CUHK(SZ)-CSC3170 Final Exam

Dec 12, 2024

Name:

Student ID:

Seat No:

Note:

1. **DO NOT** open the exam paper until you are instructed to open.
2. Please read all instructions carefully.
3. **This is a <u>closed-book exam</u>. You are allowed 1 page of note sheet that you can write on both sides.**
4. Answer all questions within 120 minutes. **Write your answer to answer book.** Not this paper.
5. If you are not done after 1 hour and 45 minutes, please **stay in your seat** until the end.
6. **NO** electronic devices are allowed, including cell phones and smart watches. Silence your cell phones and place them in your bag.
7. This exam paper has 10 pages in double-sided printing, including the cover page.
8. The student must **NOT** take the exam paper away from the exam venue.
9. *If you get stuck on a question move on and come back to it later.*

1. **Relational Model and SQL**

Consider the following schema for a simple social network. It maintains a collection of users, identified by unique "handles" (short name).

```
User(handle, name, home_city, bio)
Friend(handle1, handle2)
Message(handle, text, from_city)
```

The next few problems will consider queries for answering the following questions:

A. Find the handles of users who wrote messages while in cities other than the one in which they live.
B. Find the handles of users who are friends with someone who is friends with them.
C. Find the names of users who either live in Shenzhen or have written a message from Shenzhen.
D. Find the names of users who both live in Shenzhen and have written a message from Shenzhen.

Note, the friendship is directed, and more like following.

For each relational algebra expression below, indicate which of the questions above (A – D), if any, it answers. If no question matches, then write **NONE**.

(a) (3 points)

$$\delta\left(\pi_{\text{handle1}}\sigma_{\text{handle1=handle3}}\left[\text{Friend} \bowtie \left(\rho_{\text{handle2,handle3}}\text{Friend}\right)\right]\right)$$

Hint: Recall that the ρ operator simply renames the columns of a table.

(b) (3 points)

$$\delta\left(\pi_{\text{handle}}\left[\delta\left(\pi_{\text{handle,from\_city}}\text{Message}\right) - \pi_{\text{handle,home\_city}}\text{User}\right]\right)$$

(c) (3 points)

$$\delta\left(\pi_{\text{name}}\left[\left[\pi_{\text{handle}}\sigma_{\text{home\_city='Shenzhen'}}\text{User}\right] \cup \left[\pi_{\text{handle}}\sigma_{\text{from\_city='Shenzhen'}}\text{Message}\right] \bowtie \text{User}\right]\right)$$

(d) (3 points)

$$\delta\left[\pi_{\text{handle}}\sigma_{\text{from\_city}\neq\text{home\_city}}\left(\text{User} \bowtie \text{Message}\right)\right]$$

For each SQL query below, indicate which of the questions from before (A – D), if any, it answers. If no question matches, then then write **NONE**.

(e) (2 points)

```
SELECT name
FROM User U
WHERE home_city = 'Shenzhen' OR
      EXISTS (SELECT name FROM Message
              WHERE handle = U.handle AND
                    from_city = 'Shenzhen')
```

(f) (2 points)

```
(SELECT name FROM User
 WHERE home_city = 'Shenzhen')
INTERSECT
(SELECT name FROM User U, Message M
 WHERE U.handle = M.handle AND
       from_city = 'Shenzhen')
```

(g) (2 points)

```
SELECT DISTINCT U.handle
FROM User U, Message M
WHERE U.handle = Message.handle AND
      from_city <> home_city
```

(h) (2 points)

```
SELECT F1.handle1
FROM Friend F1, Friend F2
WHERE F1.handle2 = F2.handle2 AND
      F1.handle1 = F2.handle1
GROUP BY F1.handle1
```

(i) (2 points) Consider the following SQL query:

```
SELECT handle, bio, min(home_city)
FROM User NATURAL JOIN Message
GROUP BY handle, from_city
```

Which section of the query is illegal?

A. `handle`

B. `bio`

C. `min(home_city)`

D. `from_city`

(j) (2 points) Consider the following expression in relational algebra:

$$\pi_{\text{from\_city}}\left[\text{Message} \bowtie \left(\sigma_{\text{handle}>'c'}\text{User} - \sigma_{\text{handle}>'d'}\text{User}\right)\right]$$

Can the same result be computed using a simple select-from-where query (i.e., one with no subqueries, grouping, or aggregation)?

