1.

a)

$\text{PROJ}_1 = \sigma_{\text{LOC="New York"}}\text{PROJ}$

$\text{PROJ}_2 = \sigma_{\text{LOC="Montreal"}}\text{PROJ}$

$\text{PROJ}_3 = \sigma_{\text{LOC="Paris"}}\text{PROJ}$

(*better have one more fragment: $\text{PROJ}_4 = \sigma_{\text{LOC}\neq\text{"New York" and LOC}\neq\text{"Montreal" and LOC}\neq\text{"Paris"}}\text{PROJ}$ unless we have the assumption that "New York", "Montreal" and "Pairs" are the only values of LOC in PROJ table.)

$\text{EMP}_1 = \sigma_{\text{SAL}>5000}\text{EMP}$

$\text{EMP}_1 = \sigma_{\text{SAL}\leq 5000}\text{EMP}$

b)

$\text{ASG}_1 = \text{ASG} \rhd\!\lhd \text{PROJ}_1$

$\text{ASG}_2 = \text{ASG} \rhd\!\lhd \text{PROJ}_2$

$\text{ASG}_3 = \text{ASG} \rhd\!\lhd \text{PROJ}_3$

$(*\ \text{ASG}_4 = \text{ASG} \rhd\!\lhd \text{PROJ}_4)$

c)

$\text{EMP}_1 = \text{EMP} \rhd\!\lhd \text{ASG}_1$

$\text{EMP}_2 = \text{EMP} \rhd\!\lhd \text{ASG}_2$

$\text{EMP}_3 = \text{EMP} \rhd\!\lhd \text{ASG}_3$

$(*\ \text{EMP}_4 = \text{EMP} \rhd\!\lhd \text{ASG}_4)$

Apparently, the fragmentation on PROJ table is both complete and disjoint. ASG is derived fragmented based on the fragments of PROJ (PNO is a foreign key). Since PROJ_1 and PROJ_2 are disjoint, the fragments of ASG are also disjoint. Because the PNO in any record in ASG must appear in the PROJ table (foreign key constraint) and must be in one of the fragment of ASG, the fragmentation of ASG is complete.

The completeness and disjointness of the fragmentation on EMP table are not guaranteed. An employee can work on different projects (at different time), which locate at different places, "New York" and "Pairs" for example. His/Her record in EMP might be in more than one fragment (EMP_1 and EMP_3). Therefore, the fragmentation is not disjoint. If there exists some employee that not attach to any of the projects, his/her record in EMP will be dropped after the fragmentation as stated. So the fragmentation is not complete either.

2.
Access profiles:

Applications	Fragment											
	1		2		3		4		5		6	
	R	W	R	W	R	W	R	W	R	W	R	W
1	-	30	-	10	10	-	30	-	-	-	50	10
2	20	-	40	-	-	30	-	50	10	40	-	-
3	10	20	-	50	-	-	20	-	30	-	30	-
4	20	10	20	-	-	50	10	-	-	10	-	30
5	-	-	10	10	100	-	-	20	50	80	10	40
6	10	60	10	-	-	-	20	40	-	-	-	-

Cost Table:

Applications	Fragment					
	1	2	3	4	5	6
1	60	20	10	30	0	70
2	20	40	60	100	90	0
3	50	100	0	20	30	30
4	40	20	100	10	20	60
5	0	30	100	40	210	90
6	130	10	0	100	0	0

Benefit table = Frequency table^{T} x Cost table

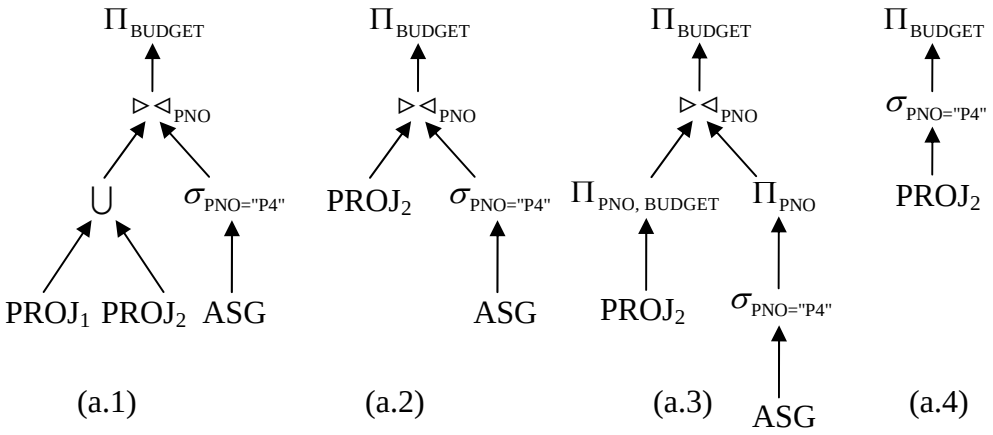
Sites	Fragment					
	1	2	3	4	5	6
A	1700	2000	2200	1600	**4500**	3500
B	3000	**3800**	1600	3000	3900	1800
C	**4800**	1600	**5600**	**3500**	1900	3000
D	3600	3000	2300	1500	1000	**3900**

