Query Optimization.

Clustered Index → B+ tree → Good for range selections.
eg: age between 20 & 30.

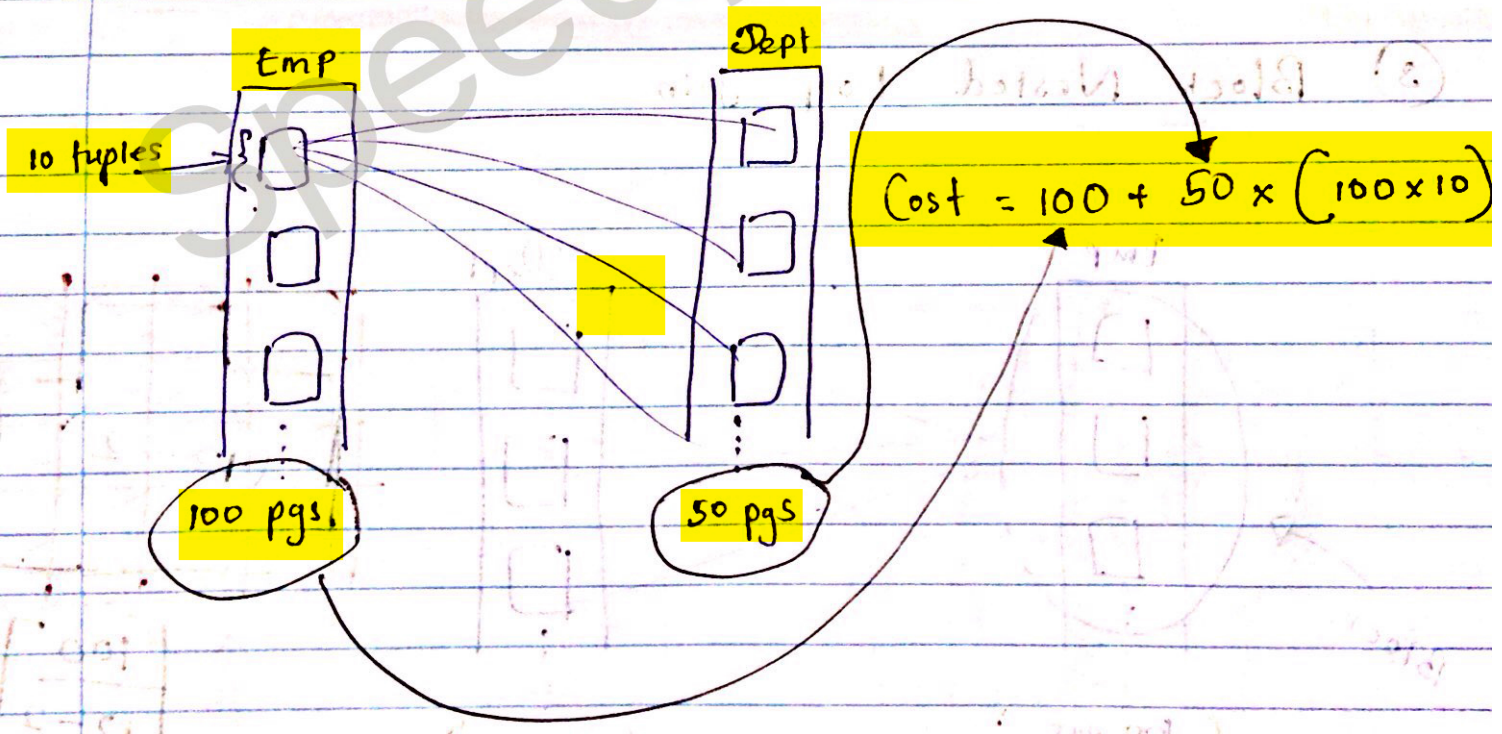
Unclustered Index → Hash → Good for equal selections
eg: age = 20

