

<<精通Hibernate>>

图书基本信息

书名：<<精通Hibernate>>

13位ISBN编号：9787121093739

10位ISBN编号：7121093731

出版时间：2010-2

出版时间：孙卫琴 电子工业出版社 (2010-02出版)

作者：孙卫琴

页数：716

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## 前言

在如今的企业级应用开发环境中，面向对象的开发方法已成为主流。

众所周知，对象只能存在于内存中，而内存不能永久保存数据。

如果要永久保存对象的状态，需要进行对象的持久化，即把对象存储到专门的数据存储库中。

目前，关系数据库仍然是使用最广泛的数据存储库。

关系数据库中存放的是关系数据，它是非面向对象的。

对象和关系数据其实是业务实体的两种表现形式。

业务实体在内存中表现为对象，在数据库中表现为关系数据。

内存中的对象之间存在关联和继承关系，而在数据库中，关系数据无法直接表达多对多关联和继承关系。

因此，把对象持久化到关系数据库中，需要进行对象-关系的映射（Object/Relation Mapping，简称ORM），这是一项烦琐耗时的工作。

在实际应用中，除了需要把内存中的对象持久化到数据库外，还需要把数据库中的关系数据再重新加载到内存中，以满足用户查询业务数据的需求。

频繁地访问数据库，会对应用的性能造成很大影响。

为了降低访问数据库的频率，可以把需要经常被访问的业务数据存放在缓存中，并且通过特定的机制来保证缓存中的数据与数据库中的数据同步。

在Java领域，可以直接通过JDBC编程来访问数据库。

JDBC可以说是访问关系数据库的最原始、最直接的方法。

这种方式的优点是运行效率高，缺点是在Java程序代码中嵌入大量SQL语句，使得项目难以维护。

在开发企业级应用时，可以通过JDBC编程来开发单独的持久化层，把数据库访问操作封装起来，提供简洁的API，供业务层统一调用。

但是，如果关系数据模型非常复杂，那么直接通过JDBC编程来实现持久化层就需要有专业的知识。

对于企业应用的开发人员，花费大量时间从头开发自己的持久化层不是很可行的。

幸运的是，目前在持久化层已经有好多种现成的持久化中间件可供选用，有些是商业性的，

如TopLink；有些是非商业性的，如JDO和Hibernate。

Hibernate是一个基于Java的开放源代码的持久化中间件，它对JDBC做了轻量级封装，不仅提供ORM映射服务，还提供数据查询和数据缓存功能，Java开发人员可以方便地通过Hibernate API来操纵数据库。现在，越来越多的Java开发人员把Hibernate作为企业应用和关系数据库之间的中间件，以节省和对对象持久化有关的30%的JDBC编程工作量。

2005年，Hibernate作为优秀的类库和组件，荣获了第15届Jolt大奖。

Hibernate之所以能够流行，归功于它的以下优势。

（1）它是开放源代码的，允许开发人员在需要的时候研究源代码，改写源代码，定制客户化功能。

（2）具有详细的参考文档。

（3）对JDBC仅做了轻量级封装，若有必要的话，用户还可以绕过Hibernate，直接访问JDBC API。

（4）具有可扩展性。

（5）使用方便，容易上手。

（6）Hibernate既适用于独立的Java程序，也适用于Java Web应用，而且还可在J2EE架构中取代CMP（Container-managed Persistence，由容器管理持久化），完成对象持久化的重任，Hibernate能集成到会话EJB和基于BMP的实体EJB中，BMP（Bean-managed Persistence）是指由实体EJB本身管理持久化。

（7）Hibernate可以和多种Web服务器、应用服务器良好集成，并且支持几乎所有流行的数据库服务器。

本书结合大量典型的实例，详细介绍运用目前最成熟的Hibernate 3.3版本进行Java对象持久化的技术。Hibernate是连接Java对象模型和关系数据模型的桥梁，通过本书，读者不仅能掌握用Hibernate工具对这两种模型进行映射的技术，还能获得设计与开发Java对象模型和关系数据模型的先进经验。

