第 1 章

1 ．试述数据、数据库、数据库系统、数据库管理系统的概念。
答：

（ 1 ）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。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。

DB DBMS DBA ⇒ DBS

（ 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。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。

（ 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。

2. 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。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。

3 ．定义并解释以下术语：模式、外模式、内模式、 DDL 、 DML

答：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。

4 ．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。

都要改上应用程序不需要改变

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。

5 ．试述数据库系统的组成。

答：

DB DBMS DBS

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。

第 2 章 关系数据库

主码属性非空

外码要么空要么有对应的主码在原表中

1. 试述关系模型的完整性规则。

答：关系模型的完整性规则有实体完整性规、参照完整性规则和用户定义完整性规则。

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

参照完整性规则是指若属性(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。

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。包括某属性取唯一值，某一属性值应满足一定的函数关系，某一属性的取值范围等等。

2. 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：外部码属性不一定是主码属性，所以外部码属性的值可能为空。当外部码属性不是主码属性时，可以取空值，否则不能取空值。

3.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。

4.解释候选码和主码的关系。候选码可有多个，主码只有一个，其中可有多个属性
答：
若关系中的某一属性组的值能唯一标识一个元组，而其子集不能，该属性组称为候选码；如果一个关系有多个候选码，可以选择其中一个为主码。候选码可以有多个，主码是其中的一个。

第 3 章

"虚表"
1 ．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。

2 ．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。

3 ．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.

4 ．哪类视图是可以更新的？哪类视图是不可更新的？

答：一般而言，单表、原始属性构成的视图可以更新(可转换成对基本表的操作)。对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时，不能更新视图。一般当视图由两个以上的基本表导出，或者视图的字段来自字段表达式或常数，或者视图包含 GROUP BY,DISTINCT、聚集函数和嵌套查询时，该视图是不可以更新的。

5. SQL 一般包括哪些操作？

答：①定义和修改删除关系模式、②定义和删除视图，③插入数据，④建立数据库

⑤对数据库中的数据进行查询和更新

⑥数据库的重构和维护

⑦数据库安全性、完整性控制及事务控制

⑧嵌入式 SQL 和动态 SQL 的定义

6. 解释相关子查询和不相关子查询

答：在嵌套查询中，如果子查询的查询条件不依赖于父查询，称为不相关子查询；如果子查询的查询条件依赖于父查询，称为相关子查询

第 4 章

1 ．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。

2 ．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。

3.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。

4. 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。

5 . 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 ）。

6 . 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。

7 . 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敏感的信息，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

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？ 试举例说明

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。比如：（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）实体完整性约束；(6）参照完整性约束等等

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 ．关系数据库系统中，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

1. 理解并给出函数依赖、传递依赖的定义

答: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。）

在 R(U)中，如果 X→Y，(Y⊄ X) ,Y↛X Y→Z， 则称 Z 对 X 传递函数依赖。

2. 理解并给出部分函数依赖、完全函数依赖的定义

答: 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

3. 理解并给出 1 NF 、ZNF 、3NF 、BNF 的定义

答: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)。

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）。

若关系模式 R∈2NF， 且关系模式 R 中的所有非主属性对任何候选关键字都不存在传递信赖，则称关系 R 是属于第三范式的，则称 R<U，F>∈3NF。

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。

4. 给出分解为 3NF 并保持依赖和无损分解的算法

答:（1）F 的最小覆盖仍记为 F，找出不在 F 中出现的 R 中属性，组成一个关系模式，将其从 R 中分离，其余属性仍组成的关系模式仍记为 R，属性集记为 U

（2）若最小覆盖 F 中有一函数依赖含有 R 中的所有属性，则把整个 R 作为输出，算法结束，否则进入（3）

（3）对于 F 中所有函数依赖 X →A，将 XA 作为σ中的一个关系模式输出，但如果 F 中有函数依赖 X →A1，X →A2，…，X →An，可将 XA1…An 作为一个关系模式输出

（4）设 X 为 R 的一个键，则 加上 {R（X）}，消除重复。

第 7 章

1. 数据库设计过程有哪几个阶段。

答:数据库设计过程通常分为六个阶段：（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．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。

3．什么是数据库概念结构设计？设计好E-R图

答：将需求分析得到的用户需求抽象为信息结构即概念模型的过程就是概念结构设计。概念结构是各种数据模型的共同基础，它比数据模型更独立于机器、更抽象，从而更加稳定，概念结构设计是整个数据库设计的关键。

4．什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等模型；（2）将转换来的关系等模型向特定 DBMS 支持下的数据模型转换；（3）对数据模型进行优化。

5．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，使数据库设计工作有了严格的理论基础。

6．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：（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。

7．什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。数据库应用环境也常常发生变化，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 8 章

1. 什么是嵌入式 SQL?
答：嵌入式 SQL 是将 SQL 语句嵌入程序设计语言中，被嵌入的程序设计语言可以是 C、C++、Java 等语言

2. 什么是动态 SQL? run时确定
答：动态 SQL 是应用程序中的 SQL 命令在运行时才能确定。

3. 什么是静态 SQL compile时确定
答：静态 SQL 的 Sql 语句在写应用程序时指明，应用程序中的 SQL 命令在编译时已经确定.

4. 什么是 ODBC?
答：ODBC（开放的数据库互连，Open DataBase Connectivity）是 Microsoft 公司开发的一套开放的数据库系统应用程序接口规范，它为应用程序提供了一套高层调用接口规范和基于动态链接库的运行支撑环境。它是数据库服务器的一个标准协议，它向访问数据库的应用程序提供了一种通用的语言，应用程序开发人员不必知道所连接的数据库类型，就可以用标准的 SQL 语言访问数据库中的数据。

5. 简述存储过程的优点
答：
a) 存储过程经编译和优化后存储在数据库服务器中，运行效率高
b) 降低客户机和服务器之间的通信量
c) 有利于集中控制，方便维护

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

答：关系数据库系统的查询优化既是 RDBMS 实现的关键技术又是关系数据库系统的优点所在。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”做得更好。

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

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：Q1 = Q1 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 ．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。

7．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。

8．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。

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”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”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。

9．什么是检查点记录？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。

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。 不用重新 redo 已保存的事务

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最后一个检查点时刻 Tc 开始扫描日志，这就缩短了扫描日志的时间。

11．什么是数据库镜像？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。

第 11 章　并发控制

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’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 一 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。

4．什么是活锁？什么是死锁？

答：

如果事务 T1 封锁了数据 R ，事务T2 又请求封锁 R ，于是 T2 等待。T3 也请求封锁 R ，当 T1 释放了 R 上的封锁之后系统首先批准了 T3 的请求，T2 仍然等待。然后 T4 又请求封锁 R ，当 T3 释放了 R 上的封锁之后系统又批准了 T4 的请求 … … T2 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 T1 封锁了数据 R1 ，T2 封锁了数据 R2，然后 T1 又请求封锁 R2，因 T2 已封锁了 R2，于是 T1 等待 T2 释放 R2 上的锁。接着 T2 又申请封锁 R1 ，因 T1 已封锁了 R1 ，T2 也只能等待 T1 释放 R1 上的锁。这样就出现了 T1 在等待 T2，而 T2 又在等待 T1 的局面， T1 和 T2 两个事务永远不能结束，形成死锁。

5 ．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。

6 ．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。

7 ．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。

8 ．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：

8．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

9．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。

10．试述常用的意向锁： IS 锁、 ix 锁、 SIX 锁。

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