

<<数据库原理与应用>>

图书基本信息

书名：<<数据库原理与应用>>

13位ISBN编号：9787121124617

10位ISBN编号：7121124610

出版时间：2011-1

出版时间：高凯 电子工业出版社 (2011-01出版)

作者：高凯 编

页数：314

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>

<<数据库原理与应用>>

前言

在社会信息化的今天，信息已经成为全社会宝贵的资源。

就信息本身而言，大致可将其分为结构化信息、半结构化信息和非结构化信息三类。

作为管理结构化信息的有效手段，数据库系统对于当今科研部门、政府机关、企事业单位等都是至关重要的；作为数据库系统中的核心，数据库管理系统DBMS——特别是关系型数据库管理系统(如Microsoft SQL Server、Oracle、Sybase等)——用于高效创建数据库和存储大量数据并对其进行有效管理和维护，其本身具有很大的工程实用价值。

同时，数据库技术也是计算机科学与技术领域中发展飞快的一个分支，在其短短40年的发展历程中，已造就了包括C. W. Bachman、E. F. Code、James Gray等在内的多位图灵奖得主，发展成为以数据建模和DBMS为核心、具有很大工程实践价值的学科，数据库及其应用技术也因此成为计算机科学与技术及相关专业的重要专业课程。

同时也要看到，虽然传统的关系型数据库系统在管理结构化信息方面具有很大优势，但在网络迅速普及的今天，半结构化信息和非结构化信息所占的比重正在逐步增大，已经逐渐成为重要的信息组织方式。

因此，在本书前几章对传统的关系数据库系统理论、关系数据库标准语言SQL、规范化理论、数据库设计、关系查询处理和查询优化、数据库完整性和安全性、数据库恢复和并发控制等基础理论进行综述的基础上，针对当前的应用实际，对基于数据库的Web应用程序开发与实现、XML技术在数据库中的应用等进行概述，以使读者能对包括结构化、半结构化信息在内的数据管理技术有较全面的了解。

<<数据库原理与应用>>

内容概要

《数据库原理与应用》全面、系统地介绍了关系数据库基础理论及应用。在介绍关系数据库系统理论、SQL语言、规范化理论、数据库设计、关系查询处理和查询优化、数据库完整性和安全性、数据库恢复和并发控制等理论的基础上，针对实际应用，介绍了以VB、Delphi、VC++、C#等常用编程工具开发数据库应用程序的常用方法和技巧，介绍了基于数据库的Web应用程序开发、XML技术等。书后配有7个数据库实验。本书配有电子课件、完整的开发案例源代码等教学资源，方便教学和工程实践应用。

<<数据库原理与应用>>

书籍目录

第一部分 原理篇第1章 绪论 (2) 1.1 数据管理技术的产生和发展 (2) 1.1.1 人工管理阶段 (2) 1.1.2 文件系统阶段 (3) 1.1.3 数据库系统阶段 (3) 1.2 数据库基本概念 (5) 1.2.1 数据 (5) 1.2.2 数据库 (5) 1.2.3 数据库管理系统 (6) 1.2.4 数据库系统 (6) 1.3 数据模型 (7) 1.3.1 数据模型的组成要素 (7) 1.3.2 数据模型分类 (8) 1.3.3 概念模型及表示方法 (8) 1.3.4 数据模型 (12) 1.4 数据库系统的结构 (14) 1.4.1 数据库系统的三级模式结构 (15) 1.4.2 数据库系统的二级映像 (15) 1.5 数据库管理系统 (16) 1.5.1 DBMS的工作模式 (16) 1.5.2 DBMS的功能 (17) 1.5.3 DBMS的组成 (17) 1.6 小结 (18) 习题1 (19) 第2章 关系数据库基础 (20) 2.1 关系模型的基本概念 (20) 2.1.1 域、笛卡儿积、关系、关系模式 (20) 2.1.2 码 (23) 2.2 关系模型的完整性约束 (24) 2.2.1 实体完整性规则 (24) 2.2.2 参照完整性规则 (25) 2.2.3 用户自定义完整性规则 (25) 2.3 关系操作 (26) 2.3.1 关系数据语言 (26) 2.3.2 关系代数 (26) 2.4 小结 (34) 习题2 (34) 第3章 关系数据库标准语言SQL (37) 3.1 SQL概述 (37) 3.1.1 SQL的发展历史 (37) 3.1.2 SQL的特点 (37) 3.1.3 SQL数据库的体系结构 (38) 3.2 数据定义 (39) 3.2.1 模式的定义与删除 (40) 3.2.2 基本表的定义、删除和修改 (40) 3.2.3 索引的建立和删除 (43) 3.3 数据查询 (44) 3.3.1 单表无条件查询 (45) 3.3.2 单表带条件查询 (48) 3.3.3 分组查询和排序查询 (51) 3.3.4 连接查询 (53) 3.3.5 嵌套查询 (56) 3.3.6 集合查询 (60) 3.4 数据更新 (62) 3.4.1 插入数据 (62) 3.4.2 删除数据 (63) 3.4.3 修改数据 (63) 3.5 视图 (64) 3.5.1 建立视图 (64) 3.5.2 删除视图 (65) 3.5.3 查询视图 (66) 3.5.4 更新视图 (66) 3.5.5 视图的作用 (67) 3.6 小结 (68) 习题3 (68) 第4章 规范化理论 (69) 4.1 问题的提出 (69) 4.1.1 关系模式中可能存在的问题 (69) 4.1.2 解决的方法 (70) 4.2 函数依赖 (70) 4.2.1 函数依赖的基本概念 (71) 4.2.2 函数依赖的推理规则 (72) 4.2.3 码的函数依赖表示 (75) 4.2.4 最小函数依赖集 (75) 4.3 规范化 (77) 4.3.1 范式 (77) 4.3.2 模式分解 (80) 4.3.3 关系模式规范化步骤 (86) 4.4 小结 (87) 习题4 (87) 第5章 数据库设计 (89) 5.1 数据库设计概述 (89) 5.1.1 数据库设计的任务和特点 (89) 5.1.2 数据库设计方法 (90) 5.1.3 数据库设计步骤 (90) 5.1.4 数据库设计过程中的各级模式 (92) 5.2 需求分析 (92) 5.2.1 需求分析的任务 (92) 5.2.2 需求分析的方法和步骤 (93) 5.2.3 需求分析的结果 (94) 5.3 概念结构设计 (95) 5.3.1 概念结构的特点 (95) 5.3.2 概念设计方法 (95) 5.3.3 概念设计的步骤 (95) 5.4 逻辑结构设计 (100) 5.4.1 E-R模型到关系模式的转换 (100) 5.4.2 逻辑模式的优化 (103) 5.4.3 设计用户子模式 (103) 5.5 数据库的物理设计 (104) 5.5.1 数据库物理设计的内容和方法 (104) 5.5.2 确定数据库的物理结构 (105) 5.5.3 评价物理结构 (106) 5.6 数据库的实施和维护 (106) 5.6.1 建立数据库和应用程序的调试 (106) 5.6.2 数据库的试运行 (106) 5.6.3 数据库的运行和维护 (107) 5.7 小结 (108) 习题5 (108) 第6章 关系查询处理和查询优化 (109) 6.1 关系数据库系统的查询处理 (109) 6.1.1 查询优化的必要性 (109) 6.1.2 查询处理的步骤 (111) 6.1.3 查询的执行代价 (111) 6.2 代数优化 (112) 6.2.1 关系代数表达式等价变换规则 (112) 6.2.2 查询优化一般策略 (113) 6.2.3 关系代数表达式的优化算法 (114) 6.3 物理优化 (116) 6.3.1 基于规则的优化方法 (116) 6.3.2 基于代价估算的优化方法 (116) 6.4 SQL查询语句的优化 (117) 6.4.1 合理使用索引 (117) 6.4.2 书写高效率的SQL查询语句 (118) 6.4.3 使用存储过程 (120) 6.5 小结 (123) 习题6 (123) 第7章 数据库完整性和安全性 (124) 7.1 数据库完整性 (124) 7.1.1 实体完整性 (124) 7.1.2 参照完整性 (125) 7.1.3 用户定义的完整性 (127) 7.1.4 完整性约束命名子句 (128) 7.1.5 触发器 (129) 7.2 数据库安全性 (130) 7.2.1 数据库安全性概述 (130) 7.2.2 用户标识与鉴别 (130) 7.2.3 存取控制 (131) 7.2.4 其他安全控制方法 (135) 7.3 SQL Server的完整性控制和安全控制 (137) 7.3.1 SQL Server的完整性控制 (137) 7.3.2 SQL Server的安全控制 (139) 7.4 小结 (141) 习题7 (142) 第8章 数据库恢复和并发控制 (143) 8.1 事务 (143) 8.1.1 事务的基本概念 (143) 8.1.2 事务的性质 (143) 8.2 数据库恢复技术 (144) 8.2.1 故障的种类 (144) 8.2.2 恢复的实现技术 (145) 8.2.3 故障恢复的策略 (146) 8.2.4 SQL Server的数据库恢复技术 (149) 8.3 并发控制 (151) 8.3.1 并发操作带来的问题 (151) 8.3.2 封锁 (152) 8.3.3 并发调度的可串行性 (155) 8.3.4 两段锁协议 (156) 8.3.5 封锁的粒度 (157) 8.3.6 SQL Server的封锁方式 (159) 8.4 小结 (159) 习题8 (160) 第二部分 应用篇第9章 数据库应用技术概述 (162) 9.1 数据库访问接口 (162) 9.1.1 ODBC简介 (162) 9.1.2 DAO简介 (163) 9.1.3 OLE DB简介 (164) 9.1.4 ADO简介 (165) 9.1.5 ADO.NET简介 (165) 9.1.6 内嵌式SQL、RDO、DB库简介 (166) 9.2 数据库应用体系结构

