

EZDML 脚本简介

<http://www.ezdml.com>

——HUZ 2019 年 10 月

——本文最后修订日期：20200411，对应 EZDML 版本：3.01。

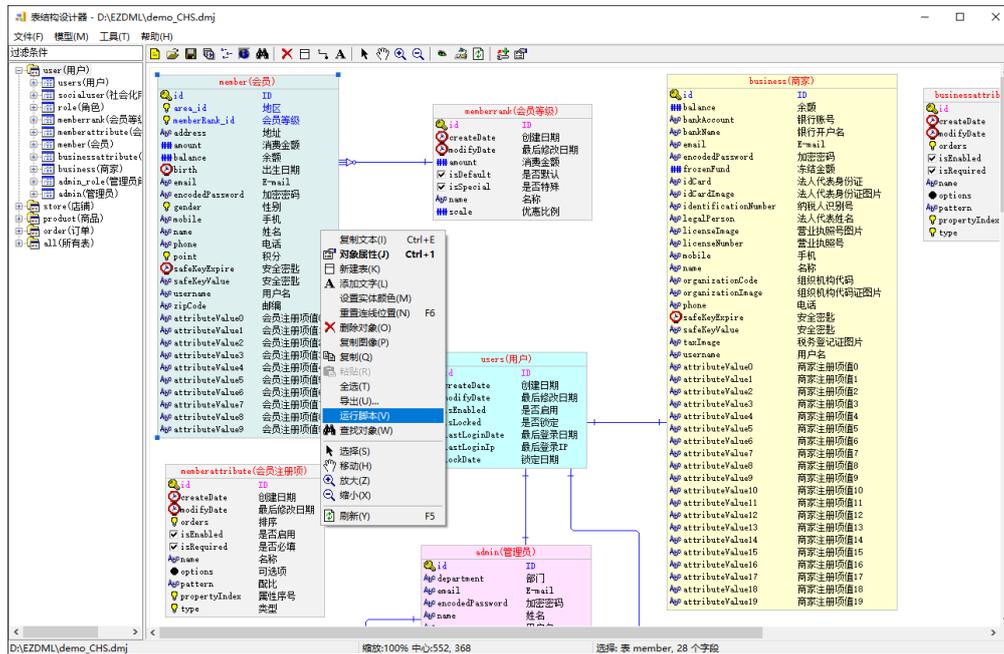
脚本能够访问模型中的所有对象及其属性，能够批量进行各种增删改查，能生成你想要的各种格式。

注意：模型目录树上有批量添加删除字段功能，比写脚本简单多了，如果仅仅是想批量增减字段，可能没有必要用脚本。

EZDML 同时支持 Javascript 和 Pascal 脚本（以下简称 JS 和 PAS），你可以根据需要选择一种。

一、JAVASCRIPT 脚本

本文档主要说如何用 JS 实现一些目的，不讲解 javascript 的基础知识。打开示例文件，选中会员表，右键弹出菜单，选择“执行脚本”：



弹出脚本编辑窗口，默认就是 JAVASCRIPT 的示例了：

```

1 //JavaScript
2 //This example shows how to traverse all Models, Tables and Fields with javascript (click New again will switch)
3 //本示例演示如何编写JavaScript脚本（再次新建可切换为PascalScript）遍历所有模型、表和字段
4
5 for (var i=0; i<allModels.count; i++)
6 //if(allModels.getItem(i) == allModels.curDataModel)
7 {
8     var md=allModels.getItem(i);
9     curOut.add("Model"+i+": "+md.name);
10
11     for (var j=0; j<md.tables.count; j++)
12         //if(md.tables.getItem(j).isSelected)
13         //if(curTable == md.tables.getItem(j).name == curTable.name)
14         {
15             var tb = md.tables.getItem(j);
16             curOut.add(" Table"+j+": "+tb.name);
17
18             for (var k=0; k<tb.metaFields.count; k++)
19             {
20                 var fd = tb.metaFields.getItem(k);
21                 curOut.add(" Field"+k+": "+fd.name);
22             }
23         }
24 }
25
26
27 /*
28 <!--
29 //This example shows how to write script-page template
30 //以下示例演示如何编写脚本页面模板
31 var md=curModel;
32 -->

```

简单说下这个脚本窗口：执行 NEW 命令新建时会自动初始化为示例脚本，默认是 JS 脚本（再点一下 NEW 命令就切换为 PAS）。JS 脚本目前不支持断点和单步调试，按 F9 运行，F1 显示帮助。脚本窗口关闭时会自动保存当前内容到临时文件（每次 F9 运行脚本时也会）。在这个脚本窗口执行“Help”菜单或按 F1 可显示一些简单的帮助：

```

PASCAL: CurOut.Add("Current table: "+CurTable.Name);
JAVASCRIPT: curOut.add("Current table: "+curTable.name);

Main variables:
var
AllModels: TctDataModelGraphList;
CurModel: TctDataModelGraph;
CurTable: TctMetaTable;
CurField: TctMetaField;
CurOut: TStrings;
CurSettingsPanel: TDmlScriptControlList;

Application: TApplication;
Screen: TScreen;

Pascal脚本须采用GBK或ASCII编码，JavaScript文件请使用UTF-8编码。
Pascal脚本包含引用文件：{$INCLUDE 文件名} 或 {$I 文件名}
JavaScript脚本包含引用文件：#include "文件名"
JavaScript脚本目前不支持单步调试。
JavaScript和Pascal脚本之间可以通过RunDmlScript函数互相混合调用。

后面所列所有类的方法和属性均为Pascal方式，在JavaScript中使用时，首字母要小写，如：
PASCAL: CurOut.Add("当前表名: "+CurTable.Name);
JAVASCRIPT: curOut.add("当前表名: "+curTable.name);

```

回到 JAVASCRIPT 的示例，往下翻，可以看到一共有五个示例：

1. 演示如何编写 JavaScript 脚本（再次新建可切换为 PascalScript）遍历所有模型、表和字段
2. 演示如何编写脚本页面模板
3. 演示如何编写脚本处理文本文件
4. 演示如何编写脚本遍历对象属性

