第 1 章 绪论

1．试述数据、数据库、数据库系统、数据库管理系统的概念。

答：

(l）数据（ Data)：描述事物的符号记录称为数据。数据的种类有数字、文字、图形、图像、声音、正文等。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。现代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。

(2）数据库（ DataBase，简称 DB)：数据库是长期储存在计算机内的、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。

(3）数据库系统（ DataBas 。 Sytem，简称 DBS)：数据库系统是指在计算机系统中引入数据库后的系统构成，一般由数据库、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一机系统，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。

(4）数据库管理系统（ DataBase Management sytem，简称 DBMs)：数据库管理系统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 的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 是一个大型的复杂的软件系统，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 DBZ 关系数据库管理系统和 IMS 层次数据库管理系统、美国 Oracle 公司的 orade 关系数据库管理系统、 s 油 ase 公司的 s 油 ase 关系数据库管理系统、美国微软公司的 SQL Serve，关系数据库管理系统等。

2．使用数据库系统有什么好处？

答：

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，而不必为数据管理的许许多多复杂的细节操心。还有，当应用逻辑改变，数据的逻辑结构也需要改变时，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，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，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。

答：

文件系统与数据库系统的区别是：文件系统面向某一应用程序，共享性差，冗余度大，数据独立性差，记录内有结构，整体无结构，由应用程序自己控制。数据库系统面向现实世界，共享性高，冗余度小，具有较高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的安全性、完整性、并发控制和恢复能力。

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的文件系统来实现的。

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。

答 ：

(l）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。

（2）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。例如，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），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。希望读者能举出自己了解的应用例子。

5．试述数据库系统的特点。

答：

数据库系统的主要特点有：

（l）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。解析注意这里的“整体’夕两个字。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。

（2）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、多个应用以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，只需做很少的改动。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。

（3）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像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。

(4）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？

答：

（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。

答：

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。这些概念精确描述了系统的静态特性、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：是一组完整性规则的集合。完整性规则是给定的数据模型中数据及其联系所具有的制约和依存规则，用以限定符合数据模型的数据库状态以及状态的变化，以保证数据的正确、有效、相容。解析数据模型是数据库系统中最重要的概念之一。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者说两个层次：一类是概念模型，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；另一类是数据模型，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。

答：

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。

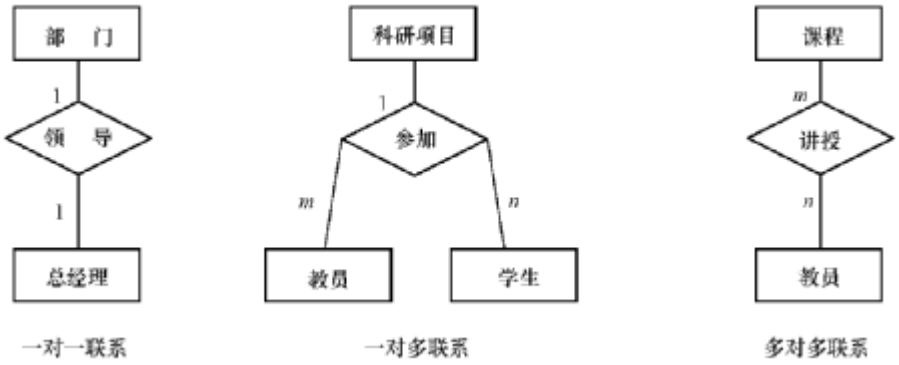
9．定义并解释概念模型中以下术语：实体，实体型，实体集，属性，码，实体联系图（E — R 图）

答：

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：惟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的类型（ 1 : 1 , 1 : n 或 m : n ）。

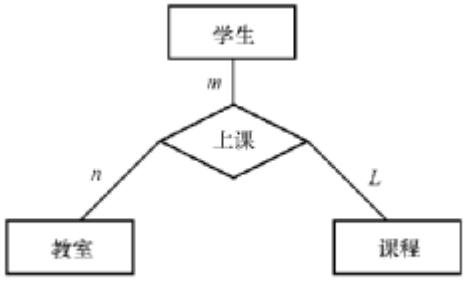
10．试给出 3 个实际部门的 E — R 图，要求实体型之间具有一对一、一对多、多对多各种不同的联系。

答：

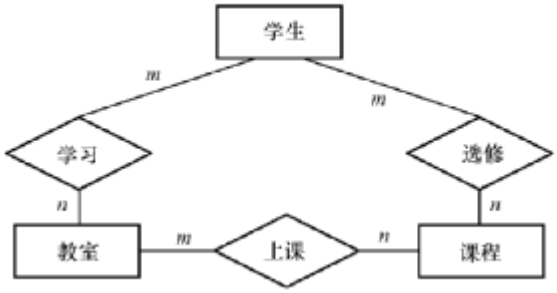


11．试给出一个实际部门的 E — R 图，要求有三个实体型，而且 3 个实体型之间有多对多联系。 3 个实体型之间的多对多联系和三个实体型两两之间的三个多对多联系等价吗？为什么？

答：

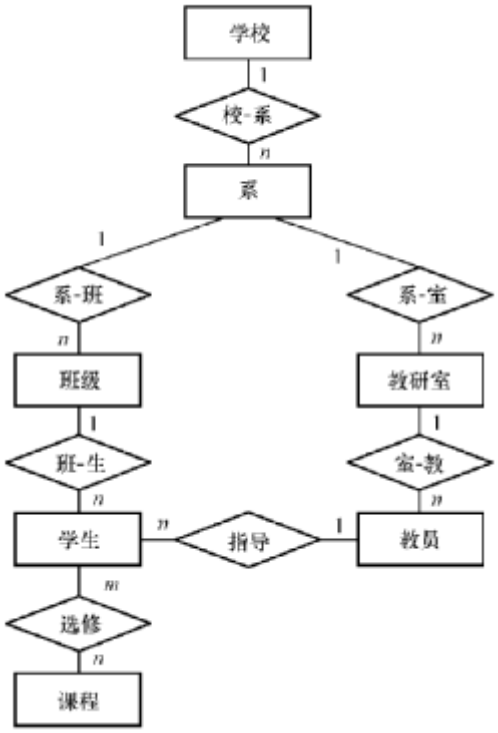


3 个实体型之间的多对多联系和 3 个实体型两两之间的 3 个多对多联系是不等价，因为它们拥有不同的语义。 3 个实体型两两之间的三个多对多联系如下图所示。



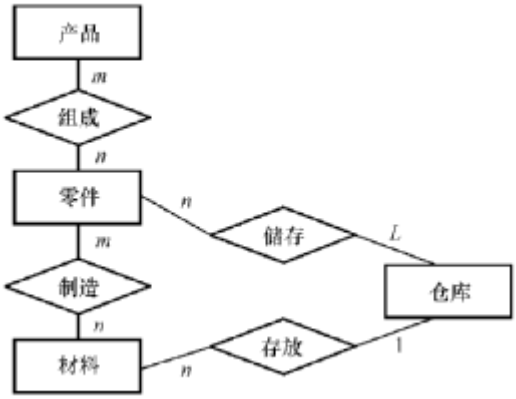
12．学校中有若干系，每个系有若干班级和教研室，每个教研室有若干教员，其中有的教授和副教授每人各带若干研究生；每个班有若干学生，每个学生选修若干课程，每门课可由若干学生选修。请用 E 一 R 图画出此学校的概念模型。

答：



13．某工厂生产若干产品，每种产品由不同的零件组成，有的零件可用在不同的产品上。这些零件由不同的原材料制成，不同零件所用的材料可以相同。这些零件按所属的不同产品分别放在仓库中，原材料按照类别放在若干仓库中。请用 E 一 R 图画出此工厂产品、零件、材料、仓库的概念模型。

