1.设教学管理数据库有如下3个关系模式：

S(S#, SNAME, AGE, SEX)

C(C#, CNAME, TEACHER)

SC(S#, C#, GRADE)

其中S为学生信息表、SC为选课表、C为课程信息表；S#、C#分别为S、C表的主码，(S#, C#)是SC表的主码，也分别是参照S、C表的外码

用户有一查询语句：

Select SNAME

From S, SC, C

Where SC.S#=S.S# and SC.C#=C.C# and CNAME=“数据库”

检索选学“数据库”课程的学生的姓名。

(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。

(3)设SC表有10000条元组，C表有50条元组，S表中有1000条元组，SC中满足选修数据库课程的元组数为150，计算优化前与优化后的查询计划中每一步所产生的中间结果大小

2、某个数据库的一个关系模式如下，其中主码用下划线标出

Branch(branch\_name, branch\_city, assets)

考虑下面的SQL语句：

**select** T.branch\_name

**from** branch T,branch S

**where** T.assert>S.assert **and** S.branch\_city=’’BeiJing’’

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：

T1:read(A)

read(B)

if A=0 then B:=B+1

write(B);

T2:read(B)

read(A)

if B=0 then A:=A+1

write(A).

给事务T1、T2增加加锁和解锁指令，使它们遵从两段锁协议。这两个事务会引起死锁吗？如果会，请举出例子。

4、考虑时间戳排序协议，以及两个事务，一个执行写两个数据项p和q，另一个执行读这两个数据项，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：

T1：A:=A+4

T2：A:=A\*3

T3：A:=A2

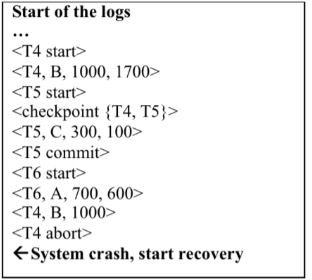
初始A=2

设三个事务都遵守两段锁协议，按 T2-T3-T1的顺序执行，请给出一个不产生死锁的可串行化调度（在下表中填写相应代码即可），并给出最终 A 的结果

|  |  |  |  |
| --- | --- | --- | --- |
| 时间 | T1 | T2 | T3 |
| 1 |  | LOCK-S(A) |  |
| 2 |  | READ(A) |  |
| 3 |  | A:=A\*3 |  |
| 4 |  | WRITE(A) |  |
| 5 |  | UNLOCK(A) |  |
| 6 |  |  | LOCK-S(A) |
| 7 |  |  | READ(A) |
| 8 |  |  | A:=A\*\*2 |
| 9 |  |  | WRITE(A) |
| 10 |  |  | UNLOCK(A) |
| 11 | LOCK-S(A) |  |  |
| 12 | READ(A) |  |  |
| 13 | A:=A+4 |  |  |
| 14 | WRITE(A) |  |  |
| 15 | UNLOCK(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。



定位到checkpoint {T4,T5}，考虑checkpoint list中的事务以及之后开始的事务，即T4,T5,T6。T5在checkpoint前开始，在checkpoint之后，故障之前提交，故需要Redu。T4在checkpoint前开始，故障时没有提交，故需要Undo。T6在checkpoint后开始，故障时没有提交，需要Undo。所以Undo-List:{T4,T6}，Redo-List:{T5}。

回复后A=700，B=1000，C=100。

7、有一个带检查点的日志内容如下，请完善日志。

<T0 start> //T0 事务开始

<T0,A,-30,10>

<T0, commit> //T0 事务提交

<T1, start> //T1 事务开始

<T1,B,-10,10>

<T2, start> //T2 事务开始

<T2,A,10,20>

<T2,A,\_\_20,-20>

<checkpoint{\_\_T1,\_\_T2}> //检查点标记

<T2, commit> //T2 事务提交

<T3, start> //T3 事务开始

<T3,C,10,-20>

<T3,D,-20,30>

结束处<T3,D,-20,30>发生了故障，请问恢复后数据库中 A,B,C,D 的值各为多少？ (注：更新日志记录的格式为< Ti, X, Vold, Vnew>)

Undo-List:{T1,T3}，Redo-List:{T2}

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。

（1）无故障发生时。

（2）从系统崩溃中恢复。

（3）从介质（磁盘）故障中恢复。

目的：在当系统故障发生时，检查日志，决定哪些事务需要重做，哪些事务需要撤销。为了减小搜索过程，使用检查点机制，只对有必要的事务进行处理。

可以定期或不定期做一次检查点。如一个小时做一次，或者日志文件写满一半时做一次。

影响：

（1）无故障发生时，频率越高，检查时间越长，越影响性能。

（2）从系统崩溃中恢复，频率越高，恢复时需要应用的重做或撤销的日志就越少，一定程度内会缩短检查时间。

（3）从介质故障中恢复，由于日志文件存储在介质中，所以频率越高，发现故障就越及时。

9.

已知2个关系R(A, B)和S(B, C)，其主键分别为R.A和S.B。R有20000个元组，S有60000个元组，一块中可以容纳20个R元组或30个S元组。设2个关系均采用聚簇存储，且每个关系中的元组均已按照其主键值递增排序。现在要执行自然连接操作R ⨝ S。设缓冲区中可用内存页数为M = 40。回答下列问题：

（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。

B(R)=20000/20=1000块

B(S)=60000/30=2000块

假设自然连接的结果的块个数为B(T)

嵌套循环连接：

S作为外层，I/O次数为2000+1000\*60000=6.0002e7

块嵌套循环连接：

使用R作为外层，38个内存页作为R的缓冲。外层循环需要1000/38=27轮。对于R需要的I/O次数为1000，对于S需要的I/O次数为27\*2000=54000，共计55000+ B(T)次。

（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。

R和S已经有序，按序扫描R和S，I/O次数为B(R)+B(S)+ B(T)=3000+ B(T)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。

由于B是R的外键，S的主键，所以连接后元组个数为20000。

由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，所以一块可以容纳10个R ⨝S元组。

故需要占用20000/10=2000个块。