

## CS 186 Fall 2004: Exercise Solutions for Week of 10/11/04

### Exercise

1. In this question you will consider the join of two relations from an e-commerce database. The CARTS relation represents shopping carts of items. The CONTENTS relation has the contents of all the carts, with a foreign key called cartID to CARTS:

CARTS(cartID, customerID, date, comment)

CONTENTS(cartID, productID, quantity, price)

Assume that tuples in each relation are of fixed size. There are 10 CARTS tuples per page, and 100 CONTENTS tuples per page. Also assume that there are 1000 pages in the CARTS relation, and 5000 pages in the CONTENTS relation. The buffer pool is of size 1002 frames. You need to join on  $\text{CARTS.cartID} = \text{CONTENTS.cartID}$ .

a) How many I/Os are required for a page-oriented nested loops join, with CARTS as the outer relation and CONTENTS as the inner relation? Assume that the buffer manager is not used (i.e. every page reference generates an I/O).

b) Consider the same question as part (a), assuming now that the buffer manager is being used fully for this query, and that:

- The buffer manager starts out empty
- One output buffer frame is pinned by the join, and is used to hold output tuples until they are ready to be flushed to disk. This frame is not unpinned by the join until the end of the query.
- One input buffer frame is pinned by the join, and holds the current page of the outer relation at all times. All I/Os to the outer relation are placed explicitly into this frame, which is not unpinned until the end of the query
- The buffer manager is using the MRU replacement policy
- There is no other I/O happening in the system

Given these assumptions, how many I/Os are required for a page-oriented nested loops join in this case?

c) How many I/Os are required for a block nested loops join, with CONTENTS as the outer relation, CARTS as the inner relation, and 1000 pages per block?

d) Assume now that there are only 52 frames in the buffer pool total. Give the I/Os for the following join algorithms for this query

- Hash Join, using CARTS to build the hash table in the second phase
- Sort-Merge Join
- Block Nested Loops, CARTS as outer, block-size = 50

2. Consider the same database in Question 1, now running a selection range query on CONTENTS.quantity. Let the selectivity of this query be .4.

a) How many I/Os do we need if the whole table is scanned, and CONTENTS is unsorted? What about if it is sorted on CONTENTS.quantity?

b) Let there be an unclustered 3 level B+ tree index on CONTENTS.quantity. Suppose data is distributed evenly among the data record pages, and each data entry page can hold 500 data entries per page. How many I/Os are required using the index if the

query only requires the rids of the CONTENTS records? What if it requires entire data records?

- c) Let there be an index similar to that in part b, except it is clustered. Assume that each data records page has a 10% chance of having an overflow page. How many I/Os are required using the index if the query requires entire data records?
- d) Let there be an unclustered 3 level B+ tree index on (CONTENTS.price, CONTENTS.quantity). How many I/Os are required using the index if the query requires the entire data record?