

《数据库原理及应用》复习重点

第一章 数据库系统基本概念

要求、目标：

了解和掌握数据管理技术的发展阶段，数据描述的术语，数据抽象的四个级别，数据库管理系统的功能，数据库系统的组成。

一、数据管理技术的发展

1. 分为四个阶段：人工管理阶段、文件系统阶段、数据库阶段和高级数据库阶段。
2. 数据库阶段数据管理的特点：
 - 1) 采用数据模型表示复杂的数据结构。
 - 2) 有较高的数据独立性。
 - 3) 数据库系统为用户提供了方便的用户接口。
 - 4) 提供四方面的数据控制功能：数据库的恢复、数据库的并发控制、数据的完整性、数据安全性。
 - 5) 增加了系统的灵活性。
3. 数据库（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。
4. 数据库管理系统（DBMS）：是位于用户与操作系统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，包括 DB 的建立、查询、更新及各种数据控制。
5. 数据库系统（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，即它是采用数据库技术的计算机系统。

二、数据描述

1. 分为三个阶段：概念设计、逻辑设计和物理设计。
2. 概念设计中的术语：
 - 1) 实体：客观存在，可以相互区别的事物称为实体。
 - 2) 实体集：性质相同的同类实体的集合。
 - 3) 属性：实体有很多特性，每一个特性称为属性。
 - 4) 实体标识符（关键码或键）：能惟一标识实体的属性或属性集。以上概念均有类型和值之分。
3. 逻辑设计中的术语：
 - 1) 字段（数据项）：标记实体属性的命名单位称为字段或数据项。
 - 2) 记录：字段的有序集合。
 - 3) 文件：同一类记录的集合。
 - 4) 关键码：能惟一标识文件中每个记录的字段或字段集。以上概念均有类型和值之分。
4. 概念设计和逻辑设计中术语的对应关系：

概念设计		逻辑设计
实体	—	记录
属性	—	字段（数据项）
实体集	—	文件
实体标识符	—	关键码

5. 实体之间联系的元数：与一个联系有关的实体集个数。常用二元联系。二元联系的类型有三种：一对一联系、一对多联系、多对多联系。
6. 一对一联系：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，那么实体集 E1 和 E2 的联系称为“一对一联系”，记为“1: 1”。
7. 一对多联系：如果实体集 E1 中每个实体与实体集 E2 中任意个（零个或多个）实体间有联系，而 E2 中每个实体至多和 E1 中一个实体有联系，那么称 E1 对 E2 的联系是“一对多联系”，记为“1: N”。
8. 多对多联系：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）实体间有联系，反之亦然，那么称 E1 和 E2 的联系是“多对多联系”，记为“M: N”。

三、数据抽象的级别

1. 数据模型：描述数据库的结构和定义，对现实世界的数据进行抽象。
2. 从现实世界的信息到数据库存储的数据以及用户使用的数据库数据是一个逐步抽象过程，根据数据抽象的级别定义了四种模型：概念模型、逻辑模型、外部模型和内部模型。
3. 概念模型：表达用户需求观点的数据全局逻辑结构的模型。
4. 逻辑模型：表达计算机实现观点的 DB 全局逻辑结构的模型。
5. 外部模型：表达用户使用观点的 DB 局部逻辑结构的模型。
6. 内部模型：表达 DB 物理结构的模型。
7. 数据抽象的过程、即数据库设计的过程具体步骤：
 - 1) 根据用户需求，设计数据库的概念模型；
 - 2) 根据转换规则，把概念模型转换成数据库的逻辑模型；
 - 3) 根据用户的业务特点，设计不同的外部模型，给程序员使用；
 - 4) 数据库实现时，要根据逻辑模型设计其内部模型。
 通常分为概念设计、逻辑设计（2 和 3 步）和物理设计三个阶段。
8. 常用的概念模型是实体联系（ER）模型，ER 模型主要用 ER 图来表示。
9. 逻辑模型的分类：层次模型、网状模型、关系模型等。
10. 层次模型：用树型（层次）结构表示实体及实体间联系的数据模型。
11. 1969 年，美国 IBM 公司的 IMS 系统是典型的层次模型系统。
12. 网状模型：用有向图结构表示实体及实体间联系的数据模型。
13. 1969 年，CODASYL 组织提出 DBTG 报告中的数据模型是网状模型的主要代表。
14. 关系模型：是由若干个关系模式组成的集合。关系模式即记录类型，它的实例称为关系，每个关系实际上是一张二维表格。
15. 1970 年，美国 IBM 公司的 E.F.Codd 连续发表论文，提出关系模型，奠定了关系数据库的理论基础。关系数据库是目前的主流数据库。
16. 外部模型中的模式称为视图。
17. 三级模式：

从用户（或应用程序）到数据库之间，DB 的数据结构描述有三个层次：

 - 1) 外模式：用户与数据库系统的接口，是用户用到的那部分数据的描述。外模式由若干个记录类型组成。
 - 2) 逻辑模式：是数据库中全部数据的整体逻辑结构的描述。
 - 3) 内模式：是数据库在物理存储方面的描述。

注意：外模式是逻辑模式的子集。

18. 两级映像：

1) 外模式/逻辑模式映像：存在于外模式和逻辑模式之间，用于定义外模式和逻辑模式之间的对应性。

2) 逻辑模式/内模式映像：存在于逻辑模式和内模式之间，用于定义逻辑模式和内模式之间的对应性。

19. 数据库系统的三级模式、两级映像结构使数据库系统达到了高度的数据独立性。

20. 数据独立性：是指应用程序与数据库的数据结构之间相互独立，在修改数据结构时，尽可能不修改应用程序。分为逻辑数据独立性和物理数据独立性。

21. 逻辑数据独立性：如果数据库的逻辑模式要修改，那么只要对外模式/逻辑模式映像作相应的修改，可以使外模式和应用程序尽可能保持不变。这样就认为数据库达到了逻辑数据独立性。

22. 物理数据独立性：如果数据库的内模式要修改，即数据库的物理结构有所变化，那么只要对逻辑模式/内模式映像作相应的修改，可以使逻辑模式尽可能保持不变。也就是对内模式的修改尽量不影响逻辑模式，当然对外模式和应用程序的影响更小，这样就认为数据库达到了物理数据独立性。

四、数据库管理系统（DBMS）

1. DBMS 的主要功能：数据库的定义功能（DBMS 提供 DDL 定义数据库的三级模式、两级映像等）、数据库的操纵功能（DBMS 提供 DML 实现对数据的操作，基本的数据操作有检索和更新两类）、数据库的保护功能、数据库的维护功能、数据字典。

五、数据库系统（DBS）

1. DBS 的组成：是数据库、硬件、软件和数据库管理员的集合体。
2. 软件包括 DBMS、OS、各种主语言和应用开发支撑软件等程序。其中，DBMS 是 DBS 的核心软件，要在 OS 支持下才能工作。
3. 数据库管理员（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、监控和维护数据库结构的责任。

第一章 复习题

一、单项选择题

1. 在数据库系统中，当数据库的内模式发生改变时，应用程序也可以不变。这是（ A ）
A) 物理数据独立性 B) 逻辑数据独立性
C) 位置数据独立性 D) 存储数据独立性
2. 在下面列出的数据模型中，哪一个是概念数据模型（ D ）
A) 关系模型 B) 层次模型 C) 网状模型 D) 实体-联系模型
3. 下面列出的数据管理技术发展阶段中，哪个(些)阶段数据不能保存在计算机中？（ A ）
I. 人工管理阶段
II. 文件系统阶段
III. 数据库阶段

- A) 只有 I B) 只有 II C) I 和 II D) II 和 III
4. 用二维表结构表示实体以及实体间联系的数据模型称为 (C)
A) 网状模型 B) 层次模型 C) 关系模型 D) 实体-联系模型
5. 下面列出的条目中, 哪些是数据库技术的主要特点 (A)
I. 数据共享度高
II. 数据的冗余小
III. 较高的数据独立性
IV. 程序的标准化
A) I、II 和 III B) I 和 II C) I、II 和 IV D) 都是
6. 数据库管理系统 DBMS 中用来定义逻辑模式、内模式和外模式的语言是 (C)
A) DML B) C C) DDL D) Basic
7. 层次模型的典型代表是 (A) 数据库管理系统。
A) IMS B) IBM C) DBTG D) ATM
8. 负责数据库系统的正常运行, 承担创建、监控和维护数据库结构责任的是 (C)
A) 应用程序员 B) 终端用户
C) 数据库管理员 D) 数据库管理系统的软件设计员
9. 在以下各条叙述中, 正确的叙述有几条 (B)
1) 数据库避免了一切数据重复
2) 数据库减少了数据冗余
3) 数据库中, 如果逻辑模式改变, 则需将与其有关的外模式做相应改变, 否则应用程序需改写
4) 数据库中的内模式如有改变, 逻辑模式可以不变
A) 1 B) 2 C) 3 D) 4
10. 在三级模式之间引入两级映像, 其主要功能之一是 (A)
A) 使数据与程序具有较高的独立性
B) 使系统具有较高的通道能力
C) 保持数据与程序的一致性
D) 提高存储空间的利用率
11. 视图对应于数据库系统三级模式结构中的 (A)
A) 外模式 B) 逻辑模式 C) 内模式 D) 都不对应
12. 对数据库中数据可以进行查询、插入、删除、修改, 这是因为数据库管理系统提供了 (B)
A) 数据库定义功能 B) 数据库操纵功能
C) 数据库维护功能 D) 数据库控制功能
13. 1970 年, 美国 IBM 公司研究员 E.F.Codd 提出了数据库的 (C)
A) 层次模型 B) 网状模型 C) 关系模型 D) 实体联系模型
14. 具有数据冗余度小、数据共享、以及较高数据独立性等特征的系统是 (B)
A) 文件系统 B) 数据库系统 C) 管理系统 D) 高级程序
15. 在概念设计中的事物称为 (A)
A) 实体 B) 记录 C) 对象 D) 结点
16. 层次模型表达实体及实体之间联系用的数据结构是 (D)

- A) 网状 B) 表 C) 有向图 D) 树型
17. DB 是 (A)
A) 数据库 B) 数据库管理系统
C) 数据处理系统 D) 数据库系统
18. DBMS 是 (B)
A) 数据库 B) 数据库管理系统
C) 数据处理系统 D) 数据库系统
19. 已知在一个工厂中有多个车间，每一个车间有多名职工，工厂的产品要经过多个车间的多道工序加工。具体来说，一个产品要经过多个工人加工，一位工人要加工多个产品。问：工厂与车间之间属于 (A) 联系，车间与工人之间属于 (A) 联系，工人与产品之间属于 (D) 联系。
A) 一对多 B) 一对一 C) 未知 D) 多对多
20. (D) 是存储在计算机内的有结构的数据集合。
A) 网络系统 B) 数据库系统 C) 操作系统 D) 数据库
21. 数据库系统的核心是 (D)
A) 编译系统 B) 数据库 C) 操作系统 D) 数据库管理系统
22. 数据库 (DB)、数据库系统 (DBS) 和数据库管理系统 (DBMS) 三者之间的关系是 (A)
A) DBS 包括 DB 和 DBMS B) DBMS 包括 DB 和 DBS
C) DB 包括 DBS 和 DBMS D) DBS 就是 DB，也就是 DBMS
23. 数据库管理系统 (DBMS) 是 (B)
A) 一组硬件 B) 一组软件 C) 既有硬件，也有软件
24. 在 DBS 中，DBMS 和 OS 之间关系是 (B)
A) 相互调用 B) DBMS 调用 OS
C) OS 调用 DBMS D) 并发运行
25. 数据库系统是在 (A) 的基础上发展起来的。
A) 文件系统 B) 应用程序系统
C) 数据库管理系统 D) 编译系统
26. 三级模式间存在两种映射，它们是 (A)
A) 逻辑模式与外模式间，逻辑模式与内模式间
B) 外模式与内模式间，外模式与逻辑模式间
C) 内模式与外模式间，内模式与逻辑模式间
27. 表达用户使用观点的数据库局部逻辑结构的模型是 (C)
A) 概念模型 B) 逻辑模型
C) 外部模型 D) 内部模型

二、填空题

1. 数据管理技术的发展经历了如下四个阶段：人工管理阶段、文件系统阶段、数据库阶段和高级数据库阶段。
2. 用二维表结构表示的实体及实体间联系的数据模型称为关系模型。
3. 两个实体集之间的联系有三种，分别是一对一联系，一对多联系和多对多联系。
4. 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，那么实体集 E1 和 E2 的联系称为一对一联系。

5. 数据库系统的三级模式、两级映像结构使数据库系统达到了高度的数据独立性。

第二章 数据库设计和 ER 模型

要求、目标：

了解和掌握数据库应用系统设计的全过程，掌握 ER 模型和关系模型的基本概念，掌握概念设计中 ER 模型的设计方法，掌握逻辑设计中 ER 模型向关系模型转换的方法。

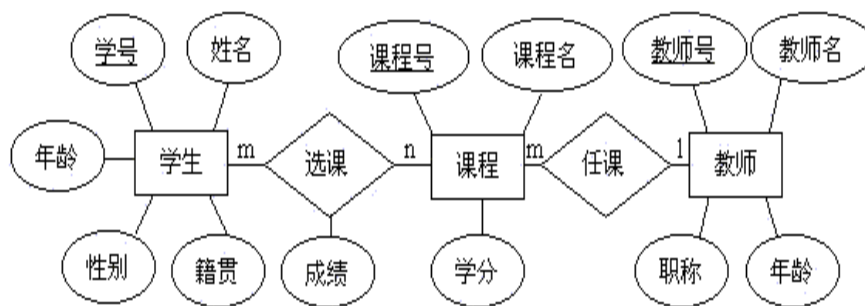
一、数据库系统生存期

1. 数据库系统生存期：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。
2. 数据库系统生存期分七个阶段：规划、需求分析、概念设计、逻辑设计、物理设计、实现、运行维护。
3. 规划阶段三个步骤：系统调查、可行性分析、确定数据库系统总目标。
4. 需求分析阶段：主要任务是系统分析员和用户双方共同收集数据库系统所需要的信息内容和用户对处理的需求，并以需求说明书的形式确定下来。
5. 概念设计阶段：产生反映用户单位信息需求的概念模型。与硬件和 DBMS 无关。
6. 逻辑设计阶段：将概念模型转换成 DBMS 能处理的逻辑模型。外模型也将在此阶段完成。
7. 物理设计阶段：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。数据库的物理结构主要指数据库的存储记录格式、存储记录安排和存取方法。
8. 数据库的实现：包括定义数据库结构、数据装载、编制与调试应用程序、数据库试运行。

二、ER 模型的基本概念 ER 模型的基本元素是：实体、联系和属性。

2. 实体：是一个数据对象，指应用中可以区别的客观存在的事物。
实体集：是指同一类实体构成的集合。
实体类型：是对实体集中实体的定义。
一般将实体、实体集、实体类型统称为实体。
3. 联系：表示一个或多个实体之间的关联关系。
联系集：是指同一类联系构成的集合。
联系类型：是对联系集中联系的定义。
一般将联系、联系集、联系类型统称为联系。
4. 同一个实体集内部实体之间的联系，称为一元联系；两个不同实体集实体之间的联系，称为二元联系，以此类推。
5. 属性：实体的某一特性称为属性。在一个实体中，能够惟一标识实体的属性或属性集称为实体标识符。
6. ER 模型中，方框表示实体、菱形框表示联系、椭圆形框表示属性、实体与联系、实体与其属性、联系与其属性之间用直线连接。实体标识符下画横线。联系的类型要在直线上标注。注意：联系也有可能存在属性，但联系本身没有标识符。

