UCLA
Computer Science Department

Instr: C. Zaniolo
TAs: Jiaqi Gu, Jia Teoh, Zijun Xue

Your Name and ID: Kevin Qian 604574464

CS143, Fall 2017: MIDTERM EXAM: Closed Book, 110 minutes.

Please Read:

- Attach extra pages as needed. Write your name and ID on the extra pages.
- If you need to make any assumptions to solve a problem, please write you assumptions clearly in your answer.
- Simplicity and clarity of your solutions will count. You may get as few as 0 point for a problem if your solution is far more complicated than necessary, or if we cannot understand your solution.
- Please write neatly.

Problem	Score	
1	(40%)	34
2	(40%)	38
3	(20%)	20
Total	(100%)	92

Extra Credit (9 Points) 7

Midterm Score: 99

Another A1 sol:
SELECT SupplierNo FROM warehouse
EXCEPT
SELECT w1.SupplierNo FROM warehouse w1
WHERE w1.Price ≤ MIN(SELECT w2.Price FROM warehouse w2 WHERE w2.Supplier <> w1.Supplier AND w2.PartNo = w1.PartNo) } call this <subquery>
COUNT(<subquery>) =0 OR
GROUP BY w1.SupplierNo HAVING COUNT(w1.*) ≥ 2;

Another A2 sol:
change "w1.price ≤ MIN(..." to "w1.Price < MIN(..."

Problem 1: 40 points–all questions have the same weight

The relation warehouse(PartNo, SupplierNo, Price) describes the suppliers for each part, along with the price they charge for the part. To cut inventory, the manager of our warehouse wants to eliminate non-competitive suppliers. A supplier is competitive if either (i) he is the only supplier of some part, or (ii) he supplies at least two parts at a minimum cost (however the supplier could share this minimum cost for the with other suppliers). All the others are non-competitive suppliers.

A1: Write an SQL query to find all non-competitive suppliers.

A2: Let us now modify the definition of competitive supplier as follows: A supplier is competitive if either (i) he is the only supplier of some part, or (ii) he supplies at least two parts at a strictly minimum cost (i.e., ties are not viewed as min). All the others are non-competitive suppliers. Please write an SQL query to find the non-competitive suppliers according to this modified definition

A3: Write an SQL statement to delete from the warehouse all the part-suppler tuples where the supplier is non-competitive (as defined in A2):

A4: Is query A2 expressible in basic RA, which does not have aggregates? (Hint: re-express "minimum cost" and "least cost" without using aggregates.)

ignored (wrong anyways)

ANOTHER sol for A1, A2, see above

A1.
```
SELECT SupplierNo FROM warehouse
WHERE SupplierNo NOT IN
(SELECT w1.SupplierNo FROM (warehouse AS w1) LEFTOUTER JOIN (warehouse AS w2) ON
 (w1.PartNo = w2.PartNo AND w1.SupplierNo <> w2.SupplierNo)
 WHERE (w2.SupplierNo IS NULL) OR w1.SupplierNo IN (
   SELECT a.SupplierNo FROM warehouse AS a
   WHERE NOT EXISTS (
     SELECT b.SupplierNo FROM warehouse AS b
     WHERE a.SupplierNO <> b.SupplierNo AND a.PartNo = b.PartNo
     AND a.Price > b.Price) /* ① */
   GROUP BY a.SupplierNo HAVING COUNT(a.*) ≥ 2 ));
```

* actually the ONLY SUPPLIER part is REDUNDANT, we can remove it (as the later NOT EXIST secure this already)

(i) only supplier; (ii) ≥2 min price

most creative solution so far...

* technically you might end up w/ extra duplicates of SNo but no difference

A2. Simply modify ① in (A1) to " AND a.Price ≥ b.Price)" would do the trick (i)✓ (ii)✓ like

-2 A3. DELETE FROM warehouse WHERE SupplierNo NOT IN (/* subquery from A2 */)

'competitive' only supplier (this part actually redundant, discardable)

-4 A4. Yes.

$\pi_{SupplierNo}(warehouse) - (\pi_{SupplierNo}(warehouse) - \pi_{S.SupplierNo}(\sigma_{S.SupplierNo <> T.SupplierNo \wedge S.PartNo = T.PartNo}(\rho_S(warehouse) \times \rho_T(warehouse))))$ (i) -4

Let R be $(\pi_{SupplierNo, PartNo}(warehouse) - (\pi_{A.SupplierNo, A.PartNo}(\sigma_{A.PartNo=B.PartNo \wedge A.SupplierNo <> B.SupplierNo \wedge A.Price \geq B.Price}(\rho_A(warehouse) \times \rho_B(warehouse)))))$

b/c Suppliers with non-unique PNo can still have unique p

$-(\pi_{M.SupplierNo}(\sigma_{M.SupplierNo=N.SupplierNo \wedge M.PartNo <> N.PartNo}(\rho_M(R) \times \rho_N(R))))$ (ii)✓

≥2 min supplier *

Problem 2, 40 Points. All questions have the same weight.

