Course Number: 420-DW4-AS
Title: Web Server Applications Development II
Teacher: Quang Hoang Cao
Session: Winter 2024

Formative Evaluation
Database Programming (connected mode) in ASP.NET
LAB 1

Due Date: January 31, 2024 (Demo in class and submission via LEA)

Objective

At the end of this lab, you are able to

1. create and use Web forms (Properties, events and methods)
2. use server controls,
3. use validation controls,
4. use ADO.NET object model (classes: SqlConnection, SqlCommand, SqlDataReader, SqlParameter),
5. develop the data access classes with ADO.NET,
6. use data access classes in business classes,
7. create and use an SQL Server database in a Web application and perform CRUD operations.

Performance Criteria

2.1 Proper installation of the Web development platform and the development database management system
2.2 Proper installation of software and libraries
3.1 Suitable creation or adaptation of the database
3.2 Proper insertion of initial or test data
3.3 Compliance with the data model
4.1 Appropriate use of markup language
4.2 Suitable creation and use of style sheets
4.3 Proper integration of images
4.4 Suitable creation of Web forms
4.5 Adaptation of the interface based on the display format and resolution
5.2 Proper programming of interactions between the Web interface and the user
5.3 Appropriate choice of clauses, operators, commands or parameters in database queries
5.4 Correct handling of database data

Problem Statement

Design and implement an **ASP.Net Web Forms Application** using Visual Studio 2022, C# and SQL Server 2022/2019.The application allows the user to manage the employee information for a given company.

Following is the list of the application's operations

1. Save: Save new employee information to the database **EmployeeDB.mdf.**
Note: For simplicity (in this lab), each employee information should include the following:

- Employee number
- First name
- Last name
- JobTitle

2. Search: Search an employee information by

- an employee number
- a first name /last name

3. Update: Update employee information. Confirmation message is required.

4. Delete: Delete employee information. Confirmation message is required.

5. List All: List all the employees

Data Validation

1. The employee number must be a 4-digit number and unique.
2. The first name / las name contains only letters. A whitespace can be used only to separate the name component if required.

	Input	Valid	invalid
Employee Number	*123456; 123abcd; abcd1234*		***V***
	1111	***V***	
First Name	*123Mary; Mary12*		***V***
	Mary	***V***	
Last Name	*123Brown; Brown12*		***V***
	Brown	***V***	

Important Notes

Add a text file to your project. In this file, specify the following:
Your name :
Auto-evaluation : /100
Suggestions:

Step 1 Creating the Database using SQL Server Management Studio 18 (or higher)

Database Name: **EmployeeDB**

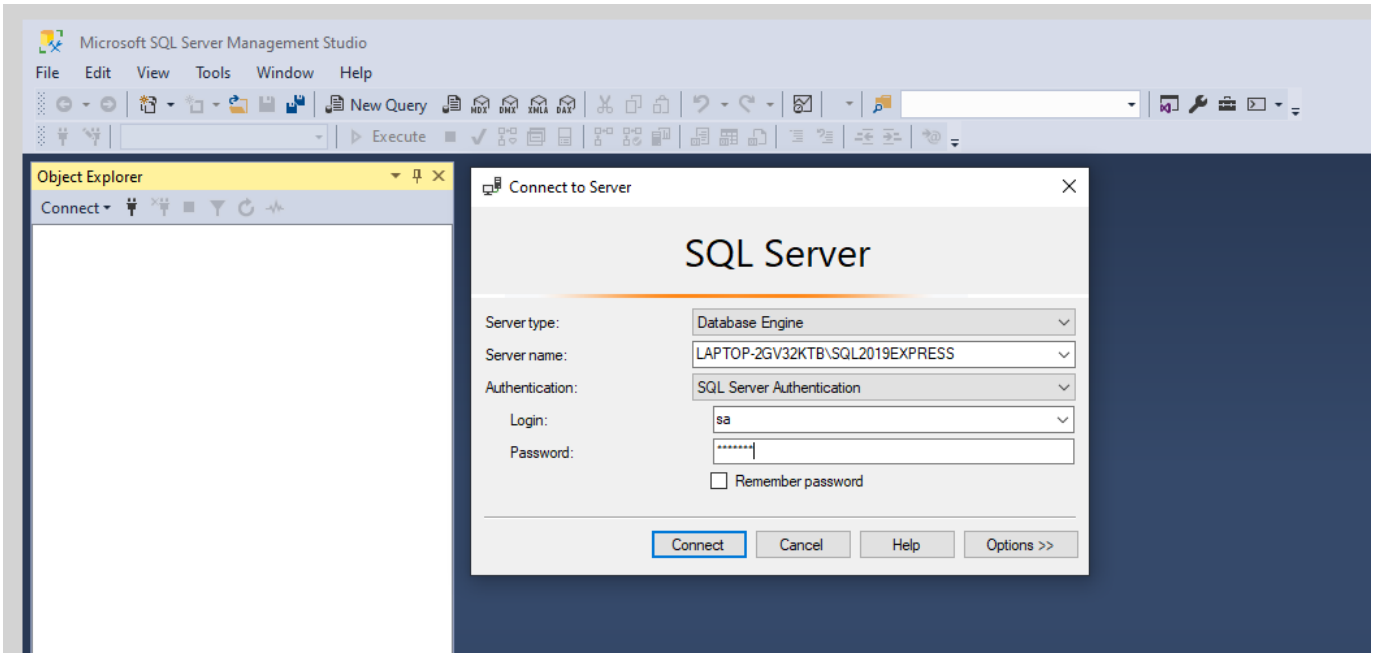
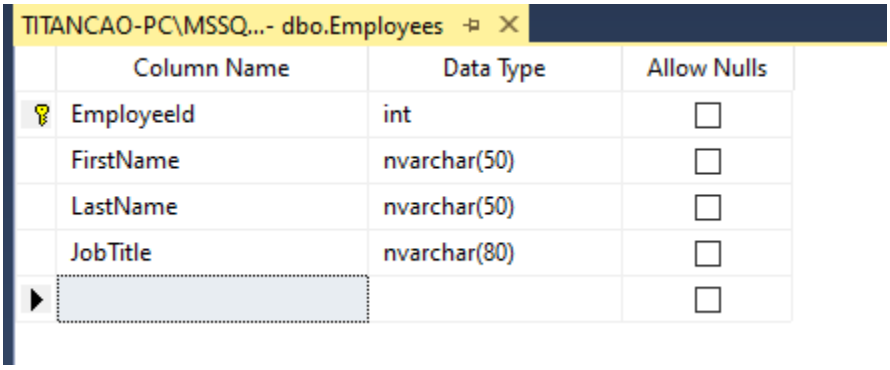


Table Name: **Employees**



Important Note

Follow the naming conventions discussed in class.