## &lt;&lt;精通Hibernate&gt;&gt;

本书的组织结构和主要内容本书按照由浅入深、前后照应的顺序来安排内容，主要包含以下内容。

1. Hibernate入门（第1章、第2章和第3章）第1章、第2章和第3章为入门篇。  
第1章和第2章概要介绍和Java对象持久化相关的各种技术，详细阐述中间件、Java对象的持久化、持久化层、数据访问细节、ORM、域模型和关系数据模型等概念。  
第3章以一个Hibernate应用实例—helloapp应用为例，引导读者把握设计、开发和部署Hibernate应用的整体流程，理解Hibernate在分层的软件结构中所处的位置。  
对于已经在Java对象持久化领域有一定工作经验的开发人员，可以从第1章和第2章入手，高屋建瓴地把握持久化领域的各种理论，对于新手，不妨先阅读第3章，以便快速获得开发Hibernate应用的实际经验。
2. Hibernate工具（第4章和附录C）第4章和附录C介绍Hibernate的一些代码转换工具的用法，例如，hbm2java工具能根据映射文件自动生成Java源文件，hbm2ddl功能能根据映射文件自动生成数据库Schema。
3. 对象-关系映射技术（第5、6、7、10、11、12、13、14和15章）本书重点介绍的内容之一就是如何运用Hibernate工具，把对象模型映射到关系数据模型，相关章节包括：第5章：介绍对象-关系映射的基础知识。  
第6章：介绍对象标识符的映射方法。  
第7章：介绍一对多关联关系的映射方法。  
第10章：介绍组成关系的映射方法。  
第11章：介绍Java类型、SQL类型和Hibernate映射类型之间的对应关系。  
第12章：介绍继承关系的映射方法。  
第13章：介绍Java集合类的用法，这一章主要是为第14章做铺垫的。  
第14章：介绍Java集合的映射方法。  
第15章：介绍一对一和多对多关联关系的映射方法。
4. 通过Hibernate API操纵数据库（第8、9和23章）第8章介绍运用Hibernate API来保存、更新、删除、加载或查询Java对象的方法，并介绍Java对象在持久化层的4种状态：临时状态、持久化状态、游离状态和删除状态。  
深入理解Java对象的4种状态及状态转化机制，是编写健壮的Hibernate应用程序的必要条件。  
第9章介绍Hibernate与触发器协同工作的技巧、拦截器（Interceptor）的用法，以及扩展Hibernate的事件监听器的方法。  
此外，还介绍Hibernate提供的批量处理数据的各种方法。  
第23章介绍Session的生命周期的管理方式，以及会话的实现方式。  
这一章的内容将帮助读者简化Hibernate应用的程序代码，并且为应用设计合理的软件架构。
5. Hibernate的检索策略和检索方式（第16、17和18章）第16章介绍Hibernate的各种检索策略，对每一种检索策略，都介绍它的适用场合。  
第17章和第18章介绍HQL查询语句的语法，以及QBC API的使用方法。  
合理运用Hibernate的检索策略及检索技巧，是提高Hibernate应用性能的重要手段。
6. 数据库事务、并发、缓存与性能优化（第20、21和22章）第20章先介绍数据库事务的概念，接着介绍运用Hibernate API和JTA API来声明事务边界的方法。  
第21章介绍在并发环境中出现的各种并发问题，然后介绍采用Hibernate的悲观锁，以及版本控制功能来避免并发问题的方法。  
第22章介绍Hibernate的二级缓存机制，并介绍如何根据实际需要来配置Hibernate的第二级缓存，以提高应用的性能。
7. Hibernate高级配置（第19章）第19章主要介绍Hibernate应用的两种运行环境：受管理环境与不受管理环境，然后介绍在这两种环境中配置数据库连接池、SessionFactory实例及事务的方法。
8. 综合实例（第24、25章）第24章和第25章介绍一个名为netstore应用的电子商务网站的实例，netstore应用是利用Struts作为Java Web框架，用Hibernate来完成对象持久化的任务，并且分别用普通的JavaBean及EJB组件来实现业务逻辑。

9. 附录本书的附录介绍标准SQL语言的主要用法、Java的反射机制、XDoclet工具的用法, 以及把Hibernate 2应用升级到Hibernate 3应用的方法。

在介绍标准SQL语言和Java反射机制时, 都不是泛泛而谈, 而是有针对性地介绍与Hibernate紧密相关的知识, 如SQL连接查询, 以及运用Java反射机制来实现持久化中间件的基本原理。

