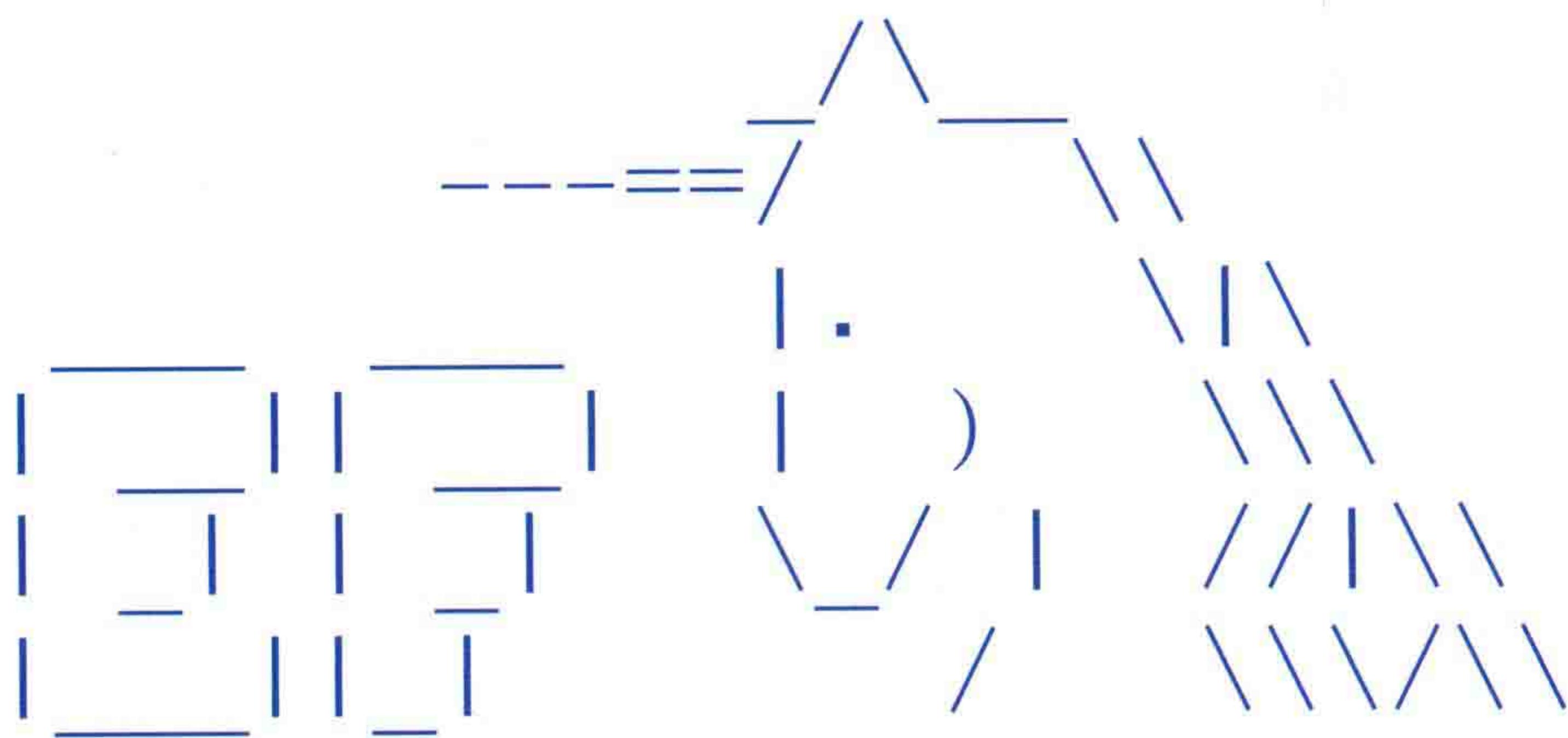


## 深入EF与EF Core技术，掌握跨平台ORM框架

- 由基础到进阶，经作者精心编排，均结合实际小型案例有理有据论证而来
- 掌握基础是必备前提，在完善基础的必要条件下，你会深化基础、步向进阶
- 吸收他人经验，但仍需进行反思，继而抽象概括，最终应用于实践、达到内化



