عنوان: **آشنایی با Oslo - قسمت اول**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۱۲/۰۲ ۱۶:۴۲:۴۱
آدرس: www.dotnettips.info
برچسب‌ها: Oslo

Oslo

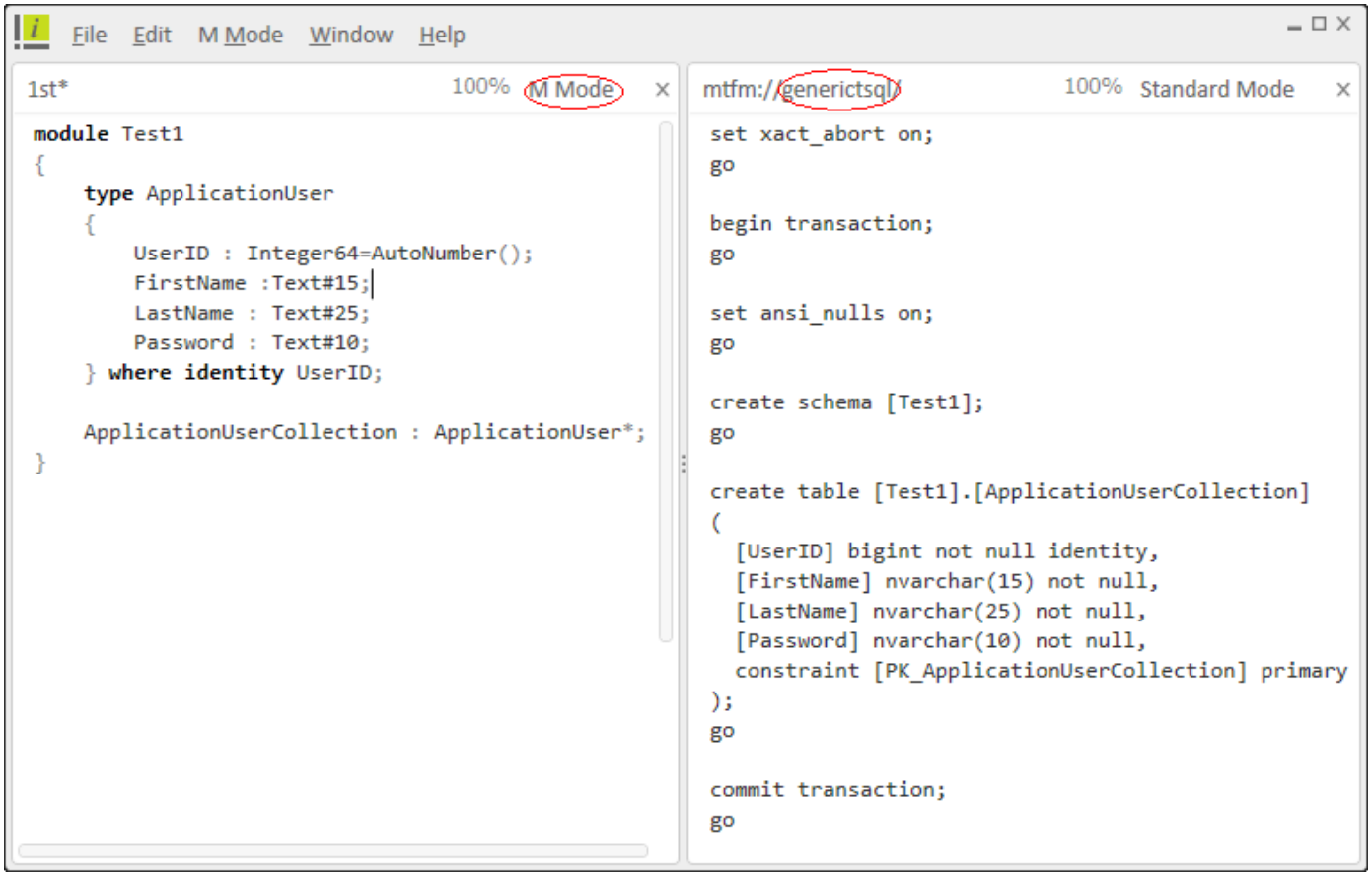
پلتفرم جدید مدل‌سازی مایکروسافت است که در سال‌های آتی مورد استفاده قرار خواهد گرفت و همچنین این روزها در مجامع توسعه و طراحی برنامه‌ها به شدت مورد بحث و توجه است. به همین جهت در طی مقالاتی با این پلتفرم جدید بیشتر آشنا خواهیم شد.

