第 1 章 数据库概论

1.1 基本内容分析

1.1.1 本章的重要概念

（1）DB、DBMS 和 DBS 的定义

（2）数据管理技术的发展阶段

人工管理阶段、文件系统阶段、数据库系统阶段和高级数据库技术阶段等各阶段的特点。

（3）数据描述

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

（4）数据模型

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

（5）DB 的体系结构

三级结构，两级映像，两级数据独立性，体系结构各个层次中记录的联系。

（6）DBMS

DBMS 的工作模式、主要功能和模块组成。

（7）DBS

DBS 的组成，DBA，DBS 的全局结构，DBS 结构的分类。

1.1.2 本章的重点篇幅

（1）教材 P23 的图 1.24（四种逻辑数据模型的比较）。

（2）教材 P25 的图 1.27（DB 的体系结构）。

（3）教材 P28 的图 1.29（DBMS 的工作模式）。

（4）教材 P33 的图 1.31（DBS 的全局结构）。

1.2 教材中习题 1 的解答

1.1 名词解释

• 逻辑数据：指程序员或用户用以操作的数据形式。

• 物理数据：指存储设备上存储的数据。

• 联系的元数：与一个联系有关的实体集个数，称为联系的元数。

•1:1 联系：如果实体集 E1 中每个实体至多和实体集 E2 中的一个实体有联系，反之亦然，那么 E1 和 E2 的联系称为“1:1 联系”。

• 1:N 联系：如果实体集 E1 中每个实体可以与实体集 E2 中任意个（零个或多个）实体有联系，而 E2 中每个实体至多和 E1 中一个实体有联系，那么 E1 和 E2 的联系是“1:N 联系”。

• M:N 联系：如果实体集 E1 中每个实体可以与实体集 E2 中任意个（零个或多个）实体有联系，反之亦然，那么 E1 和 E2 的联系称为“M:N 联系”。

• 数据模型：能表示实体类型及实体间联系的模型称为“数据模型”。

• 概念数据模型：独立于计算机系统、完全不涉及信息在计算机中的表示、反映企业组织所关心的信息结构的数据模型。

• 结构数据模型（或逻辑数据模型）：与 DBMS 有关的，直接面向 DB 的逻辑结构、从计算机观点对数据建模的数据模型。

• 层次模型：用树型（层次）结构表示实体类型及实体间联系的数据模型称为层次模型。

• 网状模型：用有向图结构表示实体类型及实体间联系的数据模型称为网状模型。

• 关系模型：用二维表格表达实体集的数据模型。

• 外模式：是用户用到的那部分数据的描述。

• 概念模式：数据库中全部数据的整体逻辑结构的描述。

• 内模式：DB 在物理存储方面的描述。

• 外模式/模式映象：用于定义外模式和概念模式之间数据结构的对应性。

• 模式/内模式映象：用于定义概念模式和内模式之间数据结构的对应性。
• 数据独立性：应用程序和 DB 的数据结构之间相互独立，不受影响。
• 物理数据独立性：在 DB 的物理结构改变时，尽量不影响应用程序。
• 逻辑数据独立性：在 DB 的逻辑结构改变时，尽量不影响应用程序。
• 主语言：编写应用程序的语言（如 C 一类高级程序设计语言），称为主语言。
• DDL：定义 DB 三级结构的语言，称为 DDL。
• DML：对 DB 进行查询和更新操作的语言，称为 DML。
• 过程性语言：用户编程时，不仅需要指出“做什么”，还需要指出“怎么做”的语言。
• 非过程性语言：用户编程时，只需指出“做什么”，不需要指出“怎么做”的语言。
• DD（数据字典）：存放三级结构定义的 DB，称为 DD。
• DD 系统：管理 DD 的软件系统，称为 DD 系统。

1.2 试解释 DB、DBMS 和 DBS 三个概念。

答：DB 是长期存储在计算机内、有组织的、统一管理的相关数据的集合。

DBMS 是位于用户与 OS 之间的一层数据管理软件，它为用户或应用程序提供访问 DB 的方法。

DBS 是实现有组织地、动态地存储大量关联数据、方便多用户访问的计算机硬件、软件和数据资源组成的系统，即采用数据库技术的计算机系统。

1.3 人工管理阶段和文件系统阶段的数据管理各有哪些特点？

答：人工管理阶段主要有四个特点：数据不保存在计算机内；没有专用的软件对数据进行管理；只有程序的概念，没有文件的概念；数据面向程序。

文件系统阶段主要有五个特点： 数据以“文件”形式长期保存；数据的逻辑结构与物理结构有了区别；文件组织已多样化；数据面向应用；对数据的操作以记录为单位。

1.4 文件系统阶段的数据管理有些什么缺陷？试举例说明。

答：主要有三个缺陷：数据冗余；数据不一致性；数据联系弱。

例如学校里教务处、财务处、保健处建立的文件中都有学生详细资料，譬如联系电话，家庭住址等。这就是“数据”冗余；如果某个学生搬家，就要修改三个部门文件中的数据，否则会引起同一数据在三个部门中不一致；产生上述问题的原因是这三个部门的文件中数据没有联系。

1.5 数据管理的数据库阶段产生的标志是哪三件事情？

答：进入数据库阶段的标志是 20 世纪 60 年代末发生的三件事件：

• 1968 年 IBM 公司研制的 IMS 系统是一个典型的层次 DBS；
• 1969 年美国 CODASYL 组织 DBTG 报告，提出网状 DBS 的概念；
• 1970 年美国 IBM 公司的 E.F.Codd 发表论文，提出关系模型的思想。

1.6 数据库阶段的数据管理有哪些特点？

答：主要有五个特点：

采用数据模型表示复杂的数据结构；有较高的数据独立性；为用户提供了方便的用户接口；提供了四个方面的数据控制功能；对数据的操作以数据项为单位，增加了系统的灵活性。

1.7 与“文件”结构相比，“数据库”结构有些什么不同？

答：与文件结构相比，数据库结构主要有下面三点不同：

• 数据的结构化。文件由记录组成，但各文件之间缺乏联系。数据库中数据在磁盘中仍以文件形式组织，但这些文件之间有着广泛的联系。数据库的逻辑结构用数据模型来描述，整体结构化。数据模型不仅描述数据本身的特点，还要描述数据之间的联系。

• 数据独立性。文件只有设备独立性，而数据库还具有逻辑独立性和物理独立性。

• 访问数据的单位。访问文件中的数据，以记录为单位。访问数据库中的数据，以数据项（字段）为单位，增加了系统的灵活性。

1.8 什么是数据独立性？在数据库中有哪两级独立性？

答：数据独立性是指应用程序与 DB 的数据结构之间相互独立。在物理结构改变时，尽量不影响应用程序，称为物理数据独立性；在逻辑结构改变时，尽量不影响应用程序，称为逻辑数据独立性。

1.9 分布式数据库系统和面向对象数据库系统各有哪些特点？

答：DDBS 主要有三个特点：

• 数据物理上分布在各地，但逻辑上是一个整体；

• 每个场地既可以执行局部应用，也可以执行全局应用；

• 各地的计算机由数据通信网络相连接。

面向对象数据系统主要有两个特点：

• 面向对象数据模型能完整地描述现实世界的数据结构，能表达数据间嵌套、递归的联系。

• 具有面向对象技术的封装性和继承性的特点，提高了软件的可重用性。

1.10 逻辑记录与物理记录，逻辑文件与物理文件有些什么联系和区别？

答：逻辑数据是用户用以操作的数据形式，是抽象的概念化数据。物理数据是实际存放在存储设备上的数据。

逻辑数据与物理数据在结构上可以差别很大，需通过两级映象来进行数据传输和格式转换。

从以上的解释可以看出，逻辑记录和逻辑文件是用户在程序中使用的记录和文件，而物理记录和物理文件是指磁盘上的记录和文件。逻辑记录、文件与物理记录、文件在结构、组成上有很大的差异，而数据管理软件就是通过三级结构两级映象来实现逻辑数据与物理数据之间的转换。

1.11 试述 ER 模型、层次模型、网状模型、关系模型和面向对象模型的主要特点。

答：ER 模型直接表示实体类型及实体间联系，与计算机系统无关，充分反映用户的需求，用户容易理解。

层次模型的数据结构为树结构，记录之间联系通过指针实现，查询较快，但 DML 属于过程化的，操作复杂。

网状模型的数据结构为有向图，记录之间联系通过指针实现，查询较快，并且容易实现 M:N 联系，但 DML 属于过程化的语言，编程较复杂。

关系模型的数据结构为二维表格，容易为初学者理解。记录之间联系通过关键码实现。DML 属于非过程化语言，编程较简单。

面向对象模型能完整描述现实世界的数据结构，具有丰富的表达能力，能表达嵌套、递归的数据结构。但涉及的知识面较广，用户较难理解，这种模型尚未普及。

1.12 数据之间联系在各种结构数据模型中是怎么实现的？

答：在层次、网状模型中，数据之间的联系通过指针实现的；

在关系模型中，数据之间联系通过外键和主键间联系实现的；

在面向对象模型中，数据之间嵌套、递归联系通过对象标识符（OID）实现的（见第 8 章）。

1.13 DB 的三级模式结构描述了什么问题？试详细解释。

答：DB 的三级模式结构是对数据的三个抽象级别，分别从外部（用户）级、概念级和内部级去观察数据库。

外部级是用户使用的局部数据库的逻辑结构，其描述称为外模式。

概念级是 DB 的整体逻辑结构，其描述称为概念模式。

内部级是 DB 的物理结构，其描述称为内模式。

1.14 试述概念模式在数据库结构中的重要地位。

答：数据按外模式的描述提供给用户，按内模式的描述存储在磁盘中，而概念模式提供了连接这两级的相对稳定的中间观点，并使得两级的任何一级的改变都不受另一级的牵制。

1.15 试叙述用户、DB 的三级模式结构、磁盘上的物理文件之间有些什么联系和不同？

答：用户、外模式、概念模式、内模式和物理文件中的记录分别称为用户记录、外部记录、概念记录、内部记录和物理记录。

用户记录与外部记录的结构是一致的，它们之间只是数据传输问题。

而外部记录、概念记录和内部记录之间的结构可能是不一致的，除了数据传输问题，还有格式转换问题。

内部记录与物理记录的结构是一致的，它们之间只是数据传输问题。

1.16 数据独立性与数据联系这两个概念有什么区别？

答：数据独立性是指应用程序和 DB 的数据之间相互独立，不受影响，对系统的要求是“数据独立性要高”，而数据联系是指记录之间的联系，对系统的要求是“数据联系密切”。

1.17 试述 DBMS 的工作模式和主要功能。

答：DBMS 的工作模式有六点：

- 接受应用程序的数据请求和处理请求；
- 将用户的数据请求转换成低层指令；
- 实现对 DB 的操作；
- 从对 DB 的操作中接受查询结果；
- 对查询结构进行处理；
- 将处理结果返回给用户。

DBMS 的主要功能有 DB 的定义、操纵、保护、维护和数据字典等五个功能。

1.18 试叙述 DBMS 对数据库的维护功能。

答：包括 DB 的数据载入、转换、转储、DB 的改组以及性能监控等功能。这些功能分别由各个实用程序完成。

1.19 从模块结构观察，DBMS 由哪些部分组成？

答：DBMS 由两大部分组成：查询处理器和存储管理器。（解释略）

1.20 DBS 有哪几部分组成？其中 DD 有什么作用？

答：DBS 由 DB、硬件、软件和 DBA 等四个部分组成。（解释略）

在 DBS 中，DD 是存储三级结构描述（即元数据）的 DB。DBMS 的所有工作都要以 DD 中的元数据为依据，也就是所有工作都要通过 DD 访问 DB。

1.21“元数据”与“数据”之间有些什么联系与区别？

答：元数据（metadata）是指“数据的数据”，即数据的描述。DB 中的元数据是指三级模式结构的详细描述。

数据（data），一般是指用户使用的具体值。

1.22 什么是 DBA？DBA 应具有什么素质？DBA 的职责是什么？

答：DBA 是控制数据整体结构的一组人员，负责 DBS 的正常运行，承担创建、监控和维护 DB 结构的责任。

DBA 必须具备下列 4 条素质：熟悉企业全部数据的性质和用途；对所有用户的需求有充分的了解；对系统的性能非常熟悉；兼有系统分析员和运筹学专家的品质和知识。

DBA 的主要职责有 6 点：定义模式；定义内模式；与用户的联络；定义安全性规则；定义完整性规则；DB 的转储与恢复。

1.23 试对 DBS 的全局结构作详细解释。

答：从四个方面解释：

- 数据库用户有四类：DBA，专业用户，应用程序员，终端用户。
- DBMS 的查询处理器有四个模块：DML 编译器，嵌入型 DML 预编译器，DDL 编译器，查询运行核心程序。
- DBMS 的存储管理器有四个模块：授权和完整性管理器，事务管理器，文件管理器，缓冲区管理器。
- 磁盘存储器中有五种数据结构：数据文件，数据字典，索引文件，统计数据组织和日

志。

1.24 使用 DBS 的用户有哪几类？

答：（略，见习题 1.23）

1.25 DBMS 的查询处理器和存储管理器各有哪些功能？

答：（略，见习题 1.23）

1.26 磁盘存储器中有哪五类主要的数据结构？

答：（略，见习题 1.23）

1.27 根据计算机的系统结构，DBS 可分成哪四种?各有什么特点?

答：根据计算机的系统结构，DBS 可分成集中式、C/S 式、并行式和分布式等四种

集中式 DBS 的特点是单点数据（DB 集中在一个场地）单地处理（单个 CPU)。

C/S 式 DBS 的特点是计算机的功能分放在客户机和服务器上（即功能的分布)。客户机上专门实现前端处理和用户界面。服务器上完成事务处理和数据访问控制。

并行式 DBS 的特点是使用多个 CPU 和多个磁盘进行并行操作。

分布式 DBS 的特点是多点数据（DB 分布在多个场地）多点处理（多个 CPU)。数据具有物理分布性和逻辑整体性特点。系统中事务有本地事务（访问本地 DB）和全局事务（访问至少两个场地的 DB）之分。

1.28 DBS 能产生哪些效益？

答：DBS 的应用，使计算机应用深入到社会的每个角落。其效益有以下 7 个方面：灵活性，简易性，面向用户，有效的数据控制，加快应用开发速度，维护方便，标准化。

1.3 自测题

1.3.1 填空题

1．数据管理技术的发展，与_____、_____和_____有密切的联系。

2．文件系统中的数据独立性是指_____独立性。

3．文件系统的缺陷是：_____、_____和_____。

4．就信息处理的方式而言，在文件系统阶段，_____处于主导地位，_____只起着服从程序设计需要的作用；而在数据库方式下，_____占据了中心位置。

5．对现实世界进行第一层抽象的模型，称为_____模型；对现实世界进行第二层抽象的模型，称为_____模型。

6．层次模型的数据结构是_____结构；网状模型的数据结构是_____结构；关系模型的数据结构是_____结构；面向对象模型的数据结构之间可以_____。

7．在层次、网状模型中，用_____导航数据；而在关系模型中，用_____导航数据。

8．数据库的三级模式结构是对_____的三个抽象级别。

9．DBMS 为应用程序运行时开辟的 DB 系统缓冲区，主要用于_____和_____。

10．在数据库技术中，编写应用程序的语言仍然是 C 一类高级语言，这些语言被称为_____语言。

11．在 DB 的三级模式结构中，数据按_____的描述提供给用户，按_____的描述存储在磁盘中，而_____提供了连接这两级的相对稳定的中间观点，并使得两级中的任何一级的改变都不受另一级的牵制。

12．层次、网状的 DML 属于_____语言，而关系型 DML 属于_____语言。

13．DBS 中存放三级结构定义的 DB 称为_____。

14．从模块结构考察，DBMS 由两大部分组成：_____和_____。

15．DBA 有两个很重要的工具：_____和_____。

16．DBS 是_____、_____、_____和_____的集合体。

17．DBS 的全局结构体现了其_____结构。

18．在 DBS 中，DB 在磁盘上的基本组织形式是_____，这样可以充分利用 OS _____的功能。

19．根据计算机的系统结构，DBS 可分成四种类型：_____、_____、_____和_____。
20．数据独立性使得修改 DB 结构时尽量不影响已有的_____。

1.3.2 单项选择题（在备选答案中选出一个正确答案）

1．在 DBS 中，DBMS 和 OS 之间关系是 [ ]
A．并发运行 B．相互调用
C．OS 调用 DBMS D．DBMS 调用 OS

2．在数据库方式下，信息处理中占据中心位置的是 [ ]
A．磁盘 B. 程序 C. 数据 D. 内存

3．DB 的三级体系结构是对_____抽象的三个级别。 [ ]
A. 存储器 B. 数据 C. 程序 D. 外存

4．DB 的三级模式结构中最接近外部存储器的是 [ ]
A. 子模式 B. 外模式 C. 概念模式 D. 内模式

5．DBS 具有“数据独立性”特点的原因是因为在 DBS 中 [ ]
A. 采用磁盘作为外存 B. 采用三级模式结构
C. 使用 OS 来访问数据 D. 用宿主语言编写应用程序

6．在 DBS 中，“数据独立性”和“数据联系”这两个概念之间联系是 [ ]
A．没有必然的联系 B．同时成立或不成立
C．前者蕴涵后者 D．后者蕴涵前者

7．数据独立性是指 [ ]
A．数据之间相互独立 B．应用程序与 DB 的结构之间相互独立
C．数据的逻辑结构与物理结构相互独立 D．数据与磁盘之间相互独立

8．DB 中数据导航是指 [ ]
A．数据之间联系 B．数据之间指针联系
C．从已知数据找未知数据的过程 D．数据的组合方式

9．用户使用 DML 语句对数据进行操作，实际上操作的是 [ ]
A．数据库的记录 B．内模式的内部记录
C．外模式的外部记录 D．数据库的内部记录值

10．对 DB 中数据的操作分成两大类： [ ]
A．查询和更新 B．检索和修改
C．查询和修改 D．插入和修改

1.3.3 问答题

1．试对数据管理技术三个发展阶段作一详细的比较。
2．在用户访问数据库中数据的过程中，DBMS 起着什么作用？
3．什么是“DB 的系统缓冲区”？
4．DBS 中有哪些系统软件？

1.4 自测题答案

1.4.1 填空题答案

1．硬件 软件 计算机应用 2．设备
3．数据冗余 数据不一致 数据联系弱 4．程序设计 数据 数据
5．概念 结构（或逻辑） 6．树 有向图 二维表 嵌套和递归
7．指针 关键码（或外键与主键） 8．数据
9．数据的传输 格式的转换 10．宿主语言(或主语言，host language)
11．外模式 内模式 概念模式 12．过程性 非过程性
13．数据字典（DD） 14．查询处理器 存储管理器
15．一系列实用程序 DD 系统 16．数据库 硬件 软件 DBA
17．模块功能 18. 文件 管理外存(或文件系统)

19．集中式 C/S 式　并行式　分布式　20．应用程序

1.4.2 单项选择题答案

1．D　　2．C　　3．B　　4．D　　5．B

6．A　　7．B　　8．C　　9．C　　10．A

1.4.3 问答题答案

1. 答：数据管理技术三个发展阶段的详细比较见图 1. 1。

	人工管理阶段	文件系统阶段	数据库阶段
时　间	20 世纪 50 年代	60 年代	70 年代
环境 外存	纸带、卡片、	磁带　磁盘	大容量磁盘
环境 软件	汇编语言	3GL、OS	DBMS
计算机应用	科学计算	进入企业管理	企业管理
数据的管理者	用户（程序员）	文件系统	DBS
数据的针对者	面向某一应用程序	面向某一应用	面向现实世界
数据的共享程度	无共享	共享性差、冗余度大	共享性高、冗余度小
数据独立性	无独立性，数据完全依赖于程序	独立性差，有设备独立性	有高度的物理独立性，一定的逻辑独立性
数据的结构化	无结构	记录内有结构，整体结构性差	整体结构化，用数据模型描述

图 1. 1

2．答：在用户访问数据的过程中，DBMS 起着核心的作用，实现“数据三级结构转换”的工作。

3．答：在应用程序运行时，DBMS 在内存为其开辟一个程序工作区，称为“DB 的系统缓冲区”。这个工作区主要用于“数据的传输和格式的转换”。

4．答：DBS 应包括 DBMS、OS、宿主语言和应用开发支撑软件等四部分系统软件。

DBMS 是管理 DB 的软件系统，但对硬盘的管理是由 OS 实现的，因此 DBS 中应包括 DBMS 和 OS 这两个主要的系统软件。

编写应用程序仍然是用 C 一类高级程序设计语言，这些语言在 DBS 中称为宿主语言。

为提高应用程序开发效率，需要像 Dephi、PowerBuilder 一类软件开发工具（即应用开发支撑软件）开发应用程序。这些软件属于 4GL 范畴，可使应用系统开发生产率提高 20~100 倍。

第 2 章 关系模型和关系运算理论

2.1 基本内容分析

2.1.1 本章重要概念

（1）基本概念

关系模型，关键码（主键和外键），关系的定义和性质，三类完整性规则，ER 模型到关系模型的转换规则，过程性语言与非过程性语言。

（2）关系代数

五个基本操作，四个组合操作，七个扩充操作。

（3）关系演算

元组关系演算和域关系演算的原子公式、公式的定义。关系演算的安全性和等价性。

（4）关系代数表达式的优化

关系代数表达式的等价及等价转换规则，启化式优化算法。

（5）关系逻辑

谓词、原子、规则和查询，规则的安全性，用规则模拟关系代数表达式。

2.1.2 本章的重点篇幅

（1）教材中 P56 的例 2.7（关系代数表达式的应用实例）。

（2）教材中 P63 的例 2.19（元组表达式的应用实例）。

（3）教材中 P81 的例 2.36（关系逻辑的规则表示）。

2.1.3 重要内容分析

1．关系代数表达式的运用技巧

（1）一般规则

• 对于只涉及到选择、投影、联接的查询可用下列表达式表示：

$$\pi_{\ldots}(\sigma_{\ldots}(R\times S)) \quad \text{或者} \quad \pi_{\ldots}(\sigma_{\ldots}(R\bowtie S))$$

• 对于否定的操作，一般要用差操作表示，例如“检索不学 C2 课的学生姓名”。

• 对于检索具有“全部”特征的操作，一般要用除法操作表示，例如“检索学习全部课程的学生姓名”。

（2）“检索不学 C2 课的学生姓名”，决不能用下式表示：

$$\pi_{SNAME,\ AGE}(\sigma_{C\#\neq'C2'}(S\bowtie SC))$$

一定要用“差”的形式：

$$\pi_{SNAME,\ AGE}(S) - \pi_{SNAME,\ AGE}(\sigma_{C\#='C2'}(S\bowtie SC))$$

（3）“检索学习全部课程的学生学号”，要用 $\pi_{S\#,\ C\#}(SC)\div\pi_{C\#}(C)$ 表示，

而不能写成 $\pi_{S\#}(SC\div\pi_{C\#}(C))$ 形式。这是因为一个学生学的课程的成绩可能是不一样的。

（4）对于教材 P56 的例 2.7 的 8 个查询语句的关系代数表达式，考生一定要掌握，这是基础。

2．非过程性语言与过程性语言的区别

编程时必须指出“干什么”及“怎么干”的语言，称为过程性语言；编程时只须指出“干什么”，不必指出“怎么干”的语言，称为非过程性语言。

两种语言的主要区别见图 2.1。

过程性语言	非过程性语言
编程时，必须指出“怎么干”	编程时，不必指出“怎么干”
由用户进行数据导航	由系统进行数据导航
单记录处理方式	集合处理方式
属于 3GL 范畴	属于 4GL 范畴
C 语言，层次、网状 DML 等	关系 DML，软件开发工具等

图 2.1

2.2 教材中习题 2 的解答

2.1 名词解释

• 关系模型：用二维表格表示实体集，外键和主键表示实体间联系的数据模型，称为关系模型。

• 关系模式：是对关系的描述，包括模式名、诸属性名、值域名和模式的主键。

• 关系实例：关系模式具体的值，称为关系实例。

• 属性：即字段或数据项，与二维表中的列对应。属性个数，称为元数（arity）。

• 域：属性的取值范围，称为域。

• 元组：即记录，与二维表中的行对应。元组个数，称为基数（cardinality）。

• 超键：能惟一标识元组的属性或属性集，称为关系的超键。

• 候选键：不含有多余属性的超键，称为候选键。

• 主键：正在使用的、用于标识元组的候选键，称为主键。

• 外键：属性集 F 是模式 S 的主键，在模式 R 中也出现，那么称 F 是模式 R 的外键。

• 实体完整性规则：实体的主键值不允许是空值。

• 参照完整性规则：依赖关系中的外键值或者为空值，或者是相应参照关系中某个主键码。

• 过程性语言：编程时必须给出获得结果的操作步骤，即指出“干什么”及“怎么干”的语言。

• 非过程性语言：编程时，只需指出需要什么信息，不必给出具体的操作步骤，即只要指出“干什么”，不必指出“怎么干”的语言。

• 无限关系：指元组个数为无穷多个的关系。

• 无穷验证：验证公式真假时需要进行无限次验证。

● 2.2 在关系模型中，对关系作了哪些规范性限制？

答：对关系作了一下四个限制：

属性值不可分解；没有重复元组；没有行序；使用时有列序。

2.3 为什么关系中的元组没有先后顺序，且不允许有重复元组？

答：由于关系定义为元组的集合，而集合中的元素是没有顺序的，因此关系中的元组也就没有先后的顺序（对用户而言）。这样既能减少逻辑排序，又便于在关系数据库中引进集合论的理论。

每个关系模式都有一个主键，在关系中主键值是不允许重复的。如果关系中有重复元组，那么其主键值肯定相等，起不了惟一标识作用，因此关系中不允许有重复元组。

● 2.4 外键值何时允许空？何时不允许空？

答：在依赖表中，当外键是主键的组成部分时，外键值不允许空；否则外键值允许空。

2.5 笛卡儿积、等值联接、自然联接三者之间有什么区别？

答：笛卡尔积是一个基本操作，而等值联接和自然联接是组合操作。

设关系 R 的元数为 r，元组个数为 m；关系 S 的元数为 s，元组个数为 n。

那么，R×S 的元数为 r+s，元组个数为 m×n；

$R \underset{i\,\theta\,j}{\bowtie} S$ 的元数也是 r+s，但元组个数小于等于 m×n；

R⋈S 的元数小于等于 r+s，元组个数也小于等于 m×n；

2.6 设有关系 R 和 S：

R	A	B	C
	3	6	7
	2	5	7
	7	2	3
	4	4	3

S	A	B	C
	3	4	5
	7	2	3

计算 R∪S，R-S，R∩S，R×S，$\pi_{3,2}$（S），$\sigma_{B<'5'}$(R)，R $\bowtie_{2<2}$ S，R S。

解：

R∪S	A	B	C
	3	6	7
	2	5	7
	7	2	3
	4	4	3
	3	4	5

R－S	A	B	C
	3	6	7
	2	5	7
	4	4	3

R∩S	A	B	C
	7	2	3

R×S	R.A	R.B	R.C	S.A	S.B	S.C
	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	3	4	5
	7	2	3	7	2	3
	4	4	3	3	4	5
	4	4	3	7	2	3

$\pi_{3,2}$（S）	C	B
	5	4
	3	2

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

$R\bowtie_{2<2}S$	R.A	R.B	R.C	S.A	S.B	S.C
	7	2	3	3	4	5

R⋈S	A	B	C
	7	2	3

- 2.7 设有关系 R 和 S：

R	A	B
	a	b
	c	b
	d	e

S	B	C
	b	c
	e	a
	b	d

- 计算 R ⋈ S，R ⋈ S，$\sigma_{A=C}$(R×S)，S ⋉ R。
  $_{B<C}$

- 2.8 假设关系 U 和 V 分别有 m 个元组和 n 个元组，给出下列表达式中可能的最小和最大的元组数量：
  - (1) U∩V　　(2) U∪V
  - (3) U⋈V　　(4) σ_F(U)×V　　（F 为某个条件）

- (5) $\pi_L(U)-V$ （其中 L 为某属性集）
- 解：

操作	最小元组数	最大元组数
(1) $U\cap V$	0	min(m,n)
(2) $U\cup V$	max(m,n)	m+n
(3) $U\bowtie V$	0	m×n
(4) $\sigma_F(U)\times V$	0	m×n
(5) $\pi_L(U)-V$	0	m

2.9 如果 R 是二元关系，那么下列元组表达式的结果是什么？

$\{t|(\exists u)(R(t)\wedge R(u)\wedge(t[1]\neq u[1]\vee t[2]\neq u[2]))\}$

答：当 R 的元组数≥2 时，R 中每个元组都存在与之不相同的元组，因此表达式的结果为关系 R；

当 R 的元组数为 0 或 1 时，表达式的结果为空关系。

2.10 假设 R 和 S 分别是三元和二元关系，试把表达式 $\pi_{1,5}(\sigma_{2=4\vee 3=4}(R\times S))$ 转换成等价的:①汉语查询句子;②元组表达式;③域表达式。

解：⑴ 在关系 R 和 S 的笛卡尔积中，选取第 2 个属性值与第 4 个属性值相等，或者第 3 个属性值与第 4 个属性值相等的那些元组，再取第 1 列和第 5 列组成新的关系。

⑵ 与（R×S）等价的元组表达式是：

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

与 $\sigma_{2=4\vee 3=4}$（R×S）等价的元组表达式是：

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

与 $\pi_{1,5}(\sigma_{2=4\vee 3=4}$（R×S））等价的元组表达式是：

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

再对上述元组表达式化简（消去 t）可得：

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

在熟练后，可以直接写出上式。

⑶ 再转换成域表达式：

$\{w_1\ w_2\mid(\exists u_1)(\exists u_2)(\exists u_3)(\exists v_1)(\exists v_2)(R(u_1u_2u_3)\wedge S(v_1v_2)\wedge(u_2=v_1\vee u_3=v_1)\wedge w_1=u_1\wedge w_2=v_2)\}$

再化简（消去 u_1，v_2）可得：

$\{w_1\ w_2\mid(\exists u_2)(\exists u_3)(\exists v_1)(R(w_1u_2u_3)\wedge S(v_1w_2)\wedge(u_2=v_1\vee u_3=v_1))\}$

2.11 假设 R 和 S 都是二元关系，试把元组表达式$\{t|R(t)\wedge(\exists u)(S(u)\wedge u[1]\neq t[2])\}$转换成等价的：

①汉语查询句子；②域表达式；③关系代数表达式。

答：①在关系 R 中选取第 2 列的值与关系 S 中某个元组的第 1 列值不相等的那些元组，组成新的关系。

②域表达式为：

$\{ t_1t_2 \mid R(t_1t_2) \wedge (\exists u_1)(\exists u_2)(S(u_1u_2) \wedge u_1 \neq t_2)\}$

③关系代数表达式为：

$\pi_{1,2}(\sigma_{2\neq3}(R\times S))$ 或 $\pi_{1,2}(R \underset{2\neq1}{\bowtie} S)$

2.12 试把域表达式$\{ ab \mid R(ab) \wedge R(ba)\}$转换成等价的：⑴汉语查询句子；⑵关系代数表达式；⑶元组表达式。

解：⑴ 在关系 R 中选取属性值交换后仍是 R 中元组的那些元组，组成新的关系。

⑵ 关系代数表达式为：$\pi_{1,2}(\sigma_{1=4 \wedge 2=3}(R\times R))$

也可写成：$R \cap \pi_{2,1}(R)$

⑶ 元组表达式为：$\{ t \mid (\exists u)(\exists v)(R(u) \wedge R(v) \wedge u[1]=v[2] \wedge u[2]=v[1] \wedge t[1]=u[1] \wedge t[2]=u[2])\}$

或：$\{ t \mid (\exists v)(R(t) \wedge R(v) \wedge t[1]=v[2] \wedge t[2]=v[1])\}$

2.13 有两个关系R (A, B, C)和是S (D, E, F)，试把下列关系代数表达式转换成等价的元组表达式:

① $\pi_A(R)$；　② $\sigma_{B=' 17'}(R)$；

③ $R\times S$；　④ $\pi_{A,F}(\sigma_{C=D}(R\times S))$

解：① $\pi_A(R)$：$\{ t \mid (\exists u)(R(u) \wedge t[1]=u[1])\}$

② $\sigma_{B='17'}(R)$：$\{ t \mid R(t) \wedge t[2]='17'\}$

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

④ $\pi_{A,F}(\sigma_{C=D}(R\times S))$：$\{ t \mid (\exists u)(\exists v)(R(u) \wedge S(v) \wedge u[3]=v[1] \wedge t[1]=u[1] \wedge t[2]=v[3])\}$

● 2.14 设有关系 R（A，B，C）和 S（A，B，C），试把下列关系代数表达式转换成等价的域表达式：

① $\pi_A(R)$　② $\sigma_{2=' 17'}(R)$

③ $R\cup S$　④ $R\cap S$

⑤ $R-S$　⑥ $\pi_{1,2}(R) \bowtie \pi_{2,3}(S)$

解：① $\pi_A(R)$：$\{ t_1 \mid (\exists u_2)(\exists u_3)(R(t_1u_2u_3))\}$

② $\sigma_{2=' 17'}(R)$：$\{ t_1t_2t_3 \mid R(t_1t_2t_3) \wedge t_2='17'\}$

③ $R\cup S$：$\{ t_1t_2t_3 \mid R(t_1t_2t_3) \vee S(t_1t_2t_3)\}$

④ $R\cap S$：$\{ t_1t_2t_3 \mid R(t_1t_2t_3) \wedge S(t_1t_2t_3)\}$

⑤ $R-S$：$\{ t_1t_2t_3 \mid R(t_1t_2t_3) \wedge \neg S(t_1t_2t_3)\}$

⑥ $\pi_{1,2}(R) \bowtie \pi_{2,3}(S)$：$\{ t_1t_2t_3 \mid (\exists u_3)(\exists v_1) \mid R(t_1t_2u_3) \wedge S(v_1t_2t_3)\}$

● 2.15 设有关系 R（A，B）和 S（A，C），试把下列域表达式转换成等价的关系代数表达式：

● ① {a ｜（∃b）（R（ab）∧ b=17）}

● ② {abc ｜（R（ab）∧ S（ac））}

- ③ $\{a \mid (\exists b)(R(ab)) \vee (\forall c)((\exists d)(S(dc)) \Rightarrow S(ac))\}$
- ④ $\{a \mid (\exists c)(S(ac) \wedge (\exists b_1)(\exists b_2)(R(ab_1) \wedge R(cb_2) \wedge b_1 > b_2))\}$
- 解：① $\pi_1(\sigma_{2='17'}(R))$
- ② $R \bowtie S$
- ③ $\pi_1(R) \cup (S \div \pi_2(S))$
- ④ $\pi_1(\sigma_{1=3 \wedge 2=5 \wedge 4>6}(S \times R \times R))$

2.16 设两个关系 R（A，B）和 S（A，C）。用 null 表示空值，分别写出等价于下列表达式的元组关系演算表达式：

① R⟕ S； ② R⟖ S； ③ R⟗ S 。

解：① R⟕ S：

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

② R⟖ S:

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

③ R⟗ S:

$\{t \mid (\exists u)(\exists v)(R(u) \wedge S(v) \wedge u[1]=v[1] \wedge t[1]=u[1] \wedge t[2]=u[2] \wedge t[3]=v[2])$
$\vee (\exists u)(\forall v)(R(u) \wedge S(v) \wedge u[1] \neq v[1] \wedge t[1]=u[1] \wedge t[2]=u[2] \wedge t[3]=null)$

2.17 设有三个关系：

S（S#，SNAME，AGE，SEX)
SC（S#，C#，CNAME）
C（C#，CNAME，TEACHER）

试用关系代数表达式表示下列查询语句：

① 检索 LIU 老师所授课程的课程号和课程名。
② 检索年龄大于 23 岁的男学生的学号和姓名。
③ 检索学号为 S3 学生所学课程的课程名与任课教师名。
④ 检索至少选修 LIU 老师所授课程中一门课的女学生姓名。
⑤ 检索 WANG 同学不学的课程的课程号。
⑥ 检索至少选修两门课的学生学号。
⑦ 检索全部学生都选修的课程的课程号与课程名。
⑧ 检索选修课程包含 LIU 老师所授全部课程的学生学号。

解：(1) $\pi_{C\#,\,CNAME}(\sigma_{TNAME='LIU'}(C))$

(2) $\pi_{S\#,\,SNAME}(\sigma_{AGE>'23' \wedge SEX='M'}(SC))$

(3) $\pi_{CNAME,\,TNAME}(\sigma_{S\#='S3'}(SC \bowtie C))$

(4) $\pi_{SNAME}(\sigma_{SEX='F' \wedge TNAME='LIU'}(S \bowtie SC \bowtie C))$

(5) $\pi_{C\#}(C) - \pi_{C\#}(\sigma_{SNAME='WANG'}(S \bowtie SC))$

(6) $\pi_1(\sigma_{1=4 \wedge 2 \neq 5}(SC \times SC))$

(7) $\pi_{C\#,\,CNAME}(C \bowtie (\pi_{S\#,\,C\#}(SC) \div \pi_{S\#}(S)))$

(8) $\pi_{S\#,\,C\#}(SC) \div \pi_{C\#}(\sigma_{TNAME='LIU'}(C))$

2.18 试用元组表达式表示第 2.17 题中各个查询语句。

解：⑴ { t | (∃u) (C(u) ∧ u[3]='LIU' ∧ t[1]=u[1] ∧ t[2]=u[2])}

⑵ { t | (∃u) (S(u) ∧ u[3]＞23 ∧ u[4]='M' ∧ t[1]=u[1] ∧ t[2]=u[2])}

⑶ { t | (∃u) (∃v) (SC(u) ∧ C(v) ∧ u[1]='S3' ∧ u[2]=v[1] ∧ t[1]=v[2] ∧t[2]=v[3])}

（此处自然联接条件 u[2]=v[1]不要遗漏）

⑷ { t | (∃u) (∃v) (∃w) (S(u) ∧ SC(v) ∧ C(w) ∧ w[3]='LIU' ∧ u[4]='F' ∧u[1]=v[1] ∧ v[2]=w[1] ∧ t[1]=u[2])}

（此处自然联接条件 u[1]=v[1]和 v[2]=w[1]不要遗漏）

⑸ { t | (∃u) (∃v) (∀w) (C(u) ∧ S(v) ∧ SC(w) ∧ v[2]='WANG' ∧
(w[1]=v[1] => w[2]≠u[1]) ∧ t[1]=u[1])}

其意思是：在关系 C 中存在一门课程，在关系 S 中存在一个 WANG 同学，在关系 SC 中要求不存在 WANG 同学学这门课程的元组。也就是要求在关系 SC 中，WANG 同学学的课程都不是这门课程（因此在元组表达式中要求全称量词∀）。

⑹ { t | (∃u) (∃v) (SC(u) ∧ SC(v) ∧ u[1]=v[1] ∧ u[2]≠v[2] ∧ t[1]=u[1])}

⑺ { t | (∃u) (∀v) (∃w) (C(u) ∧ S(v) ∧ SC(w) ∧ w[2]=u[1] ∧ w[1]=v[1] ∧t[1]=u[1] ∧ t[2]=u[2])}

其意思是：在关系 C 中找一课程号，对于关系 S 中每一个学生，都应该学这门课（即在关系 SC 中存在这个学生选修这门课的元组）。

⑻ { t | (∃u) (SC(u) ∧ (∀v) (C(v)
∧(v[3]='LIU' => (∃w) (SC(w) ∧ w[1]=u[1] ∧ w[2]=v[1]))) ∧ t[1]=u[1])}

其意思是：在关系 SC 中找一个学号，对于关系 C 中 LIU 老师的每一门课，这个学生都学了（即在关系 SC 中存在这个学生选修这门课的元组）。

由于在括号中出现“=>”符号（包含有“∨”的语义），因此括号中的量词（∃w）就不能随意往左边提了。

2.19 试用域表达式表示第 2.17 题的各个查询语句。

解：① { t_1 t_2 | (∃u_1 u_2 u_3) (C(u_1 u_2 u_3) ∧ u_3='LIU' ∧ t_1=u_1 ∧ t_2=u_2)}

再简化成：{ t_1 t_2 | C(t_1 t_2 'LIU')}

此处(∃u_1 u_2 u_3)是(∃u_1) (∃u_2) (∃u_3) 的简写，下同。

② { t_1 t_2 | (∃u_1 u_2 u_3 u_4) (S(u_1 u_2 u_3 u_4) ∧ u_3>'23' ∧ u_4='M' ∧ t_1=u_1∧ t_2=u_2)}

再简化成：{ t_1 t_2 | (∃u_3) (S(t_1 t_2 u_3 'M') ∧ u_3>'23')}（以下各题的化简略）

③ { t_1 t_2 | (∃u_1 u_2 u_3) (∃v_1 v_2 v_3) (SC(u_1 u_2 u_3) ∧ C(v_1 v_2 v_3) ∧ u_1='s3' ∧ u_2= v_1
∧ t_1=v_2 ∧ t_2=v_3)}

④ { t_1 | (∃u_1 u_2 u_3 u_4) (∃v_1 v_2 v_3) (∃w_1 w_2 w_3) (S(u_1 u_2 u_3 u_4) ∧ SC(v_1 v_2 v_3)
∧ C(w_1 w_2 w_3) ∧w_3='LIU' ∧ u_4='F' ∧ u_1=v_1 ∧ v_2=w_1 ∧ t_2=u_2)}

（⑤~⑧题的域表达式，读者可以很容易写出，此处略）

2.20 设关系 R 和 S 的属性集相同，W 是 R 的属性集的子集，试说明下列等式是否成立，并指出它们的正确表示：

① $\pi_W(R-S) = \pi_W(R) - \pi_W(S)$

② $\pi_W(R \cap S) = \pi_W(R) \cap \pi_W(S)$

③ $\pi_W(R \cup S) = \pi_W(R) \cup \pi_W(S)$

答：① $\pi_W(R-S) = \pi_W(R) - \pi_W(S)$是一个错误的式子。

譬如 R 只有一个元组（1，2，3），S 只有一个元组（1，2，4），W 为 R、S 中前两个属性。

显然 R 和 S 不满足上式。正确的式子应该是 $\pi_W(R-S) = \pi_W(R) - S$。

② $\pi_W(R \cap S) = \pi_W(R) \cap \pi_W(S)$是一个错误的式子。

譬如 R 只有一个元组（1，2，3），S 只有一个元组（1，2，4），W 为 R、S 中前两个属性。显然 R 和 S 不满足上式。此时不可以把 π 操作往里移。

③ $\pi_W(R \cup S) = \pi_W(R) \cup \pi_W(S)$是一个正确的式子。

2.21 在教学数据库的关系 S、SC、C 中，用户有一查询语句：检索女同学选修课程的课程名和任课教师名。

① 试写出该查询的关系代数表达式。

② 画出查询表达式的语法树。

③ 使用启发式优化算法，对语法树进行优化，并画出优化后的语法树。

解：① 关系代数表达式为：

$\pi_{CNAME,\ TEACHER}(\sigma_{SEX='F'}(S \bowtie SC \bowtie C))$

②上述关系代数表达式的语法树如图 2.2 所示。

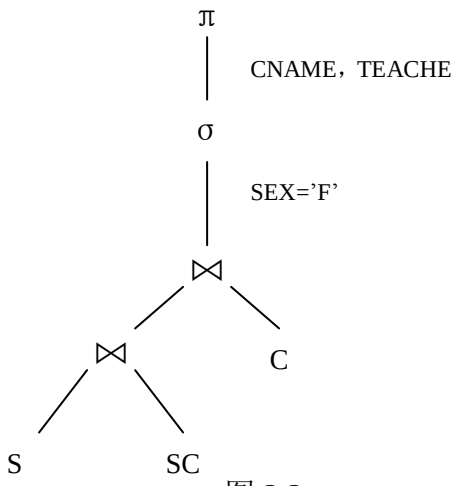


图 2.2

③ 上述的关系代数表达式为：

$\pi_{CNAME,\ TEACHER}(\sigma_{SEX='F'}(\pi_L(\sigma_{S.S\#=SC.S\# \wedge SC.C\#=C.C\#}((S \times SC) \times C))))$

此处 L 为 S、SC、C 中全部属性（公共属性只取一次）。

设 $L1=\pi_{S\#}(\sigma_{SEX='F'}(S))$

$L2=\pi_{S\#,C\#}(SC)$

则优化的关系代数表达式为：

$\pi_{CNAME,\ TEACHER}(\sigma_{SC.C\#=C.C\#}(\pi_{SC.C\#}(\sigma_{S.S\#=SC.S\#}(L1 \times L2)) \times C))$

优化后的语法树如图 2.3 所示。

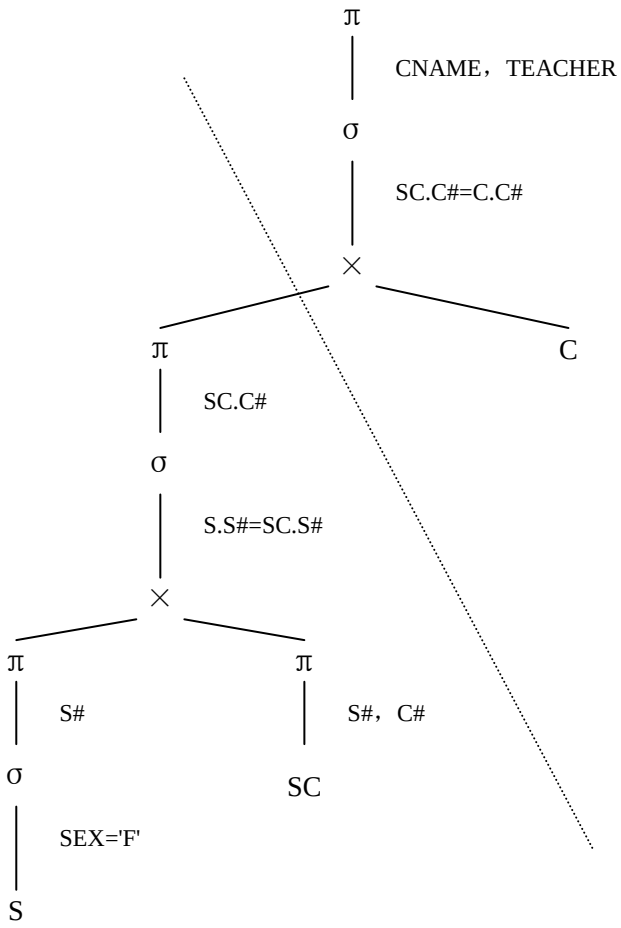


图 2.3

●

● 2.22 为什么要对关系代数表达式进行优化？有哪三条启发式规则？对优化起什么作用？

答：关系代数表达式由关系代数操作组合而成。操作中，以笛卡尔积和联接操作最费时，并生成大量的中间结果。如果直接按表达式书写的顺序执行，必将花费很多时间，并生成大量的中间结果，效率较低。在执行前，由 DBMS 的查询子系统先对关系代数表达式进行优化，尽可能先执行选择和投影操作，以便减少中间结果，并节省时间。

优化工作是由 DBMS 做的，用户书写时不必关心优化一事，仍以简练的形式书写。

● 三条启发式规则是：尽可能早执行选择操作；尽可能早执行投影操作；把笛卡尔积与附近的一连串选择和投影合并起来做。

● 使用这三条规则，可以使计算时尽可能减少中间关系的数据量。

● 2.23 试解释关系逻辑中的名词:

● ·谓词：在关系逻辑中，每一个谓词符号表示了一个关系，但在规则中谓词符号类似于关系演算中的公式。

● ·外延谓词：其关系存储在数据库中的谓词称为“外延谓词”。

● ·内涵谓词：由逻辑规则定义的谓词称为“内涵谓词”。

·外延数据库：用“外延数据库”的缩写 EDB 来引用外延谓词或相应关系。

·内涵数据库：用“内涵数据库”的缩写 IDB 来引用内涵谓词或相应关系。

● ·原子：关系逻辑中的基本成分，称为原子。原子有关系原子和算术原子两种。

● ·关系原子：关系原子是一个谓词符号，带一个参数表，每个参数可以是变量或常量。用大写字母表示谓词符号，用小写字母表示变量，常量用引号括起来。

● ·算术原子：算术原子是算术比较表达式。

·规则：规则是形为 $W \leftarrow P_1 \wedge P_2 \wedge \cdots \wedge P_n$ 的式子，规则有三部分组成：

① 一个称为头部（head）的关系原子；

② 符号“←”，通常读作“if”；

③ 包括一个或多个原子的体（body），称为子目标（subgoal），它可能是关系原子，也可能是算术原子。各子目标用“与”运算符 ∧ 连接，并且子目标前面可以有“非”运算符 ¬，也可以没有。

● ·查询：关系逻辑中的查询是一个或多个规则的聚集，规则之间的顺序无关紧要。

● 2.24 假设 R（A，B，C），S（A，B，C）和 T（A，B，C）为三个关系。试对下列关系代数表达式写出关系逻辑的规则或规则集：

● ① R∪S ② R∩S ③ R－S ④（R∪S）－T

● ⑤（R－S）∩（R－T） ⑥ $\pi_{a,b}(R)$

● 解：① R∪S:W(a，b，c)←R(a，b，c)

● W(a，b，c)←S(a，b，c)

● ② R∩S: W(a，b，c)←R(a，b，c) ∧ S(a，b，c)

● ③ R－S: W(a，b，c)←R(a，b，c) ∧¬ S(a，b，c)

● ④（R∪S）－T: W(a，b，c)←R(a，b，c) ∧¬ T(a，b，c)

● W(a，b，c)←S(a，b，c) ∧¬ T(a，b，c)

● ⑤（R－S）∩（R－T）: W(a，b，c)←R(a，b，c) ∧¬ S(a，b，c) ∧¬ T(a，b，c)

● ⑥ $\pi_{a,b}(R)$: W(a，b)←R(a，b，c)

● 2.25 假设 R（X，Y，Z）为一个关系，试写出下列关系代数表达式 $\sigma_F(R)$ 的关系逻辑规则。其中 F 为以下条件：

● ① x = y ② x<y ∧ y<z ③ x<y ∨ y<z

● ④ ¬(x<y ∨ x>y) ⑤ ¬((x<y ∨ x>y) ∧ y<z)

● ⑥ ¬((x<y ∨ x<z) ∧ y<z)

● 解：① F 为 x = y，此时关系选择规则为：

● W(x，y，z)←R(x，y，z) ∧ x=y

● ② F 为 x<y ∧ y<z，此时关系选择规则为：

● W(x，y，z)←R(x，y，z) ∧x<y ∧ y<z

● ③ F 为 x<y ∨ y<z，此时关系选择规则为：

● W(x，y，z)←R(x，y，z) ∧x<y

● W(x，y，z)←R(x，y，z) ∧y<z

● ④ F 为¬(x<y ∨ x>y)，即 x≥y ∧ x≤y，也就是 x=y,此时关系选择规则为：

● W(x，y，z)←R(x，y，z) ∧ x=y

● ⑤ F 为¬((x<y ∨ x>y) ∧ y<z)，即（x≥y ∧ x≤y）∨ y≥z，即 x=y ∨ y≥z，此时关系选择规则为：

● W(x，y，z)←R(x，y，z) ∧ x=y

● W(x，y，z)←R(x，y，z) ∧ y≥z

● ⑥ F 为¬((x<y ∨ x<z) ∧ y<z)，即（x≥y ∧ x≥z）∨ y≥z，此时关系选择规则为：

● W(x，y，z)←R(x，y，z) ∧ x≥y ∧ x≥z

● W(x，y，z)←R(x，y，z) ∧ y≥z

●

● 2.26 假设R（A，B，C），S（B，C，D）和T（D，E）为三个关系。对每个自然联接写出单一的规则：

● ① R⋈S　② S⋈T　③ （R⋈S）⋈T

● 解：① R⋈S:

● W（a，b，c，d）← R（a，b，c）∧ S（b，c，d）

● ② S⋈T:

● W（b，c，d，e）← R（b，c，d）∧ S（d，e）

● ③ （R⋈S）⋈ T

● W（a，b，c，d，e）← R（a，b，c）∧ S（b，c，d）∧ T（d，e）

● 2.27 对下列每个规则，写出关系代数表达式来定义与规则头部相同的关系：

● ① W（x，y）← Q（x，z）∧ R（z，y）

● ② W（x，y）← Q（x，z）∧ Q（z，y）

● ③ W（x，y）← Q（x，z）∧ R（z，y）∧ x<y

● 解：① $\pi_{1,4}(\sigma_{2=3}(Q\times R))$

● ② $\pi_{1,4}(\sigma_{2=3}(Q\times Q))$

● ③ $\pi_{1,4}(\sigma_{2=3\wedge 1<4}(Q\times R))$

● 2.28 试用关系逻辑的规则来定义第2.17题的各个查询语句。

解：① 检索LIU老师所授课程的课程号和课程名。

W（a，b）← C（a，b，'LIU'）

② 检索年龄大于23岁的男学生的学号和姓名。

W（a，b）← S（a，b，h，'M'）∧ h>23

③ 检索学号为S3学生所学课程的课程名与任课教师名。

W（a，b）← SC（'S3'，e，f）∧ C（e，a，b）

④ 检索至少选修LIU老师所授课程中一门课的女学生姓名。

W（f）← S（e，f，g，'F'）∧ SC（e，h，i）∧ C（h，j,'LIU'）

⑤ 检索WANG同学不学的课程的课程号。

W（a）← C（a，b，d）∧ S（e，'WANG',f，g）∧ ¬SC（e，a，h）

⑥ 检索至少选修两门课的学生学号。

W（a）← SC（a，e，f）∧ SC（a，g，h）∧ e≠g

⑦ 检索全部学生都选修的课程的课程号与课程名。

W（a,b）← C（a，b，e）∧ ¬S（f，g，h，i）∧ ¬SC（f，a，j）

⑧ 检索选修课程包含LIU老师所授全部课程的学生学号。

W（a）← SC（a，b，e）∧ ¬C（f，g，'LIU'）∧ ¬SC（a，f，h）

● 2.29 试撰写短文，对关系运算的三种形式作一评估。

● 答：短文应提到以下几点：

● （1）三种关系运算的理论基础。

● （2）三种关系运算的等价性。

● 关系代数和关系演算在关系代数的五个基本操作的基础上是等价的。

● 关系代数和关系逻辑在表达功能方面不相适应，每个都能表达另一个不能表达的内容。在作了严格的限制后，才能等价。但关系逻辑比关系代数更富于表现力。

- （3）三种关系运算非过程性的强弱不一样。

2.3 自测题

2.3.1 填空题

1．关系中没有行序的原因是_____。

2.

