Entity Framework is a new technology from Microsoft. More likely than not, you are just beginning to explore Entity Framework and you are probably asking the question, “Okay, how do I get started?” If this describes you, this chapter is a great place to start. If, on the other hand, you have built some working models and feel comfortable with a few key modeling concepts such as entity splitting and inheritance, you can skip this chapter.

In this chapter, we will walk you through the basic examples of modeling with Entity Framework. Modeling is the core feature of Entity Framework and what distinguishes Entity Framework from previous Microsoft data access platforms. Once you have built your model, you can write code against the model rather than against the rows and columns in the relational database.

We start off this chapter with an example of how to create a simple conceptual model and let Entity Framework create the underlying database. In the remaining examples, we will show you how to create models from existing tables and relationships in your databases.

2-1. Creating a Simple Model

Problem

You have a brand new project and want to create a model with just one entity.

Solution

Let’s imagine you want to create an application to hold names and phone numbers of people you know. To keep things simple, let’s assume you need just one entity type: Person.

To create the new model, do the following:

1. Right-click your project and select Add ➤ New Item.
2. From the templates, select ADO.NET Entity Data Model and click Add. This template is located in Data under Visual C# Items. See Figure 2-1.
3. In the first step of the wizard, choose Empty Model and click Finish. The wizard will create a new conceptual model with an empty design surface.
4. Right-click the design surface and select Add ➤ Entity.
5. Type Person in the Entity name field and select the box to Create a key property. Use Id as the Key Property. Make sure its Property Type is `Int32`. Click OK, and a new Person entity will appear on the design surface. See Figure 2-2.

6. Right-click near the top of the Person entity and select Add ➤ Scalar Property. A new scalar property will be added to the Person entity.

7. Rename the scalar property `FirstName`. Add scalar properties for LastName, MiddleName, and PhoneNumber.

8. Right-click the Id property and select Properties. In the properties view, change the StoreGeneratedPattern property to Identity. This flags the Id property as a value that will be computed by the store layer (database). The database script we get at the end will flag the Id column as an identity column, and the storage model will know that the database will automatically manage the values in this column.

The completed conceptual model should look like the model in Figure 2-3.

---

**Figure 2-1.** Adding a new .edmx file that contains XML describing the conceptual model, storage model, and mapping layer