本书的范例程序为了使读者不但能掌握用Hibernate来持久化Java对象的理论, 并且能迅速获得开发Hibernate应用的实际经验, 彻底掌握并会灵活运用Hibernate技术, 本书几乎为每一章都提供完整的Hibernate应用范例, 本书配套光盘中包含所有范例源文件。

为了方便初学者能顺利地运行本书的范例, 光盘上提供的所有范例程序都是可运行的。

读者只要把它们复制到本地机器上, 就能够运行, 不需要再做额外的配置。

此外, 在每个范例的根目录下还提供了ANT工程的工程文件build.xml, 它用于编译和运行范例程序。

本书最后还提供了一个完整的netstore应用例子, 它实现了一个购物网站, 更加贴近实际应用。

本书以netstore应用为例, 介绍软件的MVC框架, 控制层与模型层之间通过游离对象来传输数据的方式, 以及模型层采用合理的检索策略来控制检索出来的对象图的深度, 从而优化应用的性能的技巧。

这本书是否适合您把Java对象持久化到关系数据库, 几乎是所有企业Java应用必不可少的重要环节, 因此本书适用于所有从事开发Java应用的读者。

Hibernate是Java应用和关系数据库之间的桥梁, 阅读本书, 要求读者具备Java语言和关系数据库的基础知识。

如果您是开发Hibernate应用的新手, 建议按照本书的先后顺序来学习。

可以先从简单的Hibernate应用实例下手, 把握开发Hibernate应用的大致流程, 然后逐步深入地了解把对象模型映射到关系数据模型的各种细节。

如果您已经在开发Hibernate应用方面有着丰富的经验, 则可以把本书作为实用的Hibernate技术参考资料。

本书深入探讨把复杂的对象模型映射到关系数据模型的各种映射方案, 详细介绍通过Hibernate API来操纵数据库的方法。

灵活运用本书介绍的Hibernate最新技术, 将使您开发Hibernate应用更加得心应手。

实践是掌握Hibernate的好方法。

为了让读者彻底掌握并学会灵活运用Hibernate, 本书为每一章都提供了典型范例, 本书配套光盘中提供了完整的源代码, 以及软件安装程序。

建议读者在学习Hibernate技术的过程中, 善于将理论与实践相结合, 以达到事半功倍的效果。

光盘使用说明本书配套光盘包含以下目录。

1. software目录在该目录下包含本书内容涉及的所有软件的最新版本的安装程序, 包括: (1)

) Hibernate核心软件包 (Hibernate 3.3)。

(2) Hibernate工具软件包 (HibernateTools 3.2)。

(3) MySQL服务器的安装软件 (MySQL 5)。

(4) MySQL的JDBC驱动程序 (Mysql-Connector-Java-3.1.7) (5) ANT的安装软件 (Ant 1.7)。

(6) Tomcat的安装软件 (Tomcat 6)。

(7) JBoss的安装软件 (JBoss 4)。

2. sourcecode目录在该目录下提供本书所有的源程序。

3. ppt目录在该目录下提供由本书作者亲自制作的配套语音讲座。

写作规范为了节省文章的篇幅, 在本书中显示范例的源代码时, 有时做了一些省略。

对于Java类, 省略显示package语句和import语句。

除了netstore应用外, 本书其他范例创建的Java类都位于mypack包下。

对于持久化类, 还省略显示了属性的getXXX()和setXXX()方法。

对于对象-关系映射文件, 省略显示开头的和元素。

但在配套光盘中可获得完整的源代码。

在本书提供的SQL语句中, 表名和字段名都采用大写形式, 而SQL关键字, 如select、from、insert、update和delete等, 都采用小写形式。

在本书中，有时把运用了Hibernate技术的Java应用简称为Hibernate应用。

此外，对象和实例是相同的概念；覆盖方法、重新定义方法，以及重新实现方法是相同的概念；继承和扩展是相同的概念；表的记录和表的数据行是相同的概念；表的字段和表的数据列是相同的概念；查询与检索是相同的概念；持久化类和POJO都是指其实例需要被持久化的基于JavaBean形式的实体域对象；对象-关系映射文件和映射文件是相同的概念；本书中的应用服务器主要指J2EE服务器。

