Exercise 06 Multi Record Query

Fork the repo called “WebAppNW02-MultiRecord” from “RobbinLawCPSC1517/” to your github account.

Clone the newly forked repo from your account into Visual Studio.

Restore the packages, and then restart the Visual Studio IDE.

Change the name of the solution form “WebAppNW02-MultiRecord” to “WebAppFSIS”.

Run the code to make sure it works before making changes as per Exercise 05. When it works commit and push back to your github repo.

For this exercise replace the Entities that are already in place as described below. Remember to look at the actual database data types and reflect them properly in the Entity definitions. Also remember that when you define something with say “int?”, this means that the database field in the actual database is NULLABLE. Strings don’t need this and of course the primary key as it will never be nullable.

In the Entities01.cs class file change the Categories definition of the Northwind database to the Teams definition of the FSIS database.

In the Entities02.cs class file change the Products definition of the Northwind database to the Players definition of the FSIS database.

In the Entities03.cs class file change the Suppliers definition of the Northwind database to the Guardians definition of the FSIS database.

For this exercise change the name of the database from “NWDB” to “FSIS\_db” in the Context.cs file.

For this exercise change Controller02.cs to reflect the changes from the Northwind database to the FSIS database. This means calling the right Stored Procedure.

In the ExcercisePages/SimpleQuery Form add the tags necessary to present labels, a textbox, and a button, for a simple PKey Query, similar to the Northwind Demo, but for the Team Table of the FSIS Database instead of the Region Table of the Northwind Database.

In the SimpleQuery code behind add the code necessary to make everything work when the button is pressed. This code will be similar to the code behind in the Northwind Demo. Remember to put in the proper using directives, so your code can access the BLL, and ENTITIES code.

Test your code and when it works, save it to GitHub.

At this point you can demonstrate all of Exercise 05 to your instructor for evaluation.

FSIS Client-Server Query – MultiRecord Query using Code Behind

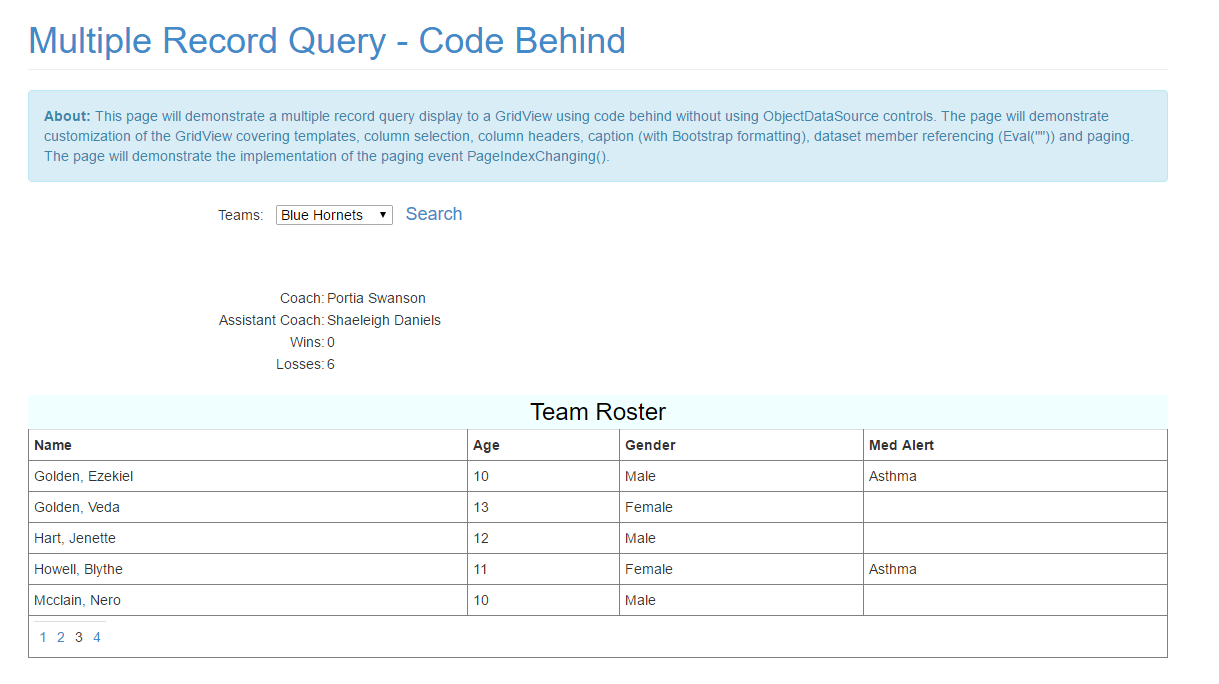
Displaying Multiple Rows

Modify the Visual Studio solution from your previous exercise, to add a form to display multiple rows of data using a GridView control. Base it upon the screenshot below. Perform the lookup of data for the GridView by the DropDownList in the code-behind of the form. All code to load the DropDownList and GridView is to be done in the code behind, **NOT** using an ObjectDataSource. Include a prompt line for the DropDownList.

This exercise will require you to make 2 separate queries to the database under the one button event. One query will retrieve the Team data record, and then display the data in individual fields. The second query will retrieve all the team player records (List<>), and then assign the collection to a GridView. You will need to customize the GridView for column headers, columns displayed, caption and paging (5 items/page). Use templates for customizing your GridView columns.

Create the following:

* BLL
  + Create the PlayerController, TeamController and GuardianController BLL Classes
  + Create the following method in the appropriate TeamController BLL Classes
    - Team\_List() which will return a List<T> containing all the Team records
    - Team\_Find(int teamid) which will return a single Team record matching the input parameter
  + Create the following methods for the PlayerController
    - Player\_GetByTeam(int teamid) which will return a List<T> containing all the Player records for that team



Team with no players should have a EmptyTemplate message shown.

