

Lesson 20: Northwind Store

In this lesson, you will continue the development of the Northwind Store application.

In Part 1 of this lesson, you will add some finishing touches to the Product search capability.

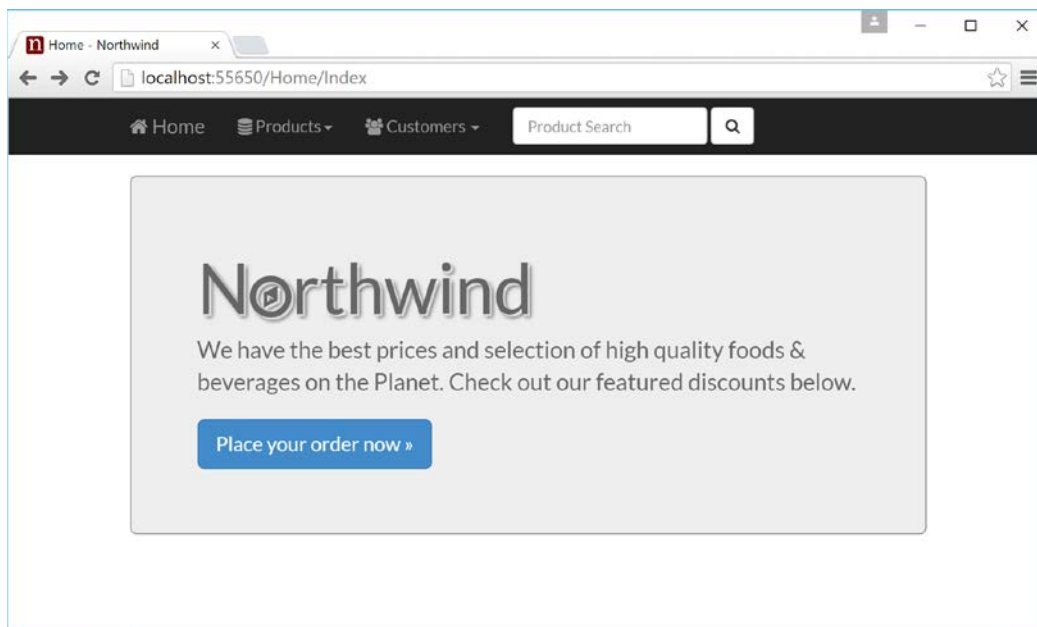
In Part 2 of this lesson, you will add a new table to the database for product discounts. Three product discounts will be featured on the Home page and all discounts will be displayed in a new view.

Agenda

1. Demonstration
2. Lab/Homework

Demonstration – Part 1

1. Review the product search feature.
2. Open the Northwind project in Visual Studio. We are going to add a search box to the navbar. A product search will be accessible from any page in our website (see below).

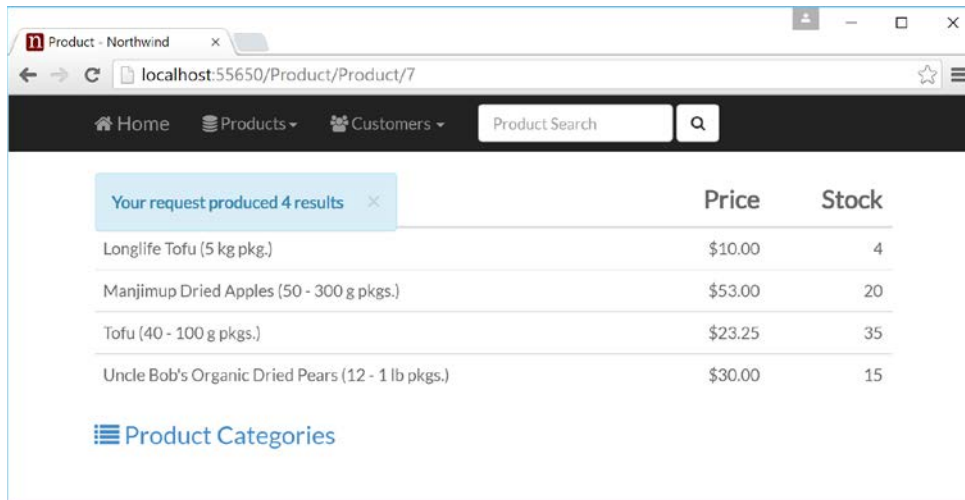


3. Open the `_Layout` view. Add the search form inside the navbar after the Products ``.

```
<form class="navbar-form navbar-left hidden-xs" role="search" method="post"
action="~/Product/SearchResults">
    @Html.AntiForgeryToken()
    <div class="form-group">
        <input type="text" class="form-control" name="SearchString" id="SearchString"
placeholder="Product Search">
    </div>
    <button type="submit" class="btn btn-default"><i class="fa fa-search"></i></button>
</form>
```