More records from selection

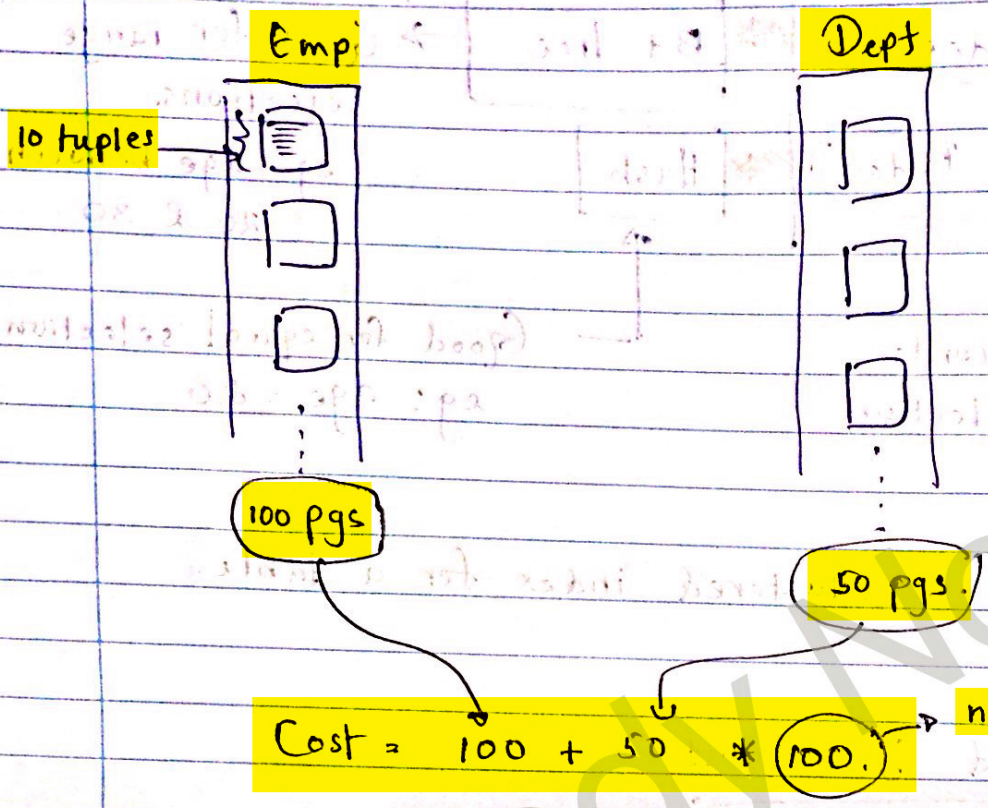
Less records from selection

(*) Maximum 1 clustered index for a table.

1 (*) Simple Nested Join Cost

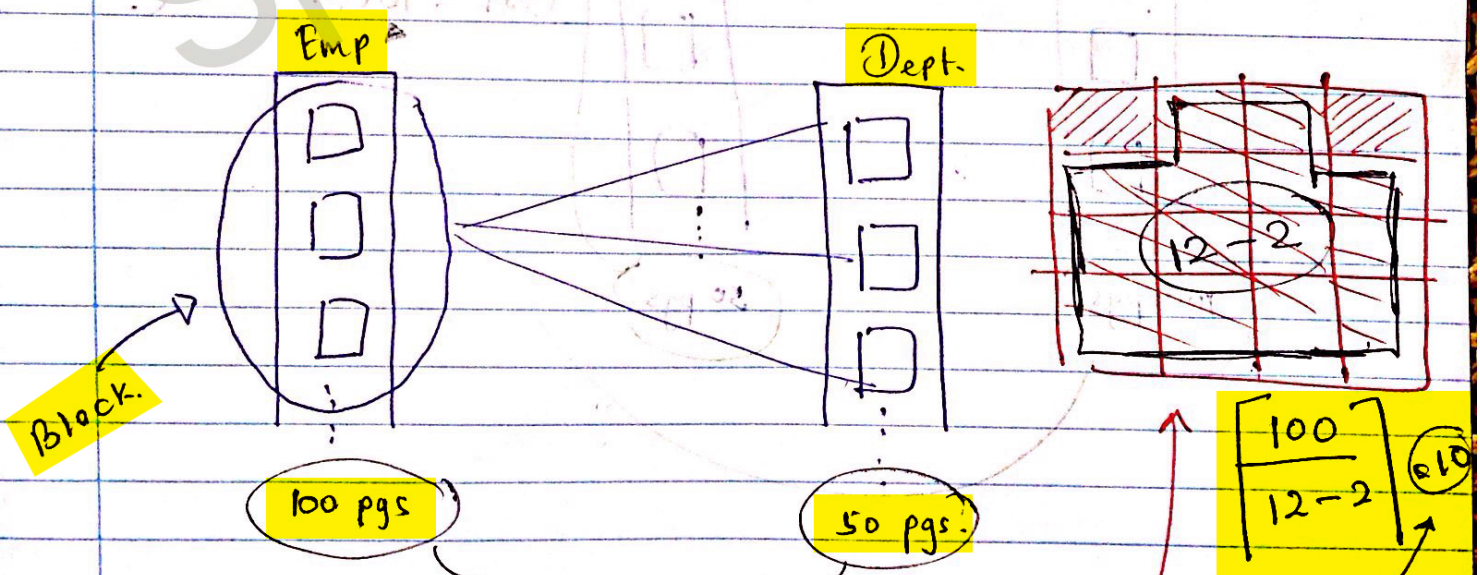


(2) Page Oriented Nested Loop Join.



