Database Exam

Computer Science Research Area

25 June, 1996

General Instructions:

- Answer 3 of the following questions. Be sure you turn in only 3 answer sheets.

- Answer each question on a separate sheet of paper supplied for the test. In the upper right-hand corner of each answer sheet, write "Database Exam", the problem number, and your code number.

Please answer only one question per sheet.

When you finish the database exam, please staple your three answers and the exam sheet together and turn them in.

1. Basic relational database material
   Answer the following questions using the relational schema below. Primary keys are italicized.

   Course(course_no, course_name)
   Prereqs(course_no, pre_req_no)
   Student(student_id, lname, fname, major)
   Faculty(faculty_id, lname, fname, department)
   TakenBy(student_id, course_no, grade)
   TaughtBy(faculty_id, course_no, quarter, year)

(a) What is referential integrity? Give an example from the following relational schema.
(b) Can more than one student take a particular course? Why or why not?
(c) Can a faculty member teach more than one course? Why or why not?
(d) Can a student have more than one grade for the same course? Why or why not?
(e) Give a query in relational algebra to answer the following:
    give names of students who have never taken a course taught by Terwilliger.
(f) Give a query in tuple calculus or domain calculus to answer the following:
    give the course names for each course and its prerequisites (course names) for all courses taught in the fall of 1995.
2. Normal Forms

(a) Explain what a dependency-preserving decomposition is in terms of a relation scheme \( R \), with dependencies \( F \), into a relation scheme \( \rho = (R_1, \ldots, R_k) \).

(b) What are the practical implications of not preserving dependencies?

(c) Describe an algorithm for lossless-join decomposition of a relation scheme \( R \), with dependencies \( F \), into a relation scheme \( \rho = (R_1, \ldots, R_k) \) in Boyce-Codd Normal Form.

(d) Decompose \( R(a, b, c, d, e, f) \) into BCNF using your algorithm where \( F \) is

\[
\begin{align*}
a & \rightarrow b, c, d, e, f \\
b, c & \rightarrow a, d, e, f \\
b & \rightarrow f \\
d & \rightarrow e.
\end{align*}
\]

Label all steps of your computation.

(e) Is the decomposition dependency-preserving? Explain.

3. Comparative Data Models

(a) Discuss the relative merits (advantages and disadvantages) of the following data models: relational, hierarchical, network (CODASYL DBTG), and object-oriented. Give specific reasons and examples to illustrate your points.

(b) Summarize your answer as a matrix with data models listed on the vertical axis and features listed on the horizontal axis. Fill in the matrix elements with a check to indicate compliance, an X to indicate non-compliance, and a question mark to indicate unknown.

4. Equivalence of Relational Query Languages

(a) Identify a minimal set of relational operators.

(b) Prove that every query of the relational algebra is equivalent to a query in the domain relational calculus.

(c) Explain the concept of safety. How does it impact the other direction of the proof of equivalence of relational algebra and relational calculus?
5. **Functional Dependencies**

Let \( R \) be a relation schema; \( \mathcal{F}, \mathcal{G} \), sets of functional dependencies; and \( X \), a set of attributes of \( R \) (i.e., \( X \subseteq R \)).

Recall Armstrong's axioms:

**Reflexivity:** If \( Y \subseteq X \subseteq R \), then \( X \rightarrow Y \) follows from \( F \) by Armstrong's axioms.

**Augmentation:** If \( X \rightarrow Y \) follows from \( F \) by Armstrong's axioms and \( Z \subseteq R \), then \( XZ \rightarrow YZ \) follows from \( F \) by Armstrong's axioms.

**Transitivity:** If \( X \rightarrow Y \) and \( Y \rightarrow Z \) follow from \( F \) by Armstrong's axioms, then \( X \rightarrow Z \) also follows from \( F \) using Armstrong's axioms.

**Union:** If \( X \rightarrow Y \) and \( X \rightarrow Z \) follow from \( F \) by Armstrong's axioms, then \( X \rightarrow YZ \) follows from \( F \) by Armstrong's axioms.

**Pseudotransitivity:** If \( X \rightarrow Y \) and \( WY \rightarrow Z \) follow from \( F \) by Armstrong's axioms, then \( WX \rightarrow Z \) follows from \( F \) by Armstrong's axioms.

**Decomposition:** If \( X \rightarrow Y \) follows from \( F \) by Armstrong's axioms and \( Z \subseteq Y \), then \( X \rightarrow Z \) follows from \( X \) by Armstrong's axioms.

It is understood that every dependency in \( F^+ \) follows from \( F \) by Armstrong's axioms.

(a) Prove the Decomposition axiom from the Reflexivity, Augmentation, and Transitivity axioms.

(b) Let \( R = \{A, B, C, D, E, F\} \).

Let \( \mathcal{F} = \{AB \rightarrow CDE, AC \rightarrow D, AD \rightarrow B, E \rightarrow F\} \).

Is \( AE \rightarrow DF \in \mathcal{F}^+ \)? Prove your answer.

6. **Physical Storage**

(a) Describe an extensible (or dynamic) hashing algorithm.

(b) Using your algorithm, make the necessary assumptions and insert the following keys: 10, 14, 40, 88, 97, 62, 33, 20, 2, 18. Draw the appropriate structure and label each insertion clearly.

(c) Describe how a query of the following form can be evaluated when the data is stored in an unsorted file and there is a hash function on the name field.

```sql
SELECT name
FROM student
WHERE name > "Jones" AND name < "Williams"
```
Course Purpose: The development and use of highly structured databases. The focus is on agents that support the solution of computer problems requiring the use of large amounts of interrelated data. Particularly, data that are stored in a complex manner and are used to develop information. The emphasis is on theoretical, underlying issues; a practicum in database programming is not provided.

Course Materials:

Suggested Texts (available from the bookstore):
Elmasri, Navaehe, Database Systems, Addison-Wesley
Martin, Managing the Data-Base Environment, Prentice-Hall
Ullman, Principles of Database and Knowledge-Based Systems, vol 1, Computer Science Press

Supplemental Books (at the library, held in reserve):
Date, Database, vol 1, Addison-Wesley
Martin, Database Principles, Prentice-Hall

Supplementary Journal Articles (found in journals in the library):
Codd, E.F., “A Relational Model For Large Shared Data Banks,” Communications ACM, v 13 n 6, p377-387
Kent, W., “A Simple Guide To Five Normal Forms In Relational Database Theory,” Communications ACM, v 26, n2, p120-126

Topic List:
The following is a list of the topics included in this course.

<table>
<thead>
<tr>
<th>subject</th>
<th>Elmasri/Navathe</th>
<th>Martin</th>
<th>Ullman</th>
</tr>
</thead>
<tbody>
<tr>
<td>definition of a DBMS</td>
<td>1-36</td>
<td>1-69</td>
<td>1-31</td>
</tr>
<tr>
<td>DBMS capabilities</td>
<td>↓</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>DBMS issues, facilities, paradigms</td>
<td>1-36</td>
<td>1-69</td>
<td>1-31</td>
</tr>
<tr>
<td>DBMS advantages/disadvantages</td>
<td>↓</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>building logical records</td>
<td>171-181</td>
<td></td>
<td></td>
</tr>
<tr>
<td>traditional data organization and retrieval</td>
<td>65-132</td>
<td></td>
<td>294-306</td>
</tr>
<tr>
<td>sequential, index sequential (ISAM)</td>
<td>↓</td>
<td>310-313</td>
<td>↓</td>
</tr>
<tr>
<td>direct (random), hashing</td>
<td>↓</td>
<td></td>
<td>306-310</td>
</tr>
<tr>
<td>secondary index structures</td>
<td>267-268</td>
<td>310-322</td>
<td></td>
</tr>
<tr>
<td>semantic data modeling</td>
<td>253-352</td>
<td>171-202, 235-236</td>
<td>32-71</td>
</tr>
<tr>
<td>needs of data modeling</td>
<td>↓</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>schemas and subschemas</td>
<td>↓</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>data models</td>
<td>↓</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>flat files</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>trees: homogeneous, heterogeneous</td>
<td>253-281</td>
<td></td>
<td>342-346</td>
</tr>
<tr>
<td>CODASYL</td>
<td>↓</td>
<td>102,109</td>
<td>240-246</td>
</tr>
<tr>
<td>networks</td>
<td>281-327</td>
<td></td>
<td>342-346</td>
</tr>
<tr>
<td>independence (from Codd)</td>
<td>23-28</td>
<td></td>
<td>376-377</td>
</tr>
<tr>
<td>normal forms (also summarized in Kent)</td>
<td>133-174</td>
<td>203-226</td>
<td>376-401</td>
</tr>
<tr>
<td>definition</td>
<td>↓, 355-373</td>
<td></td>
<td></td>
</tr>
<tr>
<td>operations on relations (relational algebra)</td>
<td>145-160</td>
<td></td>
<td>53-61</td>
</tr>
<tr>
<td>first</td>
<td>133-174, 373-376</td>
<td>203-226</td>
<td>376-401</td>
</tr>
<tr>
<td>second</td>
<td>↓, 376-386</td>
<td>↓</td>
<td>401-412</td>
</tr>
<tr>
<td>third (including Boyce-Codd)</td>
<td>↓, 376-378</td>
<td>227-234</td>
<td></td>
</tr>
<tr>
<td>fourth</td>
<td>↓, 387-403</td>
<td>↓</td>
<td>420-423</td>
</tr>
<tr>
<td>fifth</td>
<td>↓, 403-404</td>
<td></td>
<td></td>
</tr>
<tr>
<td>object oriented model</td>
<td>435-449</td>
<td></td>
<td>21-23, 82-87</td>
</tr>
<tr>
<td>distributed databases</td>
<td>611-634</td>
<td>95-96,616-617</td>
<td>543-582</td>
</tr>
<tr>
<td>database reorganization (drawn from Sokut)</td>
<td>712-726</td>
<td></td>
<td></td>
</tr>
<tr>
<td>security</td>
<td>540-546, 588-596</td>
<td>587-607</td>
<td>446-466</td>
</tr>
</tbody>
</table>
name (please print): ________________________________
signature: _________________________________________
student id or social security number: ________________