دریافت Oslo

Oslo از سه قسمت عمده تشکیل شده است:
الف) زبان مدل سازی M
ب) ابزار مدل سازی Quadrant
ج) استفاده از SQL Server به عنوان مخزن

زبان مدل سازی M از سه قسمت به نام‌های MGrammer ، MGraph و MSchema تشکیل می‌شود.
MGrammer : گرامر مورد استفاده در SDL را تعریف می‌کند. S yntax D irected T ranslation
MSchema : طرح مدل را تعریف خواهد کرد.
MGraph : اگر MSchema بیانگر انواع باشد، MGraph بیانگر وهله‌ها خواهد بود.

یک مثال:
برنامه‌ی Intellipad را اجرا کنید (فرض بر این است که SDK فوق را نصب کرده‌اید)



در اینجا حالت را بر روی M Mode قرار دهید (مطابق تصویر) و همچنین از منوی ظاهر شده‌ی M Mode ، گزینه‌ی Generic T-SQL preview را هم انتخاب کنید.

اولین ماژول ما به صورت زیر است:

```
module Test1
{
  type ApplicationUser
  {
      UserID : Integer64=AutoNumber();
      FirstName :Text#15;
      LastName : Text#25;
      Password : Text#10;
  } where identity UserID;
}
```

ابتدا نام ماژول مشخص می‌شود. شبیه به معرفی یک فضای نام در برگیرنده‌ی اشیای مربوطه. سپس type ، بیانگر همان MSchema خواهد بود.

در این مثال شناسه‌ی کاربری از نوع Integer64 خود افزایش یابنده تعریف شده است (نوع identity در اس کیوال سرور).

فیلدهای نام ، نام خانوادگی و کلمه‌ی عبور از نوع متنی با اندازه‌های مشخص 15 ، 25 و 10 کاراکتر تعریف شده‌اند. اگر اندازه مشخص نبود نوع را تنها Text تعریف کنید.

نکته:

-1اگر پس از Text علامت ? قرار گیرد، به معنای فیلدی از نوع nullable خواهد بود و برعکس. زیبایی Intellipad هم در اینجا است که بلافاصله پس از تایپ شما، عبارت T-SQL معادل را تولید می‌کند.

-2در اینجا UserID به صورت identity معرفی شده است. در زبان ام ، identity همانند primary key در عبارات T-SQL عمل

می‌کند و نباید اشتباه گرفته شود.

تا اینجا فقط یک type تعریف شده است. برای تبدیل آن به یک جدول باید آن‌را توسعه داد.

