

## Essential topics for Final:

1. Relational algebra and calculus
2. SQL
3. B+ tree index
4. Extendible hashing
5. Linear hashing
6. Buffer manager
7. Selection and projection evaluation
8. Simple Nested Loops Join
9. Index Nested Loops Join
10. Block Nested Loops Join
11. Sort-Merge Join
12. Hash-Join
13. External sorting
14. FDs, BCNF, 3NF, Decompositions
15. Concurrency control
16. Recovery manager

## Sample problems:

### Sample Question 1:

Consider the following schema (keys are underlined):

Product(pid, name, type, mfgr, price), Buys(cid, pid), Customer(cid, cname, age, gender)

Write the following query in relational algebra: Find the cids and names of all customers who have purchased the *second most* expensive product. You can assume that all product prices are unique.

### Sample Question 2:

You only have time to build one type of index structure. Will you choose B+-trees, extendible hashing, or linear hashing?

### Sample Question 3:

Give an example of a transaction schedule that is conflict-serializable, but which is not possible using 2PL.

**Sample Question 4:**

Assume that you want to join two relations  $R(A,B)$  and  $S(B,C)$ . The two relations are stored as simple (unsorted) heap files.

Briefly describe one scenario where you would prefer a hash-join to a sort-merge join

**Sample Question 5:**

Assume that you want to sort a file of size  $N$  pages. You have  $B$  buffer pages available. Assume that in the first pass, you read in and write out blocks of  $B$  buffer pages. In subsequent passes, you read in blocks of  $k$  buffer pages and write out blocks of  $k$  buffer pages.

What is the cost (measured in number of I/Os) of sorting the file using the external merge sort algorithm?