中国科学出版集团
新世纪书局

IBM PRESS

IBM最佳实践

Java企业级持久化技术指南

Persistence in the Enterprise
A Guide to Persistence Technologies

[美] Roland Barcia, Geoffrey Hambrick, Kyle Brown,
Robert Peterson, Kulvir Singh Bhogal 著
叶斌 译

面向开发者和架构师的Java持久层技术权威指南

★ 与IBM五位经验丰富的架构师面对面交流
★ 从理论到实践深入学习JDBC、iBATIS、Hibernate Core、OpenJPA和pureQuery持久化技术
★ 掌握过硬而实用的Java技术，成为大软件企业所需的技术专才

PEARSON Education

developerWorks®

科学出版社

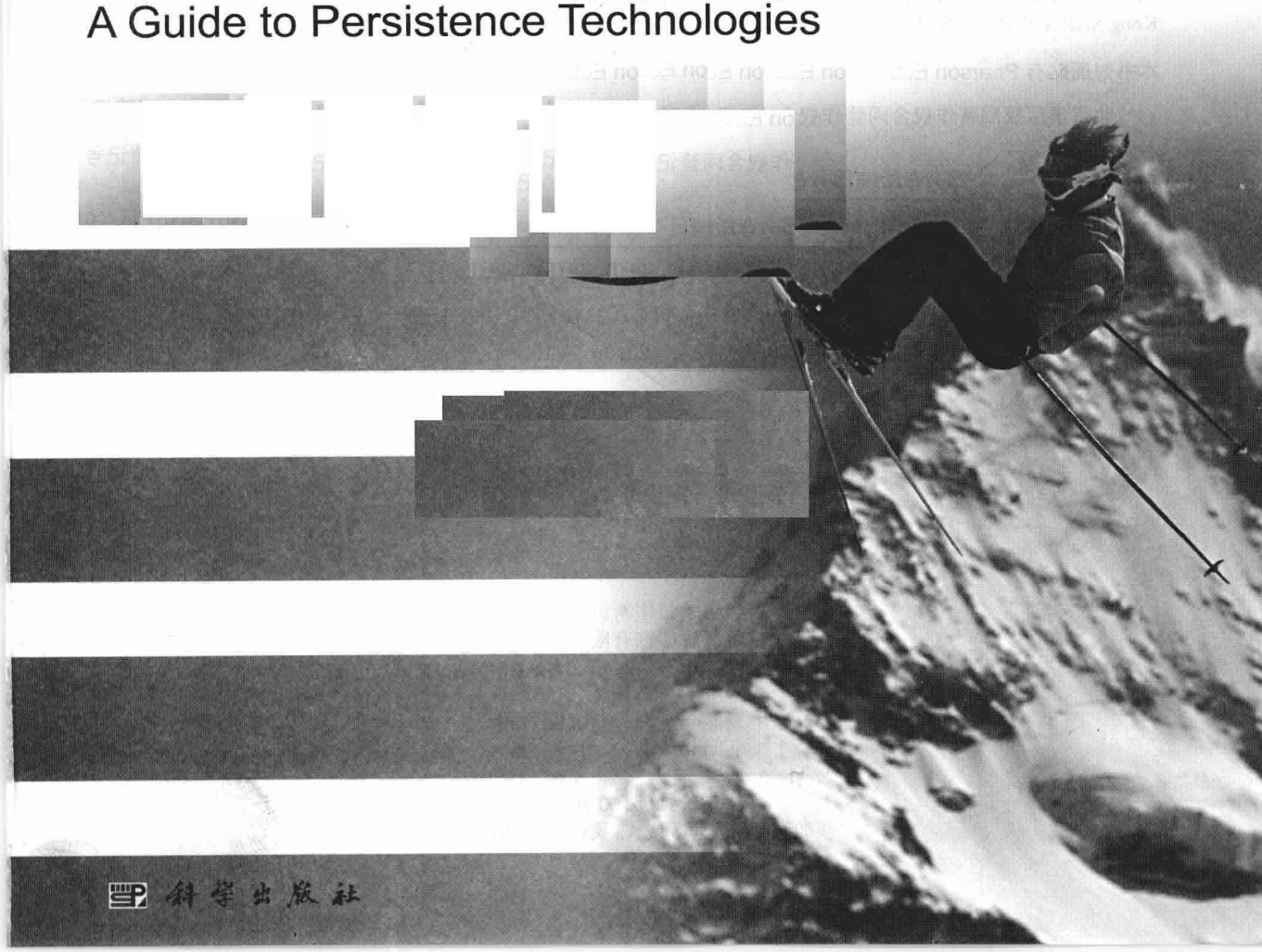
IBM PRESS

IBM最佳实践

Java企业级持久化技术指南

Persistence in the Enterprise

A Guide to Persistence Technologies



科学出版社

内 容 简 介

本书由 IBM 的 5 位资深专家合著而成，全书分为两部分。第 1 部为第 1~4 章，第 1~3 章分别讲述了持久化技术的发展历史、高层需求对选择持久化技术的影响以及如何根据企业应用的需求来设计持久数据的模式，第 4 章介绍了一种比较不同持久化技术的方法并给出一份调查问卷。第 2 部分为第 5~10 章，从技术背景、架构概况、编程模型、ORM 功能支持、调优选项和公共范例开发 6 方面详细地介绍了 JDBC、iBATIS、Hibernate、OpenJPA 和 pureQuery 这 5 种持久性框架，并就实践过程中可能遇到的问题给出各种建议，以供读者参考。

本书内容丰富，紧跟技术前进的步伐，非常适合 Java 开发人员学习使用，同时也是项目经理、系统架构师以及测试人员深入学习 Java 企业开发知识的必备参考书。

著作权声明

图书在版编目（CIP）数据

IBM 最佳实践：Java 企业级持久化技术指南/(美)巴西亚(Barcia, R.)等编著；叶斌译. —北京：科学出版社，2010.6

ISBN 978-7-03-027885-2

I. ①I…　II. ①巴…②叶…　III. ①JAVA 语言—程序设计　IV. ①TP312

中国版本图书馆 CIP 数据核字（2010）第 107625 号

责任编辑：赵东升　王海霞/ 责任校对：杨慧芳

责任印刷：新世纪书局　/ 封面设计：周智博

科学出版社 出版

北京东黄城根北街 16 号

邮政编码：100717

http://www.sciencep.com

