

COMP 231 Database Management Systems

Quiz 2 December 7, 2001 10:15 – 10:45

Name: _____ Student ID: _____

1. [10] Some People argue that extensible hashing is not a true hash technique. What is the argument?

Traditional hashing converts a key value to the location of the corresponding record by computation (i.e., using a hash algorithm).

Extensible hashing, on the other hand, requires table look up. In fact, the hash prefix table behaves like a complete binary tree. As such, the extensible hashing can be considered a combination of tree and hashing.

2. [40] Given the following relational schema and functional dependencies:

$R = (A, B, C, D, E)$

$A \rightarrow BC$

$B \rightarrow D$

$DE \rightarrow A$

- a) Identify all candidate keys of R. Suggestion: use the methods in homework 1 question 1 to try out all possibilities starting with single attributes.

Single attributes: we note immediately that no attributes can determine E, so every candidate key must contain E, but E itself is not a candidate key.

Two attributes: there are four possibilities: AE, BE, CE, and DE

$AE^+ = ABCE = ABCDE$

$BE^+ = BDE = ABDE = ABCDE$

$CE^+ = CE$

$DE^+ = ADE = ABCDE$

We need to consider 3-attribute combinations, but it is easy to see that there is no 3-attribute combination satisfying both the uniqueness and minimality properties.

There are 3 candidate keys: AE, BE, DE

- b) Is R in BCNF? Explain why.

R is not in BCNF since the left-hand-sides of the first and second FDs are not superkeys.

- c) Is R in 3NF? Explain why.

Note that A, B, D, and E are prime attributes (i.e., they are attributes in the candidate keys) but C is not.

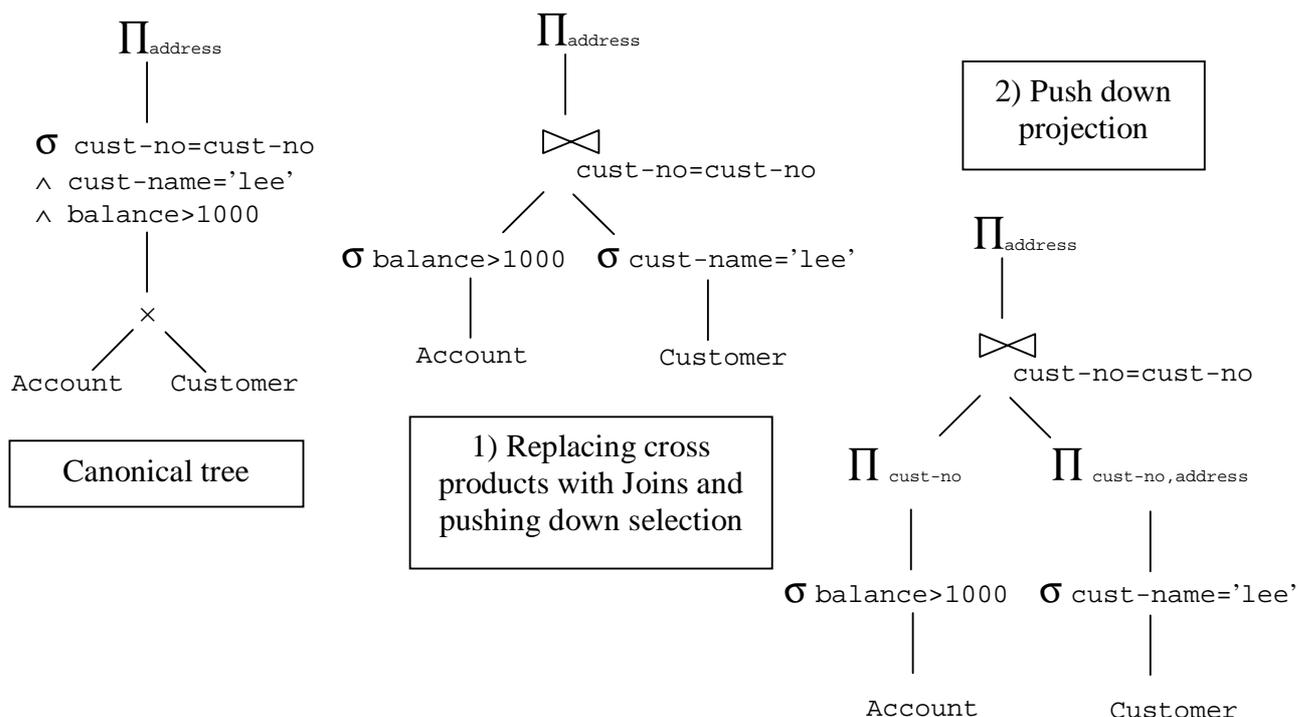
According to the definition of 3NF, the first FD violates 3NF since its left-hand-side is not a superkey and not all of the attributes on the right-hand-side are prime attributes.

Note that the second FD doesn't violate 3NF.

3. [30] Given: Customer (cust-no, cust-name, address)
 Account (cust-no, balance)

```
select  address
from    Account, Customer
where   balance >= 1000
and     cust-name = 'lee'
and     Customer.cust-no = Account.cust-no
```

Draw the canonical tree of the query, and optimise the canonical tree. Give the intermediate steps.



4. [20] Given two relations A and B of equal block size (i.e., same number of tuples contained in one disk block), which relation would you put in the outer loop in the block nested-loop join algorithm, if

i) A has 10 times more tuples than B. Briefly justify your answer.

B, the smaller relation, should be put in the outer loop. This is because the cost is $b_r * b_s + b_r$, where b_r is the outer relation. A small outer relation means a small b_r .

ii) A and B has the same number of tuples, but A has 10 times more tuples matching the join condition than B does and the join attribute of both A and B are indexed. Briefly justify your answer.

When index is available, either the smaller relation or the one with larger number of matching tuples should be put in the outer loop. Since the number of tuples in A and B are the same, A should be put in the outer loop.

The relation in the outer loop must be sequentially scanned, so even if it has a large number of matching tuples, it won't affect the performance. On the other hand, if it is placed in the inner loop, for every matching tuple in the outer relation, a large number of tuples from the inner relation has to be retrieved.