

数据库系统概论习题集

第一章 绪论

一、选择题

1. DBS 是采用了数据库技术的计算机系统，它是一个集合体，包含数据库、计算机硬件、软件和（ ）。
A. 系统分析员 B. 程序员 C. 数据库管理员 D. 操作员
2. 数据库（DB），数据库系统（DBS）和数据库管理系统（DBMS）之间的关系是（ ）。
A. DBS 包括 DB 和 DBMS B. DBMS 包括 DB 和 DBS
C. DB 包括 DBS 和 DBMS D. DBS 就是 DB，也就是 DBMS
3. 下面列出的数据库管理技术发展的三个阶段中，没有专门的软件对数据进行管理的是（ ）。
I. 人工管理阶段 II. 文件系统阶段 III. 数据库阶段
A. I 和 II B. 只有 II C. II 和 III D. 只有 I
4. 下列四项中，不属于数据库系统特点的是（ ）。
A. 数据共享 B. 数据完整性 C. 数据冗余度高 D. 数据独立性高
5. 数据库系统的数据独立性体现在（ ）。
A. 不会因为数据的变化而影响到应用程序
B. 不会因为系统数据存储结构与数据逻辑结构的变化而影响应用程序
C. 不会因为存储策略的变化而影响存储结构
D. 不会因为某些存储结构的变化而影响其他的存储结构
6. 描述数据库全体数据的全局逻辑结构和特性的是（ ）。
A. 模式 B. 内模式 C. 外模式 D. 用户模式
7. 要保证数据库的数据独立性，需要修改的是（ ）。
A. 模式与外模式 B. 模式与内模式 C. 三层之间的两种映射 D. 三层模式
8. 要保证数据库的逻辑数据独立性，需要修改的是（ ）。
A. 模式与外模式的映射 B. 模式与内模式之间的映射 C. 模式 D. 三层模式
9. 用户或应用程序看到的那部分局部逻辑结构和特征的描述是（ ），它是模式的逻辑子集。
A. 模式 B. 物理模式 C. 子模式 D. 内模式
10. 下述（ ）不是 DBA 数据库管理员的职责。
A. 完整性约束说明 B. 定义数据库模式 C. 数据库安全 D. 数据库管理系统设计

选择题参考答案：(1) C (2) A (3) D (4) C (5) B (6) A (7) C (8) A (9) C (10) D

二、简答题

1. 试述数据、数据库、数据库系统、数据库管理系统的概念。

数据：描述事物的符号记录称为数据。数据的种类有文字、图形、图象、声音、正文等等。数据与其语义是不可分的。

*解析：在现代计算机系统中数据的概念是广义的。早期的计算机系统主要用于科学计算，处理的数据是整数、实数、浮点数等传统数学中的数据等。现在计算机能存储和处理的对象十分广泛，表示这些对象的数据也越来越复杂。

数据与其语义是不可分的。500 这个数字可以表示一件物品的价格是 500 元，也可以表示一个学术会议参加的人数有 500 人。还可以表示一袋奶粉重 500 克。

数据库：数据库是长期储存在计算机内、有组织的、可共享的数据集合。数据库中的数据按一定的数据模型组织、描述和储存，具有较小的冗余度、较高的数据独立性和易扩展性，并可为各种用户共享。

*解析：简单地讲，数据库数据具有永久储存、有组织和可共享三个特点。

数据模型是数据库的核心概念。每个数据库中数据的都是按照某一种数据模型来组织的。

数据库系统：数据库系统（DBS）是指在计算机系统中引入数据库后的系统构成。数据库系统由数据库、数据库管理系统（及其开发工具）、应用系统、数据库管理员构成。

*解析：数据库系统和数据库是两个概念。数据库系统是一个人-机系统，数据库是数据库系统的一个组成部分。但是在日常工作中人们常常把数据库系统简称为数据库。希望读者能够从人们讲话或文章的上下文中区分“数据

库系统”和“数据库”。不要引起混淆。

数据库管理系统：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。用于科学地组织和存储数据、高效地获取和维护数据。DBMS 主要功能包括数据定义功能、数据操纵功能、数据库的运行管理功能、数据库的建立和维护功能。

***解析：**DBMS 是一个大型复杂的软件系统。是计算机中的基础软件。目前，专门研制 DBMS 的厂商及其研制的 DBMS 产品很多。著名的有美国 IBM 公司的 DB2 关系数据库管理系统、IMS 层次数据库管理系统；美国 ORACLE 公司的 ORACLE 关系数据库管理系统；SYBASE 公司的 SYBASE 关系数据库管理系统；美国微软公司的 SQL SERVER 关系数据库管理系统等等。

2. 使用数据库系统有什么好处？

使用数据库系统的好处是由数据库管理系统的特点或优点决定的。

使用数据库系统的好处很多，例如可以大大提高应用开发的效率，方便用户的使用，减轻数据库系统管理人员维护的负担等。

使用数据库系统可以大大提高应用开发的效率。因为在数据库系统中应用程序不必考虑数据的定义、存储和数据存取的具体路径，这些工作都由 DBMS 来完成。用一个通俗的比喻，使用了 DBMS 就如有了一个好参谋好助手，许多具体的技术工作都由这个助手来完成。开发人员就可以专注于应用逻辑的设计而不必为管理数据的许许多多复杂的细节操心。

还有，当应用逻辑改变，数据的逻辑结构需要改变时，由于数据库系统提供了数据与程序之间的独立性。数据逻辑结构的改变是 DBA 的责任，开发人员不必修改应用程序，或者只需要修改很少的应用程序。从而既简化了应用程序的编制，又大大减少了应用程序的维护和修改。

使用数据库系统可以减轻数据库系统管理人员维护系统的负担。因为 DBMS 在数据库建立、运用和维护时对数据库进行统一的管理和控制，包括数据的完整性、安全性，多用户并发控制，故障恢复等等都由 DBMS 执行。

总之，使用数据库系统的优点是很多的，既便于数据的集中管理，控制数据冗余，可以提高数据的利用率和一致性，又有利于应用程序的开发和维护。读者可以在自己今后的工作中结合具体应用，认真加以体会和总结。

3. 试述文件系统与数据库系统的区别和联系。

文件系统与数据库系统的区别：文件系统面向某一应用程序，共享性差、冗余度大，独立性差，纪录内有结构、整体无结构，应用程序自己控制。

数据库系统面向现实世界，共享性高、冗余度小，具有高度的物理独立性和一定的逻辑独立性，整体结构化，用数据模型描述，由数据库管理系统提供数据安全性、完整性、并发控制和恢复能力。

读者可以参考《概论》书中表 1.1 中的有关内容。

文件系统与数据库系统的联系是：文件系统与数据库系统都是计算机系统中管理数据的软件。

***解析：**

文件系统是操作系统的重要组成部分，而 DBMS 是独立于操作系统的软件。但是 DBMS 是在操作系统的基础上实现的。数据库中数据的组织和存储是通过操作系统中文件系统来实现的。

读者进一步学习数据库管理系统实现的有关课程后可以对本题有深入的理解和全面的解答。因为 DBMS 的实现与操作系统中的文件系统是紧密相关的。例如，数据库实现的基础是文件，对数据库的任何操作最终要转化为对文件的操作。所以在 DBMS 实现中数据库物理组织的基本问题是如何利用或如何选择操作系统提供的基本的文件组织方法。这里我们就不具体展开了。

4. 举出适合用文件系统而不是数据库系统的例子；再举出适合用数据库系统的应用例子。

- 适用于文件系统而不是数据库系统的应用例子

数据的备份，软件或应用程序使用过程中的临时数据存储一般使用文件比较合适。早期功能比较简单、比较固定的应用系统也适合用文件系统。

- 适用于数据库系统而非文件系统的应用例子

目前，几乎所有企业或部门的信息系统都以数据库系统为基础，都使用数据库。如一个工厂的管理信息系统（其中会包括许多子系统，如库存管理系统、物资采购系统、作业调度系统、设备管理系统、人事管理系统等等），还比如学校的学生管理系统，人事管理系统，图书馆的图书管理系统等等都适合用数据库系统。希望同学们能举出自己了解的应用例子。

5. 试述数据库系统的特点。

数据库系统的主要特点有：

一、数据结构化：数据库系统实现整体数据的结构化，这是数据库的主要特征之一，也是数据库系统与文件系统的本质区别。

*解析：注意这里“整体”两个字。在数据库系统中，数据不再针对某一个应用，而是面向全组织，具有整体的结构化。不仅数据是结构化的，而且数据的存取单位即一次可以存取数据的大小也很灵活。可以小到某一个数据项（如一个学生的姓名），大到一组记录（成千上万个学生记录）。而在文件系统中，数据的存取单位只有一个：记录。如一个学生的完整记录。

二、数据的共享性高，冗余度低，易扩充：数据库的数据不再面向某个应用而是面向整个系统，因此可以被多个用户、多个应用、用多种不同的语言共享使用。由于数据面向整个系统，是有结构的数据，不仅可以被多个应用共享使用，而且容易增加新的应用，这就使得数据库系统弹性大，易于扩充。

*解析：

数据共享可以大大减少数据冗余，节约存储空间，同时还能够避免数据之间的不相容性与不一致性。

所谓“数据面向某个应用”是指数据结构是针对某个应用设计的，只被这个应用程序或应用系统使用。可以说数据是某个应用的“私有资源”。

所谓“弹性大”是指系统容易扩充也容易收缩，即应用增加或减少时不必修改整个数据库的结构，或者只要做很少的修改。

我们可以取整体数据的各种子集用于不同的应用系统，当应用需求改变或增加时，只要重新选取不同的子集或加上一部分数据便可以满足新的需求。

三、数据独立性高：数据独立性包括数据的物理独立性和数据的逻辑独立性。数据库管理系统的模式结构和二级映象功能保证了数据库中的数据具有很高的物理独立性和逻辑独立性。

*解析：

所谓“独立性”即相互不依赖。数据独立性是指数据和程序相互不依赖。即数据的逻辑结构或物理结构改变了，程序不会跟着改变。数据与程序的独立，把数据的定义从程序中分离出去，加上数据的存取又由 DBMS 负责，简化了应用程序的编制，大大减少了应用程序的维护和修改。

四、数据由 DBMS 统一管理和控制：数据库的共享是并发的共享，即多个用户可以同时存取数据库中的数据甚至可以同时存取数据库中同一个数据。为此，DBMS 必须提供统一的数据控制功能，包括数据的安全性保护，数据的完整性检查，并发控制和数据库恢复。

*解析：

DBMS 数据控制功能包括四个方面：

数据的安全性保护：保护数据以防止不合法的使用造成的数据的泄密和破坏；

数据的完整性检查：将数据控制在有效的范围内或保证数据之间满足一定的关系；

并发控制：对多用户的并发操作加以控制和协调，保证并发操作的正确性；

数据库恢复：当计算机系统发生硬件故障、软件故障，或者由于操作员的失误以及故意的破坏影响数据库中数据的正确性，甚至造成数据库部分或全部数据的丢失时，能将数据库从错误状态恢复到某一已知的正确状态（亦称为完整状态或一致状态）。

下面我们可以得到“什么是数据库”的一个定义：

数据库是长期存储在计算机内有组织的大量的共享的数据集合。它可以供各种用户共享，具有最小冗余度和较高的数据独立性。DBMS 在数据库建立、运用和维护时对数据库进行统一控制，以保证数据的完整性、安全性，并在多用户同时使用数据库时进行并发控制，在发生故障后对系统进行恢复。

数据库系统的出现使信息系统从以加工数据的程序为中心转向围绕共享的数据库为中心的新阶段。

6．数据库管理系统的主要功能有哪些？

①数据库定义功能；

②数据存取功能；

③数据库运行管理；

④数据库的建立和维护功能。

7．试述数据模型的概念、数据模型的作用和数据模型的三个要素。

数据模型是数据库中用来对现实世界进行抽象的工具，是数据库中用于提供信息表示和操作手段的形式构架。

一般地讲，数据模型是严格定义的概念的集合。这些概念精确地描述系统的静态特性、动态特性和完整性约束条件。因此数据模型通常由数据结构、数据操作和完整性约束三部分组成。

①数据结构：是所研究的对象类型的集合，是对系统的静态特性的描述。

②数据操作：是指对数据库中各种对象（型）的实例（值）允许进行的操作的集合，包括操作及有关的操作规则，是对系统动态特性的描述。

③数据的约束条件：是完整性规则的集合，完整性规则是给定的数据模型中数据及其联系所具有的制约和依存规则，用以限定符合数据模型的数据库状态以及状态的变化，以保证数据的正确、有效、相容。

*解析：

数据模型是数据库系统中最重要的概念之一。同学们必须通过《概论》的学习真正掌握数据模型的概念和作用。

数据模型是数据库系统的基础。任何一个 DBMS 都以某一个数据模型为基础，或者说支持某一个数据模型。

数据库系统中模型有不同的层次。根据模型应用的不同目的，可以将模型分成两类或说两个层次：一是概念模型，是按用户的观点来对数据和信息建模，用于信息世界的建模，强调语义表达能力，概念简单清晰；另一是数据模型，是按计算机系统的观点对数据建模，用于机器世界，人们可以用它定义、操纵数据库中的数据。一般需要有严格的形式化定义和一组严格定义了语法和语义的语言，并有一些规定和限制，便于在机器上实现。

8. 试述概念模型的作用。

概念模型实际上是现实世界到机器世界的一个中间层次。概念模型用于信息世界的建模，是现实世界到信息世界的第一层抽象，是数据库设计人员进行数据库设计的有力工具，也是数据库设计人员和用户之间进行交流的语言。

9. 定义并解释概念模型中以下术语：

实体，实体型，实体集，属性，码，实体联系图（E-R 图）

实体：客观存在并可以相互区分的事物叫实体。

实体型：具有相同属性的实体具有相同的特征和性质，用实体名及其属性名集合来抽象和刻画同类实体称为实体型。

实体集：同型实体的集合称为实体集。

属性：实体所具有的某一特性，一个实体可由若干个属性来刻画。

码：唯一标识实体的属性集称为码。

实体联系图：E-R 图提供了表示实体型、属性和联系的方法：

- 实体型：用矩形表示，矩形框内写明实体名。
- 属性：用椭圆形表示，并用无向边将其与相应的实体连接起来。
- 联系：用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时，在无向边旁标上联系类型（1:1, 1:n 或 m:n）。

10. 试给出三个实际部门的 E-R 图，要求实体型之间具有一对一，一对多，多对多各种不同的联系。

关系数据库

一、选择题

1. 下面的选项不是关系数据库基本特征的是（ ）。

- A.不同的列应有不同的数据类型
- B.不同的列应有不同的列名
- C.与行的次序无关
- D.与列的次序无关

2. 一个关系只有一个（ ）。

- A.候选码
- B. 外码
- C. 超码
- D. 主码

3. 关系模型中，一个码是（ ）。

- A.可以由多个任意属性组成
- B.至多由一个属性组成
- C.可有多个或者一个其值能够唯一表示该关系模式中任何元组的属性组成
- D.以上都不是

4. 现有如下关系：

患者（患者编号，患者姓名，性别，出生日起，所在单位）

医疗（患者编号，患者姓名，医生编号，医生姓名，诊断日期，诊断结果）

其中，医疗关系中的外码是（ ）。

A. 患者编号 B. 患者姓名 C. 患者编号和患者姓名 D. 医生编号和患者编号

5. 现有一个关系：借阅（书号，书名，库存数，读者号，借期，还期），假如同一本书允许一个读者多次借阅，但不能同时对一种书借多本，则该关系模式的外码是（ ）。

A. 书号 B. 读者号 C. 书号+读者号 D. 书号+读者号+借期

6. 关系模型中实现实体间 $N:M$ 联系是通过增加一个（ ）。

A. 关系实现 B. 属性实现 C. 关系或一个属性实现 D. 关系和一个属性实现

7. 关系代数运算是以（ ）为基础的运算。

A. 关系运算 B. 谓词演算 C. 集合运算 D. 代数运算

8. 关系数据库管理系统应能实现的专门关系运算包括（ ）。

A. 排序、索引、统计 B. 选择、投影、连接 C. 关联、更新、排序 D. 显示、打印、制表

9. 五种基本关系代数运算是（ ）。

A. $\cup, -, \times, \sigma, \pi$ B. $\cup, -, \sigma, \pi$ C. $\cup, \cap, \times, \sigma, \pi$ D. \cup, \cap, σ, π

10. 关系代数表达式的优化策略中，首先要做的是（ ）。

A. 对文件进行预处理 B. 尽早执行选择运算 C. 执行笛卡尔积运算 D. 投影运算

11. 关系数据库中的投影操作是指从关系中（ ）。

A. 抽出特定记录 B. 抽出特定字段 C. 建立相应的影像 D. 建立相应的图形

12. 从一个数据库文件中取出满足某个条件的所有记录形成一个新的数据库文件的操作是（ ）操作。

A. 投影 B. 联接 C. 选择 D. 复制

13. 关系代数中的联接操作是由（ ）操作组合而成。

A. 选择和投影 B. 选择和笛卡尔积 C. 投影、选择、笛卡尔积 D. 投影和笛卡尔积

14. 自然联接是构成新关系的有效方法。一般情况下，当对关系 R 和 S 是用自然联接时，要求 R 和 S 含有一个或者多个共有的（ ）。

A. 记录 B. 行 C. 属性 D. 元组

15. 假设有关系 R 和 S ，在下列的关系运算中，（ ）运算不要求：“ R 和 S 具有相同的元数，且它们的对应属性的数据类型也相同”。

A. $R \cap S$ B. $R \cup S$ C. $R - S$ D. $R \times S$

16. 假设有关系 R 和 S ，关系代数表达式 $R - (R - S)$ 表示的是（ ）。

A. $R \cap S$ B. $R \cup S$ C. $R - S$ D. $R \times S$

17. 下面列出的关系代数表达是中，那些式子能够成立（ ）。

i. $\sigma_{f1}(\sigma_{f2}(E)) = \sigma_{f1 \wedge f2}(E)$

ii. $E1 \bowtie E2 = E2 \bowtie E1$

iii. $(E1 \bowtie E2) \bowtie E3 = E1 \bowtie (E2 \bowtie E3)$

iv. $\sigma_{f1}(\sigma_{f2}(E)) = \sigma_{f2}(\sigma_{f1}(E))$

A. 全部 B. ii 和 iii C. 没有 D. i 和 iv

18. 下面四个关系表达式是等价的，是判别它们的执行效率（ ）。

$E1 = \pi_A(\sigma_{B=C \wedge D=E'}(R \times S))$

$E2 = \pi_A(\sigma_{B=C}(R \times \sigma_{D=E'}(S)))$

$E3 = \pi_A(R \bowtie_{B=C} \sigma_{D=E'}(S))$

$E4 = \pi_A(\sigma_{D=E'}(R \bowtie_{B=C} S))$

A. $E3$ 最快 B. $E2$ 最快 C. $E4$ 最快 D. $E1$ 最快

19. 有关系 $SC(S_ID, C_ID, AGE, SCORE)$ ，查找年龄大于 22 岁的学生的学号和分数，正确的关系代数表达式是（ ）。

i. $\pi_{S_ID, SCORE}(\sigma_{age>22}(SC))$

ii. $\sigma_{age>22}(\pi_{S_ID, SCORE}(SC))$

iii. $\pi_{S_ID, SCORE}(\sigma_{age>22}(\pi_{S_ID, SCORE, AGE}(SC)))$

A. i 和 ii B. 只有 ii 正确 C. 只有 i 正确 D. i 和 iii 正确

选择题参考答案：(1) A (2) D (3) C (4) A (5) D (6) A (7) C (8) B (9) A (10) B

(11) B (12) C (13) B (14) C (15) D (16) A (17) C (18) A (19) D

二、简答题

1. 试述关系模型的三个组成部分。
2. 试述关系数据语言的特点和分类。
3. 设有一个 SPJ 数据库，包括 S, P, J, SPJ 四个关系模式：

S(SNO, SNAME, STATUS, CITY);

P(PNO, PNAME, COLOR, WEIGHT);

J(JNO, JNAME, CITY);

SPJ(SNO, PNO, JNO, QTY);

供应商表 S 由供应商代码 (SNO)、供应商姓名 (SNAME)、供应商状态 (STATUS)、供应商所在城市 (CITY) 组成；零件表 P 由零件代码 (PNO)、零件名 (PNAME)、颜色 (COLOR)、重量 (WEIGHT) 组成；工程项目表 J 由工程项目代码 (JNO)、工程项目名 (JNAME)、工程项目所在城市 (CITY) 组成；供应情况表 SPJ 由供应商代码 (SNO)、零件代码 (PNO)、工程项目代码 (JNO)、供应数量 (QTY) 组成，表示某供应商供应某种零件给某工程项目的数量为 QTY。

试用关系代数完成如下查询：

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

4. 定义并理解下列术语，说明它们之间的联系与区别：

- (1) 域，笛卡尔积，关系，元组，属性
- (2) 主码，候选码，外码
- (3) 关系模式，关系，关系数据库

5. 试述关系模型的完整性规则。在参照完整性中，为什么外码属性的值有时也可以为空？什么情况下才可以为空？

6. 试述等值连接与自然连接的区别和联系。
7. 关系代数的基本运算有哪些？
8. 试用关系代数的基本运算来表示其他运算。

SQL 语言

一、选择题

1. SQL 语言是 () 的语言，容易学习。
A. 过程化 B. 非过程化 C. 格式化 D. 导航式
2. SQL 语言的数据操纵语句包括 SELECT、INSERT、UPDATE、DELETE 等。其中最重要的，也是使用最频繁语句是 ()。
A. SELECT B. INSERT C. UPDATE D. DELETE
3. 在视图上不能完成的操作是 ()。
A. 更新视图 B. 查询 C. 在视图上定义新的表 D. 在视图上定义新的视图
4. SQL 语言集数据查询、数据操纵、数据定义和数据控制功能于一体，其中，CREATE、DROP、ALTER 语句是实现哪种功能 ()。
A. 数据查询 B. 数据操纵 C. 数据定义 D. 数据控制
5. SQL 语言中，删除一个视图的命令是 ()。
A. DELETE B. DROP C. CLEAR D. REMOVE
6. 在 SQL 语言中的视图 VIEW 是数据库的 ()。
A. 外模式 B. 模式 C. 内模式 D. 存储模式
7. 下列的 SQL 语句中，() 不是数据定义语句。

A. CREATE TABLE B. DROP VIEW C. CREATE VIEW D. GRANT

8. 若要撤销数据库中已经存在的表 S，可用（ ）。

A. DELETE TABLE S B. DELETE S C. DROP TABLE S D. DROP S

9. 若要在基本表 S 中增加一列 CN（课程名），可用（ ）。

A. ADD TABLE S (CN CHAR (8)) B. ADD TABLE S ALTER (CN CHAR (8))
C. ALTER TABLE S ADD (CN CHAR (8)) D. ALTER TABLE S (ADD CN CHAR (8))

10. 学生关系模式 S (S#, Sname, Sex, Age)，S 的属性分别表示学生的学号、姓名、性别、年龄。要在表 S 中删除一个属性“年龄”，可选用的 SQL 语句是（ ）。

A. DELETE Age from S B. ALTER TABLE S DROP Age
C. UPDATE S Age D. ALTER TABLE S 'Age'

11. 有关系 S (S#, SNAME, SAGE)，C (C#, CNAME)，SC (S#, C#, GRADE)。其中 S# 是学生号，SNAME 是学生姓名，SAGE 是学生年龄，C# 是课程号，CNAME 是课程名称。要查询选修“ACCESS”课的年龄不小于 20 的全体学生姓名的 SQL 语句是 SELECT SNAME FROM S, C, SC WHERE 子句。这里的 WHERE 子句的内容是（ ）。

A. S.S# = SC.S# and C.C# = SC.C# and SAGE >= 20 and CNAME = 'ACCESS'
B. S.S# = SC.S# and C.C# = SC.C# and SAGE in >= 20 and CNAME in 'ACCESS'
C. SAGE in >= 20 and CNAME in 'ACCESS'
D. SAGE >= 20 and CNAME = 'ACCESS'

12. 设关系数据库中一个表 S 的结构为 S (SN, CN, grade)，其中 SN 为学生名，CN 为课程名，二者均为字符型；grade 为成绩，数值型，取值范围 0—100。若要把“张二的化学成绩 80 分”插入 S 中，则可用（ ）。

A. ADD INTO S VALUES ('张二', '化学', '80')
B. INSERT INTO S VALUES ('张二', '化学', '80')
C. ADD INTO S VALUES ('张二', '化学', 80)
D. INSERT INTO S VALUES ('张二', '化学', 80)

13. 设关系数据库中一个表 S 的结构为：S (SN, CN, grade)，其中 SN 为学生名，CN 为课程名，二者均为字符型；grade 为成绩，数值型，取值范围 0—100。若要更正王二的化学成绩为 85 分，则可用（ ）。

A. UPDATE S SET grade = 85 WHERE SN = '王二' AND CN = '化学'
B. UPDATE S SET grade = '85' WHERE SN = '王二' AND CN = '化学'
C. UPDATE grade = 85 WHERE SN = '王二' AND CN = '化学'
D. UPDATE grade = '85' WHERE SN = '王二' AND CN = '化学'

14. 在 SQL 语言中，子查询是（ ）。

A. 返回单表中数据子集的查询语言
B. 选取多表中字段子集的查询语句
C. 选取单表中字段子集的查询语句
D. 嵌入到另一个查询语句之中的查询语句

15. SQL 是一种（ ）语言。

A. 高级算法 B. 人工智能 C. 关系数据库 D. 函数型

16. 有关系 S (S#, SNAME, SEX)，C (C#, CNAME)，SC (S#, C#, GRADE)。其中 S# 是学生号，SNAME 是学生姓名，SEX 是性别，C# 是课程号，CNAME 是课程名称。要查询选修“数据库”课的全体男生姓名的 SQL 语句是 SELECT SNAME FROM S, C, SC WHERE 子句。这里的 WHERE 子句的内容是（ ）。

A. S.S# = SC.S# and C.C# = SC.C# and SEX = '男' and CNAME = '数据库'
B. S.S# = SC.S# and C.C# = SC.C# and SEX in '男' and CNAME in '数据库'
C. SEX '男' and CNAME '数据库'
D. S.SEX = '男' and CNAME = '数据库'

17. 若用如下的 SQL 语句创建了一个表 SC：

CREATE TABLE SC (S# CHAR (6) NOT NULL, C# CHAR (3) NOT NULL, SCORE INTEGER, NOTE CHAR (20)); 向 SC 表插入如下行时，（ ）行可以被插入。

- A. (' 201009', ' 111', 60, 必修) B. (' 200823', ' 101', NULL, NULL)
C. (NULL, ' 103', 80, ' 选修') D. (' 201132', NULL, 86, ' ')

18. 假设学生关系 S (S#, SNAME, SEX), 课程关系 C (C#, CNAME), 学生选课关系 SC (S#, C#, GRADE)。要查询选修 “Computer” 课的男生姓名, 将涉及到关系 ()。

- A. S B. S, SC C. C, SC D. S, C, SC

选择题参考答案: (1) B (2) A (3) C (4) C (5) B (6) A (7) D (8) C (9) C (10) B
(11) A (12) D (13) A (14) D (15) C (16) A (17) B (18) D

二、简答题

1. 试述 SQL 语言的特点。

答: (1) 综合统一。SQL 语言集数据定义语言 DDL、数据操纵语言 DML、数据控制语言 DCL 的功能于一体。

(2) 高度非过程化。用 SQL 语言进行数据操作, 只要提出“做什么”, 而无须指明“怎么做”, 因此无需了解存取路径, 存取路径的选择以及 SQL 语句的操作过程由系统自动完成。

(3) 面向集合的操作方式。SQL 语言采用集合操作方式, 不仅操作对象、查找结果可以是元组的集合, 而且一次插入、删除、更新操作的对象也可以是元组的集合。

(4) 以同一种语法结构提供两种使用方式。SQL 语言既是自含式语言, 又是嵌入式语言。作为自含式语言, 它能够独立地用于联机交互的使用方式, 也能够嵌入到高级语言程序中, 供程序员设计程序时使用。