5. 演示如何在 JS 脚本中混合调用 Pascal 脚本

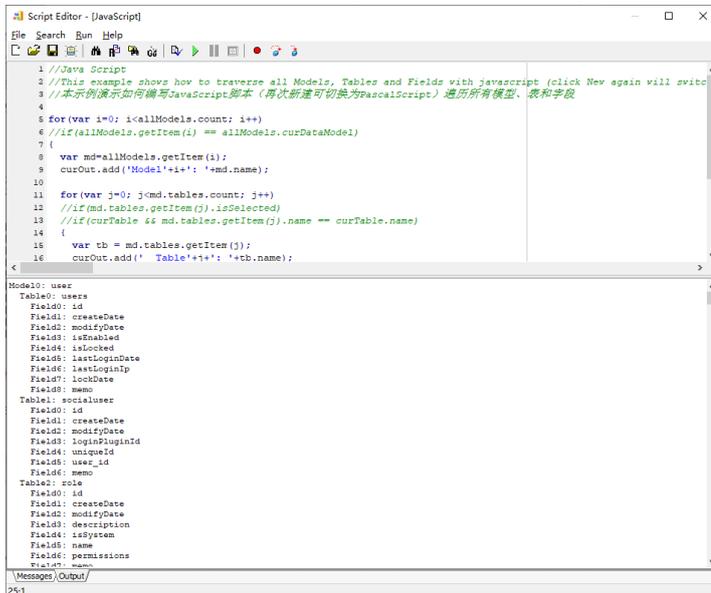
其中只有第一个没有被注释掉，我们就拿它来说事。它显然是 i j k 三个 for 循环遍历了模型、表和字段。代码很简单，不过我还是再解释一下：

```
//开始
for(var i=0; i<allModels.count; i++) //遍历所有模型，一个文件里可能有多个模型图
//if(allModels.getItem(i) == allModels.curDataModel) //判断是否为当前模型
{
    var md=allModels.getItem(i); //获得第 i 个模型
    curOut.add('Model'+i+' '+md.name); //输出模型名称

    for(var j=0; j<md.tables.count; j++) //遍历模型中的表
    //if(md.tables.getItem(j).isSelected) //判断表是否选中
    //if(curTable && md.tables.getItem(j).name == curTable.name) //判断表名是否与当前表名相同
    {
        var tb = md.tables.getItem(j); //获取表
        curOut.add(' Table'+j+' '+tb.name); //输出表名

        for(var k=0; k<tb.metaFields.count; k++) //遍历表中的字段
        {
            var fd = tb.metaFields.getItem(k); //获取字段
            curOut.add(' Field'+k+' '+fd.name); //输出字段名
        }
    }
}
//结束
```

我们把其它示例 2345 的代码删除，只留下示例 1，直接按 F9 运行，可以看到运行结果输出了所有表和字段：



```
Script Editor - [JavaScript]
File Search Run Help
1 //Java Script
2 //This example shows how to traverse all Models, Tables and Fields with javascript (click New again will switch)
3 //本示例演示如何编写JavaScript脚本(再次新建可切换为PascalScript) 遍历所有模型、表和字段
4
5 for (var i=0; i<allModels.count; i++)
6 //if(allModels.getItem(i) == allModels.curDataModel)
7 {
8     var md=allModels.getItem(i);
9     curOut.add('Model'+i+' '+md.name);
10
11     for (var j=0; j<md.tables.count; j++)
12     //if(md.tables.getItem(j).isSelected)
13     //if(curTable && md.tables.getItem(j).name == curTable.name)
14     {
15         var tb = md.tables.getItem(j);
16         curOut.add(' Table'+j+' '+tb.name);
17     }
18 }
19
20 Model0: user
21 Table0: users
22 Field0: id
23 Field1: createDate
24 Field2: modifyDate
25 Field3: isEnabled
26 Field4: isLocked
27 Field5: lastLoginDate
28 Field6: lastLoginIp
29 Field7: lockDate
30 Field8: memo
31 Table1: socialuser
32 Field0: id
33 Field1: createDate
34 Field2: modifyDate
35 Field3: loginPluginId
36 Field4: uniqueId
37 Field5: user_id
38 Field6: memo
39 Table2: role
40 Field0: id
41 Field1: createDate
42 Field2: modifyDate
43 Field3: description
44 Field4: isSystem
45 Field5: name
46 Field6: permissions
47 Field7: memo
48
49 Messages/Output/
25.1
```

把第 12 行 `//if(md.tables.getItem(j).isSelected)` 这行前面的 `//` 注释符删除，再运行，这时就只输出会员表的信息了，因为我们只选中了会员表：

```

Script Editor - [JavaScript]
File Search Run Help
6 //if(allModels.getItem(i) == allModels.curDataModel)
7 {
8   var md=allModels.getItem(i);
9   curOut.add('Model'+i+' '+md.name);
10
11   for(var j=0; j<md.tables.count; j++)
12     if(md.tables.getItem(j).isSelected)
13       //if(curTable && md.tables.getItem(j).name == curTable.name)
14       {
15         var tb = md.tables.getItem(j);
16
17         Model0: user
18         Table0: member
19         Field0: id
20         Field1: area_id
21         Field2: memberRank_id
22         Field3: address
23         Field4: amount
24         Field5: balance
25         Field6: birth
26         Field7: email
27         Field8: encodedPassword
28         Field9: gender
29         Field10: mobile
30         Field11: name
31         Field12: phone
32         Field13: point
33         Field14: safeKeyExpire
34         Field15: safeKeyValue
35         Field16: username
36         Field17: sigCode
37         Field18: attributeValue0
38         Field19: attributeValue1
39         Field20: attributeValue2
40         Field21: attributeValue3
41         Field22: attributeValue4
42         Field23: attributeValue5
43         Field24: attributeValue6
44         Field25: attributeValue7
45         Field26: attributeValue8
46         Field27: attributeValue9
47         Field28: memo
48         Model1: store
49         Model2: product
50         Model3: order
51         Model4: all
52
53         Messages/Output/
54
55         12:3

```

把第 12 行注掉，增加字段名的判断，并修改输出代码，最终改成下面的样子：

```

for(var i=0; i<allModels.count; i++)
//if(allModels.getItem(i) == allModels.curDataModel)
{
  var md=allModels.getItem(i);

  for(var j=0; j<md.tables.count; j++)
  //if(md.tables.getItem(j).isSelected)
  //if(curTable && md.tables.getItem(j).name == curTable.name)
  {
    var tb = md.tables.getItem(j);

    for(var k=0; k<tb.metaFields.count; k++)
    {
      var fd = tb.metaFields.getItem(k);
      if(fd.name.toLowerCase().indexOf("product")>=0) //判断字段名是否包含 product
      {
        curOut.add('Model_'+i+' '+md.name+' Table'+j+' '+tb.name+' Field'+k+' '+fd.name);
      }
    }
  }
}
}

