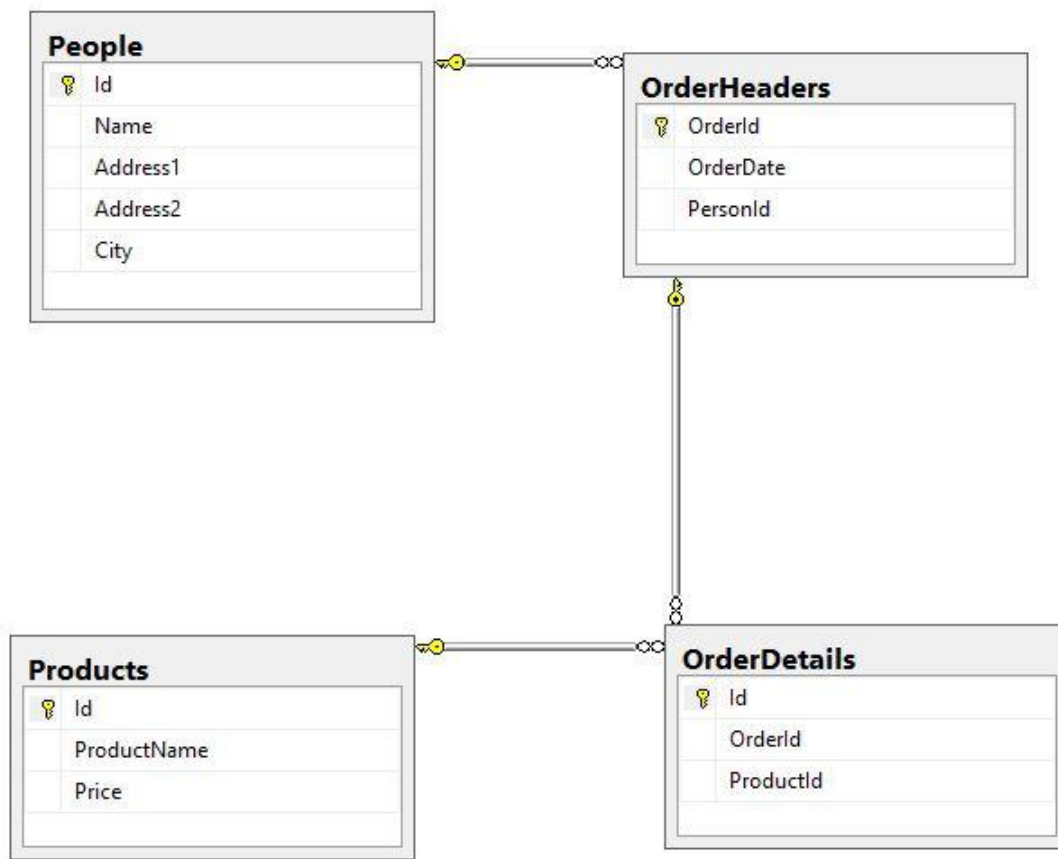


Cobra Project Onboarding Database



Onboard Solution Layout

You can follow the instruction below as a guideline

Backend

- Onboard.Controllers - Contains Customer or Order Controller
- Onboard.Model - Contains Database Model

Frontend - AngularJS - You can separate each of the modules below in separate files and reference them in `_layout.cshtml`

- App/controllers.js - AngularJS Controllers
- App/services.js - AngularJS Custom Services
- App/directives.js - AngularJS Custom Directives
- App/onboarding-login-app.js - AngularJS Module

App/onboarding-login-app.js contains the angularjs module named Onboard App. Use this in your controllers, `ng=app` is defined in the `_layout.cshtml` found in the shared views folder.

Please complete the following exercises using AngularJS and AngularJS validator.

1. Create the Database name Cobra-onboarding-db from above schema using SQL Management Studio
2. Create Web MVC Application names Cobra-onboarding
3. Implement Cobra-onboarding-db to Cobra-onboarding application using EF
4. You can use \$http.get and \$http.post in your controller

```
var onboardapp = angular.module('onboardapp', ['ui.bootstrap']);
onboardapp.controller('Customer', '$scope', '$http', function (scope, http) {
    var req = {
        method: 'POST',          ---> This can be POST or GET
        url: '/Customer/Index',  ---> This is your backend function
        data: { id: id }         ----> This is the parameter you sending to backend
    };

```

```
    http(req).then(function (response) {
        if (response.data.success) {
            // do something
        }
        else { //else do something
        }
    }); // end of http post

```


```
}); //end of controller

```

5. Create a page in a MVC Project showing a list of Customers in a table with the following columns. Name,Address,Town/City. Create a page using AngularJS and AngularJS validator to Add/Edit a customer.


Customer				
Add				
▼ Name	▼ Address1	▼ Address2	▼ Town/City	▼ Action
Tom	53 Sadi Street		Auckland	[Edit] [Delete]
Peter	12 Sadi Street		Waitoko	[Edit] [Delete]
Jack	19 Bollard Street		Otago	[Edit] [Delete]
Mary	8 Sadi Street		Wellington	[Edit] [Delete]

5.1 For Add, Edit you can either use popup modal -
A working example of popup modal can be found in this
<https://plnkr.co/edit/BvNAw51AmmTPjWYSLrYf?p=preview>
Add



A modal form titled "Create" with a light gray border. It contains four text input fields stacked vertically, each with a label to its left: "Name", "Address1", "Address2", and "Town / City". The input fields have a light gray background and a thin border. At the bottom of the modal, there are two buttons: "Save" and "Cancel", both with a light gray background and a thin border.

Edit



A modal form titled "Edit" with a light gray border. It contains four text input fields stacked vertically, each with a label to its left: "Name", "Address1", "Address2", and "Town / City". The input fields have a light gray background and a thin border. At the bottom of the modal, there are two buttons: "Save" and "Cancel", both with a light gray background and a thin border.

6. Change your \$http get and \$http post using services
https://www.w3schools.com/angular/angular_services.asp

Create Your Own Service

To create your own service, connect your service to the module:

Create a service named **hexafy**:

```
onboardapp.service('hexafy', function() {  
  this.myFunc = function (x) {  
    return x.toString(16);  
  }  
});
```

To use your custom made service, add it as a dependency when defining the filter:

Example

Use the custom made service named **hexafy** to convert a number into a hexadecimal number:

```
app.controller('myCtrl', function($scope, hexafy) {  
    $scope.hex = hexafy.myFunc(255);  
});
```

7. Create a page to list the order by date. With add/edit a sales order.

Orders					
<input type="button" value="Add"/>					
▼ Order Id	▼ Date	▼ Customer Name	▼ Product	▼ Price	▼ Action
1	12/03/17	Jack	Books	\$5.50	[Edit] [Delete]
2	15/03/17	Tom	Pencil	\$1.20	[Edit] [Delete]
3	17/03/17	Vincent	Rubber	0.50	[Edit] [Delete]
4	20/03/17	Jack	Rubber	0.50	[Edit] [Delete]

Add

Create	
OrderId	<input type="text" value="Id"/>
Date	<input type="text" value="Date"/>
Customer	<input type="text" value="Options"/> ▼
Product	<input type="text" value="Options"/> ▼
Price	<input type="text" value="Price"/>
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

Edit

Edit

OrderId

Id

Date

Date

Customer

Options

▼

Product

Options

▼

Price

Price

Save

Cancel

Note:

- OrderId can read only and read from the database
- Upon choosing the Product it automatically select the price

8. Create a menu for Customer and Order under the shared Layout.