

1. 从程序和数据之间的关系分析文件系统和数据库系统之间的区别和联系。

区别：文件系统：用文件将数据长期保存在外存上，程序和数据有一定的联系，用操作系统中的存取方法对数据进行管理，实现以文件为单位的数据共享。数据库系统：用数据库统一存储数据，程序和数据分离，用 **DBMS** 统一管理和控制数据，实现以记录和字段为单位的数据共享。

联系：均为数据组织的管理技术；均由数据管理软件管理数据，程序与数据之间用存取方法进行转换；数据库系统是在文件系统的基础上发展而来的。

2. 什么是数据库？

数据库是长期存储在计算机内、有组织的、可共享的数据集合。数据库是按某种数据模型进行组织的、存放在外存储器上，且可被多个用户同时使用。因此，数据库具有较小的冗余度，较高的数据独立性和易扩展性。

3. 什么是数据冗余？数据库系统与文件系统相比怎样减少冗余？

在文件管理系统中，数据被组织在一个个独立的数据文件中，每个文件都有完整的体系结构，对数据的操作是按文件名访问的。数据文件之间没有联系，数据文件是面向应用程序的。每个应用都拥有并使用自己的数据文件，各数据文件中难免有许多数据相互重复，数据的冗余度比较大。

数据库系统以数据库方式管理大量共享的数据。数据库系统由许多单独文件组成，文件内部具有完整的结构，但它更注重文件之间的联系。数据库系统中的数据具有共享性。数据库系统是面向整个系统的数据共享而建立的，各个应用的数据集中存储，共同使用，数据库文件之间联系密切，因而尽可能地避免了数据的重复存储，减少和控制了数据的冗余。

4. 使用数据库系统有什么好处？

查询迅速、准确，而且可以节约大量纸面文件；数据结构化，并由 **DBMS** 统一管理；数据冗余度小；具有较高的数据独立性；数据的共享性好；**DBMS** 还提供了数据的控制功能。

5. 什么是数据库的数据独立性？

数据独立性表示应用程序与数据库中存储的数据不存在依赖关系，包括逻辑数据独立性和物理数据独立性。

逻辑数据独立性是指局部逻辑数据结构（外视图即用户的逻辑文件）与全局逻辑数据结构（概念视图）之间的独立性。当数据库的全局逻辑数据结构（概念视图）发生变化（数据定义的修改、数据之间联系的变更或增加新的数据类型等）时，它不影响某些局部的逻辑结构的性质，应用程序不必修改。

物理数据独立性是指数据的存储结构与存取方法（内视图）改变时，对数据库的全局逻辑结构（概念视图）和应用程序不必作修改的一种特性，也就是说，数据库数据的存储结构与存取方法独立。

6. 什么是数据库管理系统？

DBMS 是操纵和管理数据库的一组软件，它是数据库系统的重要组成部分。不同的数据库系统都配有各自的 **DBMS**，而不同的 **DBMS** 各支持一种数据库模型，虽然它们的功能强弱不同，但大多数 **DBMS** 的构成相同，功能相似。

一般说来，**DBMS** 具有定义、建立、维护和使用数据库的功能，它通常由三部分构成：

数据描述语言及其翻译程序、数据操纵语言及其处理程序和数据库管理的例行程序。

1)数据描述语言（DDL）;2)数据操纵语言（DML）;3)数据库管理的例行程序;

7. 数据库管理系统有哪些功能?

DBMS 是位于操作系统与用户之间的一个数据管理软件, 它主要功能包括以下几个方面:

数据定义功能 DBMS 提供数据描述语言 (DDL), 用户可通过它来定义数据。

数据操纵功能 DBMS 还提供数据操纵语言 (DML), 实现对 DB 的基本操作: 查询、插入、删除和修改。

数据库的运行管理 这是 DBMS 运行时的核心部分, 包括开发控制, 安全性检查, 完整性约束条件的检查和执行, DB 的内容维护等。

数据库的建立和维护功能 它包括 DB 初始数据的输入及转换, DB 的转储与恢复, DB 的重组功能和性能的监视与分析功能等。

8. DBA 的职责是什么?

决定 DB 中的信息内容和结构; 决定 DB 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; 监控数据库的使用和运行。

9. 什么是数据字典? 数据字典包含哪些基本内容?

