Course Number: 420-DA3-AS
Course Title: Multi-tier Applications Development
Teacher: Quang Hoang Cao
Session: Autumn 2022

Formative Evaluation
LAB 4

Creating and Using the Entity Framework

Due Date: November 15, 2022 (Demo in class and submission via LEA)

Objective

- Basic concepts of Conceptual Model
- Mapping Model and Storage Model
- Creating an Entity Data Model
- Using the Entity Data Model Designer
- Using LINQ to Entities.

Performance criteria

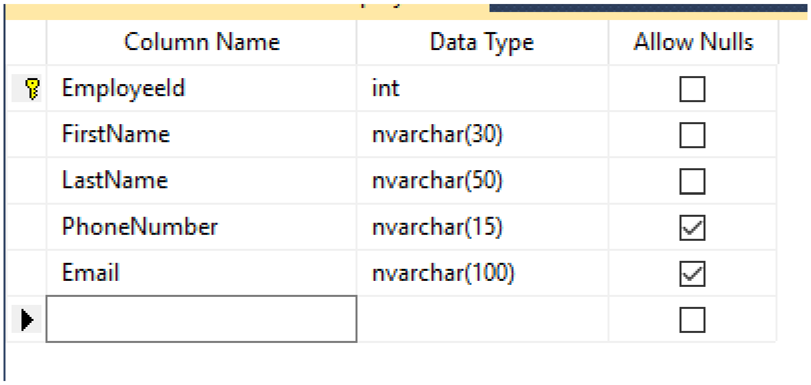
1.2 Proper identification of the tasks to be carried out
2.2 Proper configuration of the target platform
3.1 Proper creation or adaptation of the local or remote database
3.2 Proper insertion of initial or test data
3.3 Compliance with the data model
4.1 Appropriate choice and use of graphic elements for display and input
4.3 Adaptation of the interface based on the display format and resolution
5.2 Proper programming of interactions between the graphical user interface and the user
5.3 Appropriate choice of clauses, operators, commands or parameters in database queries
5.4 Correct handling of database data
6.1 Precise application of test plans in the emulator and on the target platform
6.3 Relevance of the corrective actions

- Critical thinking, methodical, analytic and synthetic;
- Programming efficiency;
- Autonomy, initiative.
- Time management

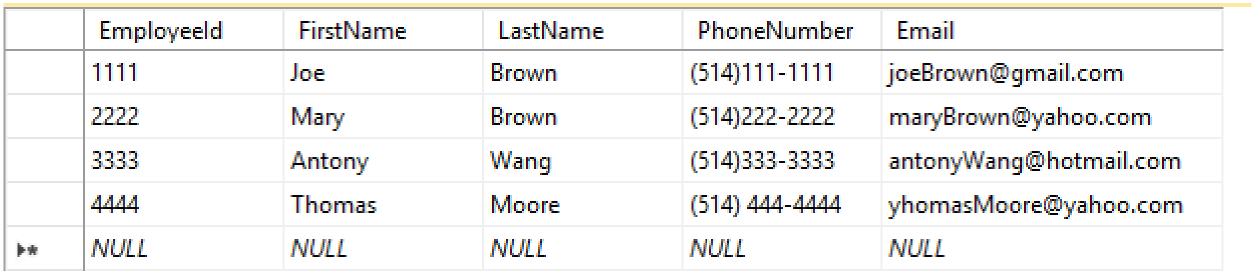
Part I: Creating the Database with SQL Server 2017

Database Name: **EmployeeProjectDB**
Tables:

- **Employees**

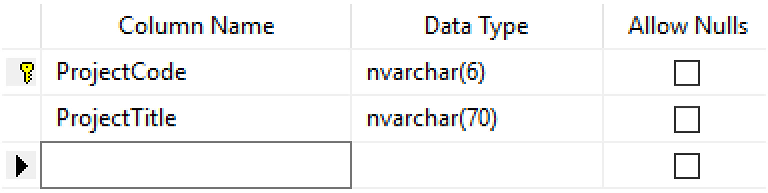


	Column Name	Data Type	Allow Nulls
🔑	EmployeeId	int	□
	FirstName	nvarchar(30)	□
	LastName	nvarchar(50)	□
	PhoneNumber	nvarchar(15)	☑
	Email	nvarchar(100)	☑
▶			□

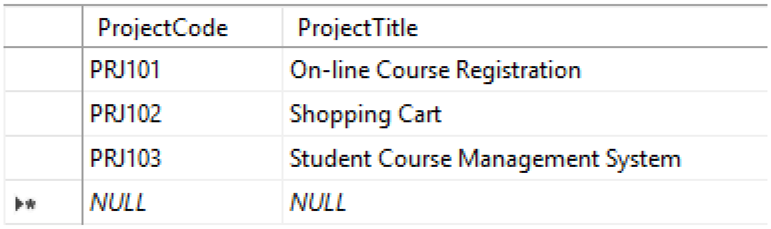


	EmployeeId	FirstName	LastName	PhoneNumber	Email
	1111	Joe	Brown	(514)111-1111	joeBrown@gmail.com
	2222	Mary	Brown	(514)222-2222	maryBrown@yahoo.com
	3333	Antony	Wang	(514)333-3333	antonyWang@hotmail.com
	4444	Thomas	Moore	(514) 444-4444	yhomasMoore@yahoo.com
▶*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

- **Projects**

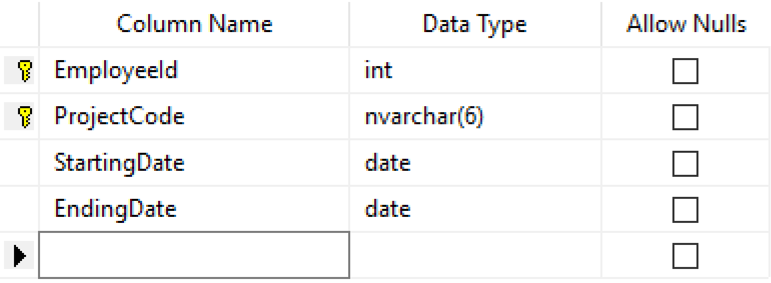


	Column Name	Data Type	Allow Nulls
🔑	ProjectCode	nvarchar(6)	□
	ProjectTitle	nvarchar(70)	□
▶			□

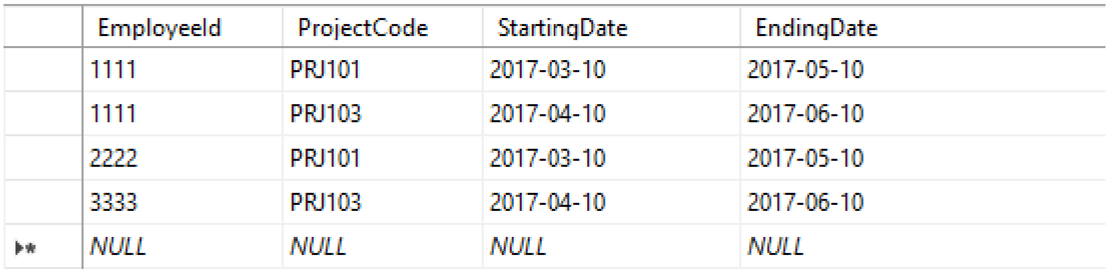


	ProjectCode	ProjectTitle
	PRJ101	On-line Course Registration
	PRJ102	Shopping Cart
	PRJ103	Student Course Management System
▶*	*NULL*	*NULL*

- **ProjectAssignments**

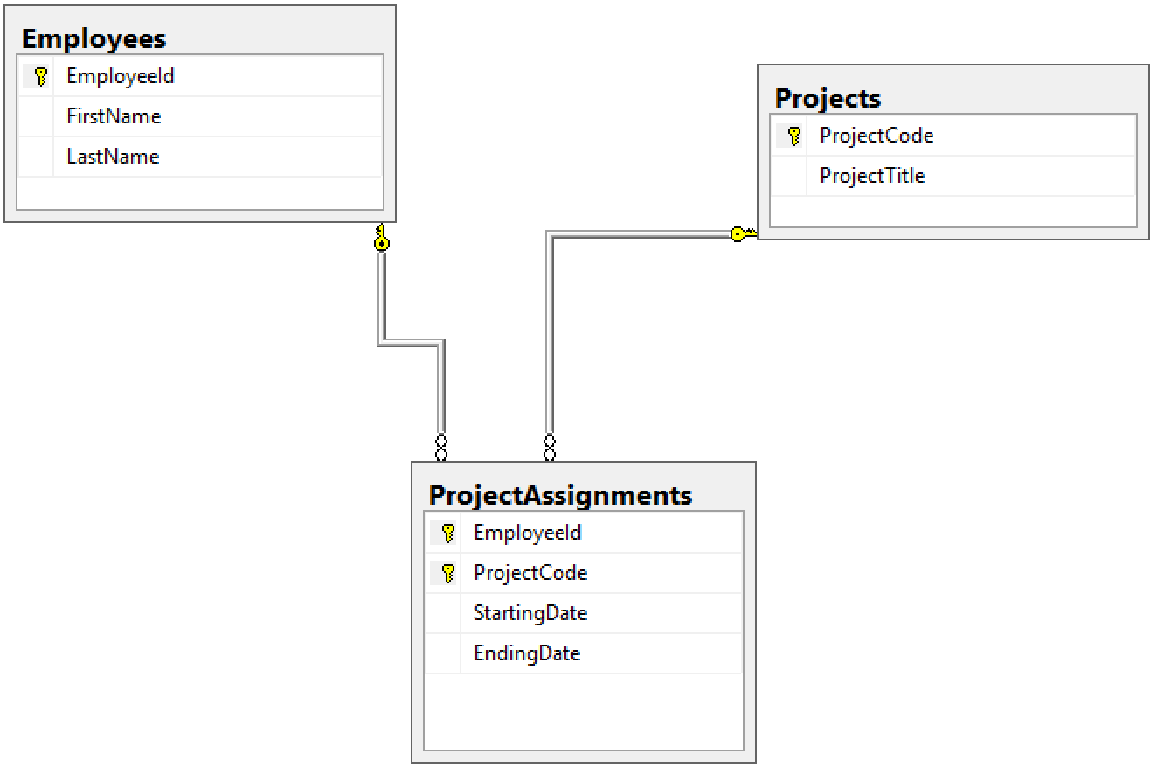


	Column Name	Data Type	Allow Nulls
🔑	EmployeeId	int	☐
🔑	ProjectCode	nvarchar(6)	☐
	StartingDate	date	☐
	EndingDate	date	☐
▶			☐



