PROG1121K – DB Programming **Lab 4**

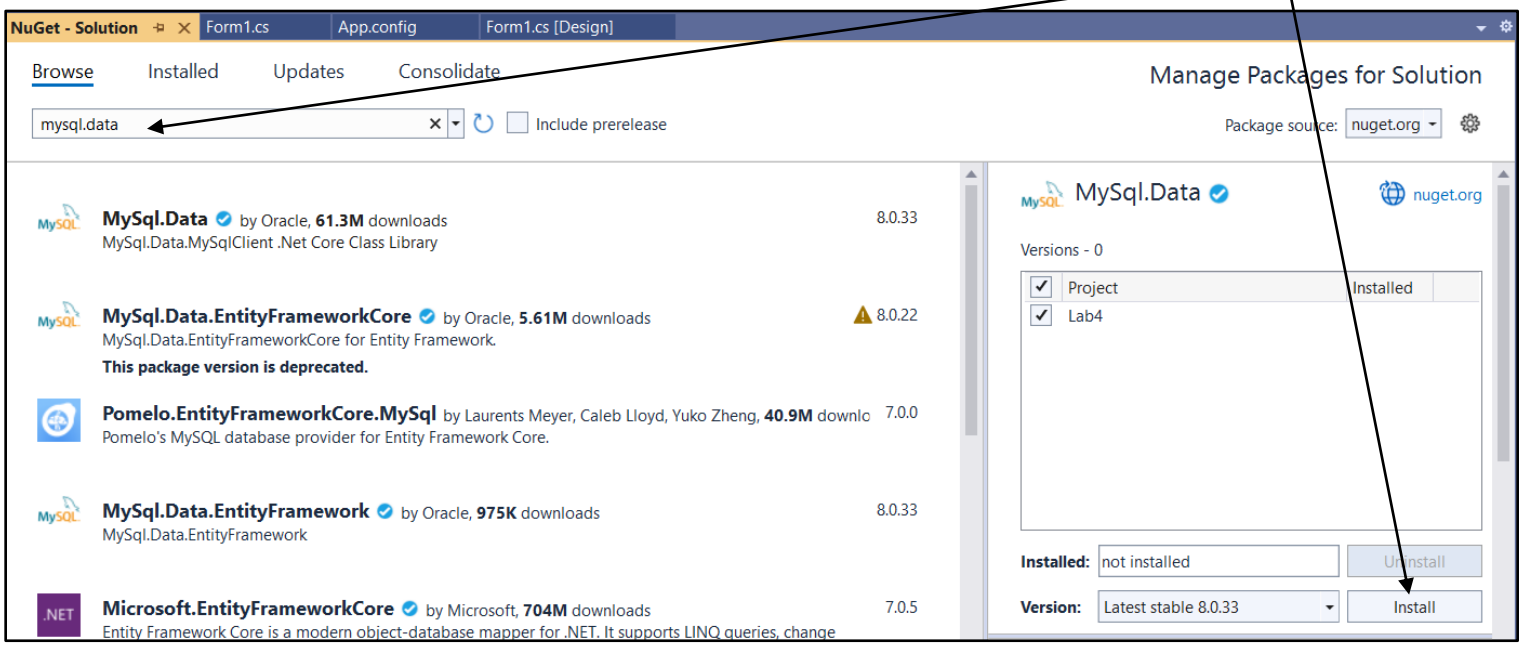
Due Date: Monday, June 5, 10:30 am **Marks: 35 (rubric)**

This lab covers connecting through code to a MySQL database, running queries and updating (update, insert, delete). **Select this link for the video run: https://youtu.be/u4sgYqM3weo**

1. Download the script **emp.sql** and execute in MySQL. You can create an ERD to help with the lab in WorkBench.

2. Download and extract **Lab4.ZIP**. Look over all controls. The combobox has items created. The listboxes will be populated by running queries. Notice the script is found within the **bin\debug** folder of your solution.

3. Install the **MySql.Data** package called **MySql.Data**
   Select **Tools/Nuget Package Manager/ Manage Nuget Packages for Solution. Browse** and **Install**:



4. Add the following using directives:

```
using MySql.Data.MySqlClient; //handles all MySql classes
using System.Configuration;   //for app.config having connection string
using System.IO;              //for files
```

5. Place the connection string in the app.config using the **name** option. Syntax given here (**yellow**: replace):

```
<configuration>
     <connectionStrings>
       <add name="SomeName"
  connectionString="server=localhost;database=dbname;userid=uid;password=pw" />
     </connectionStrings>
  </configuration>
```

6. **Class-level:**
   - ✓ Declare an output text file using **StreamWriter** named **LastnameLog**. Do not use a path and the file will be stored in the **bin\debug** folder. This file will have your name and date and **all SQL commands that are executed** during the run of your program (see sample at bottom).
   - ✓ Create the connection string using the configuration manager. Syntax given here:

```
static string connStr =
ConfigurationManager.ConnectionStrings["SomeName"].ConnectionString;
```

   - ✓ Create the connection to the connection string.