例：假设一个学生可选多门课程，而一门课程又有多个学生选修，一个教师可讲多门课程，一门课程至多只有一个教师讲授。ER 图如下：



7. 概念设计三个步骤：设计局部 ER 模型、设计全局 ER 模型和全局 ER 模型的优化。

三、关系模型的基本概念

1. 关系模型的定义：用二维表格表示实体集，用关键码表示实体之间联系的数据模型。

2. 在关系模型中，字段称为属性，字段值称为属性值，记录类型称为关系模式。记录称为元组，元组的集合称为关系或实例。有时习惯称关系为表或表格，元组为行，属性为列。关系中属性个数称为元数，元组个数称为基数。

3. 关键码（简称键）：由一个或多个属性组成。

4. 超键：在关系中能惟一标识元组的属性集称为关系模式的超键。

5. 候选键：不含有多余属性的超键。

6. 主键：用户选作元组标识的候选键。一般如不加说明，键是指主键。

7. 外键：如果模式 R 中属性集 K 是其他模式的主键，那么 K 在模式 R 中称为外键。

8. 值域：关系中每一个属性都有一个取值范围，称为属性的值域。每一个属性对应一个值域，不同的属性可对应于同一值域。

9. 关系的定义：关系是一个属性数目相同的元组的集合。

10. 关系的性质：关系是一种规范化了的二维表格。

1) 关系中每一个属性值都是不可分解的；

2) 关系中不允许出现重复元组；

3) 关系没有行序；

4) 元组中的属性在理论上也是无序的，但使用时按习惯考虑列的顺序。

11. 关系数据库中的数据与更新操作必须遵循三类完整性规则：实体完整性规则、参照完整性规则、用户定义的完整性规则。

12. 实体完整性规则：要求关系中元组在组成主键的属性上不能有空值。

13. 参照完整性规则：如果属性集 K 是关系模式 R1 的主键，K 也是关系模式 R2 的外键，那么在 R2 关系中，K 的取值只允许两种可能，或者为空值，或者等于 R1 关系中某个主键值。这条规则的实质是“不允许引用不存在的实体”。其中，R1 称为参照关系；R2 称为依赖关系。

注意：这条规则在具体使用时，有三点变通：

1) 外键和相应的主键可以不同名，只要定义在相同值域上即可；

2) R1 和 R2 可以是同一个关系模式，此时表示了同一个关系中不同元组之间的联系；

3) 外键值是否允许空，应视具体问题而定。

14. 用户定义的完整性规则：用户针对具体的数据约束，设置的完整性规则，由系统来检验实施。

四、ER 模型到关系模型的转换

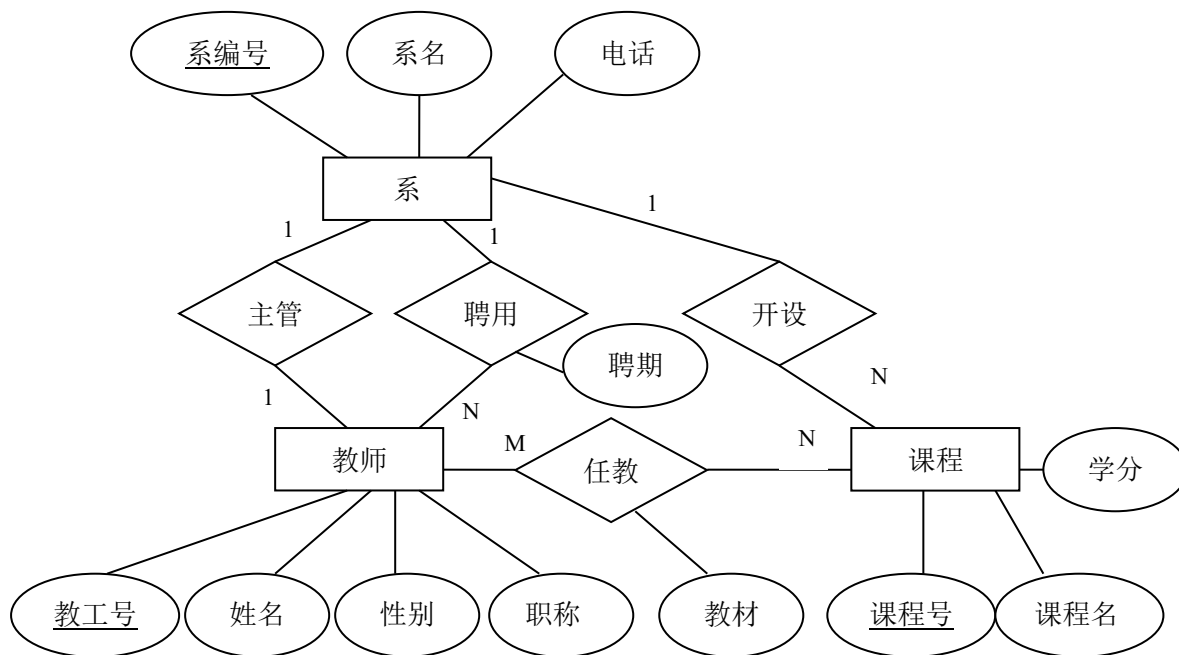
1. ER 图转换成关系模式集的算法：

1) 实体类型的转换：将每个实体类型转换成一个关系模式，实体的属性即为关系模式的属性，实体标识符即为关系模式的键。

2) 联系类型的转换：主要掌握二元联系类型的转换。

- 若实体间联系是 1: 1，可以在两个实体类型转换成的关系模式中任意一个关系模式的属性中加入另一个关系模式的键（作为外键）和联系类型的属性。
- 若实体间联系是 1: N，则在 N 端实体类型转换成的关系模式中加入 1 端实体类型的键（作为外键）和联系类型的属性。
- 若实体间联系是 M: N，则将联系类型也转换成关系模式，其属性为两端实体类型的键（作为外键）加上联系类型的属性，而键为两端实体键的组合。

例：下面是教学管理的一个可能的 ER 图。图中，有三个实体类型：系、教师和课程；有四个联系类型：主管、聘用、开设和任教。根据转换算法，把该图转换成关系模式集的步骤如下：



教学管理的 ER 图

第一步：把三个实体类型转换成三个关系模式（注意关系模式的表示方法）：

系（系编号，系名，电话）

教师（教工号，姓名，性别，职称）

课程（课程号，课程名，学分）

第二步：对于 1: 1 联系，可以在“系”模式中加入教工号（教工号为外键）。

对于 1: N 联系“聘用”，可以在“教师”模式中加入系编号和聘期两个属性（系编号为外键）；

对于 1: N 联系“开设”，可以在“课程”模式中加入系编号（系编号为外键）。

这样第一步得到的三个模式改变成如下形式（注意：外键通常使用波浪线表示）：

系（系编号，系名，电话，主管人的教工号）

教师（教工号，姓名，性别，职称，系编号，聘期）

课程（课程号，课程名，学分，系编号）

第三步：对于 M：N 联系“任教”，则生成一个新的关系模式：

任教（教工号，课程号，教材）

这样，转换成的四个关系模式如下：

系（系编号，系名，电话，主管人的教工号）

教师（教工号，姓名，性别，职称，系编号，聘期）

课程（课程号，课程名，学分，系编号）

任教（教工号，课程号，教材）

2. 采用 ER 模型的逻辑设计步骤

1) 导出初始关系模式集：把概念设计的结果（即全局 ER 模型）转换成初始关系模式集。

2) 规范化处理

3) 模式评价

4) 模式修正

5) 设计外模式

第二章 复习题

一、单项选择题

1. 下面对于关系的叙述中，哪个是不正确的？（ C ）
 - A) 关系中的每个属性是不可分解的
 - B) 在关系中元组的顺序是无关紧要的
 - C) 任意的一个二维表都是一个关系
 - D) 每一个关系只有一种记录类型
2. 设属性 A 是关系 R 的主键，则属性 A 不能取空值。这是（ A ）
 - A) 实体完整性规则
 - B) 参照完整性规则
 - C) 用户定义完整性规则
 - D) 域完整性规则
3. 对关系数据库来讲，下面（ C ）说法是错误的
 - A) 外键和相应的主键可以不同名，只要定义在相同值域上即可
 - B) 不同的属性可对应于同一值域
 - C) 行的顺序可以任意交换，但列的顺序不能任意交换
 - D) 关系中的任意两个元组不能完全相同
4. 数据库的物理设计是为一个给定的逻辑结构选取一个适合应用环境的（ B ）的过程，包括确定数据库在物理设备上的存储结构和存取方法。
 - A) 逻辑结构
 - B) 物理结构
 - C) 概念结构
 - D) 层次结构
5. 在关系中，能惟一标识元组的属性集称为（ B ）。
 - A) 外键
 - B) 超键
 - C) 域
 - D) 元组
6. 在数据库逻辑设计中，当将 E-R 图转换为关系模式时，下面的做法（ B ）是不正确的
 - A) 一个实体类型转换为一个关系模式

- B) 一个联系类型转换为一个关系模式
C) 由实体类型转换成的关系模式的主键是该实体类型的主键
7. 数据库设计的概念设计阶段, 表示概念结构的常用方法和描述工具是 (D)
A) 层次分析法和层次结构图
B) 数据流程分析法和数据流程
C) 结构分析法和模块结构图
D) 实体—联系方法和 E—R 图
8. 关系数据库中, 实现表与表之间的联系是通过 (B)
A) 实体完整性规则 B) 参照完整性规则
C) 用户定义的完整性 D) 值域
9. 有两个实体集, 并且它们之间存在着一个 M: N 联系, 那么按照 E—R 模型转换成关系数据库的规则, 这个 E—R 结构转换成表的个数为 (C)
A) 1 B) 2 C) 3 D) 4
10. 把 E—R 图转换为关系模型的过程, 属于数据库设计的 (B)
A) 概念设计 B) 逻辑设计 C) 需求分析 D) 物理设计
11. 在数据库设计的 (D) 阶段, 进行应用程序的编制和调试。
A) 概念设计 B) 逻辑设计 C) 物理设计 D) 实现
- 12.

EMP

雇员号	雇员名	部门号	工资
001	张山	02	2000
010	王宏达	01	1200
056	马林生	02	1000
101	赵敏	04	1500

DEPT

部门号	部门名	地址
01	业务部	1 号楼
02	销售部	2 号楼
03	服务部	3 号楼
04	财务部	4 号楼

- 在雇员信息表关系 EMP 中, 哪个属性是外键? (C)
A) 雇员号 B) 雇员名 C) 部门号 D) 工资

二、填空题

- 表示实体及其联系的方法为 ER 图, 基本图素包括 方框、菱形框和椭圆形框。习惯上实体用方框表示, 实体的属性用椭圆形框表示, 联系用菱形框表示。
- 在关系模型中, 二维表的列称为属性, 二维表的行称为元组。
- 假设班级和班长之间的联系是一对一联系, 则班级和学生之间是一对多联系; 另外, 学生和课程之间的联系是多对多联系。
- 关系模型的三种完整性规则是实体完整性规则、参照完整性规则和用户定义的完整性规则。
- 已知系 (系编号, 系名称, 系主任, 电话, 地点) 和学生 (学号, 姓名, 性别, 入学日期, 专业, 系编号) 两个关系, 系关系的主键是系编号, 学生关系的主键是学号, 外键是系编号。
- 数据库系统生存期分七个阶段: 规划、需求分析、概念设计、逻辑设计、物

理设计、实现、运行维护。

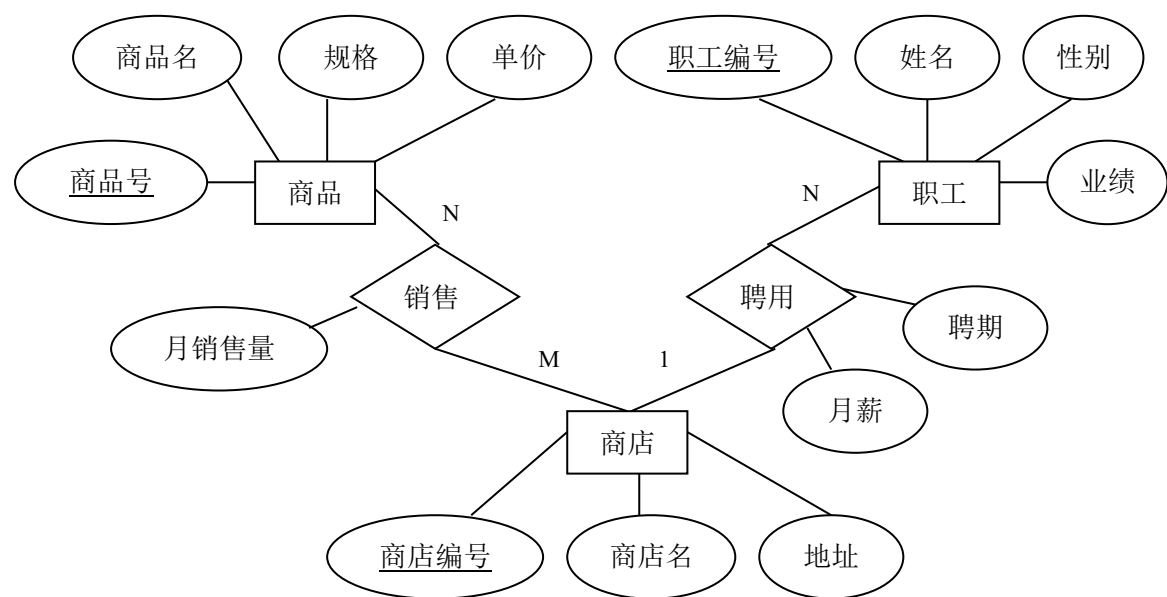
7. 关系中属性个数称为元数，元组个数称为基数。

三、综合题

1. 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，每个商店每销售一种商品，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。