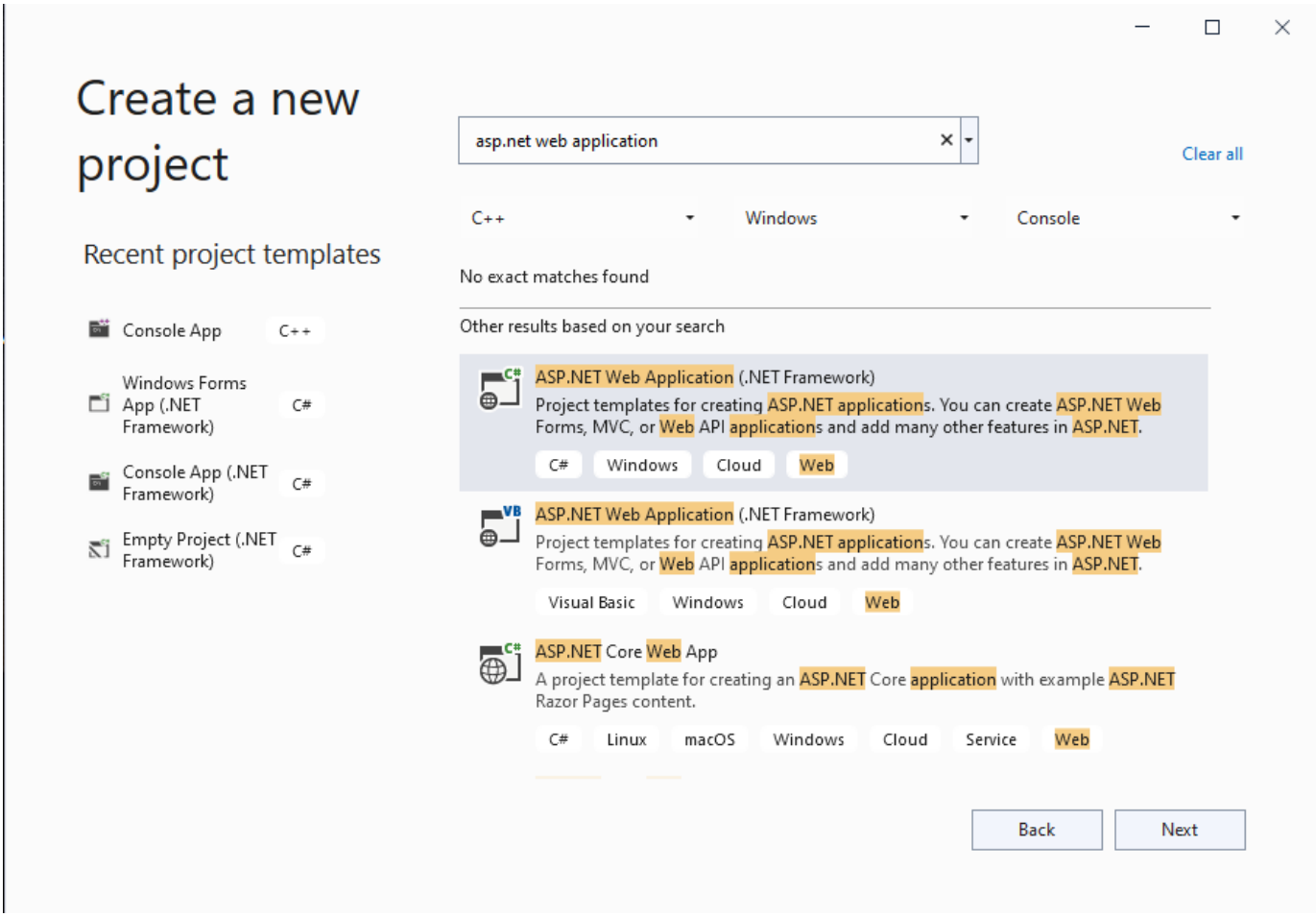
Populate the table with the data

TITANCAO-PC\MSSQ...- dbo.Employees

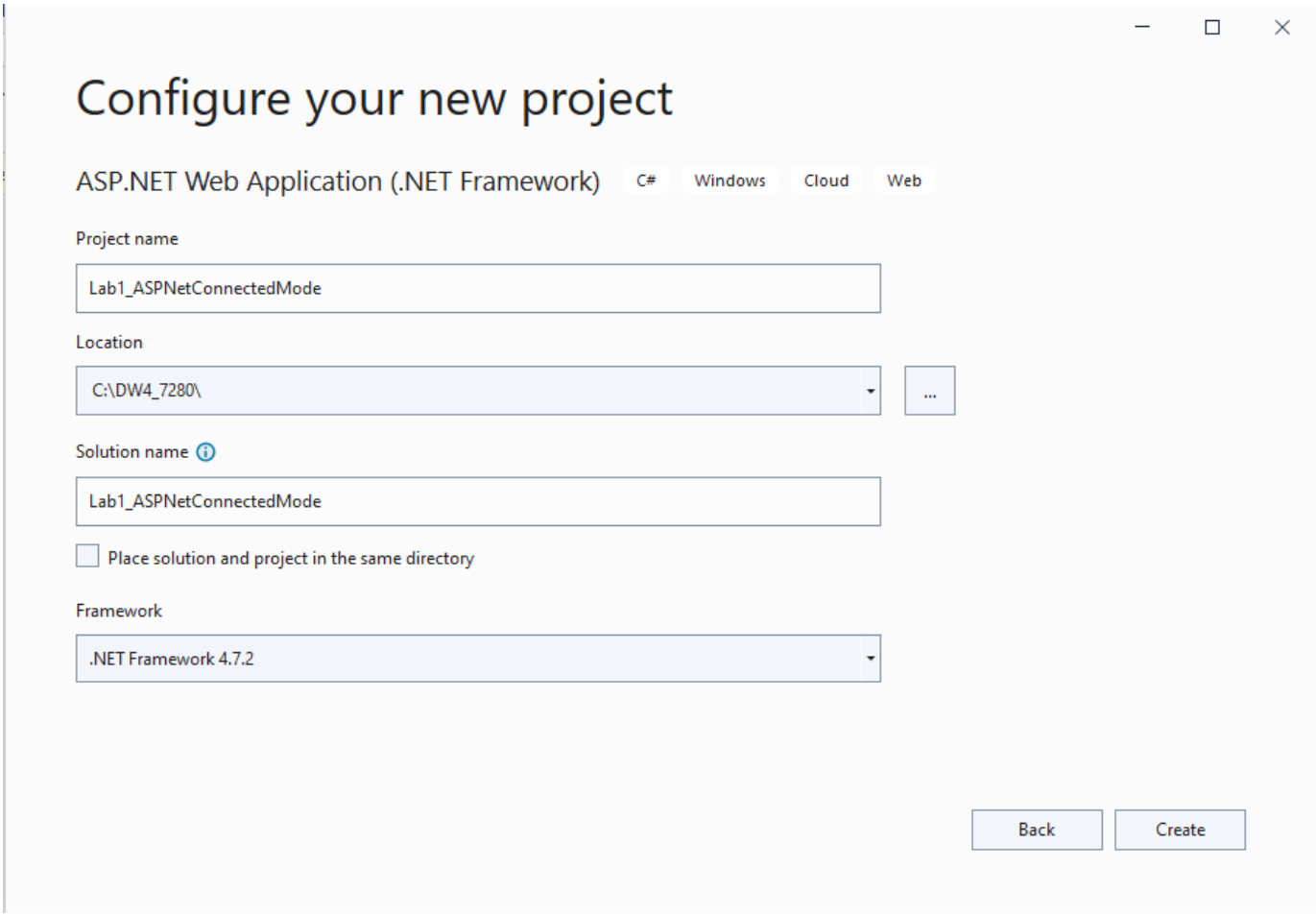
	EmployeeId	FirstName	LastName	JobTitle
	1111	John	Doe	Programmer Analyst
	2222	Mary	Ford	System Administrator
	3333	Mary	Ford	Programmer
	4444	Mary	Brown	Programmer
	5555	John	Green	Programmer
	6666	Mary	Brown	Programmer Analyst
	7777	Karan	Sood	Programmer
	8888	Kanwar	Singh	System Analyst
▶*	*NULL*	*NULL*	*NULL*	*NULL*

Step 2 Creating the ASP.Net Web Application using Visual Studio 2022, C#

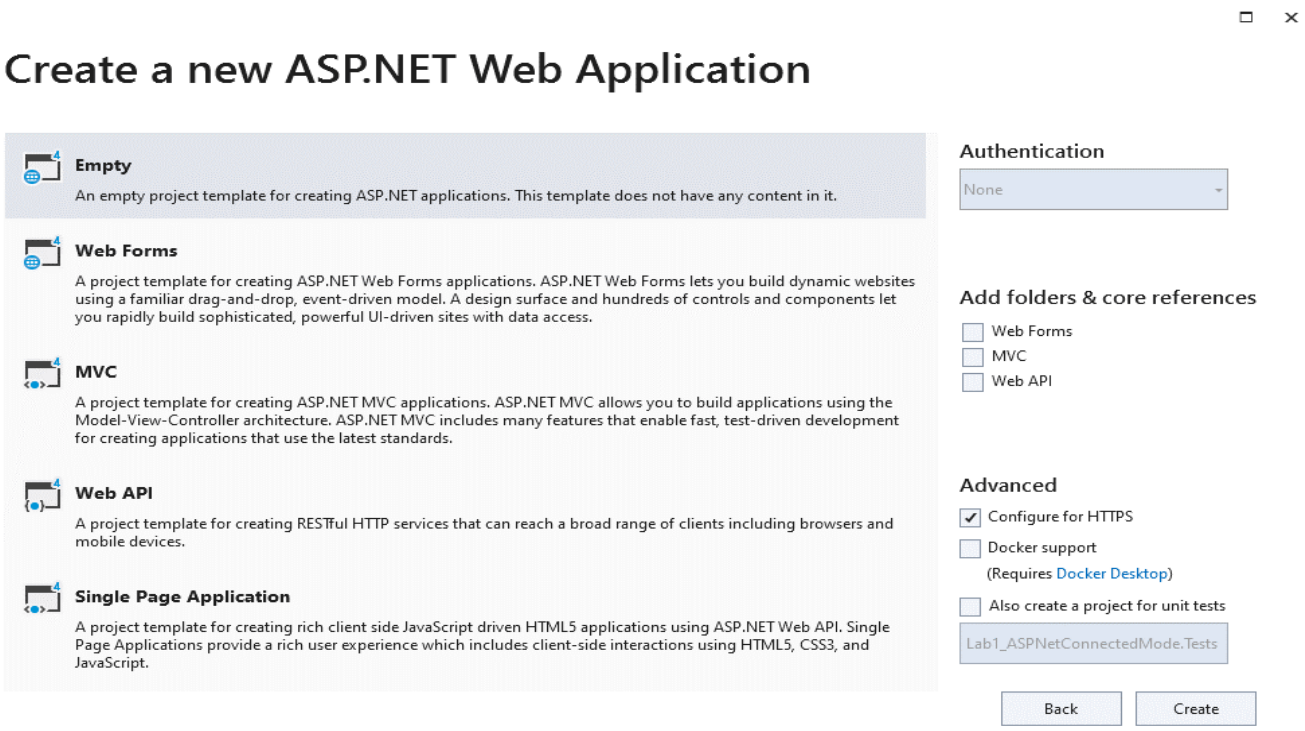
Select ASP.Net Web Application (.NET Framework) C#

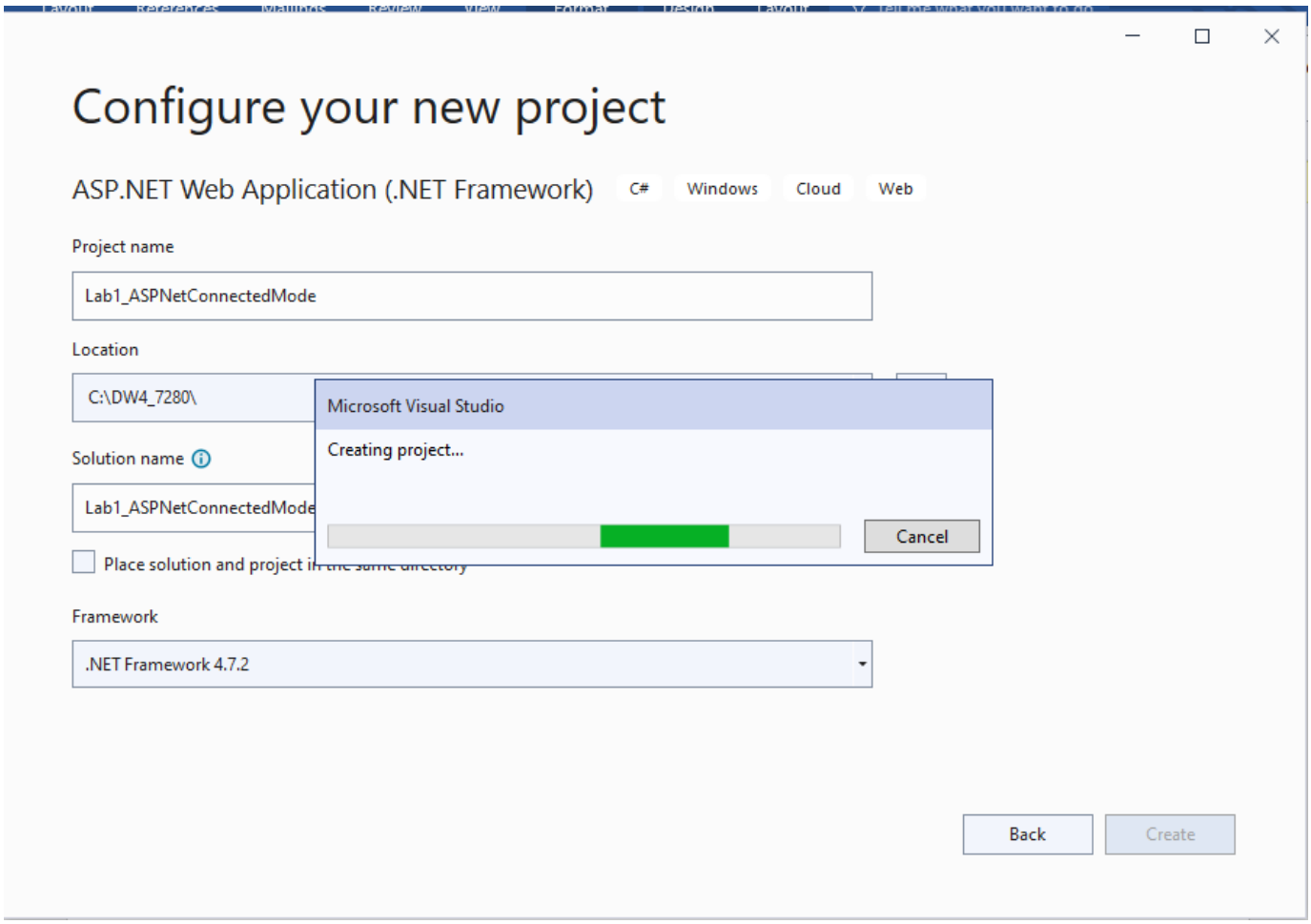


Project Name: Lab1_ASPNetConnectedMode



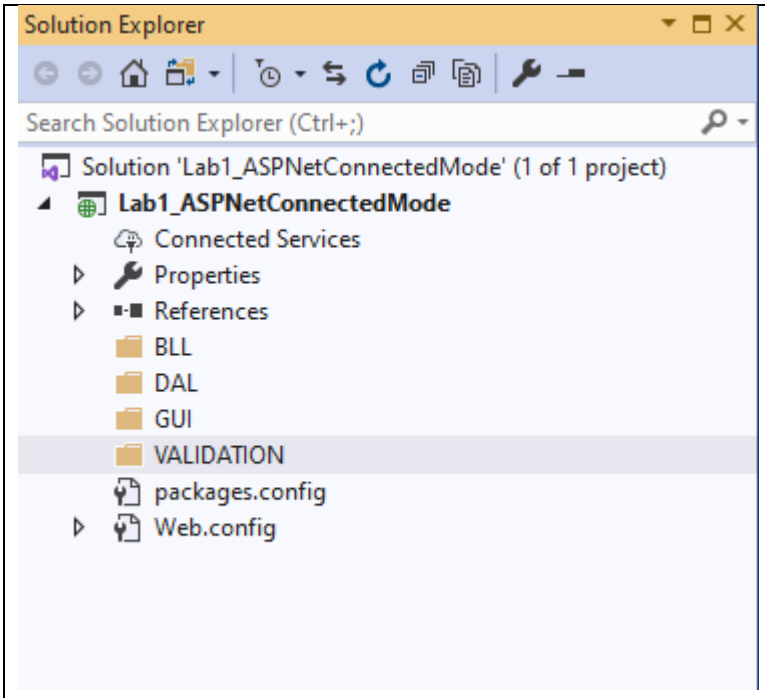
Select Empty and click Create


Create a new ASP.NET Web Application
Empty
An empty project template for creating ASP.NET applications. This template does not have any content in it.
Web Forms
A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.
MVC
A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.
Web API
A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.
Single Page Application
A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.
Authentication
None
Add folders & core references
Web Forms
MVC
Web API
Advanced
Configure for HTTPS
Docker support
(Requires Docker Desktop)
Also create a project for unit tests
Lab1_ASPNetConnectedMode.Tests
Back
Create



