数据库 第十一章

陈鑫蕾 22920202202877

1.在数据库中为什么要并发控制？并发控制技术能保证事务的哪些特性？

（1）数据库是共享资源，可以供多个用户使用，所以通常有许多个事务同时在运行。当多个事务并发地存取数据库时就会产生同时读取或修改同一数据的情况。若对并发控制不加以控制就可能会存取和存储不正确的数据，破坏数据库的一致性。所以数据库管理系统必须提供并发控制机制。

（2）并发控制可以保证事务的一致性和隔离性，保证数据库的一致性。

2.并发控制可能会产生哪几类数据不一致？用什么方法能避免各种不一致的情况？

（1）并发操作带来的数据不一致包括三类：丢失修改、不可重复读和读“脏”数据。

（2）产生上述三类数据不一致的主要原因是并发操作破坏了事务的隔离性。避免不一致性的方法和技术就是并发控制，最常用的技术是封锁技术；也可以用其他技术，例如在分布式数据库系统中可以采用时间戳的方法来进行并发控制。

3.什么是封锁？基本的封锁类型有几种？试述它们的含义。

（1）封锁是指事务 T 在对某个数据对象进行操作之前，先向系统发出请求，对其加锁。加锁后，事务 T 就对该数据对象有控制权，在事务 T 释放锁之前，其他事务不能更新此数据对象。

（2）基本的封锁类型：排它锁（X 锁），共享锁（S 锁）；

①排它锁又称写锁。事务 T 对数据对象 A 加上 X 锁后，则只允许 T 来读取和修改 A，其他任何事务都不能再对 A 加任何类型的锁，直至 T 释放 A 上的锁；

②共享锁又称读锁。事务 T 对数据对象 A 加上 S 锁后，则事务 T 可以读 A 但不能修改 A，其他事务只能再对 A 加上 S 锁，而不能加上 X 锁，直至事务 T 释放 A 上的 S 锁，保证事务 T 释放锁之前，其他事务不能对 A 进行修改。

4.如何用封锁机制保证数据的一致性？

数据库管理系统在对数据进行读、写之前首先对数据执行封锁操作。某事务在对数据对象修改之前执行 X 锁，在对数据对象读取之前执行 S 锁，直至事务结束才释放锁。使得多个并发操作有序地进行，避免了丢失修改、不可重复读和读“脏”数据等数据不一致性。

5.什么是活锁？试述活锁的产生原因和解决方法。

（1）如果事务 T1 封锁了数据 R，事务 T2 又请求封锁 R，于是 T2 等待。T3 也请求封锁 R，当 T1 释放了 R 之后，系统首先批准了 T3 的请求。然后 T4 请求封锁 R，且又先批准了 T4 的请求，则 T2 永远处于等待状态，就是活锁。

（2）活锁产生的原因：当一系列封锁不能按照其先后顺序执行时，可能导致一些事务无限期地等待某个封锁，从而导致活锁。

（3）避免活锁的解决方法就是采取先来先服务的策略。当多个事务请求封锁同一数据对象时，封锁子系统按请求封锁的先后次序对事务排队，数据对象上的锁一旦释放，就批准申请队列中第一个事务获得锁。

6.什么是死锁，请给出预防死锁的若干方法。

（1）如果事务 T1 封锁了 R1，事务 T2 封锁了 R2，T1 又请求封锁 R2，T1 等待 T2 释放 R2 上的锁，T2 又申请封锁 R1，等待 T1 释放 R1 上的锁。就出现了 T1 等待 T2，T2 等待 T1 的情况，形成了死锁。

（2）预防死锁的方式有两种：

①一次封锁法：要求每个事务必须一次性将所用的所有数据一次性全部加锁，否则就不能执行；

②顺序封锁法：预先对数据对象规定一个封锁顺序，所有事物都按照这个顺序实行封锁。

7.请给出检测死锁发生的一种方法，当发生死锁后如何解除死锁？

（1）一般采用超时法或事物等待图法：

①超时法：如果一个事务的等待时间超过规定的时限，就认为发生了死锁。该方

法实现简单，但有可能误判死锁，事务因其他原因长时间等待超过时限时，系统会误以为发生了死锁，若时限设置得太长，又不能及时发现死锁；

②事务等待图法：有用一个有向图 G=(T,U)，T 为节点的集合，每个节点表示正在运行的事务，U 表示边的集合，每条边表示事务之间的等待情况；若 T1 等待 T2，则 T1，T2 之间划一条有向边，从 T1 指向 T2。并发控制子系统周期性地生成事务等待图，并进行检测，如果发现存在回路，就表示出现了死锁。

（2）数据库管理系统并发控制子系统检测到死锁后，就要设法解除。通常采用的方法是选择一个处理死锁代价最小的事务，将其撤销，释放此事务持有的所有锁，使得其他事务得以继续运行。

8.什么样的并发调度是正确的调度？

可串行化的调度是正确的调度。

可串行化的调度是指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执行它们时的结果相同的调度策略。

9.设 T1，T2，T3 是如下的三个事务，设 A 的初值为 0.

