



Entity Framework技术概观 / 实体数据模型剖析 / 对象继承 /

数据编辑与维护——变更、存储与冲突 / ASP.NET Web Forms / MVC整合应用

Entity Framework

实用精要

吕高旭 著

大数据时代的信息检索技术



清华大学出版社



Entity Framework

实用精要

吕高旭 著

清华大学出版社
北京

内 容 简 介

Entity Framework 是微软公司 .NET 平台支持的对象关联映射技术, 开发人员可以通过实体数据模型对象与数据库进行沟通, 建立更坚固的商业应用程序系统, 避免因为通过 ADO.NET 直接处理 SQL 语句而带来的风险与可能引发的错误。本书通过介绍Entity Framework与 LINQ 开发实战的案例, 以 Entity Framework 技术内容的讨论为主线, 结合关键的 LINQ技巧说明, 提供读者系统性学习 Entity Framework 所需的内容。

本书旨在帮助读者进入 Entity Framework的世界, 建立必要的技术能力, 同时希望读者在完成本书的教学课程之后, 能够更进一步地将其运用在实际的项目开发中。

本书为荣软科技股份有限公司授权出版发行的中文简体字版本。

北京市版权局著作权合同登记号 图字: 01-2017-8673

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Entity Framework实用精要 / 吕高旭著. —北京: 清华大学出版社, 2018
ISBN 978-7-302-48593-3

I. ①E… II. ①吕… III. ①计算机网络—程序设计 IV. ①TP393.09

中国版本图书馆CIP数据核字(2017)第255028号

责任编辑: 夏毓彦

封面设计: 王 翔

责任校对: 闫秀华

责任印制: 王静怡

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦A座

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 清华大学印刷厂

经 销: 全国新华书店

开 本: 190mm×260mm

印 张: 22.25

字 数: 570千字

版 次: 2018年1月第1版

印 次: 2018年1月第1次印刷

印 数: 1~3000

定 价: 79.00元

产品编号: 076916-01

前 言

程序设计师编写网页商业应用程序时离不开对数据库的运用，基本可以认为，不懂得运用数据库的网页应用程序设计师只是完成了网页美工设计师“委托的”工作。因为几乎所有的商业化网页用程序都要在后台连接与访问数据库。

然而，没有面向对象的自动化工具，程序设计模型和数据库集成在一起就会让程序逻辑和数据模型杂乱无章地混合在一起，不但开发大型网页商业应用软件无望，就连小型的网页应用程序也没有扩展的能力，将来维护起来更是可以用“一塌糊涂”来形容。

微软公司在它的ADO.NET这个数据库访问利器的基础上融入了对象关系映射（Object-Relational Mapping）的技术，再加上一个设计的工具（Designer），就构成了现在的ADO.NET Entity Framework，就是以实体数据模型为主，通过Entity Framework Designer（EF设计器），利用抽象化数据结构的方式，将每个数据库对象转化为应用程序实体（或对象），让数据库的E/R模型（实体-关联模型）也就转换为对象模型。这样，让程序设计师能运用自己最熟悉的面向对象的程序设计语言来轻松自如且安全可靠地调用和访问数据库，让大型网页商业应用软件的开发也不是难事。

Entity Framework 是微软 .NET 平台支持的对象关联映射技术，开发人员可以通过实体数据模型对象与数据库进行沟通，建立更坚固的商业应用程序系统，避免因为通过 ADO.NET 直接处理 SQL 语句伴随的风险与可能产生的错误。

从传统的 ADO、ADO.NET 到现在的 Entity Framework，善用新技术让我们在新时代走得更稳，本书主要的目标，除了带领读者进入 Entity Framework 的世界，建立必要的技术能力，同时希望读者在完成书本的学习之后，能够更进一步地将其运用在实际项目开发中。

本书为了方便读者自学或者学校教学，所有的范例程序都是在微软公司提供的免费版 Visual Studio Community 2017中用C# 语言为主体编写完成的，所有范例程序都经过了调试和测试，可以顺利运行。

微软的Visual Studio Community 2017可以从微软公司的官方网站下载，网址如下：

<https://www.visualstudio.com/zh-hans/thank-you-downloading-visual-studio/?sku=Community&rel=15>