	EmployeeId	ProjectCode	StartingDate	EndingDate
	1111	PRJ101	2017-03-10	2017-05-10
	1111	PRJ103	2017-04-10	2017-06-10
	2222	PRJ101	2017-03-10	2017-05-10
	3333	PRJ103	2017-04-10	2017-06-10
▶*	*NULL*	*NULL*	*NULL*	*NULL*

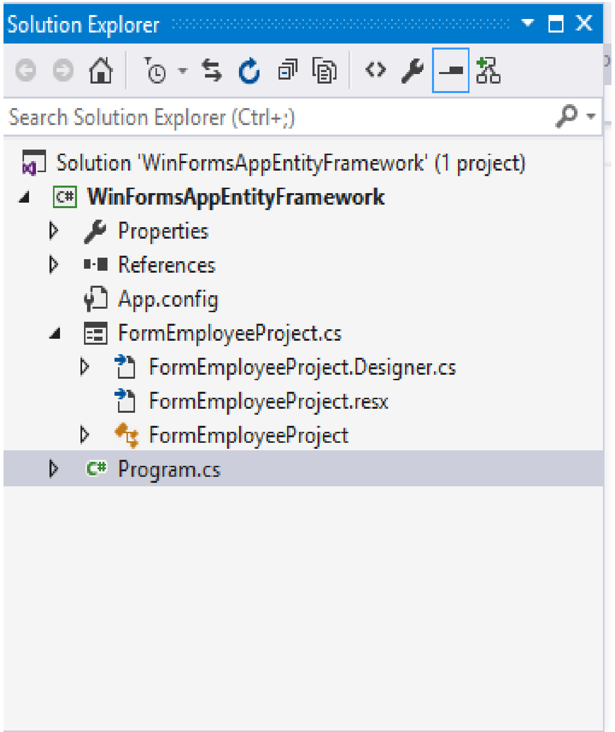
- **E-R Diagram**



Part II: Developing Windows Form Application

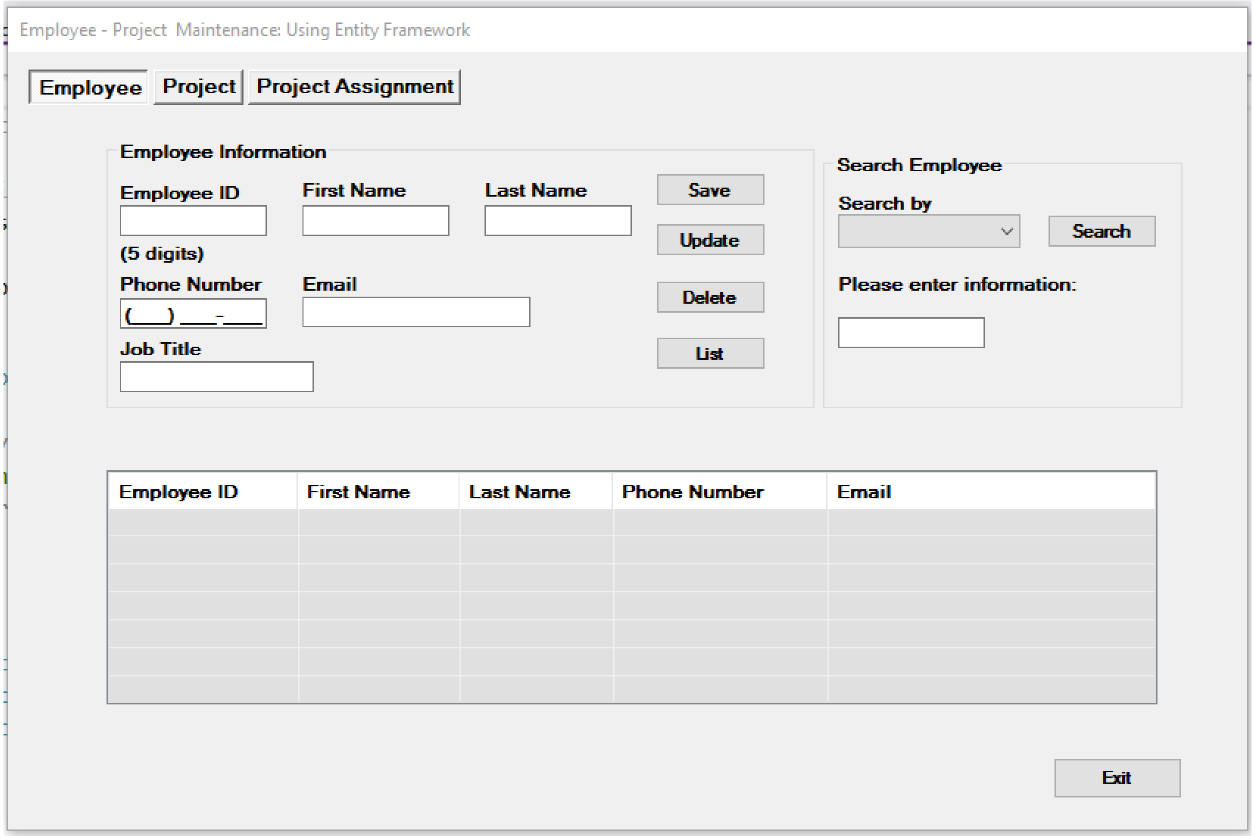
A. Creating a Windows Form Application

Project Name: **WinFormAppEntityFramework**



Solution 'WinFormsAppEntityFramework' (1 project)
WinFormsAppEntityFramework
Properties
References
App.config
FormEmployeeProject.cs
FormEmployeeProject.Designer.cs
FormEmployeeProject.resx
FormEmployeeProject
Program.cs

B. Creating the following Form Class

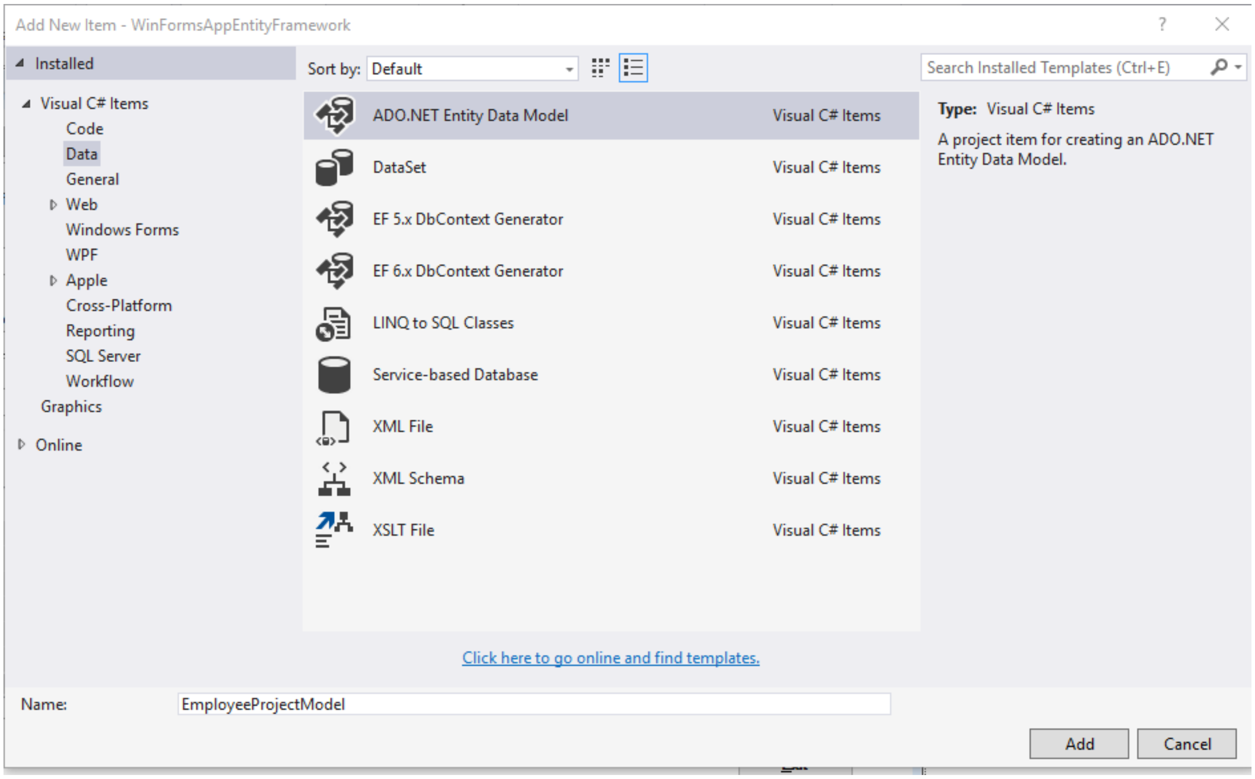


C. Creating and Using Entity Framework : Database First

Microsoft has given the following definition of Entity Framework:

The ***Microsoft ADO.NET Entity Framework is an Object/Relational Mapping (ORM) framework*** that enables developers to work with relational data as domain-specific objects, eliminating the need for most of the data access plumbing code that developers usually need to write. ***Using the Entity Framework, developers issue queries using LINQ, then retrieve and manipulate data as strongly typed objects***. The Entity Framework's ORM implementation provides services like change tracking, identity resolution, lazy loading, and query translation so that developers can ***focus on their application-specific business logic rather than the data access fundamentals***.