Cost = 100 + 50 * (100.) → no. of pages (page oriented).

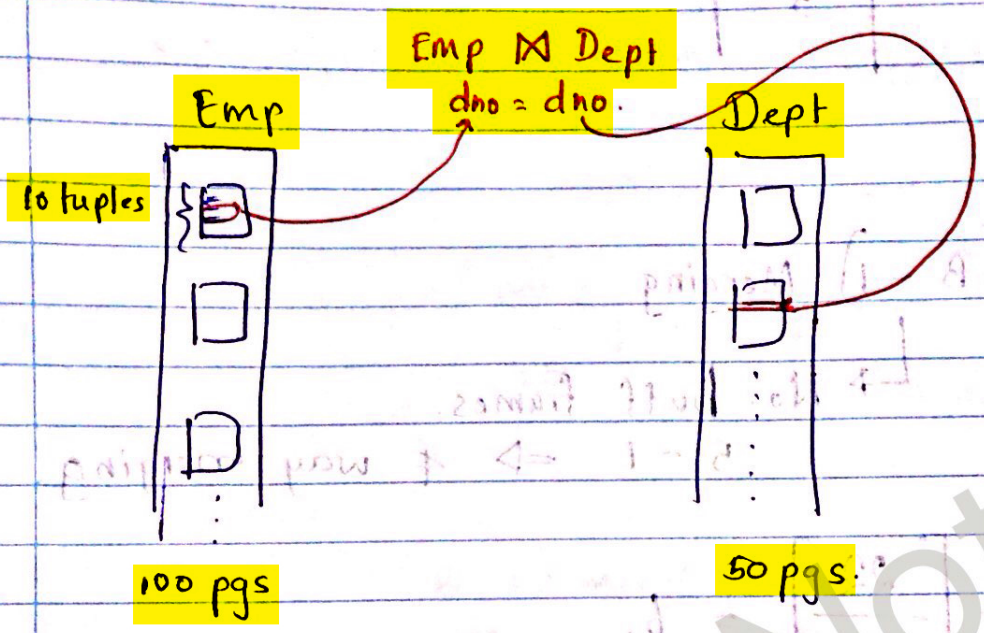
(3) Block Nested Loop Join



$\left\lceil \frac{100}{12-2} \right\rceil$ = 10

Cost = 100 + 50 × (# Blocks in emp table)