INSTRUCTIONS

(1) All questions have an equal value.
(2) You may skip three questions. Cross out the questions that you wish to skip. If you do not cross out any, it will be assumed that you wish to skip the last questions.
(3) Questions asking you to provide an example require you to provide realistic data: i.e., names and relationships that might have meaning. Examples must not be drawn from class, a group review session, or from the course’s textbooks.
(4) In all cases where you are asked to display a normal form, your answer should contain the smallest possible count of separate relations.
(5) In all logical schemas, you must label relationship lines. Labels must be unique and meaningful. Where appropriate, you must underline key(s).
(6) You may not use more than 25 words to answer any question. If you do use more than 25 words, your score will be decreased.
(7) If you use pencil, your exam will not be considered for regrading.
(8) You may use a dictionary.
(9) Breath deeply before proceeding.
(10) Good Luck.

<table>
<thead>
<tr>
<th>question</th>
<th>value received</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
</tr>
</tbody>
</table>
[23] Provide a "real life" example of an application problem that would occur in a first normal form that would not occur in the second normal form. You must specifically identify the problem. Your example must include both the first and second normal forms. Your example should be other than one presented in class or in your textbook.

[24] Given the following relation, expressed as (City, E#, EN, Skill, $, Age) with the typical data shown below, construct the 2NF. Show the answer in the notational form.

<table>
<thead>
<tr>
<th>City</th>
<th>E#</th>
<th>EN</th>
<th>Skill</th>
<th>$</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>NYC</td>
<td>3</td>
<td>John</td>
<td>3</td>
<td>4</td>
<td>24</td>
</tr>
<tr>
<td>LA</td>
<td>4</td>
<td>Sally</td>
<td>3</td>
<td>4</td>
<td>27</td>
</tr>
<tr>
<td>LA</td>
<td>2</td>
<td>Bill</td>
<td>4</td>
<td>5</td>
<td>24</td>
</tr>
<tr>
<td>LA</td>
<td>3</td>
<td>John</td>
<td>6</td>
<td>9</td>
<td>24</td>
</tr>
<tr>
<td>NYC</td>
<td>7</td>
<td>Karen</td>
<td>2</td>
<td>6</td>
<td>19</td>
</tr>
<tr>
<td>Chi</td>
<td>2</td>
<td>Bill</td>
<td>3</td>
<td>4</td>
<td>27</td>
</tr>
</tbody>
</table>

[25] Given the following relation:
(Tax, $, Farm, State, Sire, Dam, Hip#, name, Fee)
Show the 3NF. Your answer should have as few component parts as possible. Show your answer in notational form. Make sure to underline your key(s).

<table>
<thead>
<tr>
<th>Tax</th>
<th>$</th>
<th>Farm</th>
<th>State</th>
<th>Sire</th>
<th>Dam</th>
<th>Hip#</th>
<th>name</th>
<th>Fee</th>
</tr>
</thead>
<tbody>
<tr>
<td>4%</td>
<td>500</td>
<td>Spendthrift</td>
<td>KY</td>
<td>Bob</td>
<td>Ann</td>
<td>5</td>
<td>Black</td>
<td>5</td>
</tr>
<tr>
<td>4%</td>
<td>600</td>
<td>Claybrook</td>
<td>KY</td>
<td>Ted</td>
<td>Mary</td>
<td>7</td>
<td>Spot</td>
<td>6</td>
</tr>
<tr>
<td>4%</td>
<td>700</td>
<td>White</td>
<td>OH</td>
<td>John</td>
<td>Sue</td>
<td>9</td>
<td>Spot</td>
<td>7</td>
</tr>
<tr>
<td>5%</td>
<td>600</td>
<td>Lucky</td>
<td>TN</td>
<td>Ray</td>
<td>Sue</td>
<td>9</td>
<td>Black</td>
<td>6</td>
</tr>
<tr>
<td>4%</td>
<td>550</td>
<td>Spendthrift</td>
<td>KY</td>
<td>Bob</td>
<td>Ann</td>
<td>4</td>
<td>Nasty</td>
<td>5</td>
</tr>
<tr>
<td>4%</td>
<td>500</td>
<td>White</td>
<td>OH</td>
<td>Ralph</td>
<td>Lily</td>
<td>1</td>
<td>Nasty</td>
<td>5</td>
</tr>
<tr>
<td>4%</td>
<td>750</td>
<td>Claybrook</td>
<td>KY</td>
<td>Bill</td>
<td>Sally</td>
<td>2</td>
<td>Fast</td>
<td>7</td>
</tr>
<tr>
<td>4%</td>
<td>550</td>
<td>Spendthrift</td>
<td>KY</td>
<td>Bob</td>
<td>Zoe</td>
<td>8</td>
<td>Quick</td>
<td>5</td>
</tr>
<tr>
<td>5%</td>
<td>600</td>
<td>Lucky</td>
<td>TN</td>
<td>Ray</td>
<td>Zoe</td>
<td>8</td>
<td>Sludge</td>
<td>6</td>
</tr>
</tbody>
</table>
[1] Identify two important ways that access to data and the organization of the data in a DBMS (database management system) is different than how the data is accessed and how the data is organized in a multi-file system that does not make use of a DBMS. (Note: this asks how they are different, it does not ask you to define what is done.)

[2] Which of the following can be said to be True of normalization or particular normal forms? Answer by circling the letter of the correct answers. (Grading: +1 for every correct answer circled, -1 for every wrong answer circled, 0 if not circled.)
   (a) Normalization reduces hierarchy
   (b) Moving from 1NF to 2NF introduces partial dependencies
   (c) A 3NF is a flat file
   (d) A 3NF is in 2NF
   (e) Repeating attribute names are OK as long as the values are different
   (f) A candidate key is a collection of attributes whose value uniquely identifies each column
   (g) A relation is in 1NF if and only if every determinant is a candidate key
   (h) A Boyce-Codd normal form is a weaker normal form than 3NF
   (i) Row order does not matter
   (j) All rows must be distinct
   (k) Attribute values are consistently non-atomic

[3] True or False: In databases, the term integrity constraint means something done to keep the database and DBMS user’s morally honest and/or ethical. Explain your answer of True or False. Your explanation should contain an example. Both parts must be correct to receive credit for the question.

[4] Given the following relationships

\[ R_1 = \begin{array}{ccc} X & Y & Z \\ A & D & G \\ B & E & H \\ C & F & J \end{array} \quad R_2 = \begin{array}{ccc} M & X & Y & N \\ K & A & D & R \\ C & L & F & P \end{array} \]

what is the result of the natural join of \( R_1 \) and \( R_2 \)? The join can be expressed as \( R_1 \Join R_2 \).
[5] Given: a very large collection (or file) of identically formatted physical records. Each record contains the attributes: employee number, employee name, birth date, age, title, year-to-date compensation, citizenship. Each employee number is unique. The records are stored sequentially and ordered on employee number. Which of the following things can be said to be true? Answer by circling the letter of the correct answers. (Grading: +1 for every correct answer circled, -1 for every wrong answer circled, 0 if not circled.)

(a) employee number is a good attribute for the primary key for the file.
(b) employee number is a good attribute to use as a secondary index.
(c) year-to-date compensation would be a good attribute for the primary key for the file.
(d) year-to-date compensation is a good attribute to use as a secondary index.
(e) title is a good attribute to use as the primary key for the file.
(f) title is a good attribute to use as a secondary index.
(g) The data can be stored in a flat file.
(h) citizenship would not be a good ISAM index attribute.
(i) It is not possible to construct a good ISAM index.
(j) employee number is the only simple candidate key.
(k) If the data would be stored using a hashing function, applications similar to payroll would run faster.

