COMP 3311
Database Management Systems

Lab 8

Accessing Oracle Database Via A Web Application

Lab Topics

- **Programmatically** connecting to Oracle Database from a web application.
- **Retrieving** data from Oracle Database tables and displaying them in a web browser.
- **Updating** Oracle Database tables via a web browser.

Accessing Visual Studio

- You have the following options for accessing Visual Studio.

1. Use the ITSC Programming Software virtual desktop (see Accessing the Virtual Desktop in these lab notes).

2. If you have a Windows PC, you can install the free version of Visual Studio (see Installing Visual Studio in Windows in these lab notes).

3. If you have a Mac PC, you can use Bootcamp to install Windows 10 in a Windows partition on your Mac and then install the free version of Visual Studio. As a HKUST student, you should be able to get a free copy of Windows 10 from ITSC.

Prepare The Database

- **Download to the Desktop** the file Lab8Exercise.zip from the Accessing Oracle Database Via A Web Application entry of the Lab Schedule course web page and unzip it.

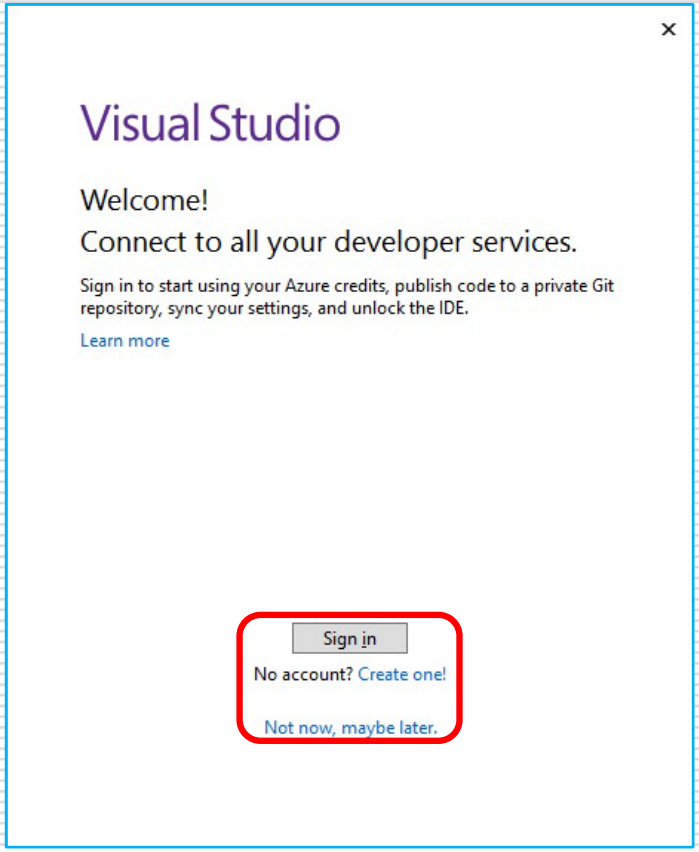
- Execute the Lab8DB.sql script file inside the Lab8Exercise folder in SQL Developer.

Start Visual Studio

- From the Start menu, find the Visual Studio 2019 app or search for it and double-click it.

DO NOT USE
Visual Studio 2015 or 2017!

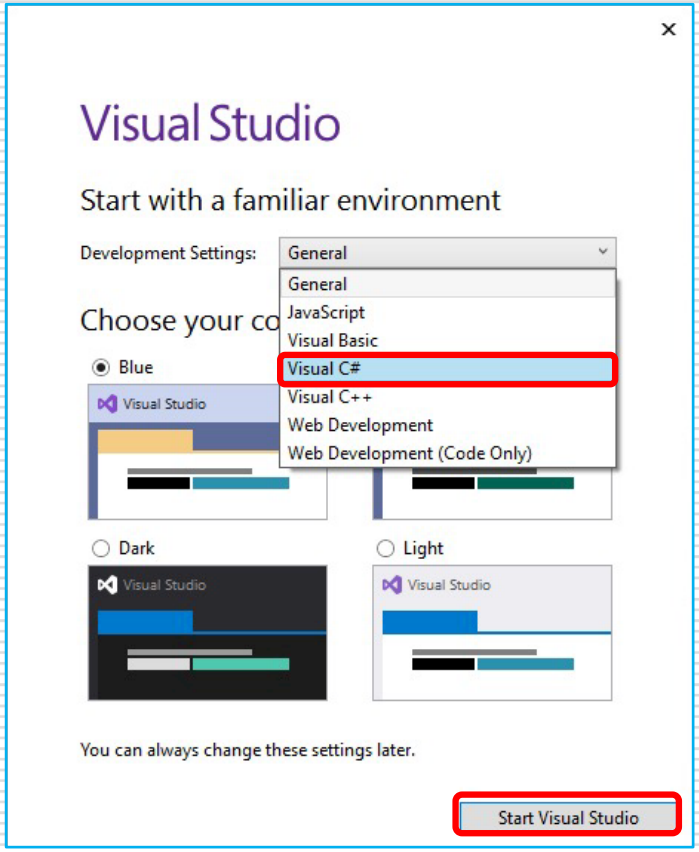
- In the Welcome dialog,
  - If you are using the virtual barn, sign in using your ITSC account.
  - If you are using the CSE lab computers, select the Not now, maybe later. link at the bottom of the dialog.



Select the Environment

- When prompted,
  - select Visual C# as the development setting (highlighted in red on the right).

    DO NOT SELECT C++ or any other development setting!
  - click the Start Visual Studio button (highlighted in red on the right).