```

运行结果如下，只输出了符合条件的字段：

查询遍历基本上就这样了，下面我们要动手修改了。

我要给每一个包含字段 id 的表，增加一个 memo 备注字段（如果已存在则跳过），类型为字符串，长度 512，普通索引（只是为了演示哈，其实模型目录树上有批量添加字段功能，比写脚本简单多了），代码如下：

```
for(var i=0; i<allModels.count; i++)
{
  var md=allModels.getItem(i);

  for(var j=0; j<md.tables.count; j++)
  {
    var tb = md.tables.getItem(j);

    var fd=tb.metaFields.fieldByName("id"); //查找 ID 字段
    if(!fd) //未找到 ID 字段, 跳过
    {
      curOut.add(' Table'+j+: '+tb.name +' no ID found, skipped');
    }
    else //找到 ID 字段了
    {
      fd=tb.metaFields.fieldByName("memo"); //查找 memo 字段
      if(!fd) //不存在就添加
      {
        fd=tb.metaFields.newMetaField();
        fd.name='memo';
        fd.displayName='备注';
        fd.memo='这是一个演示添加的字段';
        fd.dataType='cfdtString';
        fd.dataLength=512;
        fd.indexType='cfitNormal';
        fd.nullable=true;
        curOut.add(' Table'+j+: '+tb.name +' memo field added');
      }
      else //已存在则提示
        curOut.add(' Table'+j+: '+tb.name +' memo field exists');
    }
  }
}
```

正常来说，我们改了一个表，需要调_syncTableProps(tb)同步属性到模型中其它所有同名的表。不过，这里我们直接遍历了所有模型中的表全改了一遍，就不需要同步了。

运行结果：

```

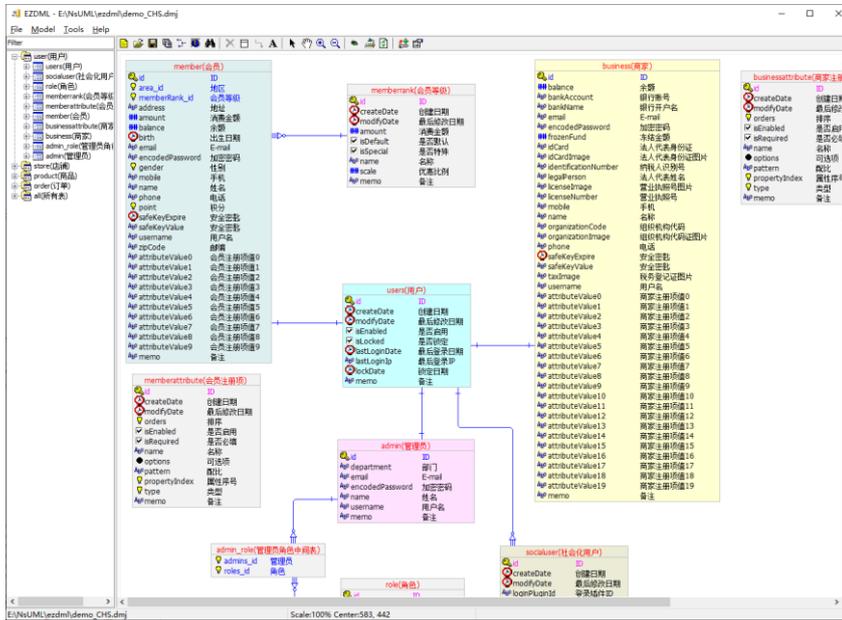
Script Editor - [JavaScript]
File Search Run Help
8
9 var fd=tb.metaFields.fieldByName("id"); //查找ID字段
10 if(!fd) //未找到ID字段, 跳过
11 {
12   curOut.add(' Table'+j+': '+tb.name + ' no ID found, skipped');
13 }
14 else //找到ID字段了
15 {
16   fd=tb.metaFields.fieldByName("memo"); //查找memo字段
17   if(!fd) //不存在就添加
18   {
19     fd=tb.metaFields.newMetaField();
20     fd.name='memo';
21     fd.displayName='备注';
22     fd.memo='这是一个演示添加的字段';
23     fd.dataType='cfdtString';
24     fd.dataLength=512;
25     fd.indexType='cfitNormal';
26     fd.nullable=true;
27     curOut.add(' Table'+j+': '+tb.name + ' memo field added');
28   }
29   else //已存在则提示
30     curOut.add(' Table'+j+': '+tb.name + ' memo field exists');
31 }
32 }
33 }
34 }

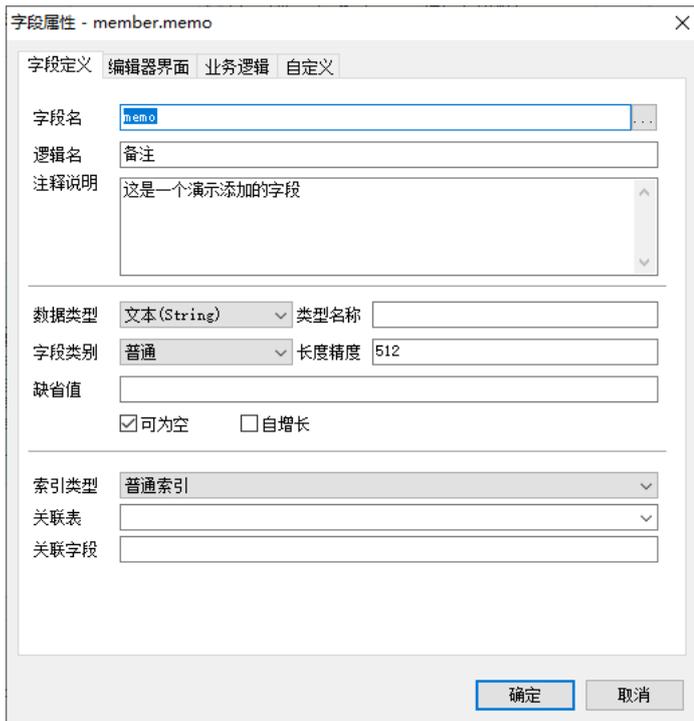
```

Table0: users memo field added
Table1: socialuser memo field added
Table2: role memo field added
Table3: memberank memo field added
Table4: memberattribute memo field added
Table5: member memo field added
Table6: businessattribute memo field added
Table7: business memo field added
Table8: admin_role no ID found, skipped
Table9: admin memo field added
Table0: store memo field added
Table1: store_productcategory no ID found, skipped
Table2: storeimage memo field added
Table3: storecategory memo field added
Table4: storefavorite memo field added
Table5: storeproductcategory memo field added
Table6: storeproducttag memo field exists
Table7: business memo field added
Table8: storerank memo field exists
Table9: svc memo field added
Table10: promotionpluginsvc memo field added
Table11: promotion_sku no ID found, skipped