7. **Form Load**:
   - ✓ Run the script found in **bin\debug** (need to start over with database as changes are happening). Syntax given here (**yellow**: replace):

```
string script = File.ReadAllText("filename.sql");
 MySqlCommand command = new MySqlCommand(script, nameofconnectionstring);
command.ExecuteNonQuery();
```

   - ✓ Delete the table **families** from the database.
   - ✓ **View tab**: Populate the listbox with all tables found in the database by running a command.
   - ✓ **View tab**: Set the start date to the earliest **hire_date** found in the employee's table and the end date to the latest **hire_date** found in the employee's table.
   - ✓ **Revise tab**: Populate the listbox with all unique titles found in the titles tables sorted by title.
   - ✓ Fix the invalid **to_date** entries in the following tables: dept_emp, dept_manager, salaries, titles. An invalid date is anything after today's date. Set the invalid dates to today's date.

8. **View Tab**
   a. **Listbox Selection Changed:** Set the combobox for searching to display nothing as tables get selected. Run a query to show all fields and records for the table selected. Display the binding navigator. Use the pseudocode here to link the binding navigator to the datagrid information:
      - Create a binding source (class **BindingSource**)
      - Set the data source of the binding source to the data table that contains the results of the query
      - Set the datagrid data source to the binding source
      - Set the binding navigator binding source to the binding source created

   b. **Combobox Selection Changed**: set the listbox to not have anything selected. As the user selects different searches, do the following:
      - Average Salaries: for each emp_no (employees table), list their name and average salary (salaries table). Check screenshots for column headings and formatting. Sort by name ascending.
      - Hired Between…: use the date picker to list emp_no, name and hire_date for employees hired between the start and end dates (employees table only). Sort by hire date ascending.

9. **Revise Tab:** as different radiobuttons are selected, the corresponding groupbox is enabled for use.
   a. Add a Department: The department numbers start with **d0** (2nd character is a zero). User selects from numericupdown for last two digits. The department name must be at least 5 characters. Ask the user to confirm the addition and add the record to the departments table. Display confirmation if added.
   b. Remove a Title: the unique titles are populated on form load. When the user selects a title, the employee numbers who have that title appear in the other listbox. To delete a title, the user must have a title and an employee number selected. Ask the user to confirm the deletion and delete the record from the titles table. Display confirmation if deleted.

10. You **MUST** create methods for repeated code. Attempt to create methods sending in the query string only.

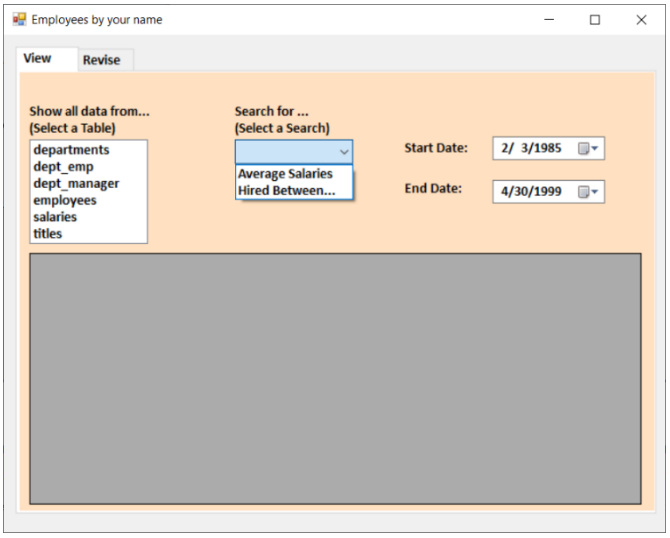
11. **Exception Handling must be on all SQL commands executed against the database**: You must handle exceptions with message boxes that display the exception code sent by the program and your solution (ie. Closing form, Reset to All Records) and a consistent title.

12. **Comments:** Header at top of program, all methods/ events with comments.