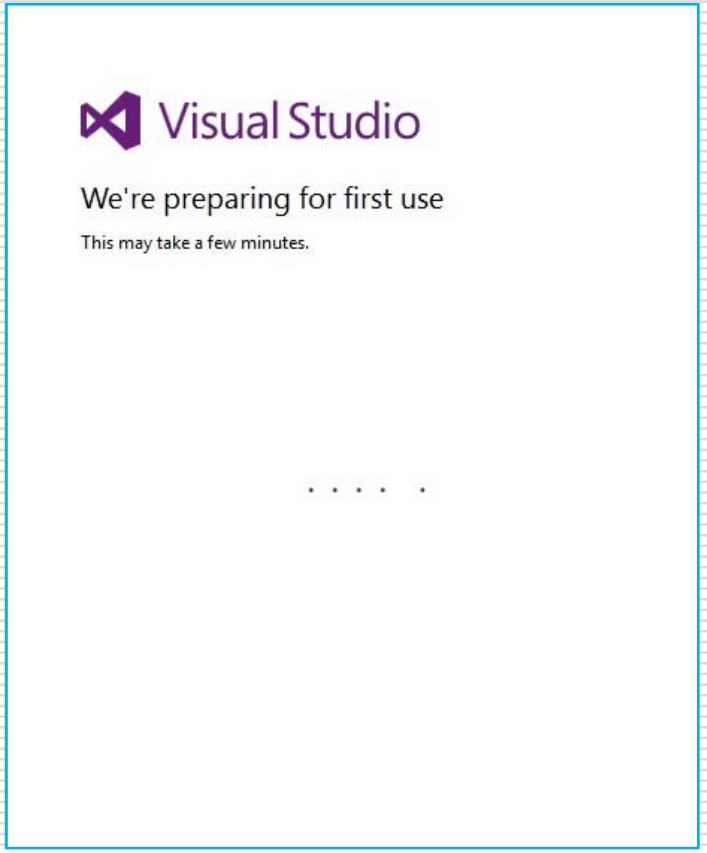


Be Patient

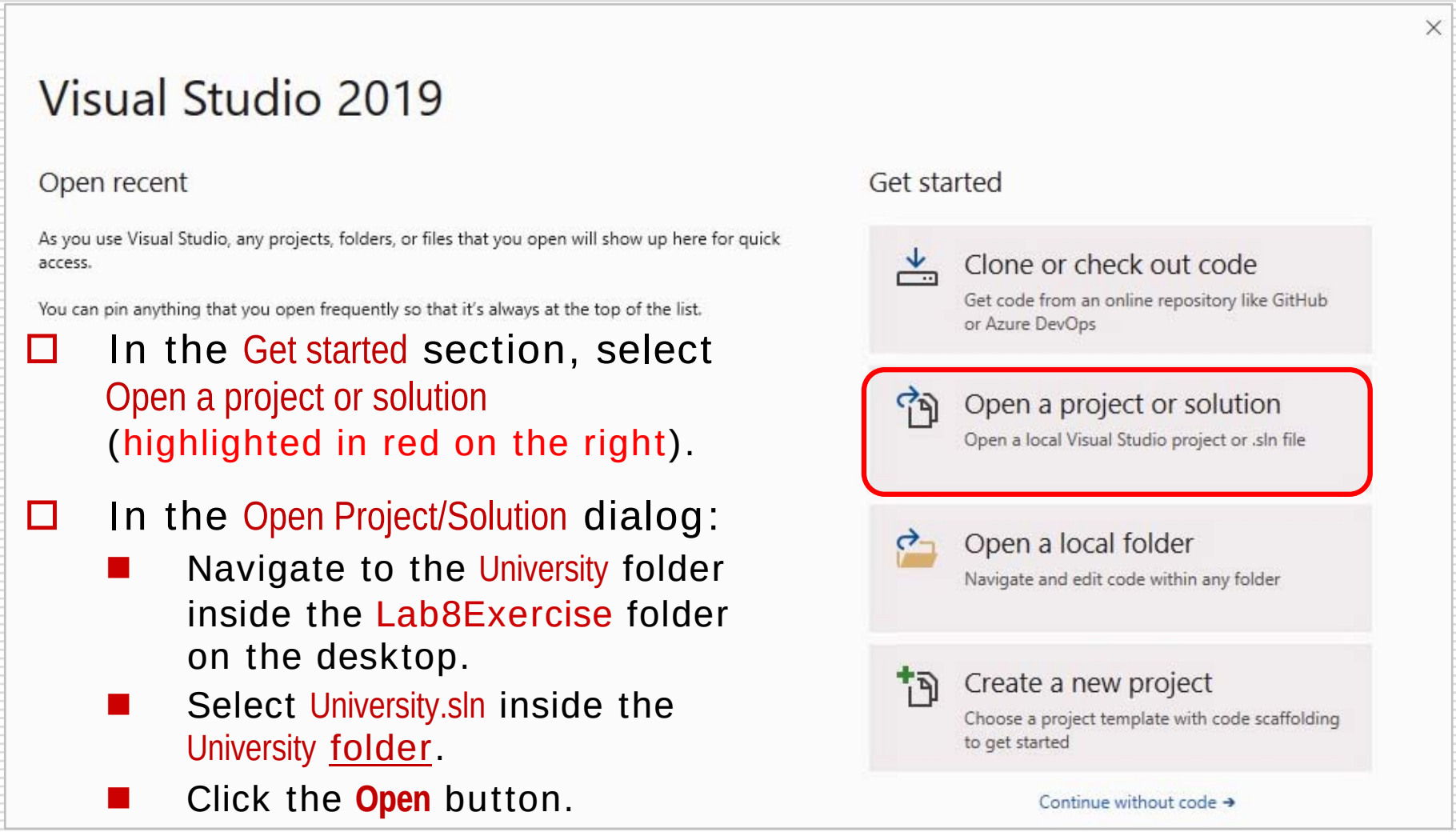
- It may take several minutes for Visual Studio to start up. We know it might be hard, but

BE VERY PATIENT!

- Eventually, the dialog shown on the next page appears.



Visual Studio Get Started Dialog



- In the Get started section, select Open a project or solution (highlighted in red on the right).
- In the Open Project/Solution dialog:
  - Navigate to the University folder inside the Lab8Exercise folder on the desktop.
  - Select University.sln inside the University folder.
  - Click the **Open** button.

Solution Explorer – Connect To Oracle

- The Solution Explorer allows file and resource management for a project.
- In the Solution Explorer, double-click on the Web.config file (highlighted in red on the right).
- Find the line

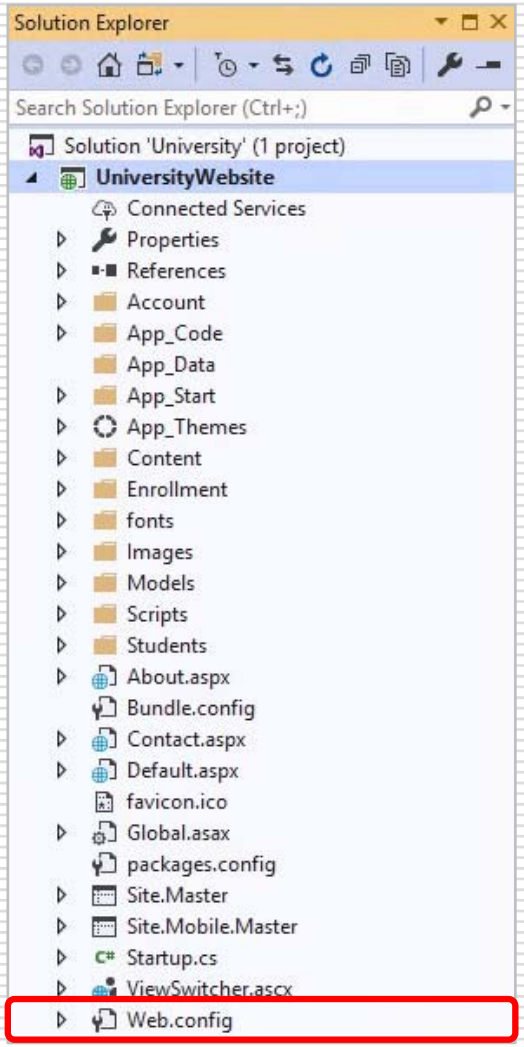
  <add name="UniversityConnectionString" ...
- On this line, find the following:

  ID=comp3311stuXXX;Password=XXXXXXXX

  and replace

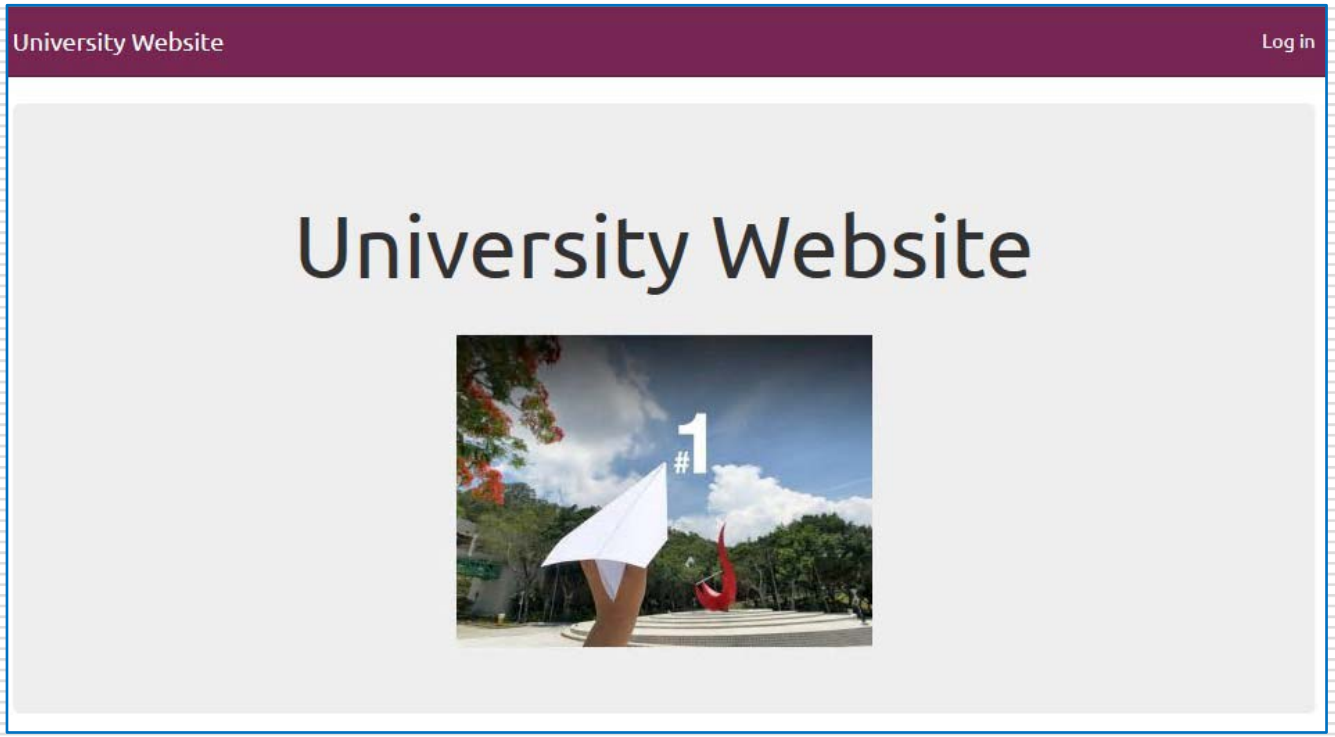
  comp3311stuXXX with your Oracle user name

  XXXXXXXX with your Oracle password
- Save and close the Web.config file.



