMARY KEY  NONCLUSTERED ( [Id] )
SET IDENTITY_INSERT [CustomerClass] ON

INSERT [CustomerClass] ([Id],[ClassName],[SortCode],[DeleteMark],[CreateOn],[CreateUserId],[CreateBy]) VALUES ( 1,'所有客户',10000000,0,'20
INSERT [CustomerClass] ([Id],[ClassName],[SortCode],[DeleteMark],[CreateOn],[CreateUserId],[CreateBy]) VALUES ( 2,'最近使用客户',10000001,0
INSERT [CustomerClass] ([Id],[ClassName],[SortCode],[DeleteMark],[CreateOn],[CreateUserId],[CreateBy]) VALUES ( 3,'经常使用客户',10000002,0
INSERT [CustomerClass] ([Id],[ClassName],[SortCode],[DeleteMark],[CreateOn],[CreateUserId],[CreateBy]) VALUES ( 4,'固定客户',10000003,0,'20
INSERT [CustomerClass] ([Id],[ClassName],[SortCode],[DeleteMark],[CreateOn],[CreateUserId],[CreateBy]) VALUES ( 5,'所有分类',10000004,0,'20
INSERT [CustomerClass] ([Id],[ParentId],[ClassName],[SortCode],[DeleteMark],[CreateOn],[CreateUserId],[CreateBy]) VALUES ( 6,5,'地区',1000C
INSERT [CustomerClass] ([Id],[ParentId],[ClassName],[SortCode],[DeleteMark],[CreateOn],[CreateUserId],[CreateBy]) VALUES ( 7,5,'关系',1000C
INSERT [CustomerClass] ([Id],[ParentId],[ClassName],[SortCode],[DeleteMark],[CreateOn],[CreateUserId],[CreateBy],[ModifiedOn],[ModifiedBy]
INSERT [CustomerClass] ([Id],[ParentId],[ClassName],[SortCode],[DeleteMark],[CreateOn],[CreateUserId],[CreateBy],[ModifiedOn],[ModifiedBy]
INSERT [CustomerClass] ([Id],[ParentId],[ClassName],[SortCode],[DeleteMark],[CreateOn],[CreateUserId],[CreateBy],[ModifiedOn],[ModifiedBy]
INSERT [CustomerClass] ([Id],[ParentId],[ClassName],[SortCode],[DeleteMark],[CreateOn],[CreateUserId],[CreateBy],[ModifiedOn],[ModifiedBy]
INSERT [CustomerClass] ([Id],[ParentId],[ClassName],[SortCode],[DeleteMark],[CreateOn],[CreateUserId],[CreateBy],[ModifiedOn],[ModifiedBy]
```

2.2.5. 导出文件（存储过程与数据脚本）

导出存储过程与导出数据脚本就是直接把数据库的存储过程、建表脚本和数据脚本保存到指定目录的文件中，而不是生成在界面上，生成的代码与上面两节的一至。

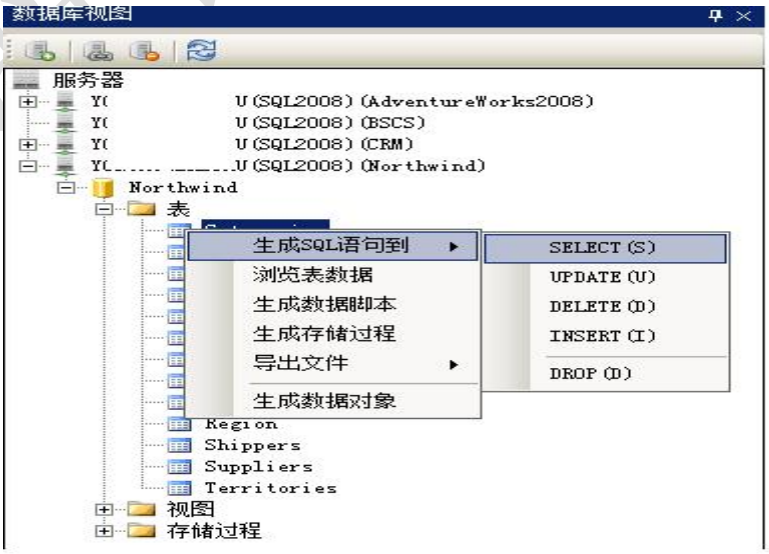
2.3. 表管理

连接到一个已注册的服务器，我们可以对当前服务器所选“数据库”的表进行相应的操作，主要包括：生成 SQL 语句（包括：SELECT 语句、UPDATE 语句、DELETE 语句、INSERT 语句、DROP 语句）、浏览表数据、生成数据脚本、生成存储过程、导出文件（存储过程与数据脚本）、生成数据对象等。

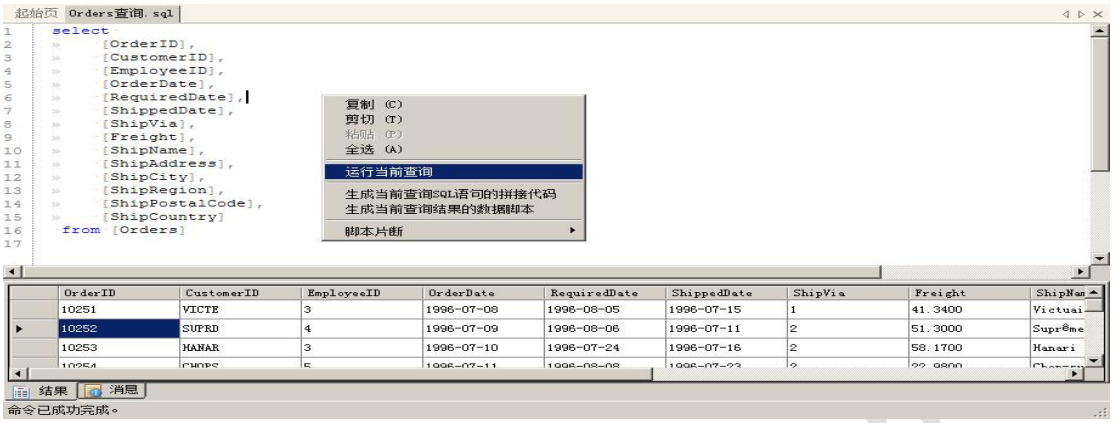


2.3.1. 生成 SQL 语句

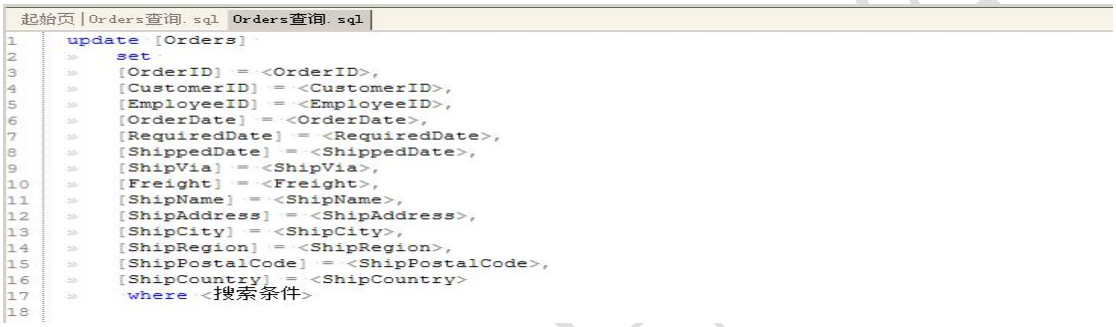
“生成 SQL 语句”功能，可能当前所选数据表自动生成其 SELECT 语句、UPDATE 语句、DELETE 语句、INSERT 语句、DROP 语句等，对生成的语句可以直接在查询窗口执行，与在数据库管理器中的执行效果一样。



1) 生成 SELECT 语句。

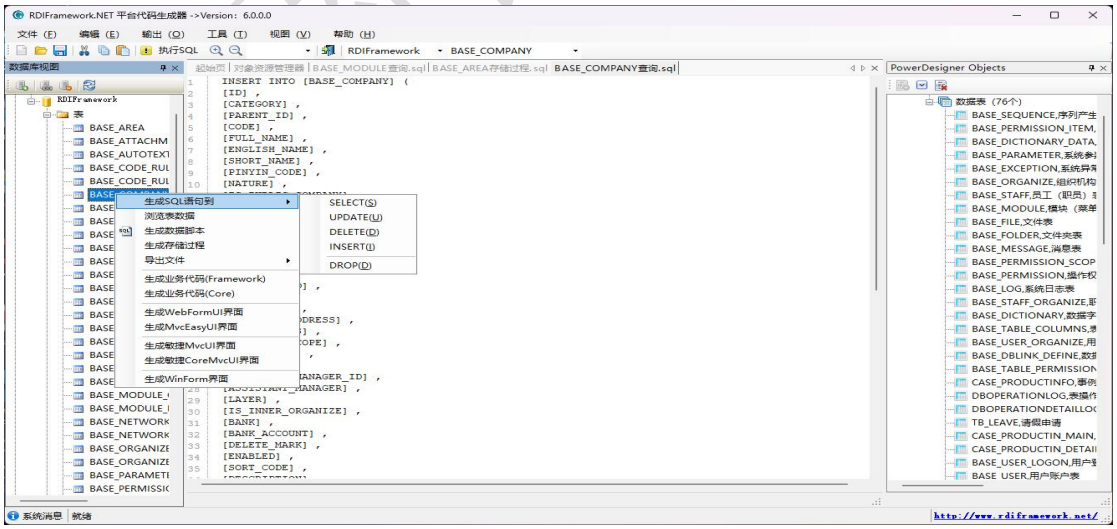


2) 生成 UPDATE 语句。

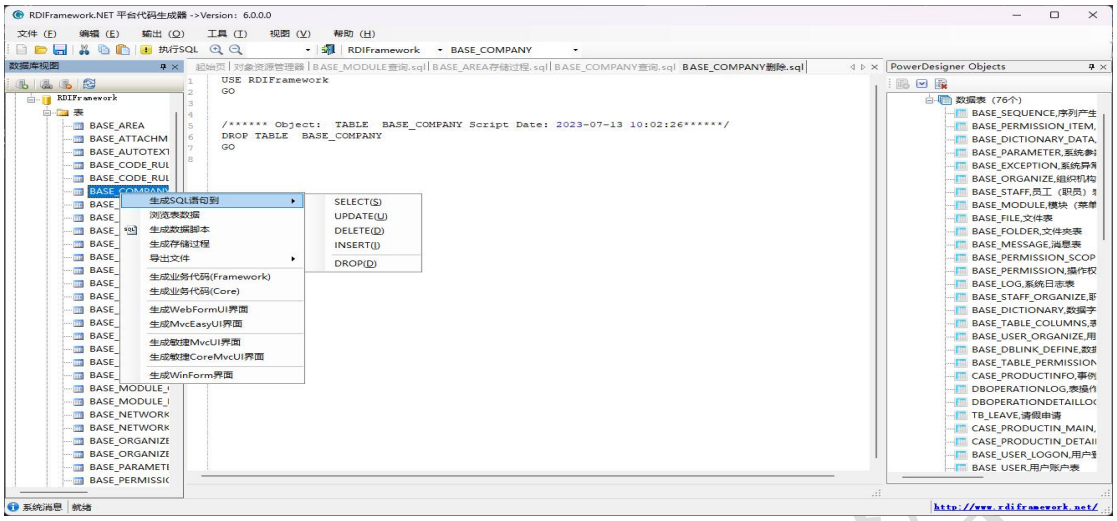


3) 生成 DELETE 语句。

4) 生成 INSERT 语句。



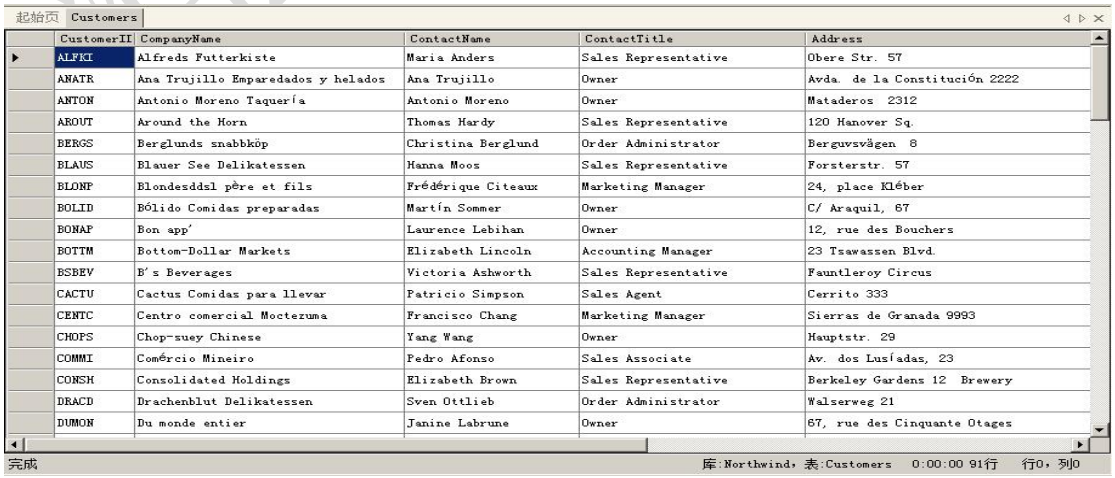
5) 生成 DROP 语句。



2.3.2. 浏览表数据

浏览表数据类似于在查询窗口执行“SELECT*FROM 表名”语句，选择一个表节点，右键单击选择 “浏览表数据”，即可查看当前表的所有表数据。

Tips：如果表数据量过大，建议不要用此方法来浏览数据，可以使用查询窗口直接输入 SELECT 语句，使用 TOP N 来限制数据的条数。



起始页 | Customers

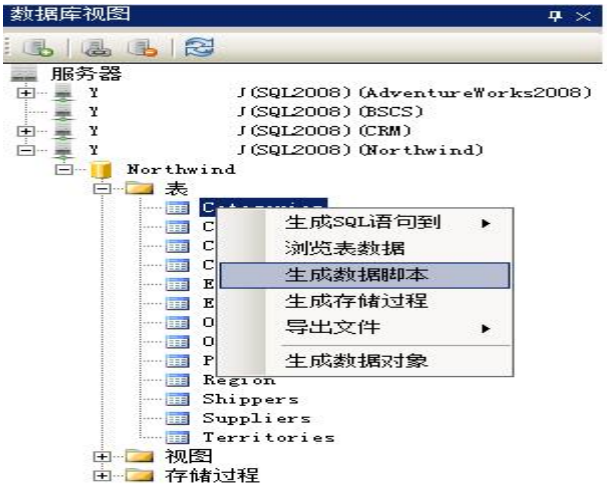
CustomerID	CompanyName	ContactName	ContactTitle	Address
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57
ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222
ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312
AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.
BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator	Berguvsvägen 8
BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative	Forsterstr. 57
BLONP	Blondesddsl père et fils	Frédérique Citeaux	Marketing Manager	24, place Kléber
BOLID	Bólido Comidas preparadas	Martín Sommer	Owner	C/ Araquil, 67
BONAP	Bon app'	Laurence Lebihan	Owner	12, rue des Bouchers
BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manager	23 Tsawassen Blvd.
BSBEV	B's Beverages	Victoria Ashworth	Sales Representative	Fauntleroy Circus
CACTU	Cactus Comidas para llevar	Patricio Simpson	Sales Agent	Cerrito 333
CENTC	Centro comercial Moctezuma	Francisco Chang	Marketing Manager	Sierras de Granada 9993
CHOPS	Chop-suey Chinese	Yang Wang	Owner	Hauptstr. 29
COMMI	Comércio Mineiro	Pedro Afonso	Sales Associate	Av. dos Lusíadas, 23
CONSH	Consolidated Holdings	Elizabeth Brown	Sales Representative	Berkeley Gardens 12 Brewery
DRACD	Drachenblut Delikatessen	Sven Ottlieb	Order Administrator	Walserweg 21
DUMON	Du monde entier	Janine Labrune	Owner	67, rue des Cinquante Otages

完成　　库:Northwind, 表:Customers　0:00:00 91行　行0，列0

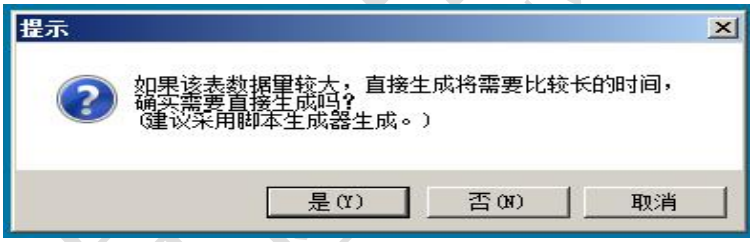
在上图中，我们可以通过右下角查看当前数据库、当前表、执行此操作所花费的时间，数据的行数等信息。

2.3.3. 生成数据脚本

生成数据脚本功能就是生成当前所选表的创建脚本与数据脚本。



在选择“生成数据脚本”功能后，会弹出对话框，让用户选择，如下图所示。



选择“否”即可打开“生成 SQL 数据库脚本”窗口，给 2.2.4 节的一至。选择“是”，则可直接生成当前表的数据脚本，如下图所示。

起始页 Categories脚本.sql

```
if exists (select * from sysobjects where id = OBJECT_ID('Categories') and OBJECTPROPERTY(id, 'IsUserTable') = 1)
DROP TABLE Categories

CREATE TABLE Categories (
[CategoryID] [int] IDENTITY (1, 1) NOT NULL,
[CategoryName] [nvarchar] (15) NOT NULL,
[Description] [ntext] NULL,
[Picture] [image] NULL)

ALTER TABLE Categories WITH NOCHECK ADD CONSTRAINT [PK_Categories] PRIMARY KEY NONCLUSTERED ( [CategoryID] )
SET IDENTITY_INSERT Categories ON

INSERT Categories ([CategoryID],[CategoryName],[Description],[Picture]) VALUES ( 1,'Beverages','Soft drinks, coffees, teas, beer
INSERT Categories ([CategoryID],[CategoryName],[Description],[Picture]) VALUES ( 2,'Condiments','Sweet and savory sauces, relish
INSERT Categories ([CategoryID],[CategoryName],[Description],[Picture]) VALUES ( 3,'Confections','Desserts, candies, and sweet b
INSERT Categories ([CategoryID],[CategoryName],[Description],[Picture]) VALUES ( 4,'Dairy Products','Cheeses',System.Byte[])
INSERT Categories ([CategoryID],[CategoryName],[Description],[Picture]) VALUES ( 5,'Grains/Cereals','Breads, crackers, pasta, an
INSERT Categories ([CategoryID],[CategoryName],[Description],[Picture]) VALUES ( 6,'Meat/Poultry','Prepared meats',System.Byte[]
INSERT Categories ([CategoryID],[CategoryName],[Description],[Picture]) VALUES ( 7,'Produce','Dried fruit and bean curd',System.
INSERT Categories ([CategoryID],[CategoryName],[Description],[Picture]) VALUES ( 8,'Seafood','Seaweed and fish',System.Byte[])