3. 关系模型的基本数据结构是_____，其数据库存储时的基本组织方式是_____。

4．实体完整性规则是对_____的约束，参照完整性规则是对_____的约束。

5．关系代数的理论基础是_____，关系演算的理论基础是_____，关系逻辑的理论基础是_____。

6．关系代数的基本操作是_____。

7．安全运算是指不产生_____和_____的运算。

8．等式 $R \bowtie S = R \times S$ 成立的条件是_____。

9．关系的并、差、交操作，要求两个关系具有_____。

10．一般，在关系代数运算中，当查询涉及到“否定”时，就要用到_____操作；当查询涉及到“全部值”时，就要用到_____操作。

11．如果关系 R 和 S 做自然联接时，只把 R 中原该舍去的元组放到新关系中，那么这种操作称为_____操作。

12．等式 $\pi_L(\sigma_F(E)) = \sigma_F(\pi_L(E))$ 成立的条件是_____。

13．等式 $\pi_{L1}(\pi_{L2}(E)) = \pi_{L1}(E)$ 成立的条件是_____。

14．等式 $\sigma_F(E_1 \times E_2) = E_1 \times \sigma_F(E_2)$ 成立的条件是_____。

15．等式 $\sigma_F(E_1 \bowtie E_2) = \sigma_F(E_1) \bowtie \sigma_F(E_2)$ 成立的条件是_____。

16. 关系逻辑中，外延谓词是指_____，内涵谓词是指_____。

17. 关系逻辑中的“安全条件”是指_____。

18. 设有关系 R（A，B，C），那么与规则 W（c，a）← R（a，b，c）
等价的关系代数操作是_____。

19. 设有关系 R（A，B，C），那么与规则 W（a，b）← R（a，b，'18'）∧b≥'15'
等价的关系代数操作是_____。

20. 设有关系 R（A，B，C）和 S（B，C，D），那么与规则
W（a，d）← R（a，b，c）∧ S（b，c，d）
等价的关系代数操作是_____。

2.3.2 单项选择题（在备选答案中选出一个正确答案）

1．在关系中，“元数”（arity）是指 [ ]
A．行数　B．元组个数　C．关系个数　D．列数

2．在关系中，“基数”（cardinality）是指 [ ]
A．行数　B．属性个数　C．关系个数　D．列数

3．由系统进行数据导航的语言称为 [ ]
A．第三代语言　B．高级程序设计语言
C．过程性语言　D．非过程性语言

4．设关系 R、S、W 各有 10 个元组，那么这三个关系的自然联接的元组个数为 [ ]
A．10　B．30　C．1000　D．不确定（与计算结果有关）

5．设 $W = R \underset{i\,\theta\,j}{\bowtie} S$，且 W、R、S 的元组个数分别为 p、m、n，那么三者之间满足 [ ]
A．p＜（m+n）　B．p≤（m+n）　C．p＜（m×n）　D．p≤（m×n）

6．设关系 R 和 S 的结构相同，且各有 10 个元组，那么这两个关系的并操作结果的元组个数为 [ ]

A．10　　B．小于等于 10　　C．20　　D．小于等于 20

7．设关系 R 和 S 的属性个数分别为 2 和 3，那么 $R \underset{1<2}{\bowtie} S$ 等价于 [ ]

A．$\sigma_{1<2}(R\times S)$　　B．$\sigma_{1<4}(R\times S)$

C．$\sigma_{1<2}(R\bowtie S)$　　D．$\sigma_{1<4}(R\bowtie S)$

8．如果两个关系没有公共属性，那么其自然联接操作 [ ]

A．转化为笛卡尔积操作　　B．转化为联接操作

C．转化为外部并操作　　D．结果为空关系

9．下列式子中，不正确的是 [ ]

A．R－S=R－（R∩S）　　B．R=（R－S）∪（R∩S）

C．R∩S=S－（S－R）　　D．R∩S=S－（R－S）

10．设关系 R 和 S 都是二元关系，那么与元组表达式

{ t | (∃u) (∃v) (R(u) ∧ S(v) ∧ u[1]=v[1] ∧ t[1]=v[1] ∧ t[2]=v[2])}

等价的关系代数表达式是 [ ]

A．$\pi_{3,4}(R\bowtie S)$　　B．$\pi_{2,3}(R\underset{1=3}{\bowtie}S)$

C．$\pi_{3,4}(R\underset{1=1}{\bowtie}S)$　　D．$\pi_{3,4}(\sigma_{1=1}(R\times S))$

11．在元组关系演算中，与公式 $P_1\wedge P_2$ 等价的公式是 [ ]

A．$\neg(P_1\vee P_2)$　　B．$\neg P_1\vee\neg P_2$

C．$\neg(\neg P_1\wedge\neg P_2)$　　D．$\neg(\neg P_1\vee\neg P_2)$

12．在元组关系演算中，与公式（∀s）（P_1（s））等价的公式是 [ ]

A．$\neg(\exists s)(P_1(s))$　　B．$(\exists s)(\neg P_1(s))$

C．$\neg(\forall s)(\neg P_1(s))$　　D．$\neg(\exists s)(\neg P_1(s))$

13．在元组关系演算中，与公式 $P_1=>P_2$ 等价的公式是 [ ]

A．$\neg P_1\vee P_2$　　B．$\neg P_2\vee P_1$

C．$\neg P_1\wedge P_2$　　D．$\neg P_2\wedge P_2$

14．与域演算表达式{ab | R（ab）∧ R（ba）}不等价的关系代数表达式是 [ ]

A．$\pi_{1,2}(\sigma_{1=4\wedge 2=3}(R\times R))$　　B．$\pi_{1,2}(R\underset{1=2\wedge 2=1}{\bowtie}R)$

C．$R\cap\pi_{2,1}(R)$　　D．$\sigma_{1=2}(R)$

15．设 R 和 S 都是二元关系，那么与元组演算表达式

{ t | (∃u) (∃v) (R(u)∧S(v)∧u[2]=v[2]∧t[1]=u[1]∧t[2]=v[1])}

等价的关系代数表达式是 [ ]

A．$\pi_{1,3}(\sigma_{2=4}(R\bowtie S))$　　B．$\pi_{1,3}(\sigma_{2=2}(R\times S))$

C．$\pi_{1,3}(R\underset{2=4}{\bowtie}S)$　　D．$\pi_{1,3}(R\underset{2=2}{\bowtie}S)$

16．设有关系 R（A，B，C）和 S（B，C，D），那么与 R⋈S 等价的关系代数表达式是

[ ]

A．$\sigma_{3=5}(R\underset{2=1}{\bowtie}S)$　　B．$\pi_{1,2,3,6}(\sigma_{3=5}(R\underset{2=1}{\bowtie}S))$

C．$\sigma_{3=5\wedge 2=4}$（R×S））　　D．$\pi_{1,2,3,6}$（$\sigma_{3=2\wedge 2=1}$（R×S））

17．设 R 和 S 都是二元关系，那么与元组演算表达式

{ t | R（t）∧ (∃u) (S(u) ∧ u[1]≠ t[2])}

不等价的关系代数表达式是　　[　]

A．$\pi_{1,2}$（$\sigma_{2\neq 3}$（R×S））　　B．$\pi_{1,2}$（$\sigma_{2\neq 1}$（R×S））

C．$\pi_{1,2}$（$R \underset{2\neq 1}{\bowtie} S$）　　D．$\pi_{3,4}$（$\sigma_{1\neq 4}$（S×R））

18．在关系代数表达式的查询优化中，不正确的叙述是　　[　]

A．尽可能早地执行联接

B．尽可能早地执行选择

C．尽可能早地执行投影

D．把笛卡尔积和随后的选择合并成联接运算

2.3.3 计算题

1．设有关系 R 和 S：

R	A	B	C
	6	4	2
	6	5	3
	5	6	8

S	B	C	D
	4	4	9
	4	2	5
	5	2	6
	5	3	8
	6	8	4

试计算：$R\underset{2=1}{\bowtie}S$，$R\underset{3=2}{\bowtie}S$，$R\underset{1>3}{\bowtie}S$，$R\bowtie S$。

2．设有关系 R 和 S：

R	A	B	C
	2	4	6
	3	2	1
	7	4	4

S	D	E	F
	3	6	9
	3	4	5
	4	4	7

试计算下面四个元组表达式的值：

R1={ t | R（t）∧ t[2]<3 }

R2={ t | (∃u) (R(t) ∧ S(u) ∧ t[1]<u[1])}

R3={ t | (∀u) (R(t) ∧ S(u) ∧ t[3]<u[3])}

R4={ t | (∃u)(∃v)(R(u) ∧ S(v) ∧ u[2]=v[2] ∧ t[1]=u[1] ∧ t[2]=v[3])}

3．在第 2 题的关系 R 和 S 中，试计算下面四个域表达式的值：

R1={ xyz | (∃u)(∃v)(R（xyz）∧ S(uzv)) }

R2={ xy | (∃u) (∃v)(∃w) (R(uxv) ∧ S(uwy) ∧ v>w)}

R3={ xyz | (∃u)(∀v) (R(xyz) ∧ S(uyv) ∧ x<v)}

R4={ xyz | (∃u)(∃v)(∃w)(R(xyu) ∧ S(vwz) ∧ u=w)}

4. 在第 2 题的关系 R 和 S 中，试计算下面四个规则的值：

规则 1：W1（b）← R（a，b，c）∧ a＞c

规则 2：W2（a，b，c，d，f）← R（a，b，c）∧ S（d，b，f）

规则 3：W3（a，b，c）← R（a，b，c）∧ S（d，e，f）∧ b＞d

规则 4：W4（a，b，c）← R（a，b，c）∧ ¬ S（c，e，f）

2.4 自测题答案

2.4.1 填空题答案

1．关系被定义为一个集合　　2．关系中主键值不允许重复

3．关系（或二维表）　文件　　4．主键　　外键　　5．集合

论（或集合代数）　　谓词演算

6．∪、－、×、π和σ　　7．无限关系　　无穷验证

8．R和S没有公共属性　　9．相同的关系模式（或相同的结构）　　10．差除法　　11．左外联接　　12．条件F只涉及到L中的属性　　13．$L1 \subseteq L2$

14．F只涉及到E_2中的属性　　15．F只涉及E_1和E_2中的公共属性

16．其关系存储在数据库中的谓词　　由逻辑规则定义的谓词

17．出现在规则中任何地方的变量必须出现在某个非求反的关系子目标中

18．$W=\pi_{C,A}(R)$　　19．$W=\pi_{A,B}(\sigma_{B\geqslant'15'\wedge C='18'}(R))$

20．$W=\pi_{1,4}(R\bowtie S)$　或　$W=\pi_{1,6}(\sigma_{2=4\wedge 3=5}(R\times S))$

2.4.2 单项选择题答案

1．D	2．A	3．D	4．D	5．D
6．D	7．B	8．A	9．D	10．C
11．D	12．D	13．A	14．D	15．D
16．B	17．B	18．A		

2.4.3 计算题答案

1．答：

R⋈S	A	B	C	D
	6	4	2	5
	6	5	3	8
	5	6	8	4

$R\bowtie_{2=1}S$	A	R.B	R.C	S.B	S.C	D
	6	4	2	4	4	9
	6	4	2	4	2	5
	6	5	3	5	2	6
	6	5	3	5	3	8
	5	6	8	6	8	4

$R\bowtie_{3=2}S$	A	R.B	R.C	S.B	S.C	D
	6	4	2	4	2	5
	6	4	2	5	2	6
	6	5	3	5	3	8
	5	6	8	6	8	4

$R\bowtie_{1>3}S$	A	R.B	R.C	S.B	S.C	D
	6	4	2	4	2	5
	6	4	2	6	8	4
	6	5	3	4	2	5
	6	5	3	6	8	4
	5	6	8	6	8	4

2．答：

R1	A	B	C
	3	2	1

R2	A	B	C
	2	4	6
	3	2	1

R3	A	B	C
	3	2	1
	7	4	4

R4	A	F
	2	5
	2	7
	7	5
	7	7

3．答：

R1	A	B	C
	2	4	6
	7	4	4

R2	B	F
	4	5
	4	7

R3	A	B	C
	2	4	6
	3	2	1

R4	A	B	F
	2	4	9
	7	4	5
	7	4	7

4.答：

W1	R.A	R.B	R.C
	3	2	1
	7	4	4

W2	R.A	R.B	R.C	S.D	S.F
	2	4	6	3	5
	2	4	6	4	7
	7	4	4	3	5
	7	4	4	3	7

W3	R.A	R.B	R.C
	2	4	6
	7	4	4

W4	R.A	R.B	R.C
	2	4	6
	3	2	1

第 3 章 关系数据库语言 SQL

3.1 基本内容分析

3.1.1 本章重要概念

（1）SQL 数据库的体系结构，SQL 的组成。

（2）SQL 的数据定义：SQL 模式、基本表和索引的创建和撤销。

（3）SQL 的数据查询；SELECT 语句的句法，SELECT 语句的三种形式及各种限定，基本表的联接操作，SQL3 中的递归查询。

（4）SQL 的数据更新：插入、删除和修改语句。

（5）视图的创建和撤消，对视图更新操作的限制。

（6）嵌入式 SQL：预处理方式，使用规定，使用技术，卷游标，动态 SQL 语句。

3.1.2 本章的重点篇幅

（1）教材中 P97 的例 3.8（SELECT 语句）。

（2）教材中 P123 的例 3.31 和 P123 的例 3.32（嵌入式 SQL）。

3.1.3 重要内容分析

SELECT 语句是 SQL 的核心内容，对于该语句考生应掌握下列内容。

1．SELECT 语句的来历

在关系代数中最常用的式子是下列表达式：

$$\pi_{A_1,\cdots,A_n}(\sigma_F(R_1\times\cdots\times R_m))$$

这里 R_1、…、R_m 为关系，F 是公式，A_1、…、A_n 为属性。

针对上述表达式，SQL 为此设计了 SELECT—FROM—WHERE 句型：

```
SELECT  A1，…，An
FROM  R1，…，Rm
WHERE  F
```

这个句型是从关系代数表达式演变来的，但 WHERE 子句中的条件表达式 F 要比关系代数中公式更灵活。

2．SELECT 语句中出现的基本表名，应理解为基本表中的元组变量，而列名应理解为元组分量。

3．SELECT 语句的语义有三种情况，下面以学生表 S（S#，SNAME，AGE，SEX）为例说明。

第一种情况：SELECT 语句中未使用分组子句，也未使用聚合操作，那么 SELECT 子句的语义是对查询的结果执行投影操作。譬如：

```
SELECT S#，SNAME
FROM S

WHERE SEX='M';
```

第二种情况：SELECT 语句中未使用分组子句，但在 SELECT 子句中使用了聚合操作，此时 SELECT 子句的语义是对查询结果执行聚合操作。譬如：

```
SELECT COUNT（*），AVG（AGE）
FROM S

WHERE SEX='M';
```

该语句是求男同学的人数和平均年龄。

第三种情况：SELECT 语句使用了分组子句和聚合操作（有分组子句时必有聚合操作），此时 SELECT 子句的语义是对查询结果的每一分组去做聚合操作。譬如：

```
SELECT AGE，COUNT（*）
```

```
FROM S
WHERE SEX='M'
GROUP BY AGE;
```

该语句是求男同学每一年龄的人数。

4. SELECT 语句中使用分组子句的先决条件是要有聚合操作。但执行聚合操作不一定要用分组子句。譬如求男同学的人数，此时聚合值只有一个，因此不必分组。

但同一个聚合操作的值有多个时，必须使用分组子句。譬如求每一年龄的学生人数。此时聚合值有多个，与年龄有关，因此必须分组。

3.2 教材中习题 3 的解答

3.1 名词解释

• 基本表：实际存储在数据库中的表，称为基本表。

• 视图：是从基本表或其他视图中导出的表，它本身不独立存储在数据库中，也就是数据库中只存放视图的定义而不存放视图的数据。

• 实表：是对基本表的别称。

• 虚表：是对视图的别称。

• 相关子查询：SELECT 语句嵌套时，子查询中查询条件依赖于外层查询中的值，因此子查询要反复求值供外层查询使用。这种子查询称为相关子查询。

•联接查询：查询时要从多个基本表中提取数据，此时把多个基本表写在同一层的 FROM 子句中，这种查询形式称为联接查询。

• 嵌套查询：查询时要从多个基本表中提取数据，此时把多个基本表分别放在不同层次上的 FROM 子句中，这种查询形式称为嵌套查询。

• 交互式 SQL：在终端交互方式使用的 SQL 语言。

• 嵌入式 SQL：嵌入在高级语言的程序中使用的 SQL 语言。

• 共享变量：嵌入的 SQL 语句和主语言语句间传递信息的变量，称为共享变量。共享变量先由主语言程序定义，再用 SQL 的说明语句说明，然后 SQL 语句就可使用这些变量。

• 游标：游标是与某一查询相联系的符号名。游标有游标关系和游标指针两层含义。在游标打开时，游标（指针）指向查询结果的第一个记录之前。

• 卷游标：在游标推进时，可以进退自如的游标。

3.2 对于教学数据库的三个基本表

S(S#,SNAME,AGE,SEX)
SC(S#,C#,GRADE)
C(C#,CNAME,TEACHER)

试用 SQL 的查询语句表达下列查询：

①检索 LIU 老师所授课程的课程号和课程名。

②检索年龄大于 23 岁的男学生的学号和姓名。

③检索学号为 S3 学生所学课程的课程名与任课教师名。

④检索至少选修 LIU 老师所授课程中一门课程的女学生姓名。

⑤检索 WANG 同学不学的课程的课程号。

⑥检索至少选修两门课程的学生学号。

⑦检索全部学生都选修的课程的课程号与课程名。

⑧检索选修课程包含 LIU 老师所授课程的学生学号。

解：①
```
SELECT   C#, CNAME
FROM   C
WHERE   TNAME=’LIU’;
```

②
```
SELECT   S#, SNAME
```

```
    FROM   S
    WHERE   AGE>23   AND   SEX='M';
③  SELECT   CNAME，TEACHER
    FROM   SC, C
    WHERE   SC.C#=C.C#   AND   S#='S3';
④  SELECT   SNAME          （联接查询方式）
    FROM   S, SC, C
    WHERE   S.S#=SC.S#    AND   SC.C#=C.C#
        AND   SEX='F' AND   TNAME='LIU';
    或：SELECT   SNAME          （嵌套查询方式）
        FROM   S
        WHERE   SEX='F'
            AND   S#   IN (SELECT   S#
                           FROM   SC
                           WHERE   C#   IN (SELECT   C#
                                            FROM   C
                                            WHERE   TNAME='LIU'));
    或：SELECT   SNAME          （存在量词方式）
        FROM   S
        WHERE   SEX='F'
            AND   EXISTS (SELECT   *
                          FROM   SC
                          WHERE   SC.S#=S.S#
                            AND   EXISTS (SELECT   *
                                          FROM   C
                                          WHERE   C.C#=SC.C#
                                            AND   TNAME='LIU'));
⑤  SELECT   C#
   FROM   C
   WHERE   NOT   EXISTS
           (SELECT   *
            FROM   S, SC
            WHERE   S.S#=SC.S#   AND   SC.C#=C.C#
              AND   SNAME='WANG');
⑥ SELECT   DISTINCT   X.S#
   FROM   SC   AS   X, SC   AS   Y
   WHERE   X.S#=Y.S#   AND   X.C#!=Y.C#;
⑦ SELECT   C#, CNAME
   FROM   C
   WHERE   NOT   EXISTS
           (SELECT   *
            FROM   S
           WHERE   NOT   EXISTS
                         (SELECT   *
                          FROM   SC
                          WHERE   S#=S.S#   AND   C#=C.C#));
```

在 1974 年的 SYSTEM　R 系统中，曾使用过“集合包含”的语法，即

（集合 1）CONTAINS（集合 2）

用这种语法也能写出本题的 SELECT 语句，即：

```
SELECT C#，CNAME
FROM　C
WHERE （SELECT S#　FROM SC　WHERE C#=C.C#）
        CONTAINS
      （SELECT S#　FROM S）;
```

由于判断“（集合 1）CONTAINS（集合 2）”与“NOT EXISTS（（集合 2）EXCEPT（集合 1））”是等价的，因此本题的 SELECT 语句也能这样写：

```
SELECT C#，CNAME
FROM　C
WHERE NOT EXISTS
        （（SELECT S#　FROM S）
          EXCEPT
         （SELECT S#　FROM SC　WHERE C#=C.C#））;
```

⑧
```
SELECT　DISTINCT　S#
FROM　SC　AS　X
WHERE　NOT　EXISTS
      (SELECT　*
       FROM　C
       WHERE　TNAME='LIU'
         AND　NOT　EXISTS
               (SELECT　*
                FROM　SC　AS　Y
                WHERE　Y.S#=X.S#　AND　Y.C#=C.C#));
```

与⑦类似，本题的 SELECT 语句也能这样写：

```
SELECT DISTINCT S#
FROM　SC　X
WHERE NOT EXISTS
        （（SELECT C#　FROM C　WHERE TEACHER='LIU'）
          EXCEPT
         （SELECT C#　FROM　SC　Y　WHERE Y.S#=X.S#））;
```

3.3 对于第 3.2 题中的 8 个查询语句，试给出 SELECT 语句的图示形式。

解：为了说明问题，这里先用高级语言的算法形式表示其执行过程，再给出图示形式。

下面给出④、⑤、⑦、⑧的算法及图示形式。

④ 如果把三个关系 S、SC、C 看成三个文件，那么可以看出这个查询语句的 SELECT 语句实际上是一个三重循环。从而可得这个查询的算法形式如下：

```
for 关系 S 的每个元组 do
{which:=false;
    if S.SEX='F' then
        for 关系 SC 的每个元组，且 NOT which do
            if　SC.S#=S.S#　then
                for 关系 C 的每个元组，且 NOT which do
                    if C.C#=SC.C#，且 TEACHER='LIU' then
                        {print(S.SNAME); which:=true; }
```

};

这个算法可以用图 3.1 表示。

S	S#	SNAME	AGE	SEX
	_X	P.		F

SC	S#	C#	GRADE
	_X	_Y	

C	C#	CNAME	TEACHER
	_Y		LIU

图 3.1

⑤for 关系 S 的每个元组 do

{if S.SNAME=’WANG’ then

for 关系 C 的每个元组 do

{which:=false;

for 关系 SC 的每个元组，且 NOT which do

if SC.S# =S.S# ，且 SC.C#=C.C# then which := true;

if NOT which then print (S.SNAME);

};

这个算法可以用图 3.2 表示。图中“¬”表示“NOT EXISTS”，即“不存在满足此条件的元组”

S	S#	SNAME	AGE	SEX
	_X	WANG		

C	C#	CNAME	TEACHER
	P._Y		

SC	S#	C#	GRADE
¬	_X	_Y	

图 3.2

⑦ for 关系 C 的每个元组 do

{which1 := false;

for 关系 S 的每个元组，且 NOT which1 do

{ which2 := false;

for 关系 SC 的每个元组，且 NOT which2 do

if SC.S# =S.S#， 且 SC.C# =C.C# then which2 := true;

if NOT which2 then which1:=true;

};

if NOT which1 then print (C.C#,C.CNAME);

};

这个算法可以用图 3.3 表示。

C	C#	CNAME	TEACHER
	P._X	P.	

S	S#	SNAME	AGE	SEX
¬	_Y			

SC	S#	C#	GRADE
¬	_Y	_X	

图 3.3

⑧ for 关系 SC 的每个元组 x do

{which1 := false;

for 关系 C 的每个元组 y，且 NOT which1 do

{ if y.TEACHER=’LIU’ then

{ which2 := false;

for 关系 SC 的每个元组 z，且 NOT which2 do

if z.S# =x.S#，且 z.C# =y.C# then which2 := true;

if NOT which2 then which1:=true;

};

if NOT which1 then print (x.S#);

}

};

这个算法可以用图 3.4 表示。

SC	S#	C#	GRADE
	P._X	P.	

C	C#	CNAME	TEACHER
┐	_Y		LIU

SC	S#	C#	GRADE
┐	_X	_Y	

图 3.4

3.4 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：

① R∪S　② R∩S　③ R－S　④ R×S　⑤ $\pi_{A,B}$(R)　$\pi_{B,C}$（S）

⑥ $\pi_{1,6}$（$\sigma_{3=4}$（R×S））　⑦ $\pi_{1,2,3}$（R $\bowtie_{3=3}$ S）　⑧ R÷ π_{C}（S）

解：① (SELECT * FROM R)
UNION
(SELECT * FROM S);

② (SELECT * FROM R)
INTERSECT
(SELECT * FROM S);

③ (SELECT * FROM R)
MINUS
(SELECT * FROM S);

④ SELECT *
FROM R, S;

⑤ SELECT R.A, R.B, S.C
FROM R, S
WHERE R.B=S.B;

⑥ SELECT R.A, S.C
FROM R, S
WHERE R.C=S.A;

⑦ SELECT R.*　（R.*表示 R 中全部属性）
FROM R, S
WHERE R.C=S.C;

⑧ R÷ π_{C}（S）的元组表达式如下：

{ t |（∃u）（∀v）（∃w）（R（u）∧ S（v）∧ R（w）∧ w[1]=u[1] ∧ w[2]=u[2] ∧ w[3]=v[3] ∧ t[1]=u[1] ∧ t[2]=u[2]）}

据此，可写出 SELECT 语句：

```
SELECT A, B
FROM R  RX
WHERE NOT EXISTS
       ( SELECT *
         FROM S
         WHERE NOT EXISTS
               ( SELECT *
                 FROM R  RY
                 WHERE RY.A=RX.A AND RY.B=RX.B AND RY.C=S.C));
```

3.5 设有两个关系 R（A，B）和 S（A，C），试用 SQL 查询语句表示下列域表达式：

① { a |（∃b）（R（ab）∧ b= ‘17’）}

② { abc | R（ab）∧ S（ac）}

③ { a | (∃c)(∃b_1)(∃b_2)(S (ac) ∧R (ab_1) ∧ R (cb_2) ∧ b_1>b_2) }

解：① SELECT A
FROM R
WHERE B=17;

② SELECT R.A, R.B, S.C
FROM R, S
WHERE R.A=S.A;

③ SELECT S.A
FROM S, R RX, R RY
WHERE S.A=RX.A AND RX.B>RY.B;

3.6 试叙述 SQL 语言的关系代数特点和元组演算特点。

答：SQL 的关系代数特点如下：

① 有关系代数运算的并、交、差、自然联接等运算符；

② FROM 子句体现了笛卡尔积操作，WHERE 子句体现了选择操作，SELECT 子句体现了投影操作。

SQL 的元组演算特点如下：

① FROM 子句中的基本表名应视为“元组变量”，属性名应视为“元组分量”；

② 有存在量词 EXISTS 符号。

3.7 试用 SQL 查询语句表达下列对 3.2 题中三个基本表 S、SC、C 的查询：

① 在表 C 中统计开设课程的教师人数。

② 求选修 C4 课程的女学生的平均年龄。

③ 求 LIU 老师所授课程的每门课程的平均成绩。

④ 统计每个学生选修课程的门数（超过 5 门的学生才统计）。要求输出学生学号和选修门数，查询结果按门数降序排列，若门数相同，按学号升序排列。

⑤ 检索学号比 WANG 同学大，而年龄比他小的学生姓名。

⑥ 在表 SC 中检索成绩为空值的学生学号和课程号。

⑦ 检索姓名以 L 打头的所有学生的姓名和年龄。

⑧ 求年龄大于女同学平均年龄的男学生姓名和年龄。

⑨ 求年龄大于所有女同学年龄的男学生姓名和年龄。

解：① SELECT COUNT(DISTINCT TEACHER)
FROM C;

② SELECT AVG(AGE)
FROM S, SC
WHERE S.S#=SC.S# AND C#='C4' AND SEX='F';

③ SELECT C.C#, AVG(GRADE)
FROM SC, C
WHERE SC.C#=C.C# AND TEACHER= 'LIU'
GROUP BY C.C#;

④ SELECT S#, COUNT(C#)
FROM SC
GROUP BY S#
HAVING COUNT(*)>5
ORDER BY 2 DESC, 1;

⑤ SELECT SNAME
FROM S
WHERE S#>ALL(SELECT S#

```
                FROM S
                WHERE SNAME='WANG')
    AND AGE<ALL(SELECT S#
                FROM S
                WHERE SNAME='WANG');
```

⑥
```
SELECT S#, C#
FROM SC
WHERE GRADE IS NULL;
```

⑦
```
SELECT SNAME, AGE
FROM S
WHERE SNAME LIKE 'L%';
```

⑧
```
SELECT SNAME, AGE
FROM S
WHERE SEX='M'
   AND AGE>(SELECT AVG(AGE)
            FROM S
            WHERE SEX='F');
```

⑨
```
SELECT SNAME, AGE
FROM S
WHERE SEX='M'
   AND AGE>ALL(SELECT AGE
               FROM S
               WHERE SEX='F');
```

3.8 对于下面的关系 R 和 S，试求出下列各种联接操作的执行结果：

① R NATURAL INNER JOIN S

② R NATURAL RIGHT OUTER JOIN S

③ R RIGHT OUTER JOIN S USING (C)

④ R INNER JOIN S

⑤ R FULL OUTER JOIN S ON false

R	A	B	C
	a_1	b_1	c_1
	a_2	b_2	c_2
	a_3	b_3	c_3

S	B	C	D
	b_1	c_1	d_1
	b_2	c_2	d_2
	b_4	c_4	d_4

解：

①

A	B	C	D
a_1	b_1	c_1	d_1
a_2	b_2	c_2	d_2

②

A	B	C	D
a_1	b_1	c_1	d_1
a_2	b_2	c_2	d_2
null	b_4	c_4	d_4

③

A	R.B	C	S.B	D
a_1	b_1	c_1	b_1	d_1
a_2	b_2	c_2	b_2	d_2
null	null	c_4	b_4	d_4

④

A	R.B	R.C	S.B	S.C	D
a_1	b_1	c_1	b_1	c_1	d_1
a_1	b_1	c_1	b_2	c_2	d_2
a_1	b_1	c_1	b_4	c_4	d_4
a_2	b_2	c_2	b_1	c_1	d_1
a_2	b_2	c_2	b_2	c_2	d_2
a_2	b_2	c_2	b_4	c_4	d_4
a_3	b_3	c_3	b_1	c_1	d_1
a_3	b_3	c_3	b_2	c_2	d_2
a_3	b_3	c_3	b_4	c_4	d_4

⑤

A	R.B	R.C	S.B	S.C	D
a_1	b_1	c_1	null	null	null
a_2	b_2	c_2	null	null	null
a_3	b_3	c_3	null	null	null
null	null	null	b_1	c_1	d_1
null	null	null	b_2	c_2	d_2
null	null	null	b_4	c_4	d_4

3.9 SQL2 提供 CASE 表达式操作，这个操作类似于程序设计语言中的多分支选择结构，其句法如下：

```
CASE
    WHEN  条件 1  THEN  结果 1
    WHEN  条件 2  THEN  结果 2
    ……
    WHEN  条件 n  THEN  结果 n
    ELSE  结果 m
END
```

如果自上而下“条件 i”首先被满足，那么这个操作返回值“结果 i”（可以是某个表达式的值）；如果没有一个条件被满足，那么返回值“结果 m”。

在基本表 SC（S#，C#，GRADE）中，GRADE 值是百分制。如果欲转换成“成绩等第”，则规则如下：若 GRADE＜40 则等第为 F，若 40≤GRADE＜60 则等第为 C，若 60≤GRADE＜80 则等第为 B，若 80≤GRADE 则等第为 A。试写出下列两个查询语句：

① 检索每个学生的学习成绩，成绩显示时以等第（SCORE）形式出现。

② 检索每个等第的学生人次数。

解：①

```
SELECT S#, C#, CASE
                    WHEN GRADE >= 80 THEN 'A'
                    WHEN GRADE >= 60 THEN 'B'
                    WHEN GRADE >= 40 THEN 'C'
                    ELSE 'F'
                  END AS SCORE
FROM SC;
```

② SELECT SCORE，COUNT（S#）
FROM （SELECT S#，C#，CASE

```
                    WHEN GRADE >= 80 THEN 'A'

                    WHEN GRADE >= 60 THEN 'B'

                    WHEN GRADE >= 40 THEN 'C'

                    ELSE 'F'

                  END
        FROM SC） AS RESULT（S#，C#，SCORE）
```

GROUP BY SCORE;

3.10 用第 3.9 题给出的 CASE 操作在下列更新语句中完成 SC 表中的元组更新：

① 若课程号为 C5 则增加 6 分，若课程号为 C8 则增加 10 分，其他一律增加 5 分。

② 若 C4 课程的成绩低于该门课平均成绩时，提高 5%，否则提高 4%。

解：①

```
UPDATE SC
SET GRADE = GRADE + CASE
                    WHEN C# ='C5' THEN 6

                    WHEN C# ='C8' THEN 10

                    ELSE 5
                END ;
```

②

```
UPDATE SC
SET GRADE = GRADE * CASE
                    WHEN GRADE < （SELECT AVG（GRADE）
                                FROM SC

                                WHERE C# ='C4'）

                    THEN 1.05
                    ELSE 1.04
                END

 WHERE C# ='C4';
```

3.11 设零件之间有组合联系，其关系模式如下：

PART（P#，PNAME，SUBP#，TOTAL）

其属性表示零件编号、零件名称、所需子零件编号及数量。设临时关系 W（P#，SUBP#）的属性分别表示零件编号、这种零件的直接或间接子零件编号。

① 试写出表示关系 W 的规则。

② 写出计算 W 的递归查询语句。

解：①

```
W(x, y)←PART(x, g, y, h)
W(x, y)←W(x, z)∧ W(z, y)
```

②

```
WITH RECURSIVE W(P#, SUBP#) AS
    (SELECT P#, SUBP# FROM PART)
    UNION
```

```
        (SELECT W1.P#, W2.SUBP#
         FROM W AS W1, W AS W2
         WHERE W1.SUBP#=W2.P#)
    SELECT * FROM W;
```

3.12 试用 SQL 更新语句表达对 3.2 题教学数据库中关系 S、SC、C 的更新操作：

① 往关系 C 中插一个课程元组（'C8', 'VC++', 'BAO'）。

② 检索所授每门课程平均成绩均大于 80 分的教师姓名，并把检索到的值送往另一个已存在的表 FACULTY（TNAME）。

③ 在 SC 中删除尚无成绩的选课元组。

④ 把选修 LIU 老师课程的女同学选课元组全部删去。

⑤ 把 MATHS 课不及格的成绩全改为 60 分。

⑥ 把低于所有课程总平均成绩的女同学成绩提高 5%。

⑦ 在表 SC 中修改 C4 课程的成绩，若成绩小于等于 70 分时提高 5%，若成绩大于 70 分时提高 4%（用两种方法实现，一种方法是用两个 UPDATE 语句实现，另一种方法是用带 CASE 操作的一个 UPDATE 语句实现）。

⑧ 在表 SC 中，当某个成绩低于全部课程的平均成绩时，提高 5%。

解：①

```
INSERT INTO C
    VALUES('C8', 'VC++', 'BAO');
```

②

```
INSERT INTO FACULTY(TNAME)
    SELECT DISTINCT TEACHER
    FROM (SELECT TEACHER, C.C#, AVG(GRADE)
          FROM S, SC
          WHERE SC.C#=C.C#
          GROUP BY TEACHER, C.C#)
         AS RESULT(TEACHER, C#, AVG_GRADE) AS X
    WHERE 80<=ALL(SELECT AVG_GRADE
                  FROM RESULT AS Y
                  WHERE Y.TEACHER=X.TEACHER);
```

③

```
DELETE FROM SC
 WHERE GRADE IS NULL;
```

④

```
DELETE FROM SC
WHERE S# IN(SELECT S# FROM S WHERE SEX='F')

  AND C# IN(SELECT C# FROM C WHERE TEACHER='LIU');
```

⑤

```
UPDATE SC
SET GRADE=60
WHERE GRADE<60

  AND C# IN(SELECT C# FROM C WHERE CNAME='MATHS');
```

⑥

```
UPDATE SC
SET GRADE=GRADE*1.05

WHERE S# IN(SELECT S# FROM S WHERE SEX='F')
```

```
      AND GRADE<(SELECT AVG(GRADE) FROM SC);
```

⑦ 用两个 UPDATE 语句实现：

```
UPDATE SC
SET GRADE=GRADE*1.04

WHERE C#='C4' AND GRADE>70;

UPDATE SC
SET GRADE=GRADE*1.05

WHERE C#='C4' AND GRADE<=70;
```

（这两个 UPDATE 语句的顺序不能颠倒。）

用一个 UPDATE 语句实现：

```
UPDATE SC
SET GRADE=GRADE*CASE
          WHEN GRADE>70 THEN 1.04
          ELSE 1.05
          END

WHERE C#='C4';
```

⑧

```
UPDATE SC
SET GRADE=GRADE*1.05
WHERE GRADE<(SELECT AVG(GRADE)
             FROM SC);
```

3.13 设数据库中有三个关系：

职工表 EMP（E#，ENAME，AGE，SEX，ECITY），
其属性分别表示职工工号、姓名、年龄、性别和籍贯。

工作表 WORKS（E#，C#，SALARY），
其属性分别表示职工工号、工作的公司编号和工资。

公司表 COMP（C#，CNAME，CITY），
其属性分别表示公司编号、公司名称和公司所在城市。

试用 SQL 语句写出下列操作：

① 用 CREATE TABLE 语句创建上述三个表，需指出主键和外键。

② 检索超过 50 岁的男职工的工号和姓名。

③ 假设每个职工只能在一个公司工作，检索工资超过 1000 元的男性职工工号和姓名。

④ 假设每个职工可在多个公司工作,检索在编号为 C4 和 C8 公司兼职的职工工号和姓名。

⑤ 检索在“联华公司”工作、工资超过 1000 元的男性职工的工号和姓名。

⑥ 假设每个职工可在多个公司工作,检索每个职工的兼职公司数目和工资总数.显示（E#，NUM，SUM_SALARY），分别表示工号、公司数目和工资总数。

⑦ 工号为 E6 的职工在多个公司工作，试检索至少在 E6 职工兼职的所有公司工作的职工工号。

⑧ 检索联华公司中低于本公司平均工资的职工工号和姓名。

⑨ 在每一公司中为 50 岁以上职工加薪 100 元（若职工为多个公司工作，可重复加）。

⑩ 在 EMP 表和 WORKS 表中删除年龄大于 60 岁的职工有关元组。

解：①

```
CREATE TABLE EMP
( E#        CHAR(4) NOT NULL,
  ENAME         CHAR(8) NOT NULL,
```

```
      AGE      SMALLINT,
      SEX      CHAR(1),
      ECITY    CHAR(20),
      PRIMARY KEY(E#));
CREATE TABLE COMP
    ( C#       CHAR(4) NOT NULL,
      CNAME        CHAR(20) NOT NULL,
      CITY     CHAR(20),
      PRIMARY KEY(C#));
CREATE TABLE WORKS
    ( E#       CHAR(4) NOT NULL,
      C#       CHAR(4) NOT NULL,
     SALARY        SMALLINT,
     PRIMARY KEY(E#, C#),
     FOREIGN KEY(E#) REFERENCES EMP(E#),
     FOREIGN KEY(C#) REFERENCES COMP(C#));
```

②
```
SELECT E#, ENAME
FROM EMP
WHERE AGE>50 AND SEX='M';
```

③
```
SELECT EMP.E#, ENAME
FROM EMP, WORKS
WHERE EMP.E#=WORKS.E# AND SALARY>1000;
```

④
```
SELECT A.E#, A.ENAME
FROM EMP A, WORKS B, WORKS C
WHERE A.E#=B.E# AND B.E#=C.E#
  AND B.C#='C4' AND C.C#='C8';
```

⑤
```
SELECT A.E#, A.ENAME
FROM EMP A, WORKS B, COMP C
WHERE A.E#=B.E# AND B.C#=C.C#
  AND CNAME='联华公司' AND SALARY>1000

  AND SEX='M';
```

⑥
```
SELECT E#, COUNT(C#) AS NUM, SUM(SALARY) AS SUM_SALARY
FROM WORKS
GROUP BY E#;
```

⑦
```
SELECT X.E#
FROM WORKS X
WHERE NOT EXISTS
       (SELECT *
         FROM WORKS Y
         WHERE E#='E6'
           AND NOT EXISTS
```

```
                    (SELECT *
                      FROM WORKS Z
                      WHERE Z.E#=X.E#
                        AND Z.C#=Y.C#));
```

⑧
```
SELECT A.E#, A.ENAME
FROM EMP A, WORKS B, COMP C
WHERE A.E#=B.E# AND B.C#=C.C#
  AND CNAME='联华公司'
  AND SALARY<(SELECT AVG(SALARY)
              FROM WORKS, COMP
              WHERE WORKS.C#=COMP.C#
                AND CNAME='联华公司');
```

⑨
```
UPDATE WORKS
SET SALARY=SALARY+100
WHERE E# IN (SELECT E# FROM EMP WHERE AGE>50);
```

⑩
```
DELETE FROM WORKS
WHERE E# IN (SELECT E# FROM EMP WHERE AGE>60);
DELETE FROM EMP
WHERE AGE>60;
```

3.14 对第 3.13 题中的关系建立一个有关女职工信息的视图 EMP_WOMAN,属性包括(E#,ENAME, C#, CNAME, SALARY)。

然后对视图 EMP_WOMAN 操作，检索每一位女职工的工资总数。(假设每个职工可在多个公司兼职)

解：
```
CREATE VIEW EMP_WOMAN
    AS SELECT A.E#, A.ENAME, C.C#, CNAME, SALARY
       FROM EMP A, WORKS B, COMP C
       WHERE A.E#=B.E# AND B.C#=C.C#
         AND SEX='F';

SELECT E#, SUM(SALARY)
FROM EMP_WOMAN
GROUP BY E#;
```

3.15 在第 1 章中提到的仓库管理数据库中有五个基本表:

零件　PART(P#, PNAME, COLOR, WEIGHT)
项目　PROJECT (J#, JNAME, DATE)
供应商 SUPPLIER (S#, SNAME, SADDR)
供应　P_P(J#, P#, TOTAL)
采购　P_S(P#, S#, QUANTITY)

① 试用 SQL DDL 语句定义上述五个基本表，需说明主键和外键。

② 试将 PROJECT、P_P、PART 三个基本表的联接定义为一个视图 VIEW1,将 PART、P_S、SUPPLIER 三个基本表的联接定义为一个视图 VIEW2。

③ 试在上述两个视图的基础上进行查询操作：

a）检索上海的供应商所供应的零件的编号和名称。

b）检索项目 J4 所用零件的供应商的编号和名称。

解：① CREATE TABLE PART
(P# CHAR(6),
PNAME CHAR(10) NOT NULL,
COLOR CHAR(6),
WEIGHT FLOAT(6),
PRIMARY KEY(P#));