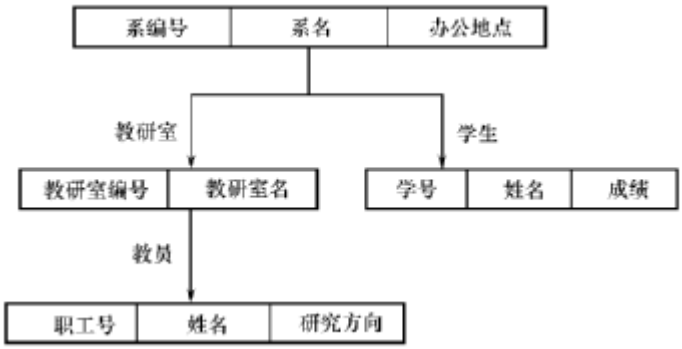
答：



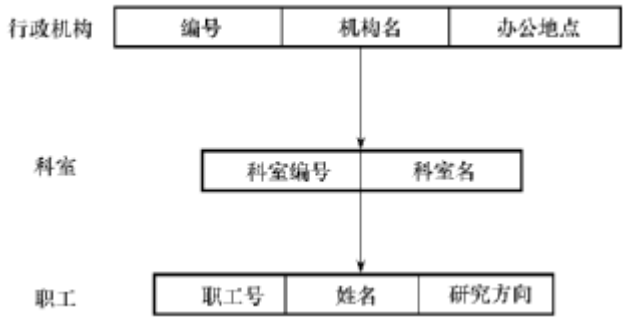
14．试述层次模型的概念，举出三个层次模型的实例。

答：

（1）教员学生层次数据库模型



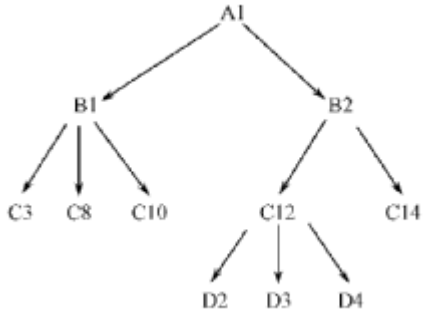
（2）行政机构层次数据库模型



（3）行政区域层次数据库模型

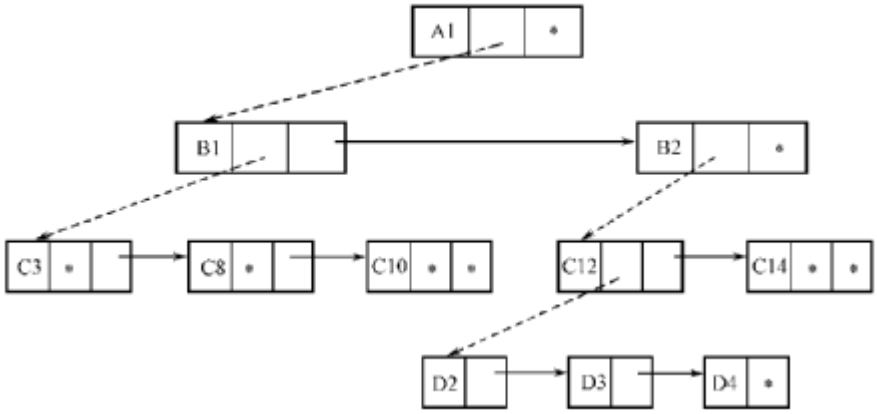
15．今有一个层次数据库实例，试用子女一兄弟链接法和层次序列链接法画出它的存储结

构示意图。

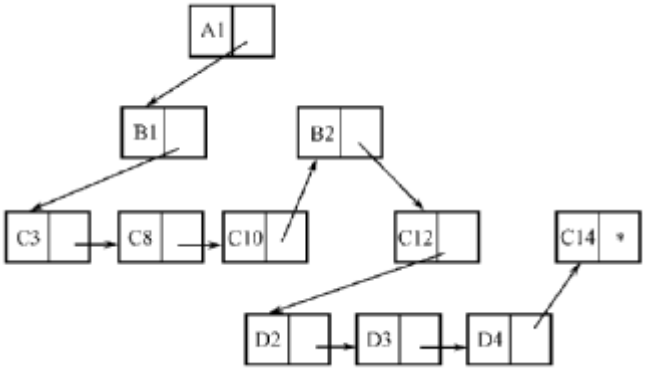


答：

子女兄弟链接法：



层次序列链接法：



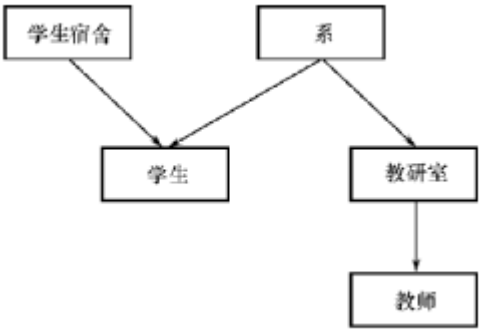
16．试述网状模型的概念，举出三个网状模型的实例。

答：

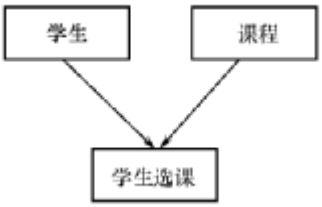
满足下面两个条件的基本层次联系集合为网状模型。

（1）允许一个以上的结点无双亲；（2）一个结点可以有多于一个的双亲。

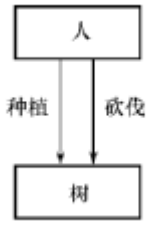
实例 1：



实例 2：



实例 3：



17．试述网状、层次数据库的优缺点。

答：

层次模型的优点主要有：（1）模型简单，对具有一对多层次关系的部门描述非常自然、直观，容易理解，这是层次数据库的突出优点；（2）用层次模型的应用系统性能好，特别是对于那些实体间联系是固定的且预先定义好的应用，采用层次模型来实现，其性能优于关系模型；（3）层次数据模型提供了良好的完整性支持。

层次模型的缺点主要有：（1）现实世界中很多联系是非层次性的，如多对多联系、一个结点具有多个双亲等，层次模型不能自然地表示这类联系，只能通过引入冗余数据或引入虚拟结点来解决；（2）对插入和删除操作的限制比较多；（3）查询子女结点必须通过双亲结点。

网状数据模型的优点主要有：（1）能够更为直接地描述现实世界，如一个结点可以有多个双亲；（2）具有良好的性能，存取效率较高。

网状数据模型的缺点主要有：（1）结构比较复杂，而且随着应用环境的扩大，数据库的结构就变得越来越复杂，不利于最终用户掌握；（2）其 DDL、DML 语言复杂，用户不容易使用。由于记录之间联系是通过存取路径实现的，应用程序在访问数据时必须选择适当的存取路径。因此，用户必须了解系统结构的细节，加重了编写应用程序的负担。

18．试述关系模型的概念，定义并解释以下术语：（1）关系（2）属性（3）域（4）元组（5）主码（6）分量（7）关系模式

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。在用户观点下，关系模型中数据的逻辑结构是一张二维表，它由行和列组成。（1）关系：一个关系对应通常说的一张表；（2）属性：表中的一列即为一个属性；（3）域：属性的取值范围；（4）元组：表中的一行即为一个元组；（5）主码：表中的某个属性组，它可以惟一确定一个元组；（6）分量：元组中的一个属性值；（7）关系模式：对关系的描述，一般表示为关系名（属性 1，属性 2，…，属性 n）