SET IDENTITY_INSERT Categories OFF
```

Tips:如果所选表的数据过多，通过上面的方式生成数据脚本可能会花很长时间，此时建议在单击“生成数据脚本”后弹出的提示框时选择“否”打开“生成 SQL 数据库脚本”窗口来进行表数据脚本的生成。

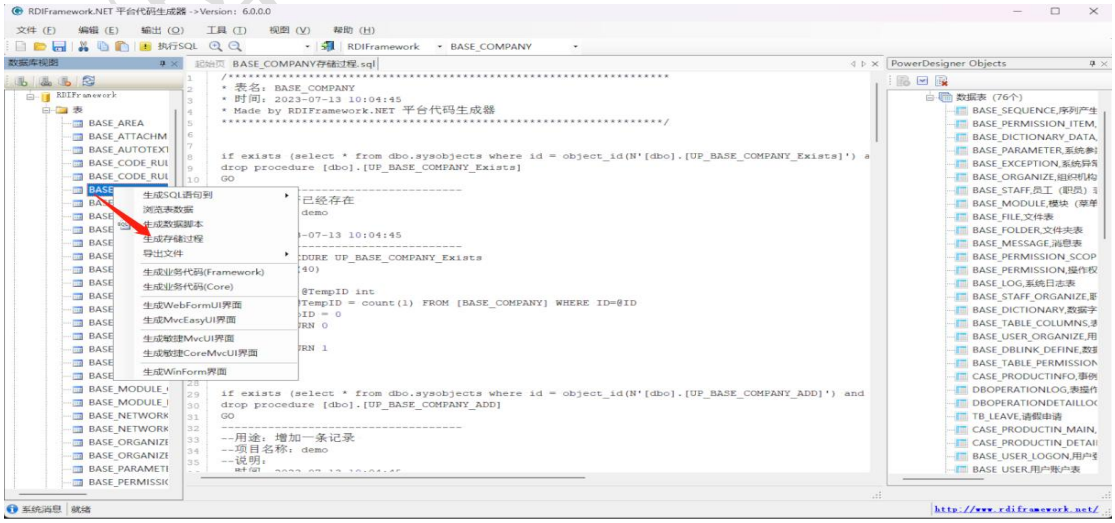
2.3.4. 生成存储过程

“生成存储过程”就是生成当前所选表的业务逻辑存储过程。

生成的存储过程包括以下几种类型：

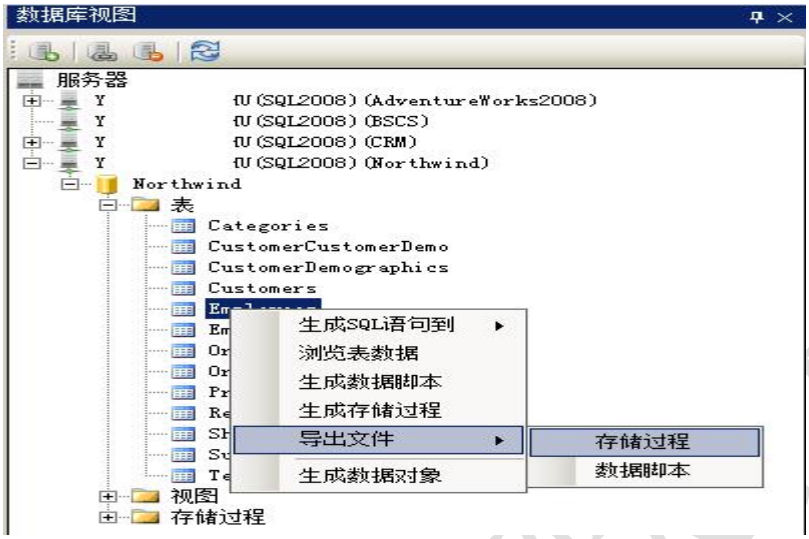
a) 得到主键字段的最大值（过程命名方式为：表名+“_GetMaxId”）。
b) 数据存在性判断（过程命名方式为：表名+“_Exists”）。
c) 新增数据（过程命名方式为：表名+“_ADD”）。
d) 修改数据（过程命名方式为：表名+“_Update”）。
e) 删除数据（过程命名方式为：表名+“_Delete”）。
f) 得到实体模型（过程命名方式为：表名+“GetEntity”）。
g) 得到数据列表（过程命名方式为：表名+“_GetList”）。

部分截图如下：



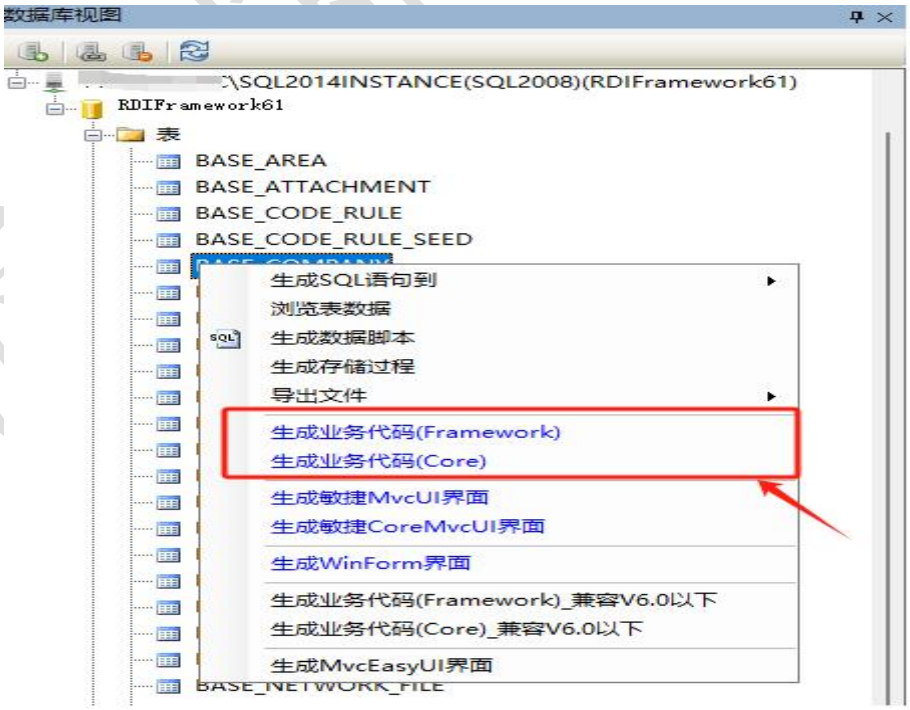
2.3.5. 导出文件（存储过程与数据脚本）

导出存储过程与导出数据脚本就是直接把所选表的存储过程、建表脚本和表数据脚本保存到指定目录的文件中，而不是生成在界面上，生成的代码与上面两节的一至。

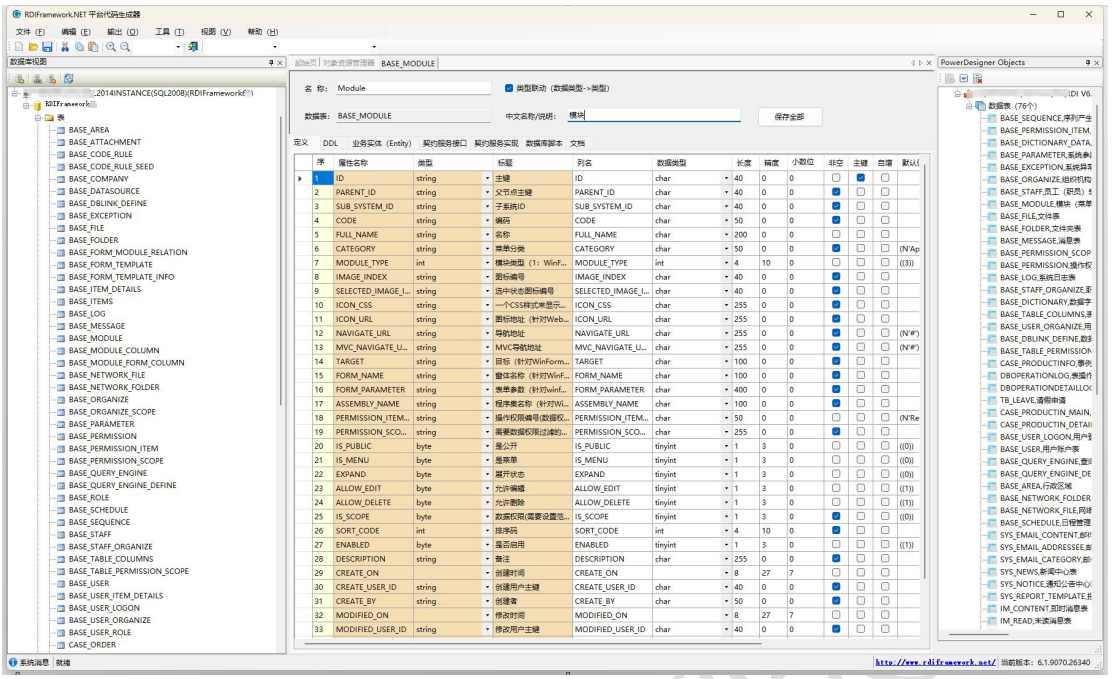


2.3.6. 生成业务代码（重量级）

“生成业务代码”分为生成支持.NET Framework 与.NET Core 版本的业务逻辑代码，通过此模块，可以生成所需的核心业务逻辑，注意：我们的代码生成器，是基于 RIDFramework.NET 框架的代码生成器。选择一个数据表，单击右键，选择“业务逻辑代码（Framework）”或“生成业务逻辑代码（Core）”功能，如下图所示。



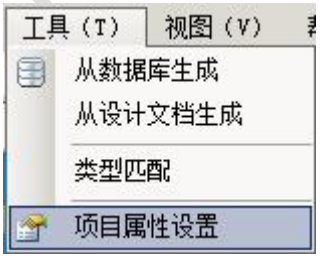
打开所选表的数据库对象界面，在该界面中，显示了当前所选表的概要信息以及包含八个选项卡信息，下面的讲解主要围绕这八个选项卡进行，如下图所示。



在进入下面的内容前，需要简单说明一下针对代码生成的公开设置部分，那就是项目属性设置。

2.3.6.1. 项目属性设置

“项目属性”设置，是对代码生成器的公共设置，如设置项目的名称、公司名称、作者名称、代码输出目录等，后面的代码生成相关的信息都要以此设置作为公共引用，要打开“项目属性”窗口，需要通过主菜单“工具”菜单，选择“项目属性设置”，即可打开“项目属性设置”窗口。



打开的“项目属性设置”窗口如下：

在“项目属性设置”的“项目”Tab 窗口，设置代码生成的所需的公共信息如下：

项目名称(程序集名称)：代码生成所需的命名空间的名称。

公司名称：代码生成时版权部分需引用。

开发人员：代码生成的作者信息。

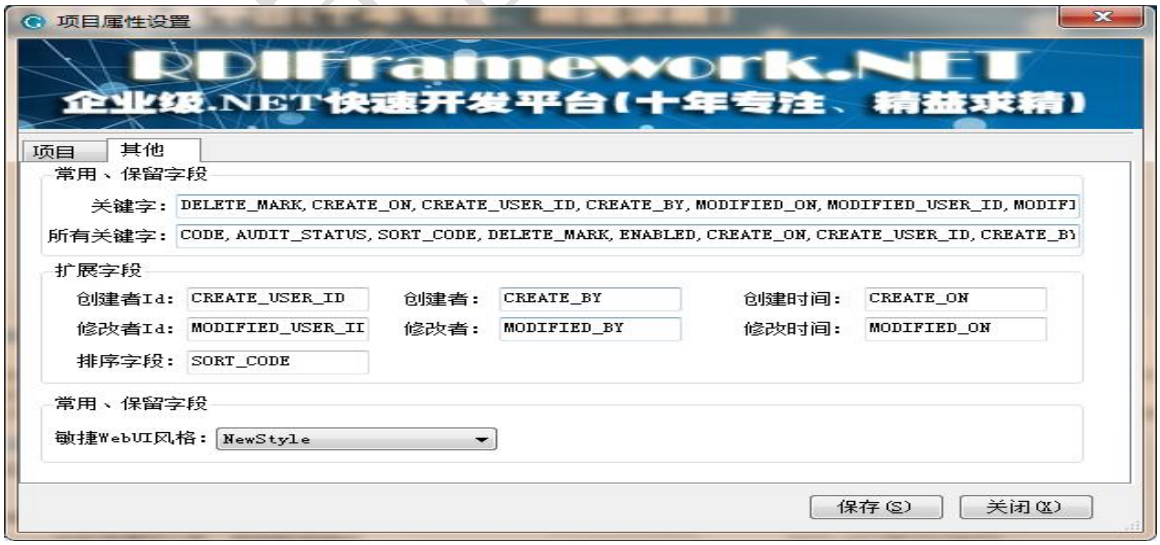
输出区域（Area 名称）：针对 Mvc 的区域名称设置。

代码输出目录：此主要用于代码批量生成时，保存代码的目录。

字段规范化：选中则会对生成的代码做规范化处理，如：FULL_NAME 为变为：FullName，不选中就会原样处理。

去掉的前缀：生成的表需要去掉的前缀，如：设置为 BASE_,则针对 BASE_AREA，会去掉 BASE_，多个以英文逗号隔开。

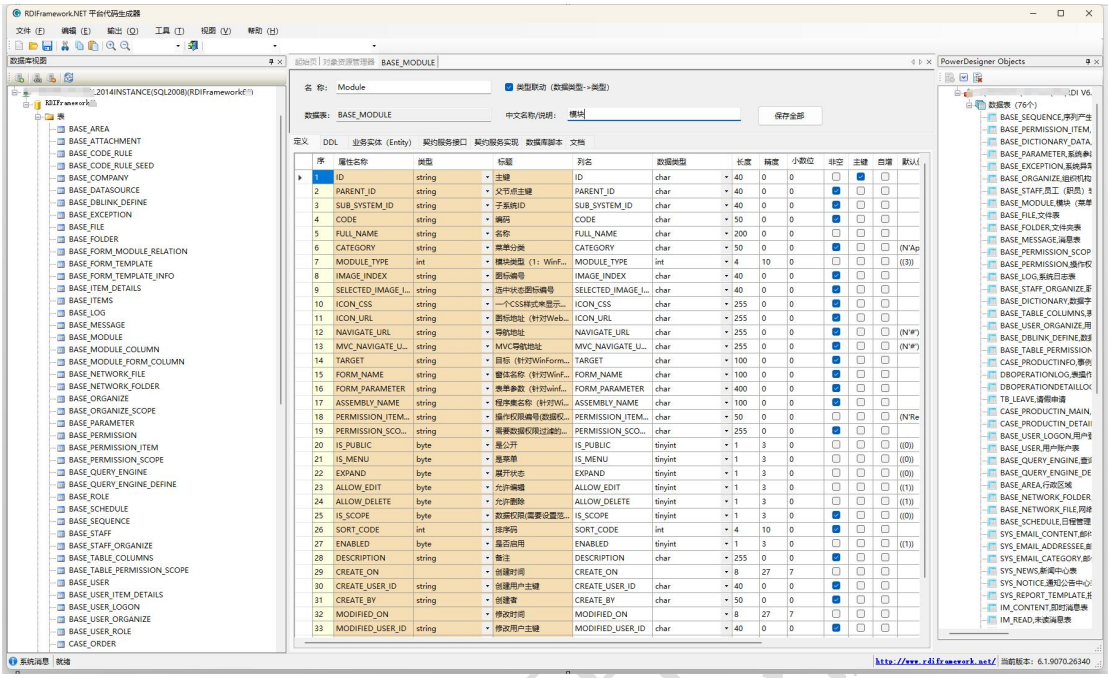
在项目属性的“其他”Tab 页，可以针对一些保留字段、扩展字段等进行设置。



通过以上设置，接下来我们就可以进行下面的工作了。

2.3.6.2. 数据表的定义展示

数据表的定义主要显示的是当前表的列信息与生成的实体代码的对应关系，后期还会增加在此就可以新增列、修改列等，表的定义（DDL）展示如下图所示。



在上图中，以淡黄色底色显示的是列对应到实体的类型对应关系，后半部分显示的是表列的详细信息。

2.3.6.3. 表的 DDL

表的 DDL 可以查看当前表的 CREATE 代码，如下图所示。

定义　DDL　业务实体（Entity）　契约服务接口　契约服务实现　数据库脚本　文档

文件　选项

```
if exists (SELECT * FROM sysobjects where id = OBJECT_ID('[BASE_MODULE]') and OBJECTPROPERTY(id, 'IsUserTable') = 1)
DROP TABLE [BASE_MODULE]

CREATE TABLE [BASE_MODULE] (
[ID] [varchar]  (40) NOT NULL,
[PARENT_ID] [varchar]  (40) NULL,
[SUB_SYSTEM_ID] [varchar]  (40) NULL,
[CODE] [nvarchar]  (50) NULL,
[FULL_NAME] [nvarchar]  (200) NOT NULL,
[CATEGORY] [nvarchar]  (50) NULL DEFAULT (N'Application'),
[MODULE_TYPE] [int]  NOT NULL DEFAULT ((3)),
[IMAGE_INDEX] [varchar]  (40) NULL,
[SELECTED_IMAGE_INDEX] [varchar]  (40) NULL,
[ICON_CSS] [nvarchar]  (255) NULL,
[ICON_URL] [nvarchar]  (255) NULL,
[NAVIGATE_URL] [nvarchar]  (255) NULL DEFAULT (N'#'),
[MVC_NAVIGATE_URL] [nvarchar]  (255) NULL DEFAULT (N'#'),
[TARGET] [nvarchar]  (100) NULL,
[FORM_NAME] [nvarchar]  (100) NULL,
[FORM_PARAMETER] [nvarchar]  (400) NULL,
[ASSEMBLY_NAME] [nvarchar]  (100) NULL,
[PERMISSION_ITEM_CODE] [nvarchar]  (50) NOT NULL DEFAULT (N'Resource.AccessPermission'),
[PERMISSION_SCOPE_TABLES] [nvarchar]  (255) NULL,
[IS_PUBLIC] [tinyint]  NOT NULL DEFAULT ((0)),
[IS_MENU] [tinyint]  NOT NULL DEFAULT ((0)),
[EXPAND] [tinyint]  NOT NULL DEFAULT ((0)),
[ALLOW_EDIT] [tinyint]  NOT NULL DEFAULT ((1)),
[ALLOW_DELETE] [tinyint]  NOT NULL DEFAULT ((1)),
[IS_SCOPE] [tinyint]  NULL DEFAULT ((0)),
[SORT_CODE] [int]  NULL,
[ENABLED] [tinyint]  NOT NULL DEFAULT ((1)),
[DESCRIPTION] [nvarchar]  (255) NULL,
[CREATE_ON] [datetime2]  NOT NULL,
[CREATE_USER_ID] [varchar]  (40) NULL,
[CREATE_BY] [nvarchar]  (50) NULL,
[MODIFIED_ON] [datetime2]  NOT NULL
```

2.3.6.4. 业务实体（Entity）

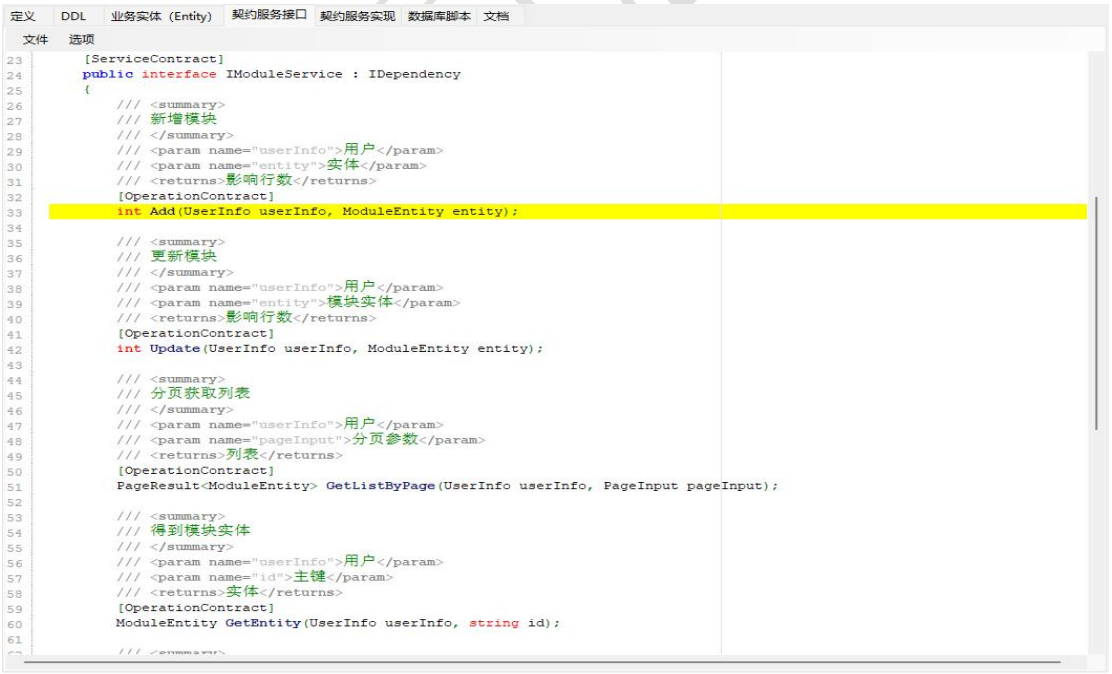
根据数据表，自动生成业务实体（也可称 Model），如下图所示。