数据字典是数据库系统中各种描述信息和控制信息的集合, 它是数据库设计与管理的有力工具。基本内容有: 数据项; 组项; 记录; 文件; 外模式; 概念模式; 内模式; 外模式/概念模式映象; 概念模式/内模式映象; 用户管理信息; 数据库控制信息。

10. 叙述数据字典的主要任务和作用?

数据字典的任务就是管理有关数据的信息, 所以又称为“数据库的数据库”。它的任务主要有:

1) 描述数据库系统的所有对象, 并确定其属性。2) 描述数据库系统对象之间的各种交叉联系。3) 登记所有对象的完整性及安全性限制等。4) 对数据字典本身的维护、保护、查询与输出。

数据字典的主要作用是: 1) 供数据库管理系统快速查找有关对象的信息。2) 供数据库管理员查询, 以掌握整个系统的运行情况。3) 支持数据库设计与系统分析。

11. 叙述模型、模式和具体值三者之间的联系和区别。

数据模型是用来表示信息世界中的实体及其联系在数据世界中的抽象描述, 它描述的是数据的逻辑结构。模式的主体就是数据库的数据模型。数据模型与模式都属于型的范畴。所谓型, 是指只包含属性的名称, 不包含属性的值, 而所谓值, 是型的具体实例值, 即赋了值的型 (在 Visual Foxpro 中, 型就是数据库表的结构, 而值就是数据库表中存储的记录)。

12. 什么是层次模型?

在数据库中, 把满足以下两个条件的基本层次联系的集合称为“层次模型”:

1) 有且仅有一个结点无双亲, 这个结点称为“根结点”; 2) 其他结点有且仅有一个双亲。

13. 什么是网状模型?

在数据库中, 把满足以下两个条件的基本层次结构的集合称为“网状模型”:

1) 允许一个以上结点无双亲; 2) 一个结点可以有多个双亲。

14. 简要叙述关系数据库的优点?

关系数据库是以关系模型作为数据的组织方式, 关系模型是建立在严格的数学概念基础上的, 关系数据库的主要优点是概念简单清晰, 用户不需了解复杂的存取路径, 不需说明“怎么干”, 只需说明“干什么”, 易懂易学。

15. 层次模型、网状模型和关系模型等三种基本数据模型是根据什么来划分的?

数据模型是现实世界中各种实体之间存在着联系的客观反映, 是用记录描述实体信息的基本结构, 它要求实体和记录——一对应; 同一记录类型描述同一类实体且必须是同质的。目前应用在数据库技术中的模型有关系、网状和层次模型, 它们是依据描述实体与实体之间联系的不同方式来划分的; 用二维表格来表示实体和实体之间联系的模型叫做关系模型; 用图结构来表示实体和实体之间联系的模型叫做网状模型; 用树结构来表示实体和实体之间联系的模型叫做层次模型。

16. 层次模型、网状模型和关系模型这三种基本数据模型各有哪些优缺点?

层次模型的优点是: 结构清晰, 表示各结点之间的联系简单; 容易表示如“家族关系”等现实世界的层次结构的事物及其之间的联系。

缺点是: 不能表示两个以上实体型之间的复杂联系和实体型之间的多对多联系;

严格的层次顺序使数据插入和删除操作变得复杂, 如父结点的删除导致子结点的删除。

网状模型的优点是: 能够表示实体之间的多种复杂联系。

缺点是: 网状模型比较复杂, 需要程序员熟悉数据库的逻辑结构; 在重新组织数据库时容易失去数据独立性。

关系模型的优点是: 使用表的概念, 简单直观; 直接表示实体之间的多对多联系; 具有更好的数据独立性; 具有坚实的理论基础。

缺点是: 关系模型的联结等操作开销较大, 需要较高性能计算机的支持。

17. 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?