- Right-Click Project Folder's Name (**Lab1_ASPNetConnectedMode**) and add 4 folders
    1. **BLL** (Business Logic Layer)
    2. **DAL** (Data Access Layer)
    3. **GUI** (Graphical User Interface)
    4. **VALIDATION**

Your project structure looks like this:



BLL : This folder contains all the business classes required for the project.
e.g Employee.cs

DAL : All the data access classes are included in this folder.
e.g
EmployeeDB.cs
UtilityDB.cs

GUI: This folder contain all the Web Forms Classes required for the project
e.g FormEmployee.cs

VALIDATION: This folder stores all the input validation methods

Step 3 Developing the Business Class(es)

Recommendations :
I strongly recommend you take your time to review the following topics already discussed in the previous course (420-CT2-AS)

1. Class ,Objects, Properties and Methods
2. Default constructors; Parameterized constructors
3. Abstraction, Encapsulation
4. Inheritance, Base Class, Derived Class
5. Polymorphism
6. Method Overloading
7. Method Overriding
8. Class Relationships (Inheritance, Aggregation, Composition)

For Lab1, to keep it simple, you add only one business class.

Right-Click the folder **BLL** and add the following class to this folder.

Class: **Employee.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace Lab1_ASPNetConnectedMode.BLL
{
    public class Employee
    {
        //private class variables
        private int employeeId;
```

```
        private string firstName;
        private string lastName;
        private string jobTitle;
        //properties
        public int EmployeeId { get => employeeId; set => employeeId = value; }
        public string FirstName { get => firstName; set => firstName = value; }
        public string LastName { get => lastName; set => lastName = value; }
        public string JobTitle { get => jobTitle; set => jobTitle = value; }
        //custom methods

    }
}
```

Step 4 Developing the Data Access Class(es)

Recommendation : Review the ADO.Net (A: Activex, D: Data, O: Object) Object Model

1. Class SqlConnection (Properties and Methods)
2. Class SqlCommand (Properties and Methods)
3. Class SqlDataReader (Properties and Methods)
4. SQL Statements (INSERT, UPDATE, DELETE, SELECT)

If you forget the syntax, the following web site will help you to review:

SQL Keywords Reference

https://www.w3schools.com/sql/sql_ref_keywords.asp

Right-Click the folder **DAL** and add the following class to this folder.

Class: **UtilityDB.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data.SqlClient;


namespace Lab1_ASPNetConnectedMode.DAL
{
    public class UtilityDB
    {
        //Version 1: Working but not good.Why? Another better solution?
        /// <summary>
        /// This method returns an active database connection
        /// </summary>
        /// <returns>object of type SqlConnection</returns>
        public static SqlConnection ConnectDB()
        {
            SqlConnection conn = new SqlConnection();
            conn.ConnectionString = "server=TITANCAO-
PC\\MSSQLSERVER2017;database=EmployeeDB;user=sa;password=lasalle";
            conn.Open();
```

```
            return conn;
        }

    }
}
```

Class: **EmployeeDB.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data.SqlClient;
using Lab1_ASPNetConnectedMode.BLL;

namespace Lab1_ASPNetConnectedMode.DAL
{
    public class EmployeeDB
    {
        // <summary>
        // This method saves an Employee object data to the database; Version 1
        // </summary>
        // <param name = "emp" ></ param >
        public static void SaveRecord(Employee emp)
        {
            SqlConnection connDB = UtilityDB.ConnectDB();
            SqlCommand cmd = new SqlCommand();
            cmd.Connection = connDB;
            cmd.CommandText = "INSERT INTO Employees " +
                              "VALUES (" + emp.EmployeeId + ",'" +
                                           emp.FirstName + "','" +
                                           emp.LastName + "','" +
                                           emp.JobTitle + "')";
            cmd.ExecuteNonQuery();
            connDB.Close();
        }

    }
}
```

Using the method **SaveRecord**() in the business class **Employee.cs**
Modify the code in the class **Employee.cs**

```
public class Employee
    {
        private int employeeId;
        private string firstName;
        private string lastName;
        private string jobTitle;

        public int EmployeeId { get => employeeId; set => employeeId = value; }
        public string FirstName { get => firstName; set => firstName = value; }
        public string LastName { get => lastName; set => lastName = value; }
        public string JobTitle { get => jobTitle; set => jobTitle = value; }

         public void SaveEmployee(Employee emp)
         {
            EmployeeDB.SaveRecord(emp);
```

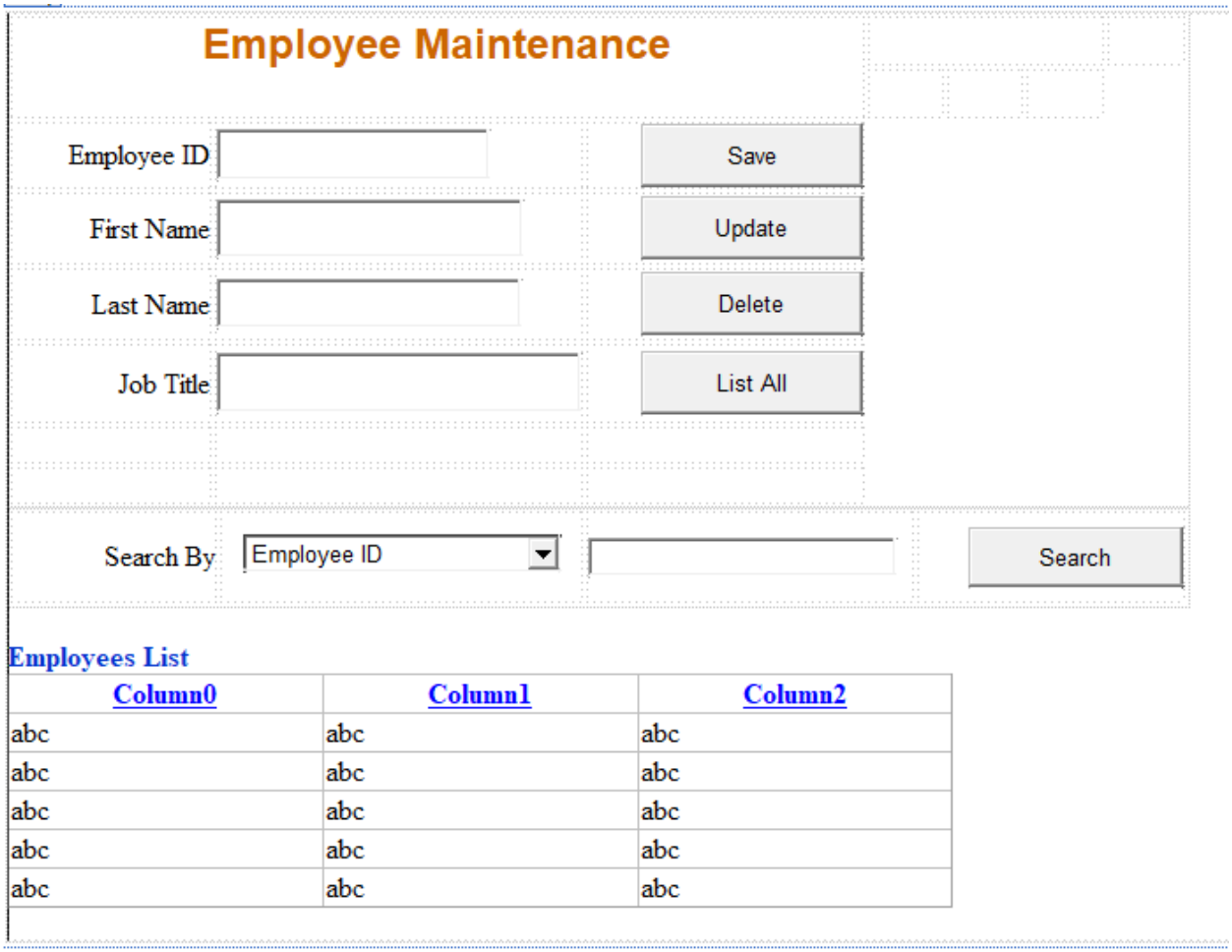
```
        }
    }
```

Step 5 Developing the GUI Class(es)

Right-Click the folder **GUI** and add the following classes to this folder.

1. Add a new Form: **WebFormEmployee.cs**

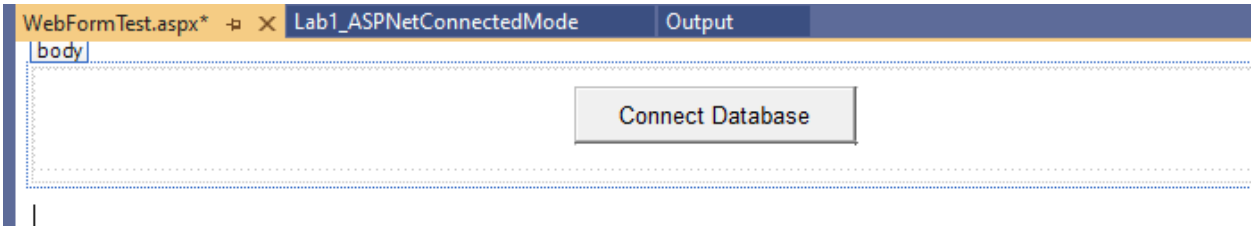
And design the form as follows



Inportant Note : You will use this Web form to implement all the CRUD operations required by the project.

2. Add a another new Form: **WebFormTest.cs**

Use this Web form to test the *database connection* to make sure your app can connect the SQL Server database **EmployeeDB.mdf** created in **Step 1**.



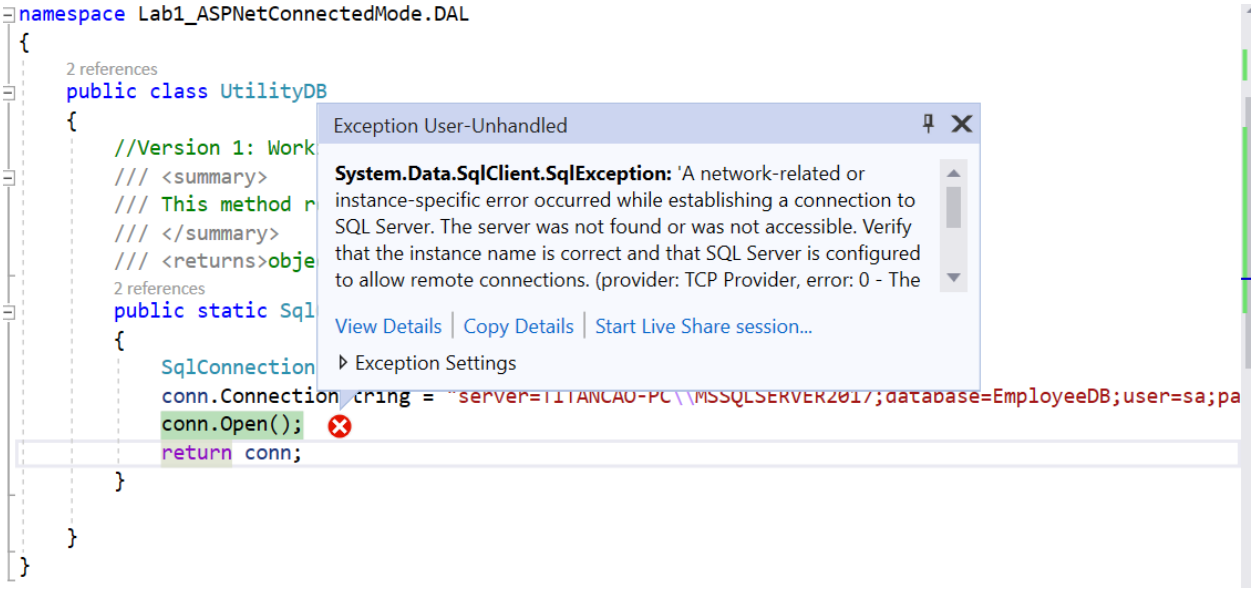

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Lab1_ASPNetConnectedMode.DAL; //just for testing the database connection
using System.Data.SqlClient; // for using ADO.Net classes with Sql Server database
using System.Windows.Forms;// for using MessageBox.Show()

namespace Lab1_ASPNetConnectedMode.GUI
{
    public partial class WebFormTest : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void ButtonConnectDB_Click(object sender, EventArgs e)
        {
            SqlConnection connDB = UtilityDB.ConnectDB();
            MessageBox.Show(Convert.ToString(connDB.State), "Database Connection");

        }

    }
}
```