Messages/Output/
34.1

回到模型图刷新一下，大部分表都加上了 memo 字段：





接下来我觉得 memo 这名字不好, 我要把这些 memo 字段改名为 remark, 逻辑名为“评论”, 长度改为 255, 代码如下:

```

for(var i=0; i<allModels.count; i++)
{
    var md=allModels.getItem(i);

    for(var j=0; j<md.tables.count; j++)
    {
        var tb = md.tables.getItem(j);

        var fd=tb.metaFields.fieldByName("memo"); //查找 memo 字段
        if(!fd) //未找到 memo 字段, 跳过
        {
            curOut.add(' Table'+j+': '+tb.name +' no memo field found, skipped');
        }
        else //找到 memo 字段了
        {
            fd.name='remark';
            fd.displayName='评论';
            fd.dataLength=255;
            curOut.add(' Table'+j+': '+tb.name +' memo field modified');
        }
    }
}
}

```

运行结果:

```

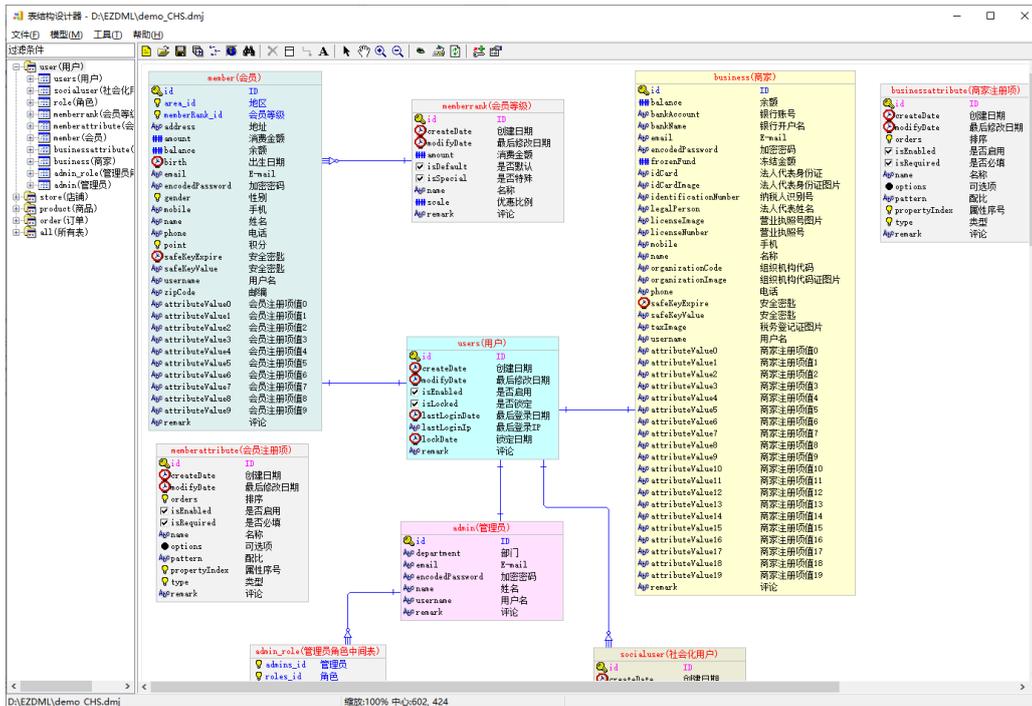
Script Editor - [JavaScript]
File Search Run Help
1 for(var i=0; i<allModels.count; i++)
2 {
3   var md=allModels.getItem(i);
4
5   for(var j=0; j<md.tables.count; j++)
6   {
7     var tb = md.tables.getItem(j);
8
9     var fd=tb.metaFields.fieldByName("memo"); //查找memo字段
10    if(!fd) //未找到memo字段, 跳过
11    {
12      curOut.add(' Table'+j+' : '+tb.name + ' no memo field found, skipped');
13    }
14    else //找到memo字段了
15    {
16      fd.name='remark';
17      fd.displayName='评论';
18      fd.dataLength=255;
19      curOut.add(' Table'+j+' : '+tb.name + ' memo field modified');
20    }
21  }

```

Table0: users memo field modified
Table1: socialuser memo field modified
Table2: role memo field modified
Table3: memberrank memo field modified
Table4: memberattribute memo field modified
Table5: member memo field modified
Table6: businessattribute memo field modified
Table7: business memo field modified
Table8: admin_role no memo field found, skipped
Table9: admin memo field modified
Table10: store memo field modified
Table11: store productcategory no memo field found, skipped

Messages/Output/
21:25

回到模型图, 刷新:



字段属性 - member.remark

字段定义 编辑器界面 业务逻辑 自定义

字段名: remark

逻辑名: 评论

注释说明: 这是一个演示添加的字段

数据类型: 文本(String) 类型名称:

字段类别: 普通 长度精度: 255

缺省值:

可为空 自增长

索引类型: 普通索引

关联表:

关联字段:

确定 取消

接下来我又后悔了，我觉得 remark 这字段没什么用，决定删除掉，代码如下：

```

for(var i=0; i<allModels.count; i++)
{
    var md=allModels.getItem(i);

    for(var j=0; j<md.tables.count; j++)
    {
        var tb = md.tables.getItem(j);

        var fd=tb.metaFields.fieldByName("remark"); //查找 remark 字段
        if(!fd) //未找到 remark 字段，跳过
        {
            curOut.add(' Table'+j+: '+tb.name +' no remark field found, skipped');
        }
        else //找到 remark 字段了，删除之
        {
            tb.metaFields.remove(fd);
            curOut.add(' Table'+j+: '+tb.name +' remark field removed');
        }
    }
}

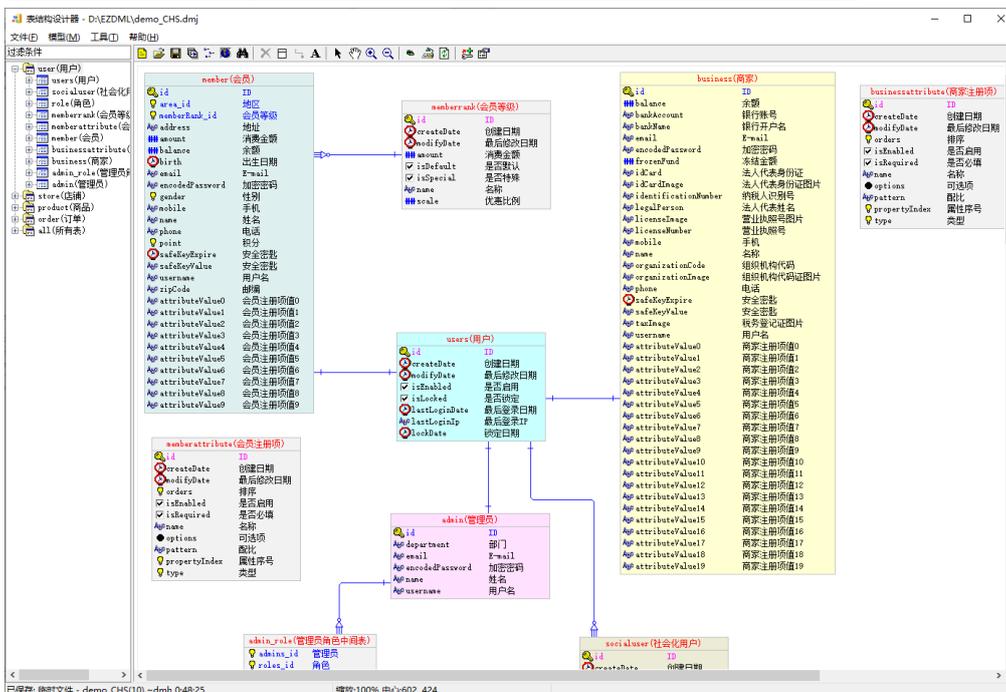
```