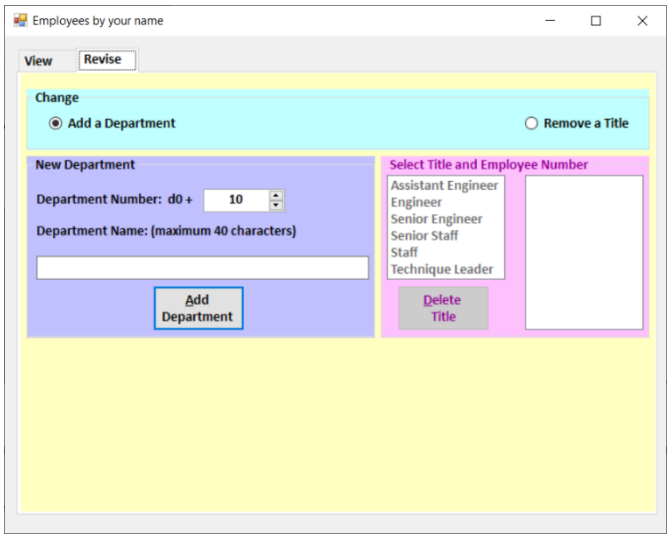
Submission Requirements: (3 files to Brightspace)
1. **Word document: Memo:** paragraphs explaining problems, time spent. Code with event/ methods highlighted
2. **Zip**: zip of entire project renamed **LastnameLab4.ZIP**
3. Text file created: **LastnameLog.txt**

Screenshots

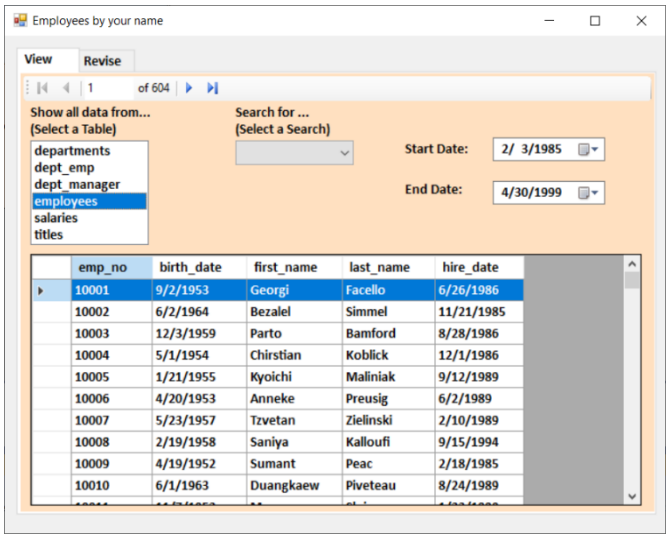
Form Load: View tab
(list of tables, dates updated)



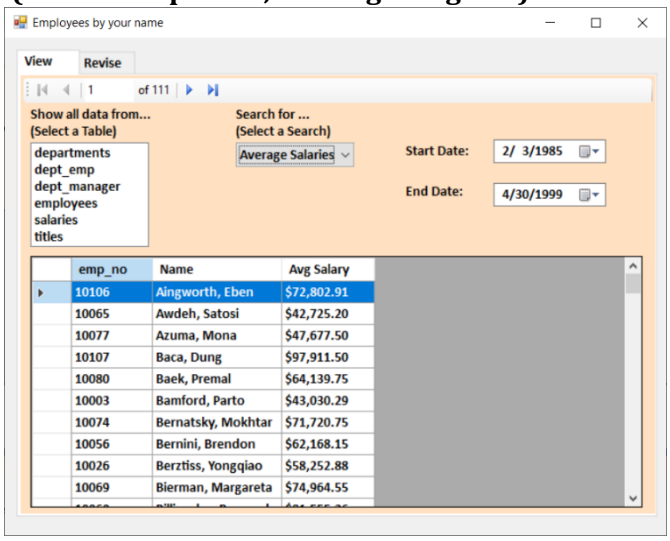
Form Load: Revise Tab
(list of titles, New Department group enabled)



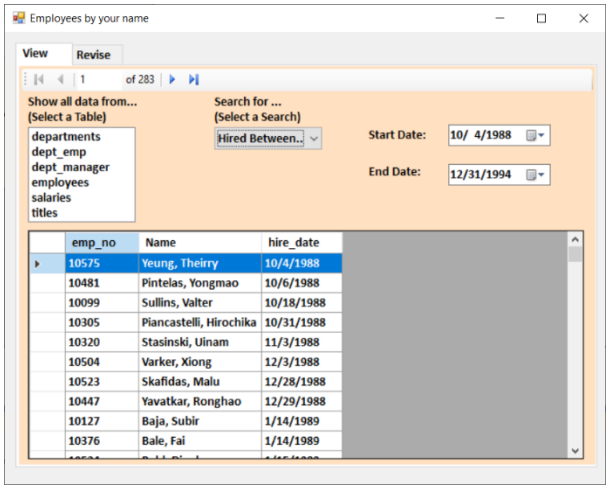
View Tab: select table
(DataGrid updated, binding navigator)