Step 6 Testing the Application

1. Testing the Database Connection

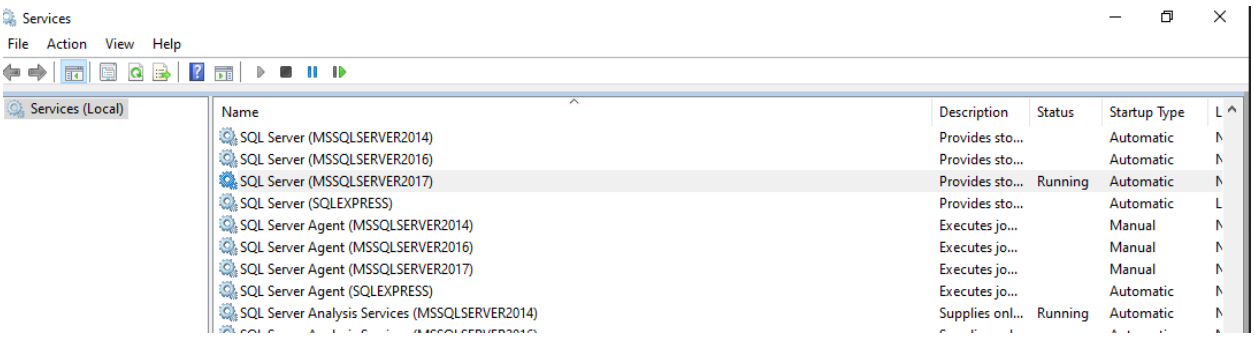
Right-Click WebFormTest.aspx, select **Set as Start Page**

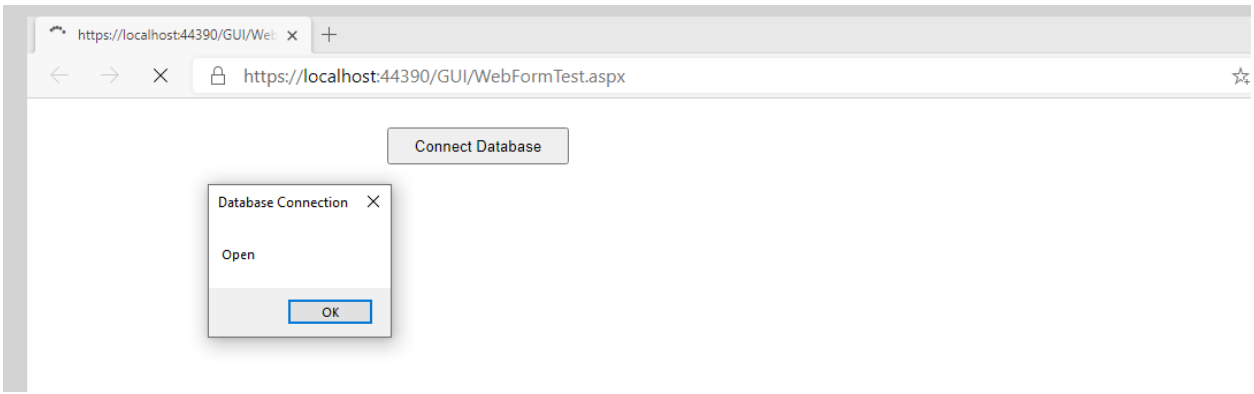
Run the application



How to fix the problem?