实体完整性规则: 若属性 A 是基本关系 R 的主属性, 则属性 A 不能取空值。
参照完整性规则: 若属性 (或属性组) F 是基本关系 R 的外码, 它与基本关系 S 的主码 K_s 相对应 (基本关系 R 和 S 不一定是不同的关系), 则对于 R 中每个元组在 F 上的值必须为: 或者取空值 (F 的每个属性值均为空值); 或者等于 S 中某个元组的主码值。外部属性可以为空, 因为外部属性对其基本关系本身是非码, 不是标识元组的属性值, 故可以为空。当此属性值尚未确定时, 此外部属性为空。

18. 名词解释:

最小关系系统: 仅支持关系数据结构和三种关系操作的关系系统。

关系上完备的系统: 支持关系数据结构和所有的关系代数操作的关系系统。

全关系型的关系系统: 支持关系模型的所有特征的关系系统。

19. 叙述数据库实现完整性检查的方法?

要实现数据库完整性的检查, 系统要提供定义完整性约束条件的机制, 和提供检查是否违背完整性约束条件的方法。当 **DBMS** 发现用户的操作违背了完整性约束条件时, 就采取一定的控制。具体的控制是随系统而定的。

20. 什么是事务?

事务是 **DBMS** 的基本工作单位。它是用户定义的一组逻辑一致的程序序列。它是一个不可分开的工作单位, 其中包含的所有操作, 要么都执行, 要么都不执行。

数据库是一个共享资源, 它允许多个用户程序并发地存取数据库中的数据。但是, 如果系统对并发执行的操作不加以控制, 就会存取和存储不正确的数据, 破坏数据库的完整性。并发控制的主要方法就个采用封锁机制。封锁就是一个事务在对某个数据对象进行操作前, 先向系统发出请求, 对其加锁。

21. 事务中的提交和回滚是什么意思?

事务中的提交（COMMIT）是提交事务的所有操作。具体说就是将事务中所有对数据库的更新写回到磁盘上的物理数据库中去，事务正常结束。

事务中的回滚（ROLLBACK）是数据库滚回到事务开始时的状态。具体地说就是，在事务运行的过程中发生了某种故障，事务不能继续执行，系统将事务中对数据库的所有已完成的更新操作全部撤消，使数据库回滚到事务开始时的状态。

22. 在数据库中为什么要有并发控制？

数据库是一个共享资源，它允许多个用户同时并行地存取数据。若系统对并行操作不加控制，就会存取和存储不正确的数据，破坏数据库的完整性（或称为一致性）。并发控制的目的是，就是要以正确的方式调度并发操作，避免造成各种不一致性，使一个事务的执行不受另一个事务的干扰。

23. 并发操作会产生几种不一致情况？用什么方法避免各种不一致的情况？

操作可能会产生丢失修改、不能重复读和读“脏”数据这三种不一致情况。采用封锁机制来进行并发控制，可避免各种不一致情况。

24. 叙述数据库中数据的一致性级别。

在数据库的并发控制中，数据一致性级别的概念有三个：丢失修改，不能重复读和读“脏”数据。

丢失修改是指 T1 和 T2 先后读取了同一个数据，T1 把数据修改了并写回库中，T2 也将读取的数据修改了并写回库中。这样，T2 提交的结果导致 T1 对数据库的修改丢失了。

不能重复读是指 U 读取 A、B 两个数据并进行了运算之后，T2 读了其中的数据 B，把它修改后写回数据库，最后当 T1 为了对读取值进行校对而再重读 B 时，读的是 T2 修改后的值，而不是 T1 开始读的值。

读“脏”数据是指 T1 修改了某一数据，并将其写回库中，T2 读了这修改后的数据，而事务 T1 由于某种原因撤消了，被它修改的数据恢复了原来的值，这时，T2 读的数据就与库中的数据不一致了，即 T2 读了不正确的数据，也称为 T2 读了“脏”数据。