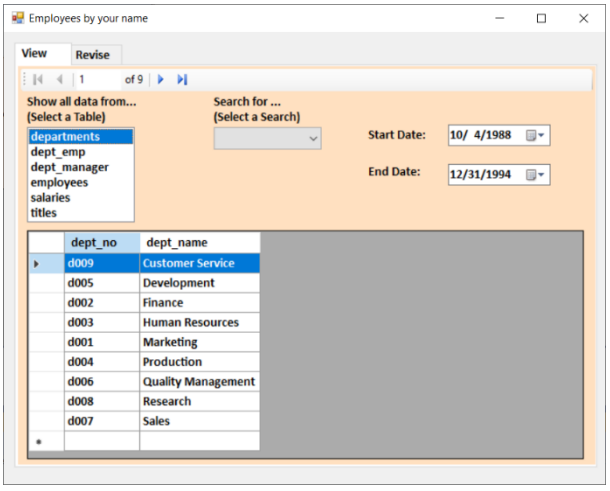
View tab: Select Average Salaries
(DataGrid updated, binding navigator)



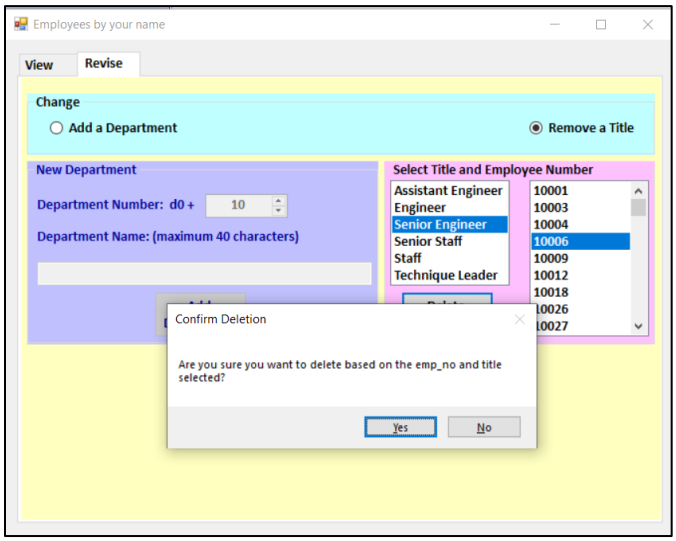
View Tab: Select dates and Hired Between



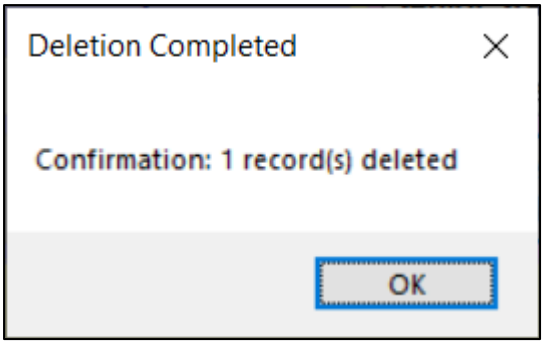
View Tab: select Departments table



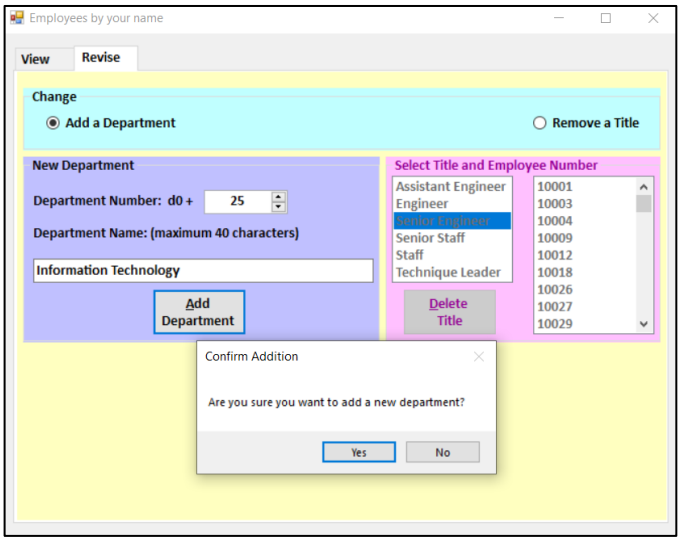
Revise Tab: Remove a Title, select Title and Number
Click Button Delete Title



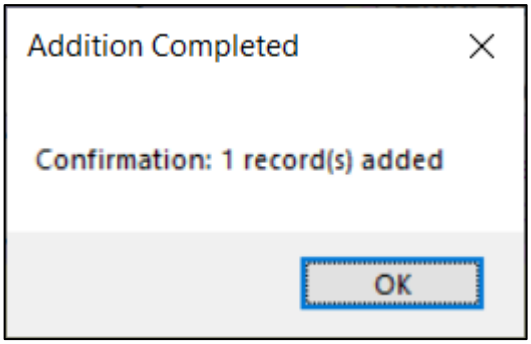
Yes (confirm deletion)



Revise Tab: Add a Department, select 25 and type name
Click button Add Department



Yes (confirm addition)



Log file as program runs: