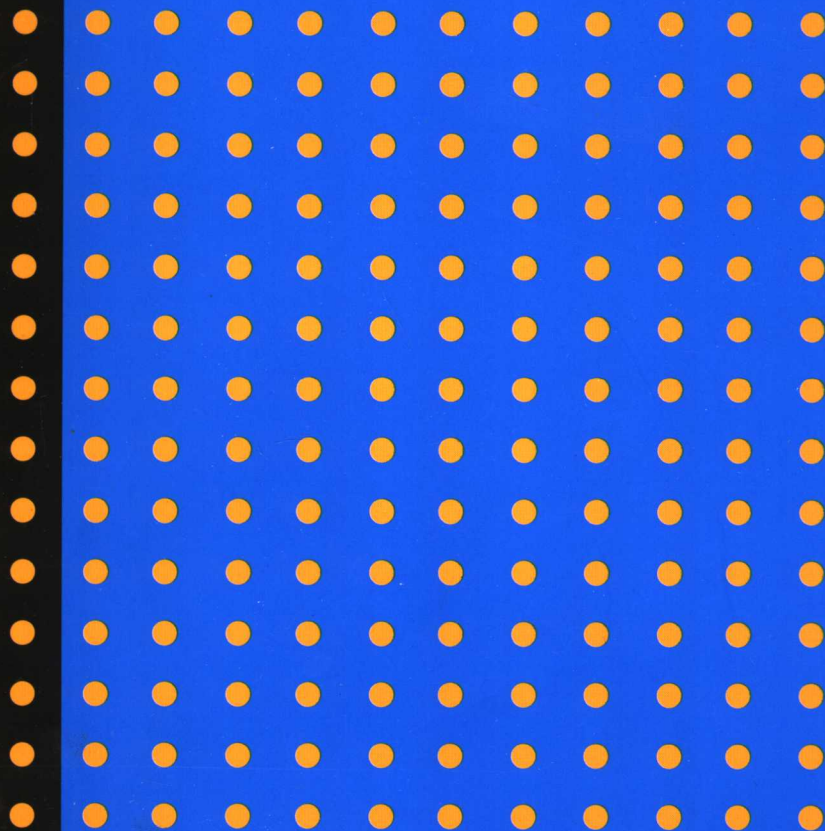


重点大学计算机专业系列教材

数据库系统实验指导教程

汤娜 汤庸 叶小平 刘海 编著

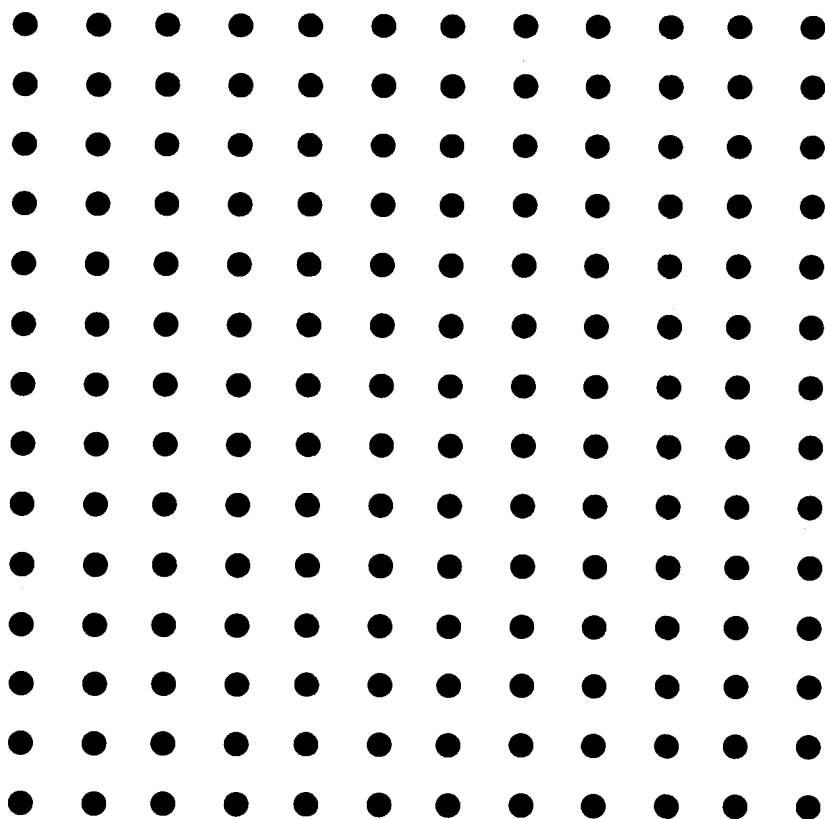


清华大学出版社

重点大学计算机专业系列教材

数据库系统实验指导教程

汤娜 汤庸 叶小平 刘海 编著



清华大学出版社
北京

内 容 简 介

本书是数据库的实验教材,从测试 DBMS 边界的角度出发,通过案例现象引导读者主动思考现象的成因,对 SQL 的基本知识,数据库系统的完整性控制、安全性控制、并发控制、数据备份与恢复、性能检测、索引进行实验内容组织。每个实验还有习题及其参考答案。

通过本书附录可了解数据库的逻辑结构,搭建实验环境,并将数据库结构和数据直接导入到计算机中,相关数据和资料可在清华大学出版社网站下载。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

数据库系统实验指导教程/汤娜等编著. —北京:清华大学出版社,2006.4

(重点大学计算机专业系列教材)

ISBN 7-302-12560-0

I. 数… II. 汤… III. 数据库系统—高等学校—教材 IV. TP311.13

中国版本图书馆 CIP 数据核字(2006)第 011112 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦
http://www.tup.com.cn 邮 编: 100084
社 总 机: 010-62770175 客 户 服 务: 010-62776969

组稿编辑: 丁 岭

文稿编辑: 张为民

印 装 者: 北京鑫海金澳胶印有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185 × 260 印 张: 14.5 字 数: 337 千字

版 次: 2006 年 4 月第 1 版 2006 年 4 月第 1 次印刷

书 号: ISBN 7-302-12560-0/TP · 8035

印 数: 1 ~ 3000

定 价: 20.00 元

INTRODUCTION

出版说明

随着国家信息化步伐的加快和高等教育规模的扩大,社会对计算机专业人才的需求不仅体现在数量的增加上,而且体现在质量要求的提高上,培养具有研究和实践能力的高层次的计算机专业人才已成为许多重点大学计算机专业教育的主要目标。目前,我国共有 16 个国家重点学科、20 个博士点一级学科、28 个博士点二级学科集中在教育部部属重点大学,这些高校在计算机教学和科研方面具有一定优势,并且大多以国际著名大学计算机教育为参照系,具有系统完善的教学课程体系、教学实验体系、教学质量保证体系和人才培养评估体系等综合体系,形成了培养一流人才的教学和科研环境。

重点大学计算机学科的教学与科研氛围是培养一流计算机人才的基础,其中专业教材的使用和建设则是这种氛围的重要组成部分,一批具有学科方向特色优势的计算机专业教材作为各重点大学的重点建设项目成果得到肯定。为了展示和发扬各重点大学在计算机专业教育上的优势,特别是专业教材建设上的优势,同时配合各重点大学的计算机学科建设和专业课程教学需要,在教育部相关教学指导委员会专家的建议和各重点大学的大力支持下,清华大学出版社规划并出版本系列教材。本系列教材的建设旨在“汇聚学科精英、引领学科建设、培育专业英才”,同时以教材示范各重点大学的优秀教学理念、教学方法、教学手段和教学内容等。

本系列教材在规划过程中体现了如下一些基本组织原则和特点。

1. 面向学科发展的前沿,适应当前社会对计算机专业高级人才的培养需求。教材内容以基本理论为基础,反映基本理论和原理的综合应用,重视实践和应用环节。

2. 反映教学需要,促进教学发展。教材要能适应多样化的教学需要,正确把握教学内容和课程体系的改革方向。在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

3. 实施精品战略，突出重点，保证质量。规划教材建设的重点依然是专业基础课和专业主干课；特别注意选择并安排了一部分原来基础比较好的优秀教材或讲义修订再版，逐步形成精品教材；提倡并鼓励编写体现重点大学计算机专业教学内容和课程体系改革成果的教材。

4. 主张一纲多本，合理配套。专业基础课和专业主干课教材要配套，同一门课程可以有多本具有不同内容特点的教材。处理好教材统一性与多样化的关系；基本教材与辅助教材以及教学参考书的关系；文字教材与软件教材的关系，实现教材系列资源配套。

