

-第一章

1.1 名词解释

数据库 (DB)	数据库系统 (DBS)	数据库管理系统 (DBMS)
实体	实体集	属性 关键字
数据模型	概念数据模型	关系模型
模式	外模式	内模式
数据独立性	逻辑数据独立性	物理数据独立性
一对一联系	一对多联系	多对多联系

数据库 (DB): 在计算机的存储设备上合理存放、相关联、有结构的数据集合。

数据库系统 (DBS): 在计算机系统中引进数据库和数据库管理系统后的组成。或者是实现有组织地、动态地存储大量相关的结构化数据,方便用户使用数据库的计算机软件、硬件资源组成的系统。

数据库管理系统 (DBMS): 一个操纵和管理数据库的大型软件,它由一组计算机程序构成。它是位于用户与操作系统之间的一层数据管理软件。

实体: 客观存在并可相互区别的个体。

实体集: 具有相同实体特性实体的集合。

属性: 表示实体的特性。

关键字: 能够唯一标识文件中的某一个记录的最小字段集。

数据模型: 现实世界数据与信息的模拟和抽象。

概念数据模型: 简称概念模型,是独立于计算机系统的数据模型,完全不涉及信息在计算机中的表示,是面向数据库用户的现实世界模型,主要用来描述现实世界的概念化结构。

关系模型: 是最重要的一种数据模型,也是目前主要采用的数据模型,用二维表格表示现实世界实体集及实体集间的联系。

模式: 是由数据库设计者综合所有用户数据,按照统一的观点构造的对数据库全局逻辑结构的描述。

外模式: 介于模式与应用之间,是用户与数据库之间的接口,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。

内模式: 也称存储模式,是对数据库中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。

数据独立性: 应用程序和数据结构之间相互独立,不受影响。

逻辑数据独立性: 表示一旦模式发生变化,无需改变外模式或应用程序的能力。

物理数据独立性: 表示不会因为内模式发生改变而导致概念模式发生改变的能力。

一对一联系: 设有两个实体集 **A**、**B**,如果对于实体集 **A** 中的每一个实体, **B** 中至多有一个实体与之有联系,反之亦然,则称 **A**、**B** 有一对一联系 (**1 : 1** 联系)。

一对多联系: 设有两个实体集 **A**、**B**,若 **A** 中每个实体与 **B** 中任意个实体 (包括零个) 相联系,而 **B** 中每个实体至多和 **A** 中一个实体有联系,则称 **A** 和 **B** 是一对多联系 (**1 : n** 联系)。

多对多联系: 设有两个实体集 **A**、**B**,若两个实体集 **A**、**B** 中的每一个实体都和另一个实体集中任意个实体 (包括零个) 有联系,则称 **A**、**B** 是多对多联系 (**m : n** 联系)。

1.2 判断下列说法是否正确

数据库系统的一个主要特点是数据无冗余。

数据库管理系统和数据库构成了数据库系统。

数据结构化是数据库和文件系统的根本区别。

若干个属性集合就能描述一个对象。

数据库系统中数据具有完全独立性。

DBA主要职责是管理数据库中的数据。

数据库避免了一切数据重复。

数据操作语言是一个高级语言。

非过程化语言比过程化语言好。

模式是数据库全局逻辑结构的描述。

三级模式结构是数据库唯一的一种分级模式结构。

层次数据模型和网状数据模型都可用关系数据模型表示。

关系模型不仅可以描述实体，还可以描述实体及实体集之间的联系。

关系数据模型与网状数据模型相比具有查询效率高的优点。

网状数据模型可以直接表示 M: N的联系。

概念模型独立于硬件设备和 DBMS

视图对重构数据库提供了一定程度的物理独立性。

实体是信息世界中的术语，与之相对应的数据库术语为字段。

数据库系统的核心工作就是完成用户级数据库、概念级数据库和物理级数据库之间的映射。

一个网状数据结构模型可以变换为一个等价的层次数据结构模型，这种变换以存储空间为代价。

1-5 : x , x , , x , x ,

6-10 : x , x , , x , ,

11-15 : x , , , x , x

16-20 : , x , x , ,

1.3 简答下列各题

数据管理技术的发展过程。

计算机技术的发展使得数据管理技术也不断向前发展，至今，经历了人工管理阶段、文件系统阶段和数据库系统阶段。

数据库系统的三个发展阶段是什么？

1) 层次型和网状型：

代表产品是 1969 年 IBM 公司研制的层次模型数据库管理系统 IMS。

2) 关系型数据型库：

目前大部分数据库采用的是关系型数据库。1970 年 IBM 公司的研究员 E.F.Codd 提出了关系模型。典型的关系 DBMS产品有 Oracle 、 Sybase、SQLServer 、 DB2 Access、VFP等等。

3) 第三代数据库将为更加丰富的数据模型和更强大的数据管理功能为特征，以提供传统数据库系统难以支持的新应用。它必须支持面向对象，具有开放性，能够在多个平台上使用。

文件系统与数据库系统有何区别和联系？

文件系统和数据库系统之间的区别：

1) 文件系统用文件将数据长期保存在外存上，数据库系统用数据库统一存储数据；

2) 文件系统中的程序和数据有一定的联系，数据库系统中的程序和数据分离，数据具有较高的独立性；

3) 文件系统用操作系统中的存取方法对数据进行管理，数据库系统用 DBMS统一管理和控制数据；

4) 文件系统实现以文件为单位的数据共享，数据库系统实现以记录和字段为单位的数

据共享，数据结构化，冗余度小。

文件系统和数据库系统之间的联系：

- 1) 均为数据组织的管理技术；
- 2) 均由数据管理软件管理数据，程序与数据之间用存取方法进行转换；
- 3) 数据库系统是在文件系统的基础上发展而来的。

数据库系统的特点。

- 1) 采用一定的数据模型实现数据结构化
- 2) 数据共享、冗余度小、避免了数据的不一致
- 3) 具有较高的数据独立性
- 4) 数据由 DBMS 统一管理和控制

何谓数据的物理独立性与数据的逻辑独立性？

逻辑独立性表示一旦模式发生变化，无需改变外模式或应用程序的能力。

物理独立性表示不会因为内模式发生改变而导致概念模式发生改变的能力。

数据模型的三要素。

- 1) 数据结构 ;2) 数据操作 ;3) 数据完整性。

举例说明实体集之间具有 1:1、1:n、m:n 的联系。

例如，有四个实体集：学生、学校、教师、校长，校长和学校为 1:1 联系，学生和学校为 1:n 联系，学生和教师为 m:n 联系。

关系数据模型的特点是什么？

关系数据模型的主要特征是用二维表格表示现实世界实体集及实体集间的联系。具有坚实的理论基础、表达能力强、简单和数据独立性高等优点，缺点是存取效率比较低。

何谓空值？

所谓空值就是“不知道”或“无意义”的值。没有给某个属性赋值，则称其值为空 (NULL)。

层次数据模型、网状数据模型和关系数据模型的优点和缺点。

1) 层次模型具有查询效率较高、结构简单、层次分明，便于在计算机内实现的优点，在层次数据结构中，从根结点到树中任意结点均存在一条唯一的层次路径，为有效地进行数据操纵也提供了条件，并且层次数据模型提供了良好的完整性支持。

但层次模型也有缺点，一是层次数据模型缺乏直接表达现实世界中非层次型结构的复杂联系，二是层次顺序的严格限制，使得对插入或删除操作也带来了较多的限制，并且查询子女结点必须通过双亲结点，所以使得应用程序的编写也比较复杂。

2) 网状模型的主要优点是能直接描述现实世界，记录之间的联系也是通过指针来实现的，所以查询效率高。缺点是结构和编程复杂，难掌握，不易使用。

3) 关系模型有坚实的理论基础。二维表不仅能表示实体集，而且能方便地表示实体集间的联系，所以说它有很强的表达能力，这是层次模型和网状模型所不及的。关系模型具有简单、易学易用的优点。数据独立性高。

关系模型缺点是查询效率常常不如非关系数据模型。

举例说明 E-R 图的构成规则。

方框：表示一实体集，在框内写上实体集的名字

菱形框：表示联系，菱形框内标明联系名，与其相关的实体集之间用箭头表示，一个箭头代表为 1，两个箭头代表为多。

椭圆框：表示属性，在框内写上属性的名字，并用无向边连向与其相关的实体集或联系。

在 E-R 图中，有时为了突出各实体集之间的联系，可以先画出实体集及其属性，然后再重点画出实体之间的联系。如图 1，图 2 所示。

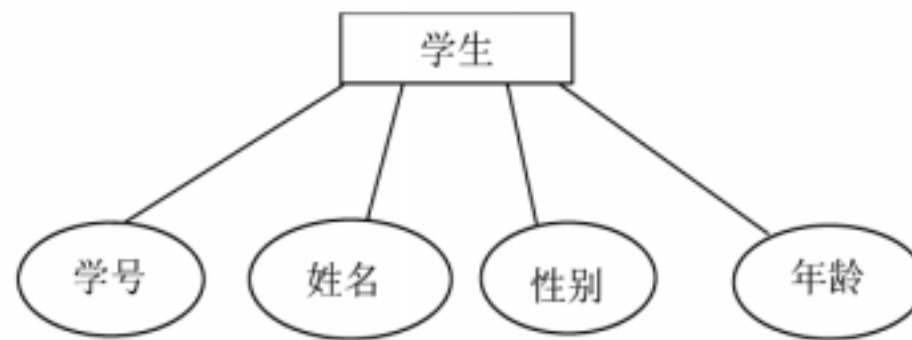


图1 学生实体及其属性

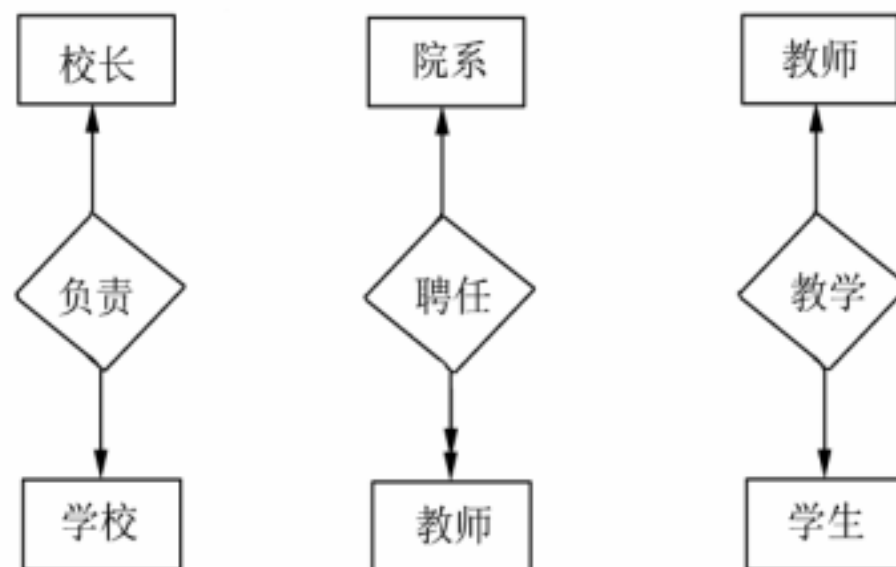


图2 实体之间联系

什么是外部模式？概念模式？内部模式？它们之间有何联系？这种分级结构的优点是什么？

外部模式又称子模式，介于模式与应用之间，是用户与数据库之间的接口，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。

概念模式简称模式，是由数据库设计者综合所有用户数据，按照统一的观点构造的对数据库全局逻辑结构的描述。

内部模式也称存储模式，是对数据库中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。

一个数据库结构从逻辑上可以划分为三个层次：外部模式（External Schema），概念模式（Conceptual Schema）和内部模式（Internal Schema），称为数据库系统的三级模式结构。

数据库系统的三级模式是对数据进行三个级别的抽象。它把数据的具体组织留给DBMS去做，用户只要抽象地处理数据，而不必关心数据在机器中的具体表示方式和存储方式。数据库的三级结构是依靠映像来联系和互相转换的。正是这两层映像保证了数据库系统中的数据具有较高的数据独立性。

什么是数据独立性？在数据库系统体系结构中是如何体现的？

1) 数据独立性：数据库系统在某一层次模式上的改变不会使它的上一层模式也发生改变的过程。数据独立性包括数据的逻辑独立性和物理独立性两种。

2) 一个数据库结构从逻辑上可以划分为三个层次：外部模式（External Schema），概念模式（Conceptual Schema）和内部模式（Internal Schema），称为数据库系统的三级模式结构。数据库的三级结构是依靠外模式 / 模式映像和模式 / 内模式映像来联系和互相转换的。正是这两层映像保证了数据库系统中的数据具有较高的数据独立性，其中外模式 / 模式映像保证逻辑独立性，模式 / 内模式映像保证物理独立性。

数据库系统的软件包括哪些主要成份？

基本的系统软件是操作系统，其它任何软件都必须在它的支持下工作。除操作系统之外，还必须配有数据库管理系统，没有数据库管理系统也就不能称其为是数据库系统。同时，为了开发数据库应用系统，还需要有各种高级语言及其编译系统。

什么是数据字典？它在数据库中的作用是什么？

数据字典（Data Dictionary）是描述各级模式的信息，主要包括所有数据的结构名、意义、描述定义、存储格式、完整性约束、使用权限等信息。由于数据字典包含了数据库系统中的大量描述信息（而不是用户数据），因此也称它为“描述信息库”。

