University of Wisconsin-Madison
Computer Sciences Department

Database Qualifying Exam
Fall 2002

Answer all five (5) questions (NOTE: This is different from some previous years, when you were only asked to answer 4 of 5.) Before beginning to answer a question make sure that you read it carefully. If you are confused about what the question means, state any assumptions that you make in formulating your answer. Good luck!

1. Database Design

The Burn'em and Bill'em (B&B) company has hired you to create an online store.

- They have products described by a *pid, price, qty-in-stock*, *short-description,* and *long-description.* The *long-description* field includes audio and video, and can be very large. It is only used when a customer browses a specific item, and is displayed on the screen. The *short-description* field is CHAR(25), and items are searched for using keyword searches on this field. Typically, a customer issues several searches (involving price and/or short-description) and looks at several item pages before placing an order.
- Each customer has an *ssn, name,* and *address.* B&B tracks some additional per-customer information as well, to support the two marketing programs described below. (It's up to you to determine exactly what information to track, and in what table!)
- All orders must be saved. The date, customer information, items ordered, and item prices for this order must be recorded.
- B&B has a Preferred Customer program, in which a customer belongs to one of several "preference levels".
    - There is a discount percentage associated with each preference level for each item that B&B sells. This percentage is adjusted by B&B, on roughly the same schedule that they adjust item prices, inventory, etc.
    - A customer can buy an item for its price less the discount for that item at the customer's preference level.
    - A customer's preference level is determined by their total purchases until the close of business on the previous day. (Level 1: Over $10,000; Level 2: $5,000 to 10,000; Level 3: $1,000 to $5,000; Level 4: $100 to $1,000; Level 5: Under $100).
- B&B also has a Corporate Affiliate program, in which a corporation can sign up as an affiliate and register the social security numbers (*ssn*'s) of its employees. This entitles each employee to get an extra 5% discount on all items; effective the day after the corporation signs up. (Of course, some employees may never shop at B&B, and some customers might not be employees of affiliates.)

Design a database schema for B&B. Assume that a single centralized DBMS is used, and is accessed by thin web-clients; no data is cached outside the DBMS.

(i) Show the schemas for the tables that hold the necessary data on products, customers, orders, affiliates, etc.
(ii) Identify all keys, foreign keys, and functional dependencies. What normal form is each table in? (Be sure to consider the Date-Fagin results in answering this question!)
(iii) Describe the kind of workload you anticipate. (For each table, what are the most updates and queries?)
(iv) Based on the expected workload, and feedback that customer queries are very frequent and very important (and remember that a customer must always receive the best price that they qualify for!), what indexes would you build?
(v) A year later, B&B approaches you and tells you that the system needs to be scaled to handle 10 times (10x) the load, and be designed to scale to handle 100x the load. How would you solve this problem? Discuss the use of view materialization within the DBMS, as well as the use of additional servers and replication.

2. Bitmap Indices

The following three questions deal with bitmap indices. You may find some of them rather open-ended; you can make (and should clearly state!) reasonable assumptions you feel you need to make to answer the question.

(i) Are bit-mapped indices more appropriate for low-cardinality or high-cardinality attributes? Why?
(ii) Describe an efficient scheme for mapping from bit-positions to records. What implications does your scheme have for supporting updates?
(iii) "Covering indices" are a common technique used in practice to speed decision support applications in RDBMS, where an index "covers" a query if the query can be answered consulting only the index (that is, without referring to the indexed table.) When would you expect covering B+ tree indices to perform better than bit-mapped indices? When would you expect them to be worse?

3. Distributed Query Optimization

(i) Describe how R* extended the System R optimizer to handle distributed queries.
(ii) What distributed join strategies did R* support?
(iii) Was the presence of replicas considered? If not, discuss how the R* optimizer could be extended to handle replicas.
(iv) As the number of replicas increases, how does this affect the cost of optimization?

4. Transaction Management: Two-Phase Commit

(i) Describe the basic two-phased commit protocol, and also the "presumed-abort" variant. In each, be sure to describe when records must be forced to disk, and also when the coordinator can safely "forget" a transaction.

(ii) Consider now the "presumed-commit" variant of the protocol. It requires a special type of log record not found in the presumed-abort variant; what is this record, and why is it necessary?

(iii) Would two-phase commit be necessary in a parallel system like Gamma? Would it be necessary in a parallel system like Gamma being run on an SMP instead of a shared-nothing multiprocessor? Explain your answer.

5. Data Mining

The following questions explore the Birch algorithm, which can be used to identify clusters in large datasets.

(i) Describe how Birch scales to large datasets. In particular, explain the use of cluster summaries, or *cluster features*, and compare the algorithm to B-tree maintenance.

(ii) What is the shape of the "clusters" discovered by Birch? How would you use the algorithm to discover clusters of arbitrary shapes?

(iii) Suppose that you want to cluster a (potentially infinite) stream of data points using Birch.

a. Explain how you would use the algorithm (What phases of the algorithm would you retain? What additional post-processing might you do?).
b. What is the impact of the periodic re-organization required when Birch runs out of memory?
c. Suggest ways to adapt Birch to overcome the problem identified in part (iii) (b).