

Name: _____.

CSE541/641 – Final Exam

Due: March 22, 2006

Late assignments will marked down.

Purpose:

- 1) To test your understanding of database implementation principles discussed in class.
- 2) To test your ability to apply what you have learned to new problems in RDBMS.

Deliverables:

- 1) For PART 1: Short answers to the given questions. If you need more than ½ a page to answer a question, you're definitely over-thinking it. Don't core dump -- you will lose marks if you provide irrelevant information, or if you give me descriptions of algorithms without an explanation of why it is relevant to the problem.
- 2) For PART 2: An analytical solution. Please include the steps you took to get to the answer, in case you were on the right path but didn't quite get the final result correct.
- 3) There are 5 questions total. **Some have multiple parts. Please read carefully and answer all parts.**

PART 1: Short answers

1. Give an example of a buffer replacement strategy that is ill suited for a DBMS under certain conditions. Specify a relational algebra operator algorithm for which your chosen buffer replacement strategy would perform poorly, and describe why by explaining when pages *should* be flushed, versus when they *would* be flushed using your chosen replacement algorithm.
2. Most modern databases implement *page cleaners*, which is a separate process or thread that periodically flushes dirty pages from the buffer to disk.
 - a. What are the advantages and disadvantages of providing asynchronous buffer cleaning in this manner?
 - b. Consider the interaction between page cleaners and the Write-Ahead Logging necessary for ARIES recovery. What implications does the page cleaning have for efficiency of the buffer page replacement algorithm under WAL?
3. A real-time database system (RTDBS) is a database system that must process transactions within definite time bounds, usually defined as a deadline.
 - a. Of the three concurrency schemes discussed in class (2-phase locking, timestamping, and validation), which would best meet this requirement of RTDBS? Why?
 - b. Under what conditions would your proposed scheme not be the best?

PART 2: Analytical questions4. Hash indexes

Consider the relation PARTS, which has Part# as hash key and which includes records with the following Part# values: 2369, 3760, 5046, 4871, 5659, 2222, 1821, 1074, 7115, 1620, 2428, 3943, 4750, 3157, 6975, 4981, 9208.

The hash function uses 8 buckets, numbered 0 to 7. Each bucket is one disk block and holds two records.

- a. Load these records into the file in the given order using the hash function $h(K)=K \bmod 8$. Calculate the average number of block accesses for a random retrieval on Part#.
- b. Now load the records into expandable hash files based on linear hashing. Start with a single disk block, using the hash function $h_0= K \bmod 2^0$, and show how the file grows and how the hash functions change as the records are inserted. Assume that blocks are split whenever an overflow occurs, and show the value of n at each stage.

5. Join algorithms

Consider a hash-join of two relations R and S having $B(R) = 1000$ and $B(S) = 500$. The values in R and S are skewed such that the hash function assigns three times as many tuples to even-numbered hash buckets as to odd-numbered buckets.

- a. How much memory would be required to perform the join in two passes?
- b. What is the performance of the hash-join given the skewed hashing?
- c. How would the performance of using the hash-join compare to using a sorted-merge algorithm?