本书在编写过程中得到了Hibernate软件组织和SUN公司在技术上的大力支持，飞思科技产品研发中心负责监制工作，此外本书第1版的读者及JavaThinker.org网站的网友为本书的编写提供了有益的帮助，在此表示衷心的感谢！

尽管我们尽了最大努力，但本书难免会有不妥之处，欢迎各界专家和读者朋友批评指正。

以下网址是作者为本书提供的技术支持网址，读者可通过它下载与本书相关的资源（如源代码、软件安装程序和讲义等），还可以与其他读者交流学习心得，以及对本书提出宝贵意见

： [http://www.javathinker.org/hibernate\\_support.jsp](http://www.javathinker.org/hibernate_support.jsp)

## <<精通Hibernate>>

### 内容概要

《精通Hibernate：Java对象持久化技术详解（第2版）》在第一版基础上，理论内容更新，更加注重理论与实践的结合，有大量具有典型性和实用价值的Hibernate应用实例。

全书将详细介绍运用目前最成熟的Hibernate 3.3版本进行Java对象持久化的技术，既能掌握用Hibernate工具对这两种模型进行映射的技术，又能获得设计与开发Java对象模型和关系数据模型的经验。

光盘包括《精通Hibernate：Java对象持久化技术详解（第2版）》范例源程序、涉及软件的最新版本的安装程序，以及配套的语音讲座。

《精通Hibernate：Java对象持久化技术详解（第2版）》无论对于Java开发的新手还是行家来说，都是精通Java对象持久化技术的必备实用手册。

## 作者简介

孙卫琴：知名IT作家的，Java系列畅销书的作者。

1997年毕业于上海交通大学，随后从事基于Java的软件开发工作，还从事Java方面的培训和咨询工作，2002年开始Java技术领域的创作，其作者深受读者欢迎。

代表作有：《精通Hibernate：Java对象持久化技术详解（第2版）》2010-1-13《Tomcat与JavaWeb开发技术详解（第2版）》2009《Java网络编程精解》2007《Java面向对象编程》2006《精通Hibernate：Java对象持久化技术详解》2005《精通Struts：基于MVC的JavaWeb设计与开发》2004《Tomcat与JavaWeb开发技术详解》2004《Java2认证考试指南与试题解析》2002