试画出 ER 图，并在图上注明属性、联系的类型。再转换成关系模式集，并指出每个关系模式的主键和外键。



关系模式集：

商店（商店编号，商店名，地址）

商品（商品号，商品名，规格，单价）

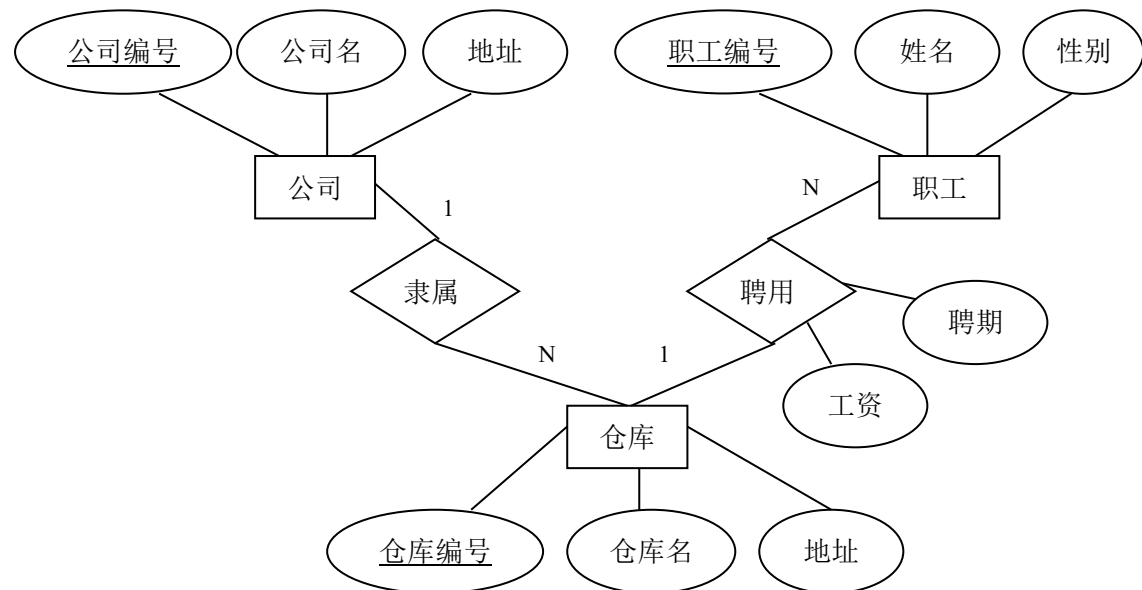
职工（职工编号，姓名，性别，业绩，商店编号，月薪，聘期）

销售（商店编号，商品编号，月销售量）

2. 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。

试画出 ER 图，并在图上注明属性、联系的类型。再转换成关系模式集，并指出每个关系模式的主键和外键。



关系模式集：

公司（公司编号，公司名，地址）

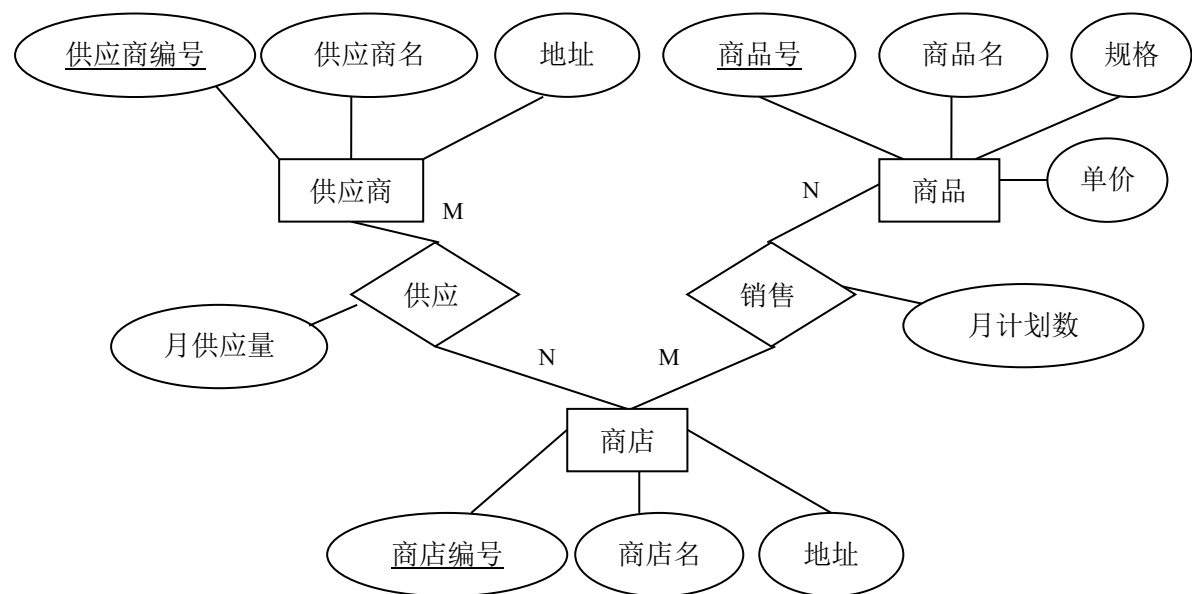
仓库（仓库编号，仓库名，地址，公司编号）

职工（职工编号，姓名，性别，仓库编号，聘期，工资）

3. 设某商业集团数据库中有三个实体集。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，属性有商店编号、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，供应商供应商品有月供应量；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，商店销售商品有月计划数。

试画出 ER 图，并在图上注明属性、联系的类型。再转换成关系模式集，并指出每个关系模式的主键和外键。



关系模式集：

供应商（供应商编号，供应商名，地址）

商店（商店编号，商店名，地址）

商品（商品号，商品名，规格，单价）

供应（供应商编号，商店编号，月供应量）

销售（商店编号，商品号，月计划数）

第三章 关系模式设计理论

要求、目标：

了解关系数据库规范化理论及其在数据库设计中的作用，重点是函数依赖和范式，要求掌握这些概念并能运用它们来进行模式分解。

一、关系模式的设计准则

1. 数据冗余：同一个数据在系统中多次重复出现。
2. 关系模式设计不当引起的异常问题：数据冗余、操作异常（包括修改异常、插入异常和删除异常）
3. 关系模式的非形式化设计准则

1) 关系模式的设计应尽可能只包含有直接联系的属性, 不要包含有间接联系的属性。也就是, 每个关系模式应只对应于一个实体类型或一个联系类型。

2) 关系模式的设计应尽可能使得相应关系中不出现插入异常、删除和修改等操作异常现象。

3) 关系模式的设计应尽可能使得相应关系中避免放置经常为空值的属性。

4) 关系模式的设计应尽可能使得关系的等值连接在主键和外键的属性上进行, 并且保证以后不会生成额外的元组。

4. 习惯使用的一些符号:

1) 英文字母表首部的大写字母 “A, B, C, ...” 表示单个的属性。

2) 英文字母表尾部的大写字母 “..., U, V, W, X, Y, Z” 表示属性集。

3) 大写字母 R 表示关系模式, 小写字母 r 表示其关系。

4) 关系模式的简化表示方法: $R(A, B, C, \dots)$ 或 $R(ABC\dots)$

5) 属性集 X 和 Y 的并集简写为 XY。

二、函数依赖

1. 函数依赖 (FD) 的定义: 设有关系模式 $R(U)$, X 和 Y 是属性集 U 的子集, 函数依赖是形成 $X \rightarrow Y$ 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 $X \rightarrow Y$ 在关系模式 $R(U)$ 中成立。

说明: 1) $t[X]$ 表示元组 t 在属性集 X 上的值, 其余类同。

2) $X \rightarrow Y$ 读作 “X 函数决定 Y” 或 “Y 函数依赖于 X”。

3) FD 是对关系模式 R 的一切可能的关系 r 定义的。对于当前关系 r 的任意两个元组, 如果 X 值相同, 则要求 Y 值也相同, 即有一个 X 值就有一个 Y 值与之对应, 或者说 Y 值由 X 值决定。

例: 设关系模式 $R(ABCD)$, 在 R 的关系中, 属性值间有这样的联系: A 值与 B 值有一对多联系; C 值与 D 值之间有一对一联系。试根据这些规则写出相应的函数依赖。

$B \rightarrow A \quad C \rightarrow D \quad D \rightarrow C$

2. 如果 $X \rightarrow Y$ 和 $Y \rightarrow X$ 同时成立, 则可记为: $X \leftrightarrow Y$

3. FD 的逻辑蕴涵: 设 F 是在关系模式 R 上成立的函数依赖的集合, $X \rightarrow Y$ 是一个函数依赖。如果对于 R 的每个满足 F 的关系 r 也满足 $X \rightarrow Y$, 那么称 F 逻辑蕴涵 $X \rightarrow Y$, 记为 $F \models X \rightarrow Y$ 。

4. 设 F 是函数依赖集, 被 F 逻辑蕴涵的函数依赖全体构成的集合, 称为函数依赖集 F 的闭包, 记为 F^+ 。即 $F^+ = \{X \rightarrow Y \mid F \models X \rightarrow Y\}$

5. FD 的推理规则 (Armstrong 公理)

设 U 是关系模式 R 的属性集, F 是 R 上成立的只涉及到 U 中属性的函数依赖集。

1) 自反性: 若 $Y \subseteq X \subseteq U$, 则 $X \rightarrow Y$ 在 R 上成立。

2) 增广性: 若 $X \rightarrow Y$ 在 R 上成立, 且 $Z \subseteq U$, 则 $XZ \rightarrow YZ$ 在 R 上成立。

3) 传递性: 若 $X \rightarrow Y$ 和 $Y \rightarrow Z$ 在 R 上成立, 则 $X \rightarrow Z$ 在 R 上成立。

6. FD 的其他五条推理规则:

1) 合并性: $\{X \rightarrow Y, X \rightarrow Z\} \models X \rightarrow YZ$

2) 分解性: $\{X \rightarrow Y, Z \subseteq Y\} \models X \rightarrow Z$

3) 伪传递性: $\{X \rightarrow Y, WY \rightarrow Z\} \models WX \rightarrow Z$

4) 复合性: $\{X \rightarrow Y, W \rightarrow Z\} \models WX \rightarrow YZ$

$$5) \{X \rightarrow Y, W \rightarrow Z\} \models X \cup (W - Y) \rightarrow YZ$$

7. 对于 FD $X \rightarrow Y$, 如果 $Y \subseteq X$, 那么称 $X \rightarrow Y$ 是一个“平凡的 FD”, 否则称为“非平凡的 FD”。通常研究非平凡 FD。

例: $X \rightarrow X$, $X \rightarrow \phi$, $\phi \rightarrow \phi$, $XY \rightarrow X$ 都是平凡函数依赖; $X \rightarrow XY$ 则是非平凡函数依赖。

8. 函数依赖是关键码概念的推广。

设关系模式 R 的属性集是 U, X 是 U 的一个子集。如果 $X \rightarrow U$ 在 R 上成立, 那么称 X 是 R 的一个超键。如果 $X \rightarrow U$ 在 R 上成立, 但对于 R 的任一真子集 X_1 都有 $X_1 \rightarrow U$ 不成立, 那么称 X 是 R 的一个候选键。在关系模式设计理论中, 键通常是指候选键。

9. 属性集的闭包

10. 设 F 是属性集 U 上的 FD 集, X 是 U 的子集, 那么 (相对于) 属性集 X 的闭包用 X^+ 表示, 它是一个从 F 集使用 FD 推理规则推出的所有满足 $X \rightarrow A$ 的属性 A 的集合: $X^+ = \{ \text{属性 } A \mid F \models X \rightarrow A \}$

11. $X \rightarrow Y$ 能用 FD 推理规则推出的充分必要条件是 $Y \subseteq X^+$, 从而避开求 F^+ , 使问题得到简化。

12. 求属性集 X 相对于 FD 集 F 的闭包 X^+ 的算法:

```

 $X^+ = X;$ 
do {old $X^+ := X^+;$ 
    for F 中每个 FD  $Y \rightarrow Z$  do
        if  $Y \subseteq X^+$  then  $X^+ := X^+ \cup Z;$ 
    }while( $X^+ \neq \text{old}X^+$ );

```

例: 属性集 U 为 ABCD, FD 集为 $\{A \rightarrow B, B \rightarrow C, D \rightarrow B\}$ 。求 A^+ 、 $(AD)^+$ 和 $(BD)^+$

$A^+ = ABC$

$(AD)^+ = ABCD$

$(BD)^+ = BCD$

13. 如果关系模式 R (U) 上的两个函数依赖集 F 和 G, 有 $F^+ = G^+$, 则称 F 和 G 是等价的函数依赖集。

三、关系模式的分解特性

1. 关系模式的分解:

设有关系模式 R (U), 属性集为 U, 而 R_1, R_2, \dots, R_k 都是 U 的子集, 并且有 $R_1 \cup R_2 \cup \dots \cup R_k = U$ 。关系模式 R_1, R_2, \dots, R_k 的集合用 ρ 表示, $\rho = \{R_1, R_2, \dots, R_k\}$ 。用 ρ 代替 R 的过程称为关系模式的分解。这里 ρ 称为 R 的一个分解, 也称为数据库模式。

一般把上述的 R 称为泛关系模式, R 对应的当前值称为泛关系。数据库模式 ρ 对应的当前值称为数据库实例, 它由数据库模式中的每一个关系模式的当前值组成。我们用 $\sigma = \langle r_1, r_2, \dots, r_k \rangle$ 表示。

因此, 在计算机中数据并不是存储在泛关系 r 中, 而是存储在数据库 σ 中。

2. σ 和 r 是否等价, 即是否表示同样的数据。这个问题用“无损分解”特性表示。

在模式 R 上有一个 FD 集 F, 在 ρ 的每一个模式 R_i 上有一个 FD 集 F_i , 那么 $\{F_1, F_2, \dots, F_k\}$ 与 F 是否等价。这个问题用“保持依赖”特性表示。

四、范式

1. 范式：衡量关系模式好坏的标准。
2. 数据库设计中最常用的是 3NF 和 BCNF。
3. 第一范式 (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 是第一范式的模式。满足 1NF 的关系称为规范化的关系，否则称为非规范化的关系。1NF 是关系模式应具备的最起码的条件。
4. 局部依赖和完全依赖：对于 FD $W \rightarrow A$ ，如果存在 $X \subset W$ 有 $X \rightarrow A$ 成立，那么称 $W \rightarrow A$ 是局部依赖 (A 局部依赖于 W)；否则称 $W \rightarrow A$ 是完全依赖。
5. 主属性和非主属性：如果 A 是关系模式 R 的候选键中的属性，那么称 A 是 R 的主属性；否则称 A 是 R 的非主属性。
6. 第二范式 (2NF)：如果关系模式是 1NF，且每个非主属性完全函数依赖于候选键，那么称 R 是第二范式 (2NF) 的模式。
7. 分解成 2NF 模式集的算法：
设关系模式 R (U)，主键是 W，R 上还存在 FD $X \rightarrow Z$ ，并且 Z 是非主属性和 $X \subset W$ ，那么 $W \rightarrow Z$ 就是一个局部依赖。此时应把 R 分解成两个模式：
R1 (XZ)，主键是 X；
R2 (Y)，其中 $Y = U - Z$ ，主键仍是 W，外键是 X (参照 R1)。
如果 R1 和 R2 还不是 2NF，则重复上述过程，一直到数据库模式中的每一个关系模式都是 2NF 为止。
8. 如果 $X \rightarrow Y$ ， $Y \rightarrow A$ ，且 $Y \not\rightarrow X$ 和 $A \not\subseteq Y$ ，那么称 $X \rightarrow A$ 是传递依赖 (A 传递依赖于 X)。
9. 第三范式 (3NF)：如果关系模式 R 是 2NF，且每个非主属性都不传递依赖于 R 的候选键，那么称 R 是第三范式 (3NF) 的模式。
10. 分解成 3NF 模式集的算法：
设关系模式 R (U)，主键是 W，R 上还存在 FD $X \rightarrow Z$ 。并且 Z 是非主属性， $Z \not\subseteq X$ ，X 不是候选键，这样 $W \rightarrow Z$ 就是一个传递依赖。此时应把 R 分解成两个模式：
R1 (XZ)，主键是 X；
R2 (Y)，其中 $Y = U - Z$ ，主键仍是 W，外键是 X (参照 R1)。
如果 R1 和 R2 还不是 3NF，则重复上述过程，一直到数据库模式中的每一个关系模式都是 3NF 为止。
11. 如果 R 是 3NF 模式，那么 R 也是 2NF 模式。如果 R 是 2NF 模式，那么 R 也是 1NF 模式。
12. BC 范式 (BCNF)：如果关系模式 R 是 1NF，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。
13. 如果 R 是 BCNF 模式，那么 R 也是 3NF 模式。
14. 分解成 BCNF 模式集的算法能保持无损分解，但不一定能保持 FD 集。而分解成 3NF 模式集的算法既能保持无损分解，又能保持 FD 集。
15. 关系模式由 1NF 分解为 2NF，消除了非主属性对键的局部函数依赖；由 2NF 分解为 3NF，消除了非主属性对键的传递函数依赖；而 BCNF 则消除了每一属性对键的传递函数依赖。
16. 关系模式设计理论主要用于数据库的逻辑设计过程中。