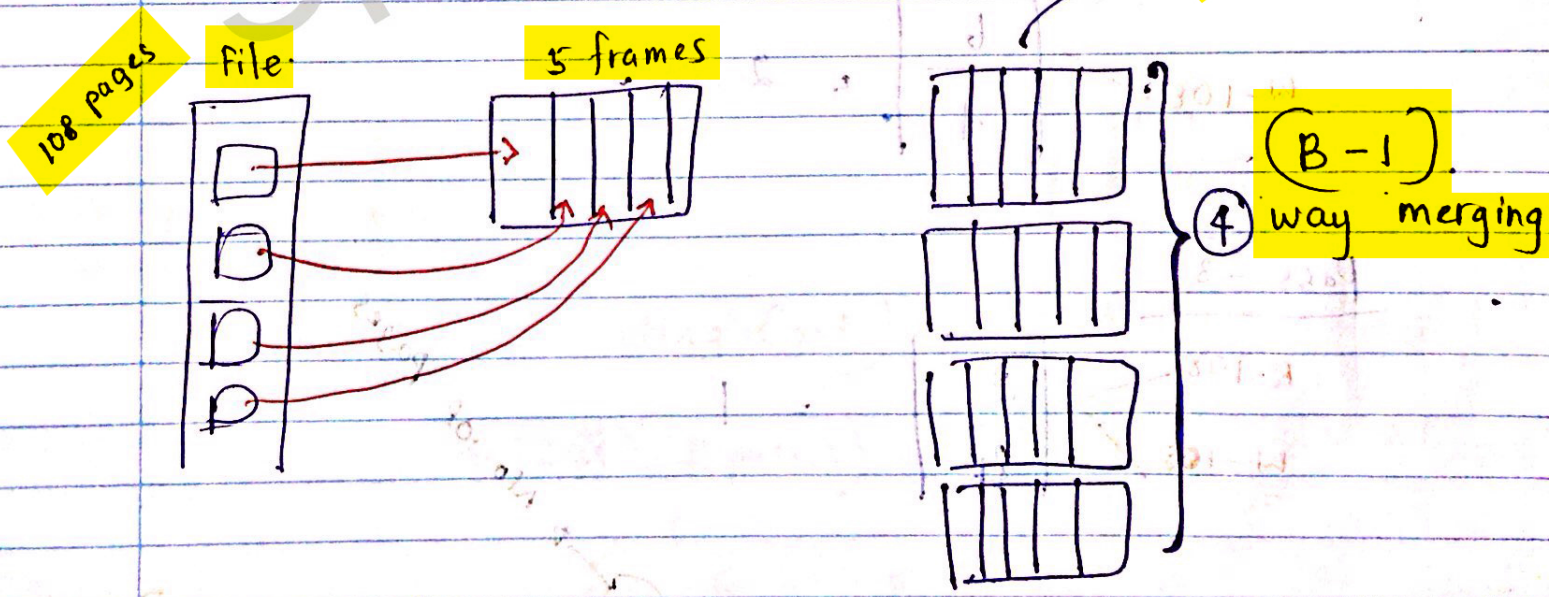
4 * Index Nested Loop Join.



Assuming → 2 - 4 → B+ tree
1.2 → hash index

$$\text{Cost} = 100 + (\text{Index Cost}) * (100 * 10)$$

* External Merge Sort.



R - 108
W - 108.

Pass - 0

$$\left\lceil \frac{108}{5} \right\rceil = 22$$

Pass - 1

(B - 1) Merging

↳ No. buff frames.

5 - 1 ⇒ 4 way merging.

R - 108
W - 108

$$\left\lceil \frac{22}{4} \right\rceil = 6.$$

Pass - 2

(B - 1) Merging

R - 108
W - 108

$$\left\lceil \frac{6}{4} \right\rceil = 2.$$

Pass - 3

R - 108
W - 108

$$\left\lceil \frac{2}{4} \right\rceil = 1$$

External Merge Sort Cost = $2(N) * (\#\text{ passes})$.

(N → No. of pages)

$= 2 \times 108 \times 4$

$= 864$

External Merge Sort Questions.

1000 pages / 50 frames what is the sorting cost?

Pass - 0

$$\left\lceil \frac{1000}{50} \right\rceil = 20$$

Pass - 1

(B - 1) merging.

$$\left\lceil \frac{20}{49} \right\rceil = 1$$

Cost = 2N * (# of passes)

= 2 x 1000 x 2

= 4000

Sort - Merge Join

R ⋈ S

SMJ = (Sorting Cost) + (Merging Cost)

= (2M * # passes) + (2N * # passes) + (M+N)

(2M * # passes): sort R; (2N * # passes): sort S; (M+N): Merge R & S

IT14063688
R. W. B. Samarasinghe

Database Management Systems III

Semester 2, 2016

1. Consider two relations R and S with the following information about them:
Relation R consists of 2,000 pages with 10 tuples per page, and relation S consists of 500 pages with 50 tuples per page. If S has a B+ tree index on the join attribute, what is the I/O cost of performing an index nested loops join? Explain your answer.

2. Consider the join of two relations R and S on attributes R.a and S.b. R contains 10,000 tuples and has 10 tuples per page. S contains 2,000 tuples also with 10 tuples per page. There are 52 buffer pages available. Estimate the minimum number of page I/Os for performing a Block Nested Loop join of R and S. Ignore the I/O cost of writing out the result. Explain your steps clearly.

3. Consider the join of two relations R and S on attributes R.a and S.b. R contains 20,000 tuples and has 10 tuples per page. S contains 1,000 tuples also with 10 tuples per page. Assume that there are clustered B+ tree indexes on R.a and S.b. Estimate the minimum number of page I/Os for performing a Sort-Merge join of R and S. Ignore the I/O cost of writing out the result. Explain your steps clearly.