中国科学出版集团新世纪书局策划

北京市艺辉印刷有限公司印刷

中国科学出版集团新世纪书局发行　各地新华书店经销

*

2010 年 7 月 第 一 版　开本：16 开

2010 年 7 月第一次印刷　印张：25.5

印数：1—3 000　字数：620 000

定价：55.00 元

（如有印装质量问题，我社负责调换）

关于作者

Roland Barcia 是一名高级技术研究人员 (Senior Technical Staff Member, STSM)，并且是 IBM Software Service for WebSphere 的 Web 2.0 首席架构师。他曾与人合著 *IBM WebSphere: Deployment and Advanced Configuration* 一书，并已发表了 40 多篇关于 Java™ 持久性、Ajax、REST、JavaServer Faces 以及消息传递技术等方面的文章和论文，经常出现在各种技术会议上，为客户介绍各种技术。他在各种平台的中间件系统实施方面积累了 10 年的经验，其中包括 Socket, CORBA, Java EE, SOA，以及最新的基于 Web 2.0 的 Project Zero 平台。Roland 拥有 New Jersey Institute of Technology 的计算机科学硕士学位，有一个名为 Web 2.0 and Middleware 的博客 (http://www.ibm.com/developerworks/blogs/page/barcia)。

Geoffrey M. Hambrick 是 IBM Software Services for WebSphere Enablement Team 的一位杰出工程师，其任务是帮助开发和传播使用 IBM WebSphere 运行时、工具和技术的最佳实践。Geoff 一直是分布式对象技术领域的先驱者，并且参与制定各种标准，例如 Object Management Group CORBA Object Services 和 Enterprise JavaBeans 规范。Geoff 有众多的工作客户，而且经常受邀出席世界各地的会议。他是 *IBM developerWorks*® 专栏 The EJB Advocate 的作者，该专栏描述各种使用 EJB 技术，特别是实体 bean 组件的最佳实践模式。Geoff 目前的重点是可用来自动化最佳实践应用的模式制作工具 (pattern authoring tools)，他和 Chris Gerken 发明了设计模式工具箱 (Design Pattern Toolkit)，该工具箱扩展了 Eclipse Java Emitter Templates 标准，并且有助于使基于模式的工程 (Pattern Based Engineering) 成为现实。