CREATE TABLE PROJECT
(J# CHAR(6),
JNAME CHAR(12) NOT NULL,
DATE DATE,
PRIMARY KEY(J#));

CREATE TABLE SUPPLIER
(S# CHAR(8),
SNAME CHAR(12) NOT NULL,
SADDR VARCHAR(30),
PRIMARY KEY(S#));

CREATE TABLE P_P
(J# CHAR(6),
P# CHAR(6),
TOTAL INTEGER,
PRIMARY KEY (J#, P#)
FOREIGN KEY(J#) REFERENCES PROJECT(J#),
FOREIGN KEY(P#) REFERENCES PART(P#));

CREATE TABLE P_S
(P# CHAR(6),
S# CHAR(8),
QUANTITY INTEGER,
PRIMARY KEY (P#, S#)
FOREIGN KEY(P#) REFERENCES PART(P#),
FOREIGN KEY(S#) REFERENCES SUPPLIER(S#));

② CREATE VIEW VIEW1
AS SELECT A.J#, JNAME, DATE, C.P#, PNAME, COLOR,
WEIGHT, TOTAL
FROM PROJECT A, P_P B, PART C
WHERE A.J#=B.J# AND B.P#=C.P#;

CREATE VIEW VIEW2
AS SELECT A.P#, PNAME, COLOR, WEIGHT, C.S#, SNAME,
SADDR, QUANTITY
FROM PART A, P_S B, SUPPLIER C
WHERE A.P#=B.P# AND B.S#=C.S#;

③ a) SELECT P#, PNAME
FROM VIEW2
WHERE SADDR LIKE ’上海%’;

b) SELECT S#, SNAME
FROM VIEW1, VIEW2
WHERE VIEW1.P#=VIEW2.P#

```
        AND  J#=’J4’;
```

3.16 对于 3.2 题的教学数据库中基本表 SC，建立一个视图：

```
CREATE VIEW S_GRADE(S#，C_NUM，AVG_GRADE)
    AS SELECT S#，COUNT(C#)，AVG（GRADE）
       FROM SC
       GROUP BY S#;
```

试判断下列查询和更新操作是否允许执行。如允许，写出转换到基本表 SC 上的相应操作。

①
```
SELECT *
FROM S_GRADE;
```

②
```
SELECT S#，C_NUM
FROM S_GRADE
WHERE AVG_GRADE>80;
```

③
```
SELECT S#，AVG_GRADE
FROM S_GRADE
WHERE C_NUM >(SELECT C_NUM
              FROM S_GRADE
              WHERE S#='S4');
```

④
```
UPDATE S_GRADE
SET S#='S3'
WHERE S#='S4';
```

⑤
```
DELETE FROM S_GRADE
WHERE C_NUM>4;
```

答：① 允许查询。相应的操作如下：

```
SELECT S#，COUNT（C#） AS C_NUM，AVG（GRADE） AS AVG_GRADE
FROM SC
GROUP BY S#;
```

② 允许查询。相应的操作如下：

```
SELECT S#，COUNT（C#） AS C_NUM
FROM SC
GROUP BY S#
HAVING AVG（GRADE）> 80;
```

③ 允许查询。相应的操作如下：

```
SELECT S#，AVG（GRADE） AS AVG_GRADE
FROM SC
GROUP BY S#
HAVING COUNT（C#）> （SELECT COUNT（C#）
                      FROM SC
                      GROUP BY S#
                      HAVING S#='S4');
```

④ 不允许。C_NUM 是对 SC 中的学生选修门数进行统计，在未更改 SC 表时，要在视图 S_GRADE 中更改门数，是不可能的。

⑤ 不允许。在视图 S_GRADE 中删除选修门数在 4 门以上的学生元组，势必造成 SC 中这些学生学习元组的删除，这不一定是用户的原意，因此使用分组和聚合操作的视图，不允许用户执行更新操作。

3.17 预处理方式对于嵌入式 SQL 的实现有什么重要意义？

答：此时宿主语言的编译程序不必改动，只要提供一个 SQL 函数定义库，供编译时使用。预处理方式只是把源程序中的 SQL 语句处理成宿主语言的函数调用形式。

3.18 在宿主语言的程序中使用 SQL 语句有哪些规定？

答：有三条规定：

① 在程序中要区分 SQL 语句与宿主语言语句，所有 SQL 语句必须加前缀标识“EXEC SQL”以及结束标志“END_EXEC”；

② 允许嵌入的 SQL 语句引用宿主语言的程序变量，而主语句不能引用数据库中的字段变量；

③ SQL 的集合处理方式与宿主语言的单记录处理方式之间要用游标机制协调。

3.19 SQL 的集合处理方式与宿主语言单记录处理方式之间如何协调？

答：用游标机制协调。把 SELECT 语句查询结果定义成游标关系，以使用文件的方式来使用游标关系。与游标有关的 SQL 语句有四个：游标定义，游标打开，游标推进，游标关闭。

3.20 嵌入式 SQL 的 DML 语句何时不必涉及到游标？何时必须涉及到游标？

答：不涉及游标的 DML 语句有下面两种情况：

① INSERT、DELETE、UPDATE 语句，只要加上前缀和结束标志，就能嵌入在宿主语言程序中使用；

② 对于 SELECT 语句，如果已知查询结果肯定是单元值，也可不必涉及游标操作。

涉及游标的 DML 语句有下面两种情况：

① 当 SELECT 语句查询结果是多个元组时，必须用游标机制把多个元组一次一个地传递给主程序处理；

② 对游标指向元组进行修改或删除操作时，也涉及到游标。

3.21 在教学数据库中检索成绩不及格的学生信息，要求显示(S#,SNAME,C#,CNAME,TEACHER)，试编写实现此功能的嵌有 SQL 语句的 C 语言程序段。

解：

```
#define NO_MORE_TUPLES    !(strcmp(SQLSTATE, "02000"))
void sel()
{ EXEC SQL BEGIN DECLARE SECTION;
      char sno[5], cno[5], sname[9], cname[11], teacher[9];
      char SQLSTATE[6];
  EXEC SQL END DECLARE SECTION;
  EXEC SQL DECLARE x CURSOR FOR
      SELECT s.s#, sname,c.c#, cname,teacher
      FROM s,sc,c
      WHERE s.s#=sc.s# and sc.c#=c.c# and grade<60;
  EXEC SQL OPEN scx;
 while (1)
      { EXEC SQL FETCH FROM x
              INTO :sno, :sname,:cno, :cname,:teacher;
        if (NO_MORE_TUPLES) break;
       printf("%s, %s, %s, %s, %s\n", sno, sname,cno, cname,teacher);
      }
  EXEC SQL CLOSE x;
}
```

3.3 练习题

3.3.1 填空题

1. 在 SQL 中，关系模式称为_____，子模式称为_____，元组称为_____，

属性称为_____。

2．SQL 中，表有两种：_____和_____，也称为_____和_____。

3．SQL 中，用户有两种：_____和_____。

4．SQL 中，外模式一级数据结构的基本单位是_____。

5．在“SQL 模式”中，主要成分有_____。

6．基本表中，“主键”概念应该体现其值的_____和_____两个特征。

7．操作“元组 IN（集合）”的语义是_____。

8．表达式中的通配符“%”表示_____，“_”（下划线）表示_____。

9．操作“元组>SOME（集合）”的语义是_____。

10．操作“元组<ALL（集合）”的语义是_____。

11．SQL 有两种使用方式：_____和_____。

12．嵌入式 SQL 的预处理方式，是指预处理程序先对源程序进行扫描，识别出_____，并处理成宿主语言的_____形式。

13．为保证嵌入式 SQL 的实现，通常 DBMS 制造商提供一个_____，供编译时使用。

14．SQL 语句嵌入在 C 语言程序中时，必须加上前缀标识_____和结束标志_____。

15．“卷游标”是指_____。

3.3.2 单项选择题（在备选答案中选出一个正确答案）

1．在 SQL 中，用户可以直接进行查询操作的是 [ ]

A．实表和虚表　　B．基本表和实表

C．视图和虚表　　D．基本表

2．SQL 中，聚合函数 COUNT（列名）用于 [ ]

A．计算元组个数　　B．计算属性的个数

C．对一列中的非空值计算个数　　D．对一列中的非空值和空值计算个数

3．SQL 中，与“NOT IN”等价的操作符是 [ ]

A．=SOME　　B．<>SOME　　C．=ALL　　D．<>ALL

4．元组比较操作（a_1，a_2）>（b_1，b_2）的意义是 [ ]

A．（a_1>b_1）OR（（a_1=b_1）AND（a_2>=b_2））

B．（a_1>=b_1）OR（（a_1=b_1）AND（a_2>=b_2））

C．（a_1>b_1）OR（（a_1=b_1）AND（a_2>b_2））

D．（a_1>=b_1）OR（（a_1=b_1）AND（a_2>b_2）

5．SQL 中，谓词 EXISTS 可用来测试一个集合是否 [ ]

A．有重复元组　　B．有重复的列名

C．为非空集合　　D．有空值

6．对于基本表 EMP（ENO，ENAME，SALARY，DNO）

其属性表示职工的工号、姓名、工资和所在部门的编号。

基本表 DEPT（DNO，DNAME）

其属性表示部门的编号和部门名。

有一 SQL 语句：

```
SELECT  COUNT（DISTINCT DNO）
FROM  EMP;
```

其等价的查询语句是 [ ]

A．统计职工的总人数　　B．统计每一部门的职工人数

C．统计职工服务的部门数目　　D．统计每一职工服务的部门数目

7．对于第 6 题的两个基本表，有一个 SQL 语句：

```
SELECT  ENO，ENAME
FROM  EMP
```

```
WHERE  DNO  NOT  IN
       （SELECT  DNO
         FROM  DEPT
         WHERE  DNAME='金工车间';
```

其等价的关系代数表达式是：[ ]

A．$\pi_{ENO,\ ENAME}$（$\sigma_{DNAME\neq'金工车间'}$（EMP⋈DEPT））

B．$\pi_{ENO,\ ENAME}$（$EMP \underset{DNAME\neq'金工车间'}{\bowtie} DEPT$）

C．$\pi_{ENO,\ ENAME}$（EMP）－$\pi_{ENO,\ ENAME}$（$\sigma_{DNAME='金工车间'}$（EMP⋈DEPT））

D．$\pi_{ENO,\ ENAME}$（EMP）－$\pi_{ENO,\ ENAME}$（$\sigma_{DNAME\neq'金工车间'}$（EMP⋈DEPT））

8．对于第 6 题的两个基本表，有一个 SQL 语句：

```
UPDATE  EMP
SET  SALARY=SALARY*1.05
WHERE  DNO='D6'
    AND  SALARY<（SELECT  AVG（SALARY）
                  FROM  EMP）;
```

其等价的修改语句为 [ ]

A．为工资低于 D6 部门平均工资的所有职工加薪 5%

B．为工资低于整个企业平均工资的职工加薪 5%

C．为在 D6 部门工作、工资低于整个企业平均工资的职工加薪 5%

D．为在 D6 部门工作、工资低于本部门平均工资的职工加薪 5%

9．有关嵌入式 SQL 的叙述，不正确的是 [ ]

A．宿主语言是指 C 一类高级程序设计语言

B．宿主语言是指 SQL 语言

C．在程序中要区分 SQL 语句和宿主语言语句

D．SQL 有交互式和嵌入式两种使用方式

10．嵌入式 SQL 实现时，采用预处理方式是 [ ]

A．把 SQL 语句和主语言语句区分开来

B．为 SQL 语句加前缀标识和结束标志

C．识别出 SQL 语句，并处理成函数调用形式

D．把 SQL 语句编译成二进制码

11．允许在嵌入的 SQL 语句中，引用宿主语言的程序变量，在引用时 [ ]

A．直接引用

B．这些变量前必须加符号“*”

C．这些变量前必须加符号“:”

D．这些变量前必须加符号“&”

12．如果嵌入的 SELECT 语句的查询结果肯定是单元组，那么嵌入时 [ ]

A．肯定不涉及游标机制

B．必须使用游标机制

C．是否使用游标，由应用程序员决定

D．是否使用游标，与 DBMS 有关

13．卷游标的推进语句“EXEC SQL FETCH RELATIVE -4”表示 [ ]

A．把游标移向查询结果的第 4 行

B．把游标移向查询结果的倒数第 4 行
C．把游标从当前位置推进 4 行
D．把游标从当前位置返回 4 行

14．卷游标的推进语句“EXEC SQL FETCH ABSOLUTE -3 ”表示 [ ]
A．把游标移向查询结果的第 3 行
B．把游标移向查询结果的倒数第 3 行
C．把游标从当前位置推进 3 行
D．把游标从当前位置返回 3 行

3.3.3 简答题

1．试叙述 SQL 的关系代数特点和元组演算特点。
2．SQL 语言对于“查询结果是否允许存在重复元组”是如何实现的？
3．试对 SELECT 语句中使用的基本表名和列名的语义作详细的解释。
4．SELECT 语句中，何时使用分组子句，何时不必使用分组子句？

3.4 练习题答案

3.4.1 填空题答案

1．基本表 视图 行 列
2．基本表 视图 实表 虚表
3．应用程序 终端用户
4．视图
5．基本表、视图、索引、完整性规则等
6．惟一 非空
7．若元组在集合中，其值为 true，否则为 false
8．与零个或多个字符组成的字符串匹配 与单个字符匹配
9．若元组值大于集合中某一元组值，则其值为 true，否则为 false
10．若元组值小于集合中每一元组值，则其值为 true，否则为 false
11．交互式 SQL 嵌入式 SQL
12．SQL 语句 函数调用
13．SQL 函数定义库
14．EXEC SQL 分号（;）
15．可以进退自如的游标（即可随意推进或返回）

3.4.2 单项选择题答案

1．A 2．C 3．D 4．C 5．C 6．C 7．C
8．C 9．B 10．C 11．C 12．C 13．D 14．B

3.4.3 简答题答案

1．答：SQL 的 SELECT 语句的基本句法来自于关系代数表达式 $\pi_L(\sigma_F(R_1\times\cdots\times R_m))$，并且 SQL 中有并（UNION）、交（INTERSECT）和差（EXCEPT）等操作，因此 SQL 具有关系代数特点。

SELECT 语句中出现的基本表名，都应该理解成基本表中的元组变量，而列名应理解成元组分量，这样 SQL 就具有了元组演算的特点。

2．答：对于 SELECT 语句中 SELECT 子句，若用“SELECT DISTINCT”形式，则查询结果中不允许有重复元组；若不写 DISTINCT 字样，则查询结果中允许出现重复元组。

3．答：在基本 SQL 中，SELECT 语句中使用的基本表名都应该理解成表中的元组变量，而列名就成了元组分量。这样就使 SELECT 语句带有元组演算的特点。

（注：实际上，在基本 SQL 中，把关系变量和元组变量混为一谈了。这在面向对象数据库中得到了纠正，在引用表时，都要为表定义一个元组变量。）

4．答：SELECT 语句中使用分组子句的先决条件是要有聚合操作。当聚合操作值与其他属性

的值无关时，不必使用分组子句。譬如求男同学的人数。此时聚合值只有一个，因此不必分组。

当聚合操作值与其他属性的值有关时，必须使用分组子句。譬如求每一性别的人数。此时聚合值有两个，与性别有关，因此必须分组。

第 4 章 模式设计理论

4.1 基本知识点

4. 1. 1 本章重要概念

（1）关系模式的冗余和异常问题。

（2）FD 的定义、逻辑蕴涵、闭包、推理规则、与关键码的联系；平凡的 FD；属性集的闭包；推理规则的正确性和完备性；FD 集的等价；最小依赖集。

（3）无损分解的定义、性质、测试；保持依赖集的分解。

（4）关系模式的范式：1NF，2NF，3NF，BCNF。分解成 2NF、3NF 模式集的算法。

（5）MVD、4NF、JD 和 5NF 的定义。

4.1.2 本章的重点篇幅

（1）教材中 P148 的例 4.13。（无损联接和保持 FD 的例子）

（2）教材中 P149 的例 4.14 和 P150 的例 4.15。（分解成 2NF 和 3NF 的例子）

4.2 教材中习题 4 的解答

4. 1 名词解释

• 数据冗余：指同一个数据在系统中多次重复出现。

• 函数依赖（FD）：在关系模式 R(U)中，FD 是形为 X→Y 的一个命题，只要 r 是 R 的当前关系，对 r 中任意两个元组 t 和 s，都有 t[X]=s[X]蕴涵 t[Y]=s[Y]，那么称 FD X→Y 在关系模式 R(U)中成立。

• 平凡的 FD：如果 X→Y，且 Y⊆X, 则称 X→Y 是一个“平凡的 FD”。

• FD 集 F 的闭包 F^+：被 F 逻辑蕴涵的函数依赖全体构成的集合，称为 F 的闭包，记为 F^+，即 $F^+=\{ X→Y \mid F⊨X→Y\}$。

• 属性集 X 的闭包 X^+：从已知的 FD 集 F 使用 FD 推理规则推出的所有满足 X→A 的属性 A 的集合，称为 X 的闭包，记为 X^+，即 $X^+=\{$ 属性 A | X→A 在 F^+中 $\}$。

• FD 的逻辑蕴涵：如果从已知的 FD 集 F 能推导出 X→Y 成立，那么称 F 逻辑蕴涵 X→Y，记为 F ⊨X→Y。

• FD 集的等价：对于两个 FD 集 F 和 G，有 $F^+=G^+$，则称 F 和 G 是等价的依赖集。

• 最小依赖集：设 F 是属性集 U 上的 FD 集，F_{min} 是 F 的最小依赖集，那么 F_{min} 应满足下列四个条件：$F_{min}^+=F^+$；每个 FD 的右边都是单属性；F_{min} 中没有冗余的 FD；每个 FD 的左边没有冗余的属性。

• 无损分解：设关系模式 R，F 是 R 上的 FD 集，$\rho=\{ R_1, \cdots, R_k \}$ 是 R 的一个分解。如果对 R 中满足 F 的每一关系 r，都有 $r=\bowtie_{i=1}^{k} \pi_{R_i}(r)$，那么称分解 ρ 相对 F 是“无损分解”。

• 泛关系假设：指数据库中每一个关系都是全部属性构成的关系的投影，此时，由全部属性构成的关系称为泛关系。

• chase 过程：根据已知 FD 集，对 R 分解成 ρ 构造的初始表格的值进行修改，使之符合 FD 集，这个过程称为 chase 过程。

• 保持 FD：设关系模式 R，F 是 R 上的 FD 分解，$\rho=\{ R_1, \cdots, R_k \}$ 是 R 的一个分解，如果有 $\bigcup_{i=1}^{k} \pi_{R_i}(F) ⊨F$，那么称分解 ρ 保持 FD 集 F。

• 1NF：如果关系模式 R 的每个关系 r 的属性值都是不可分的原子值，那么称 R 是 1NF 的模式。

• 2NF：如果 R 是 1NF 的模式，且每个非主属性完全函数依赖于 R 的候选键，那么称 R 是 2NF 的模式。

• 3NF：如果 R 是 1NF 的模式，且每个非主属性都不传递依赖于 R 的候选键，那么称 R 是 3NF 的模式。

• BCNF：如果 R 是 1NF 的模式，且每个属性都不传递依赖于 R 的候选键，那么称 R 是 BCNF 的模式。

• 4NF：设 D 是关系模式 R 上成立的 FD 和 MVD 集合。如果 D 中每个非平凡的 MVD X→→Y 的左部 X 都是 R 的超键，那么称 R 是 4NF 模式。

• 5NF：如果关系模式 R 的每个 JD 均由 R 的候选键蕴涵，那么称 R 是 5NF 的模式。

• 多值依赖（MVD）：设关系模式 R(U)，X 和 Y 是 U 的子集，Z=U-X-Y。对于 R 的关系 r，若在 r 中存在元组（x，y_1，z_1）和（x，y_2，z_2），就也应存在元组（x，y_2，z_1）和（x，y_1，z_2），那么称 MVD X→→Y 在模式 R 上成立。

• 联接依赖（JD）：设关系模式 R(U)，R_1、…、R_n 是 U 的子集，并满足 U=R_1∪…∪R_n，ρ = { R_1，…，R_n } 是 R 的一个分解。如果对于 R 的每个关系 r 都有 $m_ρ$（r）=r，那么称 JD *（R_1，…，R_n）在模式 R 上成立。

4.2 用 A1、A2 和 A3 三条推理规则来证明 4.2.3 节中的定理 4.2(推理规则 A4～A8)。

(1) A4（合并性，union)：{ X→Y，X→Z } ⊨X→YZ。

证明：已知 X→Y，根据 A2，两边用 X 扩充，得到 X→XY。从已知 X→Z，根据 A2 两边用 Y 扩充，得到 XY→YZ。再根据 A3，从 X→XY 和 XY→YZ 可得到 X→YZ。

(2) A5（分解性，decomposition)：{ X→Y，Z⊆Y } ⊨X→Z。

证明：已知 Z⊆Y，可得 Y→Z。从 Y→Z 和已知 X→Y，可得 X→Z 成立。

(3) A6（伪传递性)：{ X→Y，WY→Z } ⊨WX→Z。

证明：已知 X→Y，根据 A2，两边用 W 扩充，得到 WX→WY。据 WX→WY 和已知的 WY→Z,。再根据 A3，可得 WX→Z 成立。

(4) A7（复合性，composition)：{ X→Y，W→Z } ⊨XW→YZ。

证明：已知 X→Y，根据 A2，两边用 W 扩充，得到 WX→WY。从已知 W→Z，根据 A2 两边用 Y 扩充，得到 WY→YZ。再根据 A3，从 WX→WY 和 WY→YZ 可得到 XW→YZ 成立。

(5) A8 { X→Y，W→Z } ⊨X∪（W－Y）→YZ。

证明：已知 X→Y，根据 A2，两边用（W-Y）扩充，得到 X∪（W-Y）→Y∪（W-Y)，而 Y∪（W-Y）=WY，因此有 X∪（W-Y）→WY。从已知 W→Z，根据 A2 两边用 Y 扩充，得到 WY→YZ。再根据 A3，从 X∪（W-Y）→WY 和 WY→YZ 可得到 X∪（W－Y）→YZ。

4.3 对函数依赖 X→Y 的定义加以扩充，X 和 Y 可以为空属性集，用 φ 表示，那么 X→φ，φ→Y，φ→φ 的含义是什么？

答：据推理规则的自反律可知，X→φ 和 φ→φ 是平凡的 FD，总是成立的。

而 φ→Y 表示在当前关系中，任意两个元组的 Y 值相等，也就是当前关系的 Y 值都相等。

4.4 设关系模式 R 有 n 个属性，在模式 R 上可能成立的函数依赖有多少个？其中平凡的 FD 有多少个？非平凡的 FD 有多少个？

解：这个问题是排列组合问题。FD 形为 X→Y，从 n 个属性值中选择属性组成 X 共有 C_n^0 +C_n^1 + … +C_n^n =2^n 种方法；同理，组成 Y 也有 2^n 种方法。因此组成 X→Y 形式应该有 $2^n \cdot 2^n=4^n$ 种方法。即可能成立的 FD 有 4^n 个。

平凡的 FD 要求 Y⊆X，组合 X→Y 形式的选择有：

$C_n^0 \cdot C_0^0 + C_n^1 \cdot (C_1^0 + C_1^1) + C_n^2 \cdot (C_2^0 + C_2^1 + C_2^2) + \cdots + C_n^n (C_n^0 + C_n^1 + \cdots C_n^n)$

$= C_n^0 \cdot 2^0 + C_n^1 \cdot 2^1 + C_n^2 \cdot 2^2 + \cdots + C_n^n \cdot 2^n = (1+2)^n = 3^n$

即平凡的 FD 有 3^n。因而非平凡的 FD 有 $4^n - 3^n$ 个。

4.5 已知关系模式 R（ABC），F 是 R 上成立的 FD 集，F=｛ A→B，B→C ｝，试写出 F 的闭包 F^+。

解：据已知条件和推理规则，可知 F^+有 43 个 FD：

A→φ　AB→φ　AC→φ　ABC→φ　B→φ　C→φ

A→A　AB→A　AC→A　ABC→A　B→B　C→C

A→B　AB→B　AC→B　ABC→B　B→C　φ→φ

A→C　AB→C　AC→C　ABC→C　B→BC

A→AB　AB→AB　AC→AB　ABC→AB　BC→φ

A→AC　AB→AC　AC→AC　ABC→AC　BC→B

A→BC　AB→BC　AC→BC　ABC→BC　BC→C

A→ABC　AB→ABC　AC→ABC　ABC→ABC　BC→BC

4.6 设关系模式 R（ABCD），F 是 R 上成立的 FD 集，F=｛ A→B，C→B ｝，则相对于 F，试写出关系模式 R 的关键码。并说明理由。

解：R 的关键码为 ACD。因为从已知的 F，只能推出 ACD→ABCD。

4.7 设关系模式 R(ABCD)上 FD 集为 F，并且 F=｛AB→C，C→D，D→A｝。

① 试从 F 求出所有非平凡的 FD。

② 试求 R 的所有候选键。

③ 试求 R 的所有不是候选键的超键。

解：① 从已知的 F 可求出非平凡的 FD 有 76 个。

譬如，左边是 C 的 FD 有 6 个：C→A，C→D，C→AD，C→AC，C→CD，C→ACD。

左边是 D 的 FD 有 2 个：D→A，D→AD。

左边是 AB 的 FD 有 12 个：AB→C，AB→D，AB→CD，AB→AC，……。

感兴趣的读者可以自行把这 76 个 FD 写齐。

② 候选键是能函数决定所有属性的不含多余属性的属性集。根据这个概念可求出 R 的候选键有三个：AB、BC 和 BD。

③ R 的所有不是候选键的超键有四个：ABC、ABD、BCD 和 ABCD。

4.8 试举出反例说明下列规则不成立：

① ｛ A→B ｝⊨｛ B→A ｝

② ｛ AB→C，A→C ｝⊨｛ B→C ｝

③ ｛ AB→C ｝⊨｛ A→C ｝

答：设有三个关系：

r_1	A	B
	1	1
	2	1

r_2	A	B	C
	2	1	2
	2	2	2
	3	2	3

r_3	A	B	C
	1	2	3
	1	3	4

（1）在关系 r_1 中，A→B 成立，但 B→A 不成立。

（2）在关系 r_2 中，AB→C 和 A→C 成立，但 B→C 不成立

（3）在关系 r_3 中，AB→C 成立，但 A→C 不成立。

4.9 设关系模式 R（ABCD），F 是 R 上成立的 FD 集，F= {A→B，B→C}，

① 试写出属性集 BD 的闭包$(BD)^+$。

② 试写出所有左部是 B 的函数依赖（即形为“B→？”）。

解：①从已知的 F，可推出 BD→BCD，所以$(BD)^+$=BCD。

②由于 B^+=BC，因此左部是 B 的 FD 有四个：

B→φ，B→B，B→C，B→BC。

4.10 设关系模式 R(ABCDE)上 FD 集为 F，并且 F= {A→BC，CD→E，B→D，E→A}。

① 试求 R 的候选键。

② 试求 B^+的值。

解：① R 的候选键有四个：A、E、CD 和 BC。

② B^+=BD。

4.11 设有关系模式 R（ABC），其关系 r 如图 4.1 所示。

① 试判断下列三个 FD 在关系 r 中是否成立？

A→B　　BC→A　　B→A

② 根据关系 r，你能断定哪些 FD 在关系模式 R 上不成立？

A	B	C
1	2	3
4	2	3
5	3	3

图 4.1

解：①在关系 r 中，A→B 成立，BC→A 不成立，B→A 不成立。

②在关系 r 中，不成立的 FD 有:B→A，C→A，C→B，C→AB，BC→A。

4.12 设关系模式 R（ABC）分解成 ρ = { AB，BC }，如果 R 上的 FD 集 F= { A→B }，那么这个分解是损失分解。试举出 R 的一个关系 r，不满足 $m_ρ$（r）=r。

解：这个反例 r 可以举测试时的初始表格：

	A	B	C
AB	a_1	a_2	b_{13}
BC	b_{21}	a_2	a_3

$π_{AB}$（r）⋈ $π_{BC}$（r）有四个元组：

A	B	C
a_1	a_2	b_{13}
a_1	a_2	a_3
b_{21}	a_2	b_{13}
b_{21}	a_2	a_3

即 $m_ρ$（r）≠r。

4.13 试解释数据库“丢失信息”与“未丢失信息”两个概念。“丢失信息”与“丢失数据”有什么区别？

答：数据库中丢失信息是指 r≠$m_ρ$（r），未丢失信息是指 r=$m_ρ$（r）。

丢失信息是指不能辨别元组的真伪，而丢失数据是指丢失元组。

4.14 设关系模式 R（ABC），F 是 R 上成立的 FD 集，F= { A→C，B→C }，试分别求 F 在模式 AB 和 AC 上的投影。

答：$π_{AB}$（F）= φ（即不存在非平凡的 FD）

$π_{AC}$（F）={ A→C }

4.15 设关系模式 R（ABC），F 是 R 上成立的 FD 集，F= { B→A，C→A }，ρ = { AB，

BC ｝是 R 上的一个分解，那么分解ρ是否保持 FD 集 F？并说明理由。

答：已知 F={ B→A，C→A }，而 π_{AB}（F）={ B→A }， π_{BC}（F）=φ，

显然，分解ρ丢失了 FD　C→A。

4.16 设关系模式 R（ABC），F 是 R 上成立的 FD 集，F=｛ B→C，C→A ｝，那么分解ρ=｛ AB，AC ｝相对于 F，是否无损分解和保持 FD？并说明理由。

答：①已知 F={ B→C，C→A }，

而 π_{AB}（F）=φ， π_{AC}（F）={ C→A }

显然，这个分解丢失了 FD　B→C

② 用测试过程可以知道，ρ相对于 F 是损失分解。

4.17 设关系模式 R(ABCDEG)上 FD 集为 F，并且 F=｛D→G，C→A，CD→E，A→B}。

① 求 D^+，C^+，A^+，$(CD)^+$，$(AD)^+$，$(AC)^+$，$(ACD)^+$。

② 试求 R 的所有候选键。

③ 用ρ1=｛CDEG，ABC｝替换 R，这个分解有什么冗余和异常现象？

④ 用ρ2=｛DG，AC，CDE，AB｝替换 R，这个分解是无损分解吗？

⑤ 用ρ3=｛CDE，AC，DG，BCD｝替换 R，先求 F 在ρ3 的每个模式上的投影 π_{R_i}（F），再判断分解ρ3 保持 FD 吗？

解：① D^+=DG，C^+=ABC，A^+=AB，$(CD)^+$=ABCDEG，$(AD)^+$=ABDG，$(AC)^+$=ABC，$(ACD)^+$=ABCDEG。

② R 的候选键只有一个：CD。

③ 用ρ1=｛CDEG，ABC｝替换 R，在模式 CDEG 中，有局部依赖 CD→G，此时在关系中，一个 D 值只有一个 G 值，但当这个 D 值与 10 个 C 值对应时，就要出现 10 个元组，则 G 值就要重复 10 次。

在模式 ABC 中，有传递依赖（C→A 和 A→B），此时在关系中，一个 A 值只有一个 B 值，但当这个 A 值与 10 个 C 值对应时，就要出现 10 个元组，则 B 值就要重复 10 次。

④ 用ρ2=｛DG，AC，CDE，AB｝替换 R，据 chase 过程可知，相对于 F，R 分解成ρ是无损分解。

⑤ 用ρ3=｛CDE，AC，DG，BCD｝替换 R，

则 F 在模式 CDE 上的投影为｛CD→E｝，F 在模式 AC 上的投影为｛C→A｝，

F 在模式 DG 上的投影为｛D→G｝，F 在模式 BCD 上的投影为｛C→B｝，

显然从这四个投影集中的 FD 推不出原来 F 中的 A→B，因此分解ρ3 不保持 FD 集。

4.18 设关系模式 R（ABCD），F 是 R 上成立的 FD 集，F=｛ A→B，B→C，A→D，D→C ｝，ρ=｛ AB，AC，BD ｝是 R 的一个分解。

① 相对于 F，ρ是无损分解吗？为什么？

② 试求 F 在ρ的每个模式上的投影。

③ ρ保持 F 吗？为什么？

答：①用测试过程可以知道，ρ相对于 F 是损失分解。

② π_{AB}（F）={ A→B }， π_{AC}（F）={ A→C }， π_{BD}（F）=φ。

③显然，分解ρ不保持 FD 集 F，丢失了 B→C、A→D 和 D→C 等三个 FD。

4.19 设关系模式 R（ABCD），R 上的 FD 集 F=｛ A→C，D→C，BD→A｝，试说明ρ=｛ AB，ACD，BCD ｝相对于 F 是损失分解的理由。

答：据已知的 F 集，不可能把初始表格修改为有一个全 a 行的表格，因此ρ相对于 F 是损失分解。

4.20 设关系模式 R(ABCD)上 FD 集为 F，并且 F=｛A→B，B→C，D→B}。

① R 分解成ρ=｛ACD，BD｝，试求 F 在 ACD 和 BD 上的投影。

② ACD 和 BD 是 BCNF 吗？如不是，试分解成 BCNF。

解：① F 在模式 ACD 上的投影为｛A→C，D→C｝，F 在模式 BD 上的投影为｛D→B}。

② 由于模式 ACD 的关键码是 AD，因此显然模式 ACD 不是 BCNF。模式 ACD 应分解成｛AC，AD｝或｛CD，AD｝。但是这个分解不保持 FD，丢失了 FD D→C 或 A→C。

另外，模式 BD 已是 BCNF。

4.21 设关系模式 R(ABCD)， ρ =｛AB，BC，CD｝是 R 的一个分解。设 F1=｛A→B，B→C｝，F2=｛B→C，C→D｝。

① 如果 F1 是 R 上的 FD 集，此时 ρ 是否无损分解？若不是，试举出反例。

② 如果 F2 是 R 上的 FD 集呢？

解：① 据 chase 过程可知，相对于 F1，R 分解成 ρ 是损失分解。

据构造初始表的规则，这个反例可以是下面的表格：

r	A	B	C	D
	1	1	0	0
	0	1	1	0
	0	0	1	1

对于这个 r 而言，显然 r≠ m_ρ（r）。

② 据 chase 过程可知，相对于 F2，R 分解成 ρ 是无损分解。

4.22 设关系模式 R（ABCD），F 是 R 上成立的 FD 集，F=｛ AB→CD，A→D ｝。

① 试说明 R 不是 2NF 模式的理由。

② 试把 R 分解成 2NF 模式集。

答：①从已知 FD 集 F，可知 R 的候选键是 AB。

另外，AB→D 是一个局部依赖，因此 R 不是 2NF 模式。

②此时 R 应分解成 ρ =｛ AD，ABC ｝， ρ 是 2NF 模式集。

4.23 设关系模式 R（ABC），F 是 R 上成立的 FD 集，F=｛ C→B，B→A ｝。

① 试说明 R 不是 3NF 模式的理由。

② 试把 R 分解成 3NF 模式集。

答：①从已知 FD 集 F，可知 R 的候选键是 C。

从 C→B 和 B→A，可知 C→A 是一个传递依赖，因此 R 不是 3NF 模式。

②此时 R 应分解成 ρ =｛ CB，BA ｝， ρ 是 3NF 模式集。

4.24 设有关系模式 R（职工编号，日期，日营业额，部门名，部门经理），该模式统计商店里每个职工的日营业额，以及职工所在的部门和经理信息。

如果规定：每个职工每天只有一个营业额；每个职工只在一个部门工作；每个部门只有一个经理。

试回答下列问题：

（1）根据上述规定，写出模式 R 的基本 FD 和关键码；

（2）说明 R 不是 2NF 的理由，并把 R 分解成 2NF 模式集；

（3）进而分解成 3NF 模式集。

解：（1）基本的 FD 有三个：

（职工编号，日期）→ 日营业额

职工编号 → 部门名

部门名 → 部门经理

R 的关键码为（职工编号，日期）。

（2）R 中有两个这样的 FD：

（职工编号，日期）→（部门名，部门经理）

职工编号 →（部门名，部门经理）

可见前一个 FD 是局部依赖，所以 R 不是 2NF 模式。

R 应分解成 R1（职工编号，部门名，部门经理）

R2（职工编号，日期，日营业额）

此处，R1 和 R2 都是 2NF 模式。

（3）R2 已是 3NF 模式。

在 R1 中，存在两个 FD：职工编号 → 部门名

部门名 → 部门经理

因此，“职工编号 → 部门经理”是一个传递依赖，R1 不是 3NF 模式。

R1 应分解成 R11（职工编号，部门名）

R12（部门名，部门经理）

这样，ρ = { R11，R12，R2 }是一个 3NF 模式集。

4.25 设有关系模式

R（运动员编号，比赛项目，成绩，比赛类别，比赛主管）

存储运动员比赛成绩及比赛类别、主管等信息。

如果规定：每个运动员每参加一个比赛项目，只有一个成绩；每个比赛项目只属于一个比赛类别；每个比赛类别只有一个比赛主管。

试回答下列问题：

（1）根据上述规定，写出模式 R 的基本 FD 和关键码；

（2）说明 R 不是 2NF 的理由，并把 R 分解成 2NF 模式集；

（3）进而分解成 3NF 模式集。

解：（1）基本的 FD 有三个：

（运动员编号，比赛项目）→ 成绩

比赛项目 → 比赛类别

比赛类别 → 比赛主管

R 的关键码为（运动员编号，比赛项目）。

（2）R 中有两个这样的 FD：

（运动员编号，比赛项目）→（比赛类别，比赛主管）

比赛项目 → （比赛类别，比赛主管）

可见前一个 FD 是局部依赖，所以 R 不是 2NF 模式。

R 应分解成 R1（比赛项目，比赛类别，比赛主管）

R2（运动员编号，比赛项目，成绩）

这里，R1 和 R2 都是 2NF 模式。

（3）R2 已是 3NF 模式。

在 R1 中，存在两个 FD：比赛项目 → 比赛类别

比赛类别 → 比赛主管

因此，“比赛项目 → 比赛主管”是一个传递依赖，R1 不是 3NF 模式。

R1 应分解成 R11（比赛项目，比赛类别）

R12（比赛类别，比赛主管）

这样，ρ = { R11，R12，R2 }是一个 3NF 模式集。

4.26 设关系模式 R（ABCD），在 R 上有五个相应的 FD 集及分解：

（1）F={ B→C，D→A }， ρ ={ BC，AD }

（2）F={ AB→C，C→A，C→D }， ρ ={ ACD，BC }

（3）F={ A→BC，C→AD }， ρ ={ ABC，AD }

（4）F={ A→B，B→C，C→D }， ρ ={ AB，ACD }

（5）F={ A→B，B→C，C→D }， ρ ={ AB，AD，CD }

试对上述五种情况分别回答下列问题：

① 确定 R 的关键码。

② 是否无损分解？

③ 是否保持 FD 集？

④ 确定 ρ 中每一模式的范式级别。

解：

（1） ① R 的关键码为 BD。

② ρ 不是无损分解。

③ ρ 保持 FD 集 F。

④ ρ 中每一模式已达到 BCNF 级别。

（2） ① R 有两个关键码：AB 和 BC。

② ρ 是无损分解。

③ 因为 π_{ACD}（F）={ C→A ，C→D }， π_{BC}（F）= ϕ（没有非平凡的 FD），所以 ρ 不保持 FD，丢失了 AB→C。

④ ρ 中两模式均已达到 BCNF 级别。

（3） ① R 有两个关键码：A 和 C

② ρ 是无损分解。

③ 因为 π_{ABC}（F）={ A→BC，C→A }， π_{AD}（F）= { A→D }，所以 ρ 保持 FD。

④ 在模式 ABC 中，关键码是 A 或 BC，属性全是主属性，但有传递依赖（A→BC，BC→A）。因此模式 ABC 是 3NF，但不是 BCNF。而模式 AD 显然已是 BCNF。

（4） ① R 的关键码为 A。

② ρ 是无损分解。

③ 因为 π_{AB}（F）={ A→B }， π_{ACD}（F）={ A→C，C→D }，从这两个依赖集推不出原来的 B→C，因此 ρ 不保持 FD，丢失了 B→C。

④ 模式 AB 是 BCNF，模式 ACD 不是 3NF，只达到 2NF 级别。

（5） ① R 的关键码为 A。

② ρ 不是无损分解。

③ 因为 π_{AB}（F）={ A→B }， π_{AD}（F）={ A→D }， π_{CD}（F）={ C→D }，从这三个依赖集推不出原来的 B→C，因此 ρ 不保持 FD，丢失了 B→C。

④ ρ 中每个模式均是 BCNF 级别。

4.27 设有关系模式 R（ABC），其关系 r 如图 4.2 所示。试判断下列 FD 和 MVD 在关系 r 中是否成立？

① A→B　② A→→B　③ BC→A　④ BC→→A

⑤ B→C　⑥ B→→C

A	B	C
1	2	3
4	2	3
5	3	3
5	3	4

图 4.2

解：① A→B 在 r 中成立，　② A→→B 在 r 中成立

③ BC→A 在 r 中不成立　④ BC→→A 在 r 中成立

⑤ B→C 在 r 中不成立　⑥ B→→C 在 r 中成立

4.28 设有关系模式 R（ABCDE），现有 R 的七个关系，如图 4.3 所示。试判断 FD BC→D 和 MVD BC→→D 分别在这些关系中是否成立？

A	B	C	D	E
（空关系）				

A	B	C	D	E
a	2	3	4	5
2	a	3	5	5

A	B	C	D	E
a	2	3	4	5
2	a	3	5	5
a	2	3	4	6

（a）关系 r_1

A	B	C	D	E
a	2	3	4	5
2	a	3	4	5
a	2	3	6	5

（a）关系 r_2

A	B	C	D	E
a	2	3	4	5
2	a	3	7	5
a	2	3	4	6

（a）关系 r_3

A	B	C	D	E
a	2	3	4	5
2	a	3	4	5
a	2	3	6	5
a	2	3	6	6

（a）关系 r_4

A	B	C	D	E
a	2	3	4	5
a	2	3	6	5
a	2	3	6	6
a	2	3	4	6

（a）关系 r_5

（a）关系 r_6

（a）关系 r_7

图 4.3

解：BC→D 在 r_1、r_2、r_3、r_5 中成立，在 r_4、r_6、r_7 不成立。

BC→→D 在 r_1、r_2、r_3、r_4、r_5、r_7 中成立，在 r_6 不成立。

4.29 设关系模式 R（ABC）上有一个 MVD A→→B。如果已知 R 的当前关系存在三个元组（ab_1c_1）、（ab_2c_2）和（ab_3c_3），那么这个关系中至少还应该存在哪些元组？

解：这个关系中至少还应存在下面 6 个元组：（ab_1c_2），（ab_2c_1），（ab_1c_3），（ab_3c_1），（ab_2c_3），（ab_3c_2）。

4.30 在教材 P158 的例 4.20 中，模式 R(CSPY)上的依赖集 D={ SP→Y }。试举 r 的例子满足 SP→Y，但 π_{CS}（r）⋈ π_{CP}（r）⋈ π_{SPY}（r）≠r。验证这个 r 不满足 MVD C→→S 和 C→→P。

答：设 r 为下面的关系：

r	C	S	P	Y
	c1	s1	p1	2001
	c1	s2	p1	1999
	c1	s2	p2	2000
	c1	s1	p2	2003

在 r 中，有 SP→Y 成立，但 C→→S 和 C→→P 都不成立。譬如在前两个元组中，交换 S 的值，得到两个元组（c1，s2，p1，2001）和（c1，s1，p1，1999），但在 r 中找不到这两个元组，可见 C→→S 在 R 中不成立。同样，在中间两个元组中，交换 P 的值，得到两个元组（c1，s2，p2，1999）和（c1，s2，p1，2000），但在 r 中也找不到这两个元组，可见 C→→P 在 R 中也不成立。

关系 r 在 CSP 的投影为下面关系：

C	S	P
c1	s1	p1
c1	s2	p1
c1	s2	p2
c1	s1	p2

在上面的关系中，可以验证 C→→S 和 C→→P 都是成立的。因此多值依赖 C→→S 和 C→→P 在模式 R（CSPY）中只能是一个嵌入的多值依赖。

4.31 试举出“若 X→→Y 和 Y→→Z，则 X→→Z”不成立的一个例子。

解：设 R（ABCD），有两个 MVD A→→BC 和 BC→→CD，模式 R 的关系 r 值如下所述，显然 A→→CD 不成立，但 A→→D 是成立的。

R	A	B	C	D
	a	b_1	c_1	d_1
	a	b_2	c_2	d_2
	a	b_1	c_1	d_2
	a	b_2	c_2	d_1

4.32 下面的结论哪些是正确的？哪些是错误的？对于错误，请给出一个反例加以说明。

① 任何一个二元关系模式属于 3NF 模式。

② 任何一个二元关系模式属于 BCNF 模式。

③ 任何一个二元关系模式属于 4NF 模式。

④ 任何一个二元关系模式属于 5NF 模式。

⑤ 若 R（ABC）中有 A→B 和 B→C，则有 A→C。

⑥ 若 R（ABC）中有 A→B 和 A→C，则有 A→BC。

⑦ 若 R（ABC）中有 B→A 和 C→A，则有 BC→A。

⑧ 若 R（ABC）中有 BC→A，则有 B→A 和 C→A。

解：①、②成立。

③ 不成立。有 R(AB)但 $r=r_A \bowtie r_B$(即 $r=r_A \times r_B$)不一定成立。

④ 与③一样，不成立。

⑤、⑥、⑦成立

⑧ 不成立。例如

r	A	B	C
	3	1	2
	4	1	3
	4	2	2

BC→A 成立，但 B→A 和 C→A 都不成立。

4.33 试撰写 2000 字短文，论述泛关系假设、无损联接和保持依赖间的联系。

答：这篇短文的要点如下：

（1）“泛关系假设”是在谈论数据库时必须存在泛关系情况下再讨论分解。

（2）谈论无损分解的先决条件是泛关系假设。

（3）谈论保持 FD 时，不提泛关系假设。

（4）无损分解与保持 FD 之间，没有必然的联系。

（5）满足无损分解的数据库，有 $r=m_\rho(r)$ 性质。

（6）满足保持 FD 的数据库，数据的语义值肯定满足 FD。

4.3 自测题

4.3.1 填空题

1．关系模式的操作异常问题往往是由_____引起的。

2．函数依赖完备的推理规则集包括_____、_____和_____。

3．如果 Y⊆X⊆U，则 X→Y 成立。这条推理规则称为_____。

4．如果 X→Y 和 WY→Z 成立，则 WX→Z 成立。这条推理规则称为_____。

5．如果 X→Y 和 Y⊆X 成立，那么称 X→Y 是一个_____。这种 FD 可以根据推理规则_____律就可推出。

6．“从已知的 FD 集使用推理规则导出的 FD 在 F^+中”，这是推理规则的_____性。

7．“不能从已知的 FD 集使用推理规则导出的 FD 不在 F^+中”，这是推理规则的_____性。
8．函数依赖 X→Y 能从推理规则导出的充分必要条件是_____。
9．被函数依赖集 F 逻辑蕴涵的函数依赖的全体构成的集合，称为_____，用符号_____表示。
10．由属性集 X 函数决定的属性的集合，称为_____，用符号_____表示。
11．在关系模式 R 中，能函数决定所有属性的属性组，称为模式 R 的_____。
12．两个函数依赖集 F 和 G 等价的充分必要条件是_____。
13．关系模式 R 有 n 个属性，则在模式 R 上可能成立的函数依赖有_____个，其中平凡的 FD 有_____个，非平凡的 FD 有_____个。
14．谈论无损联接的先决条件是作了_____的假设。
15．设有关系模式 R（A，B，C，D），F 是 R 上成立的 FD 集，F={AB→C，D→B }，则 F 在模式 ACD 上的投影为_____；F 在模式 AC 上的投影为_____。
16．消除了非主属性对候选键局部依赖的关系模式，称为_____模式。
17．消除了非主属性对候选键传递依赖的关系模式，称为_____模式。
18．消除了每一属性对候选键传递依赖的关系模式，称为_____模式。
19．在关系模式的分解中，数据等价用_____衡量，依赖等价用_____衡量。

4.3.2 单项选择题（在备选的答案中选出一个正确的答案）

1．在关系模式 R 中，函数依赖 X→Y 的语义是 [ ]
A．在 R 的某一关系中，若两个元组的 X 值相等，则 Y 值也相等
B．在 R 的每一关系中，若两个元组的 X 值相等，则 Y 值也相等
C．在 R 的某一关系中，Y 值应与 X 值相等
D．在 R 的每一关系中，Y 值应与 X 值相等

2．如果 X→Y 和 WY→Z 成立，那么 WX→Z 成立。这条规则称为 [ ]
A．增广律 B．传递律 C．伪传递律 D．分解律

3．X→Y 能从推理规则导出的充分必要条件是 [ ]
A．$Y \subseteq X$ B．$Y \subseteq X^+$ C．$X \subseteq Y^+$ D．$X^+=Y^+$

