



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

陈红

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

■ 飞机订票数据库系统

■ 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

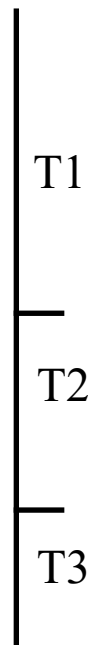
并发控制（续）



❖ 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



事务的串行执行方式

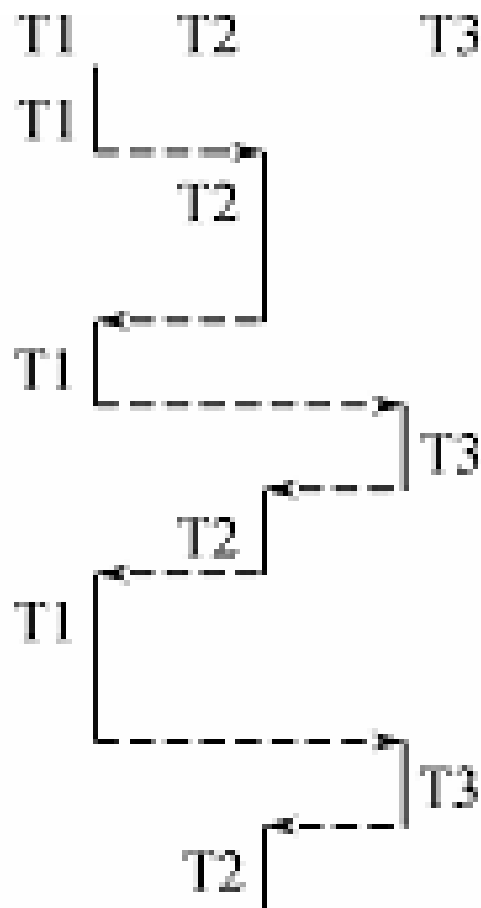
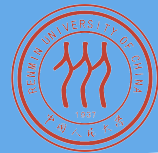
并发控制（续）



(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）



事务的交叉并发执行方式

并发控制（续）

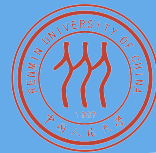


(3) 同时并发方式（ simultaneous concurrency ）

- 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

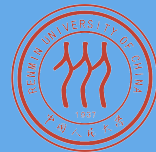
❖ 本章讨论的数据库系统并发控制技术是以单处理机系统为基础的

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
- ❖ **DBMS 必须提供并发控制机制**
- ❖ 并发控制机制是衡量一个 **DBMS** 性能的重要标志之一

第十一章 并发控制



11.1 并发控制概述

11.2 封锁

11.3 活锁和死锁

11.4 并发调度的可串行性

11.5 两段锁协议

11.6 封锁的粒度

11.7 小结

11.1 并发控制概述



❖ 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

并发控制概述（续）

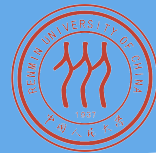


并发操作带来数据的不一致性实例

[例 1] 飞机订票系统中的一个活动序列

- ① 甲售票点（甲事务）读出某航班的机票余额 A ，设 $A=16$ ；
 - ② 乙售票点（乙事务）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 $A \leftarrow A-1$ ，所以 A 为 15，
把 A 写回数据库；
 - ④ 乙售票点也卖出一张机票，修改余额 $A \leftarrow A-1$ ，所以 A 为 15，
把 A 写回数据库
- 结果明明卖出两张机票，数据库中机票余额只减少 1

并发控制概述（续）



- ❖ 这种情况称为数据库的不一致性，是由并发操作引起的。
- ❖ 在并发操作情况下，对甲、乙两个事务的操作序列的调度是随机的。
- ❖ 若按上面的调度序列执行，甲事务的修改就被丢失。
 - 原因：第 4 步中乙事务修改 A 并写回后覆盖了甲事务的修改

并发控制概述（续）



❖ 并发操作带来的数据不一致性

- 丢失修改（Lost Update）
- 不可重复读（Non-repeatable Read）
- 读“脏”数据（Dirty Read）

❖ 记号

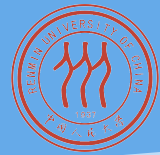
- $R(x)$: 读数据 x
- $W(x)$: 写数据 x

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。
- ❖ 上面飞机订票例子就属此类

丢失修改（续）



T_1	T_2
① $R(A)=16$	
②	$R(A)=16$
③ $A \leftarrow A-1$	
$W(A)=15$	
④	$A \leftarrow A-1$
	$W(A)=15$

丢失修改

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) 事务 T_1 读取某一数据后，事务 T_2 对其做了修改，当事务 T_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 $R(A)=50$	
$R(B)=100$	
$\square\square=150$	
②	$R(B)=100$
	$B \leftarrow B * 2$
	$(B)=200$
③ $R(A)=50$	
$R(B)=200$	
$\square=250$	
$(\square\square\square\square)$	

不可重复读

- T1 读取 $B=100$ 进行运算
- T2 读取同一数据 B ，对其进行修改后将 $B=200$ 写回数据库。
- T1 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) 事务 T1 按一定条件从数据库中读取了某些数据记录后，事务 T2 删除了其中部分记录，当 T1 再次按相同条件读取数据时，发现某些记录神秘地消失了
- (3) 事务 T1 按一定条件从数据库中读取某些数据记录后，事务 T2 插入了一些记录，当 T1 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）

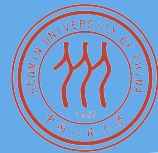
3. 读“脏”数据



读“脏”数据是指：

- 事务 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 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
- T2 读到的数据就为“脏”数据，即不正确的数据

读“脏”数据（续）



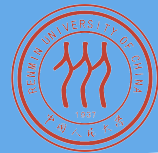
例如

T_1	T_2
① $R(C)=100$	
$C \leftarrow C * 2$	
$W(C)=200$	
②	$R(C)=200$
③ ROLLBACK	
$C \square\square\square 100$	

读“脏”数据

- 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

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，例如有些统计工作涉及数据量很大，读到一些“脏”数据对统计精度没什么影响，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

❖ 商用的 DBMS 一般都采用封锁方法



2008/10/18