数据字典是 DBMS 存取和管理数据的基本依据，主要由系统管理和使用

DBA 的主要职责是什么？

参与数据库系统的设计与建立。

对系统的运行实行监控。

定义数据的安全性要求和完整性约束条件。

负责数据库性能的改进和数据库的重组及重构工作。

用户访问数据库的过程。

1) 接受应用程序的数据请求

2) DBMS 对用户操作请求进行分析

3) 数据库管理系统要向操作系统发出操作请求

4) 操作系统接到命令后，对数据库中的数据进行处理，将结果送到系统缓冲区，并发出读完标志。

5) DBMS 接到回答信号后，将缓冲区的数据经过模式映射，变成用户的逻辑记录送到用户工作区，同时给用户回答成功与否的信息。

关系数据库完整性约束有哪些？举例说明。

1) 实体完整性约束：实体完整性约束是指任一关系中标识属性（关键字）的值，不能为 NULL，否则，无法识别关系中的元组。

2) 参照完整性约束：参照完整性是不同关系间的一种约束，当存在关系间的引用时，要求不能引用不存在的元组。若属性组 F 是关系 R(U) 的外关键字，并是关系 S(U) 的关键字（即 F 不是 R(U) 的关键字，而是 S(U) 的关键字，称 F 是 R(U) 的外关键字），则对于 R(U) 中的每个元组在属性组 F 上的值必须为：空值（NULL）或者等于 S(U) 中某个元组的关键字值。

3) 用户定义完整性约束。如值的类型、宽度等。

有控制的冗余与无控制的冗余有什么区别？

有控制的冗余是指数据库设计者在设计中为了提高操作效率有意保留的冗余数据，无控制冗余是在数据库设计中由于属性间的依赖而存在的关联数据，两者的区别是无控制冗余不是数据库设计者所要保留的冗余数据，因为它会带来很多问题，使得存储效率比较低，而有控制冗余是数据库设计者有意保留的数据，主要是为了改善对数据库的查询效率。

简述数据库、数据库管理系统和数据库系统之间的关系

数据库是指在计算机的存储设备上合理存放、相关联、有结构的数据集合。这个数据的集合由数据库管理系统进行建立和维护，用户访问数据库中的各种请求都是通过数据库管理系统完成的。数据库系统是在计算机系统中引进数据库和数据库管理系统后的组成，没有数据库和数据库管理系统不能够构成数据库系统。

1.4 选择题

在（ ）中一个结点可以有多个双亲，结点之间可以有多种联系。

A 网状模型 B 关系模型 C 层次模型 D 以上都有

数据库管理系统（DBMS）是（ ）。

- A 一个完整的数据库应用系统 B 一组硬件
C 一组软件 D 既有硬件，也有软件

用户或应用程序看到的那部分局部逻辑结构和特征的描述是（ ）模式。

- A 模式 B 物理模式 C 子模式 D 内模式

要保证数据库的逻辑数据独立性，需要修改的是（ ）。

- A 模式与外模式之间的映射 B 模式与内模式之间的映射
C 模式 D 三级模式

下列四项中，不属于数据库系统特点的是（ ）。

- A 数据共享 B 数据完整性 C 数据冗余度高 D 数据独立性高

数据库 (DB)，数据库系统 (DBS)和数据库管理系统 (DBMS)之间的关系是（ ）。

- A DBS包括 DB和 DBMS B DBMS 包括 DB和 DBS
C DB包括 DBS和 DBMS D DBS 就是 DB, 也就是 DBMS

数据库系统与文件系统的主要区别是（ ）。

- A 数据库系统复杂，而文件系统简单
B 文件系统不能解决数据冗余和数据独立性问题，而数据库系统可以解决
C 文件系统只能管理程序文件，而数据库系统能够管理各种类型的文件
D 文件系统管理的数据量较少，而数据库系统可以管理庞大的数据量。

数据库的概念模型独立于（ ）。

- A 具体的机器和 DBMS B E-R 图 C 信息世界 D 现实世界

在数据库中存储的是（ ）。

- A 数据 B 数据模型 C 数据以及数据之间的联系 D 信息

在数据库中，数据的物理独立性是指（ ）。

- A 数据库与数据库管理系统的相互独立
B 用户程序与 DBMS的相互独立
C 用户的应用程序与存储在磁盘上的数据库中的数据是相互独立的
D 应用程序与数据库中数据的逻辑结构相互独立

数据库的特点之一是数据的共享，严格地讲，这里的数据共享是指（ ）。

- A 同一应用中的多个程序共享一个数据集合
B 多个用户、同一种语言共享数据
C 多个用户共享一个数据文件
D 多种应用、多种语言、多个用户相互覆盖地使用数据集合

在数据库技术中，为提高数据库的逻辑独立性和物理独立性，数据库的结构被划分成用户级、（ ）和存储级三个层次

- A 管理员级 B 外部级 C 概念级 D 内部级

在数据库中，产生数据不一致的根本原因是（ ）。

- A 数据存储量太大 B 没有严格保护数据
C 未对数据进行完整性控制 D 数据冗余

数据库具有 、最小的 和较高的程序与数据 。

- A 程序结构化 B 数据结构化 C 程序标准化 D 数据模块化

- A 冗余度 B 存储量 C 完整性 D 有效性

- A 可靠性 B 完整性 C 独立性 D 一致性

在数据库的三级模式结构中，描述数据库中全体数据的逻辑结构和特征的是（ ）。

- A 外模式 B 内模式 C 存储模式 D 模式

1-5 : A, C, C, A, C

6-10 : A, B, A, C, C

11-15 : D, C, D, BAC,D

1.5 对于工程硕士的管理需要掌握的信息有： 学生现在的工作单位、职务、 简历情况， 其中简历情况包括开始时间、终止时间、单位、担任职务、证明人；学生目前在校情况，包括学生的学号、 所在学院、 所学专业、 入学时间、 导师；学生在校所学的课程号、 课程名、 学时、授课教师及成绩；学院包括学院代号、学院名称、院长；导师包括导师职工编号、姓名、出生日期、职称、研究方向。

要求：

确定实体及实体型属性；

找出实体间的联系；

画出 E-R 图。

(1) 学生：学号，姓名，专业，学院，入学时间，工作单位，导师号，建立号，职务

简历：简历号，开始时间，结束时间，证明人，单位，担任职务

课程：课程号，学时，课程名，授课教师

学院：学院代号，学院名称，院长

导师：导师号，姓名，出生日期，研究方向

(2) 学生 -简历： 1 : n

学生 -学院： n : 1

学生 -导师： n : 1

学生 -课程： n : m

导师 -学院： n : 1

(3)

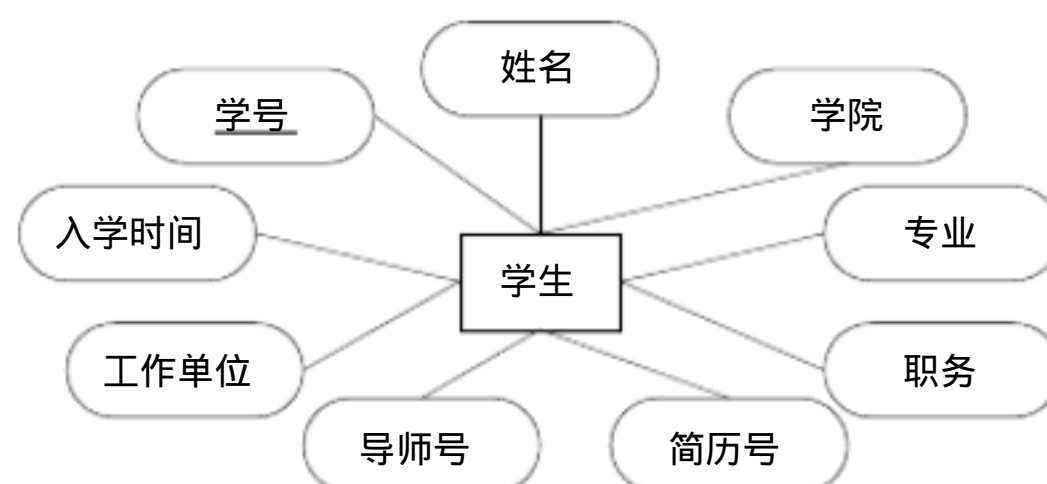


图 1 学生实体及其属性

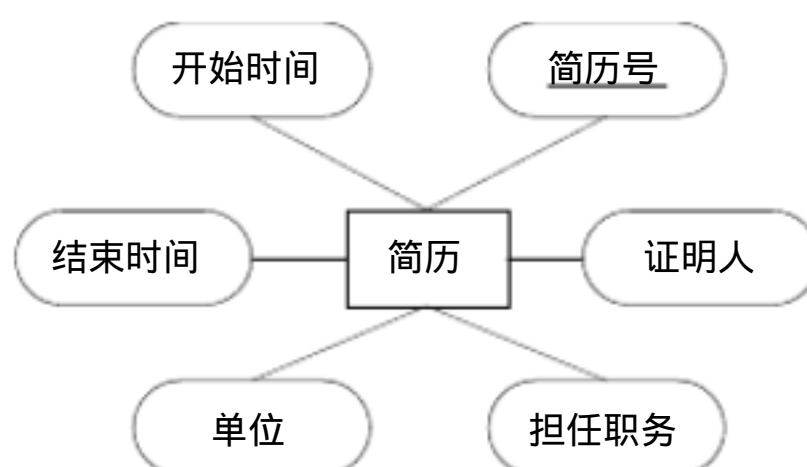


图 2 简历实体及其属性

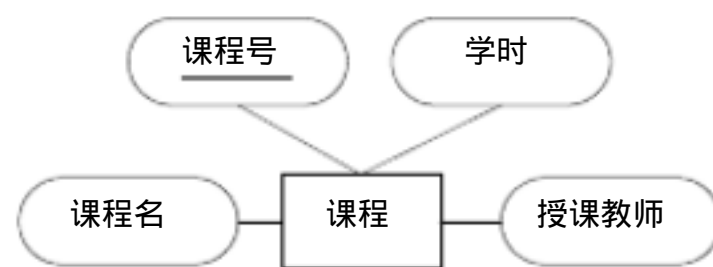


图 3 课程实体及其属性

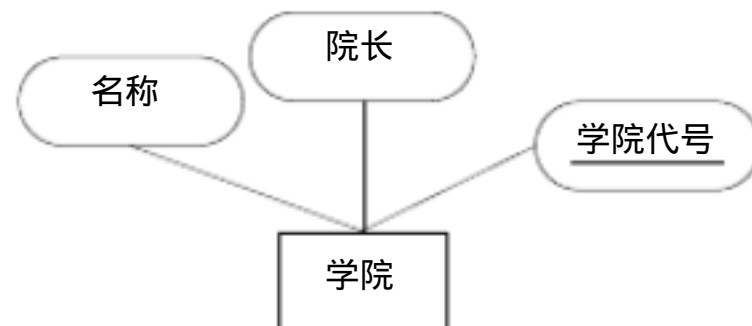


图 4 学院实体及其属性

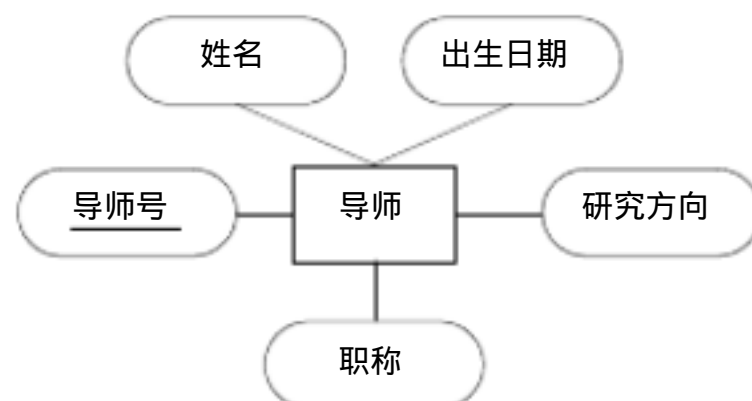


图 5 导师实体及其属性

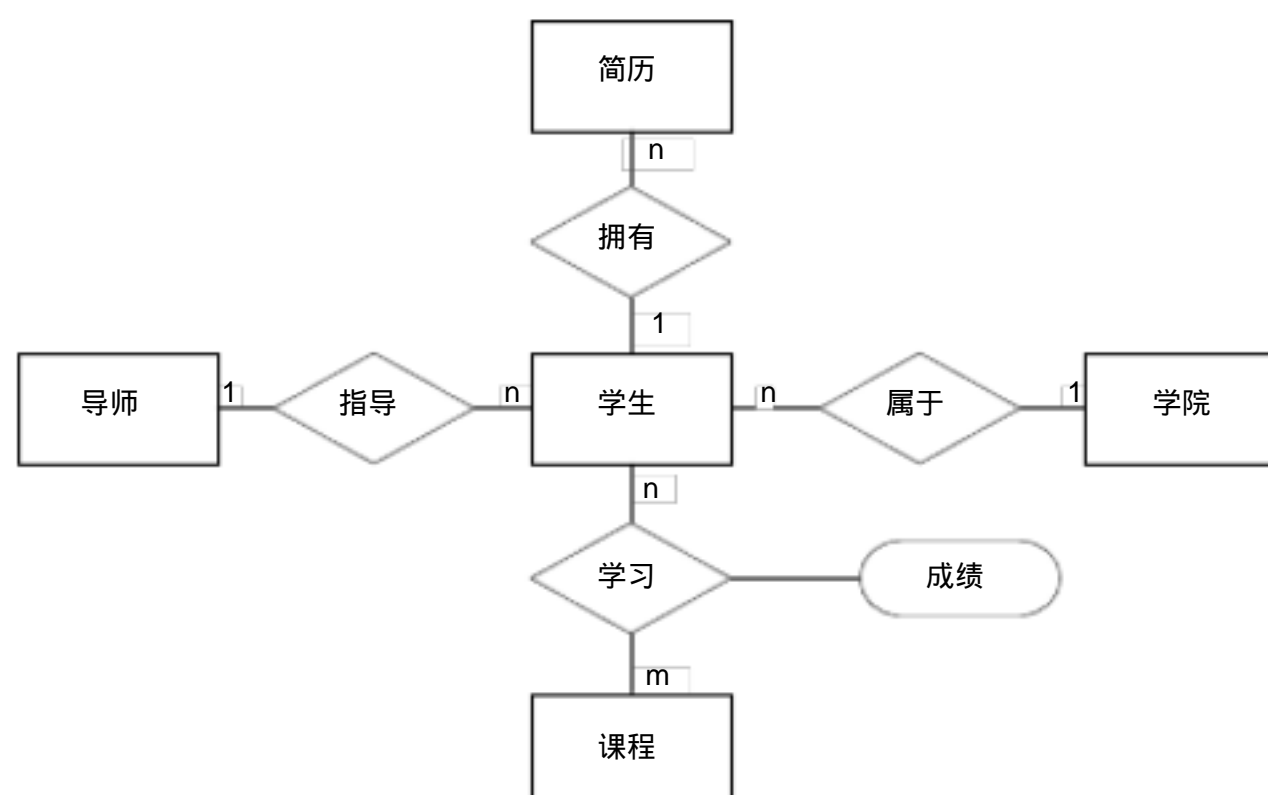


图 6 实体联系图

第二章

2.1 试述关系模型的 3 个组成部分

答：关系模型由关系数据结构、关系操作和关系完整性约束三部分组成。

2.2 一个关系模式能否没有关键字？为什么？

答：不可以没有关键字，因为关键字是元组的唯一标志，如果没有关键字则无法唯一标识一个元组。

2.3 试述关系模型的完整性规则。在参照完整性中，为什么外部关键字属性的值也可以为空？什么情况下才可以为空？

答：(1) 实体完整性规则：若属性 A 是基本关系 R 的主属性，则属性 A 不能取空值。

(2) 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。即属性 F 本身不是主属性，则可以取空值，否则不能取空值。

当外部关键字 F 对应的主键 Ks 值为空值时，说明 F 暂时没有赋值，此时 F 值必须取空值。

2.4 一个 n 行、m 列的二维表 (其中没有两行或两列全同)，将行或列交换后，能导出多少个不同的表？

答：1 个

2.5

答：

(1) 最小个数为：取 m,n 中的大者；最大个数为：m+n

(2) 最小个数为：0；最大个数为：取 m,n 中的小者

(3) 最小个数为：0；最大个数为：m*n

(4) 最小个数为：0；最大个数为：n

2.6

答：

$R \cup S = \{(3,6,7), (2,5,7), (7,2,3), (4,4,3), (3,4,5)\}$

$R - S = \{(3,6,7), (2,5,7), (4,4,3)\}$

$R \cap S = \{(7,2,3)\}$

$R \times S = \{(3,6,7,3,4,5), (3,6,7,7,2,3), (2,5,7,3,4,5), (2,5,7,7,2,3), (7,2,3,7,2,3), (7,2,3,3,4,5), (4,4,3,3,4,5), (4,4,3,7,2,3)\}$

$\Pi_{3,2}(S) = \{(5,4), (3,2)\}$

$\sigma_{B < '5'}(R) = \{(7,2,3), (4,4,3)\}$

2.7

答：该表达式的意思是：从关系 R 中选择元组，该元组满足：第 1 分量值或第 2 分量值至少有一个不等于其他某元组，即选择与其他元组不完全相同的元组。由于关系模型的性质要求，关系中不能够有重复的元组，因此上述条件一定满足，所以，这个表达式结果就是 R。

2.8

(1) 从 R 和 S 的笛卡尔积中选择满足下列条件的元组：

第 2 分量与第 4 分量相等，或第 3 分量和第 4 分量相等，并取第 1 分量和第 5 分量构成新的关系。

(2) 元组表达式

$\{t \mid (\exists u)(\exists v)(R(u) \wedge S(v) \wedge u[2] = v[1] \wedge u[3] = v[1] \wedge t[1] = u[1] \wedge t[2] = v[2])\}$

(3) 域表达式 $\{xv | (\exists y)(\exists z)(R(xyz) \wedge S(uv) \wedge (y = u \vee z = u))\}$

2.9

(1) 汉语表达式：选择 R 关系中元组第 2 分量值不等于 S 关系中某元组第 1 分量值的元组。

(2) 关系代数表达式： $\pi_{1,2}(\sigma_{2 \neq 1}(R \times S))$

(3) 域表达式： $\{xy | (\exists u)(\exists v)(R(xy) \wedge S(uv) \wedge (u \neq y))\}$

2.10

(1) 汉语表达式：选择 R 中元组第 1 分量值与第 2 分量值互换后仍存在于 R 中的元组。

(2) 关系代数表达式： $\pi_{1,2}(\sigma_{1=2}(R \times R))$

(3) 域表达式： $\{t | (\exists u)(R(t) \wedge R(u) \wedge t[1] = u[2] \wedge t[2] = u[1])\}$

2.11

(1) $\{t | (\exists u)(R(u) \wedge t[1] = u[1])\}$

(2) $\{t | (R(t) \wedge t[2] = '17')\}$

(3)

$\{t | (\exists u)(\exists v)(R(u) \wedge S(v) \wedge t[1] = u[1] \wedge t[2] = u[2] \wedge t[3] = u[3] \wedge t[4] = v[1] \wedge t[5] = v[2] \wedge t[6] = v[3])\}$

(4) $\{t | (\exists u)(\exists v)(R(u) \wedge S(v) \wedge u[3] = v[1] \wedge t[1] = u[1] \wedge t[2] = v[3])\}$

2.12

设有关系：S(SNO, SNAME, AGE, SEX, CDEPT);

SC(SNO, CNO, GRADE);

C(CNO, CNAME, CDEPT, TNAME)用关系代数表示

(1) 刘老师所教授课程的课程号、课程名

$CNO, CNAME(\pi_{TNAME='刘'}(C))$

(2) 检索年龄大于 23 的男学生的学号和姓名

$SNO, SNAME(\pi_{AGE>23, SEX='男'}(S))$

(3) 检索学号为 '090111' 学生所学课程的课程名与任课教师名

$CNAME, TNAME(\pi_{CNO='090111'}(SC \Join C))$

(4) 检索至少选修刘老师所教授课程中一门课的女学生姓名

$SNAME(\pi_{TNAME='刘'}(C) \Join SC \Join \pi_{SEX='女'}(S))$

(5) 检索 '王一新' 同学不学课程的课程号

$CNO(C) - CNO(\pi_{SNAME='王一新'}(S) \Join SC)$

(6) 检索至少选了两门课的学生学号

$SNO(\pi_{[1]=[4], [2]<=>[5]}(SC \times SC))$

(7) 检索全部学生都选修的课程的课程号和课程名

CNO, CNAME(C (SNO, CNO(SC) ÷ SNO(S)))

(8) 检索选修课程包括刘老师所教授课程的学生姓名

SNAME(SC TNAME=刘' (C))