[6] What type of applications are best served by using hashing to calculate where physical records are to be stored? Illustrate your answer by specifically listing the names of at least two such applications. The applications named should be essentially different than those presented in class or found in a course specified textbook.

[7] Given the following flat file, show its logical schema using the bubble chart notation. Your schema should be as simplest possible. This means that the schema should have as few lines as possible between the bubbles. Similarly, the schema should have as few arrows as possible. Make sure to label all of the relationship arcs.

<table>
<thead>
<tr>
<th>AGE</th>
<th>EN</th>
<th>SSN</th>
<th>NAME</th>
<th>CITY</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>25</td>
<td>381-23-1654</td>
<td>BOB</td>
<td>CINCINNATI</td>
<td>OHIO</td>
</tr>
<tr>
<td>22</td>
<td>32</td>
<td>281-41-2543</td>
<td>SUE</td>
<td>COLUMBUS</td>
<td>OHIO</td>
</tr>
<tr>
<td>32</td>
<td>19</td>
<td>931-98-7651</td>
<td>JUDY</td>
<td>ST. LOUIS</td>
<td>MISSOURI</td>
</tr>
<tr>
<td>39</td>
<td>54</td>
<td>451-23-6544</td>
<td>LARRY</td>
<td>GARY</td>
<td>INDIANA</td>
</tr>
<tr>
<td>15</td>
<td>65</td>
<td>231-45-1643</td>
<td>KAREN</td>
<td>CHICAGO</td>
<td>ILLINOIS</td>
</tr>
<tr>
<td>16</td>
<td>48</td>
<td>561-42-3164</td>
<td>AKIM</td>
<td>ATLANTA</td>
<td>GEORGIA</td>
</tr>
<tr>
<td>82</td>
<td>56</td>
<td>743-42-3465</td>
<td>SAMUEL</td>
<td>GARY</td>
<td>INDIANA</td>
</tr>
<tr>
<td>12</td>
<td>18</td>
<td>450-45-3782</td>
<td>SUE</td>
<td>BERKELEY</td>
<td>CALIFORNIA</td>
</tr>
</tbody>
</table>

EN = employee number | SSN = social security number
[8] Given the following logical schema, replace the arbitrary symbols:
A, B, C, D, E, F, G
by attributes that would represent a meaningful and reasonable collection of "real live data." All of the attribute names must be different. Your answer must be a "good" data design. Place your attributes next to the existing symbols. Remember to label all relationship lines. You do not have to show the actual data, only produce reasonable attribute names.

[9] Given the following data table, draw its logical schema. Use a bubble chart. Make sure to label all of the relationships.

<table>
<thead>
<tr>
<th>Color</th>
<th>Product</th>
<th>Company</th>
<th>Price</th>
<th>Qty</th>
</tr>
</thead>
<tbody>
<tr>
<td>chocolate</td>
<td>cake</td>
<td>Wholsome</td>
<td>1.79</td>
<td>15</td>
</tr>
<tr>
<td>chocolate</td>
<td>cake</td>
<td>Healthy</td>
<td>1.89</td>
<td>13</td>
</tr>
<tr>
<td>dark</td>
<td>beer</td>
<td>Healthy</td>
<td>1.79</td>
<td>3</td>
</tr>
<tr>
<td>dark</td>
<td>beer</td>
<td>Labatts</td>
<td>3.49</td>
<td>5</td>
</tr>
<tr>
<td>chocolate</td>
<td>candy</td>
<td>Hershey</td>
<td>0.45</td>
<td>15</td>
</tr>
</tbody>
</table>

[10] Given the data table shown in the previous question, show its second normal form (2NF) in notational form.

[11] Given the following 2NF, put it into a 3NF. You need only show the notational form.

(Age, Name, Driver, City)

<table>
<thead>
<tr>
<th>Age</th>
<th>Name</th>
<th>Driver</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>Bill</td>
<td>No</td>
<td>NYC</td>
</tr>
<tr>
<td>24</td>
<td>Karen</td>
<td>Yes</td>
<td>NYC</td>
</tr>
<tr>
<td>15</td>
<td>Sally</td>
<td>No</td>
<td>Cincinnati</td>
</tr>
<tr>
<td>12</td>
<td>Ralph</td>
<td>No</td>
<td>Chicago</td>
</tr>
<tr>
<td>27</td>
<td>William</td>
<td>Yes</td>
<td>Chicago</td>
</tr>
</tbody>
</table>
[12] Given the displayed data, supply a plausible logical schema. Your schema should have as few boxes as possible. All lines should be labeled to indicate the nature of the relationship indicated by the line.

Note: These are all automobile parts and their price

[13] Which of the following things can be said to be true of the following table of data and the diagram? Do not assume that the diagram is necessarily related to the table of data.

Supplies

<table>
<thead>
<tr>
<th>PN</th>
<th>PN</th>
<th>S#</th>
<th>SN</th>
<th>$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>nut</td>
<td>1</td>
<td>Bob</td>
<td>0.12</td>
</tr>
<tr>
<td>1</td>
<td>nut</td>
<td>2</td>
<td>Sue</td>
<td>0.13</td>
</tr>
<tr>
<td>2</td>
<td>nut</td>
<td>1</td>
<td>Bob</td>
<td>0.13</td>
</tr>
<tr>
<td>2</td>
<td>nut</td>
<td>2</td>
<td>Sue</td>
<td>0.12</td>
</tr>
<tr>
<td>3</td>
<td>bolt</td>
<td>1</td>
<td>Bob</td>
<td>0.12</td>
</tr>
<tr>
<td>3</td>
<td>bolt</td>
<td>3</td>
<td>Bob</td>
<td>0.12</td>
</tr>
</tbody>
</table>

Answer by circling the letter of the correct answers. (Grading: +1 for every correct answer circled, -1 for every wrong answer circled, 0 if not circled.)

(a) The diagram is not any good as a logical schema because all the lines in the diagram only have one way arrows.
(b) The diagram is not any good as a logical schema because there is no node with only single arrows leaving it.
(c) There is at least one simple candidate key in the diagram.
(d) The only possible primary key is a concatenated key.
(e) The table of data is a flat file.
(f) The table of data is an instantiation of the diagram.
(g) The diagram is a tree data structure.
(h) The diagram is a hierarchical structure.
(i) The diagram is a homogeneous data structure.
(j) The values of the attribute "$" would be a good to create a secondary index on.
(k) No attributes would be good to create a secondary index on.
(l) If a database was constructed to hold the data, each record of the stored data would not have to contain the same data as a line of data in the table. That is, each database record would not have to contain PN#, PN, S#, SN, $.
(m) PN is functionally dependent on PN#.
[14] Given the following figure and the relationships that are specified in it, list all of the CODASYL sets. Use the minimum number of sets possible. (Graded right/wrong)

[Diagram: A graph with nodes labeled B, C, E, F, G, K, L, M, showing relationships with arrows labeled owner and member.]

[15] How many domains are there in the following relation? You just have to identify how many there are. You do not have to provide the domain attribute names. (This question will be graded on a credit/no credit basis.)

<table>
<thead>
<tr>
<th>employee name</th>
<th>age</th>
<th>boss's name</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>234 Kim</td>
<td>23</td>
<td>Jody</td>
<td>21</td>
</tr>
<tr>
<td>178 Jody</td>
<td>21</td>
<td>Bobby</td>
<td>41</td>
</tr>
<tr>
<td>438 Blacky</td>
<td>19</td>
<td>Jody</td>
<td>21</td>
</tr>
<tr>
<td>734 Bobby</td>
<td>41</td>
<td>Whitey</td>
<td>34</td>
</tr>
<tr>
<td>685 Fran</td>
<td>25</td>
<td>Bobby</td>
<td>41</td>
</tr>
</tbody>
</table>

[16] Which of the following can be said to be true of a relation? Answer by circling the letter of the correct answer. (Grading: +1 for every correct answer, -1 for every incorrect answer, 0 if not circled.)

(a) Once a relation table is established, it doesn't change.
(b) All tuples are distinct.
(c) The order the tuples are in doesn't matter.
(d) A candidate key uniquely identifies a tuple.
(e) Each tuple represents a single entity record.
(f) Each tuple can contain a different number of attributes.
(g) The entire table represents a relation.
(h) A candidate key uniquely identifies a domain.
(i) Each tuple derives its legal values from a domain.
(j) A relation eliminates redundancy of functional dependencies.
(k) The contents of columns are distinct.
Use the following data for the next two questions:

[Diagram]

[17] List all of the simple keys (attribute names). Do not list any concatenated keys.

[18] Show the logical schema. Represent your answers using as few distinct "boxes" as possible. Your answer must make sense. Make sure to label all relationship lines and underline the key(s).

