

Copyright © 2015 by The Learning House.

All rights reserved. No part of these materials may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of The Learning House. For permission requests, write to The Learning House, addressed “Attention: Permissions Coordinator,” at the address below.

The Learning House
427 S. 4th Street #300
Louisville KY 40202

Entity Framework Model First

.NET Cohort

Coding Bootcamp

Lesson Goals

- Learn how to use the EDMX designer to create new tables and generate creation scripts

Situation

You have a brand new project and you want to create a model. No database exists yet.

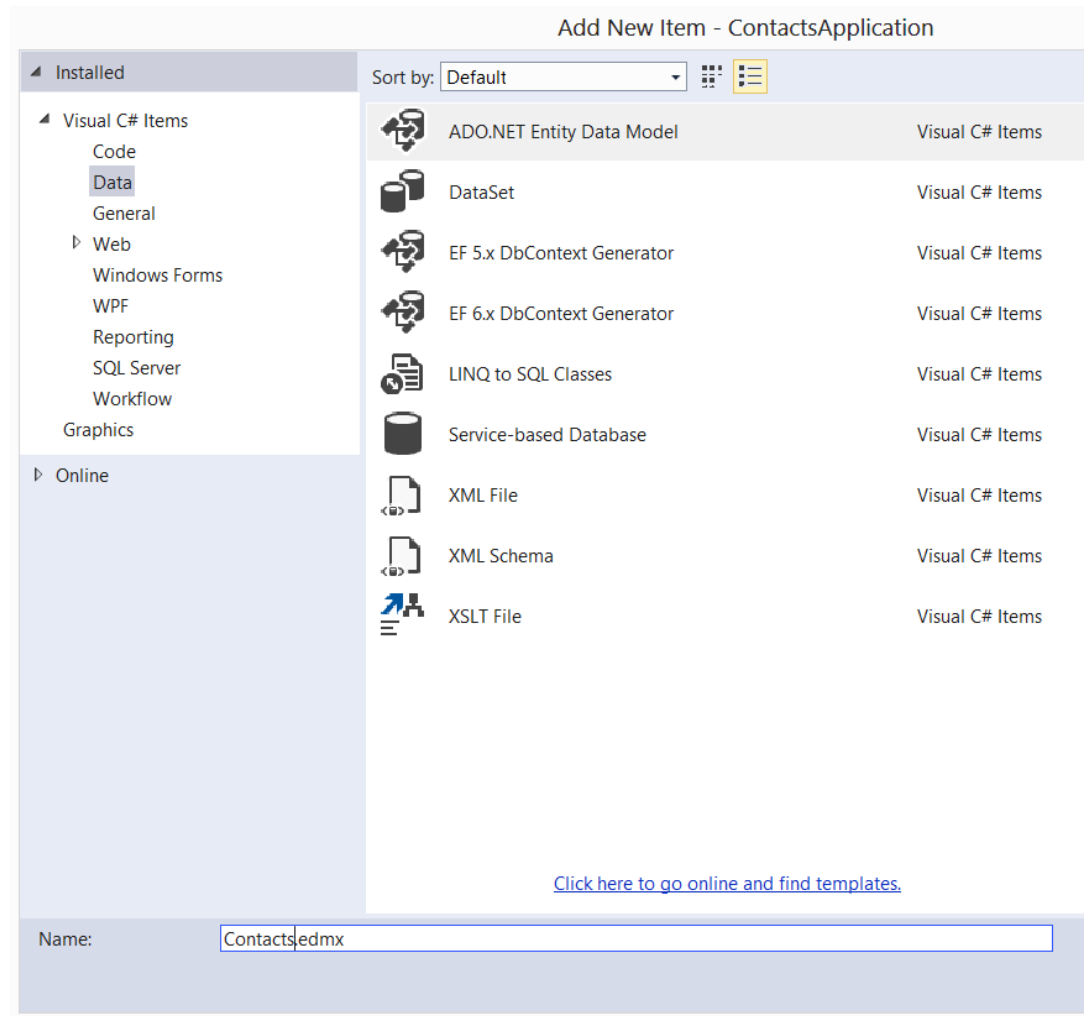
For example: let's say we want to create an app that contains the names and phone numbers of people we know.

Add a New Entity Data Model

Right-click on project → Add new item

In the Data menu, choose ADO.NET Entity Data Model and name it “Contacts.edmx”.