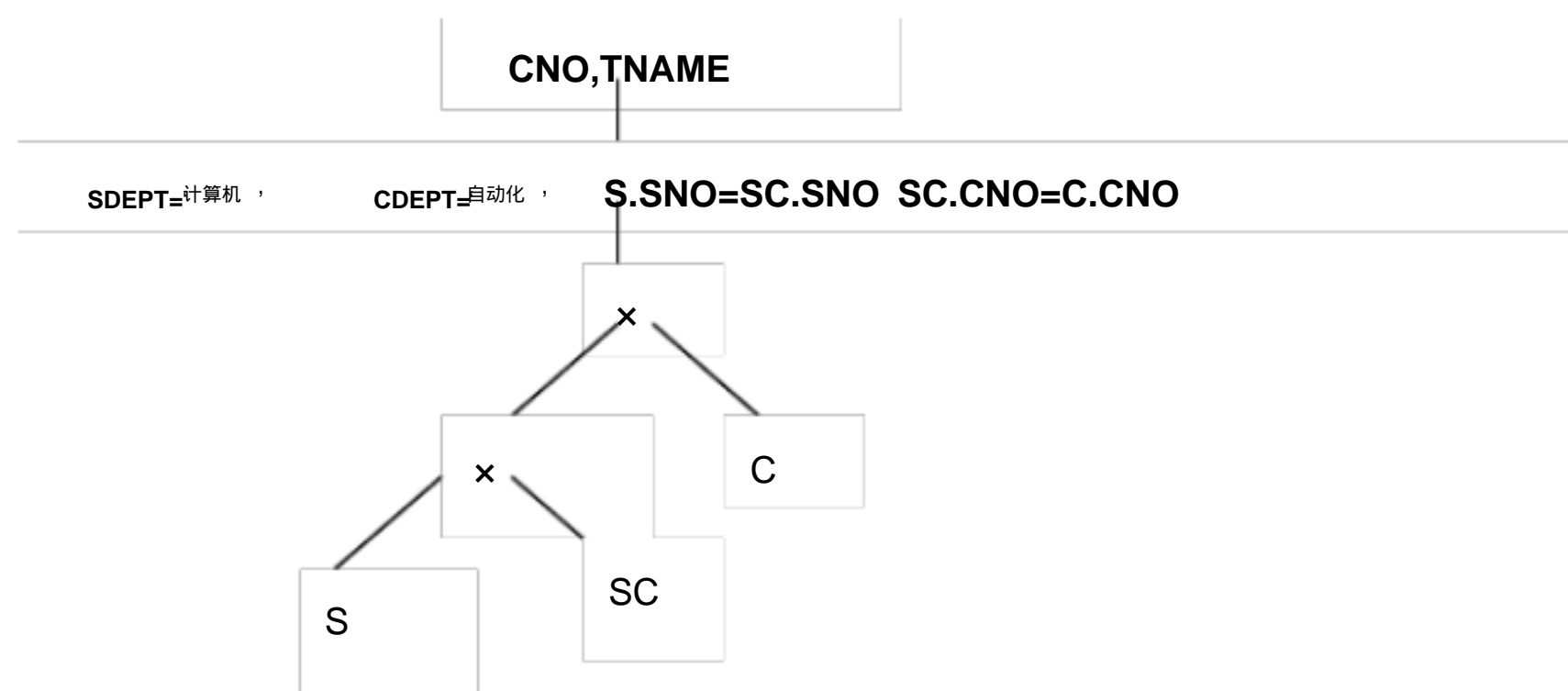
2.13

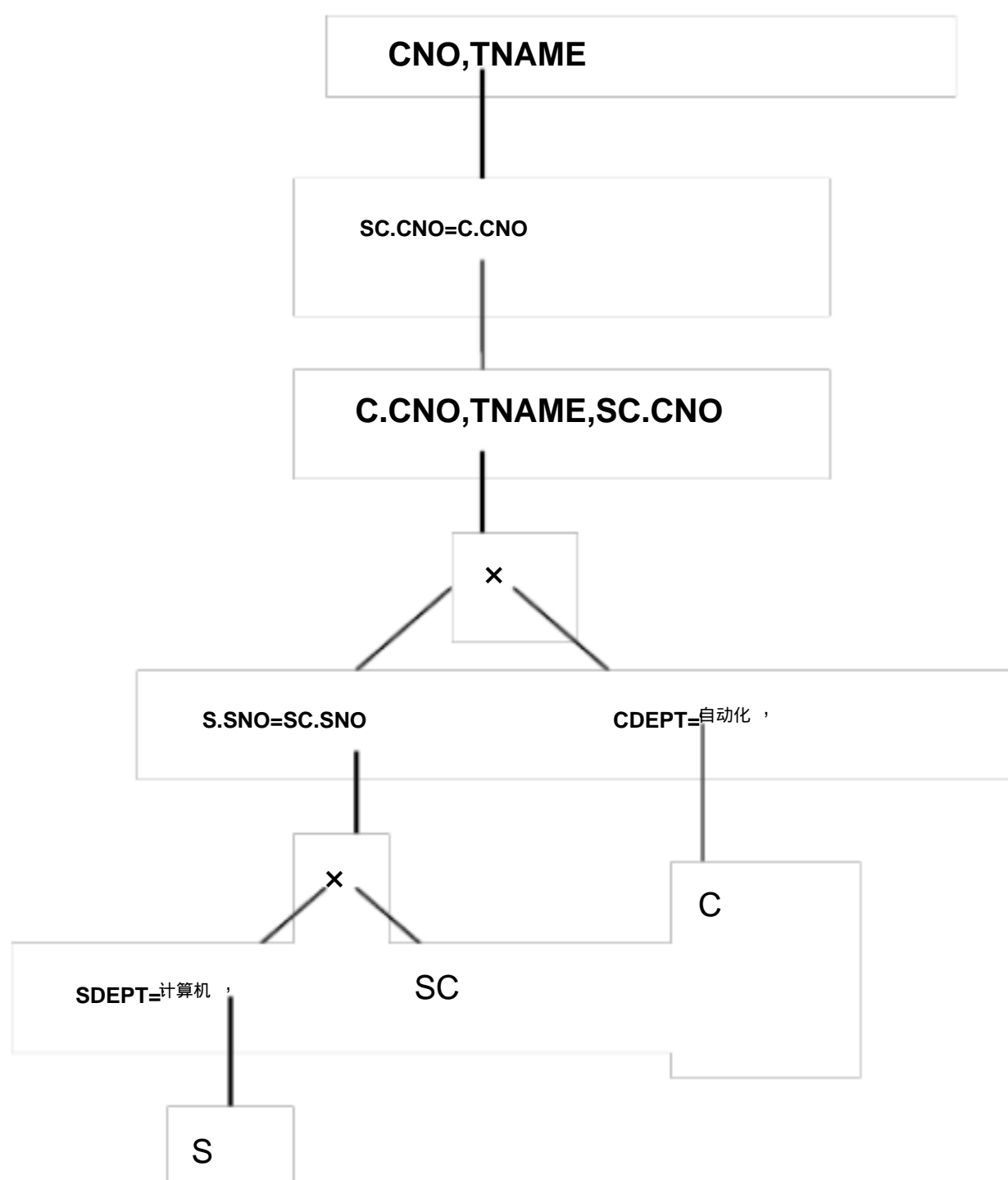
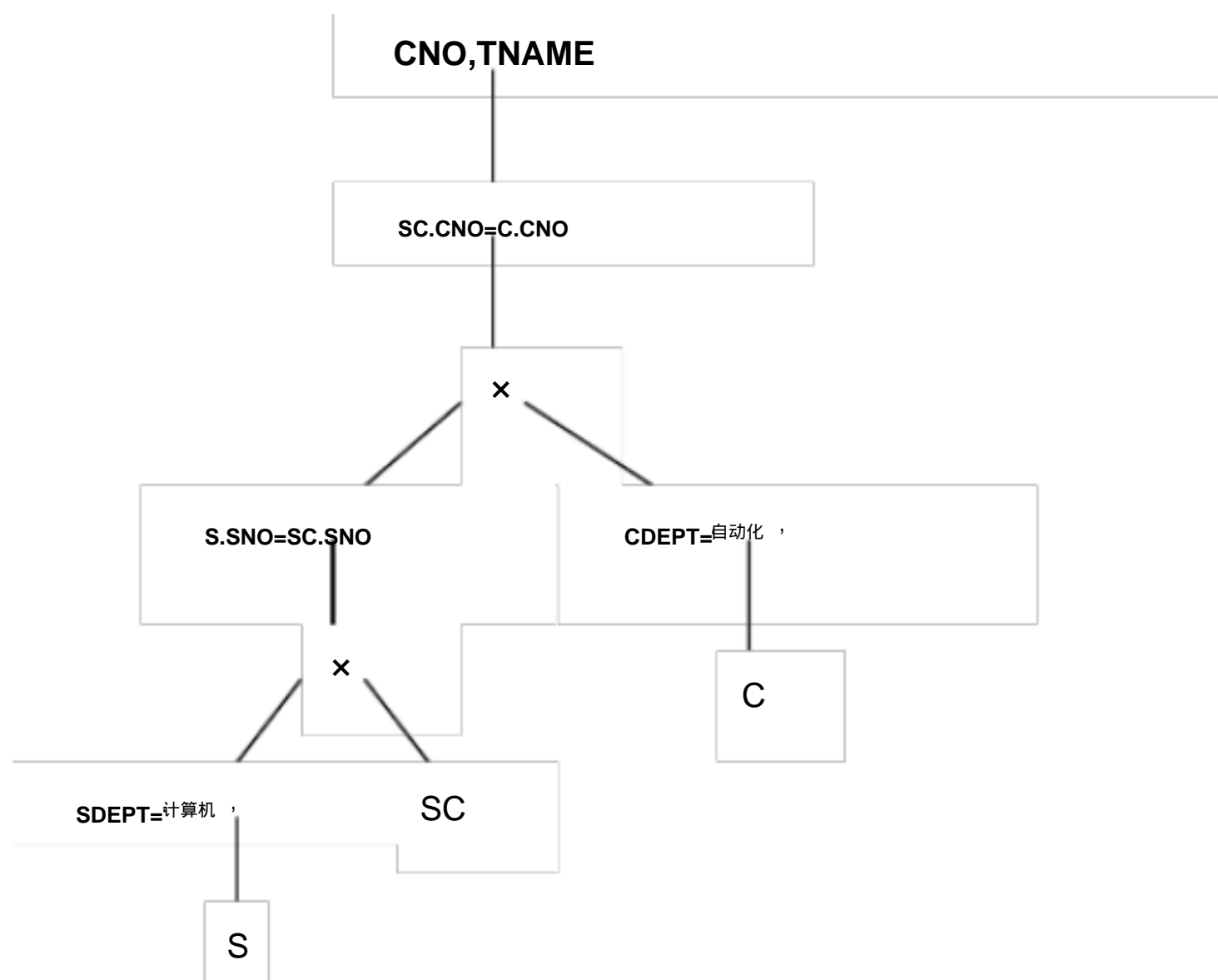
- (1) 刘老师所教授课程的课程号、课程名
- {t|(∃u)(C(u) ∧ t[1]=u[1] ∧ t[2]=u[2] ∧ C[3]= '刘')}
- (2) 检索年龄大于 23 的男学生的学号和姓名
- {t|(∃u)(S(u) ∧ t[1]=u[1] ∧ t[2]=u[2] ∧ u[3]> 23 ∧ u[4]= '男')}
- (3) 检索学号为 ' 090111 ' 学生所学课程的课程名与任课教师名
- {t|(∃u)(∃v)(C(v) ∧ SC(u) ∧ v[1]=u[2] ∧ t[1]=v[2] ∧ t[2]=v[3] ∧ v[1]= '090111')}
- (4) 检索至少选修刘老师所教授课程中一门课的女学生姓名
- {t|(∃u)(∃v)(∃w)(S(u) ∧ SC(v) ∧ C(w) ∧ u[4]= '女' ∧ v[1]=u[1] ∧ v[2]=w[1] ∧ w[3]= '刘' ∧ t[1]=u[2])}
- (5) 检索 ' 王一宁 ' 同学不学课程的课程号
- { t |(∃u)(∀v)C(u) ∧ SC(v) ∧ (u[1] = v[2] ⇒ (∃w)(s(w) ∧ w[1] = v[1] ∧ w[2] ≠ '王一宁')) ∧ t[1] = u[1]}
- (6) 检索至少选了两门课的学生学号
- (7) 检索全部学生都选修的课程的课程号和课程名
- { t |(∀u)(∃v)(∃w)(S(u) ∧ SC(v) ∧ C(w) ∧ u[1] = v[1] ∧ v[2] = w[1] ∧ t[1] = v[1] ∧ t[2] = v[2])}

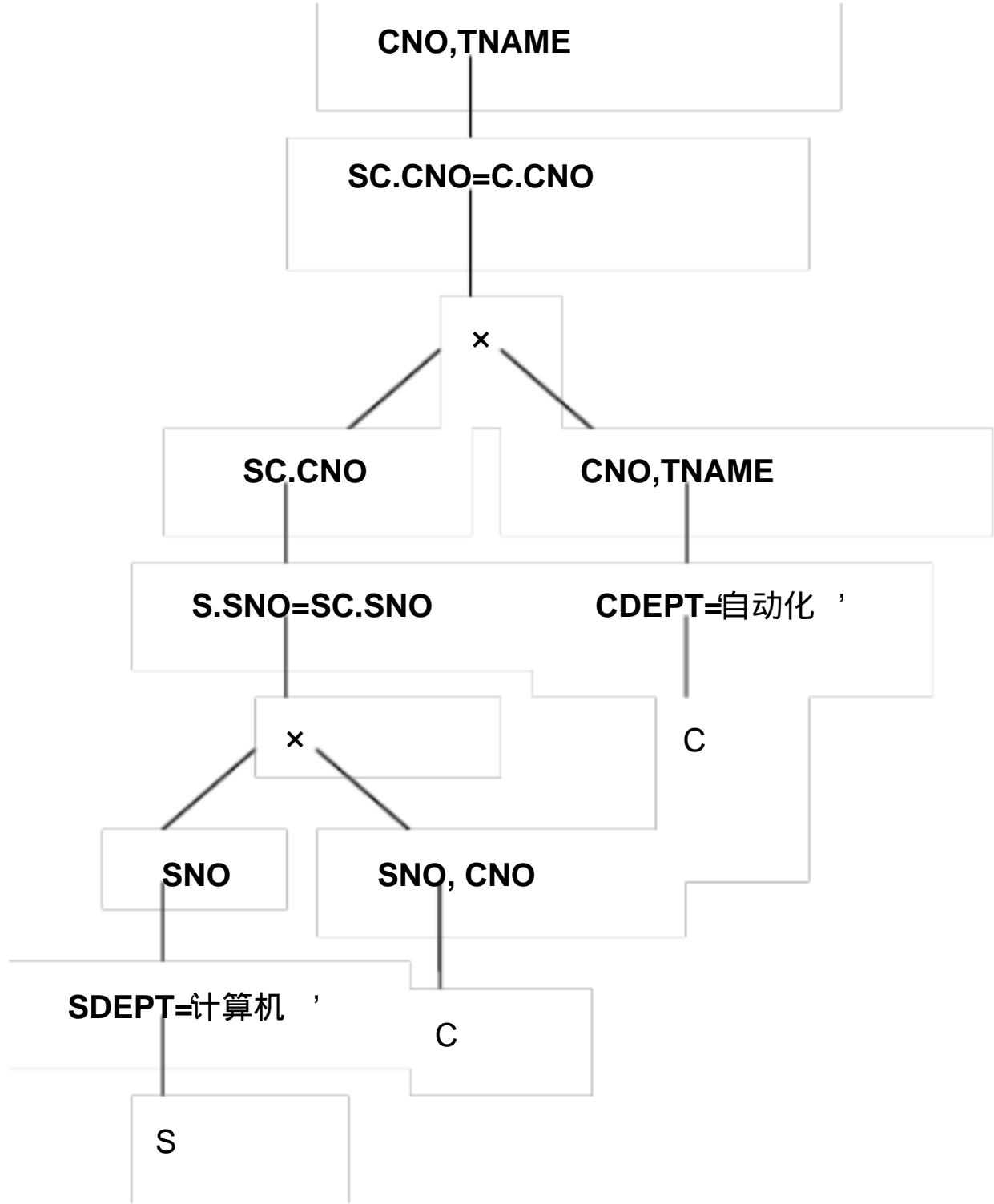
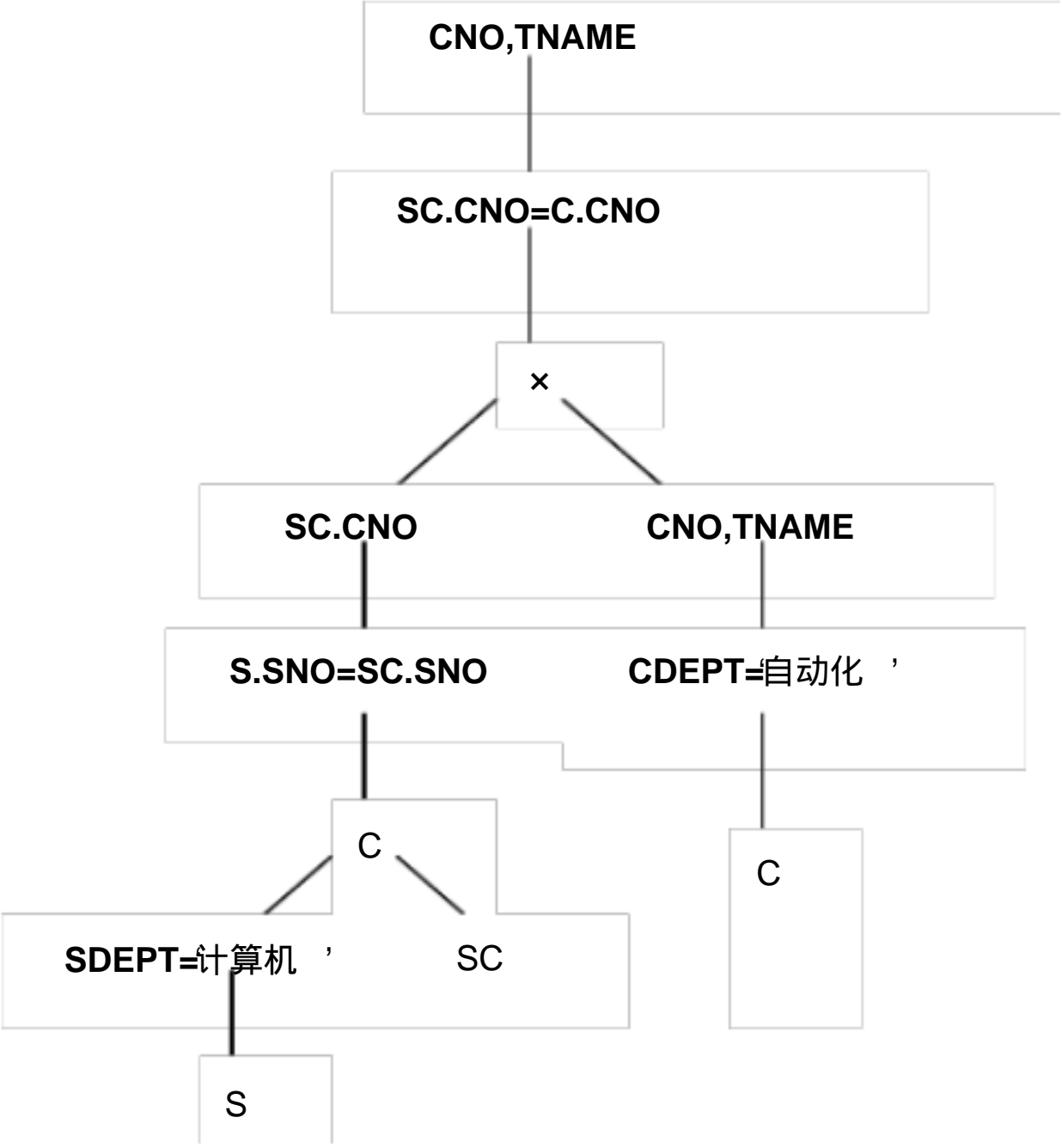
- (8) 检索选修课程包括刘老师所教授课程的学生姓名
- { t |(∃u)(∃v)(SC(u) ∧ C(v) ∧ u[2] = v[1] ∧ v[3] = '刘老师 ' ∧ t[1] = u[1])}

2.15

- (1) CNO,TNAME (SDEPT=计算机' (S) SC CDEPT=自动化' (C))
- (2) CNO,TNAME (SDEPT=计算机' CDEPT=自动化' S.SNO=SC.SNO SC.CNO=C.CNO (S× SC × C))
- (3)







2.16

在关系代数运算中，各个运算所费时间和空间是不一样的，如何安排若干关系的运算操作步骤，直接影响到整个操作所需要的时间和空间。对关系代数表达式进行优化，可以提高系统的操作效率，达到执行过程既省时间又省空间的目的。

第三章

3-1 解释下列术语：

基本表 视图 单表查询 连接查询 嵌套查询 交互式 SQL 嵌入式 SQL 宿主语言

答：

- (1) 基本表：简称基表。是数据库中实际存在的表，在 SQL 中一个关系对应于一个表。
- (2) 视图：SQL 用视图概念支持非标准的外模式概念。视图是从一个或几个基表导出的表，虽然它也是关系形式，但它本身不实际存储在数据库中，只存放对视图的定义信息（没有对应的数据）。因此，视图是一个虚表（Virtual Table）或虚关系，而基表是一种实关系（Practical Relation）。
- (3) 单表查询：只涉及到一个关系的查询。
- (4) 连接查询：同时涉及两个以上的表的查询。
- (5) 嵌套查询：将一个查询块嵌套在另一个查询块的 WHERE 子句或 HAVING 条件中的查询。
- (6) 交互式 SQL：做为独立语言由用户在交互环境下使用的 SQL 语言。
- (7) 嵌入式 SQL：将 SQL 语言嵌入到某种高级语言中使用，利用高级语言的过程性结构来弥补 SQL 语言在实现复杂应用方面的不足的 SQL 语言。
- (8) 宿主语言：嵌入 SQL 的高级语言称为主语言或宿主语言。

3-2 试述 SQL 语言的特点

答：1. 综合统一

2. 高度非过程化

3. 面向集合的操作方式

4. 以同一种语法结构提供两种使用方式

、

3-3 试述 SQL 体系结构和关系数据库模式之间的关系

答：

SQL 语言支持关系数据库体系结构，即外模式、模式和内模式，利用 SQL 语言可以实现对三级模式的定义、修改和数据的操纵功能，在此基础上形成了 SQL 体系结构；在 SQL 体系结构中的视图对应了关系数据模式的外模式，基本表对应了关系数据模式的模式，存储文件对应了关系数据模式的内模式。

3-4 SQL 是如何实现实体完整性、参照完整性和用户定义完整性的？

答：

实体完整性简单的说，就是将表中的每一行看作一个实体。实体完整性要求表的标示符列或主键的完整性。可以通过建立唯一索引、PRIMARY KEY 约束、UNIQUE 约束，以及列的 IDENTITY 属性来实施实体完整性。

参照完整性维持被参照表和参照表之间的数据一致性，他通过主键（PRIMARY KEY）约束和外键（FOREIGNKEY）约束来实现。引用完整性确保键值在所有表中一致。这样的一致性要求不能引用不存在的值，如果键值更改了，那么在整个数据库中，对该键值的所有引用要进行一致的更改。在被参照表中，当其主键值被其他表所参照时，该行不能被删除也不允许改变。在参照表中，不允许参照不存在的主键值。

用户定义完整性使得可以定义不属于其它任何完整性分类的特定业务规则。所有的完整性类型都支持用户定义完整性。

3-5 讨论当对一个视图进行更新的时候可能会出现什么样的问题？

答：

- 1) 若视图的属性来自属性表达式或常数，则不允许对视图执行 INSERT和 UPDATE操作，但允许执行 DELETE操作。
- 2) 若视图的属性来自库函数，则不允许对此视图更新。
- 3) 若视图定义中有 GROUP BY子句，则不允许对此视图更新。
- 4) 若视图定义中有 DISTINCT任选项，则不允许对此视图更新。
- 5) 若视图定义中有嵌套查询，并且嵌套查询的 FROM 子句涉及导出该视图的基本表，则不允许对此视图更新。
- 6) 若视图由两个以上的基本表导出，则不允许对此视图更新。
- 7) 如果在一个不允许更新的视图上再定义一个视图，这种二次视图是不允许更新的。

3-6 设有两个基本表 R(A,B,C) 和 S(A,B,C)试用 SQL查询语句表达下列关系代数表达式：

$R \cap S$ $R - S$ $R \cup S$ $R \times S$

- (1) (select * from R) INTERSECT (select * from S)
- (2) (select * from R) EXCEPT (select * from S)
- (3) (select * from R) UNION (select * from S)
- (4) select * from R,S

3-7 对于教学数据库的三个基本表：

S(学号, 姓名, 年龄, 性别)

SC(学号, 课程号, 成绩)

C(课程号, 课程名, 任课教师姓名)

试用 SQL语句表达下列查询：

查询“刘某”老师所授课程的课程号和课程名。

查询年龄大于 23 岁的男同学的学号和姓名。

查询学号为 S3 的学生所学课程的课程号、课程名和任课教师姓名。

查询“张小飞”没有选修的课程号和课程名。

查询至少选修了 3 门课程的学生的学号和姓名。

查询全部学生都选修了的课程编号和课程名称。

在 SC中删除尚无成绩的选课元组。

把“高等数学”课的所有不及格成绩都改为 60。

把低于总评均成绩的女同学的成绩提高 5%。

向 C 中插入元组 (' C8 ', ' VC++ ', ' 王昆 ')。

3-7

(1) 查询“刘某”老师所授课程的课程号和课程名；

SELECT课程号, 课程名

FROM C

WHERE任课教师姓名 LIKE 刘**

(2) 查询年龄大于 23 岁的男同学的学号和姓名；

SELECT学号,姓名

FROM S

WHERE 年龄 >23 AND 性别 ='男'

(3) 查询学号为 S3 的学生所学课程和课程号 ,课程名和任课教师姓名

```
SELECT C课程号 , 课程名 , 任课教师姓名
FROM C, SC
WHERE SC学号 ='S3' And C课程号 =SC课程号
```

(4) 查询 '张小飞 '同学没有选修的课程的课程号和课程名 ;

```
SELECT C课程号 , C.课程名
FROM C
WHERE 课程号 NOT IN
(SELECT课程号
FROM SC
WHERE 学号 IN
(SELECT学号 FROM S
WHERE 姓名 ='张小飞 '));
```

或者

```
SELECT *
FROM C
WHERE NOT EXISTS
(SELECT SC课程号
FROM SC,S
WHERE SC课程号 =C.课程号 AND S学号 =SC学号
AND S.姓名 ='张小飞 ')
```

或者

```
SELECT *
FROM C
WHERE 课程号 NOT IN
(SELECT课程号
FROM SC,S
WHERE S学号 =SC学号 AND S.姓名 ='张小飞 ')
```

(5) 查询至少选修了 3 门课程的学生的学号和姓名 ;

```
SELECT学号 , 姓名
FROM S
WHERE 学号 IN
(SELECT学号
FROM SC
GROUP BY学号
HAVING COUNT (课程号 )>=3)
```

(6) 查询全部学生都选修了的课程号和课程名 ;

```
SELECT课程号 ,课程名
FROM C
WHERE NOT EXISTS
(SELECT学号
```

```

FROM S
WHERE NOT EXISTS
(SELECT学号
FROM SC
WHERE SC课程号 =C课程号  AND S学号 =SC学号 ))
或者
SELECT课程号 , 课程名
FROM C
WHERE NOT EXISTS
((SELECT学号
FROM S)
EXCEPT
(SELECT学号
FROM SC
WHERE SC课程号 =C课程号 ))

```

(7) 在 SC表中删除无成绩的选课纪录 ;

```

DELETE *
FROM SC
WHERE GRADE is null;

```

(8) 把 ‘高等数学 ’的所有不及格成绩都改为 60 分 ;

```

UPDATE SC
SET 成绩 = 60
WHERE GRADE< 60 AND课程号 =
(SELECT课程号
FROM C
WHERE 课程名 ='高等数学 ')

```

(9) 把低于总平均成绩的女同学的成绩提高 5%

```

UPDATE sc
SET 成绩 = 成绩 *(1+0.05)
WHERE 学号 in
(select 学号 from s where 性别 ='女') and 成绩 < (select avg(成绩 ) from sc);

```

(10) 向 C 中插入元组 (“ C8”, “ VC++”, “ 王昆 ”)

```

INSERT INTO c
VALUES ('C8', 'VC++',王昆 ');

```

3-8 图 3-2 中显示的数据库模式 Library , 用来记录书籍、借书人和书籍借出的情况 , 参照完整性在图中用有向弧来表示 , 请用 SQL语言建立图中的关系模式 , 并完成下列操作 :

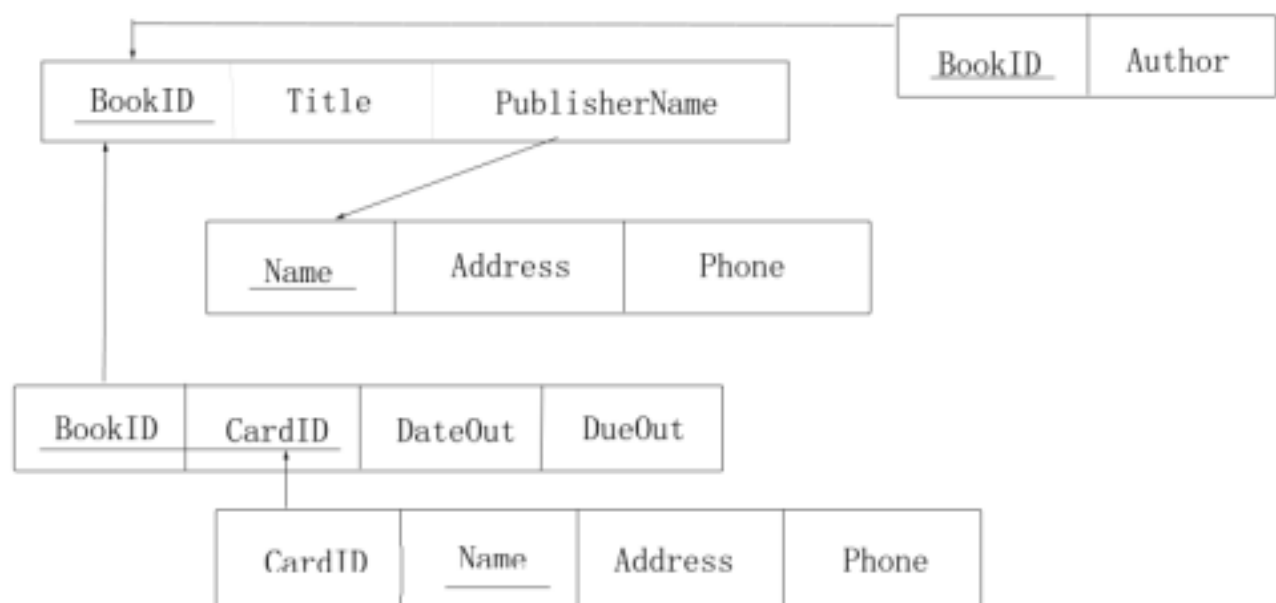


图 3-2 Library 数据库模式

查询“高等教育出版社”出版的所有图书名称和编号；

查询所有作者是“郭雨辰”的图书的编号和名称；

查询“李明”借过的所有图书的名称；

查询“李明”从 2008 年 1 月 1 日到 2009 年 1 月 1 日期间借过的图书名称；

列出从 2008 年 1 月 1 日到 2009 年 1 月 1 日期间没有被人借过的图书名称和编号；

列出从 2008 年 1 月 1 日到 2009 年 1 月 1 日期间借出次数最多的 10 本图书名称；

将“高等教育出版社”的电话改为“ 010 - 64054588 ”；

3-8

(1)

```
SELECT BookID , Title
FROM Book
WHERE PublishName=高等教育出版社 ,
```

(2)

```
SELECT BookID, Title
FROM BookID, Book
WHERE Book.BookID=Author.BookID AND Author郭雨辰 ,
```

(3)

```
SELECT Title
FROM Book, Borrow, Borrower
WHERE Borrow.CardID=Borrower.CardID AND Borrow.BookID=Book.BookID AND
Borrower.Name= 李明 ,
```

(4)

```
SELECT Title
FROM Book, Borrow, Borrower
WHERE Borrow.CardID=Borrower.CardID AND Borrow. BookID=Book.BookID AND
Borrower.Name= 李明 'AND DateOut BETWEEN 2008-1-1 AND.2009-1-1
```

(5)

```
SELECT Title
```

```
FROM Book
WHERE NOT EXISTS
    (SELECT *
     FROM Borrow,Book
     WHERE Borrow. BookID=Book.BookID AND Dateout Between 2008-1-1 AND 2009-1-1)
```

(6)

```
SELECT top 10  Title
FROM Book, Borrow
WHERE Borrow. BookID=Book.BookID AND Dateout Between 2008-1-1 AND 2009-1-1
GROUP BY Book.BookID
ORDER BY Book.BookID DESC
```

(7)

```
UPDATE Publisher
SET Phone=010-64054588 '
WHERE Name=高等教育出版社 '
```

3-9 针对 3.8 建立的表，用 SQL语言完成下列操作：

把对所有表的 INSERT权限授予“张丽”，并允许她将此权限授予其他用户；

Grant INSERT to 张丽 'with grant option

把查询和修改 BORROWER的权限受给用户“王伟”。

Grant select, update on BORROWER to王伟 '

习题四

4.1 理解并给出下列术语的定义： 函数依赖 部分函数依赖 完全函数依赖 传递函数依赖
候选码 主码 外码 全码 主属性 非主属性 1NF 2NF 3NF BCNF 4NF 函数依赖集
闭包 属性集闭包 函数依赖集等价 最小函数依赖集 无损连接 函数依赖保持

设 $R(U)$ 是属性集 U 上的关系模式。若对于 $R(U)$ 的任意一个可能的关系 r , X, Y 是属性集 U 的任意子集, 当且仅当对 r 中任意一个给定的 X 的属性值, r 中都只存在惟一的 Y 属性值与之对应。也就是说, 如果 X 相等, 就有 Y 也相等, 则称 Y 函数依赖于 X 或 X 函数确定 Y , 记作 $X \rightarrow Y$ 。

在 $R(U)$ 中, 如果 $X \rightarrow Y$, 并且对于 X 的一个真子集 X' , 有 $X' \rightarrow Y$ 成立, 则称 Y 对 X 部分函数依赖 (Partial Functional Dependency), 记作 $X \xrightarrow{p} Y$ 。

在 $R(U)$ 中, 如果 $X \rightarrow Y$, 并且对于 X 的任何一个真子集 X' , 都有 $X' \not\rightarrow Y$ 成立, 则称 Y 对 X 完全函数依赖 (Full Functional Dependency), 记作 $X \xrightarrow{f} Y$ 。

