SYSTEMS QUALIFYING EXAM

January, 1995
Department of Computer Science, Cornell University

1. **Database Question**

a. **Queries in Relational Algebra** (10 points)

The equally-attractive-non-alcoholic-beverage (EANAB) drinkers database consists of information about drinkers, EANABs, and pubs, telling:

- Which drinkers like which EANABs; likes(Drinker,EANAB).
- Which drinkers frequent which pubs; frequents(Drinker,Pub).
- Which pubs serve which EANABs; serves(Pub,EANAB).

Give a relational algebra expression for the queries below.

- (3 points) Find all happy drinkers. A drinker is happy if she frequents at least one pub that serves an EANAB she likes.

- (3 points) Find all pubs that a drinker should visit. A drinker should visit a pub if it serves an EANAB she likes.

- (4 points) Find the set of pubs at which *all* drinkers can get an EANAB that they like. That is, for pub in the set, every one of the drinkers can find at least one of the EANABs she likes. In addition, every pub on the list must be frequented by every drinker.

b. **Functional dependencies** (10 points)

- (10 points) List all non-trivial functional dependencies satisfied by the following table, where you can assume that it is the intent of the designer that exactly this set of rows should lie in the table. Include any dependencies derived from the closure.

A	B	C	D
a_1	b_2	c_1	d_1
a_1	b_1	c_2	d_2
a_2	b_2	c_1	d_3
a_2	b_1	c_2	d_4
a_2	b_3	c_4	d_5

c. **Relational Database Design** (20 points)

Suppose we have a database for an investment firm, consisting of the following attributes: B (broker), O (office of broker), I (investor), S (stock), Q (quantity of stock owned by investor), and D (dividend paid by a stock), with the following functional dependencies.

$$F = \left\{ \begin{array}{c} S \rightarrow D \\ I \rightarrow B \\ IS \rightarrow Q \\ B \rightarrow O \end{array} \right\}$$

- (2 points) Find a key for the relation scheme $R = BOSQID$

- (2 points) How many keys does the relation scheme R have?

- (4 points) Find a lossless join decomposition of R into BCNF.

- (4 points) Find a decomposition of R into 3NF that is lossless and that preserves all dependencies.

- (4 points) Is the decomposition of *R* into *(SD,IB,ISQ,BO)* lossless? Prove your answer.

- (4 points) Replace the dependency $S \rightarrow D$ by the multi-value dependency $S \twoheadrightarrow D$. That is, *D* now represents the dividend history of the stock. Find a fourth normal form decomposition of *R*.

d. **Concurrency Control** (20 points)

- (10 points) For each of the following statements, indicate and explain whether they are true or false, about 2PL.

  - 2PL is the same as serializability.

  - 2PL prevents transactions from reading dirty data.

  - 2PL requires the use of a write-ahead log. Recall that a write-ahead log is written into before writing into the database.

- (10 points) Show how a locking scheduler and a time-stamp based scheduler handle the following transactional sequence. Draw the waits-for graph at any point where a deadlock would occur.

$$W_3\,(A)\,R_1\,(A)\,W_1\,(Z)\,R_2\,(B)\,W_2\,(Y)\,W_3\,(B)\,C_1C_2C_3$$

$W_n\,(A)$ stands for transaction *n* writing the datum *A*, while $R_n\,(A)$ stands for transaction *n* reading *A*. C_i stands for commit of transaction *i*.

2. Computer Architecture

Consider a machine with a 3 way set associative cache. Suppose the line (block) size is 16 bytes, and 13 bits of an address are used to generate the index of the cache line. An LRU algorithm is used for cache replacement.

a) (15 points) What is the size of the cache? A diagram may prove helpful.

b) (15 points) Suppose the following code fragment is run on this machine, and the arrays a, b, c, and d are stored at memory locations 8192, 16384, 24576, and 32768, respectively. All data elements in this fragment are 32 bit integers, and the compilers performs no optimizations.

```
for i:= 1 to 64 do
        a[i]  :=  (b[i]  + c[i])/2  + d[i];
```

How would you expect the cache to perform while running this code fragment? Give a qualitative answer, and explain why you think your answer is correct.

c) (10 points) How would you change the hardware to improve cache performance of the code fragment in (b)?

d) (10 points) Show how to rewrite the code fragment in part (b) to improve cache performance. You can assume the machine has a large number of registers (say, 32).

e) (10 points) How would you change the memory location of the arrays to improve performance?

3. Distributed Systems

(60 points) While home for Thanksgiving, your uncle Jed requests some advice. Jed is foreman for a car repair center, and his workers are asking that Jed purchase enough repair tools so that every worker can have a full set for themselves. Jed is sure that he has enough tools to deal with any realistic scenario, but his workers don't believe him and have been collecting every tool they could possibly need for a job before even starting the job, and returning the tools only at the end. A result is that a lot of work just isn't being done for lack of tools.

Can you suggest a good solution to Jed's problem? Your answer should be based upon O/S resource allocation and deadlock considerations. A good answer does not require purchasing additional equipment or changing the number of employees, and maximizes job throughput subject to these considerations.

4. Memory management and shared libraries

a) (10 points) Briefly describe the address mapping occurring in the following memory management techniques (i.e., omit protection issues). Keep your answer to two or three sentences, i.e., keep it general and high-level and don't go into the details of the mechanisms typically used to implement the mappings.

Segmentation

Paging

b) (10 points) Again briefly (1 or 2 sentences), describe:

Two advantages of segmentation over paging

Two advantages of paging over segmentation

c) (15 points) Consider a paging system which uses a page table stored in memory:

If a single memory access takes 200 nanoseconds, how long does a paged memory reference take?

If we add a TLB (translation look-aside buffer) and 75% of all page-table references are found in the TLB in zero time, how long does a paged memory reference take on average?

d) (15 points) Recent operating systems offer the facility of shared libraries which allows a single in-memory copy of a library to be shared among multiple independent processes.

[Background info: in a system which uses dynamically loaded and linked libraries an executable does not contain the code of the libraries it was linked with. Instead, the library is dynamically loaded into memory (e.g. from /usr/lib on UNIX) when the executable starts and then the executable is dynamically linked with the library which usually involves some form of stubs or indirection tables. In a system that uses shared libraries all processes using a given library share a single in-memory copy of the library.]

Why are shared libraries highly desirable? (Mention one good reason)

Why are shared libraries sometimes undesirable? (Mention one good reason)

e) (10 points) Does the use of shared libraries make program execution faster or slower than regular libraries (e.g., that are statically linked into each executable)? Briefly justify your answer.

Give one reason for making it slower.

Give one reason for making it faster