4. The relation Executives has attributes ename, title, dname, and address; all are string fields of the same length. The ename attribute is a candidate key. The relation contains 10,000 pages. There are 10 buffer pages. Consider the query:

```
SELECT E.Ename
FROM Executives E
WHERE E.title = 'CFO' and E.dname = 'Toy';
```

Assume that only 10% of Executives tuples meet the selection conditions. If a clustered B+ tree index on <dname, title> is (the only index) available, what is the cost of the best plan for this query? Clearly describe how you arrived at the answer.

5. The relation Executives has attributes ename, title, dname, and address; all are string fields of the same length. The ename attribute is a candidate key. The relation contains 20,000 pages. There are 20 buffer pages. Consider the following query:

```
SELECT E.title, COUNT(*)
FROM Executives E
WHERE E.dname < 'C'
GROUP BY E.title;
```

Assume that 5% of the tuples satisfy the selection condition. If the only index available is an unclustered B+ tree on <dname, title>, what is the best plan for executing this query? Indicate the alternative plans you have considered, the cost of each alternative, and how you estimated the cost.

6. The relation Executives has attributes ename, title, dname, and address; all are string fields of the same length. The ename attribute is a candidate key. The relation contains 10,000 pages. There are 10 buffer pages. Consider the query:

```
SELECT E.Ename
FROM Executives E
WHERE E.title = 'CFO' and E.dname = 'Toy';
```

Assume that only 10% of Executives tuples meet the selection conditions. If a clustered B+ tree index on <dname, title, ename> is (the only index) available, what is the cost of the best plan for this query? Clearly describe how you arrived at the answer.

7. The relation Emp(empno: string, name: string, age: integer, salary: real) contains 4,000 pages. There are 10 buffer pages. The attribute empno takes up 10 bytes, name 20 bytes, age 2 bytes and salary 8 bytes. A clustered B+ tree on <age,salary> is the only index available. In estimating the cost of query plans, ignore the cost of writing the final result. Consider the following query:

```
Select age, avg(salary)
From emp
Where salary>40000
Group by age;
```

Assume that 10% of the tuples satisfy the selection condition.

(a) Describe the best plan and show an estimation of its cost.

(b) If this is the most important of all queries on the Emp table, what additional index if any would you implement to speed up this query? Justify your answer.

8. The relation Emp(empno: char(10), name: char(20), age: number(2), salary: real) contains 3,000 pages. There are 10 buffer pages. The attribute empno takes up 5 bytes, name 15 bytes, age 2 bytes and salary 8 bytes. A clustered B+ tree on <age> is the only index available. In estimating the cost of query plans, ignore the cost of writing the final result. For convenience, consider only the space for storing data ignoring control information and pointers.

(a) Consider the following query:

```
Select empno, name, salary
From emp
Where age =30 and salary >40000;
```

Assume that 20% of tuples satisfy each of the selection conditions and 10% of tuples satisfy both conditions. What is the cost of the best plan? Describe the plan you have chosen.

(b) Suppose the query is as follows:

```
Select age, avg(salary)
From emp
Group by age;
```

What is the best plan and its cost?

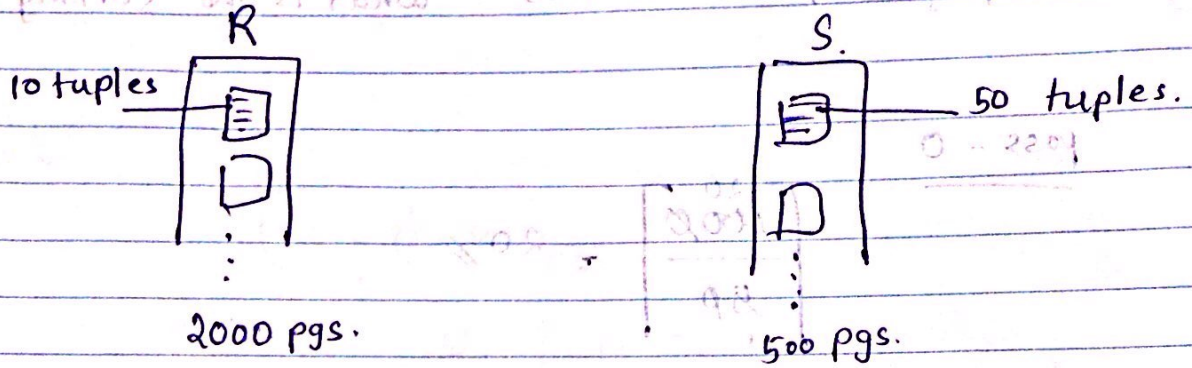
(c) Suppose the following is an important query on Emp table:

```
Select empno, name, age, salary
From emp
Where age > 16 and salary >=20000
Order by salary;
```

If 60% of tuples satisfy the selection condition, is it useful for this query to have a clustered index on <salary, age>? Explain your answer in terms of page I/Os.