本书的所有范例程序源码及其运行必需的文件，读者都可以从下面的网址下载压缩的文件包，之后解压缩到自己的Visual Studio项目文件夹中即可。

这里提供下载范例程序的网址：

<http://pan.baidu.com/s/1cuEVUm>（注意区分数字和英文字母大小写）

如果下载有问题，请发送电子邮件至booksaga@126.com，邮件主题设置为“求Entity Framework实用精要下载资源”。

最后做一点说明：我们在编写这些范例程序的时候，为了适用性，大多数范例程序都选择连接的是 Visual Studio Community 2017 集成开发环境中自带的虚拟数据库 (LocalDb)\MSSQLLocalDB，因此范例程序对数据库的操作都是对加载到内存中的这个虚拟数据库进行增、删、改等操作，范例程序运行时可以反映出这种变化，但是不会实际修改范例程序运行时载入的实际数据库文件。如果读者要把范例程序拓展到对实际数据库的操作，那么在需要时按照实际应用建立本地或者网络远端的实际数据库，并提供访问的服务即可。

编者
2017年9月

目 录

| | |
|------------------------------------|----|
| 第1章 Entity Framework概观 | 1 |
| 1.1 初探Entity Framework | 1 |
| 1.2 Entity Framework与ADO.NET | 9 |
| 1.3 对象关联映射——ORM | 13 |
| 1.4 Entity Framework开发流程 | 14 |
| 1.5 空EF设计器模型 | 16 |
| 1.6 使用Code First | 27 |
| 1.6.1 来自数据库的Code First | 27 |
| 1.6.2 空Code First模型 | 31 |
| 1.7 小结 | 34 |
| 第2章 实体数据模型 | 35 |
| 2.1 实体数据模型概观 | 35 |
| 2.2 模型工具与.edmx文件 | 36 |
| 2.2.1 概念模型 | 39 |
| 2.2.2 存储模型 | 42 |
| 2.2.3 映射 | 42 |
| 2.3 实体数据模型与类文件 | 44 |
| 2.4 POCO与实体数据模型 | 46 |
| 2.5 实体类结构映射 | 53 |
| 2.5.1 数据注解属性 | 54 |
| 2.5.2 Fluent API | 61 |
| 2.6 EF Designer数据库更新 | 65 |
| 2.7 Code First数据库迁移 | 71 |
| 2.7.1 结构变动迁移 | 76 |
| 2.7.2 查看SQL | 78 |
| 2.7.3 指定迁移 | 79 |

| | |
|---------------------------------------|-----------|
| 2.8 关于DbContext | 79 |
| 2.8.1 传送SQL语句与数据转换 | 80 |
| 2.8.2 查询与连接议题 | 81 |
| 2.8.3 管理更新操作 | 82 |
| 2.8.4 DbContext对象生命期 | 83 |
| 2.8.5 覆写OnModelCreating方法 | 83 |
| 2.8.6 管理与操作数据库——Database属性 | 83 |
| 2.8.7ObjectContext..... | 85 |
| 2.9 DbSet..... | 87 |
| 2.10 追踪本地数据集..... | 91 |
| 2.11 小结..... | 94 |
| 第3章 实体数据模型与LINQ..... | 95 |
| 3.1 LINQ与Entity Framework | 95 |
| 3.2 Select——选择与投影 | 99 |
| 3.2.1 类型转换..... | 100 |
| 3.2.2 多重from与SelectMany | 103 |
| 3.3 Where与数据筛选 | 107 |
| 3.4 排序..... | 113 |
| 3.5 分组..... | 116 |
| 3.6 转换..... | 122 |
| 3.6.1 AsEnumerable | 122 |
| 3.6.2 Array与List | 125 |
| 3.6.3 ToDictionary | 128 |
| 3.6.4 ToLookup..... | 131 |
| 3.7 异步查询..... | 134 |
| 3.8 关联——Join | 136 |
| 3.8.1 GroupJoin..... | 139 |
| 3.8.2 Inner join与Left Outer Join..... | 141 |
| 3.8.3 Entity Framework与Join | 142 |
| 3.9 LINQ、SQL与Entity SQL..... | 146 |
| 3.9.1 关于SQL语句 | 147 |
| 3.9.2 Entity SQL..... | 147 |
| 3.9.3 Entity SQL参数..... | 149 |
| 3.10 小结..... | 150 |