Solution: *Start the SQL Server 2022/2019/2017 services*



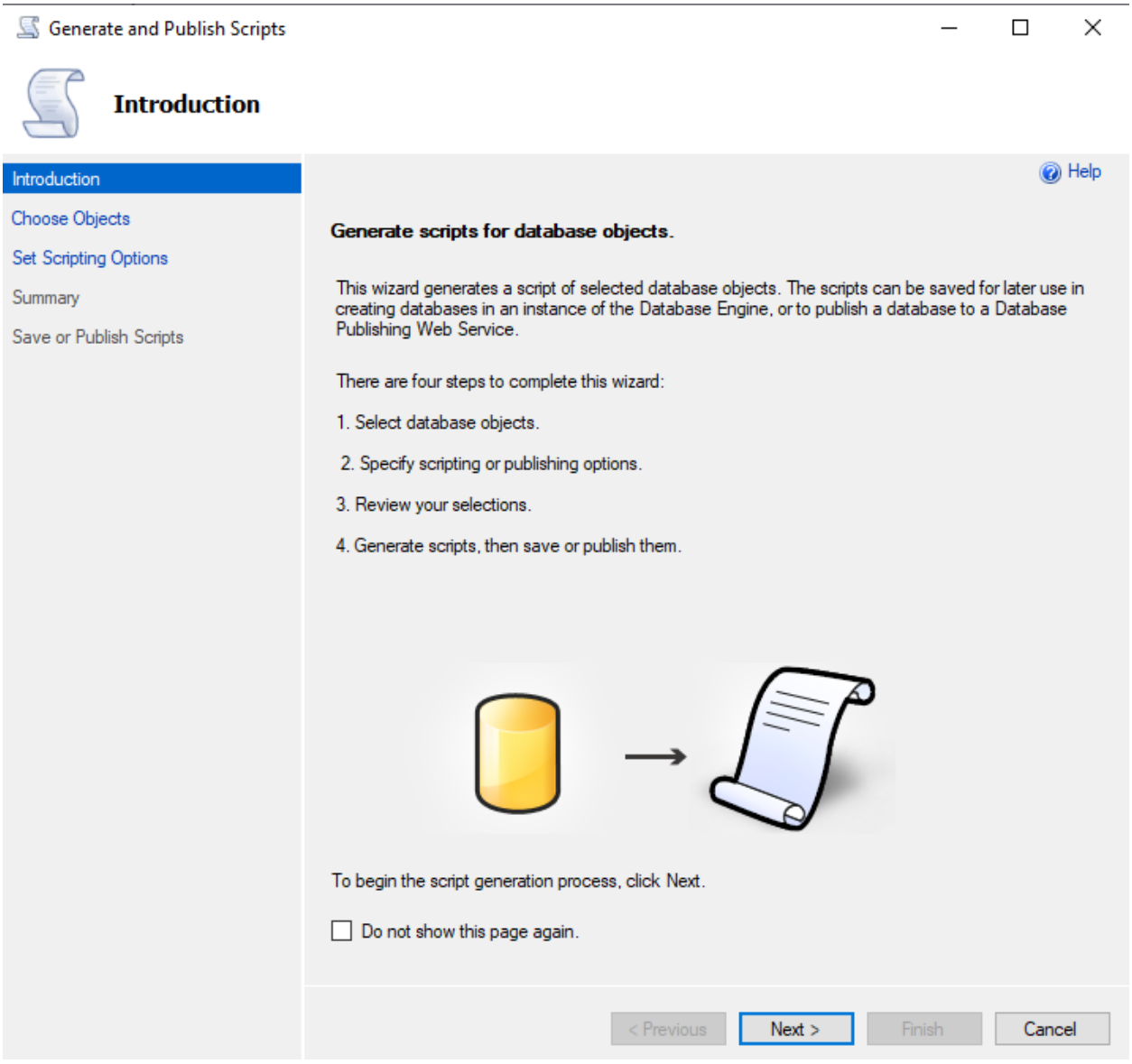


2. Testing the CRUD Operations

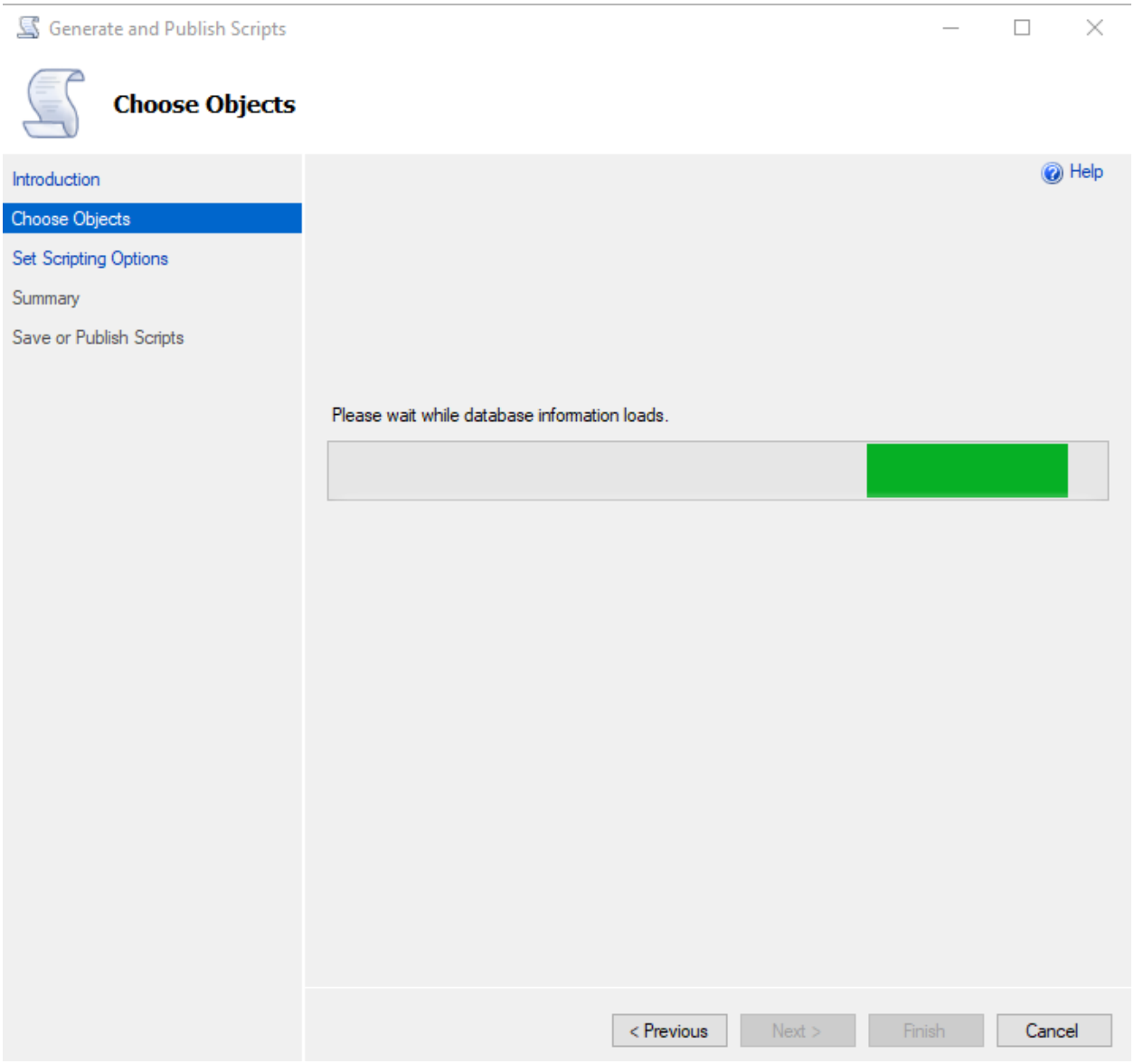
Step 7 Submitting Lab1 via LEA

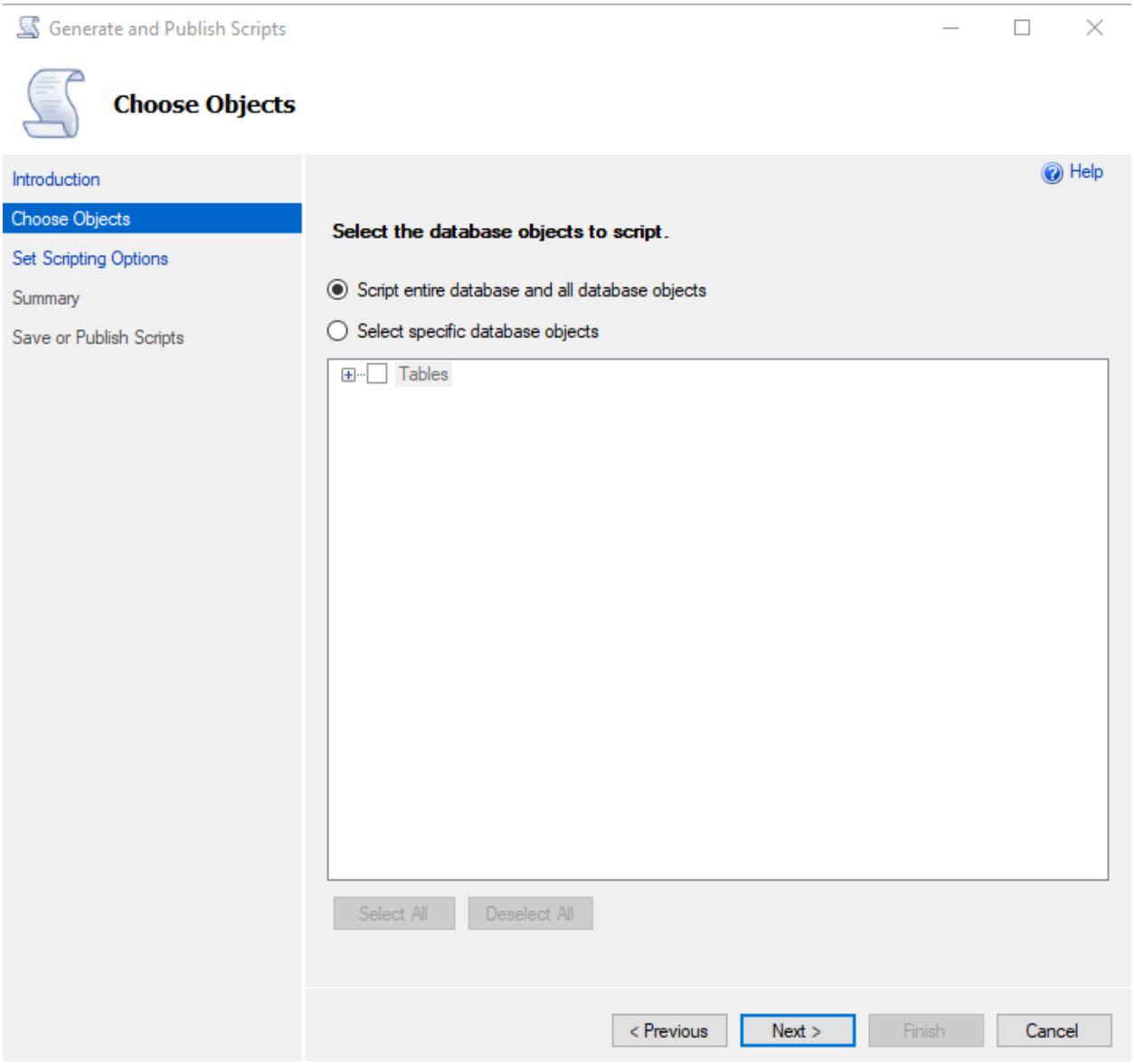
To submit your Lab1, follow these steps

1. Open SQL Server Management 17 (or higher version)
2. Right-Click the database name **EmployeeDB**
3. Select **Tasks**, then **Generate Scripts**

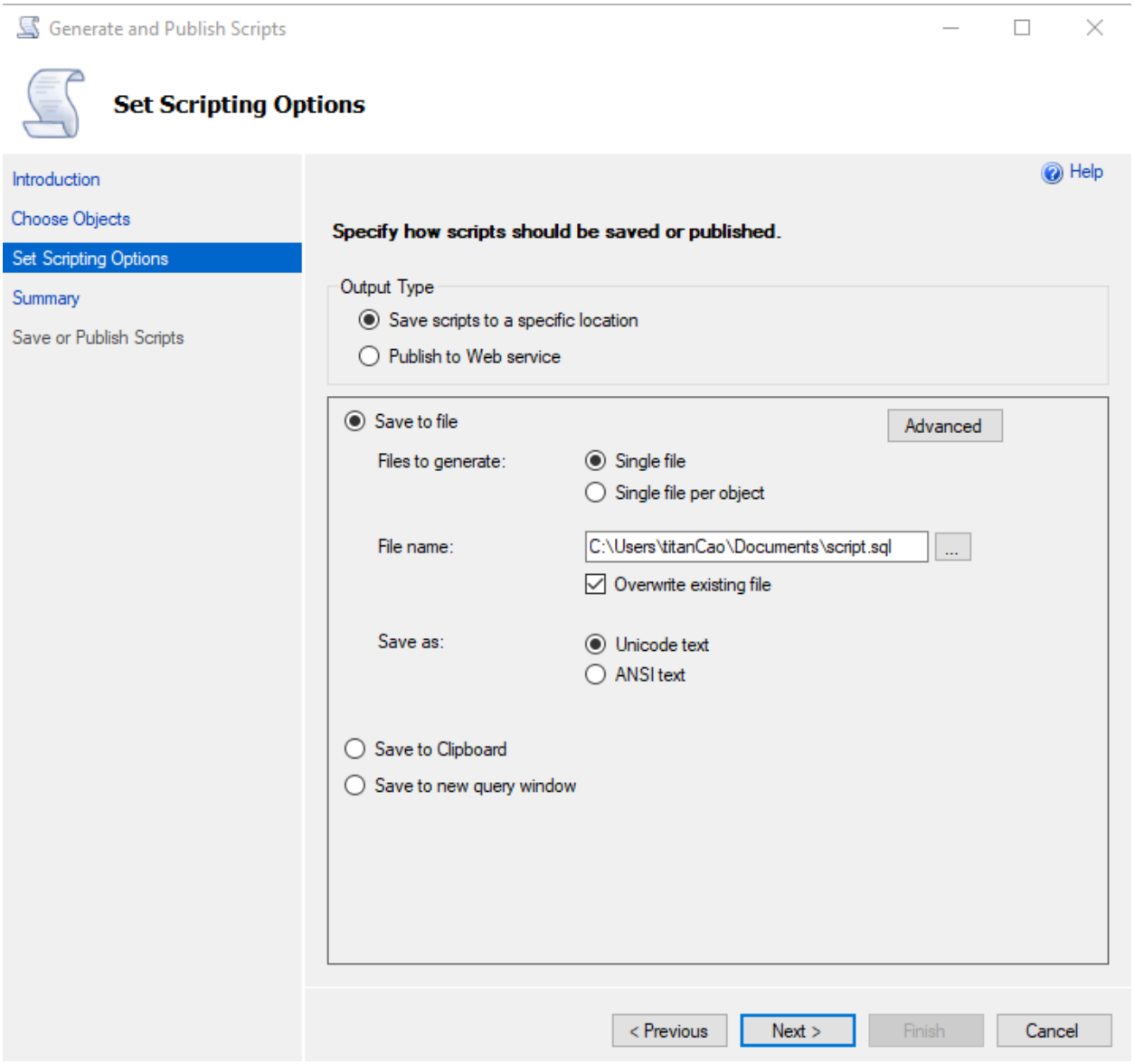


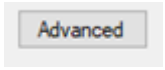
Click **Next**

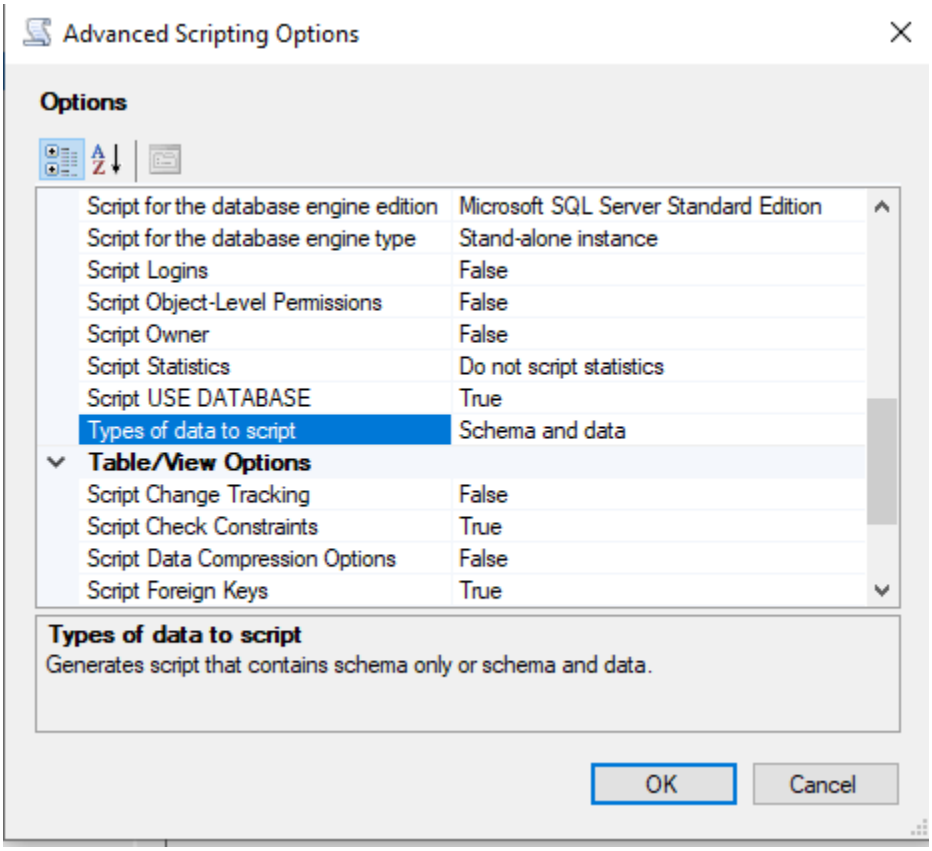

Generate and Publish Scripts
Choose Objects
Introduction
Choose Objects
Set Scripting Options
Summary
Save or Publish Scripts
Help
Please wait while database information loads.
< Previous
Next >
Finish
Cancel