在 $R(U)$ 中, 如果 $X \rightarrow Y, Y \rightarrow X, Y \rightarrow Z$, 则称 Z 对 X 传递函数依赖 (Transitive Functional Dependency), 记做 $X \xrightarrow{t} Z$ 。

设 K 为 $R\langle U, F \rangle$ 中的属性或属性组, 若 $K \xrightarrow{f} U$, 则 K 为 R 的候选码。若候选码多于一个, 则选定其中的一个为主码。包含在任何一个候选码中的属性, 叫做主属性。不包含在任何候选码中的属性称为非主属性。最简单的情况, 码只包含单个属性; 最复杂的情况是所有属性集组合成码, 称为全码。关系模式 R 中属性或属性组 X 并非 R 的主码, 但 X 是另一个关系模式的主码, 则称 X 是 R 的外码。

设 R 是一个关系模式, 如果 R 中的每一个属性 A 的属性名和属性值都是不可再分的, 则称 R 属于第一范式, 记作: $R \in 1NF$ 。

若 $R \in 1NF$, 且每一个非主属性都完全函数依赖于码, 则 $R \in 2NF$ 。

关系模式 $R(U, F)$ 中若不存在这样的码 X , 属性组 Y 及非主属性组 Z ($Z \not\subseteq Y$), 使得 $X \rightarrow Y, Y \rightarrow X, Y \rightarrow Z$ 成立, 则称 $R(U, F) \in 3NF$ 。

关系模式 $R(U, F) \in 1NF$, 若每一个决定因素都含有码, 则 $R \in BCNF$ 。

关系模式 $R(U, F) \in 1NF$, 若对 R 的每个非平凡多值依赖 $X \twoheadrightarrow Y$ ($Y \not\subseteq X$), X 都包含码, 则称 $R(U)$ 满足第四范式, 记为 $R \in 4NF$ 。

称所有被一个已知函数依赖集 F 逻辑蕴涵的那些函数依赖的集合为 F 的闭包 (Closure), 记为 F^+ 。

设有关系模式 $R(U)$, F 是 U 上的一个函数依赖集, $X \subseteq U$, 定义

$$X_F^+ = \{ A \mid X \rightarrow A \text{ 能由 } F \text{ 根据 Armstrong 公理导出} \},$$

并称 X_F^+ 为属性集 X 关于函数依赖集 F 的闭包。

如果函数依赖集 F 满足下列条件, 则称 F 是一个极小函数依赖集或最小覆盖。

F 中每一个函数依赖的右部都是单个属性。

对 F 中任一函数依赖 $X \rightarrow A$, $F - \{X \rightarrow A\}$ 都不与 F 等价。

对于 F 中的任一函数依赖 $X \rightarrow A$, $\{F - \{X \rightarrow A\}\} \cup \{Z \rightarrow A\}$ 都不与 F 等价, 其中 Z 为 X 的任一子集。

如果函数依赖集 F 与某个最小依赖集 F_m 等价, 则称 F_m 是 F 的最小覆盖或 F_m 是 F 的最小依赖集。

设 R 是一个关系模式, F 是 R 上的一个函数依赖集, R 分解为关系模式的集合 $\rho = \{R_1(U_1), R_2(U_2), \dots, R_n(U_n)\}$ 。如果对于 R 的满足 F 的每一个关系 r , 都有 $r = \prod_{R_1}(r) \bowtie \prod_{R_2}(r) \bowtie \dots \bowtie \prod_{R_n}(r)$, 则称 ρ 是一个无损连接的分解 (lossingless join decomposition)

若 $F^+ = (\bigcup_{i=1}^k F_i)^+$, 则 $R(U, F)$ 的分解 $\rho = \{R_1(U_1, F_1), \dots, R_k(U_k, F_k)\}$ 保持函数依赖。

4.2 设有关系模式 $R(A, B, C, D, E, P, G, H)$, R 的函数依赖集 $F = \{AB \rightarrow CE, A \rightarrow C, GP \rightarrow B, EP \rightarrow A, CDE \rightarrow P, HB \rightarrow P, D \rightarrow HG, ABC \rightarrow PG\}$, 求 D^+

【参考答案】

$D^+ = \{DHG\}$

4.3 证明函数依赖集 $F = \{A \rightarrow BC, A \rightarrow D, CD \rightarrow E\}$ 和函数依赖集 $G = \{A \rightarrow BCE, A \rightarrow ABD, CD \rightarrow E\}$ 的等价性

【参考答案】

$A \rightarrow BC, A \rightarrow D, CD \rightarrow E, A \rightarrow BCE, A \rightarrow ABD$, 有 $F \subseteq G^+$

$A \rightarrow BCE, A \rightarrow ABD, A \rightarrow BC, A \rightarrow D, CD \rightarrow E$, 有 $G \subseteq F^+$

所以 F 和 G 等价。

4.4 设关系模式 $R(ABCD)$, F 是 R 上成立的函数依赖集, $F = \{A \rightarrow B, C \rightarrow B\}$, 则相对于 F , 试写出关系模式 R 的候选码, 并说明理由

【参考答案】

关系模式 R 的候选码为 ACD

在关系 F 中 B 只出现在右边, 所以 B 一定不是候选码

在关系 F 中 D 没有出现 D 必然出现在候选码中

在关系 F 中 AC 出现在左边

$A \rightarrow B, C \rightarrow B, A \rightarrow A$

所以 A 能推出 ABC , 因此候选码是 ACD

4.5 设有关系模式 $R(A, B, C, D, E)$, R 的函数依赖集 $F = \{AB \rightarrow D, B \rightarrow CD, DE \rightarrow B, CD \rightarrow D, D \rightarrow A\}$

计算 $(AB)^+, (AC)^+, (DE)^+$

求 R 的所有候选码

求 F 的最小覆盖

【参考答案】

$(AB)^+ = \{ABCD\}$

$(AC)^+ = \{ACD\}$

$(DE)^+ = \{ABCDE\}$

R 属性：E, LR 属性：ABCD

$(AE)^+ = \{AE\}$

$(BE)^+ = \{ABCDE\}$

$(CE)^+ = \{ABCDE\}$

$(DE)^+ = \{ABCDE\}$

R 的候选码为：BE, CE, DE

右部属性单一化： $F_1 = \{AB \rightarrow D, B \rightarrow C, B \rightarrow D, DE \rightarrow B, C \rightarrow D, D \rightarrow A\}$

去掉多余的函数依赖： $F_2 = \{B \rightarrow C, DE \rightarrow B, C \rightarrow D, D \rightarrow A\}$

去掉冗余的属性：没有冗余属性

所以 F 的最小覆盖 $F_{\min} = F_2 = \{B \rightarrow C, DE \rightarrow B, C \rightarrow D, D \rightarrow A\}$

4.6 设有关系模式 $R(A, B, C, D)$, R 的函数依赖集 $F = \{A \rightarrow C, C \rightarrow A, B \rightarrow AC, D \rightarrow AC, BD \rightarrow A\}$, 求 F 的最小覆盖

【参考答案】

第一步：将 F 的所有函数依赖的右部都分解成单一属性：

$F_1 = \{A \rightarrow C, C \rightarrow A, B \rightarrow A, B \rightarrow C, D \rightarrow A, D \rightarrow C, BD \rightarrow A\}$

第二步：去掉冗余的函数依赖：

1 考察 $A \rightarrow C$, 令 $G = \{C \rightarrow A, B \rightarrow A, B \rightarrow C, D \rightarrow A, D \rightarrow C, BD \rightarrow A\}$, $A^+_G = \{A\}$

因为 $C \notin A^+_G$, 所以 $A \rightarrow C$ 不冗余；

2 考察 $C \rightarrow A$, 令 $G = \{A \rightarrow C, B \rightarrow A, B \rightarrow C, D \rightarrow A, D \rightarrow C, BD \rightarrow A\}$, $C^+_G = \{C\}$

因为 $A \notin C^+_G$, 所以 $C \rightarrow A$ 不冗余；

3 考察 $B \rightarrow A$, 令 $G = \{A \rightarrow C, C \rightarrow A, B \rightarrow C, D \rightarrow A, D \rightarrow C, BD \rightarrow A\}$, $B^+_G = \{ABC\}$

因为 $A \in B^+_G$, 所以 $B \rightarrow A$ 冗余, 从 F_1 中删除 $B \rightarrow A$, $F_2 = \{A \rightarrow C, C \rightarrow A, B \rightarrow C, D \rightarrow A, D \rightarrow C, BD \rightarrow A\}$;

4 考察 $B \rightarrow C$, 令 $G = \{A \rightarrow C, C \rightarrow A, D \rightarrow A, D \rightarrow C, BD \rightarrow A\}$, $B^+_G = \{B\}$

因为 $C \notin B^+_G$, 所以 $B \rightarrow C$ 不冗余；

5 考察 $D \rightarrow A$, 令 $G = \{A \rightarrow C, C \rightarrow A, B \rightarrow C, D \rightarrow C, BD \rightarrow A\}$, $D^+_G = \{ACD\}$

因为 $A \in D^+_G$, 所以 $D \rightarrow A$ 冗余, 从 F_2 中删除 $D \rightarrow A$, $F_3 = \{A \rightarrow C, C \rightarrow A, B \rightarrow C, D \rightarrow C, BD \rightarrow A\}$;

6 考察 $D \rightarrow C$, 令 $G = \{A \rightarrow C, C \rightarrow A, B \rightarrow C, BD \rightarrow A\}$, $D^+_G = \{D\}$

因为 $C \notin D^+_G$, 所以 $D \rightarrow C$ 不冗余；

7 考察 $BD \rightarrow A$, 令 $G = \{A \rightarrow C, C \rightarrow A, B \rightarrow C, D \rightarrow C\}$, $(BD)^+_G = \{ABCD\}$

因为 $A \in (BD)^+_G$, 所以 $BD \rightarrow A$ 冗余, 从 F_3 中删除 $BD \rightarrow A$, $F_4 = \{A \rightarrow C, C \rightarrow A, B \rightarrow C, D \rightarrow C\}$;

第三步：去掉冗余的属性：
 由于左边都是单属性，所以： $F_m = F_4 = \{A \rightarrow C, C \rightarrow A, B \rightarrow C, D \rightarrow C\}$;
 但是结果不唯一。

4.7 设关系模式 $R(ABC)$ ， F 是 R 上成立的 FD 集， $F = \{C \rightarrow A, B \rightarrow A\}$ ，分解 $\rho = \{AB, BC\}$ ，判断 ρ 是否具有函数依赖保持性？

【参考答案】

$F_1 = \Pi_{U_1}(F) = (B \rightarrow A)$
 $F_2 = \Pi_{U_2}(F) = \varnothing$
 $G = F_1 \cup F_2 = \{B \rightarrow A\}$
 $F = \{C \rightarrow A, B \rightarrow A\}$
 显然， G 必定包含于 F^+ 。而 F 不包含于 G^+ 。
 因此，有 $G^+ \subsetneq F^+$ ，即 ρ 不具有函数依赖保持性。

4.8 设关系模式 $R(ABC)$ ， F 是 R 上成立的 FD 集， $F = \{C \rightarrow A, B \rightarrow C\}$ ， $\rho = \{AB, AC\}$ ，判断 ρ 是否具有“无损连接性”和“函数依赖保持”性

【参考答案】

考察“无损连接性”：

首先构造初始表，结构如表 1

表 1 初始表			
$\begin{matrix} A_j \\ R \end{matrix}$	A	B	C
AB	a_1	a_2	b_{13}
AC	a_1	b_{23}	a_3

修改表

逐一考察 F 中的函数依赖：

- a) $C \rightarrow A$ ，表的结构不变；
- b) $B \rightarrow C$ ，表的结构不变；

此时，对 F 中的每个函数依赖，表的结构都不再变化。又因为表中没有出现 a_1, a_2, a_3 的行，所以该分解不具有无损连接性。

考察“函数依赖保持”

$F_1 = \Pi_{U_1}(F) = (B \rightarrow A)$
 $F_2 = \Pi_{U_2}(F) = (C \rightarrow A)$
 $G = F_1 \cup F_2 = \{B \rightarrow A, C \rightarrow A\}$
 $F = \{C \rightarrow A, B \rightarrow C\}$
 显然， G 必定包含于 F^+ 。而 F 不包含于 G^+ 。
 因此，有 $G^+ \subsetneq F^+$ ，即 ρ 不具有函数依赖保持性。

4.9 设关系模式 $R(ABCD)$ ，在 R 上有 5 个相应的 FD 集及分解：

$$F = \{B \rightarrow C, D \rightarrow A\}, \quad \rho = \{AD, BC\}$$

$$\begin{aligned}
 F &= \{ AB \rightarrow C, C \rightarrow A, C \rightarrow D \}, \quad G = \{ ACD, BC \} \\
 F &= \{ A \rightarrow BC, C \rightarrow AD \}, \quad G = \{ ABC, AD \} \\
 F &= \{ A \rightarrow B, B \rightarrow C, C \rightarrow D \}, \quad G = \{ AB, ACD \} \\
 F &= \{ A \rightarrow B, B \rightarrow C, C \rightarrow D \}, \quad G = \{ AB, AD, CD \}
 \end{aligned}$$

试对上述 5 中情况分别回答下列问题：

- 确定 R 的候选码和主码。
- 是否为无损分解？
- 是否函数依赖保持？
- 确定 G 中每一模式的范式级别。

【参考答案】

- 1 分解 $F = \{ B \rightarrow C, D \rightarrow A \}, \quad G = \{ AD, BC \}$
- A) $(BD)^+ = \{ ABCD \}$ BD 是候选码，也是主码
- B) 首先构造初始表，结构如表 2

表 2 初始表

$\begin{matrix} A_j \\ R_i \end{matrix}$	A	B	C	D
AD	a ₁	b ₁₂	b ₁₃	a ₄
BC	b ₂₁	a ₂	a ₃	b ₂₄

