第一章 绪论

1．试述数据、数据库、数据库管理系统、数据库系统的概念。

答：

（1）数据：是数据库中存储的基本对象，是用来描述事物的符号。

（2）数据库：是长期储存在计算机内的、有组织的、可共享的、大量数据的集合。

（3）数据库管理系统：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。

（4）数据库系统：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。

2．使用数据库系统有什么好处?

答：使用数据库系统的好处主要是：可以大大提高应用系统的开发效率；方便用户的使用；减轻数据库系统管理人员维护数据库系统的负担，等等。

3．试述文件系统与数据库系统的区别和联系。

答：

（1）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，由应用程序自行控制，系统难以扩充。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。

（2）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：

（1）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等，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。

5．试述数据库系统的特点。

答：数据库系统的主要特点有：

（1）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。

（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。

（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。

6．数据库管理系统的主要功能有哪些?

答:

（1）数据定义功能;

（2）数据的组织、存储和管理;

（3）数据操纵功能;

（4）数据库的事务管理和运行管理

（5）数据库的建立和维护功能;

（6）其他功能，如不同数据库之间的互访和互操作，不同数据库管理系统之间的转换功能等。

7．什么是概念模型？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，用于信息世界的建模，是对现实世界的第一层抽象，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所使用的语言。

8．定义并解释概念模型中以下术语:

实体、实体型、实体集、实体之间的联系

答:

（1）实体：客观存在并可相互区别的事物。

（2）实体型：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。

（3）实体集：同一类型实体的集合。

（4）实体之间的联系：不同实体集之间的联系，有一对一、一对多和多对多等多种类型。

9．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：

（1）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。

（2）数据模型用于抽象地表示现实世界的数据和信息。

（3）数据模型的三个要素是：数据结构、数据操作及完整性约束条件。

10．试述层次模型的概念，并举出三个层级模型的实例。

答：

（1）在数据库中定义满足下面两个条件的基本层次联系的集合为层次模型：其一，有且只有一个结点没有双亲结点，这个结点称为根结点；其二，根结点以外的其它结点有且只有一个父结点。

（2）实例略，参考教材 1.2.5。

11．试述网状模型的概念，并举出三个网状模型的实例。

答：

（1）在数据库中，把满足下面两个条件的基本层次联系的集合为网状模型：其一，允许一个以上的结点无双亲；其二，一个结点可以有多于一个的双亲。

（2）实例略，参考教材 1.2.6。

12．试述网状、层次数据库的优缺点。

答：

（1）网状数据库的优点是：能够更为直接地描述现实世界；具有良好的性能，存取效率较高。网状数据库的缺点是：结构比较复杂，而且随着应用规模的扩大，数据库的结构会变得越来越复杂，不利于最终用户掌握；操作语言比较复杂，用户不易使用。

（2）层次数据库的优点是：数据结构比较简单、清晰；查询效率高；提供了良好的完整性支持。层次数据库的缺点是：现实世界中很多联系是非层次的，

层次模型在表示这类联系时只能通过引入冗余数据或创建非自然的数据结构，对插入和删除操作的限制比较多，因此应用程序编写比较复杂；查询孩子结点时必须通过双亲结点；由于结构严密，层次命令趋向于程序化。

13．试述关系模型的概念，定义并解释以下术语。

关系、属性、域、元组、码、分量、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 3 部分组成，从用户的角度看，关系模型中数据的逻辑结构是一张由行和列组成的二维表。

（1）关系：一个关系对应一张二维表。

（2）属性：关系表中的一列即为一个属性。

（3）域：属性的取值范围。

（4）元组：关系表中的一行即为一个元组。

（5）码：关系表中的某个属性组，它可以唯一确定一个元组。

（6）分量：元组中的一个属性值。

（7）关系模式：通过关系名和属性名列表对关系进行描述，相当于二维表的表头部分（即表的描述部分）。

14．试述关系数据库的特点。

答:

（1）关系数据模型具有下列优点：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；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；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。

（2）关系数据模型最主要的缺点是：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。

15．试述数据库系统三级模式结构，这种结构的优点是什么?

答:

（1）数据库系统三级模式结构由外模式、模式和内模式组成。① 模式（也称为逻辑模式）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。② 外模式（也称为子模式或用户模式）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。外模式涉及的是数据的局部逻辑结构，通常是模式的子集。③ 内模式（也称为存储模式）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。

（2）数据库系统三级模式结构的优点在于：这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。

16．定义并解释以下术语。

模式、外模式、内模式、数据定义语言、数据操纵语言

答:

（1）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。

（2）外模式：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。

（3）内模式：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。

（4）数据定义语言：用来定义数据库模式、外模式、内模式的语言。

（5）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。

17．什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：

（1）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。

（2）数据与程序的逻辑独立性是指：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。

（3）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。

18．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、数据库管理员和最终用户所组成。

19 .试述数据库管理员、系统分析员、数据库设计人员、应用程序员的职责。

答：

（1）数据库管理员的职责是：负责全面地管理和控制数据库系统。具体职责包括： 决定数据库的信息内容和结构；决定数据库的存储结构和存取策略；定义数据的安全性要求和完整性约束条件；监督和控制数据库的使用和运行；改进、重组和重构数据库系统。

（2）系统分析员的职责是：负责应用系统的需求分析和规范说明；确定系统的软、硬件配置；参与数据库系统的概要设计。

（3）数据库设计人员的职责是：负责数据库中数据的确定以及数据库各级模式的设计。

（4）应用程序员的职责是：设计和编写应用系统的程序模块，并进行调试和安装。

第二章 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个部分组成。

2．试述关系数据语言的特点和分类。

答：

（1）关系数据语言的特点是：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。

（2）关系数据语言可以分为三类：① 关系代数语言 ② 关系演算语言，包括元组关系演算语言和域关系演算语言 ③ 具有关系代数和关系演算双重特点的语言。

3．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。

答：

（1） 域：一组具有相同数据类型的值的集合。

笛卡尔积：给定一组域 D_1，D_2，...，D_n，这组域的笛卡尔积为：

$D_1 \times D_2 \times ... \times D_n = \{(d_1, d_2, ..., d_n) \mid d_i \in D_i, i = 1, 2, ..., n\}$ 。

关系：$D_1 \times D_2 \times ... \times D_n$ 的子集称为在域 D_1，D_2，...，D_n 上的关系，关系是一个二维表。

元组：关系中的每个元素（即表中的一行）称为关系中的元组。

属性：关系表的列名称为属性。

(2) 候选码：关系中能唯一地标识一个元组的属性组称为候选码，一个关系可以有多个候选码。

主码：从关系的多个候选码中确定其中的一个候选码为主码，主码是唯一的。

外码：某组在本关系中非码的属性组，如果与另一个关系中的主码相对应，则称之为本关系中的外码。

(3) 关系模式：是对关系的描述，可以形式化地表示为： R(U,D,DOM,F)