运行结果：

```

Script Editor - [JavaScript]
File Search Run Help
1 for(var i=0; i<allModels.count; i++)
2 {
3   var md=allModels.getItem(i);
4
5   for(var j=0; j<md.tables.count; j++)
6   {
7     var tb = md.tables.getItem(j);
8
9     var fd=tb.metaFields.fieldByName("remark"); //查找remark字段
10    if(!fd) //未找到remark字段, 跳过
11    {
12      curOut.add(' Table'+j+': '+tb.name +' no remark field found, skipped');
13    }
14    else //找到remark字段了, 删除之
15    {
16      tb.metaFields.remove(fd);
17      curOut.add(' Table'+j+': '+tb.name +' remark field removed');
18    }
19  }
20 }
21
Table80: store remark field removed
Table81: store_productcategory no remark field found, skipped
Table82: storeimage remark field removed
Table83: storecategory remark field removed
Table84: storefavorite remark field removed
Table85: storeproductcategory remark field removed
Table86: storeproducttag remark field removed
Table87: storerank remark field removed
Table88: svc remark field removed
Table89: users remark field removed

```



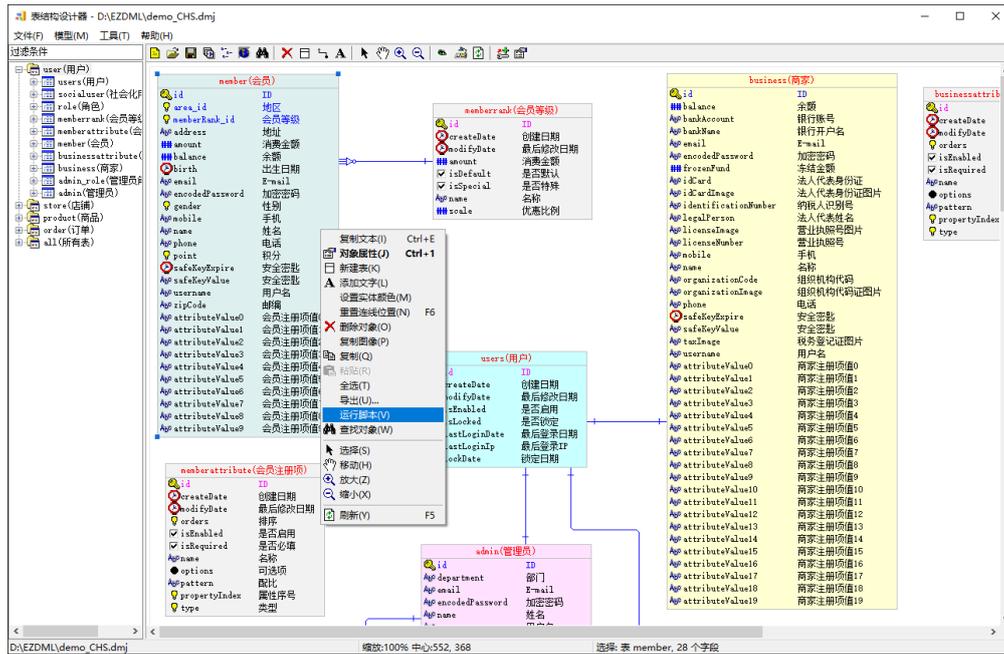
好了，用 JS 增删改查了一遍，想必你已经有个大概了解了。

二、PASCAL 脚本

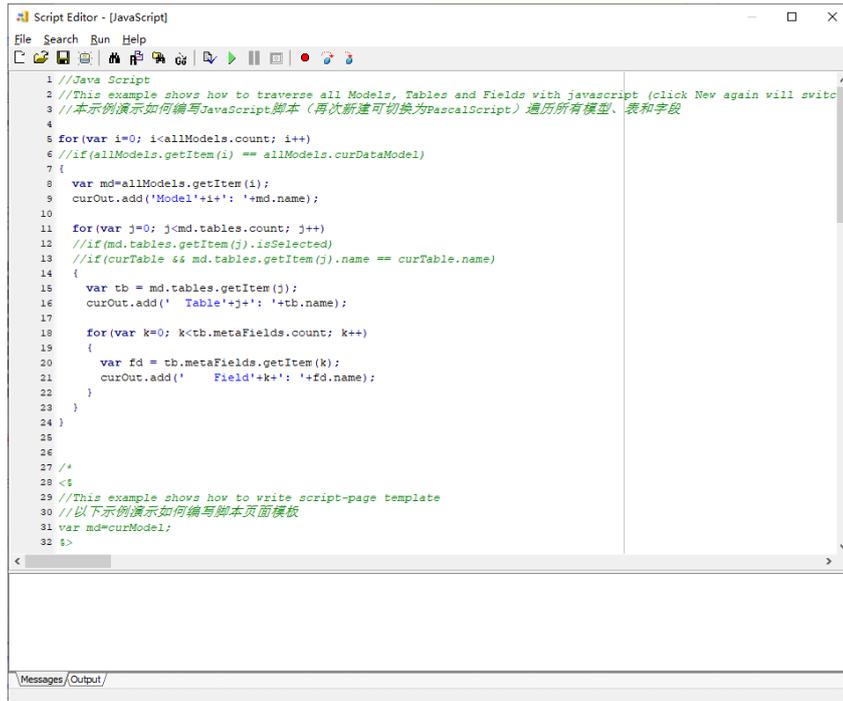
会 JS 的人没必要看这一节。

写了 JS 的内容后，感觉 PASCAL 似乎没有必要写了，不过空着也不好，还是写点凑数吧。

还是打开示例文件，选中会员表，右键弹出菜单，选择“执行脚本”：



弹出脚本编辑窗口，默认就是 JAVASCRIPT 的示例：



再执行一次 File/New 菜单命令，切换为 PASCAL 脚本（如果还是 JS，就再点一次）：

```

1 //Pascal Script
2 //This example shows how to traverse all Models, Tables and Fields with PascalScript (click New again wi
3 //本示例演示如何编写PascalScript脚本 (再次新建可切换为JavaScript) 遍历所有模型、表和字段
4 var
5   I, J, K: Integer;
6   md: TCtDataModelGraph;
7   tb: TCtMetaTable;
8   fd: TCtMetaField;
9 begin
10  for I:=0 to AllModels.Count-1 do
11    //if AllModels.Items[I] = AllModels.CurDataModel then