| | |
|--|-----|
| 第4章 Entity Framework与扩充方法 | 151 |
| 4.1 Enumerable扩充方法 | 151 |
| 4.2 聚合运算 | 152 |
| 4.2.1 计算平均值 | 153 |
| 4.2.2 Average方法与Func | 155 |
| 4.2.3 聚合函数 | 158 |
| 4.3 设置操作 | 160 |
| 4.4 产生操作 | 164 |
| 4.4.1 Empty与DefaultIfEmpty | 164 |
| 4.4.2 Range | 166 |
| 4.4.3 Repeat | 167 |
| 4.5 计数运算 | 169 |
| 4.5.1 All | 169 |
| 4.5.2 Any | 171 |
| 4.5.3 Contains | 172 |
| 4.6 分割运算 | 173 |
| 4.6.1 Skip与SkipWhile | 174 |
| 4.6.2 Take与TakeWhile | 176 |
| 4.7 元素操作 | 178 |
| 4.7.1 First与FirstOrDefault | 178 |
| 4.7.2 Last与LastOrDefault | 180 |
| 4.7.3 ElementAt与ElementAtOrDefault | 182 |
| 4.7.4 Single与SingleOrDefault | 183 |
| 4.8 扩展方法的异步版本 | 185 |
| 4.9 小结 | 186 |
| 第5章 数据关联与继承 | 187 |
| 5.1 与关联数据互动 | 187 |
| 5.2 Code Frist与关联设计 | 194 |
| 5.2.1 Code First关联设置惯例 | 196 |
| 5.2.2 数据注解 | 200 |
| 5.3 处理特定关联定义 | 203 |
| 5.3.1 一对一 (1 to 0..1) | 203 |
| 5.3.2 多对多 (* to *) | 208 |
| 5.3.3 多对多关联——附加数据字段 | 213 |
| 5.4 使用CodeFirst设计关联 | 216 |

| | | |
|-------|---|-----|
| 5.5 | Lazy Loading与Eager Loading | 223 |
| 5.5.1 | 调整virtual属性 | 229 |
| 5.5.2 | Include方法与对象加载 | 231 |
| 5.5.3 | Include方法与异步数据加载 | 234 |
| 5.6 | 继承 | 235 |
| 5.6.1 | Table Per Hierarchy (TPH) | 235 |
| 5.6.2 | Table Per Type (TPT) | 241 |
| 5.6.3 | Table Per Concrete Class (TPC) | 244 |
| 5.7 | 复杂类型 | 248 |
| 5.8 | 小结 | 251 |
| 第6章 | 数据编辑与维护 | 252 |
| 6.1 | 数据编辑操作 | 252 |
| 6.1.1 | 实体数据对象状态 | 255 |
| 6.1.2 | 更新与删除 | 256 |
| 6.1.3 | Attach | 259 |
| 6.1.4 | 变更追踪——DbContext.ChangeTracker | 263 |
| 6.1.5 | 更新验证异常——DbEntityValidationException | 268 |
| 6.1.6 | DbContext.ValidateEntity方法 | 271 |
| 6.1.7 | 覆写SaveChange | 272 |
| 6.1.8 | 异步的变更更新 | 274 |
| 6.2 | SQL语句 | 275 |
| 6.2.1 | 使用参数 | 279 |
| 6.2.2 | 执行非查询变更指令——ExecuteSqlCommand | 280 |
| 6.3 | 存储过程 | 281 |
| 6.3.1 | 直接调用存储过程 | 288 |
| 6.3.2 | 获取返回值 | 289 |
| 6.3.3 | CodeFirst自动设置存储过程 | 290 |
| 6.4 | 数据变更冲突 | 294 |
| 6.4.1 | 并发冲突处理——Database Wins或Client Wins | 298 |
| 6.4.2 | SQL语句与数据冲突 | 300 |
| 6.4.3 | 存储过程与数据冲突 | 300 |
| 6.4.4 | ConcurrencyCheck注解 | 302 |
| 6.5 | 事务处理 | 304 |
| 6.5.1 | 共享事务处理连接 | 305 |
| 6.5.2 | 使用TransactionScope | 305 |
| 6.6 | 复杂类型的数据对象变更 | 308 |

| | |
|--------------------------------------|-----|
| 6.7 小结..... | 310 |
| 第7章 ASP.NET整合应用 | 311 |
| 7.1 Web Forms与Entity Framework | 311 |
| 7.1.1 查询..... | 314 |
| 7.1.2 更新..... | 315 |
| 7.1.3 删除..... | 316 |
| 7.1.4 处理单项数据..... | 316 |
| 7.1.5 关联设计..... | 319 |
| 7.2 MVC与Entity Framework..... | 321 |
| 7.2.1 查询..... | 325 |
| 7.2.2 更新/删除..... | 326 |
| 7.2.3 添加..... | 330 |
| 7.2.4 关系型数据..... | 332 |
| 7.3 WCF服务与实体数据模型 | 335 |
| 7.3.1 WCF应用程序项目 | 335 |
| 7.3.2 数据合约..... | 339 |
| 7.3.3 整合Entity Framework..... | 340 |
| 7.4 小结..... | 346 |