Query Optimization Questions

(1) R ⋈ S.



Index Nested Loop Join ? If S has B + tree index.

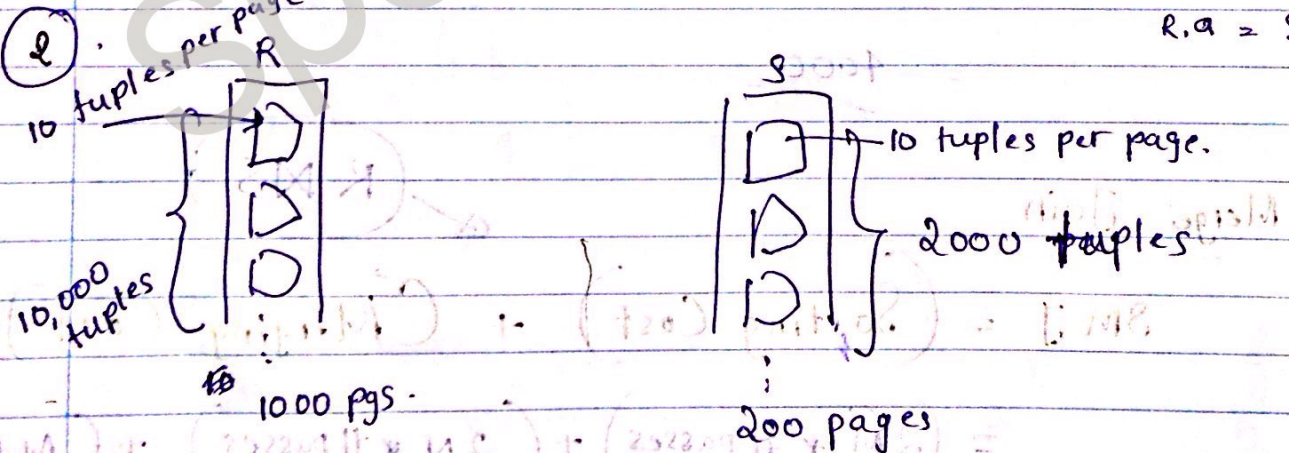
2 - 4

Cost = Scan R + (Index Cost) * (2000 x 10)

= 2000 + 3 * 20000

= 62000 I/O

(2) R ⋈ S.
R.a = S.b.



52 buffer pages.

SNLJ = 1000 + (200 x 10000)

or

= 1000 + 200 x (1000 x 10).

= 2001000 I/O.

Contd.

PONLJ

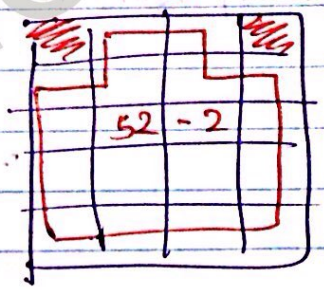
$$\text{Cost} = 1000 + (200 \times 1000)$$

$$= 201000 \text{ I/O}$$

BNLJ.

