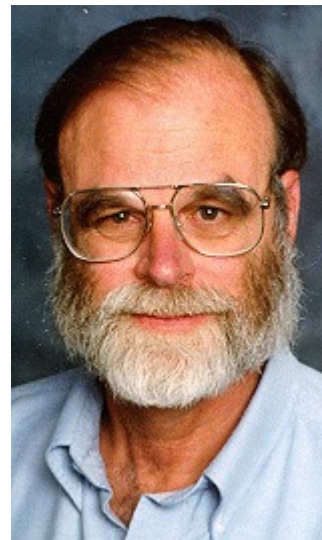


# 事务的隔离级别



胡卉芪  
华东师范大学  
数据科学与工程学院  
[hqhu@dase.ecnu.edu.cn](mailto:hqhu@dase.ecnu.edu.cn)

# 事务的概念



- ACID
  - 原子性 ( Atomicity )
    - 一个事务 ( transaction ) 要么没有开始，要么全部完成，不存在中间状态。
  - 一致性 ( Consistency )
    - 事务的执行不会破坏数据的正确性，即符合约束。
  - 隔离性 ( Isolation )
    - 多个事务不会相互破坏。
  - 持久性 ( Durability )
    - 事务一旦提交成功，对数据的修改不会丢失。

请举例  
Recall 第一节课我们讲  
的一致性是什么？

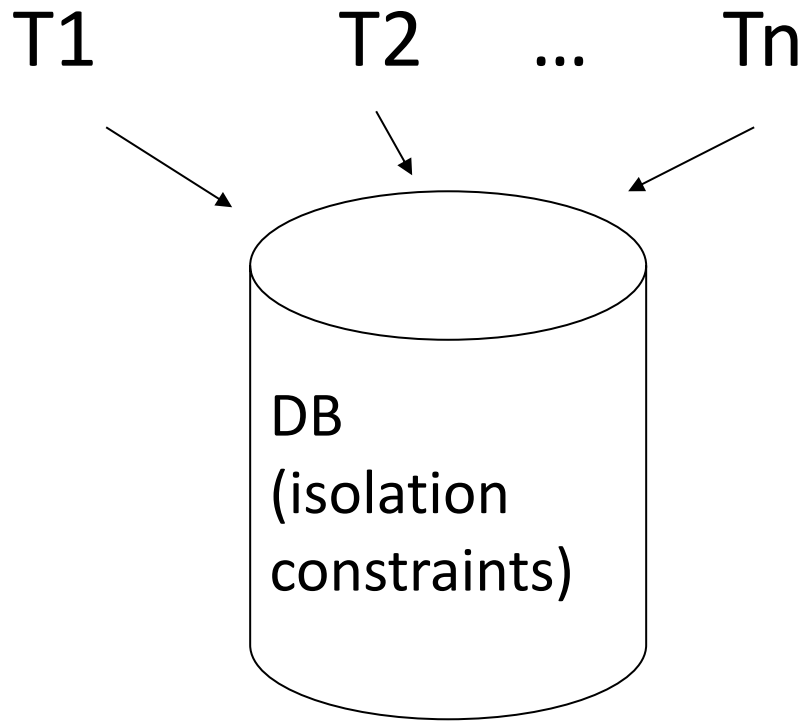
# 事务面向的负载

- 从查询到更新
- 主要特征
  - 查询：较大数据集合的计算
  - 更新：通常是小部分数据，点数据的更新

# 事务的实现

- 并发控制
  - 隔离性
- 日志
  - 原子性，持久性
- 一致性
  - 应用或其他实现保证

# 并发控制的主要内容



- 事务的隔离级别
  - 多个读写集合操作要达到一定的隔离性要求
- 并发控制算法
  - 如何实现上述要求

# 隔离级别

- 单个数据的读/写操作（加上c和a）构成的调度
  - c : commit a: abort
- $r_1(x_1), w_1(x_1), r_2(x_2), r_1(x_1), w_2(x_1), c_1, a_2$
- Critique of ANSI SQL Isolation Levels
- 假设数据只有一个版本
- 最高的隔离级别
  - 可串行化/冲突可串行化