Generate and Publish Scripts
Choose Objects
Introduction
Choose Objects
Set Scripting Options
Summary
Save or Publish Scripts
Help
Select the database objects to script.
Script entire database and all database objects
Select specific database objects
Tables
Select All
Deselect All
< Previous
Next >
Finish
Cancel

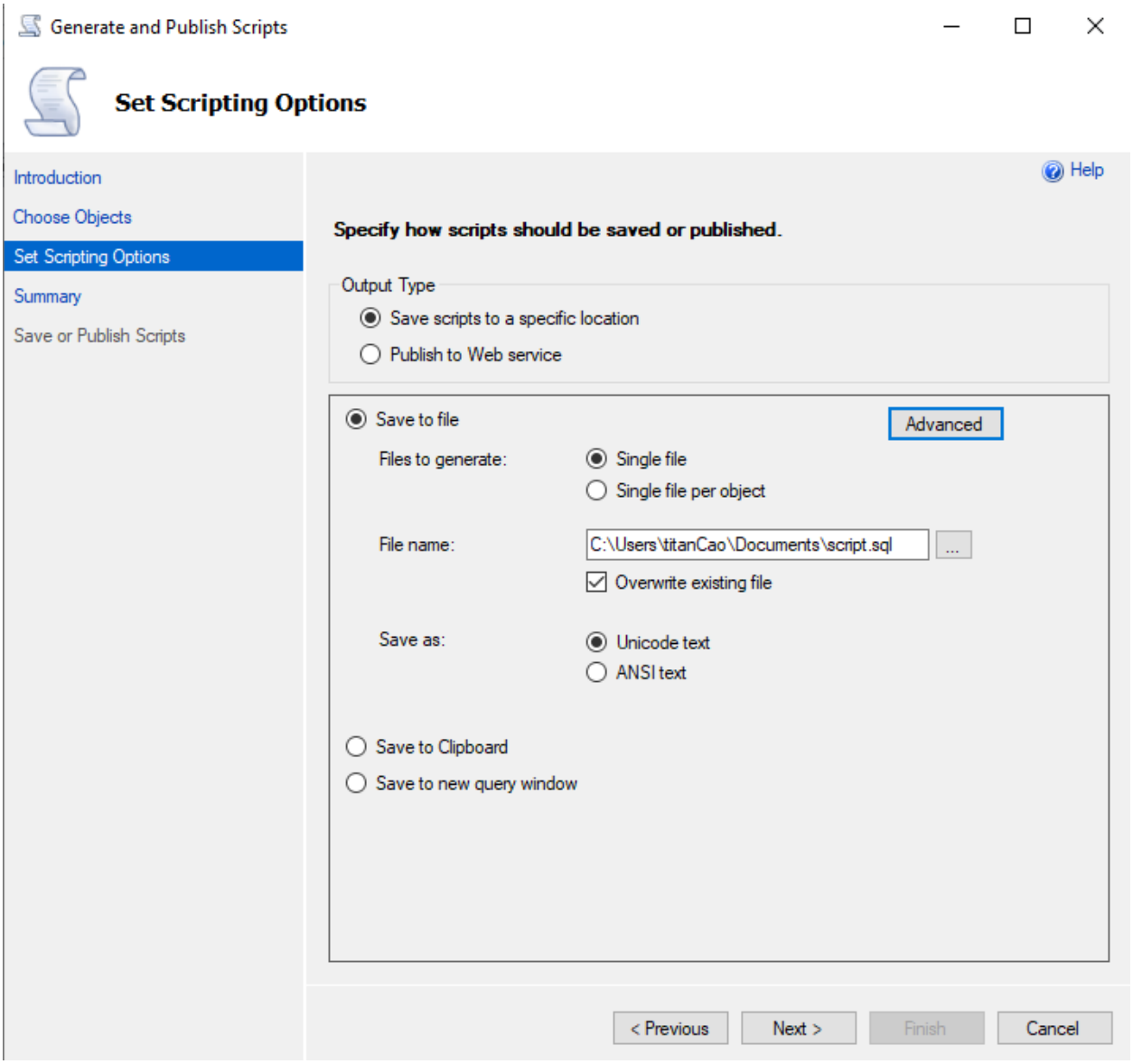
Click **Next**



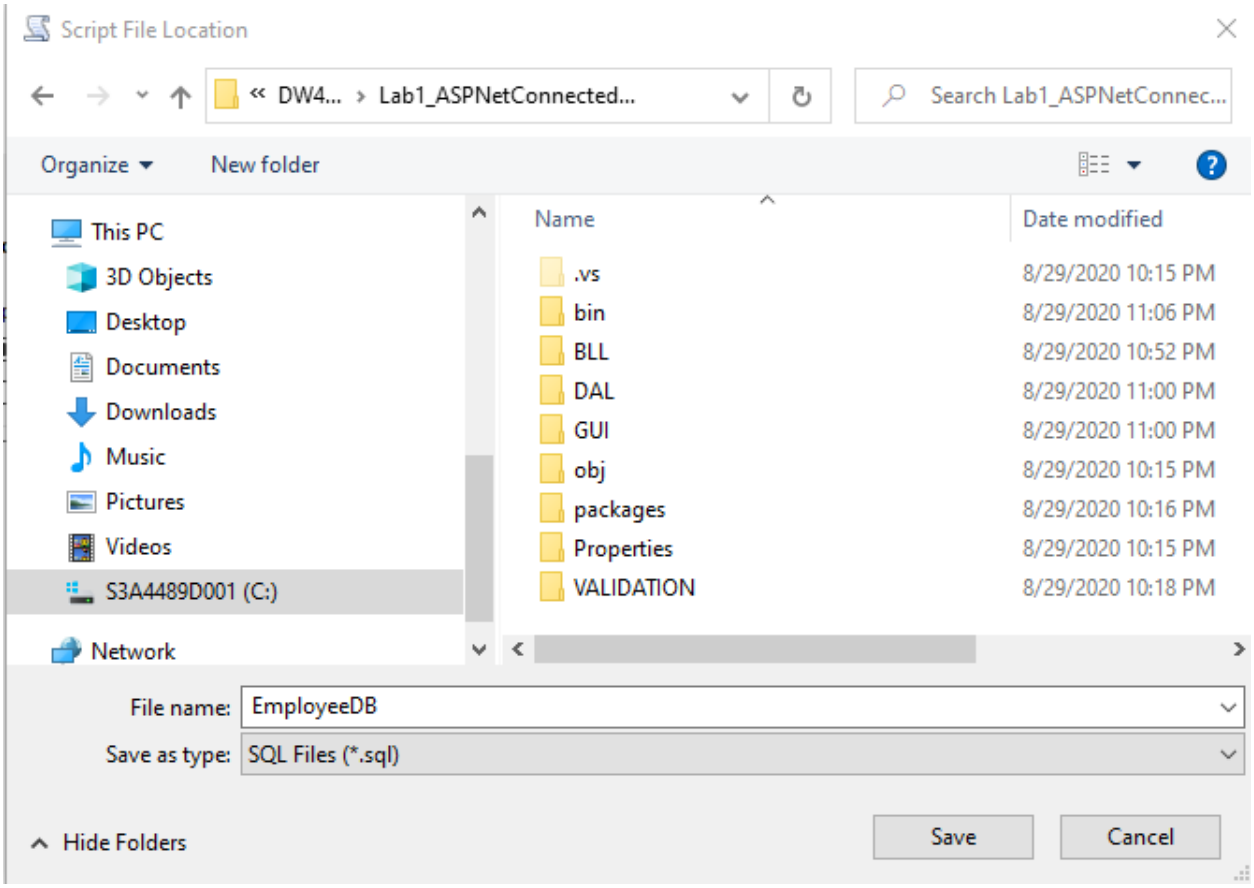
Click the button 


Advanced Scripting Options
Options
Script for the database engine edition
Microsoft SQL Server Standard Edition
Script for the database engine type
Stand-alone instance
Script Logins
False
Script Object-Level Permissions
False
Script Owner
False
Script Statistics
Do not script statistics
Script USE DATABASE
True
Types of data to script
Schema and data
Table/View Options
Script Change Tracking
False
Script Check Constraints
True
Script Data Compression Options
False
Script Foreign Keys
True
Types of data to script
Generates script that contains schema only or schema and data.
OK
Cancel

