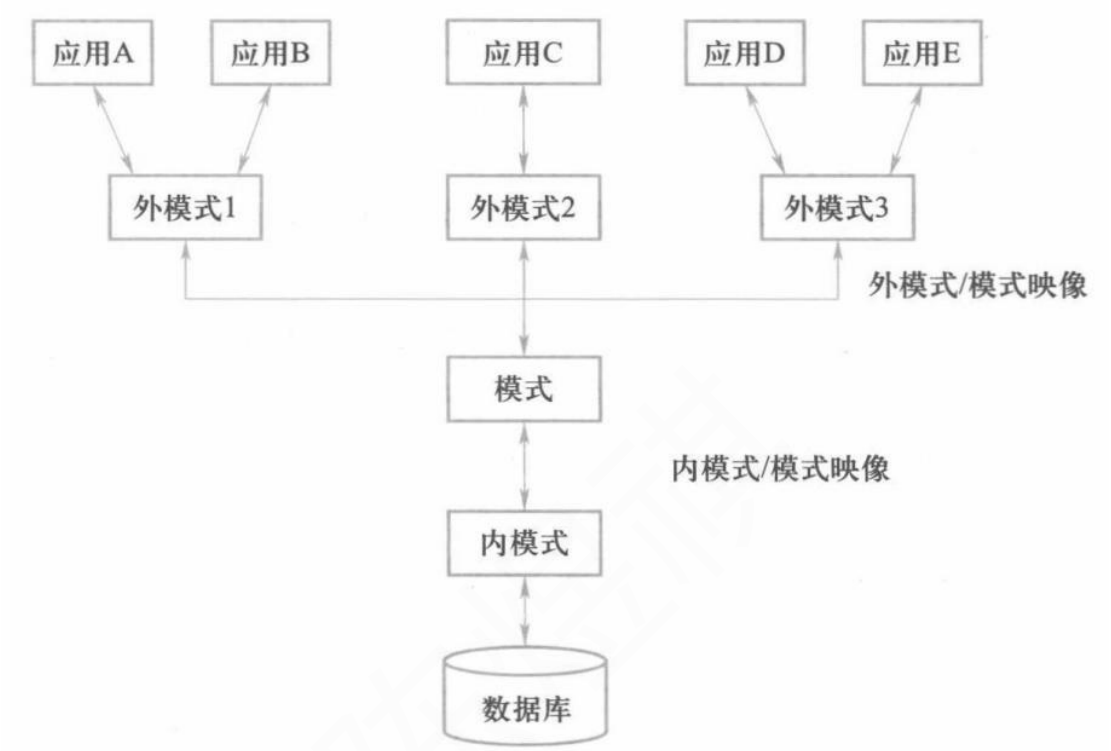
第 1 章　绪论

1. 四个基本概念
   (1) 数据：描述事物的符号记录称为数据，是数据库中存储的基本对象
   (2) 数据库：长期存储在计算机内有组织、可共享的大量数据的集合
   (3) 数据库管理系统：是位于用户与操作系统之间的数据管理软件，是计算机的基础软件
   (4) 数据库系统：是指由数据库、数据库管理系统(及其应用开发工具)、应用系统和数据库管理员(DBA)组成的存储、管理、处理和维护数据的系统
2. 数据模型三要素
   (1) 数据结构：描述数据库的组成对象以及对象之间的联系。
   (2) 数据操纵：指对数据库中各种对象(型)的实例(值)允许执行的操作的集合，包括操作及有关的操作规则。
   (3) 完整性约束：是一组完整性规则，完整性规则是给定的数据模型中的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。
3. 数据库系统的三级模式结构
   (1) 外模式：是模式的子集，是数据库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。一个数据库可以有多个外模式。
   (2) 模式：是所有用户的公共数据视图，是对数据库中全体数据的逻辑结构和特征的描述。一个数据库只有一个模式。
   (3) 内模式：是对数据物理结构和存储方式的描述，是数据在数据库内部的组织方式。一个数据库只有一个内模式。
   (4) 三层模式结合两级映象保证了数据库系统中的数据具有较强的逻辑独立性和物理独立性。
4. 两级映像与数据独立性
   (1) 逻辑独立性：当模式(全体逻辑结构)改变时，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。因而建立在外模式上的应用程序也不必修改，保证了数据与程序间的逻辑独立性。