A. Yes

B. No

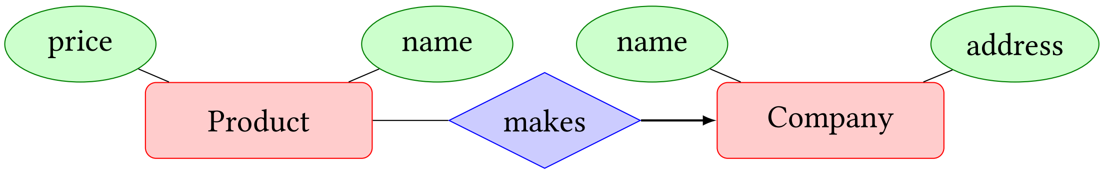
2. **DB Design**

   (a) (2 points) Consider the following SQL schema:

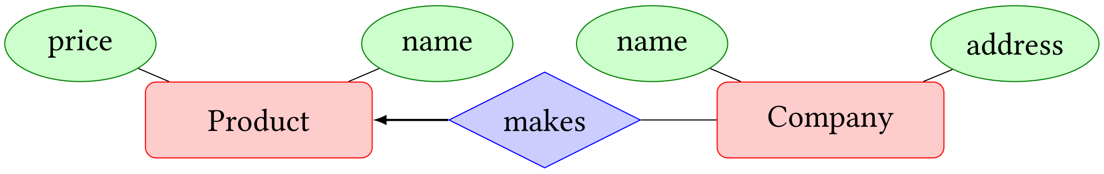
```
CREATE TABLE Company(
    name VARCHAR(100) PRIMARY KEY,
    address VARCHAR(200));

CREATE TABLE Product(
    name VARCHAR(100) PRIMARY KEY,
    price FLOAT,
    made_by VARCHAR(100) FOREIGN KEY REFERENCES Company);
```

   Which of the following E/R diagrams could produce that schema?

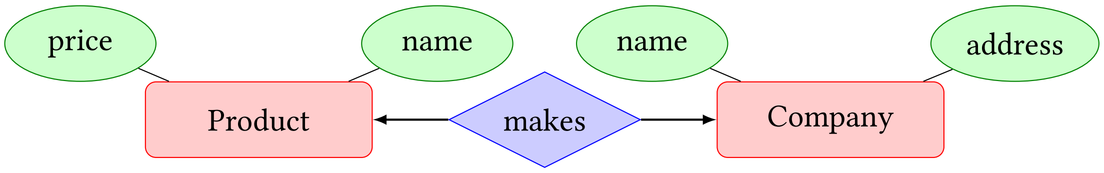
   A.



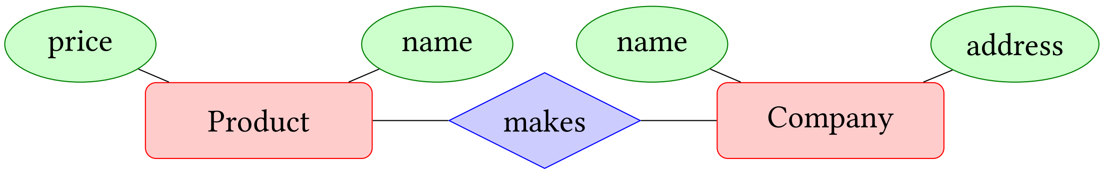
   B.



   C.



   D.



   (b) (4 points) Write down at least two functional dependencies implied by the SQL schema.

   (c) (4 points) You are given a table named `Sales` with the following structure and data: Please give the expected output of executing the following SQL:

```
SELECT
    SalesID,
    SUM(SalesAmount) OVER (PARTITION BY Salesperson ORDER BY
       SaleDate) AS CumulativeSales,
```

SalesID	Salesperson	Region	SalesAmount	SaleDate
1	Alice	East	500	2024-12-01
2	Bob	West	700	2024-12-01
3	Alice	East	300	2024-12-02
4	Bob	West	400	2024-12-02
5	Charlie	East	600	2024-12-03
6	Alice	East	800	2024-12-03

Table 1: `Sales` Table

```
    RANK() OVER (PARTITION BY Region ORDER BY SalesAmount DESC)
       AS SalesRank
FROM
    Sales
ORDER BY
    SalesID;
```

3. **Transaction**

   (a) (5 points) Consider the following schedule involving three transactions T_1 , T_2 , and T_3 :

$$R_1(X), R_2(X), W_1(X), R_3(Y), W_2(X), R_3(X), W_3(Y), W_1(Y)$$

Identify the conflicting operations between transactions in the schedule (do not need to list all of them). Draw the dependency (precedence) graph based on the conflicts you identified and determine whether the schedule is conflict serializable.