Adding ADO.NET Entity Data Model



Add New Item - WinFormsAppEntityFramework

Sort by: Default

ADO.NET Entity Data Model	Visual C# Items
DataSet	Visual C# Items
EF 5.x DbContext Generator	Visual C# Items
EF 6.x DbContext Generator	Visual C# Items
LINQ to SQL Classes	Visual C# Items
Service-based Database	Visual C# Items
XML File	Visual C# Items
XML Schema	Visual C# Items
XSLT File	Visual C# Items

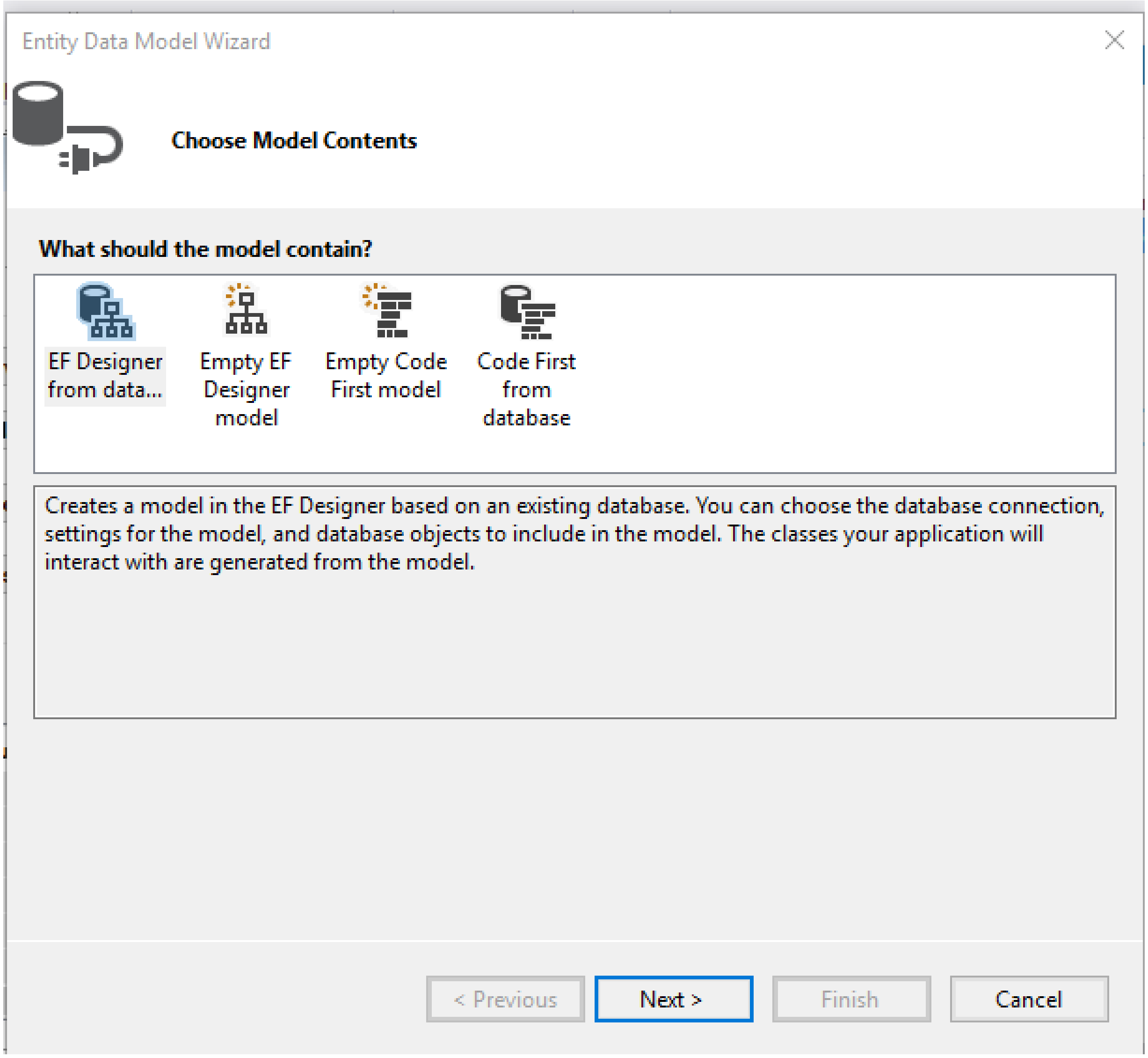
Type: Visual C# Items

A project item for creating an ADO.NET Entity Data Model.

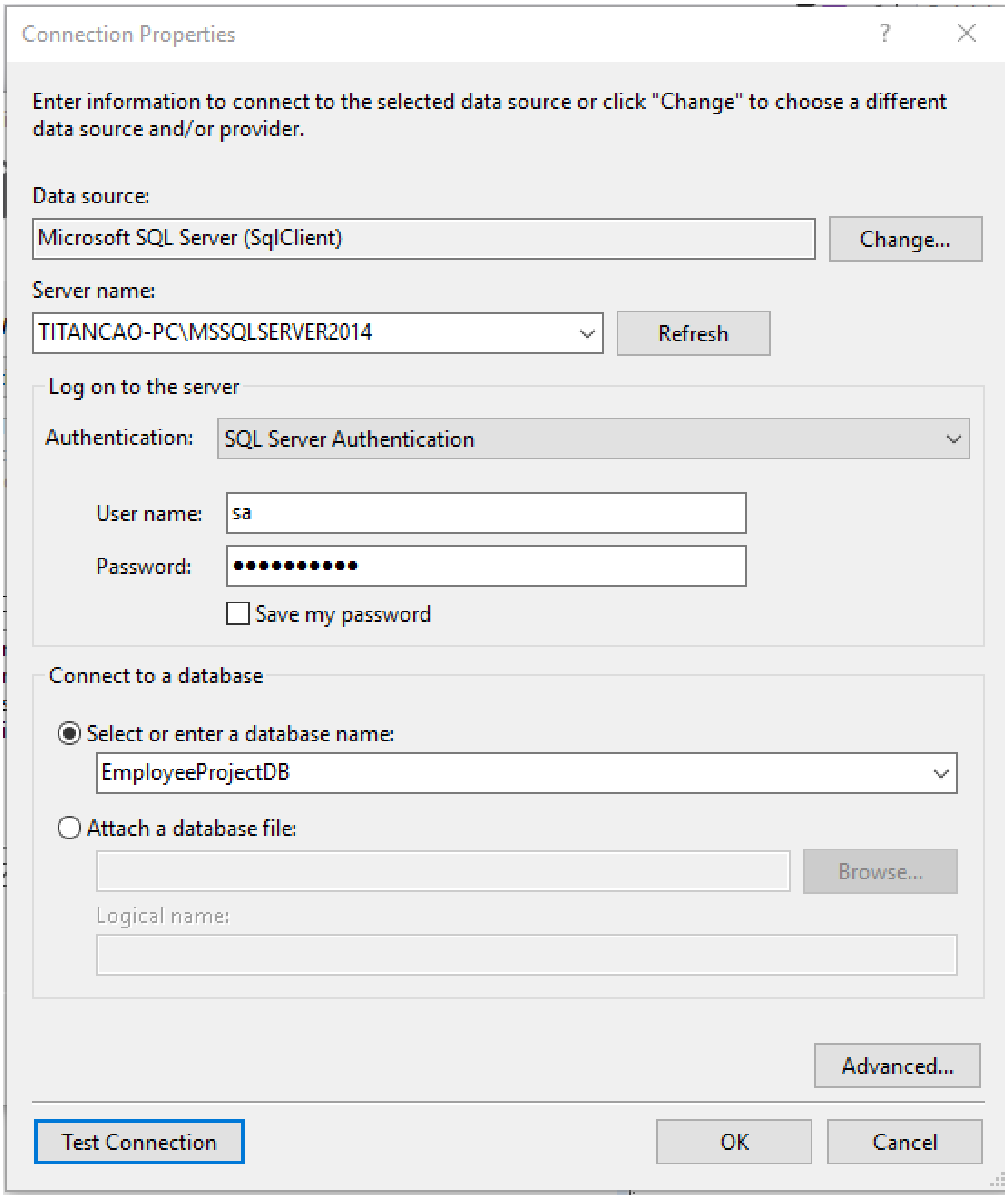
Click here to go online and find templates.

Name: EmployeeProjectModel

Select EF Designer from database (first option)



Creates a model in the EF Designer based on an existing database. You can choose the database connection, settings for the model, and database objects to include in the model. The classes your application will interact with are generated from the model.



Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft SQL Server (SqlClient)