5. 依靠专家，择优落实。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时，要引入竞争机制，通过申报、评审确定主编。书稿完成后要认真实行审稿程序，确保出书质量。

繁荣教材出版事业，提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量，希望有志于教材建设的教师能够加入到我们的编写队伍中来。

教材编委会

FOREWORD

前言

数据库技术出现于 20 世纪 60 年代末,经过 30 多年的迅速发展,已经成为计算机科学技术领域中最重要学科之一。数据库领域已经形成了一大批实用系统,这些系统作为数据管理最有效的手段,深入到了计算机应用的各个方面,极大地促进了计算机应用的发展。统计说明,数据库技术在计算机应用中所占的比例已经超过了 60%。

数据库系统原理作为大学计算机及相关专业的必修主干课程,也是其他许多专业学生的选修课程。数据库虽然是一门实践的学科,但目前大多数学校都没有设置数据库的实验环节,这和图书市场上与本科教学配套的实验教材相对匮乏有关。本书能为数据库实验教材的建设提供一定的角度和思路。本书从测试 DBMS 边界的角度出发,通过案例现象引导读者主动思考现象的成因,再通过新的现象测试验证对成因的揣度是否正确。实验的设计思路是现象→原理→现象,锻炼学生的抽象、归纳和演绎的能力,让学生能清晰了解 DBMS 与用户的边界、管理和调优系统的手段,帮助学生深入了解系统,而非停留在仅会使用 and 操作的层次。

为了使教学内容和实验内容更容易被接受,本书在现有的关系数据库产品中选用了 Microsoft SQL Server 2000 作为实验平台,该产品以简单、实用且界面友好著称。这对于数据库系统知识的入门者,既能全面了解数据库的系统知识,又能避免读者陷入 DBMS 技术要点的海洋中。本书着眼于帮助读者了解 DBMS 的基本工作原理,并清楚了解系统的边界,不是一本数据库产品的说明书。所以,如果读者需要对数据库产品的操作有更细致的了解,可以参考产品的帮助文件和相关支持网站。

本书是为了配合本科教学中的数据库实践部分编写的,所以在内容组织上紧贴本科教学的教学内容来组织每一章的实验内容,由于本书的实验内容可以独立使用,可以配合任何一本本科数据库教材。所以在每一章中首先对实验中涉及的知识点作了回顾,然后每个实验在组织实验数据及现象的观察过程中,为了阐述现象后的本质,对涉及的知识点会做进一步的解释,以帮助了解 DBMS 的基本工作原理,并清楚了解系统的边界。本书毕

竟是实验讲义,并不能替代本科的数据库课程教材,如果读者需要对数据库更系统的学习,还需要一本本科数据库课程教材。

全书分为 7 章。第 1 章针对本科教学中 SQL 语句涉及的所有基本知识点进行实验内容组织,第 2~5 章分别就系统中的完整性控制、安全性控制、并发控制、数据备份与恢复等系统知识进行实验内容组织,第 6 章对如何进行性能检测进行讨论,第 7 章针对索引知识进行实验内容组织。每个实验都有习题,可以进一步检验读者对相关知识的掌握程度。

本书相关资料放在清华大学出版社网站上,在下载后的文件中,code 目录下存放本书各章节涉及的代码,database 目录下是本书采用的实验数据库和实验数据。读者可以先通过阅读本书附录 A 了解数据库的逻辑结构,然后通过附录 B 搭建实验环境,并将数据库结构和数据直接导入到用户的计算机中。如果想使用下载文件中的代码,可以通过查询分析器中的“文件”菜单下的“打开”命令选择代码文件进行使用。

编写一本适用本科教学的实验教材并非易事,由于作者水平所限,上述目的是否能完全达到,还需要实践检验;同时对于本书的不足与疏漏之处,也恳请读者和专家批评指正。读者的建议、意见可以通过发电子邮件到 isstn@mail.sysu.edu.cn 或 issty@mail.sysu.edu.cn 与我们联系。

本书由汤娜和汤庸统稿,叶小平、刘海、刘晓玲、林伟嘉、陈子聪、吴凌坤等参与了部分内容的编写。在本书编写过程中,还参考了国内外数据库相关教材等书籍,在此一并表示谢意。