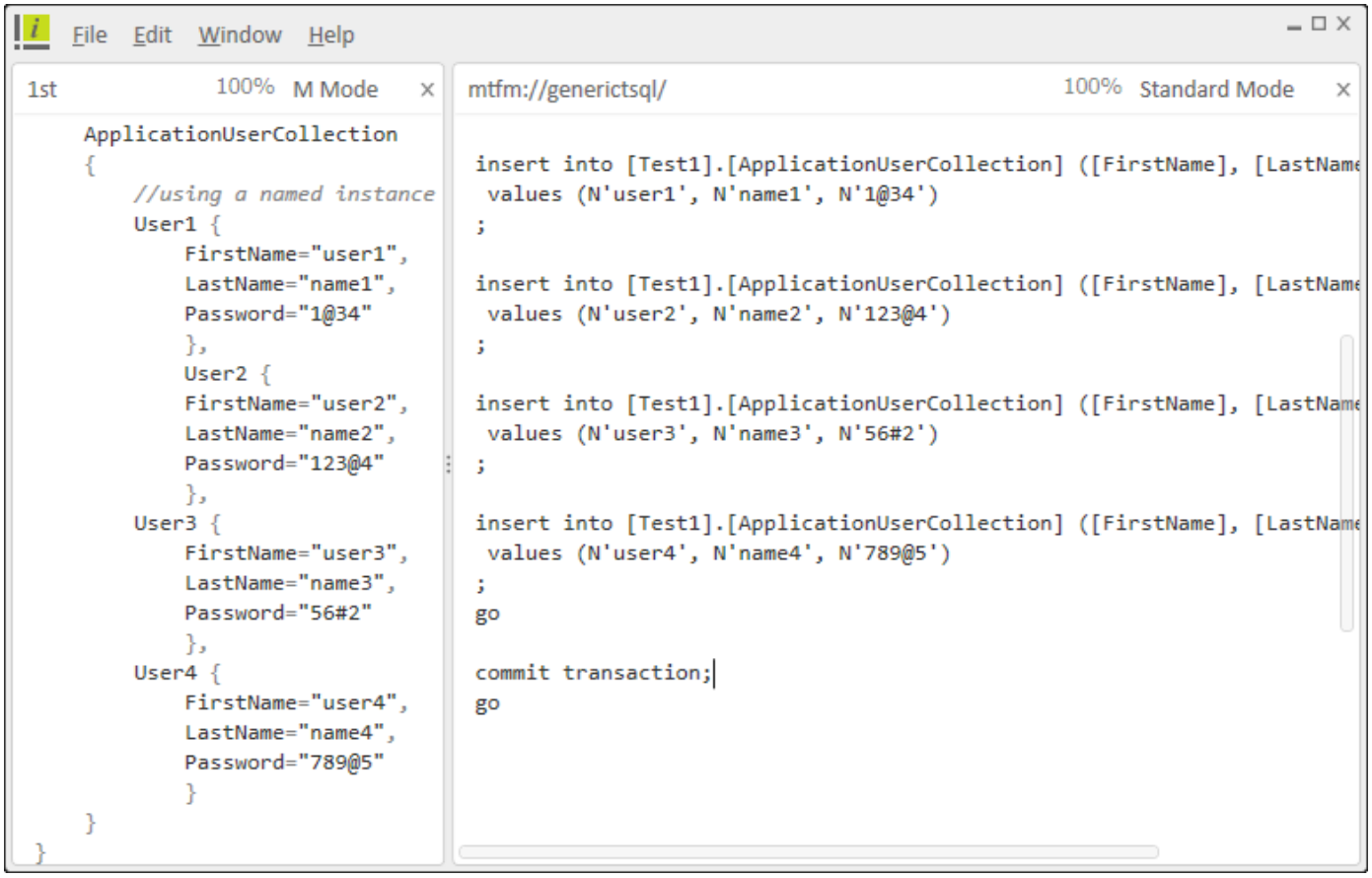
```
ApplicationUserCollection : ApplicationUser*;
```

این سطر را به پس از تعریف type اضافه نمائید. علامت ستاره در اینجا به معنای صفر یا بیشتر است و جهت بسط نوع تعریف شده به یک مجموعه به کار می‌رود. اکنون با اضافه شدن این سطر، Intellipad بلافاصله عبارات T-SQL معادل را تولید خواهد کرد که در تصویر مشخص است. به این صورت MGraph ما که بیانگر وهله‌هایی از نوع ApplicationUser هستند تولید گردید.

اکنون قصد داریم گروهی از کاربرها را به صورت نمونه ایجاد کنیم:

```
ApplicationUserCollection
  {
      //using a named instance
      User1 {
          FirstName="user1",
          LastName="name1",
          Password="1@34"
          },
          User2 {
          FirstName="user2",
          LastName="name2",
          Password="123@4"
          },
      User3 {
          FirstName="user3",
          LastName="name3",
          Password="56#2"
          },
      User4 {
          FirstName="user4",
          LastName="name4",
          Password="789@5"
          }
  }
```

سطرهای فوق را پس از تعریف ApplicationUserCollection در Intellipad اضافه کنید. بلافاصله Intellipad عبارات T-SQL معادل را برای ما تولید خواهد کرد.