(2) 物理独立性：当数据库的存储结构改变时，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，因而应用程序也不必改变，保证了数据与程序间的物理独立性。应用程序与数据存储互相独立，数据的物理存储方式可以灵活调整，而不会影响应用程序的正常运行。

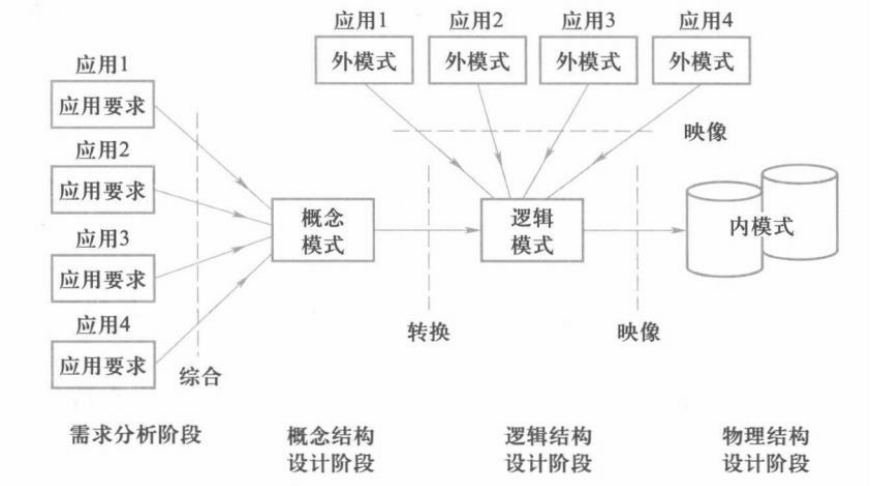


第 4 章　数据库安全性

1. 定义：是指保护数据库，以防不合法使用所造成的数据泄露、篡改或破坏。
2. 不安全因素
   (1) 非授权用户对数据库的恶意存取和破坏
   (2) 数据库中重要或敏感的数据被泄露
   (3) 安全环境的脆弱性
3. 安全技术：用户身份鉴别、存取控制、视图机制、审计、数据加密
4. 存取控制
   (1) 定义用户权限和合法权限检查机制一起组成了数据库管理系统的存取控制子系统。
   (2) 在自主存取控制方法中，用户对于不同的数据库对象有不同的存取权限，不同的用户对同一对象也有不同的权限，而且用户还可将其拥有的存取权限转授给其他用户。因此自主存取控制非常灵活。主题资源的拥有者可以自主地选择特定的用户或用户组授予资源的某些权限。
   (3) 在强制存取控制方法中，每一个数据库对象被标以一定的密级，每一个用户也被授予某一个级别的许可证。对于任意一个对象，只有具有合法许可证的用户才可以存取。强制存取控制因此相对比较严格。

第 7 章　数据库设计

1. 定义：数据库设计是指对于一个给定的应用环境，设计数据库的外模式、模式和内模式，并据此建立数据库及其应用系统，使之能够有效地存储和管理数据，满足各种用户的应用需求，包括信息管理需求和数据操作需求。数据库设计的目标是为用户和各种应用系统提供信息基础设施和高效的运行环境。
2. 数据库设计的基本步骤
   (1) 需求分析：准确了解与分析用户的应用需求，包括“数据”和“处理”，建立数据字典。
   (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数据库管理系统的概念模型。形成 E-R 图。
   (3) 逻辑结构设计：是按某种转换规则将概念结构设计转换为某个数据库管理系统所支持的逻辑数据模型如关系模型，并对其进行优化。
   (4) 物理结构设计：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：运用数据库管理系统提供的数据库语言及高级语言，根据逻辑结构设计和物理结构设计的结果创建数据库。编写与调试应用程序，组织数据入库并进行试运行。
   (6) 数据库运行与维护：将数据库应用系统投入正式运行，并在运行过程中不断对其进行维护（评估、调整与修改）。