[19] Provide a "real life" example of an application problem that would occur in a third normal form that would not occur in the Boyce-Codd normal form. You must specifically identify the problem. (By identifying the problem, you specify what can go wrong in BCNF that will not go wrong in 3NF.) Your example must include both the third and Boyce-Codd normal forms. Your example should be other than one presented in class or in your textbook.
[20] Assume that a record can be assigned a sequence number indicating the relative time when the record was entered into the system. For example, in a system designed to handle telephone orders for "Mister Microphones," the identifying number 1 is assigned to the first order for a microphone, the identifying number 2 is assigned to the second order for a microphone, the identifying number 3 is assigned to the third order for a microphone, etc. Assume that the sequence number describing the order will always be unique and will always uniquely separate one Mister Microphone order from another. (Note: this is, in fact, basically how many of the over the telephone ordering systems work.) Assume that the file is to be sequentially ordered and stored in a sequential file. Discuss the necessity of ordering the file on the record sequence number if the number uniquely identifies each record and it is the only attribute that can be used to uniquely identify each record.

[21] Given the following relation: (sname, supplier#, part#, supplier city, sales tax, warehouse stored in), Which of the following things can be said to be true? Answer by circling the letter of the correct answers. (Grading: +1 for every correct answer circled, -1 for every wrong answer circled, 0 if not circled.)

<table>
<thead>
<tr>
<th>supplier#</th>
<th>sales tax</th>
<th>warehouse stored in</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob</td>
<td>10</td>
<td>0.13</td>
</tr>
<tr>
<td>Bob</td>
<td>11</td>
<td>0.14</td>
</tr>
<tr>
<td>Sue</td>
<td>12</td>
<td>0.12</td>
</tr>
<tr>
<td>Sue</td>
<td>12</td>
<td>0.13</td>
</tr>
<tr>
<td>Bob</td>
<td>10</td>
<td>0.11</td>
</tr>
<tr>
<td>Bob</td>
<td>11</td>
<td>0.12</td>
</tr>
</tbody>
</table>

(a) The highest normal form is 1.
(b) Is in 3NF if the attribute "sname" and its values are removed.
(c) Has a concatenated key.
(d) Has a transitive dependency.
(e) Would NOT be normalized if the attribute "warehouse stored in" and its data are removed.
(f) The attribute "price" is functionally dependent on a concatenated key.
(g) The attribute "supplier city" is functionally dependent on a simple key.
(h) The attribute "supplier name" is functionally dependent on a concatenated key.
(i) Has repeating groups.
(j) All functional dependencies are on concatenated keys.
(k) In order to get to 3NF, the relation has to be decomposed into more than two relations.

[22] Given the following relation, SUPPLIES, what would be the resulting value of Q?

\[ Q \leftarrow \pi_{item, price} \sigma_{\text{weight}=15 \land \text{price}<6} (\text{SUPPLIES}) \]

<table>
<thead>
<tr>
<th>supplier#</th>
<th>item</th>
<th>weight</th>
<th>price</th>
<th>city</th>
<th>tax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kroger</td>
<td>Cheddar</td>
<td>16</td>
<td>2.39</td>
<td>Cincinnati</td>
<td>5.5</td>
</tr>
<tr>
<td>Acme</td>
<td>Brie</td>
<td>15</td>
<td>4.56</td>
<td>Toulon</td>
<td>6.0</td>
</tr>
<tr>
<td>Biggs</td>
<td>Brie</td>
<td>22</td>
<td>6.89</td>
<td>Cincinnati</td>
<td>5.5</td>
</tr>
<tr>
<td>Acme</td>
<td>Brie</td>
<td>22</td>
<td>8.42</td>
<td>Pamplona</td>
<td>4.0</td>
</tr>
<tr>
<td>Kroger</td>
<td>Swiss</td>
<td>15</td>
<td>5.49</td>
<td>Dayton</td>
<td>5.0</td>
</tr>
<tr>
<td>Acme</td>
<td>Swiss</td>
<td>15</td>
<td>5.49</td>
<td>Toulon</td>
<td>6.0</td>
</tr>
<tr>
<td>Biggs</td>
<td>Havarti</td>
<td>16</td>
<td>3.48</td>
<td>Cincinnati</td>
<td>5.5</td>
</tr>
<tr>
<td>Acme</td>
<td>Brie</td>
<td>8</td>
<td>3.48</td>
<td>Kabul</td>
<td>0.0</td>
</tr>
<tr>
<td>Kroger</td>
<td>Brie</td>
<td>8</td>
<td>3.26</td>
<td>Dayton</td>
<td>5.0</td>
</tr>
<tr>
<td>Acme</td>
<td>Brie</td>
<td>12</td>
<td>4.56</td>
<td>Kandahar</td>
<td>0.0</td>
</tr>
</tbody>
</table>
III. Data Base Theory (551)

15-625-551:CS551:Database Theory

course description

Course Purpose: The development and use of highly structured databases. The focus is on agents that support the solution of computer problems requiring the use of large amounts of interrelated data. Particularly, data that are stored in a complex manner and are used to develop information. The emphasis is on theoretical, underlying issues; a practicum in database programming is not provided.

Course Materials:

Suggested Texts (available from the bookstore):
- Elmasri, Navathe, Database Systems, Addison-Wesley
- Martin, Managing the Data-Base Environment, Prentice-Hall
- Ullman, Principles of Database and Knowledge-Base Systems, vol 1, Computer Science Press

Supplemental Books (at the library, held in reserve):
- Date, Database, vol 1, Addison-Wesley
- Martin, Database Principles, Prentice-Hall

Supplementary Journal Articles (found in journals in the library):
- Codd, E.F., "A Relational Model For Large Shared Data Banks," Communications ACM, v 13 n 6, p377-387
- Kent,W., "A Simple Guide To Five Normal Forms In Relational Database Theory," Communications ACM, v 26, n2, p120-126

Topic List:
The following is a list of the topics included in this course.

<table>
<thead>
<tr>
<th>subject</th>
<th>Elmasri/Navathe</th>
<th>Martin</th>
<th>Ullman</th>
</tr>
</thead>
<tbody>
<tr>
<td>definition of a DBMS</td>
<td>1-36</td>
<td>1-69</td>
<td>1-31</td>
</tr>
<tr>
<td>DBMS capabilities</td>
<td>↓</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>DBMS issues, facilities, paradigms</td>
<td>1-36</td>
<td>1-69</td>
<td>1-31</td>
</tr>
<tr>
<td>DBMS advantages/disadvantages</td>
<td>↓</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>building logical records</td>
<td></td>
<td>171-181</td>
<td></td>
</tr>
<tr>
<td>traditional data organization and retrieval</td>
<td>65-132</td>
<td></td>
<td>294-306</td>
</tr>
<tr>
<td>sequential, index sequential (ISAM)</td>
<td>↓</td>
<td>310-313</td>
<td></td>
</tr>
<tr>
<td>direct (random), hashing</td>
<td>↓</td>
<td></td>
<td>306-310</td>
</tr>
<tr>
<td>secondary index structures</td>
<td></td>
<td>267-268</td>
<td>310-322</td>
</tr>
<tr>
<td>semantic data modeling</td>
<td>253-352</td>
<td>171-202, 235-236</td>
<td>32-71</td>
</tr>
<tr>
<td>needs of data modeling</td>
<td>↓</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>schemas and subschemas</td>
<td>↓</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>data models</td>
<td>↓</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>flat files</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>trees: homogeneous, heterogeneous</td>
<td>253-281</td>
<td></td>
<td>342-346</td>
</tr>
<tr>
<td>CODASYL</td>
<td>↓</td>
<td>102,109</td>
<td>240-246</td>
</tr>
<tr>
<td>networks</td>
<td>281-327</td>
<td></td>
<td>342-346</td>
</tr>
<tr>
<td>independence (from Codd)</td>
<td>23-28</td>
<td></td>
<td>376-377</td>
</tr>
<tr>
<td>normal forms (also summarized in Kent)</td>
<td>133-174</td>
<td>203-226</td>
<td>376-401</td>
</tr>
<tr>
<td>definition</td>
<td>↓, 355-373</td>
<td></td>
<td></td>
</tr>
<tr>
<td>operations on relations (relational algebra)</td>
<td>145-160</td>
<td></td>
<td>53-61</td>
</tr>
<tr>
<td>first</td>
<td>133-174, 373-376</td>
<td>203-226</td>
<td>376-401</td>
</tr>
<tr>
<td>second</td>
<td>↓, 376-386</td>
<td>↓</td>
<td>401-412</td>
</tr>
<tr>
<td>third (including Boyce-Codd)</td>
<td>↓, 376-378</td>
<td>227-234</td>
<td>↓</td>
</tr>
<tr>
<td>fourth</td>
<td>↓, 387-403</td>
<td>↓</td>
<td>420-423</td>
</tr>
<tr>
<td>fifth</td>
<td>↓, 403-404</td>
<td>↓</td>
<td></td>
</tr>
<tr>
<td>object oriented model</td>
<td>435-449</td>
<td></td>
<td>21-23, 82-87</td>
</tr>
<tr>
<td>distributed databases</td>
<td>611-634</td>
<td>95-96,616-617</td>
<td>543-582</td>
</tr>
<tr>
<td>database reorganization (drawn from Sokut)</td>
<td></td>
<td>712-726</td>
<td></td>
</tr>
<tr>
<td>security</td>
<td>540-546, 588-596</td>
<td>587-607</td>
<td>446-466</td>
</tr>
</tbody>
</table>
that could occur in second normal form that would not occur in third normal form. Make sure to identify what could go wrong if the third normal form was not reached.