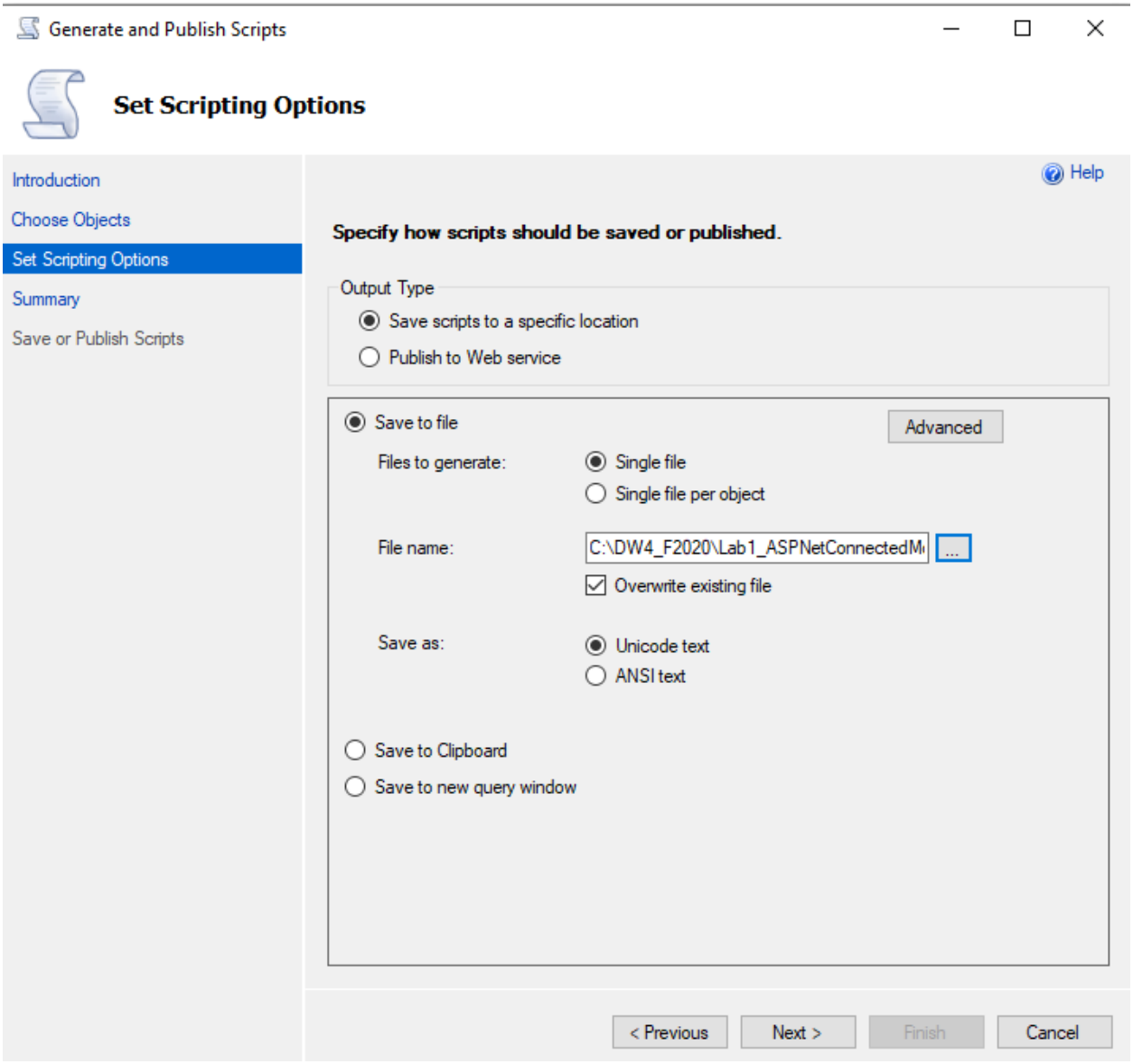
Click **OK**



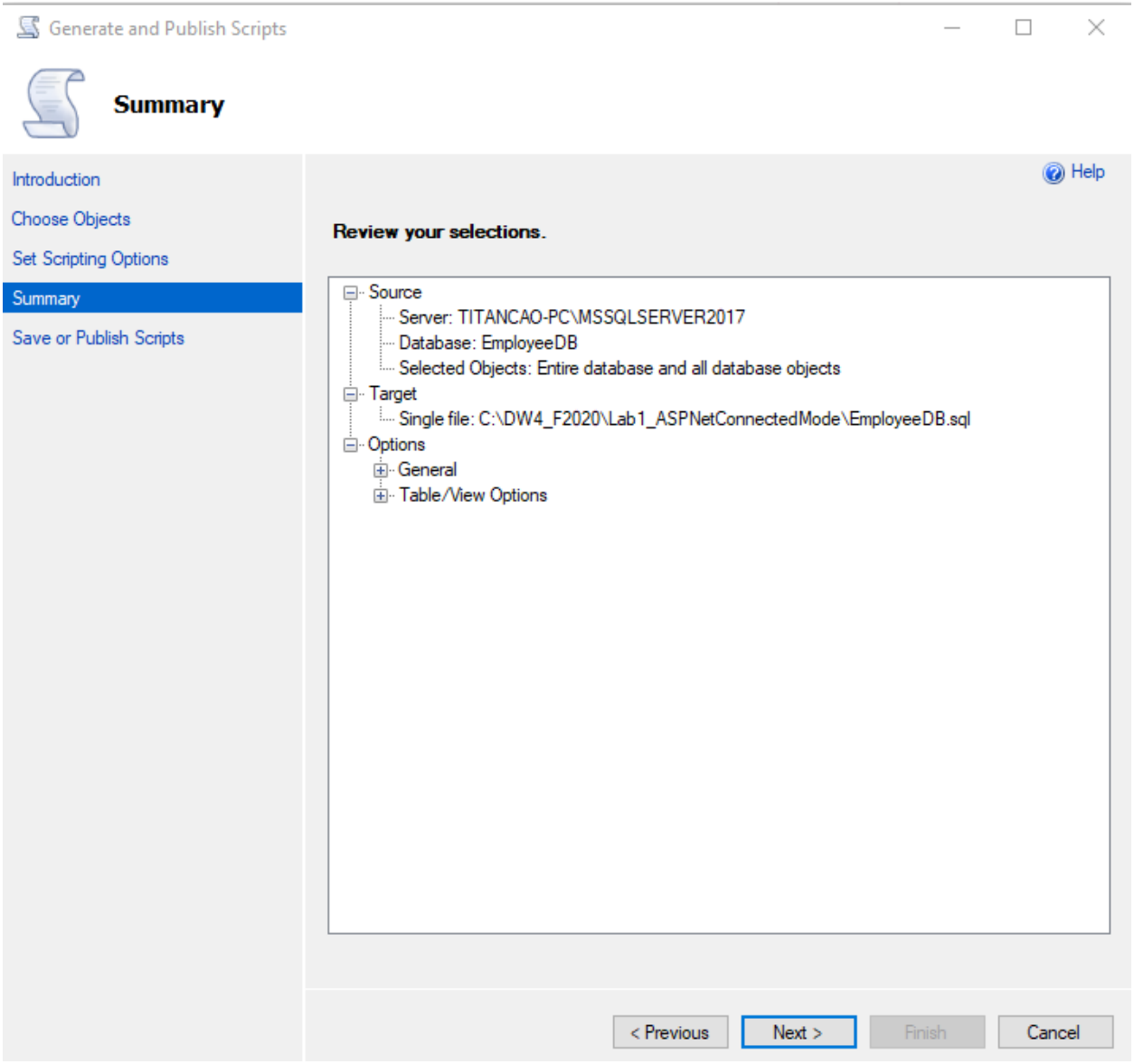
Click ... to save the script file to your project folder


Script File Location
« DW4... › Lab1_ASPNetConnected...
Search Lab1_ASPNetConnec...
Organize
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
S3A4489D001 (C:)
Network
Name
Date modified
.vs 8/29/2020 10:15 PM
bin 8/29/2020 11:06 PM
BLL 8/29/2020 10:52 PM
DAL 8/29/2020 11:00 PM
GUI 8/29/2020 11:00 PM
obj 8/29/2020 10:15 PM
packages 8/29/2020 10:16 PM
Properties 8/29/2020 10:15 PM
VALIDATION 8/29/2020 10:18 PM
File name: EmployeeDB
Save as type: SQL Files (*.sql)
Hide Folders
Save
Cancel

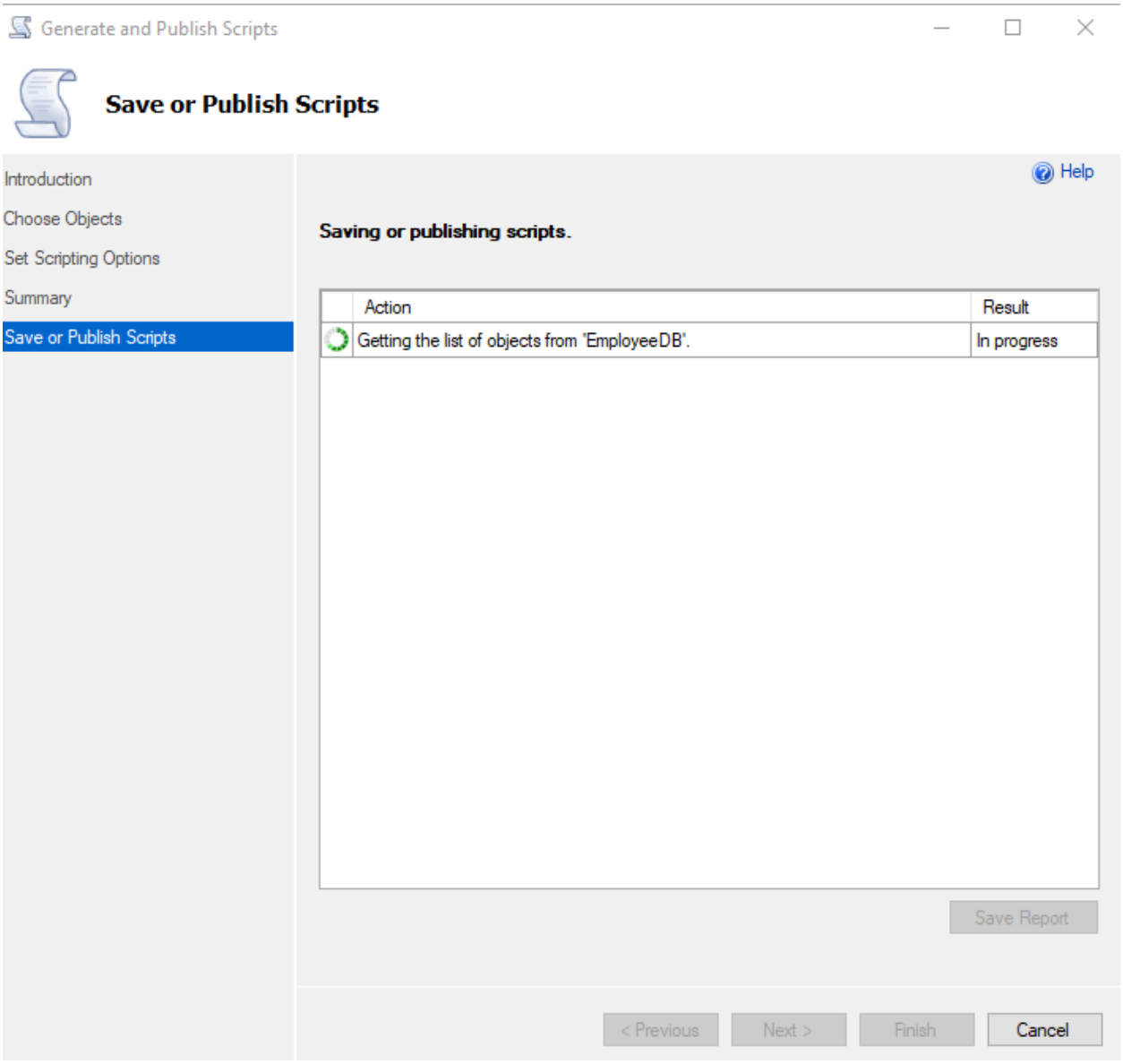
Click **Save**

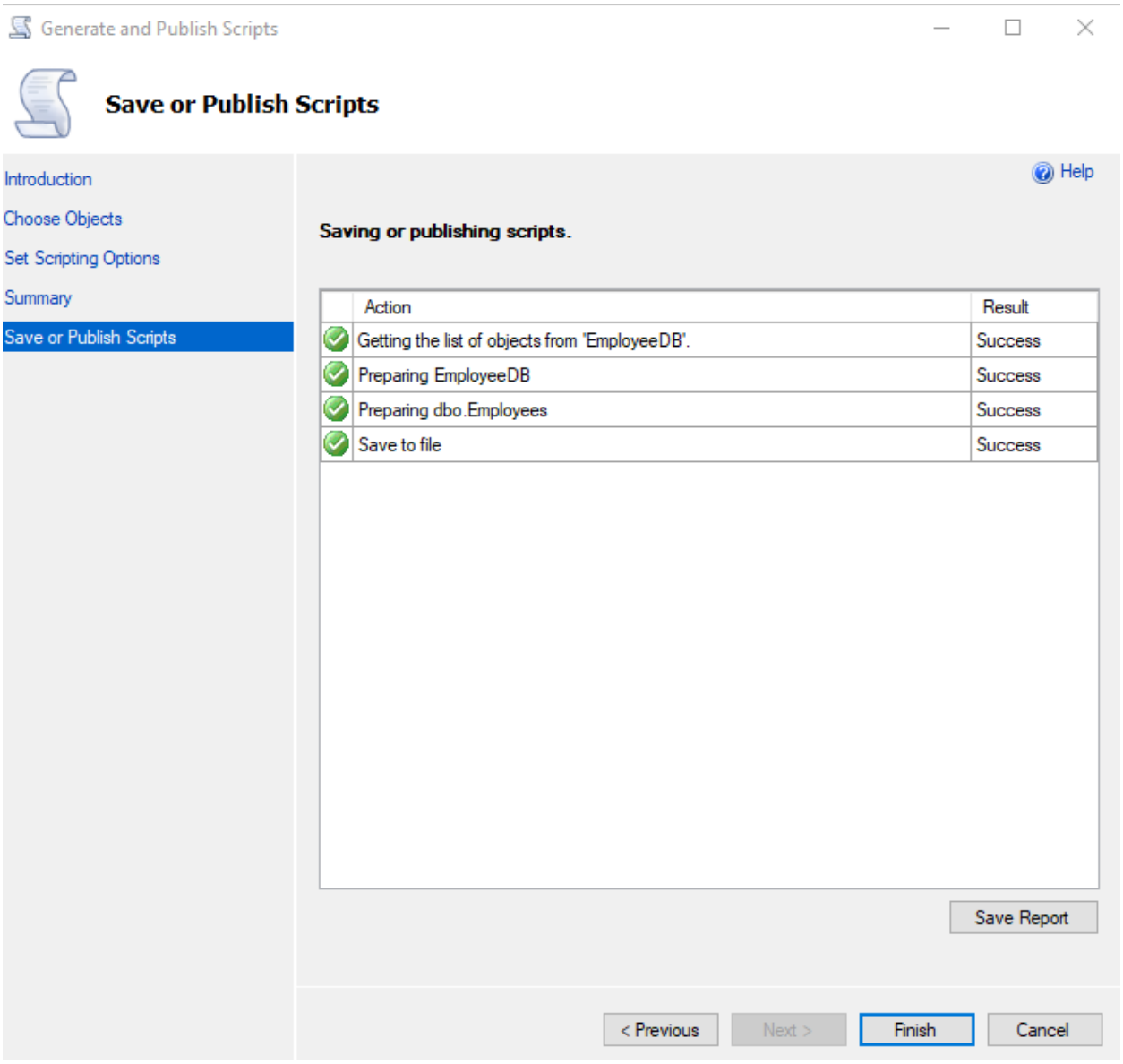

Generate and Publish Scripts
Set Scripting Options
Introduction
Choose Objects
Set Scripting Options
Summary
Save or Publish Scripts
Help
Specify how scripts should be saved or published.
Output Type
Save scripts to a specific location
Publish to Web service
Save to file
Advanced
Files to generate:
Single file
Single file per object
File name:
C:\DW4_F2020\Lab1_ASPNetConnectedM
Overwrite existing file
Save as:
Unicode text
ANSI text
Save to Clipboard
Save to new query window
< Previous
Next >
Finish
Cancel