# 异常等级

- 若不满足冲突可串行化，应用会产生何种异常？
  - 注：不是全部可能的异常，仅是应用中主要异常
  - 我们通常将异常分成4级
- A0-Dirty Write: T2 writes value modified by T1 before T1 commits
  - T2在T1提交之前修改同一数据
- A1 – Dirty Read: T2 reads value modified by T1 before T1 commits
  - T2在T1提交之前读T1修改的数据
- A2 – Non-Repeatable Read: T2 reads value, after which T1 modifies it
  - T2在T1修改并提交之前读了同一数据项，且有可能再次读同一数据项
- A3 – Phantom: (see next)

# A0-脏写 Dirty Write

- A0: T2 writes value modified by T1 before T1 commits
  - T2在T1提交之前修改同一数据
  - 下例中如果初始X.bal=50, 那么结果有X.bal=60的可能性, 左边事务的更新被覆盖掉.

Deposit 1  
read(X.bal)  
X.bal := X.bal + \$50  
write(X.bal)

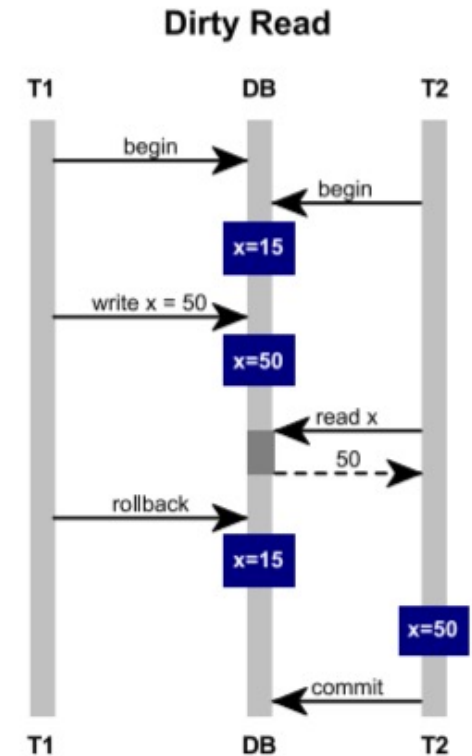
Deposit 2  
read(X.bal)  
X.bal:= X.bal + \$10  
write(X.bal)

- A0即所谓的写写冲突, 是任何系统都要避免的



# A1-脏读 Dirty Read

- A1-Dirty Read: T2 reads value modified by T1 before T1 commits
  - T2在T1提交之前读T1修改的数据
  - 图中T2读了一个未提交的x=50
- A1可看做一种读写冲突(先写后读)



# A1-脏读 Dirty Read

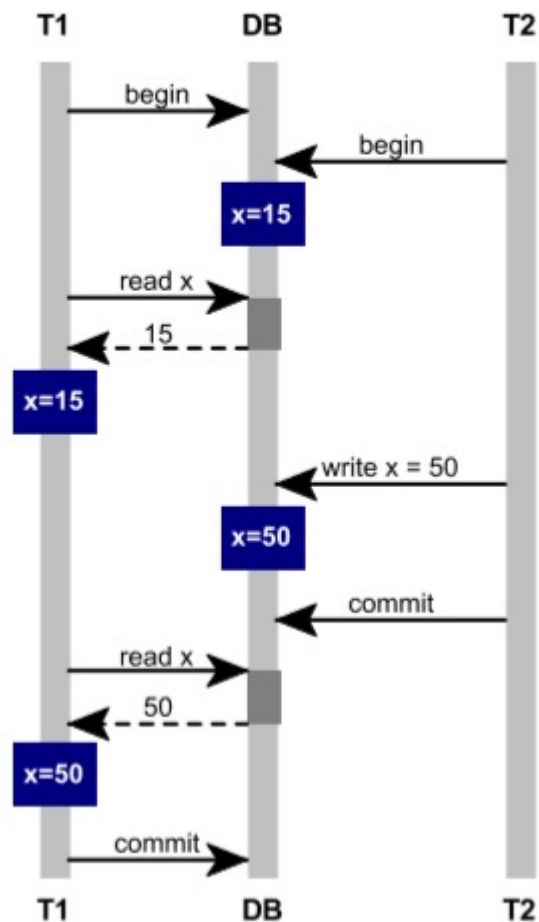
1		开始事务
2	开始事务	
3		查询账户余额为2000元
4		取款1000元，余额被更改为1000元
5	查询账户余额为1000元 (产生脏读)	
6		取款操作发生未知错误，事务回滚，余额变更为2000元
7	转入2000元，余额被更改为3000元 (脏读的1000+2000)	
8	提交事务	
备注	按照正确逻辑，此时账户余额应该为4000元	

