



SOFE 3700U Data Management Systems

## Lab # 5: Creating Databases in Visual Studio

Submission Type: **Individual Work**

### Objectives:

- This lab will enable you to create databases in Microsoft Visual Studio
- Understand how to create database diagrams using an integrated development environment
- How to enforce and test for key constraints

### 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. would like to create an employee phone directory that allows clients to search for its employees (i.e. by department, etc.) via its online corporate website. You have been hired by this company to build the backend for this website. You will be responsible for creating SQL database, populating some entries manually, and creating database diagrams.

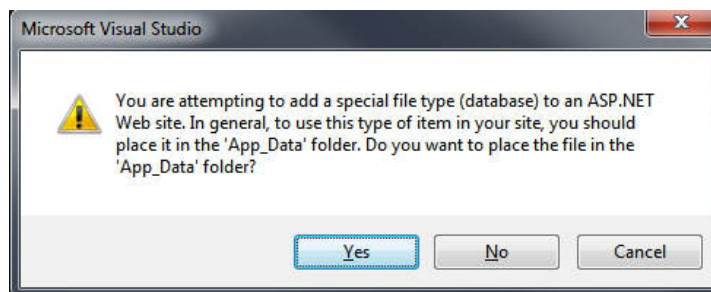
Preparatory Step: Creating an empty ASP.NET website and saving it locally on your machine

- Open **Microsoft Visual Studio**
- Click on **File** → **New Web Site** → Select **ASP.NET Empty Web Site (default)** → Click **Browse** and select a location where you would like to save your files → Click **Ok**

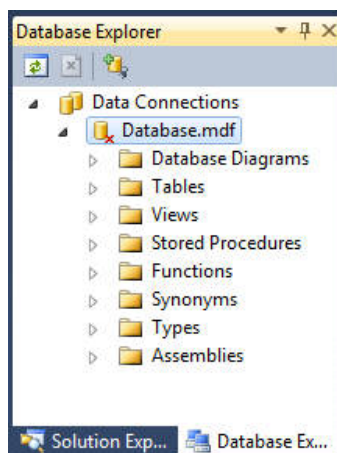
## Step 1: Creating an SQL database: Database.mdf

Using Microsoft Visual Studio, you can create a new database. Databases for ASP.NET applications are usually saved in a folder called App\_Data.


- In Microsoft Visual Studio, press 'Ctrl + Alt + L' on your keyboard. This will point you to the **Solution Explorer** (right hand side)
- On the Solutions Explorer, right click on the website name and select "Add New Item" → **SQL Server Database** → name it **database.mdf** → Click **Add**
- A pop-up window will prompt you to save the database in **App\_Data** folder. Click **Yes**



- Microsoft Visual Studio will switch automatically to the **Server (Database) Explorer**. You should see a list of database elements as follows:



- Expand the **Tables** node. It should be empty. We would like to add a new table. Right click on Tables and select “**Add New Table**”
- Create the following columns as shown below:

	Column Name	Data Type	Allow Nulls
	EmployeeID	int	<input type="checkbox"/>
	EmployeeName	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Department	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Telephone	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Extension	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Title	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Email	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

- Set the **EmployeeID** column as the **Primary key**. Save the file → Name the table **Employees**. Navigate through the Column Properties window and familiarize yourself with the options associated for each column.

## Step 2: Populating Rows in Database.mdf

[illegible]

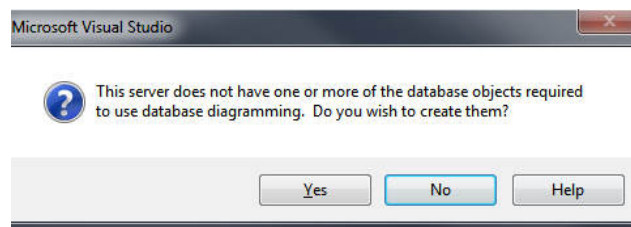
- Create another table called “**Departments**” with two columns: (a) DepartmentID as integer, and (b) DepartmentName as nvarchar(max). Set DepartmentID as Primary Key.
- Populate four rows in the Departments table as shown below:

	DepartmentID	DepartmentName
▶	100	Human Resources
	200	Graphic Design
	300	Software Development
	400	Accounting
*	NULL	NULL

### Step 3: Creating Database Diagrams

Microsoft Visual Studio contains a Database Diagram Designer

- From the Server (or Database) Explorer, right click on Database Diagrams → **Add New Diagram**
- You may get the following messages



Click “**Yes**” to ignore this warning. This message relates to database ownership and the fact that any user who has access to a database can create a diagram. You can simply disregard this warning at this stage.