$$\text{Cost} = 1000 + 200 \times (\#\text{Blocks in R})$$

$$\left\lceil \frac{1000}{52-2} \right\rceil = 20$$

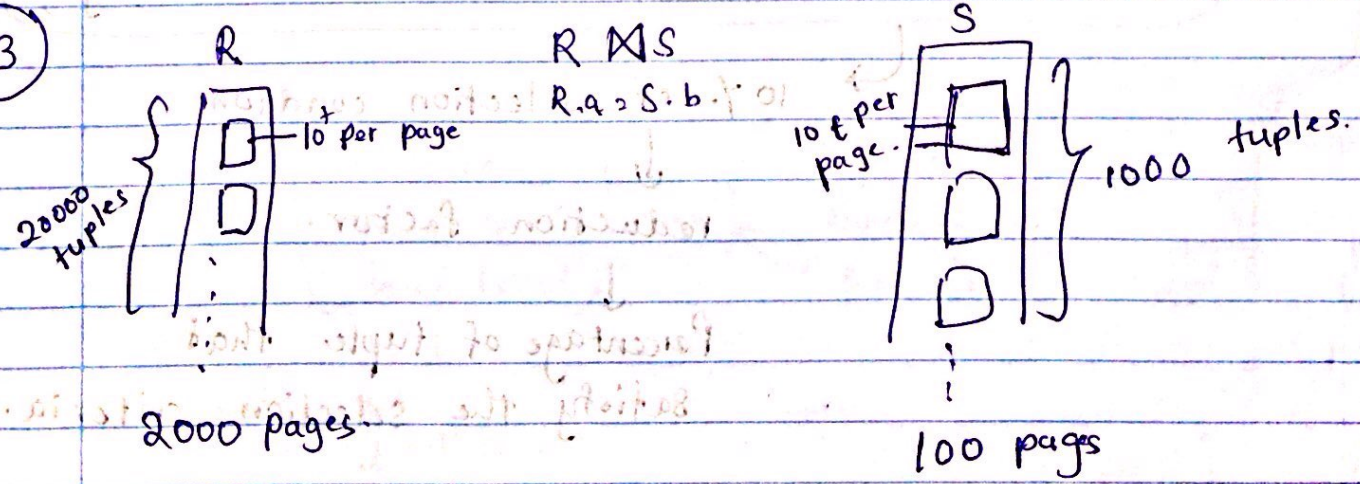


$\Rightarrow$ 52.

$$= 1000 + 200 \times 20.$$

$$= 5000 \text{ I/O}$$

(3)



Clustered B+ tree $\rightarrow$ Already sorted.

SMJ = (Sorting Cost) + (Merging Cost).

↓

= (Sorting Cost of R + Sorting Cost of S) + (Merging Cost)

= (0 + 0) + (2000 + 100).

Clustered Already Sorted.

2000 → No. of pages in R

100 → No. of pages in S.

= 2100 I/O.

(4) Executives (ename, title, dname, address)

```
Select e.ename
from Executives e
where e.title = 'CFO' and e.dname = 'Toy';
```

↳ 10% of selection condition.

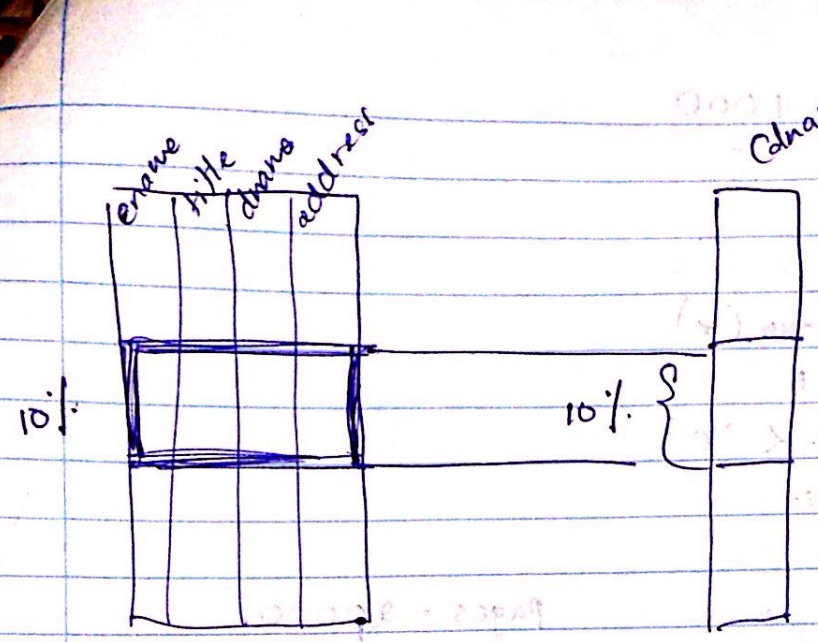
↓

reduction factor.

↓

Percentage of tuple that satisfy the selection criteria.

Clustered B + tree (dname, title)



10 buffer pages.

$\pi_{e.ename}$

σ e.title = 'CFO' AND e.dname = 'Toy' } Index Scan.

Executive

Cost = (To travels root to leaf level) + (To travels through B+ tree leaf level) + (table size x 10

(2-4) → height of the B+ tree.