Always select the site which has the largest benefits. The final allocation:

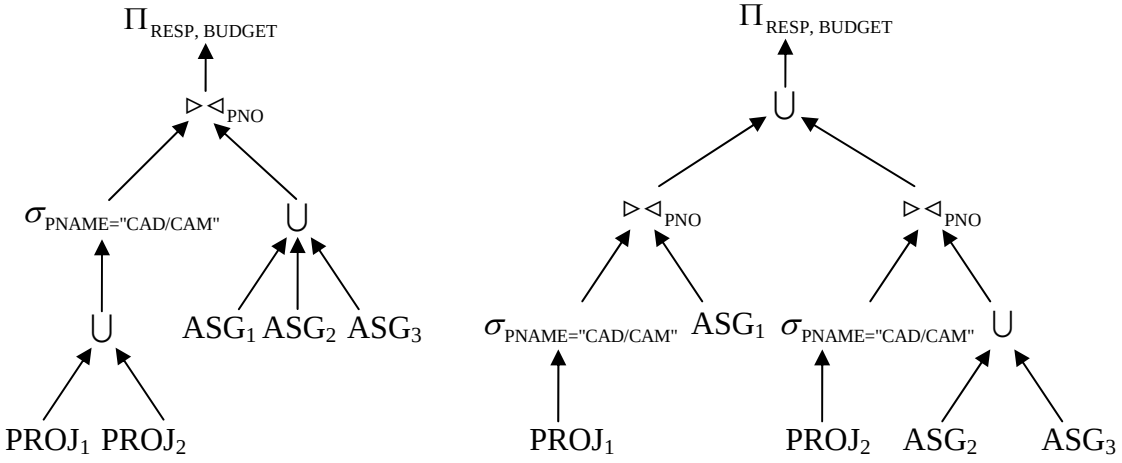
Sites	Fragment
A	5
B	2
C	**1,3,4**
D	6

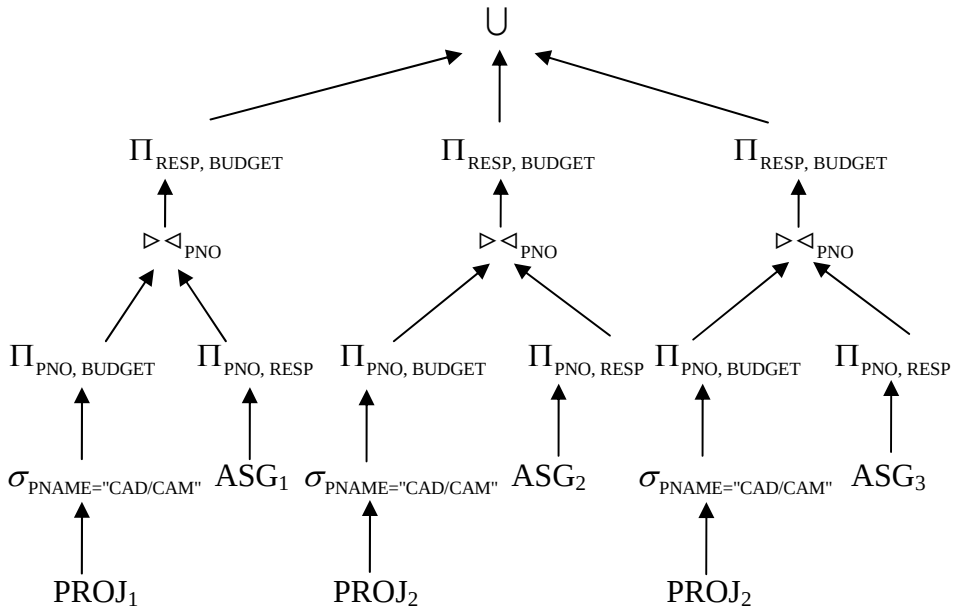
3.

a) Given the SQL query, we get the initial query tree in Figure a.1. Then, remove the empty relation $PROJ_1 \triangleright\triangleleft (\sigma_{PNO="P4"}ASG)$ and get the query tree in Figure a.2. Finally we get the reduced query tree in Figure a.3 by pushing down the projections. Figure a.3 can be further optimized to Figure a.4 (The selection is only on PNO).