- Add the two tables (Employees and Departments) to the diagram

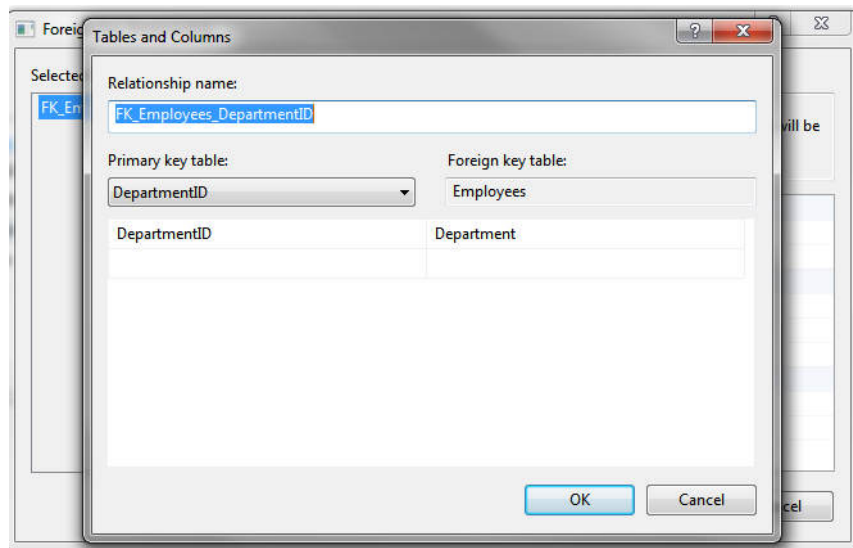
You should have something similar as below



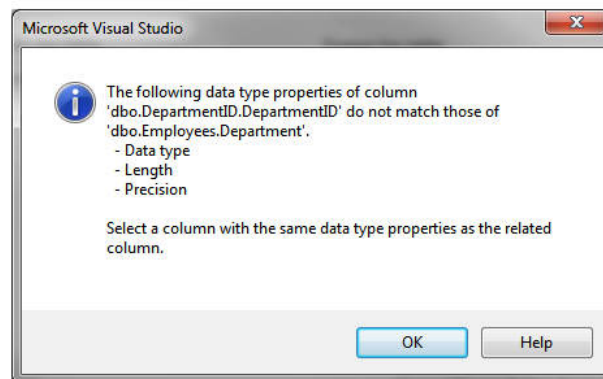
- Create a relation between the Employee table and the Department table.

### Question 1: Which keys are suitable for this relationship?

To create a relation, highlight the column in the referenced table by clicking to the left of the column name. A black arrow should appear and the column name becomes highlighted. Then, drag this column to the column in the referencing table. A pop-up window will appear as shown below:

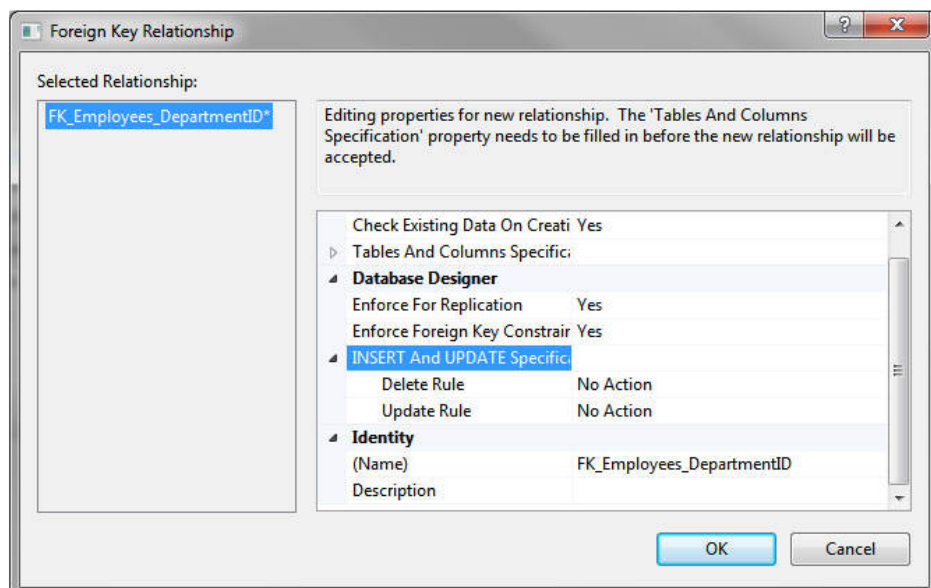


Click OK. A pop-up message will appear as shown below:



## **Question 2: Why are you receiving this error? How can this error be corrected?**

- Once you have fixed the error, you will get the following window:



- Set the Update Rule and Delete Rule to **Cascade**. Verify that the “**Enforce Foreign Key ...**” option is set to Yes.
- Save the diagram.

#### **Enforcing Foreign Key Constraint:**

- Right-click on the Employees table and select “**Show Table Data**”. Go to the Department column and attempt to enter a department id that does not exist (i.e. 500). (To reset values to original values, press ESC key).

#### **Question 3: What happens when attempting to enter a department id that does not exist?**

- Choose two or more employees working in different departments and note the department ids for these employees. Close the Employees table.
- Show the data for the departments table. Change the DepartmentID values for the two employees you noted in the previous step. Close the Department table.
- Re-show the data for the Employees table. Point to the two employees you noted and check the values of the department column. Are the department values the same or they have been updated?
- Modify the relationship between Employees and Departments tables such that the UPDATE Rule is set to NULL. Attempt to make changes to the values of the department ids in both tables and check the results.
- At this stage, ask the lab instructor to check your results and demonstrate to the instructor examples of how to enforce foreign key constraints.