12    begin
13      md := AllModels.Items[I];
14      CurOut.Add('Model'+IntToStr(I)+' '+md.Name);
15    end;
16    for J:=0 to md.Tables.Count-1 do
17      //if md.Tables.Items[J].IsSelected then
18      //if CurTable <> nil then if md.Tables.Items[J].Name = CurTable.Name then
19      begin
20        tb := md.Tables.Items[J];
21        CurOut.Add(' Table'+IntToStr(J)+' '+tb.Name);
22      end;
23      for K:=0 to tb.MetaFields.Count -1 do
24        begin
25          fd := tb.MetaFields.Items[K];
26          CurOut.Add(' Field'+IntToStr(K)+' '+fd.Name);
27        end;
28      end;
29    end;
30 end;
31
32 (*

```

PASCAL 脚本支持断点和单步调试，按 F9 运行，F1 显示帮助。脚本窗口关闭时会自动保存当前内容到临时文件（每次 F9 运行脚本时也会）。

照例解释一下：

```

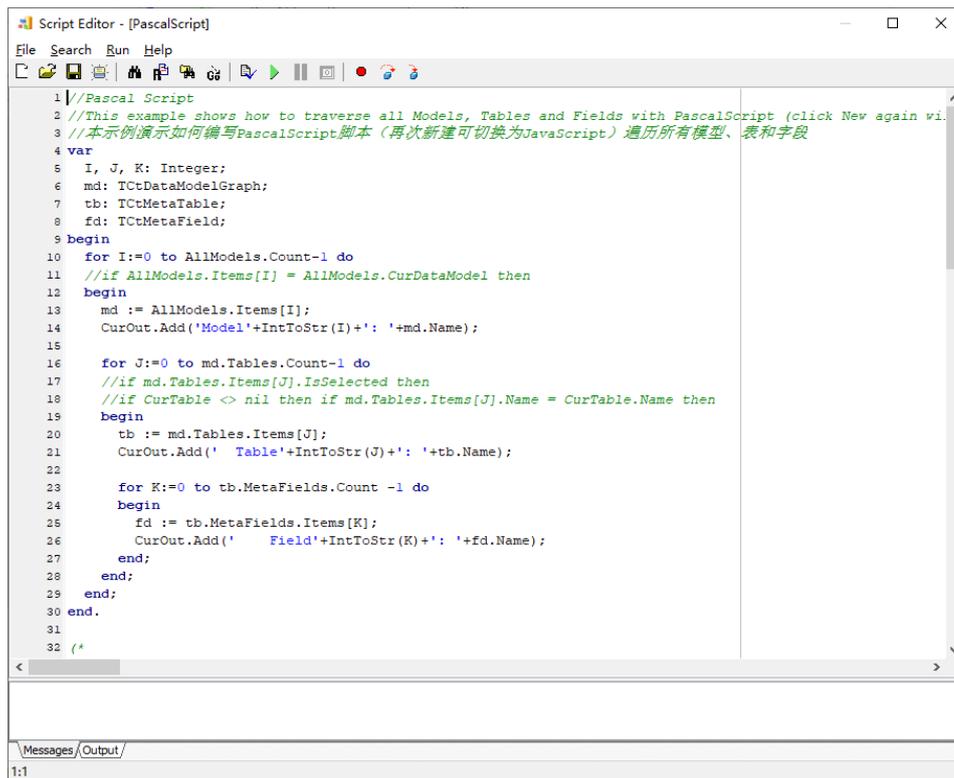
var //变量声明, PASCAL 比较烦人的一点就是变量声明要在最前面
  I, J, K: Integer;
  md: TCtDataModelGraph;
  tb: TCtMetaTable;
  fd: TCtMetaField;
begin //开始
  for I:=0 to AllModels.Count-1 do //遍历所有模型
    //if AllModels.Items[I] = AllModels.CurDataModel then //判断如果是当前模型
    begin
      md := AllModels.Items[I]; //获取模型
      CurOut.Add('Model'+IntToStr(I)+' '+md.Name); //输出模型名称

      for J:=0 to md.Tables.Count-1 do //遍历该模型中的所有表
        //if md.Tables.Items[J].IsSelected then //如果表被选中
        //if CurTable <> nil then if md.Tables.Items[J].Name = CurTable.Name then //或者表名与当前表一样
        begin
          tb := md.Tables.Items[J]; //获取表
          CurOut.Add(' Table'+IntToStr(J)+' '+tb.Name); //输出表名

          for K:=0 to tb.MetaFields.Count -1 do //遍历该表的所有字段
            begin
              fd := tb.MetaFields.Items[K]; //获取字段
              CurOut.Add(' Field'+IntToStr(K)+' '+fd.Name); //输出字段名
            end;
          end;
        end;
      end;
    end;
  end. //PASCAL 的最后结束是 end 加一个点