第三章 复习题

一、单项选择题

1. 由于关系模式设计不当所引起的插入异常指的是 (B)
 - A) 两个事务并发地对同一关系进行插入而造成数据库不一致
 - B) 由于键值的一部分为空而不能将有用的信息作为一个元组插入到关系中
 - C) 未经授权的用户对关系进行了插入
 - D) 插入操作因为违反完整性约束条件而遭到拒绝
2. 下面有关模式分解的叙述中, 不正确的是 (D)
 - A) 若一个模式分解保持函数依赖, 则该分解一定具有无损连接性
 - B) 若要求分解保持函数依赖, 那么模式分解可以达到 3NF, 但不一定能达到 BCNF
 - C) 若要求分解既具有无损连接性, 又保持函数依赖, 则模式分解可以达到 3NF, 但不一定能达到 BCNF
 - D) 若要求分解具有无损连接性, 那么模式分解一定可以达到 BCNF
3. 下述哪一条不是由于关系模式设计不当而引起的 (B)
 - A) 数据冗余
 - B) 丢失修改
 - C) 插入异常
 - D) 修改异常
4. 根据数据库规范化理论, 下面命题中正确的是 (D)
 - A) 若 $R \in 2NF$, 则 $R \in 3NF$
 - B) 若 $R \in 3NF$, 则 R 不属于 BCNF
 - C) 若 $R \in 3NF$, 则 $R \in BCNF$
 - D) 若 $R \in BCNF$, 则 $R \in 3NF$
5. 若关系模式 $R \in 3NF$, 则下面最正确的说法是 (C)
 - A) 某个主属性不传递依赖于码
 - B) 某个非主属性不部分依赖于码
 - C) 所有的非主属性都不传递依赖于码
 - D) 所有的非主属性都不部分依赖于码
6. 给定关系模式 $R \langle U, F \rangle$, 其中, U 是所有属性的集合, F 是 FD 集。如果 X, Y 是 U 的子集, 且 $X \rightarrow Y \in F$, 则 X 和 Y 之间必然存在 (C)
 - A) 一对一联系
 - B) 一对多联系 (含一对一联系)
 - C) 多对一联系 (含一对一联系)
 - D) 多对多联系
7. 设 $R(U)$, 其中, U 是所有属性的集合。如果存在 U 的子集 K , 且 $K \rightarrow U$, 则 K 为 R 的 (D)
 - A) 外键
 - B) 候选键
 - C) 主键
 - D) 超键
8. 任何一个二元关系在函数依赖的范畴内必能达到 (D)
 - A) 1NF
 - B) 2NF
 - C) 3NF
 - D) BCNF
9. 在关系模式设计理论中, 如果一个关系 R 满足 1NF, 但 R 的某个非主属性传递依赖于键, 则关系 R 至多属于 (B)
 - A) 1NF
 - B) 2NF
 - C) 3NF
 - D) BCNF
10. 在一个 BCNF 关系模式中, 所有的非主属性对每一个键都是 (D)
 - A) 部分函数依赖
 - B) 平凡函数依赖
 - C) 传递函数依赖
 - D) 完全函数依赖

11. 在一个关系模式 $R(A, B, C, D)$ 中, 若各个属性间没有任何函数依赖关系, 则该模式的主属性有 (A)
- A) A, B, C, D B) R, A C) A, B D) R, A, B, C, D
12. 当下述哪一条成立时, 称 $X \rightarrow Y$ 为平凡的函数依赖 (B)
- A) $X \subseteq Y$ B) $Y \subseteq X$ C) $X \cap Y = \phi$ D) $X \cap Y \neq \phi$
13. 当关系模式 $R(A, B)$ 已属于 3NF, 下列 (B) 说法是正确的。
- A) 它一定消除了插入和删除异常
B) 仍可能存在着一一定的插入和删除异常
C) 一定属于 BCNF
D) A 和 C 都是
14. 关系模型中的关系模式至少是 (A)
- A) 1NF B) 2NF C) 3NF D) BCNF
15. 下列函数依赖中, (C) 是平凡的函数依赖。
- A) $AB \rightarrow BC$ B) $AB \rightarrow CD$ C) $AB \rightarrow A$ D) $AB \rightarrow D$
16. 下列命题中, 不正确的是 (D)
- A) 若 $X \rightarrow Y$ 在 R 上成立, 且 $Z \subseteq U$, 则 $XZ \rightarrow YZ$ 在 R 上成立。
B) 若 $X \rightarrow Y$ 和 $Y \rightarrow Z$ 在 R 上成立, 则 $X \rightarrow Z$ 在 R 上成立。
C) 若 $X \rightarrow Y$, $X \rightarrow Z$ 在 R 上成立, 则 $X \rightarrow YZ$ 在 R 上成立。
D) 若 $X \rightarrow Y$, $WY \rightarrow Z$ 在 R 上成立, 则 $WX \rightarrow Z$ 在 R 上不成立。
17. 设关系模式 $R(ABCDE)$, F 是 R 上成立的 FD 集, $F = \{AB \rightarrow C, CD \rightarrow E, DE \rightarrow B\}$, 则下列哪一项不是关系模式 R 的候选键 (D)
- A) ACD B) ABD C) AED D) AD
18. 设关系模式 $R(ABCD)$ 上 FD 集为 F , 并且 $F = \{AB \rightarrow C, C \rightarrow D, D \rightarrow A\}$, 则下列哪一项不是关系模式 R 的候选键 (B)
- A) AB B) AD C) BC D) BD

二、填空题

- 关系模式规范化过程中, 若要求分解保持函数依赖, 那么模式分解一定可以达到 3NF, 但不一定能达到 BCNF。
- 将一个关系从 1NF 规范到 2NF, 目的是消除非主属性对键的部分函数依赖, 若进一步规范到 3NF, 目的是消除非主属性对键的传递函数依赖。
- 在关系数据库的规范化设计中, 对模式分解的等价性进行评价的两条主要标准是具有无损连接性和保持函数依赖。
- 若关系为 1NF, 且它的每一非主属性都完全函数依赖于候选键, 则该关系为 2NF。
- 衡量关系模式好坏的标准称为范式。
- 满足第一范式的关系称为规范化的关系。
- 设关系模式 $R(ABCD)$, F 是 R 上成立的 FD 集, $F = \{A \rightarrow B, C \rightarrow B\}$, 则相对于 F , 关系模式 R 的候选键是 AC。

三、综合题

- 设关系模式 $R(ABCD)$, F 是 R 上成立的 FD 集, $F = \{A \rightarrow B, B \rightarrow C\}$ 。
 - 试写出属性集 BD 的闭包 $(BD)^+$ 。
 - 试写出所有左部是 B 的函数依赖 (即形为 “ $B \rightarrow ?$ ”)。

$(BD)^+ = BCD$

左部是 B 的函数依赖有: $B \rightarrow \phi$, $B \rightarrow B$, $B \rightarrow C$, $B \rightarrow BC$

2. 设关系模式 R (ABCDE) 上 FD 集为 F, 并且 $F = \{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$ 。

1) 试求 R 的候选键。

2) 试求 B^+ 的值。

R 的候选键为: A、E、BC、CD

$B^+ = BD$

3. 设关系模式 R (ABCD), F 是 R 上成立的 FD 集, $F = \{AB \rightarrow CD, A \rightarrow D\}$ 。

1) 试说明 R 不是 2NF 模式的理由。

2) 试把 R 分解成 2NF 模式集。

理由: R 的候选键是 AB, 则非主属性为 C 和 D, 并且 $AB \rightarrow D$ 成立。而已知 $A \rightarrow D$, 因此 $AB \rightarrow D$ 为非主属性 D 对候选键的局部依赖。

R 分解为: R1 (AD) 主键是 A;

R2 (ABC) 主键是 AB, 外键是 A。

4. 设关系模式 R (ABCD), F 是 R 上成立的 FD 集, $F = \{C \rightarrow B, B \rightarrow A\}$ 。

1) 试说明 R 不是 3NF 模式的理由。

2) 试把 R 分解成 3NF 模式集。

理由: R 的候选键是 C, 则非主属性为 A 和 B。因为 $C \rightarrow B$, $B \rightarrow A$, 则 $C \rightarrow A$ 为非主属性 A 对候选键的传递依赖。

R 分解为: R1 (CB) 主键是 C, 外键是 B;

R2 (AB) 主键是 B。

5. 设有关系模式 R (职工编号, 日期, 日营业额, 部门名, 部门经理), 该模式记录了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。

答: 1) R 的基本 FD: (职工编号, 日期) \rightarrow 日营业额, 职工编号 \rightarrow 部门名, 部门名 \rightarrow 部门经理

R 的关键码: (职工编号, 日期)

2) R 不是 2NF 的理由: R 的候选键是 (职工编号, 日期), 则部门名和部门经理为非主属性, 并且 (职工编号, 日期) \rightarrow 部门名和 (职工编号, 日期) \rightarrow 部门经理成立。而职工编号 \rightarrow 部门名, 部门名 \rightarrow 部门经理, 因此职工编号 \rightarrow 部门经理, 因此 (职工编号, 日期) \rightarrow 部门名为非主属性部门名对候选键的局部依赖, (职工编号, 日期) \rightarrow 部门经理为非主属性部门经理对候选键的局部依赖

R 分解为: R1 (职工编号, 部门名, 部门经理) 主键是职工编号;

R2 (职工编号, 日期, 日营业额) 主键是 (职工编号, 日期), 外键是职工编号。

3) R 分解为: R11 (职工编号, 部门名) 主键是职工编号, 外键是部门名 (参照 R12);

R12（部门名，部门经理）主键是部门名；

R2（职工编号，日期，日营业额）主键是（职工编号，日期），外键是职工编号（参照 R11）。

6. 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。

答：1) R 的基本 FD：（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：（运动员编号，比赛项目）

2) R 不是 2NF 的理由：R 的候选键是（运动员编号，比赛项目），则比赛类别和比赛主管为非主属性，并且（运动员编号，比赛项目）→比赛类别和（运动员编号，比赛项目）→比赛主管成立。而比赛项目→比赛类别，比赛类别→比赛主管，因此比赛项目→比赛主管成立，因此（运动员编号，比赛项目）→比赛类别为非主属性比赛类别对候选键的局部依赖，（运动员编号，比赛项目）→比赛主管为非主属性比赛主管对候选键的局部依赖。

R 分解为：R1（比赛项目，比赛类别，比赛主管）主键是比赛项目；

R2（运动员编号，比赛项目，成绩）主键是（运动员编号，比赛项目），外键是比赛项目。

3) R 分解为：R11（比赛项目，比赛类别）主键是比赛项目，外键是比赛类别（参照 R12）；

R12（比赛类别，比赛主管）主键是比赛类别；

R2（运动员编号，比赛项目，成绩）主键是（运动员编号，比赛项目），外键是比赛项目（参照 R11）。

第四章 关系运算

要求、目标：

理解关系模型的运算理论，了解关系演算和查询优化，熟练掌握关系代数运算，掌握关系代数表达式的构造方法。

一、简介

1. 关系模型的三个组成部分：数据结构、数据操纵和数据完整性规则。

2. 数据结构：数据库中全部数据及其相互联系都被组织成“关系”（二维表格）的形式。关系模型基本的数据结构是关系。

3. 数据操纵：关系模型提供一组完备的高级关系运算，以支持对数据库的各种操作。关系运算分成关系代数和关系演算两类。

4. 数据完整性规则：数据库中数据必须满足实体完整性、参照完整性和用户定义的完整性等三类完整性规则。

5. 关系数据库的数据操纵语言（DML）的语句分成查询语句和更新语句两大类。查询语句用于描述用户的各种检索要求；更新语句用于描述插入、删除、修改等

操作。前者是基础。

6. 关系查询语言分两类：关系代数语言（查询操作以集合操作为基础）和关系演算语言（查询操作以谓词演算为基础）

二、关系代数

1. 关系代数中的操作可以分为两类：

1) 传统的集合操作：并、差、交、笛卡儿积（乘法）、笛卡儿积的逆运算（除法）

2) 扩充的关系操作：投影、选择、连接等。

2. 关系代数的五个基本操作：并、差、笛卡儿积、投影和选择。

3. 并：设关系 R 和 S 具有相同的模式， R 和 S 的并是由属于 R 或属于 S 的元组构成的集合，记为 $R \cup S$ 。

4. 差：设关系 R 和 S 具有相同的模式， R 和 S 的差是由属于 R 但不属于 S 的元组构成的集合，记为 $R - S$ 。

5. 笛卡儿积：设关系 R 和 S 的元数分别为 r 和 s ， R 和 S 的笛卡儿积是一个 $(r+s)$ 元的元组集合，每个元组的前 r 个分量（属性值）来自 R 的一个元组，后 s 个分量来自 S 的一个元组。若 R 有 m 个元组， S 有 n 个元组，则 $R \times S$ 有 $m \times n$ 个元组。

6. 投影：对一个关系进行垂直分割，消去某些列，并重新安排列的顺序。例：

$\pi_{3,1}(R)$ 下标也可以用属性名表示。

7. 选择：对关系做水平分割，即选取符合条件的元组。表示为： $\sigma_F(R)$ 从 R 中挑选满足公式 F 为真的元组所构成的集合。

F 中有两种成分：

1) 运算对象：常数（用引号括起来）、元组分量（属性名或列的序号）

2) 运算符：算术比较运算符（ $<$ 、 \leq 、 $>$ 、 \geq 、 $=$ 、 \neq 、也称为 θ 符）、逻辑运算符（ \wedge 、 \vee 、 \neg ）

例：两个关系 R 和 S 如下：求 $R \cup S$ 、 $R - S$ 、 $R \times S$ 、 $\pi_{C,A}(R)$ 、 $\sigma_{B>4}(R)$ 。

关系 R		
A	B	C
1	2	3
4	5	6
7	8	9

关系 S		
A	B	C
2	4	6
4	5	6

答案：

$R \cup S$		
A	B	C
1	2	3
4	5	6
7	8	9
2	4	6

$R - S$		
A	B	C
1	2	3
7	8	9

$R \times S$					
R.A	R.B	R.C	S.A	S.B	S.C
1	2	3	2	4	6
1	2	3	4	5	6
4	5	6	2	4	6
4	5	6	4	5	6
7	8	9	2	4	6
7	8	9	4	5	6

$\pi_{C, A}(R)$

C	A
3	1
6	4
9	7

$\sigma_{B > 4}(R)$

A	B	C
4	5	6
7	8	9

8. 交：设关系 R 和 S 具有相同的模式， R 和 S 的交是由属于 R 又属于 S 的元组构成的集合，记为 $R \cap S$ 。 $R \cap S = R - (R - S)$ 或 $R \cap S = S - (S - R)$

9. 连接：从关系 R 和 S 的笛卡儿积中选取属性值满足某一 θ 操作的元组，记为 $R \bowtie_{\theta} S$ 。 i 和 j 分别是关系 R 和 S 中的第 i 个和第 j 个属性名或序号。

连接是由笛卡儿积和选择操作组合而成。

如果 θ 为 “=”，该连接操作称为 “等值连接”。

例：已知关系 R 和 S ，求 $R \bowtie_{B=1} S$

关系 R			关系 S	
A	B	C	D	E
1	2	3	2	4
4	5	6	5	6
7	2	9	7	8

答案： $R \bowtie_{B=1} S$

A	B	C	D	E
1	2	3	2	4
4	5	6	5	6
7	2	9	2	4

10. 自然连接：公共属性只出现一次的等值（公共属性值全部相等）连接。

记为： $R \bowtie S$

一般自然连接使用在 R 和 S 有公共属性的情况中。如果两个关系没有公共属性，那么其自然连接就转化为笛卡儿积操作。

例：已知关系 R 和 S，求 $R \bowtie S$

关系 R

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

关系 S

B	C	D
5	7	3
4	6	2
5	7	9
5	6	3

答案：

A	B	C	D
2	4	6	2
3	5	7	3
3	5	7	9
7	4	6	2

11. 除法：设关系 R (X, Y) 和关系 S (Y, Z)，则 $R \div S$ 定义为：

$$R(X, Y) \div S(Y, Z) = \Pi_X(R) - \Pi_X((\Pi_X(R) \times \Pi_Y(S)) - R)$$

例：关系 R 是学生选修课程的情况，COURSE 表示课程情况，而操作 $R \div \text{COURSE}$ 表示至少选修 COURSE 中列出课程的学生名单。

R

S#	SNAME	C#
S1	BAO	C1
S1	BAO	C2
S1	BAO	C3
S1	BAO	C4
S2	GU	C1
S2	GU	C2
S3	AN	C2
S4	LI	C4
S4	LI	C2

COURSE

C#	CNAME
C2	OS
C4	MIS

$R \div \text{COURSE}$

S#	SNAME
S1	BAO
S4	LI

12. 关系代数表达式：由五个基本操作经过有限次复合的式子称为代数表达式。这种表达式的运算结果仍是一个关系。可以用关系代数表达式表示各种数据查询操作。

例：教学数据库中的四个关系如下：

教师关系 T (T#, TNAME, TITLE)

