Question No # 02:

Slide 25:

Solution 1:

$\Pi_{sname}((\sigma_{bid=103}\ Reserves) \bowtie Sailors)$

→ first we solve $(\sigma_{bid=103}\ Reserve)$

Sid	bid	day
58	103	11/12/96

Now

$((\sigma_{bid=103}\ Reserve) \bowtie Sailors)$

Sid	Sname	rating	age	bid	day
58	Rusty	10	35	103	11/12/96

Now

$\Pi_{sname}((\sigma_{bid=103}\ Reserves) \bowtie Sailors)$

Sname
Rusty

Date: __/__/20

Slide 26:

$\pi_{sname}((\sigma_{color='red'} Boats) \bowtie Reserves \bowtie Sailors)$

→ first we solve $(\sigma_{color='red'} Boats)$

bid	bname	Colour
102	Interlake	Red
104	Marine	Red

Now

$(\sigma_{colour='red'} Boats) \bowtie Reserves$

bid	bname	Colour	sid	day

Now

$(\sigma_{colour='red'} Boats) \bowtie Reserves \bowtie Sailers)$

Sid	Sname	rating	age	bid	day	bname	Colour

Now

$\pi_{Sname}((\sigma_{color='red'} Boats) \bowtie Reserve \bowtie Sailers)$

Sname

Date: __/__/20

Method 2:

$\pi_{sname}(\pi_{sid}((\pi_{bid}\,\sigma_{color='red'}\,Boats) \bowtie Res) \bowtie Sailers)$

-> first we solve $(\pi_{bid}\,\sigma_{color='red'}\,Boats)$

bid
102
104

Now $(\pi_{bid}\,\sigma_{color='red'}\,Boats) \bowtie Reserves$

bid	Sid	day

~~Now $(\pi_{bid}\,\sigma_{color='red'}\,Boats) \bowtie$~~

Now $\pi_{sid}((\pi_{bid}\,\sigma_{color='red'}\,Boats) \bowtie Reserve)$

Sid

Now

$(\pi_{sid}((\pi_{bid}\,\sigma_{color='red'}\,Boats) \bowtie Reserve) \bowtie Sailers)$

Sid	Sname	rating	age

Now

$\pi_{sname}(\pi_{sid}((\pi_{bid}\,\sigma_{color='red'}\,Boats) \bowtie Reserve) \bowtie Sail$

Sname

Slide-27:

ρ (Tempboats, ($\sigma_{color = 'red' \vee color = 'green'}$ Boats))
π_{sname} (Tempboats $\bowtie$ Reserves $\bowtie$ Sailors)

→ first we solve ($\sigma_{color = 'red' \vee color = 'green'}$ Boats)

bid	bname	color
102	Interlake	Red
103	Clipper	Green
104	Marine	Red

Now
ρ (Tempboats, ($\sigma_{color = 'red' \vee color = 'green'}$ Boats))

Tempboats

bid	bname	color
102	Interlake	Red
103	Clipper	Green
140	Marine	Red

Now
Tempboats $\bowtie$ Reserves

bid	Sid	day	bname	color
103	58	11/12/96	Clipper	Green

Now
Tempboats $\bowtie$ Reserves $\bowtie$ Sailors

Sid	sname	rating	age	bid	day	bname	color
58	rusty	10	35.0	103	11/12/96	clipper	Green

Now π_{sname} (T $\bowtie$ R $\bowtie$ S)

Sname
rusty

Slide - 28:

→ ρ (Tempred, π_{sid} (($\sigma_{color='red'}$ Boats) ⋈ Reserves))

→ ρ (Tempgreen, π_{sid} (($\sigma_{color='green'}$ Boats) ⋈ Reserves))

→ π_{sname} ((Tempred ∩ Tempgreen) ⋈ Sailors)

Sol

first we solve ($\sigma_{color='red'}$ Boats)

bid	bname	color
102	Interlake	Red
104	Marine	Red

Now

(($\sigma_{colors='red'}$ Boats) ⋈ Reserves)

bid	Sid	day	bname	color

Now

ρ (Tempred, π_{sid} ((—) ⋈ R))