(7) Normalization removes various kinds of data anomalies. The potential for a different type of data insertion anomaly is removed when moving between each level of normalization. Data insertion anomalies occur when either (a) new data is added or (b) existing data is modified, but not deleted. Identify and display a specific, real life example of a data insertion anomaly that could occur in third normal form that would not occur in fourth normal form. Make sure to identify what could go wrong if the fourth normal form was not reached.

(8) **True or False:** Non-loss decomposition means that when relations are decomposed, no data values are lost. **Explain.** Both parts of your answer must be correct to receive credit. As part of your explanation, provide an example.

(9) **Background:** The CEO of your company is supremely confident in his personal knowledge. This means that you can never explain things to him in terms that he does not understand (or, at least, if you wish to keep your $200,000/year job, company car, and company condo). As the corporate CIO (Chief Information Officer), you have just authorized the conversion of your IBM hierarchical IMS database to a relational database (also an IBM product as your president has a close relationship with IBM’s current CEO - a man who came from a cookie company - True!!!)

**QUESTION:** In simple, general terms that even a non-computer trained person could understand, what is the advantage of normalization?

(10) Provide a "real life" example of an application problem that would occur in a third normal form that would not occur in the fourth normal form. You must specifically identify the problem. (By identifying the problem, you specify what can go wrong in 3NF that will not go wrong in 4NF.) Your example must include both the third and fourth normal forms. Your example should be other than one presented in class or in your textbook.

(11) Create a "real life" example of data that is stored in fourth normal form (4NF) that is not stored in fifth normal form (5NF). Use the fewest possible relations to construct the normal form. Show both the 4NF and 5NF and explain why the 4NF is not in 5NF. Your example may not be one presented in class materials.
(12) Relations with cyclic constraints are not in fifth normal form. **True or False:** Cyclic constraints are essentially looping network structures.

(13) **True or False:** The term "networked heterogeneous databases" refers to multiple databases that use a network data model to store records of more than one data type. **Explain** your answer of True or False. Both parts must be correct to receive credit for the question.

(14) **True or False:** Database concurrency is not a problem when there is only a single database instead of distributed databases because there is only a single place where the data is stored. **Explain.** (Both parts of your answer must be correct to receive credit for this question.)

(15) **True or False:** A lattice data type (found in object oriented databases as well as in other places), contains mutually exclusive type components. **Explain** your answer of True or False. As part of your explanation, illustrate your answer by an example. The example should be essentially different than one presented in class or found in a course specified textbook. Both parts must be correct to receive credit for the question.
There are three main questions. Each of the three questions is equal valued.

Each question is broken into several parts. Within the same question, each part is equal valued.
1. This question is on basic database issues. There are several parts to this question. Within this question, each part of this question is equal valued. You must do all of the parts of the question when answering this question.

[1-a] Identify two important ways that access to data and the organization of the data in a DBMS (database management system) is different than how the data is accessed and how the data is organized in a multi-file system that does not make use of a DBMS. (Note: this asks how they are different, it does not ask you to define what is done.)

[1-b] One of the things that a database does for you is that it reduces redundancy. Other than reduction in needed storage, identify at least two reasons why reducing redundancy is a good thing.

[1-c] True or False: The DBMS provides tools that help the end user to directly and explicitly manipulate the stored data. Explain your answer of True or False. Both parts must be correct to receive credit for the question.

[1-d] True or False: In the context of databases, the term data agent and the term database administrator are equivalent. Explain your answer of true or false. Both parts of the answer must be correct to receive credit for the question.

[1-e] True or False: In databases, the term integrity constraint means something done to keep the database and DBMS user's morally honest and/or ethical. Explain your answer of True or False. Your explanation should contain an example. Both parts must be correct to receive credit for the question.

[1-f] True or False: Other than restricting access to particular fields, It is useful to have two distinctly different user logical views of the same collection of data. Explain your answer of True or False. As part of your explanation, illustrate your answer by an example. Both parts must be correct to receive credit for the question.
2. **This question is on normalization issues.** There are several parts to this question. Within this question, each part of this question is equal valued. You must do all of the parts of the question when answering this question. Most of the parts of this question ask you to provide examples. This means that you are to provide explicit examples that could be drawn from "real life."

**[2-a] Background:** The CEO of your company is supremely confident in his personal knowledge. This means that you can never explain things to him in terms that he does not understand (or, at least, if you wish to keep your $200,000/year job, company car, and company condo). As the corporate CIO (Chief Information Officer), you have just authorized the conversion of your IBM hierarchical IMS database to a relational database (also an IBM product as your president has a close relationship with IBM's current CEO - a man who came from a cookie company - True!!!)

**Test Question:** In simple, general terms that even a non-computer trained person could understand, what is the advantage of normalization?

**[2-b] True or False:** In normalization, the higher the normalization level (i.e., 2NF is higher than 1NF, 3NF is higher than 2NF, etc.) the more complex the dependencies in the relation. Another way of saying this is that relations that are in the highest normal form have more complex data dependencies than relations in a lower normal form. **Explain** your answer of True or False. Both parts must be correct to receive credit for the question.

**[2-c] The term "non-loss decomposition" is relevant to normalization.** When is it relevant? Identify and display a specific, real life example where "non-loss decomposition" is relevant.
Normalization removes various kinds of data anomalies. The potential for a different type of data insertion anomaly is removed when moving between each level of normalization. Data insertion anomalies occur when either (a) new data is added or (b) existing data is modified, but not deleted. Identify and display a specific, real life example of a data insertion anomaly that could occur in second normal form that would not occur in third normal form. Make sure to identify what could go wrong if the third normal form was not reached.

Normalization removes various kinds of data anomalies. The potential for a different type of data insertion anomaly is removed when moving between each level of normalization. Data insertion anomalies occur when either (a) new data is added or (b) existing data is modified, but not deleted. Identify and display a specific, real life example of a data insertion anomaly that could occur in third normal form that would not occur in fourth normal form. Make sure to identify what could go wrong if the fourth normal form was not reached.
3. This question focuses on data and data organization. There are several parts to this question. Within this question, each part of this question is equal valued. You must do all of the parts of the question when answering this question.

[3-a] Describe a specific application of the use of a "real life" database where indexing of the database on an attribute other than the primary key would be useful. (Items in your example do not count towards your maximum word limitation.) Explain why it would be useful.

[3-b] True or False: There is only one reasonable way to hierarchically organize data. Explain, using logical schemas drawn from "real life". (Your explanation must be correct to receive any credit for this question.) Your example should have at least three levels of hierarchy.

[3-c] True or False: The term "networked heterogeneous databases" refers to multiple databases that use a network data model to store records of more than one data type. Explain your answer of True or False. Both parts must be correct to receive credit for the question.

[3-d] True or False: Database concurrency control means keeping the logical database model consistent with the real or external world. Explain. (Both parts of your answer must be correct to receive credit for this question.)

[3-e] Provide a specific example of object data stored for an object oriented database. Your example should not be storable in a hierarchical or relational database.
Qualifying Test
Database Theory
November 12, 1994

- The test is closed book and closed notes.
- The total time duration is 60 minutes.
- Each question carries 25 points.
- To disprove something, it is enough if you give a counter-example.
- To prove something, an example won't suffice.
- Do not forget to write your Name and Social-Security Number.
- Do not start working on the test until you are told to do so.

Name:__________________________  SSN:__________________________
1. Questions below are concerned with the definitions of keys and primary keys.

(a) 10 points
Given a relation scheme $R[A_1, A_2, \ldots, A_n]$, what is the maximum number of primary keys $R$ can have?

(b) 3 points
Can the union of two keys be a key? Justify your answer.

(c) 4 points
Is the intersection of two keys necessarily a key? Justify your answer.

(d) 8 points
If $K$ is a key for instances $R$ and $S$ of a relation scheme $\mathcal{R}$, then which of the following must necessarily have a key $K$?

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) $R \cap S$</td>
<td>:Y/N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(b) $\Pi_A(R)$ where $A \subseteq K$</td>
<td>:Y/N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c) $(R \times S) \div S$</td>
<td>:Y/N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(d) $R \bowtie S$</td>
<td>:Y/N</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2. Let \( A \) be the attribute set of a relation scheme \( R \), and let \( X \subseteq A \). Let \( P \) and \( Q \) be some relation instances of \( R \). Then, prove or disprove the following.

(a) 13 points
\[ \Pi_X(P \cap Q) = \Pi_X(P) \cap \Pi_X(Q) \]

(b) 12 points
\[ \Pi_X(P - Q) = \Pi_X(P) - \Pi_X(Q) \]
3. Consider the following suppliers–parts–projects database.

