

DAIS Qualifying Examination  
*Fall 2005*

Department of Computer Science  
University of Illinois at Urbana-Champaign

Sept. 26, 2005

Time Limit: 180 minutes

- You are required to answer the **one** required question and **five** other questions.
- The required question will test the examinee's understanding of the basic concepts from all the three subareas of the exam. The required question has **10** subquestions.
- You are required to answer other **5** questions out of ten. If you answer more than five questions, the committee will randomly select five to grade. The distribution of the questions by areas is as follows.
  - **Database Systems:** 4 questions
  - **Data Mining:** 3 questions
  - **Information Retrieval:** 3 questions

For each such question, there are three subquestions. The first is to test the basic knowledge related to a topic, the second is to test current research related to the topic, and the third is to test research capability.

- Answer one question on a booklet. With one required and five other questions, you will use six booklets for your answers.
- To ensure the fairness of grading, all the exams will be kept anonymous. Your answer sheets should only bear the assigned ID but *no names*.
- Please make your answers clear and succinct; you will lose credit for verbose, convoluted, or confusing answers. *Simplicity does count!*

## Required Question (Basic Concepts): Problem 0

Short answers: The answer of each of the following questions is expected to be no more than two lines (i.e., no explanation nor reasoning is needed).

- (1) (*how to optimize a query*) Suppose that we have the relations Patient(pname, dname, lifeExpectancy) and Doctor(dname, email), where pname is a key for Patient and dname is a key for Doctor. Consider the SQL query

```
SELECT Doctor.dname, Patient.pname, Patient.lifeExpectancy
FROM Doctor, Patient
WHERE Doctor.dname = Patient.dname
      AND Patient.lifeExpectancy < 1
ORDER BY Doctor.dname
```

Many query optimizers would choose a sort-merge join to execute this query, because with a sort-merge join, the result of the join operation will be sorted on Doctor.dname. Describe one situation where the data distribution and the available indices would lead a good query optimizer to choose another kind of join, and explain why the chosen kind of join is better in the situation that you describe.

- (2) (*dependency theory*) Suppose we have two functional dependencies,  $AB \rightarrow C$  and  $BC \rightarrow A$ . Either justify why  $B \rightarrow C$  must also hold, or give a counterexample that shows why it does not hold.
- (3) (*two-phase locking*) Consider the following schedule:

```
Transaction 1 writes object x
Transaction 2 reads object x
Transaction 2 writes object y
Transaction 1 writes object z
```

Is this schedule allowable under two phase locking? Why or why not?

- (4) (*B-trees*) Suppose that you want to index fixed-length records, each 256 bytes long. The key of each record is 4 bytes long. Records are stored on disk so that they do not cross page boundaries. If you use a three-level B+ tree to index this file, with the records themselves occupying the third level of the tree, then what is the maximum number of records that you can index if the B+ tree pointers are 4 bytes long and the B+ tree nodes are 4096 bytes long?
- (5) (*data cubes*) List (the names of) four methods that can be used to compute *iceberg cubes* efficiently.
- (6) (*clustering*) List four *clustering* methods, one from each of the following categories: (a) partitioning method, (b) hierarchical method, (c) density-based method, and (d) model-based method.
- (7) (*classification*) List four *classification* methods, one from each of the following categories: (a) scalable decision-tree induction, (b) statistical approach, (c) associative classification, and (d) lazy evaluation approach.

- (8) (*TF-IDF weighting*) Consider 3 English words “baseball”, “the”, and “is”. In a random collection of English news articles, which of them do you expect to have the highest IDF? In a news article about “Iraq war”, which of them do you expect to have the highest TF?
- (9) (*relevance/pseudo feedback*) Use no more than 2 sentences to explain the difference between relevance feedback and pseudo feedback.
- (10) (*language model smoothing*) List three methods for smoothing language models in information retrieval.

# Database Systems: Problem 1

## Part 1

1. Undo/redo logging is more flexible than both undo and redo logging. So are there situations where it would be more desirable to use either undo or redo logging, but not undo/redo logging?
2. Briefly explain the notions of conflict serializability and view serializability. Describe the key contributions of the ARIES transaction recovery method.

