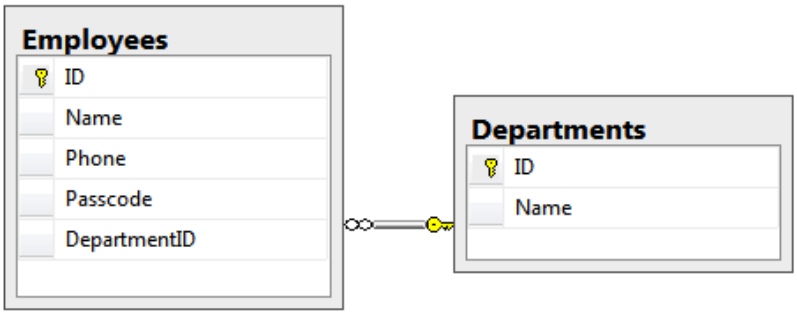
LAB 3

You will use the same web site you submitted for Lab2 for this lab. For this lab, we are going to upgrade all pages so they interact with a database.

First, you will need to create a database with two tables. Once completed, you should have two tables with a relationship between them as shown. For those who missed the class where we created the database.



The database definitions are as follows:
Employee:

Column Name	Data Type	Allow Nulls
ID	int	☐
Name	varchar(50)	☐
Phone	varchar(10)	☐
ITPasscode	varchar(10)	☑
DepartmentID	int	☐
Email	varchar(50)	☐

Department:

Column Name	Data Type	Allow Nulls
ID	int	☐
Name	varchar(50)	☐

NOTE: Make sure both ID's are specified as identity so they auto increment for you.

NOTE #2: Make sure the fields all match the name & data type, including the 10 character lengths for both Phone & ITPasscode.

As you recall from Lab2, we have a web site which provides 4 links (All.aspx, Manage.aspx, Department.aspx, Contact.aspx) and should all share a master page which provides a header to click to each of them as shown below.

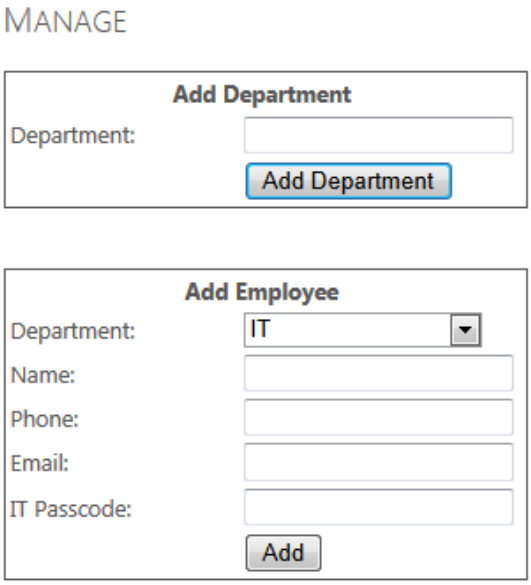
Rename the tab containing the word “Lab2” and change it to “Lab3”. You will find this on the main master page.

Manage.aspx page

In the previous lab, we created Employee objects and added them to the session. For this lab, you will modify this code so the employees are added to the database we created using the Entity Framework data model for the database as shown in class. In addition, we will add a new field for **Email** and we will also add the ability to add departments to the database as well.

NOTE: The departments drop down list will need to be populated from the database. I recommend first making sure you can add departments first. Afterwards make sure the dropdown list shows all the departments available in the **Departments** table.

The validations and rules for showing the IT passcode for IT departments should remain just like we specified for Lab2.



The new email field should be required and should also validate to valid email address.

The validation rules from Lab2 should all still apply and are included here:

- *All visible fields need to be required*
- *Phone must be entered in the dash format (333-444-5555)*
- *Names must contain only letters and consist of two parts with only once space in between (Peter Piper)*
- *Both client and server validation needs to be applied.*
- *All Validation controls should use the css class “lab2Validation” which should make them*

appear in red & italic.

Once you create the entity framework for you database model, you will automatically have an Employee object available for you to use similar to the code below:

```
Employee e = new Employee();
```

NOTE: If you do not see the Employee object and you've already added the entity framework to the project pointing it to your database, then you need to make specify the namespace for the entity framework and then include the same namespace in your code behind file.

