Fall 2013

SOFE 3700U Data Management Systems

Lab # 6: SQL Queries Using Visual Studio

Submission Type: **Individual Work**

Objectives:

- This lab will enable you to write SQL queries in Microsoft Visual Studio
- Understand how to use aggregate functions and sub-queries
- How to use EXISTS, JOIN, and UNION

Important Notes:

- Save all your lab-related files as you may need them for future labs.
- Once you are done with your work, ask the lab instructor to check your work to assign you a mark.

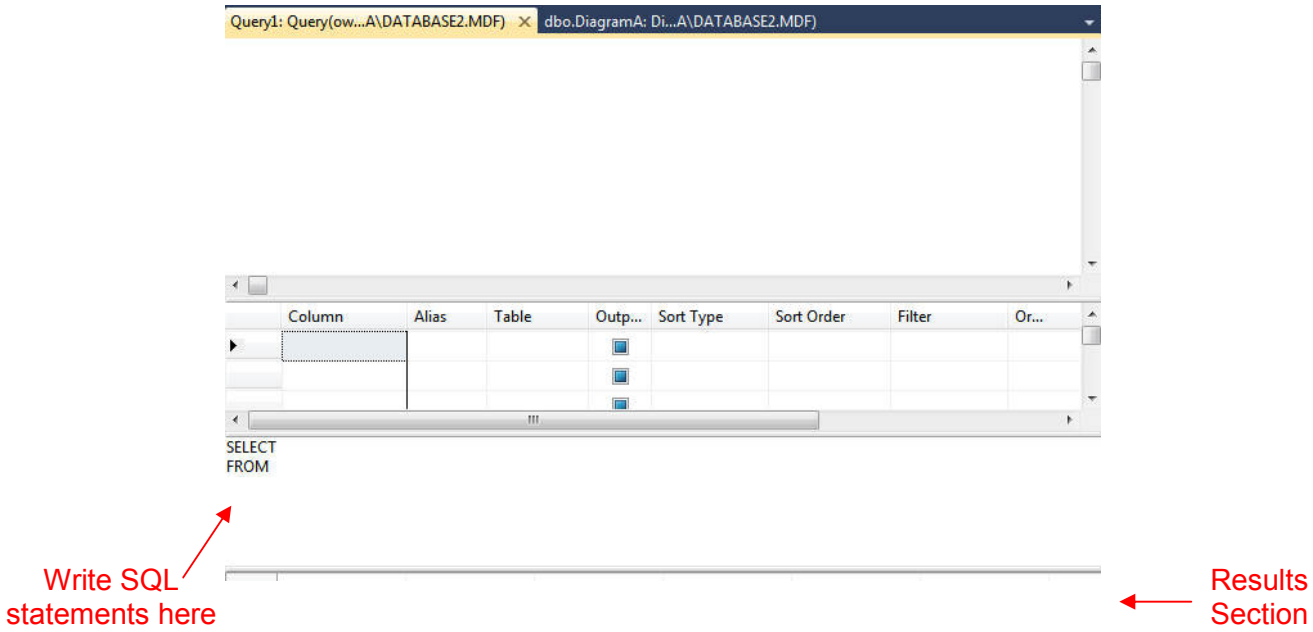
What to submit:

- No written report or online submission is required for this lab.
- To receive marks for this lab, ask Lab TA to grade your work after finishing the steps and answering/demonstrating the Steps and questions throughout the lab.

Scenario: DatabaseManage Inc. has been very happy with your work on the creation of the backend corporate website. The company now wishes to proceed with the second step in which you will be responsible for creating the SQL queries that the company believes satisfy the requirements for the project. You will be responsible for writing and executing several SQL queries using Microsoft SQL query designer.

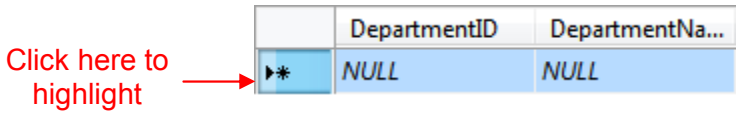
Preparatory Steps: Open the Web Site folder that you created in Lab 5.