(5) 语言简捷, 易学易用。

2. 试述 SQL 的定义功能。

答: SQL 的数据定义功能包括定义表、定义视图和定义索引。

SQL 语言使用 CREATE TABLE 语句定义建立基本表,; ALTER TABLE 语句修改基本表定义, DROP TABLE 语句删除基本表; 建立索引使用 CREATE INDEX 语句建立索引, DROP INDEX 语句删除索引表; SQL 语言使用 CREATE VIEW 命令建立视图, DROP VIEW 语句删除视图。

3. 用 SQL 语句建立第 3 章习题 3 中的四个表。

答: 对于 S 表: S(SNO, SNAME, STATUS, CITY);

建 S 表: CREATE TABLE S(SNO CHAR(3), SNAME CHAR(10), STATUS CHAR(2), CITY CHAR(10));
P(PNO, PNAME, COLOR, WEIGHT);

建 P 表: CREATE TABLE P(PNO CHAR(3), PNAME CHAR(10), COLOR CHAR(4), WEIGHT INT);
J(JNO, JNAME, CITY);

建 J 表: CREATE TABLE J(JNO CHAR(3), JNAME CHAR(10), CITY CHAR(10));

SPJ(SNO, PNO, JNO, QTY);

建 SPJ 表: CREATE TABLE SPJ(SNO CHAR(3), PNO CHAR(3), JNO CHAR(3), QTY INT);

4. 针对上题中建立的四个表试用 SQL 语言完成第 3 章习题 3 中的查询。

答: (1) 求供应工程 J1 零件的供应商号码 SNO;

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

(2) 求供应工程 J1 零件 P1 的供应商号码 SNO;

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

(3) 求供应工程 J1 零件为红色的供应商号码 SNO;

```
SELECT SNO
FROM SPJ
WHERE JNO= 'J1'
AND PNO IN
(SELECT PNO
```



```
FROM P
WHERE COLOR= '红' );
```

或

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

(4) 求没有使用天津供应商生产的红色零件的工程师 JNO;

解析:

用 SQL 语言表示如下:

```
SELECT JNO
FROM J
WHERE NOT EXISTS
(SELECT *
FROM SPJ
WHERE SPJ.JNO=J.JNO
AND SNO IN
(SELECT SNO
FROM S
WHERE CITY= '天津' )
AND PNO IN
(SELECT PNO
FROM P
WHERE COLOR= '红' ));
```

或

```
SELECT JNO
FROM J
WHERE NOT EXISTS
(SELECT *1
FROM SPJ, S, P
WHERE SPJ.JNO=J.JNO
AND SPJ.SNO=S.SNO
AND SPJ.PNO=P.PNO
AND S.CITY= '天津'
AND P. COLOR= '红' );
```

注意: 从 J 表入手, 以包含那些尚未使用任何零件的工程师。

(5) 求至少用了供应商 S1 所供应的全部零件的工程师 JNO 。

解析:

用 SQL 语言表示如下:

```
SELECT DISTINCT JNO
FROM SPJ SPJZ
WHERE NOT EXISTS
(SELECT *
FROM SPJ SPJX
WHERE SNO='S1'
AND NOT EXISTS
```

```

(SELECT *
FROM SPJ SPJY
WHERE SPJY.PNO=SPJX.PNO
AND SPJY.JNO=SPJZ.JNO
AND SPJY.SNO=' S1' ));
AND SPJY.SNO='S1' ));

```

5. 针对习题 3 中的四个表试用 SQL 语言完成以下各项操作：

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

答： (1) 找出所有供应商的姓名和所在城市。

```

SELECT SNAME, CITY
FROM S;

```

- (2) 找出所有零件的名称、颜色、重量。

```

SELECT PNAME, COLOR, WEIGHT
FROM P;

```

- (3) 找出使用供应商 S1 所供应零件的工程号码。

```

SELECT JNO
FROM SPJ
WHERE SNO= 'S1' ;

```

- (4) 找出工程项目 J2 使用的各种零件的名称及其数量。

```

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

```

- (5) 找出上海厂商供应的所有零件号码。

```

SELECT DISTINCT PNO
FROM SPJ
WHERE SNO IN
( SELECT SNO
FROM S
WHERE CITY='上海' );

```

- (6) 找出使用上海产的零件的工程名称。

```

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

```

或

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

- (7) 找出没有使用天津产的零件的工程号码。

```
SELECT JNO
FROM J
WHERE NOT EXISTS
(SELECT *
FROM SPJ
WHERE SPJ.JNO=J.JNO AND SNO IN
(SELECT SNO
FROM S
WHERE CITY= '天津' ));
```

或

```
SELECT JNO
FROM J
WHERE NOT EXISTS
(SELECT *1
FROM SPJ, S
WHERE SPJ.JNO=J.JNO AND SPJ.SNO=S.SNO AND S.CITY= '天津' );
```

- (8) 把全部红色零件的颜色改成蓝色。

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

- (9) 由 S5 供给 J4 的零件 P6 改为由 S3 供应，请作必要的修改。

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

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

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

或

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

解析：注意删除顺序，应该先从 SPJ 表中删除供应商 S2 所供应零件的记录，然后从 S 表中删除 S2。

- (11) 请将 (S2, J6, P4, 200) 插入供应情况关系。

```
INSERT INTO SPJ(SNO, JNO, PNO, QTY)
VALUES (S2, J6, P4, 200);
```

或

```
INSERT INTO SPJ
```

VALUES (S2, P4, J6, 200);

6. 什么是基本表？什么是视图？两者的区别和联系是什么？

答：基本表是本身独立存在的表，在 SQL 中一个关系就对应一个表。

视图是从一个或几个基本表导出的表。视图本身不独立存储在数据库中，是一个虚表。即数据库中只存放视图的定义而不存放视图对应的数据，这些数据仍存放在导出视图的基本表中。视图在概念上与基本表等同，用户可以如同基本表那样使用视图，可以在视图上再定义视图。

7. 试述视图的优点。

答：(1)视图能够简化用户的操作。

(2)视图使用户能以多种角度看待同一数据。

(3)视图对重构数据库提供了一定程度的逻辑独立性。

(4)视图能够对机密数据提供安全保护。

8. 所有的视图是否都可以更新？为什么？

答：不是。视图是不实际存储数据的虚表，因此对视图的更新，最终要转换为对基本表的更新。因为有些视图的更新不能唯一地有意义地转换成对相应基本表的更新，所以，并不是所有的视图都是可更新的。如《概论》3.5.1 中的视图 S_G（学生的学号及他的平均成绩）

```
CREATE VIEW S_G(Sno, Gavg)
AS SELECT Sno, AVG(Grade)
FROM SC
GROUP BY Sno;
```

要修改平均成绩，必须修改各科成绩，而我们无法知道哪些课程成绩的变化导致了平均成绩的变化。

9. 哪类视图是可以更新的，哪类视图是不可更新的？各举一例说明。

答：基本表的行列子集视图一般是可更新的。如《概论》3.5.3 中的例 1。

若视图的属性来自集函数、表达式，则该视图肯定是不可以更新的。

如《概论》3.5.3 中的 S_G 视图。

10. 试述某个你熟悉的实际系统中对视图更新的规定。

答：（略）

解析：不同的系统对视图更新的规定是不同的，读者必须了解你所用系统对视图更新的规定。

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

（1）找出三建工程项目使用的各种零件代码及其数量。

（2）找出供应商 S1 的供应情况。

答：建视图：

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

对该视图查询：

(1) 找出三建工程项目使用的各种零件代码及其数量。

```
SELECT PNO, QTY
FROM V_SPJ;
```

(2) 找出供应商 S1 的供应情况。

```
SELECT PNO, QTY /* S1 供应三建工程的零件号和对应的数量 */
FROM V_SPJ
WHERE SNO='S1';
```

关系系统及其优化

一、选择题

1.概念模型是现实世界的第一层抽象，这一类最著名的模型是（ ）。

A.层次模型 B. 关系模型 C. 网状模型 D. 实体-关系模型

2.区分不同实体的依据是（ ）。

A. 名称 B. 属性 C. 对象 D. 概念

3.关系数据模型是目前最重要的一种数据模型，它的三个要素分别为（ ）。

A.实体完整、参照完整、用户自定义完整 B.数据结构、关系操作、完整性约束

C.数据增加、数据修改、数据查询 D.外模式、模式、内模式

4.在（ ）中一个结点可以有多个双亲，节点之间可以有多种联系。

A.网状模型 B. 关系模型 C.层次模型 D. 以上都有

5.（ ）的存取路径对用户透明，从而具有更高的数据独立性、更好的安全保密性，也简化了程序员的工作和数据库开发建立的工作。

A.网状模型 B. 关系模型 C.层次模型 D. 以上都有

6.在关系数据库中，要求基本关系中所有的主属性上不能有空值，其遵守的约束规则是（ ）。

A.数据依赖完整性规则 B. 用户定义完整性规则 C.实体完整性规则 D. 域完整性规则

选择题参考答案：(1) D (2) B (3) B (4) A (5) B (6) C

二、简答题

1.试述关系模型的三个组成部分。

答：关系模型由关系数据结构、关系操作集合和关系完整性约束三部分组成。

2.试述关系数据语言的特点和分类。

答：关系数据语言可以分为三类：

关系代数语言 例如 ISBL

关系演算语言 （元组关系演算语言 例如 APLHA, QUEL 和 域关系演算语言 例如 QBE）

具有关系代数和关系演算双重特点的语言 例如 SQL

这些关系数据语言的共同特点是，具有完备的表达能力，是非过程化的集合操作语言，功能强，能够嵌入高级语言中使用。

3. 定义并理解下列术语，说明它们之间的联系与区别：

（1）域，关系，元组，属性

答：域：域是一组具有相同数据类型的值的集合。

关系：在域 D_1, D_2, \dots, D_n 上笛卡尔积 $D_1 \times D_2 \times \dots \times D_n$ 的子集称为关系，表示为 $R(D_1, D_2, \dots, D_n)$

元组：关系中的每个元素是关系中的元组。

属性：关系也是一个二维表，表的每行对应一个元组，表的每列对应一个域。由于域可以相同，为了加以区分，必须对每列起一个名字，称为属性（Attribute）。

（2）主码，候选码，外部码

答：候选码：若关系中的某一属性组的值能唯一地标识一个元组，则称该属性组为候选码（Candidate key）。

主码：若一个关系有多个候选码，则选定其中一个为主码（Primary key）。

外部码：设 F 是基本关系 R 的一个或一组属性，但不是关系 R 的码，如果 F 与基本关系 S 的主码 K_s 相对应，则称 F 是基本关系 R 的外部码（Foreign key），简称外码。

基本关系 R 称为参照关系（Referencing relation），基本关系 S 称为被参照关系（Referenced relation）或目标关系（Target relation）。关系 R 和 S 可以是相同的关系。

(3)关系模式，关系，关系数据库

关系模式：关系的描述称为关系模式（Relation Schema）。它可以形式化地表示为：

$R(U, D, \text{dom}, F)$

其中 R 为关系名， U 为组成该关系的属性名集合， D 为属性组 U 中属性所来自的域， dom 为属性向域的映象

集合，F 为属性间数据的依赖关系集合。

关系：在域 D_1, D_2, \dots, D_n 上笛卡尔积 $D_1 \times D_2 \times \dots \times D_n$ 的子集称为关系，表示为

$R(D_1, D_2, \dots, D_n)$

关系是关系模式在某一时刻的状态或内容。关系模式是静态的、稳定的，而关系是动态的、随时间不断变化的，因为关系操作在不断地更新着数据库中的数据。

关系数据库：关系数据库也有型和值之分。关系数据库的型也称为关系数据库模式，是对关系数据库的描述，它包括若干域的定义以及在这些域上定义的若干关系模式。关系数据库的值是这些关系模式在某一时刻对应的关系的集合，通常就称为关系数据库。

4. 试述关系模型的完整性规则。在参照完整性中，为什么外部码属性的值也可以为空？什么情况下才可以为空？

答：关系模型的完整性规则是对关系的某种约束条件。关系模型中可以有三类完整性约束：实体完整性、参照完整性和用户定义的完整性。

其中实体完整性和参照完整性是关系模型必须满足的完整性约束条件，被称作是关系的两个不变性，应该由关系系统自动支持。

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

2) 参照完整性规则：若属性（或属性组）F 是基本关系 R 的外码，它与基本关系 S 的主码 K_s 相对应（基本关系 R 和 S 不一定是不同的关系），则对于 R 中每个元组在 F 上的值必

须为：

- 或者取空值（F 的每个属性值均为空值）；

- 或者等于 S 中某个元组的主码值。

3) 用户定义的完整性是针对某一具体关系数据库的约束条件。它反映某一具体应用所涉及的数据必须满足的语义要求。

在参照完整性中，外部码属性的值可以为空，它表示该属性的值尚未确定。但前提条件是该外部码属性不是其所在关系的主属性。

例如，在下面的“学生”表中，“专业号”是一个外部码，不是学生表的主属性，可以为空。其语义是，该学生的专业尚未确定。

学生（学号，姓名，性别，专业号，年龄）

专业（专业号，专业名）

而在下面的“选修”表中的“课程号”虽然也是一个外部码属性，但它又是“选修”表的主属性，所以不能为空。因为关系模型必须满足实体完整性。

课程（课程号，课程名，学分）

选修（学号，课程号，成绩）

5. 等值连接与自然连接的区别是什么？

答：连接运算中有两种最为重要也最为常用的连接，一种是等值连接（equi-join），另一种是自然连接（Natural join）。

θ 为“=”的连接运算称为等值连接。

它是从关系 R 与 S 的笛卡尔积中选取 A、B 属性值相等的那些元组。即等值连接为：

$RA=BS = \{tr \ ts | tr \in R \wedge ts \in S \wedge tr[A]=ts[B]\}$

自然连接（Natural join）是一种特殊的等值连接，它要求两个关系中进行比较的分量必须是相同的属性组，并且要在结果中把重复的属性去掉。即若 R 和 S 具有相同的属性组 B，则自然连接可记作：

$RS = \{tr \ ts | tr \in R \wedge ts \in S \wedge tr[B]=ts[B]\}$

关系数据理论

一、选择题

1. 为了设计出性能较优的关系模式，必须进行规范化，规范化主要的理论依据是（ ）。

A. 关系规范化理论 B. 关系代数理论 C. 数理逻辑 D. 关系运算理论

2. 规范化理论是关系数据库进行逻辑设计的理论依据，根据这个理论，关系数据库中的关系必须满足：每一个属性都是（ ）。

A. 长度不变的 B. 不可分解的 C. 互相关联的 D. 互不相关的

3. 已知关系模式 $R(A, B, C, D, E)$ 及其上的函数相关性集合 $F=\{A \rightarrow D, B \rightarrow C, E \rightarrow A\}$, 该关系模式的候选关键字是 ()。