4．两个函数依赖集 F 和 G 等价的充分必要条件是 [ ]
A．F=G B．$F^+=G$ C．$F=G^+$ D．$F^+=G^+$

5．在最小依赖集 F 中，下面叙述不正确的是 [ ]
A．F 中每个 FD 的右部都是单属性
B．F 中每个 FD 的左部都是单属性
C．F 中没有冗余的 FD
D．F 中每个 FD 的左部没有冗余的属性

6．设有关系模式 R（A，B，C，D），F 是 R 上成立的 FD 集，F={ B→A，D→C }，则 F^+中左部为（BC）的函数依赖有 [ ]
A．2 个 B．4 个 C．8 个 D．16 个

7．设有关系模式 R（A，B，C，D），F 是 R 上成立的 FD 集，F={ A→B，B→C，C→D，D→A }，则 F^+中，左部为（CD）的函数依赖有 [ ]
A．2 个 B．4 个 C．8 个 D．16 个

8．设有关系模式 R（A，B，C，D），F 是 R 上成立的 FD 集，F={ AB→C，D→A }，则属性集（CD）的闭包$(CD)^+$为 [ ]
A．CD B．ACD C．BCD D．ABCD

9．设有关系模式 R（A，B，C，D），F 是 R 上成立的 FD 集，F={ AB→C，D→A }，则 R 的关键码为 [ ]
A．AB B．AD C．BC D．BD

10．在关系模式 R 分解成 ρ ={R_1，……，R_k}时，R 上的关系 r 和其投影联接表达式 $m_\rho(r)$之间满足 [ ]

A．r= $m_\rho(r)$　　B．r⊆$m_\rho(r)$　　C．$m_\rho(r)$ ⊆r　　D．r≠$m_\rho(r)$

11．设关系模式 R（A，B，C，D），F 是 R 上成立的 FD 集，F={ B→A，A→C }， ρ ={ AB，AC，AD }是 R 上的一个分解，那么分解 ρ 相对于 F [ ]

A．是无损联接分解，也是保持 FD 的分解

B．是无损联接分解，但不保持 FD 的分解

C．不是无损联接分解，但保持 FD 的分解

D．既不是无损联接分解，也不保持 FD 的分解

12. 设关系模式 R(A,B,C,D),F 是 R 上成立的 FD 集,F={ A→B,B→C,C→D, D→A }， ρ ={ AB，BC，AD }是 R 上的一个分解，那么分解 ρ 相对于 F [ ]

A．是无损联接分解，也是保持 FD 的分解

B．是无损联接分解，但不保持 FD 的分解

C．不是无损联接分解，但保持 FD 的分解

D．既不是无损联接分解，也不保持 FD 的分解

13．设关系模式 R（A，B，C，D），F 是 R 上成立的 FD 集，F={ AB→C，D→B }，那么 F 在模式 ACD 上的投影 π_{ACD}（F）为 [ ]

A．{ AB→C，D→B }　　B．{ AC→D }

C．{ AD→C }　　D．ϕ（即不存在非平凡的 FD）

14．设关系模式 R（A，B，C，D），F 是 R 上成立的 FD 集，F={ AB→C，D→B }，ρ ={ ACD，BD }是 R 上的一个分解，那么分解 ρ [ ]

A．保持函数依赖集 F　　B．丢失了 AB→C

C．丢失了 D→B　　D．是否保持 FD，由 R 的当前关系确定

15．在关系模式 R 分解成数据库模式 ρ 时，谈论无损联接的先决条件是 [ ]

A．数据库模式 ρ 中的关系模式之间有公共属性 B．保持 FD 集

C．关系模式 R 中不存在局部依赖和传递依赖　　D．存在泛关系

16．无损联接和保持 FD 之间的关系是 [ ]

A．同时成立或不成立　　B．前者蕴涵后者

C．后者蕴涵前者　　D．没有必然的联系

4.3.3 简答题

1．为什么要进行关系模式的分解？分解的依据是什么？

2．分解有什么优缺点？

4.4 自测题答案

4.4.1 填空题答案

1．数据冗余　　2．自反律　增广律　传递律

3．自反律　　4．伪传递律

5．平凡的 FD　自反　　6．正确

7．完备　　8．Y⊆X^+

9．函数依赖集 F 的闭包　F^+　　10．属性集 X 的闭包　X^+

11．超键（注：不能回答“候选键”）　　12．F^+= G^+（注：不能回答 F=G）

13．4^n　3^n　4^n-3^n　　14．存在泛关系

15．{AD→C }　ϕ (即没有非平凡的函数依赖)　16．2NF

17．3NF　　18．BCNF

19．无损联接　保持 FD

4.4.2 单项选择题答案

1．B	2．C	3．B	4．D	5．B	6．C
7．D	8．B	9．D	10．B	11．C	12．A
13．C	14．B	15．D	16．D		

4.4.3 简答题答案

1．答：由于数据之间存在着联系和约束，在关系模式的关系中可能会存在数据冗余和操作异常现象，因此需把关系模式进行分解，以消除冗余和异常现象。

分解的依据是数据依赖和模式的标准（范式）。

2．答：分解有两个优点：① 消除冗余和异常；② 在分解了的关系中可存储悬挂元组。

但分解有两个缺点：① 可能分解了的关系不存在泛关系；② 做查询操作，需做联接操作，增加了查询时间。

第 5 章 数据库设计与 ER 模型

5.1 基本内容分析

5. 1. 1 本章重要概念

（1）DBS 生存期及其 7 个阶段的任务和工作，DBD 过程的输入和输出。

（2）概念设计的重要性、主要步骤。逻辑设计阶段的主要步骤。

（3）ER 模型的基本元素，属性的分类，联系的元数、连通词、基数。采用 ER 方法的概念设计步骤。

（4）ER 模型到关系模型的转换规则。采用 ER 方法的逻辑设计步骤。

（5）ER 模型的扩充：弱实体，超类和子类。

5.1.2 本章的重点篇幅

（1）教材中 P193-194 的转换规则和实例。

（2）教材中 P196-200 的四个 ER 模型实例。

5.1.3 对 ER 模型的理解

ER 模型是人们认识客观世界的一种方法、工具。ER 模型具有客观性和主观性两重含义。ER 模型是在客观事物或系统的基础上形成的，在某种程度上反映了客观现实，反映了用户的需求，因此 ER 模型具有客观性。但 ER 模型又不等同于客观事物的本身，它往往反映事物的某一方面，至于选取哪个方面或哪些属性，如何表达则决定于观察者本身的目的与状态，从这个意义上说，ER 模型又具有主观性。

ER 模型的设计过程，基本上是两大步：

• 先设计实体类型（此时不要涉及到“联系”）；

• 再设计联系类型（考虑实体间的联系）。

具体设计时，有时“实体”与“联系”两者之间的界线是模糊的。数据库设计者的任务就是要把现实世界中的数据以及数据间的联系抽象出来，用“实体”与“联系”来表示。

另外，设计者应注意，ER 模型应该充分反映用户需求，ER 模型要得到用户的认可才能确定下来。

5.2 教材中习题 5 的解答

5. 1 名词解释

（1）• 软件工程：研究如何用科学知识、工程方面的纪律指导软件开发的过程，以提高软件质量和开发效率，降低开发成本，这样的一门学科称为“软件工程”。

• 软件生存期：软件生存期是指从软件的规划、研制、实现、投入运行后的维护，直到它被新的软件所取代而停止使用的整个期间。软件生存期通常分为六个阶段：规划阶段，需求分析阶段，设计阶段，程序编制阶段，调试阶段，运行维护阶段。

• 数据库工程：数据库应用系统的开发是一项软件工程，但又有自己特有的特点，所以特称为“数据库工程”。

• 数据库系统生存期：我们把数据库应用系统从开始规划、设计、实现、维护到最后被新的系统取代而停止使用的整个期间，称为数据库系统生存期。这个生存期一般可划分成下面七个阶段：规划，需求分析，概念设计，逻辑设计，物理设计，实现，运行和维护

（2）• 实体：可以区别的客观存在的事物，称为实体。

• 实体集：同一类实体构成的集合，称为实体集。

• 实体类型：实体集中实体的定义，称为实体类型。

• 实体标识符：能惟一标识实体的属性或属性集，称为实体标识符。有时也称为关键码（key），或简称为键。

（3）• 联系：一个或多个实体之间的关联关系，称为联系。

• 联系集：同一类联系构成的集合，称为联系集。

• 联系类型：联系集中联系的定义，称为联系类型。

（4）• 属性：实体的某一特性，称为属性。

• 基本属性：不可再分割的属性，称为基本属性。
• 复合属性：可再分解成其他属性的属性，称为复合属性。
• 单值属性：同一实体的属性只能取一个值，称为单值属性。
• 多值属性：同一实体的属性可能取多个值，称为多值属性。
• 导出属性：通过具有相互依赖的属性推导而产生的属性，称为导出属性。

（5）• 联系：

• 联系的元数：一个联系涉及到的实体集个数，称为该联系的元数。

• 联系的连通词：联系涉及到的实体集之间实体对应的方式（指对应一个还是多个实体），称为联系的连通词。

• 实体的基数：是对连通词更为精确的描述。譬如有两个实体集 E_1 和 E_2，E_1 中每个实体与 E_2 中有联系实体数目的最小值 Min 和最大值 Max，称为 E_1 的基数。

（6）• 弱实体：一个实体对于另一些实体（父实体）具有很强的依赖联系，而且该实体主键的部分或全部从其父实体中获得，则称该实体为弱实体。

• 子类实体和超类实体：某个实体类型中所有实体同时也是另一个实体类型中的实体，此时称前一实体类型是后一实体类型的子类，后一实体类型称为超类。其实体分别称为子类实体和超类实体。

• 继承性：指子类继承其超类上定义的所有属性，但其本身还可以包含其他的属性。

5.2 数据库系统的生存期分成哪几个阶段？数据库结构的设计在生存期中的地位如何？

答：对 DBS 生存期的划分，一般分为七个阶段，即规划、需求分析、概念设计、逻辑设计、物理设计、实现和运行维护。

DB 结构设计的任务就是把概念设计阶段设计好的基本 ER 图转换成与选用的具体机器上的 DBMS 所支持的数据模型相符合的逻辑结构。

5.3 基于数据库系统生存期的数据库设计分成哪几个阶段？

答：基于 DBS 生存期的 DBD 分成以下五个阶段：

规划；需求描述和分析；概念设计；逻辑设计；物理设计。

5.4 数据库设计的规划阶段应做哪些事情？

答：DBD 中规划阶段的主要任务是进行建立 DB 的必要性及可行性分析，确定 DBS 在组织中和信息系统中的地位，以及各个 DB 之间的联系。

5.5 数据库设计的需求分析阶段是如何实现的？目标是什么？

答：需求分析阶段的工作由下面四步组成：

• 分析用户活动，产生用户活动图；
• 确定系统范围，产生系统范围图；
• 分析用户活动所涉及的数据，产生数据流图；
• 分析系统数据，产生数据字典。

需求分析阶段的目标是对系统的整个应用情况作全面的、详细的调查，确定企业组织的目标，收集支持系统总的设计目标的基础数据和对这些数据的要求，确定用户的需求；并把这些要求写成用户和数据库设计者都能接受的文档。

5.6 概念设计的具体步骤是什么？

答：概念设计的主要步走可分为三步：

（1） 进行数据抽象，设计局部概念模式；
（2） 将局部概念模式综合成全局概念模式；
（3） 评审。

5.7 逻辑设计的目的是什么？试述逻辑设计阶段的主要步骤及内容。

答：逻辑设计的目的是把概念设计阶段设计好的基本 ER 图转换成与选用的具体机器上的 DBMS 所支持的数据模型相符合的逻辑结构（包括数据库模式和外模式）。这些模式在功能、性能、完整性和一致性约束及数据库的可扩充性等方面均应满足用户的各种要求。

逻辑设计阶段主要有五步：形成初始模式，设计子模式，设计应用程序梗概，评价模式和修改模式。（解释略）

5.8 什么是数据库结构的物理设计？试述其具体步骤。

答：对于给定的基本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 DB 的物理设计。

物理设计有五步：

确定 DB 的存储记录结构；确定数据存储按排；存取方法的设计；完整性和安全性的设计；应用程序设计。

5.9 数据库实现阶段主要做哪几件事情？

答：数据库实现阶段主要有以下三项工作：

建立实际 DB 结构；装入试验数据调试应用程序；装入实际数据进入试运行状态。

5.10 数据库系统投入运行后，有哪些维护工作？

答：DBS 投入运行以后，就进入运行维护阶段。其主要工作有四项：

维护 DB 的安全性与完整性及系统的转储和恢复；

DB 性能的监督、分析与改进；

增加 DB 新功能；

改正运行中发现的系统错误。

5.11 设某商业集团数据库中有三个实体集。一是“商店”实体集，属性有商店编号、商店名、地址等；二是“商品”实体集，属性有商品号、商品名、规格、单价等；三是“职工”实体集，属性有职工编号、姓名、性别、业绩等。

商店与商品间存在“销售”联系，每个商店可销售多种商品，每种商品也可放在多个商店销售，每个商店销售一种商品，有月销售量；商店与职工间存在着“聘用”联系，每个商店有许多职工，每个职工只能在一个商店工作，商店聘用职工有聘期和月薪。

（1） 试画出 ER 图，并在图上注明属性、联系的类型。

（2） 将 ER 图转换成关系模型，并注明主键和外键。

解：(1) ER 图如图 5.1 所示。

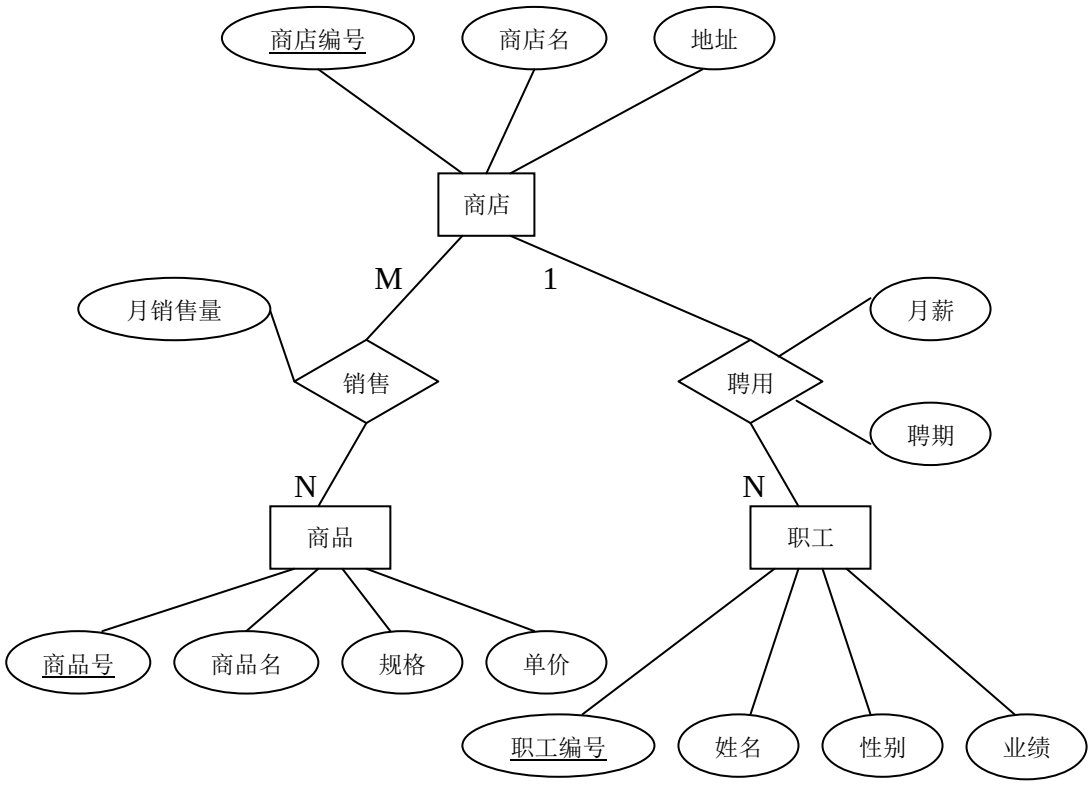


图 5.1

（2）这个 ER 图可转换 4 个关系模式：

商店（商店编号，商店名，地址）

职工（职工编号，姓名，性别，业绩，商店编号，聘期，月薪）

商品（商品号，商品名，规格，单价）

销售（商店编号，商品号，月销售量）

5.12 设某商业集团数据库中有三个实体集。一是“公司”实体集，属性有公司编号、公司名、地址等；二是“仓库”实体集，属性有仓库编号、仓库名、地址等；三是“职工”实体集，属性有职工编号、姓名、性别等。

公司与仓库间存在“隶属”联系，每个公司管辖若干仓库，每个仓库只能属于一个公司管辖；仓库与职工间存在“聘用”联系，每个仓库可聘用多个职工，每个职工只能在一个仓库工作，仓库聘用职工有聘期和工资。

i. 试画出 ER 图，并在图上注明属性、联系的类型。

ii. 将 ER 图转换成关系模型，并注明主键和外键。

解：(1) ER 图如图 5.2 所示。

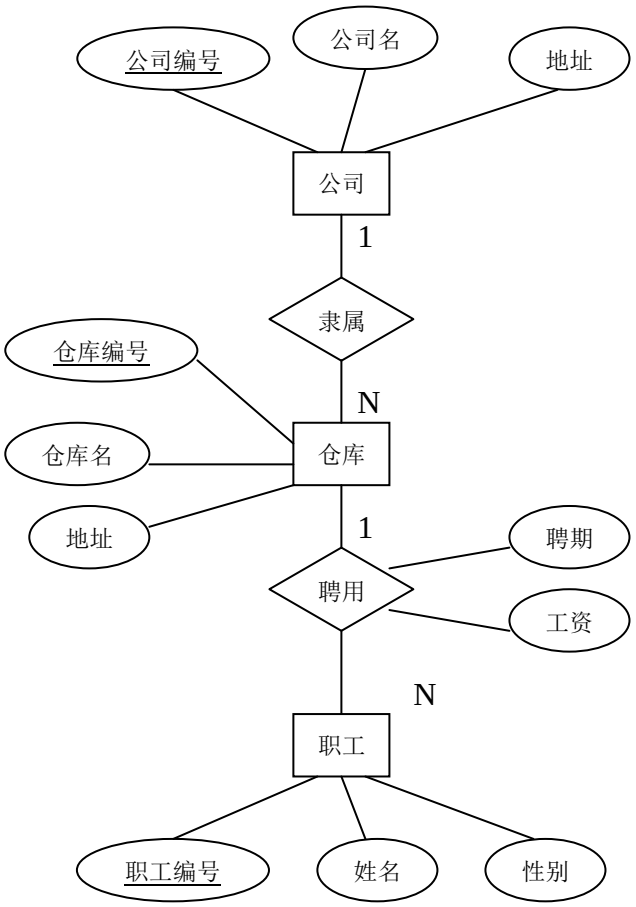


图 5.2

（2）这个 ER 图可转换 3 个关系模式：

公司（公司编号，公司名，地址）
仓库（仓库编号，仓库名，地址，公司编号）
职工（职工编号，姓名，性别，仓库编号，聘期，工资）

5.13 设某商业集团数据库有三个实体集。一是“商品”实体集，属性有商品号、商品名、规格、单价等；二是“商店”实体集，属性有商店号、商店名、地址等；三是“供应商”实体集，属性有供应商编号、供应商名、地址等。

供应商与商品之间存在“供应”联系，每个供应商可供应多种商品，每种商品可向多个供应商订购，每个供应商供应每种商品有个月供应量；商店与商品间存在“销售”联系，每个商店可销售多种商品，每种商品可在多个商店销售，每个商店销售每种商品有个月计划数。试画出反映上述问题的 ER 图，并将其转换成关系模型。

解：ER 图如图 5.3 所示。

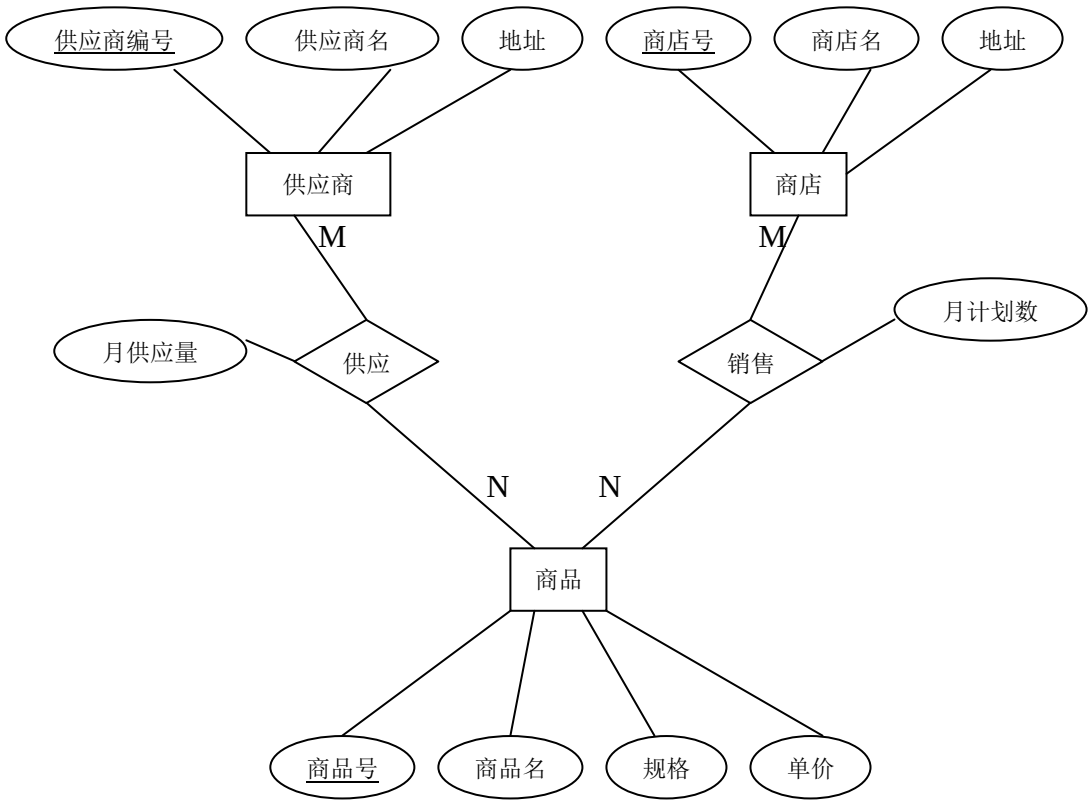


图 5.3

（2）这个 ER 图可转换 5 个关系模式：
供应商（供应商编号，供应商名，地址）
商店（商店号，商店名，地址）
商品（商品号，商品名，规格，单价）
供应（供应商编号，商品号，月供应量）
销售（商店号，商品号，月计划数）

5.14 假设要为银行的储蓄业务设计一个数据库，其中涉及到储户、存款、取款等信息。试设计 ER 模型。

解：储蓄业务主要是存款、取款业务，可设计如图 5.4 所示的 ER 图。

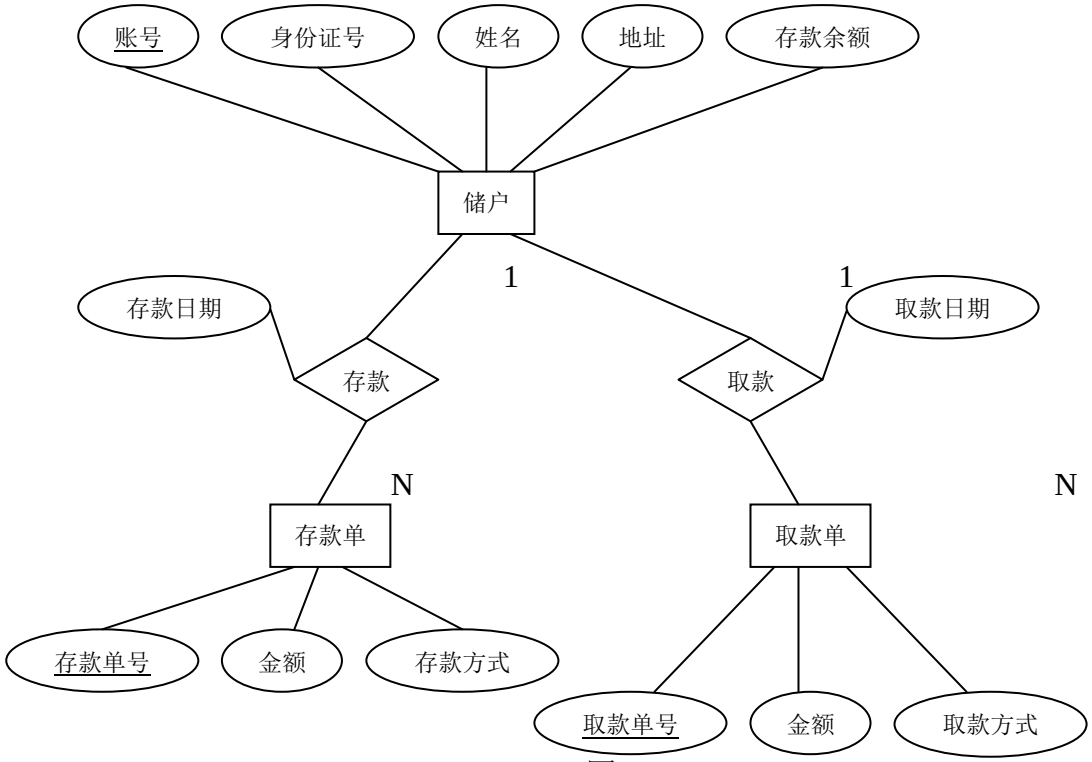


图 5.4

5.15 某体育运动锦标赛有来自世界各国运动员组成的体育代表团参赛各类比赛项目。试为该锦标赛各个代表团、运动员、比赛项目、比赛情况设计一个 ER 模型。

解：图 5.5 是 ER 图的一种设计方案。

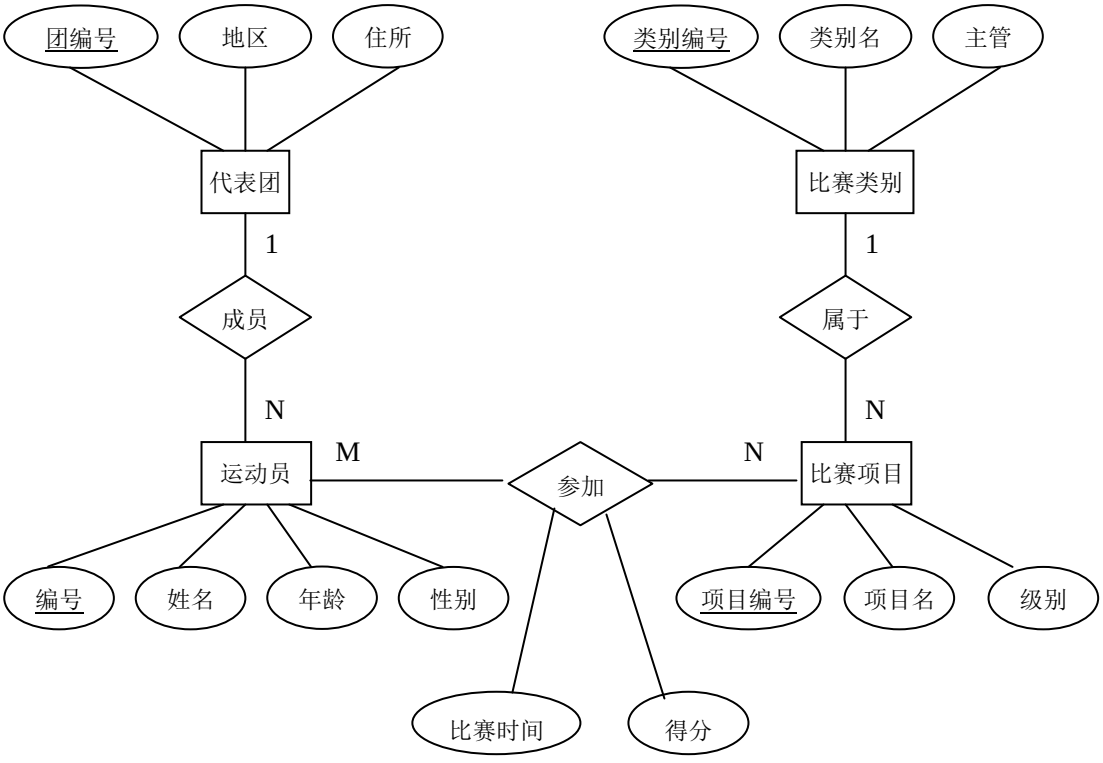


图 5.5

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:

⑴该超市公司有若干仓库，若干连锁商店，供应若干商品。

⑵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。

⑶每个商店销售多种商品，每种商品可在不同的商店销售。

⑷每个商品编号只有一个商品名称，但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。

⑸超市公司的业务员负责商品的进货业务。

试按上述规则设计 ER 模型

解：图 5.6 是 ER 图的一种设计方案。

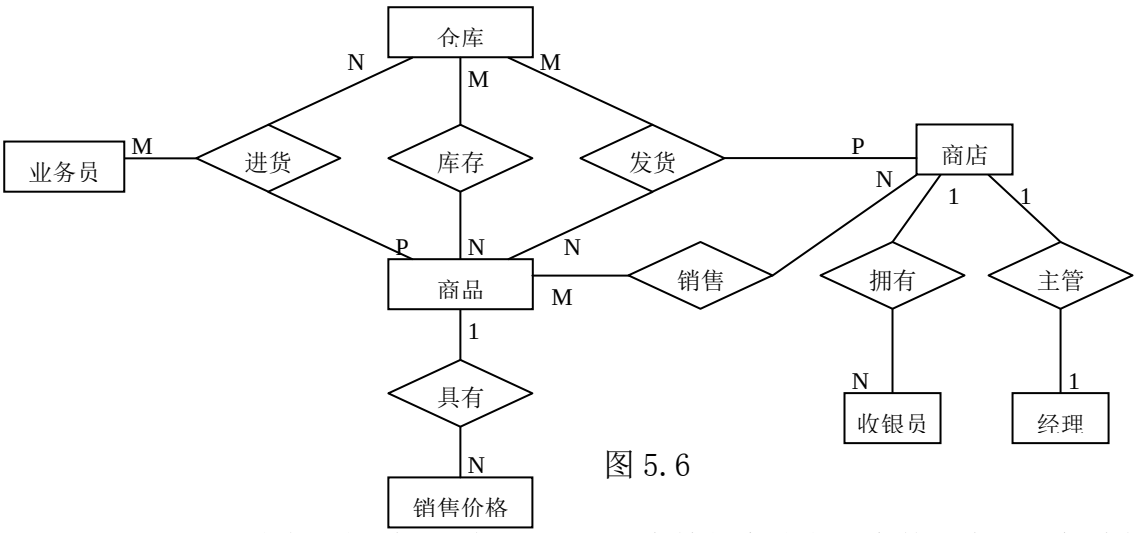


图 5.6

5.17 假设要根据某大学的系、学生、班级、学会等信息建立一个数据库，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。每个学生可以参加多个学会，每个学会有若干学生，学生参加某学会有个入会年份。试为该大学的系、学生、班级、学会等信息设计一个 ER 模型。

解：图 5.7 是 ER 图的一种设计方案。

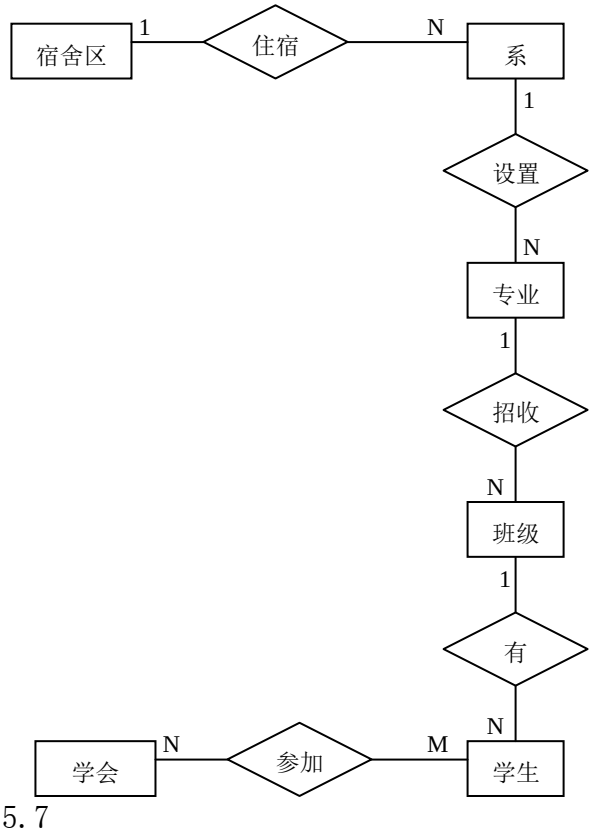


图 5.7

5.18 试把教材中 5.5.2、5.5.3、5.5.4 等三小节中的 ER 模型转换成关系模型，并指出每个关系模式的主键和外键。

（1）（教材中 P197 的 5.5.2 节）公司车队信息系统的 ER 模型

本例为某货运公司设计了车队信息管理系统，对车辆、司机、维修、保险、报销等信息和业务活动进行管理。其 ER 图如图 5.8 所示。

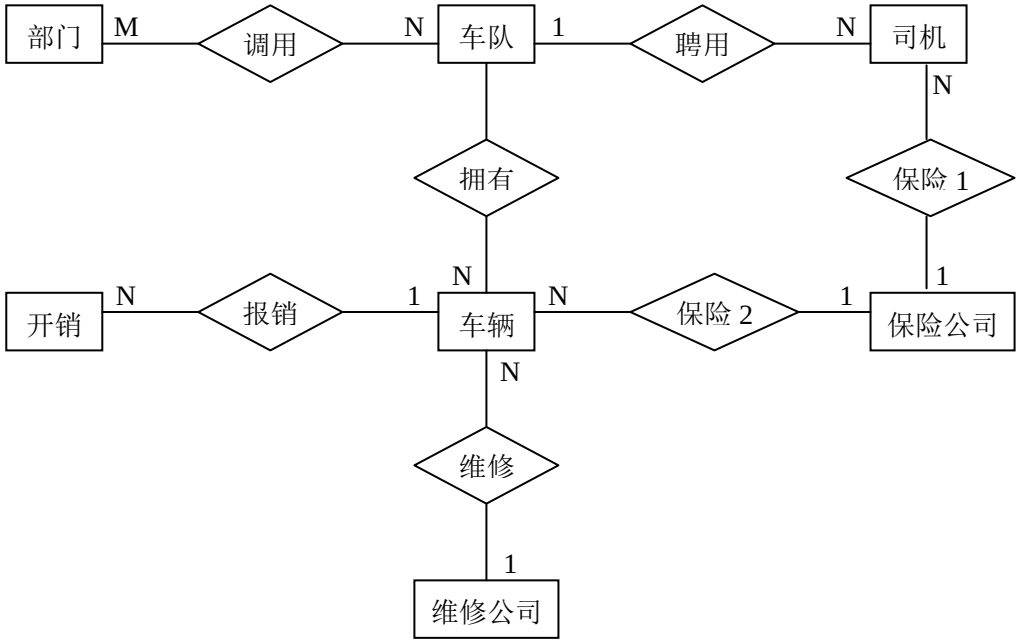


图 5.8　公司车队信息系统的 ER 模型

该 ER 图有 7 个实体类型，其结构如下：

部门（部门号，名称，负责人）

车队（车队号，名称，地址）

司机（司机号，姓名，执照号，电话，工资）

车辆（车牌号，车型，颜色，载重）

保险公司（保险公司号，名称，地址）

维修公司（维修公司号，名称，地址）

开销（顺序号，费用类型，费用，日期，经手人）

实体之间有 7 个联系，其中 6 个是 1:N 联系，1 个是 M:N 联系。其中联系的属性如下：

调用（出车编号，出车日期，车程，费用，车辆数目）

保险 1（投保日期，保险种类，费用）

保险 2（投保日期，保险种类，费用）

进而，读者可以很容易地转换成关系模式集。

解：根据 ER 图和转换规则，7 个实体类型转换成 7 个关系模式，1 个 M:N 联系转换成 1 个关系模式，共 8 个关系模式，如下：

部门（部门号，名称，负责人）

车队（车队号，名称，地址）

司机（司机号，姓名，执照号，电话，工资，车队号，保险公司号，投保日期，保险种类，费用）

车辆（车牌号，车型，颜色，载重，车队号，保险公司号，投保日期，保险种类，费用，维修公司号）

保险公司（保险公司号，名称，地址）

维修公司（维修公司号，名称，地址）

开销（顺序号，车牌号，费用类型，费用，日期，经手人）

调用（出车编号，车队号，部门号，出车日期，车程，费用，车辆数目）

（2）（教材中 P198 的 5.5.3 节）人事管理信息系统的 ER 模型

上海交通电器有限公司设计了人事管理信息系统，其中涉及到职工、部门、岗位、技能、培训课程、奖惩记录等信息。其 ER 图如图 5.9 所示。

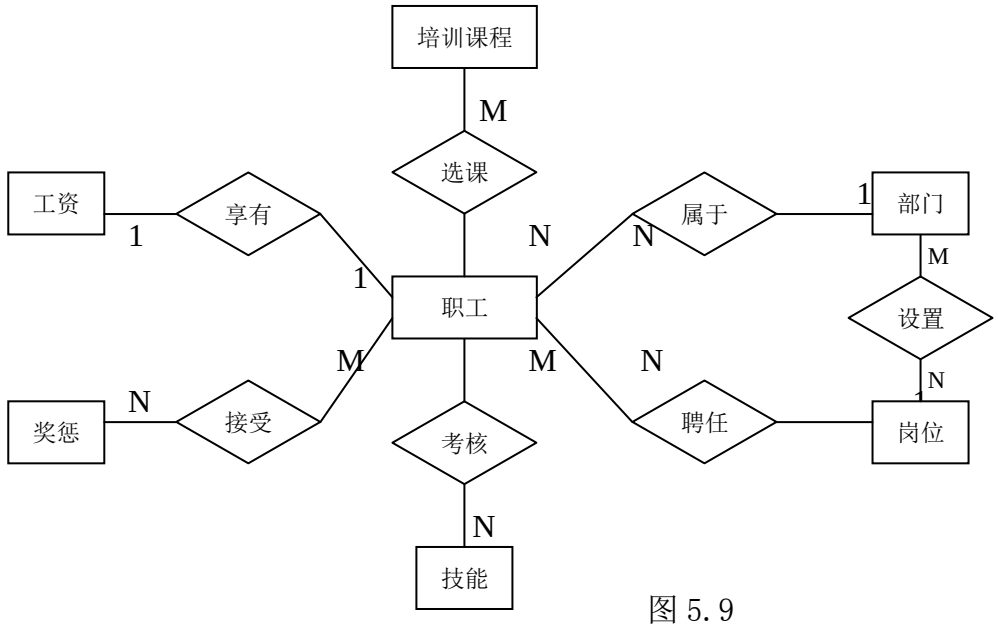


图 5.9

这个 ER 图有 7 个实体类型，其属性如下：

职工（工号，姓名，性别，年龄，学历）

部门（部门号，部门名称，职能）

岗位（岗位编号，岗位名称，岗位等级）

技能（技能编号，技能名称，技能等级）

奖惩（序号，奖惩标志，项目，奖惩金额）

培训课程（课程号，课程名，教材，学时）

工资（工号，基本工资，级别工资，养老金，失业金，公积金，纳税）

这个 ER 图有 7 个联系类型，其中 1 个 1:1 联系，2 个 1:N 联系，4 个 M:N 联系。联系类型的属性如下：

选课（时间，成绩）

设置（人数）

考核（时间，地点，级别）

接受（奖惩时间）

解：根据 ER 图和转换规则，7 个实体类型转换成 7 个关系模式，4 个 M:N 联系转换成 4 个关系模式，共 11 个模式，如下：

职工（工号，姓名，性别，年龄，学历，部门号，岗位编号）

部门（部门号，部门名称，职能）

岗位（岗位编号，岗位名称，岗位等级）

技能（技能编号，技能名称，技能等级）

奖惩（序号，奖惩标志，项目，奖惩金额）

培训课程（课程号，课程名，教材，学时）

工资（工号，基本工资，级别工资，养老金，失业金，公积金，纳税）

选课（工号，课程号，时间，成绩）

设置（部门号，岗位编号，人数）

考核（工号，技能编号，时间，地点）

接受（工号，序号，奖惩日期）

（3）（教材中 P199 的 5.5.4 节）旅游管理信息系统的 ER 模型

上海普教旅行社设计了一个小型的国内旅游管理信息系统，其中涉及到与业务有关的信息有旅游线路、班次、团体、旅客、保险员、导游、宾馆、交通工具等。其 ER 图如图 5.10 所示。

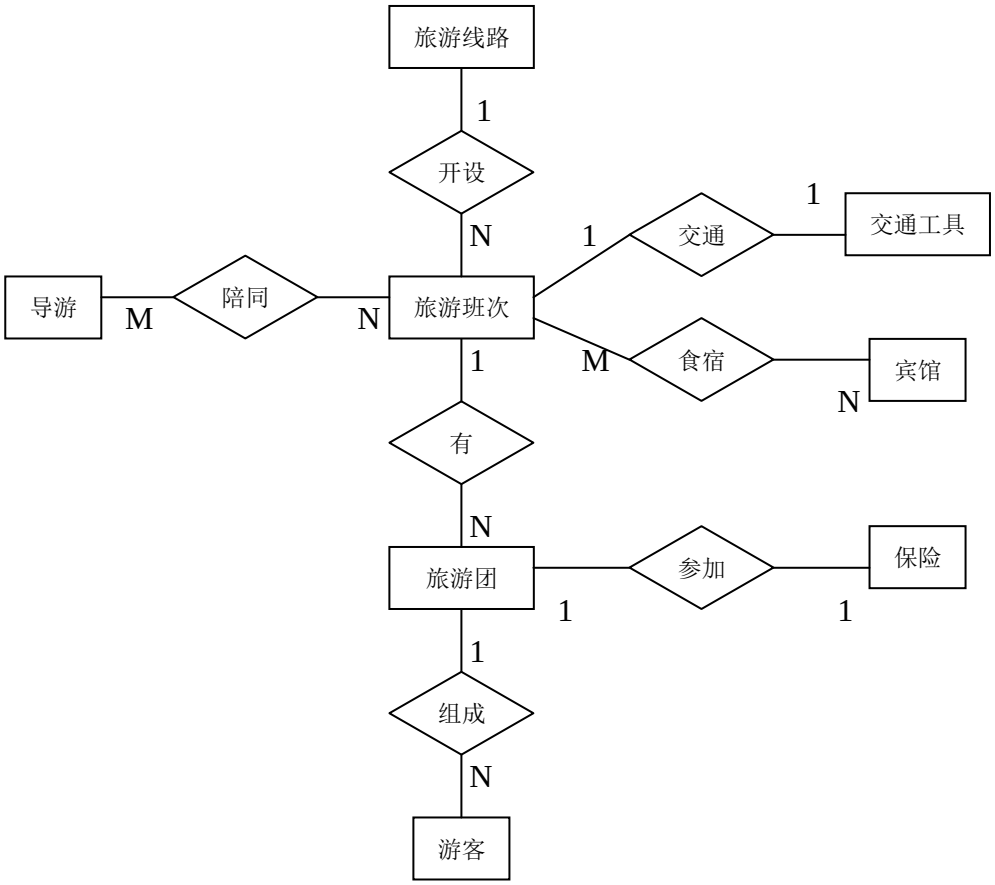


图 5.10

这个 ER 图有 8 个实体类型，其属性如下：

旅游线路（路线号，起点，终点，天数，主要景点）

旅游班次（班次号，出发日期，回程日期，旅游标准，报价）

旅游团（团号，团名，人数，联系人，地址，电话）

游客（游客编号，姓名，性别，年龄，身份证号码，住址，电话）

导游（导游编号，姓名，性别，年龄，身份证号码，住址，电话，语种，等级，业绩）

交通工具（旅游班次号，出发工具，出发日期，出发班次，出发时间，回程工具，回程日期，回程班次，回程时间）

宾馆（宾馆编号，宾馆名，城市，星级，标准房价，联系人，职务，地址，电话，传真）

保险单（保险单编号，保险费，投保日期）

这个 ER 图有 7 个联系类型，其中 2 个 1:1 联系，3 个 1:N 联系，2 个 M:N 联系。

解：根据 ER 图和转换规则，8 个实体类型转换成 8 个关系模式，2 个 M:N 联系转换成 2 个关系模式，共 10 个关系模式，如下：

旅游线路（路线号，起点，终点，天数，主要景点）

旅游班次（班次号，路线号，出发日期，回程日期，旅游标准，报价）

旅游团（团号，旅游班次号，团名，人数，联系人，地址，电话）

游客（游客编号，团号，姓名，性别，年龄，身份证号码，住址，电话）

导游（导游编号，姓名，性别，年龄，身份证号码，住址，电话，语种，等级，业绩）

交通工具（旅游班次号，出发工具，出发日期，出发班次，出发时间，回程工具，回程日期，回程班次，回程时间）

宾馆（宾馆编号，宾馆名，城市，星级，标准房价，联系人，职务，地址，电话，传真）

保险（保险单编号，团号，人数，保险费，投保日期）

陪同（旅游班次号，导游编号）

食宿（旅游班次号，宾馆编号）

5.3 自测题

5.3.1 填空题

1．数据库设计过程的输入包括四部分内容：_____，_____，_____和_____。

2．数据库设计过程的输出主要有两部分：_____和_____。

3．规划阶段具体可以分成三个步骤：_____、_____和_____。

4．需求分析的工作主要有下面四步组成：分析用户活动，产生_____；确定系统范围，产生_____；分析用户活动涉及的数据，产生_____；分析系统数据，产生_____。

5．需求分析中的数据字典通常包含以下五个部分：_____，_____，_____，_____和_____。

6．概念设计的目标是产生反映_____的数据库概念结构，即概念模式。

7．概念设计阶段可分为三步来完成：_____，_____和_____。

8．就方法的特点而言，需求分析阶段通常采用_____的分析方法；概念设计阶段通常采用_____的设计方法。

9．逻辑设计的主要工作是：_____。

10．逻辑设计的步骤有五步：_____，_____，_____，_____和_____。

11．物理设计可分成五步进行：_____，_____，_____，_____和_____。

12．DBS 的维护工作由_____承担的。

13．DBS 的维护工作主要包括以下四个部分：_____，_____，_____，_____。

5.3.2 单项选择题（在备选的答案中选出一个正确答案）

1．需求分析阶段设计数据流程图（DFD）通常采用 [ ]

A．面向对象的方法　　B．回溯的方法

C．自底向上的方法　　D．自顶向下的方法

2．概念设计阶段设计概念模型通常采用 [ ]

A．面向对象的方法　　B．回溯的方法

C．自底向上的方法　　D．自顶向下的方法

3．设计子模式属于数据库设计的 [ ]

A．需求分析　　B．概念设计　　C．逻辑设计　　D．物理设计

4．概念结构设计的主要目标是产生数据库的概念结构，该结构主要反映 [ ]

A．应用程序员的编程需求　　B．DBA 的管理信息需求

C．数据库系统的维护需求　　D．企业组织的信息需求

5．数据库设计人员和用户之间沟通信息的桥梁是 [ ]

A．程序流程图　　B．实体联系图　　C．模块结构图　　D．数据结构图

6．有两个不同的实体集，它们之间存在着一个 1:1 联系和一个 M:N 联系，那么根据 ER 模型

转换成关系模型的规则，这个 ER 结构转换成的关系模式个数为 [ ]

A．2 个 B．3 个 C．4 个 D．5 个

7．如果有 10 个不同的实体集，它们之间存在着 12 个不同的二元联系（二元联系是指两个实体集之间的联系），其中 3 个 1:1 联系，4 个 1:N 联系，5 个 M:N 联系，那么根据 ER 模型转换成关系模型的规则，这个 ER 结构转换成的关系模式个数为 [ ]

A．14 个 B．15 个 C．19 个 D．22 个

8．在 ER 模型转换成关系模型的过程中，下列叙述不正确的是 [ ]

A．每个实体类型转换成一个关系模式

B．每个联系类型转换成一个关系模式

C．每个 M:N 联系类型转换一个关系模式

D．在处理 1:1 和 1:N 联系类型时，不生成新的关系模式

9．当同一个实体集内部的实体之间存在着一个 1:N 联系时，那么根据 ER 模型转换成关系模型的规则，这个 ER 结构转换成的关系模式个数为 [ ]