Server name: TITANCAO-PC\MSSQLSERVER2014

Log on to the server

Authentication: SQL Server Authentication

User name: sa

Password: ●●●●●●●●●●●

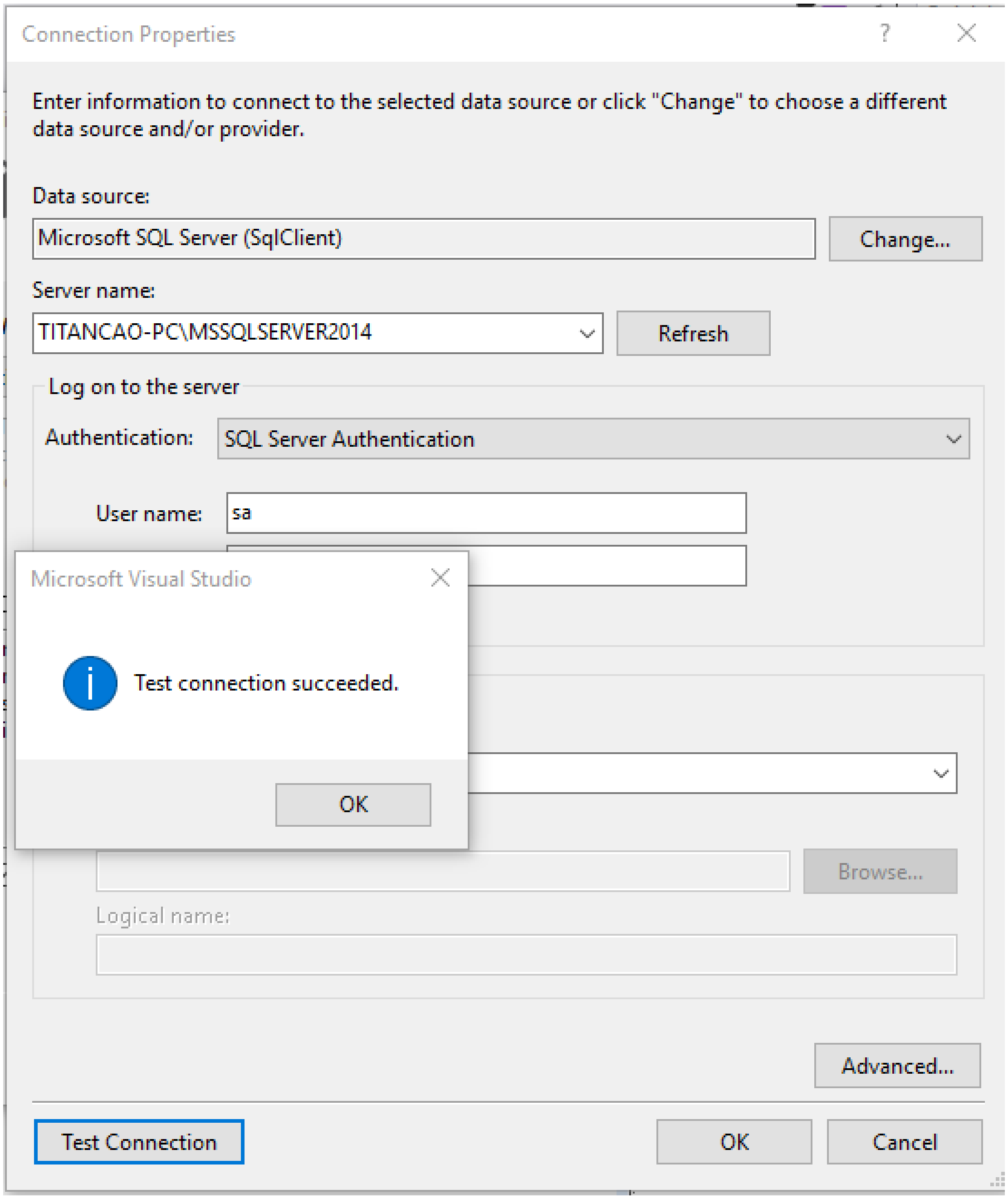
□ Save my password

Connect to a database

● Select or enter a database name: EmployeeProjectDB

○ Attach a database file:

Logical name:



Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft SQL Server (SqlClient)

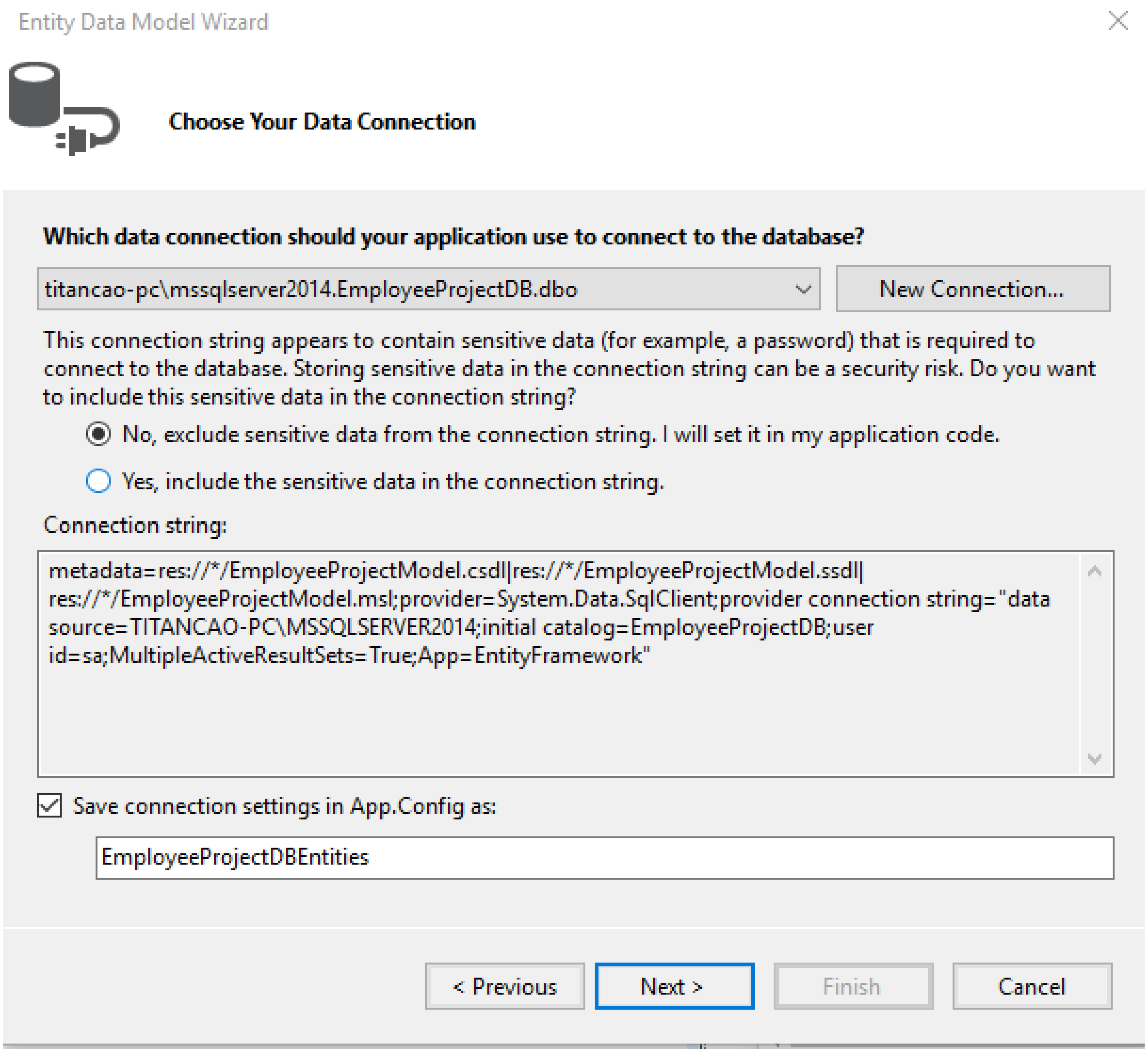
Server name: TITANCAO-PC\MSSQLSERVER2014

Authentication: SQL Server Authentication

User name: sa

Test connection succeeded.

Note: ***Use your Server Name***



Choose Your Data Connection

Which data connection should your application use to connect to the database?

titancao-pc\mssqlserver2014.EmployeeProjectDB.dbo

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

● No, exclude sensitive data from the connection string. I will set it in my application code.

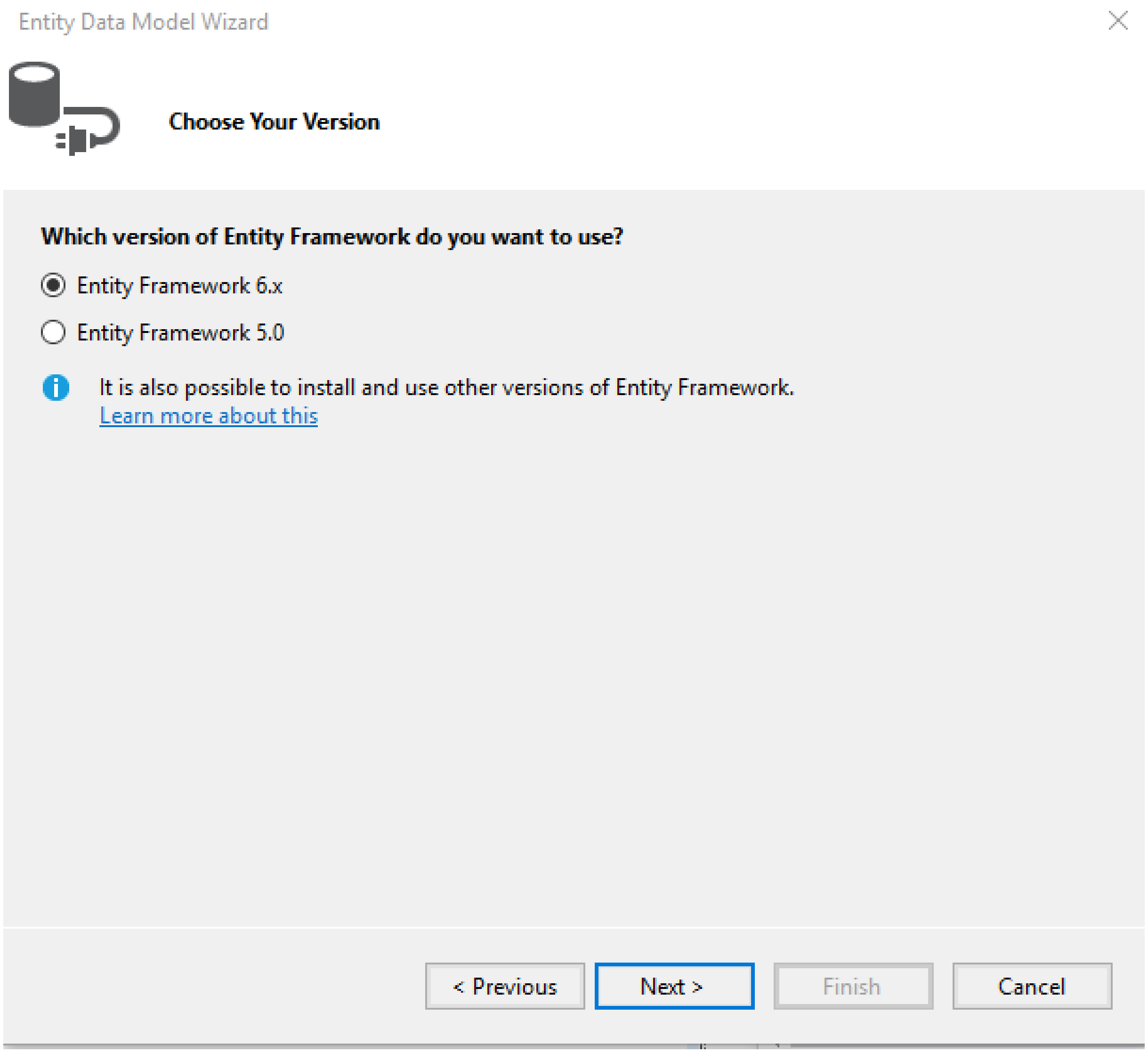
○ Yes, include the sensitive data in the connection string.