4. That is all we need. A product search is now accessible using the navbar. Notice the `hidden-xs` class will hide the form when the navbar is collapsed.

- Next, we will modify the search results view. Let's display the total number of results in a Bootstrap alert (see below).



- Open the Views\Product\SearchResults view. Add the alert right before the table. Test in browser. Notice the alert can be closed.

```
<div id="alert-search" class="alert alert-info alert-dismissible fade in"
style="position:fixed;">
  <button type="button" class="close" data-dismiss="alert" aria-label="Close">
    <span aria-hidden="true">&times;</span>
  </button>
  <strong id="alert-search-msg">Your request produced @Model.Count() results</strong>
</div>
```

- Although the alert can be closed, it is still a bit of a distraction. Let's have it close automatically after 3 seconds. Add the setTimeout javascript method to the jquery ready function.

```
setTimeout(function () { $('<div id="alert-search" class="alert alert-info alert-dismissible fade in"
style="position:fixed;">
  <button type="button" class="close" data-dismiss="alert" aria-label="Close">
    <span aria-hidden="true">&times;</span>
  </button>
  <strong id="alert-search-msg">Your request produced @Model.Count() results</strong>
</div>').alert().fadeOut(); }, 3000);
```

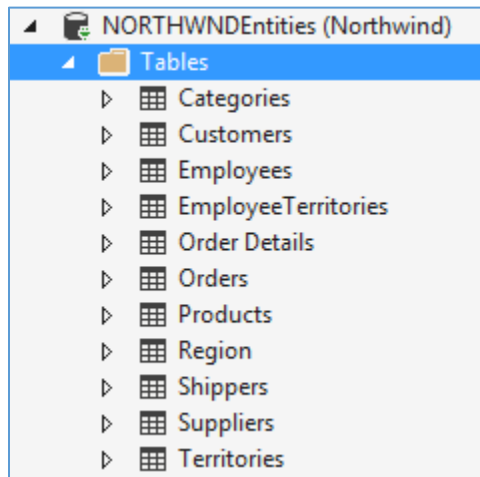
- The setTimeout function executes after 3000 milliseconds. In this case we are calling closing the alert. Test in browser.

Demonstration – Part 2

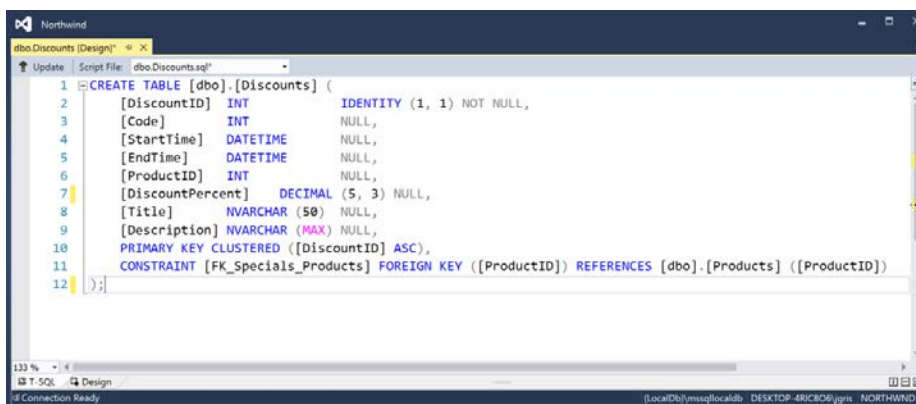
- Let's begin by adding a new table to the database. See below for the table schema. This table contains the ProductID as a foreign key (ProductID is the primary key in the Product table).

