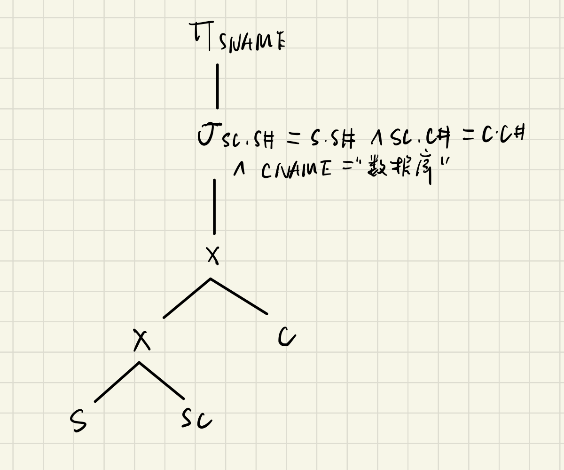
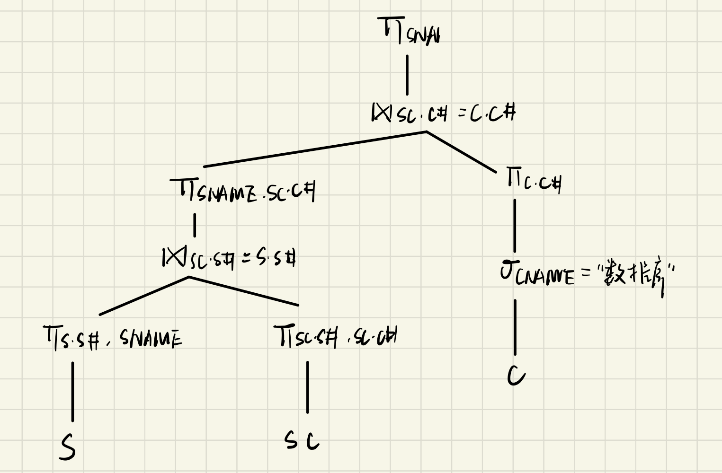
1.设教学管理数据库有如下3个关系模式：

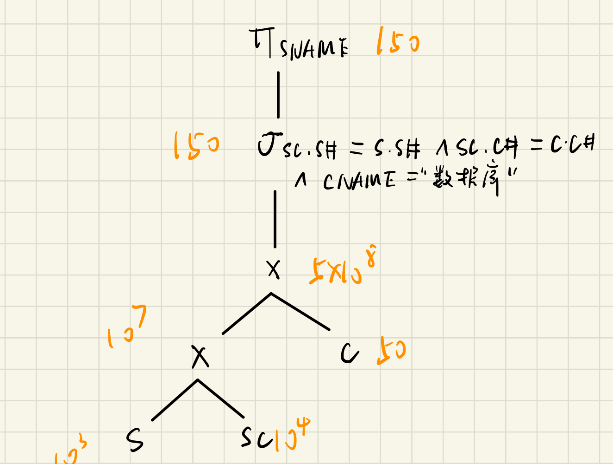
1. 写出以上SQL语句所对应的关系代数表达式。
2. 画出上述关系代数表达式所对应的查询计划树。使用启发式查询优化算法，对以上查询计划树进行优化，并画出优化后的查询计划树。

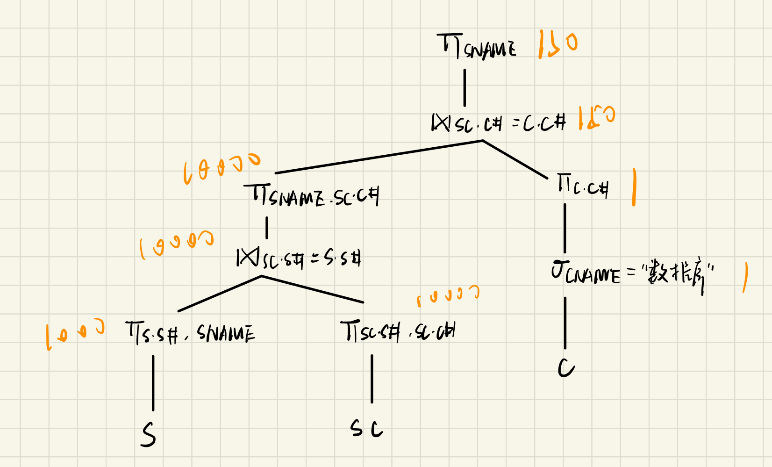
优化前 优化后

(3)