File Edit Window Help
1st 100% M Mode
ApplicationUserCollection
{
//using a named instance
User1 {
FirstName="user1",
LastName="name1",
Password="1@34"
},
User2 {
FirstName="user2",
LastName="name2",
Password="123@4"
},
User3 {
FirstName="user3",
LastName="name3",
Password="56#2"
},
User4 {
FirstName="user4",
LastName="name4",
Password="789@5"
}
}
}
mtfm://genericsql/ 100% Standard Mode
insert into [Test1].[ApplicationUserCollection] ([FirstName], [LastName
values (N'user1', N'name1', N'1@34')
;
insert into [Test1].[ApplicationUserCollection] ([FirstName], [LastName
values (N'user2', N'name2', N'123@4')
;
insert into [Test1].[ApplicationUserCollection] ([FirstName], [LastName
values (N'user3', N'name3', N'56#2')
;
insert into [Test1].[ApplicationUserCollection] ([FirstName], [LastName
values (N'user4', N'name4', N'789@5')
;
go
commit transaction;
go

ادامه دارد ...

نظرات خوانندگان

نویسنده: farasun
تاریخ: ۱۳۸۷/۱۲/۰۲ ۱۹:۴۹:۰۰

بسیار عالی. من در مورد Oslo مقالاتی خونده بودم و میخواستم در موردش بنویسم. چه خوب که شما زودتر نوشتین.البته شما همیشه با ذکر مثال جلو میرین که این خیلی خوبه. حالا من فقط لینک میدم به نوشته شما.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۱۲/۰۳ ۰۰:۲۷:۰۰

با تشکر.
لطفا شما هم بنویسید. از خواندن مقالات شما لذت می‌برم. ;)

نویسنده: حسین
تاریخ: ۱۳۸۷/۱۲/۰۵ ۱۳:۲۲:۰۰

مطلب جالبی بود. . . از فید دنبالت میکنیم ...

عنوان: **آشنایی با Oslo - قسمت دوم**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۱۲/۰۳ ۱۱:۳۷:۵۱
آدرس: www.dotnettips.info
برچسب‌ها: Oslo

قبل شروع این قسمت بد نیست با یک سری از وبلاگ‌های اعضای تیم Oslo آشنا شویم:

"Oslo" Modeling Language Team Blog

Intellipad Team Blog

Adventures in the guts of Oslo

!Pinky

در ادامه‌ی مثال قسمت قبل، اکنون می‌خواهیم entity جدیدی به نام Project را به مدل اضافه کنیم:

```
//mschema to define a Project type
type Project
{
    ProjectID : Integer64 = AutoNumber();
    ProjectName : Text#25;
    ConectionStringSource : Text;
    ConectionStringDestination : Text;
    DateCompared: DateTime;
    Comment: Text?;
    ProjectOwner: ApplicationUser;
} where identity ProjectID;
```

مطابق تعاریف فوق، فیلد ProjectOwner ارجاعی را به نوع ApplicationUser که پیشتر ایجاد کردیم دارد. اکنون برای مشاهده‌ی تغییرات حاصل شده نیاز به ایجاد یک جدول از روی این نوع جدید است که foreign key آن به صورت زیر تعریف می‌شود:

```
//this will define a SQL foreign key relationship
ProjectCollection : Project* where item.ProjectOwner in ApplicationUserCollection;
```

پس از افزودن این سطر، Intellipad بلافاصله اسکریپت T-SQL آن‌را برای ما ایجاد می‌کند که به شرح زیر است:

```
set xact_abort on;
go

begin transaction;
go

set ansi_nulls on;
go

create schema [Test1];
go

create table [Test1].[ApplicationUserCollection]
(
 [UserID] bigint not null identity,
 [FirstName] nvarchar(max) null,
 [LastName] nvarchar(25) not null,
```

```
 [Password] nvarchar(10) not null,
 constraint [PK_ApplicationUserCollection] primary key clustered ([UserID])
);
go

create table [Test1].[ProjectCollection]
(
 [ProjectID] bigint not null identity,
 [Comment] nvarchar(max) null,
 [ConectionStringDestination] nvarchar(max) not null,
 [ConectionStringSource] nvarchar(max) not null,
 [DateCompared] datetime2 not null,
 [ProjectName] nvarchar(25) not null,
 [ProjectOwner] bigint not null,
 constraint [PK_ProjectCollection] primary key clustered ([ProjectID]),
 constraint [FK_ProjectCollection_ProjectOwner_Test1_ApplicationUserCollection] foreign key
([ProjectOwner]) references [Test1].[ApplicationUserCollection] ([UserID])
);
go

insert into [Test1].[ApplicationUserCollection] ([FirstName], [LastName], [Password])
values (N'user1', N'name1', N'1@34')
;

insert into [Test1].[ApplicationUserCollection] ([FirstName], [LastName], [Password])
values (N'user2', N'name2', N'123@4')
;

insert into [Test1].[ApplicationUserCollection] ([FirstName], [LastName], [Password])
values (N'user3', N'name3', N'56#2')
;

insert into [Test1].[ApplicationUserCollection] ([FirstName], [LastName], [Password])
values (N'user4', N'name4', N'789@5')
;
go

commit transaction;

Go
```