修改表

逐一考察 F 中的函数依赖：

- a) $B \rightarrow C$ ，表的结构不变；
- b) $D \rightarrow A$ ，表的结构不变；

此时，对 F 中的每个函数依赖，表的结构都不再变化。 又因为表中没有出现 a₁ , a₂ , a₃ , a₄ 的行，所以该分解不具有无损连接性。

- C) $F_1 = \Pi_{U_1}(F) = (B \rightarrow C)$
- $F_2 = \Pi_{U_2}(F) = (D \rightarrow A)$
- $G = F_1 \cup F_2 = \{ B \rightarrow C, D \rightarrow A \}$
- $F = \{ B \rightarrow C, D \rightarrow A \}$
- 显然，G 必定包含于 F^+ 。而 F 包含于 G^+ 。
- 因此，有 $G^+ = F^+$ ，即 G 具有函数依赖保持性。

- D) 模式 ad(A,D) \in BCNF，模式 bc(B,C) \in BCNF

- 2 分解 $F = \{ AB \rightarrow C, C \rightarrow A, C \rightarrow D \}, \quad G = \{ ACD, BC \}$

- A) L 属性 B，LR 属性 AC，R 属性 D
- $(B)^+ = \{ B \}$
- $(AB)^+ = \{ ABCD \}$ 所以 AB 是候选码
- $(BC)^+ = \{ ABCD \}$ 所以 BC 是候选码
- 选择 AB 做为主码

- B) 构造初始表

$\begin{matrix} A_j \\ R_i \end{matrix}$	A	B	C	D
ACD	a ₁	b ₁₂	a ₃	a ₄
BC	b ₂₁	a ₂	a ₃	b ₂₄

修改表

<div><div>A_j</div><div>R_i</div></div>	A	B	C	D
ACD	a ¹	b ¹²	a ³	a ⁴
BC	a ¹	a ²	a ³	a ⁴

因为表中出现 a₁ , a₂ , a₃ , a₄ 的行 , 所以该分解具有无损连接性。

C) $F_1 = \Pi_{U_1}(F) = (C \rightarrow A, C \rightarrow D)$

$F_2 = \Pi_{U_2}(F) = \emptyset$

$G = F_1 \cap F_2 = \{C \rightarrow A, C \rightarrow D\}$

$F = \{B \rightarrow C, D \rightarrow A\}$

显然 , G 必定包含于 F⁺。而 F 不包含于 G⁺。

因此 , 有 G⁺ ≠ F⁺ , 即 不具有函数依赖保持性。

D) 模式 ad(ACD) ∈ BCNF, 模式 bc(BC) ∈ BCNF

3 分解 $F = \{A \rightarrow BC, C \rightarrow AD\}, \quad = \{ABC, AD\}$

- A) 候选码为 : A , C 选主码为 : A
- B) 具有无损连接性。
- C) 具有函数依赖保持性
- D) 均为 BCNF

4 分解 $F = \{A \rightarrow B, B \rightarrow C, C \rightarrow D\}, \quad = \{AB, ACD\}$

- A) 候选码为 : A , 主码为 : A
- B) 具有无损连接性。
- C) 不具有函数依赖保持性
- D) 模式 ad(AB) ∈ BCNF, 模式 bc(ACD) ∈ 2NF

5 分解 $F = \{A \rightarrow B, B \rightarrow C, C \rightarrow D\}, \quad = \{AB, AD, CD\}$

- A) 候选码为 : A 选主码为 : A
- B) 不具有无损连接性。
- C) 不具有函数依赖保持性
- D) 均为 BCNF

4.10 设有关系模式 R (A , B , C , D , E), R 的函数依赖集 $F = \{ AB \rightarrow C, C \rightarrow D, D \rightarrow E \}$ 。判断分解 $= \{R_1(ABC), R_2(CD), R_3(DE) \}$ 是否为无损连接分解。并且 :

求 R 的所有候选码

求 F 的最小覆盖

将 R 分解为 3NF 并具有无损连接性和函数依赖保持性

【参考答案】

初始表

<div><div>A_j</div><div>R_i</div></div>	A	B	C	D	E
ABC	a ¹	a ²	a ³	b ¹⁴	b ¹⁵

CD	b ₂₁	b ₂₂	a ₃	a ₄	b ₂₅
DE	b ₃₁	b ₃₂	b ₃₃	a ₄	a ₅

修改表

逐一考察 F 中的函数依赖：

a) AB C 表的结构不变；

b) C D, 将 b₁₄ 改为 a₄

$\begin{matrix} A_j \\ R_i \end{matrix}$	A	B	C	D	E
ABC	a ₁	a ₂	a ₃	a ₄	b ₁₅
CD	b ₂₁	b ₂₂	a ₃	a ₄	b ₂₅
DE	b ₃₁	b ₃₂	b ₃₃	a ₄	a ₅

c) D E, 将 b₁₅ 改为 a₅, 将 b₂₅ 改为 a₅

$\begin{matrix} A_j \\ R_i \end{matrix}$	A	B	C	D	E
ABC	a ₁	a ₂	a ₃	a ₄	a ₅
CD	b ₂₁	b ₂₂	a ₃	a ₄	a ₅
DE	b ₃₁	b ₃₂	b ₃₃	a ₄	a ₅

是无损连接分解

求 R 的所有候选码

L 类属性 AB, LR 类属性为 CD

$(AB)^+ = \{ABCDE\}$, 所以候选码为 AB

求 F 的最小覆盖

第一步：将 F 的所有函数依赖的右部都分解成单一属性：

由于均为单一属性，所以 $F_1 = F = \{AB \rightarrow C, C \rightarrow D, D \rightarrow E\}$

第二步：去掉冗余的函数依赖：

没有冗余函数依赖，所以 $F_2 = F_1 = \{AB \rightarrow C, C \rightarrow D, D \rightarrow E\}$

第三步：去掉冗余的属性：

没有冗余的属性，所以 $F_m = F_2 = \{AB \rightarrow C, C \rightarrow D, D \rightarrow E\}$

将 R 分解为 3NF 并具有无损连接性和函数依赖保持性

$= \{R_1(ABC), R_2(CD), R_3(DE)\}$

4.11 设有关系模式 R (职工名, 项目名, 工资, 部门名, 部门经理)

如果规定每个职工可参加多个项目, 各领一份工资; 每个项目只属于一个部门管理; 每个部门只有一个经理。

试写出关系 R 的基本 FD 和候选码, 确定主码。

说明 R 不是 2NF 的理由, 并把 R 分解成 2NF 的模式集。

把 R 分解成 3NF 模式集, 说明理由。

【参考答案】

(1) 由题意得函数依赖集 F

$F = \{\text{职工名} \rightarrow \text{工资}, \text{项目名} \rightarrow \text{部门名}, \text{部门名} \rightarrow \text{部门经理}\}$

候选码：职工名 + 项目名

由于只有一个候选码，故主码即为职工名 + 项目名

(2) 由于项目名 部门名, 所以存在非主属性 (部门名) 对码 (职工名 + 项目名) 的部分依赖, 故 R 没有达到 2NF。

可以分解为如下两个关系模式：

R1= (职工名 ,项目名 , 工资)

R2= (项目名 , 部门名 , 部门经理)

(3)由于 R2中存在函数依赖：项目名 → 部门名, 部门名 → 部门经理, 即存在非主属性 (部门经理) 对码 (项目名) 的传递依赖, 故 R2 没有达到 3NF。可以在 (2)的基础上把 R2进一步分解为如下两个关系模式：

R1= (项目名 ,部门名)

R2= (部门名 , 部门经理)

综上, R 分解成 3NF 模式集为：

R1= (职工名 ,项目名 , 工资)

R2= (项目名 ,部门名)

R3= (部门名 , 部门经理)

第五章

5.1 名词解释

数据库设计 基于 3NF 的数据库设计方法 基于 E-R模型的数据库设计方法

数据库设计是指对于一个给定的应用环境，根据用户的需求，在某一具体的数据库管理系统上，构造一个性能良好的数据模式，建立数据库及其应用系统，使之能够有效地存储数据，满足各种用户的信息要求和处理要求。

基于 E-R 模型的数据库设计方法是由 P.P.S.chen 于 1976 年提出的数据库设计方法，其基本思想是在需求分析的基础上，用 E-R 图构造一个反映现实世界实体之间联系的企业模式，然后再将此企业模式转换成基于某一特定的 DBMS 的概念模式。

基于 3NF 的数据库设计方法是由 S·Atre 提出的结构化设计方法，其基本思想是在需求分析的基础上，确定数据库模式中的全部属性和属性间的依赖关系，将它们组织在一个单一的关系模式中，然后再分析模式中不符合 3NF 的约束条件，将其进行投影分解，规范成若干个 3NF 关系模式的集合。

5.2 什么是数据库设计？试述数据库设计的过程

数据库设计 基于 3NF 的数据库设计方法 基于 E-R模型的数据库设计方法

数据库设计是指对于一个给定的应用环境，根据用户的需求，在某一具体的数据库管理系统上，构造一个性能良好的数据模式，建立数据库及其应用系统，使之能够有效地存储数据，满足各种用户的信息要求和处理要求。

按照规范设计方法、考虑数据库及其应用系统开发全过程，并仿照软件生存周期，将数据库设计分为需求分析、概念结构设计、逻辑结构设计、物理结构设计、数据库实施和运行维护六个阶段。数据库设计过程可以用图 5-1 表示。

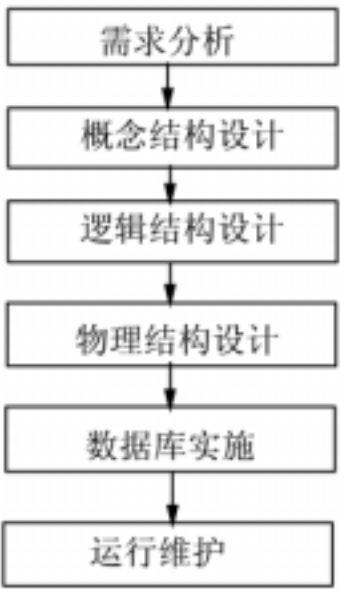


图 5-1 数据库设计步骤

(1) 需求分析

需求分析是对具体应用环境的业务流程和用户提出的各种要求加以调查研究和分析，并和用户共同对各种原始数据加以综合、整理的过程，是形成最终设计目标的首要阶段，也是整个数据库设计过程中最困难的阶段。该阶段任务的完成，将为以后各阶段任务打下坚实的基础。因此，对用户各种需求及数据，能否做出准确无误、充分完备的分析，并在此基础上形成最终目标，是整个数据库设计成败的关键。

(2) 概念结构设计

概念结构设计是对用户信息需求所进行的进一步抽象和归纳，结果为数据库概念结构，通常用 E-R 模型来表示。

数据库的概念结构与 DBMS 和相关软硬件无关。它是对现实世界中具体数据的抽象，实现了从现实世界到信息世界的转化过程。概念结构设计是数据库设计的一个重要环节，是数据库的逻辑结构设计和物理结构设计的基础。

（3）逻辑结构设计

概念结构设计的结果是得到一个与 DBMS 无关的概念模式，而逻辑结构设计就是将概念模式转化为选用的具体 DBMS 所支持的数据模型相符合的逻辑结构。所以，在逻辑结构设计阶段选择什么样的数据模型和哪一个具体 DBMS 尤为重要，它是能否满足用户各种要求的关键。

在逻辑结构设计阶段还有一个很重要的工作就是模式优化，该工作主要以用规范化理论为指导，目的是能够合理存放数据集合。逻辑结构设计阶段的模式优化，已成为影响数据库设计质量的一项重要工作。

（4）物理结构设计

数据库物理设计是将逻辑结构设计阶段所产生的逻辑数据模型，转换为某一计算机系统所支持的数据库物理结构的实现过程。

数据库的物理结构主要指数据库的存储记录格式、存储记录安排和存储方法，完全依赖于给定的硬件环境、具体的 DBMS 和操作系统，。

存储记录格式的设计包括记录的组成、数据项的类型、长度，以及逻辑记录到存储记录的映射。存储记录的安排是指可以把经常同时被访问的数据组合在一起。存取方法的设计主要是指存取路径，存取路径分为主存取路径与辅存取路径，前者用于主键检索，后者用于辅助键检索。

除此之外，物理结构设计还要进行完整性和安全性考虑，设计者应在完整性、安全性、有效性和效率方面进行分析，做出权衡。

完成物理结构设计后，对该物理结构做出相应的性能评价，若评价结果符合原设计要求，则进一步实现该物理结构。否则，对该物理结构做出相应的修改，若属于最初设计问题所导致的物理结构的缺陷，必须返回到概念设计阶段修改其概念数据模型或重新建立概念数据模型，如此反复，直至评价结果最终满足原设计要求为止。

（5）数据库实施

数据库实施阶段，即数据库调试、试运行阶段。一旦数据库物理结构形成，就可以用已选定的 DBMS 定义、描述相应的数据库结构，装入数据库数据，以生成完整的数据库，编制有关应用程序，进行联机调试并转入试运行，同时进行时间、空间等性能分析，若不符合要求，则需调整物理结构、修改应用程序，直至高效、稳定、正确地运行该数据库系统为止。

（6）数据库运行和维护

数据库实施阶段结束，标志着数据库系统投入正常运行的开始。严格地说，数据库运行和维护不属于数据库设计的范畴，早期的新奥尔良法明确规定数据库设计的四个阶段，不包括运行和维护内容。随着人们对数据库设计的深刻了解和设计水平的不断提高，已经充分认识到数据库运行和维护工作与数据库设计的紧密联系。数据库设计是一种动态和不断完善的运行过程，运行和维护阶段开始，并不意味着设计过程的结束，任何哪怕只有细微的结构改变，也许就会引起对物理结构的调整、修改，甚至物理结构的完全改变，因此数据库运行和维护阶段是保证数据库日常活动的一个重要阶段。

5.3 试述数据库设计的需求分析阶段的目标和方法

需求分析的任务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。新系统必须充分考虑今后可能的扩充和改变，不能仅仅按当前应用需求来设计数据库。