A. AB B. BE C. CD D. DE

4. 设学生关系 $S(SNO, SNAME, SSEX, SAGE, SDPART)$ 的主键为 SNO , 学生选课关系 $SC(SNO, CNO, SCORE)$ 的主键为 SNO 和 CNO , 则关系 $R(SNO, CNO, SSEX, SAGE, SDPART, SCORE)$ 的主键为 SNO 和 CNO , 其满足 ()。

A. 1NF B. 2NF C. 3NF D. BCNF

5. 设有关系模式 $W(C, P, S, G, T, R)$, 其中各属性的含义是: C 表示课程, P 表示教师, S 表示学生, G 表示成绩, T 表示时间, R 表示教室, 根据语义有如下数据依赖集: $D=\{C \rightarrow P, (S, C) \rightarrow G, (T, R) \rightarrow C, (T, P) \rightarrow R, (T, S) \rightarrow R\}$, 关系模式 W 的一个关键字是 ()。

A. (S, C) B. (T, R) C. (T, P) D. (T, S)

6. 关系模式中, 满足 2NF 的模式 ()。

A. 可能是 1NF B. 必定是 1NF C. 必定是 3NF D. 必定是 BCNF

7. 关系模式 R 中的属性全是主属性, 则 R 的最高范式必定是 ()。

A. 1NF B. 2NF C. 3NF D. BCNF

8. 消除了部分函数依赖的 1NF 的关系模式, 必定是 ()。

A. 1NF B. 2NF C. 3NF D. BCNF

9. 如果 $A \rightarrow B$, 那么属性 A 和属性 B 的联系是 ()。

A. 一对多 B. 多对一 C. 多对多 D. 以上都不是

10. 关系模式的候选关键字可以有 1 个或多个, 而主关键字有 ()。

A. 多个 B. 0 个 C. 1 个 D. 1 个或多个

11. 候选关键字的属性可以有 ()。

A. 多个 B. 0 个 C. 1 个 D. 1 个或多个

12. 关系模式的任何属性 ()。

A. 不可再分 B. 可以再分 C. 命名在关系模式上可以不唯一 D. 以上都不是

13. 设有关系模式 $W(C, P, S, G, T, R)$, 其中各属性的含义是: C 表示课程, P 表示教师, S 表示学生, G 表示成绩, T 表示时间, R 表示教室, 根据语义有如下数据依赖集: $D=\{C \rightarrow P, (S, C) \rightarrow G, (T, R) \rightarrow C, (T, P) \rightarrow R, (T, S) \rightarrow R\}$, 若将关系模式 W 分解为三个关系模式 $W_1(C, P)$, $W_2(S, C, G)$, $W_3(S, T, R, C)$, 则 W_1 的规范化程序最高达到 ()。

A. 1NF B. 2NF C. 3NF D. BCNF

14. 在关系数据库中, 任何二元关系模式的最高范式必定是 ()。

A. 1NF B. 2NF C. 3NF D. BCNF

15. 在关系规范式中, 分解关系的基本原则是 ()。

- I. 实现无损连接
- II. 分解后的关系相互独立
- III. 保持原有的依赖关系

A. I 和 II B. I 和 III C. I D. II

16. 不能使一个关系从第一范式转化为第二范式的条件是 ()。

- A. 每一个非属性都完全函数依赖主属性
- B. 每一个非属性都部分函数依赖主属性
- C. 在一个关系中没有非属性存在
- D. 主键由一个属性构成

17. 任何一个满足 2NF 但不满足 3NF 的关系模式都不存在 ()。

- A. 主属性对键的部分依赖
- B. 非主属性对键的部分依赖
- C. 主属性对键的传递依赖
- D. 非主属性对键的传递依赖

18. 设数据库关系模式 $R=(A, B, C, D, E)$, 有下列函数依赖: $A \rightarrow BC, D \rightarrow E, C \rightarrow D$; 下述对 R 的分解中, 哪些分解是 R 的无损连接分解 ()。

I. $(A, B, C)(C, D, E)$

II. (A, B) (A, C, D, E)

III. (A, C) (B, C, D, E)

IV. (A, B) (C, D, E)

A. 只有IV B. I 和 II C. I、II 和 III D. 都不是

19. 设 U 是所有属性的集合, X 、 Y 、 Z 都是 U 的子集, 且 $Z=U-X-Y$ 。下面关于多值依赖的叙述中, 不正确的是 ()。

A. 若 $X \twoheadrightarrow Y$, 则 $X \twoheadrightarrow Z$

B. 若 $X \rightarrow Y$, 则 $X \twoheadrightarrow Y$

C. 若 $X \twoheadrightarrow Y$, 且 $Y' \in Y$, 则 $X \twoheadrightarrow Y'$

D. 若 $Z = \phi$, 则 $X \twoheadrightarrow Y$

20. 若关系模式 $R(U, F)$ 属于 3NF, 则 ()。

A. 一定属于 BCNF

B. 消除了插入的删除异常

C. 仍存在一定的插入和删除异常

D. 属于 BCNF 且消除了插入和删除异常

21. 下列说法不正确的是 ()。

A. 任何一个包含两个属性的关系模式一定满足 3NF

B. 任何一个包含两个属性的关系模式一定满足 BCNF

C. 任何一个包含三个属性的关系模式一定满足 3NF

D. 任何一个关系模式都一定有码

22. 设关系模式 $R(A, B, C)$, F 是 R 上成立的 FD 集, $F=\{B \rightarrow C\}$, 则分解 $P=\{AB, BC\}$ 相对于 F ()。

A. 是无损联接, 也是保持 FD 的分解

B. 是无损联接, 也不保持 FD 的分解

C. 不是无损联接, 但保持 FD 的分解

D. 既不是无损联接, 也不保持 FD 的分解

23. 关系数据库规范化是为了解决关系数据库中 () 的问题而引入的。

A. 插入、删除和数据冗余

B. 提高查询速度

C. 减少数据操作的复杂性

D. 保证数据的安全性和完整性

24. 关系的规范化中, 各个范式之间的关系是 ()。

A. $1NF \in 2NF \in 3NF$

B. $3NF \in 2NF \in 1NF$

C. $1NF=2NF=3NF$

D. $1NF \in 2NF \in BCNF \in 3NF$

25. 数据库中的冗余数据是指可 () 的数据。

A. 容易产生错误

B. 容易产生冲突

C. 无关紧要

D. 由基本数据导出

26. 学生表 ($id, name, sex, age, depart_id, depart_name$), 存在函数依赖是 $id \rightarrow name, sex, age, depart_id$; $dept_id \rightarrow dept_name$, 其满足 ()。

A. 1NF

B. 2NF

C. 3NF

D. BCNF

27. 设有关系模式 $R(S, D, M)$, 函数依赖集: $F=\{S \rightarrow D, D \rightarrow M\}$, 则关系模式 R 的规范化程度最高达到 ()。

A. 1NF

B. 2NF

C. 3NF

D. BCNF

28. 设有关系模式 $R(A, B, C, D)$, 其数据依赖集: $F=\{(A, B) \rightarrow C, C \rightarrow D\}$, 则关系模式 R 的规范化程度最高达到 ()。

A. 1NF

B. 2NF

C. 3NF

D. BCNF

29. 下列关于函数依赖的叙述中, 哪一条是不正确的 ()。

A. 由 $X \rightarrow Y, Y \rightarrow Z$, 则 $X \rightarrow YZ$

B. 由 $X \rightarrow YZ$, 则 $X \rightarrow Y, Y \rightarrow Z$

C. 由 $X \rightarrow Y, WY \rightarrow Z$, 则 $XW \rightarrow Z$

D. 由 $X \rightarrow Y, Z \in Y$, 则 $X \rightarrow Z$

30. $X \rightarrow Y$, 当下列哪一条成立时, 称为平凡的函数依赖 ()。

A. $X \in Y$

B. $Y \in X$

C. $X \cap Y = \phi$

D. $X \cap Y \neq \phi$

31. 关系数据库的规范化理论指出: 关系数据库中的关系应该满足一定的要求, 最起码的要求是达到 1NF, 即满足 ()。

A. 每个非主键属性都完全依赖于主键属性

B. 主键属性唯一标识关系中的元组

C. 关系中的元组不可重复

D. 每个属性都是不可分解的

32. 根据关系数据库规范化理论, 关系数据库中的关系要满足第一范式, 部门 (部门号, 部门名, 部门成员, 部门总经理) 关系中, 因哪个属性而使它不满足第一范式 ()。

A. 部门总经理

B. 部门成员

C. 部门名

D. 部门号

33. 有关系模式 $A(C, T, H, R, S)$, 其中各属性的含义是: C : 课程 T : 教员 H : 上课时间 R : 教室 S : 学生。根据语义有如下函数依赖集: $F=\{C \rightarrow T, (H, R) \rightarrow C, (H, T) \rightarrow R, (H, S) \rightarrow R\}$

(1) 关系模式 A 的码是 ()。

A. C B. (H, S) C. (H, R) D. (H, T)

(2) 关系模式 A 的规范化程度最高达到 ()。

A. 1NF B. 2NF C. 3NF D. BCNF

(3) 现将关系模式 A 分解为两个关系模式 A1 (C, T), A2 (H, R, S), 则其中 A1 的规范化程度达到 ()。

A. 1NF B. 2NF C. 3NF D. BCNF

参考答案: (1) A (2) B (3) B (4) A (5) D (6) B (7) C (8) B (9) B (10) C (11) D (12) A (13) D (14) D (15) B (16) B (17) D (18) B (19) C (20) C (21) C (22) A (23) A (24) A (25) D (26) B (27) B (28) B (29) B (30) B (31) D (32) B (33) B B D

二、简答题

1. 理解并给出下列术语的定义:

函数依赖、部分函数依赖、完全函数依赖、候选码、主码、外码、全码。

解析: 解答本题不能仅仅把《概论》上的定义写下来。关键是真正理解和运用这些概念。

答: 函数依赖: 设 $R(U)$ 是一个关系模式, U 是 R 的属性集合, X 和 Y 是 U 的子集。对于 $R(U)$ 的任意一个可能的关系 r , 如果 r 中不存在两个元组, 它们在 X 上的属性值相同, 而在 Y 上的属性值不同, 则称" X 函数确定 Y "或" Y 函数依赖于 X ", 记作 $X \rightarrow Y$ 。

解析: 1) 函数依赖是最基本的一种数据依赖, 也是最重要的一种数据依赖。

2) 函数依赖是属性之间的一种联系, 体现在属性值是否相等。由上面的定义可以知道, 如果 $X \rightarrow Y$, 则 r 中任意两个元组, 若它们在 X 上的属性值相同, 那么在 Y 上的属性值一定也相同。

3) 我们要从属性间实际存在的语义来确定他们之间的函数依赖, 即函数依赖反映了(描述了)现实世界的一种语义。

4) 函数依赖不是指关系模式 R 的在某个时刻的关系(值)满足的约束条件, 而是指 R 任何时刻的一切关系均要满足的约束条件。

答: 完全函数依赖、部分函数依赖: 在 $R(U)$ 中, 如果 $X \rightarrow Y$, 并且对于 X 的任何一个真子集 X' , 都有 $X' \not\rightarrow Y$, 则称 Y 对 X 完全函数依赖; 若 $X \rightarrow Y$, 但 Y 不完全函数依赖于 X , 则称 Y 对 X 部分函数依赖;

候选码、主码: 设 K 为 $R(U, F)$ 中的属性或属性组合, 若 $K \rightarrow U$ 则 K 为 R 的候选码。若候选码多于一个, 则选定其中的一个为主码。

答: 外码: 关系模式 R 中属性或属性组 X 并非 R 的码, 但 X 是另一个关系模式的码, 则称 X 是 R 的外部码也称外码。

全码: 整个属性组是码, 称为全码 (All-key)。

2. 建立一个关于系、学生、班级、学会等诸信息的关系数据库。

描述学生的属性有: 学号、姓名、出生年月、系名、班号、宿舍区。

描述班级的属性有: 班号、专业名、系名、人数、入校年份。

描述系的属性有: 系名、系号、系办公室地点、人数。

描述学会的属性有: 学会名、成立年份、地点、人数。

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

请给出关系模式, 写出每个关系模式的极小函数依赖集, 指出是否存在传递函数依赖, 对于函数依赖左部是多属性的情况讨论函数依赖是完全函数依赖, 还是部分函数依赖。

指出各关系的候选码、外部码, 有没有全码存在?