25. 叙述封锁的概念。

封锁就是事务 T 在对某个数据对象操作之前，先向系统发出请求，对其加锁，这样事务 T 就对这个数据对象有了一定的控制，其他事务就不能更新此数据，直到事务 T 释放它的锁为止。

26. 叙述数据库中死锁产生的原因和解决死锁的方法。

死锁产生的原因：封锁可以引起死锁。比如事务 T1 封锁了数据 A，事务 T2 封锁了数据 B。T1 又申请封锁数据 B，但因 B 被 T2 封锁，所以 T1 只能等待。T2 又申请封锁数据 A，但 A 已被 T1 封锁，所以也处于等待状态。这样，T1 和 T2 处于相互等待状态而均不能结束，这就形成了死锁。解决死锁的常用方法有如下三种：

1) 要求每个事务一次就要将它所需要的数据全部加锁。2) 预先规定一个封锁顺序，所有的事务都要按这个顺序实行封锁。

3) 允许死锁发生，当死锁发生时，系统就选择一个处理死锁代价小的事务，将其撤消，释放此事务持有的所有的锁，使其他事务能继续运行下去。

27. 基本的封锁类型有几种？试叙述它们的含义。

基本的封锁的类型有排它锁（“X”锁）和共享锁（“S”锁）两种。

若事务 T 对数据 A 加上 X 锁，则只允许事务 T 读取和修改数据 A，其他事务

都不能再对 A 加任何类型的锁, 直到 T 释放 A 上的锁。

若事务 T 对数据 A 加上 S 锁, 则其他事务只有再对 A 加 S 锁, 而不能加 X 锁, 直到 T 释放 A 上的锁。

28. 数据库中为什么要有恢复子系统? 它的功能是什么?

由于硬件的故障、系统软件和应用软件的错误、操作的失误以及恶意的破坏都是不可避免的, 这些故障, 轻则会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, 使数据库中的数据部分丢失或全部丢失。为了保证各种故障发生后, 数据库中的数据都能从错误状态恢复到某种逻辑一致状态, DBMS 中的恢复子系统是必不可少的。

恢复子系统的功能就是利用冗余数据, 再根据故障的类型采取相应的恢复措施, 把数据库恢复到故障前的某一时刻的一致性状态。

29. 什么是数据库的转储? 转储的意义是什么?

对数据库制作后备副本的过程称为转储。转储是由 DBA 定期将数据库复制到另一个存档用的存储器上, 转储的意义就在于, 当数据库遭到破坏时, 可将后备副本重新装入, 从而把数据库恢复到某个一致性状态。

30. 为什么要设立日志文件?

设立日志文件的目的是, 为了记录对数据库中数据的每一次更新操作。从而 DBMS 可以根据日志文件进行事务故障的恢复和系统故障的恢复, 并可结合后备副本进行介质故障的恢复。

31. 怎样进行系统故障的恢复?

当系统故障发生时, 造成数据库处于不一致状态的原因有两个, 一个是一些未完成事务对数据库的部分更新已写入数据库中, 另一个是一些已提交事务对数据库的更新还留在内存缓冲区, 来不及写回数据库。因此, 基本的恢复分为以下两步:

1) 根据日志文件建立重做队列和撤消队列

从头扫描日志文件, 找出故障前已提交的事务记录, 将其事务标识放入重做队列; 还要找出故障前尚未完成的事务, 将该事务放入撤消队列。

2) 对重做队列中的事务进行重做处理, 对撤消队列中的事务进行撤消处理

对于重做处理 (REDO), 正向扫描日志文件, 再根据重做队列, 将已完成的事务的所有操作重新执行; 对于撤消处理 (UNDO), 反向扫描日志文件, 再根据撤消队列, 将未完成的事务的更新操作执行逆操作

32. 怎样进行介质故障的恢复?

在发生介质故障时, 磁盘上的物理数据库被破坏, 这时的恢复操作分为以下几步:

1) 重装转储后备副本, 使数据库恢复到转储时的一致状态;

2) 从故障开始, 反向阅读日志文件, 找出已提交事务标记作重做队列;

3) 从起始点开始正向阅读日志文件, 根据重做队列的记录, 重做所有已完成的事务, 将数据库恢复至故障前某一时刻的一致状态。

33. 数据库在运行过程中可能产生的故障有哪几类?