作 者

2005 年 12 月于中山大学

CONTENTS

目录

第 1 章 SQL 语言	1
实验 1.1 数据定义	1
实验 1.2 数据查询	6
实验 1.3 数据更新	24
实验 1.4 视图	30
实验 1.5 数据控制	37
实验 1.6 空值和空集的处理	42
习题答案	50
第 2 章 数据库的完整性控制	57
实验 2.1 实体完整性	58
实验 2.2 参照完整性	62
实验 2.3 用户自定义完整性	70
实验 2.4 触发器	76
习题答案	79
第 3 章 数据库的安全性控制	82
实验 3.1 用户标识与鉴别	83
实验 3.2 自主存取控制	87
实验 3.3 视图机制在自主存取控制上的应用	96
习题答案	98
第 4 章 数据库事务	100
实验 4.1 SQL Server 中事务的定义	100
实验 4.2 SQL Server 2000 事务与锁	106
实验 4.3 SQL Server 2000 事务与隔离级别	112

实验 4.4 锁冲突与死锁	119
实验 4.5 SQL Server 2000 的事务与事务日志	124
习题答案	130
第 5 章 数据库的备份与还原	131
实验 5.1 SQL Server 数据库的备份	131
实验 5.2 SQL Server 数据库的还原	138
实验 5.3 系统数据库的备份与恢复	143
习题答案	146
第 6 章 性能监测	147
实验 6.1 查看事务的执行计划	150
实验 6.2 查看事务执行的统计信息	156
实验 6.3 运用查询调控器	162
实验 6.4 运用事件探查器	164
实验 6.5 运用性能监测器	170
实验 6.6 运用锁监测工具	173
实验 6.7 综合应用	178
习题答案	183
第 7 章 物理优化	184
实验 7.1 建立索引	184
实验 7.2 覆盖索引的作用	187
实验 7.3 聚簇索引	189
实验 7.4 维护索引的代价	191
实验 7.5 冗余索引	193
实验 7.6 表中多个索引的选择	195
实验 7.7 索引与表的连接查询	199
实验 7.8 填充因子	202
实验 7.9 组合索引	207
习题答案	210
附录 A 实验数据环境说明	216
附录 B 实验环境构建指南	217

SQL 语言

第 1 章

SQL 是 Structured Query Language 的缩写,译成“结构化查询语言”,是一种介于关系代数与关系演算之间的语言,其功能包括数据查询(data query)、数据操纵(data manipulation)、数据定义(data definition)和数据控制(data control)四个方面,是一个通用的、功能极强的关系数据库语言。目前,SQL 语言已经成为关系数据库的标准语言。

SQL 具有高度综合统一、高度非过程化、采取面向集合操作方式、支持三级模式结构、具有一种语法两种使用方式、结构简洁、易学易用的特点,所以为广大用户和业界所接受,成为国际标准。

SQL 支持关系数据库三级模式结构。其中,内模式对应于存储文件,模式对应于基本表,外模式对应于视图(view)。

存储文件的逻辑结构组成了关系数据库的内模式,存储文件的物理文件结构是透明的。基本表是本身独立存在的表,在 SQL 中,一个关系就对应一个表。一些基本表对应一个存储文件,一个表可以带若干索引,索引也存放在存储文件中。

视图是从基本表或其他视图中导出的一个虚表。在数据库中只存放视图的定义而不存放视图对应的数据。用户可以用 SQL 对视图和基本表进行查询。在用户看来,视图和基本表都是关系,而存储文件对用户是透明的。

SQL 的基本功能包括数据定义功能、数据查询功能、数据更新功能、视图管理功能和数据控制功能等五个方面。其中,数据查询是 SQL 的最主要功能。

实验 1.1 数据定义

1. 实验目的

熟悉 SQL 的数据定义语言,能够熟练地使用 SQL 语句来创建和更改基本表,创建和取消索引。

2. 原理解析

SQL 的数据定义包括基本表的定义、索引的定义和视图的定义三部分。这里的“定义”包括创建(create)、取消(drop)和更改(modify)三部分内容,如表 1.1 所示。在本实验中,主要介绍基本表和索引的定义,视图部分的内容将在实验 1.4 视图中讨论。

