九章高校经典教材同歩辅导愚书

丛书 配套高教版•王珊、萨师嬪编著

*敎*你）11更多的11信血可未来,

**—书两用**

同步辅导+考研复习

数据库系统概论

（第5版）

同步辅导及习题全解

主编吴宗岱

**习题超全解 $■奇F於后**

名师一线经验**大汇集**，解题步骤**超详细**，方法技巧**最实用** *nfA*



让你的学习更简单!

**[www.waterpub.com.cn](http://www.waterpub.com.cn)**

高校经典教材同步辅导丛书

物理类 数学类经济类

■《通信原理（第七版）同步辅导及习题全解》酉蘋国防工^出版社樊昌信、曹丽娜主编

■《通信風里（第三版）同步辅导及习题全解》酉噎d啷版^炯槃、庞沁华、续大我、吴伟陵、杨鸿文主编

■《电路（第五版）同步辅导及习题全解》配套高教版邱关源主编

・《电路分析基?出（第四版• ±J»）同步辅导及习题全解》酉唸高教版李瀚茹主编

■《电路分析基础（第四版•下册）同步辅导及习题全解》配套高教版李瀚茹主编

■《信号与系统（第三版•上册）同步辅导及习题全解》配套高教版郑君里主编

■《信号与系统（第三版.下册）同步辅导及习题全解》配套高教版郑君里主编

■《信号与线性系统分析（第四版）同步辅导及习题全解》配套高教版吴大正主编

・《电工学电工技术（第七版•上册）同步辅导及习题全解》配套高教版秦曾煌主编

・《电工学电子技术（第七版•下册）同步辅导及习题全解》酉陰高教版秦曾煌主编

・《樹以电子技术基础（第四版）同步辅导及习题全解》酉味高教版童诗白主编

.《模^电子技术基础（第五版）同步辅导及习题全解》配套高教版童诗白主编

■《数字电子技术基础（第五版）同步辅导及习题全解》酉味高教版阎石主编

■《構以电子技术基础简明教程（第三版）同步辅导及习题全解》酉德高教版杨素行主编

■《数字电子技术基?出简明教程（第三版）同步辅导及习题全解》酉陰高教版余孟尝主编

■《电子技术基础（模拟部分•第五版）同步辅导及习题全解》酉高教版康华光主编

■《电子技术基础（数字部分•第五版）同步辅导及习题全解》配套高教版康华光主编

■《电子技术基础（模^部分•第六版）同步辅导及习题全解》配套高教版康华光主编

■《电子技术顎出（数字部分•第六版）同步辅导及习题全解》配套高教版康华光主编

■《数字输殳计（臉中与数字电路•第）同步辅导及习题全解》程高離王毓丰畦编

■《电磁场与电磁波（第四版）同步辅导及习题全解》酉味高教版谢处方、饶克谨主编

■《自动控制原理（第六版）同步辅导及习题全解》酉皤科学版胡寿松主编

■《计算机网络（第六版）同狗导及习题全解》配套电子工\*版^2±编



销售分类：数据库技术/数据库系统概论

高校经典教材同步辅导丛书

数据库系统概论（第**5**版）  
同步辅导及习题全解

主编吴宗岱

嚓吨

rS

•图书馆/

6Tb才圓衣1你史^風a

**1 [www.waterpub.com.cn](http://www.waterpub.com.cn)**

・北京・

内容提要

本书是与高等教育出版社岀版，王珊、萨师煩编著的《数据库系统概论》 （第5版）一书配套的同步辅导和习题解答辅导书。

本书共16章：绪论、关系数据库、关系数据库标准语言SQL、数据库安 全性、数据库完整性、关系数据理论、数据库设计、数据库编程、关系査询处 理和查询优化、数据库恢复技术、并发控制、数据库管理系统、数据库技术发 展概述、大数据管理、内存数据库系统、数据仓库与联机分析处理技术；按教 材内容安排全书结构，各章均包括知识网络图、重难点、知识点归纳、典型例 题、课后习题解答5部分内容；针对各章节习题给出详细解答，思路清晰、逻 辑性强，循序渐进地帮助读者分析并解决问题，内容详尽、简明易懂。

本书可作为高等院校学生学习“数据库系统概论”课程的辅导教材，也 可作为考研人员复习备考及教师备课命题的参考资料。

图书在版编目（C I P）数据

数据库系统概论（第5版）同步辅导及习题全解/吴 宗岱主编.一北京：中国水利水电出版社，2017. 10

（高校经典教材同步辅导丛书）

ISBN 978-7-5170-5960-8

I.①数…II.①吴…III.①数据库系统一高等学校 一教学参考资料IV.①TP311. 13

中国版本图书馆CIP数据核字（2017）第257174号

策划编辑：杨庆川 责任编辑：周益丹加工编辑：任红歌 封面设计：李佳

|  |  |
| --- | --- |
| 书 名  作 者  出版发行  经 售 | 高校经典教材同步辅导丛书  数据库系统槪论（第5版）同步辅导及习题全解  SHUJUKU XITONG GAILUN（ DI-WU BAN）TONGBU FUDAO JI XITI QUANJIE  主编吴宗岱  中国水利水电出版社  （北京市海淀区E渊潭南路1号D座100038）  网址： [www.wateq3ub.com.cn](http://www.wateq3ub.com.cn)  E-mail： [mchannel@263.net](mailto:mchannel@263.net) （万水）  sales@ waterpub .com.cn  电话：（010） 68367658 （营销中心）、82562819 （万水） 全国各地新华书店和相关出版物销售网点 |
| 版刷格次数价 排印规版印定 | 北京万水电子信息有限公司  北京正合鼎业印刷技术有限公司  170mm X 227mm 16 开本 13、印张 290 千字  2017年10月第1版 2017年10月第1次印刷 0001—5000 册  23.80 元 |

凡购买我社图书，如有缺页、倒页、脱页的，本社营销中心负责调换 版权所有•侵权必究

王珊、萨师煩编著的《数据库系统概论》（第5版）以体系完整、结构严谨、层次清 晰、深入浅岀等特点成为这门课程的经典教材，被全国许多院校采用。为了帮助读者 更好地学习这门课程、掌握更多知识，我们根据多年教学经验编写了这本配套辅导教 材，旨在帮助读者理解基本概念、掌握基本知识、学会基本解题方法与解题技巧，进而 提高应试能力。

本书作为一种辅助性教材，具有较强的针对性、启发性、指导性和补充性。考虑到 “数据库系统概论”这门课程的特点，我们在内容上作了以下安排：

1. 知识网络图。全面涵盖本章知识点，内容框架结构一目了解。
2. 重难点。每章前面均对本章知识要点进行了整理，综合众多参考资料，归纳了 本章几乎所有的考点，便于读者学习与复习。
3. 知识点归纳。对每章知识点做了简练概括，梳理了各知识点之间的脉络联系， 突出各章节主要定理及重要公式，使读者在学习过程中目标明确、有的放矢。
4. 典型例题。该部分选取了一些启发性或综合性较强的经典例题,对所给例题先 进行分析，再给出详细解答，意在抛砖引玉。
5. 课后习题解答。教材中课后习题丰富、层次多样,许多基础性问题从多个角度 帮助学生理解基本概念和基本理论，促其掌握基本解题方法。我们对教材的课后习题 给出了详细解答。

由于时间仓促及编者水平有限,书中难免有疏漏之处，敬请各位同行和读者给予 批评指正（邮箱:yapai2004@126. com 或微信:JZCS15652485156） o

编者

2017年8月

第一章绪论 1

[知识网络图 1](#bookmark21" \o "Current Document)

[重难点 1](#bookmark39" \o "Current Document)

[知识点归纳 2](#bookmark60" \o "Current Document)

典型例题 …… 7

课后习题解答 ••••• - •• 9

■第二章关系数据库 17

知识网络图 17

重难点 17

知识点归纳 18

[典型例题 20](#bookmark101" \o "Current Document)

[课后习题解答 22](#bookmark107" \o "Current Document)

■第三章 关系数据库标准语言SQL 25

[知识网络图 25](#bookmark47" \o "Current Document)

[重难点 26](#bookmark147" \o "Current Document)

知识点归纳 26

典型例题 33

课后习题解答 36

■第四章 数据库安全性 41

知识网络图 41

重难点 42

知识点归纳 42

典型例题 49

课后习题解答 50

第五章数据库完整性 55

知识网络图 55

重难点 55

知识点归纳 56

典型例题 58

课后习题解答 61

員第六章 关系数据理论 63

知识网络图 63

重难点 64

知识点归纳 64

典型例题 68

课后习题解答 72

第七章数据库设计 78

知识网络图 78

重难点 79

知识点归纳 79

典型例题 85

课后习题解答 91

■第八章 数据库编程 95

知识网络图 95

重难点 96

知识点归纳 96

典型例题 110

课后习题解答 V 112

■第九章 关系查询处理和查询优化 123

知识网络图 123

重难点 124

知识点归纳 124

典型例题 127

课后习题解答 129

■第十章 数据库恢复技术 133

知识网络图 133

重难点 134

知识点归纳 134

典型例题 141

课后习题解答 142

■第十一章并发控制 147

知识网络图 147

重难点 148

知识点归纳 148

典型例题 150

课后习题解答 153

■第十二章数据库管理系统 163

知识网络图 163

重难点 164

知识点归纳 164

典型例题 166

课后习题解答 170

contents

J第十三章数据库技术发展概述 175

知识网络图 175

重难点 175

知识点归纳 176

典型例题 178

课后习题解答 179

湧第十四章大数据管理 181

知识网络图 181

重难点 182

知识点归纳 182

典型例题 •-•••••• 184

课后习题解答 185

|第十五章内存数据库系统 187

知识网络图 187

重难点 7 187

知识点归纳 188

典型例题 189

课后习题解答 191

g第十六章 数据仓库与联机分析处理技术 193

知识网络图 193

重难点 194

知识点归纳 194

典型例题 196

课后习题解答 v 198

知识网络图

数据库的4个基本概念

数据库系统概述〈

数据管理技术的产生

数据管理技术的发展

I数据库系统的特点

［两类数据模型

概念模型

组成要素 绪论（数据模型｛常用的数据模型

层次模型

网状模型

关系模型

［数据库系统模式的概念

数据库系统结构＜

三级模式结构

二级映像功能

数据独立性

重难点

1. DB.DBMS和DBS的定义
2. 数据管理技术的发展阶段

数据库系统概论(第**5**版)同步辅导及习题全解

人工管理阶段、文件系统阶段、数据库系统阶段的特点。

1. 数据描述

概念设计、逻辑设计和物理设计等各阶段中数据描述的术语，概念设计中实体间二元联系的描 述(1 : 1,1 : n,m : n)。

1. 数据模型

数据模型的定义,两类数据模型，逻辑模型的形式定义,E-R模型，层次模型、网状模型、关系模 型和面向对象模型的数据结构以及联系的实现方式。

1. 数据库的体系结构

三级结构，二级映像，二级数据独立性。

1. 数据库管理系统

数据库管理系统的工作模式、主要功能和模块组成。

1. 数据库系统

数据库系统的组成,数据库，数据库系统的全局结构，数据库系统结构的分类。

1. 数据库系统的特点
2. 面向全组织的复杂的数据结构。
3. 数据的冗余度小，易扩充。
4. 具有较高的数据和程序的独立性。
5. 统一的数据控制功能，数据共享程度高。

知识点归纳

. 一、数据库系统概述

1. 数据、数据库、数据库管理系统、数据库系统

数据(Data)：数据库中存储的基本对象。描述事物的符号记录称为数据。

数据的表现形式不能完全表达其内容,需要经过解释，数据和关于数据的解释(语义)是不可 分的。

数据库(DB)：严格地讲，数据库是长期储存在计算机内、有组织的、可共享的大量数据的集合。

数据库中的数据按一定的数据模型组织、描述和储存，具有较小的冗余度、较高的数据独立性 和易扩展性，并可为各种用户共享。

数据库中的数据具有三个特点:永久存储、有组织、可共享。

数据库管理系统(DBMS)：位于用户与操作系统之间的一层数据管理软件。用于科学地组织和 存储数据、高效地获取和维护数据

主要功能：

（1） 数据定义功能。 \

（2） 数据组织、存储和管理。

（3） 数据操纵功能。

（4） 数据库的事务管理和运行管理。

（5） 数据库的建立和维护功能。

（6） 其他功能（DBMS与网络中其他软件系统的通信功能;DBMS与另一个DBMS或文件系统 的数据转换功能;异构数据库之间的互访和互操作功能等）。

数据库系统（DBS）：在计算机系统中引入数据库后的系统,一般由数据库、数据库管理系统（及 其开发工具）、应用系统、数据库管理员（DBA）构成。

1. 数据管理技术的产生和发展

对数据进行分类、组织、编码、存储、检索和维护，是数据处理的中心问题。

数据管理技术经历了人工管理、文件系统、数据库系统3个阶段。

1. 数据库系统的特点

数据库系统与人工管理和文件系统相比的特点有：

（1） 数据结构化

数据库系统实现整体数据的结构化，这是数据库系统与文件系统的本质区别。

所谓整体数据的结构化指数据不仅针对某一应用，而是面向全组织，且数据之间是具有联 系的。

（2） 数据的共享性高，冗余度低、易扩充

数据共享可以大大减少数据冗余,节约存储空间，还能避免数据之间的不相容性与不一致性。

（3） 数据独立性高

包括数据的物理独立性和数据的逻辑独立性。

数据独立性由DBMS的二级映像功能来保证。

（4） 数据由DBMS统一管理和控制

由于数据库的共享是并发的共享，即多个用户可以同时存取数据库中的数据甚至是存储同一 个数据，因而DBMS必须提供以下几方面的数据控制功能：

1. 数据的安全性保护
2. 数据的完整性检査
3. 并发控制
4. 数据库恢复

■二、数据模型

数据库系统概论(第**5**版)同步辅导及习题全解

1. 数据模型是对现实世界数据特征的抽象，它是用来描述数据、组织数据和对数据进行操 作的。

根据模型应用的不同目的,数据模型划分为两类：

第一类是概念模型。也称信息模型，是按照用户的观念对数据和信息建模，主要用于数据库 设计。

第二类是逻辑模型和物理模型。

1. 概念模型

概念模型用于信息世界的建模，是现实世界到信息世界的第一层抽象。

信息世界中的基本概念。

1. 最常用的数据模型

层次模型和网状模型统称为格式化模型。

在格式化模型中，实体用记录表示，实体的属性对应记录的数据项(或字段)。实体之间的联系 在格式化模型中转换成记录之间的两两联系。

格式化模型中的数据结构的单位是基本层次联系。所谓基本层次联系是指两个记录以及它们 之间的一对多的联系。

1. 层次模型

层次模型用树形结构来表示各类实体以及实体间的联系。

层次模型的数据结构：

满足以下两个条件的基本层次联系的集合为层次模型：

1. 有且只有一个结点没有双亲结点，这个结点称为根结点；
2. 根以外的其他结点有且只有一个双亲结点。

层次模型的一个基本特点：

任何一个给定的记录纸只有按其路径查看时，才能显示出它的全部意义，没有一个子女记录纸 能脱离双亲记录纸而独立存在。

多对多联系在层次模型中的表示：

层次模型中表示多对多联系，需将多对多联系分解成一对多联系，分解方法有两种:冗余结点 法和虚拟结点法。

层次模型的优缺点：

优点:数据结构比较简单清晰;查询效率高;提供了良好的完整性支持。

缺点:现实世界中很多联系是非层次性的，一个结点具有多个双亲等，层次模型表示这类联系 的方法很笨拙,对插入和删除操作的限制比较多;查询子女结点必须通过双亲结点；由于结构严密， 层次命令趋于程序化。

1. 网状模型

网状模型的数据结构： 、

满足以下两个条件的基本层次联系集合称为网状模型:允许一个以上的结点无双亲;一个结点 可以有多于一个的双亲。

网状模型中要为每个联系命名，并指出与该联系有关的双亲记录和子女记录。

网状数据模型的优缺点：

优点:能更为直接地描述现实世界;具有良好的性能，存取效率较高。

缺点:结构比较复杂，不利于最终用户掌握;网状模型的DDL.DML复杂,并且要嵌入某一种高 级语言中。用户不容易掌握,不容易使用。

1. 关系模型

关系模型的数据结构：

从用户观点看，关系模型由一组关系组成。每个关系的数据结构是一张规范化的二维表。

关系模型中的一些术语：

关系:一个关系对应一张表；

元组:表中的一行即为一个元组；

属性:表中的一列即为一个属性；

码:也称为码键。表中的某个属性组，它可以唯一确定一个元组，也就称为本关系的码；

域:属性的取值范围；

分量:元组中的一个属性值；

关系模式:对关系的描述，一般表示为关系名（属性1，属性2,…,属性n）

关系的每一个分量必须是一个不可分的数据项，也就是说,不允许表中还有表。

关系模型的优缺点：

优点:建立在严格的数学概念的基础上;概念单一,数据结构简单、清晰，用户易懂易用;存取路 径对用户透明，从而具有更高的数据独立性、更好的安全保密性，简化了程序员的工作和数据库开 发建立的工作。

缺点：由于存取路径对用户透明，查询效率不如格式化模型。

.三、数据库系统结构

1. 从数据库管理系统角度看，数据库系统通常采用三级模式结构,这是数据库管理系统内部的 系统结构;从数据库最终用户角度看,数据库系统的结构分为单用户结构、主从式结构、分布式结 构、客户/服务器、浏览器/应用服务器/数据库服务器多层机构等，这是数据库系统外部的体系

结构。

数据库系统概论(第**5**版)同步辅导及习题全解

1. 数据库系统的三级模式结构

数据库系统的三级模式结构是指数据库系统是由外模式、模式和内模式三级构成。

模式:模式也称逻辑模式，是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共 数据视图。

定义模式时不仅要定义数据的逻辑结构,而且要定义数据之间的联系，定义与数据有关的安全 性、完整性要求。

外模式:也称为子模式或用户模式，它是数据库用户能够看见和使用的局部数据的逻辑结构和 特征的描述，是数据库用户的数据视图，是与某一应用有关的数据的逻辑表示。

一个数据库可以有多个外模式；同一外模式可以为某一用户的多个应用系统所使用，但一个应 用程序只能使用一个外模式。

内模式:也称存储模式，它是数据物理结构和存储方式的描述，是数据在数据库内部的表示 方式。

一个数据库只有一个内模式。

1. 数据库的二级映像功能与数据独立性

外模式/模式映像:对于每一个外模式，都有一个外模式/模式映像，这些映像定义通常包含在 各自外模式的描述中。

当模式改变时，由数据库管理员对外模式/模式映像作相应改变，可以使外模式保持不变。应 用程序是依据数据的外模式编写的，从而应用程序不必修改，保证了数据与程序的逻辑独立性，简 称数据的逻辑独立性。

模式/内模式映像:模式/内模式映像是唯一的，它定义了数据全局逻辑结构与存储结构之间的 对应关系，该映像定义通常包含在模式描述中。

当数据库的存储结构改变了，由数据库管理员对模式/内模式映像作相应改变,可以使模式保 持不变，从而应用程序也不必改变。保证了数据与程序的物理独立性，简称数据的物理独立性。

数据与程序之间的独立性，使得数据的定义和描述可以从应用程序中分离出去。

1. 数据库系统的组成

数据库管理员(DBA)：

负责全面管理和控制数据库系统，具体职责包括：

决定数据库中的信息内容和结构;决定数据库的存储结构和存储策略;定义数据的安全性要求 和完整性约束条件;监控数据库的使用和运行;数据库的改弛和重组重构。

系统分析员和数据库设计人员:

系统分析员负责应用程序的需求分析和规范说明，和用户及数据库管理员相结合，确定系统的 硬件、软件配置，并参与数据库系统的概要设计。

数据库设计人员负责数据库中数据的确定、数据库各级模式的设计。数据库设计人员必须参 加用户需求调查和系统分析，然后进行数据库设计。

应用程序员： \

应用程序员负责设计和编写应用系统的程序模块，并进行调试和安装。

用户：

最终用户通过应用系统的用户接口使用数据库，最终接口方式有浏览器、菜单驱动、表格操作、 图形显示、报表书写等。

最终用户可分为三类:偶然用户、简单用户、复杂用户。

典型例题

例**1**什么是数据库？

解析数据库是长期存储在计算机内、有组织的、可共享的大量数据的集合。数据库中的数据是按 某种数据模型进行组织、描述和存放在外存储器上，且可被多个用户同时使用。因此，数据 库具有较小的冗余度、较高的数据独立性和易扩展性。

例**2** 什么是数据库的数据独立性？

解析数据独立性表示应用程序与数据库中存储的数据不存在依赖关系，包括逻辑数据独立性和 物理数据独立性。

逻辑数据独立性是指局部逻辑数据结构（外视图，即用户的逻辑文件）与全局逻辑数据结构 （概念视图）之间的独立性。当数据库的全局逻辑数据结构（概念视图）发生变化（数据定义 的修改、数据之间联系的变更或增加新的数据类型等）时，它不影响某些局部的逻辑结构的 性质,应用程序不必修改。

物理数据独立性是指数据的存储结构与存取方法（内视图）改变时，对数据库的全局逻辑数 据结构（概念视图）和应用程序不必作修改的一种特性，也就是说，数据库数据的存储结构与 存取方法独立。

例**3**什么是数据库管理系统？

解析数据库管理系统（DBMS）是操纵和管理数据库的一组软件，它是数据库系统（DBS）的重要组 成部分。不同的数据库系统都配有各自的DBMS,而不同的DBMS各支持一种数据库模型, 虽然它们的功能强弱不同，但大多数DBMS的构成相同、功能相似。

一般说来,DBMS具有定义、建立、维护和使用数据库的功能，它通常由三部分构成:数据描 述语言及其翻译程序、数据操纵语言及其处理程序和数据库管理的例行程序。

例**4**假设教学管理规定：

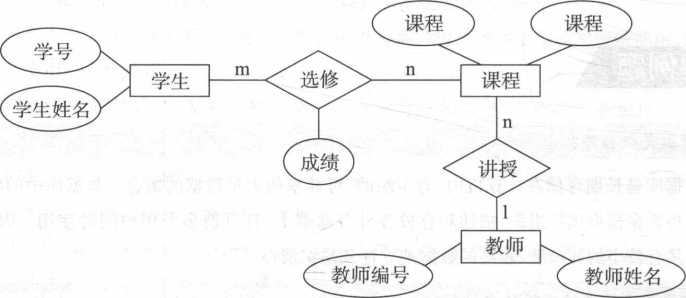
数据库系统概论(第**5**版)同步辅导及习题全解

1. 一个学生可选修多门课，一门课有若干学生选修；
2. 一个教师可讲授多门课,一门课只有一个教师讲授；
3. 一个学生选修一门课，仅有一个成绩。

学生的属性有学号、学生姓名；教师的属性有教师编号，教师姓名；课程的属性有课程号、课 程名。

要求:根据上述语义画出**E-R**图，要求在图中画出实体的属性并注明联系的类型。

解析如图1.1所示。

图1.1

例**5** 某系有若干个课程组，每个课程组有若干位教师，每个教师可参加若干个课程组，每个课程 组管理若干门课程，每门课程只属于一个课程组。教师有工号、姓名、职称的属性，课程组 有名称、专业方向的属性，课程有名称、学时、考核方式的属性。

请根据给定语义画出**E-R**图，并在图上注明联系类型。

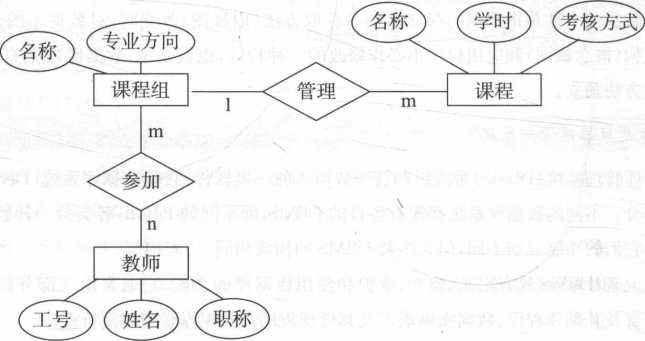
解析如图1.2所示。

图1.2

课后习题解答

1.!解题陲\ （1）数据（Data）：描述事物的符号记录称为数据。数据的种类有数字、文字、图形、 图像、声音、正文等。

解析在现代计算机系统中数据的概念是广义的。早期的计算机系统主要用于科 学计算，处理的数据是整数、实数、浮点数等传统数学中的数据。现代计算机能存储 和处理的对象十分广泛，表示这些对象的数据也越来越复杂。数据与其语义是不可 分的。500这个数字可以表示一件物品的价格是500元，也可以表示一个学术会议 参加的人数有500人，还可以表示一袋奶粉重500克。

（2） 数据库（DataBase, DB）：数据库是长期储存在计算机内的、有组织的、可共享的 数据集合。数据库中的数据按一定的数据模型组织、描述和储存，具有较小的冗余 度、较高的数据独立性和易扩展性，并可为各种用户共享。

（3） 数据库系统（DataBase System, DBS）：数据库系统是指在计算机系统中引入数 据库后的系统构成,一般由数据库、数据库管理系统（及其开发工具）、应用系统、数 据库管理员构成。

解析数据库系统和数据库是两个概念。数据库系统是一个人机系统，数据库是 数据库系统的一个组成部分。但是在日常工作中人们常常把数据库系统简称为数 据库。希望读者能够从人们讲话或文章的上下文中区分“数据库系统”和“数据库”, 不要引起混淆。

（4） 数据库管理系统（DataBase Management System,DBMS）:数据库管理系统是位 于用户与操作系统之间的一层数据管理软件，用于科学地组织和存储数据、高效地 获取和维护数据。DBMS的主要功能包括数据定义功能、数据操纵功能、数据库的 运行管理功能、数据库的建立和维护功能。

［知识点播：数据管理技术经历了人' J工管理、文件系统、数据库系统三 个阶段。数据库系统是在文件系统 的基础上发展而成的，同时又克服 了文件系统的三个缺陷：数据的冗 余、不一致性和联系弱。

解析DBMS是一个大型的复杂的软件系

统，是计算机中的基础软件。目前,专门研

制DBMS的厂商及其研制的DBMS产品有

很多。

使用数据库系统的好处是由数据库管理系统的特点或优点决定的。使用数据库系

统的好处很多，例如，可以大大提高应用开发的效率、方便用户的使用、减轻数据库 系统管理人员维护的负担等等。使用数据库系统可以大大提高应用开发的效率。 因为在数据库系统中应用程序不必考虑数据的定义、存储和数据存取的具体路径, 这些工作都由DBMS来完成。用一个通俗的比喻,使用了 DBMS就如有了一个好

参谋、好助手，许多具体的技术工作都由这个助手来完成。开发人员就可以专注于 应用逻辑的设计,而不必为数据管理的许许多多复杂的细节操心。还有，当应用逻 辑改变，数据的逻辑结构也需要改变时，由于数据库系统提供了数据与程序之间的 独立性，数据逻辑结构的改变是DBA的责任，开发人员不必修改应用程序,或者只 需要修改很少的应用程序，从而既简化了应用程序的编制，又大大减少了应用程序 的维护和修改。使用数据库系统可以减轻数据库系统管理人员维护系统的负担。 因为DBMS在数据库建立、运用和维护时对数据库进行统一的管理和控制，包括数 据的完整性、安全性、多用户并发控制、故障恢复等，都由DBMS执行。总之，使用 数据库系统的优点是很多的，既便于数据的集中管理、控制数据冗余、提高数据的利 用率和一致性，又有利于应用程序的开发和维护。

数据库系统概论(第**5**版)同步辅导及习题全解

I过程

文件系统与数据库系统的区别是:文件系统面向某一应用程序、共享性差、冗余度 大、数据独立性差、记录内有结构、整体无结构、由应用程序自己控制。数据库系统 面向现实世界、共享性高、冗余度小、具有较高的物理独立性和一定的逻辑独立性, 整体结构化、用数据模型描述、由数据库管理系统提供数据的安全性、完整性、并发 控制和恢复能力。文件系统与数据库系统的联系是:文件系统与数据库系统都是计 算机系统中管理数据的软件。

解析 文件系统是操作系统的重要组成部分;而DBMS是独立于操作系统的软件。

但是DBMS是在操作系统的基础上实现的;数据库中数据的组织和存储是通过操 作系统中的文件系统来实现的。

**4**.區题gk （1）适用于文件系统而不是数据库系统的应用例子:数据的备份、软件或应用程序使 用过程中的临时数据存储一般使用文件比较合适。早期功能比较简单、比较固定的 应用系统也适合用文件系统。

（2）适用于数据库系统而非文件系统的应用例子：目前,几乎所有企业或部门的信息 系统都以数据库系统为基础，都使用数据库。例如，一个工厂的管理信息系统（其中 会包括许多子系统，如库存管理系统、物资采购系统、 j小零示：在数据库领域，'

作业调度系统、设备管理系统、人事管理系统等）、学郭鷲齬'爲 校的学生管理系统、人事管理系统、图书馆的图书管 捐命系畐意义

理系统等等，都适合用数据库系统。

**5**.陸慮亀数据库系统的主要特点有：

（1）数据结构化。数据库系统实现整体数据而结构化，这是数据库的主要特征之一, 也是数据库系统与文件系统的本质区别。

解析 注意这里的“整体”两个字。在数据库系统中，数据不再针对某一个应用，而 是面向全组织，具有整体的结构化。不仅数据是结构化的，而且数据的存取单位即一次可以存取数据的大小也很灵活，可以小到某一个数据项（如一个学生的姓名）， 大到一组记录（如成千上万个学生记录）。而在文件系统中，数据的存取单位只有一 个记录（如一个学生的完整记录）。

（2） 数据的共享性高，冗余度低，易扩充。数据库的数据不再面向某个应用而是面向 整个系统,因此可以被多个用户、多个应用以多种不同的语言共享使用。由于数据 面向整个系统，是有结构的数据，不仅可以被多个应用共享使用，而且容易增加新的 应用,这就使得数据库系统弹性大、易于扩充。解析数据共享可以大大减少数据冗 余、节约存储空间，同时还能够避免数据之间的不相容性与不一致性。所谓“数据面 向某个应用”是指数据结构是针对某个应用设计的，只被这个应用程序或应用系统 使用，可以说数据是某个应用的“私有资源”。所谓“弹性大”是指系统容易扩充也容 易收缩，即应用增加或减少时不必修改整个数据库的结构，只需做很少的改动。可 以取整体数据的各种子集用于不同的应用系统，当应用需求改变或增加时，只要重 新选取不同的子集或加上一部分数据，便可以满足新的需求。

11

（3） 数据独立性高。数据独立性包括数据的物理独立性和数据的逻辑独立性。数据 库管理系统的模式结构和二级映像功能保证了数据库中的数据具有很高的物理独 立性和逻辑独立性。

（4） 数据由DBMS统一管理和控制。数据库的共享是并发的共享，即多个用户可以 同时存取数据库中的数据甚至可以同时存取数据库中同一个数据。为此，DBMS 必须提供统一的数据控制功能，包括数据的安全性保护、数据的完整性检查、并发控 制和数据库恢复。

解析 DBMS数据控制功能包括四个方面。数据的安全性保护:保护数据以防止不 合法的使用造成的数据泄密和破坏;数据的完整性检查:将数据控制在有效的范围 内,或保证数据之间满足一定的关系;并发控制:对多用户的并发操作加以控制和协 调，保证并发操作的正确性;数据库恢复:当计算机系统发生硬件故障、软件故障，或 者由于操作员的失误以及故意的破坏影响数据库中数据的正确性，甚至造成数据库 部分或全部数据的丢失时，能将数据库从错误状态恢复到某一已知的正确状态（亦 称为完整状态或一致状态）。下面可以得到“什么是数据库”的一个定义:数据库是 长期存储在计算机内有组织的大量的共享的数据集合，它可以供各种用户共享，具 有较小冗余度和较高的数据独立性。DBMS在数据库建立、运用和维护时对其进行 统一控制，以保证数据的完整性、安全性,并在多用户同时使用数据库时进行并发控 制，在发生故障后对系统进行恢复。数据库系统的出现使信息系统从以加工数据的 程序为中心转向围绕共享的数据库为中心的新阶段。

（1）数据定义功能；

（2） 数据存取功能；

数据库系统概论(第**5**版)同步辅导及习题全解

（3） 数据库运行管理功能；

（4） 数据库的建立和维护功能。

1. 匯厘函k概念模型实际上是现实世界到机器世界的一个中间层次。概念模型用于信息世界

的建模，是现实世界到信息世界的第一层抽象,是数据库设计人员进行数据库设计 的有力工具，也是数据库设计人员和用户之间进行交流的语言。

1. 庭函k实体:客观存在并可以相互区分的事物叫实体。实体型:具有相同属性的实体具有

相同的特征和性质，用实体名及其属性名集合来抽象和刻画同类实体,称为实体型。 实体集：同型实体的集合称为实体集。属性:实体所具有的某一特性,一个实体可由 若干个属性来刻画。实体之间的联系:用菱形表示,菱形框内写明联系名，并用无 向边分别与有关实体连接起来，同时在无向边旁标上联系的类型（1 ： 1,1 ： n或 m ： n）。

1. 匯厘数据模型是数据库中用来对现实世界进行抽象的工具,是数据库中用于提供信息表

示和操作手段的形式构架。一般地讲，数据模型是严格定义的概念的集合。这些概 念精确描述了系统的静态特性、动态特性和完整性约束条件。因此数据模型通常由 数据结构、数据操作和完整性约束三部分组成。

（1） 数据结构:是所研究的对象类型的集合,是对系统静态特性的描述。

（2） 数据操作:是指对数据库中各种对象（型）的实例（值）允许进行的操作的集合，包 括操作及有关的操作规则,是对系统动态特性的描述。

（3） 数据的约束条件:是一组完整性规则的集合。完整性规则是给定的数据模型中

数据及其联系所具有的制约和依存规则,用以限定符合数据模型的数据库状态以及 状态的变化，以保证数据的正确、有效、相容。解析数据模型是数据库系统中最重要 的概念之一。数据模型是数据库系统的基础。任何一个DBMS都以某一个数据模 型为基础，或者说支持某一个数据模型。数据库系统中，模型有不同的层次。根据 模型应用的不同目的，可以将模型分成两类或者说两个层次:一类是概念模型，是按 用户的观点来对数据和信息建模,用于信息世界的建模，强调语义表达能力，概念简 单清晰;另一类是数据模型，是按计算机系统的观点对数据建模,用于机器世界,人 们可以用它定义、操纵数据库中的数据，一般 ［小提示：数据模型是对现实世界进 需要有严格的形式化定义和-组严格定\*了 V囂農的讀波于聽蠶鷲 语法和语义的语言,并有一些规定和限制，便 ［据约束等方面内容〔

于在机器上实现。

1. 邂题陸' 层次模型是数据库系统中最早岀现的数据模型，层次数据库系统采用层次模型作

为数据的组织方式。层次模型是用树形结构来表示各类实体以及实体间的联系的

数据模型。

（1）教员学生层次数据库模型（如图1. 3所示）

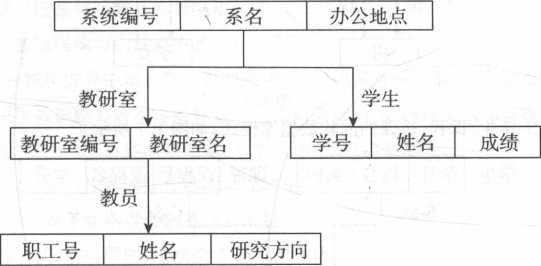
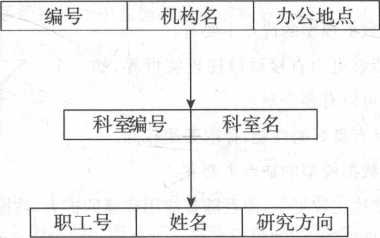


图1.3

（2）行政机构层次数据库模型（如图1. 4所示）



行政机构

科室

职工

图1.4

（3）行政区域层次数据库模型（如图1.5所示）

国家



自治区直辖市

图1.5

1. **®**题幽，把满足以下两个条件的基本层次联系集合称为网状模型：

（1） 允许一个以上的结点无双亲。

（2） 一个结点可以有多于一个的双亲。

例：1）网状模型的两个结点之间有多种联系（如图1.6所示）

父母

子女

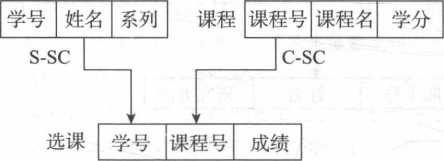
图1.6

数据库系统概论(第**5**版)同步辅导及习题全解

**2.**

**IX**

2）学生/课程/选课的网状数据库模式（如图1.7所示）

图1.7

学生

f小提示：数据模型分成概念模型和

I逻辑模型两大类。前•者的代表是实 \ 体联系模型，后者的代表是层次、 网状、关系和面向对象模型。关系 模型是当今的主流模型，面向对象 模型是今后发展的方向。

网状数据模型的优点主要有：

（1） 能够更为直接地描述现实世界，如一个 结点可以有多个双亲。

（2） 具有良好的性能，存取效率较高。

网状数据模型的缺点主要有：

（1）结构比较复杂，而且随着应用环境的扩大，数据库的结构就变得越来越复杂，不 利于最终用户掌握。

⑵其数据定义语言（DDL）、数据操作语言（DML）复杂，用户不容易使用。

（3） 数据独立性较差。由于记录之间联系是通过存取路径实现的，应用程序在访问 数据时必须选择适当的存取路径，因此，用户必须了解系统结构的细节，加重了编 写应用程序的负担，从而影响数据独立性。

层次数据模型的优点主要有：

（1） 层次数据模型本身比较简单，层次分明，便于在计算机内实现。

（2） 在层次数据结构中，从根结点到树中任一结点均存在一条唯一的层次路径，为 有效地进行数据操纵提供条件。

（3） 由于层次结构规定除根结点外所有结点有且仅有一个双亲，故实体集之间的联 系可用双亲结点唯一地表示，并且层次数据模型中的基本层次联系总是从双亲记 录指向子女记录，所以记录类型之间的联累名可省略。由于实体集间的联系固定, 所以层次数据模型DBMS对层次结构的数据有较高的处理效率。

（4） 层次数据模型提供了良好的完整性支持。

（5） 实体间联系是固定的，且预先定义好的应用系统，采用层次数据模型来实现，其

性能优于关系数据模型，不低于网状数据模型。

可见用层次数据模型对具有一对多的层次关系的部门描述非常自然、直观，容易理 解。这是层次数据库的卖出优点。

层次数据模型的缺点主要有：

（1） 现实世界中很多联系是非层次性的，如多对多联系、一个结点具有多个双亲等， 层次数据模型表示这类联系的方法很笨拙，只能通过引入冗余数据（易产生不一致 性）或创建非自然的数据组织（引入虚拟结点）来解决。

（2） 对插入和删除操作的限制比较多。

（3） 查询子女结点必须通过双亲结点。

（4） 由于结构严密，层次命令趋于程序化。

.（1）关系 （2）属性 （3）域 （4）元组 （5）主码 （6）分量 （7）关系模式

关系数据模型由关系数据结构、关系操作集合和关系完整性约束三部分组成。在 用户观点下，关系数据模型中数据的逻辑结构是一张二维表，它由行和列组成。

（1）关系:一个关系对应通常说的一张表；（2）属性:表中的一列即为一个属性；（3） 域:属性的取值范围；（4）元组:表中的一行即为一个元组；（5）主码:表中的某个属 性组,它可以唯一确定一个元组；（6）分量:元组中的一个属性值；（7）关系模式:对 关系的描述，一般表示为关系名（属性1,属性2,…，属性n）

关系数据模型具有下列优点：（1）关系数据模型与非关系数据模型不同，它是建立 在严格的数学概念的基础上的；（2）关系数据模型的概念单一，无论实体还是实体 之间的联系都用关系表示,操作的对象和操作的结果都是关系，所以其数据结构简 单、清晰，用户易懂易用；（3）关系数据模型的存取路径对用户透明，从而具有更高 的数据独立性、更好的安全保密性，也简化了程序员的工作和数据库开发建立的工 作。当然，关系数据模型也有缺点,其中最主要的缺点是，由于存取路径对用户透 明，查询效率往往不如非关系数据模型。因此为了提高性能,必须对用户的查询请 求进行优化，增加了开发数据库管理系统的难度。

.MSBk数据库系统的三级模式结构由外模式、模式和内模式组成。外模式，亦称子模式或 用户模式,是数据库用户（包括应用程序员和最终用户）能够看见和使用的局部数 据的逻辑结构和特征的描述，是数据库用户的数据视图，是与某一应用有关的数据 的逻辑表示。模式,亦称逻辑模式，是数据库中全体数据的逻辑结构和特征的描 述，是所有用户的公共数据视图。模式描述的是数据的全局逻辑结构。外模式涉 及的是数据的局部逻辑结构，通常是模式的子集。内模式，亦称存储模式,是数据 在数据库系统内部的表示，即对数据的物理结构和存储方式的描述。数据库系统 的三级模式是对数据的三个抽象级别，它把数据的具体组织留给DBMS管理，使

用户能逻辑抽象地处理数据，而不必关心数据在计算机中的表示和存储。为了能 够在内部实现这三个抽象层次的联系和转换,数据库系统在这三级模式之间提供 了二层映像:外模式/模式映像和模式/内模式映像。正是这二层映像保证了数据 库系统中的数据能够具有较高的逻辑独立性和物理独立性。

数据库系统概论(第**5**版)同步辅导及习题全解

1. 陸画**L**外模式、内模式，亦称逻辑模式，是数据库中全体数据的逻辑结构和特征的描述,是

所有用户的公共数据视图。模式描述的是数据的全局逻辑结构。外模式涉及的是 数据的局部逻辑结构，通常是模式的子集。内模式，亦称存储模式,是数据在数据 库系统内部的表示，即对数据的物理结构和存储方式的描述。数据定义语言是用 来定义数据库模式、外模式、内模式的语言。数据操纵语言是用来对数据库中的数 据进行查询、插入、删除和修改的语言。

1. 癖，瞄区 数据与程序的逻辑独立性:当模式改变时（例如增加新的关系、新的属性、改变属性

的数据类型等），由数据库管理员对各个外模式/模式的映像做相应改变，可以使外 模式保持不变。应用程序是依据数据的外模式编写的，从而应用程序不必修改，保 证了数据与程序的逻辑独立性，简称数据的逻辑独立性。

数据与程序的物理独立性:当数据库的存储结构改变了，由数据库管理员对模式/ 内模式映像做相应改变，可以使模式保持不 变，从而应用程序也不必改变，保证了数据与 程序的物理独立性，简称数据的物理独立性。

数据库管理系统在三级模式之间提供的二层

映像保证了数据库系统中的数据能够具有较高的逻辑独立性和物理独立性。

［小提示：由于三级结构之间往往 T： -I差别很大，存在着二级映象，因 此使数据库系统具有较高的数据 独立性：物理数据独立性和逻辑 、数据独立性。

**18** .陸曲，数据库系统一般由数据库、数据库管理系统（及其开发工具）、应用系统、数据库管 理员和用户构成。

**19**.陸距，数据库管理员负责全面地管理和控制数据库系统。具体职责包括：①决定数据库 的信息内容和结构；②决定数据库的存储结构和存取策略；③定义数据的安全 性要求和完整性约束条件；④监督和控制数据库的使用和运行；⑤改进和重组 数据库系统。

系统分析员负责应用系统的需求分析和规范说明，系统分析员要和用户及DBA相 结合，确定系统的硬件、软件配置，并参与数据库系统的概要设计。

数据库设计人员负责数据库中数据的确定、数据库各级模式的设计。数据库设计 人员必须参加用户需求调查和系统分析，凍后进行数据库设计。在很多情况下，数 据库设计人员就由数据库管理员担任。

应用程序员负责设计和编写应用系统的程序模块，并进行调试和安装。

**7**

**1—**

关系数据库

二章

关系数据库

知识网络图

［关系模型

关系数据结构及形式化定义＜

关系模式

关系数据库

〔关系模型的存储结构

实体完整性

关系数据库＜

关系的完整性＜

参照完整性

〔用户定义的完整性

关系代数

J传统的集合运算  
［专门的关系运算

J元组关系演算语言

1域关系演算语言

重难点

重点:掌握关系数据结构及其定义;关系的三类完整性约束的概念。需要举一反三的是:关系 代数（包括抽象的语言及具体的语言）；关系代数中的各种运算（包括并、交、差、选择、投影、连接、 除、及广义笛卡尔积等）、元组关系演算语言ALPHA及域关系演算语言QBE等，能够使用这些语 言完成各种数据操做。

难点：由于关系代数较为抽象,因此在学习的过程中一定要结合具体的实例进行学习。同时,要注意把握由具体语言到抽象语言的原则，即通过对具体语言如ALPHA和QBE的学习过渡到对 抽象的关系演算的把握。关系数据库系统是目前使用最广泛的数据库系统。了解关系数据库系统 与非关系数据库系统的区别。

■ 一、关系

数据库系统概论(第**5**版)同步辅导及习题全解

知识点归纳

1. 域(domain)：域是一组具有相同数据类型的值的集合。
2. 笛卡尔积(cartesian product)：笛卡尔积是域上面的一种集合运算。笛卡尔积可表示为一个 二维表,表中的每一行对应一个元组,表中的每一列的值来自一个域。

R： n目关系,kl个元组;S： m目关系,k2个元组;RXS0

1. 关系(relation)

候选码、主码、外部码的概念

候选码:若关系中的某一属性组的职能唯一地标识一个元组,则称该属性组为候选码(candidate key)。

候选码的诸属性称为主属性(prime attribute),不包含在任何候选码中的属性称为非主属性 (non—prime attribute)或3E码属'性(non—key attribute)o

主码:若一个关系有多个候选码，则选定其中一个为主码(primary key)。

外部码:设F是基本关系R的一个或一组属性，但不是关系R的码。如果F与基本关系S的主 码K相对应，则称F是基本关系R的外部码,简称外码。

关系是笛卡尔积的有限子集，所以关系也是一个二维表，表的每行对应一个元组，表的每列对 应一个域。

关系可以有三种类型:基本关系(通常又称为基本表或基表)、查询表和视图表。

■二、关系操作

\

关系模型中常用的关系操作包括查询操作(query)和插入(insert)、删除(delete)、修改(update) 操作两大部分。

■三、关系的完整性

关系模型的完整性规则是对关系的某种约束条件，也就是说关系的值随着时间的变化应该满 足一些约束性条件。

1. 关系模型的三类完整性约束

第二章

关系中有三类完整性约束:实体完整性、参照完整性和用户定义的完整性。

1. 实体完整性(entity integrity):

关系数据库

实体完整性规则：

若属性(指一个或一组属性)A是基本关系R的主属性，则A不能取空值。

实体完整性规则的说明：

1. 实体完整性规则是针对基本关系而言的。一个基本表通常对应现实世界的一个实体集。
2. 现实世界中的卖体是可区分的,即它们具有某种唯一性标识。
3. 关系模型中以主码作为唯一性标识。
4. 主码中的属性即主属性不能取空值。主属性取空值，就说明存在某个不可标识的实体，即 存在不可区分的实体，这与第(2)点相矛盾，因此这个规则称为实体完整性。
5. 参照完整性(referential integrity)

参照完整性规则

若属性(或属性组)F是基本关系R的外码它与基本关系S的主码Ks相对应(基本关系R和S 不一定是不同的关系)•则对于R中每个元组在F上的值必须为：

或者取空值(F的每个属性值均为空值)；

或者等于S中某个元组的主码值。

1. 用户定义的完整性(user—defined integrity)

关系的完整性约束

用户定义的完整性:应用领域需要遵循的约束条件，体现了具体领域中的语义约束。

针对某一具体关系数据库的约束条件，反映某一具体应用所涉及的数据必须满足的语义要求

关系模型应提供定义和检验这类完整性的机制，以便用统一的系统的方法处理它们，而不需由 应用程序承担这一功能。

关系操作的特点，关系代数中的各种运算

关系操作的特点是集合操作方式，即操作的对象和结果是集合。

**20**

数据库系统概论(第**5**版)同步辅导及习题全解

■四、关系代数

1. 并(RUS)仍为n目关系，由属于R或属于S的元组组成。RUS= { tltERVtES }
2. 差(R-S)仍为n目关系，由属于R而不属于S的所有元组组成。R—S = { t|t£RAt£S}
3. 交(RDS)仍为n目关系，由既属于R又属于S的元组组成。Rp|S= { t|t€RAt£S} RQS =R-(R-S)
4. 选择:选择又称为限制(Restriction);选择运算符对元组按照条件进行筛选。在关系R中 选择满足给定条件的诸元组aF(R) = {t|tERAF(t)=，真，}。
5. 投影运算符m从R中选择出若干属性列组成新的关系。irA(R) = { t[A] I tER }A：R中 的属性列投影操作主要是从列的角度进行运算。但投影之后不仅取消了原关系中的某些列，而且 还可能取消某些元组(避免重复行)。
6. 连接:两张表中的元组有条件的串接，运算符为0。从两个关系的笛卡尔积中选取属性间满 足一定条件的元组 RXS= {| tr6RAtSeSAtr[A]0ts[B] }

外连接:如果把舍弃的元组也保存在结果关系中，而在其他属性上填空值(Null),这种连接就叫 做外连接。

左外连接:如果只把左边关系R中要舍弃的元组保留就叫做左外连接(LEFT OUTER JOIN或 LEFT JOIN)o

右外连接:如果只把右边关系S中要舍弃的元组保留就叫做右外连接(RIGHT OUTER JOIN 或 RIGHT JOIN) o

1. 除+：给定关系R(X,Y)和S(Y,Z),其中X,Y,Z为属性组;R中的Y与S中的Y可以有不 同的属性名，但必须出自相同的域集;R与S的除运算得到一个新的关系P(X),P是R中满足下列 条件的元组在X属性列上的投影:元组在X上分量值x的像集Yx包含S在Y上投影的集合。

典型例题

例1给出3个域：

D1 =导师集合SUPERVISOR= (张清玫，刘逸}

D2=专业集合SPECIALITY= {计算机专业，信息专业}

D3=研究生集合POSTGRADUATE= {李勇，刘晨，王敏}

解析D1,D2,D3的笛卡尔积为D1XD2XD3={

（张清玫，计算机专业，李勇），（张清玫，计算机专业，刘晨）, （张清玫，计算机专业，王敏），（张清玫，信息专业，李勇）， （张清玫，信息专业，刘晨），（张清玫，信息专业，王敏）， （刘逸，计算机专业，李勇），（刘逸，许算机专业，刘晨）， （刘逸，计算机专业，王敏），（刘逸，信息专业，李勇）， （刘逸，信息专业，刘晨），（刘逸，信息专业,王敏）｝ 基数为2X2X3 = 12

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 例**2**关系**R**和关系**S**如下所示。  **R S** |  |  |  |  |
| A 1 B 1 C | B | E |  |  |
| al bl | 5 | b】 | 3 |  |  |
| al b2 6\_\_  -^2 b3 8 -  一： | b2  b3  b3  b2 | *±* |  |  |
| 解析 一般连续RxS的结果如下：  C<E |  |  |  |  |
| A | R. B | *c* | S. B | E |
| al | bl | 5 | b2 | .7 |
| al | bl | 5 | b3 | 10 |
| al | b2 | 6 | b2 | 7 |
| al | b2 | 6 | b3 | 10 |
| a2 | b3 | 8 | b3 | 10 |
| 等值连续RxS的结果如下：  RB=SB |  |  |  |  |
| A | R. B | C | S. B | E |
| al | bl | 5 | bl | 3 |
| al | b2 | 6 | b2 | 7 |
| a2 | b3 | 8 | b3 | 10 |
| a2 | b3 | 8 | b3 | 2 |

自然连续RxS的结果如下:

第二章

关系数据库

22

数据库系统概论(第**5**版)同步辅导及习题全解

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | E |
| al | bl | 5 | 3 |
| al | b2 | 6 | 7 |
| a2 | b3 | 8 | 10 |
| a2 | b3 | 8 | 2 |
| 下图(a)是例2中关系R和关系S的外连续。 | | | |
| A | B | C | E |
| al | bl | 5 | 3 |
| al | b2 | 6 | 7 |
| a2 | b3 | 8 | 10 |
| a2 | b3 | 8 | 2 |
| a2 | b4 | 12 | NULL |
| NULL | b5 | NULL | 2 |

图(a)

下图(b)、图(c)分别是例2中关系R和关系S的左外连续、右外连续。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | E | A | B | C | E |
| al | bl | 5 | 3 | al | bl | 5 | 3 |
| al | b2 | 6 | 7 | al | b2 | 6 | 7 |
| a2 | b3 | 8 | 10 | a2 | b3 | 8 | 10 |
| a2 | b3 | 8 | 2 | a2 | b3 | 8 | 2 |
| a2 | b4 | 12 | NULL | NULL | b5 | NULL | 2 |

图(b) 图(c)

课后习题解答

1. 医厘关系模型由关系数据结构、关系操作集合和关系完整性约束三部分组成。
2. 匯厘国**k**关系数据语言可以分为三类：

关系代数语言。

关系演算语言:元组关系演算语言和域关系演算语言。

SQL：具有关系代数和关系演算双重特点的语言。

这些关系数据语言的共同特点是，语言具有完备的表达能力，是非过程化的集合操 作语言，功能强,能够嵌入高级语言中使用*。*

1. 域:域是一组具有相同数据类型的值的集合。

笛卡尔积:给定一组域D1,D2,…,Dn,这些域可以有相同的。这组域的笛卡尔积为 DlXD2X・・・XDn={(dl,d2,・・・，dn)|diWDi,i=L2,・・・，n}

第二章

其中每一个元素(dl,d2,…,dn)叫做一个n元组或简称元组。元素中的每一个值di 叫做一个分量。

关系数据库

关系:在域D1,D2,…,Dn上笛卡尔积DlXD2X-XDn的子集称为关系,表示为R (DI ,D2,・・・ ,Dn)。

元组:关系中的每一个元素是关系中的元组。

属性:关系也是一个二维表，表的每行对应一个元组,表的每列对应一个域。由于域 可以相同,为了加以区分，必须对每列起一个名字，称为属性。

1. 候选码:若关系中的某一属性组的值能唯一地标识一个元组，则称该属性组为候 选码。

主码:若一个关系有多个候选码，则选定其中一个为主码。

外部码:设F是基本关系R的一个或一组属性，但不是关系R的码，如果F与基本 关系S的主码Ks相对应，则称F是基本关系R的外部码，简称外码。基本关系R 称为参照关系，基本关系S称为被参照关系或目标关系。关系R和S可以是相同的 关系。

1. 关系模式:关系的描述称为关系模式。它可以形式化地表示为R(U,D,dom, F)，其中R为关系名，U为组成该关系的属性名集合,D为属性组U中属性所来自 的域,Dom为属性向域的映像集合,F为属性间数据的依赖关系集合。

关系数据库:关系数据库有型和值之分。关系数据库的型也称为关系数据库模式， 是对关系数据库的描述,它包括若干域的定义以及在这些域上定义的若干关系模 式。关系数据库的值是这些关系模式在某一时刻对应的关系的集合，通常称为关系 数据库。

|  |  |
| --- | --- |
| 关系模式 | 关系 |
| 对关系的描述 | 关系模式在某一时刻的状态或内容 |
| 静态的、稳定的 | 动态的、随时间不断变化的 |

1. 实体完整性规则是指若属性A是基本关系R的主属性,则属性A不能取空值。

**24**

数据库系统概论(第**5**版)同步辅导及习题全解

若属性(或属性组)F是基本关系R的外码，它与基本关系S的主码Ks相对应(基本 关系R和S不一定是不同的关系)，则对于R中每个元组在F上的值必须为:或者 取空值(F的每个属性值均为空值)；或者等于S中某个元组的主码值。即属性F本 身不是主属性，则可以取空值，否则不能取空值。

1. 1)求供应工程J1零件的供应商号码SNO：

7TSNO( GJN()= JP( SPJ) ) o

1. 求供应工程J1零件Pl的供应商号码SNO：

KSNO((JJNO= 'JI', PNO= ,P1 ，(SPJ))。

1. 求供应工程JI零件为红色的供应商号码SNC)：

7CSnO(7tSn().Pn()(bjnO'Jl' (SPJ) 7TPn() ( (Jcolor'红'(P)))o

7rsno (aPno=' Pl' (oC()L()R='红'(P)ooSPJ)) o

1. 求没有使用天津供应商生产的红色零件的工程号JNO：

irjno( SPJ) - 7tJNO( *acity=*'天津'Color='红'(ASPJsp)。

1. 求至少用了供应商S1所供应的全部零件的工程号JNO：

ttJ no, Pno( SPJ) -r- 7tPno((jSno=' SI' (SPJ)) o

1. **遂题国区** 连接运算符是“=”的连接运算称为等值连接。它是从关系R与S的广义笛卡尔积

中选取A,B属性值相等的那些元组。

自然连接是一种特殊的等值连接,它要求两个关系中进行比较的分量必须是相同的 属性组，并且在结果中把重复的属性列去掉。

自然连接与等值连接的区别：自然连接中相等的分量必须是相同的属性组,并且要 在结果中去掉重复的属性,而等值连接则不必。

自然连接与等值连接的联系：自然连接是等值连接的特殊处理，当R与S无相同属 性时,R8S = RXSO

1. **匯亜区** 并、差、笛卡尔积、投影和选择5种运算为基本的运算。其他3种运算，即交、连接和

除，均可以用这5种基本运算来表达。

用基本运算表示其他运算如下:RQS = R - (R - S)

RooS =aR[A]=S[A](RXS) \

R4-S = kX( R) - 7tX( kX( R) X k Y( S) - R)

第三章

关系数据库标准语言**SQL**

知识网络图

第三章

关系数据库标准语言SQL

fSQL的产生与发展

{ SQL的特点

ISQL的基本概念 模式的定义与删除 基本表的定义、删除与修改 索引的建立与删除 数据字典 单表查询 连接查询 嵌套查询 集合查询

、基于派生表的查询 （插入数据

数据更新{修改数据

I删除数据

'空值的产生 空值的判断 空值的约束条件

、空值的算术运算、比较运算和逻辑运算 定义视图 查询视图 更新视图 视图的作用

rSQL概述•

SQL语言〈

数据定义＜

数据查询＜

空值的处理〈

视图

26

数据库系统概论(第**5**版)同步辅导及习题全解

1. SQL语言的特点、与SQL语言相关的基本概念及与数据库系统结构的三级模式的关系。
2. 数据定义的概念及操作.SQL查询前的准备与最简单的SQL查询语法格式。
3. 条件语句及各种条件运算符的使用。

如：IN、LIKE、BETWEEN等运算以及集函数的使用。

1. 查询分组:GROUP BY子句。

-含IN、LIKE、BETWEEN等运算的条件表达式的构建。

-分组查询。

1. 重要概念。

* SQL的数据定义部分包括对SQL模式、基本表、视图、索引的创建和撤消。

-SQL的数据操纵分成数据查询和数据更新两部分。

* SQL的数据查询是用SELECT语句实现,兼有关系代数和元组演算的特点。

-SQL的数据更新包括插入、删除和修改三种操作，在视图中只有行列子集视图是可以更 新的。

1. 熟练使用SQL语言,在数据库中正确地插入数据、修改数据、删除数据、进行更新操作及视图 操作。学会使用多个元组的插入、多个元组的修改、删除多个元组的值。

知识点归纳

■ 一、**SQL**语言的特点

1.综合统一。

3.面向集合的操作方式。

5 .语言简洁、易学易用。

2.高度非过程化。

4.以同一种语法结构提供多种使用方式。

■二、**SQL**语言的数据查询、数据定义、数据操纵功能

1 .数据定义:定义数据库中的基本对象，包含模式（架构）、表、视图和索引。

第三章

|  |  |  |  |
| --- | --- | --- | --- |
| 操作对象 | 操作方式 | | |
| 创建 | 删除 | 修改 |
| 模式 | CREATE SCHEMA | DROP SCHEMA |  |
| 表 | CREATE TABLE | DROP TABLE | ALTER TABLE |
| 视图 | CREATE VIEW | DROP VIEW |  |
| 索引 | CREATE INDEX | DROP INDEX |  |

⑴模式：

关系数据库标准语言SQL

定义模式:CREATE SCHEMA〈模式名〉AUTHORIZATION V用户名＞[V表定义子句〉 I v视图定义子句＞ I v授权定义子句〉]。

删除模式:DROP SCHEMA V模式名＞〈CASCADE| RESTRICT〉。

CASCADE（级联）：删除模式的同时把该模式中所有的数据库对象全部删除。

RESTRICT（限制）；没有任何下属的对象时才能执行。

（2）表：

定义基本表:CREATE TABLE V表名〉

（V列名〉V数据类型〉[V列级完整性约束条件＞ ]

[,V列名＞ V数据类型〉[V列级完整性约束条件＞]]…

[,〈表级完整性约束条件〉]）。

数据类型：

|  |  |
| --- | --- |
| 数据类型 | 含义 |
| CHAR(n) | 长度为n的定长字符串 |
| VARCHAR(n) | 最大长度为n的变长字符串 |
| INT | 长整数（也可以写作INTEGER） |
| SMALLINT | 短整数 |
| NUMERIC(p,d) | 定点数，由P位数字（不包含符号、小数点）组成，小数点后面有d位数字 |
| REAL | 取决于机器精度的浮点数 |
| DOUBLE RECISION | 取决于机器精度的双精度浮点数 |
| FLOAT(n) | 浮点数，精度至少为n位数字 |
| DATE | 日期,包含年、月、日，格式为YYYY-MM-DD |
| TIME | 时间,包含一日的时、分、秒，格式为HH：MM：SS |

修改基本表:ALTER TABLE〈表名〉

[ADD V新列名＞ V数据类型〉[完整性约束]]

数据库系统概论(第**5**版)同步辅导及习题全解

| 序号 | 标准及主流数 库的处理方式 依赖基本表的对插\* | SQL99 | | Kingbase ES | | ()RACLE9i | | MS SQL SERVER 2005 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| R | c | R |  |  | c |  |
| 1 | 索引 | 无规定 | | J | 7 | V | 7 | J |
| 2 | 视图 | X | J | *X* | J | J  保留 | V  保留 | J 保留 |
| 3 | DEFAULT, PRIMARY KEY,  CHECK(只含该表的列)NOT  NULL等约束 | V | V | J | J | J | V | J |
| 4 | FORELGN KEY | X | V | X | J | X |  | X |
| 5 | TRIGGER | X | J | X | J | V | J | J |
| 6 | 函数或存储过程 | X | J | V  保留 | J  保留 | V  保留 | J  保留 | J 保留 |

L DROP〈列名＞ I〈完整性约束名〉]

E ALTER COLUMN＜列名〉〈数据类型〉]。

删除基本表:DROP TABLE ＜表名＞[RESTRICT| CASCADE]o

RESTRICT（受限）：欲删除的基本表不能被其他表的约束所引用，如果存在依赖该表的对象 （触发器、视图等），则此表不能被删除。

CASCADEC级联）:在删除基本表的同时,相关的依赖对象一起删除。

当基本表定义被删除、数据被删除时，表上建立的索引、视图、触发器等一般也将被删除。

当DROP TABLE时,SQL99与3个RDBMS的处理策略比较：

R 表示 RESTRICT]表示 CASCADE；

‘X，表示不能删除基本表，，J，表示能删除基本表，保留，表示删除基本表后，还保留依赖对象。

（3）索引：

建立索引的目的:加快査询速度。

DBA或表的属主（即建立表的人）（显式）DBMS 一般会自动建立以下约束列上的索弓|（隐式）

PRIMARY KEY UNIQUE

建立索引：

CREATE [UNIQUE] [CLUSTER] INDEX ＜索引名＞

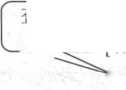
ON V表名〉（V列名＞[V次序＞][,〈列名＞[V次序＞]]・・・）；

用V次序〉指定索引值的排列次序,升序:ASC,降序:DESC。默认:ASC。

UNIQUE表明此索引的每一个索引值只对应唯一的数据记录。

CLUSTER表示要建立的索引是聚集索引。索引项顺序与表中记录的物理顺序一致。 删除索引：DROP INDEX V索引名〉。

■三、数据查询



基本格式：

Select Al,A2,…，An

From R1, R2,.. , Rm

Where F

要显示的属性列

第三章

•:・语句格式

**SELECT[ALL|D|STINCT]<** 目标有凑达式〉

[,〈目标列表达式>] （查询

保存查询结果）

**[INT**灵表名>]

查询结**]FROM<**表名或视图名>[,菜名或视图申 壁釘**[WHERE**〈条件表达式才<.弯鬱佳筛选组）

**[Croup by<** 列名 **i>[having**〈条件表达式 >]]

关系数据库标准语言SQL

**[ORDER BY<** 列名 2>[ASCIDESC]]；

最终查询结果排序:

单表查询：

-选择表中的若干列（投影）

査询指定列（相当于7tA（R）,A= Al,A2,…，An）

选出所有属性列:在SELECT关键字后面列出所有列名，并按用户指定顺序显示。 将V目标列表达式>指定为\*，并按关系模式中的属性顺序显示。

算术表达式：

•选择表中的若干元组（选择）

消除重复性：

指定DISTINCT关键词，去掉表中重复的行；

SELECT DISTINCT Sno FROM SC； 注意DISTINCT短语的作用范围是所有目标列。

使用比较运算符或逻辑运算符NOT+比较运算符

**30**

| 查询条件 | 谓词 | |
| --- | --- | --- |
| 比较 | =,>,<,>=,<=,!=, <>, | !>, !<; NOT+ |
|  | 比较运算符 |  |
| 确定范围 | BETWEEN AND, NOT BETWEEN AND | |
| 确定集合 | IN, NOTIN |  |
| 字符匹配 | LIKE, NOT LIKE *「* | *小* 小4n I |
| 空值 | IS NULL, IS NOTNULlJ^ | 优无级 J |
| 多重条件（逻辑运算） | AND, OR, NOT |  |

数据库系统概论(第**5**版)同步辅导及习题全解

匹配串为含通配符的字符串

%：代表任意长度（可以是0）的字符串

\_：代表任意单个字符

字符串本身就含有％或\_时，在％或\_之前加上转义符“\”要使用ESCAPE ‘V换码字符〉，

将通配符转义为普通字符。如果'\'要作为一个普通字符，用连续两个‘\’表示一个真正的

* ORDER BY子句:对查询结果排序。

可以按一个或多个属性列排序，升序:ASC；降序:DESC ；缺省值为升序。

当排序列含空值时，空值最大。

ASC：排序列为空值的元组最后显示;DESC：排序列为空值的元组最先显示。

-聚集函数:对查询结果集中的某列进行计算或统计。

计数：

C（）UNT（ [DISTINCT | ALL] \*）

CC）UNT（ [DISTINCT | ALL]〈列名 ＞）

计算总和 SUM（ [DISTINCT I ALL]〈列名〉）

计算平均值 AVG（:DISTINCT| ALL]〈列名〉）

最大最小值：

MAX（ [DISTINCT | ALL] V列名〉）

MIN（[DISTINCTI ALL] ＜列名〉）

* GROUP BY子句:对查询结果分组。

用途细化聚集函数的作用对象：

未对查询结果分组,聚集函数将作用于整个查询结果八

对查询结果分组后，聚集函数将分别作用于每个组。

使用GROUP BY后，其SELECT子句的列名列表中只能出现分组属性和集函数。

如果分组后还要按照条件对这些组进行筛选,可使用HAVING短语指定筛选条件。

HAVING短语与WHERE子句的区别:作用对象不同。

WHERE子句作用于基表或视图,从中选择满足条件的元组。

HAVING短语作用于组，从中选择满足条件的组。

**1—**

**3**

\

.四、数据更新

第三章

1. 插入数据

关系数据库标准语言SQL

插入元组

INSERT

[INTO] V表名〉[（V属性列1＞[,＜属性列2 ＞・・・）]

VALUES （V常量 1〉[,＜常量 2＞]…）

功能:将新元组插入指定表中;新元组的属性列1的值为常量1，属性列2的值为常量2,…。

INTO子句:属性列的顺序可与表定义中的顺序不一致，但须指定列名；没有指定属性列，表示 要插入的是一条完整的元组;指定部分属性列，未指定的属性列取空值，具有NOT NULL的属性列 除外。VALUES子句:提供的值必须与INTO子句匹配,包括值的个数、值的类型。

1. 修改数据

UPDATE V 表名〉

SETV列名＞=V表达式＞[,V列名 ＞=＜ 表达式 …

[WHERE V条件＞]；

SET子句:指定修改方式,要修改的列,修改后取值:〈表达式〉。

WHERE子句:指定要修改的元组,缺省表示要修改表中的所有元组。

功能:修改指定表中满足WHERE子句条件的元组;修改某一个元组的值

1. 删除数据

DELETE

FROM〈表名＞

[WHERE V条件＞]；

功能:删除指定表中满足WHERE子句条件的元组。

■五、视图

特点:虚表，是从一个或几个基本表（或视图）导出的表；只存放视图的定义，不存放视图对应的 数据;基表中的数据发生变化，从视图中查询出的数据也随之改变。

基于视图的操作：查询、删除、受限更新、定义基于该视图的新视图。

32

数据库系统概论(第**5**版)同步辅导及习题全解

1. 定义视图
2. 建立视图

CREATE VIEW V视图名＞[(V列名＞[,〈列名＞]・・・)]

AS＜子查询＞

[WITH CHECK OPTION]；

子查询:不允许含有ORDER BY子句和DISTINCT短语。

WITH CHECK OPTION：表示对视图进行UPDATE .INSERT和DELETE操作时要保证更 新、插入或删除的行满足视图定义中的谓词条件(即子查询中的条件表达式)。

1. 删除视图DROP VIEWV视图名〉；

该语句从数据字典中删除指定的视图定义。如果该视图上还导出了其他视图，使用CASCADE 级联删除语句，把该视图和由它导出的所有视图一起删除。删除基表时，由该基表导出的所有视图 定义都必须显式地使用DROP VIEW语句删除。

1. 查询视图:查询视图与查询基本表相同，视图定义后，就可以像对待基本表一样对视图进行 查询(SELECT)操作。

使用视图消解法(VIEW RESOLUTION)进行有效性检查,检查查询的表、视图等是否存在。 如果存在，则从数据字典中取出视图的定义，并把对视图的查询转换成等价的对基本表的查询，把 视图定义中的子查询与用户的查询结合起来，最后执行修正后的查询。

1. 更新视图

(DDBMS实现视图更新的方法:转换为对基本表的更新。

只有对“可更新”视图才能进行更新操作。

1. 对于视图元组的更新操作(INSERT、DELETE、UPDATA),有以下三条规则：

如果一个视图是从多个基本表使用联接操作导出的，那么不允许对这个视图执行更新操作。 如果在导出视图的过程中,使用了分组和聚集函数操作，也不允许对这个视图执行更新操作。 行列子集视图可以执行更新操作。•

1. 更新视图的限制:一些视图是不可更新的，因为对这些视图的更新不能唯一地有意义地转 换成对相应基本表的更新。对其他类型视图的更新，不同系统有不同限制：
2. 若视图是由两个以上基本表导出的，则此视图不允许更新。
3. 若视图的字段来自字段表达式或常数，则不允许对此视图执行INSERT和UPDATE操作, 但允许执行DELETE操作。
4. 若视图的字段来自集函数，则此视图不允许更新。
5. 若视图定义中含有GROUP BY子句，则此视图不允许更新。
6. 若视图定义中含有DISTINCT短语，则此视图不允许更新。
7. 若视图定义中有嵌套查询，并且内层查询的FROM子句中涉及的表也是导岀该视图的基本

表，则此视图不允许更新。 、

1. 一个不允许更新的视图上定义的视图也不允许更新。

例1 建立一个“学生”表Student,它由学号Sno、姓名Sname、性别Ssex、年龄Sage、所在系Sdept 五个属性组成。其中学号不能为空，值是唯一的，并且姓名取值也唯一。

解析 CREATE TABLE Student

(Sno CHAR(5) NOT NULL UNIQUE,

Sname CHAR(20) UNIQUE,

Ssex CHAR(l),

Sage INT,

Sdept CHARC15))

例2设学生课程数据库中有三个关系：

学生关系 S(S# ,SNAME, AGE,SEX)

学习关系 SC(S# ,C# ,GRADE)

课程关系C(C# ,CNAME)

其中S#、C#、SNAME、AGE、SEX、GRADE、CNAME分别表示学号、课程号、姓名、年龄、 性别、成绩和课程名。

用SQL语句表达下列操作：

1. 检索选修课程名称为“MATHS"的学生的学号与姓名
2. 检索至少学习了课程号为“C1"和“C2"的学生的学号
3. 检索年龄在18到20之间(含18和20)的女生的学号、姓名和年龄
4. 检索平均成绩超过80分的学生学号和平均成绩
5. 检索选修了全部课程的学生姓名
6. 检索选修了三门课以上的学生的姓名 解析 （l）SELECT SNAME,AGE

FROM S,SC,C

WHERE S.S#=SC.S#

34

数据库系统概论(第**5**版)同步辅导及习题全解

AND C.C#=SC.C#

AND CNAME=' MATHS'

1. SELECT S#

FROM SC

WHERE CNO=4C1, AND S# IN(SELECT S#

FROM SC

WHERE CNO=\*C2,)

1. SELECT S# ,SNAME,AGE

FROMS

WHERE AGE BETWEEN 18 AND 20

1. SELECTS#,AVG(GRADE)，平均成绩，

FROM SC

GROUP BY S# HAVING AVG(GRADE)>80

1. SELECT SNAME

FROM S

WHERE NOT EXISTS

(SELECT \*

FROM C

WHERE NOT EXISTS

(SELECT \*

FROM SC

WHERE S# =S. S# AND C# =C. C#))

1. SELECT SNAME

FROM S,SC

WHERE S.S#=SC.S#

GROUP BY SNAME

HAVING COUNT( \* )>3

例 **3** 设有关系 S (S#,SNAME,SAGE,SEX),C(C#,CNAME),SC(S#,C 井，GRADE)。其中 S#是学生号，SNAME是学生姓名,SAGE是学生年龄,SEX是学生的性别，C#是课程号， CNAME是课程名称,GRADE是成绩。

解析(1)用SQL语句创建S表，声明“S\*”为主码，“SNAME”不能为空，“SEX”的取值为男或女。

CREATE TABLE Student

(SNO CHAR(9) PRIMARY KEY,

SNAME CHAR(20) UNIQUE,

\

SEX CHAR(2) check(SEX=男 or SEX=女)，

SAGE SMALLINT)；

第三章

1. 用SQL语句修改C表中的列CNAME为CNAME CHAR(40)o

ALTER TABLE C

关系数据库标准语言SQL

ALTER COLUMN CNAME CHAR(40)；

1. 用SQL语句向原表S中插入一个学生信息(具体信息自定)。

INSERT

INTO Student (SNO,SNAME,SEX,SAGE)

VALUES ('2Q0215128',4 陈冬'，'男'，18)；

1. 分别用关系代数表达式和SQL语句查询出所有女同学的姓名、年龄信息。

SELECT SNAME,SAGE

FROM S

WHERE S. SEX =4女气

Rl = aSEX ='女'(S)

R2=kSNAME,SAGE (RD

例4设某工厂数据库中有两个基本表：

车间基本表:DEPT( DNO, DNAME, MGR\_ENO)，其属性分别表示车间编号、车间名和车 间主任的职工号。

职工基本表:ERP( ENO, ENAME, AGE, SEX, SALARY, DNO)，其属性分别表示职工号、 姓名、年龄、性别、工资和所在车间的编号。建立一个有关女车间主任的职工号和姓名的视 图，其结构如下：VIEW6(ENO,ENAME)o试写出创建该视图VIEW6的SQL语句。

解析 CREATE VIEW VIEW6

AS

SELECT ENO, ENAME FROM ERP

WHERE SEX='女'AND ENO IN

(SELECT MGR\_EN() FROM DEPT)

例5用SQL语句使用嵌套查询，查询出没有选修1号课程的学生学号和姓名。

解析 SELECT S# ,Sname,

FROM Student

WHERE NOT EXISTS

(SELECT \*

FROM SC

36

数据库系统概论(第**5**版)同步辅导及习题全解

WHERE Sno = Student. Sno AND Cno=' 1');

例6 (2015年深圳大学)查询Student表中的所有非空Email信息，以下语句正确的是( )。

1. SELECT Email FROM Student WHERE Email ! =NULL
2. SELECT Email FROM Student WHERE Email NOT IS NULL
3. SELECT Email FROM Student WHERE EmailONULL
4. SELECT Email FROM Student WHERE Email IS NOT NULL

答案D

解析 在数据库SQL语句中,WHERE子句表示条件，用IS NULL来表示空的记录，IS NOT

NULL表示非空信息。

1. 匯题函k （1）综合统一:SQL语言集数据定义语言DDL、数据操纵语言DML、数据控制语言

课后习题解答

DCL的功能于一体。

（2） 高度非过程化:用SQL语言进行数据操作，无需了解存取路径。

（3） 面向集合的操作方式:SQL语言釆用的集合操作方式,不仅操作对象、查询结果 可以是元组的集合,而且一次插入、删除、更新操作的对象也可以是元组的集合。

（4） SQL语言既是自含式语言，能独立地用于联机交互，又是嵌入式语言，能嵌入到 高级语言中进行混合编程。

（5） 语言简捷，易学易用；

1. 匯厘函k在DROP TABLE语句删除基本表时，使用RESTRICT（受限）是为了确保只有不存

在相关视图和完整性约束的表才能被删除。欲删除的基本表不能被其他表的约束 所引用，如果存在依赖该表的对象（触发器，视图等），则此表不能被删除。

而CASCADE（级联）是任何的相关视图和定整性约束一并都被删除,在删除基本表 的同时，相关的依赖对象一起删除。

\ (1) select \* from S where A=' 10'；

园过程

**3.**

1. select A,B from S；
2. select A,B,S. C,S. DEF from S,T 日小提示：掌握基本的 SQL

where S. C=T. C and S. D=T. D；

1. select \* from S ,T where S. C=T. C；

\

37

第三章

关系数据库标准语言SQL

1. select \* from S ,T where S. AVT. E；
2. select S. C,S. D,T. \* from S,T；

建立表 S(SNO, SNAME,STATUS,CITY)

CREATE TABLE S(SNO CHAR(3) ,SNAME CHAR(IO),

STATUS CHAR(2),CITY CHAR(IO))；

建立表 P(PNO,PNAME,COLOR, WEIGHT)

CREATE TABLE P(PNO CHAR(3) ,PNAME CHAR(IO),

COLOR CHAR(4),WEIGHT INT)；

建立表 J(JN(),JNAME,CITY)

CREATE TABLE J(JNO CHAR(3) ,JNAME CHAR(IO) ,CITY CHAR(IO))； 建立表 SPJ(SN(),PNO,JNO,QTY)

CREATE TABLE SPJ(SNO CHAR(3), PNO CHAR(3),

JNO CHAR(3),QTY INT)；

或 SELECT JNO FROM J WHERE NOT EXISTS

数据库系统概论(第**5**版)同步辅导及习题全解

(SELECT \* FROM SPJ,S,P WHERE SPJ. JNO=J. JNO

AND SPJ. SNO=S. SNO AND SPJ. PNO=P. PNC)

AND S. CITY='天津'AND P. COL()R='红')；

1. 求至少用了 SI供应商所供应的全部零件的工程号JN()O

SELECT DISTINCT JNO FROM SPJ SPJZ WHERE NOT EXISTS

(SELECT \* FROM SPJ SPJX WHERE SNO='S1' AND NOT EXISTS

(SELECT \* FROM SPJ SPJY WHERE SPJY. PNO=SPJX. PNC)

AND SPJY. JNO=SPJZ. JNO))；

「知识点播：上述查询可以抽象为：要求这样的工程X,使(』y)p—>q为真。即对于所 有的零件y,满足逻辑蕴涵piq： P为谓词“供应商S1供应了零件y” ; q为谓词 \ “工程X选用了零件y” o即，只要“供应商S1供应了零件y”为真，则“工程x 选用了零件y”就为真。逻辑蕴涵可以转换为等价形式：(Vy)p-qM—>(Vy(—Xp-\*q)) —-(Vy(—1(—p Vq))) — 一 Vy(p A—q)表达语义是：不存在这禅的泰件y,供应商S1 <供应了，而工程x没有选用。

1. 找出所有供应商的姓名和所在城市。

SELECT SNAME,CITY FROM S；

1. 找出所有零件的名称、颜色、重量。

SELECT PNAME,COLOR, WEIGHT FROM P；

1. 找出使用供应商S1所供应零件的工程号码。

SELECT JNO FROM SPJ WHERE SNO= 'SI'；

1. 找出工程项目J2使用的各种零件的名称及其数量。

SELECT P. PNAME,SPJ. QTY FROM P,SPJ

WHERE P. PN()=SPJ. PNO AND SPJ. JN()= <J2,；

1. 找出上海厂商供应的所有零件号码。

SELECT DISTINCT PNO FROM SPJ

WHERE SNO IN (SELECT SNO FROM S WHERE CITY=\*上海')；

1. 找出使用上海产的零件的工程名称。

SELECT JNAME FROM J,SPJ,S

WHERE J. JNO=SPJ. JNO AND SPJ. SN()=S. SNO AND S. CITY='上海'； 或:SELECT JNAME FROM J WHEREJNO INC SELECT JNO FROM SPJ,S

WHERE SPJ. SNO=S. SNO AND S. CITY='上海')；

1. 找出没有使用天津产的零件的工程号码。

SELECT JNO FROM J WHERE NOT EXISTS

(SELECT \* FROM SPJ WHERE SPJ. JN()=J. JNO

AND SNO IN (SELECT SNO FROM S WHERE CITY='天津'))；

或:SELECT JNO FROM J WHERE NOT EXISTS

(SELECT \* FROM SPJ,S

WHERE SPJ. JNO=J. JNO AND SPJ. SNO=S. SNO AND S. CITY='天津')；

第三章

1. 把全部红色零件的颜色改为蓝色。

UPDATE P SET C()L()R='蓝'WHERE C()L()R='红'；

关系数据库标准语言SQL

1. 由S5供给J4的零件P6改为由S3供应，请做必要的修改。

UPDATE SPJ SET SNO=' S3'

WHERE SNO='S5' ANDJN()='J4' AND PNO='P6'；

1. 从供应商表中删除S2的记录，并从供应情况表中删除相应记录。

DELETE FROM SPJ WHERE SN()= 'S2'；

DELETE FROM S WHERE SNO=，S2，；

1. 请将(S2,J6,P4,200)插入供应情况表。

INSERT INTO SPJ(SNO, JNO,PNO, QTY) VALUES (' S2',' J6',' P4', 200)； **6**」解题靂区 基本表:是独立存在的表。在SQL中，一个关系对应于一个表。

视图:是从一个或多个基本表所导出的表。视图本身并不独立存储在数据库中，是 一个虚表，即数据库中只存放视图的定义而不存放其所对应的数据,这些数据仍然 存放在导出的视图的基本表中。视图在概念 上与基本表等同,用户可以像使用基本表那样 使用视图，可以在视图上再定义视图。

知识点播：SELECT语句中出现 \ 的基本表名，应理解为基本表中 I的元组变量，而列名应理解为元 组分量。

**7.**曰解题通区

1. 视图能够简化用户的操作，聚焦于所关心的数据上。
2. 视图使用户能以多种角度看待同一数据,增加灵活性。
3. 视图对重构数据库提供了一定程度的逻辑独立性。 V小提示：适当地利用视图
4. 视图能够对机密数据提供安全保护。 p竺~~\*清商云表达查询。“~~

**8.1**解题明詛基本表的行列子集视图一般是可更新的;若视图的属性来自集函数、表达式,则该视 图肯定是不可更新的。视图是不实际 存储数据的虚表，因此对视图的更新，

J小提示：有些视图的更新不能唯一有意义 下地转换成对相应基本表的更新，所以并不 ［是所有的视图都是可更新的。

最终要转换为对基本表的更新。

例如:对视图S\_G(学生的学号及其平均成绩)而言，要修改平均成绩,必须修改各科 成绩，但无法知道哪些课程成绩的变化导致了平均成绩的变化。

CREATE VIEW S G(Sno,Gavg)

AS SELECT Sno,AVG(Grade) FROM SC GROUP BY Sno；

**9.；**應题函**k**建立视图：

数据库系统概论(第**5**版)同步辅导及习题全解

CREATE VIEW V\_SPJ AS SELECT SN(),PN(),QTY FROM SPJ WHERE JNO= (SELECT JNO FROM J WHERE JNAME='三建')；

1. SELECT PNO, QTY FROM V\_SPJ；
2. SELECT PNO,QTY FROM V\_SPJ WHERE SN()='S1'；

［知识点播：建立视图的操作语句 VJ CREATE VIEW v 视图名 >

\ ［（v列名>［,v列名习...）］

AS v子查询〉

［WITH CHECK OPTION］；

41

*in*

JEE.