# 你必须掌握的 Entity Framework 6.x与Core 2.0

汪鹏 著

 示例项目源代码

清华大学出版社



内容简介



# 你必须掌握的 Entity Framework 6.x与Core 2.0

汪鹏 著

清华大学出版社  
北京

## 内 容 简 介

本书针对 Entity Framework 6.x 和 Entity Framework Core 2.0 以通俗易懂的语言进行深入、全面的探讨，并结合实例来介绍如何灵活运用。

全书共四篇，内容分为 19 章，第一篇（第 1~3 章）讲述 Entity Framework 6.x 基础，并深入讲解如何创建过滤索引、迁移的本质等；第二篇（第 4~8 章）讲述 Entity Framework 6.x 进阶，充分阐述 Entity Framework 6.x 背后隐藏的原理、性能优化、并发解析等；第三篇（第 9、10 章）讲述 Entity Framework Core 2.0 基础，并对 Entity Framework Core 每一版本所出现的新特性进行详细介绍和深入研究；第四篇（第 11~19 章）讲述 Entity Framework Core 2.0 进阶，内容包括 Entity Framework Core 变更追踪策略、创建上下文实例方式、性能优化，结合 Entity Framework Core 2.0 特性实现多租户模式，利用 Entity Framework Core 进行数据审计、并发解析等。

本书适合 ASP.NET 开发人员阅读，也适合高等院校和培训学校相关专业的师生作为教学参考书。

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

你必须掌握的 Entity Framework 6.x 与 Core 2.0 / 汪鹏著. — 北京：清华大学出版社，2018

ISBN 978-7-302-50017-9

I. ①你… II. ①汪… 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（2018）第 081995 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：宋 林

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](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈：010-62772015，[zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 装 者：三河市君旺印务有限公司

经 销：全国新华书店

开 本：190mm×260mm 印 张：31.25 字 数：800 千字

版 次：2018 年 7 月第 1 版 印 次：2018 年 7 月第 1 次印刷

印 数：1~3500

定 价：99.00 元

产品编号：078113-01

# 推荐序

作者是我们博客园的知名博主，我读过他写的很多关于 EF 与 EF Core 的博文，从中学到很多，作者对待技术的认真、严谨、刨根问底让人印象深刻。深厚的技术功底，对 EF 家族的持续探索，通过博客的千锤百炼，在我眼里，这本书是作者长期辛勤耕耘的自然成果。

博客园创始人 杜勇

当我第一次看到这本书目录的时候，就感到很惊喜，因为这正是我所需要的 EF 书籍。在过去，一些从事.NET 开发工作的朋友让我推荐几本有关 Entity Framework 方面的书，可找来找去一本国内的书都找不出，最终只好推荐几本国外的英文书。国外的一些开发书籍比较侧重理论，喜欢讲原理，而我相信大多数国内的开发人员更喜欢理论和实践相结合的书，这是由国内的行情决定的。而此书的出现无疑弥补了国内书籍在 Entity Framework 方面的空白，书中的内容几乎涵盖了 EF 在工作当中的各种知识点及应用场景，不得不说这是作者多年 EF 开发经验的精华，对于所有想学习 EF 人来说，此书都将是一本非常不错的参考书。

深圳市紫衡技术有限公司.NET 软件架构师 邹琼俊

Entity Framework 和 Entity Framework Core 都是微软的 ORM 技术，在社区它们有着褒贬不一的评价，这本书同时涵盖了 Entity Framework 和 Entity Framework Core 的技术，让你一窥 EF 技术体系，做到知己知彼，避开错误使用的坑，不人云亦云。