第1章 Entity Framework 概观

Entity Framework是.NET平台支持的对象关联映射技术,以实体数据模型搭配LINQ支持对象化的数据操作,和SQL语句相比,结构化的数据对象除了易于应用程序的整合外,同时可以避免常见的类型错误问题。本书第1章介绍Entity Framework的概念,同时示范各种不同模式的Entity Framework开发流程。

1.1 初探Entity Framework

过去我们通过ADO.NET提供的类把SQL语句传送到底层数据库引擎进行解析,执行特定的数据操作,然后获取返回的结果。ADO.NET/SQL数据存取架构示意图如图1-1所示。

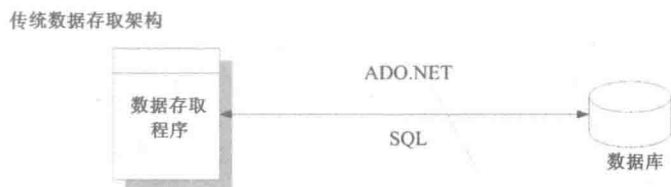


图 1-1

因为直接处理SQL语句这种系统开发方式具有非常大的弹性,但相对的也存在一些风险,从SQL语句的处理到数据类型的转换都可能发生错误,特别是随着系统规模逐渐增长,这些错误造成的问题将越发严重。

Entity Framework通过实体数据模型映像到底层的数据结构,应用程序开发人员只需编写C#程序代码来处理实体数据对象,从而降低了因为不当的SQL语句操作而导致的程序错误与安全性问题。由于数据对象本身具有类型,因此在程序开发的过程中避免了因类型转换可能导致错误的程序代码。

图1-2所示为Entity Framework的数据存取架构示意图。

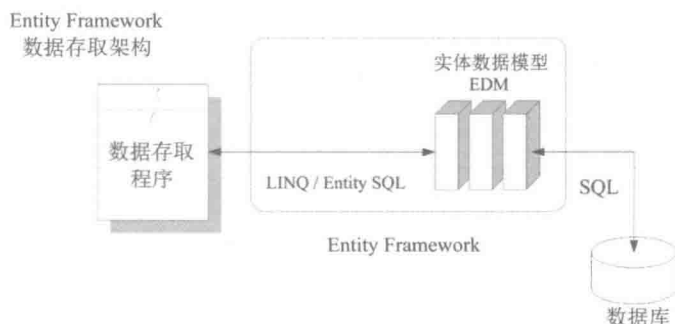


图 1-2

比较图1-1与图1-2，我们可以看到传统ADO.NET的数据存取架构是由开发人员直接处理SQL语句来建立应用程序功能的，而Entity Framework则是通过LINQ操作实体数据模型，由Entity Framework接手处理应用程序与后端数据库互动操作的细节。

接下来从最简单的控制台应用程序项目开始，说明如何通过Entity Framework进行数据库连接与存取操作。

通过Entity Framework进行数据存取

为了方便示范说明，本章范例文件夹配置了一个简单的SQL Server数据库文件KTStore.mdf，其中仅包含单个数据表Product，如表1-1所示。

表1-1 KTStore数据库/Product数据表

| Id | Name | Category | Price |
|----|---------------------------|----------|-------|
| 1 | ASP.NET MVC商业应用开发实践 | Book | 85 |
| 2 | Entity Framework实用精解 | Book | 80 |
| 3 | HTML5从零开始 | Book | 55 |
| 4 | HTML5完美风暴 | Book | 100 |
| 5 | Entity Framework与LINQ开发实战 | Book | 59 |
| 6 | ASP.NET商业级数据库网站开发实践 | Book | 65 |
| 7 | Visual C# 2010精要剖析 | Book | 52 |
| 8 | 3U工业用机箱 | PC | 400 |
| 9 | CPU散热器MITX | PC | 95 |
| 10 | 四热管多合一CPU风扇 | PC | 98 |
| 11 | PC RS232 8端口扩展卡 | PC | 102 |
| 12 | ASUC变压器24V 3.42A | PC | 360 |
| 13 | 8 Port USB充电器 | PC | 162 |
| 14 | 铝壳24000mAh+Apple 20米传输线 | Mobile | 250 |
| 15 | 移动电源超薄型85000mAh | Mobile | 150 |
| 16 | 移动电源双USB锂离子电芯65000mAh | Mobile | 180 |
| 17 | 手机Shap锂电池 | Mobile | 160 |
| 18 | AC转USB充电器10A | Mobile | 28 |
| 19 | USB复古4 英寸风扇（铁制） | USB | 85 |
| 20 | LED灯B-LED | USB | 25 |
| 21 | USB3.0鼠标键盘共享传输线 | USB | 150 |
| 22 | USB3.0 18端口HUB | USB | 145 |
| 23 | 7 Port USB充电器 | USB | 220 |