课程关系 C (C#, CNAME, T#)

学生关系 S (S#, SNAME, AGE, SEX)

选课关系 SC (S#, C#, SCORE)

使用关系代数表达式表达下列每个查询语句。

1) 检索学习课程号为 C2 课程的学生学号与成绩。

$\pi_{S\#, SCORE} (\sigma_{C\# = 'C2'} (SC))$ 或 $\pi_{1, 3} (\sigma_{2 = 'C2'} (SC))$

2) 检索学习课程号为 C2 课程的学生学号和姓名。

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

3) 检索至少选修 LIU 老师所授课程中一门课程的学生学号与姓名。

$\pi_{S\#, SNAME} (\sigma_{TNAME = 'LIU'} (S \bowtie SC \bowtie C \bowtie T))$

4) 检索选修课程号为 C2 或 C4 课程的学生学号。

$\pi_{S\#} (\sigma_{C\# = 'C2' \vee C\# = 'C4'} (SC))$

5) 检索至少选修课程号为 C2 和 C4 课程的学生学号。

$\pi_1 (\sigma_{1=4 \wedge 2 = 'C2' \wedge 5 = 'C4'} (SC \times SC))$

6) 检索不学 C2 课程的学生姓名与年龄。

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

7) 检索学习全部课程的学生姓名。

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

8) 检索所学课程包含学号为 S3 学生所学课程的学生学号。

$\pi_{S\#, C\#} (SC) \div \pi_{C\#} (\sigma_{S\# = 'S3'} (SC))$

总结：查询语句的关系代数表达式的一般形式是：

$\pi \dots (\sigma \dots (R \times S))$ 或 $\pi \dots (\sigma \dots (R \bowtie S))$

即首先把查询涉及到的关系取来，执行笛卡儿积或自然连接操作得到一张大的表格，然后对大表格执行水平分割（选择操作）和垂直分割（投影操作）。

但这种形式不适用于否定或全部值的查询。这时要用差或除法操作。

13. 外连接：如果 R 和 S 做自然连接时，把原该舍弃的元组也保留在新关系中，同时在这些元组新增加的属性上填上空值（Null），这种操作称为“外连接”操作。

14. 左外连接：如果 R 和 S 做自然连接时，只把 R 中原该舍弃的元组放到新关系中，那么这种操作称为“左外连接”操作。

15. 右外连接：如果 R 和 S 做自然连接时，只把 S 中原该舍弃的元组放到新关系中，那么这种操作称为“右外连接”操作。

16. 外部并：两个关系 R 和 S 做并操作时，如果它们的关系模式不同，构成的新关系的属性由 R 和 S 的所有属性组成（公共属性只取一次），新关系的元组由属于 R 或属于 S 的元组构成，同时元组在新增加的属性上填上空值，那么这种操作称为“外部并”操作。

三、关系演算

关系演算又可分为元组关系演算和域关系演算，前者以元组为变量，后者以

属性（域）为变量。

四、关系代数表达式的优化

1. 目的：提高系统效率。
2. 三条启发式规则：
 - 1) 尽可能早地执行选择操作；
 - 2) 尽可能早地执行投影操作；
 - 3) 避免直接做笛卡儿积，把笛卡儿积操作之前和之后的一连串选择和投影合并起来一起做。

第四章 复习题

一、单项选择题

1. 设关系 R 和 S 的元组个数分别为 100 和 300，关系 T 是 R 与 S 的笛卡儿积，则 T 的元组个数是（ C ）
A) 400 B) 10000 C) 30000 D) 90000
2. 设关系 R 与关系 S 具有相同的模式，则 $R - (R - S)$ 等于（ B ）
A) $R \cup S$ B) $R \cap S$ C) $R \times S$ D) $R - S$
3. 在关系代数中，从两个关系的笛卡儿积中，选取它们属性间满足一定条件的元组的操作，称为（ D ）
A) 投影 B) 选择 C) 自然连接 D) 连接
4. 设关系 R 和关系 S 的元数分别是 3 和 4，关系 T 是 R 与 S 的笛卡儿积，即： $T = R \times S$ ，则关系 T 的元数是（ A ）
A) 7 B) 9 C) 12 D) 16
5. 在关系代数中，自然连接的运算符为（ A ）
A) ∞ B) \times C) π D) σ
6. 设有关系 R, S 和 T 如下。关系 T 由关系 R 和 S 经过（ B ）操作得到。

R

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

S

A	B	C
4	1	6
2	7	1

T

A	B	C
1	2	3
3	2	4

7. 查询优化策略中最重要、最基本的一条原则是（ C ）
A) 投影运算应尽可能先做
B) 连接运算应尽可能先做
C) 选择运算应尽可能先做

D) 把投影运算和选择运算同时进行

8. 假定有两个关系 R 与 S, 其内容分别为:

R 关系

A	B	C
1	2	5
2	5	6
3	5	4

S 关系

B	C	D
2	5	17
2	5	9
5	4	1

则 $R \bowtie S$ 的运算结果为 (B)

A)

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

B)

A	B	C	D
1	2	5	17
1	2	5	9
3	5	4	1

C)

A	B	C	S. B	S. C	D
1	2	5	2	5	17
2	5	6	2	5	9
3	5	4	5	4	1

D)

A	B	C
2	5	6

9. 对表进行垂直方向的分割用的运算是 (B)

A) 交 B) 投影 C) 选择 D) 连接

10. 当关系 R 与 S 做自然连接时, 能够把 R 和 S 原该舍弃的元组放到结果关系中的操作是 (D)

A) 左外连接 B) 右外连接 C) 外部并 D) 外连接

11. 关系笛卡儿积运算记号 $R \bowtie S$ 中, (D)

A) R 为关系名, S 为属性名 B) R, S 均为属性名

B) R 为属性名, S 为关系名 D) R, S 均为关系名

12. 关系模型通常由 3 部分组成, 它们是 (B)

A) 数据结构、数据通信、数据操作

B) 数据结构、数据操作、完整性规则

C) 数据通信、数据操作、完整性规则

D) 数据结构、数据通信、完整性规则

13. 如果两个关系没有公共属性，那么其自然连接操作 (A)

A) 转化为笛卡儿积操作 B) 转化为连接操作

C) 转化为外部并操作 D) 结果为空关系

二、填空题

1. 关系模型的三个组成部分分别是：数据结构、数据操纵和数据完整性规则。

2. 已知关系 R, T, 试求下列运算结果。

R:

A	B
a	d
b	c
f	f

T:

B	C
b	b
c	c
b	g

(1) $\sigma_{A=C} (R \times T)$

A	R.B	T.B	C
b	c	c	c

(2) $R \bowtie T$

A	B	C
b	c	c

3. 给定两个关系 S1, S2,

S1:

B	C	D
2	5	7
3	6	7
2	5	8
3	6	9

S2:

B	C
2	5
3	6

求 $\pi_B (\sigma_{C>5} (S1)) \bowtie S2$

B	C
---	---

3	6
---	---

4. 关系代数中，从两个关系中找出相同元组的运算称为 交 运算。
 5. 关系代数的五个基本操作为：并、差、笛卡儿积、投影和选择。
 6. 关系运算分成关系代数和关系演算两类。

三、计算题

1. 设有关系 R 和 S:

R			S		
A	B	C	A	B	C
2	4	6	2	5	7
3	5	7	4	6	8
4	6	8	3	5	9

计算 $R \cup S$ 、 $R - S$ 、 $R \cap S$ 、 $R \times S$ 、 $\pi_{3,1}(S)$ 、 $\sigma_{C>6}(R)$ 、 $R \bowtie_{2=2} S$ 、 $R \bowtie S$

答：

$R \cup S$			$R - S$		
A	B	C	A	B	C
2	4	6	2	4	6
3	5	7	3	5	7
4	6	8			
2	5	7			
3	5	9			

$R \cap S$			$R \times S$					
A	B	C	R.A	R.B	R.C	S.A	S.B	S.C
4	6	8	2	4	6	2	5	7
			2	4	6	4	6	8
			2	4	6	3	5	9
			3	5	7	2	5	7
			3	5	7	4	6	8
			3	5	7	3	5	9
			4	6	8	2	5	7
			4	6	8	4	6	8
			4	6	8	3	5	9

$\pi_{3,1}(S)$			$\sigma_{C>6}(R)$					
C	A		A	B	C			
7	2		3	5	7			
8	4		4	6	8			
9	3							

$\sigma_{C>6}(R)$

A	B	C
3	5	7
4	6	8

$R \bowtie_{2=2} S$

R.A	R.B	R.C	S.A	S.B	S.C
-----	-----	-----	-----	-----	-----

$R \bowtie S$

3	5	7	2	5	7
3	5	7	3	5	9
4	6	8	4	6	8

A	B	C
4	6	8

2. 设有关系 R 和 S, 计算 $R \bowtie S$ 、 $R \bowtie S$ 、 $\sigma_{3=6} (R \times S)$
 $1=1$

R

A	C	D
2	5	8
7	4	1
4	5	8
3	4	9

S

B	C	D
3	5	8
4	4	1
4	1	8
6	4	1

答: $R \bowtie S$

A	C	D	B
2	5	8	3
7	4	1	4
7	4	1	6
4	5	8	3

$R \bowtie S$
 $1=1$

A	R.C	R.D	B	S.C	S.D
4	5	8	4	4	1
4	5	8	4	1	8
3	4	9	3	5	8

$\sigma_{3=6} (R \times S)$

A	R.C	R.D	B	S.C	S.D
2	5	8	3	5	8
2	5	8	4	1	8
7	4	1	4	4	1
7	4	1	6	4	1
4	5	8	3	5	8
4	5	8	4	1	8

四、设计题

1. 设教学数据库中有四个关系:

教师关系 T (T#, TNAME, TITLE)

课程关系 C (C#, CNAME, T#)

学生关系 S (S#, SNAME, AGE, SEX)

选课关系 SC (S#, C#, SCORE)

试用关系代数表达式表示下列查询语句:

- 1) 检索年龄小于 17 岁的女学生的学号和姓名。
- 2) 检索男学生所学课程的课程号和成绩。
- 3) 检索男学生所学课程的任课教师的工号和姓名。
- 4) 检索至少选修两门课程的学生学号。
- 5) 检索至少有学号为 S2 和 S4 学生选修的课程的课程号。
- 6) 检索 WANG 同学不学的课程的课程号。

- 7) 检索全部学生都选修的课程的课程号和课程名。
- 8) 检索选修课程包含 LIU 老师所授全部课程的学生学号。

答:

- 1) $\pi_{S\#, SNAME} (\sigma_{AGE < '17' \wedge SEX = '女'} (S))$
- 2) $\pi_{C\#, SCORE} (\sigma_{sex = '男'} (S \bowtie SC))$
- 3) $\pi_{T\#, TNAME} (\sigma_{sex = '男'} (S \bowtie SC \bowtie C \bowtie T))$
- 4) $\pi_1 (\sigma_{1=4 \wedge 2 \neq 5} (SC \times SC))$
- 5) $\pi_{C\#} (\sigma_{S\# = 'S2'} (SC)) \cap \pi_{C\#} (\sigma_{S\# = 'S4'} (SC))$
- 6) $\pi_{C\#} (C) - \pi_{C\#} (\sigma_{SNAME = 'WANG'} (S \times SC))$
- 7) $\pi_{c\#, CNAME} (C \bowtie (\pi_{S\#, C\#} (SC) \div \pi_{S\#} (S)))$
- 8) $\pi_{S\#, C\#} (SC) \div \pi_{C\#} (\sigma_{TNAME = 'LIU'} (C \bowtie T))$

第五章 SQL 语言

要求、目标:

SQL 语言是关系数据库的标准语言, 是本课程的一个重点。本章总的要求是: 全面掌握、深刻理解、熟练应用。

要求了解嵌入式 SQL 和存储过程, 掌握 SQL 定义语句的应用, 熟练掌握 SQL 查询语句、视图、SQL 更新语句的应用。

一、SQL 简介

1. SQL: 结构化查询语言, 关系数据库的标准语言。
2. SQL 数据库的体系结构

SQL 数据库的体系结构基本上也是三级结构, 但术语与传统的关系模型术语不同。在 SQL 中, 关系模式称为“基本表”, 存储模式称为“存储文件”, 子模式称为“视图”, 元组称为“行”, 属性称为“列”。

SQL 数据库的体系结构要点:

- 1) 一个 SQL 模式是表和约束的集合。
- 2) 一个表由行集构成, 一行是列的序列, 每列对应一个数据项。
- 3) 表有三种类型: 基本表、视图和导出表。

基本表: 实际存储在数据库中的表。

视图: 由若干基本表或其他视图构成的表的定义。

导出表: 执行了查询时产生的表。

4) 一个基本表可以跨一个或多个存储文件, 一个存储文件也可以存放一个或多个基本表。每个存储文件与外部存储器上一个物理文件对应。

5) 用户可以用 SQL 语句对基本表和视图进行查询等操作。在用户看来, 两者一样, 都是表。

6) SQL 语句可嵌在 C、FORTRAN 等主语言的程序中使用, 也可在交互环境下供终端用户使用。

3. SQL 的组成

核心 SQL 主要有四个部分：

1) 数据定义语言，即 DDL，用于定义 SQL 模式、基本表、视图、索引等结构。

2) 数据操纵语言，即 SQL DML。数据操纵分成数据查询和数据更新两类。而数据更新又分成插入、删除和修改三种操作。

3) 嵌入式 SQL 语言的使用规定

4) 数据控制语言，即 SQL DCL，这一部分包括对基本表和视图的授权、完整性规则的描述、事务控制等内容。

4. SQL 的特点

1) SQL 具有十分灵活和强大的查询功能。

2) SQL 不是一个应用开发语言，它只提供对数据库的操作功能。但 SQL 既可作为交互式语言独立使用，也可作为子语言嵌入在主语言中使用，成为应用开发语言的一部分。

3) SQL 是国际标准语言，有利于各种数据库之间交换数据，有利于程序的移植，有利于实现高度的数据独立性，有利于实现标准化。

4) SQL 完成核心功能只用 9 个英语动词，语法结构接近英语，容易学习和使用。

二、SQL 的数据定义

1. SQL 模式的创建

在 SQL 中，一个 SQL 模式定义为基本表的集合。一个 SQL 模式由模式名和模式拥有者的用户名或账号来确定，并包含模式中每一个元素（基本表、视图、索引等）的定义。创建 SQL 模式，就是定义了一个存储空间。

创建模式语法：

```
CREATE SCHEMA <模式名> AUTHORIZATION <用户名>
```

2. SQL 模式的撤销

语法：DROP SCHEMA <模式名> [CASCADE | RESTRICT]

说明：

CASCADE（级联式）方式：执行 DROP 语句时，把 SQL 模式及其下属的基本表、视图、索引等所有元素全部撤销。

RESTRICT（约束式）方式：只有当 SQL 模式中没有任何下属元素时，才能撤销 SQL 模式，否则拒绝执行 DROP 语句。

3. SQL 的基本数据类型

1) 数值型

INTEGER（或 INT） 长整数

SMALLINT 短整数

REAL 浮点数

DOUBLE PRECISION 双精度浮点数

FLOAT（n） 浮点数，精度至少为 n 位数字

NUMERIC（p，d） 定点数，有 p 位数字（不包括符号、小数点）组成，小数点后有 d 位数字，也可写成 DECIMAL（p，d）或 DEC（p，d）

2) 字符串型

CHAR（n） 长度为 n 的定长字符串

VARCHAR（n） 具有最大长度为 n 的变长字符串

3) 位串型

BIT (n) 长度为 n 的二进制位串
BIT VARYING (n) 最大长度为 n 的变长二进制位串

4) 时间型

DATE 日期, 包含年、月、日, 形为 YYYY-MM-DD
TIME 时间, 包含时、分、秒, 形为 HH: MM: SS

4. 基本表结构的创建

语法: CREATE TABLE <基本表名>
 (<列名 类型>,

 <完整性约束>,
 )