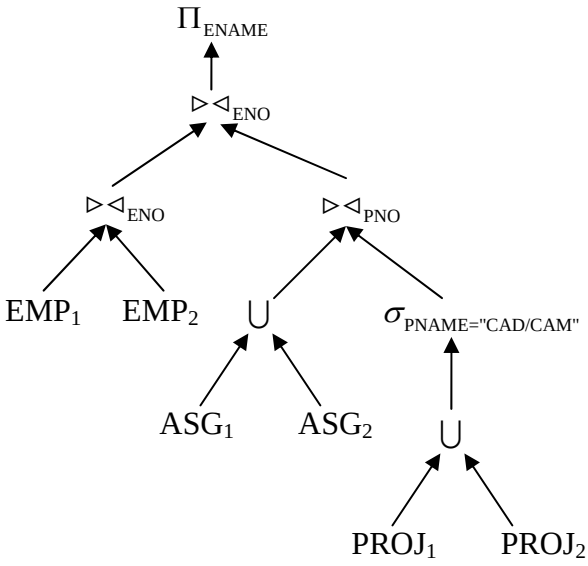
b) Given the SQL query, we get the initial query tree in Figure b.1. Allocate the query to the fragments and eliminate empty results. Then we get the query tree in Figure b.2. Finally, push projections and selections down and push unions up. The final reduced query tree is shown in Figure b.3.



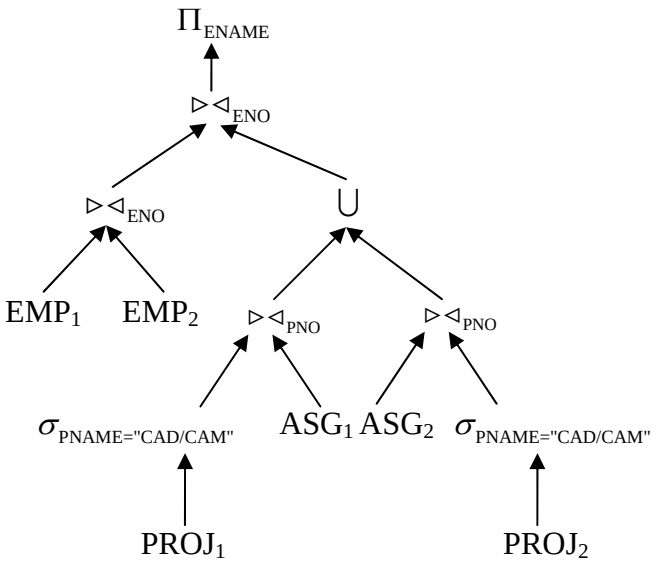


(b.3)

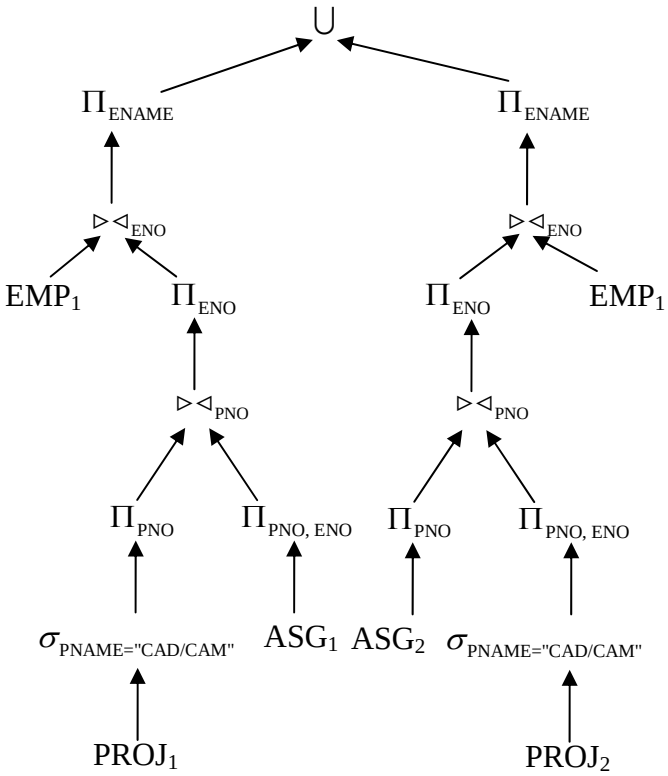
c) Given the SQL query and fragments, the initial query tree is shown in Figure c.1. After push unions up, localization and elimination on the fragments of ASG and PROJ, we get query tree in Figure c.2. Consider the vertical fragment of EMP, we find that EMP_2 is not related to the query. After pushing down projections, we finally get the reduced query tree in Figure c.3.