第 11 章　数据库恢复技术

1. 事务
   (1) 定义：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) 原子性 A：指事务中包括的操作要么都做，要么都不做。
   (3) 一致性 C：事务执行的结果必须是使数据库从一个一致性状态转为另一个一致性状态。
   (4) 隔离性 I：指一个事务的执行不能被其他事务干扰。
   (5) 持续性 D：一个事务一旦提交，它对数据库中数据的改变就是永久的
   (6) ACID 特性遭到破坏的因素
      ① 多个事务并行运行时，不同事务的操作交叉执行。
      ② 事务在运行过程中被强行停止。
2. 故障的种类
   (1) 事务内部故障：事务没有达到预期的终点。恢复：事务撤销 UNDO
   (2) 系统故障（软故障）：是指造成系统停止运转的任何事件如断电，使得系统要重新启动。恢复：撤销未完成的事务 UNDO、重做已提交的事务 REDO
   (3) 介质故障（硬故障）：指外存故障，直接破坏了数据库。恢复：利用后备副本重装数据库，然后重做已提交的事务。
   (4) 计算机病毒带来的故障
   (5) 恢复的基本原理：冗余。撤销都是逆向扫描，重做都是正向扫描。
3. 数据转储
   (1) 定义：数据库管理员定期将整个数据库复制到磁带、磁盘或其他存储介质上保存的过程。备用的数据称为后备副本。
   (2) 静态转储：是在系统中无运行事务时进行的转储操作。静态转储操作开始的时刻数据库处于一致性状态，在转储期间不允许对数据库的任何存取、修改活动。如此得到的一定是一个数据一致性的副本。
      ① 优点：实现简单

② 缺点：降低了数据库的可用性

(3) 动态转储：转储期间允许对数据库进行存取或修改，即转储和用户事务可以并发执行。

① 优点：不用等待事务结束、不会影响新事务运行

② 缺点：不能保证副本中的数据正确有效

(4) 海量转储：每次转储全部数据库（更方便）

(5) 增量转储：只转储上次转储后更新过的数据（更实用）

转储方式	转储状态	
	动态转储	静态转储
海量转储	动态海量转储	静态海量转储
增量转储	动态增量转储	静态增量转储

4. 登记日志文件

(1) 定义：是用来记录事务对数据库的更新操作的文件。

(2) 两种格式：以记录为单位、以数据块为单位。

(3) 作用

① 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。

② 在动态转储方式中必须建立日志文件，后备副本和日志文件结合起来才能有效地恢复数据库。

③ 在静态转储方式中也可以建立日志文件，利用日志文件把已完成的事务进行重做处理。

(4) 登记日志文件遵循的原则

① 登记的次序严格按并发事务执行的时间次序。

② 必须先写日志文件，后写数据库。

第 12 章　并发控制

1. 数据不一致
   (1) 丢失修改：两个事务 T1 和 T2 读入同一数据，各自进行修改，T2 提交的结果破坏了 T1 提交的结果，导致 T1 的修改被丢失。
   (2) 脏读：因事务撤销读到了不正确的数据。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，这时被 T1 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。
   (3) 不可重复读：指事务 T1 读取数据后，事务 T2 执行更新操作，当事务 T1 再次读该数据时，得到与前一次不同的值。
   (4) 幻读
   (5) 主要原因：并发操作破坏了事务的隔离性。
   (6) 主要技术：封锁、时间戳、乐观方法、多版本并发控制
2. 事务的隔离级别
   (1) 读未提交
   (2) 读已提交
   (3) 可重复读
   (4) 可串行化
3. 封锁
   (1) 定义：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。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，其他事务不能更新此数据对象。
   (2) 排它锁（X 锁、写锁）：只允许事务 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 上的锁为止。
   (3) 共享锁（S 锁、读锁）：事务 T 可以读但不能修改 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 锁为止。
4. 封锁协议
   (1) 一级封锁协议：指事务在修改数据之前必须先对其加 X 锁，直到事务结束才释放。防止了丢失修改。