答: 关系模式: 学生 $S(S\#, SN, SB, DN, C\#, SA)$

班级 $C(C\#, CS, DN, CNUM, CDATE)$

系 $D(D\#, DN, DA, DNUM)$

学会 $P(PN, DATE1, PA, PNUM)$

学生--学会 $SP(S\#, PN, DATE2)$

其中, $S\#$ —学号, SN —姓名, SB —出生年月, SA —宿舍区

$C\#$ —班号, CS —专业名, $CNUM$ —班级人数, $CDATE$ —入校年份

$D\#$ —系号, DN —系名, DA —系办公室地点, $DNUM$ —系人数

PN—学会名, DATE1—成立年月, PA—地点, PNUM—学会人数, DATE2—入会年份
每个关系模式的极小函数依赖集:

S: $S\# \rightarrow SN, S\# \rightarrow SB, S\# \rightarrow C\#, C\# \rightarrow DN, DN \rightarrow SA$

C: $C\# \rightarrow CS, C\# \rightarrow CNUM, C\# \rightarrow CDATE, CS \rightarrow DN, (CS, CDATE) \rightarrow C\#$

D: $D\# \rightarrow DN, DN \rightarrow D\#, D\# \rightarrow DA, D\# \rightarrow DNUM$

P: $PN \rightarrow DATE1, PN \rightarrow PA, PN \rightarrow PNUM$

SP: $(S\#, PN) \rightarrow DATE2$

S 中存在传递函数依赖: $S\# \rightarrow DN, S\# \rightarrow SA, C\# \rightarrow SA$

C 中存在传递函数依赖: $C\# \rightarrow DN$

$(S\#, PN) \rightarrow DATE2$ 和 $(CS, CDATE) \rightarrow C\#$ 均为 SP 中的函数依赖, 是完全函数依赖
关系 候选码 外部码 全码

S $S\#$ $C\#$, DN 无

C $C\#$, (CS, CDATE) DN 无

D $D\#$ 和 DN 无 无

P PN 无 无

SP $(S\#, PN)$ $S\#, PN$ 无

3. 试由 Armstrong 公理系统推导出下面三条推理规则:

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

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

(3)分解规则: $X \rightarrow Y, Z$ 包含于 Y , 有 $X \rightarrow Z$

证: (1) 已知 $X \rightarrow Z$, 由增广律知 $XY \rightarrow YZ$, 又因为 $X \rightarrow Y$, 可得 $XX \rightarrow XY \rightarrow YZ$, 最后根据传递律得 $X \rightarrow YZ$ 。

(2) 已知 $X \rightarrow Y$, 据增广律得 $XW \rightarrow WY$, 因为 $WY \rightarrow Z$, 所以 $XW \rightarrow WY \rightarrow Z$, 通过传递律可知 $XW \rightarrow Z$ 。

(3) 已知 Z 包含于 Y , 根据自反律知 $Y \rightarrow Z$, 又因为 $X \rightarrow Y$, 所以由传递律可得 $X \rightarrow Z$ 。

4. 下面的结论哪些是正确的, 哪些是错误的? 对于错误的结论请给出理由或给出一个反例说明之。

(1) 任何一个二目关系都是属于 3NF 的。✓

(2) 任何一个二目关系都是属于 BCNF 的。✓

(3) 任何一个二目关系都是属于 4NF 的。✓

(5) 若 $R.A \rightarrow R.B, R.B \rightarrow R.C$, 则 $R.A \rightarrow R.C$ ✓

(6) 若 $R.A \rightarrow R.B, R.A \rightarrow R.C$, 则 $R.A \rightarrow R.(B, C)$ ✓

(7) 若 $R.B \rightarrow R.A, R.C \rightarrow R.A$, 则 $R.(B, C) \rightarrow R.A$ ✓

(8) 若 $R.(B, C) \rightarrow R.A$, 则 $R.B \rightarrow R.A, R.C \rightarrow R.A$ ×

反例: 关系模式 SC ($S\#, C\#, G$) ($S\#, C\#$) $\rightarrow G$, 但是 $S\# \not\rightarrow G, C\# \not\rightarrow G$

数据库设计

一、选择题

1. 数据流程图是用于描述结构化方法中 () 阶段的工具。

A. 概要设计 B. 可行性分析 C. 程序编码 D. 需求分析

2. 数据库设计中, 用 E-R 图描述信息结构但不涉及信息在计算机中的表示, 这是数据库设计的 ()。

A. 需求分析阶段 B. 逻辑设计阶段 C. 概念设计阶段 D. 物理设计阶段

3. 在数据库设计中, 将 E-R 图转换成关系数据模型的过程属于 ()。

A. 需求分析阶段 B. 逻辑设计阶段 C. 概念设计阶段 D. 物理设计阶段

4. 子模式 DDL 是用来描述 ()。

A. 数据库的总体逻辑结构 B. 数据库的局部逻辑结构
C. 数据库的物理存储结构 D. 数据库的概念结构

5. 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 ()。

A. 层次分析法和层次结构图 B. 数据流分析法和数据流程图
C. 实体联系法和实体联系图 D. 结构分析法和模块结构图

6. 在 E-R 模型向关系模型转换时, M: N 的联系转换为关系模式时, 其关键字是 ()。
- A.M 端实体的关键字 B.N 端实体的关键字 C.M、N 端实体的关键字组合 D.重新选取其他属性
7. 某学校规定, 每一个班级最多有 50 名学生, 至少有 10 名学生; 每一名学生必须属于一个班级。在班级与学生实体的联系中, 学生实体的基数是 ()。
- A. (0, 1) B. (1, 1) C. (1, 10) D. (10, 50)
8. 在关系数据库设计中, 设计关系模式是数据库设计中 () 阶段的任务。
- A. 逻辑设计阶段 B. 概念设计阶段 C. 物理设计阶段 D. 需求分析阶段
9. 关系数据库的规范化理论主要解决的问题是 ()。
- A.如何构造合适的数据逻辑结构 B.如何构造合适的数据物理结构
C.如何构造合适的应用程序界面 D.如何控制不同用户的数据操作权限
10. 数据库设计可划分为七个阶段, 每个阶段都有自己的设计内容, “为哪些关系, 在哪些属性上、键什么样的索引”这一设计内容应该属于 () 设计阶段。
- A. 概念设计 B. 逻辑设计 C. 物理设计 D. 全局设计
11. 假设设计数据库性能用“开销”, 即时间、空间及可能的费用来衡量, 则在数据库应用系统生存期中存在很多开销。其中, 对物理设计者来说, 主要考虑的是 ()。
- A. 规划开销 B. 设计开销 C. 操作开销 D. 维护开销
12. 数据库物理设计完成后, 进入数据库实施阶段, 下述工作中, () 一般不属于实施阶段的工作。
- A. 建立库结构 B. 系统调试 C. 加载数据 D. 扩充功能
13. 从 ER 图导出关系模型时, 如果实体间的联系是 M: N 的, 下列说法中正确的是 ()。
- A. 将 N 方关键字和联系的属性纳入 M 方的属性中 B. 将 M 方关键字和联系的属性纳入 N 方的属性中
C. 增加一个关系表示联系, 其中纳入 M 方和 N 方的关键字
D. 在 M 方属性和 N 方属性中均增加一个表示级别的属性
14. 在 ER 模型中, 如果有 3 个不同的实体集, 3 个 M: N 联系, 根据 ER 模型转换为关系模型的规则, 转换为关系的数目是 ()。
- A. 4 B. 5 C. 6 D. 7

参考答案: (1) D (2) C (3) B (4) B (5) C (6) C (7) B (8) A (9) A (10) C (11) C (12) D (13) C (14) C

二、简答题

1. 试述数据库设计过程。

*解析: 希望同学能够认真阅读《概论》的内容, 了解并掌握数据库设计过程。这里只概要列出数据库设计过程的六个阶段:

- 1) 需求分析
- 2) 概念结构设计
- 3) 逻辑结构设计
- 4) 数据库物理设计
- 5) 数据库实施
- 6) 数据库运行和维护

这是一个完整的实际数据库及其应用系统的设计过程。不仅包括设计数据库本身, 还包括数据库的实施、数据库运行和维护。

设计一个完善的数据库应用系统往往是上述六个阶段的不断反复。

2. 试述数据库设计过程的各个阶段上的设计描述。

*解析: 这是进一步了解数据库设计的具体内容。设计描述是指在各个阶段体现设计内容, 描述设计结果的各种文档、程序。

答: 各阶段的设计要点如下:

- 1) 需求分析: 准确了解与分析用户需求 (包括数据与处理)。
- 2) 概念结构设计: 通过对用户需求进行综合、归纳与抽象, 形成一个独立于具体 DBMS 的概念模型。
- 3) 逻辑结构设计: 将概念结构转换为某个 DBMS 所支持的数据模型, 并对其进行优化。
- 4) 数据库物理设计: 为逻辑数据模型选取一个最适合应用环境的物理结构 (包括存储结构和存取方法)。

5) 数据库实施：设计人员运用 DBMS 提供的数据库语言、工具及宿主语言，根据逻辑设计和物理设计的结果建立数据库，编制与调试应用程序，组织数据库入库，并进行试运行。

6) 数据库运行和维护：在数据库系统运行过程中对其进行评价、调整与修改。

3. 试述数据库设计过程中结构设计部分形成的数据库模式。

答：数据库结构设计的不同阶段形成数据库的各级模式，即：

在概念设计阶段形成独立于机器特点，独立于各个 DBMS 产品的概念模式，在本篇中就是 E-R 图；

在逻辑设计阶段将 E-R 图转换成具体的数据库产品支持的数据模型，如关系模型，形成数据库逻辑模式；然后在基本表的基础上再建立必要的视图(View)，形成数据的外模式；

在物理设计阶段，根据 DBMS 特点和处理的需要，进行物理存储安排，建立索引，形成数据库内模式。

概念模式是面向用户和设计人员的，属于概念模型的层次；逻辑模式、外模式、内模式是 DBMS 支持的模式，属于数据模型的层次。可以在 DBMS 中加以描述和存储。

4. 试述数据库设计的特点。

答：数据库设计既是一项涉及多学科的综合性技术又是一项庞大的工程项目。其主要特点有：

- 1) 数据库建设是硬件、软件和干件（技术与管理的界面）的结合。
- 2) 从软件设计的技术角度看，数据库设计应该和应用系统设计相结合，也就是说，整个设计过程中要把结构（数据）设计和行为（处理）设计密切结合起来。

5. 需求分析阶段的设计目标是什么？调查的内容是什么？

答：需求分析阶段的设计目标是通过详细调查现实世界要处理的对象（组织、部门、企业等），充分了解原系统（手工系统或计算机系统）工作概况，明确用户的各种需求，然后在此基础上确定新系统的功能。

调查的内容是“数据”和“处理”，即获得用户对数据库的如下要求：

（1）信息要求。指用户需要从数据库中获得信息的内容与性质。由信息要求可以导出数据要求，即在数据库中需要存储哪些数据。

（2）处理要求。指用户要完成什么处理功能，对处理的响应时间有什么要求，处理方式是批处理还是联机处理。

（3）安全性与完整性要求。

6. 数据字典的内容和作用是什么？

答：数据字典是系统中各类数据描述的集合。数据字典的内容通常包括：数据项、数据结构、数据流、数据存储、处理过程五个部分

其中数据项是数据的最小组成单位，若干个数据项可以组成一个数据结构。数据字典通过对数据项和数据结构的定义来描述数据流、数据存储的逻辑内容。

数据字典的作用：

数据字典是关于数据库中数据的描述，在需求分析阶段建立，是下一步进行概念设计的基础，并在数据库设计过程中不断修改、充实、完善。

（注意，数据库设计阶段形成的数据字典与后面讲到的数据字典不同，后者是 DBMS 关于数据库中数据的描述，当然两者是有联系的）。

7. 什么是数据库的概念结构？试述其特点 and 设计策略。

答：概念结构是信息世界的结构，即概念模型，其主要特点是：

（1）能真实、充分地反映现实世界，包括事物和事物之间的联系，能满足用户对数据的处理要求。是对现实世界的一个真实模型。

（2）易于理解，从而可以用它和不熟悉计算机的用户交换意见，用户的积极参与是数据库的设计成功的关键。

（3）易于更改，当应用环境和应用要求改变时，容易对概念模型修改和扩充。

（4）易于向关系、网状、层次等各种数据模型转换。

概念结构的设计策略通常有四种：

- 自顶向下。即首先定义全局概念结构的框架，然后逐步细化；
- 自底向上。即首先定义各局部应用的概念结构，然后将它们集成起来，得到全局概念结构；
- 逐步扩张。首先定义最重要的核心概念结构，然后向外扩充，以滚雪球的方式逐步生

成其他概念结构，直至总体概念结构；

- 混合策略。即将自顶向下和自底向上相结合，用自顶向下策略设计一个全局概念结构的框架，以它为骨架集成由自底向上策略中设计的各局部概念结构。

8. 什么叫数据抽象？试举例说明。

答：数据抽象是对实际的人、物、事和概念进行人为处理，抽取所关心的共同特性，忽略非本质的细节，并把这些特性用各种概念精确地加以描述，这些概念组成了某种模型。

如分类这种抽象是：定义某一类概念作为现实世界中一组对象的类型。这些对象具有某些共同的特性和行为。它抽象了对象值和型之间的“is member of”的语义。在 E-R 模型中，实体型就是这种抽象。例如在学校环境中，李英是老师，表示李英是教师类型中的一员，则教师是实体型，李英是教师实体型中的一个实体值，具有教师共同的特性和行为：在某个系某个专业教学，讲授某些课程，从事某个方向的科研。

9. 试述数据库概念结构设计的重要性和设计步骤。

答：重要性：

数据库概念设计是整个数据库设计的关键，将在需求分析阶段所得到的应用需求首先抽象为概念结构，以此作为各种数据模型的共同基础，从而能更好地、更准确地用某一 DBMS 实现这些需求。

设计步骤：

概念结构的设计方法有多种，其中最经常采用的策略是自底向上方法，该方法的设计步骤通常分为两步：第 1 步是抽象数据并设计局部视图，第 2 步是集成局部视图，得到全局的概念结构

10. 什么是 E-R 图？构成 E-R 图的基本要素是什么？

答：E-R 图为实体-联系图，提供了表示实体型、属性和联系的方法，用来描述现实世界的概念模型。

构成 E-R 图的基本要素是实体型、属性和联系，其表示方法为：

- 实体型：用矩形表示，矩形框内写明实体名；
- 属性：用椭圆形表示，并用无向边将其与相应的实体连接起来；
- 联系：用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时在无向边旁标上联系的类型（1:1，1:n 或 m:n）。

11. 为什么要视图集成？视图集成的方法是什么？

答：在对数据库系统进行概念结构设计时一般采用自底向上的设计方法，把繁杂的大系统分解子系统。首先设计各个子系统的局部视图，然后通过视图集成的方式将各子系统有机的融合起来，综合成一个系统的总视图。这样设计清晰，由简到繁。由于数据库系统是从整体角度看待和描述数据的，因此数据不再面向某个应用而是整个系统。因此必须进行视图集成，使得数据库能被全系统的多个用户、多个应用共享使用。

一般说来，视图集成可以有两种方式：

- 多个分 E-R 图一次集成；
- 逐步集成，用累加的方式一次集成两个分 E-R 图。

无论采用哪种方式，每次集成局部 E-R 图时都需要分两步走：

- （1）合并。解决各分 E-R 图之间的冲突，将各分 E-R 图合并起来生成初步 E-R 图。
- （2）修改和重构。消除不必要的冗余，生成基本 E-R 图。

12. 什么是数据库的逻辑结构设计？试述其设计步骤。

答：数据库的逻辑结构设计就是把概念结构设计阶段设计好的基本 E-R 图转换为与选用的 DBMS 产品所支持的数据模型相符合的逻辑结构。

设计步骤为：

- （1）将概念结构转换为一般的关系、网状、层次模型；
- （2）将转换来的关系、网状、层次模型向特定 DBMS 支持下的数据模型转换；
- （3）对数据模型进行优化。