表 1.1 SQL 的数据定义语句

操作对象	创 建	取 消	更 改
表	CREATE TABLE	DROP TABLE	ALTER TABLE
视图	CREATE VIEW	DROP VIEW	
索引	CREATE INDEX	DROP INDEX	

(1) SQL 的基本数据类型。

SQL 在定义表的各个属性时,要求指明其中的数据类型和长度。不同的 DBMS 支持的数据类型不完全一样。表 1.2 列举了 Microsoft SQL Server 支持的主要数据类型。

表 1.2 Microsoft SQL Server 支持的主要数据类型

序号	符 号	数据类型	说 明
1	TINYINT	整数类型	其值按 1 个字节存储
2	SMALLINT	整数类型	其值按 2 个字节存储
3	INTEGER or INT	整数类型	其值按 4 个字节存储
4	REAL	实数类型	其值按 4 个字节存储
5	FLOAT	实数类型	其值按 8 个字节存储
6	CHARACTER(n) or CHAR(n)	长度为 n 的字符类型	1 个字符占 1 个字节
7	VARCHAR(n)	最大长度为 n 的 变长字符类型	所占空间与实际字符数有关
8	DATETIME	日期时间类型	默认格式为 MM-DD-YYYY, HH: MM: AM/PM
9	TIMESTAMP	时间戳	更新或插入一行时,系统自 动记录的日期时间类型

(2) SQL 语句用 CREATE 语句来创建基本表,它的一般形式是:

```
CREATE TABLE <表名>(<列名>,<数据类型>[列级完整性约束条件]
[,<列名>,<数据类型>[列级完整性约束条件]]...
[,<表级完整性约束条件>])
```

其中,[]内的内容是可选项,<表名>是所要定义的基本表名称。

在使用 CREATE 语句创建基本表时,必须指定基本表的表名、每个列名和数据类型,而列级的完整性约束则是备选的内容,可以指定,也可以省略。对主键和外键等的定义,一般是在基本表中各个列属性的定义之后出现。各个列的数据类型必须与表 1.1 中的数据类型相匹配,不然会产生编译错误,不能运行。

在定义外键的时候,要用保留字 REFERENCES 来指出外键来自的表名,即主表。可以通过使用引用完整性任选项 ON DELETE,来指出主表中被引用主属性被删除时,在引用表中的数据的处理方式。为了保证完整性,可能的采取方法有如下三种:

- ① 选用 RESTRICT 选项,表明被基本表所引用的主属性不得删除。
- ② 选用 CASCADE 选项,表明若主表中删除被引用的主属性,则基本表中引用该外键的对应行应随着被删除。
- ③ 选用 SET NULL 选项,表明若主表中删除被引用的主属性,则基本表中引用该外键的对应行中的该属性值会被置空。当然,这必须具有一个前提,即该属性在前面说明应该没有 NOT NULL 限制。

(3) 在更改表的时候,可以使用 ALTER 语句增加新的列,或修改列的数据类型,取消完整性约束。更改的过程是将列作为一个对象来进行。例如修改列的数据类型,则在整个表中,所有元组的这个列的取值的数据类型都发生改变。标准 SQL 没有提供直接的删除列的功能,当需要删除列的时候,必须通过间接步骤来实现。

ALTER 语句更改表的语句的一般格式如下:

```
ALTER TABLE <表名>
[ADD <新列名><数据类型>[完整性约束条件]]
[DROP CONSTRAINT[完整性约束名]]
[ALTER COLUMN<列名><数据类型>]
```

其中,ADD 子句用于增加新列和新的完整性约束条件; DROP CONSTRAINT 子句用于取消完整性约束条件; ALTER COLUMN 子句用于更改原有的定义,包括更改列名和数据类型。

(4) 取消表的语句格式:

```
DROP TABLE <表名>
```

(5) 创建索引的语句格式:

```
CREATE [UNIQUE][CLUSTERED] INDEX <索引名>
ON <表名>(<列名>[<排序方式>][,<列名>[,<排序方式>]]...)
```

(6) 取消索引的语句格式:

```
DROP INDEX 表名.索引名
```

3. 实验内容

本实验的主要内容包括:

