CS448/648: INTRODUCTION TO DATABASE SYSTEMS  
(Winter, 1996)  
Final Examination

INSTRUCTOR:  G. E. Weddell                      TIME:  3 hours

This is an open book examination. For example, class text, copies of overhead slides and printed notes may be used. Do all questions.

NOTE 1: For credit, material transcribed directly from the textbook or notes must be enclosed in clear quotation marks, and will receive less credit than material rephrased in your own words.

NOTE 2: Some of the questions in this examination are open ended; however, they can be answered to the level discussed in class by short organized answers. It is recommended that you spend part of your time organizing your answer, rather than writing down ideas in the order they occur to you. The conciseness and organization of your answers will be taken into consideration in the grading.

NOTE 3: There are 100 marks in total. As a guide to managing your time, the marks awarded for each question are indicated in parenthesis at the start of each question.

-----------------------------

I. (12 marks) Indicate whether each of the following statements is true or false. In each case, justify your answer using no more than three sentences.

(a) If an index is created on an attribute, every SQL query involving that attribute will be able to execute faster.

(b) If a database system does not maintain statistics, it will be unable to process queries.

(c) In an SQL database system, it is possible for a transaction to retrieve (read) a tuple that has been updated by another active transaction.

(d) Once defined, a view can be used exactly like a base table.
Note: Questions II, III and IV refer to the following relational schema, which describes a hypothetical banking database.

CREATE TABLE Branch
(   BNum       INTEGER NOT NULL,
    BName      VARCHAR(20) NOT NULL,
    City       VARCHAR(20) NOT NULL,
    PRIMARY KEY (BNum) );

CREATE TABLE Customer
(   CNum       INTEGER NOT NULL,
    CName      VARCHAR(20) NOT NULL,
    City       VARCHAR(20),
    PRIMARY KEY (CNum) );

CREATE TABLE Account
(   ANum       INTEGER NOT NULL,
    CNum       INTEGER NOT NULL,
    BNum       INTEGER NOT NULL,
    Balance    INTEGER NOT NULL,
    PRIMARY KEY (ANum),
    FOREIGN KEY (CNum) REFERENCES Customer,
    FOREIGN KEY (BNum) REFERENCES Branch );
II. (12 marks) Consider a relational scheme representing a simple view of the system schema in the SQL2 standard.

CREATE TABLE Table
  (  TabName   VARCHAR(20) NOT NULL,
    ColName   VARCHAR(20) NOT NULL,
    PKey      VARCHAR(5) NOT NULL,
    PRIMARY KEY (TabName, ColName)
  );

Each tuple in Table expresses the fact that table TabName has a column ColName, which is a part of the primary key of table TabName if and only if PKey has value true.

(a) Translate each of the following queries into SQL.

(i) The primary key columns of table Branch or table Table.

(ii) All primary key columns that do not occur as non-primary key columns.

(b) List the result of evaluating the previous two queries assuming Table contains the tuples encoding the banking schema augmented with the Table scheme itself.
III. (18 marks) For this question, assume that Branch contains 100 tuples, Customer contains 10,000 tuples, and Account contains 20,000 tuples. All bank branches and all bank customers are located in four cities. Balances in the bank’s accounts range from 0 to 100,000 dollars.

For each of the following queries, devise an initial execution plan and express it using the relational algebra. Then, use the optimization heuristics discussed in class to obtain a more efficient execution plan.

(a) SELECT DISTINCT Anum
    FROM Account, Branch
    WHERE Account.BNum = Branch.BNum
    AND Balance < 10,000
    AND (Balance > 1,000 OR City = ’Ottawa’)

(b) Hint: this query can be processed using a three-way join.

SELECT DISTINCT BNum
FROM Branch, Account
WHERE City = ’Ottawa’
AND Branch.Bnum = Account.Bnum
AND EXISTS (SELECT *
            FROM Customer
            WHERE Customer.CNum = Account.Cnum
            AND Customer.City = ’Montreal’
            )
IV. (18 marks) Questions on a variety of topics. Answer each part using no more than a few sentences in each case.

(a) Define each of the following terms:

(i) conceptual level,
(ii) change anomaly,
(iii) non-procedural query language,
(iv) strict two-phase locking, and
(v) relational completeness.

(b) Explain why B-trees are more appropriate for secondary storage than binary trees.

(c) For what reason, if any, might a DBA wish to vertically partition a relation in a database managed by a client-server based engine?

(d) Give two examples of SQL queries on the banking schema that cannot be expressed in the relational algebra. (Your examples should be based on different features of SQL to obtain full credit.)
V. (15 marks) Suppose the Account table contains the following tuples.

<table>
<thead>
<tr>
<th>ANum</th>
<th>CNum</th>
<th>BNum</th>
<th>Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>3</td>
<td>200.00</td>
</tr>
<tr>
<td>2</td>
<td>110</td>
<td>3</td>
<td>1020.15</td>
</tr>
<tr>
<td>4</td>
<td>120</td>
<td>4</td>
<td>777.77</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>3</td>
<td>16.22</td>
</tr>
</tbody>
</table>

Consider the following sequence of transaction requests against tuples of the Accounts table.

\begin{align*}
T_1 & : \text{begin} \\
T_1 & : \text{read the balance of account 1} \\
T_2 & : \text{begin} \\
T_2 & : \text{subtract 100 from the balance of account 4} \\
T_3 & : \text{begin} \\
T_3 & : \text{read the balance of account 4} \\
T_1 & : \text{read the balance of account 5} \\
T_2 & : \text{add 100 to the balance of account 2} \\
T_2 & : \text{commit} \\
T_4 & : \text{begin} \\
T_4 & : \text{add 10 to the balance of account 2} \\
T_4 & : \text{add 10 to the balance of account 5} \\
T_3 & : \text{read the balance of account 1} \\
T_1 & : \text{commit} \\
T_3 & : \text{commit}
\end{align*}

(a) Assuming that the database system uses strict two-phase locking, and that all transactions run at isolation level 3, in what order will the requests actually be performed by the database system?

(b) Show the contents of the transaction log after all of the requests shown above have been processed.

(c) Suppose that a system failure occurs after the requests shown above have been processed. Indicate the fate (commit or abort) of each of the transactions whose requests appear in the sequence above. Also, show the contents of the Accounts table after the system has recovered from the failure, and before any new transactions are allowed to run.
VI. (25 marks) Questions on design theory and functional dependencies. Answer each part.

(a) Recall that an initial design for a relational database is usually obtained by first deriving an E-R model of the data, and then translating the E-R model to a relational model. Also recall that a simple methodology for obtaining the E-R diagram (outlined by slide IV-30 of the course overheads) is to start by first recognizing entity sets, then to proceed with recognizing relationship sets and their participating entity sets, and so on. In no more than two paragraphs, suggest an alternative methodology to follow if one wishes to obtain the initial relational database design using normalization theory.

(b) Indicate whether or not an efficient algorithm is known for each of the following problems.

(i) Is a given relational scheme in BCNF?
(ii) Is a given relational scheme in 3NF?
(iii) Can a given functional dependency be derived from a finite set of given functional dependencies?
(iv) Is a given superkey of a relational scheme also a candidate key?

(c) Briefly comment on the relationship between the notion of candidate key and of functional dependency.

(d) Assume the functional dependency $X \rightarrow Y$ holds on relation $R$. State the conditions on which the same functional dependency will continue to hold on the relations produced by each of the following relational algebra queries. Justify your answer in each case.

(i) $(R \text{ WHERE } C)$ (where $C$ is an arbitrary condition)
(ii) $R[Z]$ (where all attributes in $X$ or $Y$ occur in $Z$)