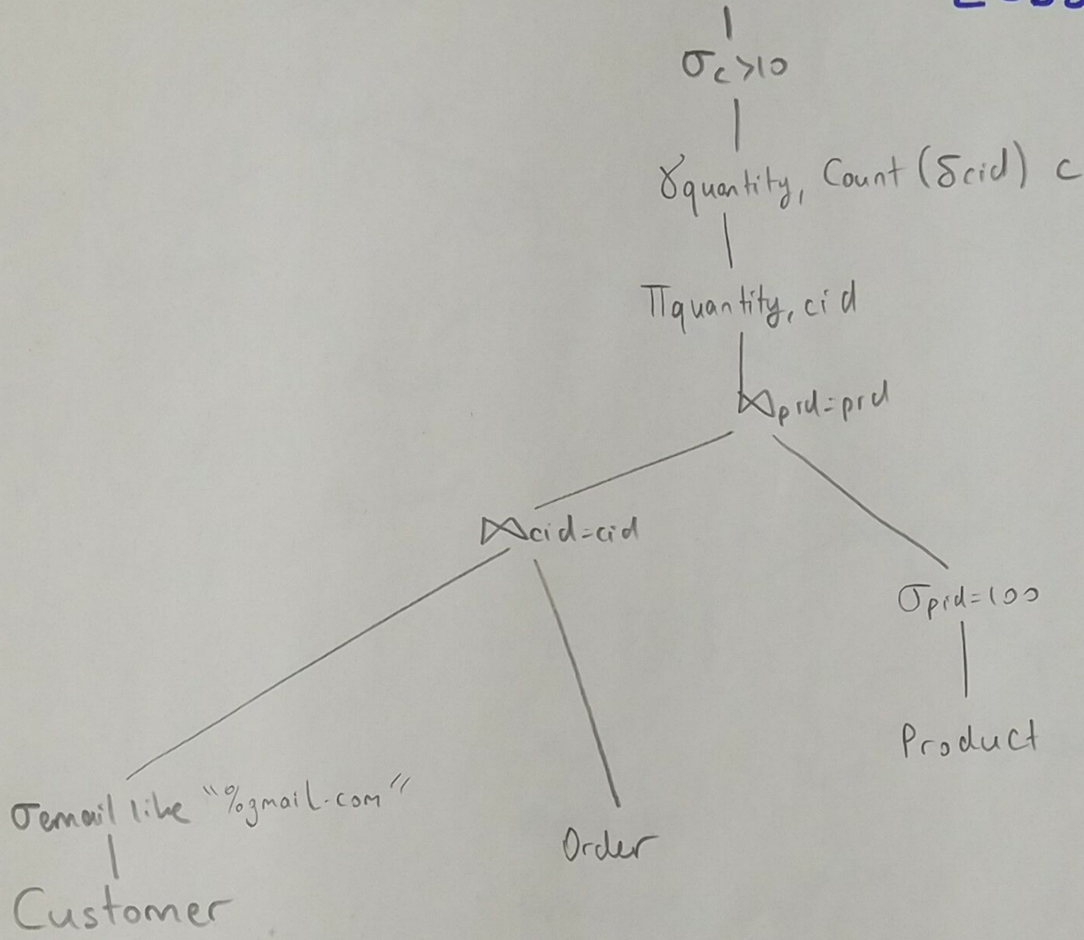
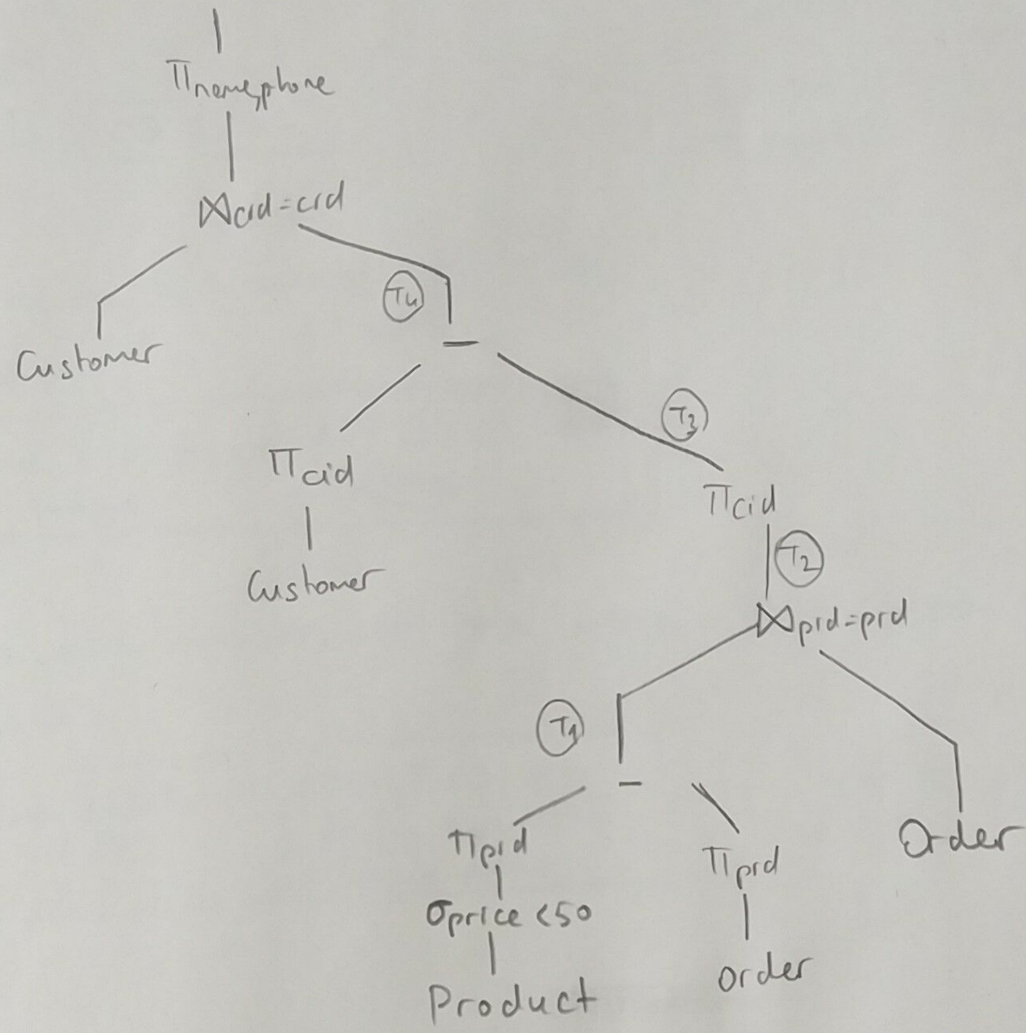
CENG 352 - WA3