We have 1 million employee tuples with unique key Eno. The length of each tuple is 50 bytes. These are stored as fixed-length unspanned records in a file consisting of blocks of size 2048 bytes.

On this file, we build a **sparse** index on Eno. The index is is organized as a B+ tree, where each key takes 18 bytes and each pointer takes 22 bytes (the leaf nodes are chained together as in the textbook). The B+ tree blocks contain 2048 bytes.

A. How many blocks are needed to store all the records in the file.

B. Compute the blocks used at each level of the B+ tree, for the best case and the worst case.

C. Is this index a primary index or a secondary one?

D. A query asking for the records of all employees whose Eno falls in a certain range returns 800 records. Estimate the number of pages accessed to retrieve this query in the worst case situation.

A. Each Block stores $\lfloor 2048/50 \rfloor = 40$ tuples. Thus We need $\lceil 1{,}000{,}000/40 \rceil = 25000$ blocks

B. Since using sparse index, we do have 25000 (per per block) leaf key entries. Max Ptr No: $N = 1 + \lfloor \frac{2048-22}{18+22} \rfloor = 51$

Min Ptrs { Leaf : $\lceil \frac{N+1}{2} \rceil = 26$; Non-Leaf : $\lceil \frac{N}{2} \rceil = 26$

WORST: Thus: 1st Level: $\lfloor \frac{25000}{26-1} \rfloor = 1000$ blocks

WORST: 2nd Level: $\lfloor \frac{1000}{51} \rfloor = 19$ blocks (next block ptr); 3rd Level (Root) $19 < 26$, thus 1 block

BEST: 1st Level: $\lceil \frac{25000}{51-1} \rceil = 500$ blocks

2nd Level: $\lceil \frac{500}{51} \rceil = 10$ blocks

3rd Level/Root: $10 < 51$, thus 1 block

−2

C. This index is a Primary index (sparse index MUST BE primary) ↙ clustering

D. Worst case has tree height 3. Thus a basic 3 blocks for searching is needed

800 records are spanned on MAX $1 + \frac{800}{40} = 21$ blocks (continuous due to sparse index) (adding 1 so that maybe some page are NOT completely filled with the requested tuples)

21 block ptrs could span on 2 different tree leaf nodes. (1 more node)

Thus in total $3 + 1 + 21 = 25$ pages will be accessed

Problem 3, 20 Points. All questions have the same weight.

A file is indexed using an extensible hash table which applies the function h(n) = n mod 256 to the search keys of the first seven records in the file. The records are stored in buckets that contain 3 records.

The keys of the first seven records are: {106, 115, 916, 126, 16, 15, 31} which produce the following hash values:

```
h(106) = 106  = 01101010
h(115) = 115  = 01110011
h(916) = 148  = 10010100
h(126) = 126  = 01111110
h(16) =  16   = 00010000
h(15) =  15   = 00001111
h(31) =  31   = 00011111
```

Assuming that that the leftmost i bits of the hash key are used for a bucket address table of size 2^i, please answer the following questions:

A. What is the size of the hash table after all the seven items above are inserted? (Explain how that can be determined without drawing the actual table.)

B. As more records in the file are are indexed using this extensible hash function, could there be situations where overflow buckets must be used? If your answer is yes, give the search keys of three records, that once they are added to those first seven, will cause the creation of an overflow bucket.

A. The size is 4 (for table entry only). Notice that we have 3 entries prefixed by 01, 3 by 00 and 1 by 10. Thus they can perfectly satisfy the hashing bucket size requirement by just using the leftmost 2 bits, yielding $2^2 = 4$ entries in hash table.
The number of buckets would be 3, though. (as seen above in same 2 bit values)

B. Yes. We can just supply 3 records that are same as one of the already inserted entries (4 would have same hash value no matter how many bits used, thus must overflow)
Thus we can just insert 15, 15, 15 to cause the overflow bucket.

Extra Credit. Each question 3 Points

Consider the relations `R(A, B)` and `S(B, C)` `R` contains $n > 0$ tuples and `S` contains $m > 0$ tuples; NULLs are not allowed in either relation.

A • What is the minimum number of tuples that may result from the natural join of these two relations:?

B • What is the maximum?

C • How will your previous answers change if instead of natural join we take the full outer join?

A. 0. Suppose none of the R's B values appears in S's B values

B. m·n, Suppose EVERY R's B values appears in S's B values, (as if doing cross product)

C. W/ full outer join, min would become (MAX of m and n)
Max would still be m·n though.

-2.