Tempred

Sid

Now

($\sigma_{color='green'}$ Boats).

bid	bname	color
103	Clipper	Green

Now

$((\sigma_{color = 'green'}(Boats)) \bowtie Reserves)$

bid	Sid	day	bname	Color
103	58	11/12/96	clipper	green

Now

$\rho(Tempgreen, \pi_{sid}((\sigma_{color = 'green'}(Boats)) \bowtie Reserves))$

Tempgreen

Sid
58

Now

$(Tempred \cap Tempgreen)$

Sid
58

Now

$((Tempred \cap Tempgreen) \bowtie Sailors)$

Sid	Sname	rating	age
58	rusty	10	35.0

Now

$\pi_{sname}((Tempred \cap Tempgreen) \bowtie Sailors)$

Sname
rusty

Slide - 29:

(1) ρ (Tempsids, ($\pi_{sid, bid}$ Reserves) / (π_{bid} Boats))

π_{sname} (Tempsids $\bowtie$ Sailors)

→ first we solve ($\pi_{sid, bid}$ Reserves)

Sid	bid
22	101
58	103

Now (π_{bid} Boats)

bid
101
102
103
104

Now ($\pi_{sid, bid}$ Reserves) / (π_{bid} Boats).

Sid

ρ (tempsid, ($\pi_{sid, bid}$ Reserves) / (π_{bid} Boats))

tempsid

Sid

Now (Tempsid ⋈ Sailors)

Sid	Sname	rating	age

Now

π_{sname} (Tempsid ⋈ Sailors)

Sname

(2) ρ(Tempsids, ($\pi_{sid, bid}$ Reserves) / π_{bid} ($\sigma_{bname = 'Interlake'}$ Boats))

π_{sname} (Tempsids ⋈ Sailors)

Sol

→ ($\pi_{sid, bid}$ Reserves)

Sid	bid
22	101
58	103

Now

($\sigma_{bname = 'Interlake'}$ Boats)

bid	bname	color
101	Interlake	Blue
102	Interlake	Red

Now π_{bid} ($\sigma_{bname = 'Interlake'}$ Boats)

bid
101
102

Now $(\pi_{sid, bid} \cdot Reserves) / \pi_{bid}(\sigma_{bname='Interlake'} Boats)$

Sid

Now

$\rho(TempSids, (\pi_{sid, bid} Reserves) / \pi_{bid}(\sigma_{bname='Interlake'} Boats))$

TempSids

Sid

Now

$TempSids \bowtie Sailors$

Sid	Sname	rating	age

Scanned with CamScanner

Question: 03

A) File Organization & Indexing:

(1) Ans: Following are three alternative file organizations.

a) Heap files: Suitable when typical access is file scan of all records.

b) Sorted Files:
- → best for retrieval in search key order.
- → also good for search based on search key.

c) Indexes:
Organize data via trees or hashing.

(2) Ans: **Index:** A database index is a disk-based datastructure that improves the speed of data retrieval.

Indexing organize records via trees or hashing.

(3): (1) Selection of form field <OP> constant
(2) Equality Selections (op is =)
(3) Range selections (op is one of <, >, =>, =<, BETWEEN)
(4) One common n-dimensional index: R-tree

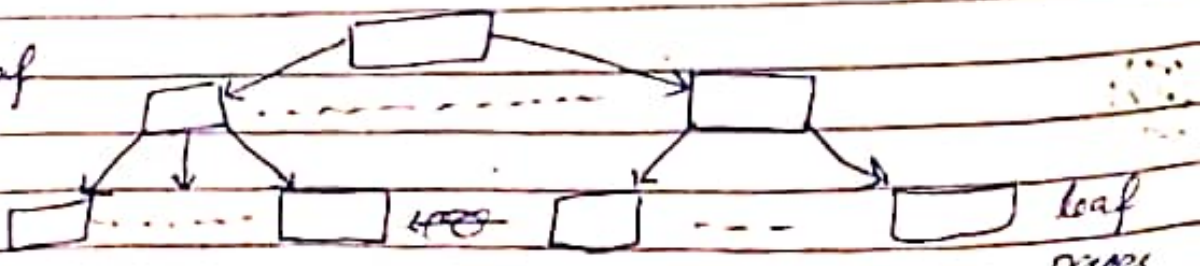
(4) (1) data record with key value K.
(2) <K, rid of data record with search key value K>
(3) <K, List of rids of data records with search key K>

(5) * Good for equality selections.
→ Index is collection of buckets.
Bucket = primary page plus zero or more overflow page.