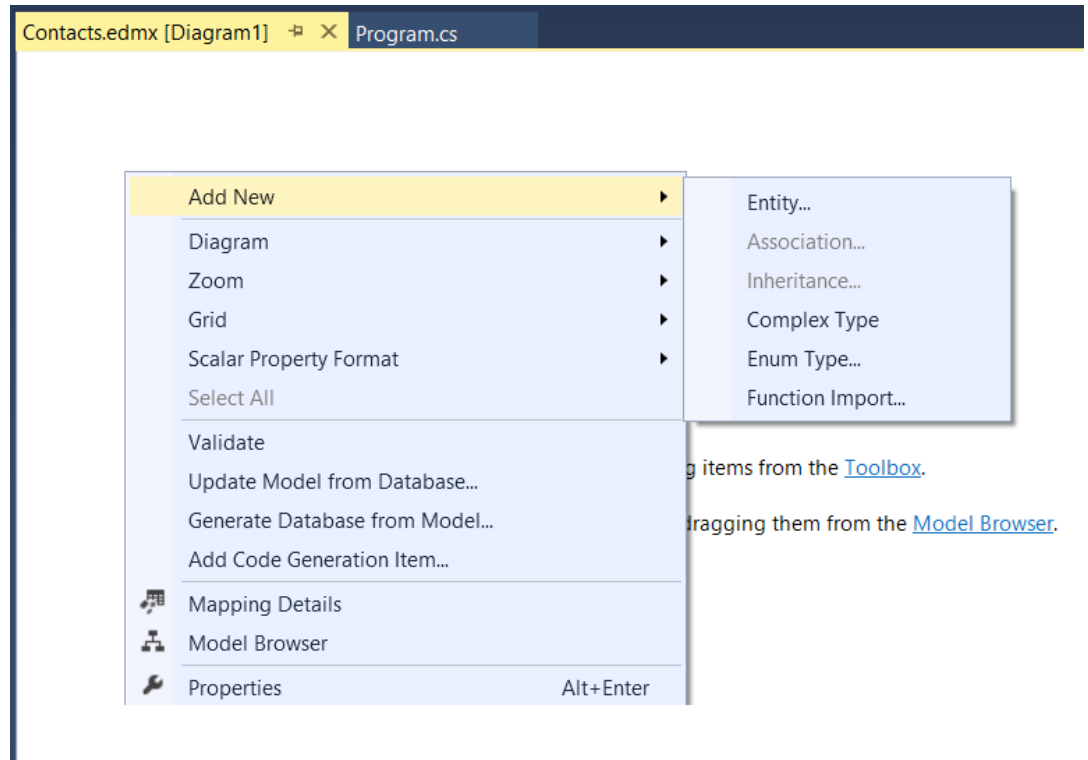
Choose “Empty Model” in the next step of the wizard and click finish.



Add a New Entity

Right-click the design surface and choose Add New → Entity

Give the Entity a name (Person), a set name (People), and check the box to create a key property of type Int32 called Id.



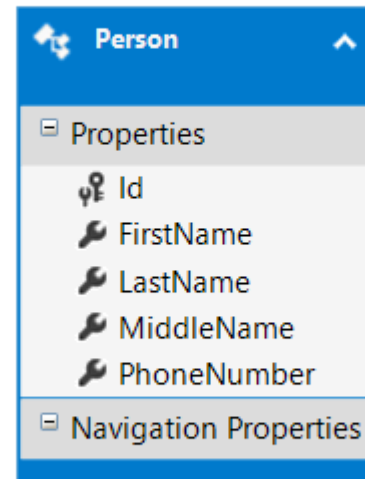
Add Scalar Properties

Right-click near the top of the Person entity. Select Add → Scalar Property.

Name the property FirstName.

Add additional properties for LastName, MiddleName, and PhoneNumber.

Right-click the Id property and select Properties. Make sure the StoreGeneratedPattern property is set to Identity. This tells the database to manage the keys and it will automatically create Ids when new people are added.



Set Up the Context

Right-click the design surface (not the Person entity) and select Properties. We need to set up the Entity Container Name and, optionally, we can set a Schema.

In SQL Server, tables and objects can be grouped into a schema. It makes securing things easier because you can give users permissions to a schema instead of each individual object.