## 书籍目录

第1章 Java应用分层架构及软件模型 11.1 应用程序的分层体系结构 11.1.1 区分物理层和逻辑层 21.1.2 软件层的特征 31.1.3 软件分层的优点 41.1.4 软件分层的缺点 41.1.5 Java应用的持久化层 51.2 软件的模型 61.2.1 概念模型 71.2.2 关系数据模型 81.2.3 域模型 101.2.4 域对象 101.2.5 域对象之间的关系 111.2.6 域对象的持久化概念 171.3 小结 191.4 思考题 19第2章 Java对象持久化技术概述 212.1 直接通过JDBC API来持久化实体域对象 212.2 ORM简介 272.2.1 对象-关系映射的概念 292.2.2 ORM中间件的基本使用方法 312.2.3 常用的ORM中间件 332.3 实体域对象的其他持久化模式 342.3.1 主动域对象模式 352.3.2 JDO模式 372.3.3 CMP模式 372.4 Hibernate API简介 382.4.1 Hibernate的核心接口 392.4.2 事件处理接口 412.4.3 Hibernate映射类型接口 412.4.4 可供扩展的接口 422.5 小结 432.6 思考题 45第3章 第一个Hibernate应用 473.1 创建Hibernate的配置文件 473.2 创建持久化类 483.3 创建数据库Schema 513.4 创建对象-关系映射文件 523.4.1 映射文件的文档类型定义 (DTD) 523.4.2 把Customer持久化类映射到CUSTOMERS表 543.5 通过Hibernate API操纵数据库 583.5.1 Hibernate的初始化 613.5.2 访问Hibernate的Session接口 633.6 运行helloapp应用 673.6.1 创建运行本书范例的系统环境 673.6.2 创建helloapp应用的目录结构 723.6.3 把helloapp应用作为独立应用程序运行 733.6.4 把helloapp应用作为Java Web应用运行 773.7 小结 783.8 思考题 80第4章 hbm2java和hbm2ddl工具 834.1 创建对象-关系映射文件 834.1.1 定制持久化类 854.1.2 定制数据库表 884.2 建立项目的目录结构 904.3 运行hbm2java工具 934.4 运行hbm2ddl工具 944.5 使用XML格式的配置文件 964.6 小结 974.7 思考题 98第5章 对象-关系映射基础 1015.1 持久化类的属性及访问方法 1015.1.1 基本类型属性和包装类型属性 1025.1.2 Hibernate访问持久化类属性的策略 1045.1.3 在持久化类的访问方法中加入程序逻辑 1045.1.4 设置派生属性 1075.1.5 控制insert和update语句 1085.2 处理SQL引用标识符 1095.3 创建命名策略 1105.4 设置数据库Schema 1125.5 设置类的包名 1135.6 运行本章的范例程序 1145.7 小结 1205.8 思考题 121第6章 映射对象标识符 1236.1 关系数据库按主键区分不同的记录 1236.1.1 把主键定义为自动增长标识符类型 1236.1.2 从序列 (Sequence) 中获取自动增长的标识符 1246.2 Java语言按内存地址区分不同的对象 1256.3 Hibernate用对象标识符 (OID) 来区分对象 1266.4 Hibernate的内置标识符生成器的用法 1286.4.1 increment标识符生成器 1316.4.2 identity标识符生成器 1336.4.3 sequence标识符生成器 1346.4.4 hilo标识符生成器 1356.4.5 native标识符生成器 1376.5 映射自然主键 1386.5.1 映射单个自然主键 1386.5.2 映射复合自然主键 1406.6 小结 1436.7 思考题 144第7章 映射一对多关联关系 1477.1 建立多对一的单向关联关系 1487.1.1 元素的not-null属性 1537.1.2 级联保存和更新 1557.2 映射一对多双向关联关系 1567.2.1 元素的inverse属性 1617.2.2 级联删除 1637.2.3 父子关系 1647.3 映射一对多双向自身关联关系 1657.4 改进持久化类 1717.5 小结 1757.6 思考题 176第8章 通过Hibernate操纵对象 (上) 1798.1 Java对象在JVM中的生命周期 1798.2 理解Session的缓存 1818.2.1 Session的缓存的作用 1828.2.2 脏检查及清理缓存的机制 1848.3 Java对象在Hibernate持久化层的状态 1878.3.1 临时对象的特征 1888.3.2 持久化对象的特征 1898.3.3 被删除对象的特征 1908.3.4 游离对象的特征 1918.4 Session接口的详细用法 1918.4.1 Session的save()和persist()方法 1918.4.2 Session的load()和get()方法 1948.4.3 Session的update()方法 1958.4.4 Session的saveOrUpdate()方法 1978.4.5 Session的merge()方法 1988.4.6 Session的delete()方法 2008.4.7 Session的replicate()方法 2018.5 级联操纵对象图 2028.5.1 级联保存临时对象 2068.5.2 更新持久化对象 2078.5.3 持久化临时对象 2078.5.4 更新游离对象 2098.5.5 遍历对象图 2108.6 小结 2118.7 思考题 211第9章 通过Hibernate操纵对象 (下) 2159.1 与触发器协同工作 2159.2 利用拦截器 (Interceptor) 生成审计日志 2179.3 Hibernate的事件处理机制 2249.4 批量处理数据 2269.4.1 通过Session来进行批量操作 2279.4.2 通过StatelessSession来进行批量操作 2299.4.3 通过HQL来进行批量操作 2309.4.4 直接通过JDBC API来进行批量操作 2329.5 使用元数据 2339.6 通过Hibernate调用存储过程 2349.7 小结 2349.8 思考题 235第10章 映射组成关系 23710.1 建立精粒度对象模型 23810.2 建立粗粒度关系数据模型 23910.3 映射组成关系 24010.3.1 区分值 (Value) 类型和实体 (Entity) 类型 24210.3.2 在应用程序中访问具有组成关系的持久化类 24410.4 映射复合组成关系 24710.5 小结 24810.6 思考题 249第11章 Hibernate的映射类型 25111.1 Hibernate的内置映射类型 25111.1.1 Java基本类型的Hibernate映射类型 25111.1.2 Java时间和日期类型的Hibernate映射类型 25211.1.3 Java大对象类型的Hibernate映射类型 25311.1.4 JDK自带的个别Java类的Hibernate映射类型 25311.1.5 使



