Written Assignment - 3

CRN : [redacted] 21563

Bhanu Prakash Nimmagadda
700756164

1A) 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' $n_s = 2000$

no. of tuples for block = 5

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

$$\text{Cost} = \left(\left[\frac{b_r}{m-1}\right] * b_s\right] + b_r$$

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

$$= ((6.25) * 400) + 1000 \Rightarrow (63 \times 400) + 1000 = 26,200$$

∴ Total cost = 26,200

1.2) Total cost using merge join

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

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

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

$$\Rightarrow B_s = 1000\ (2(2)+2) + 400\ (2(2)+2)$$

$B_s = 6000 + 2400 = 8400$

cost = 8400 + 1000 + 400 = 9800

∴ Total cost = 9800

1.3) Total cost using mesh Join (Recursive partition)

$$\text{cost} = 2(b_{r1} + b_s)\left(\lceil \log_{m-1}(b_s) - 1 \rceil\right) + b_{s1} + b_s$$

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

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

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

$$= 7000$$

$\therefore$ Total cost = 7000

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

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

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

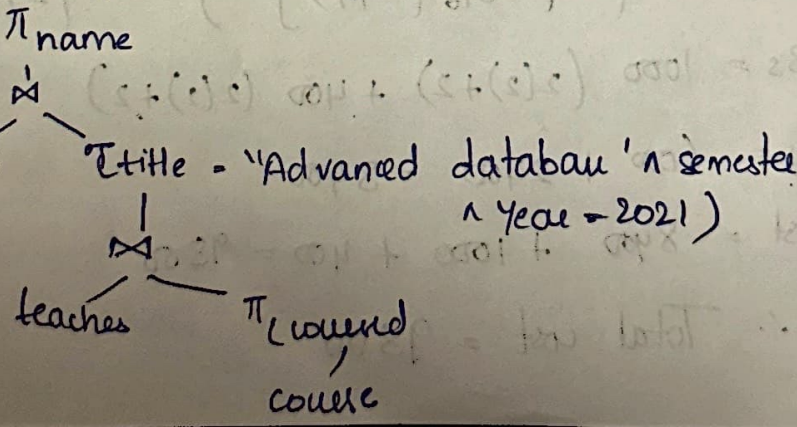
$\therefore$ Total cost = 4200

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 "$3(b_r + b_s)$"

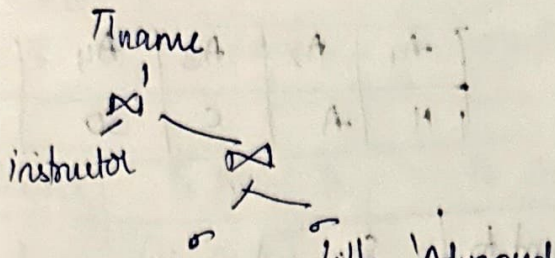
2.1) A) find the relational Algebra Expression of this SQL command. $\pi_{name}(\sigma_{File = \text{'Advance Database'} \wedge semester = \text{"Fall"} \wedge year = 2021}(instructor \bowtie (teaches \bowtie course)))$

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 at 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_R→	11	A	C
P_R→	12	F	A
P_R→	12	L	K
P_R→	14	T	P
P_R→	15	I	O
P_R→	16	P	L
P_R→	17	K	C

P_R → null

S

	A_1	A_4
P_S→	10	30
P_S→	11	30
P_S→	12	20
P_S→	14	40
P_S→	17	10
P_S→	17	50

P_S → 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:-

$t_s = (14, 40)$

$t_s = (14, 10)$

$t_s' = (17, 50)$

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

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

$t_r = (14, T, P)$

$t_r = (15, I, O)$

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	(15, I, O)	(17, 50)	{(14, 40), (14, 10)}

Round 5:-

$t_s = (17, 50)$

$t_r = (15, I, O)$

$t_r = (16, P, L)$

$t_r = (17, P, L)$

$t_r = \text{null}$

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

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	10
17	P	L	50

Round	Pr Point to	Ps point to	&s
5	null	null	{(17, 50)}

Round	Pr point to	Ps Point to	&s
1	(11, A, C)	(11, 30)	{(10, 30)}
2	(12, F, A)	(12, 20)	{(11, 30)}
3	(14, T, P)	(14, 40)	{(12, 20)}
4	(15, I, O)	(17, 50)	{(14, 40) (14, 10)}
5	null	null	{(17, 50)}

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