## Part 2

1. Assume that at the crash time a transaction T1 has committed. It has read a value A written by a transaction T2, which has not committed. The DBMS attempts to use an undo/redo scheme to recover from this crash. Explain complications that can arise from this scenario.
2. Describe in details how a DBMS addresses the complications that you have mentioned in the above question.

## Part 3

1. Consider a Web-scale information retrieval system such as Google. Discuss various recovery and concurrency control problems that arise in this context, and sketch solutions. Would any recovery and concurrency control techniques from the database context apply to these scenarios?

## Database Systems: Problem 2

### Part 1

1. What is data integration? Name three key reasons why it is a difficult problem.
2. Briefly describe the key ideas behind model management.

### Part 2

1. Briefly describe the merge operator in the model management context (as detailed in the paper "Applying Model Management to Classical Metadata Problems" by Philip A. Bernstein). Is this operator completely automatic? Are all operators of the proposed model management framework automatic? If yes, explain why. If not, explain why and name examples of operators that are not completely automatic.
2. Define the schema matching problem. Discuss any possible connection between this problem and the information retrieval (IR) problem. Is there any IR techniques that can be applied to the schema matching context, and vice versa?

### Part 3

1. On-the-fly data integration refers to settings in which the user may ask only a few queries, and want answers to those queries as quickly as possible. Give a few examples of scenarios where on-the-fly data integration is desirable, and the current paradigms of data integration are not well suited.
2. Discuss what key issues we must solve to enable on-the-fly data integration, and sketch possible solutions. Note: you can consider only certain types of data integration if you like (that is, you can scope your problem, but clearly state the scope).

## Database Systems: Problem 3

### Part 1

- (a) What are the three types of hardware architectures for parallel multiprocessor systems? Draw a diagram to illustrate each of the three architectures.
- (b) “*Parallelism is an unanticipated benefit of the relational model.*” Explain why.

### Part 2

- (a) In the performance metrics for parallel systems, we define both *speedup* and *scaleup*. In the context of databases, explain why we need both measurements.
- (a) Consider the current application scenarios in which many Web sites are supported by back-end databases for their online transactions. Which measure, speedup or scaleup, makes more sense? Explain why.

### Part 3

- (a) There are three types of data partitioning schemes commonly used in parallel DBMS. What are these schemes? Give an advantage for each scheme.
- (b) The above schemes are proposed for relational databases. For the new setting of query processing for XML data, are any of them applicable? If not, suggest a scheme for XML data partitioning. Explain your answers.

# Database Systems: Problem 4

## Part 1

- (a) Why a database system should care about supporting user defined data types?
- (b) What issues are involved in supporting user defined data types in a database system? Name three such issues.

## Part 2

- (a) The idea of *extended type system* (which supports user-defined data types) is related to that of an *object-oriented DBMS*. Identify how they are similar and different.
- (b) Why defining a new data type may impact the correctness of transaction management? Explain. How does the idea of GiST address this issue?

## Part 3

When a system supports user-defined data types, it often also needs to support *user-defined functions*. For instance, the following query looks for houses that are located in a “nice location” in the same city as UIUC. The Boolean function *NiceLocation*, as a user-defined function, returns *true* or *false* for the new data type “location.”

```
SELECT h.address
FROM Houses h, Schools s
WHERE NiceLocation(h.location) and s.name = “UIUC” and h.city = s.city
```

Such user-defined functions can often complicate query processing and optimization.

- (a) Name one *significant* issue that such user-defined functions may impact query processing and optimization. Explain why the issue is significant. Use the example query to illustrate.
- (b) Propose a solution, by briefly sketching your idea, for the issue you identified above.

# Data Mining and Data Warehousing: Problem 1

## Part 1

A data cube of  $n$  dimensions contains exactly  $p$  nonempty cells in the base cuboid. Assume that there are no concept hierarchies associated with the dimensions.

- (a) What is the *maximum number of nonempty cells* (including the cell in the base cuboid) possible in such a materialized datacube?
- (b) If the minimum support is 2, what is the *minimum number of nonempty cells* possible in the materialized iceberg cube?