19．试述关系数据库的特点。
答：
关系数据模型具有下列优点：（1）关系模型与非关系模型不同，它是建立在严格的数学概念的基础上的。（2）关系模型的概念单一，无论实体还是实体之间的联系都用关系表示，操作的对象和操作的结果都是关系，所以其数据结构简单、清晰，用户易懂易用。（3）关系模型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。当然，关系数据模型也有缺点，其中最主要的缺点是，由于存取路径对用户透明，查询效率往往不如非关系数据模型。因此为了提高性能，必须对用户的查询请求进行优化，增加了开发数据库管理系统的难度。

20．试述数据库系统三级模式结构，这种结构的优点是什么？
答：
数据库系统的三级模式结构由外模式、模式和内模式组成。（参见书上图 1.29）外模式，亦称子模式或用户模式，是数据库用户（包括应用程序员和最终用户）能够看见和使用的局部数据的逻辑结构和特征的描述，是数据库用户的数据视图，是与某一应用有关的数据的逻辑表示。模式，亦称逻辑模式，是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图。模式描述的是数据的全局逻辑结构。外模式涉及的是数据的局部逻辑结构，通常是模式的子集。内模式，亦称存储模式，是数据在数据库系统内部的表示，即对数据的物理结构和存储方式的描述。数据库系统的三级模式是对数据的三个抽象级别，它把数据的具体组织留给 DBMs 管理，使用户能逻辑抽象地处理数据，而不必关心数据在计算机中的表示和存储。为了能够在内部实现这三个抽象层次的联系和转换，数据库系统在这三级模式之间提供了两层映像：外模式／模式映像和模式／内模式映像。正是这两层映像保证了数据库系统中的数据能够具有较高的逻辑独立性和物理独立性。

21．定义并解释以下术语：模式、外模式、内模式、DDL、DML 模式、外模式、内模式，亦称逻辑模式，是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图。模式描述的是数据的全局逻辑结构。外模式涉及的是数据的局部逻辑结构，通常是模式的子集。内模式，亦称存储模式，是数据在数据库系统内部的表示，即对数据的物理结构和存储方式的描述。DDL：数据定义语言，用来定义数据库模式、外模式、内模式的语言。DML：数据操纵语言，用来对数据库中的数据进行查询、插入、删除和修改的语句。

22．什么叫数据与程序的物理独立性？什么叫数据与程序的逻辑独立性？为什么数据库系统具有数据与程序的独立性？
答：
数据与程序的逻辑独立性：当模式改变时（例如增加新的关系、新的属性、改变属性的数据类型等），由数据库管理员对各个外模式／模式的映像做相应改变，可以使外模式保持不

变。应用程序是依据数据的外模式编写的，从而应用程序不必修改，保证了数据与程序的逻辑独立性，简称数据的逻辑独立性。数据与程序的物理独立性：当数据库的存储结构改变了，由数据库管理员对模式／内模式映像做相应改变，可以使模式保持不变，从而应用程序也不必改变，保证了数据与程序的物理独立性，简称数据的物理独立性。数据库管理系统在三级模式之间提供的两层映像保证了数据库系统中的数据能够具有较高的逻辑独立性和物理独立性。

23．试述数据库系统的组成。

答：

数据库系统一般由数据库、数据库管理系统（及其开发工具）、应用系统、数据库管理员和用户构成。

24 . DBA 的职责是什么？

答：

负责全面地管理和控制数据库系统。具体职责包括： ① 决定数据库的信息内容和结构；② 决定数据库的存储结构和存取策略； ③ 定义数据的安全性要求和完整性约束条件；④ 监督和控制数据库的使用和运行； ⑤ 改进和重组数据库系统。**25．**系统分析员、数据库设计人员、应用程序员的职责是什么？答系统分析员负责应用系统的需求分析和规范说明，系统分析员要和用户及 **DBA** 相结合，确定系统的硬件、软件配置，并参与数据库系统的概要设计。数据库设计人员负责数据库中数据的确定、数据库各级模式的设计。数据库设计人员必须参加用户需求调查和系统分析，然后进行数据库设计。在很多情况下，数据库设计人员就由数据库管理员担任。应用程序员负责设计和编写应用系统的程序模块，并进行调试和安装。

第 2 章 关系数据库

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。

关系演算语言：元组关系演算语言和域关系演算语言。

SQL：具有关系代数和关系演算双重特点的语言。

这些关系数据语言的共同特点是，语言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。

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

若属性(或属性组)F 是基本关系 R 的外码，它与基本关系 S 的主码 Ks 相对应(基本关系 R 和 S 不一定是不同的关系)，则对于 R 中每个元组在 F 上的值必须为：或者取空值(F 的每个属性值均为空值)；或者等于 S 中某个元组的主码值。即属性 F 本身不是主属性，则可以取空值，否则不能取空值。

5．设有一个 SPJ 数据库，包括 S，P，J，SPJ 四个关系模式：

1）求供应工程 J1 零件的供应商号码 SNO：

πSno(σSno='J1'（SPJ）)

2）求供应工程 J1 零件 P1 的供应商号码 SNO：

πSno(σSno='J1'∧Pno='P1'(SPJ))

3）求供应工程 J1 零件为红色的供应商号码 SNO：

πSno(σPno='P1'（σCOLOR='红'（P）∞SPJ）)

4）求没有使用天津供应商生产的红色零件的工程号 JNO：

πJno(SPJ)- πJNO（σcity='天津'∧Color='红'（S∞SPJ∞P）

5）求至少用了供应商 S1 所供应的全部零件的工程号 JNO：

πJno，Pno(SPJ)÷ πPno（σSno='S1'（SPJ））

6．试述等值连接与自然连接的区别和联系。

答：连接运算符是"="的连接运算称为等值连接。它是从关系 R 与 S 的广义笛卡尔积中选取 A，B 属性值相等的那些元组

自然连接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，并且在结果中把重复的属性列去掉。

7．关系代数的基本运算有哪些 ？如何用这些基本运算来表示其他运算?

答：并、差、笛卡尔积、投影和选择 5 种运算为基本的运算。其他 3 种运算，即交、连接和除，均可以用这 5 种基本运算来表达。

第 3 章 关系数据库标准语言 SQL

1．试述 sQL 语言的特点。

答：

(l）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"，而无需指明"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；作为嵌入式语言，它能够嵌入到

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。
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；使用 CREATE INDEX 语句建立索引， DROP INDEX 语句删除索引；使用 CREATE VIEW 语句建立视图， DROP VIEW 语句删除视图。

3．用 sQL 语句建立第二章习题 5 中的 4 个表。
答：
对于 S 表： S（ SNO ， SNAME ， STATUS ， CITY ）；
建 S 表：
CREATE TABLE S (Sno C(2) UNIQUE，Sname C(6) ，Status C(2)，City C(4));

对于 P 表： P (PNO , PNAME , COLOR , WEIGHT);
建 P 表 ：
CREATE TABLE P(Pno C(2) UNIQUE，Pname C(6)，COLOR C(2)， WEIGHT INT);

对于 J 表： J (JNO , JNAME , CITY） ；
建 J 表：
CREATE TABLE J(Jno C(2) UNlQUE，JNAME C(8)， CITY C(4))

对于 sPJ 表： sPJ (sNo , PNo , JNo , QTY） ；
建 SPJ 表：SPJ(SNO,PNO,JNO,QTY)
CREATE TABLE SPJ(Sno C(2)，Pno C(2)，JNO C(2)， QTY INT))

