UCLA
Computer Science Department
Course: CS143A

Instructor: Carlo Zaniolo
TAs: Mingda Li and Jim Wang
Winter 2020

Student Name and ID: Junhong Wang 504941113

CS143 MIDTERM EXAM: Closed Book, 2 Hours

- **You are urged to write out your solution steps so you can earn partial credits even if final answer is not perfect.**
- **If you need to make some assumption to solve a problem, please write your assumptions clearly in your answer.**
- **Simplicity and clarity of your solutions are important. You might get zero points if your solution is far more complicated than needed, or if we cannot understand it.**
- **Attach extra pages as needed. Write your name and ID on them.**
- **Please write neatly.**

Problem	Score	
A	(30%)	~~0~~ 30
B	(20%)	11
C	(20%)	20
D	(30%)	30
Total	(100%)	91

Problem A. Searching and Joining— 30 points

We store the following file in blocks having size 4096 bytes.

`partSup(pid char(20), Supplier char(30))` 50 bytes / tuple

Please, answer the ~~following~~ three questions assuming that our relation has 2 millions tuples, which are stored unspanned. Also there is a sparse B+ index on pid and there is no other index: Here are the questions:

2,000,000 tuples

1. How many blocks are needed to store the whole relation?

2. Assuming that the B+ tree has two levels, how many blocks must be retrieved to execute the following query:

   `SELECT Supplier FROM partSup WHERE pid =' XBC98629450000000000'`

3. Consider now the following query that find pairs of parts supplied by the same supplier:

```
SELECT  PS1.pid, PS2.pid
FROM partSup  as  PS1 PS2
WHERE   PS1.Supplier = PS2.Supplier  and  PS1.pid < PS2.pid
```

What join method will the system use to support this query, and what will be its estimate cost in terms of blocks fetched from disk assuming that only three main-memory blocks can be used to compute the join?

81
50 | 4096
400
96
50
46

24691
81 | 2000000
162
380
324
560
486
740
729
159

1. 1 block is 4096 bytes.

   1 tuple is 20 + 30 = 50 bytes

   So $\lfloor \frac{4096}{50} \rfloor = 81$ tuples / block (∵ unspanned)

   We have 2,000,000 tuples

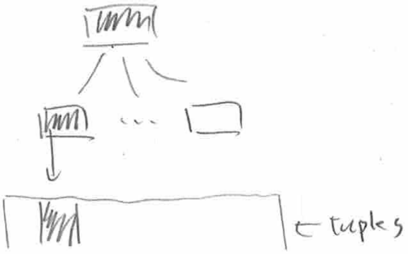
   Thus we need $\lceil \frac{2000000}{81} \rceil = \boxed{24692 \text{ blocks}}$

2. We need to access 1 block at the first level,

   1 block at the second level of the B+ tree.

   Finally we access 1 block ~~of~~ leaf node pointer is pointing to. Thus, we need to retrieve

   $\boxed{3 \text{ blocks}}$.



3. The system will use block nested join.

   One memay block is used to store PS1.

   Another memay block is used for PS2.

   The other memory block is used for output.

   Note we are joining two same relation.

   Thus $\lceil \frac{24692}{3-2} \rceil \cdot (24692 - 1) + 24692$

   $= 24692 \cdot 24691 + 24692$

   $= 24692 (24691 + 1)$

   $= \boxed{(24692)^2 \text{ block fetches}}$

137) 14706 = 107
137
1006
959

Problem B. B+ trees — 20 points

Same relation as in Problem A, with 2 millions tuples stored unspanned on blocks of size 4096 bytes We have a sparse index on pid organized as a B+ tree. **Pointers take 10 bytes**. Answer the following questions.

$30n - 20 \le 4096$
20
4116
4
4096
30) 4116 = 137
30
111
90
216
6

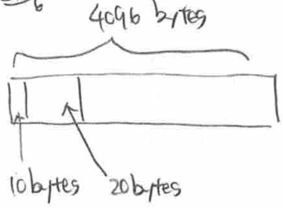
1. What is the minimum number of nodes needed for the B+ tree?

136) 2000000 = 14705
136
640
544
960
952
800
680
120

2. What is the maximum number of nodes needed for the B+ tree, in the worst case scenario (i.e., the situation which requires most blocks).

14706
108
14814

1.