Connection string:

```
metadata=res://*/EmployeeProjectModel.csdl|res://*/EmployeeProjectModel.ssdl|
res://*/EmployeeProjectModel.msl;provider=System.Data.SqlClient;provider connection string="data
source=TITANCAO-PC\MSSQLSERVER2014;initial catalog=EmployeeProjectDB;user
id=sa;MultipleActiveResultSets=True;App=EntityFramework"
```

☑ Save connection settings in App.Config as:

EmployeeProjectDBEntities



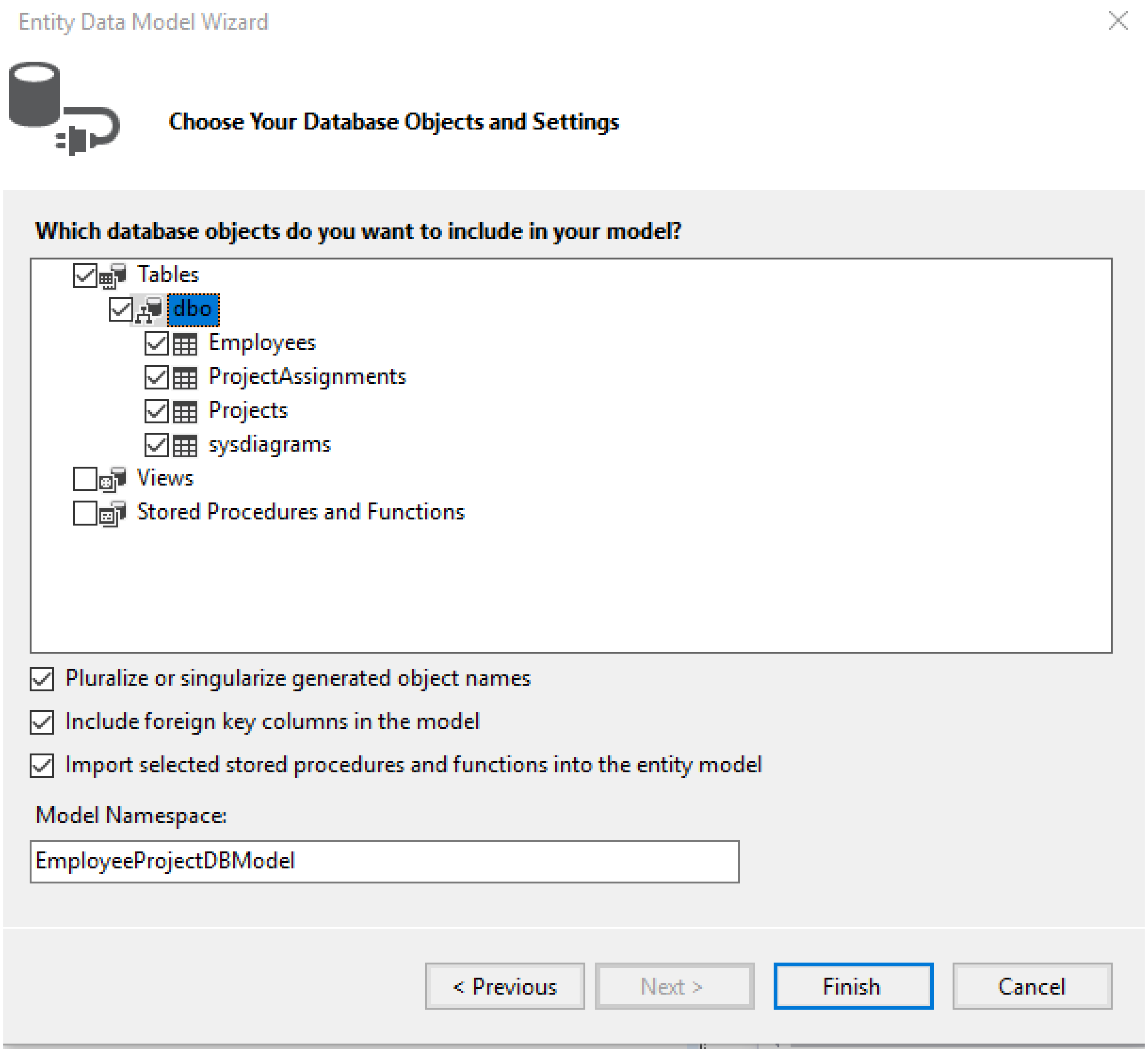
Choose Your Version

Which version of Entity Framework do you want to use?

● Entity Framework 6.x

○ Entity Framework 5.0

It is also possible to install and use other versions of Entity Framework.
Learn more about this



Entity Data Model Wizard

Choose Your Database Objects and Settings

Which database objects do you want to include in your model?

- ☑ Tables
  - ☑ dbo
    - ☑ Employees
    - ☑ ProjectAssignments
    - ☑ Projects
    - ☑ sysdiagrams
- □ Views
- □ Stored Procedures and Functions

☑ Pluralize or singularize generated object names

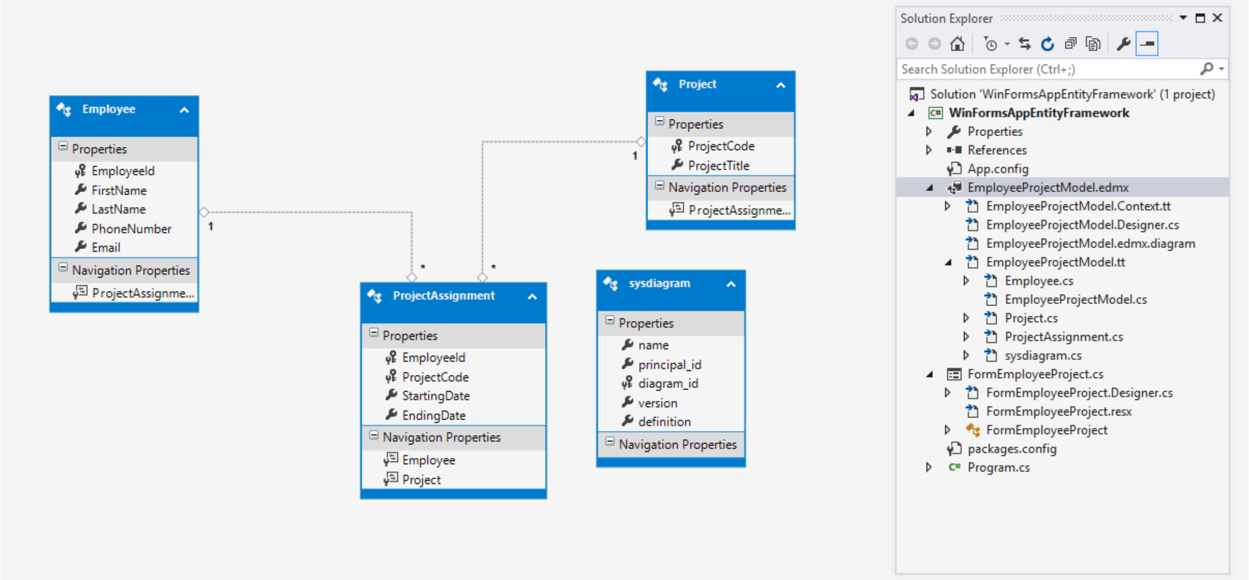
☑ Include foreign key columns in the model

☑ Import selected stored procedures and functions into the entity model

Model Namespace:

EmployeeProjectDBModel

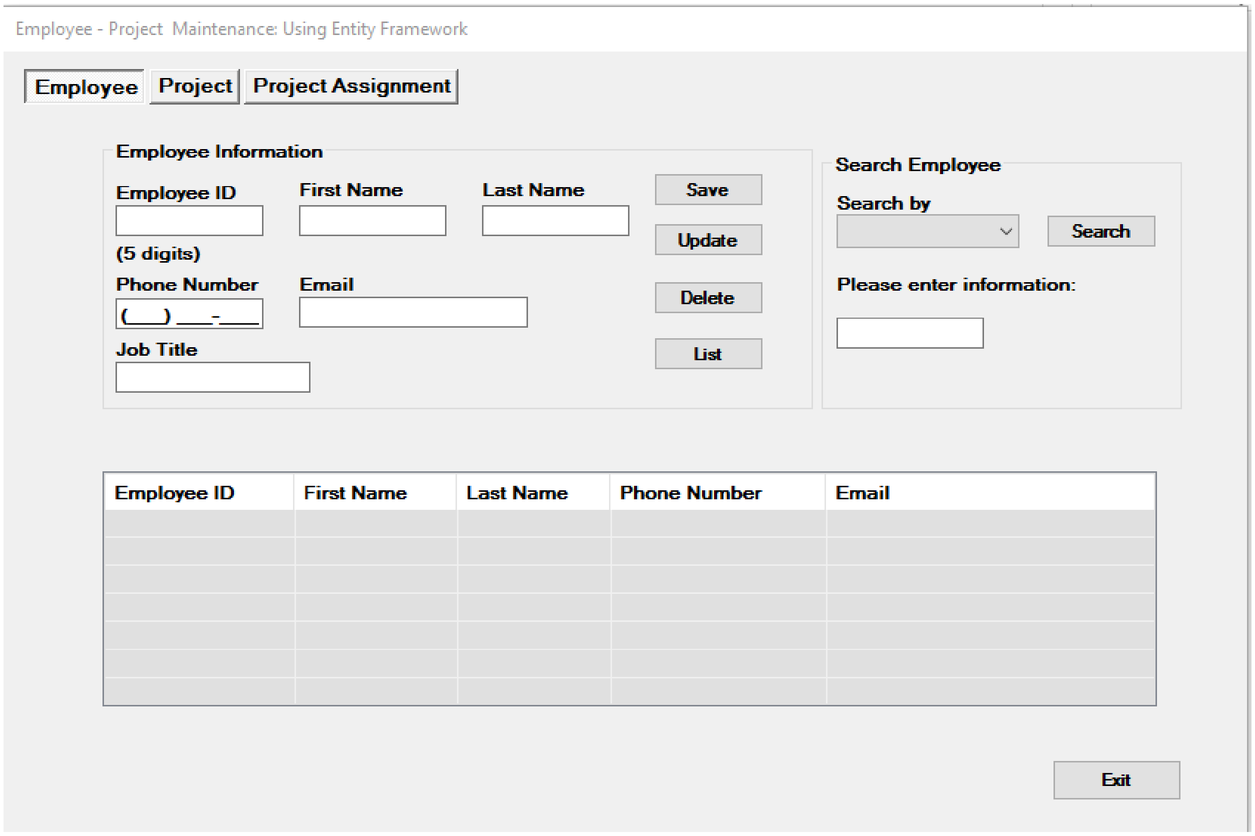
Looking at the File **EmployeeProjectModel.edmx**



```
//------------------------------------------------------------------------------
// <auto-generated>
//     This code was generated from a template.
//
//     Manual changes to this file may cause unexpected behavior in your application.
//     Manual changes to this file will be overwritten if the code is regenerated.
// </auto-generated>
//------------------------------------------------------------------------------

namespace WinFormsAppEntityFramework
{
    using System;
    using System.Collections.Generic;

    public partial class Employee
    {
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",
"CA2214:DoNotCallOverridableMethodsInConstructors")]
        public Employee()
        {
            this.ProjectAssignments = new HashSet<ProjectAssignment>();
        }

        public int EmployeeId { get; set; }
        public string FirstName { get; set; }
        public string LastName { get; set; }
        public string PhoneNumber { get; set; }
        public string Email { get; set; }

        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",
"CA2227:CollectionPropertiesShouldBeReadOnly")]
        public virtual ICollection<ProjectAssignment> ProjectAssignments { get; set; }
    }
}
```