```
[RDITable("BASE_MODULE")]
public partial class ModuleEntity : BaseCUDEntity
{
    /// <summary>
    /// 父节点主键
    /// </summary>
    [DataMember]
    [RDIColumn(ColumnName = "PARENT_ID", ColumnDescription = "父节点主键")]
    public string ParentId { get; set; }

    /// <summary>
    /// 子系统ID
    /// </summary>
    [DataMember]
    [RDIColumn(ColumnName = "SUB_SYSTEM_ID", ColumnDescription = "子系统ID")]
    public string SubSystemId { get; set; }

    /// <summary>
    /// 编码
    /// </summary>
    [DataMember]
    [RDIColumn(ColumnName = "CODE", ColumnDescription = "编码")]
    public string Code { get; set; }

    /// <summary>
    /// 名称
    /// </summary>
    [DataMember]
    [RDIColumn(ColumnName = "FULL_NAME", ColumnDescription = "名称")]
    public string FullName { get; set; }

    /// <summary>
    /// 菜单分类
    /// </summary>
    [DataMember]
    [RDIColumn(ColumnName = "CATEGORY", ColumnDescription = "菜单分类")]
    public string Category { get; set; }
```

2.3.6.5. 契约服务接口

“契约服务接口”代码，是业务的接口实现，框架的代码生成器自动生成了一些常用的业务逻辑接口，如：

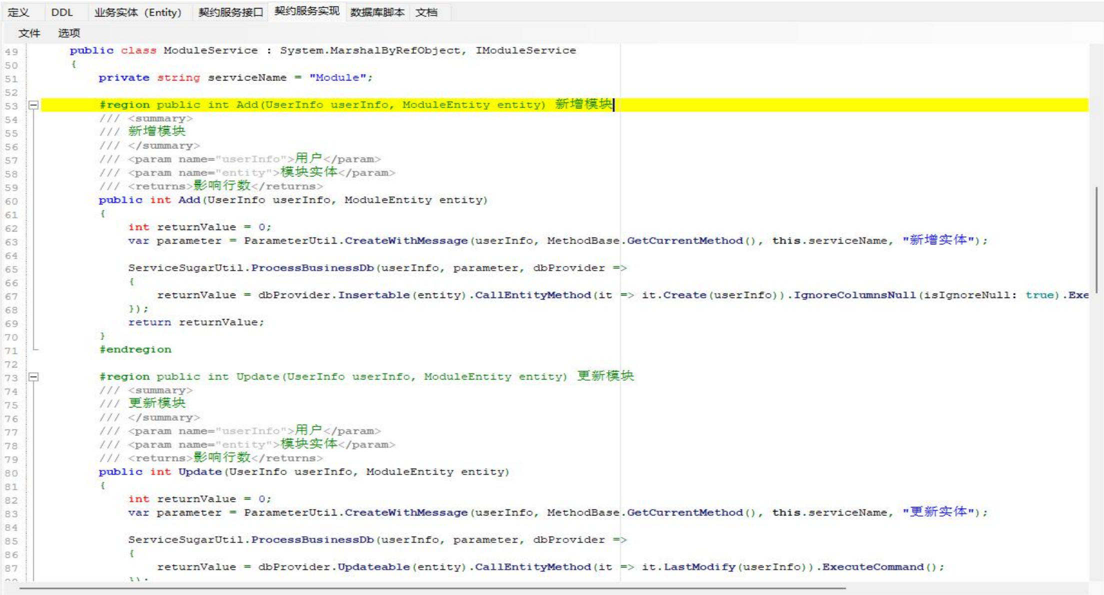
新增数据接口、修改数据接口、删除数据接口、得到数据表、得到分页数据表、得到实体，通过条件得到数据、批量保存等等，如下图所示。




```
[ServiceContract]
public interface IModuleService : IDependency
{
    /// <summary>
    /// 新增模块
    /// </summary>
    /// <param name="userInfo">用户</param>
    /// <param name="entity">实体</param>
    /// <returns>影响行数</returns>
    [OperationContract]
    int Add(UserInfo userInfo, ModuleEntity entity);

    /// <summary>
    /// 更新模块
    /// </summary>
    /// <param name="userInfo">用户</param>
    /// <param name="entity">模块实体</param>
    /// <returns>影响行数</returns>
    [OperationContract]
    int Update(UserInfo userInfo, ModuleEntity entity);

    /// <summary>
    /// 分页获取列表
    /// </summary>
    /// <param name="userInfo">用户</param>
    /// <param name="pageInput">分页参数</param>
    /// <returns>列表</returns>
    [OperationContract]
    PageResult<ModuleEntity> GetListByPage(UserInfo userInfo, PageInput pageInput);