Name	Data Type	Allow Nulls	Default
DiscountID	int	<input type="checkbox"/>	
Code	int	<input checked="" type="checkbox"/>	
StartTime	datetime	<input checked="" type="checkbox"/>	
EndTime	datetime	<input checked="" type="checkbox"/>	
ProductID	int	<input checked="" type="checkbox"/>	
DiscountPercent	decimal(5,3)	<input checked="" type="checkbox"/>	
Title	nvarchar(50)	<input checked="" type="checkbox"/>	
Description	nvarchar(MAX)	<input checked="" type="checkbox"/>	

- Open the Server Explorer (View – Server Explorer).
- Expand the NORTHWNDEntities\Tables folder.



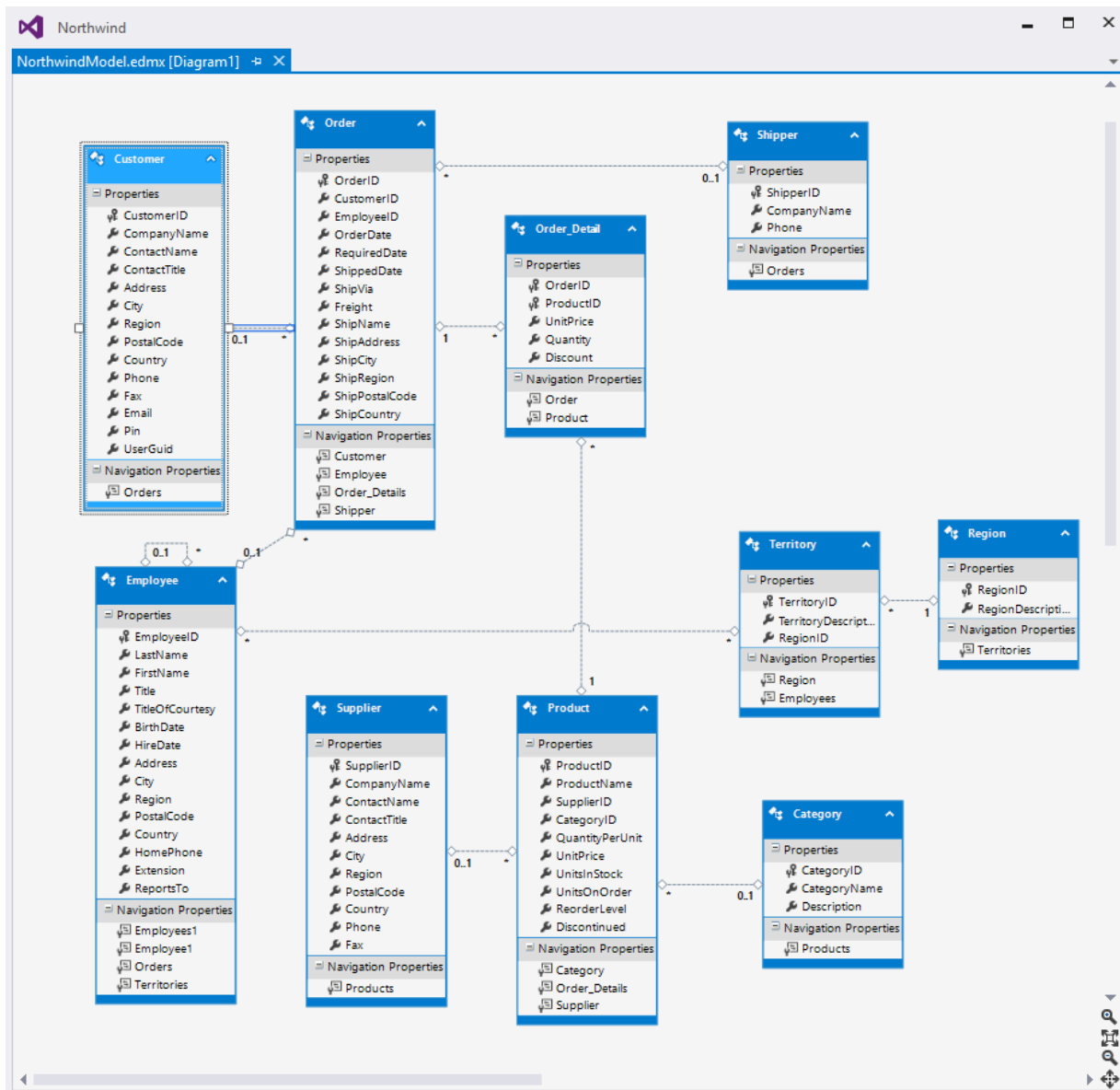
- Open the **Create_Discounts.sql** file in the Editor. It can be dragged and dropped using Windows Explorer.
- You will need to connect to the (LocalDb)\mssqllocaldb server and select the NORTHWND database.