用Hibernate内置映射类型 25411.2 客户化映射类型 25611.2.1 用客户化映射类型取代Hibernate组件  
26011.2.2 用UserType映射枚举类型 26311.2.3 实现CompositeUserType接口 26611.2.4 运行本节范例程序  
27111.3 操纵Blob和Clob类型数据 27911.4 小结 28411.5 思考题 284第12章 映射继承关系 28712.1 继承关系树的每个具体类对应一个表 28812.1.1 创建映射文件 28912.1.2 操纵持久化对象 29012.2 继承关系树的根类对应一个表 29312.2.1 创建映射文件 29412.2.2 操纵持久化对象 29612.3 继承关系树的每个类对应一个表 29712.3.1 创建映射文件 29812.3.2 操纵持久化对象 30012.4 选择继承关系的映射方式 30212.5 映射多对一多态关联 30512.6 小结 30812.7 思考题 309第13章 Java集合类 31113.1 Set (集) 31213.1.1 Set的一般用法 31213.1.2 HashSet类 31313.1.3 TreeSet类 31513.1.4 向Set中加入持久化类的对象 31913.2 List (列表) 32013.3 Map (映射) 32113.4 小结 32513.5 思考题 326第14章 映射值类型集合 32714.1 映射Set (集) 32714.2 映射Bag (包) 33114.3 映射List (列表) 33414.4 映射Map 33714.5 对集合排序 33914.5.1 在数据库中对集合排序 34014.5.2 在内存中对集合排序 34114.6 映射组件类型集合 34514.7 小结 35014.8 思考题 351第15章 映射实体关联关系 35315.1 映射一对一关联 35315.1.1 按照外键映射 35415.1.2 按照主键映射 35815.2 映射单向多对多关联 36115.3 映射双向多对多关联关系 36515.3.1 关联两端使用元素 36615.3.2 在inverse端使用元素 36715.3.3 使用组件类集合 37115.3.4 把多对多关联分解为两个一对多关联 37615.4 小结 37815.5 思考题 379第16章 Hibernate的检索策略 38116.1 Hibernate的检索策略简介 38316.2 类级别的检索策略 38616.2.1 立即检索 38716.2.2 延迟检索 38716.3 一对多和多对多关联的检索策略 39016.3.1 立即检索 (lazy属性为“false”) 39216.3.2 延迟检索 (lazy属性为默认值“true”) 39216.3.3 增强延迟检索 (lazy属性为“extra”) 39316.3.4 批量延迟检索和批量立即检索 (使用batch-size属性) 39316.3.5 用带子查询的select语句整批初始化orders集合 (fetch属性为“subselect”) 39616.3.6 迫切左外连接检索 (fetch属性为“join”) 39716.4 多对一和一对一关联的检索策略 39816.4.1 迫切左外连接检索 (fetch属性为“join”) 39816.4.2 延迟检索 (lazy属性为默认值“proxy”) 40016.4.3 无代理延迟检索 (lazy属性为“no-proxy”) 40116.4.4 立即检索 (lazy属性为“false”) 40116.4.5 批量延迟检索和批量立即检索 (使用batch-size属性) 40216.5 控制迫切左外连接检索的深度 40516.6 在应用程序中显式指定迫切左外连接检索策略 40816.7 属性级别的检索策略 40816.8 小结 40916.9 思考题 411第17章 Hibernate的检索方式 (上) 41317.1 Hibernate的检索方式简介 41317.1.1 HQL检索方式 41617.1.2 QBC检索方式 41717.1.3 本地SQL检索方式 41917.1.4 关于本章范例程序 42017.1.5 使用别名 42117.1.6 多态查询 42117.1.7 对查询结果排序 42217.1.8 分页查询 42317.1.9 检索单个对象 (uniqueResult()方法) 42417.1.10 按主键逐个处理查询结果 (iterate()方法) 42517.1.11 可滚动的结果集 42617.1.12 在HQL查询语句中绑定参数 42817.1.13 设置查询附属事项 43317.1.14 在映射文件中定义命名查询语句 43617.1.15 在HQL查询语句中调用函数 43817.2 设定查询条件 43917.2.1 比较运算 44017.2.2 范围运算 44117.2.3 字符串模式匹配 44217.2.4 逻辑运算 44317.2.5 集合运算 44417.3 小结 44517.4 思考题 446第18章 Hibernate的检索方式 (下) 44918.1 连接查询 44918.1.1 默认情况下关联级别的运行时检索策略 45018.1.2 迫切左外连接 45118.1.3 左外连接 45318.1.4 内连接 45718.1.5 迫切内连接 46118.1.6 隐式内连接 46318.1.7 右外连接 46418.1.8 使用SQL风格的交叉连接和隐式内连接 46618.1.9 关联级别运行时的检索策略 46718.2 投影查询 46818.3 报表查询 47118.3.1 使用聚集函数 47118.3.2 分组查询 47218.3.3 优化报表查询的性能 47518.4 高级查询技巧 47518.4.1 动态查询 47618.4.2 集合过滤 47818.4.3 子查询 48018.4.4 本地SQL查询 48218.4.5 查询结果转换器 48418.5 查询性能优化 48618.5.1 iterate()方法 48618.5.2 查询缓存 48718.6 小结 48818.7 思考题 489第19章 Hibernate高级配置 49119.1 配置数据库连接池 49119.1.1 使用默认的数据库连接池 49419.1.2 使用配置文件指定的数据库连接池 49519.1.3 从容器中获得数据源 49519.1.4 由Java应用本身提供数据库连接 49719.2 配置事务类型 49719.3 把SessionFactory与JNDI绑定 50019.4 配置日志 50219.5 使用XML格式的配置文件 50519.6 小结 50719.7 思考题 508第20章 声明数据库事务 51120.1 数据库事务的概念 51120.2 声明事务边界的方式 51320.3 在mysql.exe程序中声明事务 51520.4 Java应用通过JDBC API声明JDBC事务 51720.5 Java应用通过Hibernate API声明JDBC事务 51820.5.1 处理异常 52020.5.2 Session与事务的关系 52220.5.3 设定事务超时 52520.6 Java应用通过Hibernate API声明JTA事务 52520.7 Java应用通过JTA API声明JTA事务 52620.8 小结 52920.9 思考题 530第21章 处理并发问题 53321.1 多个事务并发运行时的并发问题 53321.1.1 第一类丢失更新 53421.1.2 脏读 53521.1.3 虚读 53521.1.4 不可重复读 53621.1.5 第二类丢失更新 53621.2 数据库