4.针对上题中建立的 4 个表试用 sQL 语言完成第二章习题 5 中的查询。
（l）求供应工程 Jl 零件的供应商号码 SNO；
SELECT DIST SNO FROM SPJ WHERE JNO=’J1’
（2）求供应工程 Jl 零件 Pl 的供应商号码 SNO；
SELECT DIST SNO FROM SPJ WHERE JNO='J1' AND PNO='P1'
(3）求供应工程 Jl 零件为红色的供应商号码 SNO；
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号 JNO；
SELECT DIST JNO FROM SPJ WHERE JNO NOT IN (SELE JNO FROM SPJ,P,S WHERE S.CITY='天津' AND COLOR='红' AND S.SNO=SPJ.SNO AND P.PNO=SPJ.PNO)。
(5）求至少用了供应商 Sl 所供应的全部零件的工程号 JNO；
由于 VFP 不允许子查询嵌套太深，将查询分为两步
A、查询 S1 供应商供应的零件号
SELECT DIST PNO FROM SPJ WHERE SNO=’S1’结果是（P1，P2）
B、查询哪一个工程既使用 P1 零件又使用 P2 零件。

SELECT JNO FROM SPJ WHERE PNO=’P1’
AND JNO IN (SELECT JNO FROM SPJ WHERE PNO=’P2’)

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。

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 DIST 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 PNAME,QTY FROM SPJ,P

WHERE P.PNO=SPJ.PNO AND SPJ.JNO='J2'

(5)找出上海厂商供应的所有零件号码。

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

(6)出使用上海产的零件的工程名称。

SELECT JNAME FROM SPJ,S,J

WHERE S.SNO=SPJ.SNO AND S.CITY='上海' AND J.JNO=SPJ.JNO

(7)找出没有使用天津产的零件的工程号码。

注意: SELECT DISP JNO FROM SPJ WHERE JNO NOT IN (SELECT DIST JNO FROM SPJ,S WHERE S.SNO=SPJ.SNO AND S.CITY='天津') 适用于JNO是唯一或不唯一的情况.

注意: SELECT DIST JNO FROM SPJ,S WHERE S.SNO=SPJ.SNO AND S.CITY<>'天津'适用于JNO是唯一的情况

(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。

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

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

A、DELETE FROM S WHERE SNO=’S2’

B、DELETE FROM SPJ WHERE SNO=‘S2’

(11)请将(S2，J6，P4，200)插入供应情况关系。

INSERT INTO SPJ VALUES（‘S2’，‘J6’，‘P4’，200）

6．什么是基本表？什么是视图？

答

两者的区别和联系是什么？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惟一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.

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。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。

10．试述某个你熟悉的实际系统中对视图更新的规定。
答
VFP

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)。

```
CREATE VIEW VSP AS SELECT SNO,PNO,QTY FROM SPJ,J
  WHERE SPJ.JNO=J.JNO AND J.JNAME='三建'
```

针对该视图 VSP 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。

```
SELECT  DIST  PNO,QTY  FROM  VSP
```

(2)找出供应商 S1 的供应情况。

```
SELECT  DIST * FROM VSP WHERE SNO='S1'
```

第 4 章 数据库安全性

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．试述可信计算机系统评测标准的情况，试述 TDI / TCSEC 标准的基本内容。
答：各个国家在计算机安全技术方面都建立了一套可信标准。目前各国引用或制定的一系列安全标准中，最重要的是美国国防部（DoD）正式颁布的《 DoD 可信计算机系统评估标准》

（伽 sted Co 哪 uter system Evaluation criteria ， 简称 TcsEc ， 又称桔皮书）。（TDI / TCSEC 标准是将 TcsEc 扩展到数据库管理系统，即《 可信计算机系统评估标准关于可信数据库系统的解释》（Tmsted Database Interpretation 简称 TDI , 又称紫皮书）。在 TDI 中定义了数据库管理系统的设计与实现中需满足和用以进行安全性级别评估的标准。TDI 与 TcsEc 一样，从安全策略、责任、保证和文档四个方面来描述安全性级别划分的指标。每个方面又细分为若干项。

4 ．试述 TcsEC (TDI ）将系统安全级别划分为 4 组 7 个等级的基本内容。

答：根据计算机系统对安全性各项指标的支持情况，TCSEC (TDI ）将系统划分为四组（division) 7 个等级，依次是 D 、C (CI , CZ ）、B (BI , BZ , B3 ）、A (AI) ，按系统可靠或可信程度逐渐增高。

安全级别	定　义
A1	验证设计（Verified Design）
B3	安全域（Security Domains）
B2	结构化保护（Structural Protection）
B1	标记安全保护（Labeled Security Protection）
C2	受控的存取保护（Controlled Access Protection）
C1	自主安全保护（Discretionary Security Protection）
D	最小保护（Minimal Protection）

这些安全级别之间具有一种偏序向下兼容的关系，即较高安全性级别提供的安全保护包含较低级别的所有保护要求，同时提供更多或更完善的保护能力。各个等级的基本内容为：

D 级 D 级是最低级别。一切不符合更高标准的系统，统统归于 D 组。

Cl 级只提供了非常初级的自主安全保护。能够实现对用户和数据的分离，进行自主存取控制（DAC）， 保护或限制用户权限的传播。

C2 级实际是安全产品的最低档次，提供受控的存取保护，即将 Cl 级的 DAC 进一步细化，以个人身份注册负责，并实施审计和资源隔离。

Bl 级标记安全保护。对系统的数据加以标记，并对标记的主体和客体实施强制存取控制（MAC）以及审计等安全机制。

B2 级结构化保护。建立形式化的安全策略模型并对系统内的所有主体和客体实施 DAC 和 MACo

B3 级安全域。该级的 TCB 必须满足访问监控器的要求，审计跟踪能力更强，并提供系统恢复过程。

Al 级验证设计，即提供 B3 级保护的同时给出系统的形式化设计说明和验证以确信各安全保护真正实现。

5 ．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：

(l ）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 CZ 级中的自主存取控制(DAC) , Bl 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。

6．什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。

7 .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。

8．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;
```

(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的权力。

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。

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;

9．把习题 8 中（1）---（7）的每一种情况，撤销各用户所授予的权力

(1) REVOKE SELECT ON 职工，部门 FROM 王明；

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

(3) REOVKE SELECT ON 职工 WHEN USER ( ) =NAME FROM ALI ;

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

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

(6) REVOKE ALL PRIVILIGES ON 职工，部门 FROM 周平；

(7) REVOKE SELECT ON 部门工资 FROM 杨兰；

DROP VIEW 部门工资；

10. 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。

11．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 玫vel) ，客体的敏感度标记称为密级（Classification Level ）。

13．什么是数据库的审计功能，为什么要提供审计功能？
答：审计功能是指 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。

14．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。

第 5 章 数据库完整性

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（ Garba : e In Garba : e 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，包括以下几个方面：（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之间或者若干关系之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。

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）册 l 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：1）级联删除（CASCADES）；2）受限删除（ RESTRICTED）；3）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：1）受限插入；2）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，首先要保证主码的惟一性和非空，否则拒绝修改。然后要区分是参照关系还是被参照关系。

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 RIMARY KEY(Deptno));
CREATE TABLE EMP
    (Empno NUMBER(4),
     Ename VARCHAR(10),
     Age NUMBER(2),
     CONSTRAINT C1 CHECK ( Aage<=60),
      Job VARCHAR(9),
```

```
Sal NUMBER(7,2),
Deptno NUMBER(2),
CONSTRAINT FK_DEPTNO
    FOREIGN KEY(Deptno)
    REFFERENCES 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。

第 6 章 关系数据库理论

1．理解并给出下列术语的定义：
函数依赖、部分函数依赖、完全函数依赖、传递依赖、候选码、主码、外码、全码（All 一 key ）、1 NF 、ZNF 、3NF 、BcNF 、多值依赖、4NF 。

定义 1：设 R(U)是属性集 U 上的关系模式。X，Y 是属性集 U 的子集。若对于 R(U)的任意一个可能的关系 r，r 中不可能存在两个元组在 X 上的属性值相等，而在 Y 上的属性值不等，则称 X 函数确定 Y 或 Y 函数依赖于 X，记作 X→Y。（即只要 X 上的属性值相等，Y 上的值一定相等。）

术语和记号：

X→Y，但 Y 不是 X 的子集，则称 X→Y 是非平凡的函数依赖。若不特别声明，总是讨论非平凡的函数依赖。

X→Y，但 Y 是 X 的子集，则称 X→Y 是平凡的函数依赖。

若 X→Y，则 X 叫做决定因素(Determinant)。

若 X→Y，Y→X，则记作 X←→Y。

若 Y 不函数依赖于 X，则记作 X → Y。

定义 2：在 R(U)中，如果 X→Y，并且对于 X 的任何一个真子集 X’，都有 X’ → Y，则称 Y 对 X 完全函数依赖

若 X→Y，但 Y 不完全函数依赖于 X，则称 Y 对 X 部分函数依赖

定义 3：若关系模式 R 的每一个分量是不可再分的数据项，则关系模式 R 属于第一范式(1NF)。

定义 4：若关系模式 R∈1NF，且每一个非主属性完全函数依赖于码，则关系模式 R∈2NF 。（即 1NF 消除了非主属性对码的部分函数依赖则成为 2NF）。

定义 5:关系模式 R<U，F> 中若不存在这样的码 X、属性组 Y 及非主属性 Z(Z 不是 Y 的子集) 使得 X→Y，Y → X，Y → Z 成立，则称 R<U，F>∈3NF。

定义 6:关系模式 R<U，F>∈1NF 。若 X→Y 且 Y 不是 X 的子集时,X 必含有码，则 R<U，F>∈BCNF。

定义 7:关系模式 R<U，F>∈1NF，如果对于 R 的每个非平凡多值依赖 X→→Y(Y 不是 X 的子集，Z=U-X-Y 不为空)，X 都含有码，则称 R<U，F>∈4NF。

2．建立一个关于系、学生、班级、学会等诸信息的关系数据库。

学生：学号、姓名、出生年月、系名、班号、宿舍区。

班级：班号、专业名、系名、人数、入校年份。

系：系名、系号、系办公地点、人数。

学会：学会名、成立年份、办公地点、人数。

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

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。指出各关系模式的候选码、外部码，有没有全码存在?

解：(1)关系模式如下：

学生：S(Sno，Sname，Sbirth，Dept，Class，Rno)

班级：C(Class，Pname，Dept，Cnum，Cyear)

系：D(Dept，Dno，Office，Dnum)

学会：M(Mname，Myear，Maddr，Mnum)

(2)每个关系模式的最小函数依赖集如下：

A、学生 S (Sno，Sname，Sbirth，Dept，Class，Rno) 的最小函数依赖集如下:Sno→Sname，Sno→Sbirth，Sno→Class，Class→Dept，DEPT→Rno

传递依赖如下：

由于 Sno→Dept，而 Dept→Sno ，Dept→Rno（宿舍区）

所以 Sno 与 Rno 之间存在着传递函数依赖。

由于 Class→Dept，Dept → Class，Dept→Rno

所以 Class 与 Rno 之间存在着传递函数依赖。

由于 Sno→Class，Class→Sno，Class→Dept

所以 Sno 与 Dept 之间存在着传递函数依赖。

B、班级 C(Class，Pname，Dept，Cnum，Cyear)的最小函数依赖集如下:

Class→Pname，Class→Cnum，Class→Cyear，Pname→Dept.

由于 Class→Pname，Pname→Class，Pname→Dept

所以 C1ass 与 Dept 之间存在着传递函数依赖。

C、系 D(Dept，Dno，Office，Dnum)的最小函数依赖集如下：

Dept→Dno，Dno→Dept，Dno→Office，Dno→Dnum

根据上述函数依赖可知，Dept 与 Office，Dept 与 Dnum 之间不存在传递依赖。

D、学会 M(Mname，Myear，Maddr，Mnum)的最小函数依赖集如下：

Mname→Myear，Mname→Maddr，Mname→Mnum

该模式不存在传递依赖。

(3)各关系模式的候选码、外部码，全码如下：

A、学生 S 候选码：Sno；外部码：Dept、Class；无全码

B、班级 C 候选码：Class；外部码：Dept；无全码

C、系 D 候选码：Dept 或 Dno；无外部码；无全码

D、学会 M 候选码：Mname；无外部码；无全码

3．试由 amstrong 公理系统推导出下面三条推理规则：

(1) 合并规则：若 X 一 Z，X 一 Y，则有 X 一 YZ

(2) 伪传递规则：由 x 一 Y，明吟 z 有翔一 z

(3) 分解规则：x 一 Y，zcy，有 x 一 z

证明

(l）已知 X 一 Z，由增广律知哟，YZ，又因为 X 一 Y，可得狱一 X 卜）YZ，最后根据传递律得 x 一 YZ。

(2）已知 X 一 Y，据增广律得翔一 Wy，因为阴几）Z，所以 X 林协明，Z，通过传递律可知翔一 Z。

(3）已知 zcy，根据自反律知、吟 z，又因为 x 一 Y，所以由传递律可得 x 一 Z。

5．试举出 3 个多值依赖的实例。

答:

（1）关系模式 MSC (M, S, C）中，M 表示专业，S 表示学生，C 表示该专业的必修课。假设每个专业有多个学生，有一组必修课。设同专业内所有学生选修的必修课相同，实例关系如下。按照语义对于 M 的每一个值 M i, s 有一个完整的集合与之对应而不问 C 取何值，

所以 M 一一 S 。由于 C 与 S 的完全对称性，必然有 M 一一 C 成立。

（2）关系模式 ISA (I , S , A ）中，I 表示学生兴趣小组，S 表示学生，A 表示某兴趣小组的活动项目。假设每个兴趣小组有多个学生，有若干活动项目。每个学生必须参加所在兴趣小组的所有活动项目，每个活动项目要求该兴趣小组的所有学生参加。按照语义有 I 一一 S，I 一一 A 成立。

（3）关系模式 RDP (R , D , P ）中，R 表示医院的病房，D 表示责任医务人员，P 表示病人。假设每个病房住有多个病人，有多个责任医务人员负责医治和护理该病房的所有病人。

12．下面的结论哪些是正确的? 哪些是错误的? 对于错误的请给一个反例说明之。

（1）任何一个二目关系是属于 3NF。

答：正确。因为关系模式中只有两个属性，所以无传递。

（2）任何一个二目关系是属于 BCNF.

答:正确。按 BCNF 的定义，若 X→Y,且 Y 不是 X 的子集时，每个决定因素都包含码，对于二目关系决定因素必然包含码。详细证明如下：（任何二元关系模式必定是 BCNF）。

证明：设 R 为一个二目关系 R(A1，A2)，则属性 A1 和 A2 之间可能存在以下几种依赖关系：

A、A1→A2，但 A2→A1，则关系 R 的码为 A1，决定因素都包含码，所以，R 是 BCNF。

B、A1→A2，A2→A1，则关系 R 的码为 A2，所以决定因素都包含码，R 是 BCNF。

包含码。R 是 BCNF。C、R 的码为(A1，A2)（即 A1 →A2，A2 →A1），决定因素都

（3）任何一个二目关系是属于 4NF.

答:正确。因为只有两个属性，所以无非平凡的多值依赖。

第 7 章 数据库设计

1．试述数据库设计过程。

答：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、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各个阶段上的设计描述。

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：（1）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；（2）在逻辑设计阶段将 E — R 图转换成具体的数据库产品支持的数据模型，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（Vi 娜），形成数据的外模式；（3）在物理设计阶段，根据 DBMS 特点和处理的需要，进行物理存储安排，建立索引，形成数据库内模式。

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’夕和“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：（1）数据项；（2）数据结构；（3）数据流；（4）数据存储；（5）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和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、完盖。

7．什么是数据库的概念结构？试述其特点和设计策略。

答：概念结构是信息世界的结构，即概念模型，其主要特点是：（l）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：l）自顶向下，即首先定义全局概念结构的框架，然后逐步细化；2）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；3）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；4）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’夕这种抽象是：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，' i 5 member of ”的语义。在 E 一 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．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地融合起来，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：（l）多个分 E 一 R 图一次集成;（2）逐步集成，用累加的方式一次集成两个分 E 一 R 图。无论采用哪种方式，每次集成局部 E 一 R 图时都需要分两步走：（l）合并，解决各分 E 一 R 图之间的冲突，将各分 E 一 R 图合并起来生成初步 E 一 R 图；（2）

修改和重构，消除不必要的冗余，生成基本 E 一 R 图。

11．什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 一 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 :(l）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。

14．规范化理论对数据库设计有什么指导意义？

答：规范化理论为数据库设计人员判断关系模式的优劣提供了理论标准，可用以指导关系数据模型的优化，用来预测模式可能出现的问题，为设计人员提供了自动产生各种模式的算法工具，使数据库设计工作有了严格的理论基础。

15．试述数据库物理设计的内容和步骤。

答：数据库在物理设备上的存储结构与存取方法称为数据库的物理结构，它依赖于给定的 DBMS 。为一个给定的逻辑数据模型选取一个最适合应用要求的物理结构，就是数据库的物理设计的主要内容。数据库的物理设计步骤通常分为两步：（l）确定数据库的物理结构，在关系数据库中主要指存取方法和存储结构；（2）对物理结构进行评价，评价的重点是时间效率和空间效率。

16．数据输入在实施阶段的重要性是什么？如何保证输入数据的正确性？

答：数据库是用来对数据进行存储、管理与应用的，因此在实施阶段必须将原有系统中的历史数据输入到数据库。数据量一般都很大，而且数据来源于部门中的各个不同的单位。数据的组织方式、结构和格式都与新设计的数据库系统有相当的差距，组织数据录入就要将各类源数据从各个局部应用中抽取出来，分类转换，最后综合成符合新设计的数据库结构的形式，输入数据库。因此这样的数据转换、组织入库的工作是相当费力费时的工作。特别是原系统是手工数据处理系统时，各类数据分散在各种不同的原始表格、凭证、单据之中，数据输入工作量更大。保证输入数据正确性的方法：为提高数据输入工作的效率和质量，应该针对具体的应用环境设计一个数据录入子系统，由计算机来完成数据入库的任务。在源数据入库之前要采用多种方法对它们进行检验，以防止不正确的数据入库。

17．什么是数据库的再组织和重构造？为什么要进行数据库的再组织和重构造？

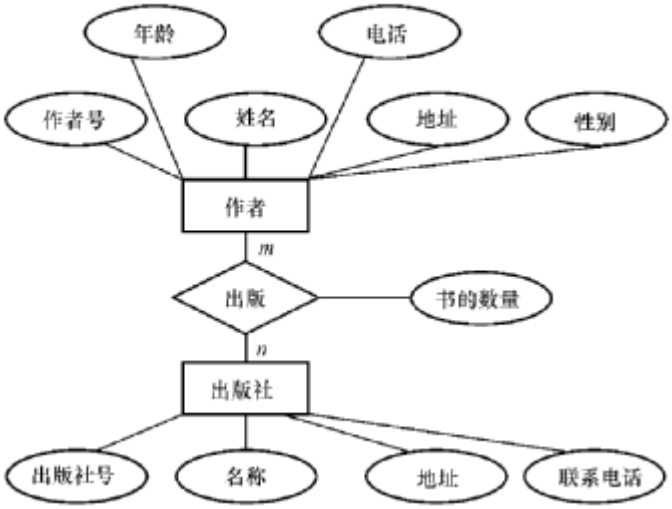
答：数据库的再组织是指：按原设计要求重新安排存储位置、回收垃圾、减少指针链等，以提高系统性能。数据库的重构造则是指部分修改数据库的模式和内模式，即修改原设计的逻辑和物理结构。数据库的再组织是不修改数据库的模式和内模式的。进行数据库的再组织和重构造的原因：数据库运行一段时间后，由于记录不断增、删、改，会使数据库的物理存储情况变坏，降低了数据的存取效率，数据库性能下降，这时 DBA 就要对数据库进行重组织。 DBMS 一般都提供用于数据重组织的实用程序。数据库应用环境常常发生变化，如增加新的应用或新的实体，取消了某些应用，有的实体与实体间的联系也发生了变化等，使原有的数据库设计不能满足新的需求，需要调整数据库的模式和内模式。这就要进行数据库重构造。

18．现有一局部应用，包括两个实体：“出版社”和“作者”，这两个实体是多对多的联系，请读者自己设计适当的属性，画出 E 一 R 图，再将其转换为关系模型（包括关系名、

属性名、码和完整性约束条件）。

答：

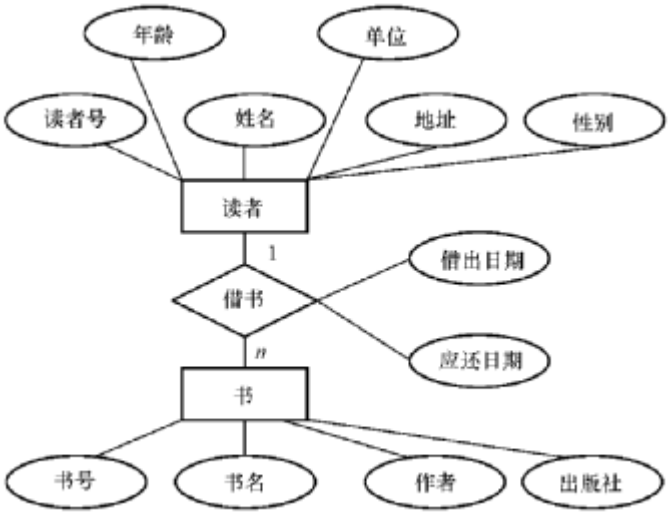
E-R 图为：



关系模型为：作者（作者号，姓名，年龄，性别，电话，地址）出版社（出版社号，名称，地址，联系电话）出版（作者号，出版社号，书的数量）出版关系的主码作者号，出版社号分别参照作者关系的主码作者号和出版社关系的主码出版社号。

19．请设计一个图书馆数据库，此数据库中对每个借阅者保存读者记录，包括：读者号，姓名，地址，性别，年龄，单位。对每本书存有：书号，书名，作者，出版社。对每本被借出的书存有读者号、借出日期和应还日期。要求：给出 E 一 R 图，再将其转换为关系模型。

答 ：E 一 R 图为：



关系模型为：读者（读者号，姓名，地址，性别书（书号，书名，作者，出版社）借书（读者号，书号，借出日期，年龄，单位）应还日期）

第 9 章 关系查询处理和查询优化

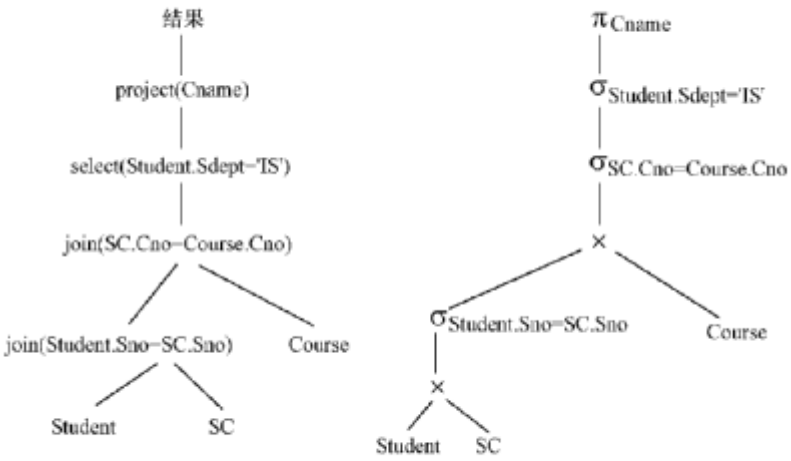
1．试述查询优化在关系数据库系统中的重要性和可能性。

答：重要性：关系系统的查询优化既是 RDBMS 实现的关键技术又是关系系统的优点所在。它减轻了用户选择存取路径的负担。用户只要提出“干什么”，不必指出“怎么干”。查询优化的优点不仅在于用户不必考虑如何最好地表达查询以获得较好的效率，而且在于系统可以比用户程序的“优化’夕做得更好。

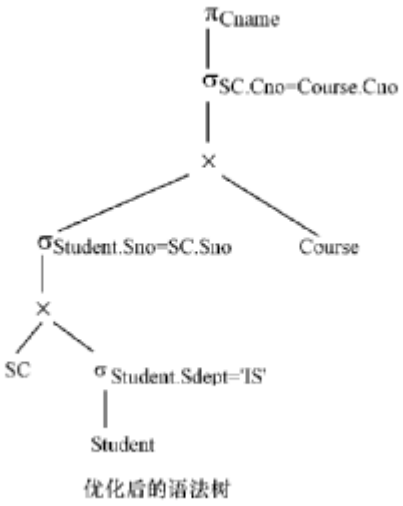
可能性：这是因为：（l）优化器可以从数据字典中获取许多统计信息，例如关系中的元组数、关系中每个属性值的分布情况、这些属性上是否有索引、是什么索引（ B ＋树索引还是 HASH 索引或惟一索引或组合索引）等。优化器可以根据这些信息选择有效的执行计划，而用户程序则难以获得这些信息。（2）如果数据库的物理统计信息改变了，系统可以自动对查询进行重新优化以选择相适应的执行计划。在非关系系统中必须重写程序，而重写程序在实际应用中往往是不太可能的。（3）优化器可以考虑数十甚至数百种不同的执行计划，从中选出较优的一个，而程序员一般只能考虑有限的几种可能性。（4）优化器中包括了很多复杂的优化技术，这些优化技术往往只有最好的程序员才能掌握。系统的自动优化相当于使得所有人都拥有这些优化技术。

2．对学生一课程数据库有如下的查询： SEI 王 CT Cn 即 ne FROM Student , Cou 拐 e , SC WHERE Student . Sno = SC . Sno AND SC . Cllo = Coll 拐 e . Cllo AND Stu 击 nt . Sdept = ! 15 ! ; 此查询要求信息系学生选修了的所有课程名称。试画出用关系代数表示的语法树，并用关系代数表达式优化算法对原始的语法树进行优化处理，画出优化后的标准语法树。

答：



最初的语法树　　关系代数语法树



优化后的语法树

3．试述查询优化的一般准则。

答：下面的优化策略一般能提高查询效率：（1）选择运算应尽可能先做；（2）把投影运算和选择运算同时进行；（3）把投影同其前或其后的双目运算结合起来执行；（4）把某些选择同在它前面要执行的笛卡儿积结合起来成为一个连接运算；（5）找出公共子表达式；（6）选取合适的连接算法。

4．试述查询优化的一般步骤。

答：各个关系系统的优化方法不尽相同，大致的步骤可以归纳如下：（1）把查询转换成某种内部表示，通常用的内部表示是语法树。（2）把语法树转换成标准（优化）形式。即利用优化算法，把原始的语法树转换成优化的形式。（3）选择低层的存取路径。（4）生成查询计划，选择代价最小的。

第 10 章 数据库恢复技术

1．试述事务的概念及事务的 4 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 4 个特性：原子性（Atomicity ）、一致性（consistency ）、隔离性(Isolation ）和持续性（ Durability ）。这 4 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（Perfnanence ），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，或者说是不一致的状态。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；Ql = Ql 一 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：

数据转储和登录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，如上图所示。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，转储结束时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(109 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 DBMs 执行恢复步骤是：
自动完成的，对用户是透明的。
（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。
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，没来得及写入数据库。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（REDO）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（REDO 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”Afte , Image）写入数据库。
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。

10．什么是检查点记录？检查点记录包括哪些内容？
答：
检查点记录是一类新的日志记录。它的内容包括：
① 建立检查点时刻所有正在执行的事务清单
② 这些事务的最近一个日志记录的地址。

11．具有检查点的恢复技术有什么优点？试举一个具体的例子加以说明。答
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。
在采用检查点技术之前，恢复时需要从头扫描日志文件，而利用检查点技术只需要从 T。开始扫描日志，这就缩短了扫描日志的时间。事务 Tl 的更新操作实际上已经写到数据库中了，进行恢复时没有必要再 REDO 处理，采用检查点技术做到了这一点。

12．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（见第 11 题的图）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 一 LIST。

这里建立两个事务队列：

1）UNDO 一 LIST：需要执行 undo 操作的事务集合；

2）REDO 一 LIST：需要执行 redo 操作的事务集合。

把 ACTIVE 一 LIST 暂时放入 UNDO 一 LIST 队列，REDO 队列暂为空。

3）从检查点开始正向扫描日志文件：

① 如有新开始的事务 T *，把 T ＊暂时放入 uNDO 一 LlsT 队列；

② 如有提交的事务毛，把毛从 UNDO 一 LIST 队列移到 REDO 一 LIST 队列，直到日志文件结束；

4）对 UNDO 一 LIST 中的每个事务执行 UNDO 操作，对 REDO 一 LIST 中的每个事务执行 REDO 操作。

13．什么是数据库镜像？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。

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会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’夕数据。(1）丢失修改（lost update) 两个事务 Tl 和 T2 读入同一数据并修改，T2 提交的结果破坏了（覆盖了） Tl 提交的结果，导致 Tl 的修改被丢失。 (2 ）不可重复读（ Non 一 Repeatable Read) 不可重复读是指事务 Tl 读取数据后，事务几执行更新操作，使 Tl 无法再现前一次读取结果。(3 ）读“脏”数据（ Dirty Read) 读“脏’夕数据是指事务 Tl 修改某一数据，并将其写回磁盘，事务几读取同一数据后， Tl 由于某种原因被撤销，这时 Tl 已修改过的数据恢复原值，几读到的数据就与数据库中的数据不一致，则几读到的数据就为“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？基本的封锁类型有几种？试述它们的含义。