数据库安全性

第四章

数据库安全性

知识网络图

非授权用户对数据库的恶意存取和破坏

概述•

不安全因素

数据库中重要和敏感的数据被泄露

安全环境的脆弱性

安全标准

用户身份鉴别

存取控制

安全性控制＜

自主存取控制方法

授权:授予与收回

数据库安全性〈

数据库角色

强制存取控制方法

审计事件

审计功能

AUDIT语句

NOAUDIT 语句

J透明

数据加密＜

存储加密

1非透明

J链路加密

1端到端加密

重难点

重点:使用SQL中的GRANT语句和REVOKE语句来实现数据库的自主存取控制功能;使用 SQL中的CREATE ROLE语句创建角色，用GRANT语句给角色授权;掌握视图机制在数据库安 全保护中的作用。

数据库系统概论(第**5**版)同步辅导及习题全解

难点:强制存取控制（MAC）机制中确定主体能否存取客体的存取规则，同学们要理解并掌握存 取规则为什么要这样规定。

1. 计算机安全性概论
2. 数据库安全性控制
3. 统计数据库安全性
4. 视图机制、审计、数据加密、统计数据安全等
5. 可信计算机系统评测标准
6. 数据库安全性控制方法（包括用户识别、自主与强制存取控制、授权等）

数据库管理员拥有对数据库中所有对象的所有操作权限，并可以根据应用的需要将不同的权 限授予不同的用户。用户对自己建立的基本表和视图拥有全部的操作权限，并且可以用**grant** 语句把其中某些权限授予其他用户。被授权的用户如果有“继续授权”的许可，还可以把获得的权 限再授予其他用户。所有授予出去的权力在必要时又都可以用REVOKE语句收回。

重点掌握强制存取控制机制中确定主体能否存取客体的存取规则。

知识点归纳

. 一、计算机安全性概述

1. 计算机系统安全性。

为计算机系统建立和采取的各种安全保护措施，以保护计算机系统中的硬件、软件及数据，防 止其因偶然或恶意的原因使系统遭到破坏，数据遭到更改或泄露等。

1. 三类计算机系统安全性问题。

技术安全类

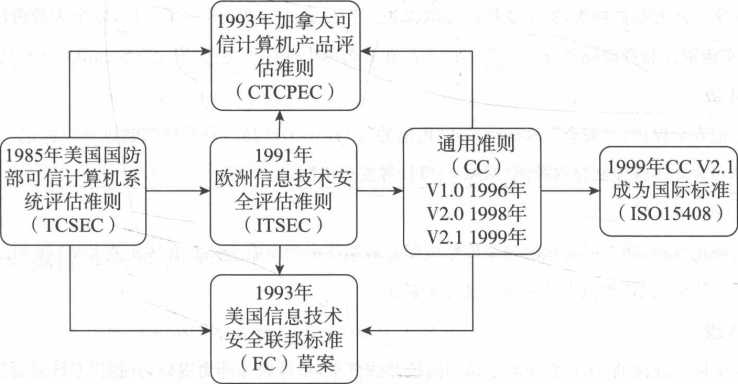
管理安全类

政策法律类

3 .为降低进而消除对系统的安全攻击，各国引用或制定了一系列安全标准。

TCSEC标准 *、*

CC标准

4.安全标准简介（结构图）。

第四章

1. TCSEC/TDI标准的基本内容。

TCSEC/TDI从四个方面来描述安全性级别划分的指标:安全策略、责任、保证、文档。

1. TCSEC/TDI安全级别划分。

数据库安全性

|  |  |
| --- | --- |
| 安全级别 | 定义 |
| A1 | 验证设计(Verified Design) |
| B3 | 安全域(Security Domains) |
| B2 | 结构化保护(Structural Protection) |
| B1 | 标记安全保护(Labeled Security Protection) |
| *C2* | 受控的存取保护(Controlled Access Protection) |
| C1 | 自主安全保护(Discretionary Security Protection) |
| D | 最小保护(Minimal Protection) |

按系统可靠或可信程度逐渐增高,各安全级别之间:偏序向下兼容。

1. TCSEC/TDI安全级别划分。

D级

・将一切不符合更高标准的系统均归于D级

・典型例子:DOS是安全标准为D的操作系统

• DOS在安全性方面几乎没有什么专门的机制来保障

44

数据库系统概论(第**5**版)同步辅导及习题全解

C1级

非常初级的自主安全保护，能够实现对用户和数据的分离，进行自主存取控制(DAC),保护或 限制用户权限的传播。

C2级

安全产品的最低档次，提供受控的存取保护，将C1级的DAC进一步细化，以个人身份注册负 责，并实施审计和资源隔离，达到C2级的产品在其名称中往往不突出“安全"(Security)这一特色。

B1级

标记安全保护。“安全"(Security)或“可信的"(Trusted)产品。对系统的数据加以标记,对标记 的主体和客体实施强制存取控制(MAC)、审计等安全机制。

B2级

结构化保护,建立形式化的安全策略模型并对系统内的所有主体和客体实施DAC和MAC。

经过认证的B2级以上的安全系统非常稀少。

B3级

安全域。该级的TCB必须满足访问监控器的要求，审计跟踪能力更强，并提供系统恢复过程。

A1级

验证设计,即提供B3级保护的同时给出系统的形式化设计说明和验证以确信各安全保护真正 实现。

B2级以上的系统，还处于理论研究阶段,应用多限于一些特殊的部门，如军队等。美国正在大 力发展安全产品,试图将目前仅限于少数领域应用的B2安全级别下放到商业应用中来，并逐步成 为新的商业标准CC。

1. CC。

提出国际公认的表述信息技术安全性的结构,把信息产品的安全要求分为以下两方面。

安全功能要求:规范产品和系统的安全行为。

安全保证要求:解决如何正确有效地实施这些功能的问题。

1. CC文本组成。

简介和一般模型 、

安全功能要求

安全保证要求

1. CC评估保证级划分。

| 评估保证级 | 定义 | TCSEC安全级别  (近似相当) |
| --- | --- | --- |
| EAL1 | 功能测试(functionally tested) |  |
| EAL2 | 结构测式(structurally tested) | Cl |
| EAL3 | 系统地测式和检查(methodically tested and checked)  系统地设计、测试和复查(methodically designed,tested,and re- | C2 |
| EAL4 | viewed) | Bl |
| EAL5 | 半形式化设计和测试(semiformally designed and tested) 半形式化验证的设计和测试(semiformally verified design and | B2 |
| EAL6 | tested) | B3 |
| EAL7 | 形式化验证的设计和测试(fomally verified design and tested) | Al |

第四章

数据库安全性

■二、数据库安全性控制概述

非法使用数据库的情况，编写合法程序绕过DBMS及其授权机制，直接或编写应用程序执行非 授权操作，通过多次合法查询数据库从中推导出一些保密数据。

-计算机系统中，安全措施是一级一级层层设置的。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用户 | < | DBMS | ► | OS | ► < | DB |
| 用户标识 | 数据库安全操作系统安全 | | | | | 数据密码 |
| 和鉴别 |  | 保护 |  | 保护 |  | 存储 |

计算机系统的安全模型

1. 数据库安全性控制的常用方法
2. 用户标识和鉴定
3. 存取控制
4. 视图
5. 审计
6. 用户标识与鉴别(Identification & Authentication)

系统核对口令以鉴别用户身份

用户名和口令易被窃取

每个用户预先约定好一个计算过程或者函数

1. 存取控制

存取控制机制的组成

46

数据库系统概论(第**5**版)同步辅导及习题全解

定义用户权限

合法权限检查

用户权限定义和合法权限检査机制一起组成了数据库管理系统的安全子系统

1. 常用存取控制方法

自主存取控制(Discretionary Access Control,DAC)

强制存取控制(Mandatory Access Control,MAC)

1. 自主存取控制方法

通过SQL的GRANT语句和REVOKE语句实现

用户权限组成

数据对象

操作类型

定义用户存取权限:定义用户可以在哪些数据库对象上进行哪些类型的操作。定义存取权限 称为授权,关系数据库系统中存取控制对象。

1. 授权与回收
2. GRANT

GRANT语句的一般格式：

GRANT V权限〉[,V权限＞1. f

[ON〈对象类型＞ ＜对象名〉]

TO〈用户＞[,V用户〉丄..

[WITH GRANT OPTION]；

语义:将对指定操作对象的指定操作权限授予指定的用户。

1. REVOKE

授予的权限可以由数据库管理员或其他授权者用REVOKE语句收回

REVOKE语句的一般格式为：

REVOKE ＜权限〉[,V权限＞]...

**Eon** ＜对象类型＞ v对象名＞]

FROM V用户＞[,〈用户;

1. 创建数据库模式的权限

数据库管理员在创建用户时实现

CREATE USER语句格式 \

CREATE USER ＜username＞

[WITH] [DBA I RESOURCE | CONNECT]

CREATE USER语句格式说明

只有系统的超级用户才有权创建一个新的数据库用户。

新创建的数据库用户有三种权限:CONNECT.RESOURCE和DBA。

如没有指定创建的新用户的权限，默认该用户拥有CONNECT权限。拥有CONNECT权限的 用户不能创建新用户，不能创建模式，也不能创建基本表，只能登录数据库。

权限与可执行的操作对照表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 拥有的权限 | 可否执行的操作 | | | |
| CREATE  USER | CREATE  SCHEMA | CREATE  TABLE | 登录数据库，  执行数据査询和操纵 |
| DBA | 可以 | 可以 | 可以 | 可以 |
| RESOURCE | 不可以 | 不可以 | 不可以 | 不可以 |
| CONNECT | 不可以 | 不可以 | 不可以 | 可以,但必须拥有相应权限 |

1. 数据库角色:被命名的一组与数据库操作相关的权限

角色是权限的集合，可以为一组具有相同权限的用户创建一个角色，简化授权的过程。

1. 数据库角色相关语句

（1） 角色的创建

CREATE ROLE V角色名〉

（2） 给角色授权

GRANT〈权限＞[,＜权限＞]・•・

ONV对象类型〉对象名

TO V角色＞[,＜角色＞]・・・

（3） 将一个角色授予其他的角色或用户

GRANT V角色1＞[,＜角色2＞]…

TO V角色3＞[,＜用户1＞]…

[WITH ADMIN OPTION] .

（4） 角色权限的收回

REVOKE V权限＞[,V权限＞]…

ON〈对象类型＞ ＜对象名＞

FROM V角色＞[,V角色…

用户可以回收角色的权限,从而修改角色拥有的权限。

三、视图机制

把要保密的数据对无权存取这些数据的用户隐藏起来,对数据提供一定程度的安全保护，主要

功能是提供数据独立性，无法完全满足要求，间接实现了支持存取谓词的用户权限定义。

■四、审计

数据库系统概论(第**5**版)同步辅导及习题全解

启用一个专用的审计日志(Audit Log),将用户对数据库的所有操作记录在上面,审计员利用审 计日志，监控数据库中的各种行为，找出非法存取数据的人、时间和内容,C2以上安全级别的DBMS 必须具有审计功能。

1. 审计事件
2. 审计功能
3. 审计分类
4. 用户级审计

针对自己创建的数据库表或视图进行审计，记录所有用户对这些表或视图的一切成功和(或) 不成功的访问要求以及各种类型的SQL操作。

1. 系统级审计

数据库管理员设置

监测成功或失败的登录要求

监测GRANT和REVOKE操作以及其他数据库级权限下的操作

AUDIT语句：设置审计功能

NOAUDIT语句:取消审计功能

.五、数据加密

1. 数据加密

防止数据库中数据在存储和传输中失密的有效手段

1. 加密的基本思想

加密方法、替换方法、置换方法、混合方法

1. 存储加密

透明存储加密.

内核级加密保护方式，对用户完全透明，数据在写到磁盘时对其进行加密，授权用户读取数据 时再对其进行解密，数据库的应用程序不需要做任何修改，只需在创建表语句中说明需加密的字段 即可，内核级加密方法：性能较好，安全完备性较高。

1. 非透明存储加密

通过多个加密函数实现，传输加密、链路加密。在链路层进行加密，传输信息由报头和报文两

部分组成。报文和报头均加密，端到端加密，在发送端加密,接收端解密；只加密报文不加密报头, 所需密码设备数量相对较少，容易被非法监听者发现并从中获取敏感信息。

■六、其他安全性保护

第四章

推理控制:处理强制存取控制未解决的问题，避免用户利用能够访问的数据推知更高密级的数据。 常用方法:基于函数依赖的推理控制,基于敏感关联的推理控制。

数据库安全性

隐蔽信道:处理强制存取控制未解决的问题。

典型例题

例1 把查询Student表权限授给用户Ulo

解析 GRANT SELECT

ON TABLE Student

TOU1；

例2 把对Student表和Course表的全部权限授予用户U2和U3。

解析 GRANT ALL PRIVILIGES

ON TABLE Student, Course

TO U2, U3；

例3 把查询Student表和修改学生学号的权限授给用户U4。

解析 GRANT UPDATE(Sno), SELECT

ON TABLE Student

TO U4；

例4把用户U5对SC表的INSERT权限收回。

REVOKE INSERT

ON TABLE SC

FROM U5 CASCADE ；

解析 将用户U5的INSERT权限收回的时候应该使用CASCADE,否则拒绝执行该语句。

如果U6或U7还从其他用户处获得对SC表的INSERT权限测他们仍具有此权限,系统只 收回直接或间接从U5处获得的权限。

50

数据库系统概论(第**5**版)同步辅导及习题全解

例**5** 把对表SC的INSERT权限授予U5用户，并允许他再将此权限授予其他用户。

GRANT INSERT

ON TABLE SC

TOU5

WITH GRANT OPTION；

解析 执行例5后,U5不仅拥有了对表SC的INSERT权限,还可以传播此权限：

GRANT INSERT ON TABLE SC TO U6

WITH GRANT OPTION；

同样,U6还可以将此权限授予U7:

GRANT INSERT ON TABLE SC TO U7；

但U7不能再传播此权限。

U5——> U6——> U7

例**6** 通过角色来实现将一组权限授予一个用户。

解析步骤如下：

1. 首先创建一个角色RE

CREATE ROLE Rl；

1. 然后使用GRANT语句，使角色R1拥有Student表的SELECT. UPDATE. INSERT 权限。

GRANT SELECT,UPDATE,INSERT

ON TABLE Student

TO Rl；

1. 将这个角色授予王平、张明、赵玲，使他们具有角色R1所包含的全部权限。

GRANT Rl

TO王平，张明，赵玲；

1. 可以一次性通过R1来回收王平的这3个权限。

REVOKE Rl

FROM王平；

课后习题解答

**L**匯厘幽**k**数据库的安全性是指保护数据库以防止不合法的使用所造成的数据泄露、更改或 破坏。

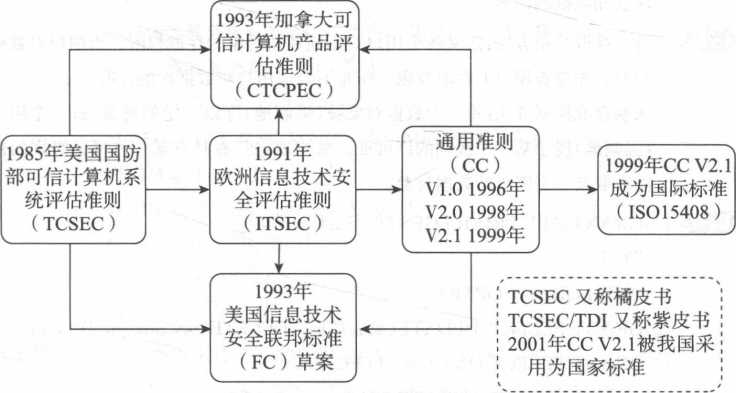
0^1^ 1.非授权用户对数据库的恶意存取和破坏。一些黑客和犯罪分子在用户存取数据 库时猎取用户名和用户口令，然后假冒合法用户偷取、修改甚至破坏用户数据。

1. 数据库中重要或敏感的数揮被泄露，黑客和敌对分子千方百计盗窃数据库中的重 要数据，一些机密信息被泄露。
2. 安全环境的脆弱性:数据库的安全性与计算机系统的安全性，包括计算机硬件、操 作系统、网络系统等的安全性是紧密联系的。

第四章

［解题圖尽 1993年加拿大可信计算机产品评估准则(CTCPEO1985年美国国防部可信计算机 系统评估准则(TCSEO1991年欧洲信息技术安全评估准则(ITSEO®用准则(CC) VI. 0 1996 年、V2. 0 1998 年、V2. 1 1999 年、1999 年 CC V2. 1 成为国际标准(ISO 15408)TCSEC又称桔皮书TCSEC/TDI又称紫皮书。2001年CC V2. 1被我国采 用为国家标准1993年美国信息技术安全联邦标准(FC)草案。TCSEC安全级别(近 似相当)Cl C2 Bl B2 B3 Alo

数据库安全性



信息安全标准的发展简史

|  |  |
| --- | --- |
| 评估保证级 | 定义 |
| EAL1 | 功能测试 |
| EAL2 | 结构测试 |
| EAU | 系统的测试和检査 |
| EAL4 | 系统的设计、测试和复查 |
| EAI5 | 半形式化设计和测试 |
| EAL6 | 半形式化验证的设计和测试 |
| EAL7 | 半形式化验证的设计和测试 |

CC评估保证级(EAL)的划分

52

数据库系统概论(第**5**版)同步辅导及习题全解

**7.**

任迥过程

，①用户王明对两个表有SELECT权力。

**4**.匯厘gL 实现数据库安全性控制的常用方法和技术有：

1. 用户标识和鉴别:该方法由系统提供一定的方式让用户标识自己的名字或身份。 每次用户要求进入系统时，由系统进行核对,通过鉴定后才提供系统的使用权。
2. 存取控制:通过用户权限定义和合法权检査,确保只有合法权限的用户可以访问 数据库，所有未被授权的人员无法存取数据。例如C2级中的自主存取控制 (DAC),B1级中的强制存取控制(MAC)O
3. 视图机制：为不同的用户定义视图，通过视图机制把要保密的数据对无权存取的 用户隐藏起来，从而自动地对数据提供一定程度的安全保护。
4. 审计:建立审计日志，把用户对数据库的所有操作自动记录下来放入审计日志 中，DBA可以利用审计跟踪的信息，重现导致数据库现有状况的一系列事件，找出 非法存取数据的人、时间和内容等。
5. 数据加密:对存储和传输的数据进行加密处理，从而使得不知道解密算法的人无 法获知数据的内容。

**5.1**标**gk**自主存取控制方法:定义各个用户对不同数据对象的存取权限。当用户对数据库访 问时首先检查用户的存取权限。防止不合法用户对数据库的存取。

强制存取控制方法:每一个数据对象被(强制地)标以一定的密级，每一个用户也被 (强制地)授予某一个级别的许可证。系统规定只有具有某一许可证级别的用户才 能存取某一个密级的数据对象。

**6.** (1)GRANT ALL PRIVILEGES ON Student,Class

TOU1

WITH GRANT OPTION ；

1. GRANT SELECT,UPDATEC家庭住址),DELETE ON Student TO U2；
2. GRANT SELECT ON Class TO PUBLIC；
3. GRANT SELECT,UPDATE ON Student TO Rl；
4. GRANT Rl TO Ul- WITH ADMIN OPTION ；

GRANT SELECT ON 职工，部门

TO王明

1. 用户李勇对两个表有INSERT和DELETE权力。

GRANT INSERT,DELETE ON 职工，标门

TO李勇

1. 每个职工只对自己的记录有SELECT权力。

GRANT SELECT ON 职工

WHEN USER() = NAME

TO ALL；

1. 用户刘星对职工表有SELECT权力，对工资字段具有更新权力。

GRANT SELECT,UPDATECT.资)ON 职工

TO刘星

第四章

数据库安全性

1. 用户张新具有修改这两个表的结构的权力。

GRANT ALTER TABLE ON 职工，部门

TO张新；

1. 用户周平具有对两个表所有权力(读，插，改，删数据)，并具有给其他用户授权的 权力。

GRANT ALL PRIVILEGES ON 职工，部门

TO周平

WITH GRANT OPTION；

1. 用户杨兰具有从每个部门职工中SELECT最高工资、最低工资、平均工资的权力, 他不能查看每个人的工资。

CREATE VIEW部门工资AS

SELECT部门.名称,MAX(工资),MIN(工资),AVG(工资)

FROM职工，部门

WHERE职工.部门号=部门.部门号

GROUP BY职工.部门号

GRANT SELECT ON 部门工资

TO杨兰；

1. REVOKE SELECT ON 职工，部门 FROM 王明；
2. REVOKE INSERT,DELETE ON 职工，部门 FROM 李勇；
3. REOVKE SELECT ON 职工

WHEN USER() = NAME

FROM ALI ；

1. REVOKE SELECT,UPDATE ON 职工

FROM刘星；

1. REVOKE ALTER TABLE ON 职工，部门

FROM张新；

1. REVOKE ALL PRIVILEGES ON 职工，部门

FROM周平；

1. REVOKE SELECT ON 部门工资

54

数据库系统概论(第**5**版)同步辅导及习题全解

FROM杨兰；

DROP VIEW部门工资；

1. 區题主体是系统中的活动实体,既包括DBMS所管理的实际用户，也包括代表用户的各

进程。

客体是系统中的被动实体，是受主体操纵的，包括文件、基表、索引、视图等。对于主 体和客体,DBMS为它们每个实例(值)指派一个敏感度标记(Label)o

敏感度标记被分成若干级别，例如绝密(Top Secret)、机密(Secret)、可信(Confiden- tial)、公开(Public)等。主体的敏感度标记称为许可证级别，客体的敏感度标记称为 密级。

1. 假设要对关系变量S进行MAC控制，为简化起见，假设要控制存取的数据单位是

元组,则每个元组标以密级,如下表所示：

假设用户U1和U2的许可证级别分别为3和2,则根据规则U1能查得元组S1和 S2,可修改元组S2；而U2只能查得元组S1，H能修改元组Sl0

| S# | SNAME | STATUS | CITY | CLASS |
| --- | --- | --- | --- | --- |
| S1 | Smith | 20 | Ix)ndon | 2 |
| S2 | Jones | 10 | Paris | 3 |
| S3 | Clark | 20 | Ix>ndon | 4 |

这里假设系统的存取规则是：

1. 仅当主体的许可证级别大于或等于客体的密级时才能读取相应的客体；
2. 仅当主体的许可证级别等于客体的密级时才能写相应的客体。
3. !解題躅L审计功能是指数据库管理系统的审计模块在用户对数据库执行操作的同时把所有 操作自动记录到系统的审计日志中。

因为任何系统的安全保护措施都不是完美无缺的，蓄意盗窃破坏数据的人总可能 存在。利用数据库的审计功能，数据库管理员可以根据审计跟踪的信息，重现导致 数据库现有状况的一系列事件，找出非法存取数据的人、时间和内容等。

第五章

数据库完整性

知识网络图

第五章

数据库完整性

实体完整性

定义实体完整性

实体完整性检查和违约处理

参照完整性（定义参照完整性

参照完整性检查和违约处理

属性上的约束条件

定义

数据库完整性〈

用户定义的完整性＜

检査和违约处理

I元组上的约束条件

定义

检查和违约条件

完整性约束命名子句

完整性约束命名子句

修改表中的完整性限制

定义触发器 激活触发器 删除触发器

重难点

重点:牢固掌握数据库的完整性概念和DBMS完整性控制机制的三个方面，即完整性约束条件 的定义、完整性约束条件的检查和违约反应。需要举一反三的:用SQL语言定义关系模式的完整性 约束条件。包括定义每个模式的主码;定义参照完整性;定义与应用有关的完整性。

56

数据库系统概论(第**5**版)同步辅导及习题全解

| 完整性的类型 | 约束的类型 | 说明 |
| --- | --- | --- |
| 实体完整性 | PRIMARY KEY | 唯一标识每一行,保证用户不输入重复的数据，且创建一个索引来提高 性能，不允许空值 |
| UNIQUE | 防止非主关键字的重复，并保证创建一个索引来提高性能*，*允许空值 |
| 用户定义完整性 | DEFAULT | 指定在INSERT语句中没有明确提供一个值时，为该列提供的值 |
| CHECK | 指定在一个列中可接受的数据值 |
| 参照完整性 | FOREIGN KEY | 定义一个列或几个列的组合，他们的值匹配同一个表或另一个表中的关  键字 |

难点:RDBMS如何实现完整性的策略,RDBMS的数据库完整性的实现机制，包括实体完整性、 参照完整性和用户自己定义的完整性约束的定义机制、完整性检查机制和违背完整性约束条件时 RDBMS采取的预防措施。

知识点归纳

. 一、数据库完整性的定义

数据库的完整性是指数据的正确性和相容性。

.二、完整性约束条件的定义

完整性约束条件也称为完整性规则，是数据库中的数据必须满足的语义约束条件。（它表达了 给定的数据模型中数据及其联系所具有的制约和依存规则，用以限定符合数据模型的数据库状态 以及状态的变化，以保证数据的正确，有效和相容）

■三、数据库管理系统的完整性控制机制的功能

1. 提供定义完整性约束条件的机制 、
2. 提供完整性检查的方法
3. 进行违约处理

■四、实体完整性的定义（应用）、检查、违约处理

定义:关系模型中的实体完整性在CREATE TABLE中用PRIMARY KEY定义（对单属性构 成的码可以定义为列级约束条件，也可定义为表级约束条件，而对于多属性构成的码只有一种说明 方法.即定义为表级约束条件）。

第五章

检查与违约处理:1.检查主码是否唯一,如果不唯一则拒绝插入或修改。

2.检查主码的各个属性是否为空，只要有一个为空就拒绝插入或修改。

数据库完整性

■五、参照完整性的定义（应用）、检查、违约处理

定义:关系模型的参照完整性在CREATE TABLE中用FOREIGN KEY短语定义哪些列为外 码，用REFERENCE短语指明这些外码参照哪些表的主码。

检査:对被参照表和参照表进行增加、删除、修改操作时有可能会破坏参照完整性，必须检查以 保证这两个表的相容性。

违约处理：

|  |  |  |
| --- | --- | --- |
| 被参照表（例如Student） | 参照表（例如SC） | 违约处理 |
| 可能破坏参照完整性V=j | 插入元组 | 拒绝 |
| 可能破坏参照完整性V=] | 修改外码值（学号） | 拒绝 |
| 删除元组 | 可能破坏参照完整性 | 拒绝/级联删除/设置为空值 |
| 修改主码值匚 A | 可能破坏参照完整性 | 拒绝/级联修改/设置为空值 |

■六、用户定义的完整性的定义、检查、违约处理

用户定义完整性就是针对某一具体应用的数据必须满足的语义要求。

1.属性上的约束条件

在CREATE TABLE中定义属性的同时,可以根据应用要求定义属性上的约束条件，即属性值 限制,包括：

（1） 列值非空（NOT NULL）

（2） 列值唯一 （UNIQUE）

（3） 检查列值是否满足一个条件表达式（CHECK短语）

58

数据库系统概论(第**5**版)同步辅导及习题全解

检查和违约处理：当往表中插入元组或修改属性的值时，关系数据库管理系统将检査属性上的

约束条件是否被满足，如果不被满足则操作被拒绝。

2.元组上的约束条件

在CREATE TABLE语句中可以用CHECK短语定义元组上的约束条件，即元组级的限制。 同属性值限制相比,元组级的限制可以设置不同属性之间的取值的相互约束条件。（例如当学生的 性别是男时,其名字不可用Ms.打头）

检查和违约处理，当往表中插入元组或修改属性值时，关系数据库管理系统将检查元组上的约 束条件是否被满足，如果不满足则操作被拒绝执行。

典型例题

例1 将Student表中的Sno属性定义为码。

解析(1)在列级定义主码

CREATE TABLE Student

(Sno CHAR(9) PRIMARY KEY,

Sname CHAR(20) NOT NULL,

Ssex CHAR(2),

Sage SMALLINT,

Sdept CHAR(20)

);

(2)在表级定义主码

CREATE TABLE Student

(Sno CHAR(9),

Sname CHAR(20) NOT NULL,

Ssex CHAR(2),

Sage SMALLINT,

Sdept CHAR( 20),

PRIMARY KEY (Sno)

);

例**2** 显示说明参照完整性的违约处理示例。

CREATE TABLE SC

解析(SnoCHAR(9) NOT NULL, 、

Cno CHAR(4) NOT NULL,

Grade SMALLINT,

第五章

PRIMARY KEY(Sno,Cno),

FOREIGN KEY (Sno) REFERENCES Student(Sno)

数据库完整性

ON DELETE CASCADE / \*级联删除SC表中相应的元组\* /

ON UPDATE CASCADE, / \*级联更新SC表中相应的元组\* /

FOREIGN KEY (Cno) REFERENCES Course(Cno)

ON DELETE NO ACTION

/ \*当删除course表中的元组造成了与SC表不一致时拒绝删除\* /

ON UPDATE CASCADE

/ \*当更新course表中的cno时，级联更新SC表中相应的元组\* /

);

例**3**当学生的性别是男时，其名字不能以Ms.打头。

CREATE TABLE Student

解析(SnoCHAR(9),

Sname CHARC8) NOT NULL,

Ssex CHAR(2),

Sage SMALLINT,

Sdept CHARC20),

PRIMARY KEY (Sno),

CHECK (Ssex='女'OR Sname NOT LIKE 'Ms. %')

/ \*定义了元组中Sname和Ssex两个属性值之间的约束条件\* /

I

性别是女性的元组都能通过该项检查，因为Ssex=，女，成立； 当性别是男性时，要通过检查则名字一定不能以Ms.打头。

例**4** 建立教师表TEACHER,要求每个教师的应发工资不低于3000元。 应发工资是工资列Sal与扣除项Deduct之和。

CREATE TABLE TEACHER

60

数据库系统概论(第**5**版)同步辅导及习题全解

解析 (Eno NUMERIC(4) PRIMARY KEY / \* 在列级定义主码 \* /

Ename CHAR(IO),

Job CHAR(8),

Sal NUMERIC(7,2),

Deduct NUMERIC(7,2),

Deptno NUMERIC]2),

CONSTRAINT TEACHERFKey FOREIGN KEY(Deptno) REFERENCES DEPT(Deptno),

CONSTRAINT Cl CHECK (Sal + Deduct >= 3000)

);

例 **5** 设有如下所示的关系 S(S# ,SNAME,AGE,SEX)、C(C# ,CNAME,TEACHER)和 SC(S

# ,C# ,GRADE),试用关系代数表达式表示下列查询语句：

1. 检索“程军”老师所授课程的课程号(C#)和课程名(CNAME)。
2. 检索年龄大于21的男学生学号(S#)和姓名(SNAME)。
3. 检索至少选修“程军"老师所授全部课程的学生姓名(SNAME)。
4. 检索”李强"同学不学课程的课程号(C#)。
5. 检索至少选修两门课程的学生学号(S#)。
6. 检索全部学生都选修的课程的课程号(C#)和课程名(CNAME)。
7. 检索选修课程包含“程军”老师所授课程之一的学生学号(S#)o
8. 检索选修课程号为kl和k5的学生学号(S#)。
9. 检索选修全部课程的学生姓名(SNAME)O
10. 检索选修课程包含学号为2的学生所修课程的学生学号(S#)o
11. 检索选修课程名为“C语言”的学生学号(S#)和姓名(SNAME)。

解析本题各个查询语句对应的关系代数表达式表示如下：

1. HC# ,CNAME(oTEACHER='程军70)
2. IIS#, SNAME( aAGE>21 A SEX='男'(C))
3. USNAME{sx[IIS#,C#(sc)：TIC#((jTEACHER='程军'(C))]}
4. IIC#(C)- HC#(aSNAME=4李强，(S)xSC)
5. US甘((£1] = [4]A[2]乂[5] (SC X SC))
6. EC# ,CNAME(Cx(US# ,C# (scEllS# (S?))
7. US#(SCxITC#((jTEACHER='程军，(C)))
8. HS#,C#(sc)：IlC#((jC# = 'kl'V C# = 'k5'(C))
9. HSNAME{sx[IlS# ,C#(sc) + HC#(C)]}
10. ns# ,C#(sc)iITC甘(<jS# = '2'(SC))

\

1. US#,SNAME{sx[IlS#(SCx aCNAME=<C语言，(C))]}

例**6** 定义一个BEFORE行级触发器，为教师表Teacher定义完整性规则“教授的工资不得低于

第五章

5000元，如果低于5000元，自动改为5000元”。

解析 CREATE TRIGGER Insert\_Or\_Update\_Sal

数据库完整性

BEFORE INSERT OR UPDATE ON Teacher

/ \*触发事件是插入或更新操作\* /

FOR EACH ROW / \*行级触发器\* /

BEGIN / \*定义触发动作体，是PL/SQL过程块\* /

IF (new. Job=4教授')AND (new. Sal V 5000)

THEN new. Sal : = 5000 ；

END IF；

END；

课后习题解答

1. 解题欧，数据库的完整性是指数据的正确性和相容性。
2. 解题**gk**数据的完整性和安全性是两个不同的概念，但是有一定的联系。前者是为了防止数

据库中存在不符合语义的数据，防止错误信息的输入和输出，即所谓垃圾进垃圾出 所造成的无效操作和错误结果。后者是保护数据库以防止恶意的破坏和非法的存 取。也就是说，安全性措施的防范对象是非法用户和非法操作，完整性措施的防范 对象是不合语义的数据。

1. 解题暨，完整性约束条件是指数据库中的数据应该满足的语义约束条件。一般分为六类:静态列

级约束、静态元组约束、静态关系约束、动态列级约束、动态元组约束、动态关系约束。

1. 匯厘关系数据库管理系统的完整性控制机制应具有三个方面的功能：（1）定义功能，即

提供定义完整性约束条件的机制；（2）检查功能，即检查用户发岀的操作请求是否 违背了完整性约束条件；（3）违约反应:如果发现用户的操作请求使数据违背了完 整性约束条件，则釆取一定的动作来保证数据的完整性。

62

数据库系统概论(第**5**版)同步辅导及习题全解

**5.**

度鉗过程

k关系数据库管理系统在实现参照完整性时需要考虑以下几个方面:

（1）外码是否可以接受空值。

（2）被参照关系的元组时的考虑，这时系统可能采取的作法有三种:

1）级联删除（CASCADES）；

2)受限删除(RESTRICTED)；

3）置空值删除（NULLIFIES）。

（3）在参照关系中插入元组时的问题,这时系统可能采取的作法有：1）受限插入;2）

递归插入。

（4）修改关系中主码的问题。一般是不能用UPDATE语句修改关系主码的。如果

需要修改主码值，只能先删除该元组,然后再把具有新主码值的元组插入到关系中。

如果允许修改主码,首先要保证主码的唯一性和非空性，否则拒绝修改。然后要区

分是参照关系还是被参照关系。

**6.**［解题灘A CREATE TABLE DEPT (Deptno NUMBER(2), Deptname VARCHAR(IO),

Manager VARCHAR( 10), PhoneNumber Char( 12)

CONSTRAINT PK\_SC RIMARY KEY(Deptno)) ； CREATE TABLE EMP

(Empno NUMBER(4), Ename VARCHAR(IO), Age NUMBER(2),

CONSTRAINT Cl CHECK (AageV = 60), Job VARCHAR(9),

Sal NUMBER(7,2), Deptno NUMBERC2), CONSTRAINT FK\_DEPTN()

FOREIGN KEY(Deptno)

REFFERENCES DEPT(Deptno))；

**7.**睥题对于违反实体完整性和用户定义的完整性的操作一般都釆用拒绝执行的方式进行

处理。而对于违反参照完整性的操作，并不都是简单地拒绝执行，有时要根据应用

语义执行一些附加的操作，以保证数据库的正确性。

**8.**瞩题gl電 CREATE ASSERTION人数限制

CHECK (50>=SELECT C()UNT( \* )

FROM (SELECT \* FROM Male

UNION

SELECT \* FROM Female)

AS Guest

);

第六章

关系数据理论

知识网络图

函数依赖

『规范化＜

（候选码 码

I外部码

2NF 3NF

范式』多值依赖 BCNF

〔4NF

关系数据理论〈

数据依赖的公理系统＜

Armstrong公理系统

自反律

增广律

传递律

合并规则

伪传递规则 分解规则

最小依赖集

'模式的分解＜

'模式分解的三个定义

分解的无损连续性和保持函数依赖性

、模式分解的算法

第六章

关系数据理论

64

数据库系统概论(第**5**版)同步辅导及习题全解

知识点归纳

重难点

分解是解决数据冗余的主要方法，也是规范化的一条原则:“关系模式有冗余问题就分解它”。

重点：了解什么是一个“不好”的数据库模式。什么是模式的插入异常和删除异常。规范化理 论的重要意义。牢固掌握数据依赖的基本概念，范式的概念，从1NF到4NF的定义，规范化的含义 和作用。需要举一反三的：四个范式的理解与应用，各个级别范式中存在的问题（插入异常、删除异 常、数据冗余）和解决方法。

难点:能够根据应用语义，完整地写出关系模式的数据依赖集合,并能根据数据依赖分析某一 个关系模式属于第几范式。各个级别范式的关系及其证明。本章内容的理论性较强。要通过具体 例子和习题练习理解和掌握理论知识。

■ 一、函数依赖：

1. 设R（U）是一个属性集U上的关系模式,X和Y是U的子集，若对于R（U）的任意一个可能 的关系■中不可能存在两个元组在X上的属性值相等，而在Y上的属性值不等，则称“X函数确 定Y”或“Y函数依赖于X”,记作X->Y0

所谓函数依赖是指关系中属性或属性组的值可以决定其他属性的值，设R（U）是属性集U上的 关系模式,X、Y是U的子集：

如果X和Y之间是1 ： 1关系（一对一关系），如学校和校长之间就是1 ： 1关系,则存在函数依 赖 X Y 和 Y X。

如果X和Y之间是1 ： n关系（一对多关系），如年龄和姓名之间就是1 ： n关系,则存在函数依 赖 Y-X。

如果X和Y之间是m： n关系（多对多关系）,如学生和课程之间就是m： n关系，则X和Y之 间不存在函数依赖。

在关系模式R（U）中，对于U的子集X和Y,如果X-Y,但Y不包含X,则称X-Y是非平凡 的函数依赖;若X-Y,但Y包含X,则称X-Y是平凡的函数依赖。

若X-Y,则X称为这个函数依赖的决定属性组,也称为决定因t（Determinant）0

1. 完全函数依赖：在R（U）中，如果X-Y,并且对于X的任何一个真子集X，，都有X，不依赖

F

Y,则称Y对X完全函数依赖，记作X—Yo

p

1. 部分函数依赖:若X—Y,但Y不完全函数依赖于X,则称Y对X部分函数依赖，记作X— Y 例中(Sno,Cno)f Grade是完全函数械赖,(Sno,Cno)->Sdept是部分函数依赖

Sno ->Sdept 成立,且 Sno 是(Sno, Cno)的真子集。

当存在部分依赖时，就会产生数据冗余。

第六章

1. 传递函数依赖:在R(U)中，如果(Y里X),Y\*X,Y—Z,Z里Y,则称Z对X传递函数依赖，记 *为:X竺Z*

关系数据理论

注：如果Y-X,即X一Y,则Z直接依赖于X。

*■二、码*

F

1. 候选码:设K为RVU,F>中的属性或属性组，若KK—UU (每个属性)，则K称为R的 候选码(Candidate Key)

K满足以下两个条件：

(DK完全函数决定该关系的所有其他属性。

(2)K的任何真子集都不能完全函数决定R的所有其他属性,K必须是最小的。

若候选码多于一个,则选定其中的一个作为主码(Primary Key),通常称之为码。

主属性(Prime Attribute):包含在任何一个候选码中的属性。

非主属性或非码属性:不包含在任何码中的属性。

由于码能唯一确定一个元组，所以关系的码函数决定该关系的所有属性;一个关系中的所有属 性都函数依赖于该关系的码。

1. 外部码：关系模式R中属性或属性组X并非R的码，但X是另一个关系模式的码，则称X 是R的外部码(Foreign Key)也称外码。

.三、范式

各种范式之间存在联系：

1NFO2NFO3NFZ)BCNKD4NFZ)5NF

1. INF：如果一个关系模式R的所有属性都是不可分的基本数据项，则R61NF。

第一范式是对关系模式的最起码的要求。不满足第一范式的数据库模式不能称为关系数据 库;简而言之，第一范式就是无重复的列，关系数据库研究的关系都是规范化的关系。但是满足第

一范式的关系模式并不一定是一个好的关系模式。

关系模式s—L—C（Sn。，Sdept, Sloe, Cno, Grade）,其中Sloe为学生住处，假设每个系的学生 住在同一个地方。

F

函数依赖包括：（Sno, Cno） ► Grade

非主属性Sdept和Sloe部分函数依赖于码（Sno, Cno） o 插入异常:如未选课的学生不能插入（见图6. 1）。

数据库系统概论(第**5**版)同步辅导及习题全解

| P  (Sno. Cno) Sdept | Sno —► Sdept |
| --- | --- |
| p  (Sno, Cno) Sloe •/ | Sdept 粵►Sloe |
| S—L—C 的码为(Sno, Cno) 0 |  |
| S-L-C满足第一范式。 |  |

S—L—C

图6.1

| Sno | Sdept | Sloe | Cno | Grade |
| --- | --- | --- | --- | --- |
| SI | CS | BLD1 | Cl | 95 |
| SI | CS | BLD1 | C2 | 90 |
| SI | CS | BLD1 | C4 | 88 |
| S3 | IS | BLD2 | C2 | 70 |
| S4 | IS | BLD2 | C3 | 78 |
| ► • | : | : | : | : |
| S7 | PHY | BLD2 | NULL | NULL |

如:学生S7已入学，且Sdept=PHY,Sloc=BLD2，但还未选课，因此主码Cno为空值，所以学生 已有信息无法插入。

删除异常:如放弃修一门课，只选修这一门课的学生被删除（见图6. 2）。

S-L-C

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sno | Sdept | Sloe | Cno | Grade |
| SI | CS | BLD1 | Cl | 95 |
| SI | CS | BLD1 | C2 | 90 |
| SI | CS | BLD1 | C4 | 88 |
| S3 | IS | BLD2 | C2  \ | 70 |
| —S4 | IS | ———— | ea |  |

“S4”的其他信息一并被删除

图6. 2

如:S4不再选择课程C3,需删除C3数据项。

由于C3为主属性，删除C3必须删除整个元组，导致S4其他信息丢失(见图6. 3)。

| S—L—C  \ | | | | |
| --- | --- | --- | --- | --- |
| Sno | Sdept | Sloe | Cno | Grade |
| S1 | CS-\*2S | BLD1-\*BLD2 | Cl | 95 |
| S1 | CS-\*2S | BLD1—BLD2 | C2 | 90 |
| S1 | CS->2S | BLD1->BLD2 | C4 | 88 |
| S3 | IS | BLD2 | C2 | 70 |
| S4 | IS | BLD2 | C3 | 78 |

第六章

关系数据理论

图6. 3

如:学生S1转系，由CS系转至IS系。

Sdept和Sloe被重复存储，重复次数为S1选课数，修改时需无遗漏地考察每个相关元组，比较 复杂。

原因:存在对码的冗余依赖。Sdept. Sloe部分函数依赖于码。

解决方法:将S-L-C分解为两个关系模式，以消除这些部分函数依赖。

S—C(Sno, Cno, Grade) S—L(Sno, Sdept, Sloe)

1. 2NF：若RE INF,且每一个非主属性完全函数依赖于码，则RE2NF。

简而言之，第二范式就是每一行被码唯一标识。

S—L—C(Sno, Sdept, Sloe, Cno, Grade) C INF

S—L—C(Sno, Sdept, Sloe, Cno, Grade) G2NF

S—C(Sno, Cno, Grade) £ 2NF

S—L(Sno, Sdept, Sloe) £ 2NF

釆用投影分解法将一个INF的关系分解为多个2NF的关系，可以在一定程度上减轻原1NF关 系中存在的插入异常、删除异常、数据冗余度大、修改复杂等问题。但将一个1NF关系分解为多个 2NF的关系，并不能完全消除关系模式中的各种异常情况和数据冗余。

1. 3NF：关系模式R<U,F>中若不存在这样的码X、属性组Y及非主属性Z(Z? Y),使得X -Y, Y-> X, Y—Z成立，即每个非主属性都不传递依赖于R的码，则称R<U,F> 6 3NFO若R 63NF,则每一个非主属性既不部分依赖于码也不传递依赖于码。

简而言之，第三范式(3NF)要求一个数据库表中不能包含其他表中已包含的非码信息。

例:S-L(Sno, Sdept, Sloe) E 2NF S-L(Sno, Sdept, Sloe) E 3NF

S—D(Sno,Sdept) £ 3NF D一L(Sdept, Sloe) E 3NF

如果R63NF,则R也是2NF。

局部依赖和传递依赖是模式产生数据冗余和操作异常的两个重要原因。

数据库系统概论(第**5**版)同步辅导及习题全解

由于3NF模式中不存在非主属性对候选码的局部依赖和传递依赖，因此一定程度上解决原 2NF关系中存在的插入异常、删除异常、数据冗余度大、修改复杂等问题具有较好的性能。将一个 2NF关系分解为多个3NF的关系后，仍然不能完全消除关系模式中的各种异常情况和数据冗余。

4. BC范式(BCNF):关系模式R<U,F>61NF,若X—Y且Y->X时X必含有码，即每个属性 都不传递依赖于R的码，则R<U,F> 6BCNFO

等价于:每一个决定因素都包含码，即消除任何属性对码的部分和传递函数依赖。

若REBCNF,所有非主属性对每一个码都是完全函数依赖;所有的主属性对每一个不包含它 的码,也是完全函数依赖;没有任何属性完全函数依赖于非码属性。

充分

R 6BCNF馈R 63NFO如果R63NF,且R只有一个候选码。

R 6BCNF ~~凄冬、~~R 63NF。

不必要

例1关系模式WSC(W,S,C)中，W表示仓库,S表示保管员，C表示商品。假设每个仓库有若 干个保管员，有若干种商品。每个保管员保管所在仓库的所有商品，每种商品被所有保管 员保管。

典型例题

|  |  |  |
| --- | --- | --- |
| W | S | C |
| W1 | S1 | C1 |
| W1 | S1 | C2 |
| W1 | S1 | C3 |
| W1 | S2 | C1 |
| W1 | S2 | C2 |
| W1 | S2 | C3 |
| W2 | S3 | C4 |
| W2 | S3 | C5 |
| W2 | S4 | C4 |
| W2 | S4 | 、 C5 |

解析•按照语义对于W的每一个值，S有一个完整的集合与之对应而不问C取何值。所以

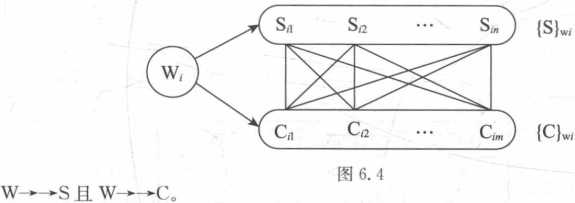
W—S。

•如图所示对应W的某一个值的全部S值记作{S}w(表示此仓库工作的全部保管员)。

•全部C值记作{C}wi(表示在此仓库中存放的所有商品)。

•应当有{S}wi中的每一个值和{C}wi中的每一个C值对应。

・于是{S}wi与{C}wi之间正好形成一个完全二分图，因而Wf f S。 由于C与S的完全对称性，必然有w'f C成立(见图6.4)0



第六章

关系数据理论

例**2**已知学生关系模式

S( Sno, Sname, SD, Sdname, Course, Grade)

其中：Sno学号Sname姓名、SD系名、Sdname系主任名、Course课程、Grade成绩。

1. 写出关系模式S的基本函数依赖和主码。
2. 原关系模式S为几范式？为什么？分解成高一级范式，并说明为什么。
3. 将关系模式分解成3NF,并说明为什么。
4. 写出关系模式S的基本函数依赖和主码。

解析关系模式S的基本函数依赖如下：

Sno-\* Sname, SE>-> Sdname, Sno—>SD, ( Sno, Course) —►Grade

关系模式S的码为：(Sno,Course) o

1. 原关系模式S为几范式？为什么？分解成高一级范式，并说明为什么。

解析 原关系模式S属于1NF。码为(Sno,Course),非主属性中的成绩完全依赖于码，而其他非主 属性对码的函数依赖为部分函数依赖，所以不属于2NF。

消除非主属性对码的函数依赖为部分函数依赖，将关系模式分解成2NF如下：

Sl( Sno, Sname, SD, Sdname)

S2 (Sno, Course, Grade)

1. 将关系模式分解成3NF,并说明为什么。

解析将上述关系模式分解成3NF如下：

关系模式S1中存在Sn(LSD,SDf Sdname,

即非主属性Sdname传递依赖于Sno,所以S1不是3NF0

进一步分解如下：

Sll( Sno, Sname, SD)

S12( SD, Sdname)

分解后的关系模式Sil. S12满足3NF。

对关系模式S2不存在非主属性对码的传递依赖,故属于3NF。

所以，原模式 S(Sno,Sname,SD,Sdname,Course,Grade)按如下分解满足 3NFO

1. 1( Sno, Sname, SD)

S12( SD, Sdname)

1. (Sno, Course*，*Grade)

数据库系统概论(第**5**版)同步辅导及习题全解

设有如下关系R

| 课程名 | 教师名 | 教师地址 |
| --- | --- | --- |
| C1 | 马千里 | D1 |
| C2 | 于得水 | D1 |
| C3 | 余快 | D2 |
| C4 | 于得水 | D1 |

1. 它为第几范式？为什么？
2. 是否存在删除操作异常？若存在，则说明是在什么情况下发生的。
3. 将它分解为高一级范式，分解后的关系是如何解决分解前可能存在的删除操作异常问 题的？

(1)它为第几范式？为什么？

解析 它是2NF。因为R的候选关键字为“课程名”。

依赖关系：课程名-教师名，教师名-课程名，教师名-教师地址，所以课程名-教师地址。

即存在非主属性“教师地址”对候选关键字课程名的传递函数，因此R不是3NF。但因为不

存在非主属性对候选关键字的部分函数依赖，所以R是2NF。

(2)是否存在删除操作异常？若存在，则说明是在什么情况下发生的。

解析存在。当删除某门课程时会删除不该删除的教师的有关信息。

(3)将它分解为高一级范式，分解后的关系是如何解决分解前可能存在的删除操作异常问题的?

解析分解为高一级范式如图所示'。

| 课程名 | 教师名 |
| --- | --- |
| C1 | 马千里 |
| C2 | 于得水 |
| C3 | 余快 |
| C4 | 于得水 |

R1如下:

| 教师名 | 教师地址 |
| --- | --- |
| 马千里 | D1 |
| 于得水\ | D1 |
| 余快 | D2 |

R2如下:

分解后，若删除课程数据时，仅对关系R1进行操作,教师地址信息在关系R2中仍然保留，不

会丢失教师方面的信息。

例**4**设某商业集团数据库中有一关系模式R如下：

R (商店编号，商品编号，数量，部门编号，负责人)

如果规定：

71

1. 每个商店的每种商品只在一个部门销售；
2. 每个商店的每个部门只有一个负责人；

第六章

1. 每个商店的每种商品只有一个库存数量。

试回答下列问题：

关系数据理论

1. 根据上述规定，写出关系模式R的基本函数依赖；

解析 关系模式S的基本函数依赖如下：

(商店编号，商品编号)部门编号，

(商店编号，部门编号)-负责人，

(商店编号，商品编号)f数量。

1. 找出关系模式R的候选码；

解析 关系模式R的码为：(商店编号，商品编号，部门编号)。

1. 试问关系模式R最高已经达到第几范式？为什么？

解析 原关系模式R是属于1NF的，码为(商店编号，商品编号，部门编号)，非主属性对码的函数 依赖全为部分函数依赖，所以不属于2NF。

消除非主属性对码的函数依赖为部分函数依赖，将关系模式分解成2NF如下：

R1(商店编号，商品编号，部门编号，数量)

R2(商店编号.部门编号，负责人)。

⑷如果R不属于3NF,请将R分解成3NF模式集。

解析将R分解为

R1(商店编号，商品编号，部门编号，数量)

R2(商店编号，部门编号，负责人)

分解后的R不存在传递的函数依赖,所以分解后的R已经是3NF。

例 **5** 设有关系 STUDENT(S# ,SNAME, SDEP1\MNAME,CNAME,GRADE), (S#， CNAME)为候选码，设关系中有如下函数依赖：

(S# ,CNAME)-\*SNAME,SDEPT,MNAME

S# — SNAME, SDEPT, MNAME

(S# ,CNAME)—GRADE

SDEPT-\* MNAME

试求下列问题：

1. 关系STUDENT属于第几范式？并说明理由。
2. 如果关系STUDENT不属于3NF,请将关系STUDENT逐步分解为3NF。

要求：写出达到每一级范式的分解过程，并指明消除什么类型的函数依赖。

解析(1)关系STUDENT是INF,因为F中存在非主属性SNAME,SDEPT,MNAME对候选码 (S# ,CNAME)的部分函数依赖，所以达不到2NF。

数据库系统概论(第**5**版)同步辅导及习题全解

(2)①首先消除部分函数依赖(S# ,CNAME)->SNAME,SDEPT,MNAME,将关系分解为： R1(S# ,SNAME,SDEPT,MNAME),R2(S# ,CNAME,GRADE)

RKS# ,SNAME,SDEPT,MNAME),S#为候选码,R1的函数依赖集为：

Fl = {S#—SNAME,SDEPT,MNAME,SDEPTf MNAME}

R2(S# ,CNAME,GRADE),S#为候选码,R2的函数依赖集为：

F2= {(S# (NAME)—GRADE}

②在关系R1中还存在非主属性MNAME对候选码S#的传递函数依赖S#-MNAME,所 以将R1进一步分解：

R11(S# ,SNAME,SDEPT) ,R12(SDEPT,MNAME)

其中：

R11(S# ,SNAME,SDEPT)以S#为候选码,Rll的函数依赖集为：

Fll = {S#->SNAME,SDEPT}

R12(SDEPT,MNAME)以SDEPT为候选码,R12的函数依赖集为：

F12 = {SDEPT—MNAME}

在R2. Rll. R12关系模式中都消除了非主属性对码的传递函数依赖,所以上述三个关系模 式均是3NFO

课后习题解答

1.匯亜留，定义1：设R(U)是属性集U上的关系模式。X,Y是属性集U的子集。若对于R

(U)的任意一个可能的关系r,r中不可能存在两个元组在X上的属性值相等，而在 Y上的属性值不等，则称X函数确定Y或Y函数依赖于X,记作X-Yo (即只要X 上的属性值相等,Y上的值一定相等。)

术语和记号：

X—Y,但Y不是X的子集，则称X—Y是非平凡的函数依赖。若不特别声明，总是 讨论非平凡的函数依赖。 '

X-Y,但Y是X的子集，则称X-Y是平凡的函数依赖。

若X—Y,则X叫做决定因素(Determinant)。

若 X->Y, Y—X,则记作 X-Y。

若Y不函数依赖于X,则记作X-Y。

定义2：在R(U)中，如果X-Y,并且对于X的任何一个真子集X，，都有X—Y,则 称Y对X完全函数依赖。

\

若X—Y,但Y不完全函数依赖于X,则称Y对X部分函数依赖。

73

第六章

关系数据理论

定义3 :若关系模式R的每一个分量是不可再分的数据项,则关系模式R属于第一 范式(1NF)O

定义4：若关系模式R61NF,且每一个非主属性完全函数依赖于码，则关系模式R E2NFO (即1NF消除了非主属性对码的部分函数依赖则成为2NF)。

定义5：关系模式R<U,F>中若不存在这样的码X、属性组Y及非主属性Z(Z不 是Y的子集)使得X—Y,Y—X,Y—Z成立，则称R<U,F>63NFO

定义6：关系模式R<U,F>E1NFO若X—Y且Y不是X的子集时,X必含有码, 则 R<U,F>eBCNFo

定义7：关系模式R<U,F>E1NF,如果对于R的每个非平凡多值依赖X-Y—Y

不是X的子集,Z=U—X—Y不为空)，X都含有码，则称R<U,F>E4NFO

关系模式:学生 S(S# ,SN,SB,DN,C# ,SA)

班级 C(C# ,CS,DN,CNUM,CDATE)

系 D(D# ,DN,DA,DNUM)

学会 P(PN,DATE1,PA,PNUM)

学生一学会 SP(S# ,PN,DATE2)

其中,S#一学号,SN一姓名，SB一出生年月，SA一宿舍区，

C甘一班号,CS—专业名，CNUM一班级人数,CDATE一入校年份，

D# 一系号,DN一系名，DA—系办公室地点,DNUM—系人数，

PN一学会名，DATE1一成立年月，PA一地点,PNUM一学会会员人数，

DATE2—入会年份

依据上面给出的语义，写出每个关系模式的极小函数依赖集：

S：S#f SN,S#f SB,S#->C# ,C#->DN,DN—SA

/ \*一个系的学生住在同一宿舍区\* /

C ： C 井 f CS, C # f CNUM ,C # f CDATE, CADN, (CS, CDATE)->C #

/\*因为每个专业每年只招一个班\* /

D：D#f DN,DN->D甘DA,D#->DNUM

/\*按照实际情况，系名和系号是一一对应的\* /

P ； PN-> DATEL PN—PA*，*PN-\*PNUM

SP：(S# ,PN1DATE2

数据库系统概论(第**5**版)同步辅导及习题全解

/ \*学生参加某学会有一个入会年份\*/

S中存在的传递函数依赖：

因为C# ,C#f DN,所以存在传递函数依赖DN,

因为C#—DN,DN->SA,所以存在传递函数依赖C#—SA,

因为S#—C# ,C#f DN,DNf SA,所以存在传递函数依赖S#-\*SA0

C中存在的传递函数依赖：

因为C#—CS,CS-DN,所以存在传递函数依赖C#—DN。

函数依赖左部是多属性的情况：

(S# ,PN)-\*DATE2和(CS,CDATE)-\*C#函数依赖左部具有2个属性，他们都是 完全函数依赖，没有部分函数依赖的情况。

关系模式C和D都有2个候选码。

| 关系 | 候选码 | 外部码 | 全码 |
| --- | --- | --- | --- |
| S | S# | C#,DN | 无 |
| C | C# 和(CS,CDATE) | DN | 无 |
| D | D# 和 DN | 无 | 无 |
| P | PN | 无 | 无 |
| SP | (S#,PN) | S#,PN | 无 |

解：(1)关系模式如下：

学生：S( Sno, Sname, Sbirth, Dept, Class ,Rno)

班级：C( Class, Pname, Dept, Cnum ,Cyear)

系：D( Dept, Dno, Office, Dnum)

学会：M( Mname, Myear, Maddr, Mnum)

(2)每个关系模式的最小函数依赖集如下：

1. 学生 S( Sno, Sname, Sbirth, Dept, Class, Rno)的最小函数依赖集如下：Sno->

Sname, Sno—►Sbirth, Sno—►Class, Class—►Dept, DEPT->Rno

传递依赖如下：

由于 Sno—► Dept,而 Dept—Sno,Dept-►Rrio(宿舍区)

所以Sno与Rno之间存在着传递函数依赖。

由 于 Class—Dept, Dept—Class, Dept-\*Rno

所以Class与Rno之间存在着传递函数依赖。

由于 Sno-^Class, Class->Sno, Class-\*Dept

所以Sno与Dept之间存在着传递函数依赖。

1. 班级C（Class, Pname, Dept, Cnum, Cyear）的最小函数依赖集如下：

、

Class-\* Pname, Class—Cnum, Class—Cyear, Pname-^Dept.

由 于 Class-\*Pname, Pname—\*-Class, Pname—►Dept

第六章

所以Class与Dept之间存在着传递函数依赖。

1. 系D（Dept,Dno,Office,Dnum）的最小函数依赖集如下：

关系数据理论

Dept—Dno, Dno—Dept, Dno-^Office, Dno—►Dnum

根据上述函数依赖可知,Dept与Office,Dept与Dnum之间不存在传递依赖。

1. 学会M（Mname,Myear,Maddr,Mnum）的最小函数依赖集如下：

Mname—\*Myear, Mname—►Maddr, Mname-^- Mnum 该模式不存在传递依赖。

（3）各关系模式的候选码、外部码，全码如下：

1. 学生S候选码:Sno ；外部码:Dept、Class ；无全码
2. 班级C候选码:Class；外部码:Dept；无全码
3. 系D候选码:Dept或Dno；无外部码;无全码
4. 学会M候选码:Mname；无外部码；无全码

**3.**匯题函k（1）合并规则:若XfY,X—Z,则有X-YZ；

设R<U,F>的任一关系r中任意的两个元组t、s：

若 t[X] = s[X],i XfY,可得 t[Y] = s[Y]；

由 X—Z,可得 t[Z] = s[Z]；

因为 t[Y] = sm，t[Z] = s|：Z],所以 t[YZ] = s[YZ]。

所以X-YZ成立，合并规则得证。

（2） 伪传递规则:若X-\*Y,XY—Z,则有XW—Z；

设RVU,F〉的任一关系r中任意的两个元组t、s：

若 t[XW] = s[XW]，则有 t[X] = s[X],t[W]=s[W]；

由 X-Y,可得 t[Y] = s[Y]；

因为 t[W] = s[W],t[Y] = s[Y],所以 t[WY]=s[WY]。

由 WY-Z,可得 t[Z] = s[Z]；

所以XW-Z成立，伪传递规则得证。

（3） 分解规则：X—Y,ZUY,有 X—Z。

设R<U,F>的任一关系r中任意的两个元组t、s：

若 t[X]=s[X],由 X-Y,可得 t[Y]=s[Y]； 由 ZUY,可得 t[Z] = s[Z]；

所以X—Z成立，分解规则得证。

**4.**[解题设Yxz = Yxz，，对于每一组(x,z,z，)都成立，现证其能推出定义6. 9的条件：设s,

76

数据库系统概论(第**5**版)同步辅导及习题全解

t是关系r中的两个元组，s[X] = t[X],由新定义的条件可知对于每一个z值，都. 对应相同的一组y值。这样一来，对相同的X值，交换y值后所得的元组仍然属 于关系r,即定义6. 9的条件成立；

如果定义6.9的条件成立，则对相同的x值，交换y值后所得的元组仍然属于关 系r,由于任意性及其对称性，可知每个z值对应相同的一组y值，所以Yxz = Yxz，，对于每一组(x,z,z，)都成立。

综上可知，新定义和定义6. 9的条件是等价的，所以新定义和定义6. 9是等价的。

**5**」解题懺區 (1)关系模式MSC (M , S , C)中，M表示专业,S表示学生,C表示该专业的必修

课。假设每个专业有多个学生，有一组必修课。设同专业内所有学生选修的必修课 相同，实例关系如下。按照语义对于M的每一个值M ,s有一个完整的集合与之对

应而不问C取何值，所以M-S。由于C与S的完全对称性，必然有M- C成立。 (2)关系模式ISA (I , S, A)中，1表示学生兴趣小组,S表示学生,A表示某兴趣小 组的活动项目。假设每个兴趣小组有多个学生，有若干活动项目。每个学生必须参 加所在兴趣小组的所有活动项目，每个活动项目要求该兴趣小组的所有学生参加。 按照语义有I-SJf A成立。

(3)关系模式RDP (R , D , P)中,R表示医院的病房,D表示责任医务人员，P表示 病人。假设每个病房住有多个病人，有多个责任医务人员负责医治和护理该病房的 所有病人。

' 假设有关系模式R(A,B,C,D,E),回答下面问题：

**6.**

(1)若A是R的候选码,具有函数依赖BC-DE,那么在什么条件下R是BCNF?

当属性组BC也是关系模式R的候选码时，R是BCNF。此时有:A-BC,BC-A 成立。

(2)如果存在函数依赖A—B,BC—D,DE-A,列出R的所有码。

R的候选码包括:ACE,BCE,CDE。

(3)如果(2)中的函数依赖成立,R属于**3NF**还是BCNF?

因为不存在传递函数依赖，所以R属于3NFO因为每个函数依赖的决定因素都不 包含码，所以R不属于BCNF。

⑴任何一个二目关系是属于3NF0

正确。因为关系模式中只有两个属性,所以无传递。

(2)任何一个二目关系是属禾BCNF。

正确。按BCNF的定义，若X-Y,且Y不是X的子集时，每个决定因素都包含码, 对于二目关系决定因素必然包含码。详细证明如下：(任何二元关系模式必定是 BCNF)O

证明:设R为一个二目关系R(A1,A2),则属性Al和A2之间可能存在以下几种依

赖关系：

1. Al—A2,但A2—Al,则关系R的码为A1,决定因素都包含码,所以，R是BC- NF。
2. Al—A2,A2—A1,则关系R的码为A2,所以决定因素都包含码,R是BCNFO

77

第六章

关系数据理论

78

数据库系统概论(第**5**版)同步辅导及习题全解

第七章

数据库设计

知识网络图

数据库设计的特点

数据库设计方法

需求分析

概念结构设计

数据库设计的基本步骤＜

逻辑结构设计

物理结构设计

数据库实施

数据库设计〈

数据库运行和维护

数据库设计过程中的各级模式

方法

'需求分析-

数据结构

数据字典＜

数据流

数据存取

处理过程

概念模型

E-R模型

概念结构设计〈扩展的E-R模型

概念结构设计

UML

E—R图向关系模型的转换

数据库设计〈

逻辑结构设计-

数据模型的优化

设计用户子模式

内容和方法

'物理结构设计〈

存取方法选择

确定数据库的存储结构

第七章

数据库设计

〔评价物理结构

重难点

重点:掌握数据库设计步骤和数据库设计过程中的各级模式设计方法。特别是数据库概念结 构的设计和逻辑结构的设计，这是数据库设计过程中最重要的两个环节。牢固掌握用E-R图来表 示概念模型的方法，掌握E-R图的设计以及E-R图向关系模型的转换。

难点：技术上的难点是E-R图的设计和数据模型的优化，包括对现实世界进行抽象的能力, 提取实体、属性、实体型之间的联系,正确划分实体与属性的能力。如何把第6章关系数据理论与 本章结合,用关系数据理论指导数据库的逻辑设计。真正的难点是理论与实际的结合。同学们一 般缺乏实际经验,缺乏对实际问题解决的能力，特别是缺乏应用领域的知识。而数据库设计需要设 计人员对应用环境、专业业务有具体深入的了解,这样才能设计出符合具体领域要求的数据库及其 应用系统。

知识点归纳

■ 一、数据库设计概述

1.数据库设计是指对于一个给定的应用环境，提供一个确定最优数据模型与处理模式的逻辑

80

数据库系统概论(第**5**版)同步辅导及习题全解

设计，以及一个确定数据库存储结构和存取方法的物理设计，建立起既能反映现实世界信息和信息 联系，满足用户数据要求和加工要求，又能被某个数据库管理系统所接受，同时能实现系统目标，并 有效存取数据的数据库。

1. 数据库系统生存期

数据库应用系统的开发也是一项软件工程，但又有自己的特点，称为数据库工程。

数据库系统的生存期:数据库应用系统从开始规划、分析、设计、实现、投入运行后的维护到最 后为新的系统取代而停止使用的整个期间。

通常包括六个阶段：

（1） 需求分析阶段

（2） 概念结构设计阶段

（3） 逻辑结构设计阶段

（4） 物理结构设计阶段

（5） 实施阶段

（6） 运行和维护阶段

1. 数据库设计过程的输入和输出

数据库设计过程的输入包括四部分内容：

（1） 总体信息需求

（2） 处理需求

（3） DBMS的特征

（4） 硬件和OS特征

数据库设计过程的输出包括两部分内容：

（1） 完整的数据结构,包括逻辑结构和物理结构。

（2） 基于数据库结构和处理需求的应用程序的设计原则。

1. 数据库设计方法学

数据库设计方法学是一些原则、工具和技术的组合，用于指导实施数据库系统的开发与研究。 一个好的数据库设计方法应该能在合理的期限内、以合理的工作量产生一个有实用价值的数 据库结构。

1. 数据库设计的步骤

数据库及其应用系统的设计分为以下六个阶段:需求分析、概念设计、逻辑设计、物理设计、实 施和运行维护。

■二、需求分析

1. 需求描述与分析

需求分析阶段应对系统的整个应用情况\*全面的、详细的调查，确定用户的目标，收集支持系 统总的设计目标的基础数据和对这些数据的要求，确定用户需求，并把这些要求写成用户和数据库 设计者都能够接受的文档。

第七章

确定用户需求可以通过对不同层次的企业管理人员进行个人访问得到。还应了解系统将来要 发生的变化，充分考虑系统可能的扩充和变动，使系统设计更符合未来发展方向。

数据库设计

1. 需求分析阶段的输入和输出

输入:总体信息需求、处理需求

输出:需求说明书

1. 需求分析的步骤

需求分析的步骤大致分为三步:即需求信息的收集、分析整理和评审。

1. 数据字典

数据字典:是对系统中数据的详尽描述，它提供对数据库数据描述的集中管理。它包含以下几 个部分:数据项、数据结构、数据流、数据存储和加工过程。

■三、概念设计

概念设计的目标是产生反映企业组织信息需求的数据库概念结构，即概念模式。

概念模式是独立于数据库逻辑结构，独立于支持数据库的DBMS,不依赖于计算机系统的。

1. 概念设计的必要性

（1） 将概念设计从设计过程中独立开来的好处；

（2） 概念模式在数据库的各级模式中的位置。

1. 概念模型

概念模型:可以看成是现实世界到机器世界过渡的一个中间层次。在设计数据库系统时，要把 现实世界的事物通过认识和抽象转换为信息世界的概念模型，再把概念模型转换为机器世界的数 据模型。

对概念模型的要求，主要有以下几个要点：

有丰富的语义表达能力，能表达用户的各种需求；

简洁、明晰、独立于机器、容易理解；

易于变动；

82

数据库系统概论(第**5**版)同步辅导及习题全解

易于向各种数据模型转换。

1. 概念设计的主要步骤

分三步完成：

1. 进行数据抽象，设计局部概念模式;
2. 将局部概念模式综合成全局概念模式；
3. 评审。
4. E-R模型的操作

E-R模型在数据库概念设计过程中经常需要进行变换，包括实体类型、联系类型和属性的分 裂、合并和增删等。

1. 实体类型的分裂：一个实体类型根据需要分裂成若干个实体类型，包括垂直分割、水平 分割。
2. 实体类型的合并:实体类型分裂的逆过程，包括垂直合并、水平合并。
3. 联系类型的分裂:一个联系类型可分裂成几个新联系类型。
4. 联系类型的合并:联系类型分裂的逆过程。

注意：

1. 在垂直分割时，码必须在分裂后的每个实体类型中出现。
2. 在联系类型合并时，所合并的联系类型必须是定义在相同的实体类型中。
3. 采用E-R方法的数据库概念设计

采用E-R方法进行数据库概念设计分成三步进行：

1. 设计局部E-R模式

确定局部结构范围;实体定义;联系定义;属性分配。

1. 设计全局E-R模式

确定公共实体类型;局部E-R模型的合并;消除冲突。

1. 全局E-R模式进行优化

实体类型的合并;冗余属性的消除;冗余联系的消除。

■四、逻辑设计

概念设计的结果是得到一个与DBMS无关的概念模式。而逻辑设计的目的是把概念设计阶段 设计好的基本ER图转换为与选用的具体机器上的DBMS所支持的数据模式相符合的逻辑结构

（包括数据库模式和外模式）。

1. 逻辑设计环境

在逻辑设计阶段主要输入下列信息：

、

（1） 独立于DBMS的概念模式

（2） 处理需求

第七章

（3） 约束条件

（4） 数据库管理系统特性

数据库设计

在逻辑设计阶段主要输出下列信息：

（1） 数据库管理系统可处理的模式

（2） 子模式

（3） 应用程序设计指南

（4） 物理设计指南

1. 逻辑设计的步骤

（1） 初步模式的形成

（2） 子模式设计

（3） 应用程序设计梗概

（4） 模式评价

（5） 修正模式

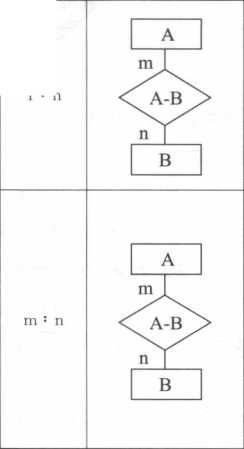
1. E-R模型向关系模型的转换

E-R模型转换关系数据库的一般规则：

（1） 将每一个实体类型转换成一个关系模式，实体的属性为关系模式的属性。

（2） 对于二元联系，按各种情况处理,如下面所示。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 二元关系 | ER图 | | | 转换成的关系 | 联系的处理 | 主码 | 外码 |
| 1: 1 |  | | | （2个关系） 模式A 模式B | （有两种） 方式（1）：把模 式B的主码，联 系的属性加入 模式A  方式（2）：把模 式A的主码，联 系的属性加入 模式B | （略） | （依据联系的 处理方式） 方式（1）：模 式B的主码 为模式A的 外码  方式（2）：表 A的主码为 表B的外码 |
|  | **A** |  |
|  | | |
|  | **B** |  |
|  | | |



84

数据库系统概论(第**5**版)同步辅导及习题全解

二元关系

ER图

1 2 n

（2个关系） 模式A 模式B

（3个关系） 模式A 模式B

模式A-B

| 联系的处理 | 主码 | 外码 |
| --- | --- | --- |
| 把模式B的主 码.联系的属性 加入模式A | （略） | 模式B的主 码为模式A 的外码 |
| 联系类型转换 成关系模式A  -B；  模式A- B的 属性  （a） 联系的属性  （b） 两端实体类 型的主码 | 两端实体类 型的主码一 起构成模式 A-B的主码 | 两端实体类 型的主码分 别为模式A 一B的外码 |

转换成的关系

4.关系数据库的逻辑设计涉及的一些概念要掌握

关系数据库的逻辑设计的过程主要是:

（1）导出初始关系模式

（2）规范化处理

■五、物理设计

对一个给定的逻辑数据模型选取一个最适合应用环境的物理结构的过程，称为数据库的物理 设计。所谓数据库的物理结构，主要指物理设备上的存储结构和存取方法。

物理设计的步骤

（1） 存储记录结构设计

（2） 确定数据存储安排

（3） 访问方法的设计

（4） 完整性和安全性

（5） 程序设计

■六、实施、运行与维护

1. 数据库的实现

实现阶段的主要工作有以下几点：

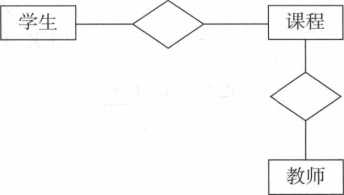
1. 建立实际数据库结构-
2. 试运行
3. 装入数据 '
4. 其他设计工作
5. 数据库的重新组织设计
6. 故障恢复方案设计
7. 安全性考虑
8. 事务控制
9. 运行与维护

运行与维护阶段的主要工作有以下几点：

1. 维护数据库的安全性与完整性控制及系统的转储和恢复;
2. 性能的监督、分析与改进；
3. 增加新功能;
4. 发现错误、修改错误。

例**1**某大学实行学分制，学生可根据自己的情况选修课程。每名学生可同时选修多门课程，每 门课程可由多位教师讲授；每位教师可讲授多门课程。其不完整的**E-R**图如图7. 1 所示。

1. 指出学生与课程的联系类型，完善**E-R**图厂
2. 指出课程与教师的联系类型，完善**E-R**图。
3. 若每名学生有一位教师指导，每个教师指导多名学生，则学生与教师是何联系？
4. 在原**E-R**图上补画教师与学生的联系，并完善**E-R**图。



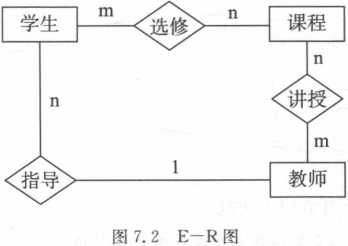
解析（1）学生与课程的联系类型是多对多联系。

86

（2） 课程与教师的联系类型是多对多联系。

（3） 学生与教师的联系类型是一对多联系。

（4） 完善本题E-R图的结果如图7. 2所示。

例**2** 假定一个部门的数据库包括以下的信息：

数据库系统概论(第**5**版)同步辅导及习题全解

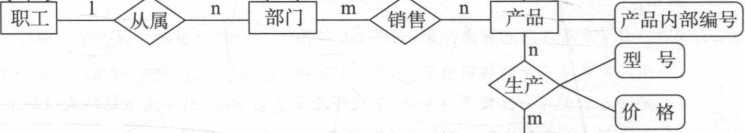
职工的信息：职工号、姓名、住址和所在部门。

部门的信息：部门所有职工、经理和销售的产品。

产品的信息：产品名、制造商、价格、型号及产品内部编号。 制造商的信息：制造商名称、地址、生产的产品名和价格。

试画出这个数据库的**E-R**图。

解析 本题对应的E-R图如图7. 3所示。

（职工号）（姓名）（牛址） （部门名）（经理） （产品名

制造商

（名称）（地址）



图7. 3 E-R图

例**3**某医院病房计算机管理中需要如下信息：

科室：科名，科地址,科电话，医生姓名

病房:病房号，床位号，所属科室名

医生：姓名，职称，所属科室名，年盼，工作证号

病人:病历号，姓名，性别，诊断，主管医生，病房号

第七章

其中，一个科室有多个病房、多个医生，一个病房只能属于一个科室，一个医生只属于一个 科室，但可负责多个病人的诊治，一个病人的主管医生只有一个。

完成如下设计：

数据库设计

1. 设计该计算机管理系统的E-R图；
2. 将该**E-R**图转换为关系模型结构；
3. 指出转换结果中每个关系模式的候选码。

解析(1)本题的E-R图如图7. 4所示。

1. 对应的关系模型结构如下：

科室(科名，科地址，科电话)

病房(病房号，床位号,科室名)

医生(工作证号，姓名，职称，科室名，年龄)

病人(病历号，姓名，性别，诊断，主管医生，病房号)

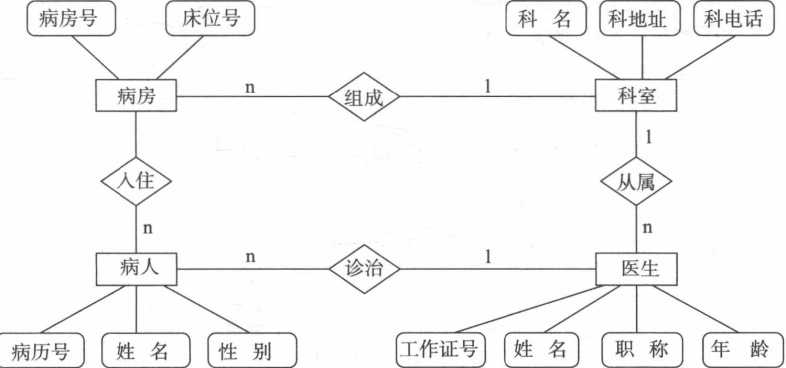
1. 每个关系模式的候选码如下：

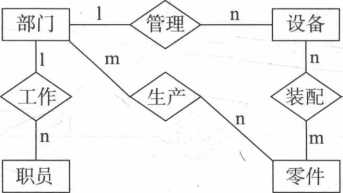
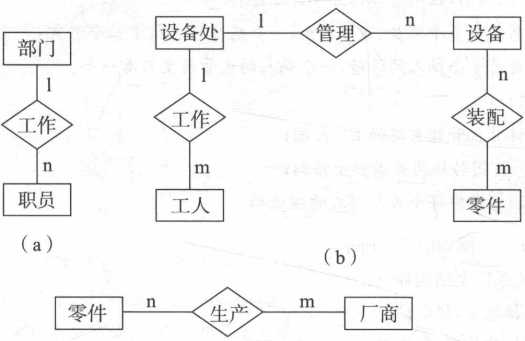
科室的候选码是科名；

病房的候选码是科室名+病房号；

医生的候选码是工作证号；

病人的候选码是病历号。





数据库系统概论(第**5**版)同步辅导及习题全解

例4图7. 5给出（a）、（b）和（c）三个不同的局部模型，将其合并成一个全局信息结构，并设置联 系实体中的属性（允许增加认为必要的属性，也可将有关基本实体的属性选作联系实体的 属性）。

解析

（c）

图7. 5各局部的E-R图 各实体构成如下：

部门:部门号，部门名，电话，地址

职员：职员号，职员名，职务（干部/工人），年龄，性别

设备处:单位号，电话，地址

工人:工人编号，姓名，年龄，性别

设备:设备号，名称，位置，价格

零件:零件号，名称，规格，价格

厂商:单位号，名称，电话，地址

解:汇总后的E-R图如图7. 6所示。

各类实体的属性为:

部门:部门号，部门名，电话，地址

职工:职工号,职工名，职务，年龄，性别

设备:设备号，名称，位置，价格

零件:零件号，名称，规格，价格

例**5** 一个图书借阅管理数据库要求提供下述服务：

（1） 可随时查询书库中现有书籍的品种、数量与存放位置。所有各类书籍均可由书号唯一 标识。

（2） 可随时查询书籍借还情况。包括借书人单位、姓名、借书证号、借书日期和还书日期。 我们约定，任何人可借多种书，任何一种书可为多个人所借，借书证号具有唯一性。

（3） 当需要时，可通过数据库中保存的出版社的电报编号、电话、邮编及地址等信息向出版 社增购有关书籍。我们约定，一个出版社可出版多种书籍，同一本书仅为一个出版社出版， 出版社名具有唯一性。

根据以上情况和假设，试作如下设计：

（1） 构造满足需求的E-R图。

（2） 转换为等价的关系模型结构。

解析（1）满足上述需求的E-R图如图7. 7所示。

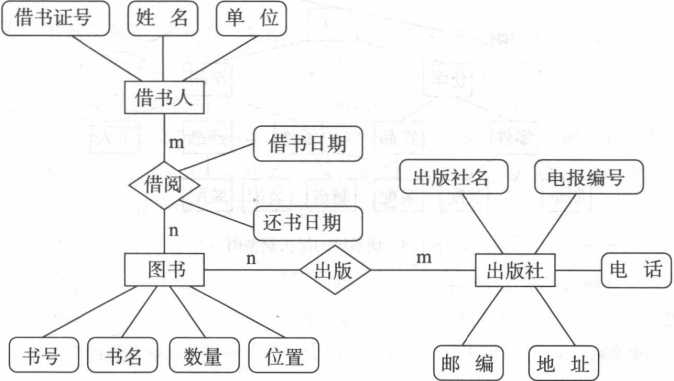
（2）转换为等价的关系模型结构如下；

借书人（借书证号，姓名，单位）

图书（书号，书名，数量，位置，出版社名）

出版社（出版社名，电报编号，电话，邮编，地址）

借阅（借书证号，书号，借书日期,还书日期）。



例**6**工厂（包括厂名和厂长名）需建立一个管理数据库存储以下信息：

90

数据库系统概论(第**5**版)同步辅导及习题全解

一个厂内有多个车间，每个车间有车间号、车间主任姓名、地址和电话。

一个车间有多个工人，每个工人有职工号、姓名、年龄、性别和工种。

一个车间生产多种产品，产品有产品号和价格。

一个车间生产多种零件,一个零件也可能为多个车间制造。零件有零件号、重量和价格。

一个产品由多种零件组成，一种零件也可装配出多种产品。

产品与零件均存入仓库中。

厂内有多个仓库,仓库有仓库号、仓库主任姓名和电话。

试：（1）给出相应的关系数据模型。

（2）画出该系统的层次模型图。

解析（1）相应的关系数据模型如下：

工厂（厂名，厂长名）

车间（车间号，主任姓名，地址，电话，厂名）

工人（职工号，姓名，年龄，性别，工种，车间号）

仓库（仓库号，主任姓名，电话，厂名）

产品（产品号，价格，车间号，仓库号）

零件（零件号，重量，价格，仓库号）

制造（车间号，零件号）

装配（产品号，零件号）

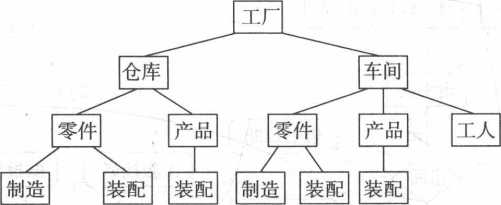
（2）该系统的层次模型图如图7. 8所示。

图7. 8该系统的层次模型图

课后习题解答

这里只概要列出数据库设计为:程的六个阶段：（1）需求分析；（2）概念结构设计;

（3）逻辑结构设计；（4）数据库物理设计;（5）数据库实施；（6）数据库运行和维护。 这是一个完整的数据库及其应用系统的设计过程。不仅包括设计数据库本身，还包 括数据库的实施、运行和维护。设计一个完善的数据库应用系统往往是上述六个阶 段的不断反复。

数据库结构设计的不同阶段形成数据库的各级模式，即：（1）在概念设计阶段形成 独立于机器特点、独立于各个数据库管理系统产品的概念模式,在本篇中就是E- R图；（2）在逻辑设计阶段将E-R图转换成具体的数据库产品支持的数据模型, 如关系模型，形成数据库逻辑模式,然后在基本表的基础上再建立必要的视图，形成 数据的外模式；（3）在物理设计阶段，根据数据库管理系统特点和处理的需要，进行 物理存储安排，建立索引，形成数据库内模式。

需求分析阶段的设计目标是通过详细调查现实世界要处理的对象（组织、部门、企业 等），充分了解原系统（手工系统或计算机系统）工作概况，明确用户的各种需求，然 后在此基础上确定新系统的功能。调查的内容是“数据”和“处理”，即获得用户对数 据库的如下要求：（1）信息要求，指用户需要从数据库中获得信息的内容与性质，由 信息要求可以导出数据要求，即在数据库中需要存储哪些数据；（2）处理要求，指用 户要完成什么处理功能，对处理的响应时间有什么要求，处理方式是批处理还是联 机处理；（3）安全性与完整性要求。

数据字典是系统中各类数据描述的集合。数据字典的内容通常包括：（1）数据项;

91

第七章

数据库设计

（2）数据结构；（3）数据流；（4）数据存储；（5）处理过程五个部分。其中数据项是数

据的最小组成单位，若干个数据项可以组成一个数据结构。数据字典通过对数据项 和数据结构的定义来描述数据流和数据存储的逻辑内容。数据字典的作用:数据字 典是关于数据库中数据的描述，在需求分析阶段建立,是下一步进行概念设计的基 础,并在数据库设计过程中不断修改、充实、完善。

**5**.座囲眉\ 概念结构是信息世界的结构，即概念模型，其主要特点是：（1）能真实、充分地反映 现实世界,包括事物和事物之间的联系，能满足用户对数据的处理要求，是对现实世 界的一个真实模型；（2）易于理解,从而可以用它和不熟悉计算机的用户交换意见, 用户的积极参与是数据库设计成功的关键;（3）易于更改，当应用环境和应用要求改 变时，容易对概念模型修改和扩充；（4）易于向关系、网状、层次等各种数据模型

92

数据库系统概论(第**5**版)同步辅导及习题全解

概念结构的设计策略通常有四种：1）自顶向下，即首先定义全局概念结构的框架,然 后逐步细化；2）自底向上，即首先定义各局部应用的概念结构，然后将它们集成起 来,得到全局概念结构；3）逐步扩张，首先定义最重要的核心概念结构，然后向外扩 充，以滚雪球的方式逐步生成其他概念结构，直至得到总体概念结构；4）混合策略, 即将自顶向下和自底向上相结合,用自顶向下策略设计一个全局概念结构的框架, 以它为骨架集成由自底向上策略中设计的各局部概念结构。

1. 匯厘W 实体:客观存在并可以相互区分的事物叫实体。

实体型:具有相同属性的实体具有相同的特征和性质,用实体名及其属性名集合来 抽象和刻画同类实体，称为实体型。

实体集：同型实体的集合称为实体集。

属性:实体所具有的某一特性，一个实体可由若干个属性来刻画。

码:唯一标识实体的属性集称为码。

实体联系图（E-R图）：提供了表示实体型、属性和联系的方法。

实体型:用矩形表示，矩形框内写明实体名；

属性:用椭圆形表示，并用无向边将其与相应的实体连接起来； 联系:用菱形表示,菱形框内写明联系名，并用无向边分别与有关实体连接起来。

同时在无向边旁标上联系的类型（1 ： 1,1 ： n或m ： n）。

1. 願函k各实体属性为：

系:系编号，系名

班级:班级号，班级名

教研室:教研室号,教研室

学生:学号，姓名，学历

课程:课程号，课程名

教员：职工号，姓名，职称

联系的属性:“选修”的属性为“成绩”

转换为关系模型如下：

系（系编号，系名，学校名）

班级（班级号,班级名，系编号）

教研室（教研室号,教研室，系编号）

学生（学号，姓名，学历，班级号，教员职工号）

课程（课程号，课程名）

教员（职工号，姓名，职称，教研室号）

选修（学号，课程号,成绩）

1. 陸■亀各实体属性为：

产品:产品号，产品名

零件:零件号，零件名

原材料:原材料号，原材料金，类别

仓库:仓库号，仓库名

各联系的属性为：

第七章

产品组成:使用零件量

零件制造:使用原材料量

数据库设计

零件存储:存储量

材料存放:存储量

转换为关系模型如下：

产品（产品号,产品名，仓库号）

零件:零件号，零件名

原材料:原材料号，原材料名，类别，仓库号，存放量）

仓库（仓库号，仓库名）

产品组成（产品号，零件号,使用零件量）

零件制造（零件号，原材料号,使用原材料量）

零件存储（零件号，仓库号,存储量）

**9.**

数据库的逻辑结构设计就是把概念结构设计阶段设计好的基本E-R图转换为与

选用的数据库管理系统产品所支持的数据模型相符合的逻辑结构。设计步骤为:

（1）将概念结构转换为一般的关系、网状、层次模型；（2）将转换来的关系、网状、层 次模型向特定数据库管理系统支持下的数据模型转换；（3）对数据模型进行优化。

**10.**匯厘函**L**其关系模型为（其中有下横线的属性是主码属性）:

系（系编号，系名，学校名）

班级（班级编号,班级名，系编号）

教研室（教研室编号，教研室，系编号）

学生（堂号，姓名，学历，班级编号，导师职工号）

课程（课程编号，课程名）

教员（职理，姓名，职称，教研室编号） 选课（学号，课程编号,成绩）

对应的关系模型为（其中有下横线的属性是主码属性）:

产品（产品号,产品名，仓库号）

零件（零件号，零件名）

原材料（原材料号，原材料名，类别，仓库号，存放量） 仓库（仓库号，仓库名）

产品组成（产品号，零件号,使用零件量）

94

数据库系统概论(第**5**版)同步辅导及习题全解

零件组成（零件号，原材料号，使用原材料量）

零件储存（零件号，仓库号，存储量）

习题7中设计的两个关系数据库的各个关系模式的码都用下划线注明，这些关系 模式都只有一个码，且都是唯一决定的因素，所以都属于BCNF。不会产生更新异 常现象。

1. 慳亟專 规范化理论为数据库设计人员判断关系模式的优劣提供了理论标准，可用以指导

关系数据模型的优化，用来预测模式可能出现的问题,为设计人员提供了自动产生 各种模式的算法工具，使数据库设计工作有了严格的理论基础。

1. 瓯囹区数据库在物理设备上的存储结构与存取方法称为数据库的物理结构，它依赖于给

定的DBMSO为一个给定的逻辑数据模型选取一个最适合应用要求的物理结构， 就是数据库的物理设计的主要内容。数据库的物理设计步骤通常分为两步：（1）确 定数据库的物理结构，在关系数据库中主要指存取方法和存储结构；（2）对物理结 构进行评价，评价的重点是时间效率和空间效率。

1. 数据库是用来对数据进行存储、管理与应用的，因此在实施阶段必须将原有系统中

的历史数据输入到数据库中。数据量一般都很大，而且数据来源于部门中各个不 同的单位。数据的组织方式、结构和格式都与新设计的数据库系统有相当的差距， 组织数据录入就要将各类源数据从各个局部应用中抽取出来，分类转换，最后组合 成符合新设计的数据库结构的形式，输入到数据库中。因此这样的数据转换、组 织入库的工作是相当费力费时的工作。特别当原系统是手工数据处理系统时，各 类数据分散在各种不同的原始表格、凭证、单据之中，数据输入工作量更大。保证 输入数据正确性的方法:为提高数据输入工作的效率和质量”，应该针对具体的应用 环境设计一个数据录入子系统，由计算机来完成数据入库的任务。在源数据入库 之前要釆用多种方法对它们进行检验,以防止不正确的数据入库。

**15**」糖题数据库的再组织是指:按原设计要求重新安排存储位置、回收垃圾、减少指针链等， 以提高系统性能。数据库的重构造则是指部分修改数据库的模式和内模式，即修 改原设计的逻辑和物理结构。数据库的再组织是不修改数据库的模式和内模式 的。进行数据库的再组织和重构造的原因:数据库运行一段时间后，由于记录不断 增、删、改，会使数据库的物理存储情况变坏，降低了数据的存取效率，数据库性能 下降，这时数据库管理员就要对数据库进行重组织。数据库管理系统一般都提供 用于数据重组织的实用程序。数据库風用环境常常发生变化，如增加新的应用或 新的实体，取消了某些应用,有的实体与实体间的联系也发生了变化等，使原有的 数据库设计不能满足新的需求,需要调整数据库的模式和内模式。这就要进行数 据库重构造了。

第八章

数据库编程

知识网络图

嵌入式SQL〈

过程化SQL】

数据库编程〈

［嵌入式SQL的处理过程

嵌入式SQL语句与主语言之间的通信 不用游标的SQL语句

使用游标的SQL语句

动态SQL

（过程化SQL的块结构

变量和常量

条件控制语句

循环控制语句

、错误处理

存储过程的优点 存储过程的用户接口

流程控制＜

存储过程

存储过程和函数

ODBC编程〈

函数的定义语句格式 函数的执行语句格式 修改函数

ODBC概述

ODBC工作原理

ODBC API 基础

函数

〔ODBC的工作流程

'JDBC编程

第八章

数据库编程

96

数据库系统概论(第**5**版)同步辅导及习题全解

知识点归纳

重点：了解SQL编程技术可以有效克服SQL实现复杂应用方面的不足，提高应用系统和RD- BMS间的互操作性。掌握嵌入SQL中游标的概念和使用方法;掌握PL/SQL和存储过程的基本概 念、基本结构、语句语法和用法。了解使用ODBC开发应用系统的体系结构，掌握ODBC API和 ODBC的应用程序的工作流程。

难点：理论联系实际，能够在实际安装的RDBMS上通过编程的方式开发应用程序，完成对数 据库的各种操作。能够使用ODBC来进行数据库应用程序的设计，使设计的应用系统可移植性好, 并且能同时访问不同的数据库，共享数据资源。

. 一、**SQL**语言两种不同的使用方式

•交互式

•嵌入式

SQL语言是非过程性语言。

事务处理应用需要高级语言。

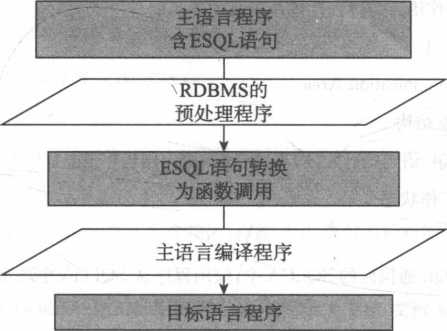
这两种方式细节上有差别，在程序设计的环境下,SQL语句要做某些必要的扩充。

主语言

•嵌入式SQL是将SQL语句嵌入程序设计语言中，被嵌入的程序设计语言，如C、C+ +、 Java,称为宿主语言，简称主语言。

处理过程

•预编译方法（见图8. 1）



第八章

数据库编程

为了区分SQL语句与主语言语句，所有SQL语句必须加前缀EXEC SQL,以分号（；）结束:

EXEC SQL VSQL 语句〉；

二、嵌入式**SQL**与主语言的通信

将SQL嵌入到高级语言中混合编程，程序中会含有两种不同计算模型的语句: •SQL语句

＞描述性的面向集合的语句

＞负责操纵数据库

•高级语言语句

＞过程性的面向记录的语句

＞负责控制程序流程

它们之间应该如何通信？

数据库工作单元与源程序工作单元之间的通信：

1. SQL通信区

•向主语言传递SQL语句的执行状态信息

•使主语言能够据此控制程序流程

1. 主变量

•主语言向SQL语句提供参数

•将SQL语句查询数据库的结果交主语言进一步处理

1. 游标

•解决集合性操作语言与过程性操作语言的不匹配

数据库系统概论(第**5**版)同步辅导及习题全解

1. SQL通信区

SQLCA： SQL Communication Area

SQLCA是一个数据结构

SQLCA的用途:SQL语句执行后,RDBMS反馈给应用程序信息。

•描述系统当前工作状态

•描述运行环境

这些信息将送到SQL通信区的SQLCA中，应用程序从SQLCA中取出这些状态信息，据此决 定接下来执行的语句。

SQLCA使用方法：

SQLCA中有一个存放每次执行SQL语句后返回代码的变量SQLCODEo

如果SQLCODE等于预定义的常量SUCCESS,则表示SQL语句成功，否则表示出错。

应用程序每执行完一条SQL语句之后都应该测试一下SQLCODE的值，以了解该SQL语句执 行情况并做相应处理。

1. 主变量(Host Variable)

在SQL语句中使用的主语言程序变量简称为主变量。

主变量的类型：

•输入主变量

•输出主变量

•指示变量：

一个主变量可以附带一个指示变量(Indicator Variable)

在SQL语句中使用主变量和指示变量的方法

1. 说明主变量和指示变量

BEGIN DECLARE SECTION

(说明主变量和指示变量)

END DECLARE SECTION

1. 使用主变量

•说明之后的主变量可以在SQL语句中任何一个能够使用表达式的地方出现

•为了与数据库对象名(表名、视图名、列名等)区别.SQL语句中的主变量名前要加冒号(：)作

为标志

3）使用指示变量

•指示变量前也必须加冒号（：）标志、

•必须紧跟在所指主变量之后

在SQL语句之外（主语言语句中）使用主变量和指示变量的方法:可以直接引用,不必加冒号。

第八章

（3）游标（cursor）

•游标是系统为用户开设的一个数据缓冲区，存放SQL语句的执行结果

数据库编程

•每个游标区都有一个名字

・用户可以用SQL语句逐一从游标中获取记录，并赋给主变量，交由主语言进一步处理

使用游标的步骤

1） 说明游标

•使用DECLARE语句

•语句格式 ■

EXEC SQL DECLARE <游标名 > CURSOR

FOR <SELECT 语句〉；

•功能

是一条说明性语句，这时DBMS并不执行SELECT指定的查询操作。

2） 打开游标

•使用OPEN语句

•语句格式

EXEC SQL OPEN V游标名〉；

•功能

打开游标实际上是执行相应的SELECT语句，把所有满足査询条件的记录从指定表取到缓冲 区中。

这时游标处于活动状态，指针指向查询结果之中的第一条记录。

3） 推进游标指针并取当前记录

•使用FETCH语句

•语句格式

EXEC SQL FETCH [[NEXT| PRIORI

FIRST I LAST] FROM] V游标名 >

INTO〈主变量〉[〈指示变量>][，〈主变量〉[V指示变量>]]...；

•功能

00

1

数据库系统概论(第**5**版)同步辅导及习题全解

向指定方向推动游标指针，然后将缓冲区中的当前记录取出来送至主变量供主语言进一步 处理。

• NEXT| PRIORI FIRSTI LAST：指定推动游标指针的方式

* NEXT：向前推进一条记录
* PRIOR：向回退一条记录
* FIRST：推向第一条记录
* LAST：推向最后一条记录

>缺省值为NEXT

4)关闭游标

•使用CLOSE语句

•语句格式

EXEC SQL CLOSE V游标名〉；

•功能

关闭游标，释放结果集占用的缓冲区及其他资源。

•说明

>游标被关闭后，就不再和原来的査询结果集相联系。

>被关闭的游标可以再次被打开，与新的查询结果相联系。

(4)建立和关闭数据库连接

•建立数据库连接

EXEC SQL CONNECT TO target [AS connection — name] [USER user— name]；

target是要连接的数据库服务器：

常见的服务器标识串，如Vdbname>@<hostname>:<port>

包含服务器标识的SQL串常量:DEFAULT

•关闭数据库连接

EXEC SQL DISCONNECT [connection]；

程序运行过程中可以修改当前连接：

EXEC SQL SET CONNECTION connection-name | DEFAULT；

■三、不用游标的**SQL**语句

不用游标的SQL语句的种类

1—

O

1

•说明性语句

•数据定义语句

第八章

•数据控制语句

•查询结果为单记录的SELECT语句

数据库编程

•非CURRENT形式的增删改语句

■四、使用游标的**SQL**语句

必须使用游标的SQL语句

•査询结果为多条记录的SELECT语句

* CURRENT形式的UPDATE语句
* CURRENT形式的DELETE语句

(一)查询结果为多条记录的SELECT语句

使用游标的步骤

1. 说明游标
2. 打开游标
3. 推进游标指针并取当前记录
4. 关闭游标

(Zl) CURRENT形式的UPDATE语句和DELETE语句

如果只想修改或删除其中某个记录

•用带游标的SELECT语句查出所有满足条件的记录

•从中进一步找出要修改或删除的记录

•用CURRENT形式的UPDATE语句和DELETE语句修改或删除之

* UPDATE语句和DELETE语句中的子句：

WHERE CURRENT OF <游标名 >

表示修改或删除的是最近一次取出的记录，即游标指针指向的记录

不能使用CURRENT形式的UPDATE语句和DELETE语句：

・当游标定义中的SELECT语句带有UNION或ORDER BY子句

•该SELECT语句相当于定义了一个不可更新的视图

■五、动态**SQL**

02

1—

数据库系统概论(第**5**版)同步辅导及习题全解

静态嵌入式SQL

•静态嵌入式SQL语句能够满足一般要求

•无法满足要到执行时才能够确定要提交的SQL语句

动态嵌入式SQL

•允许在程序运行过程中临时“组装” SQL语句

•支持动态组装SQL语句和动态参数两种形式

1. 使用SQL语句主变量

SQL语句主变量：

程序主变量包含的内容是SQL语句的内容,而不是原来保存数据的输入或输出变量。

SQL语句主变量在程序执行期间可以设定不同的SQL语句，然后立即执行。

1. 动态参数

动态参数:SQL语句中的可变元素。

使用参数符号表示该位置的数据在运行时设定。

和主变量的区别：

动态参数的输入不是编译时完成绑定，而是通过准备(prepare)语句和执行(execute)时绑定数 据或主变量来完成。

使用动态参数的步骤：

1. 声明SQL语句主变量。
2. 准备 SQL 语句(PREPARE)。

EXEC SQL PREPARE V语句名〉FROM <SQL 语句主变量〉；

1. 执行准备好的语句(EXECUTE)

EXEC SQL EXECUTE V语句名> [INTO V主变量表〉][USING < 主变量或常量>]；

■六、**PL/SQL**的块结构

PL/SQL :

SQL的扩展

增加了过程化语句功能，基本结构是块块之间可以互相嵌套,每个块完成一个逻辑操作。 PL/SQL块的基本结构： 、

1. 定义部分

DECLARE

第八章

变量、常量、游标、异常等

•定义的变量、常量等只能在该基本块中使用

数据库编程

•当基本块执行结束时，定义就不再存在

1. 执行部分

BEGIN

SQL语句、PL/SQL的流程控制语句

EXCEPTION

异常处理部分

END；

■七、变量常量的定义

1. PL/SQL中定义变量的语法形式是：

变量名数据类型［［NOT NULL］：=初值表达式］或

变量名数据类型［［NOT NULL］初值表达式］

1. 常量的定义类似于变量的定义：

常量名数据类型CONSTANT ：=常量表达式

常量必须要给一个值，并旦该值在存在期间或常量的作用域内不能改变。如果试图修改它, PL/SQL将返回一个异常。

1. 赋值语句

变量名称:=表达式

■八、**PL/SQL**功能

（一）条件控制语句

IF- THEN, IF- THEN - ELSE 和嵌套的 IF 语句

1. IF condition THEN

Sequence\_of\_statements ；

**04**数据库系统概论(第**5**版)同步辅导及习题全解

END IF

1. IF condition THEN

Sequence\_of\_statementsl;

ELSE

Sequence\_of\_statements2 ；

END IF；

1. 在THEN和ELSE子句中还可以再包括IF语句，即IF语句可以嵌套。
2. 循环控制语句

LOOP, WHILE-LOOP 和 FOR—LOOP

1. 最简单的循环语句LOOP

LOOP

Sequence\_of\_statements ；

END LOOP；

多数数据库服务器的PL/SQL功能都提供EXIT.BREAK或LEAVE等循环结束语句，保证 LOOP语句块能够结束。

1. WHILE-LOOP

WHILE condition LCXDP

Sequence\_of\_statements ；

END L(X)P；

每次执行循环体语句之前，首先对条件进行求值：

如果条件为真，则执行循环体内的语句序列；

如果条件为假，则跳过循环并把控制传递给下一个语句。

1. FOR-LOOP

FOR count IN [REVERSE]bound 1 ••• bound2 L(X)P

Sequence\_of\_statements ；

END LOOP；

1. 错误处理

如果PL/SQL语句在执行时出现异常,则应该让程序在产生异常的语句处停下来，根据异常的

类型去执行异常处理语句。

SQL标准对数据库服务器提供什么样的异常处理给出了建议，要求PL/SQL管理器提供完善 的异常处理机制。 、

05

1—

■九、存储过程

第八章

PL/SQL块类型：

数据库编程

命名块:编译后保存在数据库中，可以被反复调用，运行速度较快。存储过程和函数是命名块。

匿名块:每次执行时都要进行编译，它不能被存储到数据库中，也不能在其他的PL/SQL块中 被调用。

存储过程：由PL/SQL语句书写的过程，经编译和优化后存储在数据库服务器中，使用时只要 调用即可。

（一）存储过程的优点：

1. 运行效率高
2. 降低了客户机和服务器之间的通信量
3. 方便实施企业规则
4. 创建存储过程：

CREATE Procedure 过程名（［参数 1,参数 2,...］）AS

<PL/SQ L 块〉；

过程名：数据库服务器合法的对象标识

参数列表:用名字来标识调用时给出的参数值，必须指定值的数据类型。参数也可以定义输入 参数、输出参数或输入/输岀参数。默认为输入参数。

过程体:是一个<PL/SQL块〉。包括声明部分和可执行语句部分。

1. 重命名存储过程

ALTER Procedure 过程名 1 RENAME TO 过程名 2；

1. 执行存储过程：

CALL/PERFORM Procedure 过程名（［参数 1,参数 2,...］）；

•使用CALL或者PERFORM等方式激活存储过程的执行。

1. 删除存储过程

DROP PROCEDURE 过程名（）；

游标的使用：

**06**数据库系统概论(第**5**版)同步辅导及习题全解

・在PL/SQL中，如果SELECT语句只返回一条记录，可以将该结果存放到变量中。

•当查询返回多条记录时,就要使用游标对结果集进行处理。

* 一个游标与一个SQL语句相关联。
* PL/SQL中的游标由PL/SQL引擎管理。

■十、**ODBC**编程

ODBC优点：

•移植性好

•能同时访问不同的数据库

•共享多个数据资源

■十一、数据库互连概述

ODBC产生的原因：

・由于不同的数据库管理系统的存在，在某个RDBMS下编写的应用程序就不能在另一个

RDBMS下运行。

•许多应用程序需要共享多个部门的数据资源，访问不同的RDBMSO

■十二、**ODBC**工作原理概述

ODBC应用系统的体系结构：

（一） 用户应用程序

（二） 驱动程序管理器

（三） 数据库驱动程序

（四） ODBC数据源管理

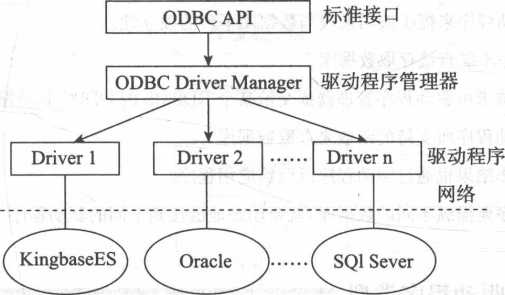
**07**

1—

第八章

数据库编程

用户应用程序



ODBC应用程序包括的内容:

•请求连接数据库;

-向数据源发送SQL语句;

•为SQL语句执行结果分配存储空间，定义所读取的数据格式;

•获取数据库操作结果或处理错误;

•进行数据处理并向用户提交处理结果;

•请求事务的提交和回滚操作;

•断开与数据源的连接。

驱动程序管理器:用来管理各种驱动程序。

・包含在ODBC32. DLL中

•管理应用程序和驱动程序之间的通信 •建立、配置或删除数据源并查看系统当前所安装的数据库ODBC驱动程序 •主要功能:

＞装载ODBC驱动程序

＞选择和连接正确的驱动程序

＞管理数据源

＞检查ODBC调用参数的合法性

＞记录ODBC函数的调用等

**08**数据库系统概论(第**5**版)同步辅导及习题全解

.十三、数据库驱动程序

ODBC通过驱动程序来提供应用系统与数据库平台的独立性。

ODBC应用程序不能直接存取数据库。

•其各种操作请求由驱动程序管理器提交给某个RDBMS的ODBC驱动程序。

•通过调用驱动程序所支持的函数来存取数据库。

•数据库的操作结果也通过驱动程序返回给应用程序。

•如果应用程序要操纵不同的数据库，就要动态地链接到不同的驱动程序上。

■十四、**ODBC**驱动程序类型

单束

•数据源和应用程序在同一台机器上

•驱动程序直接完成对数据文件的I/O操作

•驱动程序相当于数据管理器

多束

•支持客户机/服务器、客户机/应用服务器/数据库服务器等网络环境下的数据访问

•由驱动程序完成数据库访问请求的提交和结果集接收

•应用程序使用驱动程序提供的结果集管理接口操纵执行后的结果数据

ODBC数据源管理

•数据源:是最终用户需要访问的数据，包含了数据库位置和数据库类型等信息，是一种数据 连接的抽象

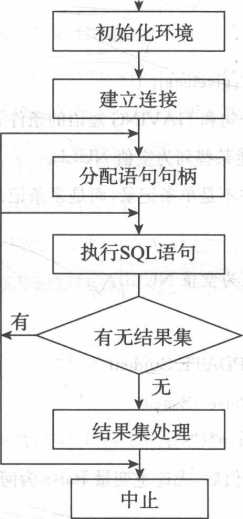
•数据源对最终用户是透明的

＞ ODBC给每个被访问的数据源指定唯一的数据源名(Data Source Name,简称DSN),并映 射到所有必要的、用来存取数据的低层软件

＞在连接中，用数据源名来代表用户名、服务器名、所连接的数据库名等

＞最终用户无需知道DBMS或其他数据管理软件、网络以及有关ODBC驱动程序的细节

■十五、**ODBC**的工作流程

配置数据源

09

1—

动态配置数据源

第八章

数据库编程

■十六、典型的**ODBC**会话

1. 完成ODBC环境句柄的申请工作，建立ODBC环境，为ODBC函数调用建立工作空间；
2. 应用程序申请连接句柄，加载驱动程序并建立到数据源的连接，用户提供数据源名，ODBC API根据数据源名建立与数据库的通信；
3. 应用程序建立语句句柄，通过它执行ODBC命令；
4. 处理执行语句句柄中包含的SQL语句，完成对数据库的查询和更新,这是应用程序的核心 内容；
5. 完成对数据库的操作后,释放语句句柄,然后释放数据库连接，最后释放ODBC环境,关闭应 用程序。

典型例题

例**1** 根据学生号码查询学生信息。假设已经把要查询的学生的学号赋给了主变量givensnoo

EXEC SQL SELECT Sno, Sname, Ssex» Sage, Sdept INTO： Hsno»: Hname,: Hsex,：

解析

数据库系统概论(第**5**版)同步辅导及习题全解

Hage,: Hdept

FR( )M Student

WHERE Sno= : givensno ；

(l)INTO子句.WHERE子句和HAVING短语的条件表达式中均可以使用主变量。

(2)查询返回的记录中，可能某些列为空值NULLO

(3)如果查询结果实际上并不是单条记录，而是多条记录,则程序出错,RDBMS会在SQL-

CA中返回错误信息。

例**2**将计算机系全体学生年龄置为空值NULLO

解析

Sageid= —1 ；

EXEC SQL UPDATE Student

SET Sage= : Raise : Sageid

WHERE Sdept='CS'；

将指示变量Sageid赋一个负值后，无论主变量Raise为何值,RDBMS都会将CS系所有学生

的年龄置空值。

等价于:

EXEC SQL UPDATE Student

SET Sage=NULL

WHERE Sdeptm'CS'；

例**3** 某个学生退学了，现要将有关他的所有选课记录删除掉。假设该学生的姓名已赋给主变量

stdname。

解析

EXEC SQL DELETE

FROM SC

WHERE Sno=

(SELECT Sno

FROM Student

WHERE Sname= : stdname)；

例**4** 某个学生新选修了某门课程，将有关记录插入SC表中。假设插入的学号已赋给主变量 stdno，课程号已赋给主变量couno0

1

1—

11

解析 gradeid=-l; / \*用作指示变量，赋为负值\* /

EXEC SQL INSERT

第八章

INTO SC( Sno, Cno, Grade)

数据库编程

VALUES(: stdno,: couno,: gr,: gradeid)；

由于该学生刚选修课程，成绩应为空，所以要把指示变量赋为负值。

例**5**向TEST中插入元组。

解析 EXEC SQL BEGIN DECLARE SECTION；

const char \* stmt = ''INSERT INTO test VALUES(?) ；"；/\* 声明 SQL 主变量 \* / EXEC SQL END DECLARE SECTION；

EXEC SQL PREPARE mystmt FROM : stmt ； / \* 准备语句 \* /

EXEC SQL EXECUTE mystmt USING 100； / \* 执行语句 \* /

EXEC SQL EXECUTE mystmt USING 200； / \* 执行语句 \* /

例**6**利用存储过程来实现下面的应用：从一个账户转指定数额的款项到另一个账户中。

CREATE PRCX3EDURE TRANSFER( inAccount INT, outAccount INT, amount FLOAT)

AS DECLARE

totalDeposit FLOAT；

BEGIN /\*检査转出账户的余额\*/

SELECT total INTO totalDeposit

FROM ACCOUNT WHERE

ACCOUNTNUM=outAccount ；

IF totalDeposit IS NULL THEN / \*账户不存在或账户中没有存款\* /

ROLLBACK；

RETURN：

END IF；

IF totalDeposit. total V amount THEN / \* 账户存款不足 \* /

**2** 数据库系统概论(第**5**版)同步辅导及习题全解

ROLLBACK；

RETURN；

END IF；

UPDATE account SET total=total—amount

WHERE ACCOUNTNUM=out Account ；

/\*修改转出账户，减去转出额\*/

UPDATE account SET total=total + amount WHERE ACCOUNTNUM = in Account ；

/\*修改转入账户，增加转出额\*/

COMMIT； /\*提交转账事务\*/

END；

1.匯厘陸' ①查询某一门课程的信息。要查询的课程由用户在程序运行过程中指定，放在主变 量中。

课后习题解答

EXEC SQL BEGIN DECLARE SECTION：/\* 说明主变量 deptname, HSno, HSname ；

\* HSsex,HSage,NEWAGEO \*/

char deptname] 20]；

char HSno[9]；

char HSname[20]；

char HSsex[2]；

int HSage；

int NEWAGE；

EXEC SQL END DECLARE SECTION； / \* 声明部分结束 \* /

long SQLCODE；

EXEC SQL INCLUDE sqlca； / \* 定义 SQL 通信区 \* /

int main(void)

{int count=0；

char yn；/ \*变量yn代表yes或者no\* /

printf(uPlease choose the department name(CS/MA/IS) ：,J) ； / \* 为主变量 dept name 赋值 \* /

scanf ( " % s," &deptname)；

EXEC SQL CONNECT TO 学生课程数据库@localhost：54321 USER “SYSTEM”/“MANAGER” ； / \* 连续数据库 TEST \* /

第八章

数据库编程

EXEC SQL DECLARE SX CURSOR FOR / \*说明游标*，*为査询和更新做准备\* / SELECT Sno,Sname, Ssex, Sage

FROM Student

WHERE SDept= :deptname；

EXEC SQL OPEN SX； / \* 打开游标 \* /

for(； ；.) /\*用循环结构逐条处理结果集的记录\* /

EXEC SQL FETCH SX INTO： HSno,: HSname,: HSsex,: HSage； / \* 游标向前 推进一行，取结果送主变量\*/

if (sqlca,sqlcode ! =0)

break； / \*其中参数0代表SUCCESS \* /

if(count+ +=0) / \*如果是第一行的话，先打出行头\* /

printf("\n% — 10s % — 20s % — 10s % — 10s\n"，“Sno",“Sname",“Ssex”,“Sage")； printf("% — 10s % — 20s % — 10s % —10d\n",HSno,HSname,HSsex,HSage)； printf(“UPDATE AGE(y/n)?,,＞；

do

{scanf("%c", &yn)；

}while(yn! =，N，&& yn! =，n，&&yn! =4Y? && yn! =，y，)；

if(yn='y，I I yn=' Y，)/ \*如果选择修改的操作\* /

(printf("INPUT NEW AGE：")；

scanf("％d”,&NEWAGE)；/ \*用户输入新年龄,存放在主变量\* NEWAGE中\* /

EXEC SQL UPDATE Student

SET Sage=： NEWAGE

WHERE CURRENT OF SX； / \*对Student表中的年龄进行更新\* /

}

4

1—

数据库系统概论(第**5**版)同步辅导及习题全解

j EPl过程

EXEC SQL CLOSE SX； / \* 关闭游标 \* /

EXEC SQL COMMIT WORK ； / \* 提交更新 \* /

EXEC SQL DISCONNECT学生课程数据库；/ \*断开连接\* /

②査询选修某一门课程的选课信息，要查询的课程号由用户在程序运行过程中指

定,放在主变量中，然后根据用户的要求修改其中某些记录的成绩字段。

k （一）统计离散数学的成绩分布情况，即按照各分数段统计人数。

1.创建存储过程

1）创建需要的表结构。

因为执行存储过程后在客户端并没有返回值，因此需要建立一个表存放执行后的结

果，并返回到客户端显示。根据实验要求，我们要统计选修了离散数学的学生的成

绩分布，因此我们建立表Rank,其中第一列division显示成绩分段划分，第二列

number显示的是成绩在该分数段的学生人数。

CREATE TABLE Rank(

division CHAR(20),

number INT)；

2）编写存储过程。

CREATE OR REPLACE R()CEDURE statistic mark (name CHAR(50))

〃存储过程带有一个字符型参数值，便于统计不同科目的分数分布情况

AS

DECLARE

less60

INT：=0； 〃分为五个分数段,并置初始值为0

b6Oa7O INT： =0；

b70a80 INT： = 0；

b80a90 INT： =0；

more90 INT: =0；

curcno CHAR(4)；

//设字符型变量curcno存放输入的课程名称参数

BEGIN

SELECT eno INTO curcno

〃根据课程名称在Course表中查询该课程号

FROM Course

WHERE cname=name；

IF NOT FOUND THEN //如果不存在用户输入的课程名称,返回错误信息

RAISE EXCEPTION，输入错误,没有该课程'；

END IF；

SELECT count( \* ) INTO less60 〃查询分类低于60分的学生人数

5

1—

1—

FROM sc

第八章

WHERE cno=curcno AND grade<60；

SELECT count( \* ) INTO b60a70 //查询分数在60到70之间的学生人数

数据库编程

FROM sc

WHERE cno=curcno AND grade〉= 60 AND grade<C70；

SELECT count( \* ) INTO b70a80 〃查询分数在70到80之间的学生人数

FROM sc

WHERE cno= curcno AND grade〉= 70 AND graded80 ；

SELECT count( \* ) INTO b80a90 //查询分数在80到90之间的学生人数

FROM sc

WHERE eno=curcno AND grade〉= 80 AND grade<C90；

SELECT count( \* ) INTO more90 〃查询分数高于90的学生人数

FROM sc

WHERE eno = curcno AND grade〉=90； 〃向表 Rank 中插入五条记录，显示

成绩分布情况

INSERT INTO Rank VALUES(' [0,60)' ； less60)；

INSERT INTO Rank VALUES(' [60,70)' ； b60a70)；

INSERT INTO Rank VALUES( '[70,80)'； b70a80)；

INSERT INTO Rank VALUES( '[80,90)'； b80a90)；

INSERT INTO Rank VALUES('[90,100)' ；more90)；

首先执行编写好的存储过程statistic mark,然后在表Rank中查看执行结果。

PERFORM PROCEDURE statistic mark(4 离散数学，);

(二)统计任意一门课的平均成绩。

1.创建存储过程

1)创建需要的表结构。

根据试验要求，我们要统计任意一门课的平均成绩，因此我们建立表Avggrade,其 中第一列cname显示统计的课程名称，第二列avgg显示选修了该课程的学生平均

成绩。

数据库系统概论(第**5**版)同步辅导及习题全解

CREATE TABLE Avggrade(

cname CHAR(50),

avgg NUMERIC] 10,6))；

CREATE OR REPLACE PROCEDURE collect avggradeO

AS

DECLARE 〃声明变量

curname CHAR(50)；

curcno CHAR(4)；

curavgg NUMERIC(10,6)；

CURSOR mycursor FOR 〃声明游标mycursor査询课程号和名称

SELECT eno,cname FROM course；

BEGIN

OPEN mycursor； //打开游标

IF mycursor%ISOPEN THEN 〃条件控制，游标打开时进行以下处理

LOOP 〃循环控制

FETCH mycursor INTO curcno,curname； //游标推进一行，取结果送变量

EXIT WHEN (mycursor%NOTFOUND) ； //如果没有返回值则退出循环

SELECT AVG(grade) INTO curavgg FROM SC //求该课程的平均值送变量

WHERE cno= curcnp； //向Avggrade表中插入记录，显示课程名称和平均 成绩

INSERT INTO Avggrade VALUES( curname, curavgg)；

END L(X)P； //结束循环控制

END IF； //结束条件控制

CLOSE mycursor ； 〃关闭游标

END；

2.执行存储过程

首先执行编写好的存储过程collect avgg整de,然后在表Avggrade中查看执行结果。

PERFORM PROCEDURE collect avggrade():

SELECT\* FORM Avggrade；

(三)将学生选课成绩从百分制改为等级制(即A、B、C、D、E)。

1. 创建存储过程

根据试验要求，在本试验中存储过程的执行不需要在客户端返回结果，因此我们不 需要建立相应的表结构来存放存储过程的执行结果。直接编写存储过程即可。

CREATE OR REPLACE PRCX3EDURE change criticalO

AS

第八章

DECLARE

chgrade CHAR(l)； 〃声明标量

数据库编程

currecord RECORD；

BEGIN //由于之前的成绩值为整数型，而修改后的结果为字符型，因此在SC

中先增加一列newgrade,存放换算后的成绩〃

ALTER TABLE SC ADD COLUMN(newgrade CHAR(l))；

FOR currecord IN SELECT \* FROM SC LOOP //FOR 循环处理每个学生的

成绩

IF currecord gradeV60 THEN

chgrade=4E, ； //如果成绩低于60分，将成绩置为E

ELSIF currecord. gradeV70 THEN

chgrade=4D,； 〃如果成绩＞ = 60&V70,将成绩置为D

ELSIF currecord. grade＜C80 THEN

chgrade=tC,； 〃如果成绩＞ = 70&V80,将成绩置为C

ELSIF currecord. grade＜C90 THEN

chgrade=tB,； 〃如果成绩＞ = 80&V90,将成绩置为B

ELSIF currecord,gradeV90 THEN

chgrade='B，； 〃如果成绩＞ = 80&V90,将成绩置为B

ELSE

chgrade=，A，； //如果成绩高于或等于90分，将成绩置为A

END IF； 〃结束条件控制

UPDATE SC SET newgrade = chgrade //修改 newgrade 列值，填入换算后的

成绩

WHERE sno=currecord sno AND eno=currecord. eno；

END LOOP；

ALTER TABLE SC DROP COLUMN grade； //删除原有的列 grade

ALTER TABLE SC RENAME newgrade TO grade； //将 newgrade 列名改为 grade

END；

1. 执行存储过程

PERFORM PROCEDURE change criticalO；

配置两个不同的数据源，编写程序连接两个不同RDBMS的数据源,对异构数据库 进行操作。例如，将KingbaseES数据库的某个表中的数据转移到SQL Server数据 库的表中。

数据库系统概论(第**5**版)同步辅导及习题全解

源代码：

* include Vstdlib. h>

井 include Vstdio. h>

* include〈windows. h>
* include <sql. h>
* include Vsqlext. h>
* include Vsqltypes. h>
* define SNO\_LEN 30
* define NAME\_LEN 50
* define DEPART\_LEN 100
* define SSEX\_LEN 5

int main()

/\* Step 1：定义句柄和变量\*/

//以king开头的表示的是连接Kingbaseeg的变量 //以server开头的表示的是连接SQL server的变量 SQLHENV

kinghenv, serverhenv；

〃环境句柄

SQLHDBC

kinghdbc*，*serverhdbc ；

//连接句柄

SQLHSTMT

kinghstmt ・ serverhstmt ；

〃语句句柄

SQLRETURN

ret；

〃结果返回集

SQLCHAR

第八章

sNameLNAME\_LEN], sDepart[DEPART\_LEN], sSex[SSEX\_LEN], sSno[SN()\_LEN]；

.数据库编程

SQLINTEGER

sAge；

SQLINTEGER

cbAge=O, cbSno=SQL\_NTS, cbSex= SQL\_NTS,

cbName= SQL\_NTS, cbDepart=SQL\_NTS；

/\* Step 2：初始化环境\*/

//分配环境句柄

ret=SQLAllocHandle(SQL\_HANDLE\_ENV, SQL\_NULL\_HANDLE, &kinghenv)；

ret=SQLAllocHandle(SQL\_HANDLE\_ENV, SQL\_NULL\_HANDLE,

&serverhenv);

//设置管理环境的属性

ret=SQLSetEnvAttr (kinghenv, SQL\_ ATTR\_ ODBC\_ VERSION, (void \* ) SQL\_

OV\_()DBC3, 0)；

ret=SQI^etEnvAttrCserverhenv, SQL\_ATTR\_ODBC\_VERSION, (void \* )SQL\_

()V\_()DBC3, 0)；

/ \* Step 3 :建立连接\* /

〃分配连接句柄

ret=SQLAllocHandle(SQL\_HANDLE\_DBC, kinghenv, &kinghdbc)；

ret=SQLAllocHandle(SQL\_HANDLE\_DBC, serverhenv, &serverhdbc);

ret = SQLConnect (kinghdbc,

〃连接 KingbaseES

"KingbaseES ODBC" ,SQL\_NTS,

"SYSTEM",SQL NTS,

” MANAGER”，SQL\_NTS)；

**20**数据库系统概论(第**5**版)同步辅导及习题全解

if (! SQL\_SUCCEEDED(ret))

〃连接失败时返回错误值

return —1；

ret=SQLConnect (serverhdbc,

〃连接 SQI^erver

"SQLServer", SQL\_NTS,

”sa”,SQL\_NTS,

” sa”，SQL\_NTS)；

if(! SQL\_SUCCEEDED( ret))

〃连接失败时返回错误值

return —1 ；

/ \* Step 4 :初始化语句句柄\* /

ret=SQLAllocHandle(SQL\_HANDLE\_STMT, kinghdbc, &kinghstmt)；

ret=SQI^etStmtAttr(kinghstmt,SQL\_ATTR\_R()W\_BIND\_TYPE, (SQLPOINT-

ER)SQL\_BI

ND\_BY\_COLUMN, SQL\_IS\_INTEGER)；

//设置语句选项

ret=SQLAllocHandle(SQL\_HANDLE\_STMT, serverhdbc, &serverhstmt)；

/ \* Step 5 :两种方式执行语句\* /

/\*预编译带有参数的语句\*/

〃需要多次执行插入，因此预先声明插入语句

ret=SQLPrepare( serverhstmt, " INSERT INTO STUDENT

(SNO,SNAME,SSEX,SAGE,SDEPT) VALUES (?, ?, ?, ?, ?)\*',

SQL\_NTS)；

if (ret==SQL\_SUCCESS | | ret==SQL\_SUCCESS\_WITH\_INFO)

(

//绑定参数 、

ret= SQLBindParameter (serverhstmt, 1, SQL\_PARAM\_INPUT, SQL\_C\_CHAR,

SQL\_CHAR, SN()\_LEN, 0,sSno,0,&cbSno)；

ret= SQLBindParameter (serverhstmt, 2, SQL\_ PARAM\_ INPUT, SQL\_C\_CHAR,

SQL\_CHAR, NAME\_LEN, 0, sName, 0, &cbName)；

ret= SQLBindParameter(serverhstmt, 3, SQL\_PARAM\_INPUT, SQL\_C\_CHAR, SQL\_CHAR, 2, 0, sSex, 0, &cbSex)；

ret= SQLBindParameter(serverhstmt, 4, SQL\_PARAM\_INPUT, SQL\_C\_LONG, SQL\_INTEGER, 0, 0, &sAge, 0, &cbAge)；

第八章

ret= SQLBindParameter (serverhstmt, 5, SQL\_ PAR AM INPUT, SQL\_C\_CHAR, SQL\_CHAR, DEPART\_LEN, 0, sDepart, 0, &cbDepart)；

数据库编程

}

/\*执行SQL语句美/

ret= SQLExecDirect( kinghstmt,nSELECT \* FROM STUDENT" ,SQL\_NTS)；

if (ret= = SQL\_SUCCESS| |ret= = SQL\_SUCCESS\_WITH\_INFO)

*{ \**

//将结果集中的属性列一一绑定至变量

ret= SQLBindCoKkinghstmt, 1, SQL\_C\_CHAR, sSno,SNO\_LEN, &cbSno)；

ret= SQLBindCoK kinghstmt, 2, SQL\_C\_CHAR, sName, NAME LEN, &cbName)；

ret= SQLBindCoKkinghstmt,3, SQL\_C\_CHAR,sSex,sSEX\_LEN, &cbSex)；

ret= SQLBindCoKkinghstmt,4, SQL\_C\_LONG,&sAge, 0, &cbAge)；

ret= SQLBindCoK kinghstmt, 5, SQL\_C\_CHAR, sDepart, DEPART LEN, &cbDepart)；

)

/\* Step 6 :处理结果集并执行预编译后的语句\* /

while ((ret=SQLFetch( kinghstmt))! = SQL\_NO\_DATA\_F()UND)

{

if(ret= = SQL\_ERR()R)

//错误处理

printfC Fetch error\n")；

else ret=SQLExecute( serverhstmt)；

〃执行语句

}

/\* Step 7：终止处理\* /

SQLFreeHandle( SQL\_H ANDLE\_STMT, kinghstmt)；

**22**数据库系统概论(第**5**版)同步辅导及习题全解

//释放Kingbase语句句柄

SQLDisconnect( kinghdbc)；

//断开与KingbaseES的连接

SQLFreeHandle( SQL\_HANDLE\_DBC, kinghdbc)；

//释放Kingbase连接句柄

SQLFreeHandle( SQL\_HANDLE\_ENV, kinghenv)；

//释放Kingbase环境句柄

SQLFreeHandle(SQL\_HANDLE\_STMT, serverhstmt)；

〃释放SQI^erver语句句柄

SQLDisconnect( serverhdbc)；

//断开与SQLServer的连接

SQLFreeHandle(SQL\_HANDLE\_DBC, serverhdbc)；

〃释放SQLServer连接句柄

SQLFreeHandleC SQL\_HANDLE\_ENV, serverhenv)；

//释放SQLServer环境句柄

return 0；

)

编译并且执行。执行完成后，进入SQL Server查询分析器,对原来为空的Student 表进行查询(SELECT \* FROM Student),可以看见SQL Server中Student表的数 据即为KingbaseES中Student表的数据备份。

3

2

1—

第九章

关系查询处理和查询优化

知识网络图

查询分析

关系数据库系统的查询处理＜

查询处理步骤'

查询检查

查询优化

第九章

关系查询处理和査询优化

查询执行

关系代数表达式等价变换规则

代数优化〈

［选择运算应尽可能先做

关系查询处理

和查询优化

查询树的启发式〈

投影和选择运算同时进行

投影同其前或后的双目运算结合

某些选择同笛卡尔积结合成连接运算

找出公共子表达式

基于启发式规则的存取路径选择优化

1. 选择操作的启发式规则
2. 连接操作的启发式规则

物理优化〈

统计信息

基于代价估算的优化

代价估算公式＜

全表扫描算法

索引扫描算法 嵌套循环连接算法

排序一合并连接算法

**24**数据库系统概论(第**5**版)同步辅导及习题全解

知识点归纳

重点：了解关系数据库查询优化的重要性。掌握查询处理各个步骤的主要功能。能够把SQL 语句转换成查询树，对查询树进行代数优化，转换成优化的查询树。掌握物理优化的基本方法。

难点:能运用本章学习的査询优化知识,对于比较复杂的查询，尤其是涉及连接和嵌套的查询, 写出适合RDBMS自动优化的SQL语句。对于RDBMS不能优化的查询需要重写查询语句，进行 手工调整以优化性能。不要把优化的任务全部放在RDBMS上。

■ 一、关系查询优化的概述

1. 查询优化在关系数据库中的重要性及必要性

关系系统的查询优化既是RDBMS实现的关键技术又是关系系统的优点所在。它减轻了用户 选择存取路径的负担。查询优化极大地影响RDBMS的性能。用户只要提出“干什么”，不必指出 “怎么干”。査询优化的优点不仅在于用户不必考虑如何最好地表达查询以获得较好的效率，而且 在于系统可以比用户程序的“优化”做得更好。

1. 查询优化的可能性和优点

（1） 优化器可以从数据字典中获取许多统计信息,而用户程序则难以获得这些信息。

（2） 如果数据库的物理统计信息改变了，系统可以自动对查询重新优化以选择相适应的执行计 划。在非关系系统中必须重写程序，而重写程序在实际应用中往往是不太可能的。

（3） 优化器可以考虑数百种不同的执行计划,程序员一般只能考虑有限的几种可能性。

（4） 优化器中包括了很多复杂的优化技术，这些优化技术往往只有最好的程序员才能全部掌 握。系统的自动优化相当于使得所有人都拥有这些优化技术。

1. 查询优化的一般准则

（1） 选择运算应尽可能先做。 、

（2） 把投影运算和选择运算同时进行。

（3） 把投影运算同其前或其后的双目运算结合起来执行。

（4） 把某些选择同在它前面要执行的笛卡儿积结合起来成为一个连接运算。

（5） 找出公共子表达式。

（6） 选取合适的连接算法。

1. 查询优化的一般步骤

5

2

1—

（1） 把查询转换成某种内部表示，通常用的内部表示是语法树。

（2） 把语法树转换成标准（优化）形式。即利用优化算法，把原始的语法树转换成优化的形式。

第九章

（3） 选择低层的存取路径。

（4） 生成查询计划，选择代价最小的。

关系查询处理和査询优化

1. 代价模型

一般DBMS采用基于代价的优化算法：

集中式数据库

单用户系统

总代价=I/O代价+ CPU代价

多用户系统

总代价=I/O代价+ CPU代价+内存代价

分布式数据库

总代价=I/O代价+ CPU代价［+内存代价］+通信代价

二、关系数据库查询优化方法

1. 代数优化

关系代数表达式等价指用相同的关系代替两个表达式中相应的关系所得到的结果是相同的。

（1）查询树启发式优化,一般规则有：

•选择运算应尽可能先做（最重要，最根本）

目的:减小中间关系

•投影运算和选择运算同时做

目的:避免重复扫描关系

•将投影运算与其前面或后面的双目运算结合

目的:减少扫描关系的遍数

•在执行连接操作前对关系适当进行预处理

•按连接属性排序

•在连接属性上建立索引

•某些选择运算同在其前面执行的笛卡尔积结合成连接运算

**26**数据库系统概论(第**5**版)同步辅导及习题全解

（2）查询树的启发式优化算法

1） 分解选择运算

2） 通过交换选择运算,将其尽可能移到叶端

3） 通过交换投影运算,将其尽可能移到叶端

4） 合并串接的选择和投影运算，以便能同时执行或在一次扫描中完成

5） 对内结点分组

6） 生成程序

1. 物理优化

物理优化就是要选择高效合理的操作算法或存取路径,求得优化的查询计划，达到查询优化的 目标。

（1） 物理优化可以选择的方法

1） 基于规则的启发式优化:大多数情况下都适用。

2） 基于代价估算的优化:优化器估算不同执行策略的代价，并选出具有最小代价的执行计划。

3） 两者结合的优化方法。

（2） 选择操作的启发式规则

对于小关系，使用全表顺序扫描，即使选择列上有索引；

对于大关系，启发式规则有：

对于选择条件是主码=值的查询，查询结果最多是一个元组，可以选择主码索引；

一般的RDBMS会自动建立主码索引；

对于选择条件是非主属性=值的查询，并且选择列上有索引，则要估算查询结果的元组数目

如果比例较小«10%）可以使用索引扫描方法，否则还是使用全表顺序扫描。

（3） 全表扫描算法的代价估算公式.

如果基本表大小为B块,全表扫描算法的代价cost=Bo

如果选择条件是码=值，则平均代价cost=B/2。

（4） 排序一合并连接算法的代价估算公式

如果连接表已经按照连接属性排好序，则cost=Br+Bs4-（Frs\* Nr\* Ns）/Mrso

如果必须对文件排序，则需要在代价函数中加上排序的代价。

对于包含B个块的文件排序的代价大约是（2 \* B） + （2\*B\*log2B）o

典型例题

2

1—

例**1**求选修了 2号课程的学生姓名。

解析用SQL表达:

SELECT Student. Sname

FROM Student, SC

WHERE Student. Sno=SC. Sno AND

SC. Cno'='2‘

•假定学生一课程数据库中有1000个学生记录aoooo个选课记录

•选修2号课程的选课记录为50个

・SQL语句的代数优化示例

⑴把SQL语句转换成查询树，如图9. 1所示。

第九章

关系査询处理和査询优化

结果

Project(Sname)  
select(SC.Cno= '2')

join( Student. Sno=SC. sno)

Student

SC

图9.1

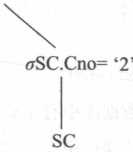
为了使用关系代数表达式的优化法，假设内部表示是关系代数语法树，则上面的查询树可表 示为图9. 2关系代数语法树图。

nSname  
oSC.Cno= ‘2’  
oStudent. Sno=SC. Sno

Student

SC

图9. 2



利用规则4. 6把选择aSC. Cno=<2>移到叶端，上图查询树便转换成图9. 3优化的査询树。

nSname

crStudent. Sno=SC. Sno

数据库系统概论(第**5**版)同步辅导及习题全解

Student

图9.3

例2 对学生一课程数据库系统设有如下数据库模式:

Student (sno, sname, age, det)

Course ( eno, cname, teacher)

SC( sno, eno, grade)

完成查询：找出“程序设计”课成绩在90分以上的学生姓名。

试画出用关系代数表示的语法树，并用关系代数表达式优化算法对原始的语法树进行优化

处理，画出优化后的标准语法树**o**

解析关系代数语法树（见图9. 4）

nSname

<rStudent.Sno=SC.sno A SC.Cno=Course.Cno

*a* Cname='程序设计'A agrade N 90

x

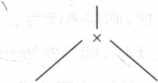
Student

SC

图9. 4

Course

优化后的标准语法树（见图9. 5）



nSname

aSC. Cno=Course.Cno

aStudent.Sno=SC.sno

nCno

Ocnan戶‘程序设计'

9

2

1—

第九章

7iSno,Sname

Curse

nSno,Cno

Student

**N90**

SC

图9.5

关系査询处理和查询优化

课后习题解答

1. 厕园区重要性:关系系统的查询优化既是RDBMS实现的关键技术又是关系系统的优点所 在。它减轻了用户选择存取路径的负担。用户只要提出“干什么”，不必指出“怎么 干”。查询优化的优点不仅在于用户不必考虑如何最好地表达查询以获得较好的效 率，而且在于系统可以比用户程序的“优化”做得更好。

可能性:这是因为：（1）优化器可以从数据字典中获取许多统计信息，例如关系中的 元组数、关系中每个属性值的分布情况、这些属性上是否有索引、是什么索引（B+ 树索引还是HASH索引或唯一索引或组合索引）等。优化器可以根据这些信息选 择有效的执行计划，而用户程序则难以获得这些信息。（2）如果数据库的物理统计 信息改变了，系统可以自动对查询进行重新优化以选择相适应的执行计划。在非关 系系统中必须重写程序，而重写程序在实际应用中往往是不太可能的。（3）优化器 可以考虑数十甚至数百种不同的执行计划，从中选出较优的一个，而程序员一般只 能考虑有限的几种可能性。（4）优化器中包括了很多复杂的优化技术，这些优化技 术往往只有最好的程序员才能全部掌握。系统的自动优化相当于使得所有人都拥 有这些优化技术。

1. ［解题(DR 上没有索引，SELECT \* FROM R；500 块
2. R 中 A 为主码,A 有 3 层 B+树索引 ,SELECT \* FROM R WHERE A=10；3+l

=4块

1. 嵌套循环连接RxS；

因为不知道内存缓冲区的块数K和存放连接结果的块因子Mrs,所以此题无解。

（4）排序合并连接Rx S,区分R和S在B属性上已经有序和无序两种情况。 假设存放连接结果的块因子Mrs=20,即一个块能装20个连接后的元组。如果R 和S在B属性上已经有序，则需要读写20000/40+1200/30+20000/20=1540块。 如果R和S在B属性上无序，则需要加上排序的代价：

30

1—

2 \* 540+（2 \* 540 \* log2 540） = 10882 块。

数据库系统概论(第**5**版)同步辅导及习题全解

**3.**

关系代数表示的语法树（见图9. 6）

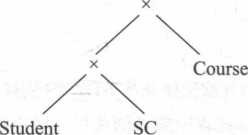
TtSname

aStudent.sdept='IS'

oSC.Cno=Course.Cno

^Student. sno=SC. sno

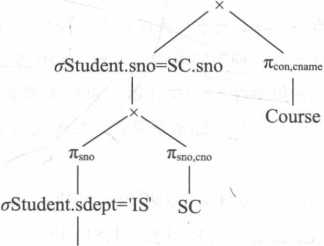
图9. 6



优化后的标准语法树（见9. 7）

TtSname  
crSC.Cno=Course.Cno

Student



(DSELECT \* FROM Teacher WHERE Tsex=4女' 最优策略:全表扫描

1. SELECT \* FROM Department WHERE Dno<301 最优策略:全表扫描
2. SELECT \* FROM Work WHERE Year<>2000 最优策略:全表扫描
3. SELECT \* FROM Work WHERE Year

> 2000 AND Salary < 5000

最优策略:利用Year属性上的B+树索引

(5) SELECT \* FROM Work WHERE Year

知识点播：对于数据库的设计， 数据库的查询优化是必不可少 的；查询处理时RDBMS的核心, 而查询优化技术是查询处理的关 键。一个好的查询优化处理能使 的执行效率更高，减小程序的设 计代价。

1—

3

1—

第九章

< 2000 AND Salary < 5000

最优策略:全表扫描

查询语法树（见图9. 8）

结果

project(Tname)

select(Department.Dname ='计算机系'and Salary>5000)

j oin(Tearcher. T no=Work. T no)

join(Work.Dno=Department.Dno)

Teacher

Department

Work

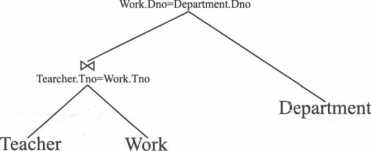
图9. 8 初始的关系代数语法树（见图9. 9）

PlTname

**^Salary>5000**

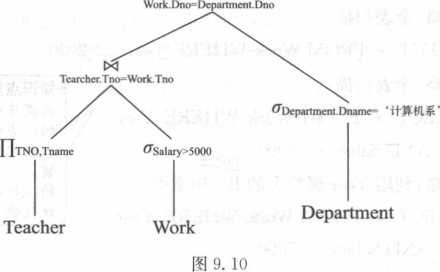
*O* **epartment. Dname= \*** 计算机系，

关系査询处理和查询优化



优化后的关系代数语法树（见图9. 10）





**32**数据库系统概论(第**5**版)同步辅导及习题全解

［小提示：给优化器提供约束和统 弋计信息，查询优化器生成的执行 计划的性能和健壮性很大程度上 依赖于可获得信息的质量。

6.匮题圖区下面的优化策略一般能提高查询效率：（1）选

择运算应尽可能先做；（2）把投影运算和选择

运算同时进行;（3）把投影运算同其前或其后的 双目运算结合起来执行；（4）把某些选择运算同在它前面要执行的笛卡儿积结合起 来成为一个连接运算；（5）找出公共子表达式；（6）选取合适的连接算法。

7.陸感亀 各个关系系统的优化方法不尽相同，大致的步骤可以归纳如下：（1）把查询转换成

某种内部表示，通常用的内部表示是语法树；（2）把语法树转换成标准（优化）形式。 即利用优化算法，把原始的语法树转换成优化的形式3 （3）选择低层的存取路径;

（4）生成查询计划，选择代价最小的。

第十章

数据库恢复技术

3

3

1—

第十章

数据库恢复技术

知识网络图

事务

事务的基本概念＜

I事务的ACID特性＜

一致性

隔离性

持续性

事务内部的故障

故障的种类＜

系统故障

介质故障

〔计算机病毒

数据库恢复技术〈

动态海量转储

数据转储〈

静态海量转储

动态增量转储

恢复的实现技术〈

静态增量转储

日志文件的格式和内容

登记日志文件•

日志文件的作用

登记日志文件

'恢复策略＜

事务故障的恢复

系统故障的恢复

介质故障的恢复

34

1—

数据库系统概论(第**5**版)同步辅导及习题全解

知识点归纳

重难点

重点:牢固掌握事务的基本概念和事务的ACID性质。要掌握数据库故障恢复的策略和方法。 数据库恢复的基本原理是数据备份，它貌似简单，实际却很复杂。数据库的事务管理策略（不仅有 数据库恢复策略,还有并发控制策略）和DBMS缓冲区管理策略、事务一致性级别密切相关，同学们 要在学习完这些知识后把这些问题联系起来，提升对这些技术的理解和掌握。

难点：掌握日志文件的使用，系统故障、介质故障的恢复方法。对于刚刚学习数据库的学生来 讲并不能体会数据库故障恢复的复杂性和重要性。在实际工作中，则必须正确了解所用的DBMS 产品提供的恢复技术和恢复方法，并且能够根据这些机制正确制定系统的恢复策略，以保证数据库 系统7 \* 24小时正确运行。保证数据库系统在遇到故障时能及时恢复正常运行，提高抗故障抗灾 难的能力。

. 一、什么是事务

1. 事务的基本概念

所谓事务是用户定义的一个操作序列，这些操作要么全做要么全不做，是一个不可分割的工作 单位。

“一荣俱荣，一损俱损”这句话很能体现事务的思想,很多复杂的事务要分步进行，但它们组成 -个整体,要么整体生效，要么整体失效。这种思想反映到数据库上，就是多个SQL语句，要么所有 执行成功,要么所有执行失败。

事务通常是以BEGIN TRANSACTION开始？以COMMIT或ROLLBACK结束。

COMMIT表示提交，即提交事务的所有操作。具体地说就是将事务中所有对数据库的更新写 回到磁盘上的物理数据库中去，事务正常结束。ROLLBACK表示回滚，即在事务运行的过程中发 生了某种故障，事务不能继续执行，系统将事务中对数据啓的所有已完成的操作全部撤消，滚回到 事务开始时的状态。这里的操作指对数据库的更新操作。

1. 事务的四个特性：简称ACID

1）原子性（Atomiciy）：事务是一个独立的逻辑单位，事务中包括的诸操作要么全做，要么全

不做。

如果把一个事务看作是一个程序,它要么完整的被执行，要么完全不执行。就是说事务的操纵 序列或者完全应用到数据库或者完全不影啊数据库。这种特性称为原子性。

假如用户在一个事务内完成了对数据库的更新,这时所有的更新对外部世界必须是可见的，或 者完全没有更新。前者称事务已提交，后者称事务撤消(或流产)。DBMS必须确保由成功提交的 事务完成的所有操纵在数据库内有完全的反映，而失败的事务对数据库完全没有影响。

**5**

**3**

**1—**

第十章

1. 一致性(Consistency)：事务的执行结果必须是使数据库从一个一致性状态变到另一个一致 性状态。

数据库恢复技术

一致性处理数据库中对所有语义约束的保护。假如数据库的状态满足所有的完整性约束，就 说该数据库是一致的。例如，当数据库处于一致性状态S1时，对数据库执行一个事务，在事务执行 期间假定数据库的状态是不一致的，当事务执行结束时，数据库处在一致性状态S2。

1. 隔离性(Isolation)：-个事务的执行不能被其他事务干扰。即并发执行的各个事务之间不能 互相干扰。隔离性是DBMS针对并发事务间的冲突提供的安全保证。DBMS可以通过加锁在并发 执行的事务间提供不同级别的分离。假如并发交叉执行的事务没有任何控制，操纵相同的共享对 象的多个并发事务的执行可能引起异常情况。
2. 持续性(Durability)个事务一旦提交，它对数据库中数据的改变使永久性的。

持久性意味着当系统或介质发生故障时,确保已提交事务的更新不能丢失。即对已提交事务 的更新能恢复。一旦一个事务被提交,DBMS必须保证提供适当的冗余，使其耐得住系统的故障。 所以，持久性主要在于DBMS的恢复性能。

事务是恢复和并发控制的基本单位。保证事务ACID特性是事务处理的重要任务。事务 ACID特性可能遭到破坏的因素有：

1. 多个事务并行运行时,不同事务的操作交叉执行。
2. 事务在运行过程中被强行停止。

在第一种情况下，数据库管理系统必须保证多个事务的交叉运行不影响这些事务的原子性。 在第二种情况下，数据库管理系统必须保证被强行终止的事务对数据库和其他事务没有任何影响。 这些就是数据库管理系统中恢复机制和并发控制机制的责任。

■二、数据库恢复概述

尽管数据库系统中采取了各种保护措施来防止数据库的安全性和完整性被破坏，保证并发事 务的正确执行，但是计算机系统中硬件的故障、软件的错误、操作员的失误以及恶意的破坏仍是不

可避免的，这些故障轻则造成运行事务非正常中断，影响数据库中数据的正确性，重则破坏数据库, 使数据库中全部或部分数据丢失，因此数据库管理系统（恢复子系统）必须具有把数据库从错误状 态恢复到某一已知的正确状态（亦称为一致状态或完整状态）的功能，这就是数据库的恢复。

**36**数据库系统概论(第**5**版)同步辅导及习题全解

■三、故障的种类

1. 事务内部的故障

事务内部的故障有的是可以通过事务程序本身发现的，有的是非预期的、不能由事务程序处 理的。

例如:银行转账事务，这个事务把一笔金额从一个账户甲转给另一个账户乙。

BEGIN TRANSACTION

读账户甲的余额BALANCE；

BALANCE = BALANCE- AMOUNT； （AMOUNT 为转账金额）

IF（BALANCE<0） THEN

｛打印“金额不足,不能转账”；

ROLLBACK"撤消刚才的修改,恢复事务）｝

ELSE

｛读账户乙的余额BALANCE1；

BALANCE1 = BALANCE1+AMOUNT ；

写回 BALANCE1 ；

COMMIT；｝

这个例子所包括的两个更新操作要么全部完成要么全部不做，否则就会使数据库处于不一致 状态，例如只把账户甲的余额减少了而没有把账户乙的余额增加。

在这段程序中若产生账户甲余额不足的情况，应用程序可以发现并让事务滚回，撤消已作的修 改，恢复数据库到正确状态。

事务内部更多的故障是非预期的,是不能由应用程序处理的。如运算溢出、并发事务发生死锁 而被选中撤消该事务、违反了某些完整性限制等。以后,事务故障仅指这类非预期的故障。

事务故障意味着事务没有达到预期的终点（COMMIT或者显式的ROLLBACK）,因此,数据库 可能处于不正确状态。恢复程序要在不影响其他事务运行的情况下，强行回滚（ROLLBACK）该事 务，即撤消该事务已经作出的任何对数据库的修改，使得该事务好象根本没有启动一样。这类恢复 操作称为事务撤消（UNDO）。

1. 系统故障

系统故障是指造成系统停止运转的任何事件，使得系统要重新启动。例如，特定类型的硬件错 误(CPU故障)、操作系统故障、DBMS代码偉误、突然停电等等。这类故障影响正在运行的所有事 务，但不破坏数据库。这时主存内容，尤其是数据库缓冲区(在内存)中的内容都被丢失，所有运行 事务都非正常终止。发生系统故障时，一些尚未完成的事务的结果可能已送入物理数据库,有些已 完成的事务可能有一部分甚至全部留在缓冲区，尚未写回到磁盘上的物理数据库中，从而造成数据 库可能处于不正确的状态。为保证数据一致性,恢复子系统必须在系统重新启动时让所有非正常 终止的事务回滚，强行撤消(UNDO)所有未完成事务。重做(REDO)所有已提交的事务，以将数据 库真正恢复到一致状态。

3

1—

第十章

数据库恢复技术

1. 介质故障

系统故障常称为软故障(Soft Crash),介质故障称为硬故障(Hard Crash)o硬故障指外存故障, 如磁盘损坏、磁头碰撞,瞬时强磁场干扰等。这类故障将破坏数据库或部分数据库,并影响正在存 取这部分数据的所有事务。这类故障比前两类故障发生的可能性小得多，但破坏性最大。

1. 计算机病毒

计算机病毒是具有破坏性、可以自我复制的计算机程序。计算机病毒已成为计算机系统的主 要威胁，自然也是数据库系统的主要威胁。因此数据库一旦被破坏仍要用恢复技术把数据库加以 恢复。

总结各类故障，对数据库的影响有两种可能性。一是数据库本身被破坏。二是数据库没有破 坏，但数据可能不正确，这是因为事务的运行被非正常终止造成的。

恢复的基本原理十分简单。可以用一个词来概括:冗余。这就是说，数据库中任何一部分被破 坏的或不正确的数据可以根据存储在系统别处的冗余数据来重建。尽管恢复的基本原理很简单但 实现技术的细节却相当复杂，下面我们将略去许多细节，介绍数据库恢复的实现技术。

.四、恢复的实现技术

恢复机制涉及的两个关键问题是：

第一,如何建立冗余数据；

第二，如何利用这些冗余数据实施数据库恢复。

建立冗余数据最常用的技术是数据转储和登录日志文件。通常在一个数据库系统中，这两种 方法是一起使用的。

1. 数据转储

所谓转储即DBA定期地将整个数据库复制到磁带或另一个磁盘上保存起来的过程。这些备 用的数据文本称为后备副本或后援副本。

**38**数据库系统概论(第**5**版)同步辅导及习题全解

当数据库遭到破坏后可以将后备副本重新装入，但重装后备副本只能将数据库恢复到转储时 的状态，要想恢复到故障发生时的状态,必须重新运行自转储以后的所有更新事务（见图10. 1）。

转储 运行事务 故障发生点

正确运行 —| 1 i -

**Ta Tb Tf**

重装后备副本 重新运行事务

恢复 > L >  
图10. 1转储和恢复

系统在Ta时刻停止运行事务进行数据库转储，在Tb时刻转储完毕，得到Tb；时刻的数据库一 致性副本。系统运行到匸时刻发生故障。为恢复数据库，首先由DBA重装数据库后备副本，将数 据库恢复至経时刻的状态，然后重新运行自K时刻至Tf时刻的所有更新事务，这样就把数据库 恢复到故障发生前的一致状态。

转储是十分耗费时间和资源的，不能频繁进行。DBA应该根据数据库使用情况确定一个适当 的转储周期。

转储可分为静态转储和动态转储。

静态转储是在系统中无运行事务时进行的转储操作。即转储操作开始的时刻，数据库处于一 致性状态，而转储期间不允许（或不存在）对数据库的任何存取、修改活动。显然，静态转储得到的 一定是一个数据一致性的副本。

静态转储简单，但转储必须等待正运行的用户事务结束才能进行，同样，新的事务必须等待转 储结束才能执行。显然,这会降低数据库的可用性。

动态转储是指转储期间允许对数据库进行存取或修改。即转储和用户事务可以并发执行。

动态转储可克服静态转储的缺点，它不用等待正在运行的用户事务结束,也不会影响新事务的 运行。但是，转储结束时后援副本上的数据并不能保证正确有效。例如，在转储期间的某个时刻 Tc，系统把数据A=100转储到磁带上，而在下一时刻Td,某一事务将A改为200。转储结束后,后 援副本上的A已是过时的数据了。为此，必须把转储期间各事务对数据库的修改活动登记下来，建 立日志文件（Log File）。这样，后援副本加上日志文件就能把数据库恢复到某一时刻的正确状态。

转储还可以分为海量转储和增量转储两种方式。海、■:转储是指每次转储全部数据库。增量转 储则指每次只转储上一次转储后更新过的数据。从恢复角度看，使用海量转储得到的后备副本进 行恢复一般来说会更方便些。但如果数据库很大，事务处理又十分频繁，则增量转储方式更实用更 有效。

数据转储有两种方式，分别可以在两种状态下进行，因此数据转储方法可以分为四类:动态海 量转储、动态增量转储、静态海量转储和静态增量转储。

1. 登记日志文件

(DH志文件的格式和内容

9

3

1—

日志文件是用来记录事务对数据库的更新操作的文件。不同数据库系统采用的日志文件格式 并不完全一样。概括起来日志文件主要有两种格式：以记录为单位的日志文件和以数据块为单位 的日志文件。

第十章

数据库恢复技术

对于以记录为单位的日志文件，其中需要登记的内容包括：

各个事务的开始(BEGIN TRANSACTION)标记

各个事务的结束(COMMIT或ROLLBACK)标记

各个事务的所有更新操作

这里每个事务开始的标记、每个事务的结束标记和每个更新操作均作为日志文件中的一个日 志十己录(log record) 0

1. 日志文件的作用

日志文件在数据库恢复中起着非常重要的作用。可以用来进行事务故障恢复和系统故障恢 复，并协助后备副本进行介质故障恢复。具体地讲:事务故障恢复和系统故障必须用日志文件。

在动态转储方式中必须建立日志文件，后援副本和日志文件综合起来才能有效地恢复数据库。

在静态转储方式中，也可以建立日志文件。当数据库毁坏后可重新装入后援副本把数据库恢 复到转储结束时刻的正确状态，然后利用日志文件，把已完成的事务进行重做处理,对故障发生时 尚未完成的事务进行撤消处理。

1. 登记日志文件(logging)

为保证数据库是可恢复的，登记日志文件时必须遵循两条原则：

1. 登记的次序严格按并发事务执行的时间次序进行。
2. 必须先写日志文件，后写数据库。

把对数据的修改写到数据库中和把表示这个修改的日志记录写到日志文件中是两个不同的操 作。有可能在这两个操作之间发生故障，即这两个写操作只完成了一个。如果先写了数据库修改, 而在运行记录中没有登记下这个修改，则以后就无法恢复这个修改了。如果先写日志，但没有修改 数据库，按日志文件恢复时只不过是多执行一次不必要的UNDO操作，并不会影响数据库的正确 性。所以为了安全，一定要先写日志文件，即首先把日志记录写到日志文件中，然后写数据库的修 改。这就是“先写日志文件”的原则。

.五、恢复策略

**40**数据库系统概论(第**5**版)同步辅导及习题全解

.

事务故障是指事务在运行至正常终止点前被中止，这时恢复子系统应利用日志文件撤消（UN­DO） 此事务已对数据库进行的修改。 事务故障的恢复是由系统自动完成的，对用户是透明的。系统 的恢复步骤是：

（1） 反向扫描日志文件（即从最后向前扫描日志文件），查找该事务的更新操作。

（2） 对该事务的更新操作执行逆操作。即将日志记录中“更新前的值”写入数据库。这样，如果 记录中是插入操作,则相当于做删除操作（因此时“更新前的值”为空）。若记录中是删除操作，则做 插入操作，若是修改操作,则相当于用修改前的值代替修改后的值。

（3） 继续反向扫描日志文件,查找该事务的其他更新操作，并做同样处理。

（4） 如此处理下去，直至读到此事务的开始标记。

系统故障造成数据库不一致状态的原因有两个，一是未完成事务对数据库的更新可能已写入 数据库，二是已提交事务对数据库的更新可能还留在缓冲区没来得及写入数据库。因此恢复操作 就是要撤消故障发生时未完成的事务,重做已完成的事务。

系统故障的恢复是由系统在重新启动时自动完成的，不需要用户干预。

系统的恢复步骤是：

（1） 正向扫描日志文件（即从头扫描日志文件），找出在故障发生前已经提交的事务（这些事务 既有BEGIN TRANSACTION记录，也有COMMIT记录），将其事务标识记入重做（REDO）队列。 同时找出故障发生时尚未完成的事务（这些事务只有BEGIN TRANSACTION记录，无相应的 COMMIT记录），将其事务标识记入撤消队列。

（2） 对撤消队列中的各个事务进行撤消（UNDO）处理。

进行UNDO处理的方法是:反向扫描日志文件，对每个UNDO事务的更新操作执行逆操作，即 将日志记录中“更新前的值”写入数据库。

（3） 对重做队列中的各个事务进行重做（REDO）处理。

进行REDO处理的方法是:正向扫描日志文件，对每个REDO事务重新执行日志文件登记的操 作。即将日志记录中“更新后的值”写入数据库。

介质故障发生后，磁盘上的物理数据和日志文件被破坏，这是最严重的一种故障，恢复方法是 重装数据库，然后重做已完成的事务。具体地说就是：

（1）装入最新的数据库后备副本（离故障发生时刻最近的转储副本），使数据库恢复到最近一次

转储时的一致性状态。

对于动态转储的数据库副本，还须同时装入转储开始时刻的日志文件副本，利用恢复系统故障 的方法(即REDO+UNDO),才能将数据底恢复到一致性状态。

1. 装入相应的日志文件副本(转储结束时刻的日志文件副本)，重做已完成的事务。

第十章

典型例题

数据库恢复技术

例**1**事务中的提交和回滚是什么意思？

解析事务中的提交(COMMIT)是提交事务的所有操作。具体说就是将事务中所有对数据库的更 新写回到磁盘上的物理数据库中去，事务正常结束。事务中的回滚(ROLLBACK)是数据库 滚回到事务开始时的状态。具体地说就是，在事务运行的过程中发生了某种故障，事务不能 继续执行，系统将事务中对数据库的所有已完成的更新操作全部撤消,使数据库回滚到事务 开始时的状态。

例**2**为什么要设立日志文件？

解析 设立日志文件的目的，是为了记录对数据库中数据的每一次更新操作。从而DBMS可以根 据日志文件进行事务故障的恢复和系统故障的恢复,并可结合后援副本进行介质故障的 恢复。

例**3**分析图10.2。

|  |  |  |  |
| --- | --- | --- | --- |
| 正确运行 | 转储  1 | 运行事务  1 | 故障发生点  i |
| **1 Ta** | 1 **Tb** | **▼**  **Tf** |
| 恢复 | 重装后备副本  **1** | 重新运行事务  L |  |
|  | 图 10.2 |  |
|  |

解析•系统在匸时刻停止运行事务，进行数据库转储

•在K时刻转储完毕，得到Tb时刻的数据库一致性副本

•系统运行到Tf时刻发生故障

•为恢复数据库，首先由数据库管理员重装数据库后备副本，将数据库恢复至外时刻的

42

1

数据库系统概论(第**5**版)同步辅导及习题全解

课后习题解答

小提示：掌握事务的基本概念和 事务的ACID性质。了解数据库 恢复技术的重要性，针对不同的 故障类型，掌握恢复数据库的策 略和方法。

状态

•重新运行自Tb〜Tf时刻的所有更新事务，把数据库恢复到故障发生前的一致状态

例**4**为什么要先写日志文件？

解析1.写数据库和写日志文件是两个不同的操作。

1. 在这两个操作之间可能发生故障。
2. 如果先写了数据库修改，而在日志文件中没有登记下这个修改，则以后就无法恢复这个修 改了。
3. 如果先写日志，但没有修改数据库，按日志文件恢复时只不过是多执行一次不必要的UN- DO操作，并不会影响数据库的正确性。
4. 陸国区 事务是用户定义的一个数据库操作序列，这些操作要么全做要么全不做，事务是一

个不可分割的工作单元。

事务具有4个特性:原子性、一致性、隔离性和持续性。

原子性:事务是数据库的逻辑工作单位，事务中包括的操作要么都做,要么都不做。

一致性:事务执行的结果必须是使数据库从一个一致性状态变到另一个一致性 状态。

隔离性:一个事务的执行不能被其他事务干扰。即一个事务内部的操作及使用的数 据对其他并发事务是隔离的，并发执行的各个事务之间不能互相干扰。

持续性:持续性也称为永久性,指一个事务一旦提交，它对数据库中数据的改变就应 该是永久生效的。接下来的其他操作或故障不应该对其执行结果有任何影响。 恢复技术能保证事务的原子性、持久性。

1. 胸厘国留，事务执行的结果必须是使数据库从一个一致性

状态变到另一个一致性状态。如果数据库系统 运行中发生故障,有些事务尚未完成就并迫中 断，这些未完成事务对数据库所做的修改有一部分已写入物理数据库，这时数据库

就处于一种不正确的状态，或者说是不一致性状态。

例如，某工厂的库存管理系统中，要把数量为Q的某种零件从仓库1移到仓库2存 放,则可以定义一个事务T。T包括两个操作:Q1 = Q1 —Q,Q2 = Q2 + Q。如果T 非正常终止时只做了第一个操作，则数据库就处于不一致性状态，库存量无缘无故 少了 Q。

43

1—

1. 【解题把对数据的修改写到数据库中和把表示这个修改的日志记录写到日志文件中是两

第十章

个不同的操作。有可能在这两个操作之间发生故障，即这两个写操作只完成了 一个。

数据库恢复技术

如果先写了数据库修改，而在运行记录中没有登记这个修改，则以后就无法恢复这 个修改了。如果先写日志，但没有修改数据库，在恢复时只不过是多执行一次不必 要的UNDO操作，并不会影响数据库的正确性。所以一定要先写日志文件，即首先 把日志记录写到日志文件中，然后写数据库的修改。

1. 匯厘W (1)如果系统故障发生在14之后，说明哪些事务需要重做,哪些事务需要回滚。

T】和T3需要重做,匸需要回滚；

1. 如果系统故障发生在10之后呢？

T】需要重做,T3需要回滚；

1. 如果系统故障发生在9之后呢？

T,需要重做,T2和T3需要回滚；

1. 如果系统故障发生在7之后呢？

T］需要重做,T2需要回滚。

1. (1)如果系统故障发生在14之后，写出系统恢复后A、B、C的值。

A = 8，B = 7,C = 11；

1. 如果系统故障发生在12之后呢？

A = 10,B = 0,C = 11 ；

1. 如果系统故障发生在10之后呢？

A = 10,B = 0,C = 11 ；

1. 如果系统故障发生在9之后呢？

A = 10,B = 0,C = 11 ；

1. 如果系统故障发生在7之后呢？

A = 10,B = 0,C = 11：

（6）如果系统故障发生在5之后呢?

A = 0,B = 0,C = 0o

小提示：事务处理技术主要包括 数据库恢复技术和并发控制技 术。事务是数据库恢复和并发控 制的基本单位。

**44**数据库系统概论(第**5**版)同步辅导及习题全解

6.廨函曇屉事务故障的恢复:

事务故障的恢复是由DBMS自动完成的，对用

户是透明的。

DBMS执行恢复步骤是：

（1） 反向扫描日志文件（即从最后向前扫描日志文件），查找该事务的更新操作；

（2） 对该事务的更新操作执行逆操作，即将日志记录中“更新前的值”写入数据库；

（3） 继续反向扫描日志文件,做同样处理；

（4） 如此处理下去，直至读到此事务的开始标记，该事务故障的恢复就完成了。

系统故障的恢复：

系统故障可能会造成数据库处于不一致状态:一是未完成事务对数据库的更新可能 已写入数据库;二是已提交事务对数据库的更新可能还留在缓冲区，没来得及写入 数据库。因此恢复操作就是要撤消（UNDO）故障发生时未完成的事务，重做（RE­DO） 已完成的事务。

系统的恢复步骤是：

（1） 正向扫描日志文件，找出在故障发生前已经提交的事务队列（REDO队列）和未 完成的事务队列（UNDO队列）。

（2） 对撤消队列中的各个事务进行UNDO处理。

进行UNDO处理的方法是:反向扫描日志文件，对每个UNDO事务的更新操作执 行逆操作，即将日志记录中“更新前的值"Before Image）写入数据库。

（3） 对重做队列中的各个事务进行REDO处理。

进行REDC）处理的方法是:正向扫描日志文件，对每个REDC）事务重新执行日志文 件登记的操作，即将日志记录中“更新后的值”After Image）写入数据库。

介质故障的恢复：

介质故障是最严重的一种故障。 \

恢复方法是重装数据库，然后重做已完成的事务。

具体过程是：

（1） DBA装入最新的数据库后备副本（离故障发生时刻最近的转储副本），使数据

库恢复到转储时的一致性状态；

（2） DBA装入转储结束时刻的日志文件副本；

（3） DBA启动系统恢复命殳，由DBMS完成恢复功能，即重做已完成的事务。 检查点记录是一类新的日志记录。它的内容包括：

45

1

第十章

数据库恢复技术

（1） 建立检查点时刻所有正在执行的事务清单。

（2） 这些事务的最近一个日志记录的地址。

利用日志技术进行数据库恢复时，恢复子系统必须搜索日志，确定哪些事务需要 REDO、哪些事务需要UNDO。一般来说,需要检查所有的日志记录。这样做有两 个问题:一是搜索整个日志将耗费大量的时间;二是很多需要REDO处理的事务实 际上已经将它们的更新操作结果写到数据库中了，恢复子系统又重新执行了这些操 作，浪费了大量时间。检査点技术就是为了解决这些问题而提出的。

在采用检査点技术之前，恢复时需要从头扫描日志文件;而利用检查点技术只需要 从检查点开始扫描日志，从而缩短了扫描日志的时间。

利用日志技术进行数据库恢复时，恢复子系统必须搜索日志，确定哪些事务需要 REDO,哪些事务需要UNDO。一般来说，需要检查所有日志记录。这样做有两个 问题:一是搜索整个日志将耗费大量的时间。二是很多需要REDO处理的事务实 际上已经将它们的更新操作结果写到数据库中了，恢复子系统又重新执行了这些操 作，浪费了大量时间。检查点技术就是为了解决这些问题。

例如某事务I的更新操作实际上已经写到数据库中了，进行恢复时没有必要再 REDO处理，而釆用检查点技术就可以做到这一点。

（1）从重新开始文件中找到最后一个检査点记录在日志文件中的地址，由该地址在 日志文件中找到最后一个检查点记录。（2）由该检查点记录得到检查点建立时刻所 有正在执行的事务清单ACTIVE-LISTO

这里建立两个事务队列：

DUNDO-LIST：需要执行UNDO操作的事务集合；

2） REDO—LIST：需要执行REDO操作的事务集合。

把ACTIVE-LIST暂时放入UNDO—LIST队列,REDO队列暂为空。

3） 从检査点开始正向扫描日志文件：

①如有新开始的事务Ti,把Ti暂时放入UNDO-LIST队列；

②如有提交的事务Tj,把Tj从UNDO-LIST队列移到REEK)-LIST队列，直到 日志文件结束；

46

1—

数据库系统概论(第**5**版)同步辅导及习题全解

1. 对UNDO-LIST中的每个事务执行UNDO操作，对REDO-LIST中的每个事 务执行REDO操作。

**10.**［解題国**k**数据库镜像即根据DBA的要求，自动把整个数据库或者其中的部分关键数据复制 到另一个磁盘上。每当主数据库更新时，DBMS自动把更新后的数据复制过去，即 DBMS自动保证镜像数据与主数据的一致性。

数据库镜像的用途有：

一是用于数据库恢复。当出现介质故障时，可由镜像磁盘继续提供使用，同时 DBMS自动利用镜像磁盘数据进行数据库的恢复，不需要关闭系统和重装数据库 副本。

二是提高数据库的可用性。在没有出现故障时，当一个用户对某个数据加排他锁 进行修改时，其他用户可以读镜像数据库上的数据，而不必等待该用户释放锁。

47

1—

第十一章

第十一章

并发控制

并发控制

知识网络图

丢失修改 不可重复读

〔读“脏”数据

排他锁

共享锁

封锁和封锁协议＜

一级封锁协议

活锁和死锁-

死锁（死锁的预防

死锁的诊断与解除

并发调度的可串行性

可串行化调度

冲突可串行化调度

多粒度封锁

显式封锁

隐式封锁

I封锁的粒度

意向共享锁

二级封锁协议 〔三级封锁协议

并发控制＜

活锁

意向排他锁 共享意向排他锁

48

1—

数据库系统概论(第**5**版)同步辅导及习题全解

知识点归纳

重难点

重点:掌握并发操作产生的数据不一致性(丢失修改、不可重复读、读“脏”数据)的确切含义。 封锁协议与数据一致性的关系;并发调度的可串行性概念。

难点：两段锁协议与串行性的关系、与死锁的关系。具有意向锁的多粒度封锁方法的封锁 过程。

■ \_、并发控制概述

并发操作带来的数据不一致性

1. 丢失修改(Lost Update)
2. 不可重复读(Non—repeatable Read)
3. 读“脏”数据(Dirty Read)

.二、封锁

1. 排他锁又称为写锁

若事务T对数据对象A加上X锁，则只允许T读取和修改A,其他任何事务都不能再对A加 任何类型的锁，直到T释放A上的锁。

1. 共享锁又称为读锁

若事务T对数据对象A加上S锁,则事务T可以读A但不能修改A,其他事务只能再对A加 S锁，而不能加X锁，直到T释放A上的S锁。

.三、封锁协议 十

在运用X锁和S锁对数据对象加锁时，需要约定一些规则，这些规则为封锁协议。

三级封锁协议：

1. 一级封锁协议
2. 二级封锁协议
3. 三级封锁协议 、

49

1—

■四、活锁和死锁

第十 一章

1. 活锁

事务r封锁了数据R,事务T2又请求封锁R,于是T2等待。T3也请求封锁R,当P释放了 R上的封锁之后系统首先批准了 T3的请求,T2仍然等待。T4又请求封锁R,当T3释放了 R上的 封锁之后系统又批准了 T.的请求……T2有可能永远等待,这就是活锁的情形。

并发控制

1. 死锁

事务匸封锁了数据Rlo T2封锁了数据**R2,Ti**又请求封锁R2,因T2已封锁了 R2,于是P等 待T2释放R2上的锁。接着T2又申请封锁R,因T】已封锁了 R,,T2也只能等待**Ti**释放R上的 锁。这样I在等待T2，而T2又在等待T］，卩和T2两个事务永远不能结束，形成死锁

.五、并发调度的可串行性

1. 可串行化调度

多个事务的并发执行是正确的，当且仅当其结果与按某一次序串行地执行这些事务时的结果 相同。

1. 冲突可串行化调度

一个调度Sc在保证冲突操作的次序不变的情况下,通过交换两个事务不冲突操作的次序得到 另一个调度Sc',如果Sc'是串行的，则称调度Sc是冲突可串行化的调度。

冲突可串行化调度是可串行化调度的充分条件，不是必要条件。还有不满足冲突可串行化条 件的可串行化调度。

I

■六、两段锁协议

1. 指所有事务必须分两个阶段对数据项加锁和解锁

在对任何数据进行读、写操作之前，事务首先要获得对该数据的封锁，在释放一个封锁之后，事务不再申请和获得任何其他封锁。

事务分为两个阶段

50

1—

数据库系统概论(第**5**版)同步辅导及习题全解

第一阶段是获得封锁，也称为扩展阶段。

事务可以申请获得任何数据项上的任何类型的锁，但是不能释放任何锁。

第二阶段是释放封锁，也称为收缩阶段。

事务可以释放任何数据项上的任何类型的锁，但是不能再申请任何锁。

■七、封锁的粒度

1. 封锁对象的大小称为封锁粒度(Granularity)

封锁的对象:逻辑单元、物理单元。

允许多粒度树中的每个结点被独立地加锁，对一个结点加锁意味着这个结点的所有后裔结点 也被加以同样类型的锁，在多粒度封锁中一个数据对象可能以两种方式封锁:显式封锁和隐式 封锁。

1. 意向锁’
2. 如果对一个结点加意向锁，则说明该结点的下层结点正在被加锁。对任一结点加基本锁, 必须先对它的上层结点加意向锁。
3. 具有意向锁的多粒度封锁方法

提高了系统的并发度；

减少了加锁和解锁的开销；

在实际的数据库管理系统产品中得到广泛应用。

典型例题

例**1**现在有两个事务，分别包含下列操作:

•事务卩：读B；A=B+1；写回A

•事务T?：读A；B=A+1；写回B

解析现给出对这两个事务不同的调度策略

第十一章

并发控制

|  |  |
| --- | --- |
| Ti | t2 |
| Slock B |  |
| Y=R(B)=2 |  |
| Unlock A |  |
| A=Y+1 = 3 |  |
| W(A) |  |
| Unlock A | •假设A、B的初值均为2。 |
|  | Slock A ・按TlT2次序执行结果为A=3,B=4 |
|  | X=R(A)=3 ・串行调度策略，正确的调度 |
|  | Unlock A |
|  | Xlock B |
|  | B=X+1=4 |
|  | W(B) |
|  | Unlock B |
| 串行调度(a) | |
| Tj | t2 |
|  | Slock A |
|  | X=R(A) = 2 |
|  | Unlock A |
|  | Xlock B |
|  | B=X+1 = 3 |
|  | W(B) |
|  | •收攻A、U的初值均为Zo  Unlock B |
| Slock B | •按T2-\*Ti次序执仃结果为B—3,A—4 |
| Y=R(B) = 3 | •串行调度策略，止确的调度 |
| Unlock B |  |
| Xlock A |  |
| A=Y+1=4 |  |
| W(A) |  |
| Unlock A |  |

2

5

1—

数据库系统概论(第**5**版)同步辅导及习题全解

|  |  |
| --- | --- |
| Ti | t2 |
| Slock B |  |
| Y=R(B)=2 |  |
|  | Slock A |
|  | X=R(A)=2 |
| Unlock B |  |
|  | Unlock A |
| Xlock A |  |
| A=Y+l=3 |  |
| W(A) |  |
|  | Xlock B |
|  | B=X+l = 3 |
|  | W(B) |
| Unlock A |  |
|  | Unlock B |
| 不可串行化的调度 | |
| Ti | t2 |
| Slock B |  |
| Y=R(B) = 2 |  |
| Unlock B |  |
| Xlock A |  |
|  | Slock A |
| A=Y+l=3 | 等待 |
| W(A) | 等待 |
| Unlock A | 等待 |
|  | X=R(A)=3 |
|  | Unlock A |
|  | Xlock B |
|  | B=X+l=4 |
|  | W(B) |
|  | Unlock B |

・执行结果与串行调度(a)(b)的结果都不同

・是错误的调度

・执行结果与串行调度(a)的执行结果相同

•是正确的调度

可串行化的调度

例2有3个事务

Ti = Wi (Y)W| (X) ,T2=W2(Y)W2 (X) ,T3 =W\3 (X)

\

解析 •调度 L1=W1(Y)W1(X)W2(Y)W2(X) W3(X)是一个串行调度。

•调度L2=Wl(Y)W2(Y)W2(X)W1(X)W3(X)不满足冲突可串行化条件。但是调度**l2**是 可串行化的，因为1-2执行的结果与调度L1相同，Y的值都等于T2的值,X的值都等于T3 的值

第十一章

并发控制

课后习题解答

1. 解题gk 数据库的一个明显的特点是多个用户共享数据库资源，尤其是多用户可以同时存

取相同的数据。在这样的系统中，在同一时刻并发运行的事务数可达数百个。若对 事务的并发操作不加以控制，就会造成数据存、取的错误，破坏了数据的一致性和完 整性。并发控制可以防止数据不 yj知识点播：充分利用数据库系统资源，但并发］

一祈柱屮削 、操作带来了数据的不一致性，主要有三种：1

一以在①说。 ［丢失修改2不可重复读3读“脏”数据

1. 陸国k并发操作带来的数据不一致性包括三类:丢失修改、不可重复读和读“脏”数据。
2. 丢失修改(lost update)：两个事务卩和T2读入同一数据并修改,T2提交的结果 破坏了(覆盖了)1提交的结果，导致匚的修改被丢失。
3. 不可重复读(Non-Repeatable Read)：事务「读取数据后，事务T2执行更新操 作.使T】无法再现前一次读取结果。
4. 读“脏”数据:事务卩修改某一数据，并将其写回磁盘，事务T2读取同一数据后,

T,由于某种原因被撤消，这时卩已修改过的数据恢复原值,T2读到的数据就与数 据库中的数据不一致，则T2读到的数据就为“脏”数据，即不正确的数据。避免不一 致性的方法和技术就是并发控制。最常用的技术是封锁技术,也可以用其他技术, 例如在分布式数据库系统中可以釆用时间戳方法来进行并发控制。

1. 厘SB 封锁就是事务T在对某个数据对象例如表、记录等进行操作之前，先向系统发出请

求，对其加锁。加锁后事务T就对该数据对象有了一定的控制,在事务T释放它的 锁之前，其他的事务不能更新此数据对象。封锁是实现并发控制的一个非常重要的 技术。

基本的封锁类型有两种：排他锁和共享锁。排他锁又称为写锁。若事务T对数据

54

1—

数据库系统概论(第**5**版)同步辅导及习题全解

对象A加上X锁，则只允许T读取和修改A,其他任何事务都不能再对A加任何 类型的锁，直到T释放A上的锁。这就保证了其他事务在T释放A上的锁之前 不能再读取和修改A。共享锁又称为读锁。若事务T对数据对象A加上S锁，则 事务T可以读A但不能修改A,其他事务只能再对A加S锁，而不能加X锁，直 到T释放A上的S锁。这就保证了其他事务 可以读A,但在T释放A上的S锁之前不能 对A做任何修改。

［小提示： 排它锁（Exclusive Locks,  
简称X锁）和龚享锁（Share

I Locks,简总S锁

数据库管理系统在对数据进行读、写操作之前首先对该数据执行封锁操作，例如下 图中事务T1在对A进行修改之前先对A执行XLock（A），即对A加X锁。这样, 当*T2*请求对A加X锁时就被拒绝,T2只能等待T1释放A上的锁后才能获得对 A的X锁,这时它读到的A是T1更新后的值，再按此新的A值进行运算。这样就

|  |  |
| --- | --- |
| Ti | t2 |
| ① Xlock A |  |
| 获得 |  |
| ②读A=l6 |  |
|  | Xlock A |
| ③ A\*-A-1 | 等待 |
| 写回A=l5 | 等待 |
| Commit | 等待 |
| Unlock A | 等待 |
| ④ | 获得Xlock A |
|  | 读 A=l5 |
|  | A—A—I |
|  | 写回A=l4 |
| ⑤ | Commit |

不会丢失T1的更新。

数据库管理系统按照一定的封锁协 议,对并发操作进行控制，使得多个 并发操作有序地执行，就可以避免 丢失修改、不可重复读和读“脏”数 据等数据不一致性问题。

Unlock A

第十一章

并发控制

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | T2 | T3 | T4 |
| lock R | 、 |  |  |
|  | lock R |  |  |
|  | 等待 | lock R |  |
| Unlock等待 |  |  | lockR |
| . 等待 |  | 等待 |  |
| , 等待 |  | 等待 |  |
| . 等待 | Unlock | 等待 |  |
| . 等待 |  | lock R |  |
| . 等待 |  |  |  |

如果事务T1封锁了数据R,事务T2又请求封锁R,于是T2等待。T3也请求封锁 R,当T1释放了 R上的封锁之后系统首先批准了 T3的请求,T2仍然等待。然后 T4又请求封锁R,当T3释放了 R上的封锁之后系统又批准了 T4的请求……T2有 可能永远等待，这就是活锁的情形。活锁的含义是该等待事务等待时间太长，似乎 被锁住了,实际上可能被激活。

活锁产生的原因：当一系列封锁不能按照其先后顺序执行时,就可能导致一些事务 无限期等待某个封锁，从而导致活锁。

避免活锁的简单方法是采用先来先服务的策略。

当多个事务请求封锁同一数据对

象时，封锁子系统按请求封锁的先后次序对事

日小提示：重点掌握封锁技术、封  
］锁协议、活锁和死锁的概念。

务排队，数据对象上的锁一旦释放就批准申请 队列中第一个事务获得锁。

如果事务T1封锁了数据R1 ,T2封锁了数据R2,然后T1又请求封锁R2,因T2已 封锁了 R2,于是T1等待T2释放R2上的锁。接着T2又申请封锁R1,因T1已封 锁了 RLT2也只能等待T1释放R1上的锁。这样就出现了 T1在等待T2,而T2又 在等待T1的局面,T1和*T2*两个事务永远不能结束，形成死锁。



**56**数据库系统概论(第**5**版)同步辅导及习题全解

| T1 | T2 |
| --- | --- |
| lock R1 |  |
|  | lock R2 |
|  |  |
| lock R2 |  |
| 等待 |  |
| 等待 | lock R1 |
| 等待 | 等待 |

知识点播：熟练掌握两端锁协议, 了解多粒度封锁的概念,掌握意 向锁的定义和强度。

在数据库中，产生死锁的原因是两个或多个事务都已封锁了一些数据对象，然后又 都请求已被其他事务封锁的数据加锁，从而出现死等待。

防止死锁的发生其实就是要破坏产生死锁的条件。预防死锁通常有两种方法：

（1） 一次封锁法，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；

（2） 顺序封锁法,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。

1. 應题匪' 数据库系统一般釆用允许死锁发生，数据库管理系统检测到死锁后加以解除的方

法。数据库管理系统中诊断死锁的方法与操作系统类似，一般使用超时法或事务等 待图法。

超时法是:如果一个事务的等待时间超过了规定的时限，就认为发生了死锁。超时 法实现简单，但有可能误判死锁，事务因其他原因长时间等待超过时限时，系统会误 认为发生了死锁。若时限设置得太长,又不能及时发现死锁发生。

数据库管理系统并发控制子系统检测到死锁后，就要设法解除。通常采用的方法是 选择一个处理死锁代价最小的事务，将其撤消，释放此事务持有的所有锁，使其他事 务得以继续运行下去。当然，对撤消的事务所执行的数据修改操作必须加以恢复。

1. 解题 可串行化的调度是正确的调度。

可串行化的调度的定义:多个事务的并发执行是正确的，当且仅当其结果与按某一 次序串行执行它们时的结果相同，称这种调度策略为可串行化的调度。

1. 匯凰蹈k （1）若这3个事务允许并行执行，则有务少可能的正确结果，请一一列举出来。

A的最终结果可能有2、4、8、16。

因为串行执行次序有 Ti T2 T3、T] T3 T2.T2 Ti T3、T2 T3 TlT3 T）T2、T3 T2 Ti o 对应的执行结果是16、8、4、2、4、2。

(2)请给出一个可串行化的调度，并给出执行结果。

|  |  |  |
| --- | --- | --- |
| Ti | t2 | t3 |
| slock A | \ |  |
| Y=A=0 |  |  |
| Unlock A |  |  |
| Xlock A |  |  |
|  | Slock A |  |
| A=Y+2 | 等待 |  |
| 写回A(=2) | 等待 |  |
| Unlock A | 等待 |  |
|  | Y=A=2 |  |
|  | Unlock A |  |
|  | Xlock A |  |
|  |  | Slock A |
|  | A=Y\*2 | 等待 |
|  | 写回A(=4) | 等待 |
|  | Unlock A | 等待 |
|  |  | Y=A=4 |
|  |  | Unlock A |
|  |  | Xlock A |
|  |  | 写回A( = 16) |
|  |  | Unlock A |

最后结果A为16,是可串行化的调度。

(3)请给出一个非串行化的调度，并给出执行结果。

|  |  |  |
| --- | --- | --- |
| Ti | t2 | t3 |
| Slock A |  |  |
| Y=A=0 |  |  |
| Unlock A |  |  |
|  | Slock A |  |
|  | Y=A=0 |  |
| Xlock A |  |  |

Unlock A

等待

57

第十一章

并发控制

**58**数据库系统概论(第**5**版)同步辅导及习题全解

|  |  |  |
| --- | --- | --- |
| Ti | t2 | t3 |
| A=Y+2 |  |  |
| 写回A(=2) | Slock A |  |
| Unlock A | 等待 |  |
|  | Y=A=2 |  |
|  | Unlock A |  |
|  |  | Xlock A |
|  | Xlock A |  |
|  | 等待 | A=Y\* \* 2 |
|  | 等待 | 写回A( = 4) |
|  | 等待 | Unlock A |
|  | A=Y\*2 |  |
|  | 写回A(=0) |  |
|  | Unlock A |  |

最后结果A为0,为非串行化的调度。

（4）若这3个事务都遵守两段锁协议,请给出一个不产生死锁的可串行化调度。

|  |  |  |
| --- | --- | --- |
| T] | t2 | t3 |
| Slock A |  |  |
| Y=A=O |  |  |
| Xlock A |  |  |
| A=Y+2 | Slock A |  |
| 写回A(=2) | 等待 |  |
| Unlock A | 等待 |  |
| Y=A=2 |  |  |
| Xlock A |  |  |
|  | 等待 | Slock A |
|  | A=Y\*2 | 等待 |
|  | 写回A(=4) | 等待 |
|  | Unlock A | 等待 |
|  |  | Y=A=4 |
|  |  | Xlock A |
|  |  | A=Y\* \*2 |

9

5

1—

第十一章

并发控制

|  |  |  |
| --- | --- | --- |
| Ti | t2 | t3 |
|  |  | 写回A( = l6) |
|  |  | Unlock A |
| (5)若这3个事务都遵守两段锁协议，请给出一个产生死锁的调度。 | | |
| Ti | t2 | t3 |
| Slock A |  |  |
| Y=A=0 |  |  |
|  | Slock A |  |
|  | Y=A=0 |  |
| Xlock A |  |  |
| 等待 |  |  |
|  | Xlock A |  |
|  | 等待 |  |
|  |  | Slock A |
|  |  | Y=A=0 |
|  |  | Xlock A |
|  |  | 等待 |

该调度是冲突可串行化的调度。

Sci=r3 (B)n (A)w3 (B)r2 (B)r2 (A)w2 (B)n (B)wi(A)

交换n(A)和W3(B)得到

r3(B)w3 (B)ri (A)r2 (B)r2 (A)W2 (B)n (A)n (B)w)(A)

再交换 ij(A)和 r2(B)r2(A)w2(B)得到

S C2 =r3 (B)w3 (B)r2 (B)r2 (A)w2 (B)r)(A)n (B)wi (A)

由于SC2是串行的，而且两次交换都是基于不冲突操作的。所以Sc】是冲突可串 行化的调度。

首先以两个并发事务I和T2为例，存在多个并发事务的情形可以类推。根据可 串行化定义可知，事务不可串行化只可能发生在下列两种情况：

1. 事务T］写某个数据对象A,T2读或写A；

［小提示：数据库管理系统必须提 3 供并发控制机制来协调并发用户 \ 的并发操作以保证并发事务的隔 离性和一致性，保证数据库的一 致性。

1. 事务T】读或写某个数据对象A,T2写A。 下面称A为潜在冲突对象。

设「和T2访问的潜在冲突的公共对象为｛A］,A2,…,A3。

**60**数据库系统概论(第**5**版)同步辅导及习题全解

不失一般性，假设这组潜在冲突对象中X={A|,A2,…，A3均符合情况（1）。

Y={Ai + l,…,A3符合所情况（2）。

那么,Ti需要Xlock x①

T2 需要 Slock x 或 Xlock x ②

1） 如果操作①先执行，则卩获得锁,T2等待

由于遵守两段锁协议,T】在成功获得X和Y中全部对象及非潜在冲突对象的锁 后,才会释放锁。

这时如果写X或Y,T2已获得写的锁，则出现死锁；

否则,1在对X、Y中对象全部处理完毕后,T2才能执行。

这相当于按T）.T2的顺序串行执行,根据可串行化定义,**Ti**和T2的调度是可串行 化的。

2） 操作②先执行的情况与（1）对称

因此,若并发事务遵守两段锁协议，在不发生死锁的情况下,对这些事务的并发调 度一定是可串行化的。

证毕。

**2.**

**1**

|  | |
| --- | --- |
| Ti | t2 |
| SIj?ck B |  |
| 读B=2 |  |
| Y=B |  |
| Unlock B |  |
| XLock A |  |
|  | SIx)ck A |
|  | 等待 |
| A=Y+1 | 等待 |
| 写回A=3 | 等待 |
|  |  |
| Unlock A | 等待 |
|  | SIx)ck A |
|  | 读A=3 |
|  | X=A |

|  |  |
| --- | --- |
| Ti | t2 |
|  | Unlock A |
|  | XLock B |
|  | b=x¥i |
|  | 写回B=4 |
|  | Unlock B |

陸遵循两阶段封锁（2PL）的调度U可串行化的调度U串行调度U正确的调度。

第十一章

'解题图区 增加加锁和解锁操作后的T］和T2事务可串行执行，不会引起死锁。调度如下表

并发控制

所示:

|  |  |
| --- | --- |
| Ti | t2 |
| SLCX2K A |  |
| XLOCK B |  |
| R(A) | SLOCK B |
| R(B) | 等待 |
| B=A+B | 等待 |
| W(B) | 等待 |
| UNLOCK A | 等待 |
| UNLOCK B | XL()CK A |
|  | R(B) |
|  | R(A) |
|  | A=A+B |
|  | UNLOCK B |
|  | UNLOCK A |

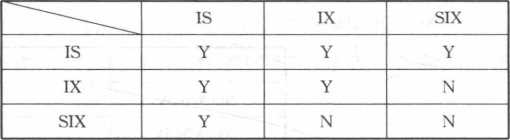
引进意向锁是为了提高封锁子系统的效率。该封锁子系统支持多种封锁粒度。原 因是:在多粒度封锁方法中一个数据对象可能以两种方式加锁——显式封锁和隐 式封锁。因此系统在对某一数据对象加锁时不仅要检查该数据对象上有无（显式 和隐式）封锁与之冲突;还要检查其所有上级结点和所有下级结点，看申请的封锁 是否与这些结点上的（显式和隐式）封锁冲突;显然,这样的检查方法效率很低。为 此引进了意向锁。

意向锁的含义是:对任一结点加锁时,必须先对它的上层结点加意向锁。例如事务 T要对某个元组加X锁,则首先要对关系和数据库加IX锁。换言之，对关系和数

据库加IX锁，表示它的后裔结点中的某个元组拟加X锁。引进意向锁后，系统对 某一数据对象加锁时不必逐个检查与下一级结点的封锁冲突了。例如，事务T要 对关系R加X锁时，系统只要检查根结点数据库和R本身是否已加了不相容的 锁（如发现已经加了 IX,则与X冲突），而不再需要搜索和检查R中的每一个元组 是否加了 X锁或S锁。

**62**数据库系统概论(第**5**版)同步辅导及习题全解

IS锁:意向共享锁。IX锁:意向排他锁°SIX锁=，+1乂：共享意向排他锁。

例:对某个表加SIX锁，表示该事务要读整个表（所以要对该表加S锁）,同时会更 新个别元组（所以要对该表加IX锁）。

原则:读锁S可读不可写，写锁X不可读写。

16：

第十二章

数据库管理系统

第十二章

数据库管理系统

知识网络图

数据库管理系统的基本功能＜

数据库定义和创建

数据组织、存储和管理 数据存取

数据库事务管理和运行管理 数据库的建立和维护

〔其他功能

层次结构＜

数据库管理系统＜

数据库管理系统的系统结构〈

1应用层

语言处理层 数据存取层 〔数据存储层

I运行过程示例

任务和工作步骤

语言处理层〈解释方法

、预编译方法

［系统结构

［记录存取、事务管理子系统 日志登记子系统 控制信息管理模块 排序合并子系统 存取路径维护子系统 封锁子系统

数据存取层

数据字典的组织

［数据库的物理组织（数据及数据联系的组织

、存取路径的组织

重难点

重点:掌握数据库管理系统(DBMS)的功能的概念。需要举一反三的是:进一步了解数据库管 理系统的基本功能、系统结构及主要的实现技术。针对数据库应用系统开发人员，使他们从宏观和 总体的角度掌握DBMS的基本概念和基本原理.以便更好地使用和维护数据库管理系统。

难点：数据库管理系统的基本功能、系统结构及主要的实现技术,是数据库管理员和数据库应 用系统开发人员应该掌握的内容。本章按照RDBMS的层次结构依次介绍了语言处理层、数据存取 层、缓冲管理层、数据物理组织层等各个层次的主要任务和功能、涉及的主要概念和问题。了解在 数据库领域从理论的创立和工作原理的推出，到实验研究和原型系统的研制，最后演化为商品的成 功过程。

知识点归纳

■ 一、数据库管理系统的基本功能

1. 数据库定义和创建
2. 数据组织、存储和管理
3. 数据存取
4. 数据库事务管理和运行管理
5. 数据库的建立和维护
6. 其他功能

■二、数据库管理系统的进程结构和多线索机制

1. DBMS与应用程序相融合的方案
2. 一个DBMS进程对应一个用户进程
3. 一个DBMS进程对应所有用户进程
4. M个DBMS进程对应N个用户进程
5. 多线索DBMS的概念

■三、数据库管理系统的系统结构

\

层次结构:第一层是应用层，第二层是语言处理层，第三层是数据存取层，第四层是数据存 储层。

■四、语言处理

1. 语言处理层的任务和工作步骤

语言处理层的任务就是把用户在各种方式下提交给RDBMS的数据库语句转换成对RDBMS 内层可执行的基本存取模块的调用序列。

1. 解释方法

解释执行方法的具体做法是:直到执行前，数据库DML语句都以原始字符串的形式保存。当 执行到该语句时，才利用解释程序去完成图中所示的全部过程，同时予以执行。这种方法通过尽量 推迟束缚过程来赢得数据独立性。

1. 预编译方法

预编译方法的基本思想是:在用户提交DML语句之后，在运行之前对它进行翻译处理,保存产 生好的可执行代码。当需要运行时，取出保存的可执行代码加以执行。

.五、数据存取层

数据存取层介于语言处理层和数据存储层之间。它向上提供单元组接，即导航式的一次一个 元组的存取操作;向下则以系统缓冲区的存储器接U1作为实现基础。

1. 数据存取层的系统结构
2. 数据存取层的功能子系统
3. 日志登记子系统

■六、缓冲区管理

系统缓冲区可由内存或虚存组成。

系统缓冲的设立是出于两方面的原因：一是它把存储层以上各系统成分和实在的外存设甘隔 离。外存设备的变更不会影响其他系统成分,使关系数据库管理系统具有设备独立性。二是提高 存取效率关系，数据库管理系统利用系统缓冲滞留数据。当存取层需要读取数据时,存储子系统首 先到系统缓冲区中查找。

**56**数据库系统概论(第**5**版)同步辅导及习题全解

只有当缓冲区中不存在该数据时才真正从外存读入该数据所在的页面。当存取层写回一元组 到数据库中时，存储子系统并不把它立即写回外存，仅把该元组所在的缓冲区页当作一标志，表示 可以释放。只有当该用户事务结束或缓冲区已满需要调入新页时才按一定的淘汰策略把缓冲区中 已有释放标志的页写回外存。这样可以减少内外存交换的次数，提高存取效率。

.七、数据库物理组织

数据库中数据组织与数据之间联系是紧密结合的。在数据的组织和存储中必须直接或间接、 显示或隐含地体现数据之间的联系，这是数据库物理组织中主要考虑和设计的内容。在网状和层 次数据库中，常用邻接法和链接法实现数据之间的联系。网状数据库中最常用的组织策略是:各记 录型分别用某种文件结构组织，记录数据之间的联系。

典型例题

例**1**试述数据库技术的发展过程。

解析(1)数据模型是数据库系统的核心和基础。数据库技术的三个发展阶段应该按照数据模型 的进展来界定。按照数据模型的进展，数据库技术可以相应地分为三个发展阶段。

1. 数据模型的发展经历了格式化数据模型(包括层次数据模型和网状数据模型)、关系数据 模型两个阶段，发展到以面向对象数据模型为代表的非传统数据模型的阶段。
2. 读者可以从每一代数据库系统的主要特征、代表性系统、主要成就、优点和不足来了解数 据库技术的发展过程。

层次数据库系统和网状数据库系统的数据模型虽然分别为层次模型和网状模型，但实质上 层次模型是网状模型的特例。它们都是格式化模型。'它们从体系结构、数据库语言到数据 存储管理均具有共同特征，是第一代数据库系统。

关系数据库系统支持关系模型。关系模型不仅简单、清晰，而且有关系代数作为语言模型, 有关系数据理论作为理论基础。因此,关系数据库系统具有形式基础好、数据独立性强、数

据库语言非过程化等特色，标志着数据库技术发展到了第二代。

第二代数据库系统的数据模型虽然描述了现实世界数据的结构和一些重要的相互联系，但 是仍不能捕捉和表达数据对象所具有的丰富而重要的语义，因此尚只能属于语法模型。

第三代的数据库系统将以更加丰富的数据模型和更强大的数据管理功能为特征，从而满足 传统数据库系统难以支持的新的应用要求。

第十二章

例**2** 试述第一、二代数据库系统的主要成就。

解析第一代数据库系统指层次和网状数据库系统，其代表是：

数据库管理系统

（1） 1969年IBM公司研制的层次模型的数据库管理系统IMS。

（2） 美国数据库系统语言协商会CODASYL下属的数据库任务组DBTG对数据库方法进行 了系统的研究、探讨，于20世纪60年代末70年初提出了若干DBTG报告。DBTG报告确定 并建立了数据库系统的许多概念、方法和技术。DBTG所提议的方法是基于网状结构的。 它是数据库网状模型的典型代表。在DBTG方法和思想的指引下数据库系统的实现技术不 断成熟，开发了许多商品化的数据库管理系统，它们都是基于网状模型或层次模型的。

可以说，层次数据库是数据库系统的先驱，而网状数据库则是数据库概念、方法、技术的奠 基。它们是数据库技术中研究得出的最早的两种数据库系统。

支持关系数据模型的关系数据库系统是第二代数据库系统。

20世纪70年代是关系数据库理论研究和原型开发的时代,其中以IBM San Jose研究室开发 的System R和Berkeley大学研制的INGRES为典型代表。经过大量的高层次的研究和开 发取得了一系列的成果,主要是：

（1） 奠定了关系模型的理论基础，给出了人们一致接受的关系模型的规范说明。

（2） 研究了关系数据语言，有关系代数、关系演算、SQL语言及QBE等。这些描述性语言一 改以往程序设计语言和网状、层次数据库系统中数据库语言的风格，以其易学易懂的优点得 到了最终用户的喜爱，为20世纪80年代数据库语言标准化打下了基础。

（3） 研制了大量的RDBMS的原型，攻克了系统实现中查询优化、并发控制等难题。建立了故 障恢复等一系列关键技术。不仅大大丰富了 DBMS实现技术和数据库理论,更重要的是促 进了 RDBMS产品的蓬勃发展和广泛应用。

在计算机领域中把20世纪70年代称为数据库时代。20世纪80年代几乎所有新开发的系 统均是关系的。关系数据库系统从实验室走向了社会,数据库技术日益广泛地应用到企业 管理、情报检索、辅助决策等各个方面，成为实现和优化信息系统的基础和基本技术。

例**3**第三代数据库系统的主要特点是什么？

解析经过多年的研究和讨论，对第三代数据库系统的基本特征已有了共识。

8

6

1—

数据库系统概论(第**5**版)同步辅导及习题全解

（1） 第三代数据库系统应支持数据管理、对象管理和知识管理。

除提供传统的数据管理服务外，第三代数据库系统将支持更加丰富的对象结构和规则，应该 集数据管理、对象管理和知识管理为一体。

（2） 第三代数据库系统必须保持或继承第二代数据库系统的技术。

第三代数据库系统应继承第二代数据库系统已有的技术，如第二代数据库系统的非过程化 数据存取方式和数据独立性。不仅能很好地支持对象管理和规则管理，而且能更好地支持 原有的数据管理,支持多数用户需要的即席查询等。

（3） 第三代数据库系统必须对其他系统开放数据库系统的开放性表现在:支持数据库语言标 准;在网络上支持标准网络协议;系统具有良好的可移植性、可连接性、可扩展性和可互操作 性等。

例**4**请阐述以下数据库系统的主要概念、研究的主要问题及其发展过程：分布式数据库系统、并 行数据库系统、主动数据库系统、多媒体数据库系统、模糊数据库系统。

解析 下面仅仅给出有关概念：

分布式数据库系统:分布式数据库是由一组数据组成的，这组数据分布在计算机网络的不同 计算机上.网络中的每个结点具有独立处理的能力（称为场地自治），可以执行局部应用。同 时，每个结点也能通过网络通信子系统执行全局应用。

并行数据库系统:并行数据库系统是在并行机上运行的具有并行处理能力的数据库系统。 并行数据库系统是数据库技术与并行计算技术相结合的产物。

主动数据库系统:主动数据库是相对于传统数据库的被动性而言的。主动数据库能根据数 据库的当前状态，主动适时地做出反应,执行某些操作，向用户提供有关信息。主动数据库 是传统数据库技术与人工智能技术、面向对象技术相结合的产物。

多媒体数据库系统:可实现对格式化和非格式化的多媒体数据的存储、管理和查询的数据库 系统。

模糊数据库系统:存储、组织、管理和操作模糊数据的数据库系统。

例**5**试述数据仓库的产生背景。

解析（1）数据库技术的发展和广泛应用使许多部门、企业积累了大量的原始数据,这些数据是宝 贵的资源。

（2） 对这些数据的分析和利用可以了解企业运行的情况，发现存在的问题，预测未来的趋势。

（3） 数据库系统作为数据管理的先进技术已经被成功用于事务处理。但是它对分析处理的

支持一直不能令人满意，具体表现在：

1. 分析处理时性能低；
2. 分析的数据对象分散,而且不一致，即缺乏对数据的清洗、集成能力；
3. 事务处理系统不具备动态集成的羸力；
4. 系统缺乏对历史数据的有效组织和存储能力，而分析方法必须以大量的历史数据为依托；

第十二章

1. 在事务处理系统中存储的是细节数据，不适合进行分析处理,而事务处理系统又不具备对 数据的综合处理能力。

总之,DSS对数据在空间和时间的广度上都有了更高的要求，而事务处理环境难以满足这些 要求。在事务型环境中直接构建分析型应用是一种失败的尝试。数据仓库正是为了构建这 种新的分析处理环境而出现的一种数据存储和组织技术。但是数据仓库的主要驱动力并不 是过去的缺点，而是市场商业经营行为的改变，市场竞争要求捕获和分析事务级的业务 数据。

数据库管理系统

例**6** 什么是联机分析处理？什么是数据挖掘？

解析联机分析处理(OLAP)是以海量数据为基础的复杂分析技术。OLAP支持各级管理决策人 员从不同的角度、快速灵活地对数据仓库中的数据进行复杂查询和多维分析处理，并且能以 直观易懂的形式将查询和分析结果提供给决策人员，以方便他们及时掌握企业内外的情况, 辅助各级领导进行正确决策，提高企业的竞争力。

数据挖掘是从超大型数据库(VLDB)或数据仓库中发现并提取隐藏在内的模式的过程，这些 模式是有效的、新颖的、有潜在使用价值的和易于理解的。目的是帮助决策者寻找数据间潜 在的关联，发现经营者忽略的要素，而这些要素对预测趋势、决策行为也许是十分有用的 信息。

例**7** 基于数据库技术的**DSS**解决方案是什么？

解析 基于数据库技术的DSS的解决方案是:DW+()LAP+DM、DSS的可行方案。

数据仓库、联机分析处理和数据挖掘是作为三种独立的信息处理技术出现的。数据仓库用 于数据的存储和组织,()LAP集中于数据的分析，数据挖掘则致力于知识的发现。由于这三 种技术内在的联系性和互补性,将它们结合起来是一种新的DSS构架，是DSS有效而可操作 的整体解决方案。

例**8**什么是工程数据库？ 解析 工程数据库是一种能存储和管理各种工程设计图形和工程设计文档,并能为工程设计提供

各种服务的数据库。

**70**数据库系统概论(第**5**版)同步辅导及习题全解

主要应用于CAD/CAM,CIM,CASE等工程应用领域。

工程数据库中，由于传统的数据模型难以满足工程应用的要求，需要运用新的模型技术，如 扩展的关系模型、语义模型、面向对象的数据模型。

工程数据库管理系统的功能与传统数据库管理系统有很大不同。

例**9** 什么是统计数据库？

解析统计数据库是一种用来对统计数据进行存储、统计、分析的数据库系统。

统计数据具有层次型特点,但并不完全是层次型结构。统计数据也有关系型特点，但关系型

也不完全满足需要。统计数据具有一些特殊的性质,例如：

（1） 分类属性和统汁属性；

（2） 多维性；

（3） 分类属性的层次结构；

（4） 微数据和宏数据之分。

统计数据库中常用的操作有：抽样、邻近搜索、估计与插值、转置、聚集及复杂的分析操作。

这些操作不同于关系数据库中传统的查询、增加、删除、修改操作。人们希望能从DBMS — 级来支持以上的数据特性和操作，因此，研究和发展了统计数据库技术。

统计数据库在安全性方面有特殊的要求,要防止某些用户在统计数据库中利用对统计数据 （如综合数据）的合法查询推导出该用户无权了解的某一个体的具体数据。

课后习题解答

1. ［解题陸，（1）数据库定义功能；

（2） 数据存取功能；

（3） 数据库运行管理；

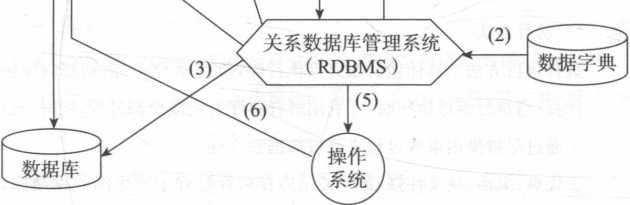
（4） 数据库的建立和维护功能。

1. 解题国，DBMS是一个复杂而有序的整体,下图是DBMS的运行过程示例。结合图12. 1,给

出DBMS插入一条记录的活动过程。

|  |  |
| --- | --- |
| 7 （8） | 应用程序 |
|  | 用户工作区  （UWA） |

系统缓冲区

图12.1数据库管理系统的运行过程

(7)\、 (1)

（1） 用户通过应用程序向DBMS（以RDBMS为例）发出调用数据库数据的INSERT 命令在命令中给出一个关系名和所插入的元组值。

（2） DBMS先对命令进行语法检查、语义检查和用户存取权限检查。语义检査的具 体做法是，DBMS读取数据字典,检査是否存在该关系及相应的字段，值的数据类型 是否正确。检查该用户是否具有该关系上执行INSERT操作的权限。若未能通过 检查就拒绝执行INSERT命令,返回有关的错误提示信息。

（3） DBMS查看存储模式，找到新记录所应插入的位置和相应的页面P。

（4） DBMS在系统缓冲区中找到一个空页。

⑸DBMS根据步骤⑶的执行结果，向操作系统发出读取物理页面P的命令。

（6） 操作系统执行读操作。将数据页P读入系统缓冲区中的空白页处。

（7） DBMS根据插入命令和数据字典的内容将数据转化成内部记录的格式。

（8） DBMS将数据记录写入系统缓冲区的页面P中。

（9） DBMS将执行事务进行提交。把状态信息（如成功或不成功的指示）、例外状态 信息等返回给应用程序。（这里未考虑多用户并发控制的问题。）

语言处理层首先要对DDL语句进行语法检查、语义检查和用户权限检查。语义检 查的具体做法是,DBMS读取数据字典，检査是否存在与该语句中的表、视图或索引 等要创建的对象相同的对象名，检查该用户是否具有创建数据库对象的权限。

然后，把DDL语句翻译成内部表示，将其存储在系统的数据字典中。例如建立一个 新表，就要把关系名、建立者、属性个数、记录长度等信息记入数据字典中。

DML语句处理的过程包括：

（1）先对DML语句进行词法分析和语法分析，并把外部关系名、属性名转换为内部

171

第十二章

数据库管理系统

名，通过词法和语法分析后生成语法分析树；

72

数据库系统概论(第**5**版)同步辅导及习题全解

**6.**

（2） 根据数据字典中的内容进行查询检查,包括审核用户的存取权限和完整性检查；

（3） 对查询进行优化,包括代数优化和存取路径优化；

（4） 把选中的查询执行方案描述出来。

（1） 解释方法

解释执行方法的具体做法是直到执行前,数据库DML语句都以原始字符串的形式 保存;当执行到该语句时，才利用解释程序去完成全部过程，同时予以执行。这种方 法通过尽量推迟束缚过程来赢得数据独立性。

1. 优点:灵活、应变性强，甚至能适应在解释过程中发生的数据结构、存储结构等的 变化，因此能保持较高的数据独立性。
2. 缺点:每次执行一个DML语句时都要执行所有步骤，开销很大，因此效率比 较低。

（2） 预编译方法

预编译方法的基本思想是:在用户提交DML语句之后,在运行之前对它进行翻译 处理,保存产生好的可执行代码。当需要运行时，取出保存的可执行代码加以执行。

1. 优点：自动重编译技术使得预编译方法既拥有了编译时进行束缚所带来的高效 率,又具备了执行时束缚带来的数据独立性。
2. 缺点:在束缚过程中进行优化所依据的条件可能在运行前已不存在，或者数据库 结构已被修改，因而导致已作出的应用规划在执行时不再有效。

数据库存取层主要包括以下几个子系统：

（1） 记录存取、事务管理子系统。

记录存取子系统提供按某个属性值直接取一个元组和顺序取一个元组的存取原语, 事务管理子系统提供定义和控制事务的操作。

（2） 封锁子系统，执行并发控制。

（3） 恢复子系统。

主要是指日志登记子系统对事物执行开始、回滚、提交;对元组进行插入、删除、修 改;对索引记录进行插入、删除、修改等等,每一个操作作为一个日志记录存入日志 文件中。对不同的故障恢复策略执行相学的恢复。

（4） 控制信息管理模块。

该模块利用专门的数据区（内存中）登记不同记录类型以及不同存取路径的说明信 息（取自数据字典）和控制信息。该信息是存取元组和管理事务的依据。它和事务管理、记录存取子系统一起保证事务的正常运行。该模块提供对数据字典中说明信 息的读取、增加、删除和修改操作。

（5） 存取路径维护子系统。

该模块在对数据进行修改為作时要对该表上已建立的所有索引进行动态维护。

73

T—

（6） 排序/合并子系统。

第十二章

该模块主要功能包括输出语序结果.删去重复值降低开销，先对两个关系按连接属 性值排序再进行连接的连接操作以提高连接运算速度,建立索引结构减少数据块的 存取次数等。

数据库管理系统

1. 匯厘画k数据库是共享资源，通常有许多个事务同时在运行。

当多个事务并发地存取数据库时就会产生同时读取和/或修改同一数据的情况。若 对并发操作不加控制就可能会存取和存储不正确的数据，破坏数据库的一致性。所 以数据库管理系统必须提供并发控制机制。

数据库的并发控制就是为了控制数据库,防止多用户并发使用数据库时造成数据错 误和程序运行错误,保证数据的完整性。

1. 瓯塑k DBMS封锁技术比操作系统封锁内容更加丰富,技术更加复杂，同时其实现手段依

赖于操作系统提供的环境。（1）操作系统封锁对象（即系统资源）单一，封锁对象状 态确定，封锁力度不能改变，排他锁类型单一。（2）DBMS封锁对象多样，包括用户 数据、索引、数据字典等，封锁对象动态改变，封锁力度可变，封锁类型多样。

两者的具体区别如下表所示：

表操作系统和数据库管理系统封锁技术的比较

|  |  |  |
| --- | --- | --- |
|  | 操作系统 | 数据库管理系统 |
| 封锁对象 | 单一，系统资源（包括  CPU、设备、表格等） | 多样.数据库中各种数据对象（包括用户数据、  索引（存取路径）、数据字典等） |
| 封锁对象的状态 | 静态，确定，各种封锁对象 在封锁表中占有一项。封 锁对象数是不变的 | 动态，不确定。封锁对象动态改变着、常常在 执行前不能确定。一个封锁对象只有当封锁 时才在封锁表中占据一项 |
| 封锁的粒度 | 不变，由于封锁对象单一、 固定，封锁粒度不会改变 | 可变,封锁可加到或大或小的数据单位上，封  锁粒度可以是整个数据库、记录或字段 |
| 封锁的类型 | 单一、排他锁 | 多样，一般有共享锁（S Lock）,排他锁（X  Lock）或其他类型的封锁，随系统而异 |

1. 瓯塑k设立系统缓冲区的原因有以下两点：（1）隔离外存设备与存储层以上各系统，保证

**74**数据库系统概论(第**5**版)同步辅导及习题全解

DBMS具有设备独立性。（2）提高存取效率。利用缓冲区滞留数据，只有数据不在 缓冲区时才从外存读入页面，写入数据先在页面作标记，当事务结束时或缓冲区满 需调入新页时才写入外存。

1. 匯厘gk数据库中存储四个方面的数据：

1） 数据描述，即数据的外模式、模式、内模式；

2） 数据本身；

3） 数据之间的联系；

4） 存取路径。

这四个方面的数据内容都要采用一定的方式组织、存储起来。

1. ［解题^^略。
2. 匯回陲，略。

**13** .［廨 略。

175

第十三章

数据库技术发展概述

知识网络图

第十三章

数据库技术发展概述

1. 支持三级模式的体系结构

第一代数据库系统＜

2.用存取路径来表示数据之间的联系

3.独立的数据定义语言

4.导航的数据操纵语言

1.奠定了关系模型的理论基础

数据库发展的三个阶段＜

第二代数据库系统•

2.研究了关系数据语言

3.研制了 RDBMS原型

支持数据管理

数据库

第三代数据库系统-

保持和继承

展概述

数据模型的发展-

1. XML数据模型

开放性

1.面向对象数据模型

1. RDF数据模型

数据库系统发展的特点＜

数据库技术与相关技术相结合

1.分布式数据库系统

2.并行数据库系统

面向应用领域的数据库新技术

1 .工程数据库

2.空间数据库

**76**数据库系统概论(第**5**版)同步辅导及习题全解

知识点归纳

数据库系统已经发展成为一个大家族。本章以数据模型、新技术内容、应用领域为三条主线, 需要了解数据库技术发展的三个阶段，数据库技术的发展及其相互关系，达到纲举目张。