系统的锁的基本原理 53721.2.1 锁的多粒度性及自动锁升级 53821.2.2 锁的类型和兼容性 53821.2.3 死锁及其防止办法 54021.3 数据库的事务隔离级别 54121.3.1 在mysql.exe程序中设置隔离级别 54321.3.2 在应用程序中设置隔离级别 54321.4 在应用程序中采用悲观锁 54421.4.1 利用数据库系统的独占锁来实现悲观锁 54421.4.2 由应用程序实现悲观锁 55021.5 利用Hibernate的版本控制来实现乐观锁 55021.5.1 使用元素 55121.5.2 使用元素 55621.5.3 对游离对象进行版本检查 55821.5.4 强制更新版本 55921.6 实现乐观锁的其他方法 55921.7 小结 56021.8 思考题 561

第22章 管理Hibernate的缓存 56322.1 缓存的基本原理 56322.1.1 持久化层的缓存的范围 56422.1.2 持久化层的缓存的并发访问策略 56622.2 Hibernate的二级缓存结构 56822.3 管理Hibernate的第一级缓存 56922.4 管理Hibernate的第二级缓存 57022.4.1 配置进程范围内的第二级缓存 57122.4.2 配置集群范围内的第二级缓存 57522.4.3 在应用程序中管理第二级缓存 57822.4.4 Session与第二级缓存的交互模式 57822.5 运行本章的范例程序 57922.6 小结 58322.7 思考题 584

第23章 管理Session和实现对话 58723.1 管理Session对象的生命周期 58723.1.1 Session对象的生命周期与本地线程绑定 59023.1.2 Session对象的生命周期与JTA事务绑定 59223.2 实现对话 59323.2.1 使用游离对象 59523.2.2 使用手工清理缓存模式下的Session 59723.3 小结 60323.4 思考题 605