    /// <summary>
    /// 得到模块实体
    /// </summary>
    /// <param name="userInfo">用户</param>
    /// <param name="id">主键</param>
    /// <returns>实体</returns>
    [OperationContract]
    ModuleEntity GetEntity(UserInfo userInfo, string id);
```

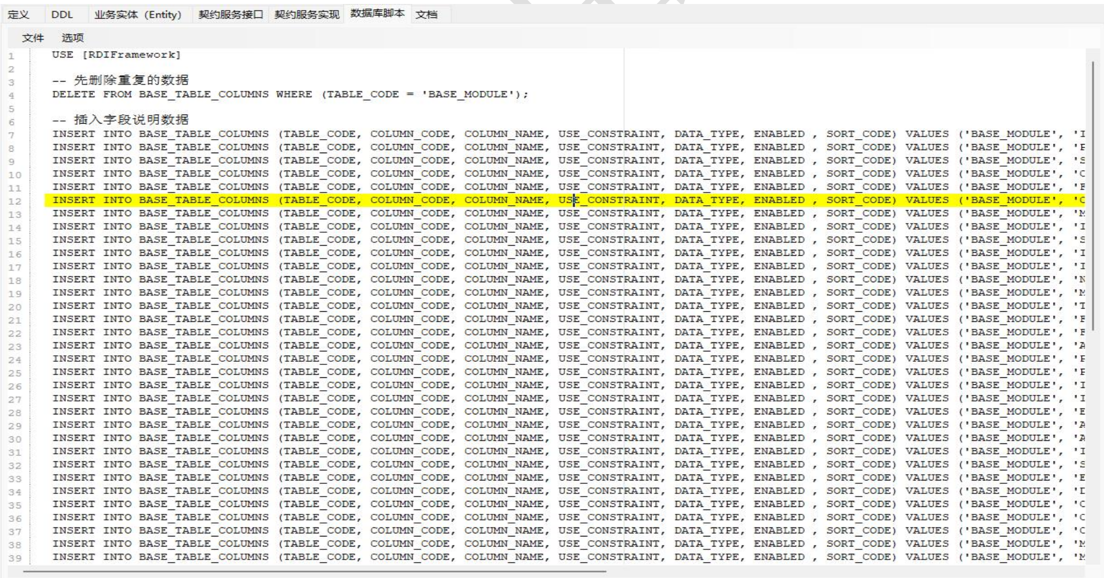
2.3.6.6. 契约服务实现

服务实现是对接口定义的实现，如下图所示。



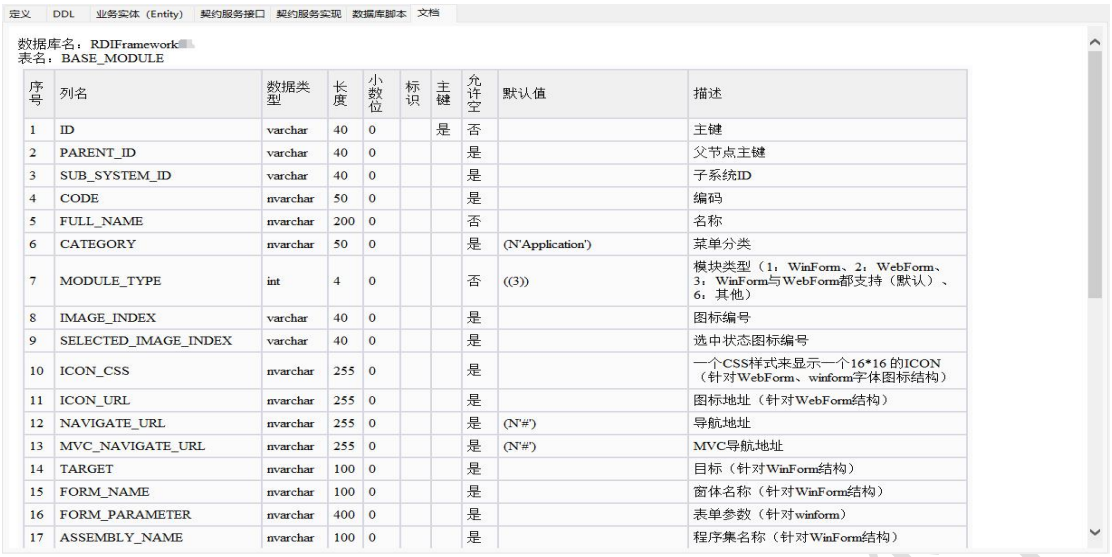
2.3.6.7. 生成数据库脚本

数据库脚本主要用于权限控制表中使用，我们在以通过代码生成器生成待控制的权限控制表脚本数据，如下图所示。



2.3.6.8. 文档

文档就是生成当前数据表的表结构文档，如下图所示。


定义 | DDL | 业务实体（Entity） | 契约服务接口 | 契约服务实现 | 数据库脚本 | 文档

数据库名：RDIFramework
表名：BASE_MODULE

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	描述
1	ID	varchar	40	0		是	否		主键
2	PARENT_ID	varchar	40	0			是		父节点主键
3	SUB_SYSTEM_ID	varchar	40	0			是		子系统ID
4	CODE	nvarchar	50	0			是		编码
5	FULL_NAME	nvarchar	200	0			否		名称
6	CATEGORY	nvarchar	50	0			是	(N'Application')	菜单分类
7	MODULE_TYPE	int	4	0			否	((3))	模块类型（1：WinForm、2：WebForm、3：WinForm与WebForm都支持（默认）、6：其他）
8	IMAGE_INDEX	varchar	40	0			是		图标编号
9	SELECTED_IMAGE_INDEX	varchar	40	0			是		选中状态图标编号
10	ICON_CSS	nvarchar	255	0			是		一个CSS样式来显示一个16*16的ICON（针对WebForm、winform字体图标结构）
11	ICON_URL	nvarchar	255	0			是		图标地址（针对WebForm结构）
12	NAVIGATE_URL	nvarchar	255	0			是	(N'#')	导航地址
13	MVC_NAVIGATE_URL	nvarchar	255	0			是	(N'#')	MVC导航地址
14	TARGET	nvarchar	100	0			是		目标（针对WinForm结构）
15	FORM_NAME	nvarchar	100	0			是		窗体名称（针对WinForm结构）
16	FORM_PARAMETER	nvarchar	400	0			是		表单参数（针对winform）
17	ASSEMBLY_NAME	nvarchar	100	0			是		程序集名称（针对WinForm结构）

2.3.6.9. 辅助功能

在生成的各个代码界面窗口中，有一个工具栏，可对生成的代码做相应的操作（如：保存当前生成的代码等），如下图所示。

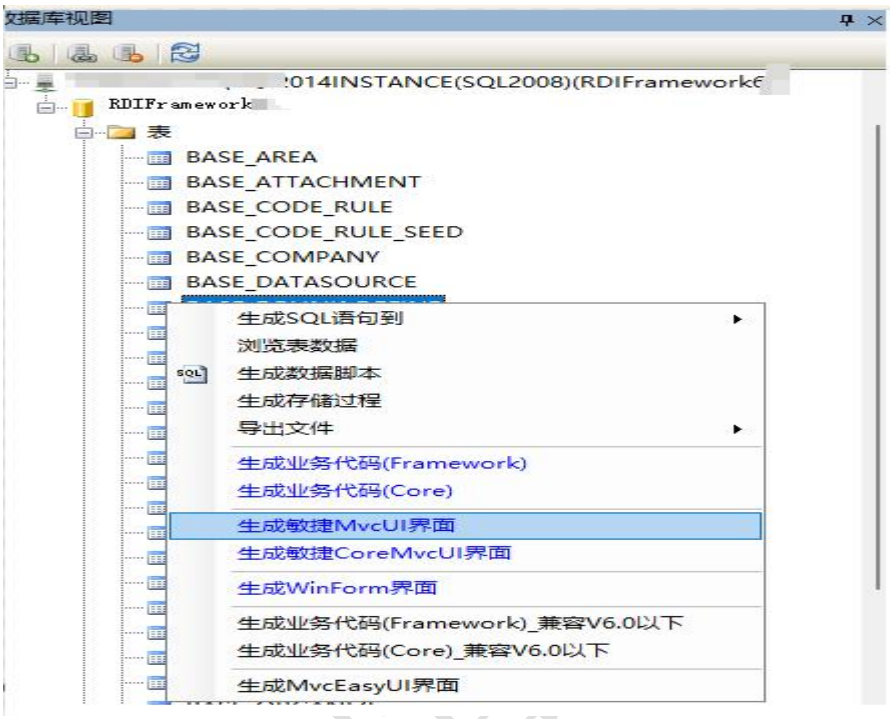
通过“文件”菜单，用户可以保存当前生成的代码。现在说明一下“选项”菜单的功能，“选项”菜单各个子菜单的功能主要是对当前代码生成窗口的代码进行格式控制，所有的代码生成窗口，都共享这些选项菜单，“选项”菜单的子菜单功能如下：

1) 拆分窗口。
2) 显示空格和制表符。
3) 显示换行标记。
4) 显示无效标记。
5) 显示行号。
6) 高亮当前行。
7) 高亮匹配括号当前光标在其后时。
8) 启用虚空格。
9) 制表符转换为空格。

10) 字体（代码的字体设置）。

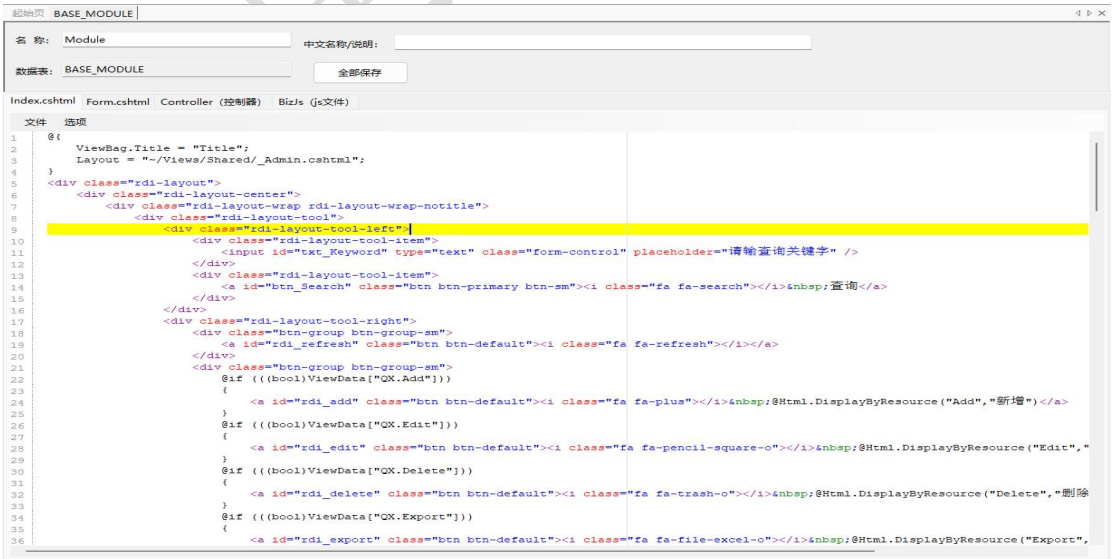
2.3.7. 生成敏捷 MvcUI 界面

“生成敏捷 MvcUI 界面”功能模块是 3.5 版本开始新增的功能，可以生成敏捷 MVC 的 UI 界面代码，这样就可以大大缩小界面的开发时间，对生成的 UI 代码做适量的修改即可。选择一个数据表，单击右键，选择“生成敏捷 MvcUI 界面”功能，如下图所示。



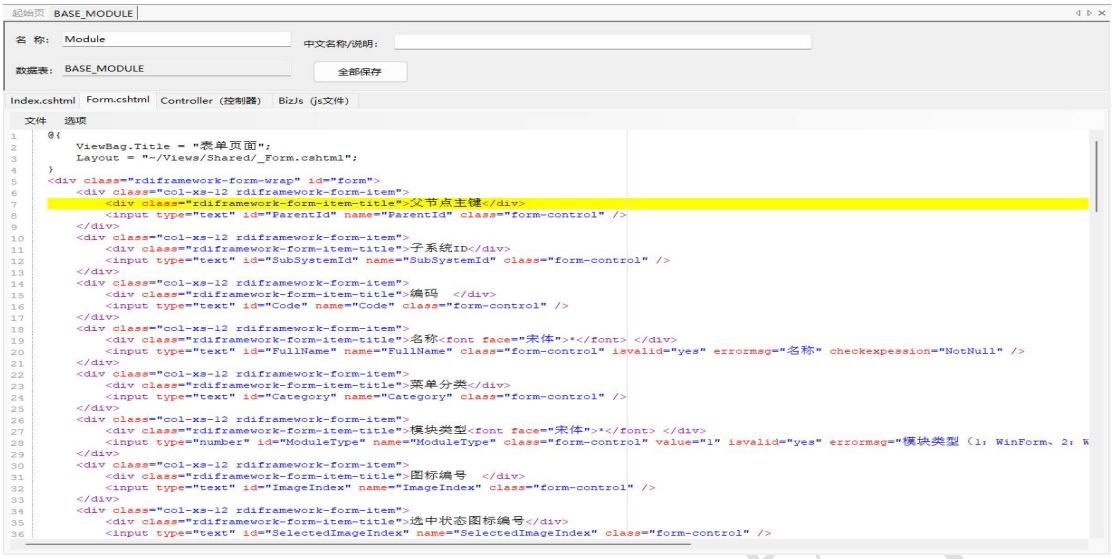
2.3.7.1. Index.cshtml 页面代码

MvcUI 的 Index.cshtml 页面代码如下图所示。



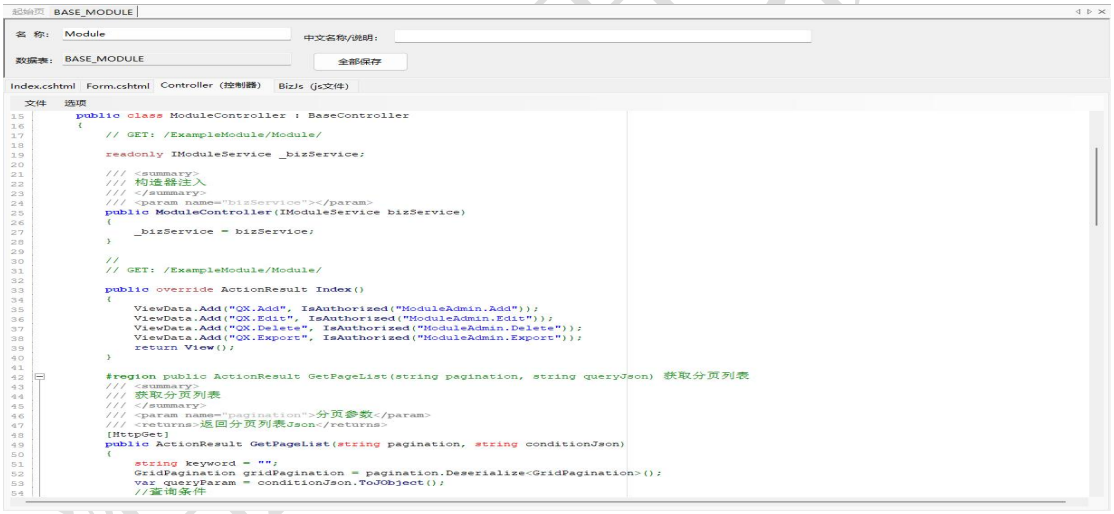
2.3.7.2. 编辑界面 Form.cshtml 代码

编辑代码 Form.cshtml 代码如下图所示。



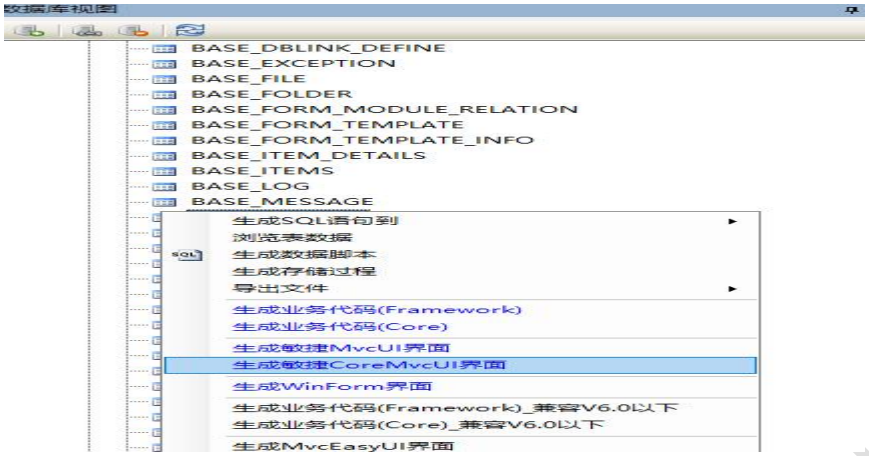
2.3.7.3. Controller 控制器

Controller 控制器代码如下图所示。



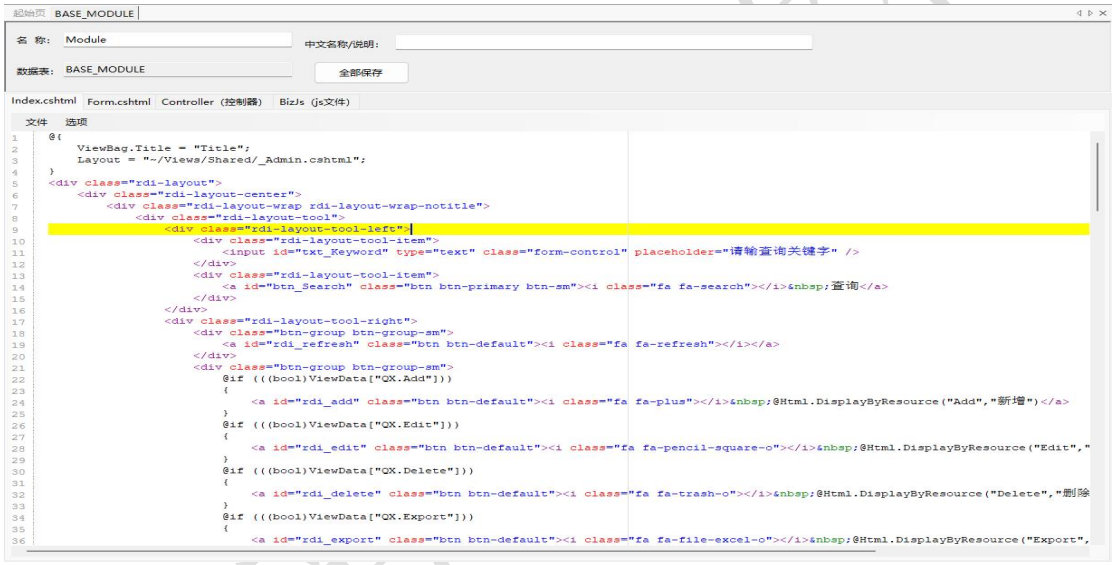
2.3.8. 生成敏捷 CoreMvcUI 界面

“生成敏捷 CoreMvcUI 界面”功能模块用于生成.NET Core 版本的敏捷 MVC 的 UI 界面代码，选择一个数据表，单击右键，选择“生成敏捷 CoreMvcUI 界面”功能，如下图所示。



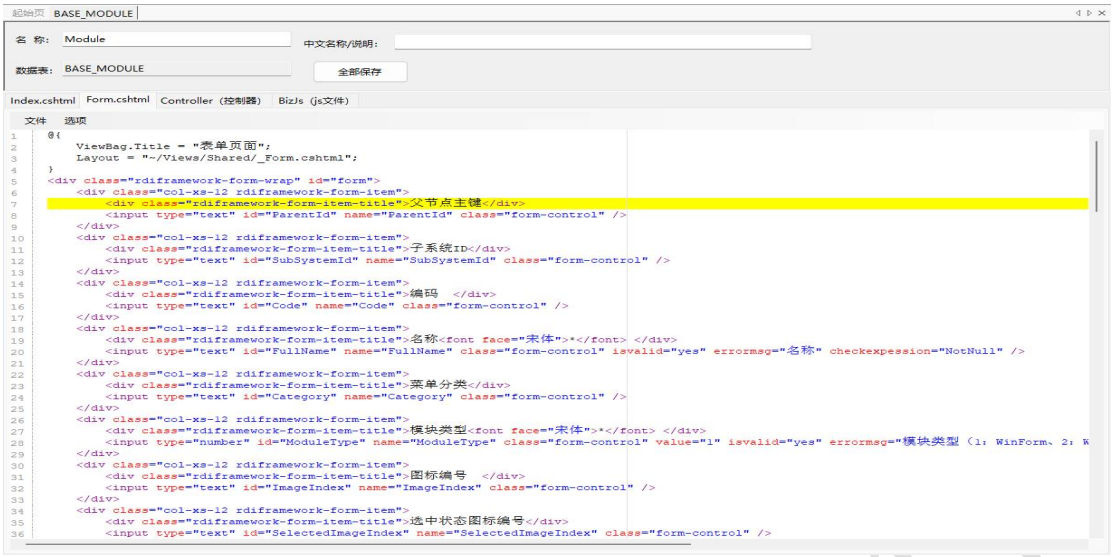
2.3.8.1. Index.cshtml 页面代码

敏捷 CoreMvcUI 的 Index.cshtml 页面代码如下图所示。



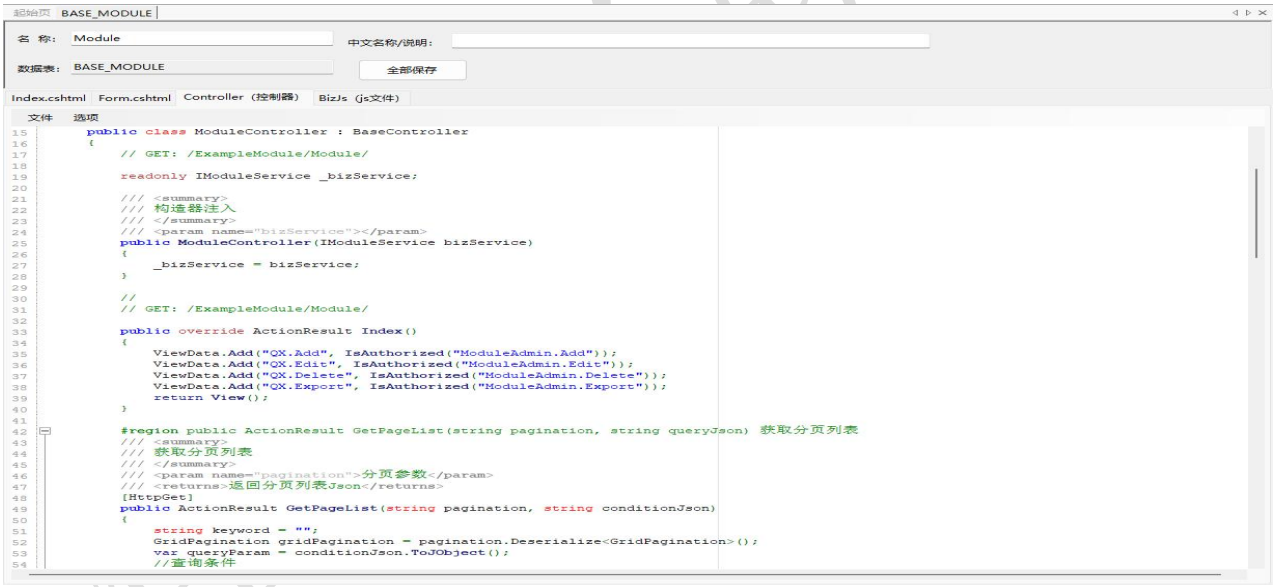
2.3.8.2. 编辑界面 Form.cshtml 代码

编辑代码 Form.cshtml 代码如下图所示。



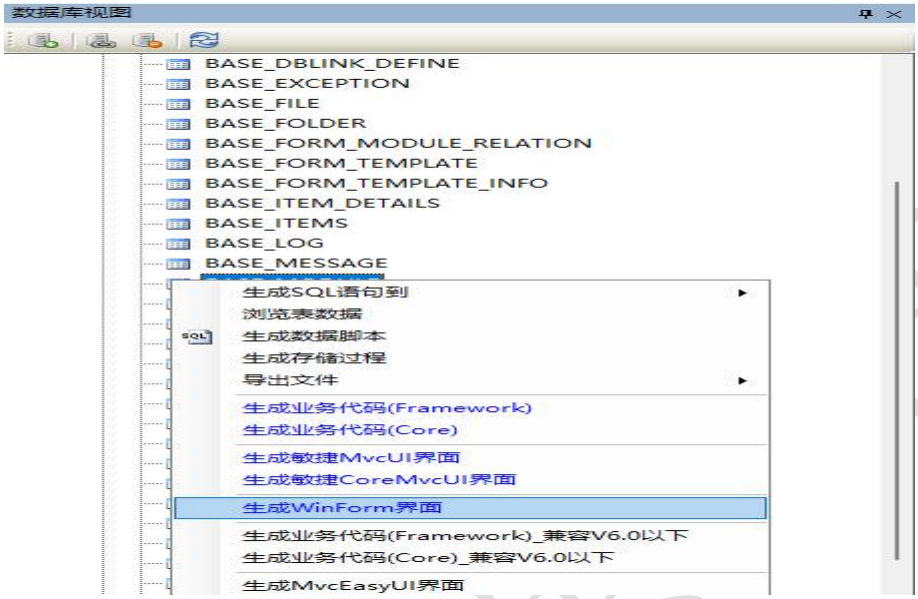
2.3.8.3. Controller 控制器

Controller 控制器代码如下图所示。

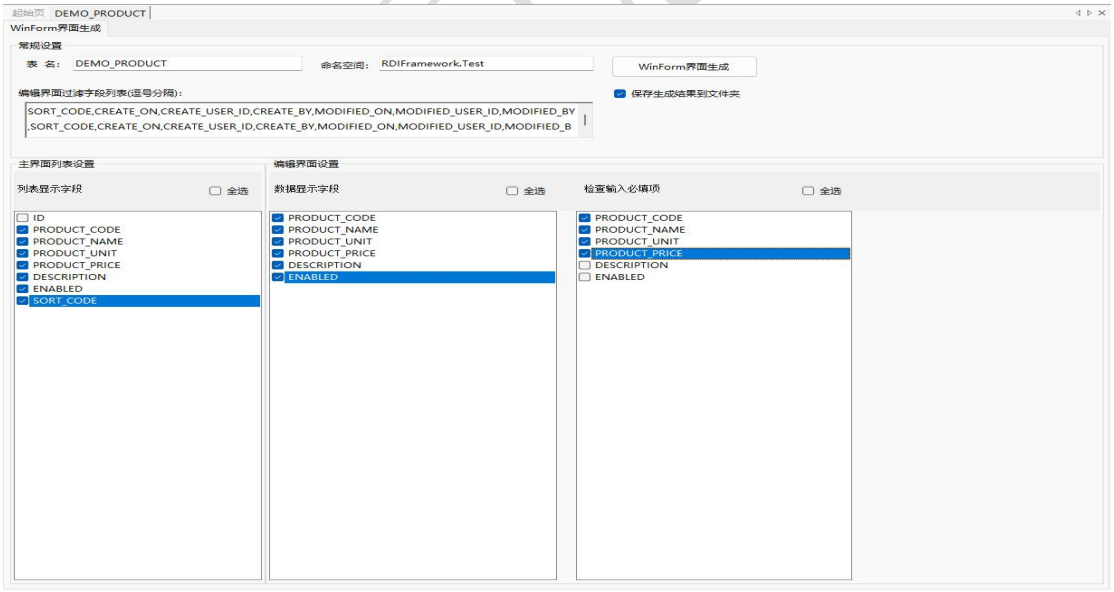


2.3.9. 生成 WinForm 界面

“生成 WinForm 界面”用于生成 WinForm 的 UI 界面代码，这样就可以大大缩小界面的开发时间，对生成的 UI 代码只需要要稍微做调整即可完成一个模块的快速开发，生成 WinForm 界面功能如下图所示。

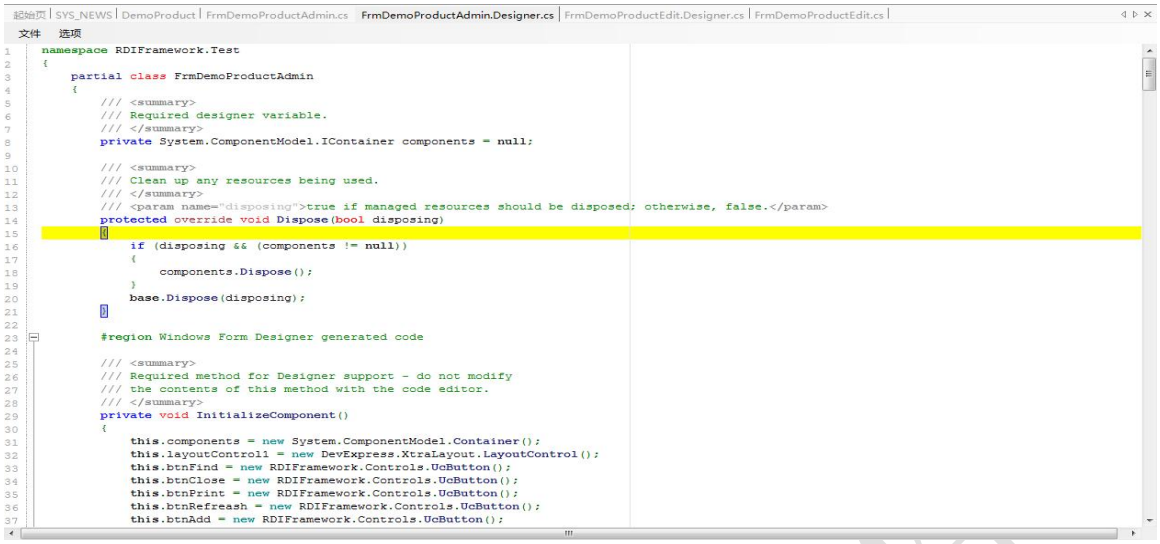


单击“生成 WinForm 界面”功能，即可打开当前表的 WinForm 界面生成设置窗口，如下图所示。



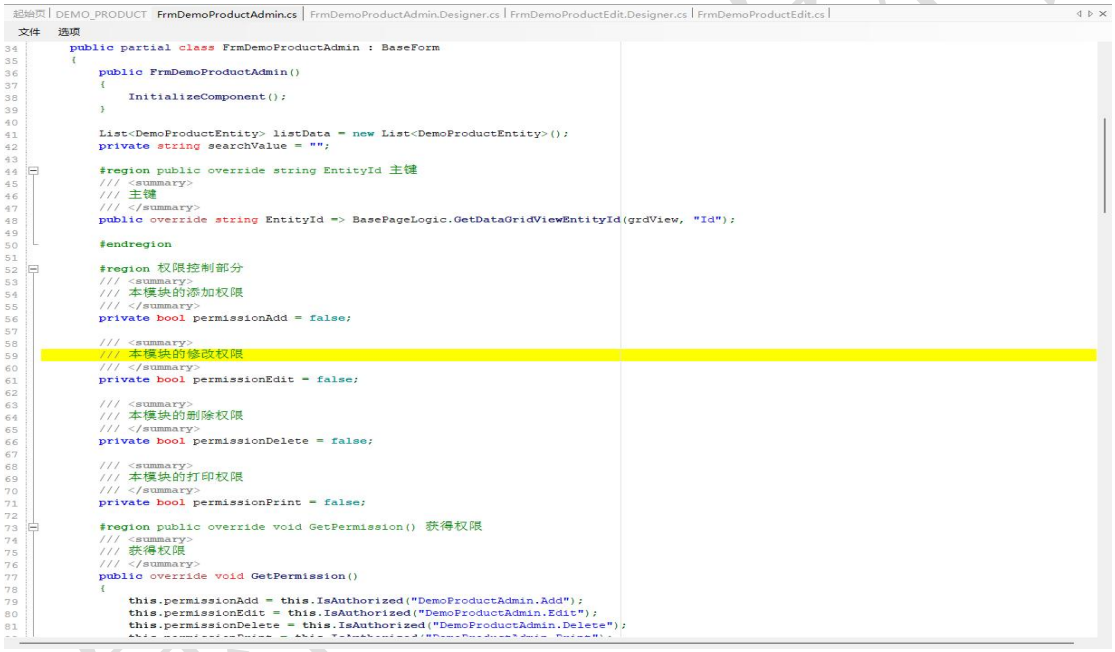
在上面的 WinForm 界面生成窗口，可以设置主界面列表需要展示的字段，编辑或新增界面需要处理的字段以及非空验证的字段，可以把生成的结果直接保存在文件夹中，也可以直接把生成的结果展示在 Tab 标签中。单击 WinForm 界面生成按钮，即可对所选表按设置进行生成，主要生成主界面与主界面的页面设计代码、编辑界面与编辑界面的页面设计代码，如下图所示。

1、主界面页面设计代码



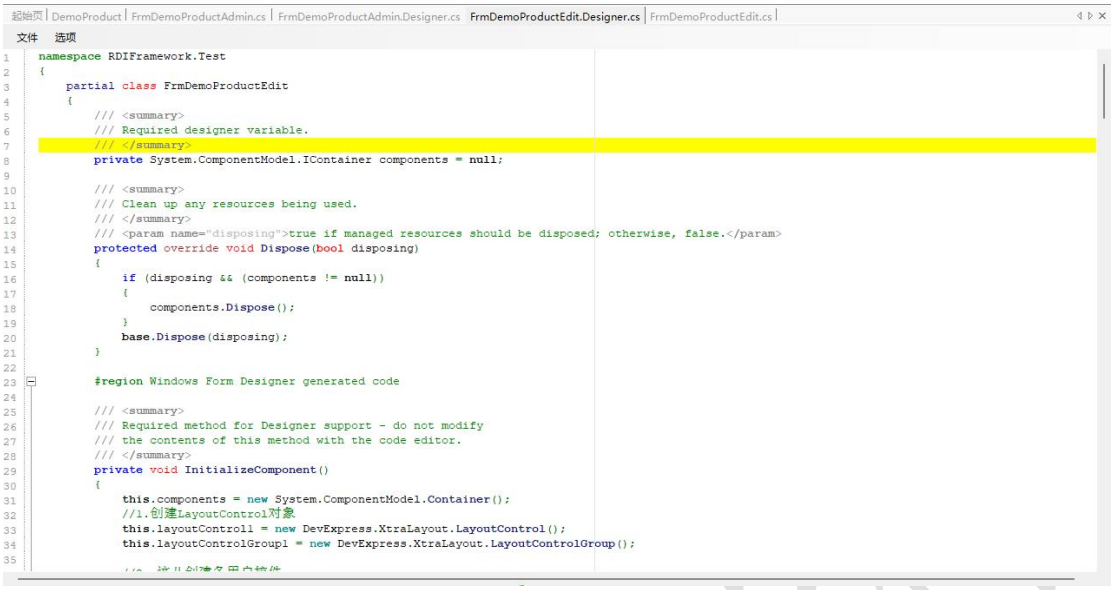
```
namespace RDIFramework.Test
{
    partial class FrmDemoProductAdmin
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.components = new System.ComponentModel.Container();
            this.layoutControl1 = new DevExpress.XtraLayout.LayoutControl();
            this.btnFind = new RDIFramework.Controls.UcButton();
            this.btnClose = new RDIFramework.Controls.UcButton();
            this.btnPrint = new RDIFramework.Controls.UcButton();
            this.btnRefreash = new RDIFramework.Controls.UcButton();
            this.btnAdd = new RDIFramework.Controls.UcButton();
```

2、主界面代码



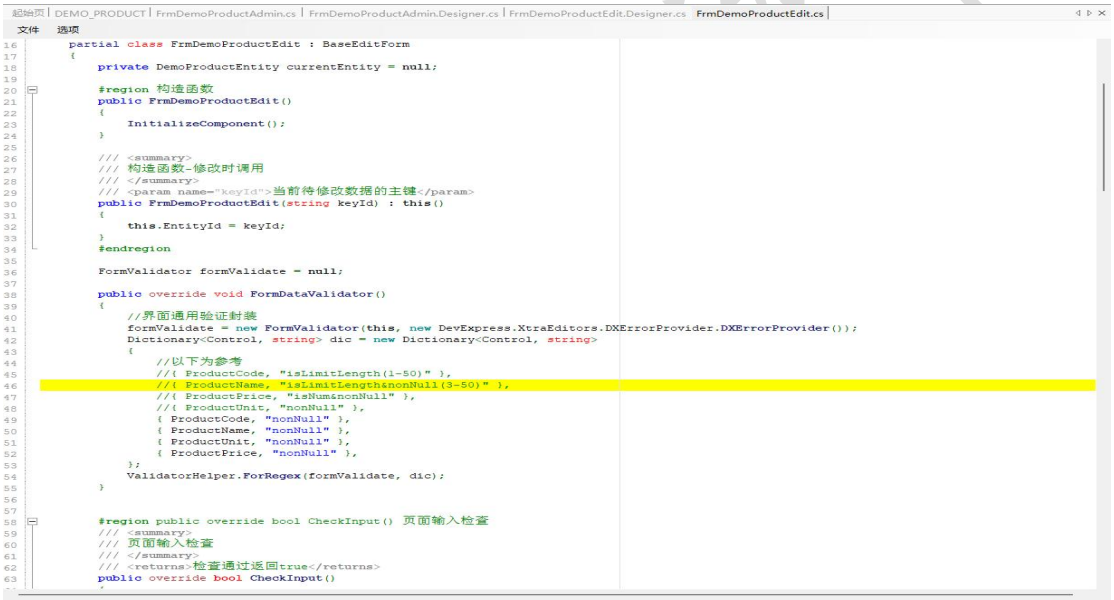
```
public partial class FrmDemoProductAdmin : BaseForm
{
    public FrmDemoProductAdmin()
    {
        InitializeComponent();
    }