→ Hashing Function h:
- h(r) = bucket in records r belongs.
- h looks at search key fields of r.

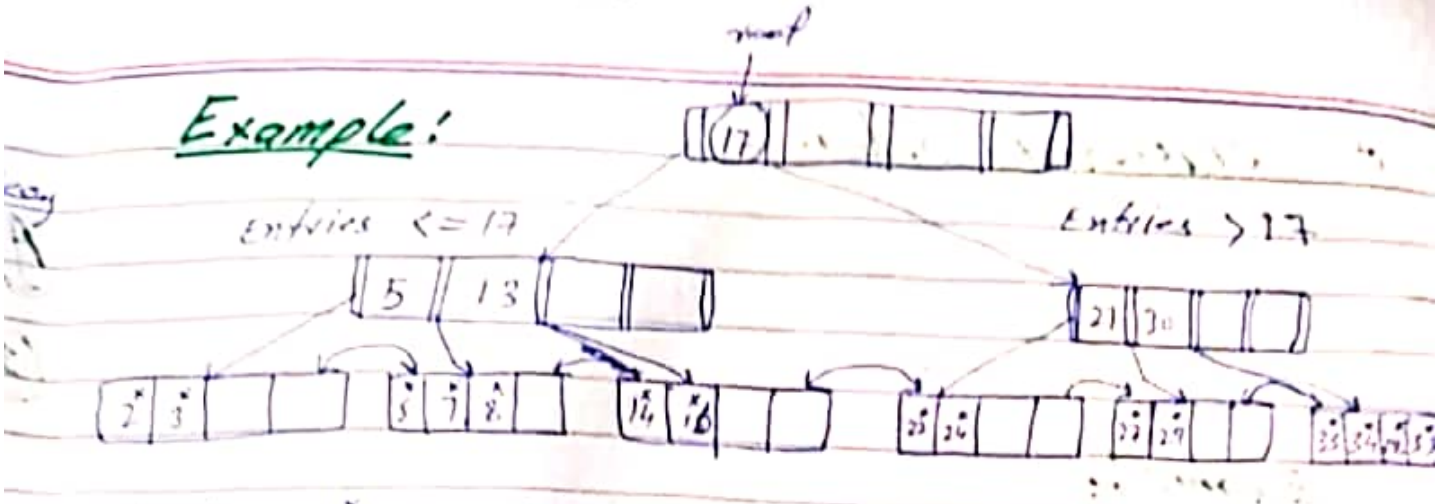
* If alternative (1) is used, the bucket contains the data records; otherwise, they contain <key, rid of r> or <key, rid-list> pairs.

(6) The B+ tree is a balanced binary search tree. It follows multi-level index format.



* leaf pages contains data entries, and are chained (prev and next).
* Non leaf pages contains index entries and direct search

Example:



- Find 28^* ? 29^* ? All $> 15^*$ and $< 30^*$
- Insert/Delete: Find data Entry in leaf, then change it. Need to adjust parent sometimes.
  - → And change sometimes bubbles up the tree.

(7) I/O Cost operations: The table shows diff

the number of data pages, Number of record per page, Fanout of B-tree, and Time required for quality search for different file organization methods, for different operations.