# A2-不可重复读

- A2 – Non-Repeatable Read: T2 reads value, after which T1 modifies it
  - T2在T1修改并提交之前读了同一数据项，且有可能再次读同一数据项
  - 读写冲突(先读后写)

# A2例子

## Non-repeatable Read



1	开始事务	
2	第一次查询, 小明的年龄为20岁	
3		开始事务
4	其他操作	
5		更改小明的年龄为30岁
6		提交事务
7	第二次查询, 小明的年龄为30岁	
备注	按照正确逻辑, 事务A前后两次读取到的数据应该一致	

# A3-幻读

- A3-Phantom
  - Transaction T1 reads a set of data items satisfying some <search condition>. Transaction T2 then creates data items that satisfy T1's <search condition> and commits. If T1 then repeats its read with the same <search condition>, it gets a set of data items different from the first read.
  - 稍微有点超过page model的描述范畴
  - 两次范围查询之间，范围内写入了数据

# 幻读例子

T1

```
Select count(*)  
where dept = "Acct"  
// find ("Sue", "Acct", 3500) and ("Tim",  
    "Acct", 2400)
```

```
Select sum(salary)  
where dept = "Acct"  
// find ("Sue", "Acct", 3500) and ("Tim",  
    "Acct", 2400) and ("Joe", "Acct", 2000)
```

T2

```
Insert ("Joe", "Acct", 2000)  
// X-lock the new record  
Commit
```

# 隔离级别

- 防止产生不同的异常等级对应不同的隔离级别
  - A0必须都满足

Level	Dirty Read	Unrepeatable Read	Phantoms
READ UN-COMMITTED	Maybe	Maybe	Maybe
READ COMMITTED	No	Maybe	Maybe
REPEATABLE READ	No	No	Maybe
SERIALIZABLE	No	No	No

# 隔离级别与实现关系

- 隔离级别越高
  - 提供的隔离性保障越强
  - 事务间并发的能力越弱
  - 实现的代价越大
  - 现实中很多关系数据库仅做到RC级别



# 用锁实现来定义不同的隔离级别

- 读锁 shared lock
  - 所有读操作
- 写锁 exclusive lock
  - 所有写操作
- Short duration lock短锁
  - 动作完成前申请，完成后即释放锁
- Long duration Lock 长锁
- 锁的兼容性

		Mode of Data Item		
		None	Shared	Exclusive
Request mode {	Shared	Y	Y	N
	Exclusive	Y	N	N

# 用锁实现定义不同的隔离级别

Table 2. Degrees of Consistency and Locking Isolation Levels defined in terms of locks.		
Consistency Level = Locking Isolation Level	Read Locks on Data Items and Predicates (the same unless noted)	Write Locks on Data Items and Predicates (always the same)
Degree 0	none required	Well-formed Writes
Degree 1 = Locking READ UNCOMMITTED	none required	Well-formed Writes Long duration Write locks
Degree 2 = Locking READ COMMITTED	Well-formed Reads Short duration Read locks (both)	Well-formed Writes, Long duration Write locks
Locking REPEATABLE READ	Well-formed Reads Long duration data-item Read locks Short duration Read Predicate locks	Well-formed Writes, Long duration Write locks
Degree 3 = Locking SERIALIZABLE	Well-formed Reads Long duration Read locks (both)	Well-formed Writes, Long duration Write locks

# 思考题

下面的调度至高是在那种隔离级别下产生的调度？

R1(x) W2(x) C2 W1(x) C1

# 思考题

下面的调度至高是在那种隔离级别下产生的调度？

R1(x) W2(x) C2 W1(X) C1

答案：RC级别下