    List<DemoProductEntity> listData = new List<DemoProductEntity>();
    private string searchValue = "";

    #region public override string EntityId 主键
    /// <summary>
    /// 主键
    /// </summary>
    public override string EntityId => BasePageLogic.GetDataGridViewEntityId(grdView, "Id");

    #endregion

    #region 权限控制部分
    /// <summary>
    /// 本模块的添加权限
    /// </summary>
    private bool permissionAdd = false;

    /// <summary>
    /// 本模块的修改权限
    /// </summary>
    private bool permissionEdit = false;

    /// <summary>
    /// 本模块的删除权限
    /// </summary>
    private bool permissionDelete = false;

    /// <summary>
    /// 本模块的打印权限
    /// </summary>
    private bool permissionPrint = false;

    #region public override void GetPermission() 获得权限
    /// <summary>
    /// 获得权限
    /// </summary>
    public override void GetPermission()
    {
        this.permissionAdd = this.IsAuthorized("DemoProductAdmin.Add");
        this.permissionEdit = this.IsAuthorized("DemoProductAdmin.Edit");
        this.permissionDelete = this.IsAuthorized("DemoProductAdmin.Delete");
```

3、编辑或新增界面设计代码



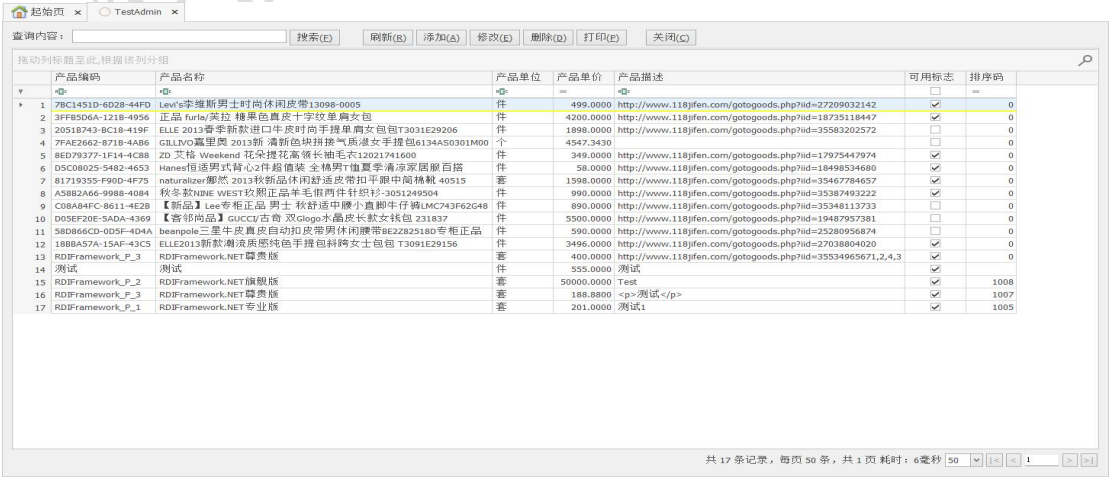
```
namespace RDIFramework.Test
{
    partial class FrmDemoProductEdit
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.components = new System.ComponentModel.Container();
            //1.创建LayoutControl对象
            this.layoutControl1 = new DevExpress.XtraLayout.LayoutControl();
            this.layoutControlGroup1 = new DevExpress.XtraLayout.LayoutControlGroup();
```

4、编辑或新增界面代码



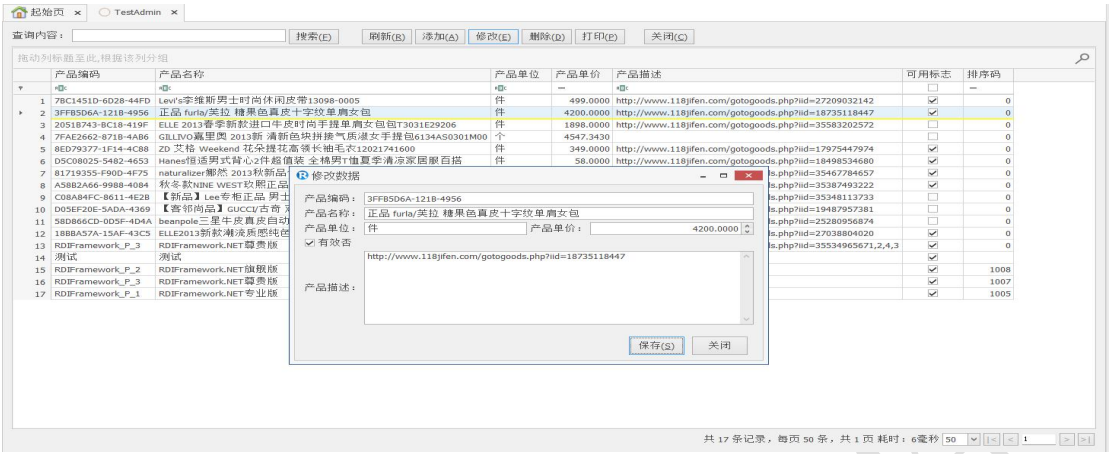
```
    partial class FrmDemoProductEdit : BaseEditForm
    {
        private DemoProductEntity currentEntity = null;

        #region 构造函数
        public FrmDemoProductEdit()
        {
            InitializeComponent();
        }

        /// <summary>
        /// 构造函数-修改时调用
        /// </summary>
        /// <param name="keyId">当前待修改数据的主键</param>
        public FrmDemoProductEdit(string keyId) : this()
        {
            this.EntityId = keyId;
        }
        #endregion

        FormValidator formValidate = null;

        public override void FormDataValidator()
        {
            //界面通用验证封装
            formValidate = new FormValidator(this, new DevExpress.XtraEditors.DXErrorProvider.DXErrorProvider());
            Dictionary<Control, string> dic = new Dictionary<Control, string>
            {
                //以下为参考
                //{ ProductCode, "isLimitLength(1-50)" },
                //{ ProductName, "isLimitLength&nonNull(3-50)" },
                //{ ProductPrice, "isNum&nonNull" },
                //{ ProductUnit, "nonNull" },
                { ProductCode, "nonNull" },
                { ProductName, "nonNull" },
                { ProductUnit, "nonNull" },
                { ProductPrice, "nonNull" },
            };
            ValidatorHelper.ForRegex(formValidate, dic);
        }