```

直接按 F9 运行，效果跟之前 JS 的示例是一样的：



```
1 //Pascal Script
2 //This example shows how to traverse all Models, Tables and Fields with PascalScript (click New again wi
3 //本示例演示如何编写PascalScript脚本 (再次新建可切换为JavaScript) 遍历所有模型、表和字段
4 var
5   I, J, K: Integer;
6   md: TCtDataModelGraph;
7   tb: TCtMetaTable;
8   fd: TCtMetaField;
9 begin
10  for I:=0 to AllModels.Count-1 do
11    //if AllModels.Items[I] = AllModels.CurDataModel then
12  begin
13    md := AllModels.Items[I];
14    CurOut.Add('Model'+IntToStr(I)+' : '+md.Name);
15
16    for J:=0 to md.Tables.Count-1 do
17      //if md.Tables.Items[J].IsSelected then
18      //if CurTable <> nil then if md.Tables.Items[J].Name = CurTable.Name then
19    begin
20      tb := md.Tables.Items[J];
21      CurOut.Add(' Table'+IntToStr(J)+' : '+tb.Name);
22
23      for K:=0 to tb.MetaFields.Count -1 do
24        begin
25          fd := tb.MetaFields.Items[K];
26          CurOut.Add(' Field'+IntToStr(K)+' : '+fd.Name);
27        end;
28      end;
29    end;
30 end.
31
32 /*
```

增加字段名的判断，并修改输出代码，最终改成下面的样子：

```
var
  I, J, K: Integer;
  md: TCtDataModelGraph;
  tb: TCtMetaTable;
  fd: TCtMetaField;
begin
  for I:=0 to AllModels.Count-1 do
    //if AllModels.Items[I] = AllModels.CurDataModel then
  begin
    md := AllModels.Items[I];

    for J:=0 to md.Tables.Count-1 do
      //if md.Tables.Items[J].IsSelected then
      //if CurTable <> nil then if md.Tables.Items[J].Name = CurTable.Name then
    begin
      tb := md.Tables.Items[J];

      for K:=0 to tb.MetaFields.Count -1 do
        begin
          fd := tb.MetaFields.Items[K];
          if Pos('product', LowerCase(fd.Name))>0 then //判断字段名是否包含 product
            CurOut.Add('Model_'+IntToStr(i)+' : '+md.name+' Table'+IntToStr(j)+' : '+tb.name+'
Field'+IntToStr(k)+' : '+fd.name);
          end;
        end;
      end;
    end.
end.
```

运行结果如下，只输出了符合条件的字段：

```

1 var
2   I, J, K: Integer;
3   md: TCtDataModelGraph;
4   tb: TCtMetaTable;
5   fd: TCtMetaField;
6 begin
7   for I:=0 to AllModels.Count-1 do
8     //if AllModels.Items[I] = AllModels.CurDataModel then
9     md := AllModels.Items[I];
10
11    for J:=0 to md.Tables.Count-1 do
12      //if md.Tables.Items[J].IsSelected then
13      //if CurTable <> All then if md.Tables.Items[J].Name = CurTable.Name then
14      begin
15        tb := md.Tables.Items[J];
16        for K:=0 to tb.MetaFields.Count-1 do
17          begin
18            fd := tb.MetaFields.Items[K];
19            if Pos('product', LowerCase(fd.Name))>0 then //判断字段名是否包含product
20              CurOut.Add('Model_'+IntToStr(I)+'_'+md.name+' Table'+IntToStr(J)+'_'+tb.name+' Field'+IntToStr(K)+'_'+fd.name);
21          end;
22        end;
23      end;
24    end;
25  end;
26 end.
27
Model_1: store Table1: store_productcategory Field1: productcategory_id
Model_1: store Table18: consultation Field1: product_id
Model_1: store Table19: categoryapplication Field1: productcategory_id
Model_2: product Table1: store_productcategory Field1: productcategory_id
Model_2: product Table2: specification Field1: productcategory_id
Model_2: product Table5: sku Field11: product_id
Model_2: product Table6: review Field19: product_id
Model_2: product Table11: productfavorite Field4: product_id
Model_2: product Table13: productcategory_permission Field10: productcategory_id
Model_2: product Table14: productcategory_baseed Field10: productcategory_id
Model_2: product Table16: product_storeproducttag Field1: products_id
Model_2: product Table17: product_storeproducttag Field1: storeProductTag_id
Model_2: product Table18: product_promotion Field10: products_id
Model_2: product Table19: product_producttag Field10: products_id
Model_2: product Table1: product_producttag Field1: productTag_id
Model_2: product Table19: product Field22: productImages
Model_2: product Table19: product Field27: productCategory_id
Model_2: product Table19: product Field19: storeProductCategory_id
Model_3: order Table2: product Field22: productImages
Model_3: order Table2: product Field19: productCategory_id
Model_3: order Table3: product Field19: storeProductCategory_id
Model_4: all Table1: attribute Field17: productCategory_id
Model_4: all Table2: categoryapplication Field1: productcategory_id
Model_4: all Table3: consultation Field1: product_id
Model_4: all Table4: categoryapplication Field1: productcategory_id
Messages (Output)
2533

```

很多时候我们只要处理当前表 CurTable，假设我要为当前表的字段生成某个赋值代码，这时可以这样遍历字段（请注意要选一个表，以保证当前表 CurTable 有值）：

```

var
  I: Integer;
  fd: TCtMetaField;
begin
  for I:=0 to CurTable.MetaFields.Count-1 do
  begin
    fd := CurTable.MetaFields.Items[I];
    CurOut.Add(' if(member1.'+fd.Name+'!=null)');
    CurOut.Add(' member1.'+fd.Name+'=member2.'+fd.Name+');
  end;
end.

```

运行效果：

```

Script Editor - [PascalScript]
File Search Run Help
1
2 var
3 I: Integer;
4 fd: TCtMetaField;
5 begin
6 for I:=0 to CurTable.MetaFields.Count -1 do
7 begin
8 fd := CurTable.MetaFields.Items[I];
9 CurOut.Add(' if(member1.'+fd.Name+'==null)');
10 CurOut.Add(' member1.'+fd.Name+'=member2.'+fd.Name+');
11 end;
12 end.
13

if(member1.id=null)
member1.id:=member2.id;
if(member1.area_id=null)
member1.area_id:=member2.area_id;
if(member1.memberRank_id=null)
member1.memberRank_id:=member2.memberRank_id;
if(member1.address=null)
member1.address:=member2.address;
if(member1.amount=null)
member1.amount:=member2.amount;
if(member1.balance=null)
member1.balance:=member2.balance;
if(member1.birth=null)
member1.birth:=member2.birth;
if(member1.email=null)
member1.email:=member2.email;
if(member1.encodedPassword=null)
member1.encodedPassword:=member2.encodedPassword;
if(member1.gender=null)
member1.gender:=member2.gender;
if(member1.mobile=null)
member1.mobile:=member2.mobile;

(Messages)Output/
9:20

