数据库系统概论

An Introduction to Database System

第十一章　并发控制

中国人民大学信息学院

并发控制

❖多用户数据库系统

允许多个用户同时使用的数据库系统

- 飞机定票数据库系统
- 银行数据库系统

特点：在同一时刻并发运行的事务数可达数百上千个

并发控制（续）

❖多事务执行方式

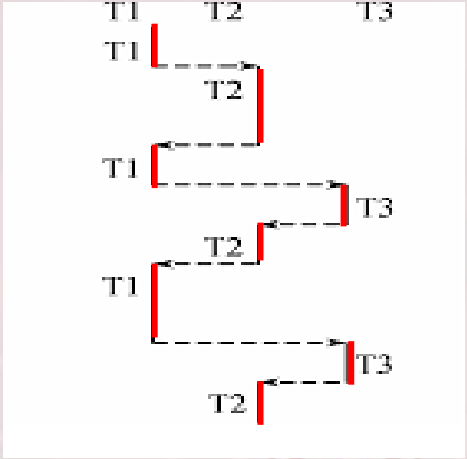
（1）事务串行执行

- 每个时刻只有一个事务运行，其他事务必须等到这个事务结束以后方能运行
- 不能充分利用系统资源，发挥数据库共享资源的特点

并发控制（续）

（2）交叉并发方式（Interleaved Concurrency）

- 在单处理机系统中，事务的并行执行是这些并行事务的并行操作轮流交叉运行
- 单处理机系统中的并行事务并没有真正地并行运行，但能够减少处理机的空闲时间，提高系统的效率



并发控制（续）

（3）同时并发方式（simultaneous concurrency）

- 多处理机系统中，每个处理机可以运行一个事务，多个处理机可以同时运行多个事务，实现多个事务真正的并行运行
- 最理想的并发方式，但受制于硬件环境
- 更复杂的并发方式机制

❖ 本章讨论的数据库系统并发控制技术是以单处理机系统为基础的

并发控制（续）

- 事务并发执行带来的问题
  - 会产生多个事务同时存取同一数据的情况
  - 可能会存取和存储不正确的数据，破坏事务隔离性和数据库的一致性
- 数据库管理系统必须提供并发控制机制
- 并发控制机制是衡量一个数据库管理系统性能的重要标志之一

T_1	T_2
① 读A=16	
②	读A=16
③ A←A-1	
写回A=15	
④	A←A-3
	写回A=13

T1的修改被T2覆盖了！

第十一章 并发控制

11.1 并发控制概述

- 事务是并发控制的基本单位
- 并发控制机制的任务
  - 对并发操作进行正确调度
  - 保证事务的隔离性
  - 保证数据库的一致性

并发控制（续）

并发操作带来数据的不一致性实例

[例11.1]飞机订票系统中的一个活动序列

① 甲售票点(事务T_1)读出某航班的机票余额A，设A=16；

② 乙售票点(事务T_2)读出同一航班的机票余额A，也为16；

③ 甲售票点卖出一张机票，修改余额A←A-1，所以A为15，把A写回数据库；

④ 乙售票点卖出三张机票，修改余额A←A-3，所以A为13，把A写回数据库

- 结果明明卖出4张机票，数据库中机票余额只减少3

T_1	T_2
① 读A=16	
②	读A=16
③ A←A-1	
写回A=15	
④	A←A-3
	写回A=13

T1的修改被T2覆盖了！

并发控制概述（续）

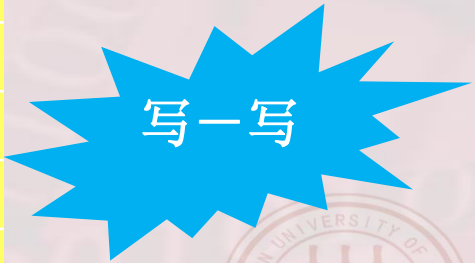
- 并发操作带来的数据不一致性
  1. 丢失修改（Lost Update）
  2. 不可重复读（Non-repeatable Read）
  3. 读“脏”数据（Dirty Read）

- 记号
  - R(x): 读数据x
  - W(x): 写数据x

1. 丢失修改

❖两个事务T_1和T_2读入同一数据并修改，T_2的提交结果破坏了T_1提交的结果，导致T_1的修改被丢失。

	T_1	T_2
①	R(A)=16	
②		R(A)=16
③	A←A-1	
	W(A)=15	
④		A←A-3
		W(A)=13



2. 不可重复读

- 不可重复读是指事务T_1读取数据后，事务T_2执行更新操作，使T_1无法再现前一次读取结果。

不可重复读（续）

❖不可重复读包括三种情况：

（1）情况1

- 事务1读取某一数据
- 事务2对其做了修改
- 当事务1再次读该数据时，得到与前一次不同的值

不可重复读（续）

例如：

T_1	T_2
① R(A)=50	
R(B)=100	
求和=150	
②	R(B)=100
	B←B*2
	W(B)=200
③ R(A)=50	
R(B)=200	
求和=250	
(验算不对)	

读一修改

- T_1读取B=100进行运算
- T_2读取同一数据B，对其进行修改后将B=200写回数据库。
- T_1为了对读取值校对重读B，B已为200，与第一次读取值不一致

不可重复读（续）

（2）情况2

- 事务T_1按一定条件从数据库中读取了某些数据记录
- 事务T_2删除了其中部分记录
- 当T_1再次按相同条件读取数据时，发现某些记录神秘地消失了。

不可重复读（续）

（3）情况3

- 事务T_1按一定条件从数据库中读取某些数据记录
- 事务T_2插入了一些记录
- 当T_1再次按相同条件读取数据时，发现多了一些记录

后两种不可重复读有时也称为幻影现象（Phantom Row）

3. 读“脏”数据

读“脏”数据是指：

- 事务T_1修改某一数据，并将其写回磁盘
- 事务T_2读取同一数据后，T_1由于某种原因被撤销
- 这时T_1已修改过的数据恢复原值，T_2读到的数据就与数据库中的数据不一致
- T_2读到的数据就为“脏”数据，即不正确的数据

读“脏”数据（续）

例如：

T_1	T_2
① R(C)=100	
C←C*2	
W(C)=200	
②	R(C)=200
③ ROLLBACK	
C恢复为100	

修改－读

- T_1将C值修改为200
- T_2读到C为200
- T_1由于某种原因撤销，其修改作废，C恢复原值100 。
- 这时T_2读到的C为200，与数据库内容不一致，就是“脏”数据

并发控制概述（续）

- 数据不一致性：由于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他事务的干扰，从而避免造成数据的不一致性
- 对数据库的应用有时允许某些不一致性，可以降低对一致性的要求以减少系统开销

并发控制概述（续）

- 并发控制的主要技术
  - 封锁(Locking)
  - 时间戳(Timestamp)
  - 乐观控制法
  - 多版本并发控制(MVCC)

小结

- 并发操作带来的数据不一致性
  1. 丢失修改 （修改-修改冲突）
  2. 不可重复读 （读-更新冲突）
     - 修改
     - 删除
     - 插入
  3. 读“脏”数据（修改-读冲突）

思考题

- 举例说明并发操作导致的幻影现象。