A．1 个 B．2 个 C．3 个 D．4 个

10．当同一个实体集内部的实体之间存在着一个 M:N 联系时，那么根据 ER 模型转换成关系模型的规则，这个 ER 结构转换成的关系模式个数为 [ ]

A．1 个 B．2 个 C．3 个 D．4 个

11．在数据库设计中，子类与超类存在着 [ ]

A．相容性联系 B．调用的联系

C．继承性的联系 D．一致性联系

5.3.3 设计题

假设要为某商业集团设计一个数据库，该集团中有若干仓库、若干商店、经销若干商品。试画一个有关仓库、商店、商品、采购员、职工、顾客、供应商、采购、入库、出库、销售聘用等信息的 ER 图。

5.3.4 ER 图实例

在数据库设计中，ER 模型的设计是一个很重要的环节。为了帮助学习者提高数据库设计水平，有利于毕业设计和今后的工作，我们从毕业生的论文中挑选了 5 个 ER 模型，供参考。这些设计并不是惟一的，可能还不完善，但大家从中可得到有益的启发，拓宽思路。

1．某学员为医院“住院管理信息系统”设计了数据库的 ER 模型，对医生、护士、病人、病房、诊断、手术、结账等有关信息进行管理，其 ER 图如图 5.11 所示。

这个 ER 图有 8 个实体类型，其属性如下：

病人（住院号，姓名，性别，地址）

医生（医生工号，姓名，职称）

护士（护士工号，姓名，职称）

病床（病床编号，床位号，类型，空床标志）

手术室（手术室编号，类型）

手术（手术标识号，类型，日期，时间，费用）

诊断书（诊断书编号，科别，诊断）

收据（收据编号，项目，金额，收款员，日期）

这个 ER 图有 11 个联系类型，其中 1 个是 1：1 联系，8 个 1：N 联系，2 个是 M：N 联系。联系的属性如下：

协助（角色）

处方（处方单号，序号，药品名称，规格，数量，费用）

入住（入院日期，出院日期）

试把这个 ER 图转换成关系模型。并指出各个关系模式的主键和外键。

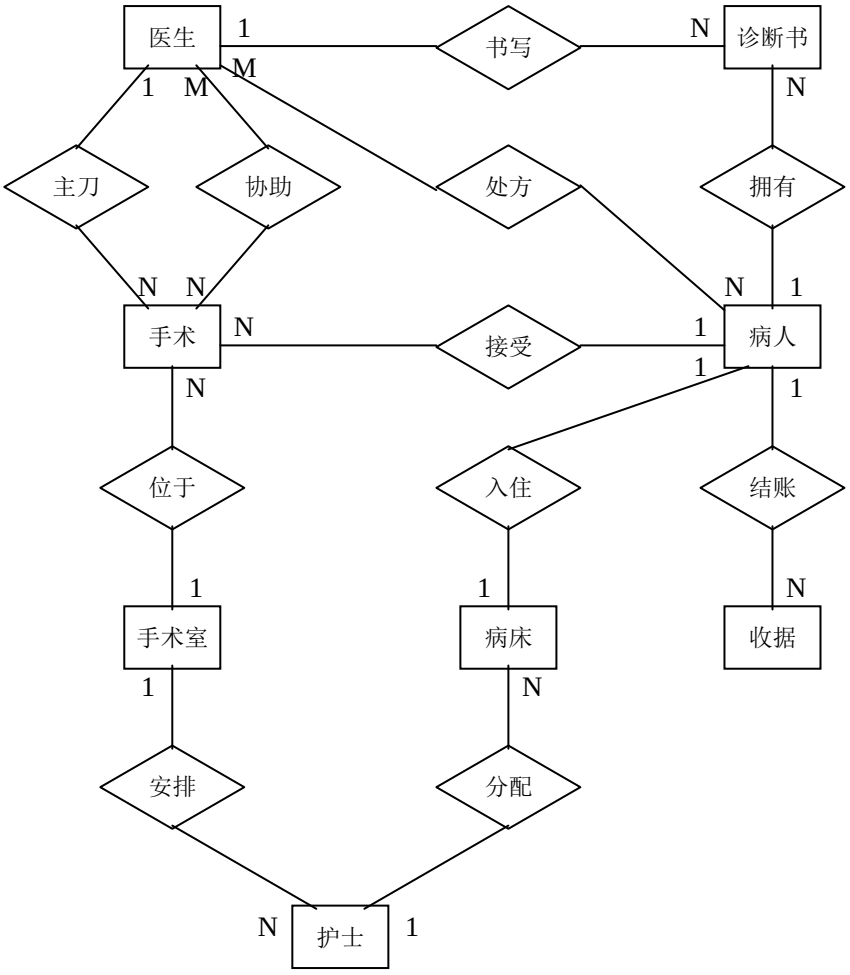


图 5.11 住院管理信息系统的 ER 图

2．某学员为电脑专卖店设计开发了“电脑销售信息管理系统”，数据库的 ER 模型对商品、供应商、仓库、营业员、门店的有关信息进行了管理，其 ER 图如图 5.12 所示。

这个 ER 图有 7 个实体类型，其属性如下：

商品（商品编号，名称，类别，单位，单价）

供应商（供应商编号，名称，账号，地址）

仓库（仓库编号，地址，负责人）

门店（门店编号，名称，地址）

采购员（采购员编号，姓名，业绩）

管理员（管理员编号，姓名，业绩）

营业员（营业员编号，姓名，业绩）

这个 ER 图有 7 个联系类型，其中 2 个是 1：N 联系，1 个 M：N 联系，4 个是 M：N：P 联系。联系的属性如下：

采购（采购单号，数量，日期）

进货（进货单号，数量，日期）

配送（配送单号，数量，日期）

销售（销售单号，数量，日期）

存储（库存量，日期，安全库存量）

试把这个 ER 图转换成关系模型。并指出各个关系模式的主键和外键。

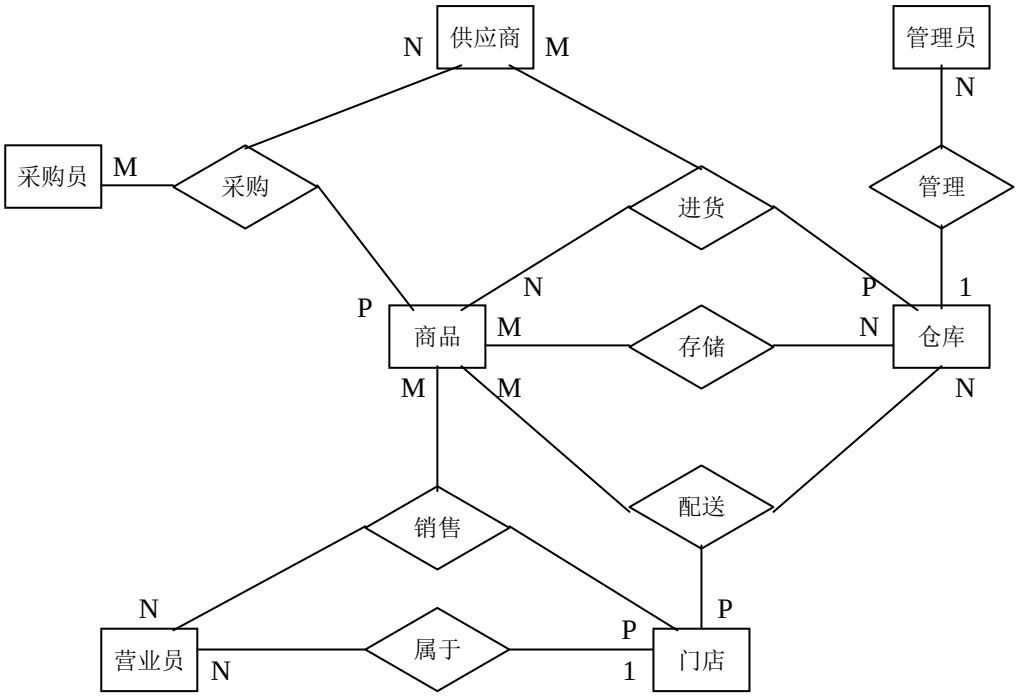


图 5.12 电脑销售信息管理系统的 ER 图

3. 某学员为证券营业网点设计的业务信息管理系统，对客户、资金、证券和业务活动进行了管理，其 ER 图如图 5.13 所示。

该 ER 图有 5 个实体类型，其结构如下：

客户（股东账号，身份证号，姓名，地址，客户类别，开户日期）

资金（资金账号，金额，可取余额，冻结金额，解冻金额，利息，日期）

证券（证券代码，名称，每手股数）

委托（委托序号，数量，买卖类别，价格，时间，操作员）

成交（成交序号，数量，买卖类别，成交价格，时间）

该 ER 图有 8 个联系类型，其中 6 个 1:N 联系，2 个 M:N 联系。其中，联系的属性如下：

持有（金额，可用数量，冻结数量，解冻数量，日期）

存取（存取单序号，存取标志，金额，日期）

试把这个 ER 图转换成关系模式集，并指出每个模式的主键和外键。

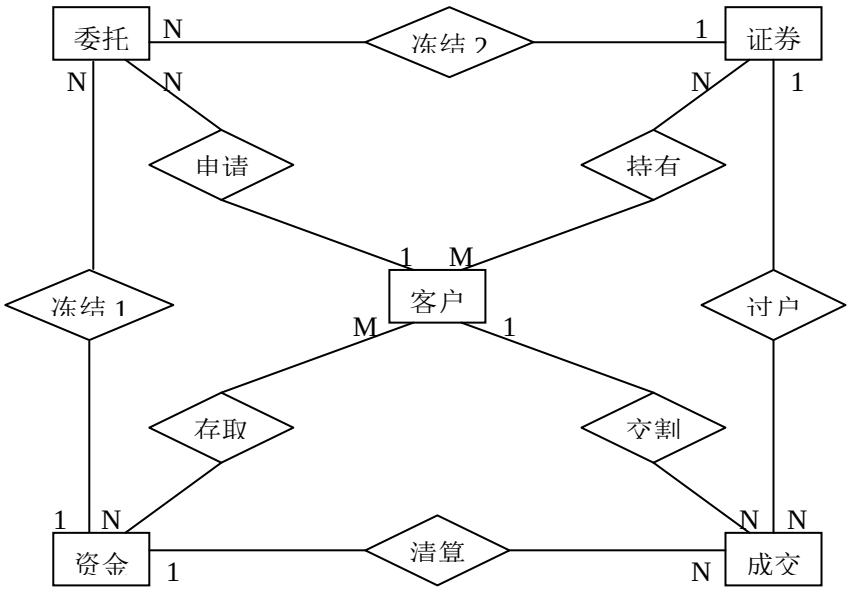


图 5.13 证券业务管理系统的 ER 图

4．某学员为某出版社设计了图书发行信息管理系统，数据涉及到图书、作者、开印、入库、客户和发行员等信息。得到的全局 ER 图如图 5.14 所示。

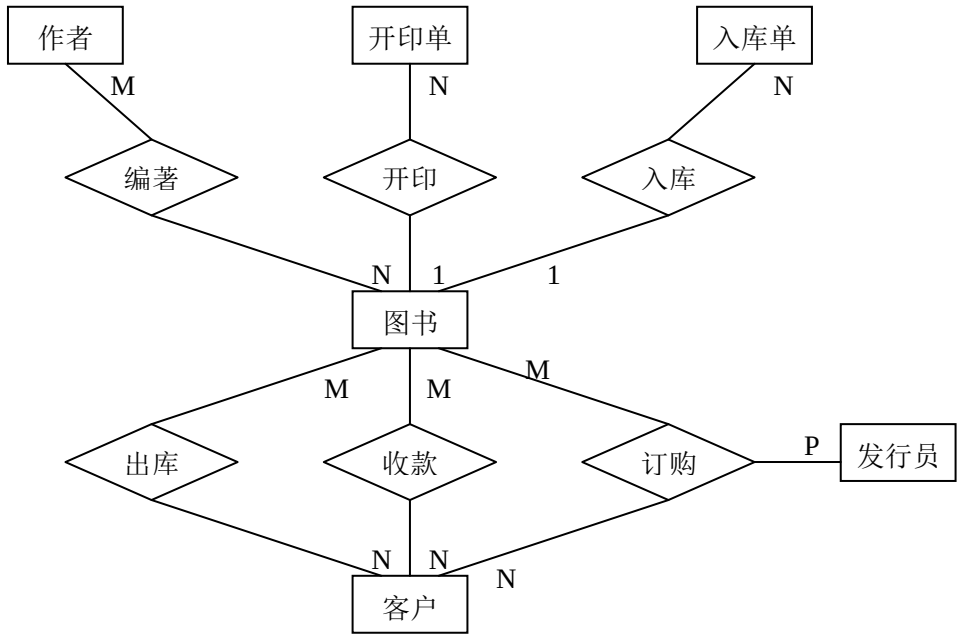


图 5.14　图书发行系统的 ER 图

该 ER 图有 6 个实体类型，其结构如下：

图书（图书编号，书名，定价，包本数，开本，统一书号，库存量）

作者（作者编号，姓名，性别，地址，电话）

开印单（印单号，开单日期，定价，印数，制单人）

入库单（入库单号，日期，送书单位，数量，包本数，版印次）

发行员（发行员代号，姓名，电话）

客户（客户编号，名称，地址，开户行，账号，税号，收款方式）

实体类型之间有 6 个联系，其中 2 个 1:N 联系，3 个 M:N 联系，1 个 M:N:P 联系，在图上均已标出。其中联系的属性如下所示。

订购（订购单号，日期，数量）

出库（出库单号，日期，数量，包本数）

收款（收款单号，金额，收款日期）

编著（日期，备注）

试将 ER 图转换成关系模型，并注明主键和外键。

5．某学员为上海闵行区物资供应公司设计了库存管理信息系统，对货物的库存、销售等业务活动进行管理。其 ER 图如图 5.15 所示。

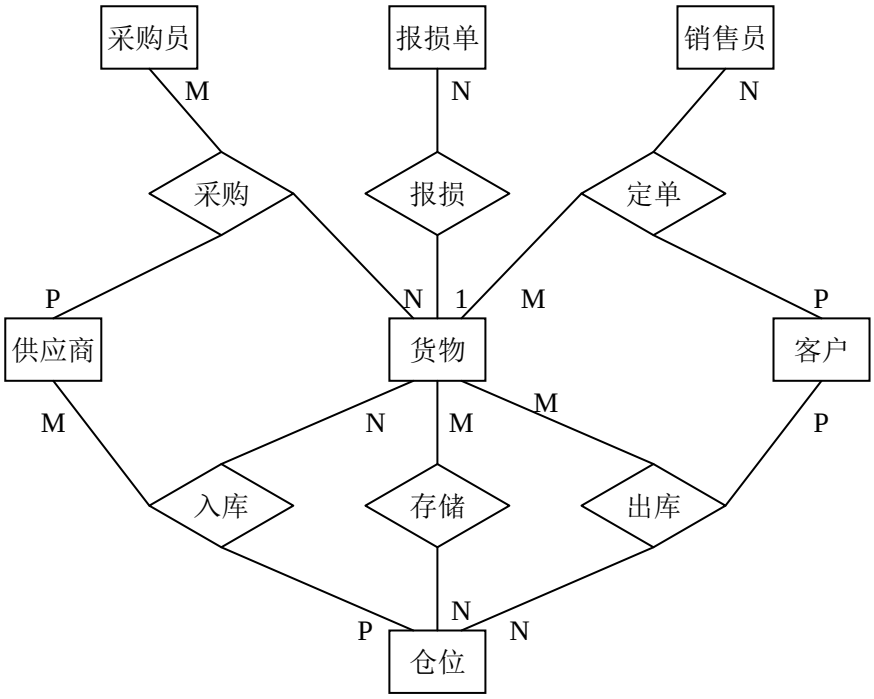


图 5.15　库存管理系统的 ER 图

该 ER 图有 7 个实体类型，其结构如下：

货物（货物代码，型号，名称，形态，最低库存量，最高库存量）

采购员（采购员号，姓名，性别，业绩）

供应商（供应商号，名称，地址）

销售员（销售员号，姓名，性别，业绩）

客户（客户号，名称，地址，账号，税号，联系人）

仓位（仓位号，名称，地址，负责人）

报损单（报损号，数量，日期，经手人）

实体间联系类型有 6 个，其中 1 个 1:N 联系，1 个 M:N 联系，4 个 M:N:P 联系。其中联系的属性如下。

入库（入库单号，日期，数量，经手人）

出库（出库单号，日期，数量，经手人）

存储（存储量，日期）

定单（定单号，数量，价格，日期）

采购（采购单号，数量，价格，日期）

试将 ER 图转换成关系模型，并注明主键和外键。

5.4 自测题答案

5.4.1 填空题答案

1．总体信息需求　　处理需求　　DBMS 特征　　硬件和 OS 特性

2．完整的数据库结构　　应用程序设计原则

3．系统调查　　可行性分析　　确定总目标和制定项目开发计划

4．业务流程图 系统范围图　　数据流程图　　数据字典

5．数据项　　数据结构　　数据流　　数据存储　　加工过程

6．企业组织信息需求

7．设计局部概念模式　　综合成全局概念模式　　评审

8．自顶向下逐步细化　　自底向上逐步综合

9．把概念模式转换成 DBMS 能处理的模式

10．形成初始模式　　设计子模式　　应用程序设计梗概　　模式评价　　模式修正

11．　存储记录结构设计　　确定数据存储安排　　访问方法的设计
　　完整性安全性设计　　程序设计

12．DBA

13．DB 的转储与恢复　　DB 的安全性与完整性控制　　DB 性能的监督、分析和改进
　　DB 的重组织和重构造

5.4.2 单项选择题答案

1．D　　2．C　　3．C　　4．D　　5．B　　6．B
7．B　　8．B　　9．A　　10．B　　11．C

5.4.3 设计题答案

这个数据库一种可能的 ER 图如图 5.16 所示，图中只画出实体、联系，未画出其属性。

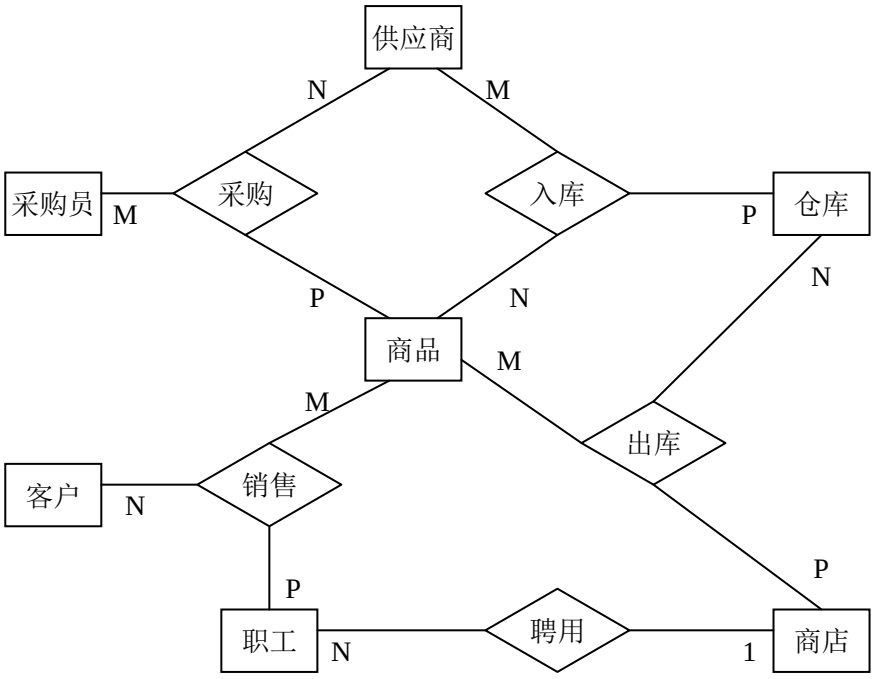


图 5. 16　库存管理系统的 ER 模型

5.4.4　ER 图实例答案

1．解：根据 ER 图和转换规则，8 个实体类型转换成 8 个关系模式，2 个 M：N 联系转换成 2 个关系模式。因此，图 5. 11 的 ER 图可转换成 10 个关系模式，如下所示：

病人（住院号，姓名，性别，地址，病房编号，床位号，入院日期，出院日期）
医生（医生工号，姓名，职称）
护士（护士工号，姓名，职称，手术室编号）
病床（病床编号，床位号，类型，空床标志，护士工号）
手术室（手术室编号，类型）
手术（手术标识号，类型，日期，时间，费用，手术室编号，医生工号，住院号）
诊断书（诊断书编号，科别，诊断，医生工号，住院号）
收据（收据编号，项目，金额，收款员，日期，住院号）
协助（手术标识号，医生工号，角色）
处方（处方单号，序号，药品名称，规格，数量，费用，住院号，医生工号）

2．解：根据 ER 图和转换规则，7 个实体类型转换成 7 个关系模式，1 个 M：N 联系和 4 个

M：N：P 联系转换成 5 个关系模式。因此，图 5.12 的 ER 图可转换成 12 个关系模式，如下所示：

商品（商品编号，名称，类别，单位，单价）
供应商（供应商编号，名称，账号，地址）
仓库（仓库编号，地址，负责人）
门店（门店编号，名称，地址）
采购员（采购员编号，姓名，业绩）
管理员（管理员编号，姓名，业绩，仓库编号）
营业员（营业员编号，姓名，业绩，门店编号）
采购（采购单号，数量，日期，采购员编号，供应商编号，商品编号）
进货（进货单号，数量，日期，供应商编号，商品编号，仓库编号）
配送（配送单号，数量，日期，商品编号，仓库编号，门店编号）
销售（销售单号，数量，日期，商品编号，门店编号，营业员编号）
存储（商品编号，仓库编号，日期，库存量，安全库存量）

3．解：根据 ER 图和转换规则，5 个实体类型转换成 5 个关系模式，2 个 M:N 联系转换成 2 个关系模式。因此，图 5.13 的 ER 图可转换成 7 个关系模式，如下：

客户（股东账号，身份证号，姓名，地址，客户类别，开户日期）
资金（资金账号，金额，可取余额，冻结金额，解冻金额，利息，日期）
证券（证券代码，名称，每手股数）
委托（委托序号，股东账号，证券代码，资金账号，数量，买卖类别，价格，时间，操作员）
成交（成交序号，股东账号，证券代码，资金账号，数量，买卖类别，成交价格，时间）
持有（股东账号，证券代码，日期，金额，可用数量，冻结数量，解冻数量）
存取（存取单序号，股东账号，资金账号，存取标志，金额，日期）

4．据转换规则，ER 图中有 6 个实体类型，可转换成 6 个关系模式，另外 ER 图中有 3 个 M:N 联系和 1 个 M:N:P 联系，也将转换成 4 个关系模式。因此，图 5.14 的 ER 图可转换成 10 个关系模式，具体如下：

图书（图书编号，书名，定价，包本数，开本，统一书号，库存量）
作者（作者编号，姓名，性别，地址，电话）
开印单（印单号，开单日期，图书编号，定价，印数，制单人）
入库单（入库单号，日期，送书单位，数量，包本数，版印次，图书编号）
发行员（发行员代号，姓名，电话）
客户（客户编号，名称，地址，开户行，账号，税号，收款方式）
订购（订购单号，日期，数量，客户编号，图书编号，发行员代号）
出库（出库单号，日期，数量，包本数，客户编号，图书编号）
收款（收款单号，金额，收款日期，客户编号，图书编号）
编著（作者编号，图书编号，日期，备注）

5．据转换规则，ER 图中有 7 个实体类型，可转换成 7 个关系模式，另外 ER 图中有 1 个 M:N 联系和 4 个 M:N:P 联系，也将转换成 5 个关系模式。因此，图 5.15 的 ER 图可转换成 12 个关系模式，具体如下：

货物（货物代码，型号，名称，形态，最低库存量，最高库存量）
采购员（采购员号，姓名，性别，业绩）
供应商（供应商号，名称，地址）
销售员（销售员号，姓名，性别，业绩）

客户（客户号，名称，地址，账号，税号，联系人）
仓位（仓位号，名称，地址，负责人）
报损单（报损号，数量，日期，经手人，货物代码）
入库（入库单号，日期，数量，经手人，供应商号，货物代码，仓位号）
出库（出库单号，日期，数量，经手人，客户号，货物代码，仓位号）
存储（货物代码，仓位号，日期，存储量）
定单（定单号，数量，价格，日期，客户号，货物代码，销售员号）
采购（采购单号，数量，价格，日期，供应商号，货物代码，采购员号）

第 6 章　数据库的存储结构

6.1 基本内容分析

6.1.1 本章重要概念

本章有以下一些重要概念：

（1）计算机系统的存储介质层次。

（2）两种文件组织：定长记录和变长记录。被拴记录，悬挂指针，分槽式页结构。

（3）四种文件结构：堆文件、顺序文件、散列文件和聚集文件。

（4）索引技术：主索引及三种实现方法（稠密、稀疏、多级索引）；辅助索引；B^+树索引文件；B 树索引文件。

（5）散列技术：散列函数；散列索引；静态散列；动态散列（可扩充散列结构）。

（6）两种多键访问技术：网格文件和分区散列。

6.1.2 本章的重点篇幅

（1）教材中 P214 的图 6.8（分槽式页结构）。

（2）教材中 P224～232 的 B^+树索引文件和 B 树索引文件。

（3）教材中 P236～241 的可扩充散列结构。

（2）教材中 P242～244 的网格文件。

6.2 教材中习题 6 的解答

6.1 名词解释

（1）• 定长记录文件：记录为定长格式的文件。

• 变长记录文件：记录为变长格式的文件。

• 被拴记录（pinned　record）：被指针指向的记录，称为被拴记录。

• 悬挂指针（dangling　pointer）：如果指针指向的记录已被删除，那么该指针称为悬挂指针。悬挂指针指向的空间称为“垃圾”，别人无法使用。

（2）• 堆文件：以输入顺序为序的文件，称为堆文件。

• 顺序文件：记录按查找键值升序或降序的顺序存储的文件，称为顺序文件。

• 散列文件：将记录的某个属性值通过散列函数求得的值作为记录的存储地址的文件，称为散列文件。

• 聚集文件：可以存储多个关系（表）的记录的文件，称为聚集文件。

（3）• 有序索引：根据记录中某种排序顺序建立的索引，称为有序索引。

• 主索引：如果索引的查找键值的顺序与主文件的顺序一致，那么这种索引称为主索引，也称为聚集索引。

• 稠密索引：对于主文件中每一个查找键值建立一个索引记录，索引记录包括查找键值和指向具有该值的记录链表的第一个记录的指针。这种索引称为“稠密索引”。

• 稀疏索引：在主文件中，对若干个查找键值才建立一个索引记录，这种索引称为“稀疏索引”。

• 多级索引：在索引很大时，还可对索引建立索引，这样就形成树结构的多级索引。

• 辅助索引：不是根据主索引的查找键值，而是根据其他查找键值来寻找主文件的记录，这种索引称为辅助索引。

• 平衡树：一棵 m 阶平衡树或者为空，或者满足以下四个条件：

每个结点至多有 m 棵子树；根结点或为叶结点，或至少有两棵子树；

每个非叶结点至少有 m/2 棵子树；叶结点在同一层次上。

• B^+树：一棵 m 阶 B^+树是平衡树，多个结点至多有 m-1 个查找键值和 m 个指向子树的指针，但叶结点中的指针指向主文件中的记录，而非叶结点形成了叶结点上的一个多级稀疏索引。

• B 树：B 树类似于 B^+树，B 树中所有查找键值只能出现一次，但可出现任何结点上。

（4）• 散列方法：根据记录的查找键值，使用一个函数计算得到的函数值，作为磁盘块的地址，对记录进行存储和访问，这种方法称为散列方法。

• 桶溢出（散列碰撞）：在散列组织中，每个桶的空间是固定的，如果某个桶内已装满记录，还有新的记录要插入到该桶，这种现象称桶溢出。

• 封闭散列法：即溢出桶拉链法。某桶号的空间分成基本桶和溢出桶两种。

• 开放式散列法：把桶的集合固定下来，也就是只考虑基本桶，不考虑溢出桶。如果有一个桶装满了记录，还需装入新记录时，就在桶集中挑选一个有空闲空间的桶去装新记录。

（5）• 散列索引：把查找键值与指针一起组合成散列文件结构的一种索引。

• 静态索引：在散列函数确定以后，所有的桶地址及桶空间都确定了。这种技术称为“静态散列”技术。

• 动态散列：桶空间可以随时申请或释放的散列技术，称为“动态散烈”技术。

• 可扩充散列：对静态散列中成倍扩充法的改进，能随时根据需要申请和释放桶。

（6）• 单键索引：只使用一个查找键的查询，称为单键查询。

• 多键查询：使用多个查找键的查询，称为多键查询。

• 网格文件：网格文件是由网格矩阵和线性标尺组成的结构，网格矩阵中每个格子中有一个指针，指向一个桶。

• 分区散列：是对散列技术的扩充，能允许在多个属性上进行索引。

6.2　试叙述计算机系统的物理存储介质层次,并说明每一种介质的数据访问速度。

答：根据访问数据的速度、成本和可靠性，计算机系统的存储介质可分成以下六类：

① 高速缓冲存储器（cache）：这是一种静态的随机访问存储器（Static Random Access Memory，简记为 SRAM）。CPU 用 cache 存储器来加快程序的执行。

② 主存或内存：这是一种动态的随机访问存储器（Dynamic RAM，简记为 DRAM）。现在微机的内存已达 200MB。

上述两种存储器是一种易失性存储器，即掉电时会丢失存储的内容。

③ 快闪存储器（Flash Memory）：这种存储器采用 EEPROM（电可擦写可编程只读存储器）技术，其优点是存取速度快，缺点是必须一次擦写或写入。其容量已达 32 兆位，存取速度 7×10^{-8}/s，写传输速度是 430KB/s。

④ 磁盘：是一种直接访问存储器，现在微机上磁盘的容量已达 180GB，I/O 传输速度达 80MB/s。

上述两种存储器属于非易失性存储器，也称为联机存储器。

⑤ 光盘：这种存储器是利用光学原理来存储数据，并通过激光元件来读取数据。自动光盘机的容量已达数千吉字节，旋传速度达 400 转/分钟，传输速度为 200KB/s。数据视频盘（DVD）的容量在 4～15 吉字节之间。

⑥ 磁带：主要用于数据存档和备份。磁带的容量为 1600～6250 字节/英寸，一般可达 50 吉字节。自动磁带机可以存储数量级达太字节的数据。

上述两种存储器是一种脱机存储器，又称为第三级存储器。

6.3 试对“被拴记录”下个确切的定义。被拴记录在物理存储中起什么作用？有什么利弊？

答：在数据库中，被指针指向的记录，称为“被拴记录”。被拴记录表示记录已被其他用户引用。如果不小心把被拴记录删掉，那么指向该记录的指针成了“悬挂指针”。悬挂指针指向的空间是“垃圾”，别人无法使用。

6.4 在教材（P212）的图 6.5 中，删除记录 5。试比较使用下面各种操作时的利弊：

①把记录 5 以下的记录依次移上一个记录位置。②把最后一个记录（记录 7）移到记录 5 的位置。③在记录 5 中置删除标志位，不移动记录。

答：①把被删记录后的记录依次移上来，平均要移动文件中的一半记录。

②把文件中最后一个记录填补到被删记录位置，这时只要移动一个记录。

③在被删结点处置删除标志位，这时使指向该记录的指针成为悬挂指针。

6.5 在教材（P213）的图 6.6 的文件中，画出执行下列三个操作后的文件结构：

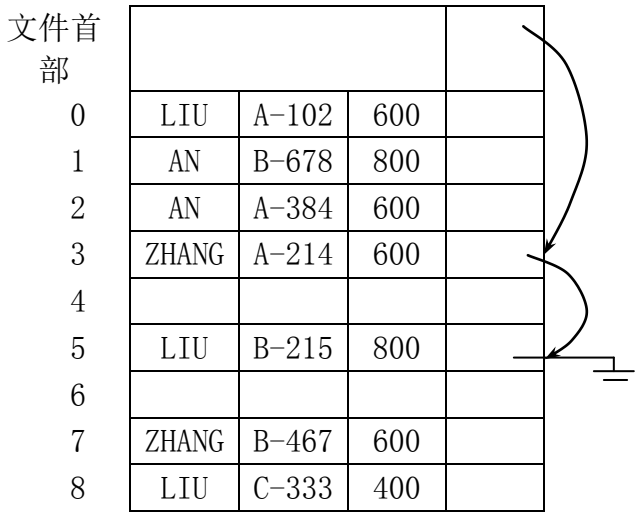
①插入（AN，B-678，800）记录；②删除记录 2；③插入（AN，A-384，600）记录。

答：解：①插入记录在“记录 1”处。

②删掉记录 2，并且记录 2 链接到被删结点链表的链首处。

③插入记录在“记录 2”处。

此时图见图 6.1。



文件首部				
0	LIU	A-102	600	
1	AN	B-678	800	
2	AN	A-384	600	
3	ZHANG	A-214	600	
4				
5	LIU	B-215	800	
6				
7	ZHANG	B-467	600	
8	LIU	C-333	400	

图 6.1

6.6 试举一个数据库应用例子，说明在表达变长记录时，有时预留空间方法要比指针形式好。并作解释。

答：譬如在文件中表达学生和选修课程情况，如果每个学生选修课程门数都在 12~15 门之间，那么此时使用预留空间形式较好，空间浪费较小。

6.7 试举一个数据库应用例子，说明在表达变长记录时，有时指针形式比预留空间方法好。并作解释。

答：譬如在文件中表达学生和其奖惩情况，每个学生的奖惩项目的差别是比较大的，那么此时用指针表示方式较好。

6.8 在教材（P215）的图 6.9 变长记录预留空间的文件中，画出执行下列三步操作后的文件结构：

①插入（HE，E-254，800）记录；②插入（LOU，C-293，600）记录；

③删除（LIU，C-333，400）记录。

解：执行题中三步操作后的文件结构如图 6.2 所示。

0	LIU	A-102	600	B-215	800	⊥	⊥
1	WEN	B-306	700	⊥	⊥	⊥	⊥
2	HE	F-257	800	E-254	800	⊥	⊥
3	ZHANG	A-214	600	B-467	600	⊥	⊥
4	ZHOU	C-343	750	⊥	⊥	⊥	⊥
5	LOU	B-428	850	C-293	600	⊥	⊥

图 6.2

6.9 在教材（P215）的图 6.9 的文件中，如果还要插入（LIU，F-834，750）记录，会发生什么现象？如何处理？有何利弊？

解：此时，LIU 的记录长度超过最大长度，此时有两种实现方式：

一种是改组文件，增加记录长度，以适应应用的需要。但这样有可能造成空间浪费。

另一种是再取一个记录，放（LIU, F-834, 750）值，但是应该使这一条记录与原来的记录联系起来，可通过指针等方法来实现。

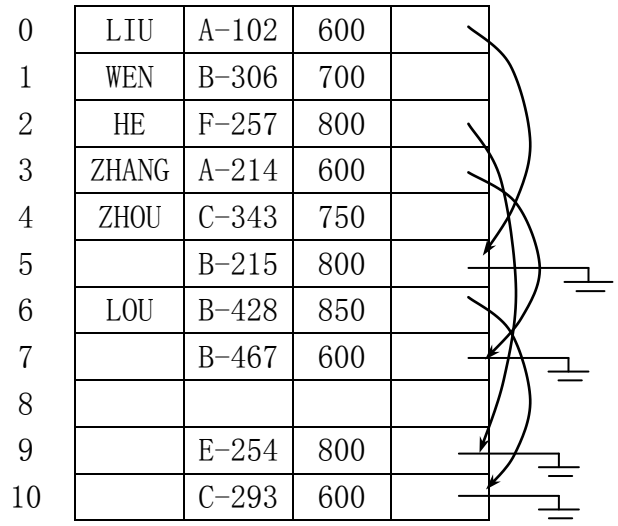
6.10 在教材（P216）的图 6.10 变长记录指针表示方式的文件中，画出执行下列三步操作后的文件结构：

①插入（HE，E-254，800）记录；

②插入（LOU，C-293，600）记录；

③删除（LIU，C-333，400）记录。

解：执行题中三步操作后的文件结构如图 6.3 所示。



0	LIU	A-102	600	
1	WEN	B-306	700	
2	HE	F-257	800	
3	ZHANG	A-214	600	
4	ZHOU	C-343	750	
5		B-215	800	
6	LOU	B-428	850	
7		B-467	600	
8				
9		E-254	800	
10		C-293	600	

图 6.3

6.11 在顺序文件组织中，为什么只有一个溢出记录时，仍然要申请一个溢出块存放这个溢出记录？

答：由于文件被组织成物理块的形式，当数据块放满时，只能再申请一个溢出块，把溢出记录放到溢出块中。

6.12 在关系数据库存储时，试说出下面每种存储技术的两个优点和两个缺点：

①每个文件中只存储一个关系；②每个文件中存储多个关系（或整个数据库）。

答：①每个文件存储一个关系。其优点是：管理简单，一般的 OS 就能实现；系统实现方便，一般适用于规模小的数据库。其缺点是文件之间分离，没有联系；数据联系通过软件实现，效率较低。

②每个文件中存储多个关系。其优点是：使有联系的数据混放在一起，提高查询速度；加强数据之间的联系。其缺点是：增加了系统的复杂性；对两个文件中的查询处理，速度将变慢。

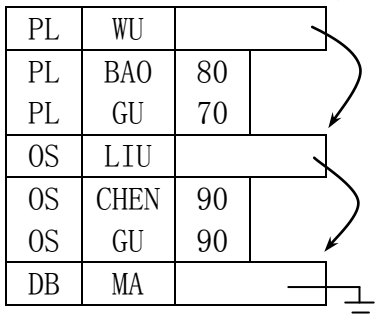
6.13 设关系数据库中有两个关系：

COURSE（COURSE_NAME，TEACHER）

ENROLLMENT（COURSE_NAME，STUDENT_NAME，GRADE）

设有三门课程，五个学生，学生与课程间有选修联系。试用聚集文件表示这两个关系的文件结构。

解：设有三门课程：PL、OS、DB，五个学生：BAO、CHEN、GU、HAI 和 YU。如果用户查询多数是从课程找选修的学生。那么可用图 6.4 表示这个文件：



PL	WU		
PL	BAO	80	
PL	GU	70	
OS	LIU		
OS	CHEN	90	
OS	GU	90	
DB	MA		

DB	BAO	85
DB	YU	95

图 6.4

6.14 什么情况下使用稠密索引比稀疏索引要好？并请作必要的解释。

答：在存取时间和空间开销方面，如果强调存取时间，那么应采用稠密索引。这是因为稠密索引是对每个查找键值建立一个索引记录，所以查找速度较快。

6.15 索引机制加快了查询处理，但为什么文件只在单属性的查找键上建索引？尽可能所出你的理由。

答：文件只在单属性的查找键上建索引，这样可以简化系统的管理。如果一个索引有多个属性构成，那么在查询时，若查询条件不是等值操作，而是比较操作，就会带来复杂性，会有较多的 I/O 操作。

6.16 主索引和辅助索引之间有什么区别？

答：两种索引的区别在于：

主索引是指索引的查找键值的顺序与主文件的顺序一致的索引。而辅助索引，是指索引的查找键值的顺序与主文件的顺序不一致的索引。

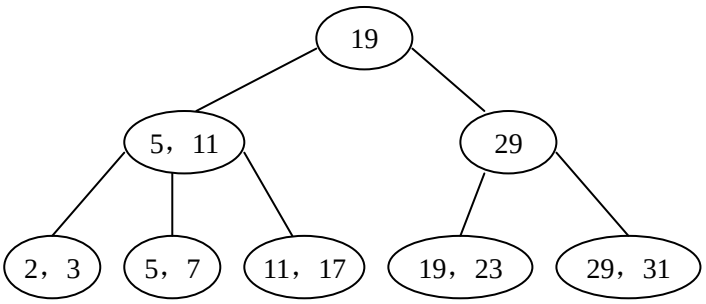
6.17 在同一关系上能否对两个不同查找键值建立两个主索引？为什么？

答：在同一关系上不允许对两个不同查找键值建立两个主索引，这是因为大多数情况下，主文件的顺序不可能同时与两个索引的查找键值顺序一致。

6.18 设查找键值集为{2，3，5，7，11，17，19，23，29，31}。假设初始时 B+树为空，按升序次序插入键值。就下面三种情况建立三棵 B+树：

① 4 阶； ② 6 阶； ③ 8 阶。

解：①4 阶 B^+树如图 6.5 所示。



符合 B^+树要求：
每个叶结点至少应有 2 个查找键，至多有 3 个查找键；每个非叶结点至少应有 2 个指针，至多有 4 个指针；根结点或者没有指针，或者至少 2 个指针至多有 4 个指针。

图 6.5

②6 阶 B^+树如图 6.6 所示。

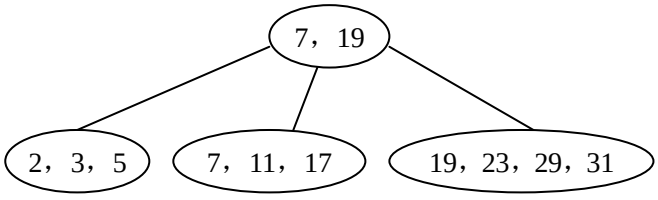


图 6.6

③8 阶 B^+树如图 6.7 所示。

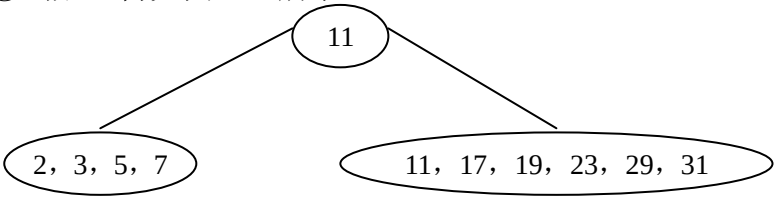


图 6.7

6.19 对 6.18 题中建立的 B+树，试叙述下列查找操作的过程：

① 找查找键值为 11 的记录；② 找查找键值在 7~17 之间的记录；

解：B^+树的查询方法如下：

要检索查找键值为 k 的所有记录，首先在根结点中找大于 k 的最小查找键值（设为 K_i），然后沿着 K_i 左边的指针 P_i 到达第二层的结点。在第二层的结点，用类似的方法找到一个指针，进入第三层的结点……，一直到进入 B^+树的叶结点，找到一个指针直接指向主文件的记录，或指向一个桶（存放指向主文件记录的指针），最后把所需记录找到。

（对 6.18 题中建立的 B^+树的查找过程略）

6.20 对 6.18 题中建立的 B+树，画出执行下列每一步操作后 B+树：

① 插入 9； ② 插入 10； ③ 插入 8； ④ 删除 23； ⑤ 删除 19

解：（1）对 6.18 题中的 4 阶 B^+树，在插入查找键值 9，10，8 后，如图 6.8 所示。

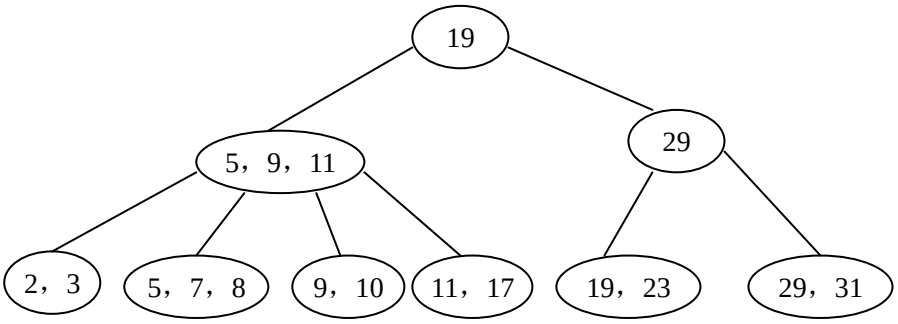


图 6.8

再删除 23 后，如图 6.9 所示。

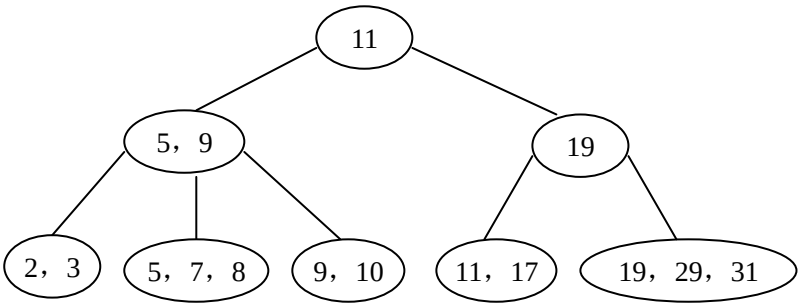


图 6.9

再删除 19 后，如图 6.10 所示。

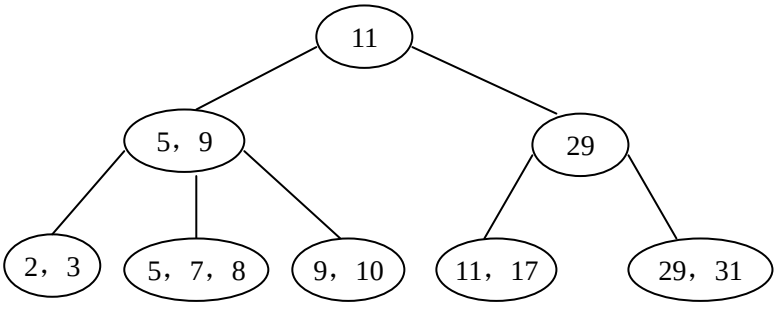


图 6.10

（2）对 6.18 题中的 6 阶 B^+树，在插入键值 9、10、8 后，如图 6.11 所示。

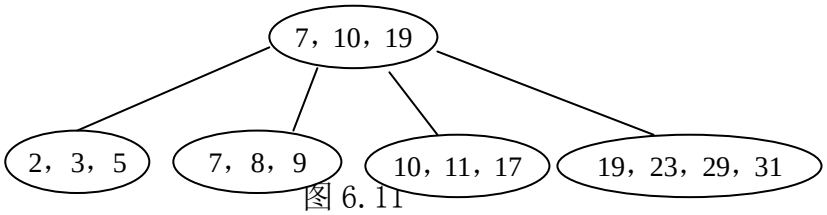


图 6. 11

再删除 23、19 后，如图 6. 12 所示。

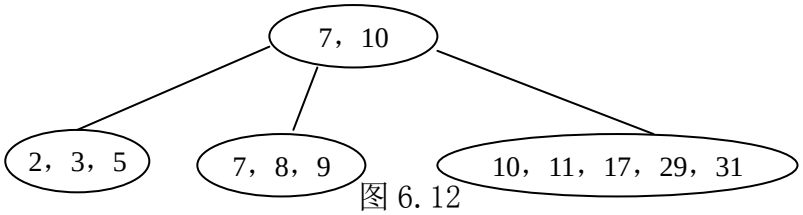


图 6. 12

（3）对 6.18 题中的 8 阶 B^+树，在插入键值 9、10、8 和删除 23、19 后，如图 6. 13 所示。

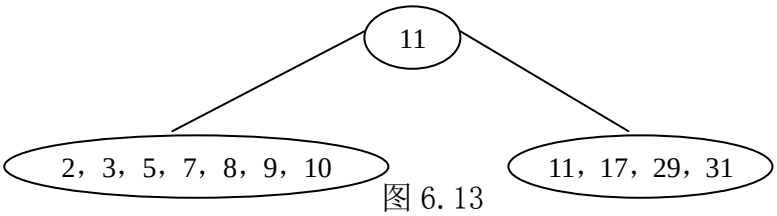


图 6. 13

6.21 对于 6. 18 题的查找键值集，建立 B 树。

解：据教材中的描述，m 阶 B 树，非叶结点中最多可包含 m 个指针和 m-1 个查找键值，而叶结点最多可包含 n 个指针和 n-1 个查找键值，而 n 和 m 间关系有 n>m。