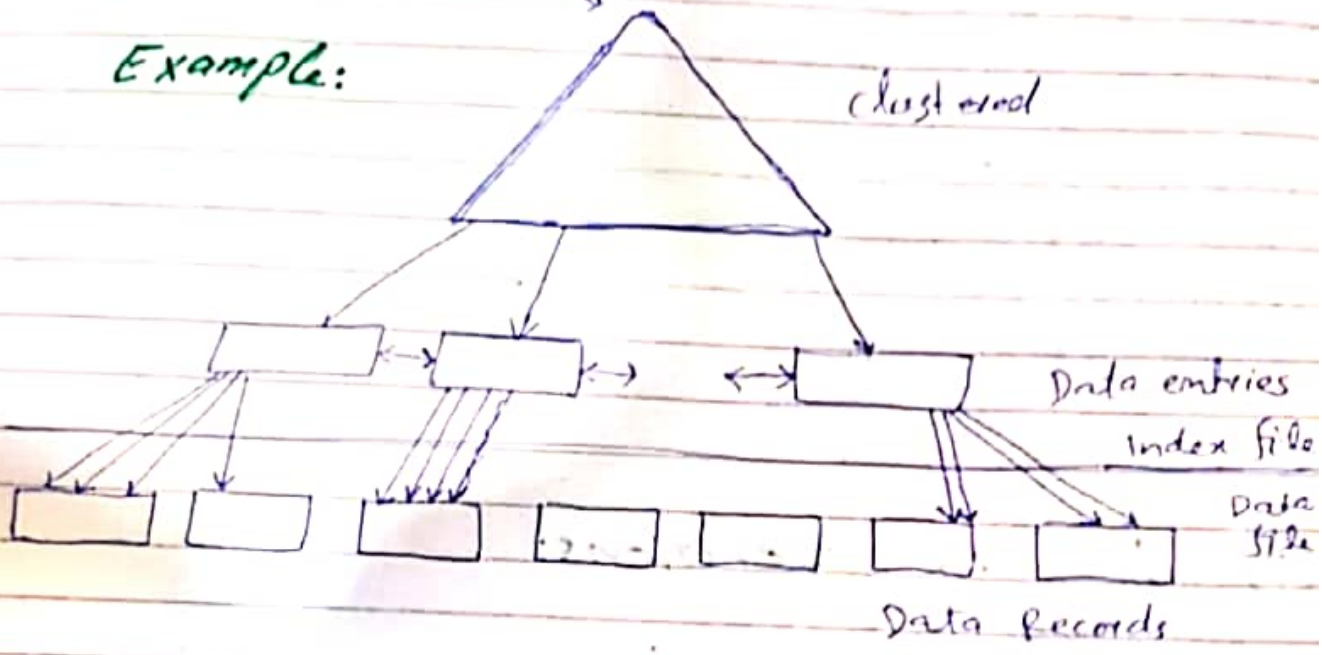
To scan all files records, Heap file and sorted file needs B number of pages, and clustered and unclustred Tree needs 1.5B number of pages while Hash index needs 1.25B number of pages.

Similar things can be interpreted from the table for Get all in Sorted form, Equality search, Range Search, Insert and delete.

(8) clustered index:

If order of data stored is the same as or 'close to' order of index data entries, then called clustered index.

Example:



B.) Query Processing joins and Sortin

1) The dbms must sort the ~~Query~~ to the data to run the Query fo

2) Different techniques used by DBMS ca
1) 2-way sort
2) Two-way External merge sort
3) General external merge sort.
4) Internal sort
5) Quick Sort
6) Heap sort

3) Ans: Scenario: Table to be sorted has B+ tree index on sorting columns.

Idea: Can retrieve records in order by traversing leaf pages.

Case to be considered:
→ B+ tree is clustered Good idea!
→ B+ tree is not clustered Bad idea.

4) To sort a file with N pages using B buffer pages:
- pass 0: use B buffer pages produce [N/B] sorted runs of B pages each.
- pass 1, 2, ... etc.: merge B-1 runs.

5): Four join algorithms:
1) Hash-join
2) Sort merge join (R ⋈ S)
3) Simple Nested loop join
4) Index Nested loop join

6) Following are three nested loop joins:
1) Block Nested loop join:
Use one page as an input buffer for scaning the inner S, one page as the output buffer and use all remaining pages to hold block of outer R.
→ for each matching tuple r in R-block, s in S-page, add ⟨r, s⟩ to result.