- 使用 CREATE 语句创建基本表。
- 更改基本表的定义,增加列,删除列,修改列的数据类型。
- 创建表的升降序索引。
- 取消表、表的索引或表的约束。

4. 实验步骤

1) 要求

(1) 使用 SQL 语句创建关系数据库表: 人员表 PERSON(P#, Pname, Page)、房间表 ROOM(R#, Rname, Rarea)、表 PR(P#, R#, Date)。其中: P# 是表 PERSON 的主键, 具有唯一性约束; Page 具有约束“大于 18”; R# 是表 ROOM 的主键, 具有唯一性约束。表 PR 中的 P#、C# 是外键。

(2) 更改表 PERSON, 增加属性 Ptype(类型是 CHAR, 长度为 10), 取消 Page 大于 18 的约束。把表 ROOM 中的属性 Rname 的数据类型改成长度为 40。

(3) 删除表 ROOM 的一个属性 Rarea。

(4) 取消表 PR。

(5) 为 ROOM 表创建按 R# 降序排列的索引。

(6) 为 PERSON 表创建按 P# 升序排列的索引。

(7) 创建表 PERSON 的按 Pname 升序排列的唯一性索引。

(8) 取消 PERSON 表 P# 升序索引。

2) 分析与解答

(1) 使用 SQL 语句创建关系数据库表: 人员表 PERSON(P#, Pname, Page)、房间表 ROOM(R#, Rname, Rarea)、表 PR(P#, R#, Date)。其中: P# 是表 PERSON 的主键, 具有唯一性约束; Page 具有约束“大于 18”; R# 是表 ROOM 的主键, 具有唯一性约束。表 PR 中的 P#、C# 是外键。

在实际情况下, 可以使用下列的 SQL 语句来实现上述功能(语句或代码参见网站提供的 1_1_1.sql 代码文件, 简称“见 1_1_1.sql”, 下同):

```
CREATE TABLE PERSON
(P# CHAR(8) NOT NULL UNIQUE,
Pname CHAR(20) NOT NULL,
Page INT,
PRIMARY KEY(P#), CHECK(Page>18))
```

```
CREATE TABLE ROOM
(R# CHAR(8) NOT NULL UNIQUE,
Rname CHAR(20),
Rarea FLOAT(10),
PRIMARY KEY(R#))
```

```
CREATE TABLE PR(
P# CHAR(8) NOT NULL UNIQUE,
R# CHAR(8) NOT NULL UNIQUE,
Date Datetime,

PRIMARY KEY(P#, R#),
```

```
FOREIGN KEY(P#) REFERENCES PERSON ON DELETE CASCADE,  
FOREIGN KEY(R#) REFERENCES ROOM ON DELETE CASCADE  
)
```

(2) 更改表 PERSON, 增加属性 Ptype(类型是 CHAR, 长度为 10), 取消 Page 大于 18 的约束。把表 ROOM 中的属性 Rname 的数据类型改成长度为 40。

可以使用下列的 SQL 语句来更改表(见 1_1_2.sql):

```
ALTER TABLE PERSON ADD Ptype CHAR(10)  
ALTER TABLE PRRSN DROP CONSTRAINT CK_PERSON_Page_78B3EFCA  
ALTER TABLE ROOM ALTER COLUMN Rname CHAR(40)
```

注意: 在取消约束的时候, 应该写出约束的名称。当定义约束的时候, 虽然用户没有指定约束的名称, 但是在数据库会为此约束起一个名称(不同的时候创建名称可能不同)。要查看这个名称, 可以通过在企业分析器中, 查看设计表中的约束选项, 可以看到这个约束的名称。在本实验中, 这个约束的名称为 CK_PERSON_Page_78B3EFCA。

(3) 删除表 ROOM 的一个属性 Rarea。

在 SQL Server 2000 中, 提供了删除列的操作, 可以使用下列的 SQL 语句实现(见 1_1_3.sql):

```
ALTER TABLE ROOM DROP COLUMN Rarea
```

在其他数据库支持的 SQL 中, 不是都有直接删除列的语句的。当 SQL 不支持直接删除列操作的时候, 要实现这一个操作, 可以通过先把表删除, 再重新定义符合要求的新表这两步的操作来完成。在这个过程中, 删除表会导致数据的丢失, 所以在删除表之前, 需要复制一份原表的数据和定义用以保存。在定义了新的符合要求的表后, 再把表的数据导入新的表中。