Koray Can
Yurtseven
2099547

1A)



1B)



Clarification:

T1 ⇒ (Select pid
From Product
wher price<50
EXCEPT
Select Pid
From Order) → unnesting the inner most query → , to get cid, we need T1 ⋈ Order ⇒ T2

↓

Then, project ~~c~~id in T2 → T3

outer query

(Select cid
From Customer
Except
Select cid
From T3 (or T3, they are the same)) = T4

In the final step

(Select name, phone
From Customer, T4
Where T4.cid = Customer.cid)

B(Prof) = 200
T(Prof) = 1,000

B(Teach) = 500
T(Teach) = 10,000

2A)

i) Table scan

B(Teach) = $\boxed{500 \text{ I/O.}}$

ii) Clustered B+ on semester

-4 different values. each semester will have approx 125 page.

Cost = $\boxed{\text{Some I/O for index} + \text{125 I/O for fetching}}$

iii) Unclustered.

-4 dif. values. Each semester will have approx. $\frac{10,000}{4} = 2500$ tuples.

In the worst case

Cost = $\boxed{\text{Some I/O for index} + \text{2500 I/O for fetching tuples from different pages}}$

2B)

i) Table scan.

B(Prof) = $\boxed{200 \text{ I/O}}$

ii) Clustered hash

-There are 100 dept. In 'CENG' dept approx 10 professor. $\left(\frac{1000 \text{ prof}}{100 \text{ dept}}\right)$

- 1000 prof → 200 pages
  10 prof → 2 pages

Cost = $\boxed{\text{Some I/O for finding correct bucket} + \text{1 I/O (for chain, since 5 prof in the 1st bucket, the remaining 5 prof. in the second bucket)}}$

iii) Unclustered hash

-In the bucket, there will be 10 values pointing different pages.

Cost = $\boxed{\text{Some I/O for finding correct bucket} + \text{10 I/O}}$

2c)

i) Table scan

B(prof) = $\boxed{200\ I/O}$

ii) Clustered B+ tree

1000 pid ⟶ 1000 prof ⟶ 200 pages

200 pid ⟶ 40 pages

Cost = $\boxed{\text{Some I/O for finding the starting index (200)} + \text{40 I/O for linear search}}$

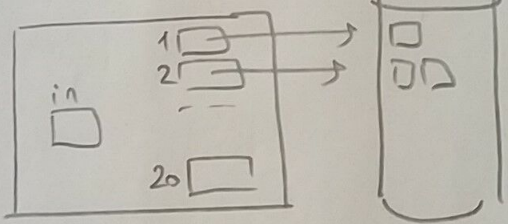
iii) Unclustered B+ tree

200 pid ⟶ 200 prof ⟶ in the index leaf we will have 200 dif. values

Cost = $\boxed{\text{Some I/O for finding the index} + 200\ I/O}$

2d)

i) Hashing



Cost = $\boxed{200\ I/O}$ for reading all professor pages

+

each partition will have approx. 5 pages, since 5 page x 10 name per page x 20 partit = 1000 rows

writing partitions to disk = 5 x 20 = $\boxed{100\ I/O}$

+

read partitions and do distinct operation = $\boxed{100\ I/O}$

= $\boxed{400\ I/O}$

- 2d) (Continued)

ii) Sorting

Read original relation $= \boxed{200\ I/O}$

+

remaining relation will have 100 page (only names) $= \boxed{100\ I/O}$

$= \boxed{400\ I/O}$

read from disk and sort, it will be in one pass since $\frac{100}{20} < 20$ ✓ $= \boxed{100\ I/O}$