13. 试述把 E-R 图转换为 DBTG 模型和关系模型的转换规则。

答：E-R 图向 DBTG 模型的转换规则：

- 1) 每个实体型转换为记录型，实体的属性转换为记录的数据项；
- 2) 实体型之间 1:n ($n \geq 1$) 的联系转换为一个系，没有任何联系的实体型转换为奇异系；
- 3) K ($K \geq 2$) 个实体型之间多对多的联系，引入一个连结记录，形成 K 个实体型和连结记录之间的 K 个系。

连结记录的属性由诸首记录的码及联系属性所组成；

4)同一实体型内的 1: n, n: m 联系，引入连结记录，转换为两个系。

*解析：根据我国实际，网状，层次数据库系统已很少使用，因此《概论》第三版把它们删去了，有关的主要概念放在第一章数据模型中介绍。对于 DBTG 模型的许多概念也介绍得很简单。本题的内容已经超出了书上的内容，同学们只要了解就可以了。但是，下面 E-R 图向关系模型的转换规则要求同学必须掌握，并且能够举一反三。

答：E-R 图向关系模型的转换规则：

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

对于实体间的联系则有以下不同的情况：

(1) 一个 1: 1 联系可以转换为一个独立的关系模式，也可以与任意一端对应的关系模式合并。如果转换为一个独立的关系模式，则与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，每个实体的码均是该关系的候选码。如果与某一端实体对应的关系模式合并，则需要在该关系模式的属性中加入另一个关系模式的码和联系本身的属性。

(2) 一个 1: n 联系可以转换为一个独立的关系模式，也可以与 n 端对应的关系模式合并。如果转换为一个独立的关系模式，则与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，而关系的码为 n 端实体的码。

(3) 一个 m: n 联系转换为一个关系模式。与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，各实体码的组合组成该关系的码，或码的一部分。

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

(5) 具有相同码的关系模式可合并。

***14. 你能给出由 E-R 图转换为 IMS 模型的转换规则吗？**

答：E-R 图向 IMS 模型的转换规则：

1)每个实体型转换为记录型，实体的属性转换为记录的数据项；

2)实体型之间 1: n ($n \geq 1$) 的联系转换记录型之间的有向边；

3)实体型之间 m: n ($m > 1, n > 1$) 的联系则分解成一对多联系，再根据 2)转换；

4)K ($K \geq 2$) 个实体型之间多对多的联系，可先转换成多对两个实体型之间的联系，再根据 3)转换。

*解析：IMS 是 IBM 公司的层次数据库管理系统。IMS 模型是层次模型。E-R 图向 IMS 模型转换的另一种方法是，先把 E-R 图转换为网状模型，再利用 IMS 逻辑数据库 LDB 的概念来表示网状模型。详细方法这里从略。

数据库恢复技术

一、选择题

1. 一个事务的执行，要么全部完成，要么全部不做，一个事务中对数据库的所有操作都是一个不可分割的操作序列的属性是 ()。

A. 原子性 B. 一致性 C. 独立性 D. 持久性

2. 表示两个或多个事务可以同时运行而不互相影响的是 ()。

A. 原子性 B. 一致性 C. 独立性 D. 持久性

3. 事务的持续性是指 ()

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。

4. SQL 语言中的 COMMIT 语句的主要作用是 ()。

A. 结束程序 B. 返回系统 C. 提交事务 D. 存储数据

5. SQL 语言中用 () 语句实现事务的回滚

A. CREATE TABLE B. ROLLBACK C. GRANT 和 REVOKE D. COMMIT

6. 若系统在运行过程中，由于某种硬件故障，使存储在外存上的数据部分损失或全部损失，这种情况称为 ()。

A. 介质故障 B. 运行故障 C. 系统故障 D. 事务故障

7. 在 DBMS 中实现事务持久性的子系统是 ()。

A. 安全管理子系统 B. 完整性管理子系统 C. 并发控制子系统 D. 恢复管理子系统

8. 后援副本的作用是 ()。

A. 保障安全性 B. 一致性控制 C. 故障后的恢复 D. 数据的转储

9. 事务日志用于保存 ()。

A. 程序运行过程 B. 程序的执行结果 C. 对数据的更新操作 D. 数据操作

10. 数据库恢复的基础是利用转储的冗余数据。这些转储的冗余数据包括 ()。

A. 数据字典、应用程序、审计档案、数据库后备副本 B. 数据字典、应用程序、审计档案、日志文件
C. 日志文件、数据库后备副本 D. 数据字典、应用程序、数据库后备副本

选择题答案: (1) A (2) C (3) B (4) C (5) B (6) A (7) D (8) C (9) C (10) C

二、简答题

1. 试述事务的概念及事务的四个特性。

答: 事务是用户定义的一个数据库操作序列, 这些操作要么全做要么全不做, 是一个不可分割的工作单位。

事务具有四个特性: 原子性 (Atomicity)、一致性 (Consistency)、隔离性 (Isolation) 和持续性 (Durability)。

这个四个特性也简称为 ACID 特性。

原子性: 事务是数据库的逻辑工作单位, 事务中包括的诸操作要么都做, 要么都不做。

一致性: 事务执行的结果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。

隔离性: 一个事务的执行不能被其他事务干扰。即一个事务内部的操作及使用的数据对其他并发事务是隔离的, 并发执行的各个事务之间不能互相干扰。

持续性: 持续性也称永久性 (Permanence), 指一个事务一旦提交, 它对数据库中数据的改变就应该是永久性的。接下来的其他操作或故障不应该对其执行结果有任何影响。

2. 为什么事务非正常结束时会影响数据库数据的正确性, 请列举一例说明之。

答: 事务执行的结果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。如果数据库系统运行中发生故障, 有些事务尚未完成就被迫中断, 这些未完成事务对数据库所做的修改有一部分已写入物理数据库, 这时数据库就处于一种不正确的状态, 或者说不一致的状态。

例如某工厂的库存管理系统中, 要把数量为 Q 的某种零件从仓库 1 移到仓库 2 存放。

则可以定义一个事务 T , T 包括两个操作: $Q1=Q1-Q$, $Q2=Q2+Q$ 。如果 T 非正常终止时只做了第一个操作, 则数据库就处于不一致性状态, 库存量无缘无故少了 Q 。

3. 数据库中为什么要有恢复子系统? 它的功能是什么?

答: 因为计算机系统中硬件的故障、软件的错误、操作员的失误以及恶意的破坏是不可避免的, 这些故障轻则造成运行事务非正常中断, 影响数据库中数据的正确性, 重则破坏数据库, 使数据库中全部或部分数据丢失, 因此必须要有恢复子系统。

恢复子系统的功能是: 把数据库从错误状态恢复到某一已知的正确状态 (亦称为一致状态或完整状态)。

4. 数据库运行中可能产生的故障有哪几类? 哪些故障影响事务的正常执行? 哪些故障破坏数据库数据?

答: 数据库系统中可能发生各种各样的故障, 大致可以分以下几类:

- (1) 事务内部的故障;
- (2) 系统故障;
- (3) 介质故障;
- (4) 计算机病毒。

事务故障、系统故障和介质故障影响事务的正常执行; 介质故障和计算机病毒破坏数据库数据。

5. 数据库恢复的基本技术有哪些?

答: 数据转储和登记日志文件是数据库恢复的基本技术。

当系统运行过程中发生故障, 利用转储的数据库后备副本和日志文件就可以将数据库恢复到故障前的某个一致性状态。

6. 数据库转储的意义是什么? 试比较各种数据转储方法。

答: 数据转储是数据库恢复中采用的基本技术。所谓转储即 DBA 定期地将数据库复制到磁带或另一个磁盘上保存起来的过程。当数据库遭到破坏后可以将后备副本重新装入, 将数据库恢复到转储时的状态。

静态转储：在系统中无运行事务时进行的转储操作。静态转储简单，但必须等待正运行的用户事务结束才能进行。同样，新的事务必须等待转储结束才能执行。显然，这会降低数据库的可用性。

动态转储：指转储期间允许对数据库进行存取或修改。动态转储可克服静态转储的缺点，它不用等待正在运行的用户事务结束，也不会影响新事务的运行。但是，转储结束后后援副本上的数据并不能保证正确有效。因为转储期间运行的事务可能修改了某些数据，使得后援副本上的数据不是数据库的一致版本。

为此，必须把转储期间各事务对数据库的修改活动登记下来，建立日志文件（log file）。这样，后援副本加上日志文件就能得到数据库某一时刻的正确状态。

转储还可以分为海量转储和增量转储两种方式。

海量转储是指每次转储全部数据库。增量转储则指每次只转储上一次转储后更新过的数据。从恢复角度看，使用海量转储得到的后备副本进行恢复一般说来更简单些。但如果数据库很大，事务处理又十分频繁，则增量转储方式更实用更有效。

7. 什么是日志文件？为什么要设立日志文件？

答：（1）日志文件是用来记录事务对数据库的更新操作的文件。

（2）设立日志文件的目的是：进行事务故障恢复；进行系统故障恢复；协助后备副本进行介质故障恢复。

8. 登记日志文件时为什么必须先写日志文件，后写数据库？

答：把对数据的修改写到数据库中和把表示这个修改的日志记录写到日志文件中是两个不同的操作。有可能在这两个操作之间发生故障，即这两个写操作只完成了一个。

如果先写了数据库修改，而在运行记录中没有登记这个修改，则以后就无法恢复这个修改了。如果先写日志，但没有修改数据库，在恢复时只不过是多执行一次 UNDO 操作，并不会影响数据库的正确性。所以一定要先写日志文件，即首先把日志记录写到日志文件中，然后写数据库的修改。

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

答：事务故障的恢复：

事务故障的恢复是由 DBMS 自动完成的，对用户是透明的。

DBMS 执行恢复步骤是：

（1）反向扫描文件日志（即从最后向前扫描日志文件），查找该事务的更新操作。

（2）对该事务的更新操作执行逆操作。即将日志记录中“更新前的值”写入数据库。

（3）继续反向扫描日志文件，做同样处理。

（4）如此处理下去，直至读到此事务的开始标记，该事务故障的恢复就完成了。

答：系统故障的恢复：

系统故障可能会造成数据库处于不一致状态：

一是未完成事务对数据库的更新可能已写入数据库；

二是已提交事务对数据库的更新可能还留在缓冲区，来不及写入数据库。

因此恢复操作就是要撤销(UNDO)故障发生时未完成的事务，重做(RED0)已完成的事务。

系统的恢复步骤是：

（1）正向扫描日志文件，找出在故障发生前已经提交的事务队列（RED0 队列）和未完成的事务队列（UNDO 队列）。

（2）对撤销队列中的各个事务进行 UNDO 处理。

进行 UNDO 处理的方法是，反向扫描日志文件，对每个 UNDO 事务的更新操作执行逆操作，即将日志记录中“更新前的值”（Before Image）写入数据库。

（3）对重做队列中的各个事务进行 REDO 处理。

进行 REDO 处理的方法是：正向扫描日志文件，对每个 REDO 事务重新执行日志文件登记的操作。即将日志记录中“更新后的值”（After Image）写入数据库。

*解析：

在第（1）步中如何找出 REDO 队列和 UNDO 队列？请大家思考一下。

下面给出一个算法：

1) 建立两个事务队列：

- UNDO-LIST: 需要执行 undo 操作的事务集合;
- REDO-LIST: 需要执行 redo 操作的事务集合;

两个事务队列初始均为空。

2) 从日志文件头开始, 正向扫描日志文件

- 如有新开始 (遇到 Begin Transaction) 的事务 T_i , 把 T_i 暂时放入 UNDO-LIST 队列;
- 如有提交的事务 (遇到 End Transaction) T_j , 把 T_j 从 UNDO-LIST 队列移到 REDO-LIST 队列;

直到日志文件结束

答: 介质故障的恢复:

介质故障是最严重的一种故障。

恢复方法是重装数据库, 然后重做已完成的事务。具体过程是:

(1) DBA 装入最新的数据库后备副本 (离故障发生时刻最近的转储副本), 使数据库恢复到转储时的一致性状态。

(2) DBA 装入转储结束时刻的日志文件副本

(3) DBA 启动系统恢复命令, 由 DBMS 完成恢复功能, 即重做已完成的事务。

*解析

1) 我们假定采用的是静态转储, 因此第 (1) 步装入数据库后备副本便可以了。

2) 如果采用的是静动态转储, 第 (1) 步装入数据库后备副本还不够, 还需同时装入转储开始时刻的日志文件副本, 经过处理后才能得到正确的数据库后备副本。

3) 第 (2) 步重做已完成的事务的算法是:

a. 正向扫描日志文件, 找出故障发生前已提交的事务的标识, 将其记入重做队列

b. 再一次正向扫描日志文件, 对重做队列中的所有事务进行重做处理。即将日志记录中 “更新后的值” 写入数据库。

10. 具有检查点的恢复技术有什么优点?

答: 利用日志技术进行数据库恢复时, 恢复子系统必须搜索日志, 确定哪些事务需要 REDO, 哪些事务需要 UNDO。一般来说, 需要检查所有日志记录。这样做有两个问题:

一是搜索整个日志将耗费大量的时间。

二是很多需要 REDO 处理的事务实际上已经将它们的更新操作结果写到数据库中了, 恢复子系统又重新执行了这些操作, 浪费了大量时间。

检查点技术就是为了解决这些问题。

11. 试述使用检查点方法进行恢复的步骤。

答: ① 从重新开始文件中找到最后一个检查点记录在日志文件中的地址, 由该地址在日志文件中找到最后一个检查点记录。

② 由该检查点记录得到检查点建立时刻所有正在执行的事务清单 ACTIVE-LIST。

这里建立两个事务队列:

- UNDO-LIST: 需要执行 undo 操作的事务集合;
- REDO-LIST: 需要执行 redo 操作的事务集合;

把 ACTIVE-LIST 暂时放入 UNDO-LIST 队列, REDO 队列暂为空。

③ 从检查点开始正向扫描日志文件

- 如有新开始的事务 T_i , 把 T_i 暂时放入 UNDO-LIST 队列;
 - 如有提交的事务 T_j , 把 T_j 从 UNDO-LIST 队列移到 REDO-LIST 队列, 直到日志文件结束;
- ④ 对 UNDO-LIST 中的每个事务执行 UNDO 操作, 对 REDO-LIST 中的每个事务执行 REDO 操作。

12. 什么是数据库镜像? 它有什么用途?