Kyle Brown 是 IBM Software Services and Support 的一位杰出工程师，他曾与人合著过几本书，其中包括 *Enterprise Java Programming with IBM WebSphere* 和 *Enterprise Integration Patterns*。他是模式方面的知名专家，而且是 PLoP (Pattern Languages of Programs) 会议的前任主席。Kyle 参与编写了首批对象关系映射模式论文之一——*Crossing Chasms*，该篇论文收录在 *Pattern Languages of Program Design 2* 一书中。在日常工作中，Kyle 帮助 IBM 客户采用新兴技术，并传授使用 IBM WebSphere 系列产品的最佳做法。

Robert R. Peterson 是 IBM Software Service for WebSphere 的一位高级管理顾问 (Senior Managing Consultant)，他往返于世界各地，为未来的 IBM 软件系统实施战略性的概念验证项目。他出版了大量的技术书籍和论文，经常在各种会议上发表演讲，拥有多项企业系统的美国专利。您可以访问他的网站 http://juzzam.org/PersonalSite/。

Kulvir Singh Bhogal 是 IBM Software Service for WebSphere 的一名高级管理顾问，在全美各地的客户网站上制定和实现以 WebSphere 为中心的 SOA 解决方案。他在众多技术领域中总共申请了 100 多项专利。Kulvir 为许多出版刊物撰写文章，包括 *JavaPro Magazine*, *IBM developerWorks*, *O'Reilly Media*, *Java Developer's Journal*, *DevX*, *InformIT* 以及 *WebSphere Advisor Magazine* 等。他还经常主持众多技术会议。

致 谢

写作小组：

任何技术书籍的背后都有能够大大提高图书质量的审阅者，本书也不例外。我们要感谢 Eberhard Wolff，Jens Coldewey，Scott Ambler，Keys Botzum 和 Brandon Smith，他们对书中的各章内容进行了审查。此外，我们还要感谢其他几位，他们对我们评估的技术提供了细节或者回答具体问题，我们要给这些人献上无尽的谢意：Kevin Sutter，Tom Alcott，Steve Brodsky，Timo Salo，Jim Knutson，Randy Schnier，Daniel Lee，Jack Woodson，Gang Chen，Billy Newport，Steve Ims，Hibernate 开发团队，以及再次提到的 Keys Botzum，Brandon Smith，Scott Ambler 和 Eberhard Wolff。最后，我们要特别感谢 IBM Press 出版社和 Pearson 的工作人员，他们对我们的繁忙日程以及经常性的延迟交稿表现出了极大的耐心，我们要感谢 Kevin Howard，Katherine Bull，Lori Lyons，Cheri Clark 和 Gloria Schurick，他们所做的一切工作最终使本书得以出版。

Roland Barcia：

所有的赞美、福祉和荣耀都归于天父和我主耶稣基督！我要感谢我的妻子 Blanca，她也是我最好的朋友，我全心全意地爱着她。写书用去了太多与家人一起的时间，因为这一点，她常常要担负起全家的生活重任。感谢你们，Savannah，Alyssa，Joseph 和 Amadeus，你们是多么棒的孩子！感谢鼓励我和爱我的父亲母亲，Rolando 和 Maria Barcia。我要特别感谢我的岳母也是我的朋友 Nery Leon，在著书期间多次照看孩子。我要感谢与我一起合著本书的其他作者，感谢 Kyle 和 Geoff 指导并帮助我实现了许多职业目标。感谢 Robert，你是一个导师能够拥有的最好门生，感谢 Kulvir 带我到 Baby Acapulcos 吃 fajitas，并成为我的朋友。感谢几位在前进的道路上帮助过我的主管和领导，特别是 Albert Scardino，Hicham Badawi，Peter Bahrs，Rachel Reinitz 和 Tom Kristek。最后，我还要感谢其他的许多朋友和家人，他们为我祈祷，支持我，并成为我的朋友：Pastor Gary Stefanski、Mchael Kenna，Alain Ratsimbazafy，Gang Chen，Avelino Barcia（Tio Cabezon），Nancy Quevedo（Tia），Javier 和 Miaria Leon，Amelia Jorge（Mami），我在 Fairview 福音教会的兄弟姐妹们，以及我的妹妹 Adriana Cantelmo，我在上一本书中忘了提及她，于是她让我至今都记得此事。