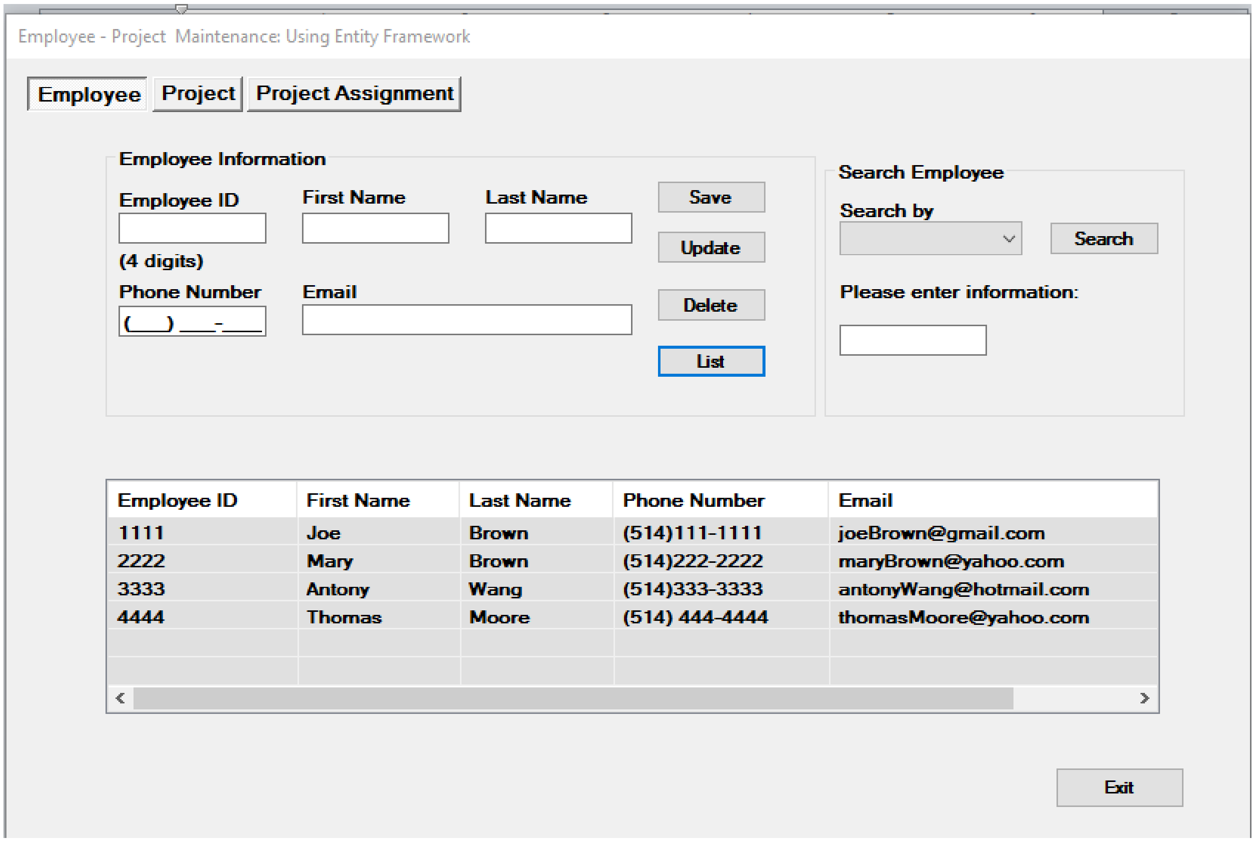
```
//------------------------------------------------------------------------------
// <auto-generated>
//     This code was generated from a template.
//
//     Manual changes to this file may cause unexpected behavior in your application.
//     Manual changes to this file will be overwritten if the code is regenerated.
// </auto-generated>
//------------------------------------------------------------------------------

namespace WinFormsAppEntityFramework
{
    using System;
    using System.Collections.Generic;

    public partial class Project
    {
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",
"CA2214:DoNotCallOverridableMethodsInConstructors")]
        public Project()
        {
            this.ProjectAssignments = new HashSet<ProjectAssignment>();
        }

        public string ProjectCode { get; set; }
        public string ProjectTitle { get; set; }

        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",
"CA2227:CollectionPropertiesShouldBeReadOnly")]
        public virtual ICollection<ProjectAssignment> ProjectAssignments { get; set; }
    }
}
//------------------------------------------------------------------------------
// <auto-generated>
//     This code was generated from a template.
//
//     Manual changes to this file may cause unexpected behavior in your application.
//     Manual changes to this file will be overwritten if the code is regenerated.
// </auto-generated>
//------------------------------------------------------------------------------

namespace WinFormsAppEntityFramework
{
    using System;
    using System.Collections.Generic;

    public partial class ProjectAssignment
    {
        public int EmployeeId { get; set; }
        public string ProjectCode { get; set; }
        public System.DateTime StartingDate { get; set; }
        public System.DateTime EndingDate { get; set; }

        public virtual Employee Employee { get; set; }
        public virtual Project Project { get; set; }
    }
}
```

Performing CRUD Operations



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WinFormsAppEntityFramework
{
    public partial class FormEmployeeProject : Form
    {
        public FormEmployeeProject()
        {
            InitializeComponent();
        }

        private void buttonExit_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
```

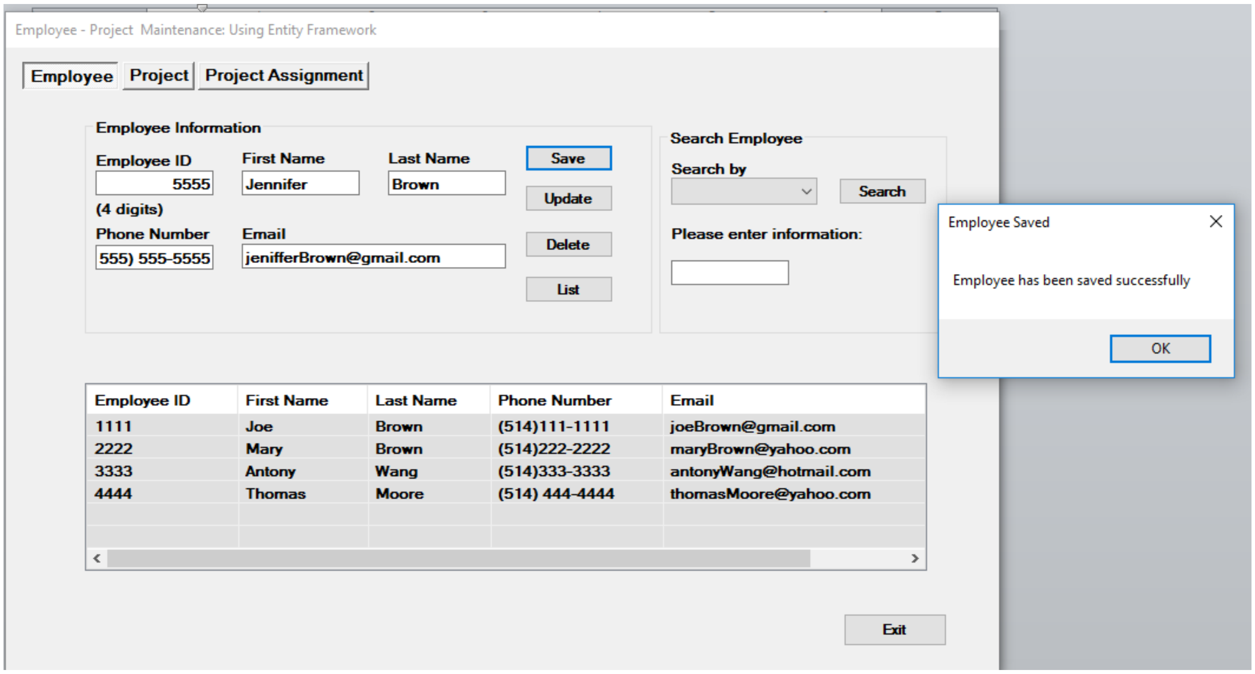
List Employees



Employee ID	First Name	Last Name	Phone Number	Email
1111	Joe	Brown	(514)111-1111	joeBrown@gmail.com
2222	Mary	Brown	(514)222-2222	maryBrown@yahoo.com
3333	Antony	Wang	(514)333-3333	antonyWang@hotmail.com
4444	Thomas	Moore	(514) 444-4444	thomasMoore@yahoo.com