说明: 完整性约束常用以下几种子句: 主键子句 (PRIMARY KEY)、外键子句 (FOREIGN KEY) 等

NOT NULL: 表示不允许某一列的值为空值

例: 对于教学数据库中的四个关系:

教师关系 T (T#, TNAME, TITLE)

课程关系 C (C#, CNAME, T#)

学生关系 S (S#, SNAME, AGE, SEX)

选课关系 SC (S#, C#, SCORE)

创建基本表。

```
CREATE TABLE T
  (T# CHAR (4) NOT NULL,
   TNAME CHAR (8) NOT NULL,
   TITLE CHAR (10),
   PRIMARY KEY (T#))
```

```
CREATE TABLE C
  (C# CHAR (4) NOT NULL,
   CNAME CHAR (10) NOT NULL,
   T# CHAR (4),
   PRIMARY KEY (C#),
   FOREIGN KEY (T#) REFERENCES T (T#))
```

```
CREATE TABLE S
  (S# CHAR (4) NOT NULL,
   SNAME CHAR (8) NOT NULL,
   AGE SMALLINT,
   SEX CHAR (2),
   PRIMARY KEY (S#))
```

```
CREATE TABLE SC
  (S# CHAR (4),
   C# CHAR (4),
```

```

SCORE SMALLINT,
PRIMARY KEY (S#, C#),
FOREIGN KEY (S#) REFERENCES S (S#),
FOREIGN KEY (C#) REFERENCES C (C#))

```

5. 基本表结构的修改

1) 增加新的列

语法: ALTER TABLE <基本表名> ADD <列名> <类型>

2) 删除原有的列

语法: ALTER TABLE <基本表名> DROP <列名> [CASCADE | RESTRICT]

说明: CASCADE 方式表示在基本表中删除某列时, 所有引用到该列的视图和约束也要一起自动地被删除; RESTRICT 方式表示在没有视图或约束引用该属性时, 才能在基本表中删除该列, 否则拒绝删除。

3) 修改原有列的类型、宽度

语法: ALTER TABLE <基本表名> MODIFY <列名> <类型>

6. 基本表的撤销

语法: DROP TABLE <基本表名> [CASCADE | RESTRICT]

7. 索引的创建

语法: CREATE [UNIQUE] INDEX <索引名> ON <基本表名> (<列名序列>)

说明: UNIQUE 表示每个索引值对应惟一的数据记录。

索引在用户查询时会自动起作用。

一个索引键可以对应多个列。索引排列时可以升序, 也可以降序, 升序排列用 ASC 表示, 降序排列用 DESC 表示, 默认时表示升序。

8. 索引的撤销

语法: DROP INDEX <索引名>

三、SQL 的数据查询

1. SELECT 查询语句句型

在关系代数中最常用的式子是下列表达式:

$$\Pi_{A1, \dots, An}(\sigma_F(R1 \times \dots \times Rm))$$

这里, $R1, \dots, Rm$ 为关系, F 是公式, $A1, \dots, An$ 为属性。

与该表达式对应, SQL 设计了 SELECT—FROM—WHERE 句型:

SELECT $A1, \dots, An$

FROM $R1, \dots, Rm$

WHERE F

注意: 1) 在学习时, 应把 SELECT 语句和关系代数表达式联系起来考虑问题。

2) 在 WHERE 子句的条件表达式 F 中可使用下列运算符:

比较运算符: $<$ 、 $<=$ 、 $>$ 、 $>=$ 、 $=$ 、 $<>$ 或 \neq

逻辑运算符: AND、OR、NOT

集合成员资格运算符: IN、NOT IN

谓词: EXISTS、ALL、SOME

聚合函数: AVG、MIN、MAX、SUM、COUNT

F 中运算对象还可以是另一个 SELECT 语句，即 SELECT 语句可以嵌套。

2. SELECT 语句的使用技术

SELECT 使用时有三种写法：连接查询、嵌套查询和带存在量词的嵌套查询。

例：针对前面使用的教学数据库，检索学习课程号为 C2 课程的学生学号与姓名。

第一种写法（连接查询）：

```
SELECT S.S#,SNAME
FROM S,SC
WHERE S.S#=SC.S# AND C#='C2'
```

第二种写法（嵌套查询）：

```
SELECT S#, SNAME
FROM S
WHERE S# IN (SELECT S#
              FROM SC
              WHERE C#='C2')
```

第三种写法（使用存在量词的嵌套查询）：

```
SELECT S#, SNAME
FROM S
WHERE EXISTS (SELECT *
              FROM SC
              WHERE SC.S#=S.S# AND C#='C2')
```

例：对于教学数据库中四个基本表 T、C、S、SC，用 SELECT 语句表示下面查询。

1) 检索学习课程号为 C2 课程的学生学号与成绩。

```
SELECT S#, SCORE
FROM SC
WHERE C#='C2'
```

2) 检索学习课程号为 C2 课程的学生学号和姓名。

见上例

3) 检索至少选修 LIU 老师所授课程中一门课程的学生学号与姓名。

```
SELECT S.S#, SNAME
FROM S, SC, C, T
WHERE S.S#=SC.S# AND SC.C#=C.C#
      AND C.T#=T.T# AND TNAME='LIU'
```

4) 检索选修课程号为 C2 或 C4 课程的学生学号。

```
SELECT S#
FROM SC
WHERE C#='C2' OR C#='C4'
```

5) 检索至少选修课程号为 C2 和 C4 课程的学生学号。

```
SELECT X.S#
FROM SC AS X, SC AS Y
WHERE X.S#=Y.S# AND X.C#='C2' AND Y.C#='C4'
```

6) 检索不学 C2 课程的学生姓名与年龄。

```
SELECT SNAME, AGE
```

```
FROM S
WHERE S# NOT IN(SELECT S#
                  FROM SC
                  WHERE C#='C2' )
```

7) 检索学习全部课程的学生姓名。(在表 S 中找学生, 在 C 中不存在一门课程, 这个学生没有学。)

```
SELECT SNAME
FROM S
WHERE NOT EXISTS
      (SELECT *
       FROM C
       WHERE NOT EXISTS
            (SELECT *
             FROM SC
             WHERE SC.S#=S.S# AND SC.C#=C.C#))
```

8) 检索所学课程包含学号为 S3 学生所学课程的学生学号。(在 SC 表中找一个学生, 不存在 S3 学的一门课, 该学生没有学。)

```
SELECT DISTINCT S#
FROM SC AS X
WHERE NOT EXISTS
      (SELECT *
       FROM SC AS Y
       WHERE Y.S#='S3'
       AND NOT EXISTS
            (SELECT *
             FROM SC AS Z
             WHERE Z.S#=X.S# AND Z.C#=Y.C#))
```

注: DISTINCT 表示去掉重复值。

3. 聚合函数

COUNT (*): 计算元组个数

COUNT (<列名>) 对一列中的值计算个数

SUM (<列名>) 求某一列值的总和 (此列值为数值型)

AVG (<列名>) 求某一列值的平均值 (此列值为数值型)

MAX (<列名>) 求某一列值的最大值

MIN (<列名>) 求某一列值的最小值

说明: 列名前均可加 DISTINCT, 表示去掉重复值。

例 1: 求男生的总人数和平均年龄。

```
SELECT COUNT (*), AVG (AGE)
FROM S
WHERE SEX= '男'
```

例 2: 统计选修了课程的学生人数。

```
SELECT COUNT (DISTINCT S#)
FROM SC
```

4. SELECT 语句完整的句法

```

SELECT <目标表的列名或列表达式序列>
FROM <基本表名和（或）视图序列>
[WHERE <行条件表达式>]
[GROUP BY <列名序列>
    [HAVING <组条件表达式>]]
[ORDER BY <列名[ASC | DESC]>, ...]

```

说明：[]表示其中的内容根据需要可以省略。

|表示二选一。

各子句的作用：SELECT 子句用于指定输出的内容；FROM 子句用于指定要检索的数据的来源表；WHERE 子句称为“行条件子句”，用于指定对元组的选取条件；GROUP BY 子句称为“分组子句”，作用是指定对元组进行分类后再检索；HAVING 子句称为“组条件子句”，用于指定对分类后的元组的选取条件；ORDER BY 子句称为“排序子句”，作用是对检索到的元组进行排序。

整个语句的执行过程为：

- 1) 读取 FROM 子句中基本表、视图的数据，执行笛卡儿积操作。
- 2) 选取满足 WHERE 子句中给出的条件表达式的元组。
- 3) 按 GROUP 子句中指定列的值分组，同时提取满足 HAVING 子句中组条件表达式的那些组。
- 4) 按 SELECT 子句中给出的列名或列表达式求值输出。
- 5) ORDER 子句对输出的目标表进行排序，按附加说明 ASC 升序排列，或按 DESC 降序排列。

例 1：统计每门课程的学生选修人数。

```

SELECT COUNT (S#)
FROM SC
GROUP BY C#

```

例 2：统计每门课程的学生选修人数，要求只查询出选修人数超过 50 人的结果。

```

SELECT COUNT (S#)
FROM SC
GROUP BY C#
HAVING COUNT (S#) > 50

```

例 3：检索每名学生的学号和年龄，查询结果按年龄降序排列，年龄相同按学号升序排列。

```

SELECT S#, AGE>
FROM S
ORDER BY AGE DESC, S# ASC

```

5. SELECT 子句的具体规定

```

SELECT [ALL | DISTINCT] <列名或列表达式序列>|*

```

说明：1) DISTINCT 选项保证重复的行将从结构中去掉；而 ALL 选项是默认的，将保证重复的行留在结果中，一般可不必写出。

2) *是对于在 FROM 子句中命名表的所有列的简写。

3) 列表达式中允许出现+、-、*、/等运算符以及列名和常数、聚合函数等。

例 1：检索男同学选修的课程的课程号。

```

SELECT DISTINCT C#

```

```
FROM S, SC
WHERE S.S#=SC.S# AND SEX='男'
```

例 2：检索所有的学生信息。

```
SELECT *
FROM S
```

例 3：检索每个学生的学号和出生年份。

```
SELECT S#, 2008-AGE
FROM S
```

6. 列和基本表的改名操作

有时，一个基本表在 SELECT 语句中多次出现，即这个表被多次调用，为区别不同的引用，应给每次的引用加上不同的名字。当然，除此情况之外，也可以给任何一个表起别名。**但要注意：一旦给表起了别名，在该语句中，凡是引用该表名时，必须用别名而不能用原表名。**

有时，用户也可以要求输出的列名与基本表中列名不一致，可在 SELECT 子句用“旧名 AS 新名”形式改名。其中 AS 可以省略。

例：检索每个学生的姓名和出生年份，输出的列名分别为 STUDENT_NAME 和 BIRTH_YEAR。

```
SELECT S# AS STUDENT_NAME, 2008-AGE AS BIRTH_YEAR
FROM S
```

7. 条件表达式中的比较操作

1) 算术比较操作：常用运算符有 <、<=、>、>=、<>或!=

[NOT] BETWEEN...AND 用来查找(不)在给定范围内的数据，下限在前，上限在后，包括上、下限。

例：检索 18—20 岁的学生姓名。

```
SELECT SNAME
FROM S
WHERE AGE >=18 AND AGE <=20
```

或：

```
SELECT SNAME
FROM S
WHERE AGE BETWEEN 18 AND 20
```

2) 字符串的匹配操作

语法：<字符串> [NOT] LIKE <匹配模式>

说明：匹配模式中常使用两个通配符：

%：与零个或多个字符组成的字符串匹配。

_：与单个字符匹配。

例：检索姓名以字符 D 打头的学生姓名。

```
SELECT SNAME
FROM S
WHERE SNAME LIKE 'D%'
```

转义字符：为使字符串中包含特殊字符（即%和_），允许定义转义字符。转义字符紧靠特殊字符并放在它前面，表示该特殊字符被当成普通字符。使用 ESCAPE 定义转义字符。

例：LIKE 'ab%cd%' ESCAPE '\' 匹配所有以 ab%cd 开头的字符串。

LIKE 'ab\cd%' ESCAPE '\ ' 匹配所有以 ab\cd 开头的字符串。

3) 空值的比较操作

使用 IS [NOT] NULL 来比较。

例：检索年龄为空值的学生姓名。

```
SELECT SNAME
FROM S
WHERE AGE IS NULL
```

4) 集合成员资格的比较

语法：[NOT] IN (<集合>)

例 1：检索年龄分别是 18、20、21 岁的学生姓名。

```
SELECT SNAME
FROM S
WHERE AGE IN (18, 20, 21)
```

例 2：检索男同学选修课程的课程号。

```
SELECT DISTINCT C#
FROM SC
WHERE S# IN (SELECT S# FROM S WHERE SEX= '男')
```

5) 集合成员的算术比较

语法：元组 θ ALL|SOME|ANY (<集合>)

说明：ALL 表示左边那个元组与右边集合中每一个元组满足 θ 运算；

ANY 和 SOME 含义相同，表示左边那个元组与右边集合中至少一个元组满足 θ 运算。

例 1：检索学习 C2 课程的学生学号和姓名。

```
SELECT S#, SNAME
FROM S
WHERE S# = SOME (SELECT S#
FROM SC
WHERE C#= 'C2')
```

例 2：检索不学 C2 课程的学生姓名与年龄。

```
SELECT SNAME, AGE
FROM S
WHERE S# <> ALL (SELECT S#
FROM SC
WHERE C#= 'C2')
```

6) 集合空否的测试

使用[NOT]EXISTS。见前面例子。

四、数据更新

1. 数据插入

语法：1) 单元组的插入：

```
INSERT INTO <基本表名> [( <列名序列> )] VALUES (<元组值>)
```

2) 多元组的插入

```
INSERT INTO <基本表名> [( <列名序列> )] VALUES (<元组值>),
(<元组值>), ..., (<元组值>)
```

3) 查询结果的插入

INSERT INTO <基本表名> [(<列名序列>)] <SELECCT 查询语句>

4) 表的插入

INSERT INTO <基本表名 1> [(<列名序列>)] TABLE <基本表名 2>

说明：在上述各种插入语句中，如果插入的值在属性个数、顺序与基本表的结构完全一致，那么基本表后的（<列名序列>）可省略，否则必须详细列出。

2. 数据删除

语法：DELETE FROM <基本表名>

[WHERE <条件表达式>]

说明：如果省略 WHERE 子句，表示删除基本表中所有元组。

例：把课程名为 MATHS 的成绩从基本表 SC 中删除。

```
DELETE FROM SC
WHERE C# IN (SELECT C# FROM C WHERE CNAME= 'MATHS')
```

3. 数据修改

语法：UPDATE <基本表名>

SET <列名>=<值表达式>[,<列名>=<值表达式>...]