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。

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。

4．如何用封锁机制保证数据的一致性？

答： DBMS 在对数据进行读、写操作之前首先对该数据执行封锁操作，例如下图中事务 Tl 在对 A 进行修改之前先对 A 执行 xock (A)，即对 A 加 x 锁。这样，当几请求对 A 加 x 锁时就被拒绝，几只能等待 Tl 释放 A 上的锁后才能获得对 A 的 x 锁，这时它读到的 A 是 Tl 更新后的值，再按此新的 A 值进行运算。这样就不会丢失 Tl 的更新。

T_1	T_2
① Xlock A 获得	
② 读 A = 16	
	Xlock A
③ A←A－1	等待
写回 A = 15	等待
Commit	等待
Unlock A	等待
④	获得 Xlock A
	读 A = 15
	A←A－1
	写回 A = 14
⑤	Commit
	Unlock A

DBMS 按照一定的封锁协议，对并发操作进行控制，使得多个并发操作有序地执行，就可以避免丢失修改、不可重复读和读“脏’夕数据等数据不一致性。

5．什么是活锁？什么是死锁？

答：

T_1	T_2	T_3	T_4
lock R	•	•	•
•	lock R	•	•
•	等待	Lock R	•
Unlock	等待	•	Lock R
•	等待	Lock R	等待
•	等待	•	等待
•	等待	Unlock	等待
•	等待	•	Lock R
•	等待	•	•

如果事务 Tl 封锁了数据 R，事务几又请求封锁 R，于是几等待。几也请求封锁 R，当 Tl 释放了 R 上的封锁之后系统首先批准了几的请求，几仍然等待。然后几又请求封锁 R，当几释放了 R 上的封锁之后系统又批准了几的请求 … … 几有可能永远等待，这就是活锁的情形。活锁的含义是该等待事务等待时间太长，似乎被锁住了，实际上可能被激活。如果事务 Tl 封锁了数据 Rl，几封锁了数据凡，然后 Tl 又请求封锁几，因几已封锁了几，于是 Tl 等待几释放几上的锁。接着几又申请封锁 Rl，因 Tl 已封锁了 Rl，几也只能等待 Tl 释放 Rl 上的锁。这样就出现了 Tl 在等待几，而几又在等待 T｝的局面， T｝和几两个事务永远不能结束，形成死锁。

T_1	T_2
lock R_1	•
•	Lock R_2
•	•
Lock R_2	•
等待	•
等待	Lock R_1
等待	等待

6．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。

11．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时，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。

12．什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执行它们时的结果相同，称这种调度策略为可串行化的调度。

9．设 Tl，几，几是如下的 3 个事务：

Tl ： A : = A + 2 ;

T2： A : = A * 2 ;

T3： A : = A **2 ; (A <-A*A）

设 A 的初值为 0 。

(l ）若这 3 个事务允许并行执行，则有多少可能的正确结果，请一一列举出来。

答 ： A 的最终结果可能有 2 、4 、8 、16 。因为串行执行次序有 Tl T2T3、Tl T3T2、T2T1T3、T2T3Tl 、T3T1T2、T3T2 Tl 。对应的执行结果是 16 、 8 · 4 · 2 · 4 · 2 。

(2 ）请给出一个可串行化的调度，并给出执行结果

答：

T_1	T_2	T_3
Slock A		
Y = A = 0		
Unlock A		
Xlock A		
	Slock A	
A = Y + 2	等待	
写回 A(= 2)	等待	
Unlock A	等待	
	Y = A = 2	
	Unlock A	
	Xlock A	
		Slock A
	A = Y * 2	等待
	写回 A(= 4)	等待
	Unlock A	等待
		Y = A = 4
		Unlock A
		Xlock A
		A = Y * * 2
		写回 A(= 16)
		Unlock A

最后结果 A 为 16，是可串行化的调度。

(3）请给出一个非串行化的调度，并给出执行结果。

答：

T_1	T_2	T_3
Slock A		
Y = A = 0		
Unlock A		
	Slock A	
	Y = A = 0	
Xlock A		
等待	Unlock A	
A = Y + 2		
写回 A(= 2)		Slock A
Unlock A		等待
		Y = A = 2
		Unlock A
		Xlock A
	Xlock A	
	等待	Y = Y * * 2
	等待	写回 A(= 4)
	等待	Unlock A
	A = Y * 2	
	写回 A(= 0)	
	Unlock A	

最后结果 A 为 0，为非串行化的调度。

(4）若这 3 个事务都遵守两段锁协议，请给出一个不产生死锁的可串行化调度。

答：

T_1	T_2	T_3
Slock A		
Y = A = 0		
Xlock A		
A = Y + 2	Slock A	
写回 A(= 2)	等待	
Unlock A	等待	
	Y = A = 2	
	Xlock A	
Unlock A	等待	Slock A
	A = Y * 2	等待
	写回 A(= 4)	等待
	Unlock A	等待
		Y = A = 4
	Unlock A	
		Xlock A
		A = Y * * 2
		写回 A(= 16)
		Unlock A
		Unlock A

(5）若这 3 个事务都遵守两段锁协议，请给出一个产生死锁的调度。

答：

T_1	T_2	T_3
Slock A		
Y = A = 0		
	Slock A	
	Y = A = 0	
Xlock A		
等待		
	Xlock A	
	等待	
		Slock A
		Y = A = 0
		Xlock A
		等待

11．试证明，若并发事务遵守两段锁协议，则对这些事务的并发调度是可串行化的。

证明：首先以两个并发事务 Tl 和 T2 为例，存在多个并发事务的情形可以类推。根据可串行化定义可知，事务不可串行化只可能发生在下列两种情况：

(l）事务 Tl 写某个数据对象 A，T2 读或写 A；

(2）事务 Tl 读或写某个数据对象 A，T2 写 A。

下面称 A 为潜在冲突对象。

设 Tl 和 T2 访问的潜在冲突的公共对象为{A1，A2 …， An｝。不失一般性，假设这组潜在冲突对象中 X＝(A 1 , A2， …， Ai｝均符合情况 1。 Y＝{A i + 1， …， An｝符合所情况（ 2）。

VX ∈ x , Tl 需要 XlockX ①

T2 需要 Slockx 或 Xlockx ②

1 ）如果操作 ① 先执行，则 Tl 获得锁，T2 等待

由于遵守两段锁协议， Tl 在成功获得 x 和 Y 中全部对象及非潜在冲突对象的锁后，才会释放锁。

这时如果存在 w ∈ x 或 Y，T2 已获得 w 的锁，则出现死锁；否则， Tl 在对 x 、Y 中对象全部处理完毕后，T2 才能执行。这相当于按 Tl 、T2 的顺序串行执行，根据可串行化定义， Tl 和几的调度是可串行化的。

2 ）操作 ② 先执行的情况与（ l ）对称因此，若并发事务遵守两段锁协议，在不发生死锁的情况下，对这些事务的并发调度一定是可串行化的。证毕。

12 ．举例说明，对并发事务的一个调度是可串行化的，而这些并发事务不一定遵守两段锁协议。

答：

T_1	T_2
Slock B	
读 B = 2	
Y = B	
Unlock B	
Xlock A	
	Slock A
A = Y + 1	等待
写回 A = 3	等待
Unlock A	等待
	等待
	Slock A
	读 A = 3
	X = A
	Unlock A
	Xlock B
	B = X + 1
	写回 B = 4
	Unlock B

13 ．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无（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。

14．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（意向功口 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：

T_1 \ T_2	S	X	IS	IX	SIX	–
S	Y	N	Y	N	N	Y
X	N	N	N	N	N	Y
IS	Y	N	Y	Y	Y	Y
IX	N	N	Y	Y	N	Y
SIX	N	N	Y	N	N	Y
–	Y	Y	Y	Y	Y	Y

15．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）

16．试述你了解的某一个实际的 DBMS 产品的并发控制机制。

答：（略，参见简单介绍了有关 Oracle 的并发控制机制。）