4. **Query Planning and Optimization**

In the next few problems, we will compute the cost of the query plan below, which finds the result of the expression:

$$\pi_{\mathrm{e,h}}\big(\sigma_{e<=100}(\mathrm{R} \bowtie \mathrm{S} \bowtie \mathrm{T})\big)$$

over the tables R(e, f), S(f, g), and T(g, h). (Both joins are natural.)

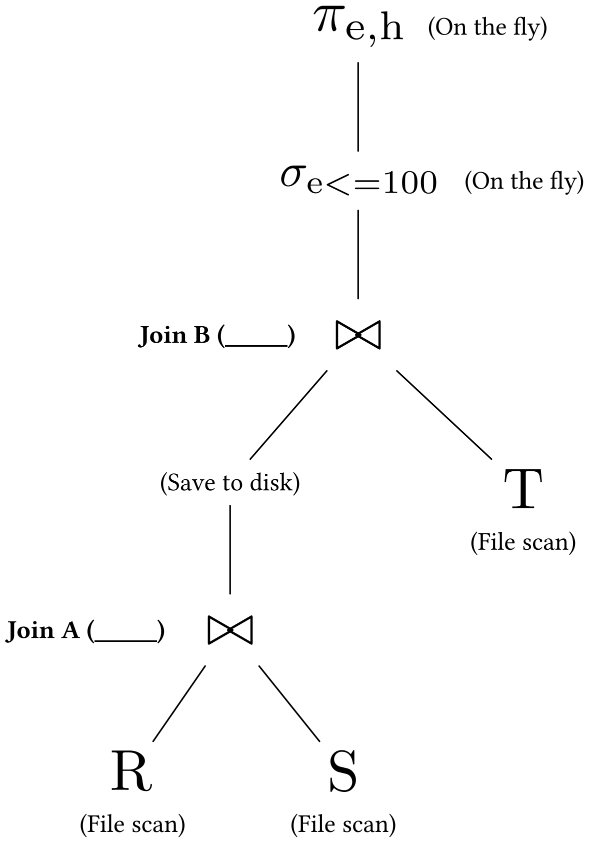
The plan is still *incomplete*, however, because the choice of algorithm for two parts, **Join A** and **Join B**, have not yet been filled in.

You can assume the following statistics about these tables:

Table	# tuples	# pages
R	1,000	100
S	5,000	200
T	100,000	10,000
$\mathrm{R} \bowtie \mathrm{S}$	5,00	20

Furthermore, you can assume the following statistics about their columns:

Column	# distinct	Low	High
R.e	80	1	400



(a) (3 points) Suppose the size of the buffer pool assigned to **Join B** is 22 pages. What is the estimated cost of **Join B** in the plan if we implement it with a block nested loop join?

(b) (3 points) Suppose the size of the buffer pool assigned to **Join A** is 22 pages. What is the estimated cost of **Join A** in the plan if we implement it with a block nested loop join? Do not count in the output cost here

(c) (3 points) What is the estimated cost of **Join A** in the plan if we implement it with a partitioned hash join? Assume the buffer pool size is 22, no recursive partitioning needed, and do not count in the output cost here.

(d) (3 points) What is the estimated cost of **Join B** in the plan if we implement it with a partitioned hash join? Assume the buffer pool size is 22 and no recursive partitioning needed.

(e) (12 points) Suppose the buffer pool sizes for both join operators are 22, and no index can be used. Use the predicate pushdown to optimize the query plan, and choose the suitable hash join algoritm (either block nested loop join or partitioned hash join) based on the cost estimation. Draw the new query plan tree (indicating the join algorithm) [6 points], and report the overall cost of the new plan [6 points].

Hint to (e): When estimating the cost, you can assume all attributes follow uniform distribution and they are independent from each other. Assume the intermediate data are pipelined among different operators, if no otherwise specified (e.g., 'save to disk' means materialization).

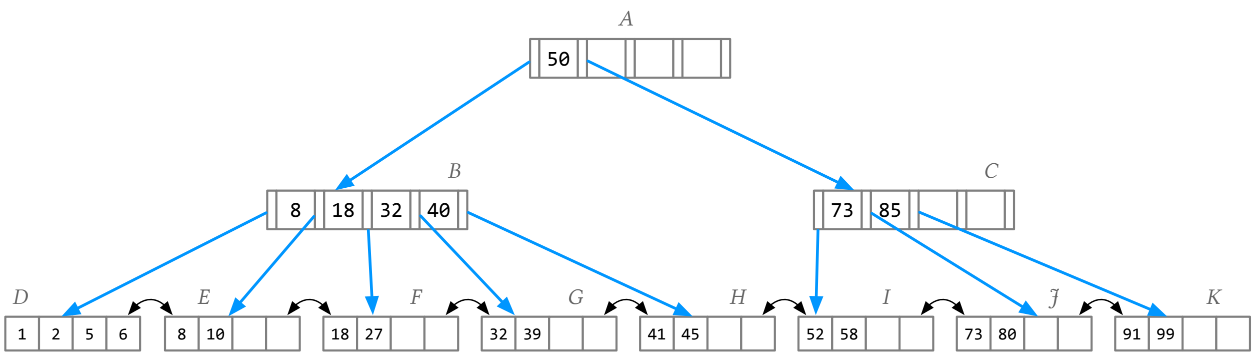


Figure 1: A 5-way search B+tree.

5. **Indexing**

   Consider the following 5-way search B+ tree index (in Figure 1, each non-leaf node can have at most 4 keys and 5 child pointers). When you split a B+tree node, for two new nodes, the left one should be larger than or equal to the right one.

   (a) (3 points) Show the tree that would result from inserting a data entry with key 9 into this tree.

   (b) (5 points) Show the tree that would result from inserting a data entry with key 3 into the original tree.

   (c) (3 points) Show the tree that would result from deleting a data entry with key 6 from the original tree.

   (d) (3 points) Suppose this is a clustered index (and the tuple data are stored within the leaf nodes). Using the original tree (Figure 1), the user wants to query all tuples where the key is less than 70. Compute the cost in terms of page input (i.e., ignore the output cost here), and write down the access order of those pages/nodes using A, B, C ... marked on nodes.

   (e) (2 points) To allow parallelism and avoid errors, we take read/write latches on the nodes of the tree. We want to allow more parallelim via the better latching algorithm introduced in the lecture. Give the latches aquired during the whole process when inserting 81 to the original tree. (Do not need to indicate when they are release. Do not consider the latches to be obtained to update the sibling pointers in leaf nodes.)

   (f) (3 points) Still use the better latching algorithm introduced in the lecture. Give the latches aquired during the whole process when inserting 3 to the original tree. (Do not need to indicate when they are release. Do not consider the latches to be obtained to update the sibling pointers in leaf nodes.)

   The next two questions are about hashing tables. Pick the best suitable answer for each question.

   (g) (2 points) In linear probing, how are collisions resolved?

      A. By chaining keys with the same hash value in a linked list

      B. By incrementally searching for the next available slot

C. By using multiple hash functions

D. By rehashing the table

(h) (2 points) Which hashing scheme minimizes collisions by using rehashing and evictions?

A. Linear probing

B. Cuckoo hashing

C. Chained hashing

D. Extendible hashing

6. **Storage**

Consider a large table with 2,000 tuples stored via N-ary Storage Model. Each tuple takes 200 bytes (including tuper header). We will use sloted pages to organize the file. The page size is 16 KB (16,384 bytes). Page header size is 128 bytes. Each element in slot array takes 8 bytes. No tuple can span two pages.

(a) (3 points) Calculate the maximum number of tuples (records) that can be stored in one page.

(b) (3 points) Calculate the minimum pages needed to store this table (including page directory). Here, the pages are organized via heap file. The page directory stores one entry per page, and each entry takes 8 bytes. The size of each page directory page is also 16 KB.

7. **Buffer Pool**

(a) (2 points) What does scan sharing allow in a DBMS?

A. Reuse data from multiple queries

B. Reduce the memory overhead of dirty pages

C. Eliminate the need for buffer pool bypass

D. Prevent sequential flooding

(b) (2 points) Which of the following is NOT a buffer replacement policy?

A. Clock

B. LRU

C. FIFO

D. Bloom Filter

(c) (4 points) Simulate a scenario using the Clock replacement policy with the following sequence of page accesses: A, B, C, A, D, E, B, F. Assume the buffer pool has 3 frames. Which pages are evicted during the simulation? Give the page evicted on each page access. If none page is evicted upon some page access, use 'None' to denote.

Question:	1	2	3	4	5	6	7	Total
Points:	24	10	5	24	23	6	8	100
Score:								