表1-1列举了Product数据表结构与存储的数据内容。下面创建一个控制台应用程序项目，新建ADO.NET实体数据模型来支持EF操作。

范例 1-1 EFDemo | 通过 EF 存取数据库

创建 Visual C# 控制台应用程序项目 EFDemo，在项目节点展开菜单，如图 1-3 所示。

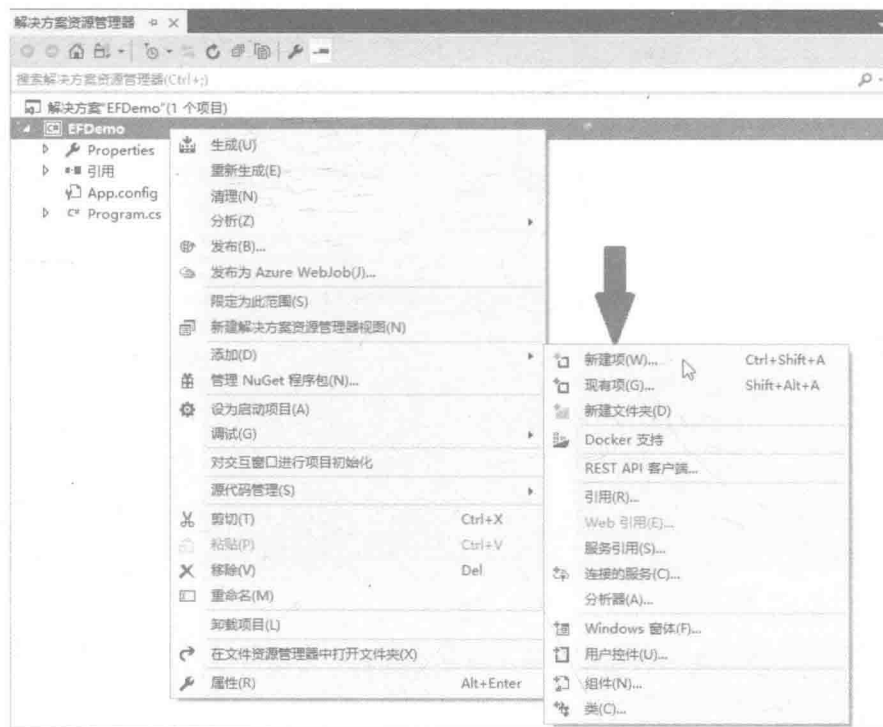


图 1-3

单击“新建项”，打开“添加新项”对话框，如图 1-4 所示。



图 1-4

在其中选择“ADO.NET 实体数据模型”，下方的名称字段输入 KTStoreModel 的名称（也可以自定义或直接保留默认名称 Model1），单击“添加”按钮开始设置程序。

接下来出现的是“实体数据模型向导”设置窗口，要求开发者选择模型内容，由于我们要通过现在的数据库建立实体数据模型，因此现在指定选择“来自数据库的EF设计器”，单击“下一步”按钮继续设置，如图1-5所示。



