[illegible]

7203610316

1. (1) $\frac{B(R)(B(S)+A)}{R} = \frac{1000\times(2000+40)}{40} = 51000$次

(2) $2B(R)+B(R)+B(S)=5000$

(3) 由于B为关系S的主键，故 $R\bowtie S$ 和S数量相等，

故 $20000\div15=1334$ 块

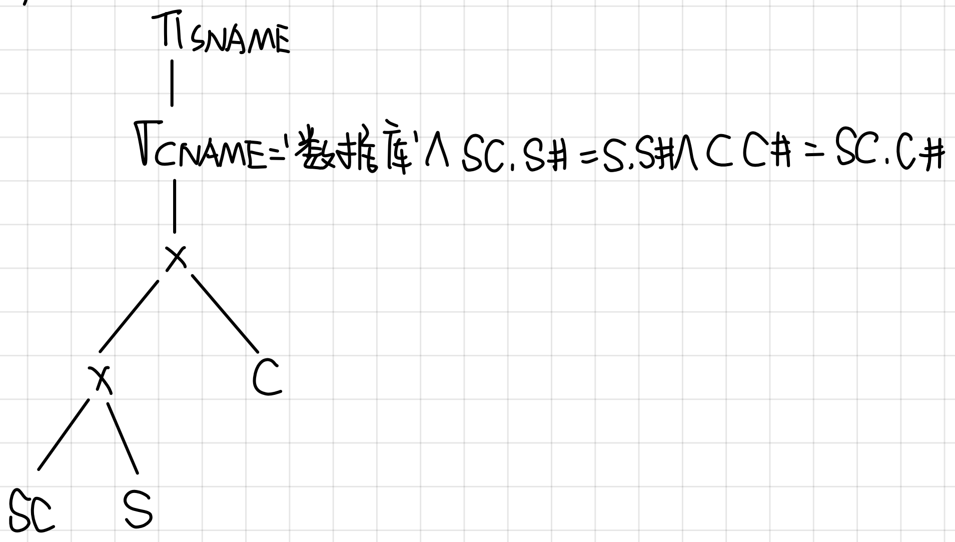
2. (1) $B(R)+T(R)\cdot\lceil\frac{B(S)}{V(S,Y)}\rceil=3050$

(2) $B(R)+\frac{T(R)\cdot T(S)}{V(S,Y)}=75050$

3. (1)

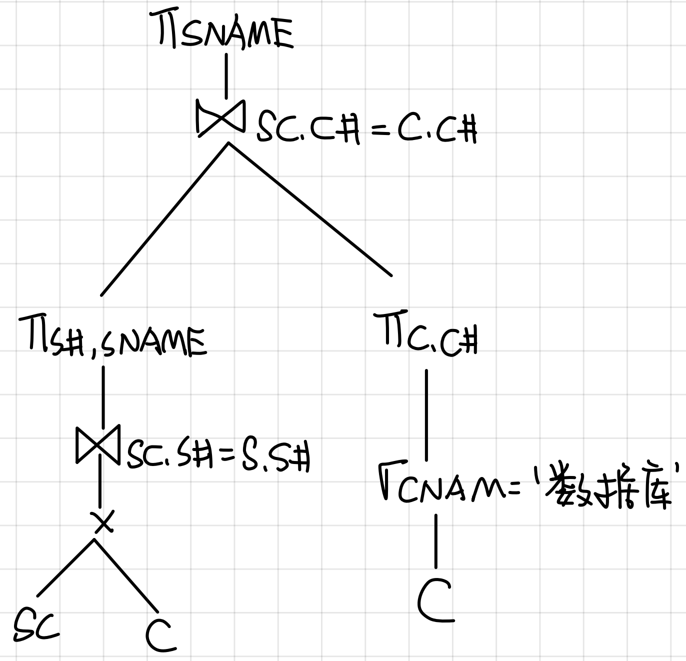
$\pi_{SNAME}(\sigma_{CNAME='数据库'\wedge SC.S\#=S.S\#\wedge C.C\#=SC.C\#}(SC\times S\times C))$

(2)



