Entity Framework

Ing. Gonzalo Alba
MCSD, MCT

gonzalo.alba@jalasoft.com

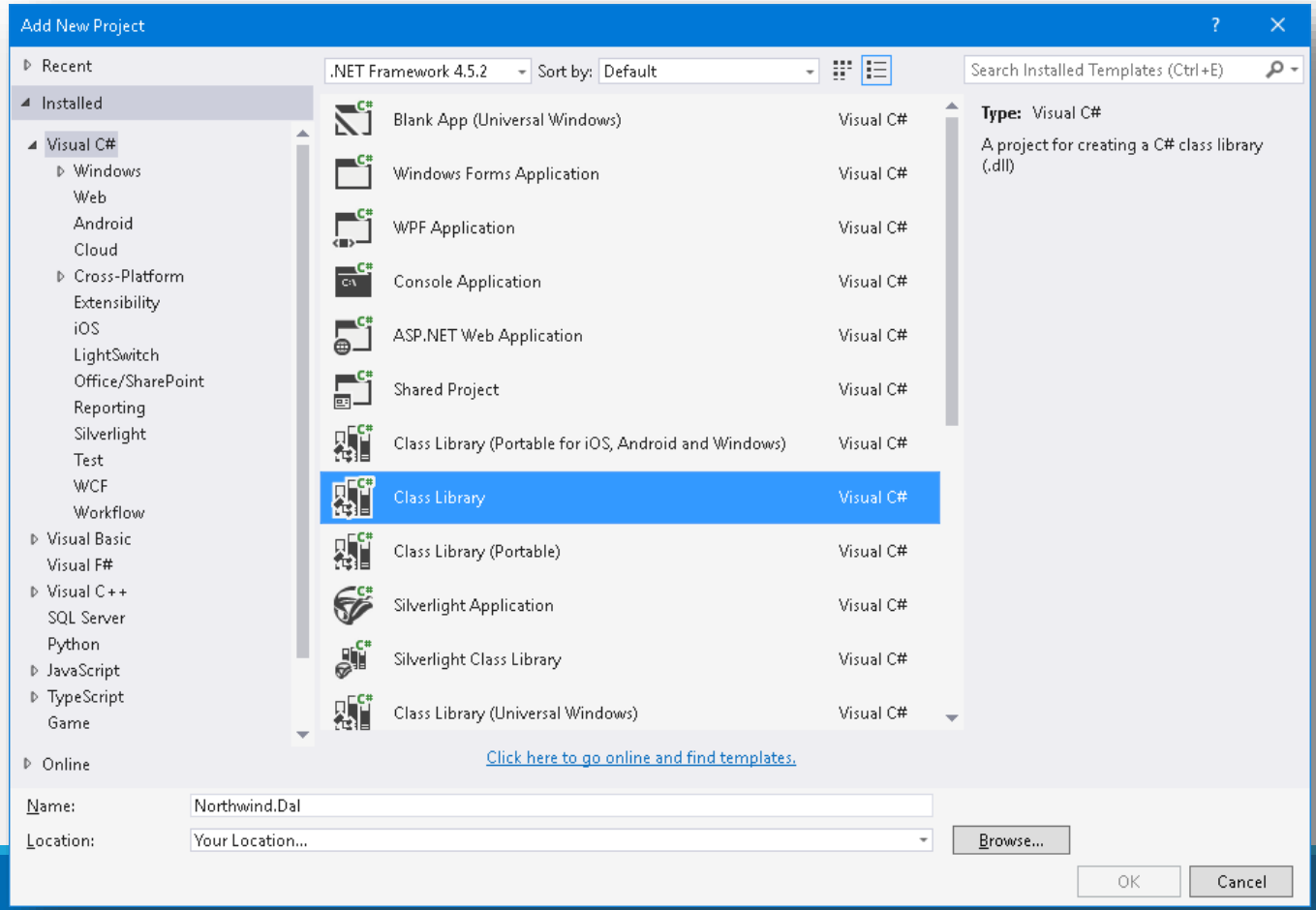
What is the ADO.NET Entity Framework?

- ADO.NET is an abbreviation for **ActiveX Data Object.NET**, which is part of Microsoft's larger .NET framework.

- ADO.NET Entity Framework is an **Object/Relational Mapping** (ORM) framework that enables developers\testers to work with relational data as domain-specific objects, eliminating the need for most of the data access plumbing code (DataSets, DataReaders, etc.).