腾讯高级工程师 张善友

Entity Framework 伴随.NET 3.5 发布以来已有近 10 年历史，但国内在这方面的资料却是七零八落，且关于 EF 和 EF Core 的中文书籍几乎是一片空白，本书从 Entity Framework 到最新的 Entity Framework Core，无论是基础内容还是进阶内容皆有详细讲解，从作者在博客上发表的文章就知其深度和独特的见解，对于想全面和深入学习 Entity Framework 和跨平台的 Entity Framework Core 的同学，此书可作为首选。

博客园知名博主 赵培

Entity Framework 是 .NET 开发平台主流的数据访问框架，虽然面世近 10 年，却始终没有一本中文教程得以出版，这增加了初学者的学习难度，本书的出版填补了国内 EF 技术的空白，实为 .NET 开发人员之幸。

EF 是一个“真正”的 ORM（对象关系映射）框架。为什么这样说，因为很多开发人员经常用一些“轻量级”的 ORM 与 EF 进行比较，进而得出 EF 性能低下难以使用的结论。他们忽视了 EF 提供的大量复杂而重要的功能，比如工作单元、类层次关系映射、值对象映射等特性。要把 EF 用好，除了了解 EF 本身外，还需要了解 EF 的关键使用场景以及面向对象设计的一些知识。

EF 的一个关键使用场景是利用面向对象思想来开发复杂的业务逻辑，这正是 EF 提供复杂特性的原因，如果你对面向对象基本方法没有认识，那么 EF 对你的帮助就会大打折扣。

当你使用 EF 来编写复杂的查询时，可能会发现编写的代码冗长，生成的 SQL 复杂且性能低下，优化也比较困难。对于这种场景，采用 EF 和 LINQ 并不是值得推荐的方案，相反，使用纯 SQL 会得到更大的自由。你可以使用 EF 来执行 SQL，不过有经验的开发人员更倾向于使用轻量级的工具，比如 Dapper。

多种数据访问工具配合是正确使用 EF 的关键。一些开发人员希望采用一种统一的数据访问技术，这种想法过于理想，每种技术有其优势和局限，技术选型的失误往往会导致项目失利。

除了错误使用场景外，EF 还可能因为错误的使用方式而导致性能故障。

EF 的头号性能杀手是延迟加载特性，在我参与的某个项目中，一位初级水平的同事使用 EF 进行数据导入操作时，大致 5 个表，每个表不到一万条记录，一次导入操作需要 3 个小时，排查原因后发现，原来他进行了嵌套循环，并使用了延迟加载，后来使用 Include 即时加载数据，一分钟不到就导入完毕。对于延迟加载，通过观察生成的 SQL 日志就可以了解发出了多少 SQL 请求，进而排查性能故障。

EF 另一个容易引发的性能问题是由更改跟踪特性引起的。对于一个导入数据的场景，先不谈是否提交到数据库，只是把 100 万个对象添加到 EF 对象集合中就需要花费数分钟时间，所以 EF 对象集合不应该当成普通的集合使用。对于这个场景，可以把对象先放入一个普通集合，再通过 AddRange 批量添加到 EF 对象集合。

本书不仅是一本 EF 的常规使用教程，更是浓缩作者心血的经验之作，本书对 EF 的原理、架构、故障排查等均有详细论述，是 .NET 开发者不可多得的优秀图书。本书的出版是中国 .NET 开发者之幸事、.NET 社区之盛事，感谢本书作者对 .NET 社区的贡献。

本人作为中国 .NET 社区 NCC (<https://github.com/dotnetcore>) 的项目成员之一，欢迎优秀框架的作者和作品加入，为壮大社区力量、扩大 .NET 技术栈的影响力共同努力。

.NET Core 平台 Util 应用框架作者 何镇汐

# 前言

## 本书背景

在 Web 开发中，ASP.NET MVC 使用相当广泛，与之呼应的是使用 ORM 框架 Entity Framework 作为概念上的数据访问层，目前有些互联网招聘公司招聘时会明确要求熟练掌握 Entity Framework。很多人对 Entity Framework 嗤之以鼻，不可否认，Entity Framework 的性能确实是一个令人头疼的问题，但是还未到弃用的地步，大部分情况下还是我们的认识太浮于表面，没有深究原理，看清其本质。微软为我们封装了一切，我们只需进行傻瓜式的操作即可，但是最终带来的却是疯狂吐槽，请深入思考自己对 Entity Framework 是否有足够了解。笔者看过并搜索过很多关于 Entity Framework 和 Entity Framework Core 的博客，发现大部分内容介绍的太过浅显，学习一门新技术需要的是细嚼慢咽而非囫圇吞枣。对于中文版的 Entity Framework 和 Entity Framework Core，市面上几乎没有一本著作进行相关介绍，导致很多人都是跟着别人的脚步笼统地学着，且一知半解、不知所云。基于以上几点，笔者打算尽最大的能力分享自己所理解的 Entity Framework 和 Entity Framework Core，让大家能够循序渐进并系统地学习。

## 本书特点

本书中 Entity Framework 和 Entity Framework Core 的内容完全不是从官方网站摘抄和翻译而来，如果你看过官方网站的目录介绍就会发现和笔者所列举的目录截然不同，从基础到进阶都经过笔者的精心编排，同时一切内容都不会蜻蜓点水似的泛泛而谈，均结合实际小型案例有理有据论证而来，掌握基础是必备前提，在完善基础的充分条件下，你会进一步深入基础。原理是必要条件，在掌握原理的条件下，从吸收到消化才是最终的归宿。如果你在实际项目中正在用 Entity Framework 6.x，相信 Entity Framework 6.x 的内容会对你有进一步提高，如果你还在担心 Entity Framework 6.x 的性能，那么可阅读 Entity Framework Core 2.0 的内容，相信会让你刮目相看。

## 本书读者

本书 Entity Framework 6.x 的内容适合想快速提高的初学读者、想基础拾遗的回炉读者、想进一步提高的进阶读者，而 Entity Framework Core 针对已跨平台的 .NET Core 应运而生，对

于想要学习新技术的初学读者和已在项目中应用而需进一步深入的进阶读者均具有很强的参考和学习价值，同时高等院校和.NET 技术培训机构也可选择本书作为参考教材。

## 代码、勘误和支持

本书中的实例代码已上传至 Github 上 (<https://github.com/wangpengxpy>)。

虽然笔者耗费大量时间对书稿进行反复的修改和推敲，但是水平有限，仍难免有疏漏之处，为了使本书能够更好地为读者服务，如果有问题，可通过 QQ 邮箱 (2752154844@qq.com) 与笔者的博客 (<http://www.cnblogs.com/createmyself>) 与本人联系。笔者将竭力解决所有问题，并向你的指正致谢。

## 致谢

感谢夏毓彦编辑以及出版社其他人对本书所做的工作，正是你们的贡献使得这本书得以出版。你们所做的工作提升了本书的质量和准确性，使本书的品质能够得到保证，也让笔者更加有信心将自己对技术的理解展示给需要的读者。

写好一本书所投入的精力和时间无法估量，没有家人的支持和女友的理解，这本书不可能面世。感谢我的父亲、母亲、哥哥、姐姐，他们永远是我归宿的港湾，是我人生的支点和明灯，谢谢他们恒久不变的支持。特别感谢我的女友刘瑶，因为写作本书花费了大量时间而忽略了对她的关心，感谢她的体贴入微。现在，本书算是告一段落，终于有时间去多陪陪她了。

当然，最重要的是感谢一直以来关注笔者的同行技术者，是他们的鼓舞和肯定使得笔者在技术路上走得更远，感谢博客园给予的平台，让笔者能够学习和分享自己的见解，同时也感谢选择本书的读者，相信本书能够让你有所收获。