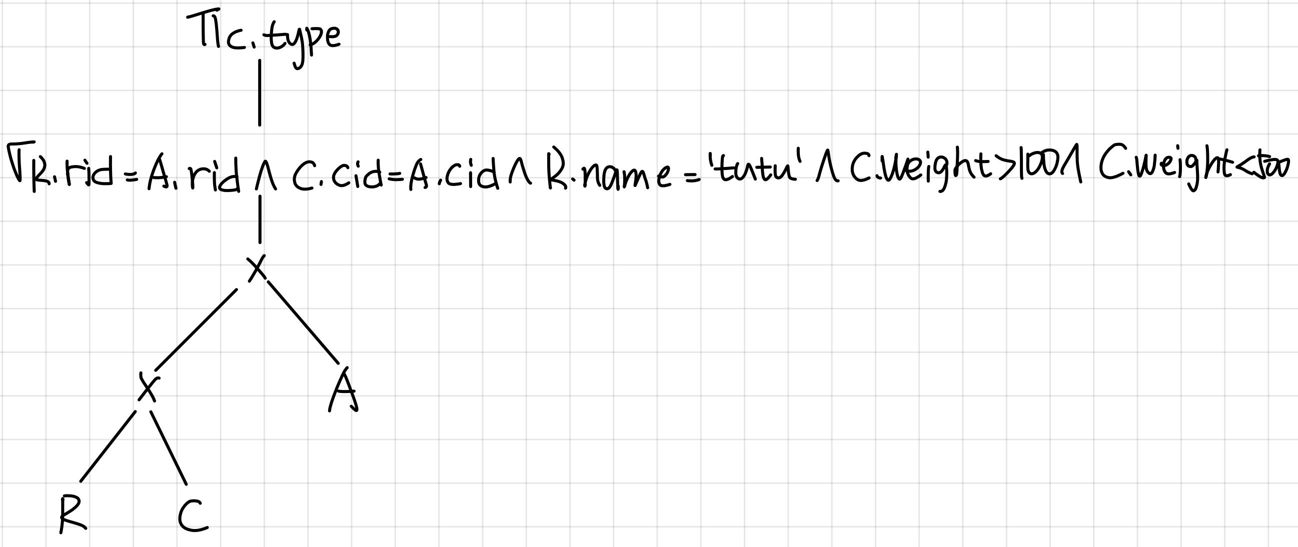
π_{SNAME}
$\sigma_{CNAME='数据库'\wedge SC.S\#=S.S\#\wedge C.C\#=SC.C\#}$
×
×
C
SC
S

优化为

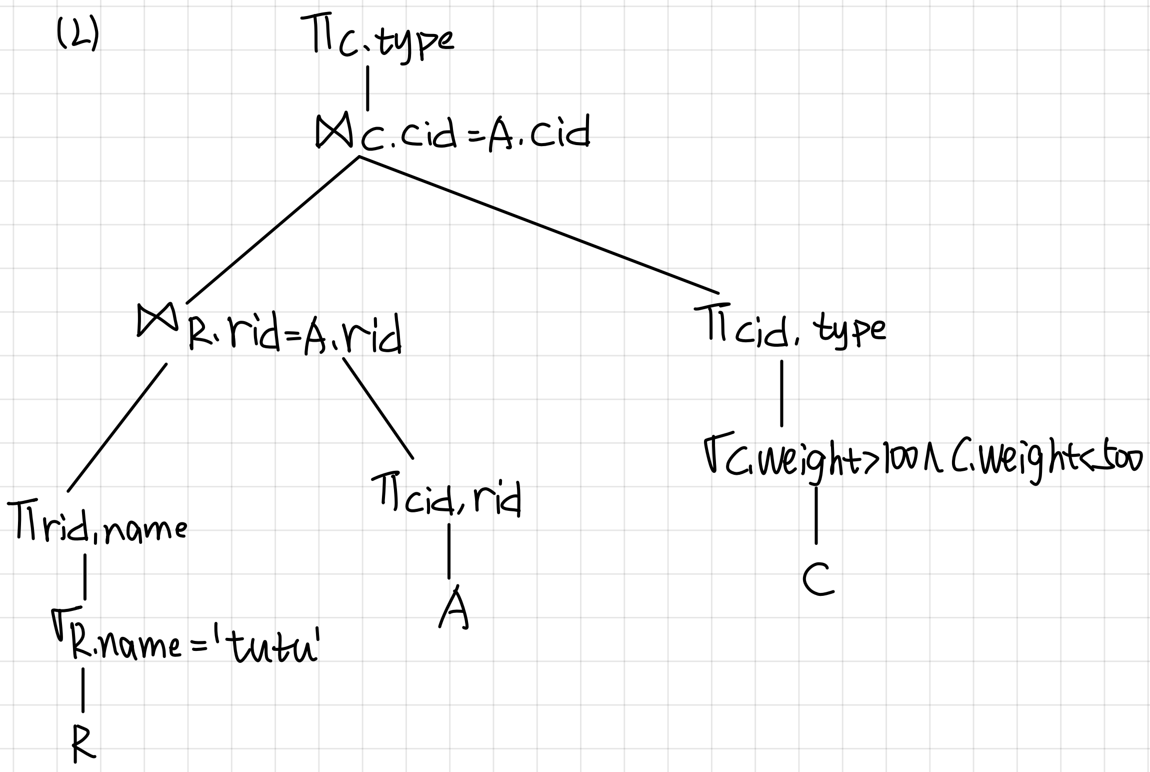


(3) 优化前：SC×S有10^7条，SC×S×C有5×10^8条
优化后：SC×S有10^4条，SC×S×C有150条

4.
(1)