NOTE #2: Even though the phone is collected with dashes, you will want to remove the dashes when storing the phone number into the database as the phone field should only have 10 characters to store.

NOTE #3: Since entity framework will automatically create an Employee class for you for the Employee table in the database, you will need to remove the Employee class provided for you in Lab 2 located in the App_Code folder.

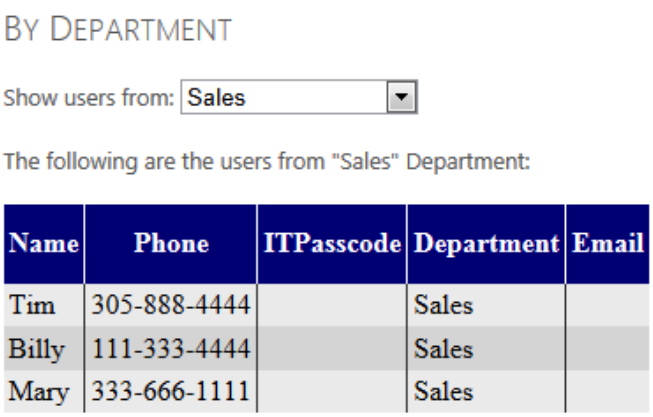
All.aspx

Will list employess in a GridView control from the database. In class we showed two ways you can do this, either through the code behind or through design time data sources. As an example, here is a sample grid view control with some formatting.

Name	Department	Phone	IT Passcode	Email
John	IT	305-777-1111	1000	
Tim	Sales	305-888-4444		
Roger	Service	786-555-9999		
Peter	Telemarketing	305-348-1100		

Department.aspx

Department tab should have a dropdown list listing the departments. Similar to the add.aspx page. This dropdown list should be populated from the departments listed in the database. It however will have autopost back enabled. When a department is selected, the grid view below it will be updated to reflect the employees who are part of that department. A sample screen shot is shown below:

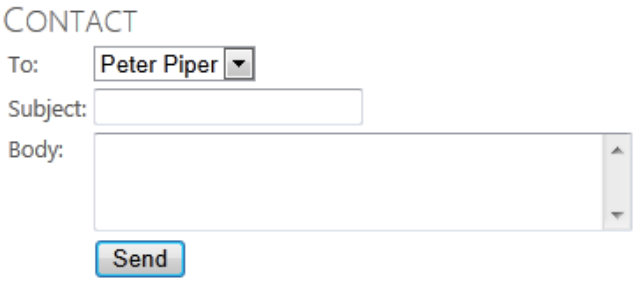


Contact.aspx

The contact page is very simple. You have to create a form similar to the one below. The dropdown list for the **To** field should list the Employees listed in the database. Both subject and body need to be required fields. When the send button is clicked, an email is sent to the selected employee. We will cover this in class Thursday.

NOTE: To ensure an email is delivered, make sure you specify the from user as you did in the homework.

NOTE #2: To ensure you do not spam anyone by accident, make sure all the employees you add have your email address for testing.



Check List of Things To Complete & Point Breakdown

- **Database (15 pts)**
  - field names and length all match the screenshot
- Rename tab to Lab3 (**5 pts**)
- **Manage.aspx (20 pts)**
  - ability to add both IT and non-IT employees to the database (10)

  - add departments to the database (5)
  - validation from lab2 remains (1)
  - validation for the new email field (both required & correctly formatted email) (4)
- **All.aspx (20)**
  - should list all the employees in the database as shown
- **Department.aspx (20)**
  - department dropdown list should show all the departments in the database (5)
  - when department is selected, it should list all the employees in the database for the selected department. (15)
- **Contact.aspx (20)**
  - The To dropdown list should list all the employees in the employee table (5)
  - The form should have all the fields required (5)
  - The form, when submitted should email the selected employee with the body & subject provided from your email address. (10)

Extra credit of 10 points are applicable to erase any missed points above if you use a single user control for the GridView and use the user control on both the all.aspx and department.aspx pages.