

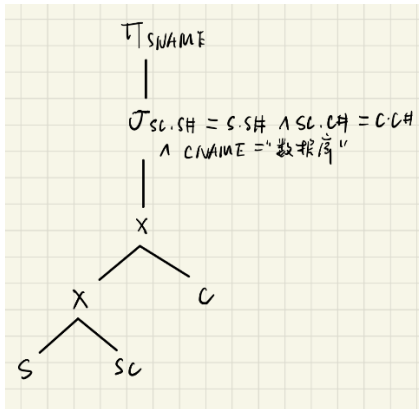
1. 设教学管理数据库有如下 3 个关系模式：

(1) 写出以上 SQL 语句所对应的关系代数表达式。

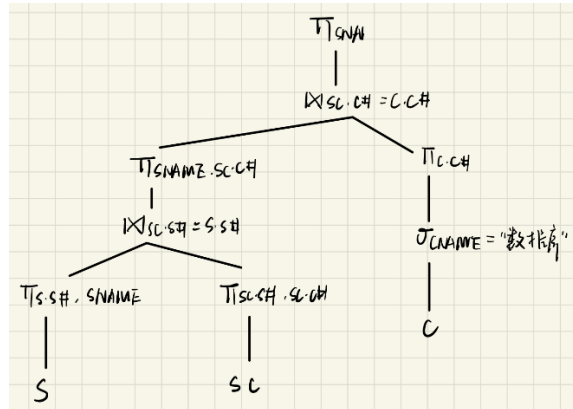
$$\Pi_{SNAME}(\sigma_{SC.S\# = S.S\# \wedge SC.C\# = C.C\# \wedge CNAME = \text{"数据库"}}(S \times SC \times C))$$

(2) 画出上述关系代数表达式所对应的查询计划树。使用启发式查询优化算法，

对以上查询计划树进行优化，并画出优化后的查询计划树。



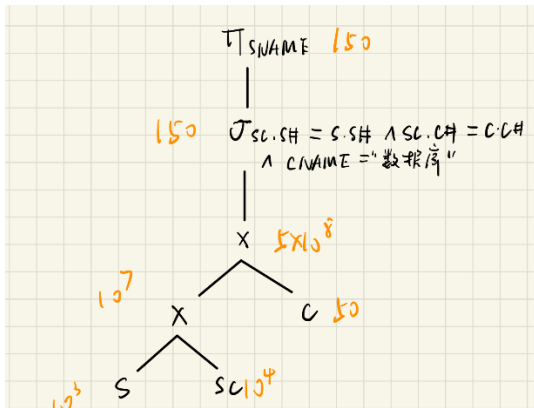
优化前



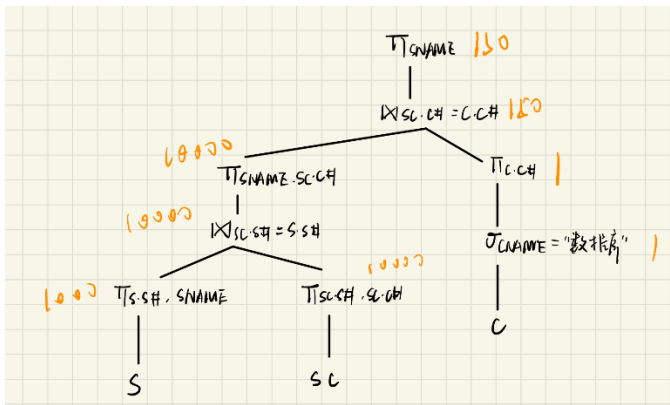
优化后

(3)

优化前：



优化后：



## 2、某个数据库的一个关系模式

(1) 写出以上 SQL 语句所对应的关系代数表达式.

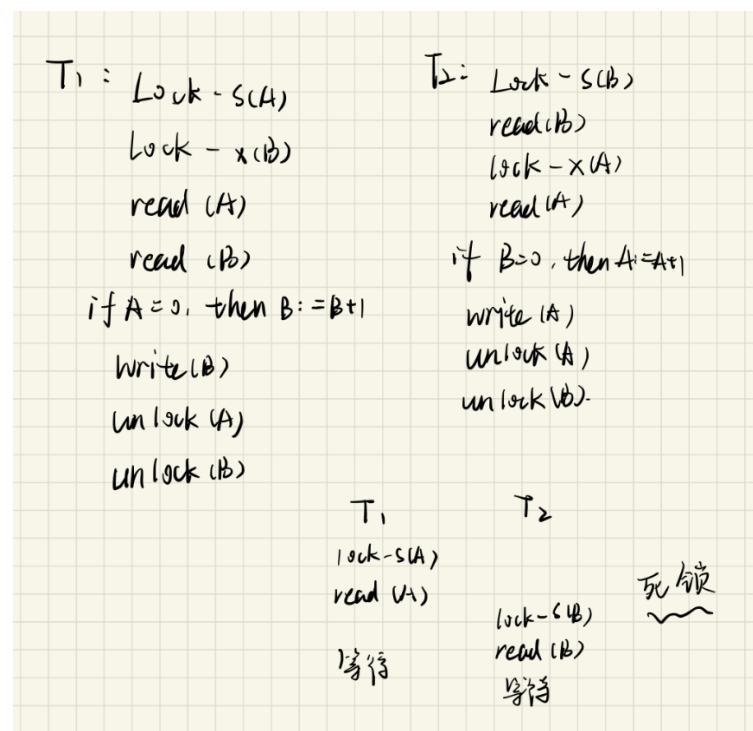
$$\Pi_{T.branch\_name} (\sigma_{T.assert > S.assert \wedge S.branch\_city = \text{"北京"}} (T \times S))$$

(2) 写出一个与 (1) 中的关系代数表达式等价的, 但更高效的关系代数表达式.