## Part 2

In certain applications, people would like to perform OLAP in a high dimensional data warehouse.

- (a) Why do most popular cubing algorithms fail in high-dimensional cubing?
- (b) Propose a method that may support efficient OLAP operations in a medium-sized (e.g., tens of mega-bytes) high-dimensional (e.g., 100 dimensions) data set.

## Part 3

Outlier detection is an important task in data cube as well. Suppose a cell  $c$  is considered as an outlier in a data cube with measure *count* if  $c$ 's value (i.e., count) is unproportionally high/low in comparison with other cells in its corresponding columns and rows. Suppose only the cells in the base cuboid are available. Outline an outlier detection method in such a data cube.

## Data Mining and Data Warehousing: Problem 2

### Part 1

Briefly outline the major differences of the following patterns:

- (a) *frequent itemsets* in transaction databases,
- (b) (*frequent*) *sequential patterns* in transaction-based sequence databases,
- (c) (*frequent*) *protein sequences* in protein sequence databases.

### Part 2

PrefixSpan is an efficient sequential pattern mining algorithm for mining transaction-based sequence databases. Suppose one wants to mine sequential patterns with the following constraints. Discuss how to extend the algorithm in each case.

- (a) Mining *closed sequential patterns*, where a pattern  $p$  is *closed* if there exists no superpattern  $s$  in the database that carries the same support as  $p$ .
- (b) Mining sequential patterns, each containing a user-specified regular expression, such as  $a^*\{b^+|X^3\}$ , i.e., containing 0 or more occurrences of  $a$ , following by one or more occurrences of  $b$  or three (repeated) occurrences of one single symbol.

### Part 3

In some applications, one may like to classify data based on their sequential patterns.

- (a) Outline such a classification method, and
- (b) Discuss how to improve the *accuracy* and *efficiency* of such a classification method.

## Data Mining and Data Warehousing: Problem 3

### Part 1

Scalability is one of the central issues in data mining. When clustering large data sets, one may need to develop scalable algorithms. Outline and compare two methodologies that may lead to the development of high-quality scalable clustering algorithms.

### Part 2

In some applications, a large amount of data may flow in and out like streams. Outline an efficient clustering method that may discover the cluster evolution regularities in data streams.

### Part 3

One major challenge in frequent itemset mining is the possible generation of a huge number of frequent itemsets in the mining process. *Clustering can be used to compress frequent itemsets and thus dramatically reduce the total number of frequent itemsets to be generated.*

- (a) Give an example to show that the above statement is true.
- (b) Propose an efficient algorithm that performs effective clustering of frequent itemsets in data mining.

# Information Retrieval: Problem 1. Information Retrieval Models

## Part 1

- a List three different information retrieval models.
- b In most retrieval formulas, long documents are penalized through length normalization. Use no more than 2 sentences to briefly explain why this is desirable.

## Part 2

Consider a document collection  $C_0 = \{D_1, \dots, D_N\}$ , where  $D_i$  is a document. We construct two other collections  $C_1$  and  $C_2$  in the following way: (1)  $C_1$  is formed by adding  $k - 1$  ( $k > 1$ ) copies of document  $D_1$  to  $C_0$ , thus it has  $N + k - 1$  documents. (2)  $C_2$  is formed by replacing  $D_1$  in  $C_0$  with a new document  $D_1'$ , which is a concatenation of  $k$  copies of  $D_1$ .  $C_2$  thus has the same number of documents as  $C_0$ .

Let  $t$  be a term occurring in  $D_1$  and  $IDF(t, C_i)$  be the IDF value of  $t$  in collection  $C_i$ . Let  $\theta_B(C_i)$  be a collection language model estimated based on collection  $C_i$ .

- a What can we say about  $IDF(t, C_1)$  and  $IDF(t, C_0)$ ? Are they the same? If not, which is higher? Why?
- b What can we say about  $IDF(t, C_2)$  and  $IDF(t, C_0)$ ? Are they the same? If not, which is higher? Why?
- c Is  $p(t|\theta(C_1))$  higher than, lower than, or equal to  $p(t|\theta(C_0))$ ? Why?
- d Is  $p(t|\theta(C_2))$  higher than, lower than, or equal to  $p(t|\theta(C_0))$ ? Why?