数据库在运行过程中可能产生的故障有如下几类:

1) 事务故障 事务在运行过程中由于种种原因, 如输入数据的错误, 运算溢出, 违反了某些完整性限制, 某些应用程序的错误, 以及并行事务发生死锁等, 使事务未能运行到正常终止点之前就被撤消了, 这种情况称为“事务故障”。

2) 系统故障 系统故障是指系统在运行过程中, 由于某种原因, 如 OS 和 DBMS

代码错误, 操作员操作失误, 特定类型的硬件错误 (如 CPU 故障), 突然停电等造成系统停止运行, 致使事务在执行过程中以非控方式终止。这时, 内存中的信息丢失, 而存储在外存储上的数据未受影响, 这种情况称为“系统故障”。

3) 介质故障 系统在运行过程中, 由于某种硬件故障, 如磁盘损坏, 磁头碰撞, 或由于 OS 的某种潜在的错误, 瞬时强磁场干扰, 使存储在外存上的数据部分损失或全部损失, 称之为“介质故障”。

4) 计算机病毒 计算机病毒是一种人为的故障和破坏, 它是一种计算机程序。通过读写染有病毒的计算机系统上的程序和数据, 这些病毒可以迅速繁殖和传播, 危害计算机系统和数据库。

34. 叙述数据库的并发控制。

数据库是一个共享资源, 它允许多个用户程序并行地存取数据库中的数据, 但是, 如果系统对并行执行的操作不加以控制就会存取和存储不正确的数据, 破坏数据库的完整性。

并发控制的主要方法是采用封锁机制。封锁是事务 T 在对某个数据对象操作之前, 先向系统发出请求对其加锁。基本的封锁类型有两种: 排它锁 (X 锁) 和共享锁 (S 锁)。所谓 X 锁, 是事务 T 对数据 A 加上 X 锁时, 只允许事务 T 读取和修改数据 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。所谓 S 锁, 是事务 T 对数据 A 加上 S 锁时, 其他事务只能再对数据 A 加 S 锁, 而不能加 X 锁, 直到 T 释放 A 上的 S 锁。

35. 在数据库的保护中有数据的安全性、完整性, 并发控制和数据库的恢复。其中的安全性和完整性有何区别?

数据的完整性是指为了防止数据中存在不合语义的数据, 防止错误的输入和输出所造成的无效操作和错误结果; 数据的安全性是指保护数据库不被恶意的破坏和非法的存取。

36. 设有三个事务 T1、T2 和 T3, 所包含的动作为:

T1: $A = A + 2$

T2: $A = A * 2$

T3: $A = A ** 2$

假定这三个事务允许并行操作, 试讨论它们可能实施的调度

它们有以下 6 种可能正确的调度:

编号 调度 A 初值为 1 时, A 最后结果

1 T1-T2-T3 36

2 T1-T3-T2 18

3 T2-T3-T1 6

4 T2-T1-T3 16

5 T3-T1-T2 6

6 T3-T2-T1 4

37. 什么是数据库的安全性、完整性?

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。

数据库的完整性是指数据的正确性和相容性。数据库是否具备完整性关系到数据库系统能否真实地反映现实世界, 因此维护数据库的完整性是非常重要的。

38. Oracle 系统提供哪些安全性措施?

Oracle 的安全性措施主要有三个方面: 一是用户标识和鉴定; 二是授权和检

查机制；三是审计技术；

除此之外 Oracle 还允许用户通过出发器灵活定义自己的安全性措施。

39. DBMS 的完整性控制机制应具有哪三个方面的功能？

- (1) 定义功能，提供定义完整性约束条件的机制。
- (2) 检查功能，检查用户发出的操作请求是否违背了完整性约束条件。
- (3) 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。