[ROW=(<元组>) [WHERE <条件表达式>]]

例 1：把 C5 课程的课程名改为 DB。

```
UPDATE C
SET CNAME= 'DB'
WHERE C#= 'C5'
```

例 2：把课程号为 C5 的元组修改为（C5，DB，T3）

```
UPDATE C
SET ROW= ( 'C5', 'DB', 'T3' )
WHERE C#= 'C5'
```

五、视图

1. 在 SQL 中，外模式一级数据结构的基本单位是视图。

2. 视图的定义：视图是从若干基本表和（或）其他视图构造出来的表。这种构造方法采用 SELECT 语句实现。在我们创建一个视图时，只是把其视图的定义存放在数据字典中，而不存储视图对应的数据，在用户使用视图时才去求对应的数据。因此，视图被称为“虚表”。

3. 视图的创建

语法：CREATE VIEW <视图名> (<列表序列>)

AS <SELECT 查询语句>

4. 视图的撤销

语法：DROP VIEW <视图名>

5. 对视图的操作

1) 视图的查询操作，与基本表一样。

2) 视图的更新操作，一般仅限于“行列子集视图”。

行列子集视图：如果视图是从单个基本表中只使用选择、投影操作导出的，并且包含了基本表的主键，那么这样的视图称为“行列子集视图”。

六、嵌入式 SQL

1. SQL 语言有两种使用方式：一种是在终端交互方式下使用，称为交互式 SQL；另一种是在主语言的程序中使用，称为嵌入式 SQL。

2. 嵌入式 SQL 的实现，有两种处理方式：一种是扩充主语言的编译程序，使之能处理 SQL 语句；另一种是采用预处理方式。目前多数系统采用后一种方式。预处理方式是先用预处理程序对源程序进行扫描，识别出 SQL 语句，并处理成主语言的函数调用形式；然后再用主语言的编译程序编译成目标程序。

3. 嵌入式 SQL 的使用规定：

- 1) 在程序中要区分 SQL 语句与主语言语句。
- 2) 允许嵌入的 SQL 语句引用主语言的程序变量（成为共享变量）。
- 3) 通过游标协调 SQL 的集合处理方式和主语言的单记录处理方式

七、存储过程

1. 存储过程是使用 SQL 语句和流程控制语句编写的模块，存储过程经编译和优化后存储在数据库服务器端的数据库中，使用时调用即可。

2. 使用存储过程的优点：

- 1) 提高运行速度；
- 2) 增强了 SQL 的功能和灵活性。
- 3) 可以降低网络的通信量；
- 4) 减轻了程序编写的工作量；
- 5) 间接实现安全控制功能。

第五章 复习题

一、单项选择题

1. 已知学生选课表 SC (S#, C#, SCORE)，其中：S#代表学生学号，C#代表课程号，SCORE 代表成绩。为了提高查询速度，对该表创建唯一索引，应该创建在哪个（组）属性上？（ A ）

A) (S#, C#) B) S# C) C# D) SCORE

2. 有如下两个关系，其中雇员信息表关系 EMP 的主键是雇员号，外键是部门号；部门信息表关系 DEPT 的主键是部门号。

EMP

雇员号	雇员名	部门号	工资
001	张山	02	2000
010	王宏达	01	1200
056	马林生	02	1000
101	赵敏	04	1500

DEPT

部门号	部门名	地址
01	业务部	1 号楼
02	销售部	2 号楼
03	服务部	3 号楼
04	财务部	4 号楼

若执行下面列出的操作，哪个操作不能成功执行？（ D ）

- A) 从 EMP 中删除行（‘010’，‘王宏达’，‘01’，1200）
- B) 在 EMP 中插入行（‘102’，‘赵敏’，‘01’，1500）
- C) 将 EMP 中雇员号= ‘056’ 的工资改为 1600 元
- D) 将 EMP 中雇员号= ‘101’ 的部门号改为 ‘05’

3. 有如下两个关系，其中雇员信息表关系 EMP 的主键是雇员号，外键是部门号；部门信息表关系 DEPT 的主键是部门号。

EMP			
雇员号	雇员名	部门号	工资
001	张山	02	2000
010	王宏达	01	1200
056	马林生	02	1000
101	赵敏	04	1500

DEPT		
部门号	部门名	地址
01	业务部	1 号楼
02	销售部	2 号楼
03	服务部	3 号楼
04	财务部	4 号楼

若执行下面列出的操作，哪个操作不能成功执行？（ C ）

- A) 从 DEPT 中删除部门号= ‘03’ 的行
- B) 在 DEPT 中插入行(‘06’, ‘计划部’, ‘6 号楼’)
- C) 将 DEPT 中部门号= ‘02’ 的部门号改为 ‘10’
- D) 将 DEPT 中部门号= ‘01’ 的地址改为 ‘5 号楼’

4. SQL 语言集数据定义、数据操纵和数据控制功能于一体，语句 INSERT、DELETE、UPDATE 实现哪类功能（ B ）

- A) 数据定义
- B) 数据操纵
- C) 数据控制

5. 下面列出的关于“视图(View)”的条目中，哪一条是不正确的（ C ）

- A) 视图是外模式
- B) 视图是虚表
- C) 使用视图可以加快查询语句的执行速度
- D) 使用视图可以简化查询语句的编写

6. 设有关系 R(A, B, C) 和 S(C, D)。与 SQL 语句

select A, B, D from R, S where R.C=S.C 等价的关系代数表达式是（ B ）

- A) $\sigma_{R.C=S.C}(\pi_{A,B,D}(R \times S))$
- B) $\pi_{A,B,D}(\sigma_{R.C=S.C}(R \times S))$
- C) $\sigma_{R.C=S.C}(\pi_{A,B}(R) \times \pi_D(S))$
- D) $\sigma_{R.C=S.C}(\pi_D(\pi_{A,B}(R) \times S))$

7. 在 SQL 语言的 SELECT 语句中，实现投影操作的是哪个子句（ A ）

- A) select
- B) from
- C) where
- D) group by

8. SQL 的 DROP INDEX 语句的作用是（ A ）

- A) 删除索引
- B) 建立索引
- C) 修改索引
- D) 更新索引

9. 在所有的关系数据库上的操作，（ D ）是核心，是其他操作的前提。

- A) 插入
- B) 删除
- C) 修改
- D) 查询

10. 设两个关系 C 和 SC 如下，它们的主键分别为 C#和 (S#, C#)。若要向关系 SC 执行下列插入操作，(A) 元组可以被插入。

C

C#	CNAME	TEACHER
c1	Database	Zhang
c2	Network	Li
c3	OS	Wu

SC

S#	C#	SCORE
993001	C1	88
991021	C3	
997345	C2	78
993069	C3	95

A) ('993001', 'C2', null)

B) ('994010', 'C4', 90)

C) ('997345', 'C2', 65)

11. 设有关系 R(A,B,C), 与 SQL 语句 SELECT DISTINCT A FROM R WHERE B=17 等价的关系代数表达式是 (C)

A) $\pi_A(R)$

B) $\sigma_{B=17}(R)$

C) $\pi_A(\sigma_{B=17}(R))$

D) $\sigma_{B=17}(\pi_A(R))$

12. SQL 语言集数据定义、数据操纵和数据控制于一体，其中 CREATE、DROP、ALTER 语句是实现哪种功能的 (A)

A) 数据定义 B) 数据操纵 C) 数据控制

13. SQL 语言具有两种使用方式，分别称为交互式 SQL 和 (C)

A) 提示式 SQL B) 多用户 SQL C) 嵌入式 SQL D) 解释式 SQL

14. 在 SQL 中，基本表的撤销可以用 (B)

A) DROP SCHEMA 命令 B) DROP TABLE 命令

C) DROP VIEW 命令 D) DROP INDEX 命令

15. 在 SQL 中，创建基本表用 (B)

A) CREATE SCHEMA 命令 B) CREATE TABLE 命令

C) CREATE INDEX 命令 D) CTEATE VIEW 命令

16. 下列说法中错误的是 (D)

A) 一个基本表可以跨一个或多个存储文件

B) 一个存储文件可存放一个或多个基本表

C) 每个存储文件与外部存储器上一个物理文件对应

D) 每一个基本表与外部存储器上一个物理文件对应

17. 实际存储在数据库中的表是 (A)

A) 基本表 B) 视图 C) 基本表和视图 D) 以上均不是

18. 在下列哪种情况下，视图可被更新 (A)

A) 行列子集视图

B) 从多个基本表中使用联接操作导出的

C) 在导出视图的过程中使用了分组操作

- D) 在导出视图的过程中使用了聚合操作
19. 在 SQL 中, SELECT 语句的“SELECT DISTINCT “表示查询结果中 (C)
- A) 属性名都不相同 B) 去掉了重复的列
C) 去掉了重复的行 D) 属性值都不相同
20. HAVING 子句中应后跟 (B)
- A) 行条件表达式 B) 组条件表达式 C) 视图序列 D) 列名序列
21. 哪种对象实质上是一个虚拟的表, 用来存储预先定义好的查询? (B)
- A) 表 B) 视图 C) 索引 D) 存储过程
22. 嵌入式 SQL 实现时, 采用预处理方式是 (C)
- 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

二、填空题

1. 在 SQL 语言中, 删除表的定义应该使用的语句是 DROP TABLE。
2. 使用 SQL 语言, 建立一个“学生”表 student, 由学号 SN, 性别 Ssex, 年龄 Sage 组成, 其中学号属性为主键。 CREATE TABLE student (SN CHAR (4), Ssex CHAR (2), Sage SMALLINT, PRIMARY KEY (SN))
3. 在 SQL 中, 建立唯一索引要用到关键字 UNIQUE。
4. 在 SQL 中, 用 DISTINCT 子句消除重复出现的元组。
5. 在 SQL 中, 用 DELETE 命令可以从表中删除行, 用 DROP TABLE 命令可以删除表。
6. 在 SQL 中, 用 UPDATE 命令可以修改表中的数据, 用 ALTER TABLE 命令可以修改表的结构。
7. 视图是从一个或几个基本表导出的表, 它与基本表不同, 是一个虚表。换句话说, 数据库中只存放视图的定义, 而不存放视图对应的数据, 这些数据仍存放在原来的基本表中。
8. SQL 的中文全称为结构化查询语言。
9. SQL 的数据操纵语言分成数据查询和数据更新两类。
10. 撤销 SQL 模式 abc, 同时把该模式下属的基本表、视图、索引等所有元素全部撤销, 完成这一操作所使用的 SQL 语句为:
DROP SCHEMA abc CASCADE

三、设计题

1. 设教学数据库中有四个关系:
教师关系 T (T#, TNAME, TITLE)
课程关系 C (C#, CNAME, T#)
学生关系 S (S#, SNAME, AGE, SEX)
选课关系 SC (S#, C#, SCORE)
试用 SQL 的查询语句表示下列查询。
 - 1) 检索年龄小于 17 岁的女学生的学号和姓名。
 - 2) 检索男学生所学课程的课程号和成绩。
 - 3) 检索男学生所学课程的任课教师的工号和姓名。

- 4) 检索至少选修两门课程的学生学号。
- 5) 检索至少有学号为 S2 和 S4 学生选修的课程的课程号。
- 6) 检索 WANG 同学不学的课程的课程号。
- 7) 检索全部学生都选修的课程的课程号和课程名。
- 8) 检索选修课程包含 LIU 老师所授全部课程的学生学号。
- 9) 统计有学生选修的课程门数。
- 10) 求选修 C4 课程的女学生的平均年龄。
- 11) 求 LIU 老师所授课程的每门课程的平均成绩。
- 12) 统计每门课程的学生选修人数 (超过 10 人的课程才统计)。要求显示课程号和人数, 查询结果按人数降序排列, 若人数相同, 按课程号升序排列。
- 13) 检索学号比 WANG 同学大, 而年龄比他小的学生姓名。
- 14) 在表 SC 中检索成绩为空值的学生学号和课程号。
- 15) 检索姓名以 L 打头的所有学生的姓名和年龄。
- 16) 求年龄大于女同学平均年龄的男学生姓名和年龄。
- 17) 求年龄大于所有女同学年龄的男学生姓名和年龄。

答: 1) SELECT S#, SNAME FROM S
WHERE AGE < 17 AND SEX = '女'

2) SELECT C#, SCORE FROM SC
FROM SC WHERE S# IN (SELECT S# FROM S WHERE
SEX = '男')

3) SELECT T.T#, TNAME FROM S, C, SC, T
WHERE S.S# = SC.S# AND SC.C# = C.C# AND T.T# = C.T#
AND SEX = '男'

4) SELECT X.S# FROM SC X, SC Y WHERE X.S# = Y.S# AND
X.C# <> Y.C#

5) SELECT C# FROM SC WHERE S# = 'S2' AND C# IN (SELECT C#
FROM SC WHERE S# = 'S4')

6) SELECT C# FROM C WHERE C# NOT IN (SELECT C# FROM SC
WHERE S# IN (SELECT S# FROM S WHERE SNAME = 'WANG'))

7) SELECT C#, CNAME FROM C WHERE NOT EXISTS (SELECT
* FROM S WHERE NOT EXISTS (SELECT * FROM SC WHERE
SC.S# = S.S# AND SC.C# = C.C#))

8) SELECT S# FROM S WHERE NOT EXISTS (SELECT *
FROM C WHERE NOT EXISTS (SELECT * FROM SC WHERE
SC.S# = S.S# AND SC.C# = C.C#) AND T# IN (SELECT T# FROM T
WHERE TNAME = 'LIU'))

9) SELECT COUNT (*) FROM SC GROUP BY C#

10) SELECT AVG (AGE) FROM S WHERE S# IN (SELECT S#
FROM SC WHERE C# = 'C4') AND SEX = '女'

11) SELECT C#, AVG (SCORE) FROM SC WHERE C# IN (SELECT
C# FROM C WHERE T# IN (SELECT T# FROM T WHERE
TNAME = 'LIU'))
GROUP BY C#

12) SELECT C#, COUNT (*) FROM SC
GROUP BY C# HAVING COUNT (*) > 10
ORDER BY 2 DESC, C#

13) SELECT SNAME FROM S WHERE S# > (SELECT S# FROM S WHERE SNAME = 'WANG') AND AGE < (SELECT AGE FROM S WHERE SNAME = 'WANG')

14) SELECT S#, C# FROM SC WHERE SCORE IS NULL

15) SELECT SNAME, AGE FROM S WHERE SNAME LIKE 'L%'

16) SELECT SNAME, AGE FROM S WHERE SEX = '男' AND AGE > (SELECT AVG (AGE) FROM S WHERE SEX = '女')

17) SELECT SNAME, AGE FROM S WHERE SEX = '男' AND AGE > ALL (SELECT AGE FROM S WHERE SEX = '女')

2. 试用 SQL 更新语句表达对上一题的教学数据库关系 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) 在 SC 中删除尚无成绩的选课元组。
- 3) 把选修 LIU 老师课程的女同学选课元组全部除去。
- 4) 把 MATHS 课不及格的成绩全改为 60 分。
- 5) 把低于所有课程总平均成绩的女同学成绩提高 5%。
- 6) 在表 SC 中, 当某个成绩低于全部课程的平均成绩时, 提高 5%。

答: 1) INSERT INTO C VALUES ('C8', 'VC++', 'BAO')

2) DELETE FROM SC WHERE SCORE IS NULL

3) DELETE FROM SC WHERE S# IN (SELECT S# FROM S WHERE SEX = '女') AND S# IN (SELECT S# FROM SC WHERE C# IN (SELECT C# FROM C WHERE T# IN (SELECT T# FROM T WHERE TNAME = 'LIU')))

4) UPDATE SC SET SCORE = 60 WHERE C# IN (SELECT C# FROM C WHERE CNAME = 'MATHS') AND SCORE < 60

5) UPDATE SC SET SCORE = SCORE + SCORE * 0.05 WHERE S# IN (SELECT S# FROM S WHERE SEX = '女') AND SCORE < (SELECT AVG (SCORE) FROM SC)

6) UPDATE SC SET SCORE = SCORE + SCORE * 0.05 WHERE SCORE < (SELECT AVG (SCORE) FROM SC)

第六章 数据库管理

要求、目标:

了解数据库的恢复、数据库的安全性, 熟悉事务、数据库的并发控制、数据库的完整性。

一、事务

1. DBMS 对 DB 的监控, 称为数据库的管理, 也称为数据库的保护。它分为四个方面: 数据库的恢复、并发控制、完整性控制和安全性控制。

2. 事务: 是构成单一逻辑工作单元的操作集合, 要么完整地执行, 要么完全不执行。DBS 运行的最小逻辑工作单位是“事务”, 所有对数据库的操作, 都要以事务作为一个整体单位来执行或撤销。