第24章 Hibernate与Struts框架 60724.1 实现业务数据 60924.2 实现业务逻辑 61224.3 netstore应用的订单业务 62224.4 小结 626

第25章 Hibernate与EJB组件 62925.1 创建EJB组件 62925.1.1 编写Remote接口 62925.1.2 编写Home接口 63125.1.3 编写Enterprise Java Bean类 63125.2 在业务代理类中访问EJB组件 63325.3 发布J2EE应用 63825.3.1 在JBoss上部署EJB组件 63825.3.2 在JBoss上部署Web应用 63925.3.3 在JBoss上部署J2EE应用 64025.4 小结 642

附录A 标准SQL语言的用法 643A.1 数据完整性 644A.1.1 实体完整性 644A.1.2 域完整性 644A.1.3 参照完整性 644A.2 DDL数据定义语言 644A.3 DML数据操纵语言 646A.4 DQL数据查询语言 647A.4.1 简单查询 648A.4.2 连接查询 648A.4.3 子查询 651A.4.4 联合查询 652A.4.5 报表查询 652

附录B Java语言的反射机制 655B.1 Java Reflection API简介 655B.2 运用反射机制来持久化Java对象 658

附录C 用XDoclet工具生成映射文件 665C.1 创建带有@hibernate标记的Java源文件 665C.2 建立项目的目录结构 669C.3 运行XDoclet工具 672

附录D 发布和运行netstore应用 675D.1 运行netstore所需的软件 675D.2 netstore应用的目录结构 676D.3 安装SAMPLEDB数据库 677D.4 安装和配置JBoss服务器 677D.5 发布netstore应用 679D.5.1 在工作模式1下发布netstore应用 679D.5.2 在工作模式2下发布netstore应用 679D.6 运行netstore应用 680

附录E Hibernate 3升级指南 685E.1 Hibernate API变化 685E.1.1 包名 685E.1.2 org.hibernate.classic包 685E.1.3 Hibernate所依赖的第三方软件包 686E.1.4 异常模型 686E.1.5 Session接口 686E.1.6 createSQLQuery() 687E.1.7 Lifecycle和Validatable接口 687E.1.8 Interceptor接口 687E.1.9 UserType和CompositeUserType接口 687E.1.10 FetchMode类 688E.1.11 PersistentEnum类 688E.1.12 对Blob和Clob的支持 688E.1.13 Hibernate中供扩展的API的变化 688E.2 元数据的变化 688E.2.1 检索策略 688E.2.2 对象标识符的映射 688E.2.3 集合映射 689E.2.4 DTD 689E.3 查询语句的变化 689E.4 把Hibernate 2应用升级到Hibernate 3应用 690

附录F 思考题答案 691参考文献 693

## 章节摘录

第9章 通过Hibernate操纵对象（下） 9.1 与触发器协同工作 数据库系统有时会利用触发器来完成某些业务规则。

触发器在接收到特定的事件时被激发，执行事先定义好的一组数据库操作。

能激发触发器运行的事件可分为以下几种： 插入记录事件，即执行insert语句。

更新记录事件，即执行update语句。

删除记录事件，即执行delete语句。

当Hibernate与数据库中的触发器协同工作时，会造成两类问题：触发器使Session的缓存中的持久化对象与数据库中对应数据不一致。

Session的update（）方法盲目地激发触发器。

下面分别介绍出现这两类问题的原因及解决办法。

1. 触发器使Session缓存中的数据与数据库不一致 当Session向数据库中保存、更新或删除对象时，如果会激发数据库中的某个触发器，常常会带来一个问题，那就是Session缓存中的持久化对象无法与数据库中的数据保持同步。

出现这一问题的原因在于触发器运行在数据库中，它执行的操作对Session是透明的，假如在Session的缓存中已经存在一个Customer对象，接下来当触发器修改数据库中CUSTOMERS表的相应记录时，Session无法检测到数据库中数据的变化，因此Session不会自动刷新缓存中的Customer对象。

下面举例说明。

<<精通Hibernate>>

编辑推荐

凝聚Java魅力，成就开发专家看清Java万花筒的本质，从复杂的表象中寻找普遍的规律，深刻理解Java的核心思想本书在第1版的基础上，并结合作者多年经验，将更新的理论知识，精心细致地介绍最新技术应用。

#### 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>