(c.1)


Π_{ENAME}
$\bowtie_{ENO}$
$\bowtie_{ENO}$
$\cup$
EMP_1
EMP_2
$\bowtie_{PNO}$
$\bowtie_{PNO}$
$\sigma_{PNAME="CAD/CAM"}$
ASG_1
ASG_2
$\sigma_{PNAME="CAD/CAM"}$
$PROJ_1$
$PROJ_2$

(c.2)


$\cup$
Π_{ENAME}
Π_{ENAME}
$\bowtie_{ENO}$
$\bowtie_{ENO}$
EMP_1
Π_{ENO}
Π_{ENO}
EMP_1
$\bowtie_{PNO}$
$\bowtie_{PNO}$
Π_{PNO}
$\Pi_{PNO, ENO}$
Π_{PNO}
$\Pi_{PNO, ENO}$
$\sigma_{PNAME="CAD/CAM"}$
ASG_1
ASG_2
$\sigma_{PNAME="CAD/CAM"}$
$PROJ_1$
$PROJ_2$

(c.3)

4.

a)

Table	Size
R	40 * 3000 = 120 KB
S	30 * 2000 = 60 KB
T	**70 * 2000 = 140 KB**
U	40 * 2500 = 100 KB

Plan (P_0)	Cost
Send S,T,U to R	60+140+100 = 300 KB
Send R,T,U to S	120+140+100 = 360 KB
Send R,S,U to T	**120+60+100=280 KB**
Send R,S,T to U	120+60+140=320 KB

Initial Plan P0: send R, S, U to T and the cost is 280 KB.

b)

	$\|X \triangleright\triangleleft Y\|$	Tuple size	Size
$S \triangleright\triangleleft R$	0.5 * |R| = 1500	40+30-10 = 60 B	1500*60 = 90 KB
$R \triangleright\triangleleft U$	0.5 * |R| = 1500	40+40-10 = 70 B	1500*70 = 105 KB
$S \triangleright\triangleleft U$	0.5 * |U| = 1250	30+40-10 = 60 B	1250*60 = 75 KB

Plan (P_1)	Cost	Plan (P_0)	Cost	Benefit
Send S to R, send out $R \triangleright\triangleleft S$	60 + 90 = 150 KB	Send S, R to T	60 + 120 = 180 KB	**30KB**
Send U to R, send out $R \triangleright\triangleleft U$	100 + 105 = 205 KB	Send R, U to T	120 + 100 = 220 KB	15KB
Send S to U, send out $S \triangleright\triangleleft U$	60 + 75 = 135 KB	Send S, U to T	60 + 100 = 160 KB	25KB

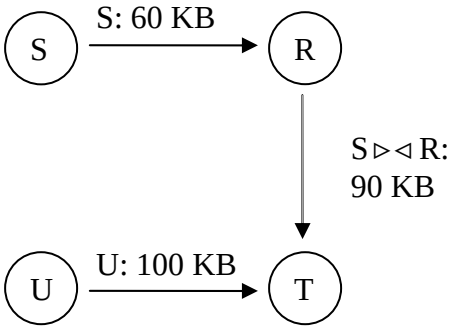
We choose the plan that benefits most. So, in the second step, we modify the initial plan P0 by sending S to R and computing $R \triangleright\triangleleft S$ on site of R.

	$\|X \triangleright\triangleleft Y\|$	Tuple size	Size
$R \triangleright\triangleleft S \triangleright\triangleleft U$	0.5 * max(|$R \triangleright\triangleleft S$|,|U|) = 0.5 * max(1250, 2500) = 1250	40+30+40-20 = 90 B	1250*90 = 112.5 KB

Plan (P_2)	cost	Plan (P_1)	Cost	Benefit
Send S ⋈ R to U, S ⋈ R ⋈ U to T	90 + 112.5 = 202.5 KB	**Send S ⋈ R to T, U to T**	90 + 100 = **190 KB**	**-12.5**KB

In the final step, since there is no benefit with the modified plan P_2, we just keep plan P_1 unchanged (send both S ⋈ R and U to T).

In summary, the complete result of hill climbing strategy is as follows:



First send S to the site of R and compute S ⋈ R at the site of R. Then, send both S ⋈ R and U to the site of T, where we perform all the remaining join operations.

The total cost is **250 KB.**