关系：$D_1 \times D_2 \times ... \times D_n$ 的子集叫作在域 D_1，D_2，...，D_n 上的关系，表示为： R（D_1，D_2，...，D_n）

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，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，通常简称为关系数据库。

4．举例说明关系模式和关系的区别。

答：略。

5．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？什么情况才可以为空？

答：

(1) 关系模型中有三类完整性规则约束：① 实体完整性：主属性不能取空值；② 参照完整性：若 F 是关系 R 的外码，且它与关系 S 的主码 K 相对应，则 F 的取值或者为空值，或者为 S 中某元组的主码 K 的值；③ 用户自定义的完整性：针对某一具体关系数据库的约束条件，反映某一具体应用所涉及的数据必须满足的语义要求。

(2) 在参照完整性中，当外码属性值可以为空，表示该属性的值尚未确定，但是前提是该外码不是其所在关系的主属性。

6．设有一个 SPJ 数据库，包括 S，P，J，SPJ 四个关系模式。试用关系代数、ALPHA 语言、QBE 语言完成如下查询：（题目略）

（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：

答：（只给出关系代数解法，ALPHA 语言和 QBE 语言解法略）

（1）$\pi_{Sno}(\sigma_{Jno='J1'}(SPJ))$

（2）$\pi_{Sno}(\sigma_{Jno='J1' \wedge Pno='P1'}(SPJ))$

（3）$\pi_{Sno}(\sigma_{Jno='J1'}(\sigma_{COLOR='红'}(P) \bowtie SPJ))$

（4）$\pi_{Jno}(SPJ) - \pi_{JNO}(\sigma_{city='天津' \wedge Color='红'}(S \bowtie SPJ \bowtie P))$

（5）$\pi_{Jno,\ Pno}(SPJ) \div \pi_{Pno}(\sigma_{Sno='S1'}(SPJ))$

7．试述等值连接与自然连接的区别和联系。

答：

（1）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。

（2）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。

8．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：

（1）关系代数的基本运算有：并、差、笛卡尔积、投影和选择。

（2）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。（表达方法略）

第三章 关系数据库标准语言 SQL

1．试述 SQL 的特点

答：SQL 是关系数据库的标准语言，是一个通用的、功能极强的关系数据库语言，它的主要特点是：

（1）综合统一：SQL 集数据定义语言（DDL），数据操纵语言（DML），数据控制语言（DCL）功能于一体；可以独立完成数据库生命周期中的全部活动；在用户数据库投入运行后，可根据需要随时逐步修改模式，不影响数据的运行；数据操作符统一，克服了非关系系统由于信息表示方式的多样性带来的操作复杂性。

（2）高度非过程化：SQL 只要提出“做什么”，无须了解存取路径（存取路径的选择以及 SQL 的操作过程由系统自动完成），减轻了用户的负担，有利于提高数据独立性。

（3）面向集合的操作方式：SQL 采用集合操作方式， 操作对象、查找结果可以是元组的集合， 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法：SQL 是独立的语言，能够独立地用于联机交互的使用方式；SQL 又是嵌入式语言，能够嵌入到高级语言程序中，供程序员设计程序时使用。

（5）语言简洁，易学易用。

2．说明在 DROP TABLE 时，RESTRICT 和 CASCADE 的区别

答：RESTRICT 和 CASCADE 是 DROP TABLE 的两种删除方式，二者必选其一。其中，RESTRICT 是约束方式，表示如果该模式下已经定义了下属的 DB 对象（如表、视图等），则拒绝该删除语句的执行；CASCADE 是级联方式，表示在删除模式的同时将该模式中的所有 DB 对象全部一起删除。

3．有两个关系 S(A, B, C, D)和 T(C, D, E, F)，写出与下列查询等价的 SQL 表达式：

（1）$\sigma_{A=10}(S)$　（2）$\pi_{A,B}(S)$　（3）$S \bowtie T$

（4）$S \underset{S.C=T.C}{\bowtie} T$　（5）$S \underset{A<E}{\bowtie} T$　（6）$\pi_{C,D}(S) \times T$

（1）$\sigma_{A=10}(S)$

```
SELECT * FROM S WHERE A=10
```

（2）$\pi_{A,B}(S)$

```
SELECT A, B FROM S
```

（3）$S \bowtie T$

```
SELECT S.*, T.E, T.F FROM S, T WHERE S.C=T.C AND S.D=T.D
```

（4）$S \underset{S.C=T.C}{\bowtie} T$

```
SELECT * FROM S, T WHRE S.C=T.C
```

（5）$S \underset{A<E}{\bowtie} T$

```
SELECT * FROM S, T WHRE A < E
```

（6）$\pi_{C,D}(S) \times T$

```
SELECT S.C, S.D, T.* FROM S, T
```

4. 用 SQL 语句建立第二章习题 6 中的 4 个表；针对建立的 4 个表用 SQL 完成第二章习题 6 中的查询。

答：

建表：

(1) S 表：　S (SNO , SNAME , STATUS , CITY) ;

```
CREATE TABLE  S  (  SNO CHAR(2) PRIMARY KEY,
                    SNAME CHAR(20),
                    STATUS CHAR(2),
                    CITY CHAR(10)  );
```

(2) P (PNO , PNAME , COLOR , WEIGHT);

```
CREATE TABLE  P  (  PNO CHAR(2) PRIMARY KEY,
                    PNAME CHAR(20),
                    COLOR CHAR(4),
                    WEIGHT  INT );
```

(3) J (JNO , JNAME , CITY) ;

```
CREATE TABLE  J  (  JNO CHAR(2)  PRIMARY KEY,
                    JNAME CHAR(20),
                    CITY CHAR(10) );
```

(4) SPJ (SNO , PNO , JNO , QTY) ;

```
CREATE TABLE  SPJ  ( SNO CHAR(2),
                     PNO CHAR(2),
                     JNO CHAR(2),
                     QTY INT,
                     PRIMARY KEY (SNO, PNO, JNO),
    FOREIGN KEY (SNO) REFERENCES S(SNO),
    FOREIGN KEY (PNO) REFERENCES P(PNO),
    FOREIGN KEY (JNO) REFERENCES J(JNO) );
```

查询:

(1) 求供应工程 J1 零件的供应商号码 SNO ;

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

(2) 求供应工程 J1 零件 P1 的供应商号码 SNO ;

```
SELECT DISTINCT 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 DISTINCT SNO
FROM SPJ, P
WHERE SPJ.PNO=P.PNO AND JNO = 'J1' 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 DISTINCT JNO
FROM SPJ
WHERE JNO NOT IN ( SELECT JNO
                   FROM SPJ, P, S
                   WHERE  S.SNO=SPJ.SNO AND
                          P.PNO=SPJ.PNO AND
                          S.CITY='天津' AND
                          COLOR='红' )
```