图 1-5

在“选择您的数据连接”界面中，单击“新建连接”按钮（见图1-6），打开“连接属性”对话框，如图1-7所示。



图 1-6

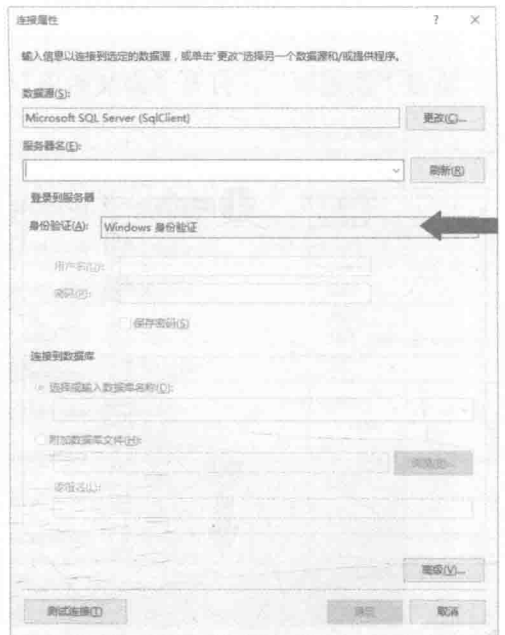


图 1-7

由于要连接的是数据库文件，因此单击“更改”按钮，打开“更改数据源”对话框，如图1-8所示。

在其中选择“Microsoft SQL Server数据库文件”，单击“确定”按钮，回到“连接属性”对话框，如图1-9所示。

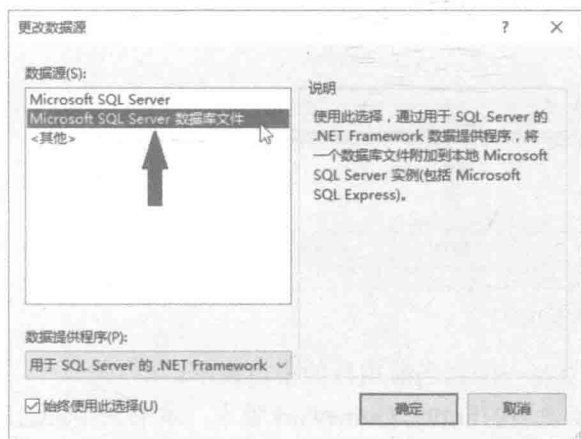


图 1-8



图 1-9

与之前的“连接属性”对话框不同，现在单击“浏览”按钮，找到本章范例文件夹中的KTStore.mdf数据库，单击界面左下角的“测试连接”按钮，出现“测试连接成功”的消息框，表示完成设置，如图1-10所示。单击“确定”按钮，回到“选择您的数据连接”界面，如图1-11所示。



图 1-10



图 1-11

界面中间的“连接字符串”文字为连接至KTStore.mdf文件的连接字符串，界面最下方保留“将App.Config中的连接设置另存为”的勾选状态，KTStoreEntities是存储在App.Config配置文件中的数据库连接名称，作为程序连接数据库所需的信息，EF通过这个名字找到此“连接字符串”，进行数据库的存取。

请特别注意其中连接字符串的内容，其中包含了KTStore.mdf的绝对路径，单击“下一步”按钮会出现一个说明消息框，询问是否将数据库复制到项目的根目录，如图1-12所示。

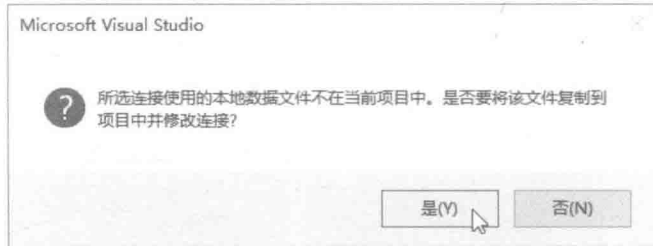


图 1-12

单击“是”按钮，此时KTStore.mdf会被复制一份到当前项目的根目录下。

接下来的界面是让开发人员自行指定所要使用的Entity Framework版本，本书采用最新的实体框架6.x，如图1-13所示。继续往下讨论之前，先单击“上一步”按钮，回到“选择您的数据连接”界面，如图1-14所示。



图 1-13



图 1-14

其中的连接字符串已经调整为根目录路径 |DataDirectory|\KTStore.mdf，为了方便查看，列举如下：

```
metadata=res://*/KTStoreModel.csdl|res://*/KTStoreModel.ssdl|res://*/KTStoreModel.msl;
provider=System.Data.SqlClient;provider connection string="data source=(LocalDB)\MSSQLLocalDB;
attachdbfilename=|DataDirectory|\KTStore.mdf;
integrated security=True;connect
timeout=30;MultipleActiveResultSets=True;App=EntityFramework"
```

|DataDirectory|表示根目录，因此会以根目录下的数据库为操作目标。单击“下一步”按钮，进入最后一个设置界面“选择您的数据库对象和设置”，如图1-15所示。



图 1-15