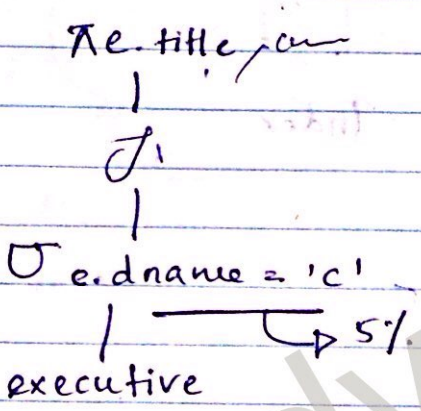
index x 10% size.

$= 3 + \left(10{,}000 \times \frac{2}{4}\right)^{\times \frac{10}{100}} + \left(10{,}000 \times \frac{10}{100}\right)$

$= 3 + 500 + 1000$

$= 1503$ I/O

(5) select e.title, count(*)
from executives E
where e.dname < 'c'
group by e.title.



Pages = 20000
buffer pgs = 20

Cost = (height of the B+ tree) + index size × 5% + Sorting Cost

Index Size = pages × 2/4

$= 20000 \times \frac{2}{4}$

$= 10,000$

$20,000 \times \frac{5}{100}$

Sorting cost = 2N * # of passes.

Dataset for sorting = $10,000 \times \frac{5}{100} = 1000$ ... 500

No. of passes:

parse (0) → $\left\lceil \frac{500}{20} \right\rceil = 25$.

parse (1) → 25

(6).

π e.ename

|

σ e.title = 'CFO' AND
e.dname = 'Toy'

|

Executive

10,000 pgs
10 buff pgs

<dname, title, ename>

No sorting cost bcz clustered.

cost = Height of B+ tree + (index size) $\times \frac{10}{100}$ + table size

= 2 + $\left(10{,}000 \times \frac{3}{4}\right) \times \frac{10}{100}$ + $10{,}000 \times \frac{10}{100}$

= 2 + 7500 $\times \frac{10}{100}$ + 1000

= 2 + 750 + 1000

= 1752 I/O

(7) Emp(empno, name, age, salary)

empno → 10 bytes.
name → 20 bytes.
age → 2 bytes.
salary → 8 bytes
40.

} field length.

Clustered B+ tree <age, salary>

4000 pgs
10 buffer pgs.

```
Select age, avg(salary)
from emp
where salary > 40000
group by age;
```

→ 10%.

(9) $\pi_{age, avg(salary)}$

$\gamma_{age, avg(salary)}$

$\sigma_{salary > 4000}$

Emp.

1000 + 400 = 1400 I/O.

Cost = (scan index) + (sorting cost)

index size = $\left(\frac{1000}{1000} \times \frac{10}{100}\right)$ +

data set for sorting = $\left(1000 \times \frac{10}{100} = \boxed{100 \text{ pgs}}\right)$.

sorting cost = $\left\lceil \frac{100}{9} \right\rceil$ (R x W) = 11 2N passes

$2 \times 100 \times 3$

$\left\lceil \frac{11}{9} \right\rceil$ (R W)

(8) Emp (empno, name, age, salary) → 3000 pgs.
5 bytes, 15 bytes, 2 bytes, 8 bytes.
10 buff pgs.

Clustered <index = age>
B+ tree index

(9). Here we cannot use index only plan. Because in select condition we have some more info like salary

π empno, name, ~~age~~, salary.
|
σ age = 30 AND salary > 40000 → 20%
|
Emp

Cost = B+ tree height + Index File Size + Pages of table.

$$= 3 + \left[3000 \times \frac{2}{30} \times \frac{20}{100}\right] + 3000 \times \frac{20}{100}$$

$$= 3 + 40 + 600$$

$$= \underline{643\ I/O}.$$

(b).

π age, avg(sal).
|
σ age
|
Emp

- No sorting cost because here the index is sorted according to age wise. So no sorting cost.
- Use index only plan.

Cost = Index File + Salary খুঁজতে cost লাগে.

$$= \left(3000 \times \frac{2}{30}\right) + 3000$$

$$= 3200\ I/O$$ Ans

c).

$\pi_{empno, name, age, salary}$

|

$\bowtie_{Salary}$

|

$\sigma_{age>16 \text{ AND } Salary>=20000}$ (Emp.)

↳ 60%.

Cost = (moment of loaded Data set) + Sorting cost.

= $\left(3000 \times \frac{10}{30} \times \frac{60}{100}\right)$ + Sorting.

= 600 + 0 + $3000 \times \frac{60}{100}$

= 600 + 1800

= 2400.