(4) 取消表 PR。

取消表, 直接使用 DROP 语句(见 1_1_4.sql):

```
DROP TABLE PR
```

(5) 为 ROOM 表创建按 R# 降序排列的索引(见 1_1_5.sql)。

创建索引使用 CREATE 语句, 需要指出索引的名称。

```
CREATE INDEX XCNO ON ROOM(R# DESC)
```

(6) 为 PERSON 表创建按 P# 升序排列的索引(见 1_1_6.sql)。

```
CREATE INDEX XSNO ON PERSON(P#)
```

在本题中, 创建的索引的名称是 XSNO, 在这里列名的排列方式省略了。在 SQL 语句中, 排列方式只有两种 ASC(升序)和 DESC(降序), 默认值为 ASC。所以, 本题排列的方式是升序。

(7) 创建表 PERSON 的按 Pname 升序排列的唯一性索引。

在前面的创建索引中,所有的保留字 UNIQUE 和 CLUSTER 都被省略了,这时取默认值 CLUSTER。当需要建立唯一性索引的时候,UNIQUE 就不能省略。例如下面的 SQL 语句创建表 PERSON 的索引 RNUA(见 1_1_7.sql):

```
CREATE UNIQUE INDEX RNUA ON PERSON (Pname ASC)
```

(8) 取消 PERSON 表 P# 升序索引。

所要求取消表 PERSON 的索引 XSNO 的 SQL 语句如下(见 1_1_8.sql):

```
DROP INDEX PERSON.XSNO
```

说明:在 Microsoft SQL Server 中进行 SQL 语句的实验,可以采取下列步骤:

- 打开企业管理器,双击选定的数据库。在菜单“工具”中,单击“SQL 查询分析器”,打开查询分析器。
- 在查询分析器中输入相应的 SQL 语句。
- 按 F5 键或是单击按钮执行。

5. 习题

(1) 创建数据库表 CUSTOMERS(CID,CNAME,CITY,DISCNT)、数据库表 AGENTS(AID,ANAME,CITY,PERCENT)、数据库表 PRODUCTS(PID,PNAME)。其中,CID、AID、PID 分别是各表的主键,具有唯一性约束。

(2) 创建数据库表 ORDERS(ORDNA,MONTH,CID,AID,PID,QTY,DOLLARS)。其中,ORDNA 是主键,具有唯一性约束。CID、AID、PID 分别是外键引用自表 CUSTOMERS、表 AGENTS,表 PRODUCTS。

(3) 增加数据库表 PRODUCTS 三个属性列: CITY、QUANTITY、PRICE。

(4) 为以上四个表建立各自的按主键增序排列的索引。

(5) 取消(4)建立的四个索引。

试一试:动手在自己的机器的 SQL Server 中进行直接删除表 PRODUCTS 的列 CITY 的操作,看看你的 SQL Server 的版本是否支持这一操作?

实验 1.2 数据查询

1. 实验目的

熟悉 SQL 语句的数据查询语言,能够使用 SQL 语句对数据库进行单表查询、连接查询、嵌套查询、集合查询和统计查询。

2. 原理解析

SQL 提供了 SQL 映像语句用于数据库查询。将一个子查询定义为 SUBQUERY,可以得到如下的查询语句的一般形式:

(1) 子查询:

```

SUBQUERY::=
SELECT [ALL|DISTINCT]<目标列表达式> [别名][,<目标列表达式> [别名]]...
FROM <表名或视图名> [别名][,<表名或视图名> [别名]]...
[WHERE <条件表达式>]
[GROUP BY <列名 1>[,<列名 2,>]...]
[HAVING <条件表达式>]]

```

(2) 查询表达式:

```

SELECT STATEMENT::=
SUBQUERY|SUBQUERY UNION[ALL]|INTERSECT[ALL]|EXCEPT[ALL]]SUBQUERY
[ORDER BY (<列名 1> [ASC|DESC] [,<列名 2>] [ASC|DESC]...)]