优化前：



优化后：



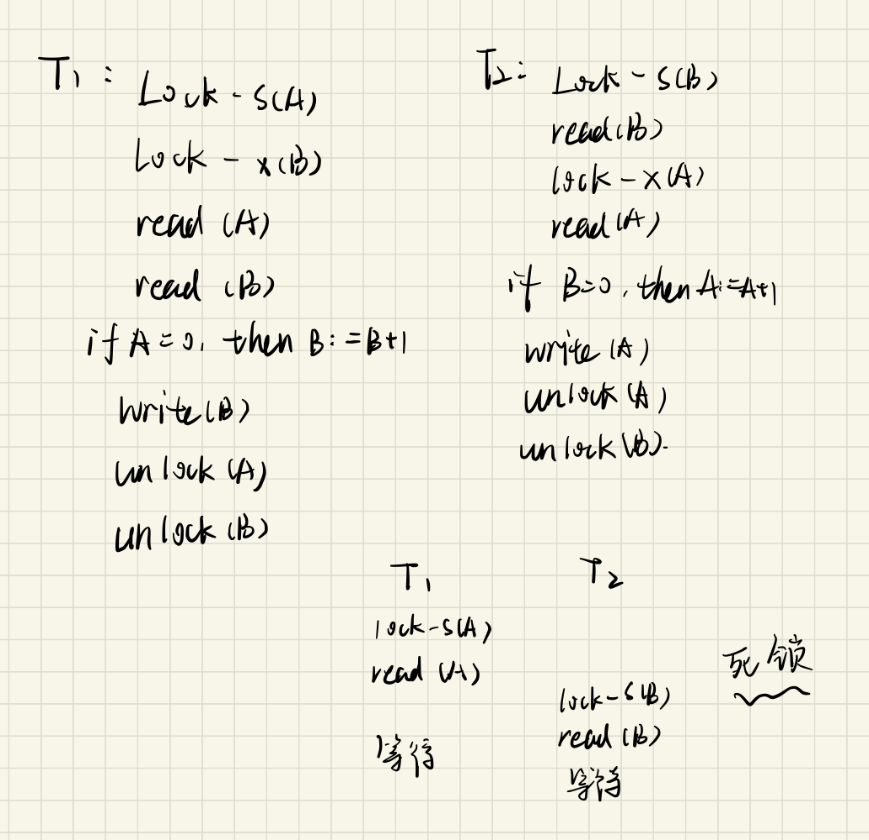
2、某个数据库的一个关系模式

1. 写出以上SQL语句所对应的关系代数表达式.

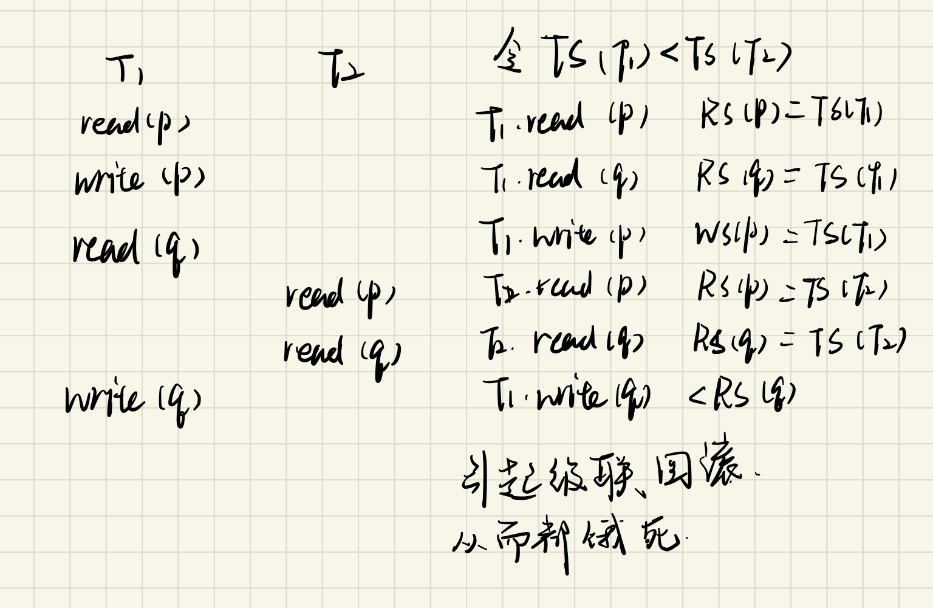
(2) 写出一个与（1）中的关系代数表达式等价的，但更高效的关系代数表达式。

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:=A2 |
| 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。

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 ⨝ 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

（2）采用归并连接算法执行R ⨝ 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 ⨝ 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 ⨝ 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 ⨝ 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。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。