همانطور که ملاحظه می‌کنید، هنگام کار کردن با یک مدل، نگهداری و توسعه‌ی آن واقعا ساده‌تر است از ایجاد این دستورات T-SQL .

نکته:

جهت آشنایی با انواع داده‌های مجاز در زبان M می‌توان به مستندات رسمی آن مراجعه نمود:

The "Oslo" Modeling Language Specification

اکنون قصد داریم همانند مثال قسمت قبل، تعدادی رکورد آزمایشی را برای این جدول تعریف کنیم:

```
ProjectCollection
   {
       Project1{
           ProjectName = "My Project 1",
           ConectionStringSource = "Data Source=.;Initial Catalog=MyDB1;Integrated Security=True;",
           ConectionStringDestination = "Data Source=.;Initial Catalog=MyDB2;Integrated
Security=True;",
           Comment="Project Comment",
           DateCompared=2009-01-01T00:00:00,
           ProjectOwner=ApplicationUserCollection.User1 //direct ref to User1 (FK)
       },
       Project2{
           ProjectName = "My Project 2",
           ConectionStringSource = "Data Source=.;Initial Catalog=MyDB1;Integrated Security=True;",
           ConectionStringDestination = "Data Source=.;Initial Catalog=MyDB2;Integrated
Security=True;",
           Comment="Project Comment",
           DateCompared=2009-01-01T00:00:00,
           ProjectOwner=ApplicationUserCollection.User2 //direct ref to User2 (FK)
       }
```

```
    }
```

چون بین ProjectOwner و ApplicationUserCollection رابطه ایجاد کرده‌ایم، هنگام استفاده از آن‌ها، برنامه Intellipad جهت سهولت کار، IntelliSense مربوطه را نیز نمایش خواهد داد :

```
ProjectCollection
{
    Project1{
        ProjectName = "My Project 1",
        ConectionStringSource = "Data Source=.;Initial Catalog=MyDB1;Integrat
        ConectionStringDestination = "Data Source=.;Initial Catalog=MyDB2;Int
        Comment="Project Comment",
        DateCompared=2009-01-01T00:00:00,
        ProjectOwner=ApplicationUserCollection.User1 //direct ref to User1 (F
    },
    Project2{
        ProjectName = "My Project 2",
        ConectionStringSource = "Data Source=.;
        ConectionStringDestination = "Data Sour
        Comment="Project Comment",
        DateCompared=2009-01-01T00:00:00,
        ProjectOwner=ApplicationUserCollection.
    }
}
```

User1
User2
User3
User4
UserID
FirstName
LastName
Password
FieldNames

ادامه دارد ...

عنوان: **آشنایی با Oslo - قسمت سوم**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۱۲/۰۴ ۱۴:۲۸:۴۳
آدرس: www.dotnettips.info
برچسب‌ها: Oslo

اگر علاقمند باشید که دو قسمت قبل را به صورت آموزش ویدیویی ملاحظه کنید، لطفا ویدیوهای رایگان زیر را دریافت نمائید. این ویدیوها قسمت اجرا کردن اسکریپت حاصل روی یک دیتابیس را هم در آخر نمایش داده (علاوه بر مدلسازی و نمایش T-SQL حاصل) و شما را با Quadrant نیز آشنا خواهند کرد.

First Look at Quadrant – Oslo’s Modeling Tool

First Look at M – Oslo’s Modeling Language

"Oslo - Data Modelling in "M

Oslo - PDC-08 CSD Bits Review

مآخذ:
biztalkgurus.com و bloggersguides.net