40. 参照完整性要考虑哪几个问题？

(1) 码能否接受空值问题在实现参照完整性时，系统除了应该提供定义外码的机制，还应提供定义外码列是否允许空值的机制。

(2) 在被参照关系中删除元组的问题

一般地，当删除被参照关系的某个元组，而参照关系存在若干元组，其外码值与被参照关系删除元组的主码值相同，这时可有三种不同的策略：

① 级联删除 (CASCADES)

② 受限删除 (RESTRICTED) 仅当参照关系中没有任何元组的外码值与被参照关系中要删除元组的主码值相同时，系统才执行删除操作，否则拒绝此删除操作。

③ 置空值删除 (NULLIFIES) 删除被参照关系的元组，并将参照关系中相应元组的外码值置空值。

(3) 在参照关系中插入元组时的问题一般地，当参照关系插入某个元组，而被参照关系不存在相应的元组，其主码值与参照关系插入元组的外码值相同，这时可有以下策略：① 受限插入 ② 递归插入

(4) 修改关系中主码的问题 ① 不允许修改主码 ② 允许修改主码

在有些 RDBMS 中，允许修改关系主码，但必须保证主码的唯一性和非空，否则拒绝修改。

41. 试述事务的概念及事务的四个特性。

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

事务具有四个特性：原子性 (Atomicity)、一致性 (Consistency)、隔离性 (Isolation) 和持续性 (Durability)。这个四个特性也简称为 ACID 特性。

(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。因此当数据库只包含成功事务提交的结果时，就说数据库处于一致性状态。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，系统将事务中对数据库的所有已完成的操作全部撤消，滚回到事务开始时的一致状态。

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

42. 数据库恢复的基本技术有哪些？

恢复机制涉及的两个关键问题是：第一，如何建立冗余数据；第二，如何利用这些冗余数据实施数据库恢复。

建立冗余数据最常用的技术是数据转储和登录日志文件。通常在一个数据库系统中, 这两种方法是一起使用的。

所谓转储即 DBA 定期地将整个数据库复制到磁带或另一个磁盘上保存起来的过程。这些备用的数据文本称为后备副本或后援副本。

日志文件是用来记录事务对数据库的更新操作的文件。不同数据库系统采用的日志文件格式并不完全一样。概括起来日志文件主要有两种格式: 以记录为单位的日志文件和以数据块为单位的日志文件。

日志文件在数据库恢复中起着非常重要的作用。可以用来进行事务故障恢复和系统故障恢复, 并协助后备副本进行介质故障恢复。

43. 试述封锁、S 锁和 X 锁的概念。

封锁是实现并发控制的一个非常重要的技术。

封锁: 所谓封锁就是事务 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, 其它的事务不能更新此数据对象。

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

排它锁: 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, 其它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。这就保证了其它事务在 T 释放 A 上的锁之前不能再读取和修改 A。

共享锁: 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, 其它事务只能再对 A 加 S 锁, 而不能加 X 锁, 直到 T 释放 A 上的 S 锁。这就保证了其它事务可以读 A, 但在 T 释放 A 上的 S 锁之前不能对 A 做任何修改。

44. 试述活锁和死锁的概念。如何预防死锁?

活锁: 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。

避免活锁的简单方法是采用先来先服务的策略。

死锁: 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。

死锁的预防:

在数据库中, 产生死锁的原因是两个或多个事务都已封锁了一些数据对象, 然后又都请求对已被其他事务封锁的数据对象加锁, 从而出现死等待。防止死锁的发生其实就是要破坏产生死锁的条件。预防死锁通常有两种方法:

① 一次封锁法 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。一次封锁法虽然可以有效地防止死锁的发生, 但也存在问题, 一次就将以后要用到的全部数据加锁, 势必扩大了封锁的范围, 从而降低了系统的并发度。

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。顺序封锁法可以有效地防止死锁, 但也同样存在问题。

事务的封锁请求可以随着事务的执行而动态地决定, 很难事先确定每一个事务要封锁哪些对象, 因此也就很难按规定的顺序去施加封锁。

可见, 在操作系统中广为采用的预防死锁的策略并不很适合数据库的特点, 因此 DBMS 在解决死锁的问题上普遍采用的是诊断并解除死锁的方法。

45. 什么是封锁协议? 不同级别的封锁协议的主要区别是什么?