<<数据库原理与应用>>

(167) 9.2.1 客户机/服务器工作模式 (168) 9.2.2 客户机/服务器模式的主要技术特征 (168) 9.2.3 客户机/服务器系统体系结构 (168) 9.2.4 分布式数据库系统 (171) 9.3 基于B/S架构的数据库应用程序 (172) 9.4 ODBC数据访问接口及实例操作 (173) 9.4.1 ODBC概述 (173) 9.4.2 MFC ODBC数据库类 (180) 9.4.3 应用案例——VC++6.0中使用ODBC开发个人通讯录系统 (185) 9.5 小结 (194) 习题9 (194) 第10章 ADO数据访问技术及操作 (195) 10.1 ADO概述 (195) 10.2 ADO数据库访问对象 (196) 10.2.1 Connection对象 (196) 10.2.2 Command对象 (197) 10.2.3 Recordset对象 (198) 10.2.4 Parameter对象 (201) 10.2.5 Field对象 (201) 10.2.6 Record对象 (202) 10.2.7 Stream对象 (203) 10.2.8 Error对象 (204) 10.2.9 Property对象 (204) 10.3 高级语言操作ADO接口 (204) 10.3.1 在VB 6.0中使用ADO (204) 10.3.2 在VC 6.0中使用ADO (207) 10.3.3 在Delphi 7中使用ADO (210) 10.4 应用案例——Delphi中使用ADO开发个人通讯录系统 (213) 10.5 小结 (215) 习题10 (215) 第11章 ADO.NET数据访问技术及操作 (216) 11.1 .NET平台及VS.NET 2008简介 (216) 11.1.1 .NET Framework简介 (216) 11.1.2 VS.NET 2008集成开发环境简介 (217) 11.1.3 C#开发语言简介 (218) 11.2 ADO.NET概述 (218) 11.2.1 ADO.NET的优点 (218) 11.2.2 ADO.NET的结构 (219) 11.3 使用ADO.NET进行数据库访问 (220) 11.3.1 ADO.NET Connection对象 (221) 11.3.2 ADO.NET Command对象 (222) 11.3.3 ADO.NET DataReader对象 (223) 11.3.4 ADO.NET DataAdapter对象 (224) 11.3.5 ADO.NET DataSet对象 (226) 11.4 应用案例——C#中使用ADO.NET开发个人通讯录系统 (227) 11.4.1 通过编写代码实现通讯录系统 (227) 11.4.2 使用ADO.NET控件实现通讯录系统开发 (229) 11.5 小结 (235) 习题11 (235) 第12章 基于C#的数据库系统应用案例——教材征订系统 (236) 12.1 系统设计 (236) 12.1.1 需求分析 (236) 12.1.2 概念结构设计 (237) 12.1.3 数据库设计 (238) 12.2 功能模块设计 (241) 12.2.1 系统功能模块 (241) 12.2.2 界面设计 (241) 12.3 功能实现步骤 (242) 12.3.1 数据库创建 (242) 12.3.2 公用模块设计 (242) 12.3.3 添加app.config文件 (247) 12.4 管理员端程序设计步骤 (247) 12.4.1 用户管理界面 (247) 12.4.2 用户管理信息编辑界面设计 (250) 12.4.3 登录窗体的设计 (254) 12.5 系统运行与调试 (256) 12.5.1 调试方法 (256) 12.5.2 调试过程 (256) 12.5.3 排错 (257) 12.6 小结 (258) 习题12 (258) 第13章 基于数据库的Web应用程序开发 (259) 13.1 概述 (259) 13.2 常见的Web服务器简介 (260) 13.2.1 IIS (260) 13.2.2 Tomcat (262) 13.2.3 WebLogic (263) 13.3 服务器端的动态网页技术 (264) 13.3.1 JSP (264) 13.3.2 基于ASP.NET构建Web应用程序 (268) 13.4 客户端网页设计相关技术简介 (271) 13.4.1 基于CSS+DIV构建网页 (271) 13.4.2 客户端脚本语言JavaScript简介 (273) 13.5 小结 (276) 习题13 (276) 第14章 XML与数据库技术 (277) 14.1 基于XML的数据库信息管理 (277) 14.1.1 XML及其与数据库的互操作 (278) 14.1.2 XML的基本规范和简单结构 (279) 14.1.3 XML数据的显示 (279) 14.1.4 数据库信息转换为XML (281) 14.1.5 XML与数据库信息的互转换 (283) 14.2 基于XML的网络异构数据的采集与处理 (285) 14.2.1 异构资源采集 (286) 14.2.2 利用HttpClient检索数据 (286) 14.2.3 构造XML文档 (288) 14.2.4 在浏览器端显示XML文档 (291) 14.2.5 提供统一检索接口 (293) 14.3 小结 (294) 习题14 (294) 第三部分 实验和辅助工具篇 实验1 熟悉SQL Server 2000环境及数据库管理 (296) 实验2 数据定义 (297) 实验3 数据操纵 (298) 实验4 简单查询 (300) 实验5 连接查询和嵌套查询 (300) 实验6 数据库的完整性控制 (301) 实验7 数据库的安全性 (302) 附录A SQL Server 2000中的系统存储过程 (303) 附录B Eclipse使用简介 (308)

<<数据库原理与应用>>

章节摘录

插图： 预防死锁通常有两种方法：一次封锁法和顺序封锁法。

一次封锁法是要求每个事务必须一次将所有要使用的数据全部加锁，否则就不能继续执行。

一次封锁法虽然可以有效防止死锁的发生，但需要将以后要用到的全部数据加锁，这势必扩大了封锁的范围，从而降低了系统的并发度。

同时，由于数据库中数据是不断变化的，原来不要求封锁的数据，在执行过程中可能会变成封锁对象，所以很难事先精确地确定每个事务所要封锁的数据对象。

顺序封锁法是预先对数据对象规定一个封锁顺序，所有事务都按这个顺序实行封锁。

顺序封锁法存在的主要问题是维护成本高。

数据库系统中可封锁的数据对象极其众多，并且随数据的插入、删除等操作而不断地变化，要维护这样极多而且变化的资源的封锁顺序非常困难，成本很高。

而且事务的封锁请求可以随着事务的执行而动态地决定，很难事先确定每一个事务要封锁哪些对象，因此也就很难按规定的顺序去施加封锁。

由此可见，在操作系统中广为采用的预防死锁的策略并不很适合数据库系统，因此DBMS在解决死锁问题上普遍采用的是诊断并解除死锁的方法。

这种方法允许数据库中出现死锁，尽可能及时发现和处理。

死锁的检测一般使用超时法或等待图法。

超时法中，如果一个事务的等待时间超过了规定的时限，则认为发生了死锁。

而等待图法则使用一个被称为“事务等待图”的有向图 $G=(T, U)$ 来判断是否存在死锁。

其中 T 为结点的集合，每个结点表示正在运行的事务； u 为边的集合，每条边表示事务等待的情况。

若事务 T_1 等待事务 T_2 ，则 T_1, T_2 之间有一条有向边，从 T_1 指向 T_2 。

如果发现图中存在回路，则表示系统中出现了死锁。

<<数据库原理与应用>>

编辑推荐

《数据库原理与应用》：高等学校计算机规划教材

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>