<table>
<thead>
<tr>
<th>Table</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suppliers</td>
<td>S (s#, sname, status, city)</td>
</tr>
<tr>
<td>Parts</td>
<td>P (p#, pname, color, weight, city)</td>
</tr>
<tr>
<td>Projects</td>
<td>J (j#, jname, city)</td>
</tr>
<tr>
<td>Sup–Part–Proj</td>
<td>SPJ (s#, p#, j#, qty)</td>
</tr>
</tbody>
</table>

Answer the following queries in Relational Algebra and Tuple Relational Calculus.

“Get J# values for projects supplied entirely by supplier (s#) S1.” (Note that J1, say, is a valid project if every part used by J1 is supplied by S1 to J1 and that some of the parts used by J1 may also be supplied by a supplier different from S1.)

(a) 15 points

Relational Algebra:

(b) 10 points

Tuple Relational Calculus:
4. For the following questions, give (at least) convincing arguments.

(a) 15 points
Show that if \( R \) is a relation scheme and \( X \subseteq R \) is a key with respect to set \( a \) of functional dependencies \( F \), then \( X \) can not have a 3NF violation with respect to the set of dependencies \( \Pi_X(F) \).

(b) 10 points
Show that if \( X \rightarrow Y \) holds, then \( X \Rightarrow Y \) holds.
Qualifying Test
Database Theory
November 12, 1994

- The test is closed book and closed notes.
- The total time duration is 60 minutes.
- Each question carries 25 points.
- To disprove something, it is enough if you give a counter-example.
- To prove something, an example won’t suffice.
- Do not forget to write your Name and Social-Security Number.
- Do not start working on the test until you are told to do so.

Name:_________________________ SSN:________________
1. Questions below are concerned with the definitions of keys and primary keys.

(a) 10 points
Given a relation scheme $R[A_1, A_2, \ldots, A_n]$, what is the maximum number of primary keys $R$ can have?

(b) 3 points
Can the union of two keys be a key? Justify your answer.

(c) 4 points
Is the intersection of two keys necessarily a key? Justify your answer.

(d) 8 points
If $K$ is a key for instances $R$ and $S$ of a relation scheme $\mathcal{R}$, then which of the following must necessarily have a key $K$?

(a) $R \cap S$ : Y/N  
(b) $\Pi_A(R)$ where $A \subseteq K$ : Y/N  
(c) $(R \times S) \div S$ : Y/N  
(d) $R \bowtie S$ : Y/N
2. Let $A$ be the attribute set of a relation scheme $R$, and let $X \subseteq A$. Let $P$ and $Q$ be some relation instances of $R$. Then, prove or disprove the following.

(a) 13 points
$$\Pi_X(P \cap Q) = \Pi_X(P) \cap \Pi_X(Q)$$

(b) 12 points
$$\Pi_X(P - Q) = \Pi_X(P) - \Pi_X(Q)$$
3. Consider the following suppliers–parts–projects database.

<table>
<thead>
<tr>
<th>Suppliers</th>
<th>S (s#, sname, status, city)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parts</td>
<td>P (p#, pname, color, weight, city)</td>
</tr>
<tr>
<td>Projects</td>
<td>J (j#, jname, city)</td>
</tr>
<tr>
<td>Sup–Part–Proj</td>
<td>SPJ (s#, p#, j#, qty)</td>
</tr>
</tbody>
</table>

Answer the following queries in Relational Algebra and Tuple Relational Calculus.

"Get J# values for projects supplied entirely by supplier (s#) S1." (Note that J1, say, is a valid project if every part used by J1 is supplied by S1 to J1 and that some of the parts used by J1 may also be supplied by a supplier different from S1.)

(a) 15 points

Relational Algebra:

(b) 10 points

Tuple Relational Calculus:
4. For the following questions, give (at least) convincing arguments.

(a) 15 points
   Show that if $R$ is a relation scheme and $X \subseteq R$ is a key with respect to set a of
   functional dependencies $F$, then $X$ cannot have a 3NF violation with respect to
   the set of dependencies $\Pi_X(F)$.

(b) 10 points
   Show that if $X \rightarrow Y$ holds, then $X \rightarrow\!\!\!\!\!\!\!\!\!\! Y$ holds.
Database, Fall, 1995: Qualifier

Instructions:
1. All questions have an equal value.
2. You may skip two questions.
3. Use your own paper for your answers.
Master's Exam
CS 551 Database Theory
April 2, 1994

- Answer 4 questions: At least 1 and at most 3 from each of the two parts.
- The test is closed book and closed notes.
- The total time duration is 80 minutes.
- Each question carries 25 points.
- To disprove something, it is enough if you give a counter-example.
- To prove something, an example won't suffice.
- Do not forget to write your Name and Social-Security Number.
- Do not start working on the test until you are told to do so.
PART-I
Answer at least 1 and at most 3
Data Models: Well over 100 different logical data models have been proposed for storing data in databases. The following parts of this question ask you to compare some of the most popular. The question does not ask you to describe what the models are. It asks you to identify the advantages of one over the other.

(a) What is the most important advantage of a relational data model over a hierarchical data model?
Provide a specific reason. Include an example illustrating this advantage.

(b) What is the most important advantage of a hierarchical data model over a relational data model?
Provide a specific reason. Include an example illustrating this advantage.

(c) What is the most important advantage of an object data model over a relational data model?
Provide a specific reason. Include an example illustrating this advantage.
Normalization is used in relational databases to remove various dependencies. There are several levels of normalization. Each level of normalization removes a particular type of anomaly. The following question asks you to supply application problems. This means that you are to provide an example with realistic data; i.e., names and relationships that might have meaning.

(a) Provide a "real life" example of an application problem that would occur in a first normal form that would not occur in the second normal form. You must specifically identify the problem. Your example must include both the first and second normal forms.

(b) Provide a "real life" example of an application problem that would occur in a second normal form that would not occur in the third normal form. You must specifically identify the problem. (By identifying the problem, you specify what can go wrong in 2NF that will not go wrong in 3NF.) Your example must include both the second and third normal forms.

(c) Provide a "real life" example of an application problem that would occur in a third normal form that would not occur in the fourth normal form. You must specifically identify the problem. (By identifying the problem, you specify what can go wrong in 3NF that will not go wrong in 4NF.) Your example must include both the third and fourth normal forms.
(3) **Modeling:**

(a) There are said to be three different "views" of a database. What are they? (Identify them by name.) What are the purposes of each of the three views of a database? Your discussion should provide information how the purposes are different from each other.

(b) What is the difference in the relationship between the data items in a **logical record** and the relationship between the data items in a **physical record**.

(c) Various methods have been used to semantically model a database. Two of the most popular are "ER Diagrams" and "bubble charts."
   - Create an example of a logical schema using either method.
   - Your example must include:
     - 1:1 relationships
     - 1:M relationships
     - at least one simple key
     - at least one concatenated key
   - Show both a data graph instantiation and the resulting schema. An example of a data graph instantiation is:

You must supply both a data graph and its logical schema using either an ER diagram or a bubble chart. You can use another page of paper for your answer.
PART–II

Answer at least 1 and at most 3
1(a) Consider a relation scheme $R(ABCD)$ with functional dependencies $F = \{A \rightarrow B, B \rightarrow C, C \rightarrow D, D \rightarrow A\}$. Let $\rho = \{R_1(AB), R_2(BC), R_3(CD)\}$ be a decomposition of $R$. Check if $\rho$ preserves $F$. 
1(b) Consider a relation scheme $R(\text{SDIM})$ with functional dependencies $F = \{ SI \rightarrow D, SD \rightarrow M \}$. Then, show that $R$ is in 2NF but not in 3NF.
2 Suppose we have a database consisting of the following three relations:

- `frequents (drinker, bar)`
- `serves (bar, beer)`
- `likes (drinker, beer)`

Express (a) in SQL and (b) in Tuple Relational Calculus.

(a) Print the drinkers who frequent only bars that serve some beer that they like. (Assume each drinker likes at least one beer and frequents at least one bar.)

(b) Print the drinkers who frequent no bar that serves a beer that they like.
3(a) Check if \((AD, AB, BE, CDE, AE)\) is a lossless-join decomposition of \(ABCDE\) with respect to the set of functional dependencies \(\{A \rightarrow C, B \rightarrow C, C \rightarrow D, DE \rightarrow A, CE \rightarrow A\}\).
3(b) Prove that if the relation scheme \( R(X, Y_1, Y_2, \ldots, Y_k) \) satisfies the multivalued dependencies \( \{X \rightarrow Y_1, X \rightarrow Y_2, \ldots, X \rightarrow Y_k\} \), then \( R \) decomposes losslessly onto the relation schemes \( (XY_1, XY_2, \ldots, XY_k) \).
4(a) Given a relation scheme $R[A_1, A_2, \ldots, A_n]$, what is the maximum number of keys (not necessarily primary keys) $R$ can have?