在运用 X 锁和 S 锁这两种基本封锁, 对数据对象加锁时, 还需要约定一些规则, 例如应何时申请 X 锁或 S 锁、持锁时间、何时释放等。我们称这些规则为封锁协议 (Locking Protocol)。对封锁方式规定不同的规则, 就形成了各种不同的封锁协议。有三级封锁协议。对并发操作的不正确调度可能会带来丢失修改、不可重复读和读“脏”数据等不一致性问题, 三级封锁协议分别在不同程度上解决了这一问题。为并发操作的正确调度提供一定的保证。不同级别的封锁协议达到的系统一致性级别是不同的。

一级封锁协议是: 事务 T 在修改数据 R 之前必须先对其加 X 锁, 直到事务结束才释放。事务结束包括正常结束 (COMMIT) 和非正常结束 (ROLLBACK)。

一级封锁协议可防止丢失修改, 并保证事务 T 是可恢复的。

在一级封锁协议中, 如果仅仅是读数据不对其进行修改, 是不需要加锁的, 所以它不能保证可重复读和不读“脏”数据。

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

二级封锁协议除防止了丢失修改, 还可进一步防止读“脏”数据。

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

三级封锁协议除防止了丢失修改和不读“脏”数据外, 还进一步防止了不可重复读。

46. 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?

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。

丢失修改是指两个事务 T1 和 T2 读入同一数据并修改, T2 提交的结果破坏了 T1 提交的结果, 导致 T1 的修改被丢失。

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。

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 由于某种原因被撤消, 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

产生上述三类数据不一致性的主要原因是并发操作破坏了事务的隔离性。并发控制就是要用正确的方式调度并发操作, 使一个用户事务的执行不受其它事务的干扰, 从而避免造成数据的不一致性。并发控制的主要技术是封锁 (Locking)。

一级封锁协议可防止丢失修改, 并保证事务 T 是可恢复的。

二级封锁协议除防止了丢失修改, 还可进一步防止读“脏”数据。

三级封锁协议除防止了丢失修改和不读“脏”数据外, 还进一步防止了不可重复读。47. 试述两段锁协议的概念。

两段锁协议: 所谓两段锁协议是指所有事务必须分两个阶段对数据项加锁和解

锁:

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

所谓“两段”锁的含义是, 事务分为两个阶段, 第一阶段是获得封锁, 也称为扩展阶段。在这阶段, 事务可以申请获得任何数据项上的任何类型的锁, 但是不能释放任何锁。第二阶段是释放封锁, 也称为收缩阶段。在这阶段, 事务可以释放任何数据项上的任何类型的锁, 但是不能再申请任何锁。

另外要注意两段锁协议和防止死锁的一次封锁法的异同之处。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, 因此一次封锁法遵守两段锁协议; 但是两段锁协议并不要求事务必须一次将所有要使用的数据全部加锁, 因此遵守两段锁协议的事务可能死锁。

1. 数据模型三要素:数据结构、数据操作、完整性约束(实体完整性、参照完整性、用户定义完整性)
2. 数据库三级模式结构: 模式、子模式、内模式
两级映射: 映像 1: 子模式/模式映像 映像 2: 模式/内模式映像
目的: 为实现数据独立性, 数据与程序间互不依赖
3. 候选键: 某一属性的值可唯一地标识每个元组, 其值对不同元组不同
4. 关系模式不合理带来的问题: 数据冗余度高; 数据修改复杂; 插入异常; 删除异常
5. 触发器是一类靠事件驱动的特殊过程, 包括事件、条件、动作
6. 数据库恢复: 数据转储、登记日志文件、检查点恢复技术、数据库镜像(介质故障)
7. 并发带来的不一致: 丢失修改、不可重复读、读脏数据
- 8.