Geoffrey Hambrick：

我要感谢我的妻子 Cindy、两个儿子 Austin 和 Alex、女儿 Lesli 和他的丈夫 Tod，以及我的两

个孙子 Ella 和 Clara。多年来，他们放弃了与我在一起的时间，让我一直忙于一个又一个写作项目，所幸的是，我的写书项目就要结束，一切就要恢复原样。我要感谢我的爸爸妈妈，James O. 和 Elizabeth A Hambrick，他们做得很出色，养育了六个孩子，而且有办法使这“六个中的第四个”自认不同并坚信只要用心去做，就能够做成任何事情。我还要感谢 Robert Peterson，是他出主意要把我们内部的比较持久性机制的白皮书变成一本书——总有一天我会原谅他的（说笑而已！）。我要感谢 Roland Barcia，实际上是他一直持着该项目的接力棒直到项目结束。最后，我想特别感谢 Tom Kristek，.作为一位超过 15 年的朋友和导师——其坦率的态度和关注问题的重要细节的敏锐能力，真正地使我成为了今天的一个行家。WATFT！

Kyle Brown：

我要感谢我出色的妻子 Ann 和我的儿子 Nathaniel 在这一项目中自始至终给予我的支持和爱。我还要感谢下面这些人，我向他们学习并与他们一起探讨对象持久性方面的知识，他们是：Ken Auer 和 Sam Adams，Bruce Whitenack，Wayne Beaton，Justin Hill，Timo Salo，Art Jolin，Mark Weitzel，Rachel Reinitz，Martin Fowler，尤其是 Scott Rich，除了在 EJB 的早期历史方面我们做了很多探讨之外，他还为本书中的 VisualAge® Persistence 的早期情况提供了极大的帮助和资料来源。我还要感谢 IBM Software Group Architecture Board 的成员，他们仔细审查了该文件的早期版本，并就审查结果做了一些热烈的讨论。最后，我要感谢我们出色的复审人，特别是 Scott Ambler 和 Eberhard Wolff，以及其他任何我们忘记了在撰写本书的过程中曾向其索求过资料、帮助或者协助的人。

Robert Peterson：

我要感谢我的父亲，在那些难挨的墨西哥童年时期，他耐心地教我如何用英语阅读和书写，还要感谢我母亲那充满慈爱的支持和安慰。我要感谢我的合著者，特别是 Roland，我的导师，我所有的成功都归于他——包括这本书。特别要感谢我的主管，Mike Morin，我花费了无数时间与我的合著者一起仔细彻底地打造这一项目直到最后开花结果，这期间他完全地支持我。

Kulvir Singh Bhogal：

我要感谢上帝给予我的所有恩典，没有他，我一无可取。我感谢我心爱的妻子 Meeti，她的微笑总是能够让我眉头舒展。我还要感谢我的父母 Kirpal 和 Manjit，他们把我带到这个世界，并耐心地教导我如何在这个世上生活。我还要感谢我的妹妹 Pummi 和我的侄女 Sajal，谢谢他们的爱和支持。Roland、Robert、Kyle 和 Geoff，感谢你们容忍我繁忙的咨询日程安排，并给了我与你们一起编撰本书的机会。最后，我要感谢 Andy Sweet，你是一个优秀的、给予我支持的主管和好朋友。

译者序

两年前，我有幸与一位毕业不久的年轻同事一起工作，我们在同一个项目组中分别负责同一个产品的不同模块，这两个模块共同用到了一些需要写到数据库中的数据，我们打算用Hibernate来实现数据的持久功能。一天，我正准备就这些数据的数据库模式和持久方式与这位同事详细地讨论一番，没想到他却不以为然："这样的做法太腐朽了！"我有些诧异于他的反应，不过我明白，这里他用"腐朽"一词的意思是说，把很多时间和精力用在数据库模式的设计和持久方式的考虑上，在现在来说，已经是一种过时的做法了。

显然，事实并非如此，可是，是什么原因能够让一个对软件的设计和开发抱有极大热忱的年轻人有如此的想法呢？我想，本书中的一句话可以回答这个问题，这句话是这样说的：不学习历史的人注定要重复历史。

在这个问题上，本书可以告诉我们，我的这位同事至少步入了两个误区：一是认为，依赖Hibernate自身就能够解决所有的数据库模式和领域模型之间的阻抗失配问题；二是认为，随着持久性机制的成熟，软件的设计者和开发者无需再花费更多的时间来处理数据持久和数据库模式方面的问题了。