答: 数据库镜像即根据 DBA 的要求, 自动把整个数据库或者其中的部分关键数据复制到另一个磁盘上。每当主数据库更新时, DBMS 自动把更新后的数据复制过去, 即 DBMS 自动保证镜像数据与主数据的一致性。

数据库镜像的用途有:

一是用于数据库恢复。当出现介质故障时, 可由镜像磁盘继续提供使用, 同时 DBMS 自动利用镜像磁盘数据进行数据库的恢复, 不需要关闭系统和重装数据库副本。

二是提高数据库的可用性。在没有出现故障时，当一个用户对某个数据加排它锁进行修改时，其他用户可以读镜像数据库上的数据，而不必等待该用户释放锁。

***13. 试述你了解的某一个实际的 DBMS 产品中采用的恢复策略。**

答：下面简单介绍一下 Oracle 的恢复技术：

Oracle 中恢复机制也采用了转储和登记日志文件两个技术。

Oracle 向 DBA 提供了多种转储后备副本的方法，如文件拷贝、利用 Oracle 的 Export 实用程序、用 SQL 命令 Spool 以及自己编程实现等。相应地，Oracle 也提供了多种重装后备副本的方法，如文件拷贝、利用 Oracle 的 Import 实用程序、利用 SQL*LOADER 以及自己编程实现等。

在 Oracle 早期版本（V.5）中，日志文件以数据块为单位，也就是说，Oracle 的恢复操作是基于数据块的，不是基于操作的。Oracle 中记录数据库更新前的旧值的日志文件称为数据库前像文件（Before Image，简称 BI 文件），记录数据库更新后的新值的日志文件称为数据库的后像文件（After Image，简称 AI 文件）。BI 文件是必须配置的，AI 文件是可以任选的。

Oracle7 为了能够在出现故障时更有效地恢复数据，也为了解决读“脏”数据问题，提供了 REDO 日志文件和回滚段(Rollback Segment)。REDO 日志文件中记录了被更新数据的前像和后像。回滚段记录更新数据的前像，设在数据库缓冲区中。在利用日志文件进行故障恢复时，为减少扫描日志文件的遍数，Oracle7 首先扫描 REDO 日志文件，重做所有操作，包括未正常提交的事务的操作，然后再根据回滚段中的数据，撤销未正常提交的事务的操作。

详细技术希望同学自己设法了解 Oracle 最新版本的介绍，例如通过 INTERNET 访问 Oracle 公司的网站。也可以了解其他 DBMS 厂商的产品情况。

***14. 试用恢复的基本技术设计一个恢复子系统，给出这个子系统的恢复策略，包括：**

- (a) 当产生某一类故障时如何恢复数据库的方法；
- (b) 日志文件的结构；
- (c) 登记日志文件的方法；
- (d) 利用日志文件恢复事务的方法；
- (e) 转储的类型；
- (f) 转储的后备副本和日志文件如何配合使用。

***解析：**这是一个大作业。可以综合复习和运用学到的知识。设计一个恢复子系统。

例如，日志文件的结构你可以记录为单位，也可以以数据块为单位。不同的日志文件结构，登记的日志内容，日志文件恢复事务的方法也就不同了。

对于研究生，还应该上机模拟实现你设计的恢复子系统。

数据库并发控制

一、选择题

1. 为了防止一个用户的工作不适当地影响另一个用户，应该采取（ ）。
A. 完整性控制 B. 访问控制 C. 安全性控制 D. 并发控制
2. 解决并发操作带来的数据不一致问题普遍采用（ ）技术。
A. 封锁 B. 存取控制 C. 恢复 D. 协商
3. 下列不属于并发操作带来的问题是（ ）。
A. 丢失修改 B. 不可重复读 C. 死锁 D. 脏读
4. DBMS 普遍采用（ ）方法来保证调度的正确性。
A. 索引 B. 授权 C. 封锁 D. 日志
5. 事务 T 在修改数据 R 之前必须先对其加 X 锁，直到事务结束才释放，这是（ ）。
A. 一级封锁协议 B. 二级封锁协议 C. 三级封锁协议 D. 零级封锁协议
6. 如果事务 T 获得了数据项 Q 上的排他锁，则 T 对 Q（ ）。
A. 只能读不能写 B. 只能写不能读 C. 既可读又可写 D. 不能读也不能写
7. 设事务 T1 和 T2，对数据库中地数据 A 进行操作，可能有如下几种情况，请问哪一种不会发生冲突操作（ ）。
A. T1 正在写 A，T2 要读 A B. T1 正在写 A，T2 也要写 A
C. T1 正在读 A，T2 要写 A D. T1 正在读 A，T2 也要读 A

8. 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（ ）。

- A. 一个是 DELETE，一个是 SELECT B. 一个是 SELECT，一个是 DELETE
C. 两个都是 UPDATE D. 两个都是 SELECT

9. 在数据库系统中，死锁属于（ ）。

- A. 系统故障 B. 事务故障 C. 介质故障 D. 程序故障

参考答案： (1) D (2) A (3) C (4) C (5) A (6) C (7) D (8) D (9) B

二、简答题

1. 在数据库中为什么要并发控制？

答：数据库是共享资源，通常有许多个事务同时在运行。

当多个事务并发地存取数据库时就会产生同时读取和/或修改同一数据的情况。若对并发操作不加控制就可能造成存取和存储不正确的数据，破坏数据库的一致性。所以数据库管理系统必须提供并发控制机制。

2. 并发操作可能会产生哪几类数据不一致？用什么方法能避免各种不一致的情况？

答：并发操作带来的数据不一致性包括三类：丢失修改、不可重复读和读“脏”数据。

(1) 丢失修改 (Lost Update)

两个事务 T1 和 T2 读入同一数据并修改，T2 提交的结果破坏了（覆盖了）T1 提交的结果，导致 T1 的修改被丢失。

(2) 不可重复读 (Non-Repeatable Read)

不可重复读是指事务 T1 读取数据后，事务 T2 执行更新操作，使 T1 无法再现前一次读取结果。

(3) 读“脏”数据 (Dirty Read)

读“脏”数据是指事务 T1 修改某一数据，并将其写回磁盘，事务 T2 读取同一数据后，T1 由于某种原因被撤销，这时 T1 已修改过的数据恢复原值，T2 读到的数据就与数据库中的数据不一致，则 T2 读到的数据就为“脏”数据，即不正确的数据。

避免不一致性的方法和技术就是并发控制。最常用的并发控制技术是封锁技术。

也可以用其他技术，例如在分布式数据库系统中可以采用时间戳方法来进行并发控制。

3. 什么是封锁？

答：封锁就是事务 T 在对某个数据对象例如表、记录等操作之前，先向系统发出请求，对其加锁。加锁后事务 T 就对该数据对象有了一定的控制，在事务 T 释放它的锁之前，其他的事务不能更新此数据对象。

封锁是实现并发控制的一个非常重要的技术。

4. 基本的封锁类型有几种？试述它们的含义。

答：基本的封锁类型有两种：排它锁(Exclusive Locks，简称 X 锁) 和共享锁(Share Locks，简称 S 锁)。

排它锁又称为写锁。若事务 T 对数据对象 A 加上 X 锁，则只允许 T 读取和修改 A，其他任何事务都不能再对 A 加任何类型的锁，直到 T 释放 A 上的锁。这就保证了其他事务在 T 释放 A 上的锁之前不能再读取和修改 A。

共享锁又称为读锁。若事务 T 对数据对象 A 加上 S 锁，则事务 T 可以读 A 但不能修改 A，其他事务只能再对 A 加 S 锁，而不能加 X 锁，直到 T 释放 A 上的 S 锁。这就保证了其他事务可以读 A，但在 T 释放 A 上的 S 锁之前不能对 A 做任何修改。

5. 什么是封锁协议？不同级别的封锁协议的主要区别是什么？

答：在运用封锁技术对数据加锁时，要约定一些规则。例如，在运用 X 锁和 S 锁对数据对象加锁时，要约定何时申请 X 锁或 S 锁、何时释放封锁等。这些约定或者规则称为封锁协议 (Locking Protocol)。对封锁方式约定不同的规则，就形成了各种不同的封锁协议。不同级别的封锁协议，例如《概论》中介绍的三级封锁协议，三级协议的主要区别在于什么操作需要申请封锁，何时申请封锁以及何时释放锁（即持锁时间的长短）。

一级封锁协议：事务 T 在修改数据 R 之前必须先对其加 X 锁，直到事务结束才释放。

二级封锁协议：一级封锁协议加上事务 T 在读取数据 R 之前必须先对其加 S 锁，读完后即可释放 S 锁。

三级封锁协议：一级封锁协议加上事务 T 在读取数据 R 之前必须先对其加 S 锁，直到事务结束才释放。

6. 不同封锁协议与系统一致性级别的关系是什么？

答：不同的封锁协议对应不同的一致性级别。

一级封锁协议可防止丢失修改，并保证事务 T 是可恢复的。在一级封锁协议中，对读数据是不加 S 锁的，所以它不能保证可重复读和不读“脏”数据。

二级封锁协议除防止了丢失修改，还可进一步防止读“脏”数据。在二级封锁协议中，由于读完数据后立即释放 S 锁，所以它不能保证可重复读。

在三级封锁协议中，无论是读数据还是写数据都加长锁，即都要到事务结束时才释放封锁。所以三级封锁协议除防止了丢失修改和不读“脏”数据外，还进一步防止了不可重复读。

7. 试述活锁的产生原因和解决方法。

答：活锁产生的原因：当一系列封锁不能按照其先后顺序执行时，就可能导致一些事务无限期等待某个封锁，从而导致活锁。

避免活锁的简单方法是采用先来先服务的策略。当多个事务请求封锁同一数据对象时，封锁子系统按请求封锁的先后次序对事务排队，数据对象上的锁一旦释放就批准申请队列中第一个事务获得锁。

8. 请给出预防死锁的若干方法。

答：在数据库中，产生死锁的原因是两个或多个事务都已封锁了一些数据对象，然后又都请求已被其他事务封锁的数据加锁，从而出现死等待。

防止死锁的发生其实就是要破坏产生死锁的条件。预防死锁通常有两种方法：

（1）一次封锁法

要求每个事务必须一次将所有要使用的数据全部加锁，否则就不能继续执行。

（2）顺序封锁法

预先对数据对象规定一个封锁顺序，所有事务都按这个顺序实行封锁。

不过，预防死锁的策略不大适合数据库系统的特点。

9. 请给出检测死锁发生的一种方法，当发生死锁后如何解除死锁？

答：数据库系统一般采用允许死锁发生，DBMS 检测到死锁后加以解除的方法。

DBMS 中诊断死锁的方法与操作系统类似，一般使用超时法或事务等待图法。

超时法是：如果一个事务的等待时间超过了规定的时限，就认为发生了死锁。超时法实现简单，但有可能误判死锁，事务因其他原因长时间等待超过时限，系统会误认为发生了死锁。若时限设置得太长，又不能及时发现死锁发生。

DBMS 并发控制子系统检测到死锁后，就要设法解除。通常采用的方法是选择一个处理死锁代价最小的事务，将其撤消，释放此事务持有的所有锁，使其他事务得以继续运行下去。当然，对撤销的事务所执行的数据修改操作必须加以恢复。

10. 什么样的并发调度是正确的调度？

答：可串行化（Serializable）的调度是正确的调度。

可串行化的调度的定义：多个事务的并发执行是正确的，当且仅当其结果与按某一次序串行地执行它们时的结果相同，我们称这种调度策略为可串行化的调度。

11. 试述两段锁协议的概念。

答：两段锁协议是指所有事务必须分两个阶段对数据项加锁和解锁。

- 在对任何数据进行读、写操作之前，首先要申请并获得对该数据的封锁；
- 在释放一个封锁之后，事务不再申请和获得任何其他封锁。

“两段”的含义是，事务分为两个阶段：

第一阶段是获得封锁，也称为扩展阶段。在这阶段，事务可以申请获得任何数据项上的任何类型的锁，但是不能释放任何锁。

第二阶段是释放封锁，也称为收缩阶段。在这阶段，事务释放已经获得的锁，但是不能再申请任何锁。

12. 为什么要引进意向锁？意向锁的含义是什么？

答：引进意向锁是为了提高封锁子系统的效率。该封锁子系统支持多种封锁粒度。

原因是：在多粒度封锁方法中一个数据对象可能以两种方式加锁—显式封锁和隐式封锁。因此系统在对某一数据对象加锁时不仅要检查该数据对象上有没有（显式和隐式）封锁与之冲突；还要检查其所有上级结点和所有下级结点，看申请的封锁是否与这些结点上的（显式和隐式）封锁冲突；显然，这样的检查方法效率很低。为此引进了意向锁。

意向锁的含义是：对任一结点加锁时，必须先对它的上层结点加意向锁。

例如事务 T 要对某个元组加 X 锁，则首先要对关系和数据库加 IX 锁。换言之，对关系和数据库加 IX 锁，表

示它的后裔结点—某个元组拟（意向）加 X 锁。

引进意向锁后，系统对某一数据对象加锁时不必逐个检查与下一级结点的封锁冲突了。例如，事务 T 要对关系 R 加 X 锁时，系统只要检查根结点数据库和 R 本身是否已加了不相容的锁（如发现已经加了 IX，则与 X 冲突），而不再需要搜索和检查 R 中的每一个元组是否加了 X 锁或 S 锁。