Q3)

a) $B(R) + B(R) \cdot B(S)$

$1000 + 1000 \cdot 1500 = $

$$\begin{array}{r} 1500{,}000 \\ +\quad 1{,}000 \\ \hline \boxed{1{,}501{,}000} \end{array}$$

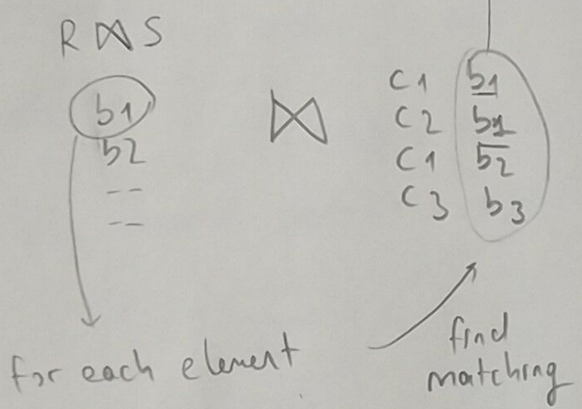
b) $B(R) + \frac{B(R)}{(M-2)} B(S) = 1000 + \frac{100\cancel{0}}{10\cancel{0}} \cdot 1500 = $

$$\begin{array}{r} 15{,}000 \\ +\ 1{,}000 \\ \hline \boxed{16{,}000} \end{array}$$

c) $B(S) + \frac{B(S)}{(M-2)} B(R) = 1500 + \frac{1500}{1\cancel{00}} \cdot 10\cancel{00} = \boxed{16{,}500}$

d)

i) Inner (S) has clustered index



$B(R) + T(R) \cdot \frac{B(S)}{V(S,b)}$

$= 1000 + 10{,}000 \cdot \frac{1{,}5\cancel{00}}{15\cancel{0}}$

10 → only need to find 10 match (in pages) ✓

$= \boxed{101{,}000}$

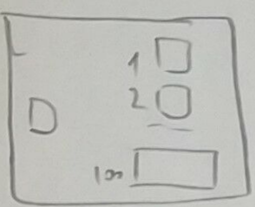
ii) $B(R) + T(R) \cdot \frac{T(S)}{V(S,b)}$

$= 1000 + 10{,}000 \cdot \frac{6\cancel{000}}{15\cancel{0}}$

40 → since it is not clustered join attribute could be in more dif. pages.

$= \boxed{401{,}000}$

Q3) (continued)

e) 

Each bucket will have $\frac{B(R)}{M} = \frac{1000}{100} = 10$ pages

Since $\frac{B(R)}{M} < M$, each bucket will fit in the memory.

Partitioning

In R: Each bucket will have $\frac{B(R)}{M} = 10$ pages (100 bucket = 1000 pages)

In S: Each bucket will have $\frac{B(S)}{M} = 15$ pages (100 bucket = 1500 pages)

(Read + Write) R + (Read + Write) S = $2000 + 3000 = \boxed{5000}$

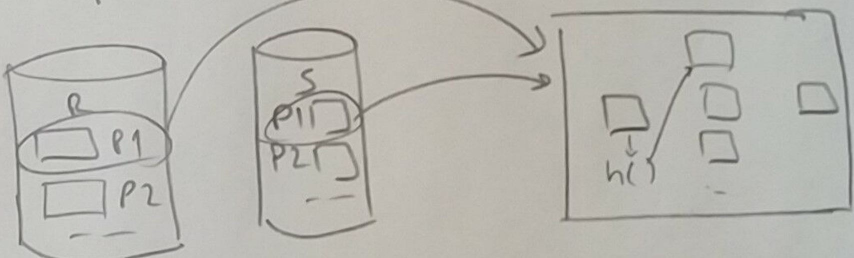
Joining

In the final step R's and S's partitions will be read from the disk

$5000 + 1000 + 1500 = \boxed{7500}$

Only needed partitions will be read.

(i.e.)



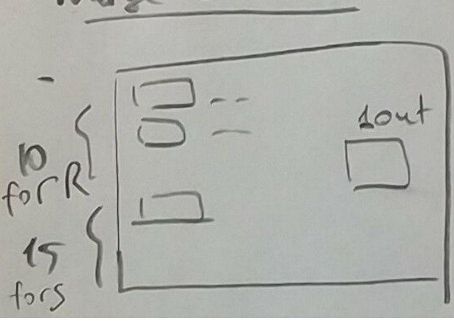
f) 1st, we have external sorting

Sorting

- Generate runs of R → 10 runs of R, last run of R is 91 pages long.
- // S → 15 runs of S, last run of S is 85 pages long.

Total cost until now = $2B(R) + 2B(S) = \boxed{5000}$
(1 write + 1 read)

Merge-Join step



Since we will read runs of R and S

$5000 + 1000 + 1500 = \boxed{7500}$

$B(R) < M^2$
$B(S) < M^2$ → Both will fit.