

数据库第十一章

事务的执行方式

- 串行 即每个时刻只有一个事务运行
- 交叉并发方式 并行事务的并行操作轮流交叉运行
- 同时并发方式 多处理机可以运行一个事务 多处理机可以同时运行多个事务

采用并发控制的原因

- 当多个用户并发的存取数据库时 就会产生多个事务同时存取同一个数据的情况 此时若不加以控制 就会存取不正确的数据 破坏事务和数据库的一致性
- 并发控制可以防止数据的不一致性出现 数据的不一致性

产生原因 并发操作破坏了事务的隔离性

分类

- 丢失修改 两个事务T1 T2读入同一个数据并修改 T2提交的结果覆盖了T1提交的结果 导致T1的修改被丢失
- 不可重复读
 - 事务T1读取数据后 事务T2执行更新操作 使得T1无法再现前一次读取结果
 - 分类
 - 修改
 - 幻影现象
 - 删除
 - 插入
- 读脏数据
 - 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被撤销 这时T1修改过的数据恢复原值 T2读到的数据与数据库中的数据不一样
 - 脏数据即不正确的数据

并发控制的技术

封锁

- 概念 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
- 分类
 - 排他锁/写锁
 - 概念 只允许T读取和修改
 - 性质 其他任何事务都不能再对该事务加任何类型的锁
 - 共享锁/读锁
 - 概念 事务T可以读对象 但是不能修改
 - 性质 其他事务只能加共享锁
- 协议
 - 一级封锁协议
 - 概念 事务在修改数据之前必须先对其加排他锁 直至事务结束才释放
 - 作用 可以防止丢失修改
 - 正常结束COMMIT
 - 非正常结束ROLLBACK
 - 二级封锁协议
 - 概念 在一级封锁协议基础上 增加事务在读取之前必须对其加共享锁 读完后即可释放共享锁
 - 作用 可以防止丢失修改和读脏数据
 - 短锁
 - 三级封锁协议
 - 概念 在一级封锁的基础上增加事务在读数据之前必须加共享锁 直至事务结束才释放
 - 作用 可以防止丢失修改 不可重复读 读脏数据
 - 长锁
- 时间戳
- 乐观控制法
- 多版本并发控制

并发调度的可串行性

可串行化调度

- 概念 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
- 正确的调度
 - 可串行性是并发事务正确调度的准则 一个给定的并发调度 当且仅当他是可串行化的 才认为是正确的调度

冲突可串行化调度

- 冲突
 - 概念 冲突操作是指不同事务对同一数据的读写操作和写写操作
- 概念 一个调度在保证冲突操作的次序不变的情况下 通过交换两个事务不冲突操作的次序 得到另一个调度 如果这个调度是串行的就称~
- 判定 若一个调度是冲突可串行化 那么一定是可串行化调度

遵循两段锁的调度<可串行化的调度<串行调度<正确的调度

故没有丢失修改 读脏数据 不可重复读等错误

封锁的粒度

- 概念 封锁对象的大小 封锁对象可以是逻辑单元也可以是物理单元
- 关系 封锁粒度越大 并发度越小 系统开销越小
- 锁的强度
 - 概念 他对于其他锁的排斥程度
 - 一个事务在申请封锁时以强锁代替弱锁是安全的
 - 申请封锁时应该按自上而下的次序进行
 - 释放封锁时应该按自下而上的次序进行