## Part 3

The Term Discrimination Constraint on a retrieval function is defined as follows [Fang et al. 04]:

**TDC:** Let  $q$  be a query,  $d_1$  and  $d_2$  be two documents, and  $w_1, w_2 \in q$  be two query terms. Assume  $|d_1| = |d_2|$ ,  $c(w_1, d_1) + c(w_2, d_1) = c(w_1, d_2) + c(w_2, d_2)$  and  $c(w, d_1) = c(w, d_2)$  for all other words  $w$ . If  $idf(w_1) \geq idf(w_2)$  and  $c(w_1, d_1) \geq c(w_1, d_2)$ , then  $f(d_1, q) \geq f(d_2, q)$ .

- a In [Fang et al. 04], it is shown that none of the existing retrieval functions can satisfy this condition *unconditionally*. However, as a necessary condition, this constraint is a bit too strong. Give a specific example to show why the TDC constraint is not always desirable.
- b Can we propose a modified version of TDC to make it more reasonable?

# Information Retrieval: Problem 2. Language Models for Retrieval

## Part 1

- a Briefly explain what are the two roles played by smoothing in the query likelihood retrieval method.
- b Write down the Dirichlet Prior smoothing formula and explain the meaning of each variable used in the formula.

## Part 2

- a In the query likelihood retrieval method, we rank documents based on the probability of query  $q$  given document  $d$ , i.e.,  $p(q|d)$ . Suppose  $d = w_1w_1w_2w_2w_2w_2$  is a very short document with only 6 words and we do *not* do smoothing. Find  $q^* = \operatorname{argmax}_q p(q|d)$ , i.e., the query that has the highest probability for document  $d$ .
- b The following aspect-based query likelihood formula is given in [Kurland & Lee 04]:

$$p(q|d) = \sum_c p(q|c)p(c|d)$$

where  $q$  is a query,  $d$  is a document, and  $c$  is a topic cluster. Show that this aspect-based query likelihood method can cover the standard query likelihood method as a special case through appropriate choices of  $c$ .

## Part 3

In the query likelihood retrieval method, we rank documents according to  $p(q|d, R = 1)$ , where  $q$  is a query,  $d$  is a document, and  $R$  is a binary relevance variable ( $R = 1$  means “relevant”). According to the derivation of this formula in [Lafferty & Zhai 03], we have made the assumption that  $p(q|d, R = 0) = p(q|R = 0)$ . If we do not make this assumption, we would have to rank documents according to  $\frac{p(q|d, R=1)}{p(q|d, R=0)}$ , which presumably would perform better. The question is how to estimate  $p(q|d, R = 0)$ ? Can you suggest a way for doing it?

# Information Retrieval: Problem 3. Collaborative Filtering & PageRank

## Part 1

- a Briefly explain what is collaborative filtering
- b What does pLSA stand for?
- c Why do we need to do “User Normalization” in collaborative filtering?

## Part 2

- a A mixture of Gaussian model is used in [Hofmann 03] to model collaborative filtering problem. Suppose we have a simple mixture model of  $k$  Gaussian models with parameters  $\{(\mu_i, \sigma_i^2)\}_{i=1}^k$ , where  $\mu_i$  is the mean and  $\sigma_i^2$  is the variance for the  $i$ -th component Gaussian model. Assuming  $\pi_i$  is the mixing weight for the  $i$ -th component Gaussian model, write down the likelihood of  $x$  given this mixture model. (The density function of Gaussian is  $\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ .)
- b Suppose we observe a set of values  $X = \{x_1, \dots, x_N\}$  from this mixture model. Suppose all the variances  $\sigma_i^2$  are known and we just want to estimate the means of all the  $k$  Gaussian models (i.e.,  $\mu_i$ 's). Sketch an EM algorithm for estimating the means using the Maximum Likelihood Estimator. Note that the mixing weights  $\pi_i$ 's are unknown.

## Part 3

Can you suggest a way for applying PageRank to collaborative filtering? In particular, we may think of an item as a web page and somehow add links. Sketch a PageRank-like formula for scoring items for a given user if possible.