汪 鹏

2018 年 1 月

# 目 录

## 第一篇 Entity Framework 6.x 基础

第 1 章 Entity Framework 的发展.....	3
1.1 Entity Framework 版本介绍.....	3
1.2 Entity Framework 领域建模方式.....	4
1.2.1 Code First.....	4
1.2.2 Model First.....	5
1.2.3 Database First.....	5
1.3 使用 Entity Framework Code First 6.x 的原因.....	6
1.4 小结.....	6
第 2 章 数据库表的创建和迁移.....	7
2.1 数据库连接和初始化策略.....	7
2.1.1 数据库连接.....	7
2.1.2 数据库初始化策略.....	10
2.2 约定.....	11
2.2.1 类型发现.....	11
2.2.2 主键约定.....	12
2.2.3 关系约定.....	13
2.2.4 复杂类型约定.....	14
2.2.5 自定义约定.....	15
2.3 基于代码配置.....	22
2.4 配置属性映射和关系映射.....	24
2.4.1 属性映射.....	25
2.4.2 关系映射.....	42
2.4.3 私有化属性映射.....	56



2.5	继承映射策略 .....	58
2.5.1	Table per Hierarchy (TPH) .....	58
2.5.2	Table per Type (TPT) .....	59
2.5.3	Table per Concrete class (TPC) .....	62
2.5.4	Entity Splitting.....	65
2.5.5	Table Splitting.....	67
2.6	代码迁移 .....	69
2.6.1	迁移命令 .....	69
2.6.2	迁移的本质 .....	73
<b>第 3 章</b>	<b>实体状态和数据操作 .....</b>	<b>82</b>
3.1	实体状态 .....	82
3.2	数据操作 .....	84
3.2.1	数据查询 .....	85
3.2.2	数据保存 .....	102
3.2.3	自动生成存储过程 .....	112
3.3	避免陷阱 .....	115
3.3.1	导航属性查询 .....	115
3.3.2	分页查询 .....	116
3.3.3	语义可空 .....	117
3.3.4	表值函数 .....	119
3.3.5	日期操作 .....	121
3.4	基础篇实战 .....	122

## 第二篇 Entity Framework 6.x 进阶

<b>第 4 章</b>	<b>上下文管理和变更追踪原理 .....</b>	<b>137</b>
4.1	上下文生命周期管理 .....	137
4.1.1	基础模式追溯 .....	137
4.1.2	生命周期追溯 .....	139
4.1.3	进化模式追溯 .....	142
4.2	变更追踪原理 .....	144
4.2.1	快照式变更追踪 .....	144
4.2.2	代理式变更追踪 .....	146

第 5 章	变更追踪原理和日志管理.....	155
5.1	DetectChanges 原理.....	155
5.1.1	DetectChanges 的用途.....	155
5.1.2	调用 DetectChanges 方法的时机.....	157
5.1.3	关闭自动调用 DetectChanges 方法.....	159
5.1.4	遗忘的二进制属性和复杂类型.....	162
5.2	日志记录.....	165
5.2.1	ToString 打印.....	165
5.2.2	简单日志打印.....	166
5.2.3	结构化日志输出.....	169
5.2.4	构造块拦截.....	171
5.2.5	拦截器异常性能监控.....	175
5.2.6	拦截器解决 EF 6.1 忽略字符串空格历史遗留问题.....	187
5.3	事务.....	191
5.4	连接弹性.....	212
第 6 章	并发冲突.....	217
6.1	初级版解析.....	217
6.2	中级版解析.....	222
6.2.1	客户端获胜.....	223
6.2.2	数据库获胜.....	224
6.2.3	客户端和数据库合并获胜.....	225
6.3	高级版解析.....	226
6.3.1	简单重试策略.....	226
6.3.2	Polly 重试策略.....	227
第 7 章	性能优化实践.....	231
7.1	优化一：预编译视图.....	231
7.2	优化二：减少首次与数据库交互的代码.....	235
7.3	优化三：NGen 安装 Entity Framework 程序集.....	237
7.4	优化四：AsNoTracking.....	239
7.5	优化五：缓存.....	239
7.6	优化六：重新编译查询.....	245
7.7	优化七：避免“N+1” Select 查询.....	249