需求分析常用的方法有：

（1）跟班作业。通过亲身参加业务工作来了解业务活动的情况。这种方法可以比较准确地理解用户的需求，但比较耗费时间。

(2) 开调查会。通过与用户座谈来了解业务活动情况及用户需求。座谈时，参加者之间可以相互启发。

(3) 请专人介绍。

(4) 询问。对某些调查中的问题，可以找专人询问。

(5) 设计调查表请用户填写。如果调查表设计得合理，这种方法是有效，也很易于为用户接受的。

(6) 查阅记录。即查阅与原系统有关的数据记录，包括原始单据、账簿、报表等。

需求分析的过程一般是：

(7) 分析用户活动，产生业务流程图：了解用户当前的业务活动和职能，理清其处理流程。把用户业务分成若干个子处理过程，使每个处理功能明确、界面清楚，画出业务流程图

(8) 确定系统范围，产生系统范围图：在和用户经过充分讨论的基础上，确定计算机所能进行数据处理的范围，确定哪些工作由人工完成，哪些工作由计算机系统完成，即确定人机界面。

(9) 分析用户活动所涉及的数据，产生数据流图：深入分析用户的业务处理，以数据流图 (Data Flow Diagram, DFD) 形式表示出数据的流向和对数据所进行的加工。DFD 有四个基本成分：数据流、加工或处理、文件、外部实体。DFD 可以形象地表示数据流与各业务活动的关系，它是需求分析的工具和分析结果的描述手段。

(10) 分析系统数据，产生数据字典：仅仅有 DFD 并不能构成需求说明书，DFD 只表示出系统有哪几部分组成和各个部分之间的关系，并没有说明各个成分的含义。数据字典提供对数据库时间描述的集中管理，它的功能是存储和检索各种数据描述 (元数据 Metadata)，数据字典是数据收集和数据分析的主要成果，在数据库设计中占有重要地位。

(11) 功能分析：数据库的设计是与应用系统的设计紧密结合的过程，离开一定的功能，数据库就失去其存在价值。数据库设计的一个重要特点是结构 (数据) 和行为 (功能) 的结合。用户希望系统能提供的功能必须有一个清晰的描述。功能分析可以采用软件结构图或模块图来表示系统的层次分解关系、模块调用关系。

5.4 评审在数据库设计中有什么作用？为什么允许设计过程中有多次的回溯与重复？

评审的作用在于确认某一阶段的任务是否全部完成，通过评审可以及早发现系统设计中的错误，并在生存期的早期阶段给予纠正，以减少系统研制的成本。

如果在数据库已经实现时再发现设计中的错误，那么代价比较大。因此应该允许设计过程的回溯与反复。设计过程需要根据评审意见修改所提交的阶段设计成果，有时修改甚至要回溯到前面的某一阶段，进行部分乃至全部重新设计。

5.5 数据字典的内容和作用是什么？

数据字典是将数据流程图中各个要素的具体内容和特征，以特定格式记录下来，所形成的文档。它主要包括：数据项、数据结构、数据流、加工、文件、外部实体等内容。在数据库设计过程中，数据字典被不断地充实、修改和完善。

对数据库设计来讲，数据字典是进行数据收集和数据分析所获得的主要成果，是各类数据描述的集合。数据字典通常包括数据项、数据结构、数据流、数据存储和处理过程五个部分。

5.6 什么是数据库的概念结构？数据库的概念结构设计的特点和方法是什么？

将需求分析得到的用户需求抽象为信息结构即概念模型的过程就是概念结构设计。

对数据库概念模型有以下要求：

(1) 有丰富的语义表达能力，能表达用户的各种需求。

(2) 易于交流和理解,从而可以用它和不熟悉计算机的用户交换意见。

(3) 要易于更改。当应用环境和应用要求改变时,概念模型要能很容易的修改和扩充以反映这种变化。

(4) 易于向各种数据模型转换。

(1) 集中式模式设计法

即首先定义全局概念结构的框架,然后逐步细化。例如,可以先确定几个高级实体类型,然后在确定其属性时,把这些实体类型分裂为更低一层的实体类型和联系。

(2) 视图集成法

以各部分的需求说明为基础,分别设计各自的局部模式,这些局部模式相当于各部分的视图,然后再以这些视图为基础,集成为一个全部模式。视图是按照某个用户组、应用或部门的需求说明,用 E-R 数据模型设计的局部模式。

(3) 混合方法

即将集中式模式设计法和视图集成法相结合,用集中式模式设计法设计一个全局概念结构的框架,以它为骨架集成由视图集成法中设计的各局部概念结构。

(4) 由内向外法

首先定义最重要的核心概念结构,然后向外扩充,考虑已存在概念附近的新概念使得建模过程向外扩展。使用该策略,可以先确定模式中比较明显的一些实体类型,然后继续添加其它相关的实体类型。

5.7 什么是视图集成?视图集成的方法是什么?

以各部分的需求说明为基础,分别设计各自的局部模式,这些局部模式相当于各部分的视图,然后再以这些视图为基础,集成为一个全部模式。视图是按照某个用户组、应用或部门的需求说明,用 E-R 数据模型设计的局部模式。

子视图的集成是一个非常复杂的过程,需要一个更加严格和系统化的方法。下面介绍一些用于视图合并的策略:

1) 二元集成

首先对两个比较类似的模式进行集成。然后把结果模式和另外一个模式集成,不断重复该过程直到所有模式被集成。可以根据模式的相似程度确定模式集成的顺序。由于集成是逐步进行的,所以该策略适用于手工集成。

2) n 元集成

对视图的集成关系进行分析和说明之后,在一个过程中完成所有视图的集成。对于规模较大的设计问题,这个策略需要使用计算机化的工具,目前有一些这种工具的原型,但还没有成熟的商业产品。

3) 二元平衡策略

首先将模式成对地进行集成,然后再将结果模式成对地进一步集成,不断重复该过程直至得到最终的全局模式。

4) 混合策略 (mixed strategy)

首先,根据模式的相似性把它们划分为不同的组,对每个组单独地进行集成。然后对中间结果进行分组并集成,重复该过程直至集成结束。

5.8 视图集成时,分 E-R图之间的冲突有哪些?解决这些冲突的方法是什么?

由于局部概念设计相对简单,因此简化了全局模式的设计。但是,在将局部视图合成为全局视图的时候,需要一个方法学在集成时具体地解决下列问题:

1) 确定模式之间的对应和冲突

由于各子模式是分开进行设计的，因此有必要在集成之前确定各模式表示的是否是同一个现实世界的概念结构。在此过程中，模式间可能会发生如下的一些冲突：

属性冲突

属性域冲突，即属性值的类型、取值范围或取值集合不同，如零件号，有的部门作为整数对待，有的部门则使用字符串。不同部门对零件号的编码也可能不同。

属性取值单位冲突：如零件重量，有的部门以公斤为单位，有的部门以克为单位。

命名冲突

包括同名异义和异名同义。如科研项目，财务科称为项目，科研处称为课题，生产管理处称为工程，这就是一个异名同义的例子。

结构冲突

同一对象在不同应用中具有不同的抽象。如在教学管理中，职称是一个属性；而在人事管理中，因为职称与工资、住房挂钩，因此是一个实体。

同一实体在不同局部视图所包含的属性不完全相同。

实体间的联系在不同分 E-R图中为不同类型：如生产子系统分 E-R图中，产品和零件构成 1:n 联系。而物资子系统分 E-R图中，产品、零件、供应商三者构成多对多联系。

2) 修改视图使得相互一致

对一些模式进行修改，以便于其它模式相符合。这一步可以解决上一步发现的冲突。

3) 合并视图

通过创建单个子视图来创建全局视图。相应的概念在全局模式中只出现一次，并且要确定子视图和全局视图之间的映射关系。在涉及到数百个实体和联系的实际数据库中，这一步是最为困难的。因为牵扯到大量的人为干预和协商来解决冲突，并且要确定全局模式的一个最为合理并且能够接受的解决方案。

4) 重构

该步骤是一个可选步骤，可能会对全局模式进行分析和重构，以删除任何冗余和不必要的内容。

5.9 试述数据库逻辑结构设计的步骤。

逻辑结构设计阶段需要完成的任务有：

- (1) 将 E-R 模型转换为等价的关系模式。
- (2) 按需要对关系模式进行规范化。
- (3) 对规范化后的模式进行评价。
- (4) 根据局部应用的需要，设计用户模式。

5.10 试述 E-R图转换成关系模型的转换规则。

E-R图向关系模型的转换一般应遵循如下原则：

(1) 一个实体型转换为一个关系模式。实体的属性就是关系的属性。实体的属性就是关系的属性。实体的码就是关系的码。

例如在例子中，学生实体可以转换为如下关系模式，其中学号为学生关系的码：学生(学号, 姓名, 出生日期, 院系, 专业, 班级, 宿舍) 同样宿舍、班级、专业、院系都分别转换为一个关系模式。

(2) 一个联系转化为一个关系模式，与该联系相连的各实体的码以及联系的属性转化为关系的属性，该关系的码则有几种情况：

若联系为 1:1，则每个实体的码均是该关系的后选码。

若联系为 1:n, 则关系的码为 n 端实体的码。

若联系为 m:n, 则关系的码为诸实体码的组合。

三个或三个以上实体间的多元联系。

同一实体集的实体间的联系, 即自联系。

1) 联系为 1:1

一个 1:1 联系可以转换为一个独立的关系模式, 也可以与任意一端对应的关系模式合并。

如果转换为一个独立的关系模式, 则与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性, 每个实体的码均是该关系的候选码。如果与某一端对应的关系模式合并, 则需要在该关系模式的属性中加入另一个关系模式的码和联系本身的属性。

2) 联系为 1:n

一个 1:n 联系可以转换为一个独立的关系模式, 也可以与 n 端对应的关系模式合并。如果转换为一个独立的关系模式, 则与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性, 而关系的码为 n 端实体的码。如果与 n 端对应的关系模式合并, 则在 n 端实体对应模式中加入 1 端实体所对应关系模式的码, 以及联系本身的属性。而关系的码为 n 端实体的码。

3) 联系为 m:n

一个 m:n 联系转换为一个关系模式。与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性。而关系的码为各实体码的组合。

4) 三个或三个以上实体间的多元联系

三个或三个以上实体间的一个多元联系转换为一个关系模式。与该多元联系相连的各实体的码以及联系本身的属性均转换为关系的属性。而关系的码为各实体码的组合。

5) 同一实体集的实体间的联系, 即自联系, 也可按上述 1:1、1:n 和 m:n 三种情况分别处理。

(3) 具有相同码的关系模式可合并。

为了减少系统中的关系个数, 如果两个关系模式具有相同的主码, 可以考虑将他们合并为一个关系模式。合并方法是将其中的一个关系模式的全部属性加入到另一个关系模式中, 然后去掉其中的同义属性(可能同名也可能不同名), 并适当调整属性的次序。

5.11 规范化理论对数据库设计有什么指导意义。

应用规范化理论对逻辑设计阶段产生的逻辑模式进行初步优化, 以减少乃至消除关系模式中存在的各种异常, 改善完整性、一致性和存储效率。规范化理论是数据库逻辑设计的指南和工具, 规范化过程分为两个步骤: 确定范式的级别和实施规范化处理(模式分解)。

5.12 试述数据库中对逻辑结构设计结果进行优化的方法。

(1) 确定范式级别

考察关系模式的函数依赖关系, 确定范式等级。找出所有“数据字典”中得到的数据之间的依赖关系, 对各模式之间的数据依赖进行极小化处理, 消除冗余的联系。按照数据依赖理论对关系模式逐一进行分析, 考察是否存在部分函数依赖、传递函数依赖和多值依赖等, 确定各关系模式属于第几范式。

(2) 实施规范化处理

确定范式级别后, 根据应用需求, 判断它们对于这样的应用环境是否合适, 确定对于这些模式是否进行合并或分解。

对关系模式进行必要的分解, 提高数据操作的效率和存储空间的利用率。常用的方法是水平分解和垂直分解两种分解方法。

5.13 试述数据库物理设计的内容和步骤。

数据库最终要存储在物理设备上。对于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，称为数据库物理设计。物理设计的任务是为了有效地实现逻辑模式，确定所采取的存储策略。此阶段是以逻辑设计的结果作为输入，结合具体 DBMS 的特点与存储设备特性进行设计，选定数据库在物理设备上的存储结构和存取方法。

数据库的物理设计可分为两步：

- (1) 确定物理结构，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，评价的重点是时间和空间效率。

5.14 数据库实施阶段主要任务？

数据库实施是指根据逻辑设计和物理设计的结果，在计算机上建立起实际的数据库结构、装入数据、进行测试和试运行的过程。

数据库实施主要包括以下工作：

- 建立实际数据库结构；
- 组织数据入库
- 编制与调试应用程序
- 数据库试运行

5.15 数据库系统投入运行后，有哪些维护工作？

在数据库运行阶段，对数据库经常性的维护工作主要是由 DBA 完成的，它包括：

- (1) 数据库的转储和恢复
- (2) 数据库的安全性、完整性控制
- (3) 数据库性能的监督、分析和改进
- (4) 数据库的重组织和重构造

5.16 假设某超市公司要设计一个数据库系统来管理该公司的业务信息。该超市公司的业务管理规则如下：

- (1) 该超市公司有若干仓库，若干连锁商店，供应若干商品。
- (2) 每个商店有一个经理和若干收银员，每个收银员只在一个商店工作。
- (3) 每个商店销售多种商品，每种商品可在不同的商店销售。
- (4) 每个商品编号只有一个商品名称，但是不同的商品编号可以有相同的商品名称。每种商品可以有多种销售价格。
- (5) 超市公司的业务员负责商品的进货。

请根据以上规则，设计出适当的属性，画出 E-R 图，再将其转换成关系模型后进行规范化。

商店：商店编号，商店名称，商店地址，商店编号，联系方式

商品：商品编号，商品名称，单价，数量，类别

员工：编号，姓名，性别，出生日期，联系电话

仓库：仓库编号，名称，联系方式

本题对应的 E-R 图如图所示