3. 一个事务由应用程序中的一组操作序列组成，在程序中，事务以 BEGIN TRANSACTION 开始，以 COMMIT 语句或 ROLLBACK 语句结束。COMMIT 语句表示事务执行成功地结束（提交）；ROLLBACK 语句表示事务执行不成功地结束（应该“回退”）。

4. 事务的 ACID 性质：

1) 原子性 (Atomicity)：一个事务对数据库的所有操作，是一个不可分割的工作单元。这些操作要么全部执行，要么什么也不做。

2) 一致性 (Consistency)：一个事务独立执行的结果，应保持数据库的一致性，即数据不会因事务的执行而遭受破坏。

3) 隔离性 (Isolation)：在多个事务并发执行时，系统应保证与这些事务先后单独执行时的结果一样。

4) 持久性 (Durability)：一个事务一旦完成全部操作后，它对数据库的所有更新应永久地反映在数据库中，不会丢失。

二、数据库的恢复

1. 数据库的可恢复性：系统能把数据库从被破坏、不正确的状态，恢复到最近一个正确的状态，DBMS 的这种能力称为可恢复性。

2. 为保证可恢复性，平时应做好转储和建立日志。日志库记录事务的开始、结束标志，记录事务对数据库的每一次插入、删除和修改前后的值。

3. 常见故障分三类：事务故障、系统故障（硬件故障、软件（DBMS、OS 或应用程序）错误、掉电等，不破坏数据库）、介质故障。

事务故障和系统故障的恢复由系统自动进行，而介质故障的恢复需要 DBA 配合执行。系统故障通常称为软故障，介质故障通常称为硬故障。

三、数据库的并发控制

1. 并发操作带来的三个问题：丢失更新、读脏数据（把未提交的随后被撤销的数据称为“脏数据”）、不可重复读。采用“封锁”技术解决这些问题。

2. 锁：是一个与数据项相关的变量，对可能应用于该数据项上的操作而言，锁描述了该数据项的状态。通常在数据库中每个数据项都有一个锁。

3. 锁的主要类型：

1) 排他锁 (X 锁、写锁)：如果事务 T 对某个数据 R（可以是数据项、记录、数据集乃至整个数据库）实现了 X 锁，那么在 T 对数据 R 解除封锁之前，不允许其他事务再对该数据加任何类型的锁，这种锁称为“X 锁”。

2) 共享锁 (S 锁、读锁)：如果事务 T 对某数据加上 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。

4. 封锁的粒度：封锁对象的大小。封锁粒度与系统的并发度成反比。

5. 调度：事务的执行次序。如果多个事务依次执行，称为事务的串行调度；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。

6. 可串行化调度：每个事务中，语句的先后顺序在各种调度中始终保持一致。在这个前提下，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”，否则是“不可串行化的调度。”

四、数据库的完整性

1. 数据库的完整性：是指数据的正确性、有效性和相容性，防止错误的数据进入数据库。

1) 正确性：是指数据的合法性。

- 2) 有效性: 是指数据是否属于所定义的有效范围。
- 3) 相容性: 是指表示同一事实的两个数据应相同, 不一致就是不相容。

2、SQL 中的完整性约束

1) 域约束

例: `CREATE DOMAIN COLOR CHAR (6) DEFAULT '???'
CONSTRAINT VALID_COLORS
CHECK (VALUE IN ('Red', 'Yellow', 'Blue', 'Green', '???'))`

定义完后, 该域便可作为一种数据类型使用, 如用在 `CREATE TABLE` 中。

2) 基本表约束

主要有三种形式: 候选键定义、外键定义和检查约束定义。这些定义都可在前面加 `CONSTRAINT <约束名>`, 也可以省略。

候选键的定义: `UNIQUE (<列名序列>)` 或 `PRIMARY KEY (<列名序列>)`

`UNIQUE` 方式: 表示值惟一;

`PRIMARY KEY`: 值惟一, 并且自动设为非空。一个基本表只能指定一个主键。

检查约束的定义:

例: `CHECK (AGE>=18 AND AGE<=20)`

3) 断言: 如果完整性约束牵涉面较广, 与多个关系有关, 或者与聚合操作有关, 应使用断言。

3. 触发器: 是一个能由系统自动执行对数据库修改的语句。有时也称为主动规则 (相对于前面的各种被动约束机制)。

五、数据库的安全性

1. 数据库的安全性: 是指保护数据库, 防止不合法的使用, 以免数据的泄密、更改或破坏。

2. SQL 中的安全性机制:

提供四个: 视图、权限、角色和审计

1) 视图: 使系统具有三个优点: 数据安全性 (用来对无权用户屏蔽数据)、逻辑数据独立性和操作简便性。视图不占存储空间。

2) 权限: 授权语句 `GRANT`、回收语句 `REVOKE`

3) 角色:

在 `SQL` 中, 用户是实际的人或是访问数据库的应用程序。而角色是一组具有相同权限的用户, 实际上角色是属于目录一级的概念。

用户和角色之间存在多对多联系, 即一个用户可以参与多个角色, 一个角色也可授予多个用户。

4) 审计:

用于安全性目的的数据库日志, 称为审计追踪。

第六章 复习题

一、单项选择题

- 1. 在关系数据库系统中, 为了简化用户的查询操作, 而又不增加数据的存储空间, 常用的方法是创建 (B)
 - A) 另一个表
 - B) 视图
 - C) 索引
 - D) 模式
- 2. 为了考虑安全性, 每个教师只能存取自己讲授的课程的学生成绩, 为此 DBA 应创建 (A)
 - A) 视图
 - B) 索引
 - C) 模式
 - D) 表
- 3. “事务的所有操作在数据库中要么全部正确反映出来, 要么全部不反映”, 这

- 是事务的哪一个特性? (A)
- A) 原子性 B) 一致性 C) 隔离性 D) 持久性
4. 事务的 ACID 特性中, C 的含义是 (A)
- A) 一致性 (Consistency) B) 邻接性 (Contiguity)
- C) 连续性 (Continuity) D) 并发性 (Concurrency)
5. 如果有两个事务, 同时对数据库中同一数据进行操作, 不会引起冲突的操作是 (C)
- A) 其中有一个是 DELETE B) 一个是 SELECT, 另一个是 UPDATE
- C) 两个都是 SELECT D) 两个都是 UPDATE
6. 下述哪一条是并发控制的主要方法 (B)
- A) 授权 B) 封锁 C) 日志 D) 索引
7. 事务的一致性是指 (D)
- A) 事务中包括的所有操作要么都做, 要么都不做
- B) 事务一旦提交, 对数据库的改变是永久的
- C) 一个事务内部的操作及使用的数据对并发的其他事务是隔离的
- D) 事务必须是使数据库从一个一致性状态变到另一个一致性状态
8. 事务的隔离性是指 (C)
- A) 事务中包括的所有操作要么都做, 要么都不做
- B) 事务一旦提交, 对数据库的改变是永久的
- C) 一个事务内部的操作及使用的数据对并发的其他事务是隔离的
- D) 事务必须是使数据库从一个一致性状态变到另一个一致性状态
9. 事务的原子性是指 (A)
- A) 事务中包括的所有操作要么都做, 要么都不做
- B) 事务一旦提交, 对数据库的改变是永久的
- C) 一个事务内部的操作及使用的数据对并发的其他事务是隔离的
- D) 事务必须是使数据库从一个一致性状态变到另一个一致性状态
10. SQL 的主码子句和外码子句属于 DBS 的 (A)
- A) 完整性措施 B) 安全性措施
- C) 恢复措施 D) 并发控制措施
11. 系统能把数据库从被破坏、不正确的状态, 恢复到最近一个正确的状态, DBMS 的这种能力称为 (C)
- A) 完整性控制 B) 安全性控制
- C) 可恢复性 D) 并发控制
12. 保证数据的正确性、有效性和相容性, 防止错误的数据进入数据库。这属于数据库保护中的 (A) 措施。
- A) 完整性控制 B) 安全性控制
- C) 可恢复性 D) 并发控制
13. 下列何种方法属于 SQL 中的主动约束机制? (D)
- A) 域约束 B) 基本表约束
- C) 断言 D) 触发器
14. 保护数据库, 防止不合法的使用, 以免数据的泄密、更改或破坏。这属于数据库保护中的 (B) 措施。
- A) 完整性控制 B) 安全性控制
- C) 可恢复性 D) 并发控制

二、填空题

1. 数据库管理系统中，为了保证事务的正确执行，维护数据库的完整性，要求数据库系统维护以下事务特性：原子性、一致性、隔离性和持久性。
2. 有两种基本类型的锁，它们是排他锁和共享锁。
3. DBS 运行的最小逻辑工作单位是事务。
4. 常见故障分三类：事务故障、系统故障和介质故障。

第七章 SQL Server 2000 简介及应用

要求、目标：

了解 SQL Server 2000，了解企业管理器的功能与操作，了解 T-SQL 的使用。

一、概述

1. SQL Server 2000 是 Microsoft 公司推出的适用于大型网络环境的关系型数据库管理系统。它支持客户机/服务器体系结构。
2. 客户机/服务器体系结构：将数据库防在服务器上，该服务器数据库中的数据可被多个用户同时存取，而用户存取服务器中的数据是通过客户机或服务端的应用程序获得的。
3. SQL Server 2000 的系统数据库：
 - 1) Master 数据库：记录了所有系统信息、登录账号、系统配置设置、系统中所有数据库及其系统信息以及存储介质信息。
 - 2) Model 数据库：为用户创建数据库提供的模板数据库。
 - 3) Msdb 数据库：主要用于存储任务计划信息、事件处理信息、备份恢复信息以及异常 SQL Server 2000 的报告等。
 - 4) Tempdb 数据库：存放所有临时表和临时的存储程序，并且存放目前使用中的表，它是一个全局的资源，临时表和存储程序可供所有用户使用。
4. SQL Server 2000 的主要工具
 - 1) 查询分析器：使用这个工具，用户可以交互地设计和测试 T-SQL 语句、批处理和脚本。
 - 2) 服务管理器：可以用来启动、暂停和停止 SQL Server 的主要服务和指示其当前状态。
 - 3) 企业管理器：用户管理 SQL Server 2000 的主要管理工具和图形界面。在企业管理器中可以完成几乎所有的管理工作，如管理登录账号、数据库用户和权限、创建和管理数据库、创建和管理表、视图、存储过程以及用户自定义数据类型等。

二、企业管理器

1. 触发器与存储过程的联系和区别

联系：触发器是一种特殊的存储过程。

区别：1) 存储过程利用 EXEC 语句调用执行，触发器为表上的 INSERT、UPDATE、DELETE 事件所触发而自动执行。

2) 触发器不能传递或接受参数。

2. 两个临时表

触发器被执行时，系统会创建一个或两个临时表 INSERTED 和 DELETED，这些表在结构上类似于定义触发器的表，用于保存用户操作更改的行的新值和旧值。

3. 一个用户要访问 SQL Server 数据库中的数据，必须经过三个认证过程：

- 1) 身份验证: 使用登录账号来标识用户, 身份验证只验证用户连接到 SQL Server 数据库服务器的资格, 即验证该用户是否具有连接到数据库服务器的“连接权”。
- 2) 当用户访问数据库时, 他必须具有对具体数据库的访问权, 即验证用户是否是该数据库的合法用户。
- 3) 当用户操作数据库中的数据或对象时, 他必须具有所要进行的操作权。
4. SQL Server 的两种身份验证模式: Windows 身份验证模式和混合验证模式。

三、T-SQL 语言

1. T-SQL: Transact-SQL 的简称, 为扩展 SQL。
2. 批处理: 由一条或多条 T-SQL 语句构成, 前端应用程序会将所有这些语句作为单个代码单元发送给 SQL Server。SQL Server 将整个批处理作为一个整体加以解析。批处理分隔符为 go。
3. SQL 脚本文件可包含多个批处理。

第七章 复习题

一、单项选择题

1. SQL Server 2000 是 (B) 公司的软件产品。
A) Sybase B) Microsoft C) Oracle D) IBM
2. SQL Server 2000 中, 负责启动、暂停和停止 SQL Server 服务的管理工具为 (D)。
A) 企业管理器 B) 查询分析器 C) 事件探查器 D) 服务管理器
3. SQL Server 2000 中, 用于配置 SQL Server 系统环境, 创建和管理所有 SQL Server 对象的管理工具为 (A)。
A) 企业管理器 B) 查询分析器 C) 事件探查器 D) 服务管理器
4. SQL Server 2000 中, 允许输入和执行 Transact-SQL 语句并返回语句的执行结果的管理工具为 (B)。
A) 企业管理器 B) 查询分析器 C) 事件探查器 D) 服务管理器
5. SQL Server 2000 中, 包含用户登录标识、系统配置信息、初始化等系统级信息的系统数据库为 (C)。
A) model B) msdb C) master D) tempdb
6. SQL Server 2000 中, (A) 系统数据库为用户数据库的模板。
A) model B) msdb C) master D) tempdb
7. SQL Server 2000 中, 存储用户建立的临时表的系统数据库为 (D)。
A) model B) msdb C) master D) tempdb
8. SQL Server 2000 是 (C) 类型的数据库管理系统。
A) 层次型 B) 网状型 C) 关系型

二、填空题

1. SQL Server 2000 中, 可供选择的身份验证模式有两种, 分别是 Windows 验证模式 和 混合验证模式。
2. SQL Server 2000 安装完成后, 包括四个系统数据库, 分别为: Master、Model、Msdb 和 Tempdb。

第八章 PowerBuilder9.0 简介及应用

要求、目标：

了解 PB9.0 的集成开发环境，了解应用程序的开发步骤，了解 PB9.0 连接数据库的方法。

一、PowerBuilder9.0 集成开发环境

1. PowerBuilder9.0 是当前最为流行的数据库前端应用开发和多层应用开发工具之一。

2. PB 的主要特点

1) 采用面向对象的编程方法和事件驱动的工作原理。

2) 支持跨平台开发，具有开放的数据库连接系统。

3. PB9.0 的开发空间

有三个层次：

1) 工作空间

2) 目标

3) 库文件

二、PB9.0 与数据库的连接

1. 连接步骤：

1) 建立 ODBC 数据源；

2) 建立数据库描述文件；

3) 用 Connect 命令连接数据库。

第八章 复习题

填空题：

1. PB9.0 的开发空间有三个层次：工作空间、目标和库文件。

第九章 数据库技术的发展

要求、目标：

了解数据库技术发展的前景，了解面向对象的概念建模和开放数据库互连技术。

一、面向对象的概念建模

1. 概念建模经历了“ER 图—对象联系图—类图”的发展历程。

2. 面向对象的数据类型系统：由基本类型、复合类型和引用类型三部分组成。

3. 对象联系图：是面向对象数据模型中数据结构的一种重要图例表示方法，由于使用了对象标识的概念，使结构的嵌套和递归成为可能。是描述面向对象数据模型的基本工具。它不仅完整地揭示了数据之间的联系，也把查询的层次观点表现的一清二楚。

4. 统一建模语言（UML）：面向对象开发的行业标准语言。

二、开放数据库互连（ODBC）

1. API：即应用程序设计接口，是为开发者提供单一的编程接口，这样同一个应用程序就可以访问不同的数据库服务器。

2. ODBC：Microsoft 公司的产品，以 C/S 结构为设计基础。该技术使得应用程序与 DBMS 之间在逻辑上可以分离，使应用程序具有数据库无关性。ODBC 定义了一个 API，每个应用程序利用相同的源代码就可访问不同的数据库系统，存取多个数据库中的数据。

ODBC 一个最显著的优点是：用它生成的应用程序与数据库或数据库引擎无关。

第九章 复习题

填空题：

1. 面向对象的数据类型系统：由基本类型、复合类型和引用类型三部分组成。
2. 概念建模经历了从 ER 图到对象联系图，再到类图的发展历程。