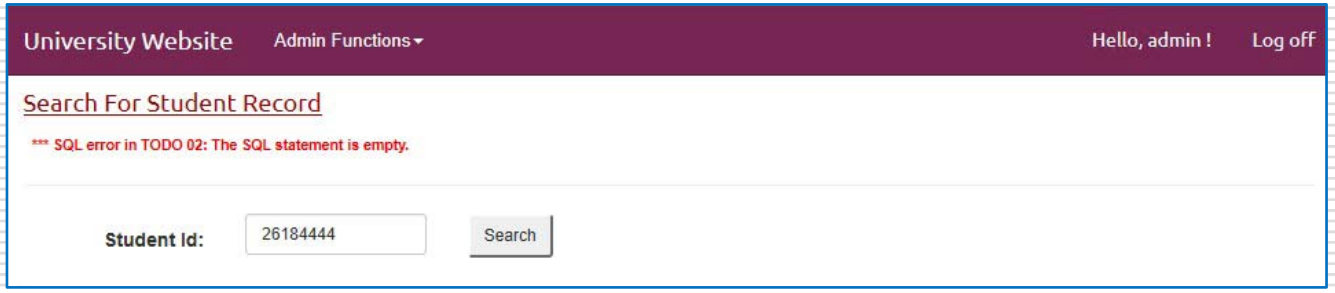
Test the Website (1)

- Select Start Debugging in the Debug menu or press the F5 function key to view the website.
- The homepage of the website, shown below, is displayed.



Test the Website (2)

- To use the website, you need to login; login as "admin".
- In the navigation bar, select Admin Functions → Search For Student Record and enter a student id in the textbox (e.g., 26184444).
- You will get an error message as shown below, indicating that there is no SQL statement defined for **TODO 02**.
  - The required SQL statement needs to be constructed as explained shortly.
  - Close the browser window.

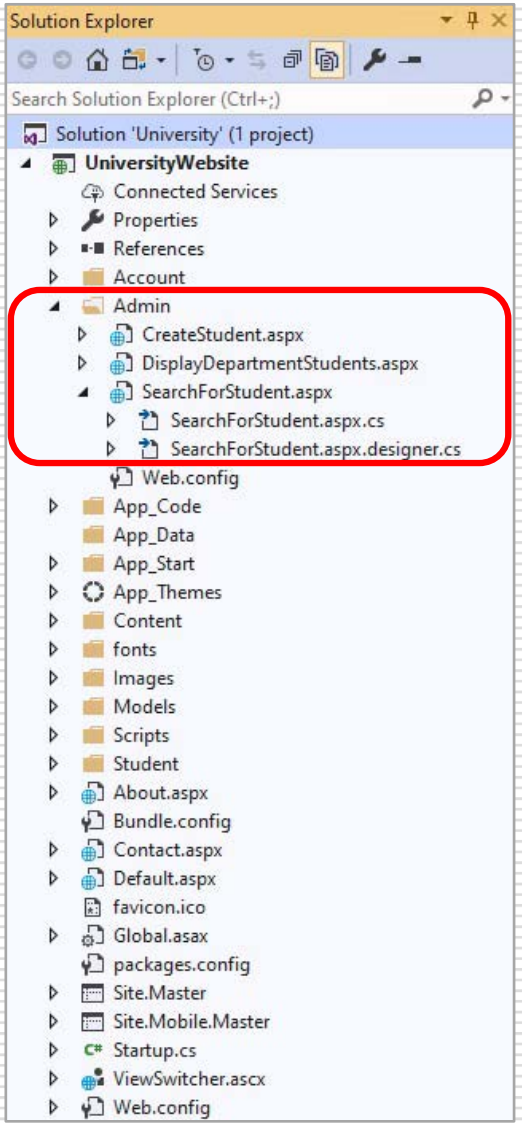


Web Forms (1)

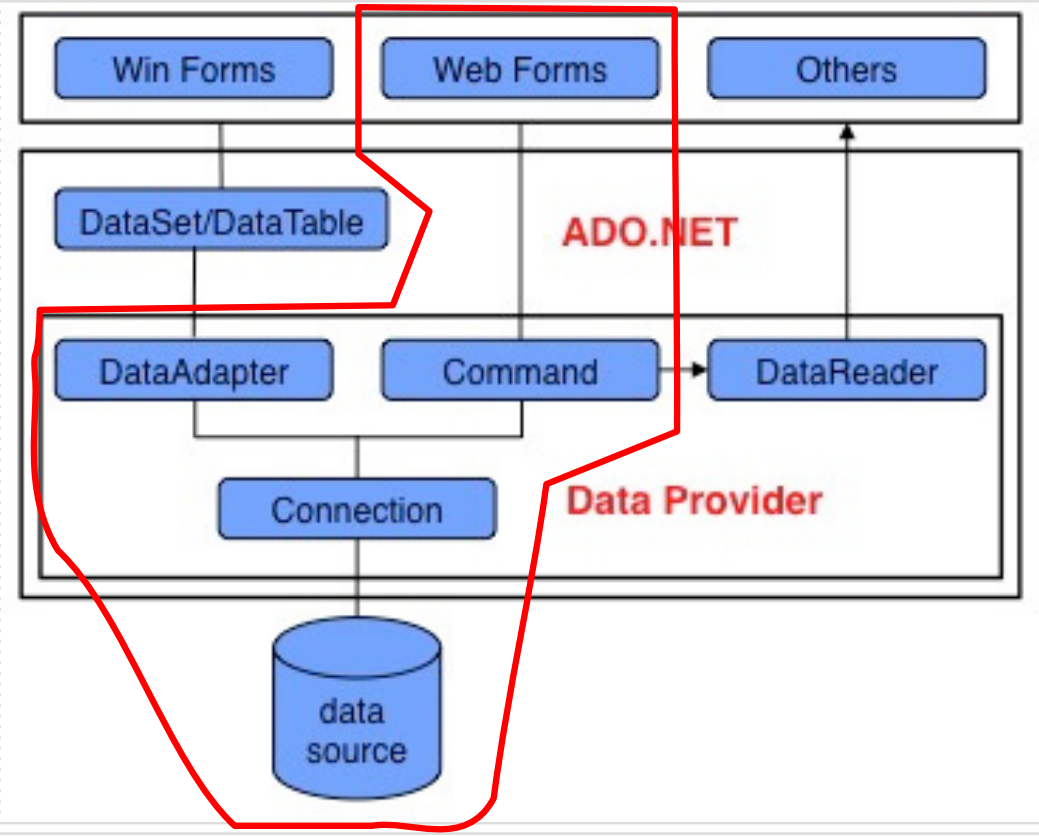
- A web form, which is rendered as a web page, provides the user interface (UI) of a website.
- Visual Studio lets you create web pages by dragging and dropping server controls, such as text boxes and buttons, onto a web form to construct a web page.
- Properties, methods and events for server controls or for the web page can be set in order to define the web page's behavior, look and feel.
- Web forms are constructed using a combination of HTML, server controls and server code.

Web Forms (2)

- A website's web forms (with extension aspx) can be found in the Solution Explorer (shown on the right).
- Each web form also has a C# code-behind file (with extension aspx.cs) that contains the code that processes the web form (more on this later).
- The web form and code-behind file for the web page that searches for a student record can be found inside the Admin folder in the Solution Explorer as highlighted in red on the right.



ASP.NET Data Access Architecture

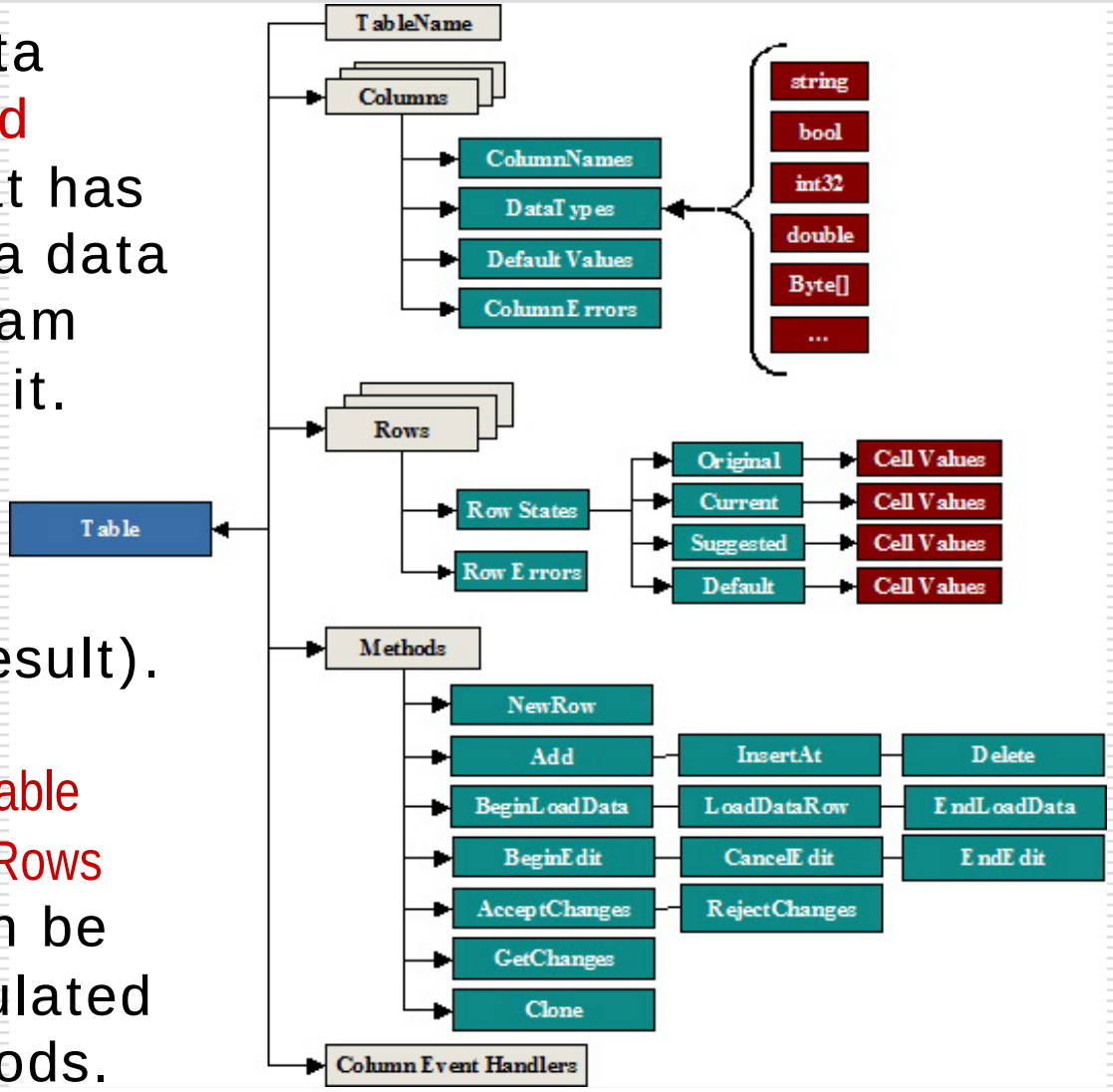


Note: The code that actually accesses Oracle Database is in the class OracleDBAccess.cs inside the App_Code folder.

DO NOT MODIFY THE CODE IN THIS CLASS!

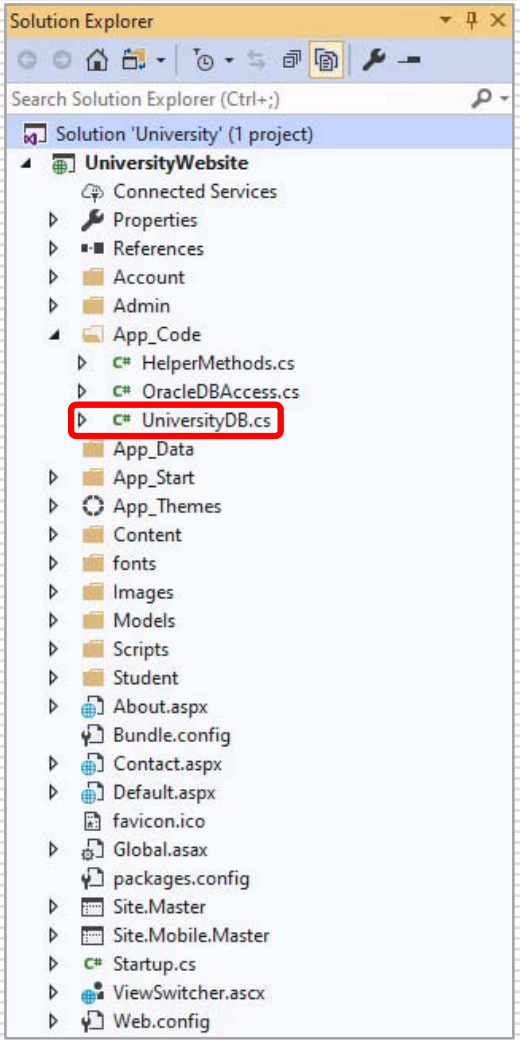
DataTable

- A DataTable is a C# data structure used to hold data *in memory*—that has been retrieved from a data source—where program code can manipulate it.
- A DataTable can hold *at most* one table (i.e., a query result).
- A table within a DataTable contains Columns and Rows collections, which can be accessed and manipulated using standard methods.



UniversityDB.cs Code File (1)

- The code that constructs the required SQL statements to access the database is contained in a C# code file named UniversityDB.cs, which is located in the App_Code folder in the Solution Explorer.
- In the Solution Explorer, expand the App_Code folder.
- Double click on the file UniversityDB.cs.



UniversityDB.cs Code File (2)

- In the code file, to retrieve the record of a student, identified by his/her student id, an SQL statement is constructed, as a string named sql, in which the value of the studentId parameter is used (1 and 2).
- Then, the string sql is passed to the procedure myOracleDBAccess.GetData, which contains the code required to access the Oracle database and the result is assigned to a DataTable (3), which is returned to the calling method.

```
public class UniversityDB
{
    private readonly OracleDBAccess myOracleDBAccess = new OracleDBAccess();
    private string sql;

    #region SQL statements for admin
    1 reference
    public DataTable GetStudentRecord(string studentId) (1)
    {
        //*************************************************************************
        // TODO 01: Used in Admin/SearchForStudent.aspx.cs                        *
        // Construct the SELECT statement to find the record (i.e., to return *
        // all the attributes) of a student identified by his/her student id. *
        //*************************************************************************
    (2) sql = "select * " +
            "from Student " +
            "where studentId='" + studentId + "'";
    (3) return myOracleDBAccess.GetData(sql);
    }

    1 reference
    public decimal StudentRecordExists(string studentId)...

    1 reference
    public DataTable GetDepartmentStudentRecords(string departmentId)...

    1 reference
    public DataTable GetDepartments()...

    1 reference
    public bool CreateStudentRecord(string studentId, string firstName,
        string lastName, string email, string phoneNo, string cga,
        string departmentId, string admissionYear)...

    #endregion SQL statements for admin

    SQL statements for students

    *** DO NOT CHANGE THE METHODS BELOW THIS LINE. THEY ARE NOT TODOS!!! ***!
}
```

Web Form Code-behind File

- The methods in the code file UniversityDB.cs are called from the code-behind files of the web forms as shown in the figure on the right for the code-behind file for the SearchForStudent web form.
- The code-behind files have comments that cross-reference the methods which they call in the UniversityDB.cs code file.

```
public partial class SearchForStudent : Page
{
    //******************
    // Uses TODO 01, 02 *
    //******************

    private readonly UniversityDB myUniversityDB = new UniversityDB();
    private readonly HelperMethods myHelperMethods = new HelperMethods();

    /*----- Private Methods -----*/
    1 reference
    private bool StudentIdIsValid(string studentId)...

    /*----- Protected Methods -----*/

    0 references
    protected void BtnFindStudent_Click(object sender, EventArgs e)
    {
        // Hide previous results.
        lblResultMessage.Visible = false;
        pnlStudentRecord.Visible = false;

        string studentId = myHelperMethods.CleanInput(txtStudentId.Text);

        if (IsValid && StudentIdIsValid(studentId))
        {
            //***************
            // Uses TODO 01 *
            //***************
            DataTable dtStudentRecord = myUniversityDB.GetStudentRecord(studentId);

            // Attributes expected to be returned by the query result.
            var attributeList = new List<string> { "STUDENTID", "FIRSTNAME", "LASTNAME", "EMAIL",
                "PHONENO", "CGA", "DEPARTMENTID", "ADMISSIONYEAR" };

            // Display the query result if it is valid.
            if (myHelperMethods.IsQueryResultValid("TODO 01", dtStudentRecord, attributeList, lblResultMessage))
            {
                // Display a no result message if nothing was retrieved from the database.
                if (dtStudentRecord.Rows.Count != 0)
                {
                    gvStudentRecord.DataSource = dtStudentRecord;
                    gvStudentRecord.DataBind();
                    pnlStudentRecord.Visible = true;
                }
                else // Display a no result message.
                {
                    myHelperMethods.DisplayMessage(lblResultMessage, "No record for the student was found.");
                }
            }
        }
    }
```

Complete UniversityDB.cs Code File (1)

- Most of the SQL statements that you need to construct require values that are passed as parameters of a method that is called from the code-behind file of a web form.
- Consider, as an example, the code on slide 17 which retrieves the record of a student with a specified student id.
- The student id value required to construct the select statement is obtained from a TextBox control on the web form, named txtStudentId, and passed to the method GetStudentRecord in the studentId parameter.

Complete UniversityDB.cs Code File (2)

- In the UniversityDB.cs code file, the SQL statement to retrieve the student record is then constructed and assigned to the variable sql as follows:

```
sql = "select * from Student where studentId='" + studentId + "'";
```

1. Since the type of the studentId attribute is char, it is necessary to put single quotes around the value of the studentId parameter so that the SQL statement will look like

```
select * from Student where studentId='26184444'
```

   if the value of the studentId parameter is 26184444.

2. Note that the C# string concatenation operator is +.

Note: You should only put single quotes around values for attributes that are strings. It is not recommended to put single quotes around values for attribute that are numbers. Moreover, you should never put single quotes around the value null.

Complete UniversityDB.cs Code File (3)

- The method parameters have been assigned names that should make it obvious what values they hold.
- You will need to use the method parameters to construct most of the SQL statements, which are marked by **TODO** comments.
- There are eight additional **TODO**s (i.e., eight SQL statements to construct) in the UniversityDB.cs code file.
- As an example, we will construct the SQL statement for **TODO 02**, which determines whether a record exists for the student id input into the web page that searches for a student.

Complete UniversityDB.cs Code File (4)

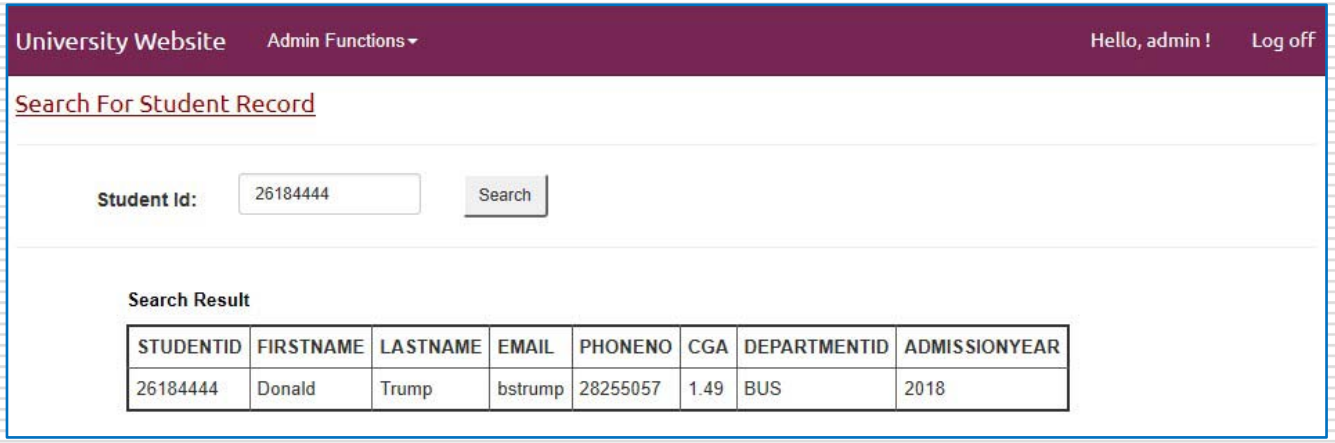
- As explained in the comments for **TODO 02** in the UniversityDB.cs code file, a student record exists if the SQL statement returns a value of 1 for a given student id; otherwise, it does not exist.

- What is the SQL statement that would return such a result given a student id?

```
public decimal StudentRecordExists(string studentId)
{
    //*****************************************************************
    // TODO 02: Used in Admin/SearchForStudent.aspx.cs                *
    // Construct the SELECT statement to determine if a record exists  *
    // in the database for a student identified by his/her student id. *
    // The SELECT statement should return:                             *
    //    0 - if the student record does not exist;                    *
    //    1 - if the student record exists.                            *
    //*****************************************************************
    sql = "";
    return myOracleDBAccess.GetAggregateValue(sql);
}
```

Complete UniversityDB.cs Code File (5)

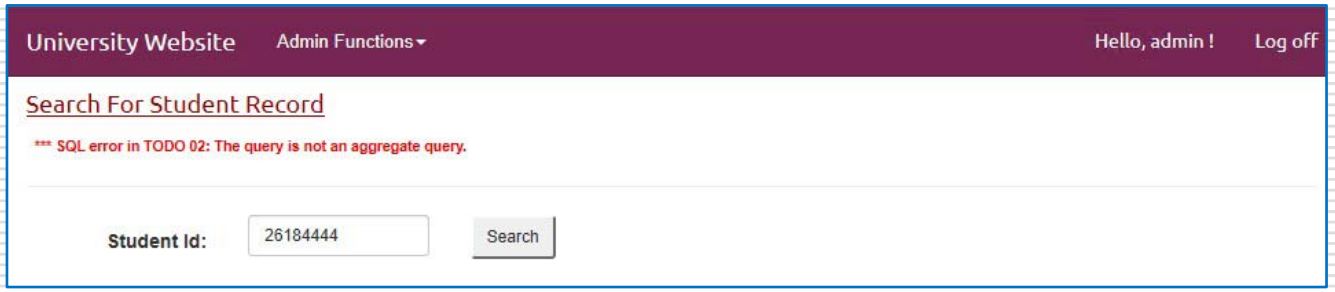
- After you correctly complete **TODO 02**, then when you search for the student record with student id 26184444, you should find the record shown below.



STUDENTID	FIRSTNAME	LASTNAME	EMAIL	PHONENO	CGA	DEPARTMENTID	ADMISSIONYEAR
26184444	Donald	Trump	bstrump	28255057	1.49	BUS	2018

Debugging Your SQL Statements (1)

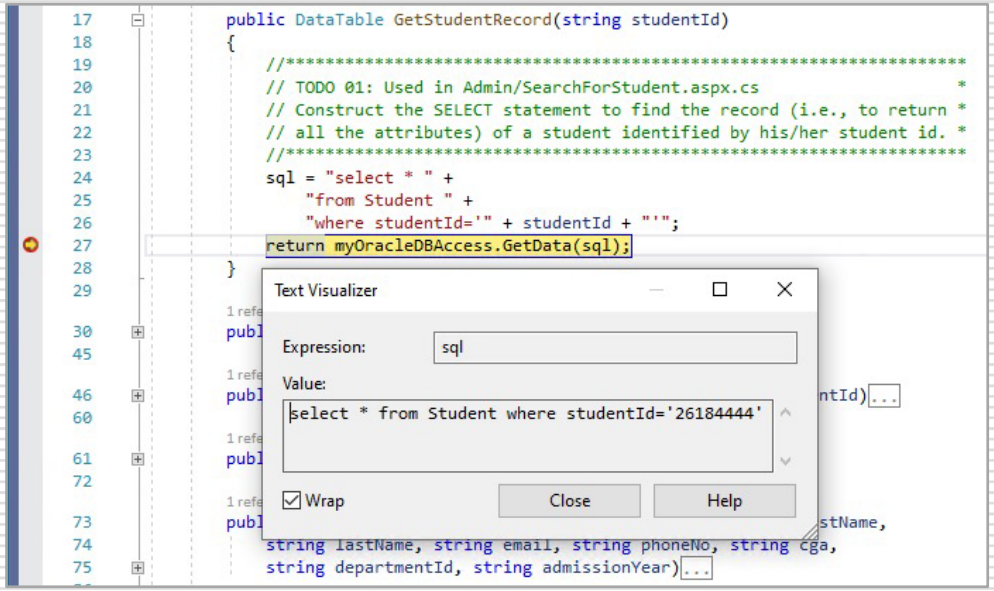
- If your SQL statement has an error in it, then when you execute it, you will get an error message indicating the type of error that occurred as shown below.



- Since the message will not indicate where in your SQL statement the error occurred, debugging is very difficult.
- Therefore, before executing an SQL statement in Visual Studio, it is highly recommended that you first "debug" it in SQL Developer using appropriate values for any variables.
- You can also set breakpoints in Visual Studio for debugging.

Debugging Your SQL Statements (2)

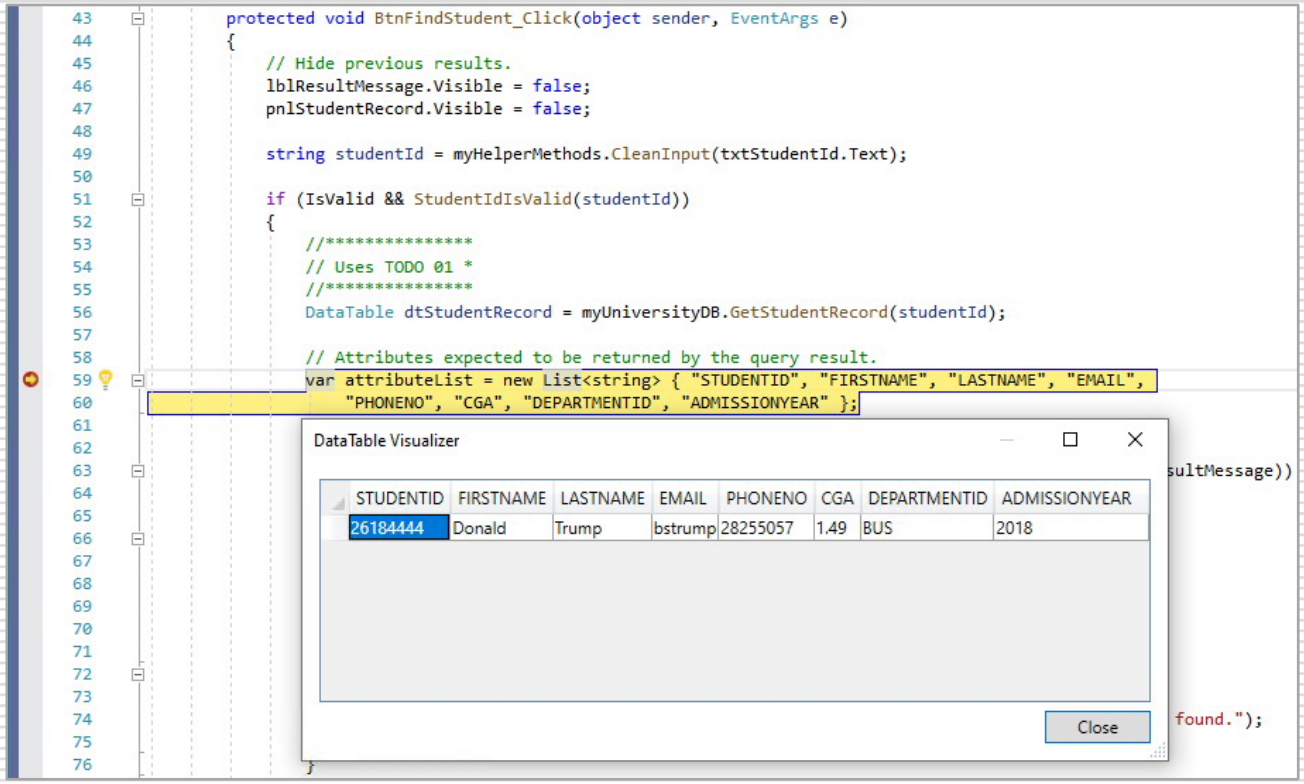
- To set a breakpoint on a code line in the UniversityDB.cs file, click in the left-most margin of the code editor window.
- Execution will then stop at this line (right-pointing yellow arrow inside the red circle).



- To see the value of the sql variable in the code line, hover the cursor over it.
- Select the magnifying glass icon in the popup to view the SQL statement in the Text Visualizer dialog box.

Debugging Your SQL Statements (3)

- To view the result of an SQL statement, set a break-point in the code-behind file where the **TODO** is used as shown in the figure.

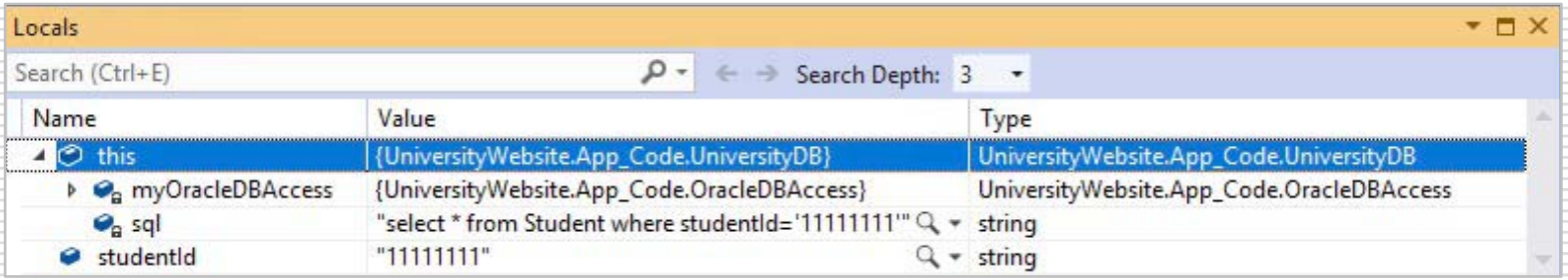


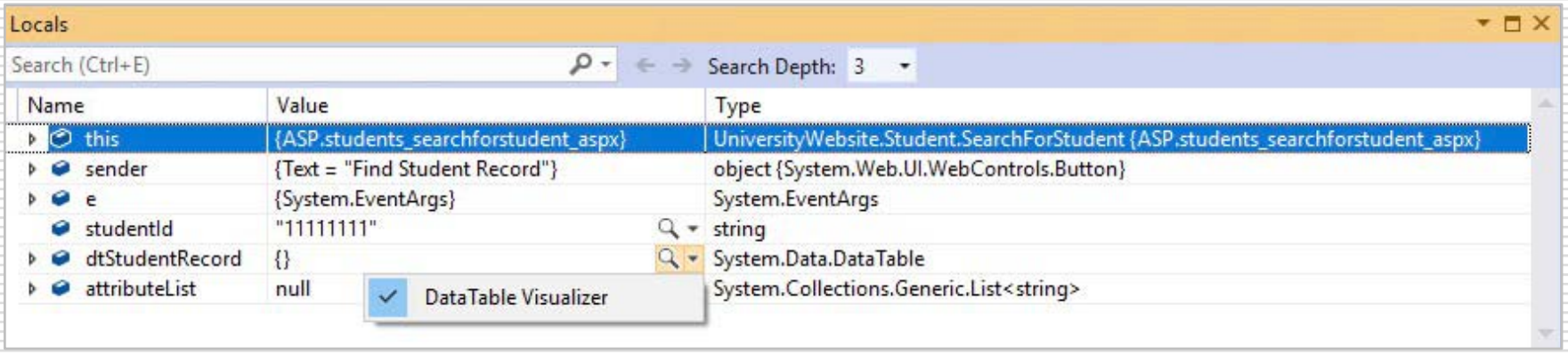
- Place the cursor over the DataTable variable dtStudentRecord.

- Select the magnifying glass icon in the popup to view the DataTable contents in the Text Visualizer dialog box.

Debugging Your SQL Statements (4)

- You can also inspect the values of the variables in the Locals tab (usually found at the bottom of the Visual Studio window) as shown in the figures below.





Lab Exercise

Ask for help if you need it!

IMPORTANT NOTE

You cannot access Oracle Database from the M drive using Visual Studio.

Your website folder must be on the local computer.

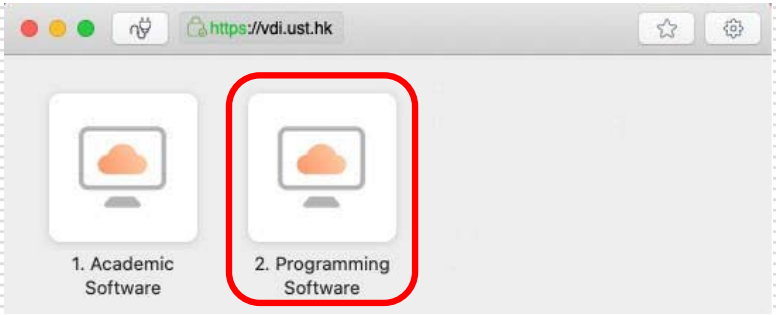
DO NOT modify any of the other code in the UniversityDB.cs code file or in any other files! In particular, do not modify the web forms. We cannot help you if you have changed these files. In this case the best thing to do is to start over by deleting the University folder and downloading it again.

Accessing the Virtual Desktop

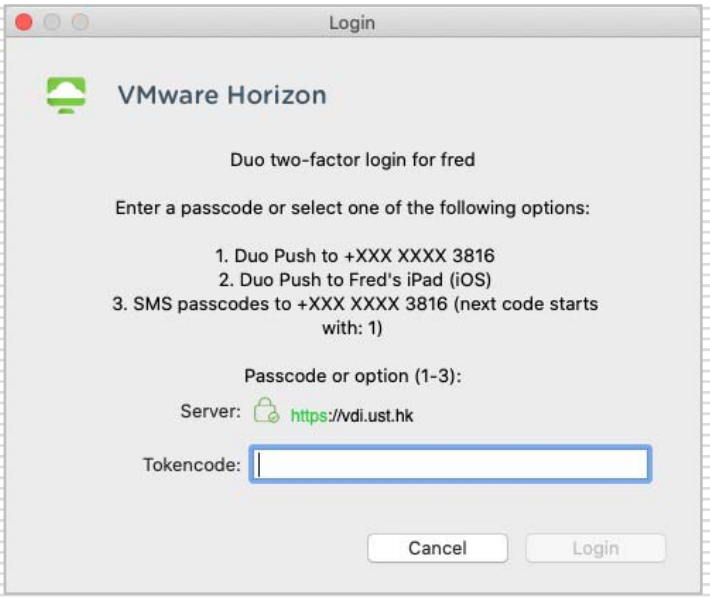
1. Login via VMWare Horizon Client.
   (Get the software here.)

2. Select the two-factor login method you want to use.

3. Select the Programming Software virtual desktop as shown below.



4. Search for Visual Studio 2019 and launch it.

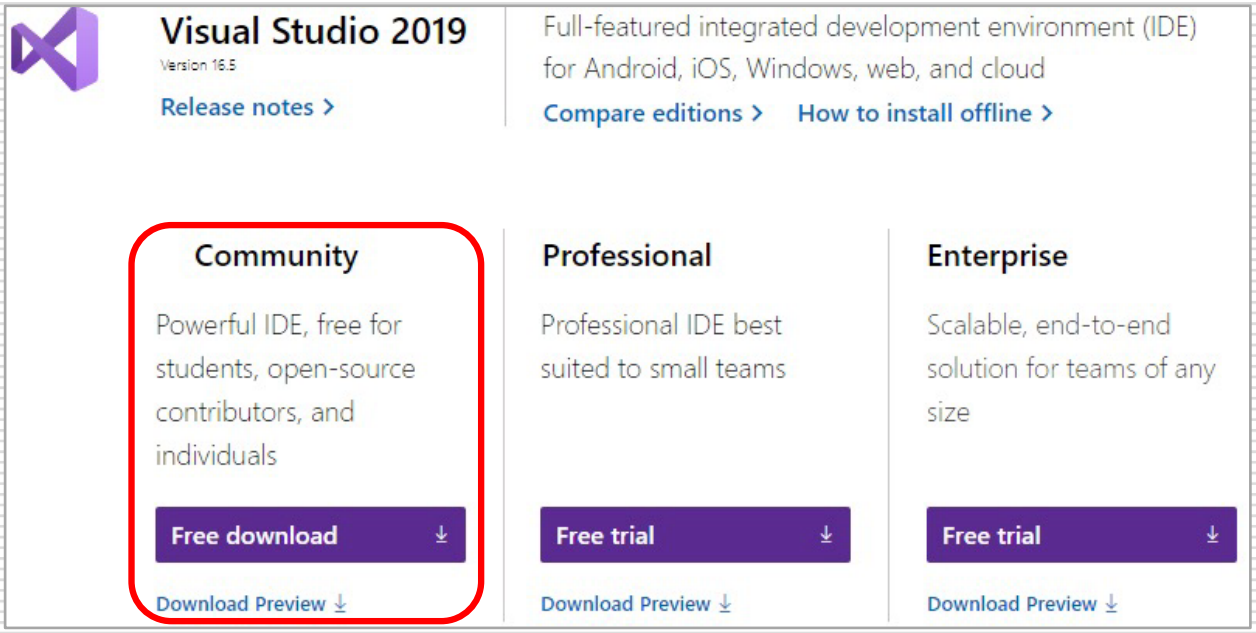


Installing Visual Studio in Windows (1)

- Download the free version of Visual Studio 2019 called, Community, from

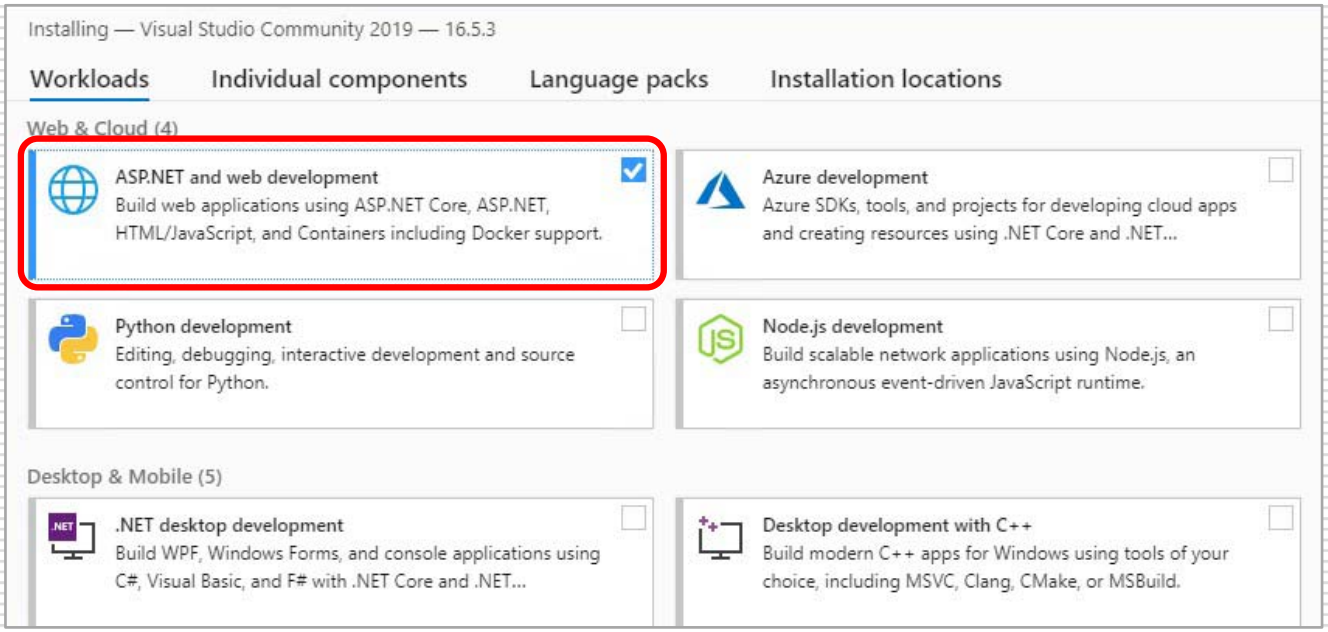
  https://visualstudio.microsoft.com/downloads/

(Note that the Mac version of Visual Studio 2019 is not compatible with the Visual Studio Windows projects used in this course. If you have a Mac, you may consider using Bootcamp to install Windows on your Mac and then install Visual Studio 2019 Community in the Bootcamp partition. Windows 10 is free to download from ITSC for HKUST students.)



Installing Visual Studio in Windows (2)

- When prompted by the Visual Studio installer, select ASP.NET and web development in the Web & Cloud section as shown in the figure below.



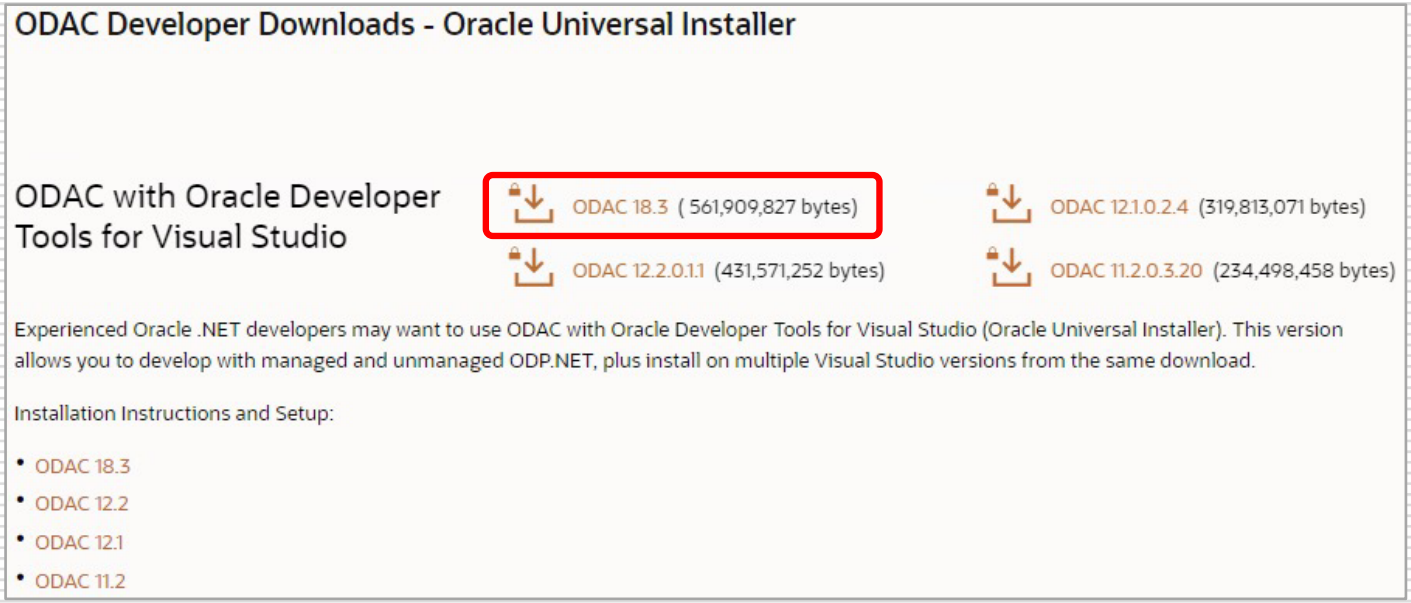
Installing Visual Studio in Windows (3)

- To access Oracle Database from within a Visual Studio project, you also need to download and install Oracle Data Access Components (ODAC) from

  https://www.oracle.com/database/technologies/dotnet-odacdev-downloads.html

  - Download ODAC 18.3 as highlighted in the figure below.

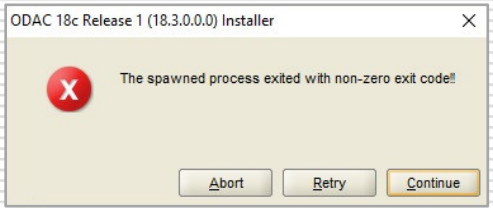
    (You will be required to login using an Oracle account to download. The account is free to create.)

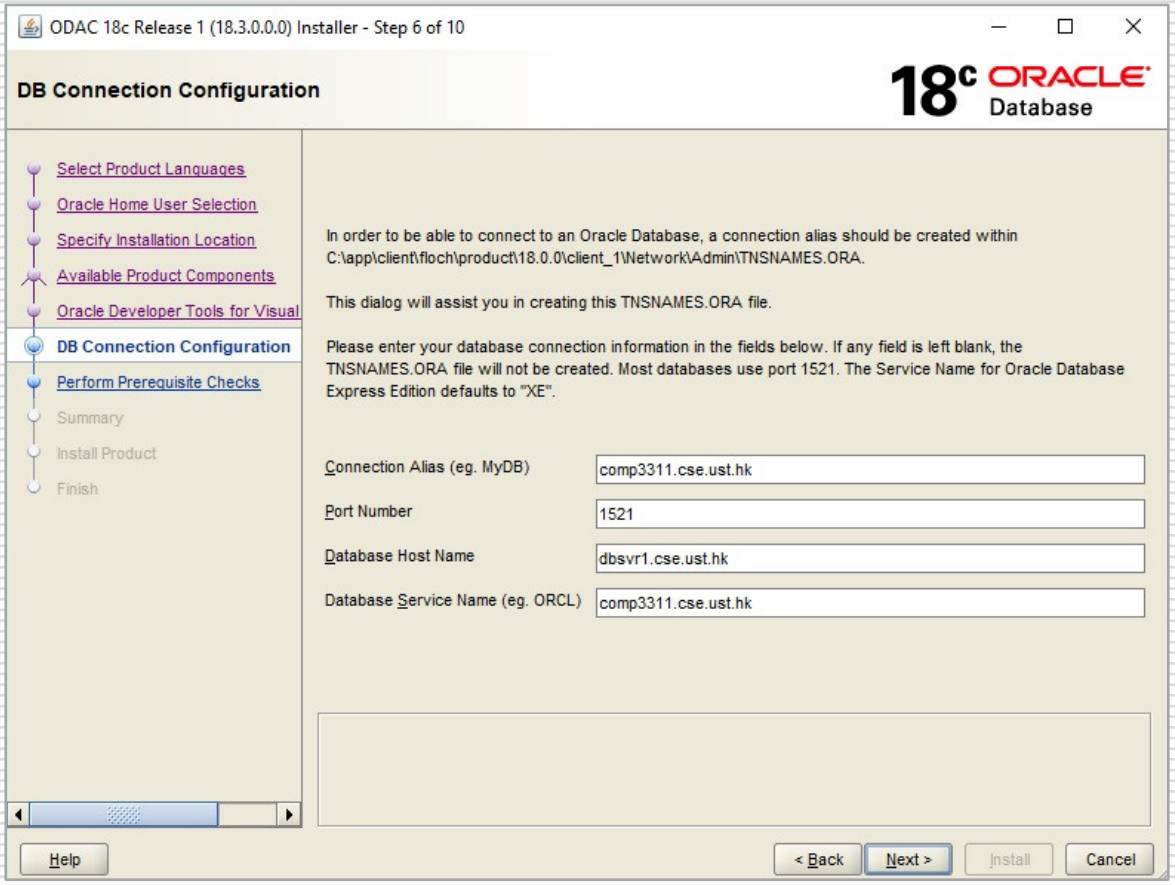


Installing Visual Studio in Windows (4)

- Start the ODAC 18.3 Installer, select the Next > button in each step and, when prompted, complete the DB Connection Configuration dialog exactly as shown in the figure.

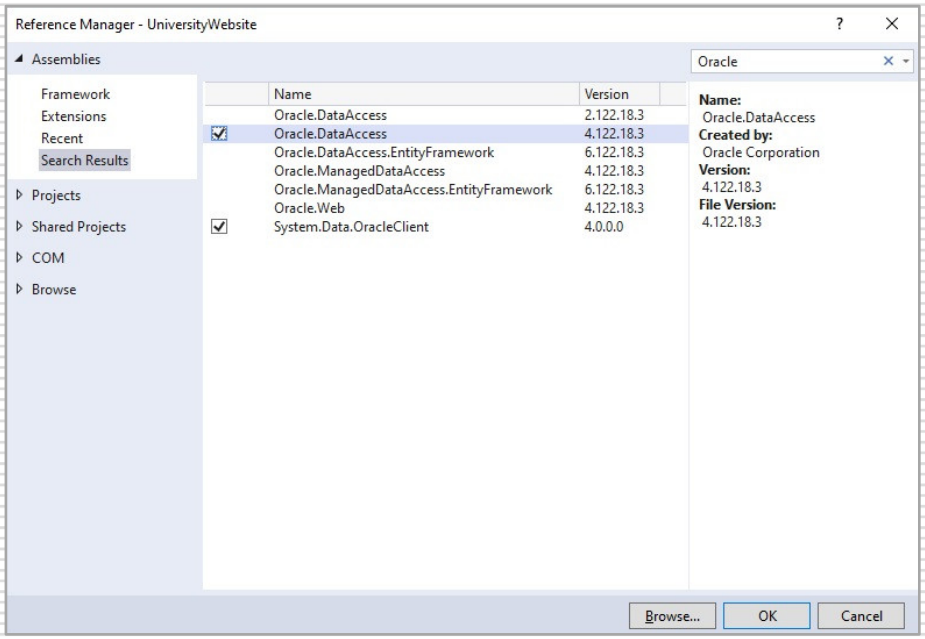
- Select the **Continue** button at the dialog shown below.





Installing Visual Studio in Windows (5)

- Start Visual Studio and add a reference to the Oracle Data Access Components as follows.
  - Right-click on the References node in the Solution Explorer.
  - Select Add reference... from the popup menu.
  - Search for Oracle.
  - Select the highest version of OracleDataAccess as shown in the figure.
  - select the OK button.



- The Oracle Database server can only be accessed within the HKUST network. To access it from outside the HKUST network, you need to use the UST VPN.

  http://itsc.ust.hk/apps/vpn/