Click **Next**


Generate and Publish Scripts
Summary
Introduction
Choose Objects
Set Scripting Options
Summary
Save or Publish Scripts
Help
Review your selections.
Source
Server: TITANCAO-PC\MSSQLSERVER2017
Database: EmployeeDB
Selected Objects: Entire database and all database objects
Target
Single file: C:\DW4_F2020\Lab1_ASPNetConnectedMode\EmployeeDB.sql
Options
General
Table/View Options
< Previous
Next >
Finish
Cancel

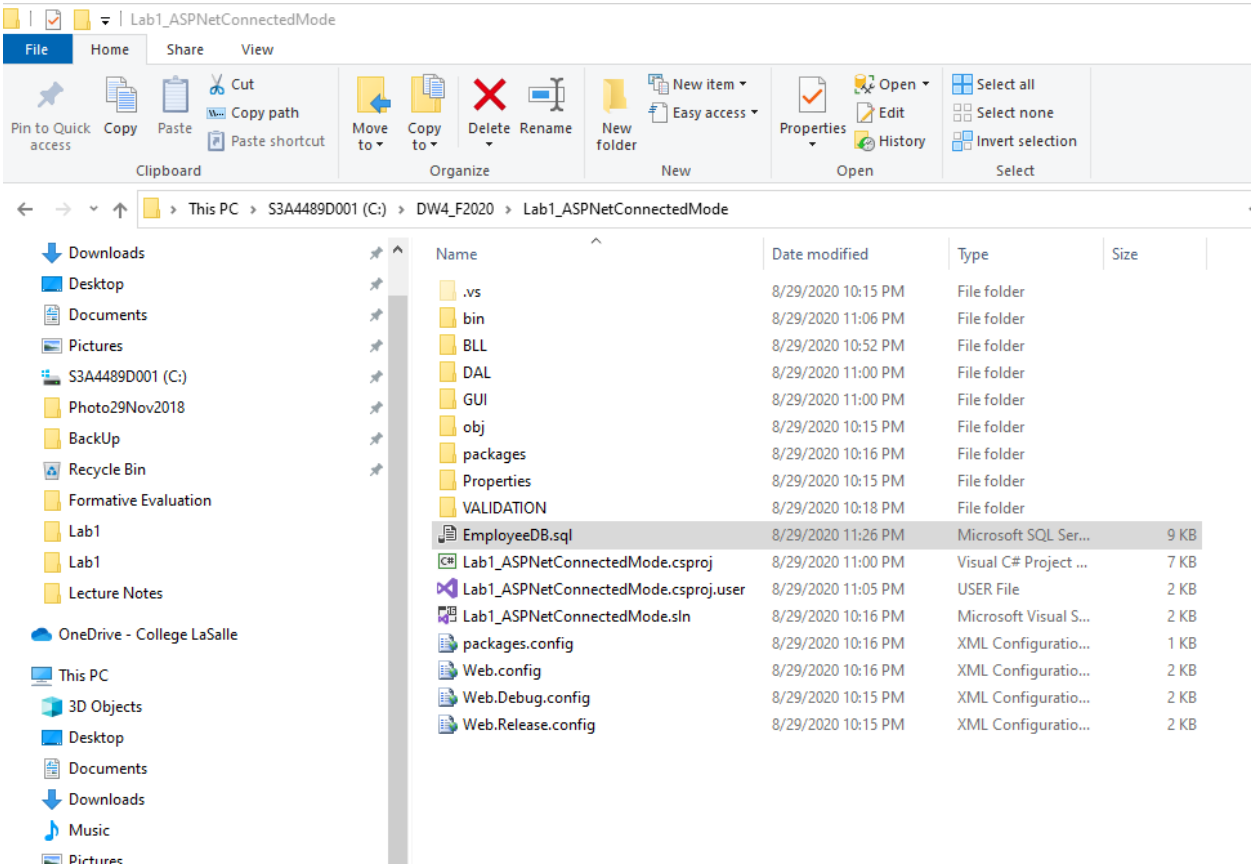
Click **Next**


Generate and Publish Scripts
Save or Publish Scripts
Introduction
Choose Objects
Set Scripting Options
Summary
Save or Publish Scripts
Help
Saving or publishing scripts.
Action
Result
Getting the list of objects from 'EmployeeDB'.
In progress
Save Report
< Previous
Next >
Finish
Cancel

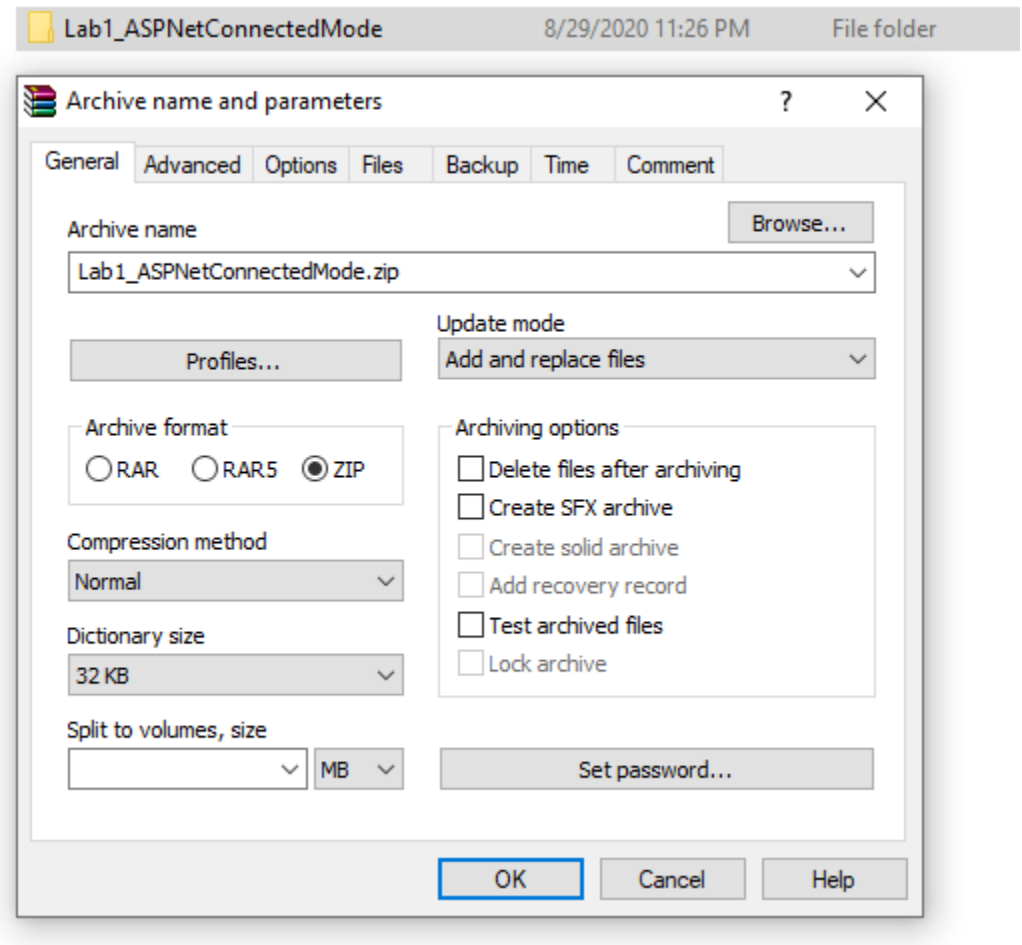


Click **Finish**

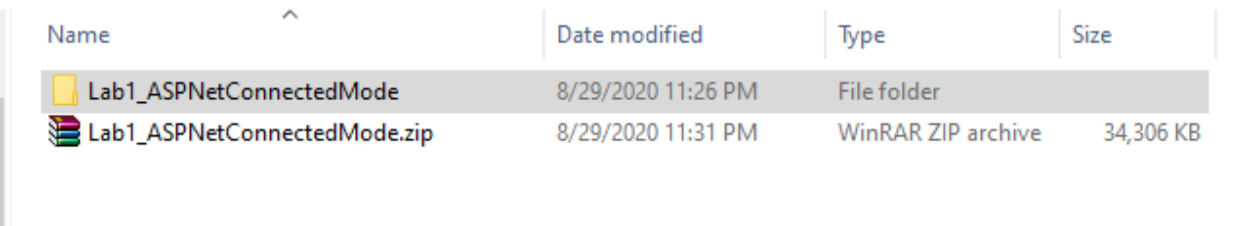
Check the project folder


Lab1_ASPNetConnectedMode
File
Home
Share
View
Pin to Quick access
Copy
Paste
Cut
Copy path
Paste shortcut
Clipboard
Move to
Copy to
Delete
Rename
Organize
New folder
New item
Easy access
New
Properties
Open
Edit
History
Open
Select all
Select none
Invert selection
Select
This PC › S3A4489D001 (C:) › DW4_F2020 › Lab1_ASPNetConnectedMode
Downloads
Desktop
Documents
Pictures
S3A4489D001 (C:)
Photo29Nov2018
BackUp
Recycle Bin
Formative Evaluation
Lab1
Lab1
Lecture Notes
OneDrive - College LaSalle
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Name
Date modified
Type
Size
.vs 8/29/2020 10:15 PM File folder
bin 8/29/2020 11:06 PM File folder
BLL 8/29/2020 10:52 PM File folder
DAL 8/29/2020 11:00 PM File folder
GUI 8/29/2020 11:00 PM File folder
obj 8/29/2020 10:15 PM File folder
packages 8/29/2020 10:16 PM File folder
Properties 8/29/2020 10:15 PM File folder
VALIDATION 8/29/2020 10:18 PM File folder
EmployeeDB.sql 8/29/2020 11:26 PM Microsoft SQL Ser... 9 KB
Lab1_ASPNetConnectedMode.csproj 8/29/2020 11:00 PM Visual C# Project ... 7 KB
Lab1_ASPNetConnectedMode.csproj.user 8/29/2020 11:05 PM USER File 2 KB
Lab1_ASPNetConnectedMode.sln 8/29/2020 10:16 PM Microsoft Visual S... 2 KB
packages.config 8/29/2020 10:16 PM XML Configuratio... 1 KB
Web.config 8/29/2020 10:16 PM XML Configuratio... 2 KB
Web.Debug.config 8/29/2020 10:15 PM XML Configuratio... 2 KB
Web.Release.config 8/29/2020 10:15 PM XML Configuratio... 2 KB

Click **Back**



Right-Click the project folder and make a zipped file.



Access Omnivox and send the zipped file via LEA.