(5) 求至少用了供应商 S1 所供应的全部零件的工程号 JNO:

```
SELECT DISTINCT JNO
FROM SPJ S1
WHERE NOT EXISTS
        (   SELECT *
            FROM SPJ S2
            WHERE S2.SNO='S1' AND NOT EXISTS
                (   SELECT *
                    FROM SPJ S3
                    S3.JNO=S1.JNO AND S3.PNO=S2.PNO) );
```

5. 针对习题 4 中的四个表试用 SQL 语言完成以下各项操作:

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

(7) 找出没有使用天津产的零件的工程号码。

```
SELECT DISTINCT JNO
FROM SPJ
WHERE JNO NOT IN ( SELECT JNO
                   FROM   SPJ, S
                   WHERE S.SNO=SPJ.SNO AND
                         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。

```
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。

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

（11）请将(S2，P4，J6，200)插入供应情况关系。

```
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：

（1）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。

（2）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。也就是说，数据库中只存放视图的定义而不存放视图对应的数据，这些数据仍存放在导出视图的基本表中。

（3）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．哪类视图是可以更新的？哪类视图是不可以更新的？各举一例说明。

答：

（1）基本表的行列子集视图一般是可更新的。例如：

```
CREAT VIEW V_Student
AS  SELECT *  FROM  Student;
```

（2）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。例如：

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

9．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 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。

② 找出供应商 S1 的供应情况。

答：供应情况视图

（1）

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

（2）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 DISTINCT PNO, QTY
FROM VIEW_SP
```

② 找出供应商 S1 的供应情况。

```
SELECT DISTINCT *
FROM VIEW_SP
WHERE SNO='S1'
```

第四章 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. 举例说明对数据库安全性产生威胁的因素。

答：略。

3. 试述信息安全标准的发展历史，试述 CC 评估保证级划分的基本内容。

答：信息安全标准的发展历史参见教材图 4.1。

CC 评估保证级划分的基本内容：

（1） EAL1：功能测试；

（2） EAL2：结果测试；

（3） EAL3：系统的测试和检查；

（4） EAL4：系统的设计、测试和复查；

（5） EAL5：半形式化设计和测试；

（6） EAL6：半形式化验证的设计与测试；

（7） EAL7：形式化验证的设计与测试。

4．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。

（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。

5．什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:

（1）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。

（2）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。

6．对下列两个关系模式:

学生（学号、姓名、年龄、性别、家庭住址、班级号）

班级（班级号、班级名、班主任、班长）

使用 GRANT 语句完成下列授权功能:

答:

（1）授予用户 U1 对两个表的所有权限，并可给其他用户授权。

```
GRANT ALL PRIVILEGES
ON TABLE 学生, 班级
TO U1
WITH GRANT OPTION
```

（2）授予用户 U2 对学生表具有查看权限，对家庭住址具有更新权限。

```
GRANT SELECT, UPDATE(家庭住址)
ON TABLE 学生
TO U2
```

（3）将对班级表查看权限授予所有用户。

```
GRANT SELECT
ON TABLE 班级
TO PUBLIC;
```

（4）将对学生表的查询、更新权限授予角色 R1。

```
GRANT SELECT, UPDATE
ON TABLE 学生
TO R1
```

（5）将角色 R1 授予用户 U1，并且 U1 可继续授权给其他角色。

```
GRANT R1
TO U1
WITH ADMIN OPTION
```

7．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。

答：

（1）用户王明对两个表有 SELECT 权限。

```
GRANT SELECT
ON TABLE 职工, 部门
TO 王明
```

（2）用户李勇对两个表有 INSERT 和 DELETE 权限。

GRANT INSERT, DELETE

ON 职工, 部门

TO 李勇

(3) 每个职工只对自己的记录有 SELECT 权限。

GRANT SELECT

ON TABLE 职工

WHEN USER( ) = 姓名

TO ALL;

这里假定系统的 GRANT 语句支持 WHEN 子句和 USER () 的使用。注意，不同的系统这些扩展语句可能是不同的。

(4) 用户刘星对职工表有 SELECT 权限，对工资字段具有更新权限。

GRANT SELECT, UPDATE(工资)

ON TABLE 职工

TO 刘星

(5) 用户张新具有修改这两个表的结构的权限。

GRANT ALTER TABLE

ON TABLE 职工,部门

TO 张新;

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

GRANT ALL PRIVILIGES

ON TBALE 职工, 部门

TO 周平

WITH GRANT OPTION;

(7) 用户杨兰具有从每个部门职工中 SELECT 最高工资、最低工资、平均工资的权限，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;
```

8 ．把习题 7 中（1）~（7）的每一种情况，撤销各用户所授予的权力。

答:

```
(1)   REVOKE SELECT
      ON TABLE  职工，部门
      FROM  王明;
(2)   REVOKE INSERT , DELETE
      ON TABLE  职工，部门
      FROM  李勇;
(3)   REOVKE SELECT
      ON TABLE  职工
      WHEN USER ( ) =  姓名
      FROM ALL;
(4)   REVOKE SELECT , UPDATE
      ON TABLE  职工
      FROM  刘星;
(5)   REVOKE ALTER TABLE
      ON TABLE  职工，  部门
      FROM  张新;
(f)   REVOKE ALL PRIVILIGES
      ON TABLE  职工，  部门
      FROM  周平;
(g)   REVOKE SELECT
```

```
ON 部门工资
FROM 杨兰;
DROP VIEW 部门工资;
```

9．解释强制存取控制机制中主体、客体、敏感度标记的含义。

答:

（1）主体：系统中的活动实体，包括 DBMS 所管理的实际用户和代表用户的各进程。

（2）客体：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。

（3）敏感度标记：对于主体和客体，DBMS 为它们的每个实例（值）指派一个敏感度标记（Label）。敏感度标记被分为若干级别，例如：绝密、机密、可信、公开等。主体的敏感度标记称为许可证级别；客体的敏感度标记称为密级。

10．举例说明 MAC 机制如何确定主体能否存取客体。

答：假设要对关系变量 S 进行 MAC 控制，为了简化起见，设所要控制存取的数据单元是元组，每个元组均被标以密级，如下表所示:

编号	姓名	敏感度
S1	TOM	可信
S2	JERRY	机密
S3	MARY	绝密

设用户 U1 的许可证级别是“机密”，则该用户可以查询元组 S1 和 S2，可以修改元组 S2；用户 U2 的许可证级别是“可信”，则该用户只能查询和修改元组 S1。

11．什么是数据库的审计功能，为什么要提供审计功能。

答:

（1）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。

（2）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。

第五章 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：

（1）数据的完整性，是指防止数据库中存在不符合语义的数据，也就是防止数据库中存在不正确的数据。所防范的对象是：不合语义的、不正确的数据。

（2）数据的安全性，是指保护数据库防止恶意的破坏和非法的存取，所防范的对象是：非法用户和非法操作。

3．什么是数据库的完整性约束条件?

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。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．关系数据库管理系统的完整性控制机制应具有哪三方面的功能?

答：

（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．关系数据库管理系统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:①级联删除；② 受限删除；③ 置空值删除。

(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。；③ 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:

(1) 定义每个模式的主码;
(2) 定义参照完整性;
(3) 定义职工年龄不得超过 60 岁。

答:

(1) 职工（职工号，姓名，年龄，职务，工资，部门号）

