1. unclustered, height = 3

① search down to find the correspondig key : 4 I/O, check if they ~~same~~ care. same with new entry

② at most 6 entry in same key : 6 I/O (unclusterell)

③ Add the new entry : 1 I/O (chache empty, page already in)

④ update the node : 1 I/O.

⇒ 12 I/O

clustered, height = 5

① Search : 6 I/O

② find the entry : 1 I/O (clustered)

③ Add new entry : 1 I/O.

④ update node : 1 I/O

⇒ 10 I/O

# CS150A Homework 2 – Writing

School of Information Science and Technology
October 29, 2022

October 29, 2022

unclustered : 1 I/O per record

clustered : 1 I/O per page of record

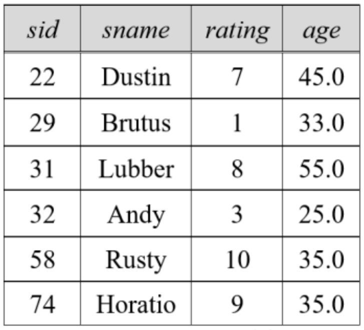
## 1 B+ tree Refinement (10 pts)

How many I/Os would it cost to insert an entry into our table if we had a height 3, unclustered alternative 3 B+ tree in the worst case? Assume that the cache is empty at the beginning and there are at most 6 entries with the same key. And what is the answer for a height 5, clustered alternative 3 B+ tree in the best case? Assume each page is at most 2/3 full.

## 2 Relational Algebra (40 pts)

As shown in the following figures, there are three instances corresponding to three relations: Sailors, Reserves and Boats.

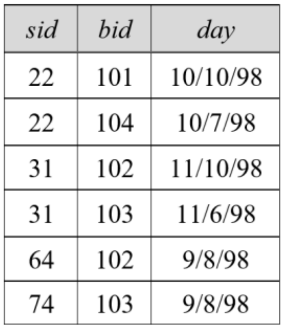
S



| sid | sname | rating | age |
|---|---|---|---|
| 22 | Dustin | 7 | 45.0 |
| 29 | Brutus | 1 | 33.0 |
| 31 | Lubber | 8 | 55.0 |
| 32 | Andy | 3 | 25.0 |
| 58 | Rusty | 10 | 35.0 |
| 74 | Horatio | 9 | 35.0 |

Fig. 1. An instance of Sailors

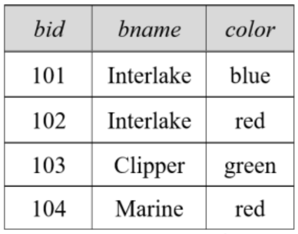
R



| sid | bid | day |
|---|---|---|
| 22 | 101 | 10/10/98 |
| 22 | 104 | 10/7/98 |
| 31 | 102 | 11/10/98 |
| 31 | 103 | 11/6/98 |
| 64 | 102 | 9/8/98 |
| 74 | 103 | 9/8/98 |

Fig. 2. An instance of Reserves

B



| bid | bname | color |
|---|---|---|
| 101 | Interlake | blue |
| 102 | Interlake | red |
| 103 | Clipper | green |
| 104 | Marine | red |

Fig. 3. An instance of Boats

projection already ensure distinct.

Use relational algebra to describe the following queries.

1. Find the names of sailors who reserved boats after 10/7/98.

   $\pi_{sname}(\sigma_{days > 10/7/98}(Sailors \bowtie_{sid=sid} Reserves.))$

2. Find the age of sailors who have reserved boat Interlake with blue color.

   $\pi_{age}(Sailors \bowtie_{sid=sid} (Reserves \bowtie_{bid=bid} (\sigma_{bname = Intertake}(\sigma_{color=blue}(Boats)))))$

3. Find the name of boats which have been reserved by sailors rating 7 or higher.

   $\pi_{sname}(Boats \bowtie_{bid=bid} (Reserves \bowtie_{sid=sid} (\sigma_{rating \geq 7}(Sailors))))$

4. Find the color of boats which once reserved by sailors Dustin on 10/10/98.

   $\pi_{color}(Boats \bowtie_{bid=bid} (\sigma_{day=10/10/98}(Reserves \bowtie_{sid=sid} (\sigma_{sname=Dustin}(Sailors)))))$

5. Find the name of sailors who didn't reserved any boat named Marine.

   $\pi_{sname}(Sailors) - \pi_{sname}(Sailors \bowtie_{sid=sid} (Reserves \bowtie_{bid=bid} [\sigma_{bname=Marine}(Boats))))$

## 3 Joins (10 pts)

Determine whether each of the following statements is True or False.

a. Block Nested Loops join will always perform at least as well as Page Nested Loops Join when it comes to minimizing I/Os. T $[R] + \lceil \frac{R}{B-2} \rceil * [S] \Rightarrow R + [R] * [S])$

b. Grace hash join is usually the best algorithm for joins in which the join condition includes an inequality. F

require equal

## 4 Query Optimization (40 pts)

Consider two relations R(a, b) and S(a), with 1000 tuples and 500 tuples respectively. We have an index on R.a with 50 unique integer values uniformly distributed in the range [1, 50], and an index on S.a with 25 unique integer values uniformly distributed in the range [1, 25]. We do not have an index on R.b.

index = clustered

Use selectivity estimation to estimate the number of tuples produced by the following queries.

1. SELECT * FROM R

   1000

2. SELECT * FROM R WHERE a = 99

   $1000 \times \frac{1}{50} = 20$

3. SELECT * FROM R WHERE b = 99

   no index on b.

   default selectivity : $\frac{1}{10}$

   $1000 \times \frac{1}{10} = 100$

$a$ : uniform$[1,50]$ $a \le 24 \rightarrow a < 25$

4. SELECT * FROM R WHERE a ≤ 24

$\text{Selectivity} = \frac{24}{50} + \frac{1}{50} = \frac{1}{2}$ $1000 \times \frac{1}{2} = 500$

5. SELECT * FROM R WHERE b ≤ 10

$\text{selectivity} = \frac{10}{50} = \frac{1}{5}$ $1000 \times \frac{1}{5} = 200$

6. SELECT * FROM R WHERE NOT a ≤ 24

where not $a \le 24 \rightarrow a \ge 25$

$\text{selectivity} = \frac{26}{50} + \frac{1}{50} = \frac{27}{50}$ $1000 \times \frac{27}{50} = 540$

7. SELECT * FROM R WHERE a ≤ 24 AND b ≤ 10

$\text{selectivity}(a \le 24) * \text{selectivity}(b \le 10) = \frac{1}{2} \times \frac{1}{5} = \frac{1}{10}$

$1000 \times \frac{1}{10} = 100$

8. SELECT * FROM R WHERE a ≤ 24 OR b ≤ 10

$\text{selectivity}(a \le 24) + \text{selectivity}(b \le 10) - \text{selectivity}(a \le 24) * \text{selectivity}(b \le 10) = \frac{1}{5} + \frac{1}{2} - \frac{1}{10} = \frac{3}{5}$

$1000 \times \frac{3}{5} = 600$

9. SELECT * FROM R WHERE a = b

$\text{selectivity} = \min\left(\frac{1}{50}, \frac{1}{10}\right) = \frac{1}{50}$ $1000 \times \frac{1}{50} = 20$

10. SELECT * FROM R, S WHERE R.a = S.a

$1000 \times \frac{1}{50} + 500 \times \frac{1}{50} = 30.$