

# 第10-11章 事务处理技术

计科2201zzy友情分享😁😁

事务处理技术主要包括数据库恢复技术和并发控制技术。

## 事务

- 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。
  - 例如在关系数据库中，一个事务可以是一条SQL语句、一组SQL语句或者是整个程序。
  - 程序和事务是两个概念，一般来说，一个程序包含多个事务。
- 特性：ACID
  - 原子性(Atomicity)：不可切割或者只完成一部分。是数据库的逻辑工作单位。
  - 一致性(Consistency)：事务的执行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。
    - 事务未完成导致数据库的修改只写入了一部分，数据库就处于不正确or不一致的状态
  - 隔离性(Isolation)：一个事务的执行不能被其他事务干扰。即一个事务的内部操作和使用的数据隔离，不会影响到其他并发的任务。
  - 持续性(Durability)：也称永久性，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会改变它的执行结果。
- 事务时恢复和并发控制的基本单位。
- 事务处理技术：维持事务的ACID特性
  - 并发控制机制：防止多个事务并行运行时，不同操作交叉执行，影响事务的原子性
  - 恢复机制：防止事务在运行过程中被强行停止后，一致性被破坏。
- 数据库系统故障的分类
  - 事务内部故障：一般指非预期的运算溢出、死锁等
  - 系统故障：系统停止运转，需要重启
  - 介质故障：存储数据库的介质受到破坏
  - 计算机病毒

## 恢复机制

- 关键问题：
  - 建立冗余数据
  - 利用这些冗余数据实施恢复
- 建立冗余数据最常用的技术：
  - 数据转储：备份整个数据库

- 登记日志文件(logging)
- 日志文件：用来记录事务对数据库的更新操作分文件
  - 作用：进行事务故障恢复和系统故障恢复，并协助后备副本(backup)进行介质故障恢复
  - 原则：一定要严格按照时间次序登记；必须先写日志后写数据库
- 介质故障的恢复流程
  - 装入最新的数据库后备副本，使数据库恢复到最近一次转储的状态
  - 装入相应的日志文件副本，重做已完成的事务

## 并发控制机制

- 事务并发进行时带来的数据不一致性
  - 丢失修改：两个事务读入同一数据并进行修改，结果其中一个事务修改的数据被覆盖导致丢失。
  - 不可重复读：在一个事务读取某一数据后，其他事务对其执行了更新/新增/删除操作，导致该事务再次读取这一数据时得到了不一致的结果。
  - 读“脏”数据：一个事务对某一数据修改并写入后，这一事务又被撤销了。但在这一过程中有别的事务读取了这一数据，最终导致读到的内容和数据库中的正确内容不一致。

## 封锁

- 实现并发控制的重要技术，即事务在进行读写操作之前先对数据对象加锁。
- **排他锁（写锁 X锁）：**
  - 排他锁是一种独占锁，它允许一个事务在持有锁期间独立地修改共享资源，并阻止其他事务同时获取排他锁或共享锁。即加上之后其它事务无法访问该数据。
  - 当一个事务持有排他锁时，其他事务无法同时持有排他锁或共享锁。这样可以确保在修改过程中不会有其他事务读取或修改相同的资源。
  - 排他锁适用于需要修改共享资源的操作，如更新、插入或删除数据。
- **共享锁（读锁 S锁）：**
  - 共享锁是一种共享访问锁，它允许多个事务同时以只读方式访问共享资源，但阻止其他事务获取排他锁。
  - 当一个事务持有共享锁时，其他事务也可以同时获取共享锁，但无法获取排他锁。这样可以确保多个事务可以同时读取相同的资源，而不会相互干扰。即加上之后其他事务也可以读，但不能修改该对象。
  - 共享锁适用于只读操作，如查询数据。
- 封锁协议：规定何时申请锁，持锁时间与释放时机。
  - 一级封锁协议：事务修改数据对象之前必须加X锁，直到事务结束才释放。
    - 无法防范：不可重复读和读脏数据
  - 二级封锁协议：在一级的基础之上新增：事务读取数据对象前必须加S锁，读完后即可释放。
    - 无法防范：读脏数据

- 三级封锁协议：S锁也必须在事务结束之后才能释放
- 活锁问题：某一事务一直被插队
- 死锁问题：两个事务分别需要使用对方已经加锁的数据对象，于是互相等待，互不相让
  - 诊断与解除
    - 超时发
    - 等待图法