        #region public override bool CheckInput() 页面输入检查
        /// <summary>
        /// 页面输入检查
        /// </summary>
        /// <returns>检查通过返回true</returns>
        public override bool CheckInput()
```

对于生成的代码放在框架中打开测试页面，可以看到生成的主界面代码如下。



	产品编码	产品名称	产品单位	产品单价	产品描述	可用标志	排序码
1	7BC1451D-6D28-44FD	Levi's李维斯男士时尚休闲皮带13098-0005	件	499.0000	http://www.118jifen.com/gotogoods.php?iid=27209032142	☑	0
2	3FFB5D6A-1218-4956	正品 furla/芙拉 橘果色真皮十字纹单肩女包	件	4200.0000	http://www.118jifen.com/gotogoods.php?iid=18735118447	☑	0
3	20518743-8C18-419F	ELLE 2013春季新款进口牛皮时尚手提单肩女包包T3031E29206	件	1898.0000	http://www.118jifen.com/gotogoods.php?iid=35583202572	☐	0
4	7FAE2662-871B-4AB6	GILLIVO嘉里奥 2013新 清新色块拼接气质退女手提包6134AS0301M00	个	4547.3430		☐	0
5	8ED79377-1F14-4C88	ZD 艾格 Weekend 花朵提花高领长袖毛衣12021741600	件	349.0000	http://www.118jifen.com/gotogoods.php?iid=17975447974	☑	0
6	05C08025-5482-4653	Hanes恒适男式背心2件超值装 全棉男T恤夏季清凉家居服百搭	件	58.0000	http://www.118jifen.com/gotogoods.php?iid=18498534680	☑	0
7	81719355-F90D-4F75	naturalizer娜然 2013秋新品休闲舒适皮带扣平跟中筒棉靴 40515	套	1598.0000	http://www.118jifen.com/gotogoods.php?iid=35467784657	☑	0
8	A5882A66-9988-4084	秋冬款NINE WEST玖熙正品羊毛假两件针织衫-3051249504	件	990.0000	http://www.118jifen.com/gotogoods.php?iid=35387493222	☑	0
9	C08A84FC-8611-4E2B	【新品】Lee专柜正品 男士 秋舒适中腰小直脚牛仔裤LMC743F62G48	件	890.0000	http://www.118jifen.com/gotogoods.php?iid=35348113733	☐	0
10	D05EF20E-5ADA-4369	【喜邻尚品】GUCCI/古奇 双Glogo水晶皮长款女钱包 231837	件	5500.0000	http://www.118jifen.com/gotogoods.php?iid=19487957381	☐	0
11	58D866CD-0D5F-4D4A	beanpole三星牛皮真皮自动扣皮带男休闲腰带BE2ZB2518D专柜正品	件	590.0000	http://www.118jifen.com/gotogoods.php?iid=25280956874	☐	0
12	18B8A57A-15AF-43C5	ELLE2013新款潮流质感纯色手提包斜跨女士包包 T3091E29156	件	3496.0000	http://www.118jifen.com/gotogoods.php?iid=27038804020	☑	0
13	RDIFramework_P_3	RDIFramework.NET尊贵版	套	400.0000	http://www.118jifen.com/gotogoods.php?iid=35534965671,2,4,3	☑	0
14	测试	测试	件	555.0000	测试	☑	
15	RDIFramework_P_2	RDIFramework.NET旗舰版	套	50000.0000	Test	☑	1008
16	RDIFramework_P_3	RDIFramework.NET尊贵版	套	188.8800	测试	☑	1007
17	RDIFramework_P_1	RDIFramework.NET专业版	套	201.0000	测试1	☑	1005

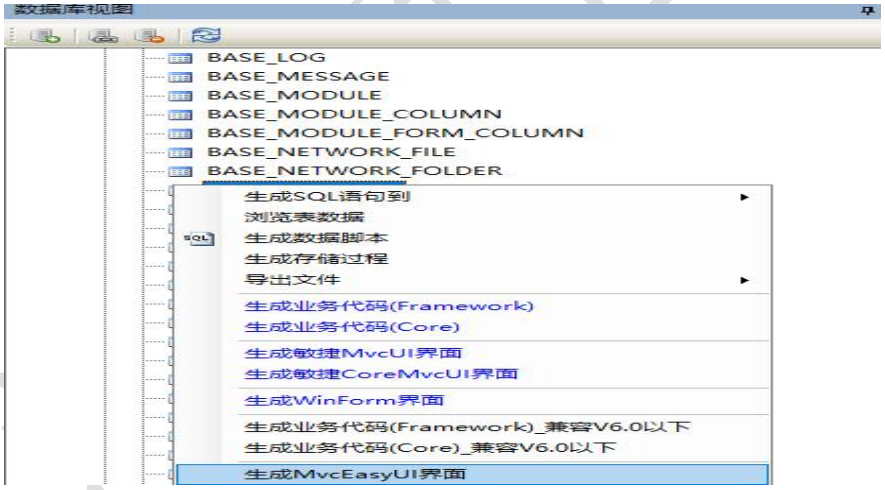
整个主界面已经集成了关键字搜索、刷新、添加、修改、删除、打印、分页、分组、表格内搜索等功能。单击添加或修改按钮即可打开编辑界面，如下图所示。



整个界面的生成如果不满意可以直接通过 VisualStudio 设计器直接拖动控件（使用了 layoutControl 布局，拖动非常方便）即可快速完成布局，非常之方便与高效。调试完成后即可在框架中定义好模块自动加载。

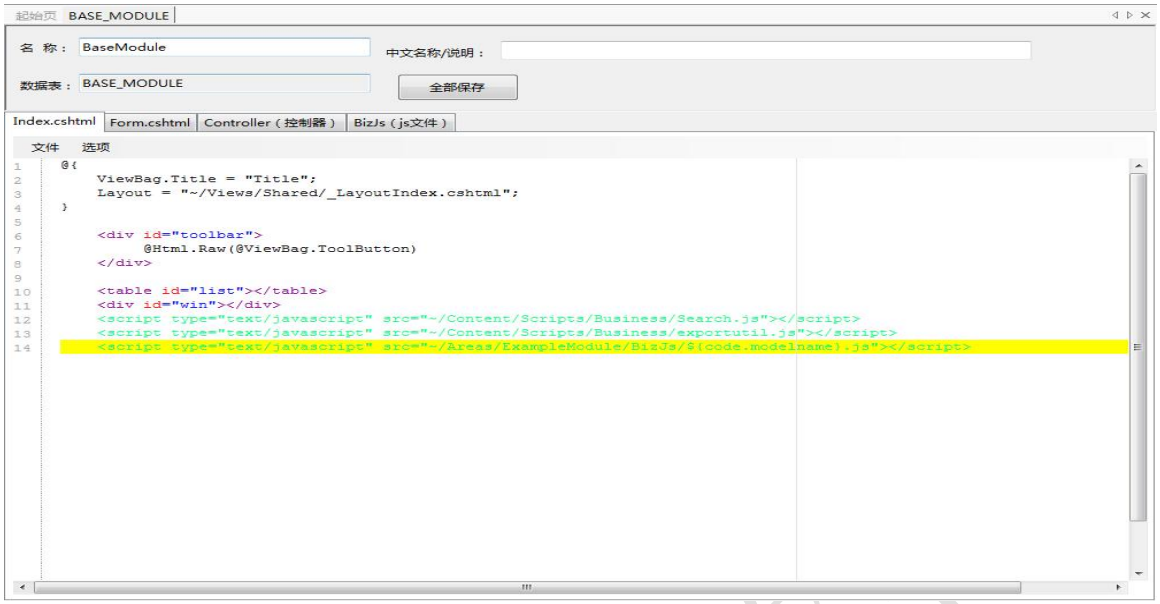
2.3.10. 生成 MvcEasyUI 界面

“生成 MvcEasyUI 界面”可以生成针对 MVC 的 EasyUI 风格的界面代码，这样就可以大大缩小界面的开发时间，对生成的 UI 代码做适量的修改即可。选择一个数据表，单击右键，选择“生成 MvcEasyUI 界面”功能，如下图所示。



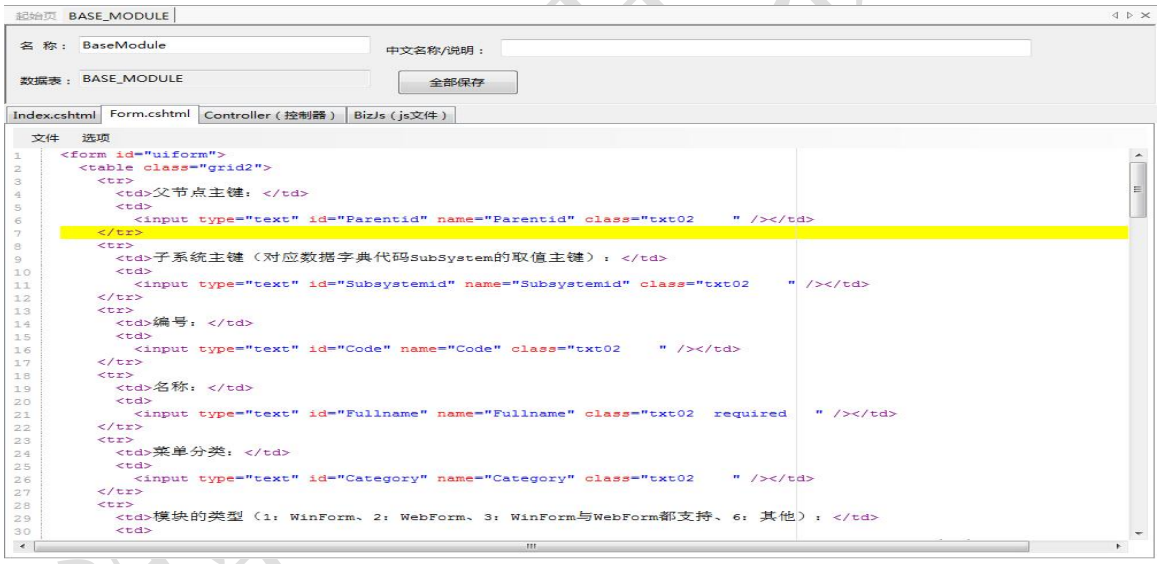
2.3.10.1. Index.cshtml 页面代码

MvcEasyUI 的 Index.cshtml 页面代码如下图所示。



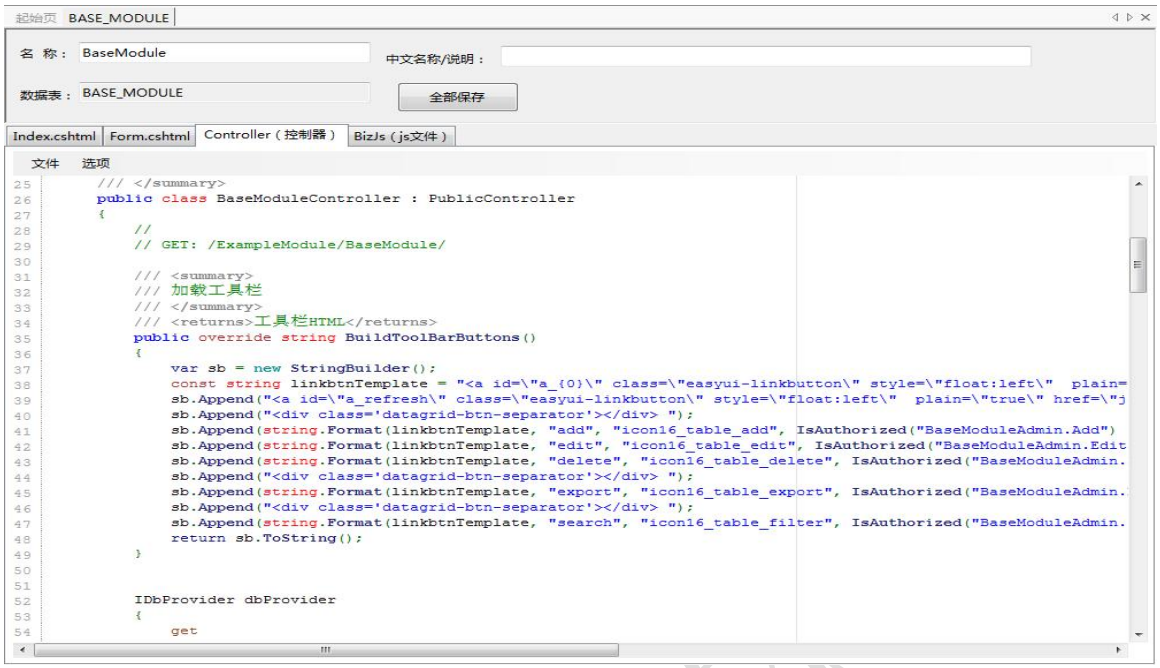
2.3.10.2. 编辑界面 Form.cshtml 代码

编辑代码 Form.cshtml 代码如下图所示。



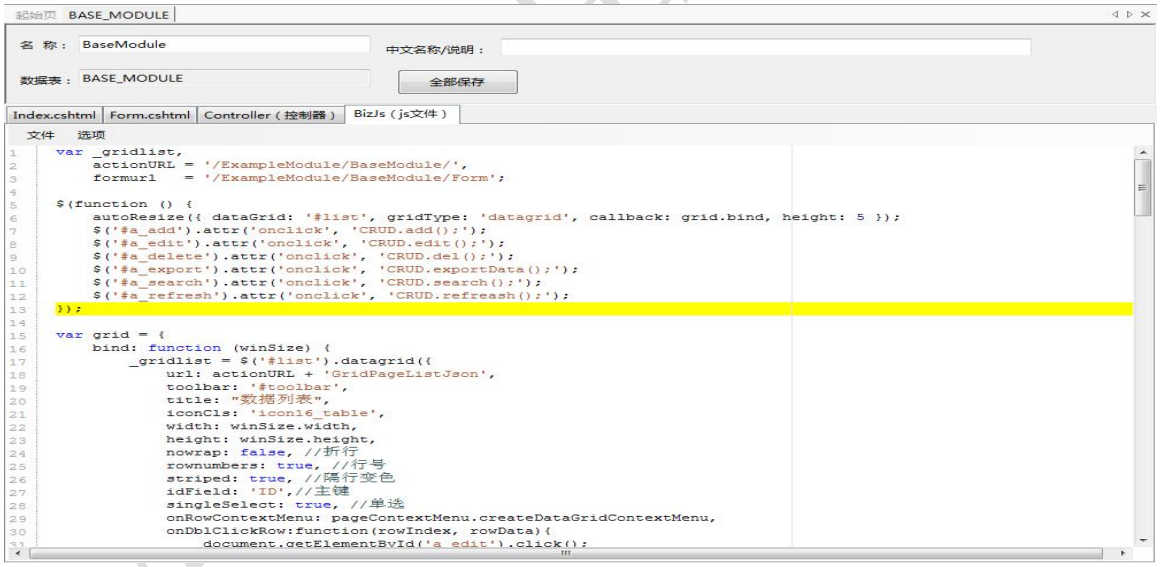
2.3.10.3. Controller 控制器

Controller 控制器代码如下图所示。



2.3.10.4. *js 页面代码*

Js 页面代码如下图所示。



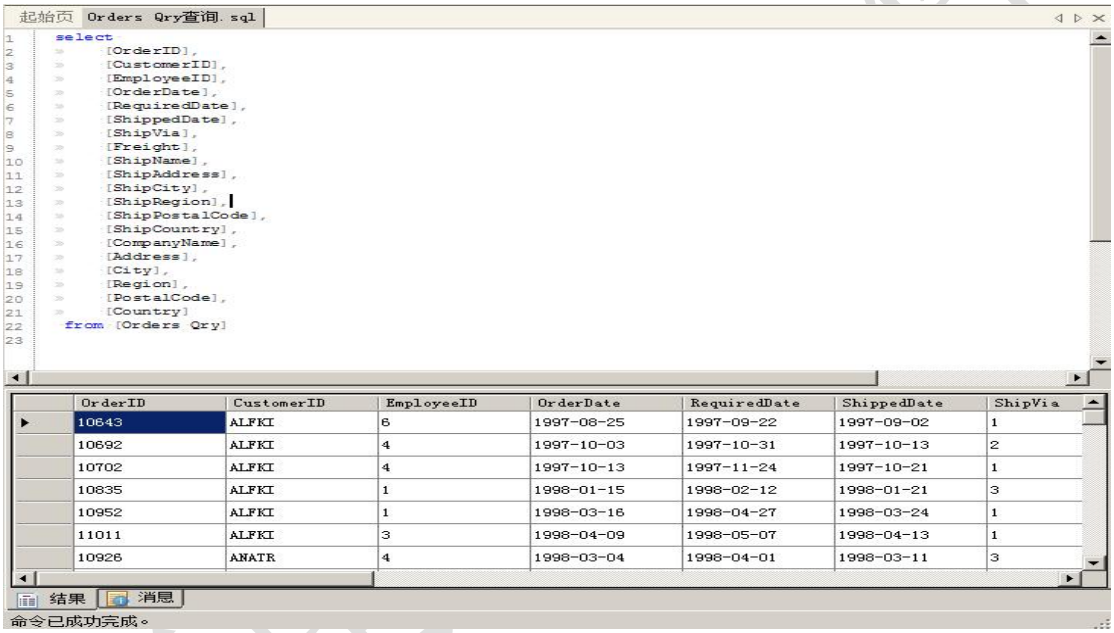
2.4. 视图管理

连接到一个已注册的服务器，我们可以对当前服务器所选“数据库”的视图进行相应的操作，主要包括：脚本生成（包括：SELECT 语句、ALTER 语句、DROP 语句）、对象定义、浏览表数据等。

2.4.1. 生成脚本

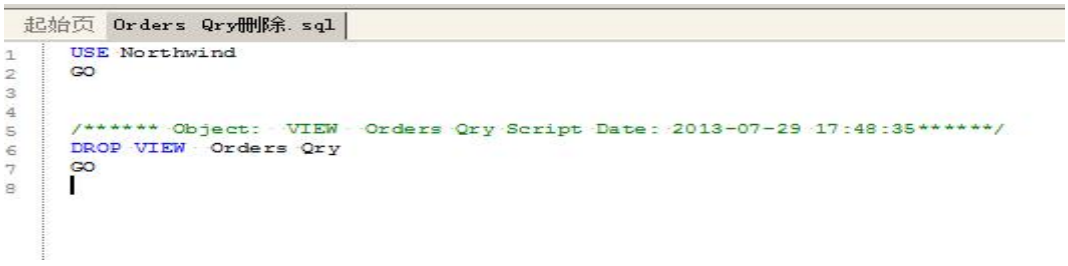
“脚本生成”可以生成当前所选视图的 SELECT 语句、ALTER 语句、DROP 语句。

1) 生成 SELECT 语句。



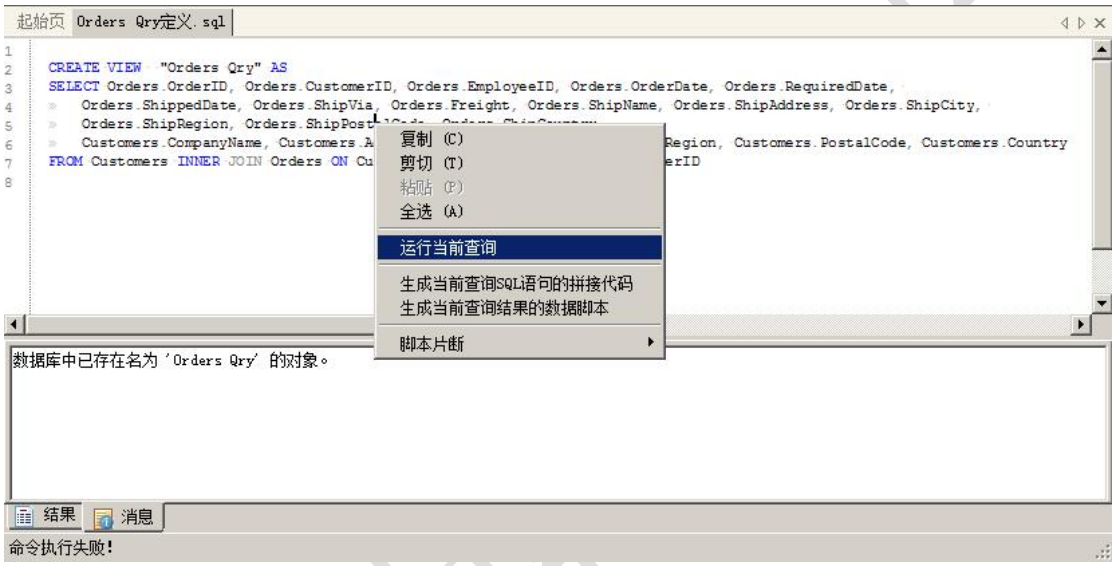
2) 生成 ALTER 语句。

3) 生成 DROP 语句。



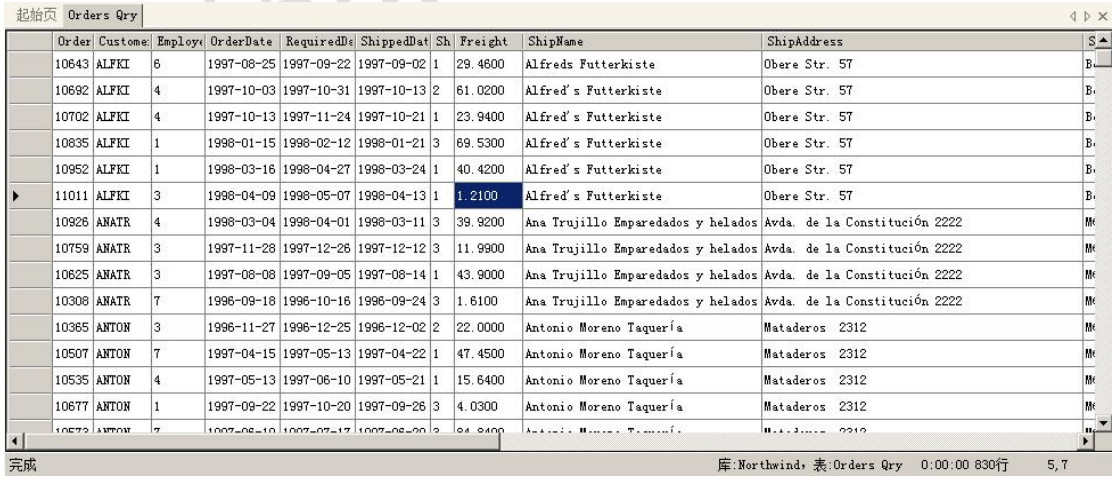
2.4.2. 对象定义

对象定义就是生成当前视图的创建脚本，如下图所示。



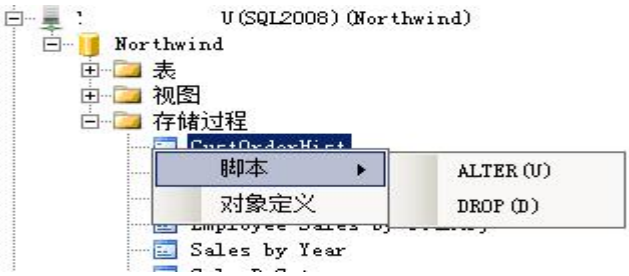
2.4.3. 浏览表数据

“浏览表数据”就是查看当前所选视图的数据，如下图所示。



2.5. 2.5 存储过程管理

连接到一个已注册的服务器，我们可以对当前服务器所选“数据库”的存储过程进行相应的操作，主要包括：脚本生成（包括： ALTER 语句、DROP 语句）、对象定义等。



2.5.1. 脚本生成

“脚本生成”可以生成当前所选存储过程的 ALTER 语句、DROP 语句。

1) 生成 ALTER 语句。

起始页 CustOrderHist修改.sql

```
ALTER PROCEDURE CustOrderHist @CustomerID nchar(5)
AS
SELECT ProductName, Total=SUM(Quantity)
FROM Products P, [Order Details] OD, Orders O, Customers C
WHERE C.CustomerID = @CustomerID
AND C.CustomerID = O.CustomerID AND O.OrderID = OD.OrderID AND OD.ProductID = P.ProductID
GROUP BY ProductName
```

2) 生成 DROP 语句。

起始页 CustOrderHist删除.sql

```
USE Northwind
GO


/****** Object:  PROCEDURE  CustOrderHist  Script Date: 2013-07-29 18:07:33******/
DROP PROCEDURE  CustOrderHist
GO
```

2.5.2. 对象定义

对象定义就是生成当前存储过程的创建脚本，如下图所示。

起始页 CustOrderHist定义.sql