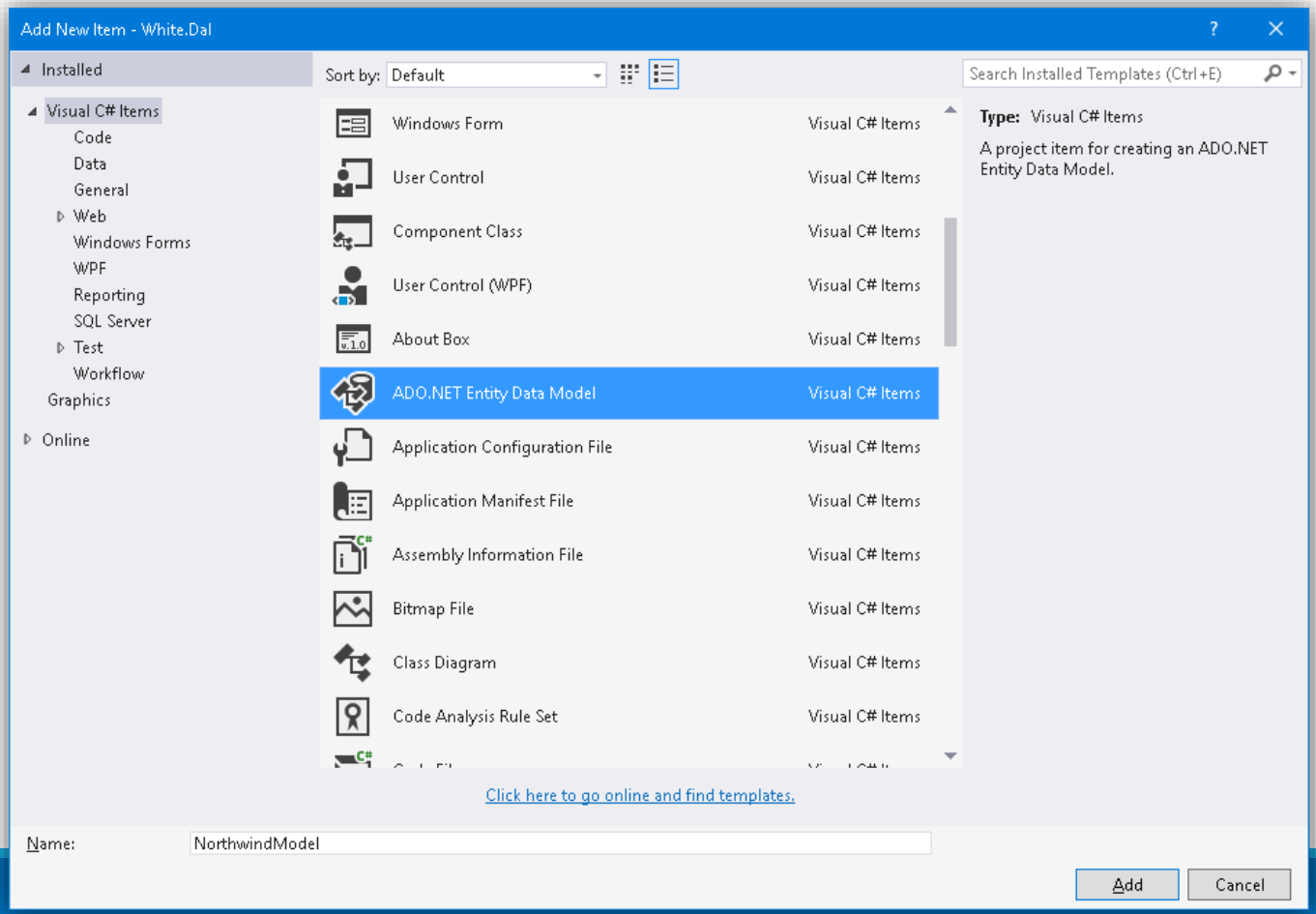
Create a Class Library project

Step 1:



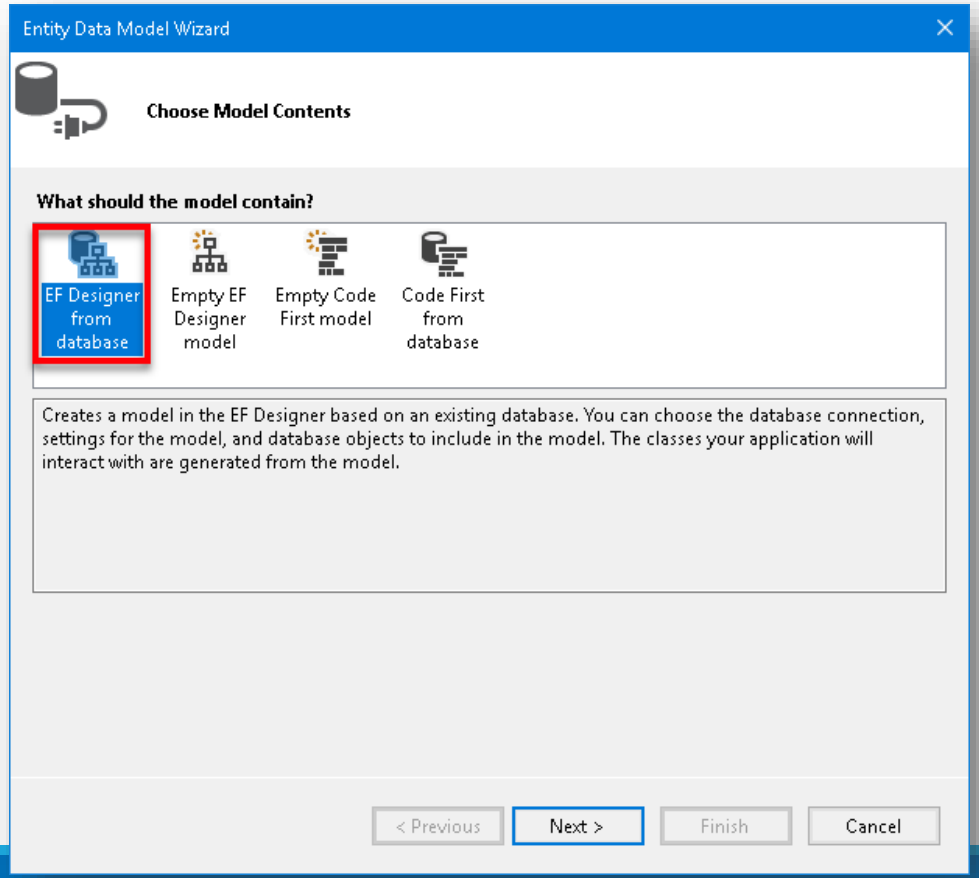
Creating ADO.NET Entity Data Model

Step 2:



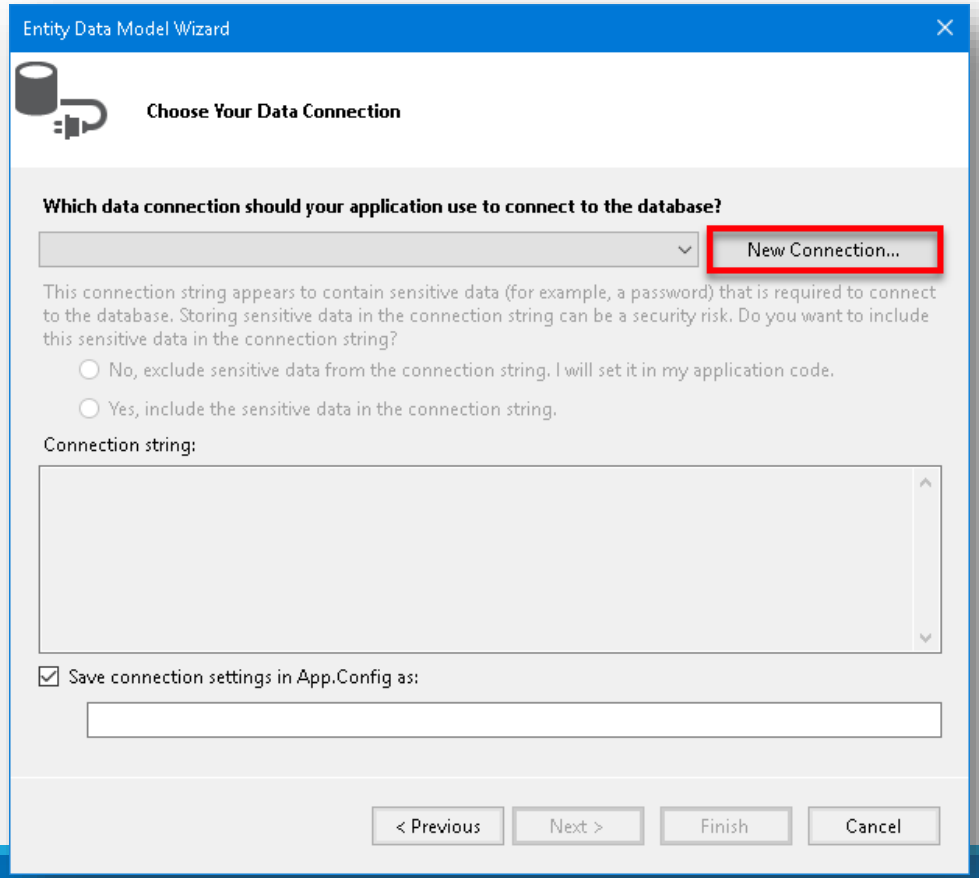
Creating ADO.NET Entity Data Model

Step 3:



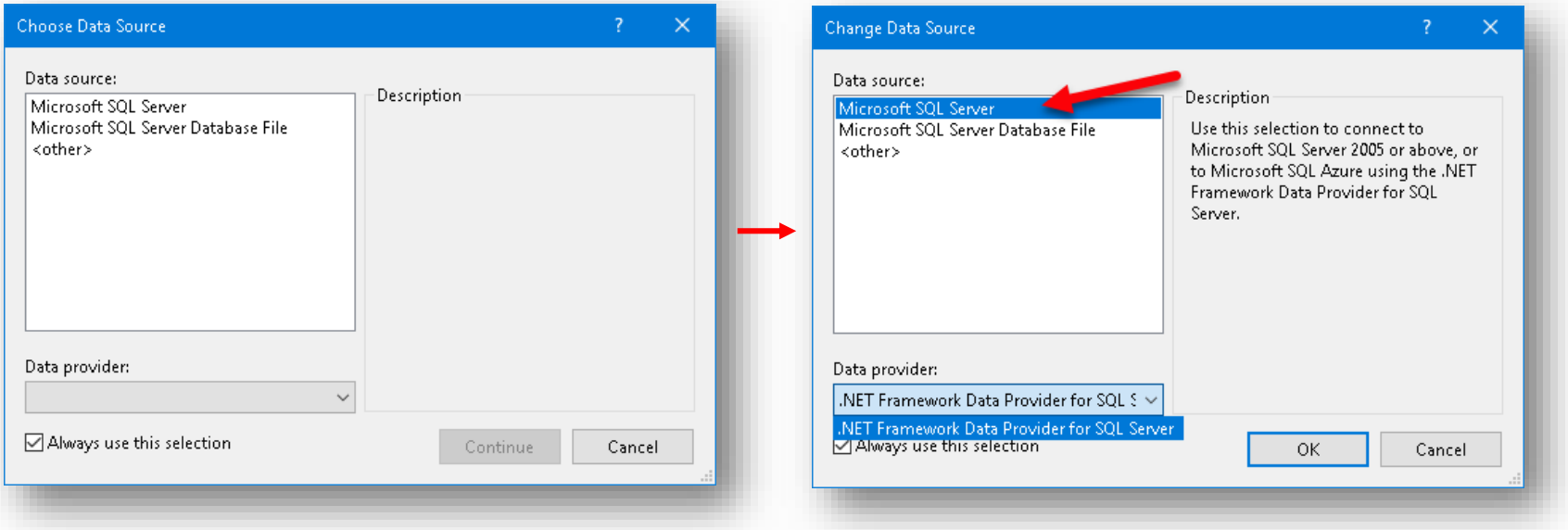
Creating ADO.NET Entity Data Model

Step 4:



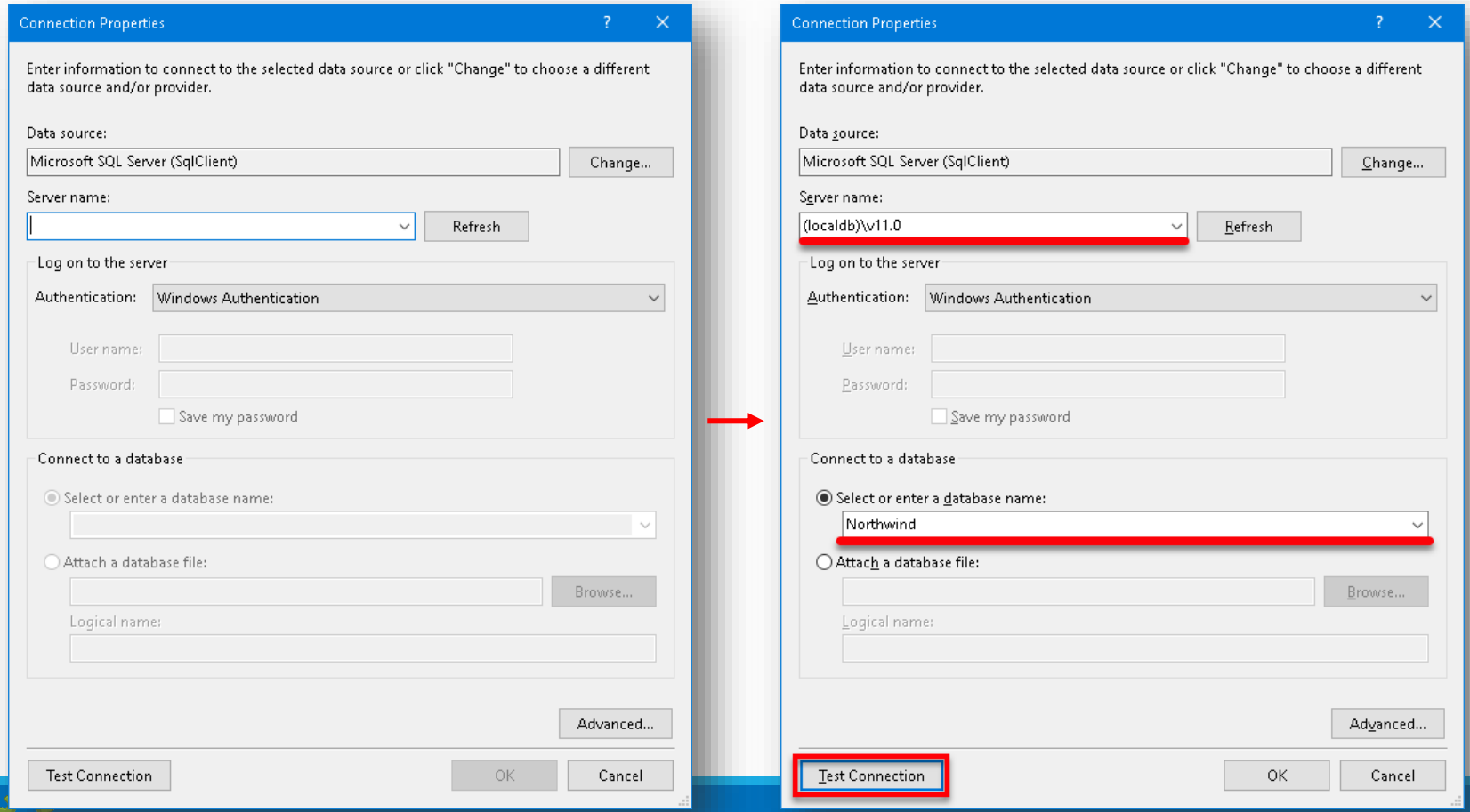
Creating ADO.NET Entity Data Model

Step 5:



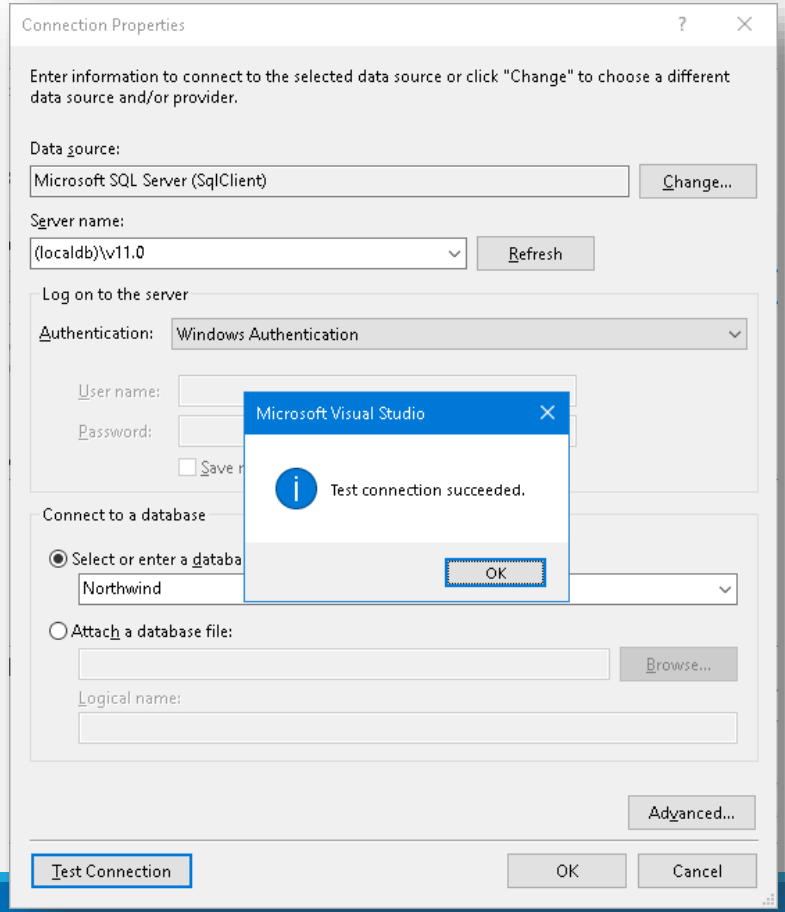
Creating ADO.NET Entity Data Model

Step 6:



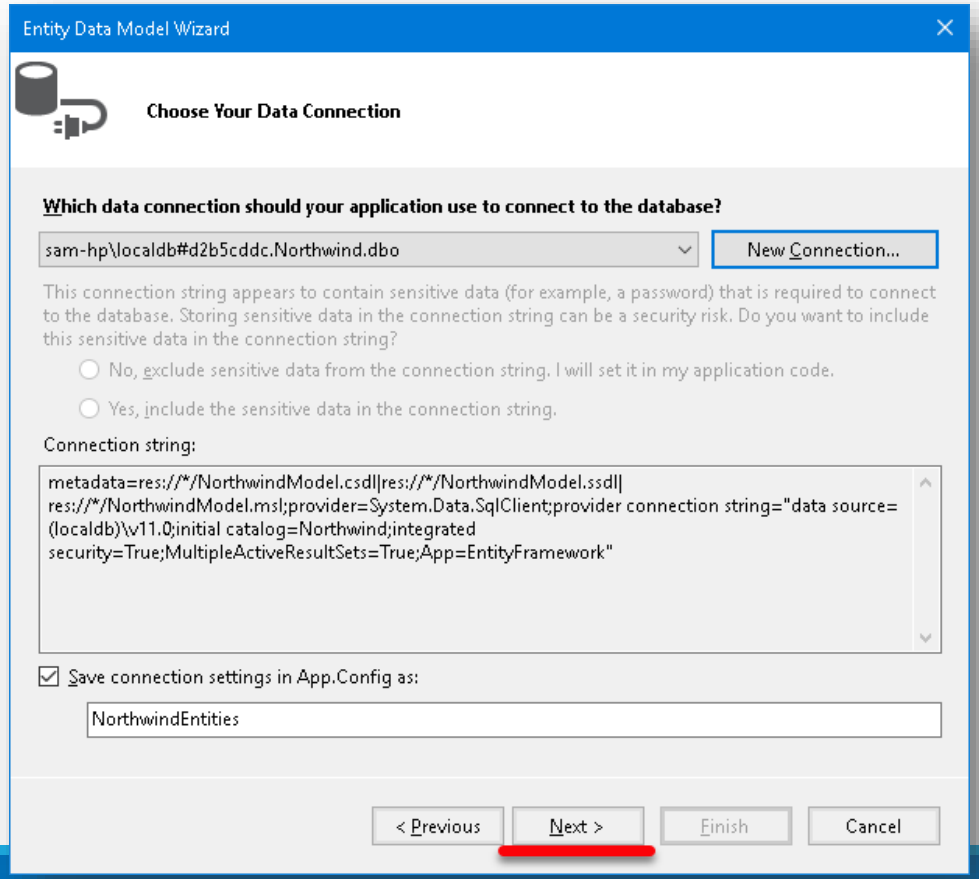
Creating ADO.NET Entity Data Model

Step 7:



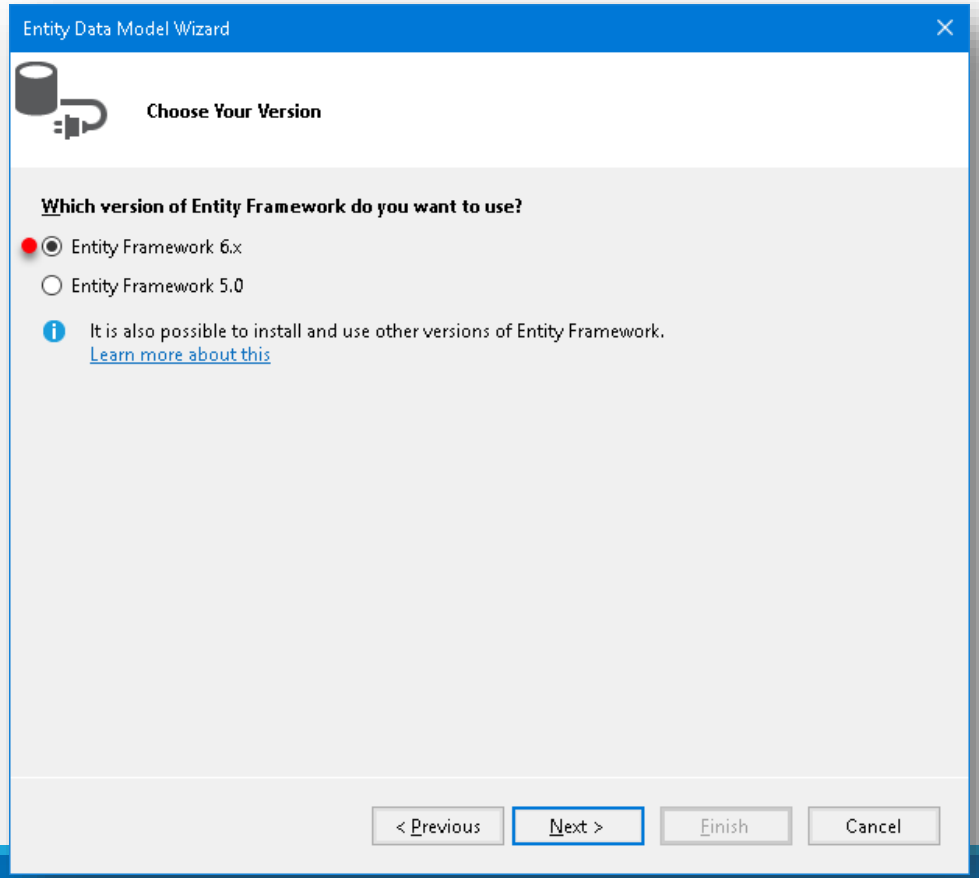
Creating ADO.NET Entity Data Model

Step 8:



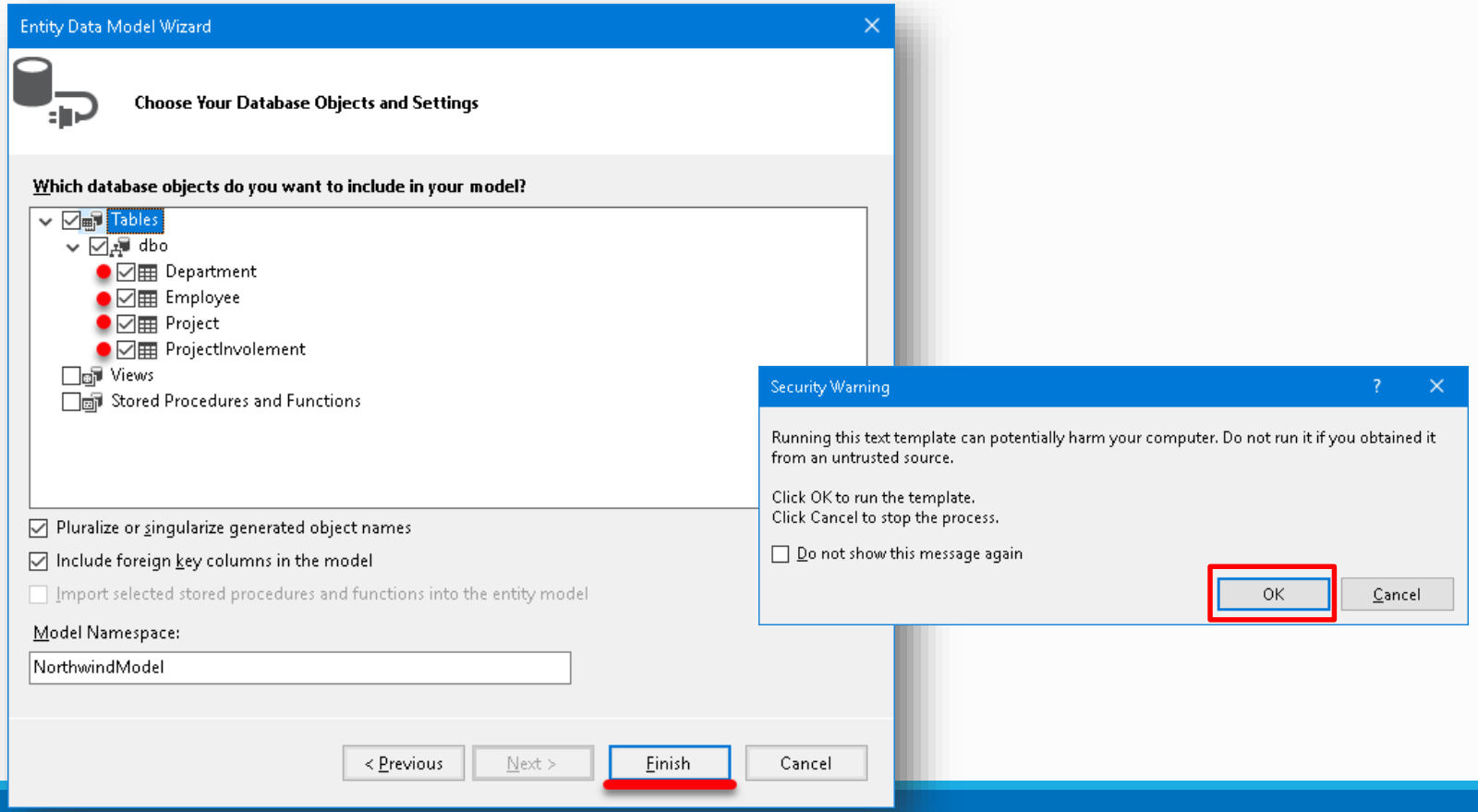
Creating ADO.NET Entity Data Model

Step 9:



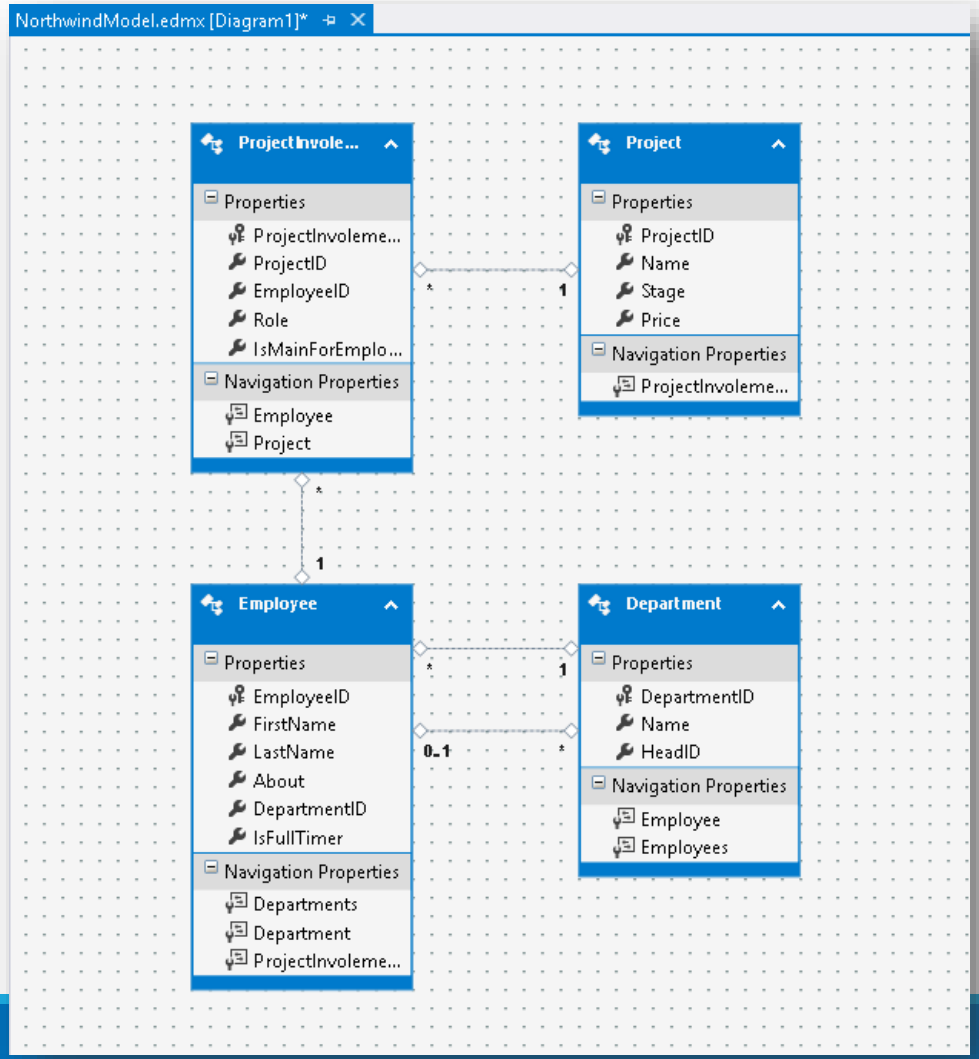
Creating ADO.NET Entity Data Model

Step 10:



Model

Step 11:



EF

Demo

Insert data with EF

```
public bool CreateDepartment(string name)
{
   Department depto = GetDepartment(name);

   if (depto == null)
   {
      using (var _dbContext = new NorthwindEntities())
      {
         depto = new Department();
         depto.Name = name;

         _dbContext.Departments.Add(depto);

         if (_dbContext.SaveChanges() > 0)
         {
            return true;
         }

          return false;
      }
   }

   return false;
}
```

Get data with EF

```
public Department GetDepartment(string name)
{
   using (var _dbContext = new NorthwindEntities())
   {
      return _dbContext.Departments.FirstOrDefault(d =>
                                    d.Name.Equals(name));
   }
}
```

```
public List<Department> GetDepartment2(string name)
{
   List<Department> departmentResult;
   using (var _dbContext = new NorthwindEntities())
   {
      var query = from depto in _dbContext.Departments
                  where depto.Name.Equals(name)
                  select depto;

      departmentResult = query.ToList();
   }

   if (departmentResult.Count() > 0)
   {
      return departmentResult;
   }

   return null;
}
```

Update data with EF

```
public bool UpdateDepartment(Department department, string newName)
{
   using (var _dbContext = new NorthwindEntities())
   {
      var departmentUpdated = _dbContext.Departments.Attach(department);

      departmentUpdated.Name = newName;

      if (_dbContext.SaveChanges() > 0)
      {
         return true;
      }

      return false;
   }
}
```

Delete data with EF

```
public bool DeleteDepartment(string name)
{
   Department depto = GetDepartment(name);

   if (depto != null)
   {
      using (var _dbContext = new NorthwindEntities())
      {
         _dbContext.Departments.Attach(depto);
         _dbContext.Departments.Remove(depto);

         if (_dbContext.SaveChanges() > 0)
         {
            return true;
         }
      }
   }

   return false;
}
```

Tests

```
[TestFixture]
public class DepartmentTests
{
    [Test]
    public void CreateDepartmentDb()
    {
        Assert.IsTrue(Entities.DepartmentEntity.CreateDepartment("New Department"));

        Assert.IsTrue(Entities.DepartmentEntity.DeleteDepartment("New Department"));
    }
}
```

Q&A