（1）对 6.18 题中的查找键值集，建立 4 阶 B 树。

此时非叶结点中最多 3 个查找键值，而假设叶结点中最多可以 4 个查找键值。

再插入 2、3、5、7、11 后的 B 树如图 6. 14 所示。

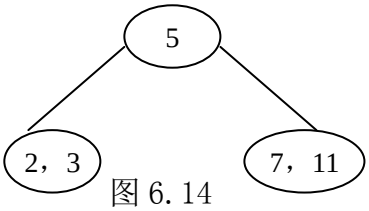


图 6. 14

再插入 17、19、23、29、31 后的 B 树如图 6. 15 所示。

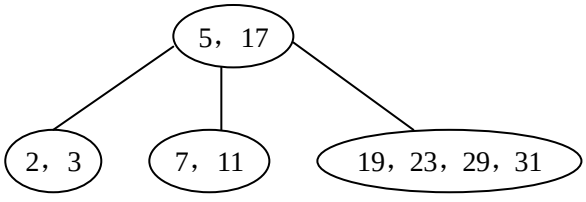


图 6. 15

（2）对 6.18 题中的查找键值集，建立 6 阶 B 树。

此时非叶结点中最多 5 个查找键值，而假设叶结点中最多可以 6 个查找键值。

在插入 10 个键值后的 B 树如图 6. 16 所示。

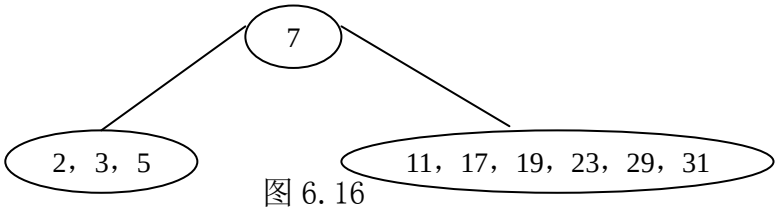


图 6. 16

（3）对 6.18 题中的查找键值集，建立 8 阶 B 树。

此时非叶结点中最多 7 个查找键值，而假设叶结点中最多可以 8 个查找键值。

在插入 10 个键值后的 B 树如图 6. 17 所示。

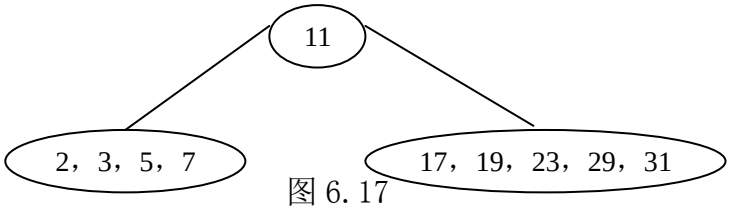


图 6. 17

6.22 封闭式散列法和开放式散列法之间有什么区别？在数据库应用中，这两种方法各有什么利弊？

答：封闭式散列法和开放式散列法是指在桶溢出时所采用的方法。

（1） 封闭式散列法是指每个桶号的存储空间分成基本桶和溢出桶两种。溢出桶链接成一条溢出链，查找某桶号的数据就在这条溢出链中进行，不会到其他溢出链中查找。

（2） 开放式散列法是把桶集固定下来，也就是只考虑基本桶，不考虑溢出桶。如果有一个桶装满了记录，那么就在桶集中挑选一个有空闲空间的桶，装入新记录。

封闭散列法的优点是查找速度快，但结构比较复杂。开放散列法的空间较省，但插入、删除操作比较复杂。所以现在大多数 DBS 中都是使用封闭散列法。

6.23 在散列文件组织中，是什么原因引起桶溢出的？有什么办法能减少桶溢出的次数？

答：产生桶溢出的原因有两个：

初始设计时桶数偏少；散列函数的“均匀分布性”不好。

对于前一个原因，在设计散列函数时，桶数应放宽些，一般存储空间应有 20%的余量，让它空闲着，以利减少桶溢出的机会。对于后一个原因，不管散列函数如何好，再留有空间余量，桶溢出现象难免还会发生，因此用封闭散列法和开放式散列法来解决桶溢出问题。

6.24 设查找键值集为{2，3，5，7，11，17，19，23，29，31}，散列函数为 h（x）=（x mod 8），每个桶可存储 3 个记录。试建立一个可扩充散列结构，并画出示意图。

解：查找键值的散列值为：

键值 X	散列值 h（X）
2	2（010）
3	3（011）
5	5（101）
7	7（111）
11	3（011）
17	1（001）
19	3（011）
23	7（111）
29	1（001）
31	7（111）

插入键值 2、3、5 后的可扩充散列结构，如图 6. 18 所示。

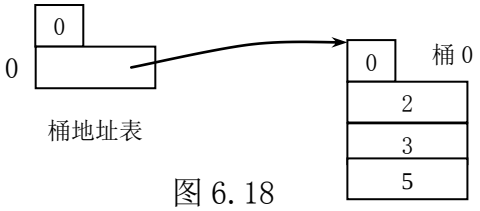


图 6. 18

在插入键值 7、11 后的结构图，如图 6.19 所示。

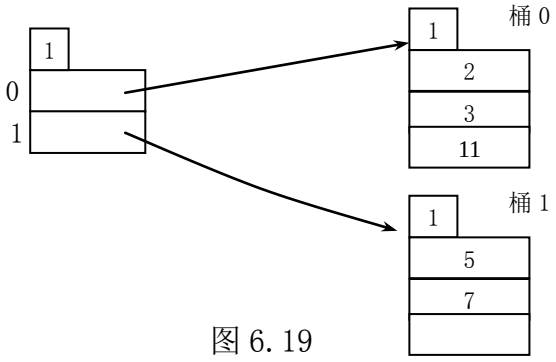


图 6.19

再插入键值 17 后的结构图，如图 6.20 所示。

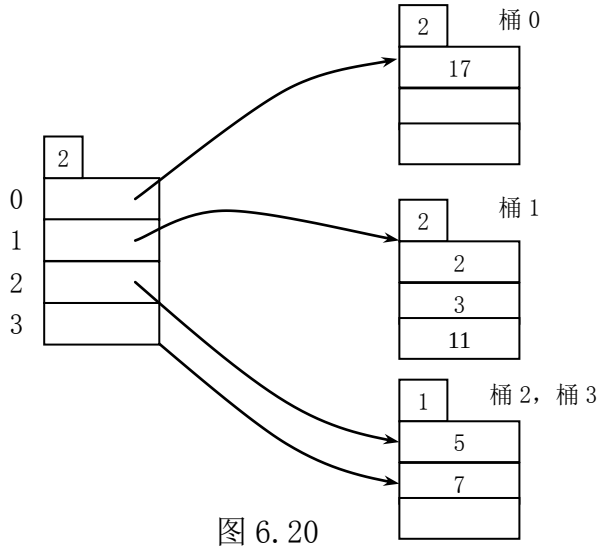


图 6.20

再插入键值 19、23、29、31 后的结构图，如图 6.21 所示。

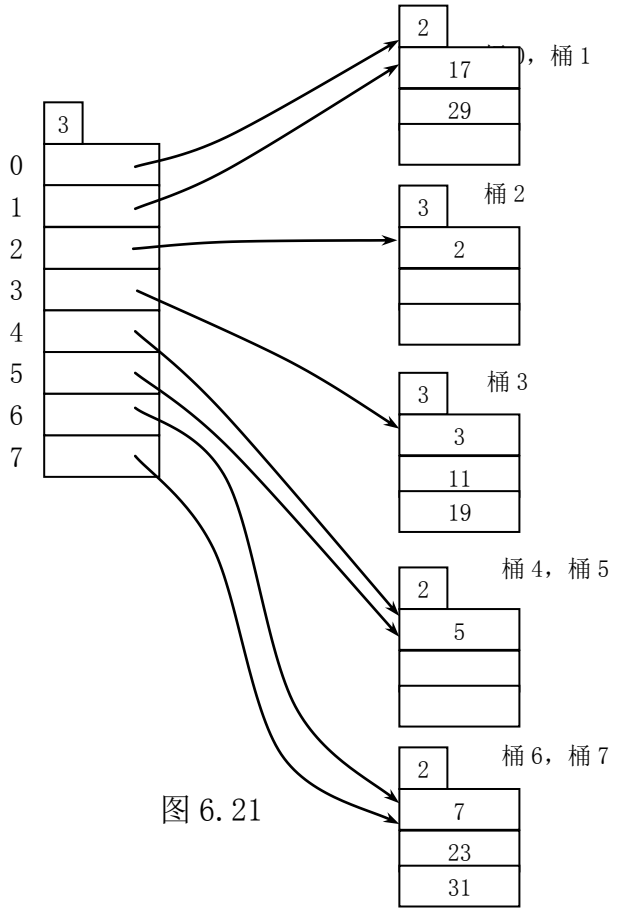


图 6.21

6.25 对 6.24 题中的可扩充散列结构，试叙述执行下列各操作后可扩充散列结构如何变化：

① 删除 11； ② 删除 31； ③ 插入 1； ④ 插入 15。

解：在 6.24 题的可扩充散列结构中，执行本题的各个操作后，得到的结构图如图 6.22 所示。

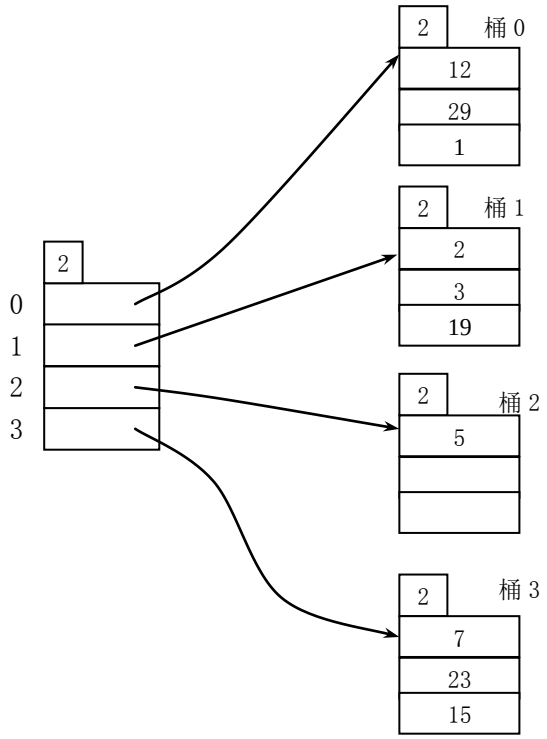


图 6.22

6.26 为什么散列结构不适宜对查找键进行范围查询？

答：由于范围查询的查找键是分散在各个桶中，因此散列结构不适宜对查找键进行范围查询。

第 7 章 系统实现技术

7.1 基本知识点

7.1.1 本章重要概念

（1）系统目录及其和 DBMS 各子系统的联系。

（2）事务的定义，COMMIT 和 ROLLBACK 的语义，事务的 ACID 性质，事务的状态变迁图。

（3）存储器类型，稳定存储器的实现，数据传送过程。

（4）恢复的定义、基本原则和实现方法，故障的类型，检查点技术，REDO 和 UNDO 操作，运行记录优先原则。

（5）并发操作带来的三个问题，X 锁、PX 协议、PXC 协议，S 锁、PS 协议、PSC 协议，活锁、饿死和死锁，并发调度，串行调度，并发调度的可串行化，两段封锁法，SQL 中事务的存取模式和隔离级别。

（6）完整性的定义，完整性子系统的功能，完整性规则的组成。SQL 中的三大类完整性约束，SQL3 中的触发器技术。

（7）安全性的定义、级别，权限，SQL 中的安全性机制，几种常用的安全性措施，自然环境的安全性。

7.1.2 本章的重点篇幅

（1）教材中 P261　的图 7.7。（检查点技术）

（2）并发操作带来的四个问题，封锁带来的三个问题，并发调度的可串行化。（教材 P265-276）

（3）SQL 中完整性约束的实现：断言（教材 P290）。

（4）安全性中的授权语句（教材 P298）。

7.2 教材中习题 7 的解答

7.1 名词解释

• 系统目录：系统目录(system catalog)是任何通用 DBMS 的核心。系统目录本身就是一个“微型数据库”，其主要功能是存储 DBMS 管理的数据库的定义或描述。这类信息被称为元数据（metadata），主要包括数据库三级结构、两级映像的定义。

• 事务：事务是构成单一逻辑工作单元的操作集合。

• DB 的可恢复性：系统能把 DB 从被破坏、不正确的状态，恢复到最近一个正确的状态，DBMS 的这种功能称为 DB 的可恢复性。

• 并发操作：在多用户共享系统中，许多事务可能同时对同一数据进行操作，这种操作称为并发操作。

• 封锁：封锁是系统保证对数据项的访问以互斥方式进行的一种手段。

• X 锁：事务 T 对某数据加了 X 锁后，其他事务要等 T 解除 X 锁后，才能对这个数据进行封锁。

• PX 协议：只有获准 X 锁的事务，才能修改数据，否则这个事务进入等待状态。

• PXC 协议：PX 协议再加上一条规则：“X 锁必须保留到事务终点才能解除”。

• S 锁：事务 T 对某数据加了 S 锁后，仍允许其他事务再对该数据加 S 锁，但在对该数据的所有 S 锁都解除之前决不允许任何事务对该数据加 X 锁。

• PS 协议：获准 S 锁的事务，只能读数据，不能修改数据。

• PSC 协议：PS 协议再加上一条规则：“S 锁必须保留到事务终点才能解除”。

• 活锁：系统可能使某个事务永远处于等待状态，得不到封锁的机会，这种现象称为“活锁”。

• 饿死：若干事务连续不断地对某数据实现加 S 锁和释放 S 锁的操作，那么若有一个事务欲对该数据加 X 锁，将永远轮不上封锁的机会。这种现象称为“饿死”。

• 死锁：若干事务都处于等待状态，相互等待对方解除封锁，结果造成这些事务都无法继续执行，这种现象称为系统进入了“死锁”状态。

• 调度：事务的执行次序称为“调度”。

• 串行调度：多个事务依次执行，称为事务的串行调度。

• 并发调度：利用分时的方法，同时处理多个事务，则称为事务的并发调度。

• 可串行化调度/不可串行化调度：如果一个并发调度的执行结果与某一串行调度的执行结果等价，那么这个并发调度称为“可串行化的调度”，否则称为“不可串行化调度”。

• 两段封锁协议：事务分成两个阶段，前一阶段只能申请封锁，后一阶段只能释放封锁，这一规则称为事务的两段封锁协议。

7.2 关系 DBMS 的系统目录中存储那些信息?

答：关系 DBMS 的系统目录存储下列信息：

① 关系名，属性名，属性域（数据类型）；

② 各种约束，主键，辅助键，外键，空值/非空值；

③ 视图的外部级描述，存储结构和索引的内部级描述；

④ 安全性和授权规则；

⑤ 数据完整性规则。

在关系 DBMS 中，系统目录被组织成关系（表格），例如 Oracle 系统中，系统目录由 42 个关系组成。DBMS 可以对目录执行查询、修改和维护操作；而用户一般只能执行查询操作不能进行修改或维护。

7.3 试叙述系统目录中三级模式的信息是如何存储的?

答：系统目录中存储三级模式结构的信息主要有以下一些表格：

① 存储数据库中关系模式定义的表格。其属性有关系模式的名字、属性名、类型名、是否主键、是否外键等。

② 存储数据库中视图定义的表格要两张。一张表格用于存储视图的定义，另一张表格用于存储视图中的属性。

③ 有关数据库中关键码、索引的信息，各用一张表格。

7.4 为什么必须有效地访问 DBMS 的系统目录?试举例说明。

答：在 DBS 运行时，DBMS 各个子系统都要频繁地使用系统目录。所有用户的逻辑数据和磁盘中物理数据，都要与系统目录中的数据定义去比较。也就是这两种数据之间的数据传输和格式转换都要在系统目录定义的框架中进行。因此 DBMS 的 DDL 编译程序，DML 的分析程序、编译程序、优化程序，安全性、完整性等检查程序都要随时访问系统目录，必须是有效地访问，否则势必造成系统效率低下，甚至瘫痪、崩溃。

7.5 试把教材中 P251 的图 7.3 扩充的 ER 图转换成等价的关系模式集。

解：这个扩充的 ER 图可转换成如下的关系模式集。由于教材中 ER 图里的属性较少，这里又增加了一些属性。

关系（关系编号，关系名，数据库编号）

视图关系（视图关系编号，视图名，创建者，创建时间）

视图属性（视图属性编号，属性名，类型，视图关系编号）

基本关系（基本关系编号，关系名，创建者，创建时间）

属性（属性编号，属性名，类型，基本关系编号，键编号，索引编号）

键（键编号，键名，键类型，基本关系编号，索引编号）

外键（外键编号，外键名，基本关系编号）

索引（索引编号，索引名，索引类型，基本关系编号）

7.6 试叙述事务的四个性质，并解释每一个性质由 DBMS 的哪个子系统实现？每一个性质对 DBS 有什么益处？

答：① 事务的原子性，是指一个事务对 DB 的所有操作，是一个不可分割的工作单元。原子性是由 DBMS 的事务管理子系统实现的。事务的原子性保证了 DBS 的完整性。

② 事务的一致性，是指数据不会因事务的执行而遭受破坏。事务的一致性是由 DBMS 的完整性子系统实现的。事务的一致性保证数据库的完整性。

③ 事务的隔离性，是指事务的并发执行与这些事务单独执行时结果一样。事务的隔离性是由 DBMS 的并发控制子系统实现的。隔离性使并发执行的事务不必关心其他事务，如同在单用户环境下执行一样。

④ 事务的持久性，是指事务对 DB 的更新应永久地反映在 DB 中。持久性是由 DBMS 的恢复管理子系统实现的。持久性能保证 DB 具有可恢复性。

7.7 事务的 COMMIT 语句和 ROLLBACK 语句各做什么事情？

答：COMMIT 语句表示事务执行成功地结束（提交），此时告诉系统，DB 要进入一个新的正确状态，该事务对 DB 的所有更新都已交付实施（写入磁盘）。

ROLLBACK 语句表示事务执行不成功地结束（应该“回退”），此时告诉系统，已发生错误，DB 可能处在不正确的状态，该事务对 DB 的所有更新必须被撤消，DB 应恢复该事务到初始状态。

7.8 试列出事务状态变迁图（教材 P255 的图 7.4）中所有可能的状态路径。并叙述路径中状态变迁的原因。

答：状态变迁如图 7.1 所示。

状态变迁	变迁的原因
活动状态 => 局部提交状态	事务的最后一个语句执行之后
局部提交状态 => 提交状态	事务未与并发事务发生干扰，并且事务对 DB 修改均已写到磁盘
活动状态 => 失败状态	事务还没到达最后一个语句就中止执行
局部提交状态 => 失败状态	局部提交的事务，发现与其他并发事务发生干扰，或未能完成对磁盘中 DB 的修改
失败状态 => 异常中止	处于失败状态的事务，应该被撤销，即回退

图 7.1

7.9 试解释三种存储器类型之间的区别。

答：从存储器的访问速度、容量和恢复能力角度考察，计算机系统的存储介质可分成三类。

① 易失性存储器，指内存和 cache 存储器。在系统发生故障时，存储的信息会立即丢失。但访问速度最快。

② 非易失性存储器，指磁盘和磁带。在系统故障时，存储的信息不会丢失。但这一类存储器受制于本身的故障，会导致信息的丢失。访问速度要比易失性存储器慢几个数量级。

③ 稳定存储器，这是一个理论上的概念。存储的信息是决不会丢失。这可以通过对非易失性存储器进行技术处理来达到稳定存储器的目标。

7.10 为什么稳定性存储器是不可能实现的？在 DBS 中采用什么方法追求这个目标？

答：由于存储器受制于本身的故障，会导致信息的丢失，因此稳定性存储器是不可能实现的。但可以通过对非易失性存储器进行技术处理来达到稳定性存储器的目标。主要是通过数据备份和数据银行形式实现。

7.11 假设一个 DBS 决不会发生故障，是否还需要有恢复管理机制？试说出理由。

答：即使一个 DBS 不会发生故障，那么仍然需要有恢复管理机制。譬如，事务并发执行时由

封锁产生的死锁问题，以及掉电引起的系统停止运行，这些都是系统运行时遇到的不可抗拒的事件，仍然需要由恢复管理机制利用日志执行 REDO 和 UNDO 处理。

7.12 数据库恢复的基本原则是什么？具体实现方法是什么？

答：恢复的基本原则是“冗余”，即数据重复存储。

为了做好恢复工作，在平时应做好两件事：定时对 DB 进行备份；建立日志文件，记录事务对 DB 的更新操作。

7.13 DBS 中有哪些类型的故障？哪些故障破坏了数据库？哪些故障未破坏数据库，但使其中某些数据变得不正确？

答：DBS 中 DB 的故障主要有三类：事务故障、系统故障和介质故障。前两类故障未破坏 DB，但使其中某些数据变得不正确，此时只要利用日志撤消或重做事务。介质故障将破坏 DB，此时只能把 DB 备份拷贝到新的磁盘，再利用日志重做事务对 DB 的修改。

7.14 “检查点机制”的主要思想是什么？COMMIT 语句与检查点时刻的操作如何协调？

答：“检查点机制”的主要思想是在检查点时刻才真正做到把对 DB 的修改写到磁盘。在 DB 恢复时，只有那些在最后一个检查点到故障点之间还在执行的事务才需要恢复。

事务在 COMMIT 时，事务对 DB 的更新已提交，但对 DB 的更新可能还留在内存的缓冲区，在检查点时刻才真正写到磁盘。因此事务的真正结束是在 COMMIT 后还要加上遇到检查点时刻。

7.15 什么是 UNDO 操作和 REDO 操作？为什么要这样设置？

答：UNDO 和 REDO 是系统内部命令

在 DB 恢复时，对于已经 COMMIT 但更新仍停留在缓冲区的事务要执行 REDO（重做）操作，即根据日志内容把该事务对 DB 修改重做一遍。

对于还未结束的事务要执行 UNDO（撤消）操作，即据日志内容把该事务对 DB 已作的修改撤消掉。

设置 UNDO 和 REDO 操作，是为了使数据库具有可恢复性。

7.16 什么是“运行记录优先原则”？其作用是什么？

答：写一个修改到 DB 中和写一个表示这个修改的登记记录到日志文件中是两个不同的操作，后者比前者重要，后者应先做。这就是运行记录优先原则。其作用是保证 DBS 具有可恢复性。

7.17 数据库的并发操作会带来哪些问题？如何解决？

答：如果不加控制，数据库的并发操作会带来四个问题：丢失更新问题、读脏数据问题、错误求和问题、不可重复读问题。

解决并发操作带来的问题，可以使用封锁技术和时标技术。

7.18 为什么 DML 可以单独提供解除 S 封锁的命令，而不单独提供解除 X 封锁的命令？

答：为防止由事务的 ROLLBACK 引起丢失更新操作，X 封锁必须保留到事务终点，因此 DML 不提供专门的解除 X 锁的操作，即解除 X 锁的操作合并到事务的终点去做。

而在未到事务终点时，执行解除 S 锁的操作，可以增加事务并发操作的程度，但对 DB 不会产生什么错误的影响，因此 DML 可以提供专门的解除 S 锁的操作，让用户使用。

7.19 为什么有些封锁需保留到事务终点，而有些封锁可随时解除？

答：（答案与 7.18 题的解答相同）

7.20 封锁会带来哪些问题？如何解决？

答：封锁技术，可以避免并发操作引起的各种错误，但有可能产生三个问题，其解决办法如下：

①“活锁”问题，可用“先来先服务”排队的方式和提高事务优先级的方法来解决。

②“饿死”问题，可用“授权加锁”方法来避免。

③“死锁”问题，可用抽取某事务作为牺牲品，把它撤销的方法来解决。

7.21 死锁的发生是坏事还是好事？试说明理由。如何解除死锁状态？

答：在 DBS 运行时，死锁状态是我们不希望发生的，因此死锁的发生本身是一件坏事。但是

坏事可以转换为好事。如果我们不让死锁发生，让事务任意并发做下去，那么有可能破坏 DB 中数据，或用户读了错误的数据。从这个意义上讲，死锁的发生是一件好事，能防止错误的发生。

在发生死锁后，系统的死锁处理机制和恢复程序就能起作用，抽取某个事务作为牺牲品，把它撤消，做 ROLLBACK 操作，使系统有可能摆脱死锁状态，继续运行下去。

7.22 试叙述“串行调度”与“可串行化调度”的区别。

答：如果多个事务依次执行，则称事务串行调度。

如果利用分时的方法，同时处理多个事务，则称为事务的并发调度。如果一个并发调度的结果与某一串行调度执行结果等价，则称这个并发调度是可串行化调度。

7.23 SQL 中事务存取模式的定义和隔离级别的定义与数据库的并发控制有什么关系？

答：事务的存取模式是对事务的读/写操作的限制。

事务的隔离机制是对事务并发控制的约束。隔离机制从高到低有四个级别：可串行化，可重复读，读提交数据和可以读未提交数据。

7.24 什么是事务的时标？时标顺序协议的基本思想是什么？

答：在事务 T_i 运行时，有惟一的时间标志，称为时标或时戳。具体地说，事务的时标是指事务的启动时间或某个逻辑编号。

时标顺序协议的基本思想有三点：

① 数据库中，所有的物理更新推迟到 COMMIT 的时候执行。，

② 不允许一个事务查看被较年轻事务更新了的记录，也不允许要求更新被较年轻事务查看过或更新过的记录。如果发生这种事情，那么提出要求的事务被重新启动。

③ 事务重启动时不需要任何回退操作。

7.25 就实现的难度和额外开销两方面，比较阴影页恢复技术和基于日志恢复技术的优缺点。

答：这个比较见图 7.2。

		阴影页恢复技术	基于日志恢复技术
实现难度		实现难度较大	系统中常用的恢复技术
额外开销	优点	减少了额外的 output 操作，恢复比较快，不必做 REDO 和 UNDO 操作	与存储技术无关，利用日志来进行恢复
	缺点	增加了存储技术的复杂性，产生了垃圾回收问题	恢复时，要执行 REDO 和 UNDO 操作

图 7.2

7.26 图 7.3 表示事务 T1 和 T2 的并发执行，在下列条件下各会发生什么情况？

① PSC 协议（并执行两段封锁法）。② PXC 协议。③ 时标顺序协议。

时间	事务 T_1	事务 T_2
t1	* start	
t2		* start
t3	FIND A	
t4		FIND B
t5	FIND B	
t6		FIND A
t7	UPD B	
T8		UPD A

图 7.3 一个并发调度

解：则三种情况的执行示意图分别如图 7.4、图 7.5、图 7.6 所示。

时间	事务 T_1	事务 T_2
t1	* start	
t2		* start
t3	SFIND A	
t4		SFIND B
t5	SFIND B	
t6		SFIND A
t7	UPDX B（失败）	
t8	wait	UPDX A（失败）
t9	wait	wait
t10	wait	wait

（两事务在 t9 时刻进入死锁状态）

图 7.4 执行 PSC 协议（并执行两段封锁法）时的示意图

时间	事务 T_1	事务 T_2
t1	* start	
t2		* start
t3	XFIND A	
t4		XFIND B
t5	XFIND B（失败）	
t6	wait	XFIND A（失败）
t7	wait	wait
t8	wait	wait

（两事务在 t7 时刻进入死锁状态）

图 7.5 执行 PXC 协议时的示意图

时间	事务 T_1	事务 T_2	R_timestamp(A)	W_timestamp(A)	R_timestamp(B)	W_timestamp(B)
$TS(T_1)$	*start*		0	0	0	0
$TS(T_2)$		*start*				
t3	read(A)		$TS(T_1)$			
t4		read(B)			$TS(T_2)$	
t5	read(B)				$TS(T_2)$	
t6		read(A)	$TS(T_2)$			
t7	write(B)					
t8	*restart*	write(A)		$TS(T_2)$		

（为方便，此图中把 FIND 和 UPD 操作改成 read 和 write 操作。
这个例子在 t7 时刻，事务 T_1 的写操作失败，将重新启动）

图 7.6 执行时标顺序协议时的示意图

7.27 什么是数据库的完整性？DBMS 的完整性子系统的主要功能是什么？

答：DB 中完整性是指数据的正确性、有效性和相容性，防止错误的数据进入 DB。

DBMS 完整性子系统的主要功能有两点：监督事务的执行，并测试是否违反完整性规则；若有违反现象，则采取恰当的操作。

7.28 完整性规则由哪几个部分组成？SQL 中的完整性约束有哪些？

答：完整性规则由三部分组成：触发条件，约束条件和 ELSE 子句。

SQL 中把完整性约束分成三大类：域约束。基本表约束和断言。

7.29 参照完整性规则在 SQL 中可以用哪几种方法实现？删除参照关系的元组时，对依赖关系有哪些影响？修改参照关系的主键值时，对依赖关系有哪些影响？

答：参照完整性规则，在 SQL 中可以用外键子句、检查子句、断言等三种方式实现。

删除参照关系的元组时，对依赖关系的影响可以采取下列三种做法之一：

RESTRICT 方式、CASCADE 方式和 SET NULL 方式。（解释略）

修改参照关系的主键值时，对依赖关系的影响也可以采取与上述类似的三种做法之一。（解释略）

7.30 试对 SQL 中检查约束（CHECK 子句）和断言两种完整性约束进行比较，各说明什么对象？何时激活？能保证数据库的一致性吗？

答：检查子句主要用于对属性值、元组值加以限制和约束。断言实际上是一种涉及面广的检查子句，用 CREATE 语句来定义。

这两种约束都是在进行插入或修改时激活，进行检查。

检查子句只在定义它的基本表中有效，而对其他基本表无约束力，因此在与检查子句有关的其他基本表进行修改时，就不能保证这个基本表中检查子句的语义了。

而断言能保证完整性约束彻底实现。

7.31 设教学数据库的关系如下：

S（S#，SNAME，AGE，SEX）

SC（S#，C#，GRADE）

C（C#，CNAME，TEACHER）

试用多种方法定义下列完整性约束：

⑴ 在关系 S 中插入的学生年龄值应在 16~25 岁之间。

⑵ 在关系 SC 中插入元组时，其 S#值和 C#值必须分别在 S 和 C 中出现。

⑶ 在关系 C 中删除一个元组时，首先要把关系 SC 中具有同样 C#值的元组全部删去。

⑷ 在关系 S 中把某个 S#值修改为新值时，必须同时把关系 SC 中那些同样的 S#值也修改为新值。

解：这里每个约束用一种方式定义。

（1）用检查子句定义：

```
CHECK（AGE  BETWEEN  16  AND  25）;
```

（2）在关系 SC 的定义中，用外键子句定义：

```
FOREIGN  KEY（S#）  REFERENCES  S（S#）;
FOREIGN  KEY（C#）  REFERENCES  C（C#）;
```

（3）在关系 SC 的定义中，用外键子句定义：

```
FOREIGN  KEY（C#）  REFERENCES  C（C#）
    ON  DELETE  CASCADE;
```

（4）在关系 SC 的定义中，用外键子句定义：

```
FOREIGN  KEY（S#）  REFERENCES  S（S#）
    ON  UPDATE  CASCADE;
```

7.32 在教学数据库中的关系 S、SC、C 中，试用 SQL 的断言机制定义下列两个完整性约束：

⑴ 学生必须在选修 MATHS 课后，才能选修其他课程。

⑵ 每个男学生最多选修 20 门课程。

解：（1）这个约束可用下列形式表达：

不存在一个学生的选课，这个学生没学过 MATHS 课。

这样就能很容易地写出断言：

```
CREATE  ASSERTION  ASSE1  CHECK
    (NOT  EXISTS( SELECT  S#
                  FROM  SC  X
```

```
                    WHERE   NOT   EXISTS
                        (SELECT   *
                          FROM   SC   Y, C
                          WHERE   Y.C#=C.C#
                             AND   Y.S#=X.S#
                             AND   CNAME='MATHS')));
```

（2）
```
CREATE   ASSERTION   ASSE2   CHECK
    (20>=ALL(SELECT   COUNT(C#)
              FROM   S, SC
              WHERE   S.S#=SC.S#   AND   SEX='M'
              GROUP   BY   S.S#));
```

7.33 什么是数据库的安全性？有哪些级别的安全措施？

答：DB 的安全性是指保护 DB，防止不合法的使用，以免数据的泄密、更新或破坏。为了保护 DB，防止恶意的滥用，可以在从低到高五个级别上设置各种安全措施：环境级、职员级、OS 级、网络级、DBS 级。（解释略）

7.34 对银行的 DBS 应采取哪些安全措施？分别属于哪一级？(解释略)

7.35 什么是“权限”？用户访问数据库有哪些权限？对数据库模式有哪些修改权限？

答：用户（或应用程序）使用 DB 的方式称为权限。

用户访问 DB 有四种权限：Read、Insert、Update 和 Delete。

用户修改 DB 模式有四种权限：Index、Resource、Alteration 和 Drop。

7.36 试解释权限的转授与回收。

答：权限的转授与回收是指 DBS 允许用户把已获得的权限再转授给其它用户，也允许把已授给其它用户的权限再回收上来，但应保证转授出去的权限能收得回来。

为了便于回收，用权限图表示转让关系。

一个用户拥有权限的充分必要条件是在权限图中从根结点到该用户结点存在一条路经。

7.37 SQL 的视图机制有哪些优点？

答：SQL 的视图机制使系统具有三个优点：数据安全性，逻辑独立性和操作简便性。

7.38 SQL2 中的用户权限有哪几类？并作必要的解释。

答：SQL2 中用户权限有六类：SELECT、INSERT、DELETE、UPDATE、REFERENCES 和 USAGE。（解释略）

7.39 安全性措施中强制存取控制是如何实现的？

答：这个方法的基本思想是把数据分成若干个密级级别，用户也分成相应的若干个许可证级别。密级和许可证级别都是严格有序的。在系统运行时，规定用户只能查看比它级别低或同级的数据，用户只能修改和它同级的数据。用这种方法来保证系统的安全性。

7.40 统计数据库是如何防止用户获取单记录信息的？

答：在统计数据系统运行时，对查询应作下列限制：

① 一个查询查到的记录个数不能太少，至少应是 n；

② 两个查询查到的记录的“交”数目不能太多，至多是 m；

系统可以调整 n 和 m 的值，使得用户很难获取个别记录的信息，但要做到完全杜绝是不可能的。

7.41 数据加密法有些什么优点？如何实现？

答：数据加密法的优点是虽然加密键是公开的，但解密键极难推出。用这个方法保证了数据的安全性。（实现方法见教材 P186-187）

7.42 数据库的并发控制、恢复、完整性和安全性之间有些什么联系和区别？

答：这四个方面是一个有机的整体，不可偏废某一方面。DBMS 的这四个子系统一起保证了 DBS 的正常运行。

DB 的可恢复性防止 DB 被破坏或 DB 中数据有错误。

DB 的并发控制是为了避免 DB 中数据有错误或用户读取“脏数据”这两件事的发生。

DB 的完整性是为了保证 DB 中数据是正确的，避免非法的更新。

DB 的安全性也是为了保护 DB，防止不合法的使用，以免数据的泄露、非法更改和破坏。

7.3 自测题

7.3.1 填空题

1．系统目录的功能是存储元数据，元数据主要包括_____的定义。

2．在 DBS 运行时，DBMS 各个子系统要频繁地访问_____，来保证系统正常运行。

3．在应用程序中，事务以 BEGIN TRANSACTION 语句开始，以_____或_____语句结束。

4．事务的原子性是由 DBMS 的_____实现的。

5．事务的一致性是由 DBMS 的_____实现的。

6．事务的隔离性是由 DBMS 的_____实现的。

7．事务的持久性是由 DBMS 的_____实现的。

8. 抽象的事务模型中，事务有五种状态：_____、_____、_____、_____和_____。

9．恢复的基本原则是_____。要使数据库具有可恢复性，在平时要做好两件事：_____和_____。

10．如果对数据库的并发操作不加以控制，则会带来四类问题：_____、_____、_____和_____。

11．锁（lock）描述了数据项的状态，其作用是使_____。

12．事务的执行次序称为_____。

13．判断一个并发调度是否正确，可以用_____概念来解决。

14．封锁能避免错误的发生，但会引起_____问题。

15．S 封锁增加了并发度，但缺点是_____。

16．两段式封锁是可串行化的_____条件。

17．数据库的完整性是指数据的_____、_____和_____。

18．错误数据的输入和输出，称为_____。

19．数据库中数据发生错误，往往是由_____引起的。

20．数据库完整性子系统是根据_____工作的。

21．数据库完整性规则由三部分组成：_____、_____和_____。

22．SQL 中完整性约束有四种：_____，_____，_____和_____。

23．SQL 中全局约束有_____和_____两种。

24．在 SQL 的外键约束中，如果“ON DELETE ……”短语不写时，系统默认是_____方式。

25．用户使用数据库的方式，称为_____。

26．SQL 中的安全性机制，主要有两个：_____和_____。

27．SQL 的授权语句中的关键字 PUBLIC 表示_____。

28．SQL 中“REVOKE GRANT OPTION FOR ……”表示_____。

7.3.2 单项选择题（在备选的答案中选出一个正确答案）

1．事务（transaction）是一个 [ ]

A．程序 B．进程 C．操作序列 D．完整性规则

2．事务对 DB 的修改，应该在数据库中留下痕迹，永不消逝。这个性质称为事务的 [ ]

A．持久化 B．隔离性 C．一致性 D．原子性

3．事务的并发执行不会破坏 DB 的完整性，这个性质称为事务的 [ ]

A．持久化 B．隔离性 C．一致性 D．原子性

4．数据库恢复的重要依据是 [ ]
A．DBA B．DD C．文档 D．事务日志
5．后备副本的主要用途是 [ ]
A．数据转储 B．历史档案 C．故障恢复 D．安全性控制
6．“日志”文件用于保存 [ ]
A．程序运行过程 B．数据操作
C．程序执行结果 D．对数据库的更新操作
7．在 DB 恢复时，对已经 COMMIT 但更新未写入磁盘的事务执行 [ ]
A．REDO 处理 B．UNDO 处理 C．ABORT 处理 D．ROLLBACK 处理
8．在 DB 恢复时，对尚未做完的事务执行 [ ]
A．REDO 处理 B．UNDO 处理 C．ABORT 处理 D．ROLLBACK 处理
9．在 DB 技术中，“脏数据”是指 [ ]
A．未回退的数据 B．未提交的数据
C．回退的数据 D．未提交随后又被撤消的数据
10．如果有 n 个事务串行调度，那么不同的有效调度有 [ ]
A．n^2 B．2^n C．4^n D．n!
11．如果 n 个事务并发调度，那么可能的并发调度数目 [ ]
A．为 n B．为 4^n C．为 n! D．大于 n!
12．事务的执行次序称为 [ ]
A．过程 B．步骤 C．调度 D．优先级
13．在事务依赖图中，如果两个事务的依赖关系形成一个循环，那么就会 [ ]
A．出现活锁现象 B．出现死锁现象
C．事务执行成功 D．事务执行失败
14．“所有事务都是两段式”与“事务的并发调度是可串行化”两者之间关系是 [ ]
A．同时成立与不成立 B．没有必然的联系
C．前者蕴涵后者 D．后者蕴涵前者
15．“断言”是 DBS 采用的 [ ]
A．完整性措施 B．安全性措施 C．恢复措施 D．并发控制措施

7.3.3 简答题

1．试叙述事务的 ACID 性质及其实现者。
2．COMMIT 操作和检查点时的操作有些什么联系？你认为应该如何恰当协调这两种操作才有利于 DB 的恢复？
3．日志文件中记载了哪些内容？
4．试比较并发与并行的区别。
5．试解释 DB 的并发控制与恢复有什么关系？
6．X 封锁与 S 封锁有什么区别？

7.3.4 设计题

设教学数据库中有四个基本表：

系 DEPT（D#，DNAME，MGR#）
其属性分别表示系编号、系名、系主任的教师工号。
教师 T（T#，TNAME，AGE，SEX，SALARY，D#）
其属性分别表示教师工号、姓名、年龄、性别、工资、所在系的编号。
任课 TC（T#，C#，TEXTBOOK）
其属性分别表示教师工号、课程号和所用的教材名。
课程 C（C#，CNAME，D#）
其属性分别表示课程号、课程名和开课系的编号。

下面的题目都是针对这四个基本表进行操作。

1. 用 SQL 的子句或语句定义下列完整性约束：
   ① 编号小于 D8 的系的教师年龄在 20~40 岁之间，其他系的教师年龄在 20~60 岁之间。
   ② 在教师表 T 中出现的 D#值必须在 DEPT 表中出现（用三种形式定义）。
   ③ 每个女教师至少要任教一门课。
   ④ 每个男教师至少要任教两门课。
   ⑤ 每个系教师的平均工资不能低于 2000 元。
   ⑥ 不允许男教师担任“艺术体操”课程。
2. 试对下列操作写出授权语句：
   ① 把对 DEPT 表的插入、修改、删除、查询的权限授给用户 LIU。
   ② 把对 TC、C 表的查询权限转授给全体用户。
   ③ 允许用户 ZHANG 引用 C 表的主键作为新表的外键，并有转让权限。
   ④ 从用户 ZHANG 回收对 C 表主键引用的转授权。

7.4 练习题答案

7.4.1 填空题答案

1．DB 三级结构、两级映象
2．系统目录
3．COMMIT　　ROLLBACK
4．事务管理子系统
5．完整性子系统
6．并发控制子系统
7．恢复管理子系统
8．活动状态　　局部提交状态　失败状态　异常中止状态　提交状态
9．冗余　　转储（备份）　　记“日志”
10．丢失更新问题　读“脏数据”问题　错误求和问题　不可重复读问题
11．并发事务对数据库中数据项的访问能够同步
12．调度
13．可串行化
14．活锁和死锁
15．容易发生死锁
16．充分
17．正确性　　有效性　　相容性
18．垃圾进垃圾出
19．非法的更新
20．完整性规则集
21．触发条件　　约束条件（谓词）　　ELSE 子句
22．主键约束　　外键约束　　属性值约束　　全局约束
23．基于元组的检查子句　　断言
24．RESTRICT
25．权限
26．视图　　授权
27．系统中当前的和未来的全体用户
28．回收转授出去的转让权限

7.4.2 单项选择题答案

1．C	2．A	3．B	4．D	5．C	6．D
7．A	8．B	9．D	10．D	11．D	12．C

13．B　　14．C　　15．A

7.4.3 简答题答案

1．答：事务的 ACID 性质及其实现者见图 7.7。

	内　容	实　现　者
原子性	事务是一个基本工作单位，不可以被分割执行	DBMS 的事务管理子系统
一致性	事务独立执行的结果，应保证 DB 的一致性	1．程序员（正确地编写事务） 2．DBMS 的完整性子系统
隔离性	在多个事务并发执行时，保证执行结果是正确的，如同单用户环境一样	DBMS 的并发控制子系统
持久性	事务对 DB 的更新，应永久地反映在 DB 中，即应保留这个事务执行的痕迹	DBMS 的恢复管理子系统

图 7.7

2．答：在 COMMIT 和检查点技术联合使用时，COMMIT 操作就不一定保证事务对 DB 的修改写到磁盘，而要到检查点时刻才保证写到磁盘。在系统恢复时，那些已经执行了 COMMIT 操作但修改仍留在内存缓冲区的事务需要做恢复工作，利用日志重做（REDO）事务对 DB 的修改。

在事务执行时，应在日志中记下事务的开始标记、结束标志以及事务对 DB 的每一个修改。在系统恢复时，要在日志中检查故障点与最近一个检查点之间，哪些事务执行了 COMMIT 操作（这些事务应重做），哪些事务还未结束（这些事务应撤销）。

3．答：日志文件中记载了事务开始标记、事务结束标记以及事务对 DB 的插入、删除和修改的每一次操作前后的值。

4．答：并发与并行的区别见图 7.8。

	定　义	方　法	对 立 面
并 发 (concurrent)	在某时间段上，有 n 个活动	单处理机（一个 CPU，利用分时方法，实现多个事务同时作）	串行（serial）（事务按顺序先后执行）
并 行 (parallel)	在某时间点上，有 n 个活动	多处理机（并行系统）	顺序（sequential）（事务按顺序先后执行）

图 7.8

5．答：如果采用封锁机制，事务并发操作时有可能产生死锁。为了解除死锁状态，就要抽取某个事务作牺牲品，把它撤消掉，做回退操作，这就属于 DB 的恢复范畴。

6．答：X 锁与 S 锁的区别见图 7.9。

X 锁	S 锁
只允许一个事务独锁数据	允许多个事务并发 S 锁某一数据
获准 X 锁的事务可以修改数据	获准 S 锁的事务只能读数据，但不能修改数据
事务的并发度低	事务的并发度高，但增加了死锁的可能性
X 锁必须保留到事务终点	根据需要，可随时解除 S 锁
解决“丢失更新”问题	解决“读不一致性”问题

图 7.9

7.4.4 设计题答案

1．解：

① 用检查子句定义：

```
CHECK（（D#<'D8' AND AGE BETWEEN 20 AND 40）
       OR（D#>='D8' AND AGE BETWEEN 20 AND 60））；
```

② 第一种形式，在 T 表中，加一个外键子句，同时在 D#定义时注明非空：

```
D# CHAR（4） NOT NULL，
FOREIGN KEY D# REFERENCES DEPT（D#）
```

第二种形式，在 T 表中用 CHECK 子句定义：

```
CHECK（D# IN（SELECT D# FROM DEPT））
```

第三种形式，用断言定义：

```
CREATE ASSERTION ASSE2 CHECK
    （NOT EXISTS
       （SELECT *
        FROM T
        WHERE D# NOT IN
            （SELECT D#
             FROM DEPT）））；
```

③ 用断言定义：

```
CREATE ASSERTION ASSE3 CHECK
    （NOT EXISTS
       （SELECT *
        FROM T
        WHERE SEX='F'
          AND T# NOT IN
              （SELECT T#
               FROM TC）））；
```

④ 用断言定义：

```
CREATE ASSERTION ASSE4 CHECK
    （2<=（SELECT COUNT（C#）
           FROM T，TC
           WHERE SEX='M' AND T.T#=TC.T#
          GROUP BY T.T#））；
```

⑤ 用断言定义：

```
CREATE ASSERTION ASSE5 CHECK
    （2000<=（SELECT AVG（SALARY）
              FROM T
             GROUP BY D#））；
```

⑥ 用断言定义：

```
CREATE ASSERTION ASSE6 CHECK
    （NOT EXISTS
       （SELECT *
        FROM TC
        WHERE T# IN（SELECT T#
                     FROM T
                     WHERE SEX='M'）
          AND C# IN（SELECT C#
                     FROM C
```

WHERE CNAME='艺术体操')));

2. 解：

① GRANT INSERT，UPDATE，DELETE，SELECT ON DEPT TO LIU；

② GRANT SELECT ON TC，C TO PUBLIC；

③ GRANT REFERENCES（C#）ON C TO ZHANG WITH GRANT OPTION；

④ REVOKE GRANT OPTION FOR REFERENCES（C#）ON C FROM ZHANG；

第 8 章 对象数据库系统

8.1 基本内容分析

8.1.1 本章重要概念

（1）新一代 DBS 的两条途径：ORDBS 和 OODBS。

（2）平面关系模型，嵌套关系模型，复合对象模型，引用类型，对象联系图的成分及表示方法，数据的概化/特化，继承性。

（3）OO 的数据类型系统：基本类型，五种复合类型，引用类型。

（4）对象关系模型的定义，两个级别的继承性，引用类型的定义，ORDB 的查询语言，路径表达式，Oracle 中查询的两种技术，嵌套与解除嵌套。

（5）OODBS 的定义，OO 数据模型的五个基本概念，ODMG1.0 标准的 ODL 和 OML，ODMG2.0 标准的数据模型、ODL 和 OQL。

（6）OODB 与 ORDB 的比较。

（7）UML 的类图，用类图表达类、关联、关联类、概化/特化、聚合。

8.1.2 本章的重点篇幅：

（1）教材 P311 的图 8.5。（对象联系图）

（2）OO 的类型系统。（教材 P313～314）

（3）教材 P318 的例 8.7，P321-322 的例 8.10～例 8.13。（ORDB 的定义语言和查询语言）

（4）教材 P339 的例 8.26，P341 的例 8.27。（ODMG2.0 标准的 ODL 和 OQL）

（5）教材 P347 的 8.9 节的 UML 类图。虽然本书对这一节加了星号“*”，但面向对象的概念建模是今后发展的方向。

8.1.3 重点内容分析

本章的核心是数据库技术发展的三个演变过程。

1．数据模型的演变

从关系模型到面向对象模型经历了下面的发展过程：

• 平面关系模型：属性都是基本数据类型。

• 嵌套关系模型：属性可以是基本数据类型，也可以是关系类型，且数据结构可以多次嵌套。

• 复合对象模型：属性可以是基本数据类型，也可以是关系类型，还可以是元组类型，且数据结构可以多次嵌套。

• 面向对象类型：在复合对象模型的基础上，数据结构的嵌套采用引用（指针）方式，并且引入面向对象技术的继承性等概念。

2．查询语言的演变

这里主要是指 SELECT 语句的演变过程：

• RDB 中的 SELECT 语句，由六个子句构成。

• ORDB 中的 SELECT 语句（SQL3 标准），引入了路径表达式、嵌套与解除嵌套等概念。

• OODB 中的 SELECT 语句（ODMG2.0 标准），有了更多的扩充，并与宿主语言语句混合起来，可以表达更为复杂的查询操作。

3．概念建模的演变

这里主要是理解 ER 图的演变历程：

• ER 图：主要用于关系数据库的设计。

• 对象联系图：它是 ER 图的扩充，使之能表达对象之间的引用。

• UML 的类图：这是一种纯 OO 技术的结构。体现了现实世界数据之间面向对象的各种联系方式。

8.2 教材中习题 8 的解答

8. 1 名词解释

• 平面关系模型：满足 INF 性质的关系模型，称为平面关系模型。

• 嵌套关系模型：允许属性值可以是一个关系，而且可以出现多次嵌套，这种关系模型称为嵌套关系模型。

• 复合对象模型：允许属性值可以是关系或元组，而且可以出现多次嵌套，这种关系模型称为复合对象模型。

• 数据的泛化/细化：是对概念之间联系进行抽象的一种方法。当在较低层上抽象表达了与之联系的较高层上抽象的特殊情况时，就称较高层上抽象是较低层上抽象的“泛化”；而较低层上抽象是较高层上抽象的“细化”。