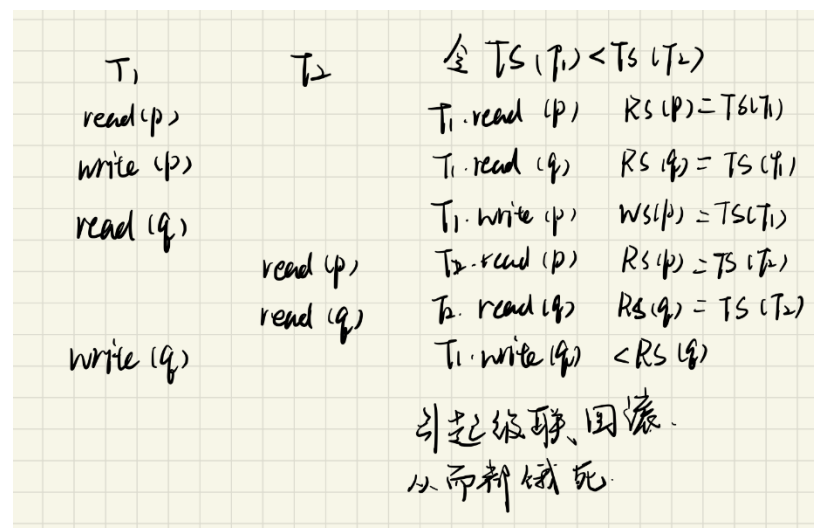
$$\Pi_{T.branch\_name} (\Pi_{T.branch\_name, T.assert} (T) \bowtie_{T.assert > S.assert} \Pi_{S.assert} (\sigma_{S.branch\_name} (S)))$$

## 3、考虑下面两个事务:

会。



4、试给出一个调度, 使得第一个事务写操作的时间戳测试失败, 引起该事务重启, 并依次引起另一个事务的级联中止。并说明是怎样导致这两个事务都饿死的。



5、设 T1、T2、T3 是如下三个事务：

时间	T1	T2	T3
1		LOCK-X(A)	
2		READ(A)	
3		A:=A*3	
4		WRITE(A)	
5		UNLOCK(A)	
6			LOCK-X(A)
7			READ(A)
8			A:=A <sup>2</sup>
9			WRITE(A)
10			UNLOCK(A)
11	LOCK-X(A)		
12	READ(A)		
13	A:=A+4		
14	WRITE(A)		
15	UNLOCK(A)		

A=40

6、一个带检查点的日志内容如下，结束处发生了故障，请简述恢复算法的过程，并给出 Undo-List 和 Redo-List，以及数据库系统恢复后的 A,B,C 的值。

重做阶段，找到最后一个检查点，进行正向扫描日志，将 undo-list 初始化为 {L}，扫描过程中，遇到<T, X, v1, v2>或<T, X, v2>重做，令 X 为 v2，一旦发现<T, start>，将 T 加入 undo-list 中，如果遇到<T, commit>或<T, abort>，将 T 从 undo-list 中删除。

撤回阶段。从尾部扫描日志，对 undo-list 中的事务对应的日志记录执行 undo 操作，如果遇到<T, start>且 T 在 undo-list 中，则写入一条<T, abort>，并从 undo-list 中删除 T。

Undo-list: {T6}

Redo-list: {T4, T5}

7、有一个带检查点的日志内容如下，请完善日志。

$A = -20, B = -10, C = 10, D = -20$

8、请解释检查点机制的目的，应该间隔多长时间做一次检查点呢，执行检查点的频率在以下各种情况下分别有何影响。

目的：

如果不设置 **checkpoint**，需要扫描整个日志，开销大且恢复过程非常漫长。有了断点之后，可以提高恢复过程的效率。

间隔多长时间：

若采用定期设置，可以设置为一小时一个检查点。如果不设置定期，可以在日志写满一半的时候设置。

影响：

- (1) 无故障发生时，频率越高会影响数据库性能，降低效率
- (2) 崩溃中恢复时，频率越高，恢复越快，提高效率
- (3) 故障中恢复时，没有影响。

9.

(1) 采用嵌套循环连接算法执行  $R \bowtie S$  分别需要进行多少次 I/O？给出具体分析过程。

53000 次。R 有 1000 块，S 有 2000 块。首先要读入全部的元组 R，然后其余  $m-1$  页读 s。即先从 R 中读入 39 块，然后遍历 S 中的每一块，因此 I/O 数应该是  $B(R)[B(S)/(M-1)] + B(R)$

(2) 采用归并连接算法执行  $R \bowtie S$  分别需要进行多少次 I/O？给出具体分析过程。

5000 次。将 R 读入内存中进行归并排序需要 1000 次。将两个关系写回需要 1000 次，最后进行归并连接需要读入内存，并遍历一次需要 3000 次。整个过程使 5000 次。

(3) 设 R.B 是关系 R 的外键，参照 S.B。如果  $R \bowtie S$  的结果中元组的平均大小是 R 中元组平均大小的 2 倍， $R \bowtie S$  的结果中元组的平均大小是 S 中元组平均大小的 1.2 倍，那么在外存中存储  $R \bowtie S$  的结果需要占用多少个块（页）？给出具体分析过程。

2000 块。计算自然连接后元组个数，由于 R.B 是参照 S.B 的外键，所以自然连接发生时，每一个 R 中的元组，必然可以与 S 中的一个元组发生自然连接，且只有一组可以，因为在 S 中 B 的取值各不相同，如此分析可知，最后结果元组的个数为  $T(R) = 20000$ 。结果元组的平均大小是 R 中元组的 2 倍，因此一块中能容纳 10 个结果元组。所以结果元组需要使用  $20000/10=2000$  块内存。