7.8	优化八：添加索引 .....	251
7.9	优化九：关闭回调 DetectChanges 方法 .....	255
7.10	优化十：使用异步查询 .....	257
<b>第 8 章</b>	<b>Entity Framework 6.x 实战 .....</b>	<b>258</b>
8.1	工作单元泛型仓储模式 .....	258
8.2	依赖注入泛型仓储模式 .....	278
<b>第三篇 Entity Framework Core 2.0 基础</b>		
<b>第 9 章</b>	<b>数据库表的创建和迁移 .....</b>	<b>295</b>
9.1	数据库初始化策略 .....	295
9.2	配置属性映射和关系映射 .....	298
9.2.1	属性映射 .....	300
9.2.2	Backing Fields .....	311
9.2.3	Alternate Keys .....	316
9.2.4	Shadow Property .....	317
9.2.5	Owned Entities .....	321
9.2.6	HasQueryFilter 修改 .....	322
9.2.7	关系映射 .....	327
9.2.8	继承映射 .....	346
9.3	代码迁移 .....	351
9.3.1	控制台程序迁移 .....	351
9.3.2	Web 应用程序迁移 .....	353
9.3.3	多个上下文迁移 .....	364
9.3.4	脚手架逆向迁移 .....	367
<b>第 10 章</b>	<b>数据操作和初始化数据 .....</b>	<b>371</b>
10.1	实体状态 .....	371
10.2	数据操作 .....	371
10.2.1	数据查询 .....	371
10.2.2	数据保存 .....	393
10.3	无连接实体 .....	398
10.4	初始化数据 .....	401

## 第四篇 Entity Framework Core 2.0 进阶

第 11 章	性能优化实践 .....	407
11.1	AsNoTracking .....	407
11.2	EF.Functions.Like .....	410
11.3	自定义标量函数 .....	412
11.4	显式编译查询 .....	416
11.5	上下文实例池 .....	419
第 12 章	查询实体元数据和模型数据验证 .....	424
12.1	查询元数据 .....	424
12.2	内置模型验证 .....	426
12.3	第三方扩展模型验证 .....	429
第 13 章	上下文实例创建方式 .....	432
13.1	显式创建上下文实例 .....	432
13.1.1	带有构造函数的 OnConfiguring 方法 .....	432
13.1.2	传递连接字符串参数到构造函数 .....	433
13.1.3	使用不带依赖注入的 DbContextOptions .....	433
13.2	依赖注入容器创建上下文实例 .....	434
13.2.1	带有无参构造函数的 DI .....	434
13.2.2	带 DbContextOptions 的 DI .....	434
13.3.3	使用泛型 DbContextOptions .....	435
13.3.4	使用 AddDbContext/AddDbContextPool .....	436
第 14 章	实现多租户 .....	437
第 15 章	捕获审计数据 .....	446
第 16 章	变更追踪策略和日志管理 .....	451
16.1	变更追踪策略 .....	451
16.1.1	INotifyPropertyChanged .....	453
16.1.2	INotifyPropertyChanging .....	457
16.2	日志管理 .....	458

16.2.1	添加控制台包输出 .....	458
16.2.2	自定义日志输出 .....	459
<b>第 17 章</b>	<b>xUnit 单元测试 .....</b>	<b>462</b>
17.1	控制台单元测试 .....	463
17.2	NET Core Web 应用程序单元测试 .....	465
<b>第 18 章</b>	<b>事务和并发冲突 .....</b>	<b>468</b>
18.1	事务 .....	468
18.2	并发冲突 .....	473
18.2.1	初级版解析 .....	474
18.2.2	中级版解析 .....	478
18.2.3	高级版解析 .....	482
<b>第 19 章</b>	<b>Entity Framework Core 2.1 待发布 .....</b>	<b>488</b>