T1：A=A+2；

T2：A=A*2；

T3：A=A*A；

（1）若这三个事务允许并发进行，则有多少种可能的正确结果？

（2）请给出一个可串行化的调度，并给出执行结果。

（3）请给出一个非串行化的调度，并给出执行结果。

（4）若这三个事务都遵守两段锁协议，请给出一个不产生死锁的可串行化调度。

（5）若这三个事务都遵守两段锁协议，请给出一个产生死锁的调度。

（1）A 可能的结果有：2，4，8，16；

（2）可串行的调度：

T1	T2	T3
Slock A		
Y=A=0		
Unlock A		
Xlock A		
	Slock A	
A=Y+2	等待	
写回 A(=2)	等待	
Unlock A	等待	
	Y=A=2	
	Unlock A	
	Xlock A	
		Slock A
	A=Y*2	等待
	写回 A(=4)	等待
	Unlock A	等待
		Y=A=4
		Unlock A
		Xlock A
		写回 A(=16)
		Unlock A

最后 A 的结果为 16；

（3）非串行化的调度：

T1	T2	T3
Slock A		
Y=A=0		
Unlock A		
	Slock A	
	Y=A=0	
Xlock A		
等待	Unlock A	
A=Y+2		
写回 A(=2)		Slock A
Unlock A		等待
		Y=A=2
		Unlock A
		Xlock A
	Xlock A	
	等待	A=Y**2
	等待	写回 A(=4)
	等待	Unlock A
	A=Y*2	
	写回 A(=0)	
	Unlock A	

最后结果为 0；

（4）

T1	T2	T3
Slock A		
Y=A=0		
Xlock A		
A=Y+2	Slock A	
A=2	等待	
Unlock A	等待	
	Y=A=2	
	Xlock A	
Unlock A	等待	Slock A
	A=Y*2	等待
	A=4	等待
	Unlock A	等待
		Y=A=A
	Unlock A	
		Xlock A
		A=Y*Y
		A=16
		Unlock A
		Unlock A

（5）

T1	T2	T3
Slock A		
Y=A=0		
	Slock A	
	Y=A=0	
Xlock A		
等待		
	Xlock A	
	等待	
		Slock A
		Y=A=0
		Xlock A
		等待

10.今有三个事务的一个调度 R3(B)R1(A)W3(B)R2(B)R2(A)W2(B)R1(B)W1(A)，该调度满足冲突可串行化的调度吗？为什么？

满足。

R1(A)分别和 W3(B)R2(B)R2(A)W2(B)交换后，得到 R3(B)W3(B)R2(B)R2(A)W2(B)R1(A)R1(B)W1(A)，是一个串行调度。

12.举例说明对并发事务的某个调度是可串行化的，而这些并发事务不一定遵守两段锁协议。

T1	T2
Slock A	
Y=A=2	
Xlock B	
	Slock B
B=Y+1	等待
B=3	等待
Unlock B	等待
	X=B
	Unlock B

13.考虑如下的调度，说明这些调度集合之间的包含关系。

（1）正确的调度；

（2）可串行的调度；

（3）遵循二段锁的调度；

（4）串行调度。

3⊂2⊂4⊂1

14.考虑 T1，T2 两个事务。

T1：R（A），R（B），B=A+B，W（B）

T2：R（B），R（A），A=A+B，W（A）

（1）改写 T1 和 T2，增加加锁和解锁操作，并遵循两段锁协议；

（2）说明 T1 和 T2 的执行是否会引起死锁，给出一个调度说明。

(1)

T1	T2
Slock A	Slock B
R(A)	R(B)
Xlock A	Xlock B
R(B)	R(S)
B=A+B	A=A+B
W(B)	W(A)
Unlock A	Unlock B
Unlock B	Unlock A

(2)可能产生死锁

T1	T2
Slock A	
R(A)	
	Slock B
	R(B)
Xlock B	
	Xlock A

15.为什么要引进意向锁，意向锁的含义是什么？

（1）引进意向锁是为了提高封锁子系统的效率，封锁子程序不必再检查所有的上级和下级结点，只需看意向锁即可。

（2）含义：对任一借点加锁时，必须对它的上层加意向锁。

16.试述常用的意向锁：IS 锁，IX 锁，SIX 锁，给出相容矩阵。

（1）IS 锁：表示它的后裔结点加了 S 锁；

（2）IX 锁：表示它的后裔结点加了 X 所；

（3）SIX 锁：表示对它加了 S 锁，再加了 IX 锁；

相容矩阵：

T_1 \ T_2	S	X	IS	IX	SIX	-
S	Y	N	Y	N	N	Y
X	N	N	N	N	N	Y
IS	Y	N	Y	Y	Y	Y
IX	N	N	Y	Y	N	Y
SIX	N	N	Y	N	N	Y
-	Y	Y	Y	Y	Y	Y