# *Programming IV (420-B42-HR)*

# *Lab 6 – Database Connections*

Date assigned: Monday, March 13, 2017

Date due: **Monday, March 13, 2017, 11:00 a.m.**

**Learning Objectives**

Upon successful completion of this lab exercise, the student will be able to:

* Connect to an Oracle database from a WPF project
* Read, update, delete and insert records in Oracle database from a WPF Application

Lab Set-Up

1. Make sure you name the controls properly. See the Moodle page for standards on how the controls are to be named.
2. There is an example on the Moodle page (the example from class); however, rather than just cut and paste, try to do it yourself so that you understand what is happening.

To Do

**Database Libraries**

1. Open Visual Studio and create a new C# WPF project solution called usernameB42L06. Add a second project to the solution called B42L06DB. Add a third project to the solution called B42L06BLL.
2. In the DB project, add a data source which accesses the Naman-Naman (NN) tables. The data source should include all the tables. The dataset should be called dsNaman.
3. Add data tables and queries as follows:
   1. Select all employees for a given department NAME
   2. Select all employees whose last name is like a name passed as a parameter
   3. Find the number of employees in a given department
   4. Find the sum of the salaries of all employees
   5. Find the sum of the salaries of all employees in a given department
   6. Find all employees with a salary above a given amount

The queries can be in the automatically created data tables (like NN\_EMPLOYEE) or in a new data table that you create. The design is up to you. I, generally, include new queries in existing data tables if they use only that table. If there is a join, I would include it in a new data table. That being said, the design is up to you.

Each of these queries should be a separate method in the class library with an appropriate name. The methods will return either a data table or scalar value depending on what they are returning.

This will be your data access layer. The class should be called after the business class and have the last two characters of DB to indicate that they access the database. Since the above are all related to employee the class should be EmployeeDB.

1. In the BLL class, add a class called Employee with the appropriate fields. You may want to add a couple of other classes, but that, once again, is up to your design. Along with the constructor(s), getters and setters, these classes should contain methods that call each of the database methods in the corresponding database class. These methods will return either a scalar value or a <list> of objects to the WPF project for displaying. These methods will receive an instance of a class or a scalar value that will be used in calling the DB method. You will likely want to have one or two private methods to convert a data table into an employee class.

When displaying the information about the employee you should not display the ID values, but should join to lookup the string values that match the IDs. You can do the join in your select statement or using multiple database queries (I would suggest the former).

Remember that you will have to include a reference to the DB project in your BLL project and add a using clause at the top of your Employee class before you can access the classes and methods.

1. In your WPF project, create a page with a menu (Menu) or tabbed interface (TabControl) that allows you to display the results of each of the queries. Sometimes (for queries that require a parameter) this will mean you will need input fields and sometimes the query will have no parameters.

The results should be on separate panels organized appropriately. For example, the first two queries can be on the same panel and the other four on a different panel. Use some basic design to make the application usable.

**To submit**

When you have completed the lab exercise, create a single zip file called YourUserNameB42L06.zip and copy the file to the Moodle page for the course.

The NN Database Tables:

