Hi Team,

Due to lesser time, unfortunately not able to complete the end-to-end task. However, I have created high level MVC application and database with necessary tables.

These are steps I followed on the given Scenario:

I followed a DB First approach.

**Step 1:** Created a database with the name “EmployeeDatabase”

* + Created a table with “Employees” having EmployeeID as a Primary Key
  + Created a table with “EmployeePhones” having EmployeePhoneID as a Primary Key
  + Created a table with “EmployeeAddress” having EmployeeAddressID as a Primary Key

**Step 2:** Inserted the data into all the tables

**Step 3:** Created the Stored Procedure “spGetAllEmployees” by joining all the 3 tables to satisfy the 1st scenario.

Note: Order by Clause is not included in SP and supposed to be handle thru a parameter from User Interface. But this is not incorporated at this moment.

SELECT DISTINCT

emp.EmployeeID,

emp.EmployeeNumber,

emp.FirstName,

emp.LastName,

emp.HireDate,

empPh.PhoneType,

empPh.PhoneNumber,

empAdd.Address1,

empAdd.Address2,

empAdd.City,

empAdd.[State],

empAdd.ZipCode

FROM Employees emp

JOIN EmployeePhones empPh ON empPh.EmployeeID = emp.EmployeeID

JOIN EmployeeAddress empAdd ON empAdd.EmployeeID = emp.EmployeeID

Step 4: Created an ASP.NET MVC application.

* Created a “Employee.cs” by defining all the properties
* Created a “EmployeeDataAccessLayer.cs” by defining GetAllEmployees() method to get the all the employees by incorporating the ADO.Net commands.

Step 5: Created an EmployeeController controller by calling the Model class method GetAllEmployees().

Before that EmployeeDataAccessLayer class has been initated.

Steps 6: Created the Employee folder in Default View folder

Index.cshtml MVC view page has been included to display Employee’s Data.

On top of this page, We are enumerating the Employee Model and iterate the model for each employee record.