```

在子查询 SUBQUERY 中,SELECT、FROM 和 WHERE 这三个保留字分别带三个子句,构成了 SQL 语句中最基本的查询语句:

- (1) SELECT 子句表示查询的目标属性。
- (2) FROM 子句表示查询所涉及的关系。
- (3) WHERE 子句表示查询的逻辑条件。

由 SELECT 子句、FROM 子句和 WHERE 子句构成的映像语句的含义是:将 FROM 子句指定的基本表或视图做笛卡儿积;再根据 WHERE 子句的条件表达式,从中找出满足条件的元组;最后根据 SELECT 子句中的目标列表达式形成结果表。

GROUP 子句的作用是将元组按后面所跟的列名 1 的值进行分组,属性列中将值相等的元组分成一个组,每个组在结果表中产生一个记录。如果 GROUP 子句带有 HAVING 子句,则表示只有满足 HAVING 子句指定的条件的元组才能在结果关系表中输出。如果有 ORDER 子句,则结果还要按列名 2 的值的升序(ASC)或降序(DISC)的排列次序输出。

上述形式中的目标列表达式可以有如下两种:

- (1) [<表名>.]或 *。
- (2) [<表名>.]<属性列名表达式>[,<表名>.]<属性列名表达式>]...

其中,<属性列名表达式>是由属性列、作用于属性列的集函数和常量的任意算术运算(+,-,*,/)组成的运算公式。

上述一般形式中的条件表达式形式如下(其中 θ 是比较运算符):

- (1) <属性列名> θ <属性列名>。
- (2) <属性列名> θ <常量>。
- (3) <属性列名> θ [ANY|ALL] (SELECT 语句)。
- (4) <属性列名> [NOT] BETWEEN<常量> AND<常量>。
- (5) <属性列名> [NOT] IN(<值 1>[,<值 2>]...)
- (6) <属性列名> [NOT] IN(SELECT 语句)。
- (7) <属性列名> [NOT] LIKE <匹配串>。
- (8) <属性列名> IS [NOT] NULL。

(9) [NOT] EXISTS (SELECT 语句)。

(10) 条件表达式 AND|OR 条件表达式[AND|OR 条件表达式]…。

(11) NOT <条件表达式>。

(12) 常量 θ <集合函数>。

注意：(12)只能用在 HAVING 的条件表达式中。

由上面的目标表达式和条件表达式的多种不同形式,SQL 语句可以提供多种数据查询方式。

1) 单表查询

使用 SELECT 语句做单表查询时,若在 SELECT 后面只指定输出某些列时,可以只查询表中的这些列;若 SELECT 后面跟 *,则可以查询所有列;若 SELECT 后面是带列名的算术表达式,可以查询经过计算之后的值。而且,用户还可以通过别名来改变查询结果的列标题,这对于含算术表达式、常量、函数名的目标列表达式尤为有用。

(1) 使用 DISTINCT。

当需要消除在查询结果中的重复行的时候,可以使用保留字 DISTINCT。DISTINCT 的对应的选项是 ALL,默认值是 ALL。所以,当要显示重复行的时候,ALL 可以写也可以省略;而要消除重复行的时候,则 DISTINCT 一定要写出来。使用 DISTINCT 的时候要注意它保持的唯一性是对 SELECT 后的列名所组成的新的元组的唯一性,而不是各个单独的列取值或原来元组(没有经过 SELECT 子句映射的原来基本表中的元组)的唯一性。

(2) 使用 WHERE 子句中条件的运算符。

在 WHERE 子句中,可以用下列方法来表示结果元组必须满足的条件:

- 使用常见的比较符号 =、>、<、>=、!=、<>、!>、!<、=*、*= 来做比较;
- 使用 BETWEEN AND、NOT BETWEEN AND 来指定范围;
- 使用 IN、NOT IN 来查找属性值属于或不属于指定集合;
- 使用谓词 LIKE、NOT LIKE 来查找匹配的元组;
- 使用 IS NULL 查找属性值为空值的元组;
- 使用逻辑运算符 AND 和 OR 可连接多个查询条件。如果这两个运算符同时出现在同一个 WHERE 条件子句中,则 AND 的优先级高于 OR,但用户可以用括号改变优先级。

(3) 使用 ORDER BY 对查询结果排序。

用户可以用 ORDER BY 子句指定按照一个或多个属性列的升序或降序重新排列查询结果,其中升序为默认值。

(4) 使用集函数。

SQL 提供集函数进行查询,可以直接返回查询的一些统计结果,方便地统计需要得到的数据。SQL 函数是一种综合信息的统计函数,包括计数,求最大值、最小值、平均值、和值等。SQL 函数一般是以列标识符出现在 SELECT 子句的目标列中,也可以出现在