(2) 二级封锁协议：一级+事务在读取数据之前必须先对其加 S 锁，读完后即可释放 S 锁。进一步防止了脏读。

(3) 三级封锁协议：一级+加事务在读取数据之前必须先对其加 S 锁，直到事务结束才释放。进一步防止了不可重复读。

封锁协议	X 锁		S 锁		一致性保证			隔离性级别保证
	操作结束释放	事务结束释放	操作结束释放	事务结束释放	不丢失修改	不脏读	可重复读	
一级封锁协议		√			√			读未提交
二级封锁协议		√	√		√	√		读已提交
三级封锁协议		√		√	√	√	√	可串行化

5. 活锁

(1) 描述：存在可能永远等待的事务

(2) 解决：采用先来先服务的策略

6. 死锁

(1) 产生原因：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。

(2) 预防死锁

① 一次封锁法：要求每个事务必须一次将所有要使用的数据全部加锁，否则就不能继续执行。

② 顺序封锁法：预先对数据对象规定一个封锁顺序，所有事务都按这个顺序实施封锁。

(3) 诊断和解除死锁

① 超时法

② 事务等待图法

③ 选择一个处理死锁代价最小的事务，将其撤销，释放此事务持有的所有锁，使其他事务得以继续运行下去。

7. 两段锁协议

(1) 可串行化调度：多个事务的并发执行是正确的，当且仅当其结果与按某一次序串行地执行这些事务时的结果相同，称这种调度策略为可串行化调度。

(2) 两段锁协议：所有事务必须分两个阶段对数据项加锁和解锁。

① 扩展阶段（获得封锁）：在对任何数据进行读、写操作之前，首先要申请并获得对该数据的封锁。

② 收缩阶段（释放封锁）：在释放一个封锁之后，事务不再申请和获得任何其他封锁。

(3) 若并发执行的所有事务均遵守两段锁协议，则对这些事务的任何并发调度策略都是可串行化的。

(4) 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发生死锁。

8. 封锁粒度

(1) 定义：封锁对象的大小称为封锁粒度。封锁对象可以是逻辑单元，也可以是物理单元。封锁对象可以是这样一些逻辑单元：属性值，属性值的集合，元组，关系，索引项，整个索引直至整个数据库；也可以是这样一些物理单元：页，物理记录等。

(2) 封锁粒度与系统的并发度和并发控制的开销密切相关。封锁的粒度越大，数据库所能够封锁的数据单元就越少，并发度就越小，系统开销也越小；反之，封锁的粒度越小，并发度越高，但系统开销也就越大。

(3) 多粒度封锁协议：允许多粒度树中的每个结点被独立地加锁，对一个结点加锁，意味着这个结点的所有后裔结点也被加以同样类型的锁。

① 显式封锁：直接加到数据对象上的锁

② 隐式封锁：该数据对象没有被独立加锁，是因其上级结点加锁而使该数据对象加上了锁。

③ 需要上下搜索检查锁的冲突

(4) 意向锁：如果对一个结点加意向锁，则说明该结点的后裔结点正在被加锁；对任一结点加锁时，必须先对它的所有上层结点加意向锁。

① IS 锁：表示它的后裔结点拟(意向)加 S 锁。

② IX 锁：表示它的后裔结点拟(意向)加 X 锁。

③ SIX 锁：加 S 锁，并且加 IX 锁。

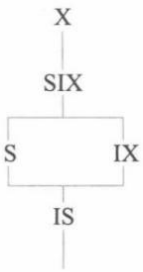
④ 次序：申请封锁时应按自上而下的次序进行，释放封锁时则应按自下而上的次序进行。

⑤ 作用：有了意向锁，数据库管理系统就无须逐个检查下一级结点的显式封锁了。具有意向锁的多粒度封锁方法提高了系统的并发度，减少了加锁和解锁的开销。

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

注：Y=Yes，表示相容的请求；N=No，表示不相容的请求

(a) 封锁类型的相容矩阵



(b) 锁的强度偏序关系

T_1	T_2		
	X	S	—
X	N	N	Y
S	N	Y	Y
—	Y	Y	Y

注：Y=Yes，相容的请求；N=No，不相容的请求。