13. 试述常用的意向锁：IS 锁，IX 锁，SIX 锁，给出这些锁的相容矩阵。

答：IS 锁：如果对一个数据对象加 IS 锁，表示它的后裔结点拟（意向）加 S 锁。例如，要对某个元组加 S 锁，则要首先对关系和数据库加 IS 锁

IX 锁：如果对一个数据对象加 IX 锁，表示它的后裔结点拟（意向）加 X 锁。例如，要对某个元组加 X 锁，则要首先对关系和数据库加 IX 锁。

SIX 锁：如果对一个数据对象加 SIX 锁，表示对它加 S 锁，再加 IX 锁，即 $SIX = S + IX$ 。

相容矩阵（略）

14. 理解并解释下列术语的含义：封锁，活锁，死锁，排它锁，共享锁，并发事务的调度，可串行化的调度，两段锁协议。

答：（略，已经在上面有关习题中解答）

* 15. 试述你了解的某一个实际的 DBMS 产品的并发控制机制。

答：（略）

数据库安全性

一、选择题

1. 以下（ ）不属于实现数据库系统安全性的主要技术和方法。

A. 存取控制技术 B. 视图技术 C. 审计技术 D. 出入机房登记和加锁

2. SQL 中的视图提高了数据库系统的（ ）。

A. 完整性 B. 并发控制 C. 隔离性 D. 安全性

3. SQL 语言的 GRANT 和 REMOVE 语句主要是用来维护数据库的（ ）。

A. 完整性 B. 可靠性 C. 安全性 D. 一致性

4. 在数据库的安全性控制中，授权的数据对象的（ ），授权子系统就越灵活。

A. 范围越小 B. 约束越细致 C. 范围越大 D. 约束范围大

选择题答案：

(1) D (2) D (3) C (4) A

三、简答题

1. 什么是数据库的安全性？

答：数据库的安全性是指保护数据库以防止不合法的使用所造成的数据泄露、更改或破坏。

2. 数据库安全性和计算机系统的安全性有什么关系？

答：安全性问题不是数据库系统所独有的，所有计算机系统都有这个问题。只是在数据库系统中大量数据集中存放，而且为许多最终用户直接共享，从而使安全性问题更为突出。

系统安全保护措施是否有效是数据库系统的主要指标之一。

数据库的安全性和计算机系统的安全性，包括操作系统、网络系统的安全性是紧密联系、相互支持的，

3. 试述实现数据库安全性控制的常用方法和技术。

答：实现数据库安全性控制的常用方法和技术有：

1) 用户标识和鉴别：该方法由系统提供一定的方式让用户标识自己的名字或身份。每次用户要求进入系统时，由系统进行核对，通过鉴定后才提供系统的使用权。

2) 存取控制：通过用户权限定义和合法权检查确保只有合法权限的用户访问数据库，所有未被授权的人员无法存取数据。例如 C2 级中的自主存取控制（DAC），B1 级中的强制存取控制（MAC）；

3) 视图机制：为不同的用户定义视图，通过视图机制把要保密的数据对无权存取的用户隐藏起来，从而自动地对数据提供一定程度的安全保护。

4) 审计：建立审计日志，把用户对数据库的所有操作自动记录下来放入审计日志中，DBA 可以利用审计跟踪的信息，重现导致数据库现有状况的一系列事件，找出非法存取数据的人、时间和内容等。

5)数据加密：对存储和传输的数据进行加密处理，从而使得不知道解密算法的人无法获知数据的内容。

具体内容请参见《概论》9.2。

4.什么是数据库中的自主存取控制方法和强制存取控制方法？

答：自主存取控制方法：定义各个用户对不同数据对象的存取权限。当用户对数据库访问时首先检查用户的存取权限。防止不合法用户对数据库的存取。

强制存取控制方法：每一个数据对象被（强制地）标以一定的密级，每一个用户也被（强制地）授予某一个级别的许可证。系统规定只有具有某一许可证级别的用户才能存取某一个密级的数据对象。

*解析：自主存取控制中自主的含义是：用户可以将自己拥有的存取权限“自主”地授予别人。即用户具有一定的“自主”权。

5.SQL 语言中提供了哪些数据控制（自主存取控制）的语句？请试举几例说明它们的使用方法。

答：SQL 中的自主存取控制是通过 GRANT 语句和 REVOKE 语句来实现的。如：

GRANT SELECT, INSERT ON Student TO 王平 WITH GRANT OPTION;

就将 Student 表的 SELECT 和 INSERT 权限授予了用户王平，后面的“WITH GRANT OPTION”子句表示用户王平同时也获得了“授权”的权限，即可以把得到的权限继续授予其他用户。

REVOKE INSERT ON Student FROM 王平 CASCADE;

就将 Student 表的 INSERT 权限从用户王平处收回，选项 CASCADE 表示，如果用户王平将 Student 的 INSERT 权限又转授给了其他用户，那么这些权限也将从其他用户处收回。

6.今有两个关系模式：

职工（职工号，姓名，年龄，职务，工资，部门号）

部门（部门号，名称，经理名，地址，电话号）

请用 SQL 的 GRANT 和 REVOKE 语句（加上视图机制）完成以下授权定义或存取控制功能：

(a) 用户王明对两个表有 SELECT 权力；

GRANT SELECT ON 职工，部门 TO 王明；

(b) 用户李勇对两个表有 INSERT 和 DELETE 权力；

GRANT INSERT, DELETE ON 职工，部门 TO 李勇；

(c) *每个职工只对自己的记录有 SELECT 权力；

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

这里假定系统的 GRANT 语句支持 WHEN 子句和 USER () 的使用。用户将自己的名字作为 ID。注意，不同的系统这些扩展语句可能是不同的。读者应该了解你使用的 DBMS 产品的扩展语句。

(d) 用户刘星对职工表有 SELECT 权力，对工资字段具有更新权力；

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

(e) 用户张新具有修改这两个表的结构权力；

GRANT ALTER TABLE ON 职工，部门 TO 张新；

(f) 用户周平具有对两个表所有权力(读，插，改，删数据)，并具有给其他用户授权的权力；

GRANT ALL PRIVILIGES ON 职工，部门 TO 周平 WITH GRANT OPTION;

(g) 用户杨兰具有从每个部门职工中 SELECT 最高工资，最低工资，平均工资的权力，他不能查看每个人的工资。

答：首先建立一个视图。然后对这个视图定义杨兰的存取权限。

CREATE VIEW 部门工资 AS

SELECT 部门.名称, MAX (工资), MIN (工资), AVG (工资)

FROM 职工, 部门

WHERE 职工.部门号 = 部门. 部门号

GROUP BY 职工.部门号;

GRANT SELECT ON 部门工资 TO 杨兰;

7.把习题 8 中(a)~(g) 的每一种情况，撤销各用户所授予的权力。

答：(a) REVOKE SELECT ON 职工，部门 FROM 王明；

(b) REVOKE INSERT, DELETE ON 职工，部门 FROM 李勇；

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

这里假定用户将自己的名字作为 ID, 且系统的 REVOKE 语句支持 WHEN 子句, 系统也支持 USER () 的使用。

(d) REVOKE SELECT, UPDATE ON 职工 FROM 刘星;

(e) REVOKE ALTER TABLE ON 职工, 部门 FROM 张新;

(f) REVOKE ALL PRIVILEGES ON 职工, 部门 FROM 周平;

(g) REVOKE SELECT ON 部门工资 FROM 杨兰;

DROP VIEW 部门工资;

8. 为什么强制存取控制提供了更高级别的数据库安全性?

答: 强制存取控制 (MAC) 是对数据本身进行密级标记, 无论数据如何复制, 标记与数据是一个不可分的整体, 只有符合密级标记要求的用户才可以操纵数据, 从而提供了更高级别的安全性。

9. 理解并解释 MAC 机制中主体、客体、敏感度标记的含义。

答: 主体是系统中的活动实体, 既包括 DBMS 所管理的实际用户, 也包括代表用户的各进程。

客体是系统中的被动实体, 是受主体操纵的, 包括文件、基表、索引、视图等。

对于主体和客体, DBMS 为它们每个实例 (值) 指派一个敏感度标记 (Label)。敏感度标记被分成若干级别, 例如绝密 (Top Secret)、机密 (Secret)、可信 (Confidential)、公开 (Public) 等。主体的敏感度标记称为许可证级别 (Clearance Level), 客体的敏感度标记称为密级 (Classification Level)。

10. 什么是数据库的审计功能, 为什么要提供审计功能?

答: 审计功能指 DBMS 的审计模块在用户对数据库执行操作的同时把所有操作自动记录到系统的审计日志中。

因为任何系统的安全保护措施都不是完美无缺的, 蓄意盗窃破坏数据的人总可能存在。利用数据库的审计功能, DBA 可以根据审计跟踪的信息, 重现导致数据库现有状况的一系列事件, 找出非法存取数据的人、时间和内容等。

11. 统计数据库中存在何种特殊的安全性问题?

答: 统计数据库允许用户查询聚集类型的信息, 如合计、平均值、最大值、最小值等, 不允许查询单个记录信息。但是, 人们可以从合法的查询中推导出不合法的信息, 即可能存在隐蔽的信息通道, 这是统计数据库要研究和解决的特殊的安全性问题。

*12. 试述你了解的某一个实际的 DBMS 产品的安全性措施。

答: 不同的 DBMS 产品以及同一产品的不同版本的安全措施各不相同, 仁者见仁, 智者见智, 请读者自己了解。《概论》上 9.4 简单介绍了有关 ORACLE 数据库的安全性措施。

数据库完整性

一、选择题

1. 完整性检查和控制的防范对象 (), 防止它们进入数据库。安全性控制的防范对象是 (), 防止他们对数据库数据的存取。

A. 不合语义的数据 B. 非法用户 C. 不正确的数据 D. 非法操作

2. 下述哪个是 SQL 语言中的数据控制命令 ()。

A. GRANT B. COMMIT C. UPDATE D. SELECT

3. 下述 SQL 语言中的权限, 哪一个允许用户定义新关系时, 引用其他关系的主码作为外码 ()。

A. INSERT B. DELETE C. REFERENCES D. SELECT

参考答案: (1) A C B D (2) A (3) C

二、简答题

1. 什么是数据库的完整性?

答: 数据库的完整性是指数据的正确性和相容性。

2. 数据库的完整性概念与数据库的安全性概念有什么区别和联系?

答: 数据的完整性和安全性是两个不同的概念, 但是有一定的联系。

前者是为了防止数据库中存在不符合语义的数据, 防止错误信息的输入和输出, 即所谓垃圾进垃圾出 (Garbage In Garbage Out) 所造成的无效操作和错误结果。

后者是保护数据库防止恶意的破坏和非法的存取。

也就是说, 安全性措施的防范对象是非法用户和非法操作, 完整性措施的防范对象是不合语义的数据。

3.什么是数据库的完整性约束条件？可分为哪几类？

答：完整性约束条件是指数据库中的数据应该满足的语义约束条件。

一般可以分为六类：

静态列级约束、静态元组约束、静态关系约束、动态列级约束、动态元组约束、动态关系约束。

静态列级约束是对一个列的取值域的说明，包括以下几方面：

1. 对数据类型的约束，包括数据的类型、长度、单位、精度等
2. 对数据格式的约束
3. 对取值范围或取值集合的约束。
4. 对空值的约束
5. 其他约束

静态元组约束就是规定组成一个元组的各个列之间的约束关系，静态元组约束只局限在单个元组上。

静态关系约束是在一个关系的各个元组间或者若干关系间常常存在各种联系或约束。常见的静态关系约束有：

1. 实体完整性约束。
2. 参照完整性约束。
3. 函数依赖约束。

动态列级约束是修改列定义或列值时应满足的约束条件，包括下面两方面：

1. 修改列定义时的约束
2. 修改列值时的约束

动态元组约束是指修改某个元组的值时需要参照其旧值，并且新旧值之间需要满足某种约束条件。

动态关系约束是加在关系变化前后状态上的限制条件，例如事务一致性、原子性等约束条件。

详细内容可以参见《概论》10.1 中的介绍。

4.DBMS 的完整性控制机制应具有哪些功能？

答：DBMS 的完整性控制机制应具有三个方面的功能：

- 1) . 定义功能，即提供定义完整性约束条件的机制。
2. 检查功能，即检查用户发出的操作请求是否违背了完整性约束条件。
- 3) . 违约反应：如果发现用户的操作请求使数据违背了完整性约束条件，则采取一定的动作来保证数据的完整性。

5.RDBMS 在实现参照完整性时需要考虑哪些方面？

答：RDBMS 在实现参照完整性时需要考虑以下几个方面：

- 1) 外码是否可以接受空值
- 2) 删除被参照关系的元组时的考虑，这时系统可能采取的作法有三种：
 - (a) 级联删除 (CASCADES)；
 - (b) 受限删除 (RESTRICTED)；
 - (c) 置空值删除 (NULLIFIES)
- 3) 在参照关系中插入元组时的问题，这时系统可能采取的作法有：
 - (a) 受限插入
 - (b) 递归插入
- 4) 修改关系中主码的问题

一般是不能用 UPDATE 语句修改关系主码的。如果需要修改主码值，只能先删除该元组，然后再把具有新主码值的元组插入到关系中。

如果允许修改主码，首先要保证主码的唯一性和非空，否则拒绝修改。然后要区分是参照关系还是被参照关系。详细讨论可以参见《概论》10.2。

6. 假设有下面两个关系模式：

职工（职工号，姓名，年龄，职务，工资，部门号），其中职工号为主码；

部门（部门号，名称，经理名，电话），其中部门号为主码；

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

定义每个模式的主码；定义参照完整性；定义职工年龄不得超过 60 岁。


```

答：CREATE TABLE DEPT
  (Deptno NUMBER(2),
  Deptname VARCHAR(10),
  Manager VARCHAR(10),
  PhoneNumber Char(12)
  CONSTRAINT PK_SC PRIMARY KEY (Deptno));
CREATE TABLE EMP
  (Empno NUMBER(4),
  Ename VARCHAR(10),
  Age NUMBER(2),
  CONSTRAINT C1 CHECK (Age <=60),
  Job VARCHAR(9),
  Sal NUMBER(7,2),
  Deptno NUMBER(2),
  CONSTRAINT FK_DEPTNO
  FOREIGN KEY (Deptno)
  REFERENCES DEPT(Deptno));

```

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

答：对于违反实体完整性和用户定义的完整性的操作一般都采用拒绝执行的方式进行处理。而对于违反参照完整性的操作，并不都是简单地拒绝执行，有时要根据应用语义执行一些附加的操作，以保证数据库的正确性。具体的处理可以参见上面第 5 题或《概论》10.2 中相应部分。

***8. 试述你了解的某一个实际的 DBMS 产品的完整性控制策略。**

答：不同的 DBMS 产品以及同一产品的不同版本的完整性控制策略各不相同，读者要去了解某一个 DBMS 产品的完整性控制策略。