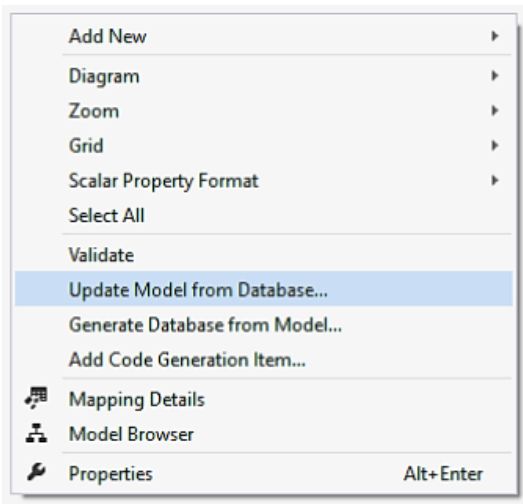
- Click the Green Execute triangle in the upper left corner of the sql window.
- Right-click the Tables folder in Server Explorer and select Refresh. You should see the new table.
- Let's populate the table with some data now. Open the **Populate_Discounts.sql** file in the Editor.
- You will need to connect to the (LocalDb)\mssqllocaldb server and select the NORTHWND database.
- Click the Green Execute triangle in the upper left corner of the sql window.
- Right-click the Discounts table in Server Explorer and select Show Table Data.



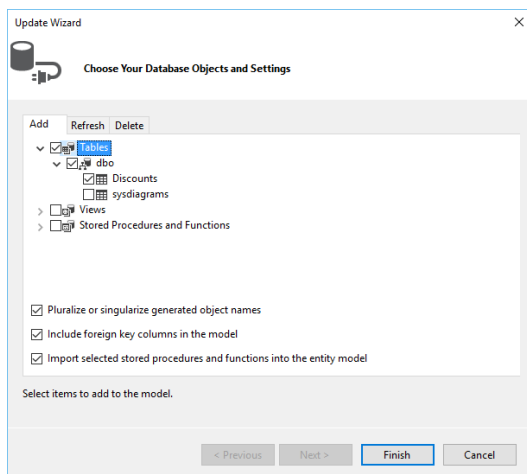
12. Now that we have our Discounts table, we will feature 3 discounts on the Home page of the Northwind store website.
13. Now our Model and database are out of sync. It's not really a problem for our database to have tables, views and stored procedures that are not part of our application's model classes, but if we want to use those tables, etc. we need to update the model.
14. Using the Solution Explorer, Open the Models\NorthwindModel.edmx file.



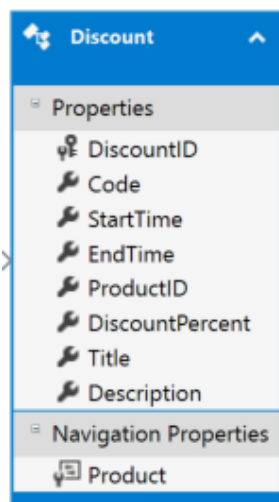
15. Right-click the Editor canvas and select Update Model from Database.



16. From the Add tab, select the Discounts table. Click Finish to update the model.



17. The Discounts table is added to the Model.



18. You MUST save the diagram.

19. Open the Home Controller in the editor. Import the Northwind.Models namespace.

```
using Northwind.Models;
```

20. Modify the Index Controller method.

```
public ActionResult Index()
{
    using(NORTHWNDEntities db = new NORTHWNDEntities())
    {
        // return a list of discounts
        return View(db.Discounts.ToList());
    }
}
```

21. This will pass an unfiltered and unsorted list of discounts to the View.

22. Open the Views\Home\Index view in the editor. Import the Northwind.Models namespace and add the model to the view.

```
@using Northwind.Models
@model IEnumerable<Discount>
```