- Make a copy of the folder of your Lab5 ASP.NET website. Rename folder to Lab 6
- Open **Microsoft Visual Studio**
- Click on **File → Open → Web Site** → Select **folder location of Lab 6** → Click **Open**.
- For your reference, the database has the following schema:
    - Employees(EmployeeID, EmployeeName, Department, Telephone, Extension, Title, Email, Salary)
    - Departments(DepartmentID, DepartmentName)
- On the Solutions Explorer, expand the App_Data folder. Then double click on database.mdf. The view will switch to the Server (or Database) Explorer
- Expand the Tables section. You should see Employees and Departments
- Right-click on the table named Employees and select New Query. A pop-up window may appear requesting addition of tables. Do not add any tables and click Close button.
- The built-in SQL query builder window will open as shown below:



- Delete all the records in both tables: Employees and Departments. To delete, you can either type the following SQL statements: **DELETE FROM EMPLOYEES** and **DELETE FROM DEPARTMENTS**. Execute these two queries:
    - Executing an SQL statement can be performed by any of the following
        - Clicking on the  icon on the Query Designer toolbox or
        - Right clicking on an empty space in the query builder and selecting Execute SQL or
        - Pressing Ctrl+R

- Add a new column to the Employees table called Salary and set the data type to integer.
- Open the Departments table data (right-click on table name and then select Show Table Data). Open the **departments.txt** file from Blackboard (Content → Labs → Lab 6 → departments.txt)
    - Select the first row on the table by clicking on the left-hand empty column (where the arrow is pointing)



    - Copy all the text from departments.txt and paste in the Show Table editor (simply do a copy/paste action). Then **save** the table.
    - Perform the same steps for Employees table: open **employees.txt** from Blackboard and then copy paste into Employees table.

You are now ready to write and execute SQL queries.

***IMPORTANT**: You will be requested to write eight SQL queries/statements in the following steps. It is recommended that you create a notepad file to save all your SQL statements for each step so that you can use them later to present to the TA for grading.*

Step 1: Review of Aggregate Functions and Subqueries

- **SQL STATEMENT 1:** Create an SQL query to determine the average salary of all employees who work at DatabaseManage Inc. Use the Employees table for this statement. Name the column of the result as **Average_Salary**
    - Execute the SQL query and verify the results in the results section
- **SQL STATEMENT 2:** We would like find all the employees who earn more than the average salary across all departments. Use nested or subqueries to perform this action. Return all employee information in your results.

- **SQL STATEMENT 3:** We would like to change SQL statement 2 to find all the employees who earn more than the average salary in their own department. Use nested or subqueries to perform this action. Return all employee information in your results.

- **SQL STATEMENT 4:** List all the employees who are either in the same department as “Dave McCain” or have a salary same or greater than “Mary West”

Step 2: Using EXISTS, JOIN, and UNION

- **SQL STATEMENT 5:** Using the EXISTS function, list all Software Development employees working at DatabaseManage Inc. Use subqueries to answer this question.

- **SQL STATEMENT 6:** Using an inner join, list all employees working at DatabaseManage Inc. Return the employee name and phone number. (After executing the statement, notice the addition of the word INNER JOIN in the SQL statement by Microsoft Visual Studio automatically)

- **SQL STATEMENT 7:** Using a RIGHT join, list all departments names stored in the Departments table and the corresponding employee's name information working in these departments. (HINT: One department – Marketing - exists but has no employees).

- **SQL STATEMENT 8:** Using UNION, **l**ist all Software Development and Accounting employees' names and salaries. (You do not need to use nested queries. To simplify, compare using department identifier → i.e. 300 and 400).

At this stage, ask the lab instructor to check your results by demonstrating the results for SQL Statements 1-8.