• 对象关系模型：在传统的关系模型基础上，提供元组、数组、集合一类丰富的数据类型以及处理新数据类型操作的能力，并且有继承性和对象标识等面向对象特点，这样形成的数据模型，称为“对象关系数据模型”。

• 类型级继承性：在 ORDB 中，数据类型之间的子类型与超类型的联系，称为“类型级继承性”。

• 表级继承性：在 ORDB 中，表之间的子表与超表的联系，称为“表级继承性”。

• 引用类型：数据类型嵌套时，在具体实现时不是引用对象本身的值，而是引用对象标识符，这种数据类型称为“引用类型”。

• ODMG 组织：致力于开发工业化 OODB 标准的国际组织。

• 对象：对象可以定义为对一组信息及其操作的描述。对现有一组变量、一组消息和一组方法三个部分组成。

• 类：类是相似对象的集合，类中所有对象共享一个公共的定义，而赋予变量的值各不相同。

• 单重继承型：一个子类只继承某一个超类的结构和特性，称为单重继承性。

• 多重继承性：一个子类继承了多个超类的结构和特性，称为多重继承性。

• 对象标识：在 DBS 内识别对象的标志，称为对象标识。对象标识具有位置独立性和数据独立性两个性质。

• 对象包含：不同类的对象之间可能存在着包含关系（即组合关系），包含是一种“一部分”（is part of）联系。

• 类继承层次图：类之间的细化层次图称为类继承层次图。

• 类包含层次图：类之间的组合层次图称为类包含层次图。

• 持久数据：所谓持久数据是指创建这些数据的程序运行终止后依然存在于系统之中的数据。

• 持久对象：创建的对象在程序终止后将被保存，这种对象称为持久对象。

• 持久指针：持久指针是指持久对象的对象标识。持久指针与内存中指针不一样，它在程序执行后及数据重组后仍保持有效。

• 持久化 C++系统：基于 C++语言的持久化扩充的 OODBS，称为持久化 C++系统。

8.2 随着计算机应用领域的扩大，关系数据库系统不能适应哪些应用需要？

答：不能适应多媒体数据、空间数据、时态数据、复合对象、演绎推理等应用的需要。

8.3 什么是对象联系图？图中，椭圆、小圆圈、单箭头（→）、双箭头（→→）、双线箭头（⇒）、双向箭头（←→）这些结构各表示什么含义？

答：描述类型定义间嵌套和递归联系的图称为对象联系图。图中，每个对象可以有若干属性，属性的类型可以是基本数据类型、元组类型或集合类型，而元组或集合是以指针形式（引用类型）实现。

对象联系图中椭圆表示对象类型（相当于实体类型）；小圆圈表示属性是基本数据类型，单箭头（→）表示属性值是单值；双箭头（→→）表示属性值是多值；双线箭头（=>）表示对象类型之间的子类与超类联系（从子类指向超类）；双向箭头（↔）表示两个属性之间值的联系为逆联系。

8.4 OO 的类型系统中有哪些基本数据类型？有哪些复合数据类型？

答：基本数据类型有整型、浮点型、字符、字符串、布尔型和枚举型等五种。

复合类型有行类型、数组类型、列表类型、包类型和集合类型等五种。

8.5 ORDB 中，子表和超表应满足哪两个一致性要求？

答：（1）超表中每个元组最多可以与每个子表中的一个元组对应。

（2）子表中每个元组在超表中恰有一个元组对应。

8.6 试解释类型 T 与引用类型 ref(T)之间的区别，什么情况下你会使用引用类型？

答：在创建表时，可定义某个属性是类型 T 或类型 ref（T），前者是引用对象本身的值，后者是引用对象标识符。而用户使用时，不必关心是何种类型。在对象的值空间很大时，或会出现递归嵌套时，那么应该定义属性的类型为 ref（T）。

8.7 图 8.1 是有关教师（Faculty）、系（Department）和系主任（Director）信息的对象联系图。

（1）试用 ORDB 的定义语言，定义这个数据库。

（2）试用 ORDB 的查询语言，分别写出下列查询的 SELECT 语句：

① 检索精通俄语（Russian）的教师工号和姓名。

② 检索复旦大学出访过瑞士（Switzerland）并且精通日语（Japanese）的系主任。

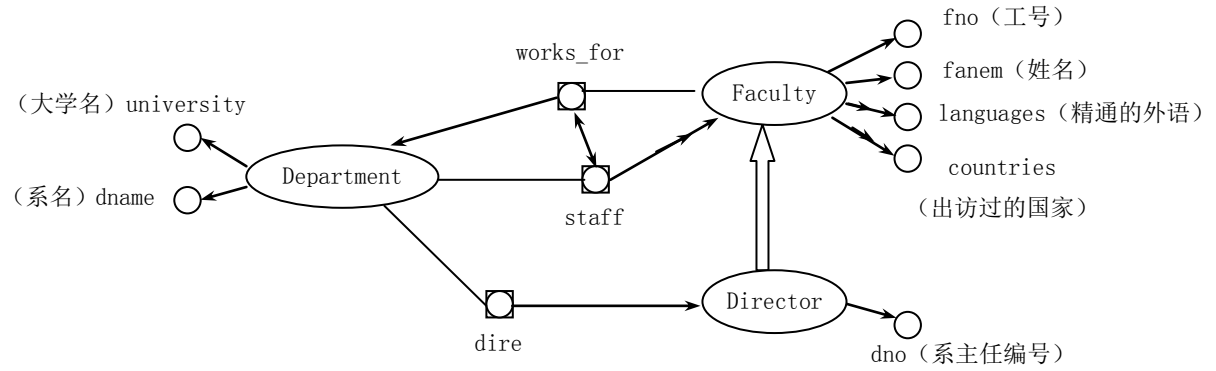


图 8.1　对象联系图

解：（1）CREATE　TYPE　MyString　char　varying;

```
CREATE　TABLE　department(university　MyString,
                          dname　MyString,
                          staff　setoff(ref(faculty)),
                          dire　ref(director));
CREATE　TABLE　faculty(fno　integer,
                       fname　MyString
                       languages　setoff(MyString),
                       countries　setoff(MyString),
                       works_for　ref(department));
CREATE　TABLE　director(dno　integer)
                    Under　faculty;
```

（2）①　SELECT　fno, fname

```
FROM　faculty
WHERE　‘Russian’　in　languages;
```

②　SELECT　D.dno, D.fname

```
FROM　director　as　D
WHERE　D.works_for.university=’Fudan　University’
```

AND ‘Switzerland’ in D.countries
AND ‘Japanese’ in D.languages;

8.8 图 8.2 是有关学生（student）和学习（study）信息的对象联系图。

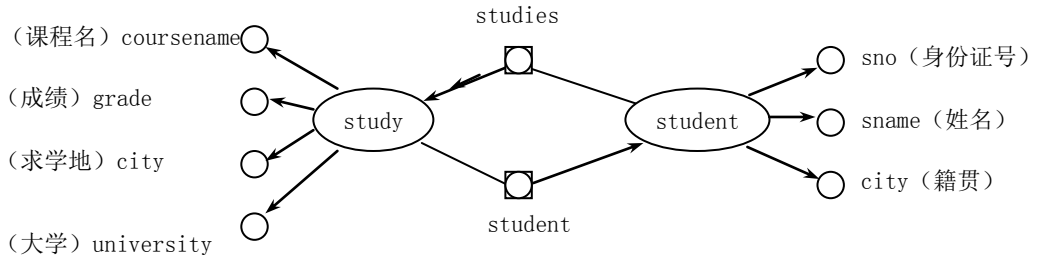


图 8.2 对象联系图

（1）试解释这个对象联系图。
（2）试用 ORDB 的定义语言，定义这个数据库。
（3）试用 ORDB 的查询语言，分别写出下列查询的 SELECT 语句：
① 检索每个学生的学习课程和成绩。
② 检索至少有一门课程的求学地与籍贯在同一城市的学生学号和姓名。

解：（1）对象 student 包含身份证号、姓名、籍贯和学习（studies）等属性，对象 study 包含课程名、成绩、求学地、大学以及学生（student）等属性。对象 student 和 study 之间联系为 1:N。

（2）CREATE TYPE MyString char varying;
CREATE TABLE student(sno integer,
sname MyString,
city MyString,
studies setoff(ref(study)));
CREATE TABLE study(coursename MyString,
grade integer,
city MyString,
university MyString,
student ref(student));

（3）① SELECT A.sname, B.coursename, B.grade
FROM student as A, A.studies as B;
② SELECT A.sno, A.sname
FROM student as A, A.student as B
WHERE A.city=B.city;

8.9 ODMG1.0 标准有哪五个核心概念？

答：ODMG1.0 标准的五个核心概念是：

对象是基本的数据结构；每个对象有一个永久的标识符；对象可以被指定为类型和子类型；对象状态由数据值与联系定义；对象行为由对象操作定义。

8.10 OODBS 应满足哪两个准则？

答：首先 OODBS 是一个 DBS，应具备 DBS 的基本功能，譬如查询语言、事务管理、一致性控制和恢复等。其次 OODBS 是一个面向对象系统，具有持久对象、自定义数据类型、自定义函数、对象封装等必不可少的特点。

8.11 继承性表现了数据间什么联系？试举例说明。

答：继承性表示数据间泛化/细化联系，是一种“is a”联系，表示了类之间的相似性。
譬如人是教师、学生、职工的超类，而教师、学生、职工等都是人的子类。

8.12 持久化语言和嵌入式 SQL 语言有什么区别？

答：有两个主要的区别：

（1）在嵌入式语言中，宿主语言的类型系统与 SQL 的类型系统不同，程序员要负责宿主语言与 DML 之间类型转换。

而持久化程序设计语言的查询语言与宿主语言完全集成在一块，具有相同的类型系统。

（2)使用嵌入式查询语言的编程人员都要负责编写程序，把数据从数据库中取出放到内存中。在更新时，还需编写程序将更新过的数据写回数据库。

8.13 持久化语言中有哪三个基本概念？

答：持久化语言中有对象的持久性、对象标识和指针、持久对象的存储和访问等三个概念。

8.14 试用 ODMG C++ ODL 定义上述第 8.7 题中的数据库。

答：

```
class Department：public Persistent_Object {
public：string university；
        string dname；
        Set<Ref<Faculty>> staff inverse Faculty：：works_for；
        Ref<Director> dire inverse Director：：works_for；
};
class Faculty：public Persistent_Object {
public：int fno；
        string fname；
        Set<string> languages；
        Set<string> countries；
        Ref<Department> works_for inverse Department：：staff；
};
class Director：public Faculty {
public：int dno；
};
```

8.15 试用 ODMG C++ ODL 定义上述第 8.8 题中的数据库。

答：

```
class Student：public Persistent_Object {
public：int sno；
        string sname；
        string city；
        Set<Ref<Study>> studies inverse Study：：student；
};
class Study：public Persistent_Object {
public：string coursename；
        int grade；
        string city；
        string university；
        Ref<Student> student inverse Student：：studies；
};
```

8.16 试用 ODMG ODL 定义上述第 8.7 题中的数据库。

解：

```
interface Department
    （extent Dept）
{attribute string university；
```

```
 attribute string dname;
 relationship Set <Faculty> staff inverse Faculty:: works_for;
 relationship Director dire inverse director:: works_for;
};
interface Faculty
    (extent teachers)
{attribute integer fno;
 attribute string fname;
 set <string> languages;
 set <string> countries;
 relationship Department works_for inverse Department:: staff;
};
interface Director: Faculty
    (extent Directors)
{attribute integer dno;
}
```

8.17 试用 ODMG ODL 定义上述第 8.8 题中的数据库。

解：

```
interface Student
    (extent Students)
{attribute integer sno;
 attribute string sname;
 attribute string city;
 relationship Set <Student> studies inverse Study:: student;
}
interface Study
    (extent St)
{attribute string coursename;
 attribute integer grade;
 attribute string city;
 attribute string university;
 relationship Student student inverse Student:: studies;
}
```

8.18 试用 ODMG OQL 表达教材中 P339 的例 8.26 数据库中的查询：

① 检索教师人数超过 1000 人的大学。要求显示大学校名及校长姓名。显示时，属性名为 univ_name 和 president_name。

② 检索每个大学里的教师平均年龄。

③ 检索上海地区与非上海地区的教师平均年龄。

④ 检索年龄最小的 10 位校长姓名。

⑤ 检索开设 MATHS 课的教师姓名。

⑥ 检索开课课程的教材全采用本校编写的教师工号与姓名。

⑦ 检索至少有 20 位教师年龄超过 80 岁的大学的编号、校名和超过 80 岁的人数。

解：

①
```
SELECT univ_name:U.uname, president_name:U.president.name
FROM University U
WHERE U.num_staff( )>1000;
```

②
```
SELECT U.uno, U.uname,
    avgAGE:AVG(SELECT P.F.age FROM partition P)
```

```
    FROM University U, U.staff F
    GROUP BY U.uno, U.uname;
③ SELECT yes, no,
          avgAGE:AVG(SELECT P.F.age FROM partition P)
    FROM University U, U.staff F

    GROUP BY yes:U.city=' shanghai', no:U.city!='shanghai';

④(SELECT U.president.name
    FROM University U
    ORDER BY U.president.age)[0:9];
⑤ SELECT F.name
    FROM Faculty F, F.teach C

    WHERE C.cname='MATHS';

⑥ SELECT F.fno, F.name
    FROM Faculty F
    WHERE FOR ALL C IN F.teach:
          F.works_for.uno=C.editor.uno;
⑦ SELECT U.uno, U.uname, count(SELECT *
                                FROM partition P, P.staff F
                                WHERE F.age>80)
    FROM university U
    GROUP BY U.uno, U.uname
       HAVING COUNT(SELECT *
                    FROM partition P, P.staff F
                    WHERE F.age>80)>20;
```

8.19 试用 ODMG OQL 的 SELECT 语句表达教材中 P339 的例 8.26 数据库中的查询：

检索上海地区大学中年龄在 25～30 岁之间的教师开设课程的课程名。（试用多种形式表达）

解：

```
SELECT Struct(U.uno, U.uname, set(F.name, set(C.cname)))
FROM university U, U.staff F, F.teach C

WHERE U.city='shanghai' AND F.age BETWEEN 25 AND 30;
```

或

```
SELECT Struct(F.works_for.uno, F.works_for.uname, set(F.name, set(C.cname)))
FROM Faculty F, F.teach C

WHERE F.works_for.city='shanghai' AND F.age BETWEEN 25 AND 30;
```

8.20 OODBS 和 ORDBS 各有什么样的市场目标？

答：ORDBS 的典型应用是涉及到复合数据（包括多媒体数据）的存储和查询。

8.21 关系 DBS、基于持久化语言的 OODBS、ORDBS 三者各有什么长处和劣势？

答：关系 DBS 的数据类型简单，查询语言功能强大，高保护性。其缺点是不能表示嵌套、递归的数据结构。

基于持久化语言的 OODBS 能支持复合数据类型，与程序设计语言集成一体化，高性能。但较难提供对说明性查询的支持。（这在 ODMG97 中已有改观）

ORDBS 能支持复合数据类型，查询语言功能强大，高保护性。其缺点是还不能说是严格意义上的面向对象数据模型。

8.22 OODB 与 ORDB 之间有些什么主要区别?

答：ORDB 是从 SQL 出发，引入复合类型、继承性、引用类型等面向对象概念，很可能是从关系世界通往面向对象世界的一条平坦之路。

OODB 是在面向对象语言（C++）基础上，引入持久数据的概念，能操作 DB，ODMG93 标准不支持 SQL 标准，因此尚难推广使用。ODMG97 标准吸收了 SQL 的 SELECT 语句特色，使 OODB 的实用性得到加强。

8.23 试写一篇 5000 字短文，对 OODB 与 ORDB 作一论述。

答：这篇论文应包括下列要点：

① 关系模型的缺陷。对新的 DBS 的研究

② ORDB 的产生和发展。

③ OODB 的产生和发展。

④ OODB 与 ORDB 的区别。

⑤ 对发展前景瞻望。

8.24 什么是 UML?

答：UML 是 20 世纪 90 年代中期 Booch、Rumbaugh 和 Jacobson 等三位专家源于早先的方法和符号，但并不拘泥于早先的方法和符号，设计了一个标准的建立模型语言。经过修改后，OMG 组织在 1997 年推出 UML 1.0 和 UML 1.1 版，确定 UML 为面向对象开发的行业标准语言，并得到了微软、Oracle、IBM 等大厂商的支持和认证。

8.25 试对“类图”下个定义?

答：类图描述了面向对象模型的静态结构，主要包括三部分内容：类，类的内部结构，类参与的联系。

8.26 试对类图中的下列术语作解释：

• 对象（Object）：对象是一个实体，在应用范围内扮演着意义明确的角色，有状态、行为和身份三重含义。

• 类（Class）：类是“对象类”的简称，类是表示共享公共结构和公共行为的对象集。

• 关联（association）：关联是对类的实例之间联系的命名。

• 关联元数：与关联有关的类的个数，称为关联元数或度数。

• 关联角色（Association role）：关联的端部，也就是与关联相连的类，称为关联角色。

• 重复度（multiplicity）：重复度是指在一个给定的联系中有多少对象参与。

• 关联类（Association class）：在关联本身也有属性或自己的操作时，或者关联也参与其他类的联系时，这种关联称为关联类。

• 鉴别器：在类之间具有概化/特化联系时，用以指出概化基础的属性，称为鉴别器。

• 抽象类（Abstract class）：抽象类是一种没有直接实例，但它的子孙可以有直接实例的类。

• 具体类（concrete class）：具体类是指有直接实例的类。

• 概化：数据的概化 / 特化（generalization / specialization）是对概念之间联系进行抽象的一种方法。当在较低层上抽象表达了与之联系的较高层上抽象的特殊情况时，就称较高层上抽象是较低层上抽象的“概化”,而较低层上抽象是较高层上抽象的“特化”。这种概化联系是一种“是”（is a）的联系。

• 聚合（Aggregation）：聚合是一种表达成分对象和聚合对象之间的 part-of（一部分）联系。

• 复合（Composition）：复合是属于整体对象的一个部分对象，并与整体对象共存亡。

8.27 类图中的重复度与 ER 图中实体的基数有什么异同?

答：重复度类似于 ER 模型中实体基数的概念。但是这是两个相反的概念。

实体基数是指与一个实体有联系的另一端实体数目的最小、最大值，基数应写在这一端实体的边上。

而重复度是指参与关联的这一端对象数目的最小、最大值，重复度应写在这一端类的边上。

8.28　试比较概化、聚合、复合等这三个概念的区别。

答：这三个概念都表达了类图中类之间的联系，但表达的内容不一样。

概化表达了子类与超类之间的“is a”联系。

聚合表达了成分对象和聚合对象之间“is part of”的联系。

复合是较强形式的聚合，此时一部分对象只属于一个整体对象，并与整体对象共存亡。

8.29　试用类图形式来表示 8.7 题的图 8.1 的对象联系图。

解：该对象联系图的类图如图 8.3 所示。

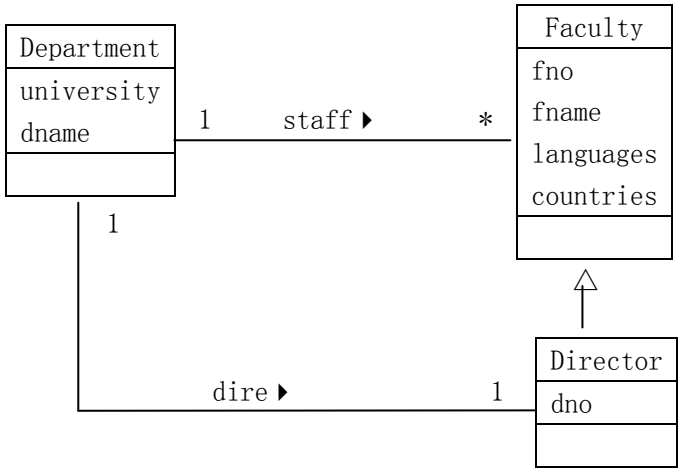


图 8.3 有关教师、系和系主任等信息的类图

8.30 试用类图形式来表示 8.8 题的图 8.2 的对象联系图。

解：解：该对象联系图的类图如图 8.4 所示。

图 8.4 有关学生和学习的类图

8.31　试写一篇 5000 字短文，对 ER 图、对象联系图和类图作一比较。根据你的理解，应该用什么图形和符号来模拟现实世界比较好。

答：这篇短文的要点如下：

① ER 图的基本成分，表达了实体之间的 1:1、1:N、M:N 联系。

② 从 ER 图到对象联系图的发展，引入了“引用类型”，对象之间可以嵌套和递归引用。

③ 从对象联系图到 UML 类图的发展，引入了数据之间的概化、聚合和复合等联系。

④ 尽可能地指出这三种图的优缺点。

8.3 练习题

8.3.1 填空题

1．OODBS 是从_____出发，引入_____技术。

2．ORDBS 是从_____出发，引入_____技术。

3. 关系模型中基本的数据结构层次是_____，并且要求关系模式具有_____性质。传统的关系模型又称为_____模型。

4．在嵌套关系模型中，数据类型可以是基本数据类型，还可以是_____类型。

5．在复合对象模型中，数据类型可以是基本数据类型，还可以是_____类型或_____类型。

6．嵌套关系模型和复合对象模型的一个明显弱点是它们无法表达_____，即类型定义不允许_____。

7．对象联系图是_____的演变，能清楚地表示 OO 数据模型_____之间的联系。

8．面向对象的类型系统中，复合类型有五种：_____，_____，_____，_____和_____。

9．ORDB 中，引用类型用关键字_____表示。

10．ORDB 中，继承性有两种级别：_____和_____。

11．第一个纯面向对象语言是_____，现在最常使用的混合型面向对象语言是_____。

12．OODB 中，对象由三部分组成：_____，_____和_____。

13．（15）面向对象技术中，封装性是一种_____技术，其目的在于将_____和_____分开。

14．类是_____的集合。

15．面向对象模型中类和对象的概念相当于 ER 模型中_____和_____的概念。

16．对象标识是指针一级的概念，是一个强有力的_____。

17．继承性是数据间的泛化/细化联系，是一种_____联系，而对象包含是一种_____联系。

18．UML 类图描述了系统的_____结构，其中包括了类和类之间联系。

19．关联是对_____的命名。

20．类、对象和关联分别相当于 ER 模型中的_____、_____和_____。

21．概化表达了类之间的_____联系，聚合表达了类之间的_____联系。

8.3.2 选择题（在备选答案中选出一个正确的答案）

1．传统的 SQL 技术中，使用“SELECT DISTINCT”方式查询得到的结果，实际上为 [ ]

A．数组　　B．列表　　C．包　　D．集合

2．传统的 SQL 技术中，在 SELECT 语句中使用了 ORDER BY 子句方式查询得到的结果，实际上为 [ ]

A．数组　　B．列表　　C．包　　D．集合

3．在面向对象系统中，不同类型元素的有序集合，称为 [ ]

A．行类型 B．数组类型　　C．列表类型　　D．包类型 E．集合类型

4．在面向对象系统中，同类元素的有序集合（大小已预置），称为 [ ]

A．行类型 B．数组类型　　C．列表类型　　D．包类型 E．集合类型

5．在面向对象系统中，同类元素的有序集合（大小未预置），称为 [ ]

A．行类型 B．数组类型　　C．列表类型　　D．包类型 E．集合类型

6．在 ORDB 中，同类元素的无序集合，并且允许一个成员多次出现，称为 [ ]

A．行类型 B．数组类型　　C．列表类型　　D．包类型 E．集合类型

7．在 ORDB 中，同类元素的无序集合，但每个成员只能出现一次，称为 [ ]

A．行类型 B．数组类型　　C．列表类型　　D．包类型 E．集合类型

8．ORDB 中，引用类型是指嵌套引用时，不是引用对象本身的值，而是引用 [ ]

A．关键码　　B．主键　　C．实体标识符　　D．对象标识符

9．面向对象技术中，封装性是一种 [ ]

A．封装技术　　B．信息隐蔽技术　　C．组合技术　　D．传递技术

10．在 OODB 中，“类”（class）是　　[　]

A．实体的集合　　B．数据类型的集合

C．表的集合　　D．对象的集合

11．在面向对象数据模型中，下列叙述不正确的是　　[　]

A．类相当于 ER 模型中实体类型　　B．类本身也是一个对象

C．类相当于 ER 模型中实体集　　D．类的每个对象也称为类的实例

12．在 OODB 中，对象可以定义为对一组信息及其_____的描述　　[　]

A．操作　　B．存取　　C．传输　　D．继承

13．在 OODB 中，包含其他对象的对象，称为　　[　]

A．强对象　　B．超对象　　C．复合对象　　D．持久对象

14．在 OODB 中，对象标识　　[　]

A．与数据的描述方式有关　　B．与对象的物理存储位置有关

C．与数据的值有关　　D．是指针一级的概念

15．UML 类图中的关联相当于 ER 模型中的　　[　]

A．实体　　B．实体集　　C．联系　　D．属性

8.3.3 简答题

1．什么是嵌套的数据结构？什么是递归的数据结构？递归的数据结构如何实现？

2．与 ER 图相比，对象联系图有哪些修改和扩充？

3．试解释 OODB 中对象、类、类对象三个概念。

4．试详细解释“对象标识”这个概念。

8.3.4 设计题

图 8.5 是关于系（Dept）、学生（Student）、成绩（SC）、课程（Course）和教师（Faculty）信息的对象联系图。

Dept 是有关学校里系信息的对象类型，有六个属性。两个是基本数据类型，系编号（dno）和系名（dname）；单值属性 director 表示有一位教师是系主任；还有三个是多值属性，staff 表示系里有若干教师，mass 表示系里有若干学生，set_up 表示系里开设了若干门课程。

Student 是有关学生信息的对象类型，有六个属性。四个是基本数据类型，学生的学号（sno）、姓名（sname）、年龄（age）和性别（sex）；单值属性 study_in 表示学生属于某个系；多值属性 study 表示该学生的学习成绩。

SC 是有关成绩信息的对象类型，有三个属性。成绩 grade 是基本数据类型；单值属性 student 表示该成绩是属于何学生；单值属性 course 表示该成绩属于何门课程。

Course 是有关课程信息的对象类型，有五个属性。两个是基本数据类型，课程号（cno）和课程名（cname）；有两个是单值属性，teacher 表示课程的任课老师，founder 表示课程由何系设置的；多值属性 learn 表示选修这门课程的学生成绩。

Faculty 是有关教师信息的对象类型，有五个属性。三个是基本数据类型，教师工号（fno）、姓名（fname）和工资（salary）；单值属性 works_for 表示教师服务的系；多值属性 teach 表示教师开设了若干门课程。

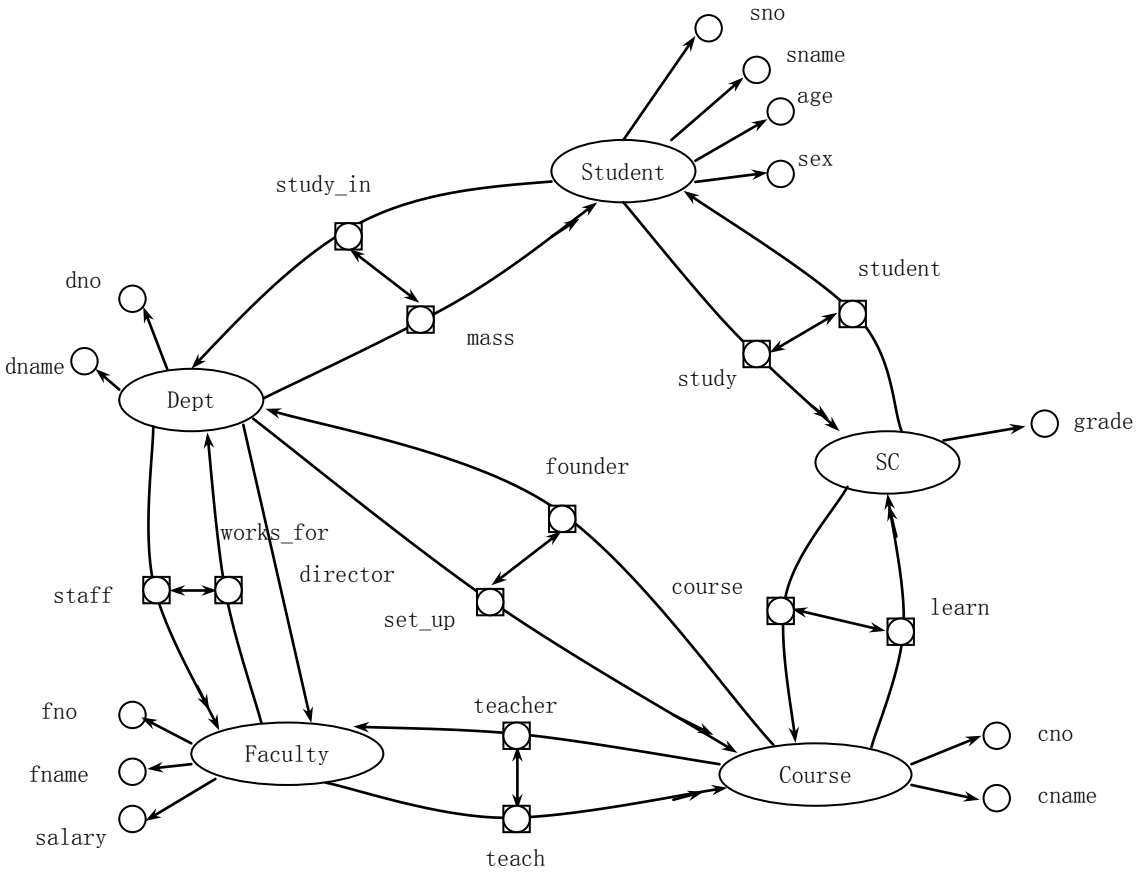


图 8.5

下面题目都是针对图 8.5 设置的。

1．试用 ORDB 的定义语言，定义图 8.5 的数据库。

2．试用 ORDB 的查询语言，分别写出下列查询的 SELECT 语句：

① 检索“计算机系”每一个学生的学习成绩，要求显示（sno，sname，cname，grade）。

②检索选修本系课程的学生选课情况，要求显示（dno，dname，sno，sname，cno，cname）。

③ 检索选修本系课程、并由本系教师任教的课程的学生选课情况，要求显示（dno，dname，sno，sname，cno，cname，fno，fname）。

④检索“计算机系”每一个男学生选修课程的门数，要求显示（sno，sname，course_num）。

3．试用 ODMG C++ ODL（伪代码）定义图 8.5 的数据库。

4．试用 ODMG ODL 定义图 8.5 的数据库。

5．试用类图形式来表示图 8.5 的对象联系图。

8.4 练习题答案

8.4.1 填空题答案

1．OOPL　　DB

2．传统的关系 DB 技术（或 SQL 语言）　　OO

3．关系——元组——属性　　1NF　　平面关系

4．关系（或集合）

5．关系（或集合）　　元组（或结构）

6．递归的结构　　递归

7．ER 图　　数据

8．行类型　　数组类型　　列表类型　　包类型　　集合类型

9．ref

10．类型级　　表级

11．smalltalk　　C++
12．一组变量　　一组消息　　一组方法
13．信息隐蔽　　使用者　　设计者
14．类似对象
15．实体集　　实体
16．数据操纵原语
17．是一个（is　a）　　是一部分（is　part　of）
18．静态
19．类的实例之间联系
20．实体集　　实体　　联系
21．is a　　is part of

8.4.2 选择题答案

1．D　　2．B　　3．A　　4．B　　5．C　　6．D　　7．E
8．D　　9．B　　10．D　　11．A　　12．A　　13．C　　14．D
15．C

8.4.3 简答题答案

1．答：嵌套的数据结构是指属性的类型可以是关系类型或元组类型。递归的数据结构是指递归的嵌套结构，譬如数据 A 中嵌有数据 B，而数据 B 中嵌有数据 A。递归的结构要用“指针”概念才能实现，以避免无穷嵌套问题。

2．答：与 ER 图相比，对象联系图从以下几个方面进行了修改和扩充：

①实体类型改称为“对象类型”，用椭圆表示。

②小圆圈表示属性，并且是基本数据类型。对象类型与属性之间联系可以是单箭头（表示单值）或双箭头（表示多值）。

③对象类型之间的连线表示指针方式的引用，连线也可以是单箭头（表示单值，即元组）或双箭头（表示多值，即集合）。

④子类与超类的联系用双线箭头（泛化边）表示。

3．答：在 OODB 中，需分清对象、类、类对象三个概念：

① 对象（object）是对客观世界中实体的进一步抽象。对象可以定义为对一组信息及其操作的描述。

② 类（class）是类似对象的集合。

面向对象模型中类和对象的概念相当于 ER 模型中实体集和实体的概念。

③ 类本身也当作一个对象，称为类对象（class object）。

4．答：面向对象中的“对象标识”（oid）与程序设计语言中的地址、关系 DB 中的主键概念都不一样。

对象标识与对象的物理存储位置无关，也与数据的描述方式、值无关。对象标识是指针一级的概念，是一个强有力的数据操纵原语，也是复合对象、引用类型操纵的基础。

8.4.4 设计题答案

1．解：对图 8.5 的对象联系图，用 ORDB 的定义语言，定义如下：

```
CREATE　TYPE　MyString　char　varying;
CREATE　TABLE　dept（dno　　MyString，
                    dname　MyString，
                    mass　　setof（ref（student）），
                    set_up　setof（ref（course）），
                    director　ref（faculty），
                    staff　　setof（ref（faculty）））;
CREATE　TABLE　student（sno　　MyString，
```

```
                        sname       MyString，
                        age         integer，
                        sex         MyString，
                        study_in    ref（dept），
                        study       setof（ref（sc））；
CREATE  TABLE  sc（grade        integer，
                  student       ref（student），
                  course        ref（course））；
CREATE  TABLE  course（cno          MyString，
                        cname       MyString，
                        founder     ref（dept），
                        teacher     ref（faculty））；
CREATE  TABLE  faculty（fno         MyString，
                        fname     MyString，
                        salary    integer，
                        works_for ref（dept），
                        teach     setof（ref（course））；
```

2．解：① SELECT A.sno，A.sname，B.course.cname，B.grade
FROM student as A，A.study as B
WHERE A.study_in.dname='计算机系'；

或者 SELECT A.sno，A.sname，B.course.cname，B.grade
FROM dept as D，D.mass as A，A.study as B
WHERE D.dname='计算机系'；

② SELECT A.study_in.dno，A.study_in.dname，A.sno，A.sname，B.course.cno，B.course.cname
FROM student as A，A.study as B
WHERE A.study_in.dno=B.course.founder.dno；

或者 SELECT D.dno，D.dname，A.sno，A.sname，B.course.cno，B.course.cname
FROM dept as D，D.mass as A，A.study as B
WHERE D.dno=B.course.founder.dno；

③ SELECT A.study_in.dno，A.study_in.dname，A.sno，A.sname，B.course.cno，B.course.cname，B.course.teacher.fno，B.course.teacher.fname
FROM student as A，A.study as B
WHERE A.study_in.dno=B.course.founder.dno
AND A.study_in.dno=B.course.teach.works_for.dno；

④ SELECT A.sno，A.sname，COUNT（SELECT ＊
FROM A.study）
FROM student as A
WHERE A.sex='M'；

3．解：对图 8.5 的对象联系图，用 ODMG C++ ODL（伪代码）定义如下：

```
class  Dept：public  Persistent_Object{
public：string  dno；
        string  dname；
        Set〈Ref〈Student〉〉mass  inverse  Student::study_in；
        Set〈Ref〈Course〉〉set_up  inverse  Course::founder；
```

```
        Ref〈Faculty〉director;
        Set〈Ref〈Faculty〉〉staff  inverse  Faculty::works_for;
};
class  Student：public  Persistent_Object{
public：string  sno;
        string  sname;
        int  age;
        string  sex;
        Ref〈Dept〉study_in  inverse  Dept::mass;
        Set〈Ref〈SC〉〉study  inverse  SC::student;
};
class  SC：public  Persistent_Object{
public：int  grade;
        Ref〈Student〉student  inverse  Student::study;
        Ref〈Course〉〉course  inverse  Course::learn;
};
class  Course：public  Persistent_Object{
public：string  cno;
        string  cname;
        Ref〈Dept〉founder  inverse  Dept::set_up;
        Ref〈Faculty〉teacher  inverse  Faculty::teach;
};
class  Faculty：public  Persistent_Object{
private：int  salary;
public：string  fno;
        string  fname;
        Ref〈Dept〉works_for  inverse  Dept::staff;
        Set〈Ref〈Course〉〉teach  inverse  Course::teacher;
};
```

4．解：对图 8.5 的对象联系图，用 ODMG ODL 定义如下：

```
interface Dept
    (extent Depts)
{attribute string dno;
 attribute string dname;
 relationship Set <Student> mass inverse Student:: study_in;
 relationship Set <Course> set_up inverse Course:: founder;
  relationship Faculty director;
  relationship Set <Faculty> staff inverse Faculty:: works_for;
}
interface Student
    (extent Students)
{attribute string sno;
 attribute string sname;
 attribute integer age;
 attribute string sex;
```

```
 relationship Dept study_in inverse Dept:: mass;
 relationship Set<SC> study inverse SC::student;
}
interface SC
    (extent SCs)
{attribute integer grade;
 relationship Student student inverse Student:: study;
 relationship Course course inverse Course:: learn;
}
interface Course
    (extent Courses)
{attribute string cno;
 attribute string cname;
 relationship Set<SC> learn inverse SC::course;
 relationship Dept founder inverse Dept:: set_up;
 relationship Faculty teacher inverse Faculty:: teach;
};
interface Faculty
{attribute string fno;
 attribute string fname;
 relationship Dept works_for inverse Dept:: staff;
 relationship Set <Course> teach inverse Course:: teacher;
};
```

5. 解：图 8.5 的对象联系图的类图如图 8.6 所示。

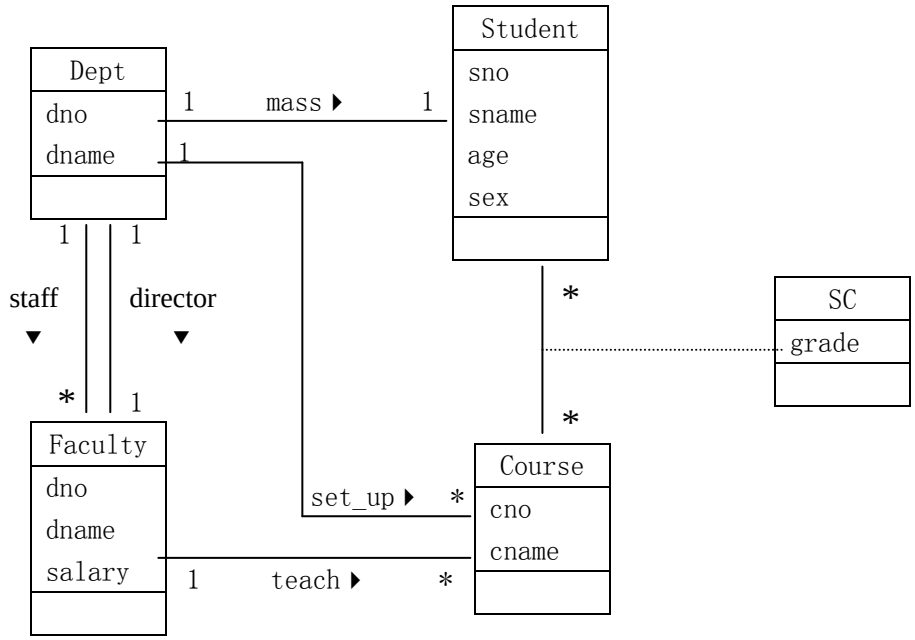


图 8.6 关于系、学生、成绩、课程和教师信息的类图

第 9 章 分布式数据库系统

9.1 基本内容分析

9.1.1 本章重要概念

（1）分布计算的三种形式：处理分布，数据分布，功能分布。

（2）C/S 系统，工作模式，技术特征，体系结构，两层、三层、多层 C/S 结构。

（3）DDBS 的定义、特点、优点、缺点和分类；分布式数据存储的两种形式（分片和分配）。

（4）DDB 的体系结构：六层模式，分布透明性的三个层次，DDBS 的组成，DDBMS 的功能和组成。

（5）分布式查询处理的查询代价，基于半联接的优化策略，基于联接的优化策略。

（6）分布式数据库的并发控制和恢复中出现的问题，以及处理机制。

9.1.2 本章的重点篇幅

（1）两层、三层、多层 C/S 结构。（教材 P365-367）

（2）分布式数据存储：分片和分配。（教材 P375-377）

（3）DDB 的体系结构。（教材 P378 的图 9.10，P381 的图 9.12）

（4）基于半联接的执行示意图。（教材 P389 的图 9.17）

9.2 教材中习题 9 的解答

9. 1 名词解释

• 集中计算：单点数据和单点处理的方式称为集中计算。

• 分布计算：随着计算机网络技术的发展，突破集中计算框架，DBMS 的运行环境逐渐从单机扩展到网络，对数据的处理从集中式走向分布式、从封闭式走向开放式。这种计算环境称为分布计算。

• 处理分布：指系统中处理是分布的，数据是集中的这种情况。

• 数据分布：指系统中数据是分布的，但逻辑上是一个整体这种情况。

• 功能分布：将计算机功能分布在不同计算机上执行，譬如把 DBMS 功能放在服务器上执行，把应用处理功能放在客户机上执行。

• 服务器位置透明性：指 C/S 系统向客户提供服务器位置透明性服务，用户不必知道服务器的位置，就可以请求服务器的服务。

• 集中式 DBS：所有工作都由一台计算机完成，这种 DBS 称为集中式 DBS。

• DDBS：是物理上分散逻辑上集中的 DBS，每一场地既能完成局部应用又能完成全局应用，这种系统称为 DDBS。

• DDB：DDB 是计算机网络环境中各场地上 DB 的逻辑集合。

• DDBMS：DDBMS 是 DDBS 中的一组软件，它负责管理分布环境下逻辑集成数据的存取、一致性和完备性。同时，由于数据的分布性，在管理机制上还必须具有计算机网络通信协议的分布管理特性。

• 分布透明性：是指用户不必关心数据的逻辑分片，不必关心数据物理位置分配的细节，也不必关心各个场地上 DB 的数据模型是哪种类型，可以像集中式 DB 一样来操作物理上分布的 DB。

• 数据分片：把全局概念模式中的全局关系划分成若干不相交部分的过程，称为数据分片。

• 水平分片：对全局关系执行选择操作得到的片段，称为水平分片。

• 垂直分片：对全局关系执行投影操作得到的片段，称为垂直分片。

• 数据分配：是指片段在场地上的分配策略。

• 分片透明性：分片透明性位于全局概念模式与分片模式之间。分片透明性是指用户或应用程序只对全局关系进行操作而不必考虑数据的分片。

• 位置透明性：位于分片模式和分配模式之间。位置透明性是指用户或应用程序应当了解分片情况，但不必了解片段的存储场地。

• 局部数据模型透明性：位于分配模式与局部概念模式之间。指用户或应用程序要了解分片及各片段存储的场地，但不必了解场地上使用的是何种数据模型。

9.2 C／S 结构的基本原则是什么？客户机和服务器的任务各是什么？

答：C/S 结构的基本原则是将计算机应用任务分解成多个子任务，由多台计算机分工完成，即“功能分布”原则。

客户机完成数据处理、数据表示、用户接口等功能。

服务器完成 DBMS 的核心功能。

9.3 一个典型的应用程序有哪四部分组成？在 C／S 结构的 DBS 中是如何实现的？

答：一个典型的应用程序有四部分组成：

用户界面的显示逻辑，应用逻辑，事务逻辑，数据管理。

在 C/S 环境下，通常把前两部分驻留在客户机上，而把后两部分驻留在服务器上。

9.4 C／S 系统有哪些主要的技术特征？

答：C/S 系统主要有下面六个技术特征：

按功能划分；共享资源；不对称协议；定位透明性；基于消息的交换；可扩展性。

9.5 C／S 系统的体系结构有几部分组成？试作必要的解释。

答：C/S 系统的体系结构由客户机、服务器和中间件等三大部分构成。（解释略）

9.6 试对 C／S 结构的两层模型、三层模型、多层模型作详细的解释。从 C／S 的结构看，其发展趋势如何？

答：两层 C/S 结构的引出主要是为了减轻集中式 DBS 主机的负担，把计算机功能分布在不同计算机上。

三层 C/S 结构的引出主要是为了减轻客户机的负担，从两层 C/S 的客户机和服务器中各抽出一部分功能组成应用服务器。

多层 C/S 结构的引出是通过引入中间层组件，扩大了两层 C/S 结构。

C/S 结构的发展趋势是：客户机越来越瘦，服务器品种越来越多。使得 C/S 结构容易组装、扩展。

9.7 网络服务器有哪几类？

答：网络服务器分成 DB 服务器、文件服务器、事务服务器、文档服务器、Web 服务器、电子邮件服务器、各种应用服务器等。

9.8 试叙述数据从集中存储、分散存储到分布存储的演变过程。

答：集中式 DBS 的数据属于集中存储方式；把数据库分成多个，建立在多台计算机上，但相互独立，这种分散式系统的数据属于分散存储；把分散在各地的 DBS 通过网络通信联接起来，这种分布式 DBS 的数据属于分布存储，兼有集中式和分散式的优点。

9.9 与集中式 DBS、分散式 DBS 相比，DDBS 的区别在哪里？

答：与集中式 DBS 的集中存储相比，分布式 DBS 的数据具有“分布性”特点：数据不是存储在一个场地，而是分布存储在各个场地。

与分散式 DBS 的分散存储相比，分布式 DBS 的数据具有“逻辑整体性”特点。

9.10 DDBS 有哪些基本特点？还可以导出哪些特点？

答：DDBS 有四个基本特点：物理分布性，逻辑整体性，场地自治性，场地之间协作性。

由此还可导出其他四个特点：数据独立性，集中与自治相结合的控制机制，适当增加数据冗余度，事务管理的分布性。

9.11 在 DDBS 中为什么需要适当增加数据冗余度？

答：在 DDBS 中希望通过冗余数据提高系统的可靠性、可用性和改善系统性能。

9.12 DDBS 有哪些优点和缺点？

答：与集中式 DBS 相比，DDBS 有六个优点：灵活的体系结构，分布式的管理和控制机构，

经济性能优越，系统可靠性高可用性好，局部应用的响应速度快，可扩展性好。

缺点有三个：花在通信部分开销较大，复杂的存取结构在分布式系统中不一定有效，数据的安全性保密性较难处理。

9.13 试解释下列术语：同构同质型 DDBS，同构异质型 DDBS，异构型 DDBS。

答：同构同质型 DDBS：系统中各个场地都采用同一类型的数据模型，并且是同一型号的 DBMS。

同构异质型 DDBS：系统中各个场地都采用同一类型的数据模型，但 DBMS 的型号可不同。

异构型 DDBS：系统中各个场地的数据模型是不同的类型。

9.14 DDB 中，数据分片有哪些策略？定义分片时必须遵守那些规则？

答：数据分片有水平分片、垂直分片、导出分片和混合分片等四种方式。

数据分片时必须遵守三条规则：完备性条件，可重构条件，不相交条件。

9.15 全局关系与片段之间映像只能是一对多，不可以是多对多，为什么？

答：在 DDB 的体系结构中，往上方向是越来越“逻辑”，往下方向是越来越“物理”。据此可看出，全局关系在上方，片段在下方，因此每个片段只能来自一个全局关系。如果来自多个全局关系的数据，那就不是片段了，而是位于全局关系上方的全局视图（全局外模式）了。因而全局关系与片段之间的映象只能是一对多。