$10n + 20(n-1) \le 4096$

$n \le 137. \dots$

2) 137 = 68
12
17
16

Every node has at most 137 pointers ✓

(Note leaf node has at most 136 pointers)

$\left\lceil \frac{2000000}{136} \right\rceil = 14706$

$\left\lceil \frac{14706}{137} \right\rceil = 108$

$\left\lceil \frac{108}{137} \right\rceil = 1$

Thus minimum number of nodes is $\cancel{14706} + 108 + 1 = \boxed{14815}$ −3

68) 2000000 = 29411
136
640
612
280
272
80
68
120
68
52

54
7
61

69) 29412 = 426
276
181
138
432
414
18

69) 427 = 6
414
13

36
5
41

29412
427
29839
18
29847

2. From 1, every node has at most 137 pointers. (Note leaf node has at most 136 pointers)

Thus every node has at least $\left\lceil \frac{137}{2} \right\rceil = 69$ pointers (Note leaf node has at least 68 pointers and root has at least 2 pointers)

$\left\lceil \frac{2000000}{68} \right\rceil = 29412$ $\left\lceil \frac{7}{69} \right\rceil = 1$ −6

$\cancel{\left\lceil \frac{29412}{69} \right\rceil} = 427$

$\left\lceil \frac{427}{69} \right\rceil = 7$

Thus maximum number of nodes is

$29412 + 427 + 7 + 1 = \boxed{29847}$

Problem C. SQL — 20 Points

Given the table: `taken(StNo, CourseID, Year, Quarter, Sec, Grade, Remarks)`: write an SQL query to find the students who got a grade less than class average in 7 or more classes they took —a class is identified by (`CourseID, Year, Quarter, Sec`) and `Grade` in `taken` is of type numeric. In your answer, the depth of nesting of sub-queries should not exceed 2.

```
SELECT AVG(Grade)
FROM taken
GROUP BY class
```

```
SELECT StNo
FROM taken AS T1
WHERE Grade < (
    SELECT AVG(Grade)
    FROM taken AS T2
    WHERE T1.Course ID = T2.Course ID
          AND T1.Year = T2.Year
          AND T1.Quarter = T2.Quarter
          AND T1.Sec = T2.sec
)
GROUP BY StNo
HAVING COUNT(*) >= 7;
```

Problem D. Potpourri: 30 points (5 Points per question):

Please answer the following questions. You must give a YES—NO answer to questions D1–D4 and if your answer is YES you must also **write the equivalent RA expression**.

D1 Can the intersection of relations **R(A,B)** and **S(A,B)** be expressed using only natural joins?

R∩S
R.A = S.A
R.B = S.B

YES, $R \bowtie S$ ✓

D2 Can the intersection of relations **R(A,B)** and **S(A,B)** be expressed using the set difference operator?

YES, $R - (R-S)$ ✓

$$\overset{R}{\begin{array}{|cc|}1 & 2\\ 2 & 3\\ 4 & 4\end{array}} - \overset{S}{\begin{array}{|cc|}1 & 2\\ 2 & 3\\ 5 & 6\end{array}} = 4\ 4$$

D3 Can the intersection of relations **R(A,B)** and **S(A,B)** be expressed using the cartesian product and projection operators?

NO ✓

D4 Can the intersection of relations **R(A,B)** and **S(A,B)** be expressed using the cartesian product, selection and projection operators?

YES, $\Pi_{R.A,\, R.B}(\sigma_{R.A=S.A \wedge R.B=S.B}(R \times S))$ ✓

D5 A relation R is indexed on its candidate key using an extendible hashing: Is this index (i) dense, (ii) sparse or (iii) it could be either way? You must explain the reason for the answer you selected.

(i), because we are using hash mapping, every record must have a key that maps to it. ✓

D6 For the extendible hashing on R described in D5: does the structure of its directory and the number of the buckets it uses depends on (i) the order in which the tuples in R have been inserted, or (ii) they only depend on the values those tuples? You must explain the reason for the answer you selected.

(ii), Extendible hashing works as follows:

1. convert the key to binary representation
2. Place the (key, value) pair based on the value of ①
3. If the bucket is overflowed, extend the bucket by splitting it.

Thus, the structure of the hash table only depends on the key of the tuple. The order of insertion does not matter. ✓