那么，在现代的软件设计和开发过程中，我们在数据的持久性方面应该持什么样的一种观念呢？本书不仅就这一问题作出了解答，而且传授了选择持久性机制和实现数据持久过程的最佳做法。

本书由IBM的5位专家共同撰写，他们都是IBM Software Services for WebSphere (ISSW)部门的成员，有着丰富的技术实践经验，他们把多年来在工作中的心得体会汇集在一起，使得本书拥有了自己独特的视角。他们在提出问题时，从自己工作中的实际出发，分析问题对企业应用的影响，在选择问题的解决方法时，则详细说明如何根据实际的情况来权衡，然后作出取舍，而且还针对读者可能会作出的决定，提出了自己的忠告。所以，由于作者本身所从事的工作的关系，本书在实践方面具有很强的指导性和针对性，这也是作者的本意所在。

另一方面，正是得益于作者的工作经验和对持久性技术的洞悉观察，本书内容丰富翔实，采用了一种端到端的视角，从架构师的观点出发，以面向对象的Java应用风格和关系型数据库作为论述的基准点，并以合理的布局和循序渐进的方式，全面详细地论述了企业级应用中的持久性问题。本书分为两大部分，第一部分从持久性技术的历史、应用的持久性需求和设计以及持久性机制的比较方法等方面进行了讨论，第二部分则针对几种常见的持久性机制，全面地分析了这些持久性机制的编程模型和对ORM功能的支持，并通过对公共范例的实现来比较不同机制之间的做法。

我们可以简单地把本书的第一部分看作是理论部分，而把第二部分看作是实践部分。第一部分分为4章，第1章"对象关系映射简史"讲述了持久性机制的起因，及其从无到有，从简单到复杂，从开始的多种概念到目前的集大家之成的演变过程，并指出了未来的发展方向。第2

章“高层需求和持久性”以如何获取高层需求这一问题的答案作为框架，针对持久性方面的需求作出了详细的论述。第 3 章“设计持久对象服务”论述了用于理解持久性机制的概念和建模方法，并说明了可以通过哪些方式来解决面向对象的领域模型和关系数据库模式之间的阻抗失配问题。这一章对于读者来说特别重要，因为通过全面介绍解决企业级数据持久这一问题需要完成的全部工作，读者可以意识到，在解决这一问题的过程中，应用软件的设计者和持久性机制分别担当着构建模型和实现模型的责任，只有通过设计者和持久性机制之间的配合，才能很好地完成这一工作。第 4 章“如何评估可选项”阐述了一种评估持久性机制的方法，并给出了一份可以作为项目持久性需求的起点的调查问卷。

本书第二部分从第 5 章到第 9 章，分别详细地论述了 JDBC，iBATIS，Hibernate Core，OpenJPA 和 pureQuery 这 5 个持久性机制的体系结构、编程模型、支持的 ORM 功能以及最佳实践等。其中，JDBC 并不是真正意义上的持久性框架，而是其他的持久性框架的基础；iBATIS 是一个表入口框架（Table Gateway Framework），使用 SQL，但分离 SQL 与代码；Hibernate 是目前最受欢迎的持久性框架，功能齐全且强大，是一个全域映射器；OpenJPA 是一个汇集了 TopLink，Hibernate 和 Java Data Objects API 等几家的最好想法的新的持久性体系结构，也是一个全域映射器；pureQuery 是一个商业性解决方案，与 iBATIS 一样，是一个表入口框架，使用 SQL，但既可以在代码中直接使用 SQL，也可以把 SQL 从代码中分离出来。pureQuery 作为 JDBC 之上很薄的一层，可以用作其他持久性机制的底层，例如现在正用作 Project Zero 的持久引擎。

第二部分的最后一章，也是本书的第 10 章“将理论付诸实践”以汇总表格的方式，横向分项比较了前面 5 章中讨论的 5 个持久性机制，然后作为本书最后要讨论的问题，也是总结性的发言，作者就读者可能会作出的决定，给出了自己的忠告。

无论你是一个刚入行的毕业生，还是一个有了多年经验的开发者和设计者，本书都值得一读再读。因为本书既可以看作是 5 种持久性机制的快速入门手册，也可以看作是持久性机制的比较方法的介绍，它还提供了持久性需求的调查问卷，你可以直接使用这一调查问卷，也可以改造它直到适合自己项目的需要，甚至还可以使用本书来指导自己捕捉和完善企业应用中的需求。如果你需要了解 ORM 映射方法的话，本书也提供了详细的说明。

这是一本信息技术方面的书籍，因此书中的叙述方式有着明显的行业风格，并且使用了大量的行业术语或者专门说法，有些术语或者专门说法在中文方面还没有正式的或者一致的译法。在这方面，译者通过互联网尽量查阅了其他书籍或者文章的翻译方式，并选择了最常用的译法，但是由于译者水平有限，译文中难免存在不当之处，恳请读者不吝指正。

叶斌

2009 年 7 月

前 言

为什么你应该偷走这本书——借用Abbie Hoffman的名著

智者常说，迈向智慧的第一步是知晓我们一无所知——甚至不懂得如何恰当地提问。这意味着我们必须留心倾听那些有经验的人：思考他们的问题和答案中更深层的含义。如此到最后，一种新的经验可能会推翻很多东西，那些每个人都曾经认为是正确的东西。

这种看法不仅适用于在智慧之颠思考深奥如哲学这样的课题的智者圣人，而且适用于我们这些需要了解某一个非常技术化主题的人，比如在企业级 Java 应用中提供持久性数据这样的主题。

想象一下，如果要求你为公司一个新的 Java 应用软件选择持久性机制，然后与公司的各利益相关方一起审查你的架构方案。在审查开始时，你将不得不准备回答一些被普遍关心的问题：

- 对哪些持久性机制进行了评估？为什么选择它们？
- 在这些经过评估的持久性机制中，你为我们选择了哪一个？为什么？

接着，根据受邀来参加审查的各利益相关方所对应的角色，你需要深入地回答具体的问题，这些问题涉及了与每位审查者相关的被选定架构的细节，以及每项需要完成的工作等。例如：

- **代表管理者和经营者的角色**，评估向新技术过渡需要在时间、金钱及其他一些资源方面的开销。他们想知道这一类问题的答案，例如哪些供应商拥有某一特定技术的市场、采用的许可协议种类（以及与之相随的限制）、技术的实用性，以及给定机制的成功案例的参考等。
- **代表分析师和架构师的角色**，需要评估给定的机制是否满足**最终用户**和**运营商**所期待的应用软件在业务和信息技术等方面的需求，他们是最终用户和运营商的倡导者。他们关心功能性方面的问题，例如，该机制是否支持关联关系、各种各样的属性类型、触发器和约束等。他们同样关心该机制是否可以达到应用的响应时间、吞吐量和其他“非功能性”目标的要求。
- **作为开发者和测试者的角色**，就代码编写方面而言，在实现对象到关系数据库层的映射服务的时候，他们最关心框架和 API 复杂性。他们想知道如何处理特定的编码工作，例如创建连接，以及依据代表用例的详细数据库设计来读取/更新数据等。

实际上，任何架构设计与其说是一门科学，还不如说是一门艺术，这就使得做好这项工作变得非常具有挑战性。它不仅要求对方法的深刻体会和以往奏效的最佳实践，而且需要一种能

够意识到何时去尝试一些创新的良好直觉。

对网页进行关于“Java 和关系型数据库”的搜索，会返回许多有帮助的文章和书籍的链接，但其中大多数是针对某一特定的机制，例如 Java 数据库连接（JDBC）、Enterprise JavaBeans（EJB）、Hibernate，或最近出现的 Java Persistence API(JPA)。这些资料有些会细谈如何设计数据库，其他则主要是指导如何使用 API 来创建一个示例应用程序，但没有资料是从端到端（end-to-end）应用架构的视角来进行阐述的，这种视角会帮助你理解在为关系数据存储选择持久性机制时所涉及的问题，然后帮助你作出选择。

端到端视角之所以重要是因为，如上所述，好的架构师知道各种问题的答案，这些问题由各个利益相关者的角色代表在审查过程中提出。比较有经验的架构师则预测问题，并首先使用这些问题来驱动他们的设计理念。但最好的架构师则依照烹调手册的格式来文档化这些问题和答案，这样可以指导应用开发项目的每一阶段，包括分析、设计、构建、测试、部署、运营和维护。以可重用的方式文档化问题的答案不仅提高了应用程序的质量，而且加速了开发进程——因为团队由于摸索和出错而花在“重建轮子”上的时间大幅度地减少。