4(b) Show that for any set of functional dependencies $F$, $((F^+)^+)^+ = F^+$. 
Master’s Exam
CS 551 Database Theory
November 13, 1993

- Answer all the questions.
- The test is closed book and closed notes.
- The total time duration is 60 minutes.
- The maximum number of points is 100.
- Check if you have seven pages including the title page.
- To disprove something, it is enough if you give a counter-example.
- To prove something, an example won’t suffice.
- Do not forget to write your Name and Social-Security Number.
- Do not start working on the test until you are told to do so.

Name: ___________________________ SSN: ________________________
1. Given a relation scheme $R[A_1, A_2, \ldots, A_n]$, what is the maximum number of primary keys $R$ can have such that the each of those primary keys has at most $k \leq n$ attributes?

2. Let $A$ be the attribute set of a relation scheme $R$, and let $X \subseteq A$. Let $P$ and $Q$ be some relation instances of $R$. Then, prove or disprove the following:
   $\Pi_X(P \cap Q) = \Pi_X(P) \cap \Pi_X(Q)$
3. Consider the following suppliers-parts-projects database.

<table>
<thead>
<tr>
<th>Suppliers</th>
<th>S (s#, sname, status, city)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parts</td>
<td>P (p#, pname, color, weight, city)</td>
</tr>
<tr>
<td>Projects</td>
<td>J (j#, jname, city)</td>
</tr>
<tr>
<td>Sup-Part-Proj</td>
<td>SPJ (s#, p#, j#, qty)</td>
</tr>
</tbody>
</table>

Answer the following queries in *tuple relational calculus* and *SQL*:
"Get j# values for projects supplied by supplier S5 with all parts that supplier S1 supplies".
4. Prove that Domain Relational Calculus when restricted to safe expressions subsumes Tuple Relational Calculus when restricted to safe expressions.
5. Check if \((AD, AB, BE, CDE, AE)\) is a lossless-join decomposition of \(ABCDE\) with respect to the set of functional dependencies \(\{A \rightarrow C, B \rightarrow C, C \rightarrow D, DE \rightarrow A, CE \rightarrow A\}\).
6. Let $R$ be any relation scheme, and let $r$ denote a relation of $R$. Let $\phi$ be a relational operator. Let $U, V \subseteq r$. Then, the operator $\phi$ is said to be monotone if whenever $U \subseteq V$, then $\phi(U) \subseteq \phi(V)$. Show that the operations selection $\sigma_F$ (where $F$ is a condition) and projection $\Pi_L$ (where $L$ is a list of attributes), are monotone.
7. Consider a relation scheme $R(SDIM)$ with $F = \{SI \rightarrow D, SD \rightarrow M\}$. Then, show that $R$ is in 2NF but not in 3NF.

8. Show that if $X \rightarrow Y$ holds, then $X \Rightarrow Y$ holds.
Master's Exam
Database Theory - 5 5
April 4, 1992

- The exam is closed book and closed notes.
- The total time duration is 60 minutes.
- The maximum number of points is 100.
- You must show all your work for full credit.
- A proper subset of a key could be a key.
- A candidate key is a key no proper subset of which is a key.
- To prove something, an example won't suffice.
- Check if you have ten pages including the title page.
- You must answer 6 questions:
  answer all of Questions 1–4;
  answer one of questions 5 and 6; and
  answer one of Questions 7, 8 and 9.
1. The following questions are concerned with the definition of keys and candidate keys.

(a) Given a relation scheme $R(A_1, A_2, \ldots, A_n)$, what is the maximum number of keys $R$ can have?

(b) Given a relation scheme $R(A_1, A_2, \ldots, A_n)$, what is the maximum number of candidate keys $R$ can have?

(c) Is the intersection of two keys of a relation scheme $R$ necessarily a key for $R$? Justify your answer.
2. Show that if $F$ is a set of functional dependencies, then $(\ldots ((F^+)^+) \ldots)^+ = F^+$.
3. Let $R(S, D, I, M)$ be a relation scheme, and let $F = \{SI \rightarrow D, SD \rightarrow M\}$ be the set of functional dependencies on $R$.

(a) Find all the keys of $R$.

(b) Find $(SD)^+$.

(c) Find a minimal set of $F$. Is it unique for $F'$?
4. Check if the following decomposition of the relation scheme $R(ABCDE)$ with the set of functional dependencies $F = \{A \rightarrow C, B \rightarrow C, C \rightarrow D, DE \rightarrow A, CE \rightarrow A\}$ is lossless: $\rho = \{R_1(ADE), R_2(ABC), R_3(BCE)\}$. 
Answer any one of Questions 5 and 6.

5. Suppose we have a database consisting of the following three relation schemes:

    frequents (drinker, bar)
    serves (bar, beer)
    likes (drinker, beer)

Express the following query either in relational algebra or in tuple relational calculus: "Print the drinkers who frequent at least one bar that serves a beer they like".
6. Consider the following suppliers-parts-projects database.

Suppliers \[ S (s\#, \text{name}, \text{status}, \text{city}) \]
Parts \[ P (p\#, \text{name}, \text{color}, \text{weight}, \text{city}) \]
Projects \[ J (j\#, \text{name}, \text{city}) \]
Sup-Part-Proj \[ \text{SPJ} (s\#, p\#, j\#, \text{qty}) \]

Express the following query either in SQL or in Quel: "Get j\# for projects supplied entirely by the supplier whose s\# is S1".
Answer any one of Questions 7, 8 and 9.

7. Prove that if the relation scheme $R(X, Y_1, Y_2, \ldots, Y_k)$ satisfies the functional dependencies $\{X \rightarrow Y_1, X \rightarrow Y_2, \ldots, X \rightarrow Y_k\}$, then $R$ decomposes losslessly onto the relation schemes $(XY_1, XY_2, \ldots, XY_k)$. 
8. Show that if $R$ is a relation scheme and $X \subseteq R$ is a candidate key with respect to set $F$ of functional dependencies, then $X$ cannot have a 3NF violation with respect to the set of dependencies $\Pi_X(F)$. 
9. Let $S \subseteq A_R$, where $A_R$ is the attribute set of the relation scheme $R$. Show that if $X \rightarrow Y$ holds in $II_S(R)$, then $X \rightarrow Y$ holds in $R$. 
Masters Exam: Database Theory

April 6, 1991

Name:

There are 7 problems on this test. Do 4 of the problems.

1. Entity-Relationship models

Design an Entity-Relationship model for a database to store and retrieve catalogue information for the campus libraries.

For each book, the database must store the call number, the title, the author, the publication date, and which library owns the book.

The queries are the following:

- A user can ask for all information about a book by entering the call number.
- A user can ask for all information about all books by a given author by entering the author's name. (You do not need to be able to distinguish different authors with the same name.)
- A user can ask for all information about all books with a given title.
- A user can ask for all information on all books with a given subject classification.
- The subject classifications are ordered in a tree structure. For example, the classification Database theory might have subclassifications Entity-Relationship models, Hierarchical models, Relational models, and Deductive databases, and Database theory itself might be a subclassification of Computer Science. The user can input a subject classification and get the computer to print out all subclassifications.

This problem concerns designing an entity-relationship model for such a database. Assume that each entity will have a primary key, and that all accesses to an entity must be either by that key or through some relationship.

(Continued on next page.)
(a) Draw an entity-relationship diagram, showing the entities, the relationships, and the attributes. Underline the primary key for each entity. Indicate which relations are 1-1, which are many-1, and which are many-many. If any attributes are multi-valued, indicate that.

(b) In your relationship between subject classifications and books, if a book has subject classification database theory, will you store only that relationship, or will you also store the relationship between the book and the superclassification computer science? Why?
2. On *virtual records in hierarchical databases*.

(a) What are virtual records?

(b) Why are they needed? (List several different kinds of circumstances where virtual records are used and describe why those circumstances require virtual records.)
3. Relational Algebra

<table>
<thead>
<tr>
<th>w</th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

(a) Compute

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

(b) Given the following relational scheme with keys underlined:

\{R_1 = \{A, B, C\}, \ R_2 = \{A, D, E, F\}, \ R_3 = \{E, G\}, R_4 = \{G, H\}\}

write a query in the relational algebra to list the set of all pairs of values \(g, h\) for \(G, H\) where some object (identified by key attributes \(A, B\)) has attribute values \(g\) for \(G\) and \(h\) for \(H\).
4. Relational Calculus

(a) For the two relation schemes \( R_1 = \{p, q, r, s\} \) and \( R_2 = \{z, y, x\} \), give a query in the domain relational calculus equivalent to

\[ R_1 =_{p=q} \land (r < s) R_2. \]

(b) For the two relation schemes \( R_1 = \{w, z, y, x\} \) and \( R_2 = \{y, z\} \), give a query in the domain relational calculus equivalent to \( R_1 \div R_2 \).