2) Simple Nested loops join:

→ For each tuple in outer relation relation R, we scan the entire inner relation S.

→ Page oriented Nested loops join: for each page of R, get each page of S, and write out matching pairs of tuples <r, s> where r is in R-page and s is in S-page

3) INdex Nested loop join:
- If there is an index on the join column of one relation (say S), can make it the inner and exploit the index.
- For each R tuple, cost of probing S index is about 1.2 for has index, 2.4 for B+ tree

) Sort-Merge Join: Sort R and S on the join column, then scan them to do a merge, and output the result tuple.
R is scanned once, each S group is scanned once per matching R tuple.

→ Advance scan of R until current R-tuple >= current S tuple, then advance scan of S until current S-tuple >= current R tuple: do this until current R = current

S tuple.

8) **Hash Join:** Partition both relations using hash function h: R tuples in partition i will only match S tuples in partition i. Read in a partition of R, hash it using h2 (<> h!). Scan matching partition of S, Search for matches.

(C) Query Optimization:

1)

Hash join Method is faster.

2)

Access path ~~is a path a Query tree traverse the table.~~ is a method of retrieving tuples

3)

a tree index matches terms that involve only attributes in a prefix of the search

4).

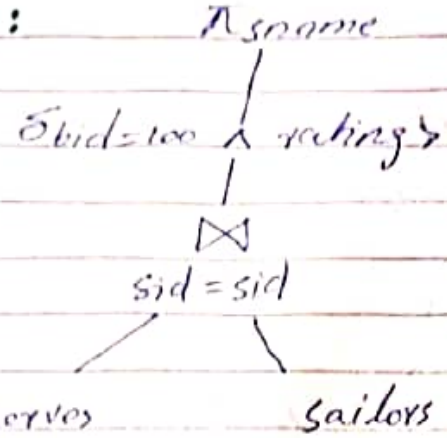
A hash index matches terms that has a term attribute = value for every attribute in the search key of the index.

5)

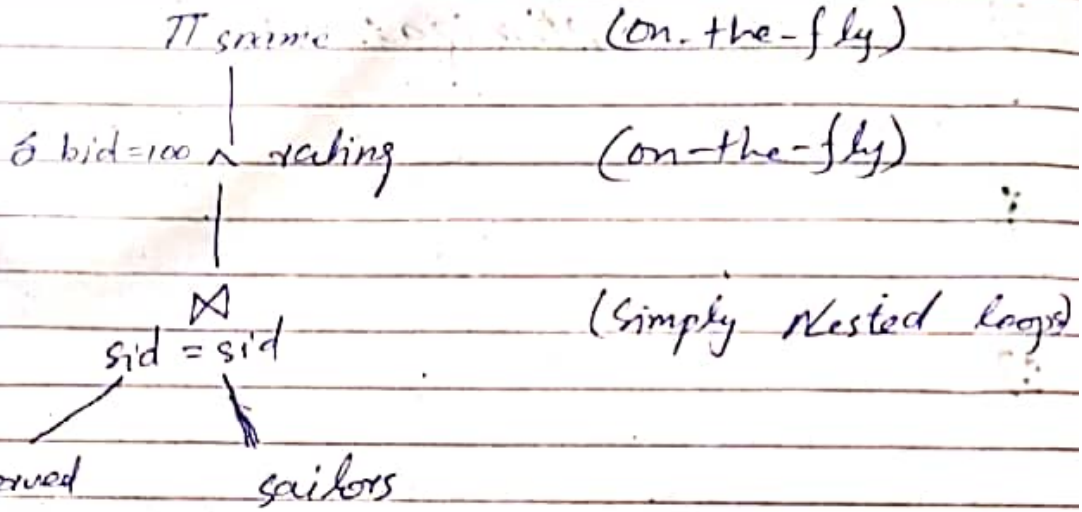
A B+ tree index on day can be used; then, bid=5 and sid=3 must be checked for each retrieved tuple.

A hash index on <bid, sid> could be used; day <8/9/94 must then be checked.

(6) RA tree:



Plan:



Cost:

500 + 500 * 1000 I/Os

(7) No index: