Written Assignment - III

700763258
P. Niteesh Kumar

CRN:- 21566

1.a) Given

Total tuples in relation 'r', $n_r = 10,000$

no. of tuples for block = 10

No. of blocks in relation 'r' $b_r = \frac{10000}{10} = 1000 \rightarrow b_r$

Total tuples in relation 's' $n_s = 2,000$

no. of blocks in relation 's' $b_s = \frac{2000}{5} = 400 \rightarrow b_s$

No. of buffer blocks in memory = 17.

1.1) Total cost using block nested loop join

$$\text{cost} = \left(\left\lceil \frac{b_r}{m-1} \right\rceil * b_s\right) + b_r$$

$$= \left(\left\lceil \frac{1000}{17-1} \right\rceil * 400\right) + 1000 \Rightarrow \left(\left\lceil \frac{1000}{16} \right\rceil * 400\right) + 1000$$

$$= \left((6.25) * 400\right) + 1000 \Rightarrow (63 \times 400) + 1000$$

$$= 26,200$$

1.2) Total cost using merge join

$$\text{cost} = B_s + b_r + b_s$$

where $B_s = b_r \times \left(2\left(\log_{n-1}\left(\frac{b_r}{m}\right)\right) + 2\right) + b_s \times \left(2\left(\log_{n-1}\right) + 2\right)$

$$\Rightarrow B_s = 1000\left(2\left(\log_{16}\left(\frac{1000}{17}\right)\right) + 2\right) + 400\left(2\left(\log_{16}\left(\frac{400}{17}\right)\right) + 2\right)$$

$$B_s = 1000\left(2(2) + 2\right) + 400\left(2(2) + 2\right)$$

$$B_s = 6000 + 2400 = 8400$$

$$\text{cost} = 8400 + 1000 + 400 = 9800$$

$\therefore$ Total cost = 9800

1.3) Total cost using mesh join (Recursive partition)

$$\text{cost} = 2(b_r + b_s)\left(\log_{M-1}(b_s) - 1\right) + b_r + b_s$$

$$= 2(1000 + 400)\left[\log_{16} 400 - 1\right] + 1000 + 400$$

$$= 2800\,(2.16 - 1) + 1400$$

$$= 2800\,(2) + 1400$$

$$= 7000$$

$\therefore$ Total cost = 7000

1.4) Total cost using Hash join (non-recursive partition)

$$\text{cost} = 3(b_r + b_s)$$

$$= 3\,(400 + 1000) = 4200$$

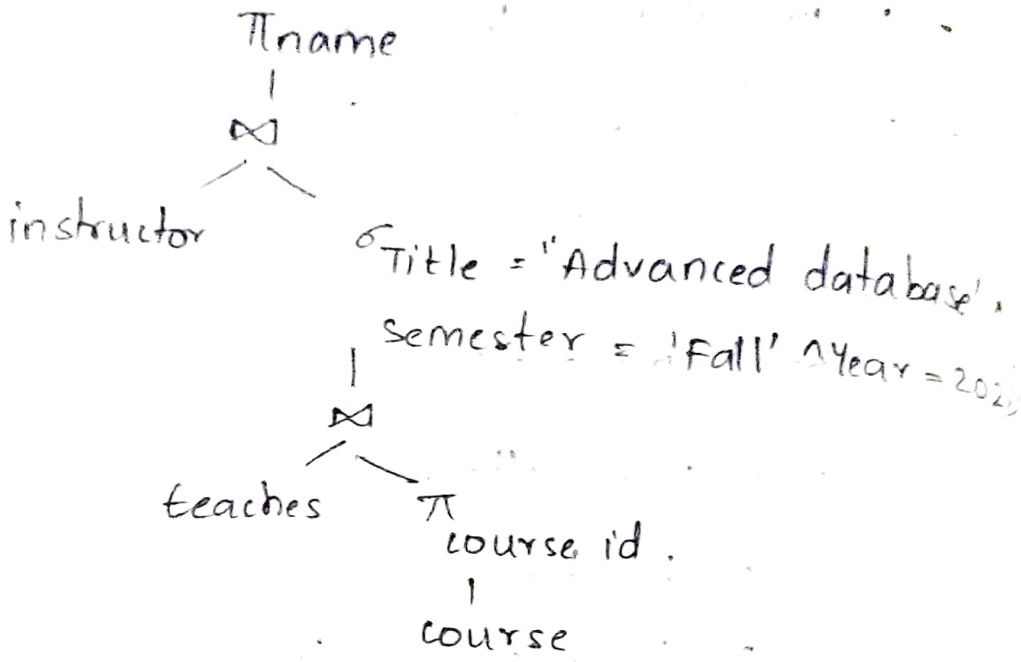
$\therefore$ Total cost = 4200

1.5) We can use any of the join algorithm for infinite memory. Because, all the algorithms will have almost same time complexity i.e., $(b_s + b_r)$ same as for Hash join the cost is given by "$3(b_r + b_s)$"

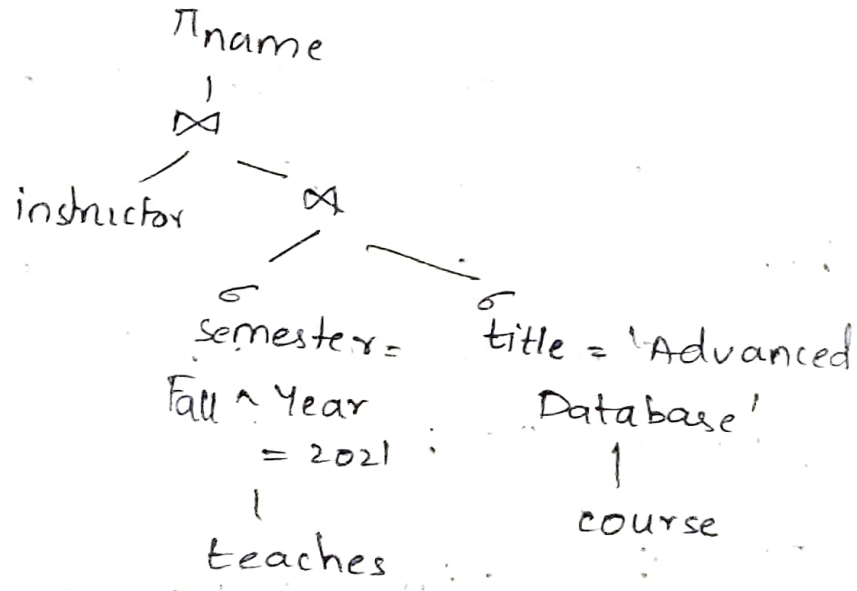
2.1 A) Find the relational Algebra expression of this SQL command. $\pi_{name}(\sigma_{title = \text{'Advance Database'} \wedge semester = \text{"Fall"} \wedge Year = 2021}(instructor \bowtie (teaches \bowtie course)))$

2.2 A) Equivalent expression :-

*) Initial expression tree



*) Transformed expression Tree



The transformed expression tree is much better than the initial expression tree because of iterating large amount data act the starting stage will decrease the number of records to be iterated at the last stage.

3A) Merge join Algorithm :-

R

	A_1	A_2	A_3
P_0 →	11	A	C
P_1 →	12	F	A
P_2 →	12	L	K
P_3 →	14	T	P
P_4 →	15	I	O
P_5 →	16	P	L
P_6 →	17	K	C

S

	A_1	A_4
P_0 →	10	30
P_1 →	11	30
P_2 →	12	20
P_3 →	14	40
P_4 →	14	10
P_5 →	17	50

P_0 → null

P_5 → null

Round (1) :-

$t_s = (10,30)$ $S_s = \{(10,30)\}$

$t_s' = (11,30)$ $t_r = (11,A,L)$

Round	P_r Point to	P_s Point to	S_s
1	(11, A, C)	(11, 30)	{(10, 30)}

Round (2) :-

$t_s = (11,30)$

$t_s' = (12,20)$

$S_s = \{(11,30)\}$

$t_r = (11,A,C)$

$t_r = (12,F,A)$

Result

A_1	A_2	A_3	A_4
11	A	C	30

Round	P_r Point to	P_s Point to	S_s
2	(12, F, A)	(12, 20)	{11, 20}

Round (3) :-

$t_s = (12,20)$

$t_s' = (14,40)$

$S_s = \{(12,20)\}$

$t_r = (12,F,A)$

$t_r = (12,L,K)$

$t_r = (14,T,P)$

A_1	A_2	A_3	A_4
11	A	L	30
12	F	A	20
12	L	K	20

Round	P_r Point to	P_s Point to	S_s
3	(14, T, P)	(14, 40)	{(12, 20)}

Round 4:-

$S_s = \{(14, 40)\}$

$t_s = (14, 40)$

$S_s = \{(14, 40), (14, 10)\}$

$t_s = (14, 10)$

$t_r = (14, T, P)$

$t_s' = (17, 50)$

$t_r = (15, I, O)$

Result:-

A_1	A_2	A_3	A_4
11	A	C	10
12	F	A	20
12	L	K	20
14	T	P	40
14	T	P	10

Round	P_r point to	P_s point to	S_s
4	(ts, I, O)	(17, 50	{(14, 40), (14, 10)}

Round 5:-

$S_s = \{(17, 50)\}$

$t_s = (17, 50)$

$t_r = (15, I, O)$

$t_r = (16, P, L)$

$t_r = (17, P, L)$

$t_r = null$

Result

A_1	A_2	A_3	A_4
11	A	C	30
12	F	A	20
12	L	K	20
14	T	P	40
14	T	P	50
17	P	L	50

Round	P_r point to	P_s point to	S_s
5	null	null	{(17, 50)}

Round	P_R point to	P_S point to	SS
1	(11, A, C)	(11, 30)	{(10, 30)}
2	(12, F, A)	(12, 20)	{(11, 30)}
3	(14, T, P)	(14, 40)	{(42, 20)}
4	(15, ±, 0)	(17, 50)	{(14, 40) (14, 10)}
5	null	null	{(17, 50)}

∴ No. of rounds for outer while loop = 5 rounds