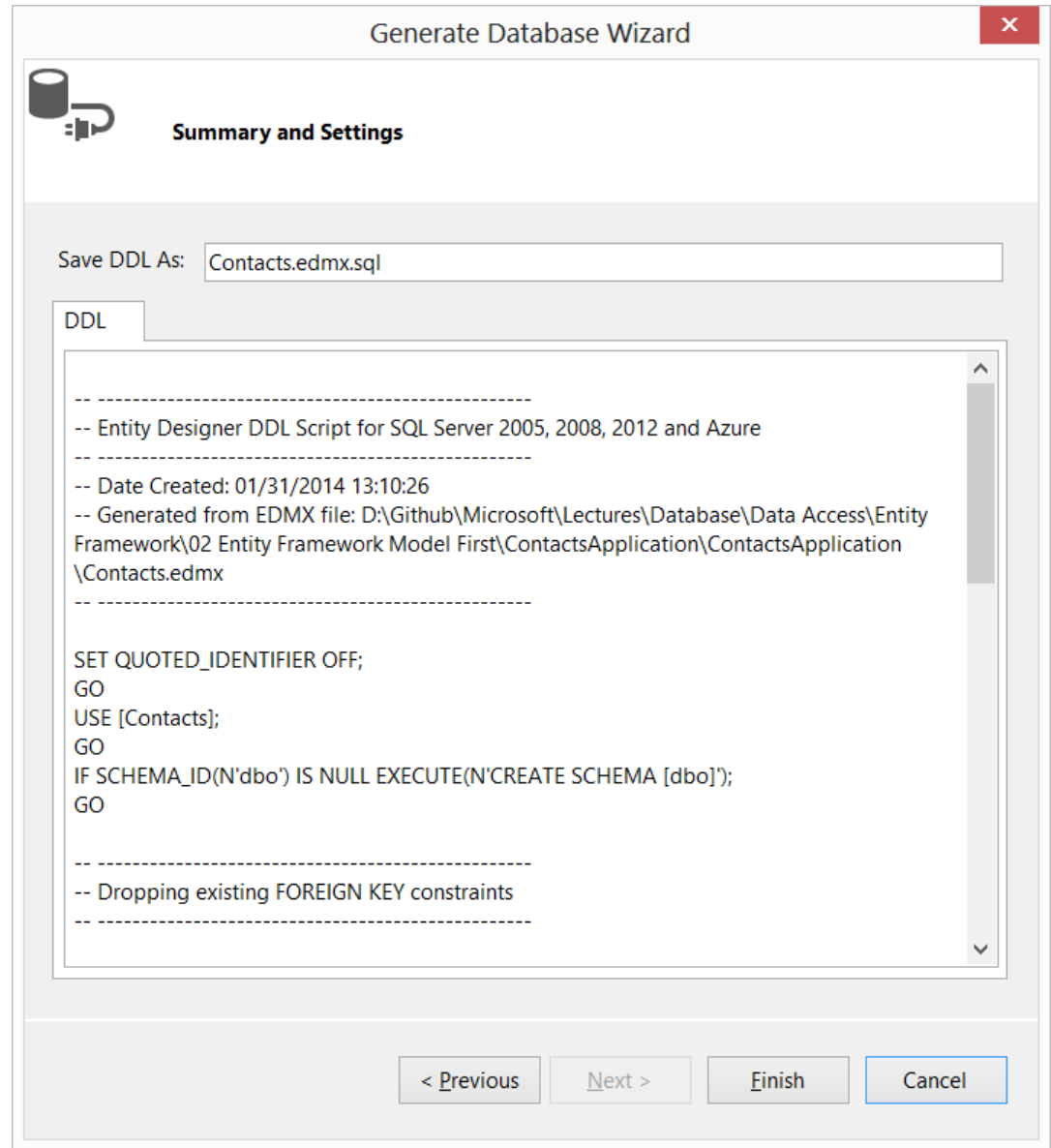
Properties	
Contacts ConceptualEntityModel	
Code Generation Strategy	T4
Connection String	
Database Generation Work	TablePerTypeStrategy.xml (VS)
Database Schema Name	dbo
DDL Generation Template	SSDLToSQL10.tt (VS)
Entity Container Access	Public
Entity Container Name	ContactsContext
Lazy Loading Enabled	True
Metadata Artifact Processing	Embed in Output Assembly
Namespace	Contacts
Pluralize New Objects	True
Transform Related Text Templates	True
Update Property Facets	True
Use Legacy Provider	True
Use Strong Spatial Types	False
Validate On Build	True

Generating a Database Script

Right-click the design surface and select Generate Database Script from Model.

Either select an existing Database Connection or Create a new one. Fill out all the login information and click Finish.

It will create a SQL Script that you can run in your database.



Run Script in Database

If you don't have a database yet, right-click on the databases folder in SQL Management Studio and add a new database called "Contacts."

Hit the New Query button, copy your generated script into the tab, and hit F5 to execute it.

```

private static void LoadSampleData()
{
    using (var context = new ContactsContext())
    {
        if (context.People.Count() == 0)
        {
            var person = new Person { FirstName = "Robert", MiddleName = " Allen", LastName = "Doe", PhoneNumber = "867-5309" };
            context.People.Add(person);

            person = new Person { FirstName = "John", MiddleName = " K.", LastName = "Smith", PhoneNumber = "824-3031" };
            context.People.Add(person);

            person = new Person { FirstName = "Billy", MiddleName = " Albert", LastName = "Minor", PhoneNumber = "907-2212" };
            context.People.Add(person);

            person = new Person { FirstName = "Kathy", MiddleName = " Anne", LastName = "Ryan", PhoneNumber = "722-0038" };
            context.People.Add(person);

            context.SaveChanges();
        }
    }
}

private static void DisplayContacts()
{
    using (var context = new ContactsContext())
    {
        foreach (var person in context.People)
        {
            Console.WriteLine("{0} {1} {2}, Phone: {3}", person.FirstName, person.MiddleName, person.LastName, person.PhoneNumber);
        }
    }
}

```