```
CREATE PROCEDURE CustOrderHist @CustomerID nchar(5)
AS
SELECT ProductName, Total=SUM(Quantity)
FROM Products P, [Order Details] OD, Orders O, Customers C
WHERE C.CustomerID = @CustomerID
AND C.CustomerID = O.CustomerID AND O.OrderID = OD.OrderID AND OD.ProductID = P.ProductID
GROUP BY ProductName
```

3. PowerDesigner Objects 导航区

RDIFramework.NET 代码生成器最大的特点就是不仅可以通过连接到数据库来生成代码，还可以直接通过 PowerDesigner（下面简称 PD）设计源文件来生成代码，通过 PD 源文件来生成代码最大的好处就是不依赖于具体的数据库类型。我们设计数据库一般都是借助于 PD 工具进行设计，使用 PD 设计数据库有什么好处呢？PowerDesigner 是 Sybase 公司开发的数据库建模 CASE 工具，它是一种数据库开发环境专门提供数据库的需求分析、概念数据模型

CDM 设计、物理数据模型 PDM 设计以及数据库建表、建索引、建视图、建存储过程、建触发器等功能，同时还可做到模型数据共享，修改、设计、沟通方便。

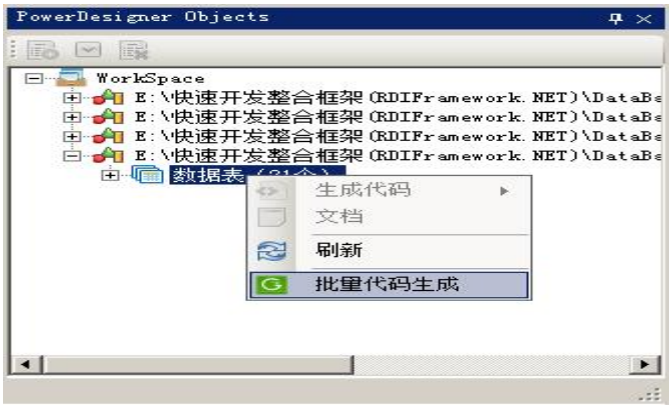
PowerDesigner Objects 导航区可对多个 PD 设计文件进行集中管理。PD 导航区如下图所示。

在上图在我们加载了 3 个 PD 设计源文件，加载成功后，我们就可以对加载的源文件进行代码的生成。在 PowerDesigner Objects 导航区，PD 源文件是以树型结构加载显示的，选择不同的树节点，会有不同的功能。树节点类型主要分为以下几类：

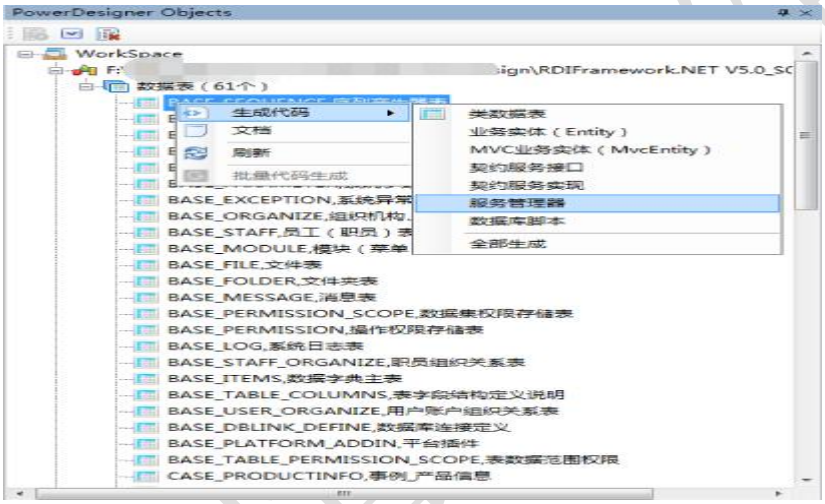
1) 工作区域（WorkSpace）树节点：在当前节点，我们可以添加 PD 设计文档到 WorkSpace 中。

2) PD 源文件树节点：在当前节点，我们可以加载当前 PD 设计文件、展开当前 PD 设计文件、移除当前 PD 设计文件。

3) 数据表父树节点：在当前节点，我们可以进行批量代码生成（生成所有表的全部代码），刷新当前数据表。



4)　具体的表节点：选择具体的表节点，我们可以生成代码（类数据表、业务实体、契约服务接口、契约服务、服务管理器、全部生成）、生成当前表的数据库文档、刷新等。



3.1. PD 设计文件管理

PD 设计文件管理包括添加 PD 设计文件到 PD 管理工作区、移除已添加到 PD 管理工作区中的 PD 设计文件、加载 PD 设计文件、刷新 PD 设计文件等。

3.1.1. 添加 PD 设计文件

要通过 PD 设计文件进行代码生成，必须先把 PD 设计文件添加到 PD 管理工作区中，添加 PD 设计文件，需要鼠标选中“WorkSpace”树节点，然后单击 PowerDesigner Objects 导航区的工具栏上的“增加 PowerDesigner 设计文件”按钮，图标为：，在弹出的文件选择对话框中，选择一个 PD 设计文件，即可把 PD 设计文件加载到工作区中。

3.1.2. 移除 PD 设计文件

对于已成功添加的 PD 设计文件，如果不再需要，我们可以把它从工作区中移除，方法就是选择需要移除的“PD 设计文件”后，单击 PowerDesigner Objects 导航区的工具栏上的“移除 PowerDesigner 设计文件”按钮，图标为：，在弹出的询问对话框，如下图所示，选择“是”即可从工作区中移除 PD 设计文件。

3.1.3. 展开 PD 设计文件

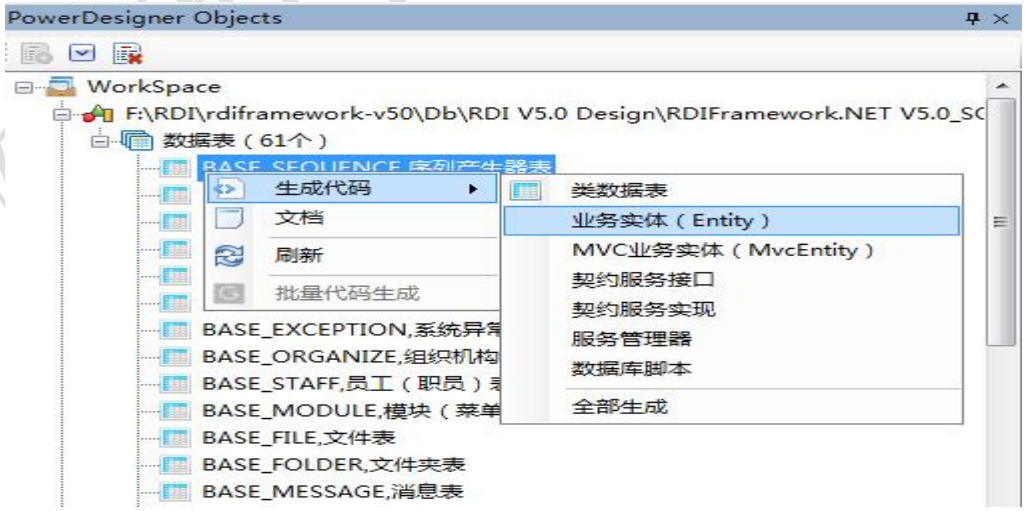
对于已经添加的 PD 设计文件，我们可以展开（如果是初次，就是加载当前设计文件的所有表）它，对应图标为：。

3.2. 代码生成

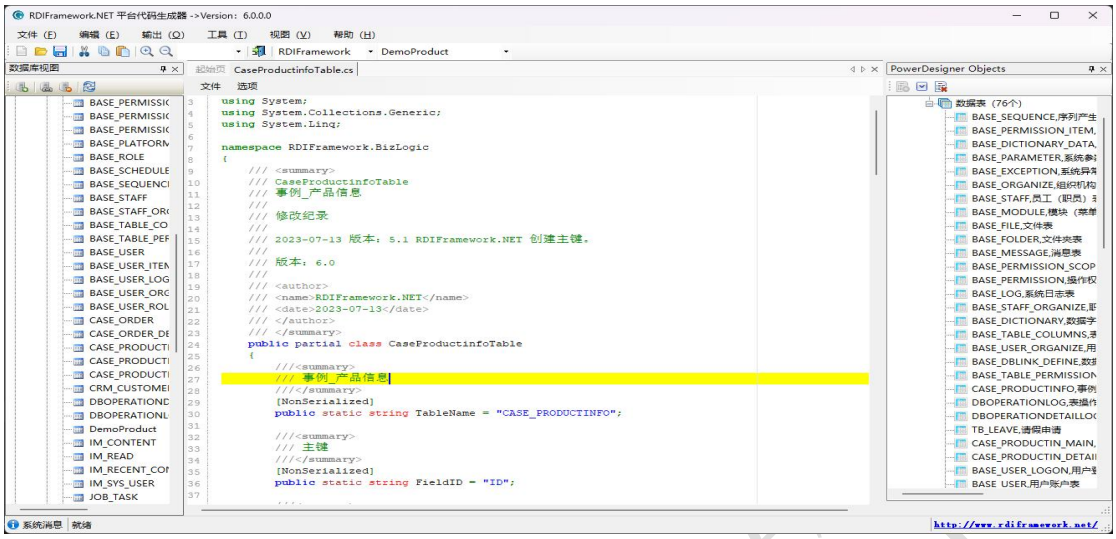
下面我们对如何通过“PD 设计文件”进行代码生成进行讲解，主要包括以下内容：

1) 生成类数据表。
2) 生成业务实体（Entity）。
3) 生成契约服务接口。
4) 生成契约服务。
5) 生成服务管理器。
6) 全部生成。
7) 生成表设计文档。

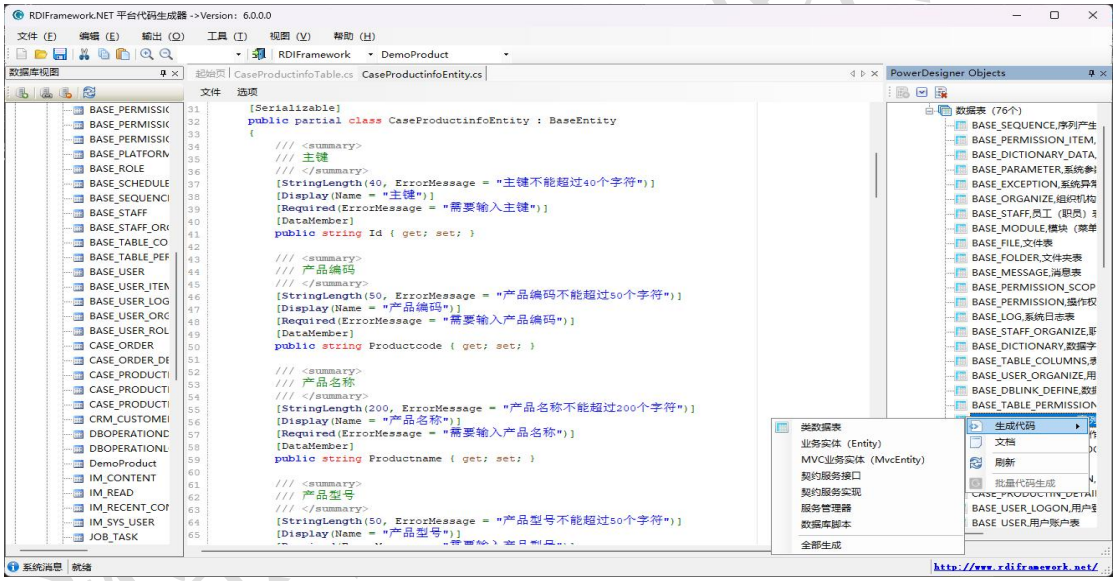
具体的代码说明与根据数据库生成的代码是一至的，具体概念就不再说明了，生成的代码公共部分的设计同样依赖于“项目属性设置”，具体可参考 2.3.6.1“项目属性设置”一节。



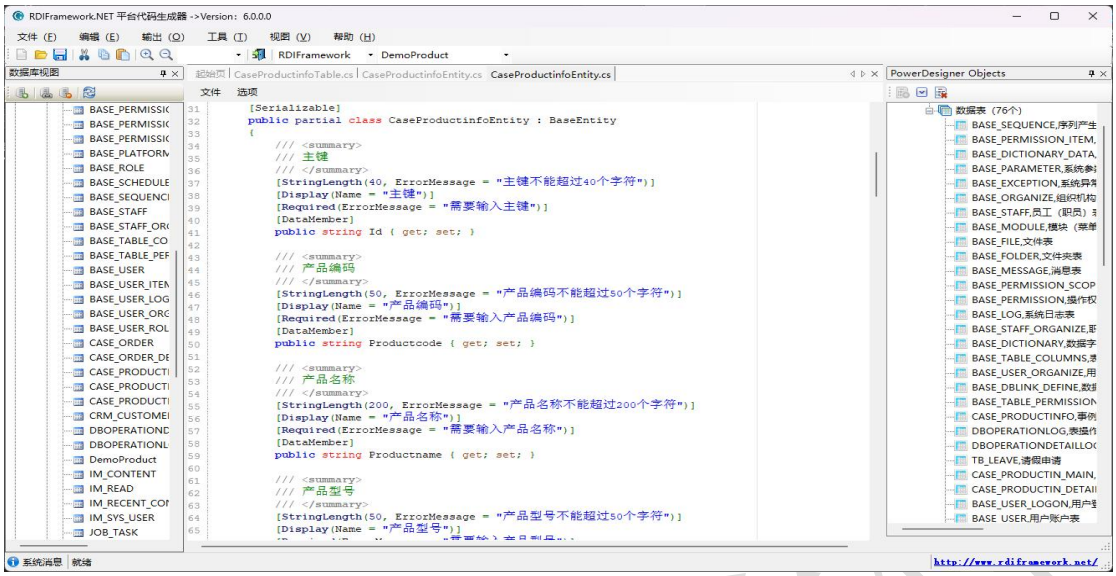
3.3.1. 生成类数据表



3.3.2. 生成业务实体（Entity）



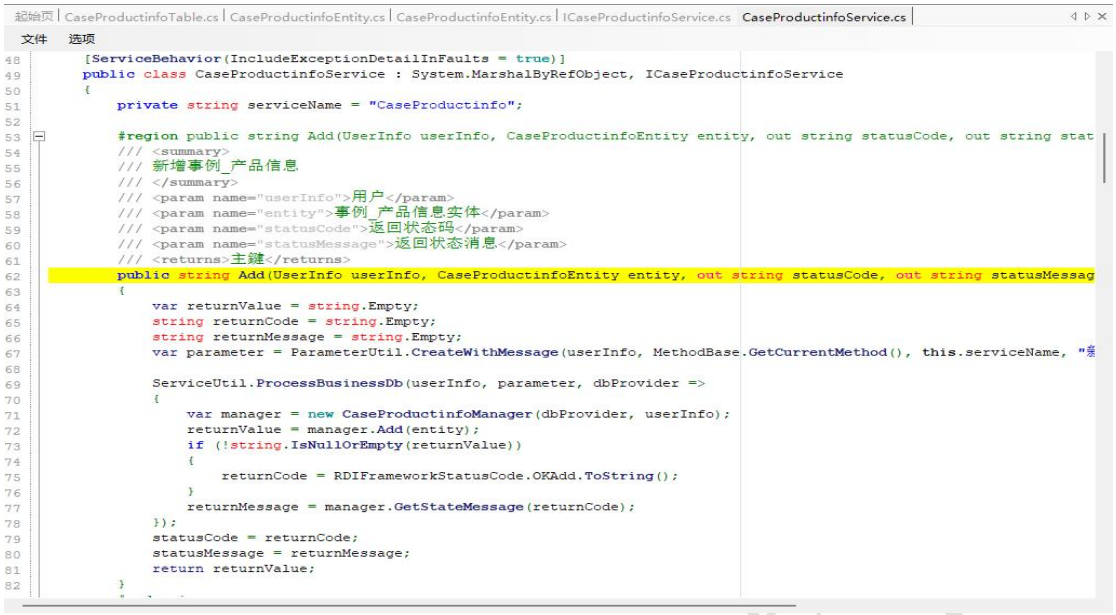
3.3.3. 生成 MVC 业务实体（MVCEntity）



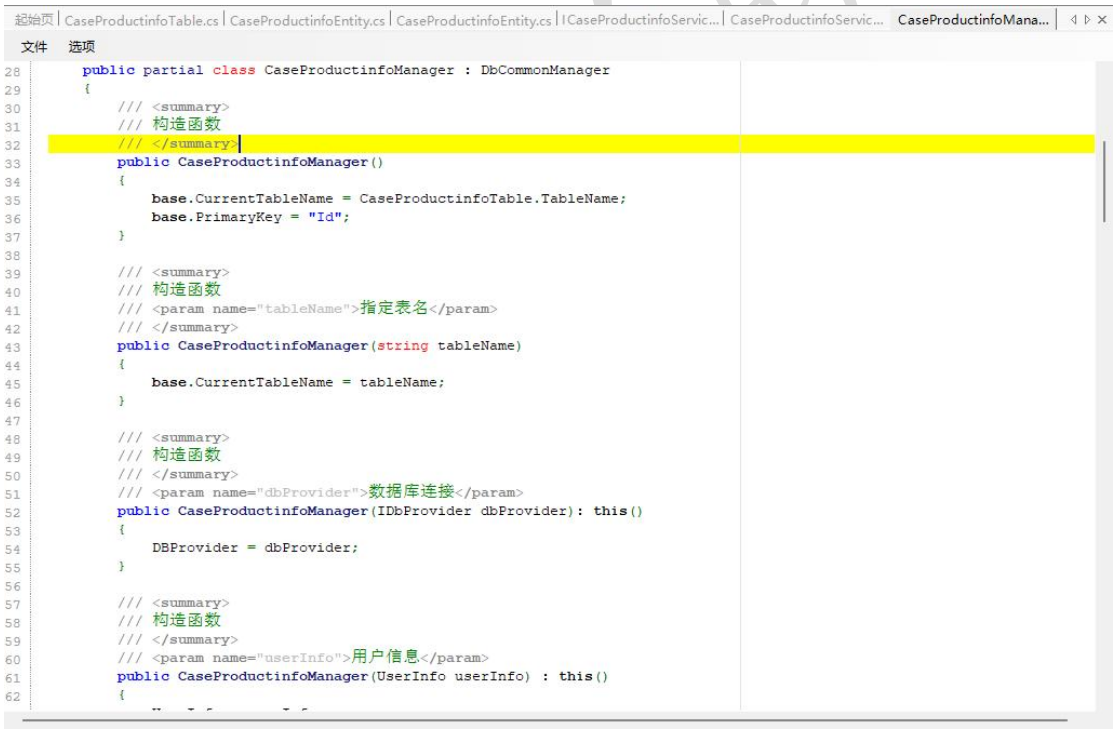
3.3.4. 生成契约服务接口