# 第一篇

---

## Entity Framework 6.x 基础



# 第 1 章

## Entity Framework的发展

### 1.1 Entity Framework 版本介绍

Entity Framework 是一个 Object/Relational (对象映射) Mapping 实体框架, 简称为 ORM, 可以在 SQL Server、Oracle、DB2、MySQL 等数据库上使用。我们可以将数据作为业务对象和实体来进行处理, 开发者可以使用 LINQ 查询, 然后使用 C#面向对象语言来操作和检索数据。表 1-1 所示为 Entity Framework 的发展历程。

表 1-1 Entity Framework 的发展历程

ADO.NET Entity Framework 版本	.NET Framework 和 IDE 版本	特性介绍
6.2-beta1	.NET Framework 4.5+ and Visual Studio 2013+	通过第一次加载并持久化来减少 Code First 模型的启动时间 增强索引映射定义 解决 EF 工具更新模型 Bug 添加 DbFunctions.Like 方法并翻译为 SQL 中的 In 语句 使用命令行自动生成 SQL 脚本 其他
6.1.3	.NET Framework 4.5+ and Visual Studio 2013+	修复遗留 Bug 其他
6.1.2	.NET Framework 4.5+ and Visual Studio 2013+	修复遗留 Bug 其他
6.1.1	.NET Framework 4.5+ and Visual Studio 2013+	修复遗留 Bug 其他
6.1.0	.NET Framework 4.5+ and Visual Studio 2013+	处理事务提交失败 合并工具 Code First 注释 增加索引特性 公共映射 API



(续表)

ADO.NET Entity Framework 版本	.NET Framework 和 IDE 版本	特性介绍
6.0.0	.NET Framework 4.5 and Visual Studio 2013	异步查询和保存 基于代码配置 依赖解析 监控 SQL 日志 改善连接管理 支持事务
5.0	.NET Framework 4.5 and Visual Studio 2012	在 Code First 模型和 EF 设计器中支持枚举 支持表值函数 支持每个模型多个图
4.3	.NET Framework 4.0 and Visual Studio 2010	Code First 迁移 自动迁移
4.2	.NET Framework 4.0 and Visual Studio 2010	1.4.1 版本遗留 Bug 修复
4.1	.NET Framework 4.0 and Visual Studio 2010	支持 Code First 模型 介绍 DbContext API 支持 Data Annotations 和 Fluent API Validation
4.0	.NET Framework 4.0 and Visual Studio 2010	支持 Model First 模型 支持 POCO 延迟加载 T4 模板代码生成
3.5	.NET Framework 3.5 SP1 and Visual Studio 2008 SP1	提供基础 ORM 映射, 支持 Database First 模型

## 1.2 Entity Framework 领域建模方式

一个业务领域由各个实体和各个相互关联且有各自的属性和行为的实体组成, 每个实体都有其状态和验证规则需要维护, Entity Framework 实体框架设计的出现就是为了允许开发人员着重关注业务领域, 开发人员就实体来建模。它产生的目的是为了了解企业快速开发和迭代出市场所需要的系统或者软件。下面我们介绍 Entity Framework 中的三种领域建模方式。

### 1.2.1 Code First

Code First 可以通过 C# 或者 VB.NET 来描述这些模型, 然后通过类来创建数据库, 这些类简称为 POCO (Plain Old CLR Object)。POCO 来源于 Java 中的 POJO, 其中的 J 为 Java, POJO 是由马丁·福勒 (Martin Fowler) 和其他人一同提出的概念, 以反对在 20 世纪 20 年代早期受