正是端到端视角使得这本书与众不同。我们都是 IBM Software Services for WebSphere （ISSW）团队的顾问，我们的工作是帮助客户充分利用我们的产品，包括 IBM WebSphere Application Server，WebSphere Portal Server，WebSphere Commerce Server 和 WebSphere Process Server 等。我们经常参与技术的论证，并与客户方的架构师进行那些趋于白热化的论战。这些客户方的架构师将会作出自行开发还是购买的决定，或是对供应商和技术做出选择。所以，在谈论到这一更广泛的以应用为中心的视角，并为棘手的问题寻求答案的时候，我们是专家。

本书着眼于持久性框架，这些框架大部分是基于 Java 的，也因此你能够提出一个令所有利益相关者——从你的 CTO 到你的架构师同事，到开发者、测试人员和运营团队都满意的解决方案。如果你担任这些特定角色中的某一个并参与方案审查，本书将帮助你提出各种正确的问题并且能够理解给出的答案。

本书不同于其他书的另一方面是，我们尽力保留了我们独特的基于辅导的在职咨询方法。简而言之，我们希望同时“授人以鱼”和“授人以渔”。一个 ISSW 合约的理想化结果是，我们的产品帮助你取得早期的成功，但在将来你会变得自给自足。

虽然一本书绝不会取代在工程中直接与你进行工作接洽的那个活生生的顾问，我们还是尝试着非正式地把本书分成两部分：

- **第 1 部分，“持久性问题”**，教你捕鱼，让你可以“吃一辈子”。具体来说，该部分的内容帮助你了解在为关系数据存储选择一个 Java 持久性机制时，会遇到的争议和需要权衡的方面。根据前面所述的各个利益相关者角色代表提出的问题，这些争议被整理编写到各章中。具体来说，前 3 章涵盖的议题为：第 1 章提供了相关的持久性机制的简要历史；第 2 章涵盖了商业驱动因素和相关的 IT 需求；第 3 章讨论了与对象关系映射相关的实现问题。第 4 章是该部分的结束章，从前 3 章所讨论

的问题和折中办法中提炼出了一份调查问卷，这样就可以使用一种统一的“尺度”和基于最佳实践的方法来评估每一种机制。

- **第 2 部分，“同型比较”**，给你一些鱼，这样就可以“今天有得吃”。每一章都使用在第 4 章中创建的方法和调查问卷来为 5 种流行的机制收集数据。在该部分中，作为第 1 章罗列的各种 Java 持久性方法的代表性例子，我们探讨了 Java Database Connectivity、iBATIS、Hibernate、Java Persistence API 和 pureQuery。在第 10 章的小结部分，我们结束了这一部分内容并总结全书，比较了这些机制并分别列举了关于这种种机制的最佳应用的案例。

有这样一种说法，“事情改变得越多，它们越发一成不变。”但是它们仍然在变化。在第 4 章“如何评估可选项”中设计的调查问卷对本书的寿命来说至关重要，因为学习了第 4 章后，当新的持久性框架和 API 可用的时候，应用同样的标准来衡量就相对容易了。所以，为了进一步提高本书的价值，我们把与评估有关的调查问卷和代码例子的下载站点纳入作为参考，当你对这些或者其他机制进行评估的时候，你可以照它们原来的样子使用这些调查问卷和代码例子，也可以为了在你自己公司使用而对它们进行定制。

因为这是一本 developerWorks 图书，于是我们使用了一些被称为“developerWorks 链接”的特殊引用，它们出现在书页边的空白处和所属章的最后一节中，并使用一个特别图标来标识。这些引用能够链接到 IBM developerWorks 网站 www.ibm.com/developerWorks/。我们发现这是一个非常令人兴奋的功能，因为当你读这本书的时候，你可以跟进网站上相应的地址，并能即时访问这些或其他相关的文章和在线书籍。这样做的最终结果是把本书扩展到了万维网上，进一步提高了它的使用寿命。

在每章的结尾处我们还加入了“参考资料”小节，列出了整章中引用的资源，这些资源通过在正文中添加“[作者]”这一标志方式加以引用。

我们希望你赞同一点，那就是这本书值得一“偷”。但请把它带到最近的结账柜台，或者现在就单击“放入购物车”按钮。如果这是朋友的藏书，请把它放回到你朋友的书桌或书架上，然后给自己买一本——在在线的零售网站上和出售有关 Java 或数据库技术的技术性书籍的传统实体书店中都可以购买到这本书。

目 录

第 1 部分 持久性的一个问题

第 2 部分 同型比较