9.16 DDB 中，数据分配有哪些策略？分配策略的评估因素有哪几个？

答：数据分配有集中式、分割式、全复制式和混合式等四种分配策略。

分配策略的评估因素有四个：存储代价，可靠性，检索代价和更新代价。

9.17 试叙述 DDB 的六层模式结构的主要成分。

答：DDB 的六层模式结构的主要成分是：六层模式、五级映像和五级独立性（透明性）。（详细解释略）

9.18 DDB 的六层模式结构是一种通用的概念结构，它有哪些显著的特征？

答：有三个特征：数据分布独立性，数据冗余的显式控制，局部 DBMS 的独立性。

9.19 DDB 的六层模式结构之间的五级映象，各体现什么独立性（或透明性）？

答：五级映像体现五个独立性（透明性），自上而下是：

逻辑独立性，分片透明性，位置透明性，局部数据模型透明性，物理独立性。

9.20 DDBMS 主要有哪些功能？DDBMS 应包括哪些基本功能模块？

答：DDBMS 的功能有五点：接受并处理用户请求，访问网络数据字典，分布式处理，通信接口功能，异构型处理。

DDBMS 应包括以下四个基本功能模块：查询处理模块，完整性处理模块，调度处理模块，可靠性处理模块。

9.21 分布式系统中影响查询的主要因素是什么？

答：主要因素是网络中数据的传输量。

9.22 基于半联接的优化策略的基本原理是什么？

答：不参与联接的数据或无用的数据不必在网络中来回传输。

9.23 什么是“半联接程序”？如何执行？

答：用半联接方法来计算自然联接的方法，称为“半联接程序”。

具体步骤为：$R \bowtie S == (R \bowtie \pi_B(S)) \bowtie S$　　（此处 B 为 R 和 S 的公共属性）

$== (R \ltimes S) \bowtie S$

（如何执行见教材 P389）

9.24 与集中式 DBMS 比较，DDBMS 环境中在并发控制和恢复方面遇到哪些新问题？

答：与集中式 DBMS 比较，DDBMS 环境中在并发控制和恢复方面会遇到以下五个问题：

① 数据项的多拷贝之间的一致性问题。
② 在单个场地故障恢复时，局部数据库的数据应和其他场地的同步问题。
③ 通信网络的故障处理能力问题。
④ 分布式提交的实现问题。
⑤ 分时式死锁的处理问题。

9.25 试解释分布式并发控制中使用的名词：

• 副本(拷贝)：DDB 中，一个数据项可以复制存放在多个场地，每个场地的数据称为副本或拷贝。

• 识别拷贝：一个数据项可以在多个场地有副本，系统为每个数据项指定一个特定的拷贝作为该数据项的识别拷贝。对该数据项的封锁应与识别拷贝相联系，并且所有的封锁和解锁请求都被传送到包含那个拷贝的场地上。

• 主场地：数据库中所有的识别拷贝都被保留在同一个场地上，该场地称为主场地。

• 备份场地：在主场点发生故障时，将接管它而成为主场地的场地称为“备份场地”。

• 主拷贝：各种数据项的识别拷贝可以存储在不同的场地上，这种识别拷贝称为主拷贝。

• 协调者场地：存储数据项识别拷贝的场地，称为该数据项的协调者场地。

9.26 试对分布式并发控制中的主场地方法和主拷贝方法作一比较。使用备份场地对它们有什么影响？

答：分布式并发控制中的主场地方法和主拷贝方法的比较以及有备份场地时对它们的影响如图 9.1 所示。

	主场地技术	主拷贝技术
优点	是集中式方案的简单扩充， 不太复杂。	一个场地的故障只会影响本场地作为主拷贝场地的那些数据项的事务。
缺点	① 主场地是瓶颈口，超负荷运行。 ② 主场地的故障会使系统瘫痪。	实现和管理较复杂。
有备份场地时的影响	能克服上述第②个缺点，简化了恢复的过程，但会使系统运行速度变慢。	提高系统的可靠性和可用性。

图 9.1

9.27 在分布式数据库中，什么时候要使用投票和选举方法？

答：在分布式并发控制中，如果不存在备份场地，或主场地和备份场地都有故障时，就要用到选举方法产生一个备份场地。

在系统中如果不采用识别拷贝技术，那么并发控制就要采用投票方法来决定封锁是成功还是失败。

9.3 自测题

9.3.1 填空题

1．C/S 结构的基本原则是_____原则。

2．C/S 结构中，客户端完成_____功能，服务器端完成_____功能。

3．DDBS 逐渐向 C/S 模式发展。单服务器的结构本质上还是_____系统。只有在网络中有多个 DB 服务器时，并可协调工作，为众多客户机服务时，才称得上是_____系统。

4．C/S 环境中，一个典型的应用程序可分解成四个组成部分：_____、_____、_____和_____。

5．两层 C/S 结构克服了_____问题，三层 C/S 结构克服了_____问题。

6．C/S 结构的发展趋势是：客户机越来越_____，服务器越来越_____。

7．DDBS 具有如下四个基本特点：_____，_____，_____和_____。
8．DDB 的数据分片有_____，_____和_____等三种分片方式。
9．在定义数据分片时，必须遵守三个条件：_____，_____和_____。
10． DDB 的数据分配有_____，_____，_____和_____四种分配策略。
11．DDBS 的体系结构自上而下有六个层次：_____，_____，_____，_____，_____和_____。
12．DDBS 的分片透明性位于_____和_____之间。
13．DDBS 的位置透明性位于_____和_____之间。
14．DDBS 中透明性层次越高，应用程序的编写越_____。
15．基于半联接的查询优化策略的主要思想是_____。

9.3.2 单项选择题（在备选的答案中选出一个正确答案）

1．C/S 体系结构的关键在于 [ ]
A．功能的分布 B．客户机的分布
C．服务器的分布 D．数据的分布

2．C/S 系统中客户机的功能是实现 [ ]
A．前端处理和事务处理 B．事务处理和用户界面
C．前端处理和用户界面 D．事务处理和数据访问控制

3．C/S 系统中服务器的功能是实现 [ ]
A．前端处理和事务处理 B．事务处理和用户界面
C．前端处理和用户界面 D．事务处理和数据访问控制

4．如果各个场地的数据模型是不同的类型（层次型或关系型），那么这种 DDBS 是 [ ]
A．同构型 B．异构型 C．同质型 D．异质型

5．DDBS 中的“数据分片”是指 [ ]
A．对磁盘的分片 B．对全局关系的分片
C．对内存的分片 D．对网络结点的分片

6．DDBS 中的“数据分配”是指在计算机网络各场地上的 [ ]
A．对磁盘的分配策略 B．对数据的分配策略
C．对内存的分配策略 D．对网络资源的分配策略

7．DDBS 的分片模式和分配模式均是 [ ]
A．全局的 B．局部的 C．集中的 D．分布的

8．在 DDBS 中，必须把全局关系映射到片段中。这个性质称为 [ ]
A．映射条件 B．完备性条件 C．重构条件 D．不相交条件

9．在 DDBS 中，必须从分片能通过操作得到全局关系。这个性质称为 [ ]
A．映射条件 B．完备性条件 C．重构条件 D．不相交条件

10．在 DDBS 中，要求一个全局关系被分片后互不重迭（主键除外）。这个性质称为[ ]
A．映射条件 B．完备性条件 C．重构条件 D．不相交条件

11．DDBS 的体系结构是 [ ]
A．分布的 B．集中的 C．全局的 D．分层的

12．DDBS 的“分片透明性”位于 [ ]
A．全局外模式与全局概念模式之间 B．全局概念模式与分片模式之间
C．分片模式与分配模式之间 D．分配模式与局部概念模式之间

13．DDBS 的“位置透明性”位于 [ ]
A．全局外模式与全局概念模式之间 B．全局概念模式与分片模式之间
C．分片模式与分配模式之间 D．分配模式与局部概念模式之间

14．DDBS 的“局部数据模型透明性”位于 [ ]
A．全局外模式与全局概念模式之间 B．全局概念模式与分片模式之间

C．分片模式与分配模式之间　　D．分配模式与局部概念模式之间

15．DDBS 中“分布透明性”可以归入　　[　]

A．逻辑独立性　　B．物理独立性　　C．场地独立性　　D．网络独立性

16．DDBS 中，透明性层次越高　　[　]

A．网络结构越简单　　B．网络结构越复杂

C．应用程序编写越简单　　D．应用程序编写越复杂

17．关系代数的半联接操作由下列操作组合而成：　　[　]

A．投影和选择　　B．联接和选择

C．联接和投影　　D．自然联接和投影

9.3.3 简答题

1．C/S 系统的基本原理是什么？有什么重要意义。

2．集中式 DBS 中和 DDBS 中影响查询的主要因素各是什么？

3．设有关系 R 和 S：

R	A	B	C
	1	2	3
	4	5	6
	8	8	9
	3	5	7

S	B	C	D
	2	3	4
	5	7	8
	8	6	4
	2	3	8

试计算下列表达式的值：

（1）$R \bowtie S$　　（2）$R \ltimes S$　　（3）$S \ltimes R$

（4）$R \underset{3=2}{\bowtie} S$　　（5）$R \underset{3=2}{\ltimes} S$　　（6）$S \underset{2=3}{\ltimes} R$

（7）$R \underset{1=3}{\bowtie} S$　　（8）$R \underset{1=3}{\ltimes} S$　　（9）$S \underset{3=1}{\ltimes} R$

4．设关系 R（A，B，C）在场地 1，关系 S（C，D，E）在场地 2，现欲在场地 2 得到 $R \bowtie S$ 的操作结果。

（1）用联接的方法，如何执行上述操作。

（2）用半联接的方法，如何执行上述操作。（需写出详细的操作式子）

9.4 练习题答案

9.4.1 填空题答案

1．功能分布

2．数据处理、数据表示和用户接口　　DBMS 的核心功能

3．集中式 DB　　分布式 DB

4．用户界面的显示逻辑　　应用逻辑　　事务逻辑　　数据管理

5．集中式主机的瓶颈口　　客户机

6．瘦　　品种繁多

7．物理分布性　　逻辑整体性　　场地自治性　　场地之间协作性

8．水平分片　　垂直分片　　导出分片　　混合分片

9．完备性条件　　重构条件　　不相交条件

10．集中式　　分割式　　全复制式　　混合式

11．全局外模式　　全局概念模式　　分片模式　　分配模式　　局部概念模式　　局部内模式

12．全局概念模式　　分片模式

13．分片模式　　分配模式

14．简单

15．不参与联接的值或无用的值不必在网络中来回传输

9.4.2 单项选择题答案

1．A	2．C	3．D	4．B	5．B	6．B	
7．A	8．B	9．C	10．D	11．D	12．B	13．C
14．D	15．B	16．C	17．D			

9.4.3 简答题答案

1．答：C/S 系统的基本原理是“功能的分布”，将计算机应用任务分解成多个子任务，由多台计算机分工完成。其重要意义是减轻了集中式系统中主机的负担（瓶颈口现象）。

2．答：在集中式系统中，影响查询的主要因素是对磁盘的访问次数。而在分布式系统中，影响查询的主要因素是通过网络传递信息的次数和传送的数据量。

3．解：

（1）$R \bowtie S$

A	B	C	D
1	2	3	4
1	2	3	8
3	5	7	8

（2）$R \ltimes S$

A	B	C
1	2	3
3	5	7

（3）$S \ltimes R$

B	C	D
2	3	4
2	3	8
5	7	8

（4）$R \overset{3=2}{\bowtie} S$

A	R.B	R.C	S.B	S.C	D
1	2	3	2	3	4
1	2	3	2	3	8
4	5	6	8	6	4
3	5	7	2	3	8

（5）$R \overset{3=2}{\ltimes} S$

A	B	C
1	2	3
4	5	6
3	5	7

（6）$S \overset{2=3}{\ltimes} R$

B	C	D
2	3	4
2	3	8
8	6	4

（7）$R \overset{1=3}{\bowtie} S$

A	R.B	R.C	S.B	S.C	D
4	5	6	2	3	4
4	5	6	8	6	4
8	8	9	5	7	8
8	8	9	2	3	8

（8）$R \overset{1=3}{\ltimes} S$

A	B	C
4	5	6
8	8	9

（9）$S \overset{3=1}{\ltimes} R$

B	C	D
2	3	4
8	6	4
5	7	8
2	3	8

4．解：

（1）用联接的方法执行，就是直接把关系 R 从场地 1 传输到场地 2，在场地 2 执行自然联接。（见图 9.2）

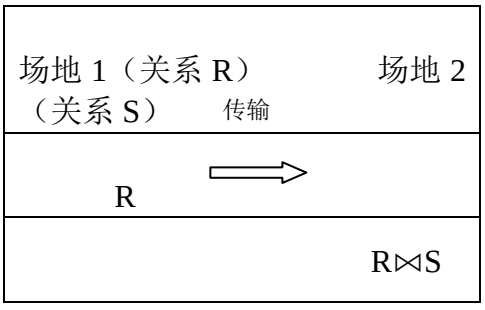


图 9.2　联接的执行示意图

（2）用半联接方法执行的过程如下（见图 9.3）：

① 在场地 2，求 π_C（S）的值；

② 把 π_C（S）的值从场地 2 传输到场地 1；

③ 在场地 1 执行 R⋈ π_C（S）操作；

④ 把（R⋈ π_C（S））的值从场地 1 传输到场地 2；

⑤ 在场地 2 执行（R⋈ π_C（S））⋈S 操作，即求得 R⋈S 的值。

即 R⋈S=（R⋈ π_C（S））⋈S

=（R ⋉S）⋈S

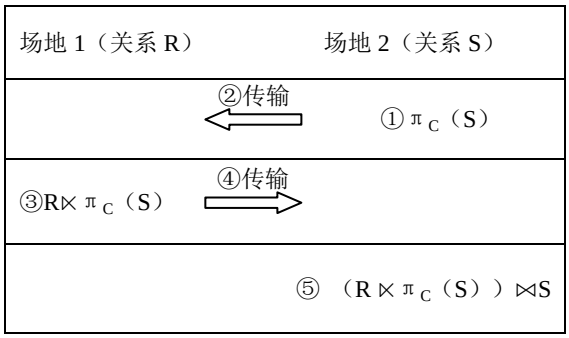


图 9.3　基于半联接的执行示意图

第 10 章 中间件技术

10.1 基本内容分析

10.1.1 本章重要概念

（1）中间件的定义和作用

（2）ODBC 分层的体系结构：应用程序，驱动程序管理器，DB 驱动程序，ODBC 数据源。

（3）ODBC 接口：应用程序基本流程，ODBC 句柄，ODBC 连接，SQL 语句的执行。ODBC 两套符合性级别。典型的 DB 应用系统开发工具。

（4）JDBC 的基本功能，JDBC 的结构，JDBC 接口。

10.1.2 本章的重点篇幅

（1）ODBC 分层的体系结构（教材 P403 的图 10.2）。

（2）ODBC 应用程序的基本流程（教材 P408 的图 10.6）。

（3）JDBC 驱动程序（教材 P427-428），JDBC API 接口（教材 P430）。

10.2 教材中习题 10 的答案

10.1 什么是中间件？有什么作用？

答：中间件是分布式环境中保证 OS、通信协议、数据库等之间进行对话、互操作的软件系统。

中间件的作用是保证了客户和服务器间的联系，使网络、数据库、操作系统对于应用软件的开发界面透明化。

10.2 试解释中间件的三个透明性。

答：中间件的网络透明性是指中间件能支持所有类型的网络。

中间件的服务器透明性是指不管服务器上的 DBMS 是何种型号（ORACLE、SYBASE、DB2 等），一个好的中间件都能通过标准的 SQL 语言与不同 DBMS 上的 SQL 语言连接起来。

中间件的语言透明性是指客户机可用任何开发语言进行发送请求和接受回答，被调用的功能应该像语言那样也是独立的。中间件还应该保证开发语言的数据类型和服务器上数据库使用的数据类型之间能够相互转换。

10.3 ODBC 技术与传统的数据库编程方式有什么区别？

答：传统的 DB 编程方式是“主语言+DML”，但一个应用程序却不能访问不同 DB 服务器上的数据。

ODBC 技术实际上是一个公共接口 API，使用 ODBC 技术，同一个应用程序就可以访问不同 DB 服务器上的数据。

10.4 ODBC 技术有什么作用？其卓越贡献是什么？

答：ODBC 技术的作用是使应用程序与 DBMS 在逻辑上可以分离，使应用程序具有数据库无关性。

ODBC 的卓越贡献是使应用程序具有良好的互用性和可移植性，并且具备同时访问多种 DBS 的能力，从而克服了传统数据库应用程序的缺陷。对用户而言，ODBC 驱动程序屏蔽掉了不同 DBS 的差异。

10.5 ODBC 的体系结构有哪几层？试叙述各层之间联系。

答：ODBC 的体系结构有四层：应用程序，驱动程序管理器，DBMS 驱动程序、数据源。

应用程序要完成 ODBC 外部接口的所有工作，用 C 语言和 ODBC 函数来编应用程序。

驱动程序管理器管理应用程序和 DBMS 驱动程序之间的交互作用，为应用程序加载和调用 DBMS 驱动程序。

DBMS 驱动程序执行 ODBC 函数，解释执行 SQL 语句。

数据源是驱动程序与 DB 连接的桥梁。

10.6 ODBC 数据库应用程序的主要功能是什么？主要完成哪些任务?

答：ODBC 数据库应用程序的主要功能有四个：调用 ODBC 函数，递交 SQL 语句给 DBMS，检索出结果，并进行处理。

应用程序要完成 ODBC 外部接口的所有工作。主要完成以下 8 个任务：

连接数据库；向数据源发送 SQL 语句；为 SQL 语句执行结果分配存储空间，定义所读取的数据格式；读取结果；处理错误；向用户提交处理结果；请求事务的提交和回退操作；断开与数据源的连接。

10.7 什么是驱动程序管理器？驱动程序管理器的主要功能是什么？

答：驱动程序管理器是一个动态连接库，用于连接各种 DBS 的 DBMS 驱动程序，管理应用程序和 DBMS 驱动程序之间的交互作用。

驱动程序管理器的主要功能有五点：为应用程序加载 DBMS 驱动程序；检查 ODBC 函数的合法性；为不同驱动程序的 ODBC 函数提供单一的入口；调用正确的 DBMS 驱动程序；提供驱动程序信息。

10.8 什么是 DBMS 驱动程序？主要任务是什么？

答：DBMS 驱动程序实际上是一个 DBMS，执行 ODBC 函数，解释执行 SQL 语句，实现对数据源的各种操作。

驱动程序主要任务有六个：建立应用程序与数据源的连接；向数据源提交用户请求执行的 SQL 语句；进行数据格式和类型的转换；把处理结果返回给应用程序；将 DBS 的错误转换成 ODBC 标准错误代码返回给应用程序；根据需要定义和使用光标。

10.9 DBMS 驱动程序有哪两种类型？主要区别是什么？

答：驱动程序有两种类型：单层驱动程序和多层驱动程序。

单层驱动程序中包含了数据库引擎，解释执行 SQL 语句。在网络中传输的是整个数据库文件，所以网络的数据通信量很大。

多层驱动程序中不包含数据库引擎，将 SQL 语句传递给数据源服务器，由 DBMS 解释执行。因此在网络中传输的只是用户请求和数据库处理的结果，从而使网络的数据通信量大大减少，减轻了网络的负担，均衡了服务器和客户机的负载，提高了应用程序的运行效率。

10.10 什么是 ODBC 数据源？有哪三类？

答：数据源是驱动程序与 DBS 连接的桥梁，用于表达一个 ODBC 驱动程序和 DBMS 特殊连接的命名。

数据源分成三类：用户数据源，系统数据源，文件数据源。

10.11 试叙述 ODBC 应用程序的基本流程。

答：应用程序的基本流程分为三个部分：初始化，SQL 处理和终止部分。每一部分按常规，要使用一些 ODBC 函数（与 DB 有关）。

10.12 什么是 ODBC 句柄？有哪几种？这几种句柄之间有什么联系？

答：ODBC 句柄就是应用程序变量，系统用来存储关于应用程序的上下文信息和应用程序所用到的一些对象。

有三种句柄：环境句柄，连接句柄和语句句柄。三者之间是嵌套的关系。一个应用程序只有一个环境句柄，在环境句柄内可以定义若干连接句柄，在连接句柄内可定义多个语句句柄。

10.13 ODBC 技术提供哪两种不同的执行 SQL 语句的方式？各用在什么场合？

答：ODBC 技术提供两种不同执行 SQL 语句的方法：直接执行和有准备地执行。前者以快捷的方式执行 SQL 语句，在 ODBC 函数中放上一条可直接执行的 SQL 语句。后者则提供了更大的灵活性，ODBC 函数 SQLPrepare 把 SQL 语句准备好（但可能条件不齐），待条件补全后，再用 ODBC 函数 SQLExecute 执行 SQL 语句。

如果 SQL 语句已经组织好了，并且只使用一次，那么可使用“直接执行 SQL 语句的函数”。如果 SQL 语句需要多次执行，或者 SQL 语句的查询条件还不齐，那么这种情况就要使用“有准备地执行 SQL 语句”的函数。

10.14 ODBC 技术中有哪两个主要的光标函数？起什么作用？

答：ODBC 中主要的光标函数有两个：移动光标函数 SQLFetch 和读光标指向行中一列值的函

数 SQLGetData。
10.15 ODBC 技术中有哪两个符合性级别？为什么要设置这两个符合性级别？
答：ODBC 定义了两套符合性级别：
一套是关于 ODBC 函数调用的 API 符合性，指出驱动程序支持哪些 ODBC 函数。
另一套是关于所支持的 SQL 的 SQL 符合性，指出驱动程序支持哪些 SQL 语句和功能。
10.16 ODBC API 与 SQL CLI 间有什么联系？
答：ODBC API 是微软公司开发的 ODBC 标准。
SQL CLI 是 SAG 财团和 X/Open 组织开发的 SQL3 标准中的一部分内容。
两者使用了不同的术语，但双方承诺要使标准统一起来。
10.17 有哪些典型的数据库应用系统开发工具？这些工具有些什么共同的特点？
答：典型的数据库应用系统开发工具有四个：PowerBuilder8.0，Delphi6.0，Visual Basic 6.0，和 Developer/2000。这些工具都属于 4GL 的软件开发工具，具有基于 Windows 界面、C/S 结构、面向对象的可视化等特点。
10.18 Java 语言对 Internet 的广泛应用起了什么作用?
答：在 Java 出现以前，Internet 主要用于信息共享，信息访问和传递方式也只是简单的链接。现在，Java 已经成为 Internet 应用的主要开发语言，将信息共享的方式往前推进了一大步，使通过 Web 提供完全交互式的应用程序成为可能。
10.19 试解释 Java 源程序的运行顺序。
答：Java 源程序分两步运行：
• Java 源代码先通过 Java 编译器产生 Java 虚拟机字节代码（bytecodes）。字节代码文件称为类文件（class files）。
• 然后，字节代码在本地或通过网络下载到客户机，再经 Java 解释器将字节代码转换成实际系统的机器代码去执行。
10.20 试解释 Java 语言的“平台无关性”？
答：开发人员在编写源程序时，不必担心程序运行的实际平台。当程序一旦编成，便可以不经修改直接运行于各种不同的平台上。这个性质称为 Java 语言的“平台无关性”。
10.21 Java 语言有哪些良好的特性?
答： Java 具有下述五个特性：简单性，可移植性，面向对象，分布式和动态结构，安全性。
10.22 Java 应用有哪两种方式?有什么区别?
答：一般可以把 Java 的应用程序分成两类：应用程序（application）和小应用程序（applet）。简单的说，小应用程序就是嵌入式 Web 文档的程序，而应用程序则是所有其它类型的程序。
小应用程序是从 Web 文档进来的 Java 程序，也就是从 HTML 文件进来的程序。而应用程序则是从命令行上运行的程序。
小应用程序需要来自 Web 浏览器的大量信息（包括何时启动、何时激活或关闭等）。而应用程序可能运行在最简单的环境中，它来自外部世界的惟一输入就是命令行参数。
10.23 什么是 JDBC?
答：JDBC 是执行 SQL 语句的 Java API。JDBC 是“Java DataBase Connectivity”（JDBC 数据库连接 ）的缩写。JDBC 由一组用 Java 语言编写的类与接口组成。
10.24 JDBC 的基本功能是什么？
答：Java 与 JDBC 的结合使程序员可以只写一次数据库应用软件便能在各种数据库系统上运行。JDBC 的基本功能包括三点：建立与数据库的连接，发送 SQL 语句，处理结果。
10.25 试比较 CGI 和 JDBC 这两种方法的程序执行过程。
答：在 CGI 方法中，当应用程序发出访问数据库的命令后，Web 服务器调用所需要的 CGI 程序，并利用相应的服务器脚本技术解释执行 CGI 程序，通过 CGI 程序实现对数据库的访问。

在 JDBC 方法中，当应用程序发出访问数据库的命令后，只需要将 SQL 命令发送给数据库的服务器，而不再需要 Web 服务器解释执行，这就大大缩短了执行 SQL 语句的时间。

在 CGI 方法中，CGI 脚本必须独立的连接数据库，处理执行结果。而 JDBC 的解决方案使应用程序直接与数据库相连，执行各种操作。因此采用 JDBC 来访问数据库比使用 CGI 方法效果更好，访问的速度也更快。

10.26 JDBC API 数据库设计方法有哪两种方式？

答：JDBC API 支持这两种应用方式：Java 应用程序和 Java 小应用程序，这两种方式分别在两层应用模型和三层应用模型中实现。

10.27 JDBC API 采取哪些措施与标准 SQL 保持一致性的？

答： JDBC API 采取三种方法与标准 SQL 保持一致：

• 允许将任何查询字符串传递给基础 DBMS 驱动器，这意味着应用可以自动地使用尽可能多的 SQL 功能，但这会使某些 DBMS 系统接收到某种错误的查询。

• 采用 ODBC 风格的方法，提供表示几种常见的 SQL 差别的标准 JDBC 语法。

• 对于复杂应用，借助于 DatabaseMetaData 接口，提供关于 DBMS 的描述性信息，使应用能适应每个 DBMS 的需求与能力。

10.28 JDBC 驱动程序有哪几类？结构如何？

答：JDBC 驱动程序有四类：

本地库 Java 驱动程序，独立于 DBMS 的网络协议驱动程序，DBMS 协议 Java 驱动程序，JDBC-ODBC 桥驱动程序。（解释略）

10.29 JDBC API 的目标是什么？

答：JDBC API 的目标主要有以下六点。

• 为 Java 定义一个“调用层”（call-level）的 SQL 接口

• 遵循 SQL2 标准

• JDBC 应建立在现存的数据库接口上

• 必须保证这个接口与 Java 系统的其他部分保持一致

• 使基本的 API 尽量简单

• 尽量保持强大的、静态的类型

10.30 JDBC 接口分为哪几个部分？

答：JDBC 接口分为两个层次：应用程序层和驱动程序层。

应用程序层是面向程序开发人员的 JDBC API，驱动程序层是由驱动厂家实现的。

10.31 在 java.sql 包中，JDBC 有哪些核心的接口和类？请对每一个接口和类作简短的解释。

答：在 java.sql 包中，包含了 JDBC 的核心接口和类。

JDBC 的核心接口主要有 8 个：

（1）java.sql.CallableStatement 接口，用于执行数据库中的 SQL 存储过程。

（2）java.sql.Connection 接口，用于与特定的数据源建立连接。

（3）java.sql.DatabaseMetaData 接口，定义了多种方法来处理在特定连接中包含的数据库信息，如基本表、存储过程、连接功能、支持的语法等信息。

（4）java.sql.Driver 接口，用于数据库驱动程序与数据库建立连接关系。

（5）java.sql.PreparedStatement 接口，这个对象用于多次执行相同查询语句时使用。

（6）java.sql.ResultSet 接口，这个接口提供访问结果集的许多方法。

（7）java.sql.ResultSetMetaData 接口，提供了获取结果集中列的数目、类型和属性等信息的方法。

（8）java.sql.Statement 接口，这个对象用来执行静态 SQL 语句（通常是没有参数的 SQL 语句）。

JDBC 的核心类主要有 9 个：

（1）java.sql.Date 类，它是 java.util.Date 类的子类，为用户提供了处理日期的方法。

（2）java.sql.DriverManager 类，提供了用于管理 JDBC 驱动程序的方法。

（3）java.sql.DriverPropertyInfo 类，一般只被高级程序员使用，通过使用 getDriverProperties 与 Driver 进行交互，获得或使用建立连接需要的资源。

（4）java.sql.Time 类，提供处理时、分和秒的方法。这个类是 java.util.Date 类的子类。

（5）java.sql.Timestamp 类，它是 java.util.Date 类的子类，用于处理时间戳问题。

（6）java.sql.Types 类，定义了一些用于表示 SQL 类型的变量，这些类型的常量值与 X/Open 标准中的类型值一致。

（7）java.sql.DataTruncation 类，用于数据截断，这个类是 SQLWarning 类的子类。

（8）java.sql.SQLExceptoin 类，它是 java.lang.SQLException 类的子类，用于提供处理访问数据库时的出错信息。

（9）java.sql.SQLWarning 类，它是 SQLExceptoin 类的子类，所以 SQLExceptoin 类的方法都可以使用。

10.32 使用 JDBC 来连接数据库的一般步骤有哪几步？

答：使用 JDBC 来连接数据库的步骤一般有以下三步：

建立数据源，建立连接，发送 SQL 语句。（解释略）

10.3 自测题及答案

10.3.1 填空题

1．在 C/S 系统中广泛地使用了中间件技术，以隐藏_____，屏蔽_____。

2．中间件是分布式环境中保证_____、_____、_____等之间进行对话、互操作的软件系统。

3．中间件的作用是保证网络中各部件（软件和硬件）之间_____地连接，即隐藏网络部件的_____性。

4．ODBC 技术以_____结构为设计基础，使得_____与_____之间在逻辑上可以分离，是应用程序具有数据库无关性。

5．ODBC 是一个分层的体系结构，由纵向四部分构成：_____、_____、_____和_____。

6．ODBC 数据库应用程序是用_____和_____编写的。

7．驱动程序管理器的作用是为_____调用和加载_____。

8．数据源是_____与_____连接的桥梁。

9．ODBC 应用程序的基本流程分为三个部分：_____、_____和_____。

10．在 ODBC 应用程序的初始部分，与数据库操作有关的步骤有四步：_____、_____、_____和_____。

11．在 ODBC 应用程序的终止部分，与数据库操作有关的步骤有四步：_____、_____、_____和_____。

12．像 PowerBuilder 一类的数据库应用系统开发工具具有四个显著的特点：_____、_____、_____和_____。

13．JDBC 的基本功能包括三个部分：_____、_____和_____。

14．JDBC API 数据库设计方法有两种方式：_____和_____。这两种方式分别在两层和三层应用模型中实现。

15．JDBC 驱动程序有四类：_____、_____、_____和_____。

10.3.2 填空题答案

1．各种复杂性 各种系统之间的差异

2．操作系统 通信协议 数据库

3．透明 异构

4．C/S 应用程序 DBMS

5．ODBC 数据库应用程序　　驱动程序管理器　　DB 驱动程序　　数据源
6．宿主语言（C 语言）ODBC 函数
7．应用程序　　DB 驱动程序
8．驱动程序　　DBS
9．初始化　SQL 处理　终止
10．分配环境句柄　分配连接句柄　与服务器连接　分配语句句柄
11．与服务器断开　　释放语句句柄　释放连接句柄　释放环境句柄
12．基于 C/S 结构　Windows 界面　OO 的开发技术 可视化技术
13．建立与数据库的连接　　发送 SQ 语句　　处理结果
14．Java 应用程序　　Java 小应用程序
15．本地库 Java 驱动程序　　网络协议 Java 驱动程序 本地协议 Java 驱动程序
JDBS-ODBC 桥驱动程序

第 11 章 数据库与 WWW

11.1 基本内容分析

本章的重要概念有：

（1）Internet、Intranet 和 WWW 的定义，IP 地址和域名。

（2）ISO/OSI/RM、实用的协议模型和 TCP/IP 协议组等三种网络层次模型结构。

（3）WWW 与 DB 交互的方法：CGI，JDBC 和 API。三种方法的比较。

（4）CGI 程序的功能、工作方式，CGI 规范，CGI 与 DB 的集成。

（5）三种常用的 Web 编程语言：ASP、JSP 和 PHP。

11.2 教材中习题 11 的解答

11.1 试对 Internet、Intranet、WWW 这三个概念写出确切的定义。三者之间有什么关系？

答：Internet 是一个“网间网”，应包含三方面的内容：网络、网络用户和信息资源。

Intranet，即企业内部网，是属于某一组织的 Web 站点或一组 Web 站点，只能由该组织的成员访问。

WWW 是一个涉及全世界的信息系统，也是一种 Internet 上高效、方便的检索工具。

三者之间是相互联系、相互依赖而存在。Internet 把不同的网络连在一起，Intranet 是企业内部使用的 Internet，WWW 统一了整个 Internet，使之变成一个超媒体形象资源的集合。

11.2 什么是 IP 地址？其结构如何？什么是域名？

答：连接到 Internet 上的每台计算机都必须有一个惟一地址。Internet 的地址用数字来表示的，称为 IP 地址。

IP 地址含有 4 个字节，32 个二进制位。在书写时，通常每个字节都用十进制表示，而字节之间用小圆点“.”分隔开来。

要记住这些枯燥的数字不是容易的事，因此大多数计算机都有一个便于记忆的别名与 IP 地址相连系，这个别名称为“域名”。

11.3 WWW 主要有哪三种标准成分组成？

答：WWW 主要由三种标准成分组成，即 URL（统一资源定位器）、HTTP（超文本传输协议）和 HTML（超文本标记语言）。

11.4 试叙述 ISO/OSI/RM、实用的协议模型和 TCP/IP 等三种网络层次模型结构。

答：为了保证计算机网络的开放性与兼容性，网络协议必须遵循标准化的体系结构。主要有下面三个：

（1）ISO/OSI/RM 网络协议层次模型：这是 ISO 于 1983 年提出的，把网络协议分成七层，自下而上分别为物理层、数据链路层、网络层、传输层、会话层、表示层和应用层。

（2）实用的协议模型：为使用方便，实际使用时把 OSI/RM 模型的上面三层统称为“应用层”。

（3）TCP/IP 协议组：已成为事实上的工业标准，把网络协议看成五个层次，自下而上分别为物理层、网络接口层、网络层、传输层和应用层，与实用的协议模型类似，但层次的命名不一样。

11.5 ISO/OSI/RM 的网络层次模型有几层？每一层的功能是什么？

答：ISO/OSI/RM 的网络层次模型有七层，自下而上每层的功能如下：

- 物理层：负责两个端点处物理信道上二进制比特的传输。
- 数据链路层：保证信息以帧（Frame）为单位在链路上的可靠传送。
- 网络层：保证信息以包为单位在两个端点处的路由传输。
- 传输层：确保较高层上的数据以可靠，高效的方式传送。
- 会话层：管理和同步较高层的数据交换的机制。
- 表示层：提供一套格式服务。
- 应用层：负责向用户提供使用各种服务软件的工具和方法。

11.6 TCP/IP 中的 TCP 协议和 IP 协议的功能各是什么？

答：TCP/IP 中的 TCP 协议是在传输层中，其功能是保证数据发送的正确性，确保数据到达顺序和发送顺序一致，如果数据发生损失或丢失，TCP 要求重新发送该数据。

TCP/IP 中的 IP 协议是在互联网层中，其功能是使用 IP 地址来确定发送端（源主机）和接收端（目的地主机），提供端到端(主机到主机)的“数据报”传递。

11.7 为什么 WWW 与数据库的结合显得越来越重要？

答：随着 Internet/Intranet 的兴起与发展，WWW 与数据库的结合显得越来越重要。这种结合有两种方式：

第一种方式：WWW 作为访问 DB 的工具，可以借用统一的浏览器软件，无需开发数据库前端。

第二种方式：DB 作为 WWW 的一种工具，为 WWW 提供更加有效的数据组织和管理方式。

无论是 DB 为 WWW 服务，还是 WWW 为 DB 服务，还是它们相互作用。将给我们带来的是一种更为有效、便利的信息管理和展示方式。

11.8 WWW 与 DB 的连接，现在有哪三种主要方式？各有什么优缺点？

答：WWW 与 DB 的连接，现在有 CGI、JDBC 和 API 三种方式。这三种方式的优缺点，见教材 P291 的表 10-1。

11.9 什么是 CGI？在开发 Web 数据库应用中起什么作用？可用哪些语言编写 CGI？

答：CGI 是一个用于定义 Web 服务器与外部程序之间通信方式的标准，使得外部程序能够生成 HTML、图像或者其它内容。

CGI 程序在开发 Web 数据库中的作用相当于一个中介，在浏览器、Web 服务器和数据库之间传递信息。

几乎任何一种高级语言都可以被用来编写 CGI 程序。在实际应用中，根据所选用的操作系统类型，用于 CGI 编程的语言可以为 Perl、C、VB 或 Shellscripts(UNIX)。

11.10 CGI 有哪两种规范？各有哪些方法或方式？

答：CGI 有两种规范：方法规范和接口规范。

CGI 方法规范定义了调用 CGI 程序的途径。有三种主要的方法：GET 方法，POST 方法和 HEAD 方法。

CGI 接口规范定义了 CGI 与 Web 服务器之间通信方式，有四种：命令行，标准输入，标准输出和环境变量。

11.11 什么是 ASP？ASP 技术有什么特点？

答：ASP 是 Active Server Pages（动态服务器页面）的缩写，它是一个服务器端的脚本环境，在站点的 Web 服务器上解释脚本，可产生并执行动态交父式高效率的站点服务器应用程序。ASP 可以胜任基于微软 Web 服务器的各种动态数据发布。

ASP 技术有下列特点：

ASP 技术主要有下列特点：：

- 使用 VBScript、JScript 等简单易懂的脚本语言；
- 程序无须编译，容易编写，可在服务器端直接执行；
- 浏览器无关性；
- 能与任何 ActiveX scripting 语言相容；
- 提高了程序的安全性；
- 具有面向对象的开发特点；
- 无限的可扩充性。

11.12 ASP 服务器端程序的原理是什么？

答：ASP 本身并不是一种脚本语言，它只是提供了一种使镶嵌在 HTML 页面中的脚本程序得以运行的环境。与一般的程序不同，ASP 程序无须编译，其控制部份是使用 VBScript、JScript 等脚本语言来设计的。执行 ASP 程序时，由脚本解释器进行翻译并将其转换成服务器所能执行的命令。

11.13 什么是 JSP？主要包含哪些内容？

答：JSP 是 Java Server Pages（Java 服务器页面）的缩写，是由 Sun 公司于 1999 年推出的一项技术，是基于 JavaServlet 以及整个 Java 体系的 Web 开发技术，进而可以建立先进、安全和跨平台的动态网站。

JSP 主要包含指令和脚本语言两大部分。

11.14 JSP 脚本语言是嵌入在页面内的 java 代码，有哪三种？

答：JSP 脚本语言是嵌入在页面内的 java 代码，分为三种：

声明（Declarations）、脚本（Scriptlet）、表达式（Expression）。

11.15 什么是 JSP 中的指令？有什么作用？

答：指令是JSP的一个重要的内容。指令提供了该页的全局信息，例如，重要的状态，错误处理，是否是session的一部分等等。这些指令包括：include指令，page指令，taglib指令以及动作指令（包括forward指令，useBean指令和plugin指令）。

Include 指令用于在 JSP 程序中包含一个静态或动态文件。

page 指令用于定义整个 JSP 文件都要使用的属性值，并在 JSP 程序进行编译时将这些文件属性传递给 JSP 引擎。

taglib 指令用于定义一个标签库以及其自定义标签的前缀。

forword 指令用于重定向一个 HTML 文件，JSP 文件，或者是一个程序段。

useBean 指令用于定位或示例一个 JavaBeans 组件。

plugin 指令用于执行一个 Applet 或 Bean，有可能的话还要下载一个 Java 插件用于执行它。

11.16 什么是 JSP 中的内置对象？有什么作用？

答：为了开发的方便，在 JSP 中已经预先定义好了一些常用的对象，在 java 脚本中不用声明就可以直接使用这些对象，称之为内置对象。

通过在 java 脚本中使用这些对象，可以方便地实现特定的功能。各个对象的功能如图 11.1 所示。

对象名称	对象功能
request 对象	用户端请求，此请求会包含来自 GET/POST 请求的参数
response 对象	网页传回用户端的回应
pageContext 对象	网页的属性是在这里管理
session 对象	与请求有关的通话期
application 对象	Servlet 正在执行的内容
out 对象	用来传送回应的输出资料流
config 对象	Servlet 的构架物件
page 对象	Jsp 网页本身

图 11.1

11.17 什么是 PHP？PHP 语言的特点是什么？

答：PHP 是 Professional Home Pages 的缩写，也是一种用服务器端 HTML 嵌入脚本创建动态网站的常用方式，其特色在于对 DB 操作的方便性。

PHP 是通过 Internet 进行合作开发的开放源代码软件，它从简单的 Perl 语言编写的 CGI 程序开始，通过不断的扩展与修改，最终称为结构完善、功能强大的动态网站的创建工具。

PHP 语言具有下列特点：

- 支持多种系统平台；
- 具有自由软件的特点；
- 版本更新速度快；
- 容易和 HTML 网页融合，执行效率高；

• 具有丰富的函数接口；
• 具有丰富的功能，几乎完整地囊括所有网站所需的功能；
• PHP 具有很高的安全性。

第 12 章 XML 技术

12.1 基本内容分析

本章的重要概念有：

（1）从 SGML、HTML 到 XML 的发展，XML 文档、DTD（文档类型定义）、XML 模式，XML 数据库的存取方法。

（2）XML 查询语言 XQUERY 的基本功能、基本概念，简单查询的表达，各种类型查询的表达，复杂查询的表达。

（3）基于关系数据库 XML 的处理，XML 存储和查询系统体系结构，XML 数据到关系数据库的存储映射，基于关数据库的 XML 查询。

12.2 教材中习题 12 的解答

12.1 名词解释

（1）SGML: SGML 的源头最早可追溯到 1969 年 IBM 公司的一个研究小组开发的“通用标记语言”GML，1986 年被完善成 ISO 国际标准（ISO 8879），即 SGML，由 ISO/IEC 联合技术组 JTC1/WG4 负责制定，目前 WG4 还在进行 SGML 相关的标准化工作。

SGML（通用标记语言标准，Standard for Generalized Markup Language）不仅仅是一种标记语言，还可认为是一种元语言。利用它可以定义各种各样的标记语言。它允许开发者根据需要制定相关的 DTD（文档类型定义）。但是这种灵活性带来的弊端是需要复杂的软件去处理它。

（2）HTML: 用于万维网的标记语言 HTML 可以看作是一个非正统的 DTD。HTML 针对 SGML 过于复杂的问题，指定很小的一组结构和语义标记，使其适合相对简单的文档的书写，另外还增加了对超文本的支持。

（3）XML: XML 是 SGML 的一个子集。XML 的设计目的明确地定位为万维网上的应用。设计工作基于两个重要的准则：易于编写处理 XML 的计算机程序，以及人和系统能花费极少的代价将 HTML 移植到 XML 中。

12.2 试描述 XML 文本的组成部分

答：一个 XML 文档由序言和文档实例两个部分组成。序言包括一个 XML 声明和一个文档类型声明，二者都是可选的。文档类型声明由 DTD 定义，它定义了文档类型结构。序言之后是文档实例，它是文档的主体，它是 DTD 的一个实现。

12.3 比较 XML 数据库的几种形式

答：XML 数据库有两种形式：

（1）纯粹的 XML 存取方法：它是专门针对 XML 格式文档进行存取管理和数据操作的数据库，数据库中的数据和元数据完全采用 XML 结构表示，其底层针对 XML 数据的特点，采用相应的存储结构，而不是采用现有的数据存储工具。

（2）基于关系数据库的存取方法：它是在关系数据库基础之上扩展了 XML 支持模块，它将 XML 数据存储在关系数据库中，在查询时将 XML 数据查询语言转换成关系数据数据库查询语言。

这两种方法各有优缺点。就前者而言，XML 文档存取无需模式转换，存取速度快；对格式复杂的 XML 文档有很好的支持；支持大多数最新的 XML 技术标准;支持层次化的数据模型;支持基于标签和路径的操作；支持对位置特征的查询。这是传统的查询语言所不具备的。但是这种方法技术比较新，没有经过时间的考验。而后者的优势在于其重发利用了传统数据库的成熟技术，如并发控制、事务处理、结构化查询等。但劣势在于 XML 文档存入到数据库时需要将其分解并进行数据映射，取出时需要重新组合，开销大，且有可能丢失某些信息。

12.4 假设 XML 文档 actor.xml 对应的 DTD 如下：

```
<?xml encoding="ISO-8859-1"?>
<!ELEMENT W4F_DOC (Actor)*>
```

```
<!ELEMENT Actor (Name,Filmography)>
<!ELEMENT Name (FirstName, LastName)>
<!ELEMENT FirstName (#PCDATA)>
<!ELEMENT LastName (#PCDATA)>
<!ELEMENT Filmography (Movie)*>
<!ELEMENT Movie (Title,Year)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Year (#PCDATA)>
```

请给出不同要求下的 XQuery 查询语句：

（1）查询陈道明参演的所有电影的名称和发行年份。

（2）找出至少出演过 4 部以上电影的演员的 Name 以及他参演的所有电影的名字。

（3）查询参演过电影<<乱世佳人>>的所有演员的 Name,结果按名字降序排序。

解：（1）

```
<result>
  for $i in document("/actor.xml")//Actor
  where $i/Name/FirstName="陈" and $i/Name/LastName="道明"
  return {$i/Filemography/Movie}
</result>
```

（2）

```
<result>
  for $i in document("/actor.xml")//Actor
  where count($i/Filemography/Movie)>=4
  return
    {$i/Name}
    {$i/Filemography/Movie/Title}
</result>
```

（3）

```
<result>
  for $i in document("/actor.xml")//Actor
  let $j:=$i/Filemography/Movie/Title
  where $j="乱世佳人"
  return
     <actor>
       {$i/Name}
     </actor>
  sortby(FirstName,LastName)
</result>
```

12.5 设有一个 XML 文档集 movie.xml，其 DTD 如下：

```
<?xml encoding="ISO-8859-1"?>
<!ELEMENT W4F_DOC (Movie)*>
<!ELEMENT Movie (Title,Year,Directed_By,Genres)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Year (#PCDATA)>
<!ELEMENT Directed_By (Director)*>
<!ELEMENT Director (#PCDATA)>
```

<!ELEMENT Genres (Genre)*>
<!ELEMENT Genre (#PCDATA)>
试完成下列查询语句：
（1）查询 2003 年发行的所有电影的 title。
（2）根据 movie.xml 和 actor.xml，查询出陈道明参演的所有电影的导演。
解：（1）

```
<result>
  for $i in document("/movie.xml")//Movie
  where $i/Year=2003
  return
     {$i/Title}
</result>
```

（2）

```
<result>
  for $i in document("/actor.xml")//Actor,$j in document("/movie.xml")//Movie
  where  $i/Name/FirstName=" 陈 "  and  $i/Name/LastName=" 道 明 "  and
$i/Filemography/Movie/Title=$j/Title
  return
    {$j/Directed_By}
</result>
```

12.6 试比较基于模板的驱动和模型驱动两种方式的优缺点。
答：基于模板方法的原理是首先定义一个模板，然后在模板中嵌入对数据库访问的命令，这些命令将交给数据库关系系统进行执行。

在基于模型驱动的方法中，数据从数据库到 XML 文档的传送用一个具体的模型，而不是用户定义的模型实现的。用户可以将相应的格式直接定义到这个模型上，从而将数据库中的数据展示各种各样的形式。

其中前者更为灵活，可以定一个各种格式，但后者对用户的使用更为方便，用户不用操心数据具体的格式。
12.7 对下面这个 DTD:

```
<!element paper(title,author*,keyword*)>
<!element title(#PCDATA)>
<!element author(name)>
<!element name(#PCDATA)>
<!element keyword (#PCDATA)>
```

描述其 Basic inlining、Shared Inlining 和 Hybrid Inlining 三种方法产生的结果。
答：（1）Basic inlining 产生的结果如下:

```
paper(paperid:integer,paper.title:string)
title(titleid:integer, title:string)
keyword(keyid:integer,key.aprentid:integer,keyword:string)
author(authorid:integer, author.parentid:integer,author.name:string)
name(nameid:integer, name:string)
```

（2）Shared inlining 产生的结果如下:

```
paper(paperid:integer,paper.title:string)
keyword(keyid:integer,key.aprentid:integer,keyword:string)
author(authorid:integer, author.parentid:integer,author.name:string)
```

（3）Hybrid Inling 产生的结果如下:

paper(paperid:integer,paper.title:string)
keyword(keyid:integer,key.aprentid:integer,keyword:string)
author(authorid:integer, author.parentid:integer,author.name:string)