```
起始页 | CaseProductinfoTable.cs | CaseProductinfoEntity.cs | CaseProductinfoEntity.cs | ICaseProductinfoService.cs
文件  选项
45      [ServiceContract]
46      public interface ICaseProductinfoService : IDependency
47      {
48          /// <summary>
49          /// 新增事例_产品信息
50          /// </summary>
51          /// <param name="userInfo">用户</param>
52          /// <param name="entity">实体</param>
53          /// <param name="statusCode">返回状态</param>
54          /// <param name="statusMessage">返回状态信息</param>
55          /// <returns>主键</returns>
56          [OperationContract]
57          string Add(UserInfo userInfo, CaseProductinfoEntity entity, out string statusCode, out st
58
59          /// <summary>
60          /// 更新事例_产品信息
61          /// </summary>
62          /// <param name="userInfo">用户</param>
63          /// <param name="entity">事例_产品信息实体</param>
64          /// <param name="statusCode">返回状态码</param>
65          /// <param name="statusMessage">返回状态信息</param>
66          /// <returns>影响行数</returns>
67          [OperationContract]
68          int Update(UserInfo userInfo, CaseProductinfoEntity entity, out string statusCode, out st
69
70          /// <summary>
71          /// 获取事例_产品信息列表
72          /// </summary>
73          /// <param name="userInfo">用户</param>
74          /// <returns>数据表</returns>
75          [OperationContract]
76          DataTable GetDT(UserInfo userInfo);
77
78          /// <summary>
79          /// 分页获取事例_产品信息列表
```

3.3.5. 生成契约服务



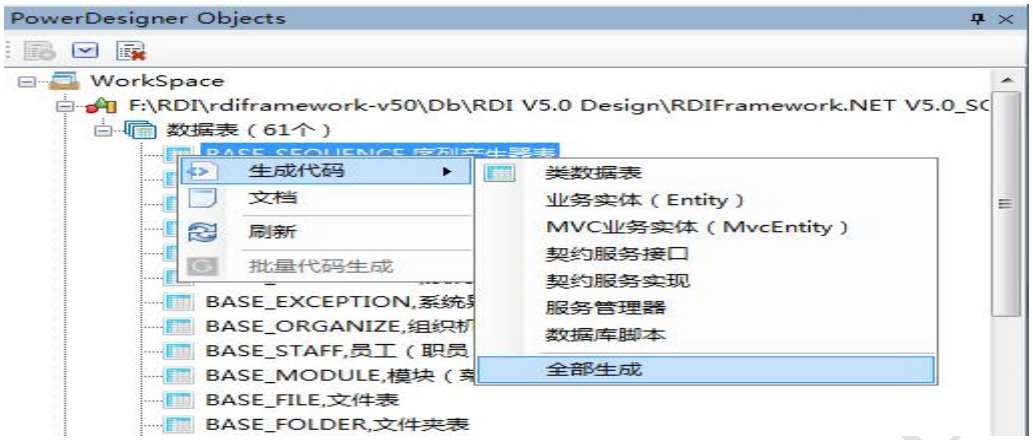
3.3.6. 生成服务管理器



3.3.7. 生成全部代码

生成当前代码就是把当前所选设计表的全部代码（类数据表、业务实体、服务接口、服务实现、服务管理器）生成到本地目录中（保存的位置可通过“项目属性设置”窗口的“代码输出目录”进行设置）。

单击“全部生成”，如下图所示。



即可把当前表的所有代码全部生成到指定目录中，生成的代码如下：

3.3.8. 生成表设计文档

表设计文档，就是当前所选表的数据库设计文档，如下图所示。

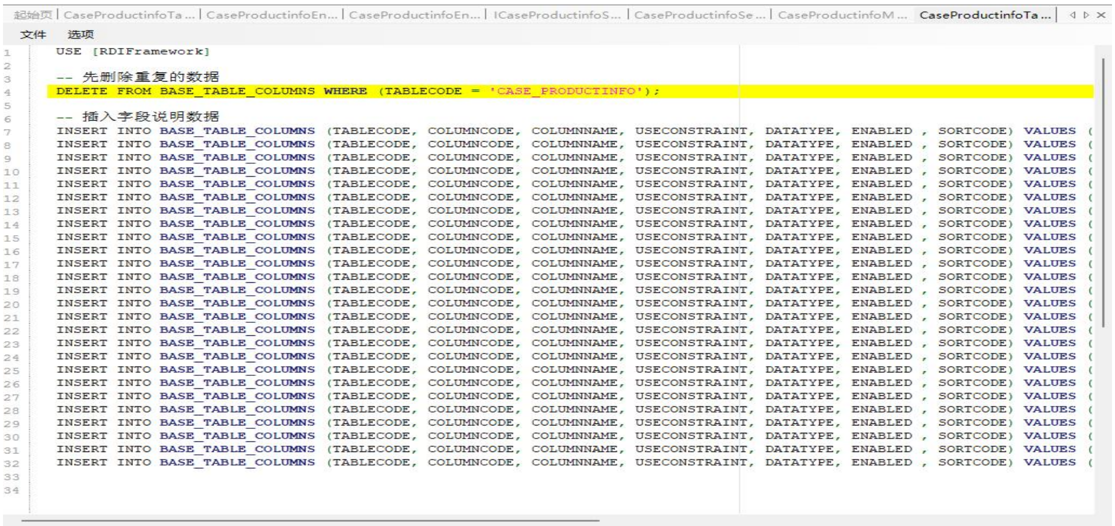
CUSTOMER（客户信息）

数据库名：
表名：CUSTOMER（客户信息）

序号	列代码	列名称	数据类型	长度	小数位	标识	主键	允许空	默认值	描述
1	ID	主键	Int			Y	Y	N		主键
2	CODE	客户编号	Nvarchar(50)	50		N	N	N		客户编号
3	FULLNAME	客户名称	Nvarchar(200)	200		N	N	Y		客户全称
4	POSTCODE	邮政编码	Nvarchar(6)	6		N	N	Y		
5	COUNTRY	国家	Nvarchar(50)	50		N	N	Y		
6	PROVINCE	省份	Nvarchar(50)	50		N	N	Y		
7	CITY	城市	Nvarchar(50)	50		N	N	Y		
8	REGISTERDATE	登记日期	date			N	N	Y		
9	NEXTCONTACTDATE	下次联系时间	date			N	N	Y		
10	LASTCONTACTDATE	最后联系时间	date			N	N	Y		
11	ADDRESS	公司地址	Nvarchar(500)	500		N	N	Y		
12	ONLINEADDRESS	公司主页	Nvarchar(500)	500		N	N	Y		
13	TURNOVER	营业额	bigint			N	N	Y		营业额（单位：人民币元）。
14	LEVEL	客户等级	nvarchar(50	50		N	N	Y		客户等级
15	STATUS	客户状态	nvarchar(50)	50		N	N	Y		
16	TYPE	客户类型	nvarchar(50)	50		N	N	N	3	信用度（1至5），默认为3。
17	SOURCE	客户来源	nvarchar(50)	50		N	N	Y		
18	TRADE	行业类型	nvarchar(50)	50		N	N	Y		
19	AREA	区域	nvarchar(50)	50		N	N	Y		
20	HONOR	信用等级	NVarchar(50)	50		N	N	Y		公司邮编
21	REMARK	备注	Nvarchar(50)	50		N	N	Y		公司电话

3.3.9. 生成数据库脚本

数据库脚本主要用于权限控制表中使用，我们在以通过代码生成器生成待控制的权限控制表脚本数据，如下图所示。

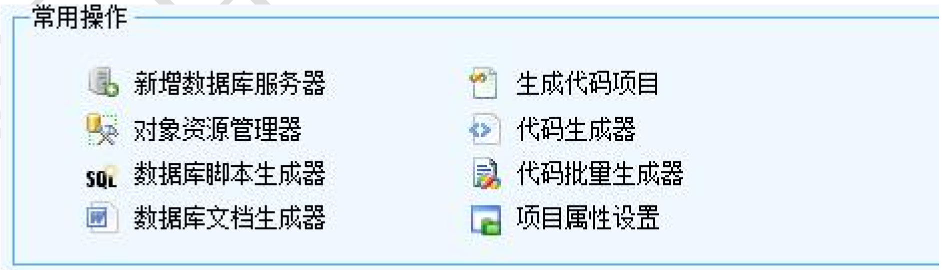


4. 代码批量生成

4.1. 基于数据库的代码批量生成

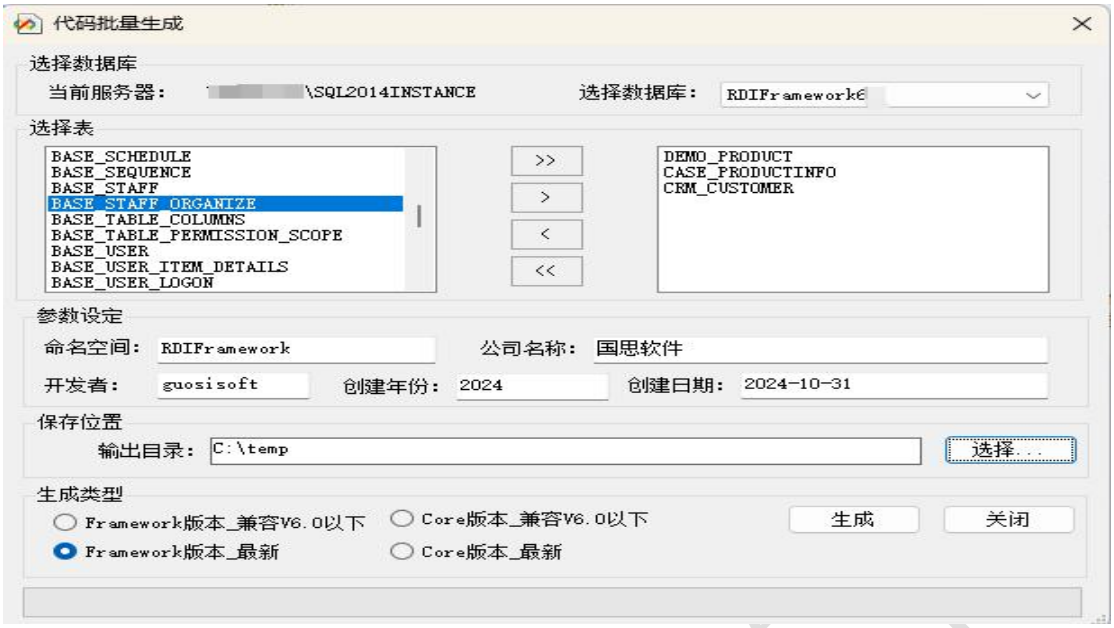
基于数据库的代码批量生成，是快速生成项目业务逻辑的最佳方式，对生成的代码只需要稍加修改，即可完成业务逻辑部分，大大提高开发的效率，节省开发时间与开发成本。基于数据库的代码批量生成，可以选择指定数据库下特定的表进行业务逻辑代码的生成。

要基于数据库进行批量代码生成，可以利用 RDIFramework.Net 代码生成器“起始页”页面上的常用操作中的“代码批量生成器”功能按钮来完成，如下图所示。



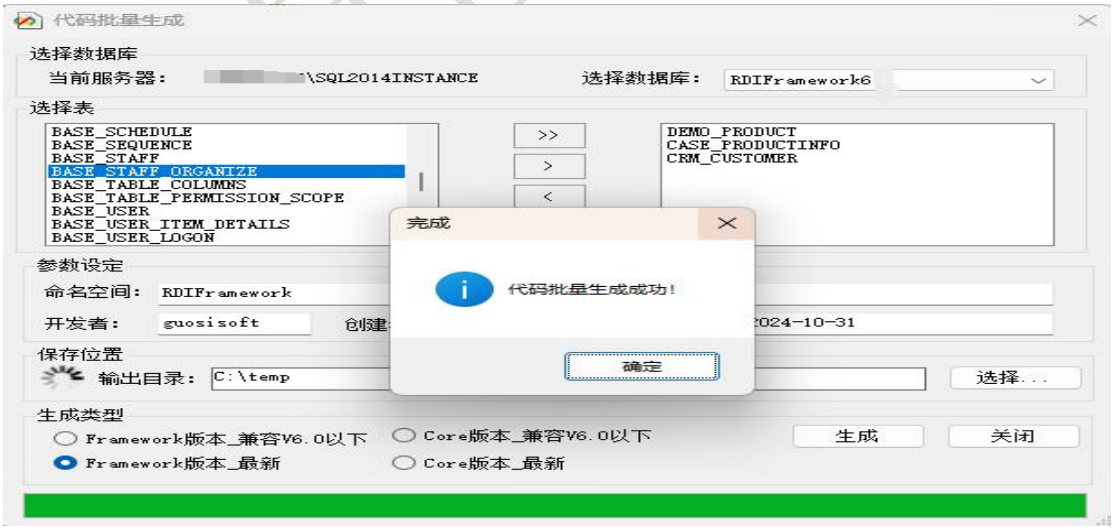
选择“代码批量生成器”，打开“代码批量生成”窗口，如下图所示。

Tips:如果提示“没有可用的数据库连接，请先连接数据库服务器”，则可在数据库视图导航区域，选择一个数据库服务器连接，再重新链接即可。

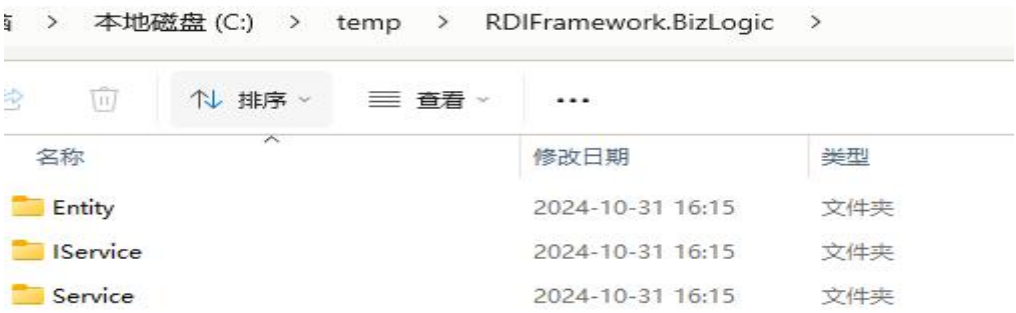


在上图中，显示了当前所选的服务器连接，针对所选“服务器连接”，在数据库列表框中列出了当前连接下的可用数据库，我们可以选择我们需要生成代码的数据库，会在选择表区域列出当前所选数据表的所有数据表，我们可以选择需要生成业务代码的数据表至右侧列表中，再设置项目的相关参数，如：命名空间、公司名称、开发者、创建年份、创建日期等，这些参数已经通过“项目属性”的设置进行了默认选择，当然你也可以在这我进行修改。在保存位置设置区域，“输出目录”就是设置代码批量生成后保存的文件目录，生成类型可以选择生成支持.NET Framework 或.NET Core 版本的代码。

这一切都设置就绪后，单击“导出”按钮，即可对当前数据库所选表批量生成代码，如下图所示。



批量生成成功后，我们可以打开代码保存的目录“C:\Temp”，查看生成的代码，如下图所示。



可以看到，我们的代码批量生成成功了。

4.2. 数据库设计文档生成

“数据库设计文档生成”是快速生成当前数据表所选数据表的数据库设计文档，一键生成，相当方便，对项目文档中自动生成数据库设计文档相当有帮助。在 RDIFramework.NET 代码生成器中，可以通过起始页的常用操作中的“数据库文档生成器”来生成数据库的设计文档，如下图所示。

单击“数据库文档生成器”功能按钮，打开“生成数据库设计文档”窗口，如下图所示。

在上图中，我们可以选择当前所选连接服务器下的数据库列表，选择一个数据库后，在选择表区域的左侧列出来当前数据库的所有可用表，我们可以选择需要生成的数据表至右侧，也可以选择全部数据表。选择好待生成的数据表后，我们还需要设置数据库设计文档输

出的格式，现在支持两种格式的生成，一种是生成 Word 文档格式，一种是生成 HTML 格式（支持三种风格）。

1) 生成 Word 格式

选择生成 Word 格式，需要本机安装 Office 软件方可，单击“生成”按钮，等待片刻（会有滚动条提示），即可完成 Word 格式的数据库设计文档的生成，生成的效果如下图所示。

数据库名：CRM1.3

表名：CUSTOMER

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	说明
1	ID	int	4	0	是	是	否		
2	CODE	nvarchar	50	0			否		
3	FULLNAME	nvarchar	200	0			是		
4	POSTCODE	nvarchar	6	0			是		
5	COUNTRY	nvarchar	50	0			是		
6	PROVINCE	nvarchar	50	0			是		
7	CITY	nvarchar	50	0			是		
8	REGISTERDATE	date	3	0			是		
9	NEXTCONTACTDATE	date	3	0			是		
10	LASTCONTACTDATE	date	3	0			是		
11	ADDRESS	nvarchar	500	0			是		
12	ONLINEADDRESS	nvarchar	500	0			是		
13	TURNOVER	bigint	8	0			是		
14	LEVEL	nvarchar	50	0			是		
15	STATUS	nvarchar	50	0			是		
16	TYPE	nvarchar	50	0			否	('3')	

2) 生成 HTML 格式

同样，数据库设计文档还可以输出为 HTML 格式，HTML 格式提供了三种风格，如下图所示，你可以选择一种适合你的风格进行生成。

风格一生成效果图如下：

数据库名：CRM1.3

表名：CUSTOMER

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	说明
1	ID	int	4	0	是	是	否		
2	CODE	nvarchar	50	0			否		
3	FULLNAME	nvarchar	200	0			是		
4	POSTCODE	nvarchar	6	0			是		
5	COUNTRY	nvarchar	50	0			是		
6	PROVINCE	nvarchar	50	0			是		
7	CITY	nvarchar	50	0			是		
8	REGISTERDATE	date	3	0			是		
9	NEXTCONTACTDATE	date	3	0			是		
10	LASTCONTACTDATE	date	3	0			是		
11	ADDRESS	nvarchar	500	0			是		
12	ONLINEADDRESS	nvarchar	500	0			是		
13	TURNOVER	bigint	8	0			是		
14	LEVEL	nvarchar	50	0			是		
15	STATUS	nvarchar	50	0			是		
16	TYPE	nvarchar	50	0			否	('3')	

风格二生成效果图如下：

数据库名：CRM1.3

表名：CUSTOMER

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	说明
1	ID	int	4	0	是	是	否		
2	CODE	nvarchar	50	0			否		
3	FULLNAME	nvarchar	200	0			是		
4	POSTCODE	nvarchar	6	0			是		
5	COUNTRY	nvarchar	50	0			是		
6	PROVINCE	nvarchar	50	0			是		
7	CITY	nvarchar	50	0			是		
8	REGISTERDATE	date	3	0			是		
9	NEXTCONTACTDATE	date	3	0			是		
10	LASTCONTACTDATE	date	3	0			是		
11	ADDRESS	nvarchar	500	0			是		
12	ONLINEADDRESS	nvarchar	500	0			是		
13	TURNOVER	bigint	8	0			是		
14	LEVEL	nvarchar	50	0			是		
15	STATUS	nvarchar	50	0			是		
16	TYPE	nvarchar	50	0			否	('3')	

风格三生成效果图如下：

数据库名：CRM1.3

表名：CUSTOMER

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	说明
1	ID	int	4	0	是	是	否		
2	CODE	nvarchar	50	0			否		
3	FULLNAME	nvarchar	200	0			是		
4	POSTCODE	nvarchar	6	0			是		
5	COUNTRY	nvarchar	50	0			是		
6	PROVINCE	nvarchar	50	0			是		
7	CITY	nvarchar	50	0			是		
8	REGISTERDATE	date	3	0			是		
9	NEXTCONTACTDATE	date	3	0			是		
10	LASTCONTACTDATE	date	3	0			是		
11	ADDRESS	nvarchar	500	0			是		
12	ONLINEADDRESS	nvarchar	500	0			是		
13	TURNOVER	bigint	8	0			是		
14	LEVEL	nvarchar	50	0			是		
15	STATUS	nvarchar	50	0			是		
16	TYPE	nvarchar	50	0			否	('3')	
17	SOURCE	nvarchar	50	0			是		

4.3. 基于 PowerDesigner 设计文件的代码批量生成

基于数据库的代码批量生成完全依赖于数据库，有一定的局限性，现在我们看看如何使用 PD 设计文档来进行代码的批量生成。PD 设计文档不限数据库类型，任意数据库类库都可进行代码生成，相当的方便。要利用 PD 设计文档进行代码生成，需要在 PowerDesigner Objects 导航区选择“数据表”树节点后单击右键，在弹出的快捷菜单中选择“批量代码生成”，如下图所示。

通过此方法即可对当前 PD 设计文件的所有表进行批量业务代码的生成，生成的业务代码与 4.1 节介绍的代码完全一至，在此就不再进行展示。需要说明的说，在批量代码生成前请先到“项目属性”设置窗口，设置代码相关信息（如：版权信息、代码的命名空间，代码的作者信息、代码的年代、日期等），“批量代码生成”依赖于这些设置。

RDIFramework.NET，基于全新.NET Framework 与.NET Core 的快速信息化系统敏捷开发、整合框架，给用户和开发者最佳的.Net 框架部署方案。为企业快速构建垮平台、企业级的应用提供了强大支持。

框架官方：http://www.rdiframework.net/
http://www.guosisoft.com/
框架其他博客：
http://www.cnblogs.com/huyong/
https://yonghu.blog.csdn.net/
联系方式：
邮件：406590790@qq.com
Q Q：406590790
电话：13005007127（同微信）

RDIFramework.NET 敏捷开发框架由海南国思软件科技有限公司专业团队长期打造、一直在更新、一直在升级，请放心使用！

欢迎关注 RDIFramework.NET 敏捷开发框架官方公众微信（微信号：guosisoft），及时了解最新动态。

扫描二维码立即关注