```
private void buttonList_Click(object sender, EventArgs e)
{



}
```

Save Employee



Employee ID: 5555 (4 digits)

First Name: Jennifer

Last Name: Brown

Phone Number: 555) 555-5555

Email: jenifferBrown@gmail.com

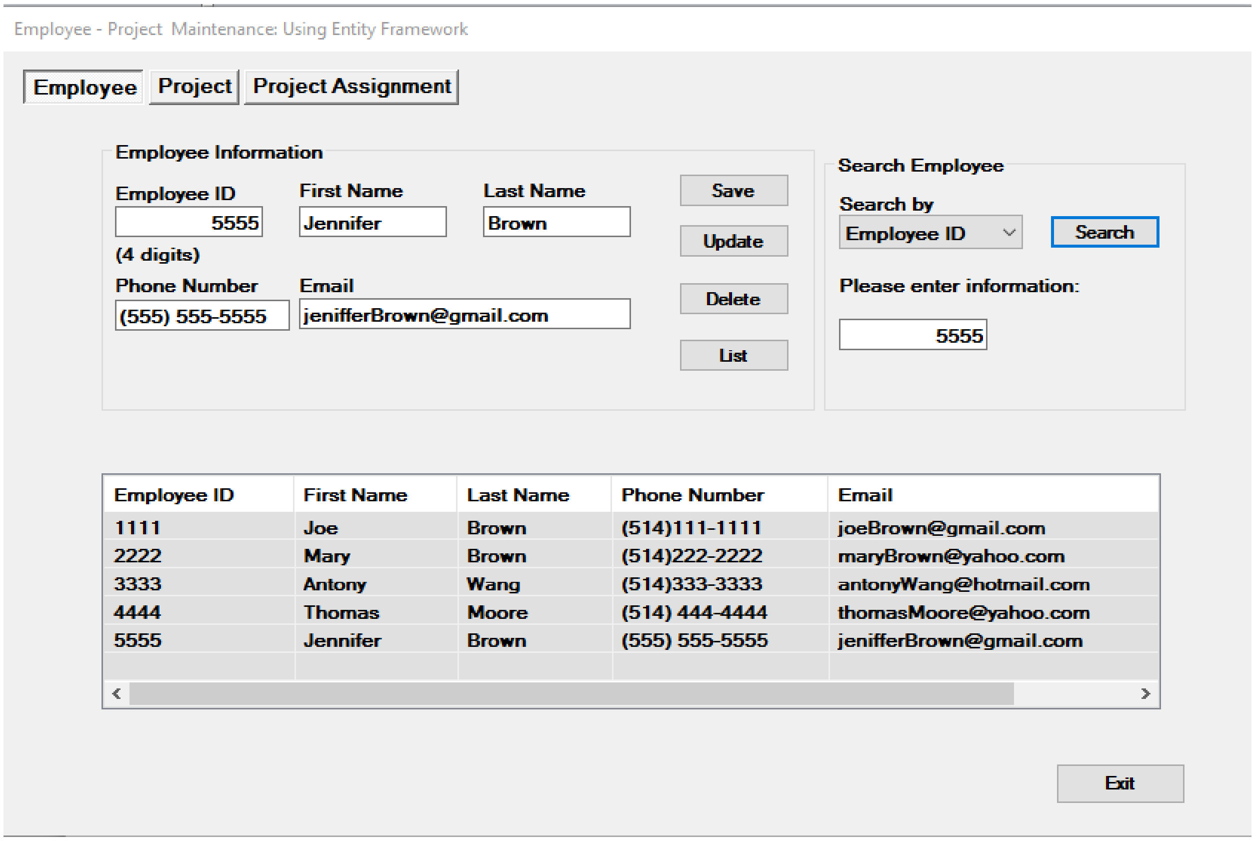
Employee ID	First Name	Last Name	Phone Number	Email
1111	Joe	Brown	(514)111-1111	joeBrown@gmail.com
2222	Mary	Brown	(514)222-2222	maryBrown@yahoo.com
3333	Antony	Wang	(514)333-3333	antonyWang@hotmail.com
4444	Thomas	Moore	(514) 444-4444	thomasMoore@yahoo.com

Employee Saved

Employee has been saved successfully

```
private void buttonSaveEmployee_Click(object sender, EventArgs e)
 {



 }
```

Search Employee



Employee ID: 5555

First Name: Jennifer

Last Name: Brown

Phone Number: (555) 555-5555

Email: jeniferBrown@gmail.com

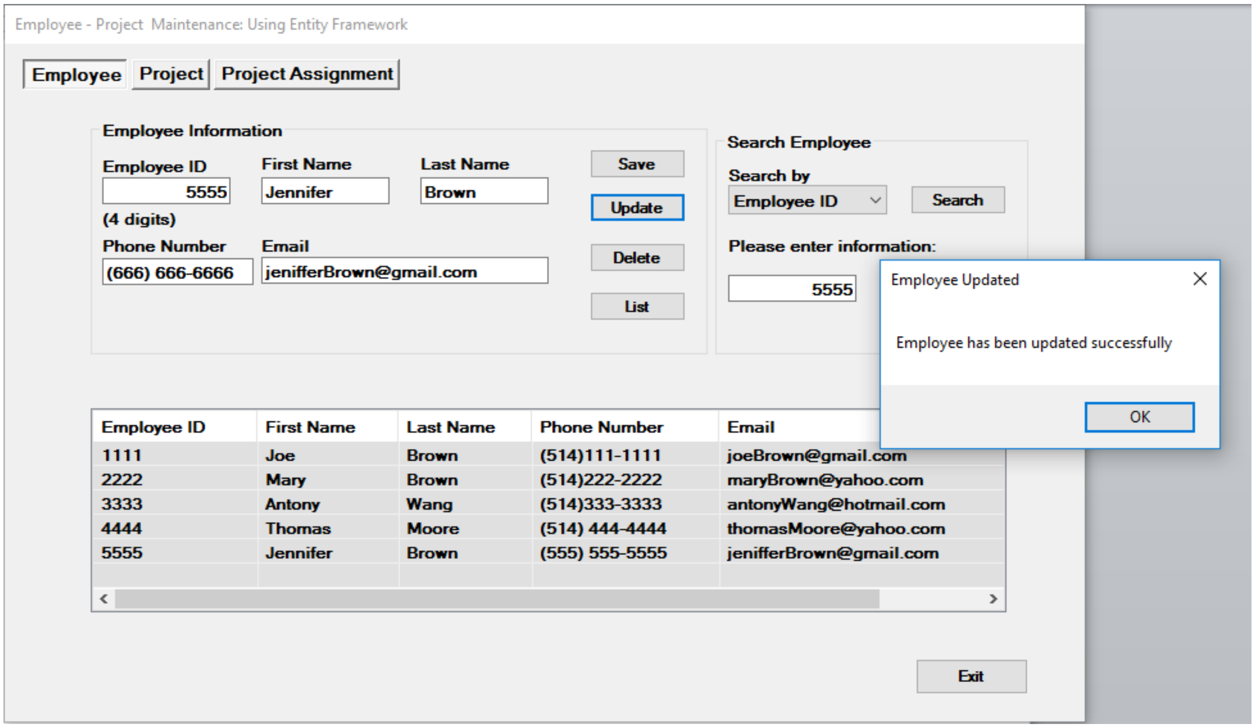
Search by: Employee ID

Please enter information: 5555

Employee ID	First Name	Last Name	Phone Number	Email
1111	Joe	Brown	(514)111-1111	joeBrown@gmail.com
2222	Mary	Brown	(514)222-2222	maryBrown@yahoo.com
3333	Antony	Wang	(514)333-3333	antonyWang@hotmail.com
4444	Thomas	Moore	(514) 444-4444	thomasMoore@yahoo.com
5555	Jennifer	Brown	(555) 555-5555	jeniferBrown@gmail.com

```
private void buttonSearchEmployee_Click(object sender, EventArgs e)
        {