5. Deductive Databases

(a) For the two relation schemes \( R_1 = \{p, q, r, s\} \) and \( R_2 = \{z, y, x\} \), give a stratified intensional database (stratified program) equivalent to

\[ R_1 =_{p=q} \land (r < s) R_2. \]

(b) For the two relation schemes \( R_1 = \{w, z, y, x\} \) and \( R_2 = \{y, z\} \), give a stratified intensional database (stratified program) equivalent to \( R_1 \div R_2 \). (Caution: this one is a bit complicated.)
6. Functional Dependencies

Let $R$ be a relation schema; $F$, $G$, sets of functional dependencies; and $X$, a set of attributes of $R$ (i.e., $X \subseteq R$).

Recall Armstrong's axioms:

**Reflexivity**: If $Y \subseteq X \subseteq R$, then $X \rightarrow Y$ follows from $F$ by Armstrong's axioms.

**Augmentation**: If $X \rightarrow Y$ follows from $F$ by Armstrong's axioms and $Z \subseteq R$, then $XZ \rightarrow YZ$ follows from $F$ by Armstrong’s axioms.

**Transitivity**: If $X \rightarrow Y$ and $Y \rightarrow Z$ follow from $F$ by Armstrong's axioms, then $X \rightarrow Z$ also follows from $F$ using Armstrong's axioms.

**Union**: If $X \rightarrow Y$ and $X \rightarrow Z$ follow from $F$ by Armstrong's axioms, then $X \rightarrow YZ$ follows from $F$ by Armstrong's axioms.

**Pseudotransitivity**: If $X \rightarrow Y$ and $WY \rightarrow Z$ follow from $F$ by Armstrong's axioms, then $WX \rightarrow Z$ follows from $F$ by Armstrong's axioms.

**Decomposition**: If $X \rightarrow Y$ follows from $F$ by Armstrong's axioms and $Z \subseteq Y$, then $X \rightarrow Z$ follows from $X$ by Armstrong's axioms.

It is understood that every dependency in $F$ follows from $F$ by Armstrong's axioms.

(a) Prove the Pseudotransitivity axiom from the Reflexivity, Augmentation, and Transitivity axioms.

(b) Let $R = \{A, B, C, D, E, F\}$.

Let $F = \{AB \rightarrow CDE, AC \rightarrow D, AD \rightarrow B, E \rightarrow F\}$.

Is $AE \rightarrow DF \in F^+$? Prove your answer.
7. Normal forms

Let relation scheme \( R = \{A, B, C, D, E\} \). Let \( \mathcal{F} = \{C \rightarrow F, E \rightarrow A, C \rightarrow D, A \rightarrow B, DE \rightarrow C\} \). You may use the fact that \( \mathcal{F} \) is minimal.

(a) Find a decomposition of \( R \) which is in 3NF, has a lossless (non-additive) join, and preserves dependencies (all with respect to \( \mathcal{F} \)).

(b) Is your decomposition in BCNF? How do you know that?
Master's Exam--May 5, 1990--CS 551, Database Theory

Directions. You must answer Question 1.
You must answer any four (4) of Questions 2-7.

1. You must answer this question.

For the given ER diagram:

a) explain the meaning of each of the symbols.

b) develop a hierarchical database from this diagram.

c) develop a network database from this diagram.

d) develop a relational database from this diagram. Clearly mark the primary key for each relation.
You must answer four (4) questions from this section.

2. a) Name the 5 basic relational operators.

b) Calculate R1 \* S1.

<table>
<thead>
<tr>
<th>R1</th>
<th>S1</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>w</td>
</tr>
<tr>
<td>v</td>
<td>x</td>
</tr>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>y</td>
<td>z</td>
</tr>
<tr>
<td>u</td>
<td>y</td>
</tr>
<tr>
<td>x</td>
<td>z</td>
</tr>
<tr>
<td>y</td>
<td>w</td>
</tr>
<tr>
<td>z</td>
<td>x</td>
</tr>
<tr>
<td>u</td>
<td>v</td>
</tr>
</tbody>
</table>

3. Suppose we have relations R(A1,A2,A3,A4) and S(B1,B2,B3,B4), where A1 = B1 and A2 = B2. Show how to compute in SQL:

a) R \* S.

b) R \* S.

4. a) State Armstrong's Axioms for functional dependencies.

b) Show that Armstrong's Axioms imply the union rule, i.e., ( X \* Y , X \* Z ) \\= X \* YZ.

5. a) Define 3NF, BCNF.

b) Given R(ABCD) with functional dependencies ( A \* C , D \* C , BD \* A ), show that (AB, ACD, BCD) is not a lossless-join decomposition.

c) For the relation R and functional dependencies in b), find a lossless-join, dependency-preserving decomposition into 3NF.
6. Suppose we have an unnormalized relation containing information about courses, teachers, and texts. Each tuple in the relation contains the name of a course, the names of all teachers teaching the course, and the names of all the different books used in the course (you may assume that each section of the course uses all of these books).

a) Find an equivalent set of relation(s) which are all in BCNF.

b) Discuss possible problems which can occur with the relation(s) in part a).

c) Show how to avoid these problems by decomposing the relation(s) of b) into 4NF.

7. For the following character set and relative frequency of each character:

   a) construct a binary tree which can be used for Huffman coding of this character set.

   b) show the coding obtained from your tree for @ + $ % #.

   character      +  -  #  @  ?  !  $  *  &  %
   frequency      5  2  7  2  3  4  9  7  6  1
Directions. You must answer Question 1.
You must answer any four (4) of Questions 2-7.

1. You must answer this question.

For the given ER diagram:

a) explain the meaning of each of the symbols

b) develop a hierarchical database from this diagram.

c) develop a network database from this diagram.

d) develop a relational database from this diagram.
Clearly mark the primary key for each relation.
Computer Science Master’s Exam—May 1989—Database Theory
One Hour

Instructions: Show all work. Define any nonstandard terms. You must answer:
The question in Section I
Any three questions from Section II
Any one question from Section III

Notation: In the following problems, \( R = R_1R_2 \ldots R_n \)
denotes a relation \( R \) with attributes \( R_1, R_2, \ldots, R_n \); and
\( A_1 \ldots A_q \rightarrow B_1 \ldots B_s \) is used to denote the fact that
attributes \( A_1 \ldots A_q \) functionally determine attributes
\( B_1 \ldots B_s \).

Section I is required for everyone.

A library system needs a database to keep track of its books
and users. Each book is identified by acquisition number,
call number, title, author, and date. Only the acquisition
number is unique. Each user is identified by social
security number, name, and address. Each book is kept in
one of four separate library buildings. Some of the
information the library will need to know is:

--for a given user, what books are checked out and when
is each one due?
--for a given book, what building does it belong in and
where is it at present (on shelf, checked out and
due back on a certain day, lost)?
--for a given book, identified by title, how many
copies are there? Where are they at present?
--what are all the books written by a given author and
where is each one?
--what are all the books on a given subject and where
is each one?

a. Design a database for this system in the entity-
relationship model. Define clearly all notation
you use and state explicitly any assumptions you
make.

b. Based on the entity-relationship diagram in (a),
   i. Design a network database for this system.
   ii. Design a hierarchical database for this system.
   iii. Design a relational database for this system.
       Identify the primary key of each relation.
       Do not leave any relations in First or Second
       Normal Form.
Section II. Answer any three of the five questions in this section.

1. Name the five primitive relational algebra operators. If R and S are two relations, show how to compute R intersect S using only primitive operators.

2. Suppose we have a relation $R = R_1R_2R_3R_4$ in which there are no functional dependencies among any of the attributes $R_1$, $R_2$, $R_3$, $R_4$. Determine whether R is in 4NF, BCNF, 3NF, 2NF, 1NF. Justify your answer.

3. Suppose we have relations $R = ABCD$ and $S = ABEF$. Show how the natural join of R and S is computed in SQL.

4. Suppose we have a relation $R = ABCDEF$ with dependencies $A \rightarrow B$, $A \rightarrow D$, $AB \rightarrow C$, $DE \rightarrow F$. Find a lossless-join, dependency-preserving 3NF decomposition of R. Justify your answer.

5. Define 4NF. Show that the relation $R = ABC$ with dependencies $A \rightarrow \rightarrow B$, $A \rightarrow \rightarrow C$ is not in 4NF, and show how it can be decomposed into a set of 4NF relations.

Section III. Answer one question from this section.

1. Suppose we have a B-tree of order M containing N levels, where the root is at level 1 and the leaves, which may be considered to be at level $N + 1$, contain no keys.
   a. Draw a picture of such a tree for $M = 3$, $N = 2$.
   b. Give the maximum and minimum numbers of keys which can be stored in this B-tree as functions of M and N (your answer must be valid for all M and N $\geq 1$).

2. Suppose we have a database containing a relation $R = (name, salary, occupation)$. Suppose R contains N tuples and queries to the database will not be answered if they involve fewer than $N - 2$ tuples. Suppose John Doe is one of the people about whom information is stored in R. Is there a sequence of queries to the database which would allow you to find out the salary of John Doe? Why or why not?