23. Modify the row div. Notice we are displaying the discount description, the discount expiration data and a link to a bootstrap popover that contains the discount code.

```
<div class="row">
    @foreach (Discount d in @Model)
    {
        <div class="col-sm-4">
            <h2 class="font-md">@d.Title</h2>
            <hr class="hr-special" />
            <p>
                @d.Description
            </p>
            <p><i>Discount Expires: @string.Format("{0:MM/dd/yyyy}", @d.EndTime)</i></p>
            <a class="special" data-container="body" data-toggle="popover" data-
placement="right" data-content="@d.Code">
                <i class="fa fa-hand-o-right"></i> Discount Code
            </a>
        </div>
    }
</div>
```

24. Open the Content\Site.css file. Add the hr-special class selector and the a-special class selector. Test in browser.

```
.hr-special{ margin:8px; }  
a.special{ cursor:pointer; }
```

25. This is not really the desired result. We really only want to feature 3 discounts on the Home Page. Open the Home Controller. Modify the Index controller method's return statement. Test in browser.

```
return View(db.Discounts.Take(3).ToList());
```

26. Notice the popover to display the discount code is not functional yet. We'll need some javascript to enable the popovers. Test in browser.

```
@section scripts  
{  
    <script>  
        $(function () {  
            $('[data-toggle="popover"]').popover();  
        });  
    </script>  
}
```

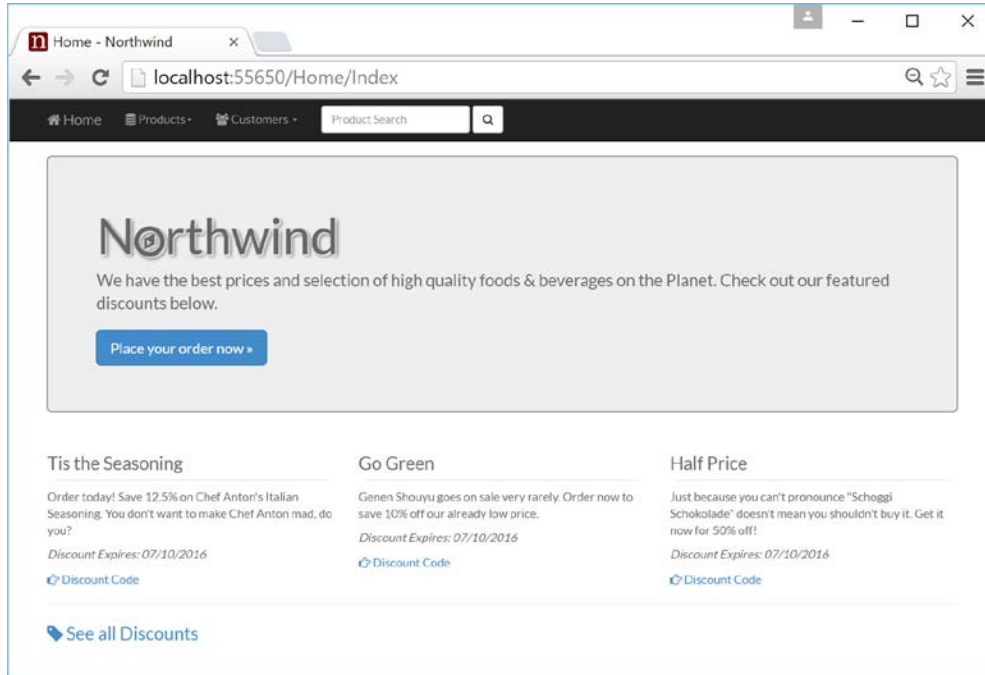
27. This looks good, but we need to properly filter the discounts. Notice that we are including discounts that have expired. We should only display discounts where the current date is within the start and end dates.

28. Open the Home Controller. Replace the return statement. Test in browser.

```
// Filter by date, return the first 3  
DateTime now = DateTime.Now;  
return View(db.Discounts.Where(s => s.StartTime <= now && s.EndTime >  
now).ToList().Take(3));
```

Homework

Your homework is to update the Product\Discount controller method & the Product\Discount view to display all product discounts where the current date is within the start and end dates. Start by adding a link at the bottom of the Home Page Display All Discounts (see below).



Then update the Product\Discount controller method & the Product\Discount view to display the discounts in a Bootstrap list-group. Include a link back to the home page.