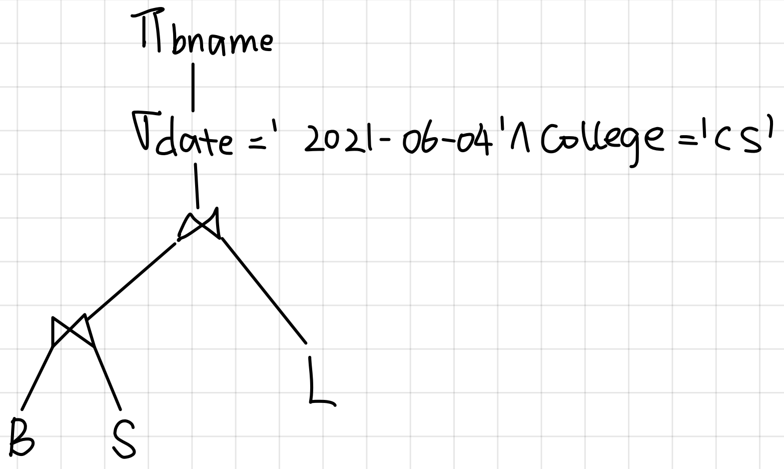

πc.type
σR.rid=A.rid ∧ C.cid=A.cid ∧ R.name='tutu' ∧ C.weight>100 ∧ C.weight<too
×
×
A
R
C

(2)

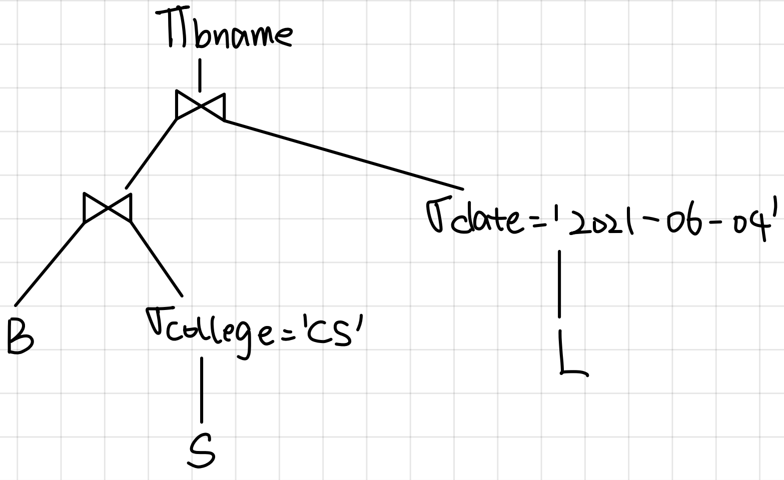


5.

(1)



(2)



理由：选择操作下推可以增加效率

6.

S可串行，它和$r_2(A)W_2(A)r_1(B)W_1(B)r_3(A)W_3(A)r_2(B)W_2(B)$等价

S'不可串行，T_1、T_2对A有读写冲突

7.

(1) T_1:

```
S-lock(A)
X-lock(B)
read(A)
unlock(A)
read(B)
if A > B then B := A
unlock(B)
```

T_2:

```
X-lock(A)
S-lock(B)
read(B)
unlock(B)
read(A)
if B < 0 then A := B*B
unlock(A)
```

(2)

T_1:	T_2:
S-lock(A)	
X-lock(B)	
read(A)	x-lock(A)
unlock(A)	
read(B)	read(A)
if (A > B) then B := A	S-lock(B)
unlock(B)	
	read(B)
	unlock(B)
	if B<0 the A:=B*B
	unlock(A)

(3)

T_1:	T_2:
S-lock(A)	
read(A)	X-lock(A)
	S-lock(B)
X-lock(B)	

(4)

① 超时法

② 等待图法：如果等待图中存在回路说明死锁

8.

(1) 对应的缓冲池策略：STEAL + NO-FORCE，即允许将未提交事务的修改写回磁盘，且不强制事务在提交之前将所做的修改回磁盘。

(2)

redo：$<T_1, A, 114, 114514>$、$<T_1, B, \text{"hit"}, \text{"hitcs"}>$

undo：$<T_3, B, \text{"hit"}, \text{"hitcsdb"}>$、$<T_2, A, 114514, 1919810>$

(3) A = 114514， B = "hitcs"

先Undo T_2、T_3，A变为114514，B变为"hitcs"

再redo T_1

9.

(1)

redo：T_4

undo：T_2、T_3、T_5

T_1 不操作

(2) redo：T_6

undo：T_7、T_8

(3) 对检查点时刻正在运行或者在其之后开始的事务检查有无Commit或absort记录；如果有则Redo，否则Undo.