        }
```

Update Employee



Employee Information

Employee ID: 5555 (4 digits)

First Name: Jennifer

Last Name: Brown

Phone Number: (666) 666-6666

Email: jenifferBrown@gmail.com

Search Employee

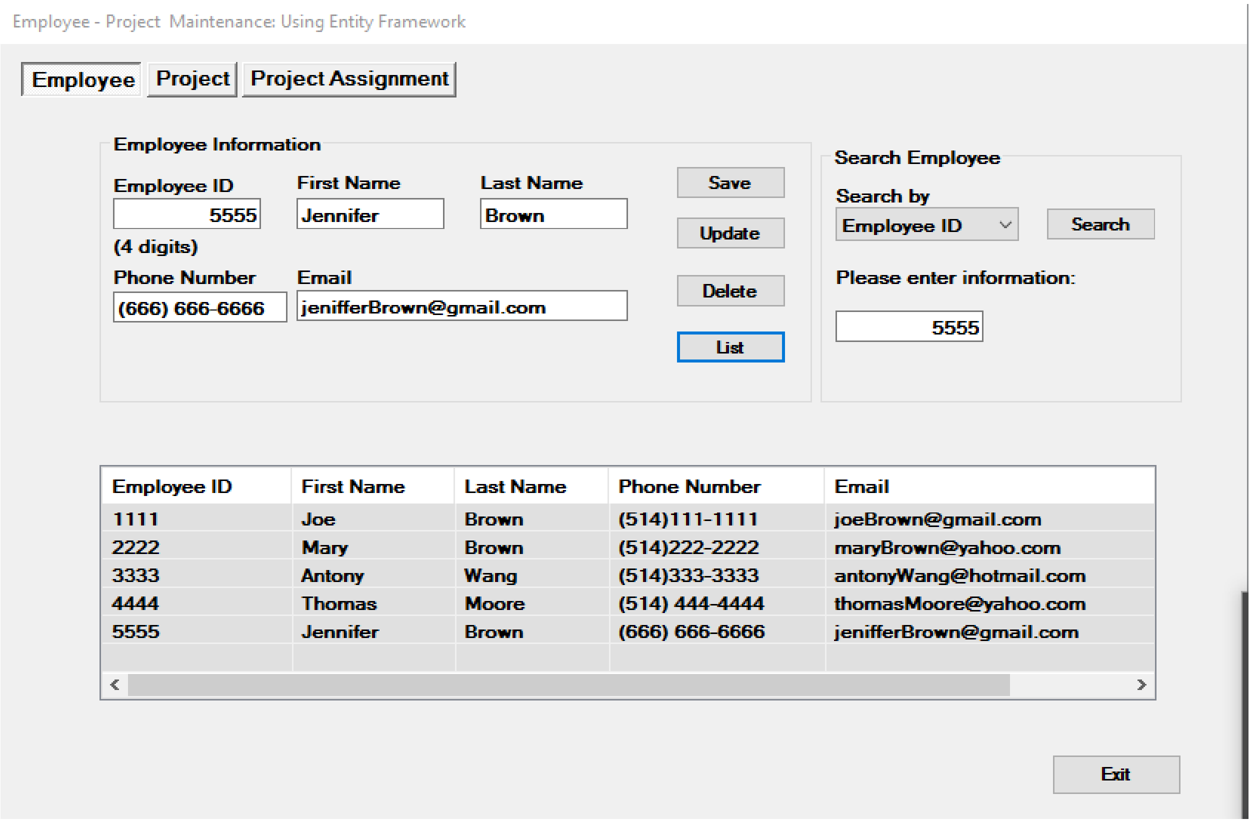
Search by: Employee ID

Please enter information: 5555

Employee Updated

Employee has been updated successfully

Employee ID	First Name	Last Name	Phone Number	Email
1111	Joe	Brown	(514)111-1111	joeBrown@gmail.com
2222	Mary	Brown	(514)222-2222	maryBrown@yahoo.com
3333	Antony	Wang	(514)333-3333	antonyWang@hotmail.com
4444	Thomas	Moore	(514) 444-4444	thomasMoore@yahoo.com
5555	Jennifer	Brown	(555) 555-5555	jenifferBrown@gmail.com



Employee Information

Employee ID: 5555 (4 digits)

First Name: Jennifer

Last Name: Brown

Phone Number: (666) 666-6666

Email: jenifferBrown@gmail.com

Search Employee

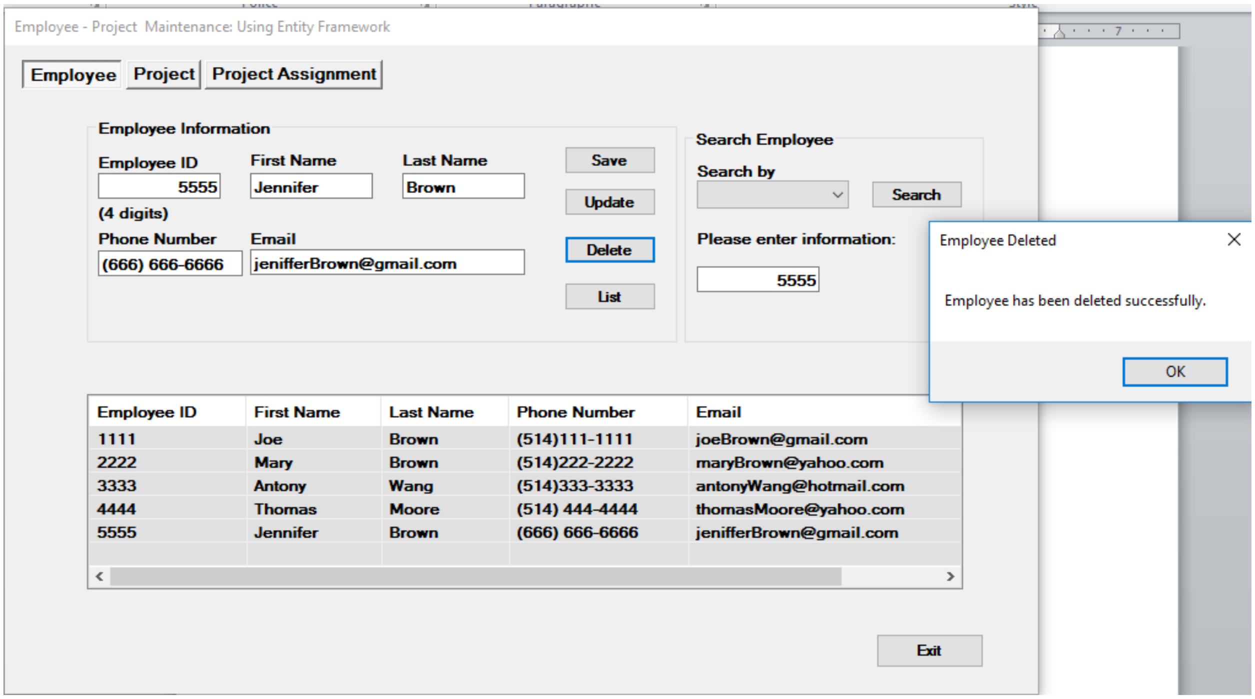
Search by: Employee ID

Please enter information: 5555

Employee ID	First Name	Last Name	Phone Number	Email
1111	Joe	Brown	(514)111-1111	joeBrown@gmail.com
2222	Mary	Brown	(514)222-2222	maryBrown@yahoo.com
3333	Antony	Wang	(514)333-3333	antonyWang@hotmail.com
4444	Thomas	Moore	(514) 444-4444	thomasMoore@yahoo.com
5555	Jennifer	Brown	(666) 666-6666	jenifferBrown@gmail.com

```
private void buttonUpdateEmployee_Click(object sender, EventArgs e)
        {









        }
```

Delete Employee



Employee Information

Employee ID	First Name	Last Name
5555	Jennifer	Brown
(4 digits)		
Phone Number	Email	
(666) 666-6666	jeniferBrown@gmail.com	

Search Employee

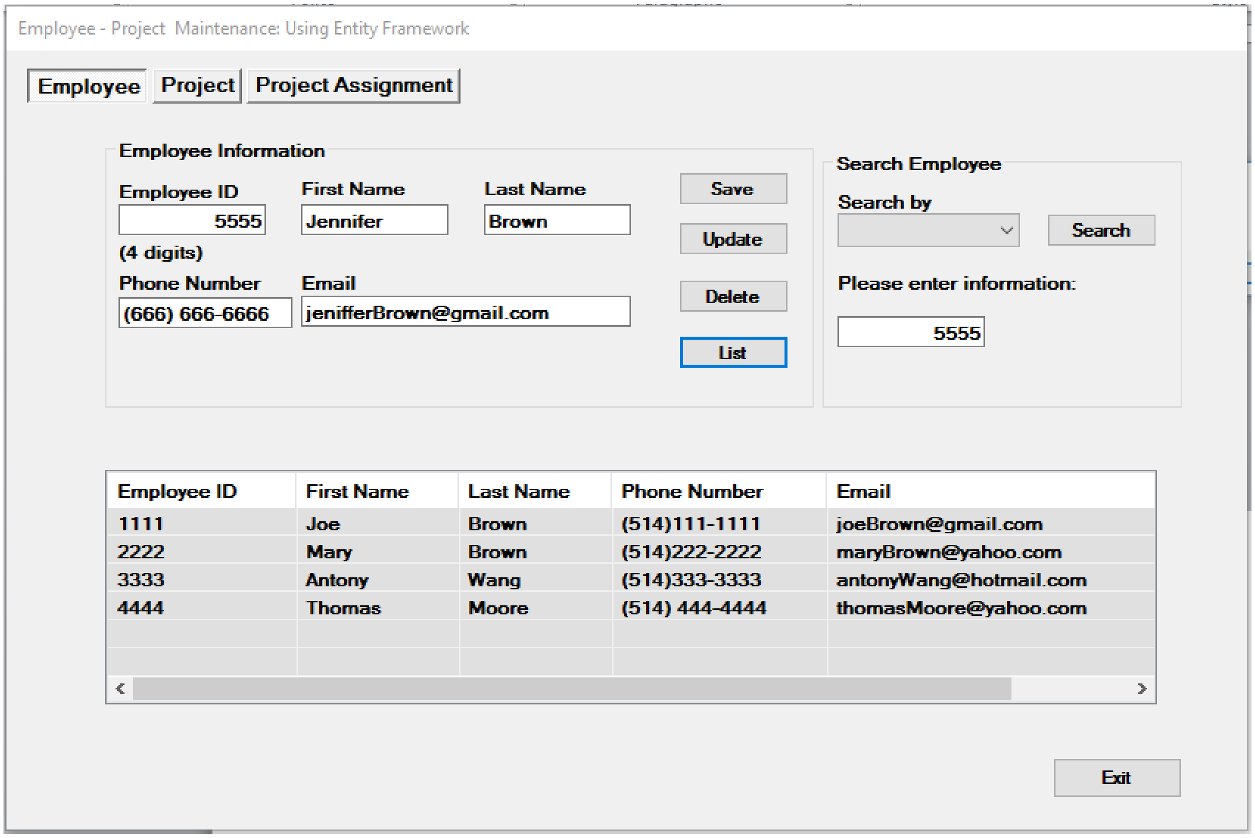
Search by

Please enter information: 5555

Employee Deleted

Employee has been deleted successfully.

Employee ID	First Name	Last Name	Phone Number	Email
1111	Joe	Brown	(514)111-1111	joeBrown@gmail.com
2222	Mary	Brown	(514)222-2222	maryBrown@yahoo.com
3333	Antony	Wang	(514)333-3333	antonyWang@hotmail.com
4444	Thomas	Moore	(514) 444-4444	thomasMoore@yahoo.com
5555	Jennifer	Brown	(666) 666-6666	jeniferBrown@gmail.com



Employee Information

Employee ID	First Name	Last Name
5555	Jennifer	Brown
(4 digits)		
Phone Number	Email	
(666) 666-6666	jenifferBrown@gmail.com	

Search Employee

Search by

Please enter information: 5555

Employee ID	First Name	Last Name	Phone Number	Email
1111	Joe	Brown	(514)111-1111	joeBrown@gmail.com
2222	Mary	Brown	(514)222-2222	maryBrown@yahoo.com
3333	Antony	Wang	(514)333-3333	antonyWang@hotmail.com
4444	Thomas	Moore	(514) 444-4444	thomasMoore@yahoo.com

```
private void buttonDelete_Click(object sender, EventArgs e)
        {



        }
    }
}
```

Important Notes

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