```
CREATE TABLE DEPT
        (   Deptno CHAR(10) PRIMARY KEY,
            Deptname CHAR(20),
            Manager CHAR(20),
            PhoneNumber Char(11) );
```

(2) 部门（部门号，名称，经理名，电话）

```
CREATE TABLE EMP
```

```
(   Empno CHAR(10) PRIMARY KEY,
    Ename CHAR(20),
    Eage SMALLINT CHECK (Eage<=60),
    Job CHAR(10),
    Salary NUMERIC (7,2),
    Deptno CHAR(10),
    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。

8．某单位想举行一个小型的联谊会，关系 Male 记录注册的男宾信息，关系 Female 记录注册的女宾信息。建立一个断言，将来宾的人数限制在 50 人以内。（提示，先创建关系 Female 和关系 Male）

答:

```
CREATE TABLE Male
        (   ID INT PRIMARY KEY,
            Name CHAR(10)
            PhoneNumber Char(11) );
CREATE TABLE Female
        (   ID INT PRIMARY KEY,
            Name CHAR(10)
            PhoneNumber Char(11) );
CREATE ASSERTION ASS_PEOPLE_NUM
```

CHECK (50>= ((SELECT count(*) FROM Male) + (SELECT count(*) FROM Female))

第六章 关系数据理论

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-key ）、1NF、2NF 、3NF 、BCNF、多值依赖、4NF。

答:

（1）函数依赖：设 R(U)是一个属性集 U 上的关系模式，X 和 Y 是 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。

（2）完全函数依赖和部分函数依赖：在关系模式 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，记作 $X \xrightarrow{F} Y$。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，记作 $X \xrightarrow{P} Y$。

（3）传递依赖：在关系模式 R(U)中，如果 X→Y，Y→Z，且 $Y \not\subseteq X$，$Y \not\rightarrow X$，则称 Z 对 X 传递函数依赖，记作 $X \xrightarrow{传递} Z$。

（4）候选码：设 K 为关系模式 R<U,F>中的属性或属性组合。若 $K \xrightarrow{F} U$，则 K 称为 R 的一个候选码（Candidate Key）

（5）主码：若关系模式 R 有多个候选码，则选定其中的一个作为主码（Primary key）。

（6）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（Foreign key），也称外码。

（7）全码：整个属性组 U 是码，称为全码（All-key）。

（8）1NF：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)。

（9）2NF：若关系模式 R∈1NF，并且每一个非主属性都完全函数依赖于 R 的码，则 R∈2NF。

（10）3NF：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⊈Y），使得 X→Y，Y ↛ X，Y→Z，成立，则称 R<U，F> ∈ 3NF。

（11）BCNF：设关系模式 R<U，F>∈1NF，如果对于 R 的每个函数依赖 X→Y，若 Y 不属于 X，则 X 必含有候选码，那么 R∈BCNF。

（12）多值依赖：设 R（U）是属性集 U 上的一个关系模式。X、Y、Z 是 U 的子集，并且 Z=U-X-Y。关系模式 R（U）中的多值依赖 X→→Y 成立，当且仅当对 R（U）的任一关系 r，给定一对（x, z）值，有一组 Y 值，这组值取决于 x 值而与 z 值无关。

（13）4NF：关系模式 R<U，F>∈1NF，若对于 R 的每个非平凡多值依赖 X→→Y（Y⊈X），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，并说明是否有全码存在。

答:

（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）每个关系模式的极小函数依赖集如下：

① 学生 S (Sno，Sname，Sbirth，Dept，Class，Rno) 的最小函数依赖集是: F_S = { Sno→Sname，Sno→Sbirth，Sno→Class，Class→Dept，DEPT→Rno }

传递依赖如下：

(i) 由于 Class→Dept，Dept↛Class，Dep→Rno

所以 Class 与 Rno 之间存在着传递函数依赖。

(ii) 由于 Sno→Class，Class↛Sno，Class→Dept

所以 Sno 与 Dept 之间存在着传递函数依赖。

(iii) 由于 Sno→Dept，而 Dept↛Sno ，Dept→Rno

所以 Sno 与 Rno 之间存在着传递函数依赖。

② 班级 C(Class，Pname，Dept，Cnum，Cyear)的极小函数依赖集如下:

F_C = { Class→Pname，Class→Cnum，Class→Cyear，Pname→Dept }

传递依赖如下：

由于 Class→Pname，Pname↛Class，Pname→Dept，

所以 C1ass 与 Dept 之间存在着传递函数依赖。

③ 系 D(Dept，Dno，Office，Dnum)的极小函数依赖集如下：

F_D = { Dept→Dno，Dno→Dept，Dno→Office，Dno→Dnum }

该模式并不存在传递函数依赖。

④ 学会 M(Mname，Myear，Maddr，Mnum)的最小函数依赖集如下：

F_M = { Mname→Myear，Mname→Maddr，Mname→Mnum }

该模式并不存在传递函数依赖。

（3）各关系模式的候选码、外部码，全码如下：

① 学生 S 候选码：Sno；外部码：Dept、Class；无全码

② 班级 C 候选码：Class；外部码：Dept；无全码

③ 系 D 候选码：Dept 或 Dno；无外部码；无全码

④ 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→Z , X→Y ， 则有 X→YZ

（2）伪传递规则：由 X →Y ， WY→Z 有 XW→Z

（3）分解规则：X→Y , Z⊆Y ， 有 X→Z

证明：

（1）合并规则：若 X→Z , X→Y ， 则有 X→YZ

已知 X→Z ，由增广律知 XY→YZ；又因为 X→Y，可得 X→XY，根据传递律得 X→YZ。

（2）伪传递规则：由 X →Y ， WY→Z 有 XW→Z

已知 X→Y，据增广律得 XW→WY，又因为 WY→Z，根据传递律得 XW→Z。

（3）分解规则：X→Y , Z⊆Y ， 有 X→Z

已知 Z⊆Y ，根据自反律知 Y→Z，又因为 X→Y，根据传递律得 X→Z。

6．有关系模式 R(A,B,C,D,E)，回答下面各个问题：

（1）若 A 是 R 的候选码，具有函数依赖 BC->DE，那么在什么条件下 R 是 BCNF?

（2）如果存在函数依赖 A->B，BC->D，DE->A，列出 R 的所有码。

（3）如果存在函数依赖 A->B，BC->D，DE->A，R 属于 3NF 还是 BCNF?

答:

（1）只有当（B,C）也是候选码的时候，R 才属于 BCNF。

（2）R 的码：（A,C,E）

（3）R 属于 3NF，由于不是所有的决定因素都包含码，故 R 不属于 BCNF。

7．下面的结论哪些是正确的？哪些是错误的？对于错误的请给出一个反例说明之。

（1）任何一个二目关系是属于 3NF 的。

（2）任何一个二目关系是属于 BCNF 的。

（3）任何一个二目关系是属于 4NF 的。

（4） 当且仅当函数依赖 A->B 在 R 上成立，关系 R(A,B,C)等于其投影 R_1(A,B)和 R_2(A,C)的连接。

（5）若 R.A->R.B，R.B->R.C，则 R.A->R.C。

（6）若 R.A->R.B，R.A->R.C，则 R.A->R.(B,C)。

（7）若 R.B->R.A，R.C->R.A，则 R.(B,C)->R.A。

（8）若 R.(B,C)->R.A，R.B->R.A，则 R.C->R.A。

答:

（1）错误。课件中的 SL（Sno, Sdept, Sloc），Sno → Sdept，Sdept → Sloc，Sno → Sloc（传递函数依赖）就不属于 3NF。

（2）错误。课件中的 STC（S，T，C），（S，C）→ T，（S，T）→ C，T → C 就不属于 BCNF。

（3）错误，举例略。

（4）错误，还应该有 A->C 在 R 上成立。例如：课件中的 SL（Sno， Sdept，Sloc），Sdept -> Sloc 成立，但是 SL 并不等于其投影（Sdept， Sloc）和（Sdept，Sno）的连接。

（5）正确。

（6）正确。

（7）正确。

（8）错误。如果 B 是候选码，则不一定 R.C->R.A。

例如 Student(Sno, Sname, Sage)，有(Sno, Sage) -> Sname, Sno-> Sname，但是没有 age -> Sname。

8．证明：

（1）如果 R 是 BCNF 关系模式，则 R 是 3NF 关系模式，反之则不然。

（2）如果 R 是 3NF 关系模式，则 R 一定是 2NF 关系模式。

答:

（1）反证法。设 R 不是 3NF，则必然存在这样的码 X，属性组 Y 和非主属性 Z （Z 不是 Y 的子集），使得 X→Y （Y 不 ->X），Y→Z，这样 Y→Z 函数依赖的决定因素 Y 不包含候选键，这与 BCNF 范式的定义相矛盾，所以如果 R ∈ BCNF，则 R 也是 3NF 。

反之则不然，课件中给出的例子 STC（S，T，C），有函数依赖：（S，C）→ T，（S，T）→ C，T → C。STC 属于 3NF，但是由于 T→C，所以 STC 不属于 BCNF。

（2）反证法。若关系 R 属于 3NF，却不属于 2NF，则不妨设 R 中非主属性 A 部分依赖于主码 K，且存在 K'⊆K，使得 K'→A。由于 K'⊆K，可知有平凡函数依赖 K→K',且 K'不→K。于是有 A 对 K 传递函数依赖,所以 R 不应该属于 3NF，与已知矛盾。

第七章 数据库设计

1．试述数据库设计过程。

答：数据库设计过程的六个阶段包括：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中形成的数据库模式。

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，如 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，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。

3．需求分析阶段的设计目标是什么？调查的内容是什么？

答：

（1）需求分析阶段的设计目标是：对现实世界要处理的对象（组织、部门、企业）等进行详细的调查，通过对原系统（手工系统或计算机系统）的了解，明确用户的各种需求，收集支持新系统的基础数据并对其进行处理，在此基础上确定新系统的功能。

（2）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。

4 ．数据字典的内容和作用是什么?

答:

(1) 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。

(2) 数据字典的作用：数据字典是各类数据描述的集合，是进行详细的数据收集和数据分析所获得的主要结果。数据字典在需求分析阶段建立，是下一步进行概念设计的基础。

5 ．什么是数据库的概念结构？试述其特点和设计策略。

答:

(1) 概念结构，即概念模型，是将需求分析得到的用户需求抽象成的信息结构。

(2) 概念结构的主要特点是： 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。

(3) 概念结构的设计策略通常有四种：① 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。

6．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

答：

（1）实体：客观存在并可相互区别的事物。

（2）实体型：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。

（3）实体集：同一类型实体的集合。

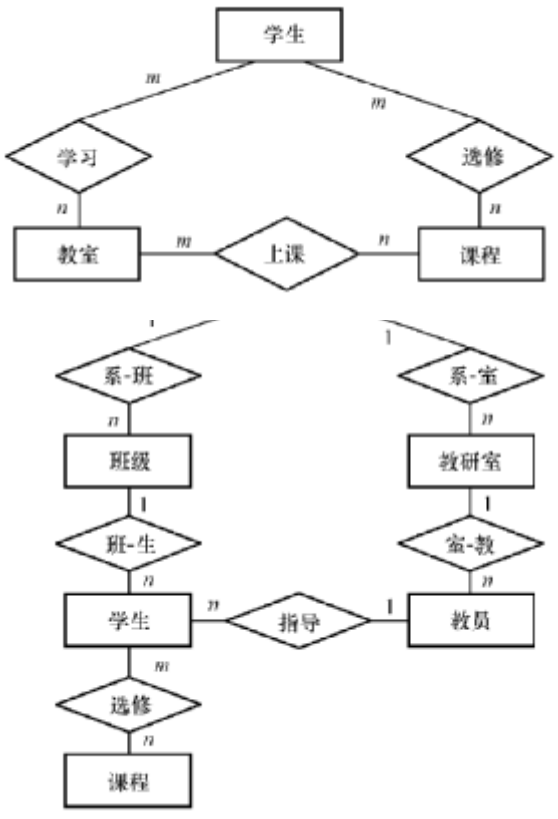
（4）属性：关系表中的一列即为一个属性。

（5）码：关系表中的某个属性组，它可以唯一确定一个元组。

（6）实体-联系图：用来表示实体型、属性和联系的方法，也成 E-R 图。

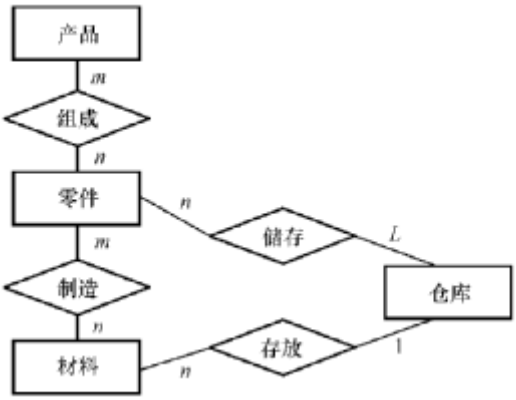
7．学校中有若干系，每个系有若干班级和教研室，每个教研室有若干教员，其中有的教授和副教授每人各带若干研究生；每个班有若干学生，每个学生选修若干课程，每门课可由若干学生选修。请用 E-R 图画出此学校的概念模型。

答：



8．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-R 图画出此工厂产品、零件、材料、仓库的概念模型。

答:



9．什么是数据库的逻辑结构设计？试述其设计步骤。

答:

（1）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。

（2）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。

10．试把习题 7 和习题 8 中的 E-R 图转换为关系模型

答：（1）习题 7

关系模型是（下划线属性是主码）

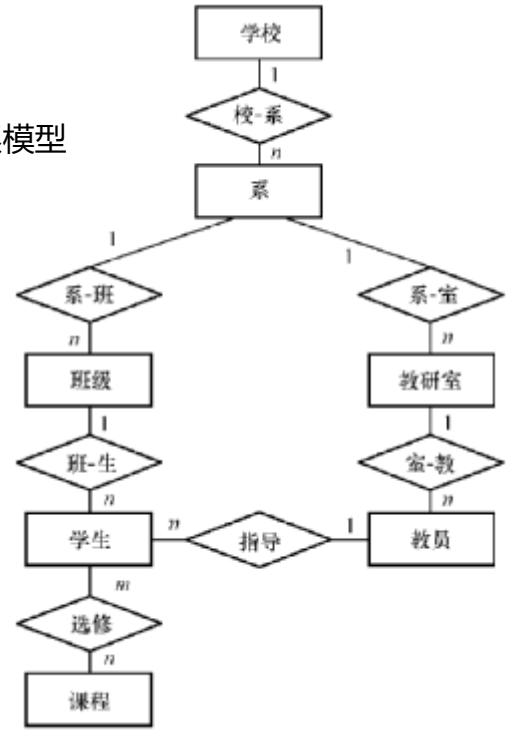
系(系编号，系名，学校名)

班级(班级编号，班级名，系编号)

教研室(教研室编号，教研室名，系编号)

学生(学号，姓名，学历

班级编号，导师教工号)



课程(课程编号，课程名)

教师(教工号，姓名，职称，教研室编号)

选课(学号，课程编号，成绩)

（2）习题 8

关系模型是（下划线属性是主码）

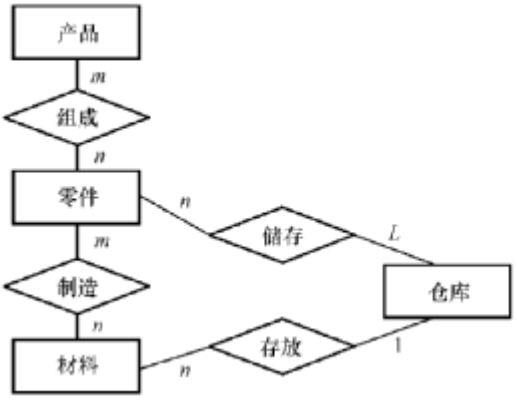
产品(产品号，产品名，仓库号)

零件(零件号，零件名)

原材料(原材料号，原材料名，类别

仓库号，存放量)

仓库(仓库号，仓库名)



产品组成(产品号，零件号，使用零件量)

零件制造(零件号，原材料号，使用原材料量)

零件存储(零件号，仓库号，存储量)

11．试用规范化理论中有关范式的概念分析习题 7 设计的关系模型中各个关系模式的候选码，它们属于第几范式？会产生什么更新异常？

答： 习题 10 答案中所设计的习题 7 的关系模型的候选码用下划线画出，都属于 BCNF，没有什么更新异常。

12．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。

13．试述数据库物理设计的内容和步骤。

答：

（1）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。

（2）数据库的物理设计步骤通常分为两步：① 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。

14．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：

（1）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。

（2）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。

15．什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：

（1）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。数据库的再组织是不修改数据库的模式和内模式的。

（2）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。

（3）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，由于记录不断增加、删除和修改，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。

第九章 关系系统和查询优化

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:

（1）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。

（2）可能性：① 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。

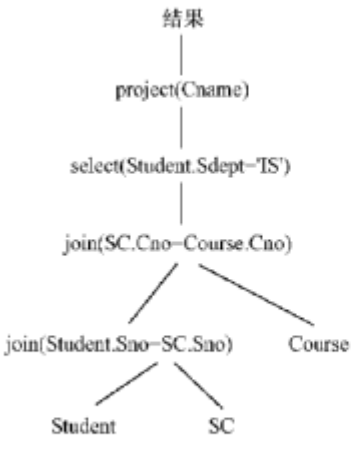
3．对学生-课程数据库，查询信息系学生选修了的所有课程名称:

```
SELECT Cname
FROM Student, Course, SC
WHERE Student.Sno = SC.Sno AND SC.Cno=Course.Cno AND
        Student.Sdept = 'IS'
```

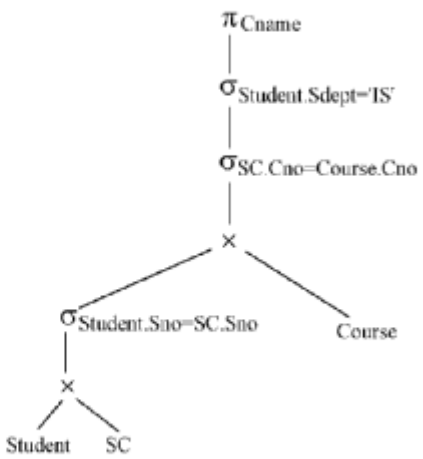
此查询要求信息系学生选修了的所有课程名称。

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。

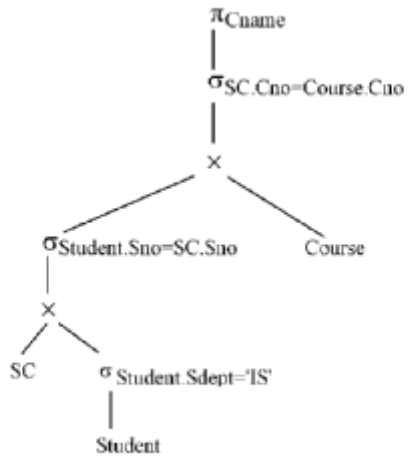
答:



最初的语法树



关系代数语法树



优化后的语法树

5．对于题 4 中的数据模式，

Teacher(Tno, Tname, Tage,Tsex)

Department(Dno, Dname, Tno)

Work(Tno, Dno, Year, Salary)

有如下查询:

```
select Tname from teacher, department, work
where teacher.tno=work.nto and department.dno=work.dno and
      department.dname='计算机系' and salary > 5000
```

画出语法树以及用关系代数表示的语法树，并对关系代数语法树进行优化，画出优化后的语法树。

答： 参考题目 3 的答案。这里略。

6．试述关系数据库管理系统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。

7．试述关系数据库管理系统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。

第十章 数据库恢复技术

1．试述事务的概念及事务的四个特性。恢复技术能保证事务的哪些特性?

答:

（1）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。

（2）事务具有 4 个特性：① 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。② 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。③ 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。④ 持续性：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。

（3）恢复技术能保证事务的原子性、一致性和持续性。

2．为什么事务非正常结束时会影响数据库数据的正确性，请举例说明之。

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 A 移到仓库 B 存放。则可以定义一个事务 T , T 包括两个操作：$Q_A = Q_A - Q$ 和 $Q_B = Q_B + Q$。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．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。

4．考虑下图所示的日志记录:

序号	日志
1	T_1:开始
2	T_1:写 A，A=10
3	T_2:开始
4	T_2:写 B，B=9
5	T_1:写 C，C=11
6	T_1:提交
7	T_2:写 C，C=13
8	T_3:开始
9	T_3:写 A，A=8
10	T_2:回滚
11	T_3:写 B，B=7
12	T_4:开始
13	T_3:提交
14	T_4:写 C，C=12

（1）如果系统故障发生在 14 之后，说明哪些事务需要重做，哪些事务需要回滚。

（2）如果系统故障发生在 10 之后，说明哪些事务需要重做，哪些事务需要回滚。

（3）如果系统故障发生在 9 之后，说明哪些事务需要重做，哪些事务需要回滚。

（4）如果系统故障发生在 7 之后，说明哪些事务需要重做，哪些事务需要回滚。

答:

（1）如果系统故障发生在 14 之后，T_1和 T_3需要重做，T_4需要回滚。

（2）如果系统故障发生在 10 之后，T_1需要重做，T_3需要回滚。

（3）如果系统故障发生在 9 之后，T_1需要重做，T_2和 T_3需要回滚。

（4）如果系统故障发生在 7 之后，T_1需要重做，T_2需要回滚。

5．考虑题 4 所示的日志记录，假设开始时 A、B、C 的值都是 0：

（1） 如果系统故障发生在 14 之后，写出系统恢复后 A、B、C 的值;

（2） 如果系统故障发生在 12 之后，写出系统恢复后 A、B、C 的值;

（3） 如果系统故障发生在 10 之后，写出系统恢复后 A、B、C 的值;

（4） 如果系统故障发生在 9 之后，写出系统恢复后 A、B、C 的值;

（5） 如果系统故障发生在 7 之后，写出系统恢复后 A、B、C 的值;

（6） 如果系统故障发生在 5 之后，写出系统恢复后 A、B、C 的值;

答：。

（1）如果系统故障发生在 14 之后，A=8，B=7，C=11；

（2）如果系统故障发生在 12 之后，A=10，B=0，C=11；

（3）如果系统故障发生在 10 之后，A=10，B=0，C=11；

（4）如果系统故障发生在 9 之后，A=10，B=0，C=11；

（5）如果系统故障发生在 7 之后，A=10，B=0，C=11；

（6）如果系统故障发生在 5 之后，A=0，B=0，C=0；

6．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，如何进行系统故障的恢复，以及如何进行介质故障的恢复。）

答:

（1）事务故障的恢复方法是：由恢复子系统应利用日志文件撤消此事务已对数据库进行的修改，事务故障的恢复由 DBMS 自动完成，对用户是透明的。

事务故障的恢复步骤是：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。

（2）系统故障的恢复方法是：撤消故障发生时未完成的事务，以及重做已完成的事务。系统故障的恢复由系统在重新启动时自动完成，不需要用户干预。

系统故障的恢复步骤是：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 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”写入数据库。③ 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”写入数据库。

（3）介质故障的恢复方法是：恢复方法是重装数据库，然后重做已完成的事务。

介质故障的恢复步骤是：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。

7．什么是检查点记录？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。

8．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：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

检查点开始扫描日志，这就缩短了扫描日志的时间。检查点之前已经提交的事务，在进行恢复时就没有必要再 REDO 处理，采用检查点技术做到了这一点。

例子略。

9．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，建立两个事务队列：UNDO-LIST 和 REDO-LIST。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，直到日志文件结束：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。

（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。

10．什么是数据库镜像？它有什么用途？

答：

（1）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。

（2）数据库镜像的用途有：① 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。② 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。

第十一章 并发控制

1. 在数据库中为什么要并发控制？并发控制技术能保证事务的哪些特性?

答:

（1）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）并发控制技术能够对并发操作进行正确调度，保证事务的隔离性和一致性。

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:

（1）并发操作带来的数据不一致性包括三类：① 丢失修改：两个事务 T_1 和 T_2 读入同一数据并修改，T_2 提交的结果破坏了（覆盖了）T_1 提交的结果，导致 T_1 的修改被丢失。② 不可重复读：事务 T_1 读取数据后，事务 T_2 执行更新操作，使 T_1 无法再现前一次读取结果。③ 读“脏”数据：指事务 T_1 修改某一数据，并将其写回磁盘，事务 T_2 读取同一数据后，T_1 由于某种原因被撤销，这时 T_1 已修改过的数据恢复原值，T_2 读到的数据就与数据库中的数据不一致，是不正确的数据，又称为“脏”数据。

（2）避免不一致性的方法是并发控制，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:

（1）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。

（2）基本的封锁类型有两种：① 排它锁（X 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 。② 共享锁 （S 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。如下图所示，事务 T_1 在对 A 进行修改之前先对 A 执行 Xlock (A)，即对 A 加 X 锁。这样，当 T_2 请求对 A 加 X 锁时就被拒绝，T_2 只能等待 T_1 释放 A 上的 X 锁后才能获得对 A 的 X 锁，这时它读到的 A 是 T_1 更新后的值，再按此新的 A 值进行运算，这样就不会丢失 T_1 对 A 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”数据等数据不一致性。

5．什么是活锁？试述活锁的产生原因和解决方法。

答：

（1）如下图所示，如果事务 T_1 封锁了数据 R，事务 T_2 又请求封锁 R，于是 T_2 等待。T_3 也请求封锁 R，当 T_1 释放了 R 上的封锁之后系统首先批准了 T_3 的请求，T_2 仍然等待。然后 T_4 又请求封锁 R，当 T_3 释放了 R 上的封锁之后系统又批准了 T_4 的请求 ... ... T_2 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。

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	•	•

（2）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。

（3）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。

6．什么是死锁？请给出预防死锁的若干方法。

答：

（1）如下图所示，如果事务 T_1 封锁了数据 R_1，T_2 封锁了数据 R_2，然后 T_1 又请求封锁 R_2，因 T_2 已封锁了 R_2，于是 T_1 等待 T_2 释放 R_2 上的锁。接着 T_2 又申请封锁 R_1，因 T_1 已封锁了 R_1，T_2 也只能等待 T_1 释放 R_1 上的锁。这样就出现了 T_1 在等待 T_2，而 T_2 又在等待 T_1 的局面， T_1 和 T_2 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

（2）预防死锁的方法有：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：顺序封锁法是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。

7．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:

（1）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。超时法是指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。

（2）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。

8．什么样的并发调度是正确的调度?

答：正确的调度是可串行化的调度，即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。

9．设 T_1，T_2，T_3 是如下的 3 个事务，设 A 的初值为 0：

T_1：A : = A + 2 ;

T_2：A : = A * 2 ;

T_3：A : = A **2 ; (A <-A^2)

设 A 的初值为 0 。

（1）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。

（2）请给出一个可串行化的调度，并给出执行结果。

（3）请给出一个非串行化的调度，并给出执行结果。

（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。

（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:

（1）A 的最终结果可能有 2、4、8、16。因为串行执行次序有 $T_1T_2T_3$、$T_1T_3T_2$、$T_2T_1T_3$、$T_2T_3T_1$、$T_3T_1T_2$、$T_3T_2T_1$。对应的执行结果是 16、8、4、2、4、2。

（2）给出一个可串行化的调度，执行结果 A 为 16。如下图所示:

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

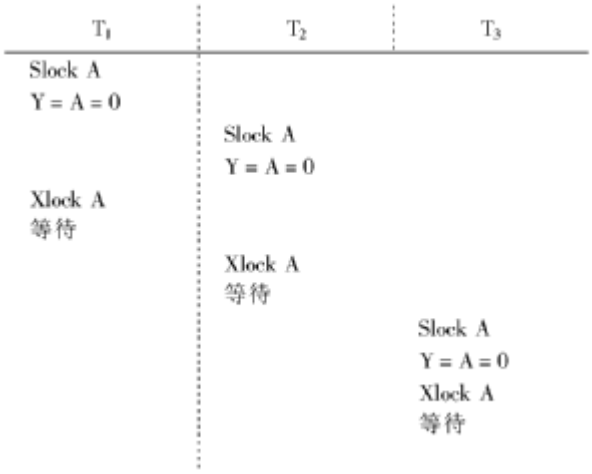
（3）给出一个非串行化的调度，执行结果 A 为 0。如下图所示：

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	

（4）给出一个不产生死锁的可串行化调度，如下图所示：

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）给出一个产生死锁的调度，如下图所示：



10．现有 3 个事务的一个调度 $r_3(B)\ r_1(A)\ w_3(B)\ r_2(B)\ r_2(A)\ w_2(B)\ r_1(B)\ w_1(A)$，该调度是冲突可串行化的调度吗？为什么？。

答：该调度是冲突可串行化的调度。

Sc0 = $r_3(B)$ **$r_1(A)\ w_3(B)$** $r_2(B)\ r_2(A)\ w_2(B)\ r_1(B)\ w_1(A)$

交换 $r_1(A)$ 和 $w_3(B)$，得到

Sc1 = $r_3(B)\ w_3(B)$ **$r_1(A)\ r_2(B)\ r_2(A)\ w_2(B)$** $r_1(B)\ w_1(A)$

再交换 $r_1(A)$ 和 $r_2(B)\ r_2(A)\ w_2(B)$，得到

Sc2 = $r_3(B)\ w_3(B)\ r_2(B)\ r_2(A)\ w_2(B)\ r_1(A)\ r_1(B)\ w_1(A)$

由于 Sc2 是串行的，而且两次交换都是基于不冲突操作的，所以 Sc0 = $r_3(B)$ **$r_1(A)\ w_3(B)$** $r_2(B)\ r_2(A)\ w_2(B)\ r_1(B)\ w_1(A)$是冲突可串行化的调度。

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：以两个并发事务 T_1 和 T_2 为例，存在多个并发事务的情形可以照此类推。根据可串行化的定义可知，事务不可串行化只能发生在下列两种情况中：

（1）事务 T_1 写某个数据对象 A，T_2 读或写 A；

（2）事务 T_1 读或写个数据对象 A，T_2 写 A。

则称 A 为潜在的冲突对象，设事务 T_1 和 T_2 所访问的潜在冲突的公共对象为 $\{A_1, A_2, \ldots\ldots A_n\}$。

为了不失一般性，假设这组潜在冲突对象中 X={A_1, A_2, ……A_i }均符合情况（1），Y={A_{i+1}, A_2, ……A_n }符合情况（2）。

对于任意 x，

操作 a：T_1 需要 XLock x；

操作 b：T_2 需要 SLock x 或 XLock x。

1）如果操作 a 先执行，则事务 T_1 获得锁，事务 T_2 等待。

由于遵守两段锁协议，事务 T_1 在在成功获得 X 和 Y 中国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，T_2 已获得 w 的锁，则会出现死锁。否则，T_1 在对 X 和 Y 中的对象全部处理完毕后，T_2 才能执行。这相当于按照 T_1 和 T_2 的顺序串行执行。根据可串行化定义，T_1 和 T_2 的调度是可串行化的。

2）如果操作 b 先执行，情况与 1）对称。

因此，若并发事务遵守两段锁协议，则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。

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：举例如下图所示：

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．考虑如下调度，说明这些调度集合之间的包含关系。

（1） 正确的调度。

（2） 可串行化的调度。

（3） 遵循两阶段封锁（2PL）的调度。

（4） 串行调度。

答：遵循两阶段封锁（2PL）的调度 $\subseteq$ 可串行化的调度 $\subseteq$ 串行调度 $\subseteq$ 正确的调度。

14．考虑 T_1 和 T_2 两个事务。

T_1: R(A); R(B); B=A+B; W(B)

T_2: R(B); R(A); A=A+B; W(A)

（1） 改写 T_1 和 T_2，增加加锁操作和解锁操作，并要求遵循两阶段封锁协议。

（2） 说明 T_1 和 T_2 的执行是否会引起死锁，给出 T_1 和 T_2 的一个调度并说明之。

答：（1）

T_1: Slock A; R(A); Slock B; R(B); B=A+B; Xlock B; W(B); Unlock A; Unlock B

T_2: Slock B; R(B); Slock A; R(A); A=A+B; Xlock A; W(A) Unlock B; Unlock A

（2）会发生死锁。

T_1:		T_2:
	Slock A;	
	R(A);	
		Slock B;
		R(B)
	Slock B;	
	R(B);	
		Slock A;
		R(A);
	B=A+B;	

	A=A+B;
Xlock B;	
	Xlock B;
Wait	Wait
.....	.....

15．为什么要引进意向锁？意向锁的含义是什么？

答：

（1）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。

（2）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。

16．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：

（1）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。

（2）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。

（3）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