需要掌握数据库技术的核心是数据管理。新应用领域不断涌现，数据对象趋于多样化,数据库 工作者应该不断拓宽数据库的研究领域,在众多新领域中承担起其中的数据管理研究开发任务。

数据库工作者应该具有这样的意识，哪里有数据，哪里就有数据管理的任务，哪里就有数据库 研究和开发的课题。面对新的挑战，了解要继承的数据库技术和其他技术相结合的优良传统，努力 探索新的途径、新的方法、新的技术,提高和改善对数据和信息的使用。

熟悉应用领域的变化，熟悉工程数据库和空间数据库的主要内容，了解数据管理技术的发展 趋势。

■ \_、数据库技术发展概述

数据库技术是计算机科学技术中发展最快的领域之一,也是应用最广的技术之一。目前，它已 成为计算机信息系统与应用系统的核心技术和重要基础。

当今数据库系统是一个大家族，数据模型丰富多样，新技术内容层出不穷,应用领域日益广泛, 当读者步入数据库领域时，面对众多复杂的数据库系统难免产生迷惑和混乱。

.二、数据库发展的**3**个阶段

（一）第一代数据库系统

1. 支持三级模式的体系结构

三级模式通常指外模式、模式、内模式。模式之间具有转换功能。

1. 用存取路径来表示数据之间的联系

这是数据库系统和文件系统的主要区别之一。数据康不仅存储数据，而且存储数据之间的联 系。数据之间的联系在层次和网状数据库系统中都是用存取路径来表示和实现的。

1. 独立的数据定义语言

层次数据库系统和网状数据库系统有独立的数据定义语言，用以描述数据库的外模式、模式、内模式以及相互映像。诸模式一经定义,就很难修改。

1. 导航的数据操纵语言

层次和网状数据库的数据查询和数据愕纵语言是一次一个记录的导航式过程化语言。

（二） 第二代数据库系统

77

T—

1. 关系数据库是以关系模型为基础的。
2. 关系操作的能力可用关系代数（或等价的关系演算）中并、交、差、广义笛卡尔积、选择、投影、 连接、除等操作。

第十三章

1. 数据完整性包括实体完整性、参照完整性以及与应用有关的完整性。实体完整性和参照完 整性是关系模型必须满足的完整性约束条件。

数据库技术发展概述

（三） 新一代数据库系统

1. 第三代数据库系统应支持数据管理、对象管理和知识管理。
2. 第三代数据库系统必须保持或继承第二代数据库系统的技术。
3. 第三代数据库系统必须对其他系统开放。

■三、数据库系统发展的特点

1. 数据模型的发展
2. 数据库技术与其他相关技术相结合

•数据库技术与分布处理技术相结合，出现了分布式数据库系统；

•数据库技术与并行处理技术相结合，出现了并行数据库系统；

•数据库技术与人工智能技术相结合，出现了知识库系统和主动数据库系统；

-数据库技术与多媒体技术相结合,出现了多媒体数据库系统；

•数据库技术与模糊技术相结合,出现了模糊数据库系统等。

■四、数据库技术发展趋势

1 .信息特性和来源的变化

数据库领域已经从获取、组织、存取、分析和恢复结构化数据扩展到文本、时间、空间、声音、图 形图像、视频等多媒体数据,HTML、XML等非结构化和半结构化数据，还有程序数据、流数据和队 列数据等。

1. 应用领域的变化

■五、当前若干研究热点

1. 信息集成

**78**数据库系统概论(第**5**版)同步辅导及习题全解

1. 移动数据管理
2. 网格数据管理
3. 传感器数据库技术
4. DBMS的自适应管理

例**1**第三代数据库管理系统的基本特征

典型例题

解析1.第三代数据库系统应支持数据管理、对象管理和知识管理。

1. 第三代数据库系统必须保持或继承第二代数据库系统的技术。
2. 第三代数据库系统必须对其他系统开放。

例**2**工程数据库的功能

解析1.支持复杂对象（如图形数据、工程设计文档）的表示和处理；

1. 可扩展的数据类型；
2. 支持复杂多样的工程数据的存储和集成管理；
3. 支持变长结构数据实体的处理；
4. 支持工程事务和嵌套事务的并发控制和恢复；
5. 支持设计过程中多个不同数据版本的存储和管理；
6. 支持模式的动态修改和扩展；
7. 支持多种工程应用程序等。

例**3**空间数据库研究的主要内容

解析1.空间数据模型

空间数据模型是描述空间实体和空间实体关系的数据模型，一般来说可以用传统的数据模 型加以扩充和修改来实现，也可以用面向对象的数据模型来实现。

1. 空间数据査询

空间数据查询包括位置查询、空间关系查询和属性查询。

1. 空间数据库系统

空间数据库管理系统的主要功能是，提供对空间数据和空间关系的定义和描述，提供空间数据查询语言,实现对空间数据的高效査询和操作，提供对空间数据的存储和组织，提供对空 间数据的直观显示等。

1. 查询语言

大多是以SQL语言为基础，增加相应的函数实现对空间对象和空间关系的查询。

79

1—

例**4** NoSQL技术的特点。

第十三章

解析（1）对数据进行划分，通过大量节点的并行处理获得高性能,釆用的是横向扩展的方式。

（2） 放松对数据的ACID一致性约束，允许数据暂时出现不一致情况，接受最终一致性。

（3） 对各个数据分区进行备份,应对节点可能的失败，提高系统可用性。

数据库技术发展概述

课后习题解答

**1**.匯函匹（1）数据模型是数据库系统的核心和基础。数据库技术的三个发展阶段应该按照数 据模型的进展来界定。按照数据模型的进展，数据库技术可以相应地分为三个发展 阶段。

（2）数据模型的发展经历了格式化数据模型（包括层次数据模型和网状数据模型）、 关系数据模型两个阶段,发展到以面向对象数据模型为代表的非传统数据模型的 阶段。

层次数据库系统和网状数据库系统的数据模型虽然分别为层次模型和网状模型，但 实质上层次模型是网状模型的特例。它们都是格式化模型。它们从体系结构、数据 库语言到数据存储管理均具有共同特征，是第一代数据库系统。

关系数据库系统支持关系模型。关系模型不仅简单、清晰，而且有关系代数作为语 言模型，有关系数据理论作为理论基础。因此，关系数据库系统具有形式基础好、数 据独立性强、数据库语言非过程化等特色，标志着数据库技术发展到了第二代。

第二代数据库系统的数据模型虽然描述了现实世界数据的结构和一些重要的相互 联系.但是仍不能捕捉和表达数据对象所具有的丰富而重要的语义，因此尚只能属 于语法模型。

第三代的数据库系统将以更加丰富的数据模型和更强大的数据管理功能为特征，从 而满足传统数据库系统难以支持的新的应用要求。

新一代数据库技术的特点是：

（1）面向对象的方法和技术对数据库发展的影响最为深远

数据库研究人员借鉴和吸收了面向对象的方法和技术，提出了面向对象数据模型 （简称对象模型）。该模型克服了传统数据模型的局限性，促进了数据库技术在一个

新的技术基础上继续发展。

80

1—

数据库系统概论(第**5**版)同步辅导及习题全解

（2） 数据库技术与多学科技术的有机结合

计算机领域中其他新兴技术的发展对数据库技术产生了重大影响。传统的数据库 技术和其他计算机技术，如网络通信技术、人工智能技术、面向对象程序设计技术、 并行计算技术、移动计算技术等的互相结合、互相渗透,使数据库中新的技术内容层 出不穷。

（3） 面向应用领域的数据库技术的研究

在传统数据库系统基础上，结合各个应用领域的特点，研究适合该应用领域的数据 库技术，如数据仓库、工程数据库、统计数据库、科学数据库、空间数据库、地理数据 库等，这是当前数据库技术发展的又一重要特征。

1. 匯壓函k （1）数据模型是数据库系统的核心和基础。

（2）数据库的发展集中表现在数据模型的发展。

数据库技术与其他学科的内容相结合，是新一代数据库技术的一个显著特征，涌现

**3.**

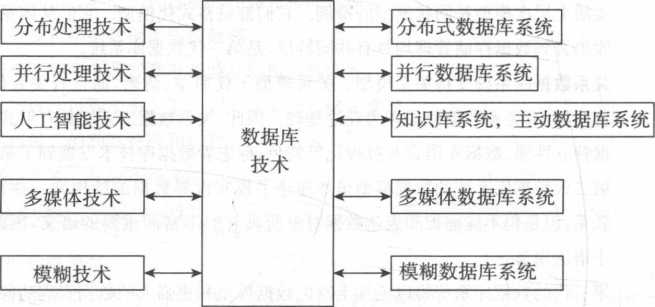
出各种新型的数据库系统（如图所示）。例如：

（1） 数据库技术与分布处理技术结合，出现了分布式数据库系统；

（2） 数据库技术与并行处理技术相结合，出现了并行数据库系统；

（3） 数据库技术与人工智能技术相结合，出现了知识库系统和主动数据库系统;

（4） 数据库技术与多媒体技术相结合,出现了多媒体数据库系统；

（5） 数据库技术与模糊技术相结合，出现了模糊数据库系统等等。



第十四章

大数据管理

第十四章

大数据管理

知识网络图

大数据定义

大数据概述-

大数据的特征＜

'巨量 多样 快变 、价值

互联网文本大数据管理与挖掘

大数据V  
管理＜

大数据的应用＜

'基于大数据分析的用户建模＜

NoSQL数据管理系统＜

实事探针

挑战

大数据管理系统

1. 面向用户建模的大数据系统架构
2. 用户建模的基础工具
3. 用户建模的价值体现

Key-Value 模型

BigTable 模型

文档模型

图模型

大数据管理系统〈

NewSQL数据库

MapReduce 技术

1. 面向操作型应用的关系数据库技术
2. 面向分析型应用的关系数据库技术

大数据管理系统的新格局＜

1. 面向操作型应用的NoSQL技术

4面向分析型应用的MapReduce技术

重难点

通过本章的学习，掌握大数据的定义和四个基本特征，熟悉大数据应用的例子一实事探针系 统。通过应用案例来了解大数据应用的特点，以及对大数据管理和大数据系统提出的技术需求。 简单了解NoSQL系统、NewSQL系统和MapReduce技术等，知道这些是在大数据管理和分析处理 领域涌现的代表性前沿技术和系统。

2

8

1—

数据库系统概论(第**5**版)同步辅导及习题全解

■ 一、概述

知识点归纳

1. 分布式数据库系统

分布式数据库由一组数据组成，这些数据物理上分布在计算机网络的不同结点(亦称场地)上, 逻辑上是属于同一个系统。

1. 分布式数据库系统的特点
2. 数据独立性

数据独立性是数据库技术追求的主要目标之一，在集中式数据库系统中，数据独立性包括两个 方面:数据的逻辑独立性与数据的物理独立性，其含义是用户程序与数据的全局逻辑结构及数据的 存储结构无关。

1. 集中与自治相结合的控制结构
2. 局部共享。即在局部数据库中存储局部场地上各用户的共享数据，这些数据是本场地用户 常用的。
3. 全局共享。即在分布式数据库系统的各个场地也存储供其他场地的用户共享的数据，支持 系统的全局共享。
4. 适当增加数据冗余度

在集中式数据库系统中,尽量减少冗余度是系统目标之一。而在分布式数据库中却要适当增 加数据冗余度。

1. 全局的一致性、可串行性和可恢复性

■二、分布式数据库系统的体系结构

分布式数据库系统的模式结构 、

3

8

1—

1. 全局外模式，它是全局应用的用户视图，是全局概念模式的子集。
2. 全局概念模式，它定义分布式数据库中数据的整体逻辑结构，使得数据如同没有分布一样。 全局概念模式中所用的数据模型应该易于向其他模式映像，全局槪念模式包括一组全局关系的 定义。

第十四章

1. 分片模式，每「个全局关系可以分为若干不相交的部分，每一部分称为一个片段，分片模式 定义片段以及全局关系到片段的映像。这种映像是一对多的，一个全局关系可对应多个片段，而一 个片段只来自一个全局关系。

大数据管理

分布模式片段是全周关系的逻辑部分，一个片段在物理上可以分配到网络的不同结点上。

.三、分布式数据库管理系统

D-DBMS由四部分组成：

1. LDBMS(Local DBMS)：局部场地上的数据库管理系统,其功能是建立和管理局部数据库, 提供场地自治能力，执行周部应用及全周查询的子查询。
2. GDBMS(Global DBMS):全局数据库管理系统，主要功能是提供分布透明性,协调全M事 务的执行，协调各局部DBMS以完成全M应用，保证数据库的全局一致性，执行并发控制，实现更 新同步，提供全局恢复功能等。
3. 全局数据字典:存放全局概念模式、分片模式、分布模式的定义以及各模式之间映像的定 义，存放有关用户存取权限的定义。以保证用户的合法权限和数据库的安全性，存放数据完整性约 朿条件的定义，其功能与集中式数据库的数据字典类似。
4. 通信管理:通信管理系统在分布式数据库各场地之间传送消息和数据，完成通信功能。

.四、查询处理和优化

1.查询分解

对于全局査询必须把它们分为若干子查间，每个査询只涉及某一结点的数据，可以由M部

DBMS处理。在分布式数据库中一个关系是分为若干逻辑片段，这些片段又对应在系统的多个结

点上存放。所以，对一个查询中所涉及的关系需要确定一个物理片段,选择不同的物理片段执行查 询操作会直接影响査询执行的效率。

**84**数据库系统概论(第**5**版)同步辅导及习题全解

1. 选择操作执行的次序

主要是确定连接和并操作的次序，其他的操作顺序是难确定的。

1. 选择执行操作的方法

这包括将若干操作组合在对数据库的一次存取中执行;选择可用的存取路径(如索引)以及选 择某一种算法等问题。连接(Join)是査询中最费时的操作，因此连接的执行方法是研究的重点。

.五、分布事务管理

一个事务的执行必须保持其原子性，即它所包含的所存更新操作要么全部都做,要么都不做。

1 .分布事务的恢复

数据库在运行过程中出现故障和错误总是难免的。不同的故障会造成数据库本身同程度的损 害，轻则事务不能正常运行，重则造成数据库数据不一致性。

1. 并发控制

集中式数据库系统中并发控制一般采用封锁技术。遵守两段锁协议，即如果一个事务已释放 了一个锁，那么它就不能再请求新的封锁。

为了减少系统开销,处理多副本的封锁可釆取如下几种方法：

1. 对写操作,要申请对所有副本的X锁。对于读操作，只要申请对某个副本的S锁。
2. 无论是写操作还是读操作都要对多数(大于半数)副本申请X锁或S锁。
3. 规定某个场地上的副本为主副本,所有的读写操作均申请对主副本的封锁。

这3个方法均可有效地发现冲突，协调并发事务的执行。

典型例题

例**1** 基于大数据分析的用户分析分为哪几类？

解析一大类用户建模方法采用的是批处理方式的离线分析方法,对结构化或半结构化的历史日 志数据进行SQL分析或者使用数据挖掘和机器学习的深度分析方法。其特点是釆用离线的 方式处理超大规模的历史数据，当数据量很大时,一些任务可能运行数个小时，甚至几天。

另一大类用户建模方法则采用实时的在线分析方法，数据即来即分析,更强调数据的实时分析处理能力。这类方法适合于捕捉一些时效性强的用户属性。这些用户属性被用来描述用 户最新的特征，是在线信息推荐算法的重要依据，其价值通常也是最高的。

例**2** NoSQL系统支持的数据模型分为哪几个类型？

解析NoSQL系统支持的数据模型通常分为Key-Value模型、Big Table模型、文档模型和图模型。

1. Key-Value模型是非常简单而容易使用的数据模型。

第十四章

1. BigTable模型能够支持结构化的数据，包括列、列簇、时间戳以及版本控制等元数据的 存储。

大数据管理

1. 文档模型，该模型在存储方面有以下改进:Value值支持复杂的结构定义，通常是被转换 成JSON或者类似于JSON格式的结构化文档;支持数据库索引的定义，其索引主要是按照 字段名来组织的。
2. 图模型，该模型支持图结构的各种基本算法。可以直观地表达和展示数据之间的联系。

例**3** 指出MapReduce技术的不足？

解析(1)基于MapReduce的应用软件较少，许多数据分析功能需要用户自行开发,从而导致使用 成本增加。

1. 原来由数据库管理系统完成的工作都转移给了程序员，导致程序员负担过重，程序与数 据缺乏独立性。
2. 在同等硬件条件下,MapReduce的性能远低于并行数据库。
3. 在数据分析领域，连接是关键操作，但MapReduce处理连接的性能尤其不尽如人意。

课后习题解答

**1**.匯厘gk 一般意义上,大数据是指无法在可容忍的时间内用现有IT技术和软硬件工具对其 进行感知、获取、管理、处理和服务的数据集合。通常被认为是PB或EB或更高数 量级的数据,包括结构化的、半结构化的和非结构化的数据。

基本特征:巨量、多样、快变、价值。

巨量:大数据的首要特征是数据量巨大，而且在持续、急剧地膨胀O

多样:大数据的多样性通常是指异构的数据类型、不同的数据表示和语义解释。

快变:大数据的快变性也称为实时性，一方面指数据到达的速度很快,另一方面指能 够进行处理的时间很短，或者要求响应速度很快,即实时响应。

价值:大数据的价值是潜在的、巨大的。大数据不仅具有经济价值和产业价值，还具 有科学价值。

**86**数据库系统概论(第**5**版)同步辅导及习题全解

**2**」解題亜，关系数据库在大数据时代丧失了互联网搜索这个机会,其主要原因是关系数据库管 理系统(并行数据库)的扩展性遇到了前所未有的障碍，不能胜任大数据分析的需 求，关系数据管理模型追求的是高度的一致性和正确性,面向超大数据的分析需求。

1. 解题距k 传统RDBMS的一致性和ACID特性在非关系数据管理系统中得到继承和发展。
2. 画陸k (DNoSQL是以互联网大数据应用为背景发展起来的分布式数据管理系统，它有两

种解释:一种是Non-Relational,即非关系数据库；另一种是Not Only SQL,即数据 管理技术不仅仅是SQL。NoSQL系统支持的数据模型通常分为:Key-Value模型、

BigTable模型、文档模型和图模型4种模型o (2)NoSQL系统为了提高存储能力和 并发读写能力采用了极其简单的数据模型，支持简单的查询操作，而将复杂操作留 给应用层实现。该系统对数据进行划分，对各个数据分区进行备份，以应对结点可 能的失败，提高系统可用性;通过大量结点的并行处理获得高性能，采用的是横向扩 展的方式。

1. 胸题置图，(DNewSQL系统是融合了 NoSQL系统和传统数据库事务管理功能的新型数据库

系统。NewSQL将SQL和NoSQL的优势结合起来，充分利用计算机硬件的新技 术、新结构，研究与开发了若干创新的实现技术。

1. MapReduce技术主要应用于大规模廉价集群上的大数据并行处理，是以key/ value的分布式存储系统为基础，通过元数据集中存储、数据以chunk为单位分布存 储和数据chunk冗余复制来保证其高可用性。
2. 优势:MapReduce是一种并行编程模型。其处理模式以离线式批量处理为主。
3. MapReduce存在如下不足：
4. 基于MapReduce的应用软件较少，许多数据分析功能需要用户自行开发,从而导 致使用成本增加；
5. 程序与数据缺乏独立性；
6. 在同等硬件条件下,MapReduce的性能远低于并行数据库；
7. MapReduce处理连接的性能尤其不尽想人意。

第十五章

内存数据库系统

知识网络图、

1内存数据库的雏形期

发展历程＜

内存数据库的研究发展期

内存数据库的产品成长期

1.高吞吐率和低访问延迟

内存数据

库系统＜

内存数据库的特性＜

2.并行处理能力

3.硬件相关性

数据存储

1.面向cache特性的查询处理与优化技术

内存数据库的关键技术＜

查询处理及优化-

2.索引技术

3 .面向多核的查询处理技术

1. 面向众核的査询处理技术

'并发与恢复

并发控制

恢复机制

需要了解内存数据库和磁盘数据库的具体区别和内存数据库的特点，重点掌握内存数据库的 定义，内存数据库和硬件相关性。需要了解哪些硬件技术影响内存数据库的性能，以及内存数据库 的性能受硬件特性的直接影响。

计算机硬件技术的发展主要体现在高端计算设备和存储设备上，内存OLTP数据库软件技术 比较成熟，已用于金融、电信等实时响应性能要求较高的应用领域，成为企业核心业务事务处理的 解决方案。

**88**数据库系统概论(第**5**版)同步辅导及习题全解

传统数据库厂商OracleJBM和Microsoft等分别推出了内存数据库产品和解决方案。

■ 一、概述

知识点归纳

1. 内存数据库是将内存作为主存储设备的数据库系统
2. 内存数据库与磁盘数据库的区别

.二、内存数据库的发展历程

1. 内存数据库的雏形期
2. 内存数据库的研究发展期
3. 内存数据库的产品成长期

.三、内存数据库的特性

1. 高吞吐率和低访问延迟

数据库的查询处理性能主要取决于数据的存储访问性能。

1. 并行处理能力

内存具有良好的并行数据访问能力和随机访问性能，因此内存数据库的查询处理技术带有天 然的并行性，并且可以充分利用随机访问能力提高查询的数据访问效率和CPU指令效率。

1. 硬件相关性 \

内存数据库的性能受硬件特性的直接影响。计算机硬件技术的发展主要体现在高端计算设备 和存储设备上。

■四、内存数据库的关键技术

1. 数据存储 \

9

8

1—

内存数据库的数据存储一般有行存储模型、列存储模型和混合模型。

1. 查询处理及优化

第十五章

内存数据访问性能

内存数据处理性能

（1） 面向cache特性的查询处理与优化技术

内存数据库系统

（2） 索引技术

（3） 面向多核的查询处理技术

（4） 面向众核的查询处理技术

1. 并发与恢复

（1） 并发控制

（2） 恢复机制

内存数据库是产品化的数据库管理软件，极大缩短了开发周期；

内存数据库有着开发的平台和接口，程序开发和移植更加灵活便捷，也便于后期维护和二次 开发；

可以通过使用统一的SQL语言方便地查询内存中的数据；

能在数据库中保障数据的安全性和完整性。

这些优势，对于快速部署和简化维护都是非常有利的。

劣势是内存数据库无法单独使用，它只能提高速率，并不能代替磁盘数据库在文件系统中的 作用。

典型例题

例**1**在数据访问方面的cache优化技术主要包括哪几类？

解析（1） cache-conscious优化技术

cache-conscious优化技术以hash连接优化为代表。在hash连接中，内层的hash表大小决定 了 hash探测过程中能否尽可能多地从高速cache中访问hash表。为提高hash探测时的 cache命中率，需要将hash表划分为小于cache容量的较小的分区，从而使hash探测时的

cache命中率提高。

90

1—

数据库系统概论(第**5**版)同步辅导及习题全解

1. cache-oblivious 优化技术

cache-oblivious查询优化技术需要根据cache层数、各层cache大小＞cache line长度、TLB条 目等硬件参数来优化算法实现，对硬件平台的特性依赖性较高。

1. page-coloring 优化技术

page-coloring是一种内存虚拟地址向cache地址映射的机制。现代多核处理器通常采用多 路组关cache。在这种地址映射机制下,物理地址被划分为以page大小为单位的地址段，称 为pagecoloro应用程序使用的虚拟内存地址被操作系统转换为物理内存地址，然后按照物 理地址中的pagecolor位将每个page映射到cache中指定的区域，通过这种机制实现在物理 地址与cache地址之间的快速转换。

例**2**当前内存数据库主要采用的三种多核并行hash连接技术有哪些？

解析(1)无分区hash连接算法

1. 基于分区的hash连接算法
2. radix hash连接算法

例**3**内存数据库的查询处理性能主要由哪些因素决定？

解析内存数据访问性能和内存数据处理性能。

内存数据访问性能由内存带宽和内存访问延迟决定。相对于CPU,内存数据访问性能的增 长速度与CPU性能增长速度之间的差距越来越大，内存访问的巨大延迟是内存数据库的性 能瓶颈。

内存数据库的查询处理性能主要受处理器性能影响。CPU的发展已经进入多核时代，不再 单一依靠CPU主频的提高,更多的处理核心提高了多核CPU的并行计算能力，因此内存数 据库的查询优化技术也进入多核并行时代，需要将内存数据库的查询处理技术全面升级为 多核CPU并行查询处理技术，并根据多核CPU的硬件特性进行算法优化，提高内存数据库 整体性能。

课后习题解答

**1.** 在内存数据库中*，*内存作为常规的数据存储设备，是数据的永久存储及后备存储设

备。内存数据库的数据组织、存储访问模型和查询处理模型都针对内存特性进行了 优化设计，内存数据可被处理器直接访问。

第十五章

在磁盘数据库中，磁盘是常规的数据存储设备，磁盘阵列或磁带机是数据的后备存 储设备,内存作为磁盘数据库的缓存使用。磁盘数据库的数据组织、存储访问模型 及处理模型都是面向磁盘访问特性而设计的，磁盘数据通过缓冲区被处理器间接访 问.查询优化的核心是减少磁盘的输入/输出。

内存数据库与磁盘数据库的区别如图15. 1所示。



处理器

物理内存

，内存数据库，  
I孟而素号I ]

内存数据库

后备存储

图15.1内存数据库和磁盘数据库对比示意图

.匯厘圖区（1）高吞吐率和低访问延迟

数据库的查询处理性能主要取决于数据的存储访问性能。

（2） 并行处理能力

内存具有良好的并行数据访问能力和随机访问性能，因此内存数据库的査询处理技 术带有天然的并行性，并且可以充分利用随机访问能力提高查询的数据访问效率和 CPU指令效率。

（3） 硬件相关性

内存数据库的性能受硬件特性的直接影响。

，.厕画区（1）内存数据库和硬件的相关性：内存数据库的性能受硬件特性的直接影响。计算

机硬件技术的发展主要体现在高端计算设备和存储设备上，如多核处理器、众核协

处理器(Many Integrated Core,MIC)、通用 GPU、PCM 存储(Phase Change Memo- ry，相变存储)、固态硬盘(Solid State Disk, SSD)存储等。这些计算能力和存储性能 的提升有助于内存吞吐率需求的提升(众核技术)、提高内存持久存储能力(PCM技 术)或为内存提供二级存储(SSD技术)。硬件技术在多核及众核处理器、高性能存 储和高速网络等方面的发展为内存数据库提供了高并行处理、高性能存储访问以及 高速连通的硬件平台。内存数据库的设计应充分考虑并有效利用由新硬件技术带 来的功能扩展和性能提高。(2)大容量内存、flash、PCM存储、多核CPU、众核处理 器、高性能网络传输等硬件技术的发展为内存数据库提供了良好的平台，直接影响 到内存数据库的性能。

2

9

<1

数据库系统概论(第**5**版)同步辅导及习题全解

大数据的特点有:数据最大(Volume).类型繁多(Variety).价值密度低(Value).速 度快时效高(Velocity)o随着大数据时代的到来，未来众核协处理器、通用计算图形 处理器(General Purpose Graphic Process Unit,GPGPU)等新的高性能计算平台进 入数据库领域，同时也对内存数据库提出了更多挑战。

1. 查询处理与优化，主要针对大数据数据量大和类型繁多的特点。pag^coloring 优化技术对于数据持久驻留内存的内存数据库来说，较大的局部性数据集往往需要 预先分配较大的内存地址范围，而较少的page color■对应的地址范围较小,难以满 足大数据集存储的要求。
2. 实时分析处理性能，主要针对大数据速度快时效高的特点。内存数据库摆脱了 I/O延迟之后，内存访问速度得到了极大的提升，在新兴的非易失性内存，如PCM 等技术支持下，内存计算和更新的速度进一步提升。事务型内存数据库的一个技术 发展趋势是将事务串行化，简化并发控制机制，提高内存数据库代码执行效率，使串 行处理性能能够满足高吞吐性能需求。分析型内存数据库则将计算最大化并行，以 提高多核处理器的并行计算效率,提高应对内存大数据实时分析处理的性能需求。
3. 并发与恢复技术,主要针对大数据数据量大和价值密度低的特点。通过对大数 据表的共享扫描减少并发查询时独立大表扫描所产生的cache缺失，主要通过査询 分组以及查询操作符批处理技术实现共享扫描基础上的高并发查询处理。

193

第十六章

数据仓库与联机分析处理技术

知识网络图

，数据仓库是面向主题的

基本特征＜

数据仓库是集成的

数据仓库是不可更新的

数据仓库与联机

分析处理技术

数据仓库技术＜

数据仓库的数据是随时间不断变化的

数据仓库中的数据组织＜

联机分析处理技术＜

数据挖掘技术＜

早期细节级

当前细节级

轻度综合级

高度综合级

多维数据模型

多维分析操作

联机分析处理的实现方式

概念

与传统分析方法的区别

数据挖掘的数据源

I数据挖掘的功能

系统需求的变化

传统数据仓库面临的问题

I大数据时代的新型数据仓库

第干六章

数据仓库与联机分析处理技术

94

1—

数据库系统概论(第**5**版)同步辅导及习题全解

知识点归纳

数据仓库、联机分析处理和数据挖掘是作为三种独立的信息处理技术出现的。需要了解数据 仓库用于数据的存储和组织，联机分析处理集中于数据的分析，数据挖掘则致力于知识的自动发 现,它们都可以分别应用到信息系统的设计和实现中，以提高相应部分的处理能力。需要掌握这三 种技术内在的联系性和互补性，将它们结合起来就成为一种新的决策支持系统构架,成为商务智能 的3个支柱。

数据挖掘表明知识就隐藏在日常积累下来的大量数据之中，仅靠复杂的算法和推理并不能发 现知识，数据才是知识的源泉。

. 一、数据仓库技术

数据仓库和数据库主要的区别是数据仓库中的数据具有以下四个基本特征：

•数据仓库的数据是面向主题的；

•数据仓库的数据是集成的；

•数据仓库的数据是不可更新的；

・数据仓库的数据是随时间不断变化的。

■二、联机分析处理技术

1. 多维数据模型
2. 多维分析操作
3. 实现方式

OLAP服务器透明地为分析软件和用户提供多维数据视图，实现对多维数据的存储、索引、查 询和优化等。OLAP服务器一般按照多维数据模型的不同实现方式，有MOLAP结构、ROLAP结 构、HOLAP结构等多种结构。 、

■三、数据挖掘技术

1 .数据挖掘的概念 \

数据挖掘是从大量数据中发现并提取隐藏在内的、人们事先不知道的但又可能有用的信息和 知识的一种新技术。

**5**

**9**

**11**

第十六章

数据挖掘的目的是帮助决策者寻找数据间潜在的关联，发现经营者被忽略的要素，而这些要素 对预测趋势、决策行为也许是十分有用的信息。

数据挖掘技术涉及数据库技术、人工智能技术、机器学习、统计分析等多种技术,使决策支持系 统跨入一个新阶段。

数据仓库与联机分析处理技术

2.数据挖掘的功能

数据挖掘的功能主要有以下几种：

（1） 概念描述:就是指归纳总结出数据的某些特征。

（2） 关联分析:若两个或多个变量的取值之间存在某种规律性,就称为关联。包括相关关联和 因果关联。关联规则小不仅是单维关联，也可能是多维之间的关联。

（3） 分类和预测:找到一定的函数或者模型来描述和区分数据类之间的区别，用这些函数和模 型对未来进行预测。这些数据类是事先已经知道的。分类的结果表示为决策树、分类规则或神经 网络。

（4） 聚类:将数据分为多个类，使得类内部数据之间的差异最小，而类之间数据的差异最大。与 分类不同的是，聚类前并不知道类的个数。聚类技术主要包括传统的模式识别方法和数学分类 学等。

（5） 孤立点的检测:孤立点是指与数据中的整体表现行为不一致的那些数据集合。这些数据虽 然是一些特例，但往往在错误检查和特例分析中是很有用的。

（6） 趋势和演变分析:描述行为随着时间变化的对象所遵循的规律或趋势。

■四、新型数据仓库

1. 系统需求的变化

\* （1）数据量急剧增长

（2） 数据类型多样

（3） 决策分析复杂

（4） 底层硬件环境变化

1. 传统数据仓库面临的问题

（1）数据移动代价过高

（2）不能快速适应变化

6

9

1—

1. 大数据时代的新型数据仓库的特点

高度可扩展、高性能、高度容错性、支持异构环境、较低的分析延迟、易用且开放接口、较低成 本、向下的兼容性。

例**1** 数据挖掘的主要功能有哪些？

数据库系统概论(第**5**版)同步辅导及习题全解

典型例题

解析数据挖掘的功能主要有以下几种：

（1） 概念描述:就是指归纳总结出数据的某些特征。

（2） 关联分析:若两个或多个变量的取值之间存在某种规律性,就称为关联。包括相关关联 和因果关联。关联规则小不仅是单维关联，也可能是多维之间的关联。

（3） 分类和预测:找到一定的函数或者模型来描述和区分数据类之间的区别,用这些函数和 模型对未来进行预测。这些数据类是事先已经知道的。分类的结果表示为决策树、分类规 则或神经网络。

（4） 聚类:将数据分为多个类，使得类内部数据之间的差异最小，而类之间数据的差异最大。 与分类不同的是,聚类前并不知道类的个数。聚类技术主要包括传统的模式识别方法和数 学分类学等。

（5） 孤立点的检测:孤立点是指与数据中的整体表现行为不一致的那些数据集合。这些数据 虽然是一些特例，但往往在错误检查和特例分析中是很有用的。

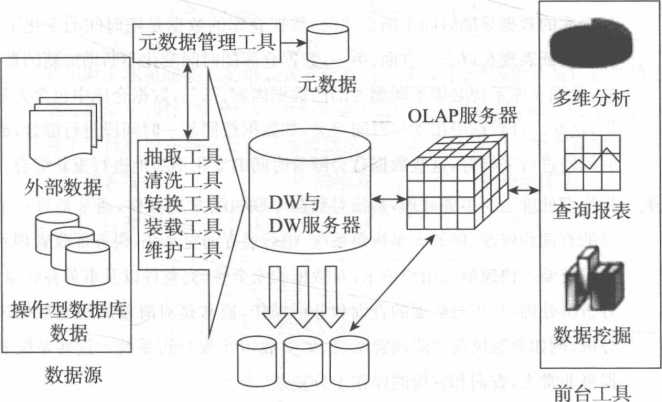
（6） 趋势和演变分析:描述行为随着时间变化的对象所遵循的规律或趋势。

例**2** 传统数据仓库所面临的问题有哪些？

解析（1）数据仓库系统的体系结构中，在数据源层和分析层之间引入了一个存储管理层,虽然可 以提升数据质量并针对査询进行优化,但也付出了较大的数据迁移代价和执行时的连接 代价。

（2）不能快速适应变化。传统的数据仓库假设主题是较少变化的，其应对变化的方式是对数 据源到前端展现的整个流程中的每个部分进行修改，然后重新加载数据，甚至重新计算数 据,导致其适应变化的周期较长，但这种模式比我适合对数据质量和查询性能要求较高、而 不太计较预处理代价的场合。但在大数据时代，分析处在变化的业务环境中，这种模式将难 以适应新的需求。

例**3**在大数据时代的背景下，新型数据仓库解决方案应具有哪些重要特性？



解析大数据分析平台需具备的特性：

（1） 高度可扩展性：横向大规模可扩展，大规模并行处理；

（2） 高性能:快速响应复杂查询与分;析；

（3） 高度容错性:查询失败时,只需重做部分工作；

（4） 支持异构环境:对硬件平台一致性要求不高，适应能力强;

（5） 较低的分析延迟:业务需求变化时，能快速反应；

（6） 易用且开放接口：既能方便查询，又能处理复杂分析；

（7） 较低成本:较高的性价比；

（8） 向下兼容性:支持传统的商务智能工具。

例**4** 数据仓库系统的体系结构是怎样的？

解析 数据仓库系统架构如图16.1所示，由数据仓库的后台工具、数据仓库服务器、（）LAP服务器

和前台工具组成。

数据集市

9

1—

第十六章

数据仓库与联机分析处理技术

图 16.1

数据仓库的后台工具包括数据抽取、清洗、转换、装载和维护工具，目前许多公司产品把后台 工具简记为ECTL工具或ETL工具。

数据仓库服务器相当于数据库系统中的DBMS,它负责管理数据仓库中数据的存储和数据 存取，并给OLAP服务器和前台工具提供存取接口。数据仓库服务器目前一般是RDBMS 或扩展的RDBMS.即由传统数据库厂商对DBMS加以扩展修改，使它能更好地支持数据仓 库的功能。

OLAP服务器透明地为前台工具和用户提供多维数据视图。用户不必关心它的分析数据到

底存储在什么地方，是怎么存储的。

前台工具包括查询报表工具、多维分析工具、数据挖掘工具和分析结果可视化工具等。

1. 陸W i.仓库的数据是面向主题的;数据仓库中的数据是面向主题进行组织的。主题是一 个抽象的概念，是在较高层次上将企业信息系统中的数据综合、归类并进行分析利 用的抽象;在逻辑意义上，它对应企业中某一宏观分析领域所涉及的分析对象。

**98**数据库系统概论(第**5**版)同步辅导及习题全解

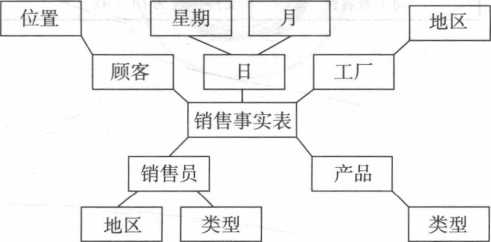
课后习题解答

1. 仓库的数据是集成的;操作型数据与分析型数据之间差别甚大，数据仓库的数据 是从原有的分散的数据库数据中抽取来的，因此数据在进入数据仓库之前,必然要 经过加工与集成，统一与综合处理。
2. 仓库的数据是不可更新的;数据仓库主要是供决策分析之用的，所涉及的数据操 作主要是数据查询，一般情况下并不进行修改操作。
3. 仓库的数据是随时间不断变化的;数据仓库的数据是随时间的变化不断变化的: 这一特征表现在以下三方面:第一，数据仓库随时间变化不断增加新的数据内容;第 二，数据仓库系统必须不断删去旧的数据内容;第三，数据仓库中包含大量的综合数 据，这些综合数据中很多与时间有关，如数据按照某一时间段进行综合，或隔一定的 时间段进行采样等，这些数据就会随着时间的变化不断地进行重新综合。

**2**」努题距\操作型处理也叫事务处理，是指对数据库联机的常规操作，通常是对一个或一组记 录的查询和修改，例如火车售票系统、银行通存通兑系统、税务征收管理系统等。这 些系统要求快速响应用户请求，对数据的安全性、完整件以及事务吞吐量要求很高。 分析型处理，是指对数据的查询和分析操作，通常是对海量的历史数据进行査询和 分析,例如金融风险预测预警系统、证券股市违规分析系统。这些系统要访问的数 据量非常大，查询和分析的操作十分复杂。

操作型数据和分析型数据的区别：

|  |  |
| --- | --- |
| 操作型数据 | 分析型数据 |
| 细节的 | 综合的，或提炼的 |
| 在存取瞬间是准确的 | 代表过去的数据 |
| 可更新 | 不可更新 |
| 操作需求事先可知道 | 操作需求事先不知道 |
| 生命周期符合软件开发生命周期 | 完全不同的生命周期 |
| 对性能要求高 | 对性能要求宽松 |
| 一个时刻操作一元组 | 一个时刻操作一集合 |



| 操作型数据 | 分析型数据 |
| --- | --- |
| 事务驱动 | 分析驱动 |
| 面向应用\ | 面向分析 |
| 一次操作数据量小 | 一次操作数据量大 |
| 支持日常操作 | 支持管理决策需求 |

9

9

1—

**3.**解题會k如图16. 2所示的星形模式的中心是销售事实表，其周围的维表有时间维表、顾客维 表、销售员维表、制造商维表和产品维表。

第十六章

图 16.2

如果对上图星状模式的维表按照层次进一步细化:顾客维表可以按照所在地区的位 置分类聚集;时间维表可以有两类层次：日、月，日、星期；制造商维表可以按照工厂 且工厂按照所在地区分层。这样就形成了如图16. 3所示的雪花模式。

数据仓库与联机分析处理技术

顾客维表

时间维表

制造商维表

销售员维表

产品维表

图 16.3

1. 解麒盐传统的决策支持系统通常是在某个假设的前提下通过数据查询和分析来验证或否 定这个假设。数据挖掘与传统的数据分析（如查询、报表、联机应用分析）的本质区 別是数据挖掘是在没有明确假设的前提下去挖掘信息、发现知识。

数据挖掘技术是基于大量的来自实际应用的数据，进行自动分析、归纳推理,从中发 掘出数据间潜在的模式或产生联想，建立新的任务模型帮助决策者调整企业发展策

略，进行正确决策。

**00**数据库系统概论(第**5**版)同步辅导及习题全解

2

数据挖掘所得到的信息应具有事先未知、有效和可实用三个特征。

事先未知的信息是指该信息是未曾预料到的，即数据挖掘是要发现那些不能靠直觉 发现的信息或知识，甚至是违背直觉的信息或知识。挖掘出的信息越是出乎意料, 就可能越有价值。

**5**.廨题暨\ （1）传统数据仓库所面临的问题：

1. 数据移动代价过高；
2. 不能快速适应变化。

（2）为了应对这些挑战，以较低的成本高效地支持大数据分析,新型的数据仓库解决 方案需具备下表所示的几个重要特性。

| 特性 | 简要说明 |
| --- | --- |
| 高度可扩展 | 横向大规模可扩展，大规模并行处理 |
| 高性能 | 快速响应复杂查询与分析 |
| 高度容错性 | 査询失败时，只需重做部分工作 |
| 支持异构环境 | 对硬件平台一致性要求不高，适应能力强 |
| 较低的分析延迟 | 业务需求变化时，能快速反应 |
| 易用且开放接口 | 既能方便査询，又能处理复杂分析 |
| 较低成本 | 较高的性价比 |
| 向下兼容性 | 支持传统的BI工具 |

(1) 求供应工程JI零件的供应商号SNOo

SELECT SNO FROM SPJ WHERE JNC)= 1 2 3 4jr ；

(2) 求供应工程JI零件Pl的供应商号SNO。

SELECT SNO FROM SPJ WHERE JNO= 4J1, AND PNO=4P1'；

(3) 求供应工程JI红色零件的供应商号SNO。

SELECT SNO FROM SPJ WHERE JNO= 'JI' AND

PNO IN (SELECT PNO FROM P WHERE COL()R='红')；

或 SELECT SNO FROM SPJ,P WHERE JNO=<J1, AND SPJ. PNO=P. PNO

ANDC()L()R=‘ 红')；

(4) 求没有使用天津供应商生产的红色零件的工程号JNO。

从J表入手,以包含那些尚未使用任何零件的工程号。

SELECT JNO FROM J WHERE NOT EXISTS

(SELECT \* FROM SPJ WHERE SPJ. JNO=J. JNO

AND SNO IN (SELECT SNO FROM S WHERE CITY=，天津，)

AND PNO IN (SELECT PNO FROM P WHERE COL()R=‘红'))；