```

再简单说下批量加字段，要给每一个包含字段 id 的表，增加一个 memo 备注字段（如
果已存在则跳过），类型为字符串，长度 512，普通索引（只是为了演示，其实模型目录树
上有批量添加字段功能，比写脚本简单），代码如下：

```

var
  I, J: Integer;
  md: TCtDataModelGraph;
  tb: TCtMetaTable;
  fd: TCtMetaField;
begin
  for I:=0 to AllModels.Count-1 do
  begin
    md := AllModels.Items[I];

    for J:=0 to md.Tables.Count-1 do
    begin
      tb := md.Tables.Items[J];
      fd := tb.metaFields.fieldByName('id');
      if fd = nil then //未找到 ID 字段，跳过
      begin
        curOut.add(' Table'+IntToStr(j)+' : '+tb.name + ' no ID found, skipped');
      end
      else
      begin
        fd := tb.metaFields.fieldByName('memo'); //查找 memo 字段
        if fd = nil then //不存在就添加
        begin
          fd:=tb.metaFields.newMetaField();
          fd.name:='memo';
          fd.displayName:='备注';
          fd.memo:='这是一个演示添加的字段';
          fd.dataType:=cfdtString;
          fd.dataLength:=512;
        end
      end
    end
  end
end

```

```

    fd.indexType:=cfitNormal;
    fd.nullable:=true;
    curOut.add(' Table'+IntToStr(j)+' '+tb.name + ' memo field added');
end
else //已存在则提示
    curOut.add(' Table'+IntToStr(j)+' '+tb.name + ' memo field exists');
end;
end;
end;
end;
end.

```

正常来说，我们改了一个表，需要调 SyncTableProps(tb)同步属性到模型中其它所有同名的表。不过，这里我们直接遍历了所有模型中的表全改了一遍，就不需要同步了。

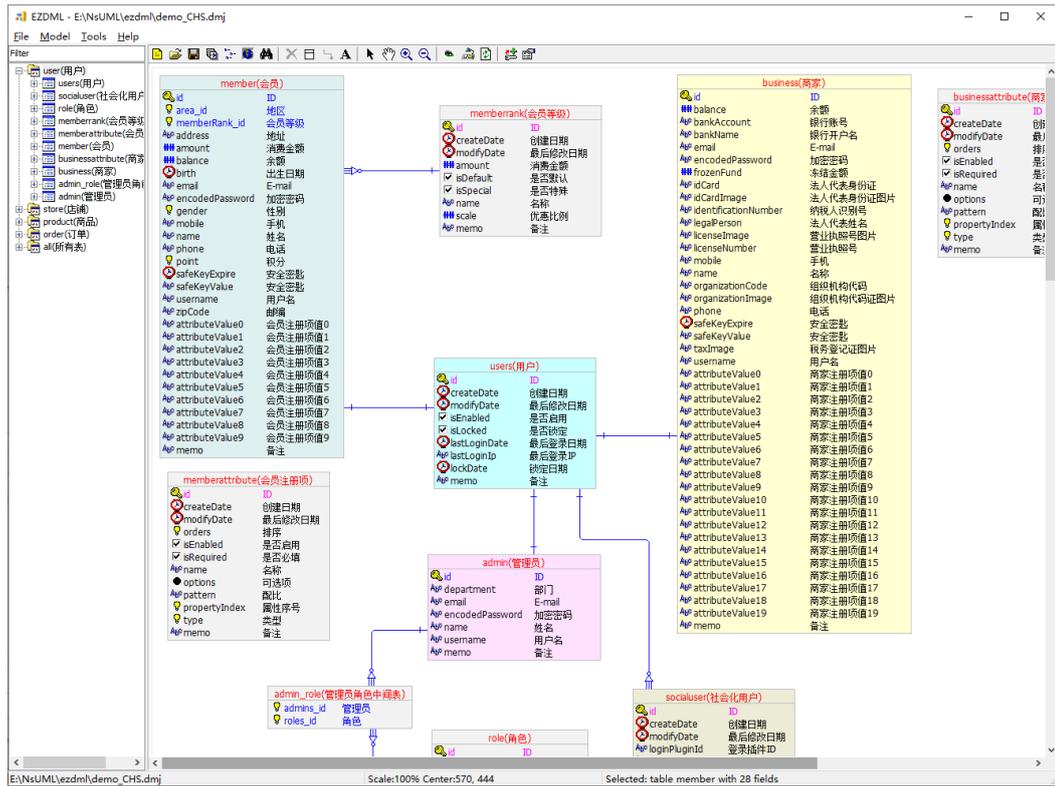
运行结果：

```

Script Editor - [PascalScript]
File Search Run Help
13 begin
14   tb := md.Tables.Items[j];
15   fd := tb.metaFields.fieldByName('id');
16   if fd = nil then //未找到ID字段, 跳过
17     begin
18       curOut.add(' Table'+IntToStr(j)+' '+tb.name + ' no ID found, skipped');
19     end
20   else
21     begin
22       fd := tb.metaFields.fieldByName('memo'); //查找memo字段
23       if fd = nil then //不存在就添加
24         begin
25           fd:=tb.metaFields.newMetaField();
26           fd.name:='memo';
27           fd.displayName:='备注';
28           fd.memo:='这是一个演示添加的字段';
29           fd.dataType:=cfdtString;
30           fd.dataLength:=512;
31           fd.indexType:=cfitNormal;
32           fd.nullable:=true;
33           curOut.add(' Table'+IntToStr(j)+' '+tb.name + ' memo field added');
34         end
35       else //已存在则提示
36         curOut.add(' Table'+IntToStr(j)+' '+tb.name + ' memo field exists');
37       end;
38     end;
39   end;
40 end.
41
Table0: users memo field added
Table1: socialuser memo field added
Table2: role memo field added
Table3: memberank memo field added
Table4: memberattribute memo field added
Table5: member memo field added
Table6: businessattribute memo field added
Table7: business memo field added
Table8: admin_role no ID found, skipped
Table9: admin memo field added
Table10: store memo field added
Table11: store_productcategory no ID found, skipped
Table12: storeadimage memo field added
Table13: storecategory memo field added
Table14: storefavorite memo field added
Table15: storeproductcategory memo field added
Table16: storeproducttag memo field exists
Table17: business memo field added
Table18: storerank memo field exists
Table19: svc memo field added
Table20: promotionpluginsvc memo field added
Table21: promotion_sfu no ID found, skipped
Table22: promotion_memberank no ID found, skipped
Table23: promotion_coupon no ID found, skipped
Messages/Output/
19:79

```

回到模型图，刷新可见效果：



PAS 就先说到这。

如需了解 EZDML 脚本中的对象及其关系等详细内容，请参考《EZDML 脚本参考》。