# CSE 494/598 Spring 2001 Exam 2

Instructor: Subbarao Kambhampati  
Given Monday, 7th May 2001 6pm  
Due Thursday, 10th May, 2001 9AM—SHARP no exceptions.  
*Must be submitted in hardcopy. You are encouraged to restrict your answer to the space provided. In cases where you absolutely must have more space, you can use the back of the sheets—but make a note so I won’t miss the work.*

Name:________________________________________

The examination is going to be conducted under the following honor-code agreement. Please read and sign it:

> “I understand that this final take-home examination is supposed to be strictly individual effort. I certify, under the penalty of academic dishonesty, that I have not consulted with anyone other than the instructor in preparing my answers.”

Signature:________________________  date:________________________

<table>
<thead>
<tr>
<th>Qn I (39pt)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Qn II (17pt)</td>
<td></td>
</tr>
<tr>
<td>Qn III (12pt)</td>
<td></td>
</tr>
<tr>
<td>Qn IV (20pt)</td>
<td></td>
</tr>
<tr>
<td>Qn V (12pt)</td>
<td></td>
</tr>
</tbody>
</table>

**Total (100pt)**
True/False questions. For each question, indicate whether the statement is true or false, with a brief justification for your answer. Answers without justifications won’t get any points. Each question carries 3 points. [13x3=39]

1. In information integration scenarios, a maximally contained plan for the query is always the right plan to be executed to satisfy the user.

2. A join algorithm of the following form is very good for information integration applications as it only requires enough memory to store just the smaller of the two relations:

   For each tuple \( t' \) in \( R_{large} \) do
   For each tuple \( t'' \) in \( R_{small} \) do
     If \( t' \) and \( t'' \) join, output the joined tuple

3. The K-means clustering algorithm is more suited to cluster query results returned by search engines than it is in clustering student marks into letter grades.

4. In an information integration scenario, if the source databases are XML databases, instead of being relational databases, then integrating the databases would be trivial.
5. In the Niagara system, suppose we replace the SEQL text-in-context search engine, and instead call Google with the union of words occurring in the XML-QL query, and feed the resulting URLs to the XML-QL query engine (assume that we are still searching in the space of pure XML documents). This approach will make the results output by Niagara both unsound and incomplete.

6. In relational databases, the result of joining two relations can never be smaller than the result of a select query on the first relation.

7. The longer the LSH signature is in the LSH scheme, the larger the number of signatures we need to compute per document in the min-hash/LSH scheme. This means that we are better off going with an LSH signature of length 1.

8. Query planning in information integration is identical to query planning in traditional databases, in the presence of materialized views (caches).

9. It always makes sense to compute a query that has a selection on the join of two relations $R_1$ and $R_2$ by first doing the selection on the individual relations and then joining the resulting smaller relations.
10. In the LAV approach, if a source S is modeled as exporting a relation R in the mediated schema, this means that every tuple of relation R is present in S.

11. The main problem with the local as view approach to information integration is that it forces the user to pose queries in terms of the source schemas.

12. Even if the original query on the mediated database is expressed in SQL, restrictions on the access capabilities of sources may sometimes make the maximally contained rewriting be inexpressible in SQL.

13. The HITS algorithm is quite stable in the sense that addition of a couple of new links is not likely to significantly change the A/H values of the pages.
Qn II.[17pt] Short answer questions.

1.[4pt] Consider the following two ways of joining the four relations A, B, C, and D. Explain, briefly, which approach is likely to be better, and why, in each of the following cases:

1. The four relations are very large, and the computer doing the join doesn’t have a whole lot of memory
2. The relations are stored in different sources spread all over the Internet, and we have an impatient user waiting at the top.

2.[2+2] We talked about the importance of “non-blocking” operators for the operations in a query plan so that first few results can be given to the user even if some of the sources have gotten “stuck”. Comment on how (and if) non-blocking operators can be designed for the following operations:

--Join of two relations
--Finding the maximum value of an attribute
3. [2+2pt] For each of the following XML fragments, write down
(1) The graphical model of the data and (2) an RDBMS table fragment that is *equivalent*
to it. If that is not possible, please explain why:

```
<student>
  <exam1> 34 </exam1>
  <project1> 77</project1>
</student>
```

```
<student>
Boy, this guy is a really smart student.
  <exam1> 98 </exam1>
But he seems to have trouble with projects
  <project1> 33 </project1>
</student>
```

4.[5pt] Suppose you were to list the topics we covered in this course in the order of their interestingness to you—with the most interesting topics at the top of the list, while least interesting topics in terms of the time spent on them at the bottom. Write down the top 5 and bottom five topics from the list. Note that I fully realize that you may have liked this course so much that even the least liked topic is something you liked quite a bit; so please don’t tell me that there are no “not liked” topics…

<table>
<thead>
<tr>
<th>Most liked topics</th>
<th>Least liked topics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
III. [12pt] Consider the following fragment of XML data.

```xml
<cse494-marks-db>
  <student>
    <exam1>34</exam1>
    <project1>77</project1>
  </student>
  <student>
    <exam1>34</exam1>
    <project1>77</project1>
    <athome-exam1>33</athome-exam1>
  </student>
</cse494-marks-db>
```

1. [4pt] Write down a DTD for this file.

2. [2pt] Many XML files come without DTDs, and it may be important to have algorithms that can automatically infer DTDs given the XML file. Having just done what a DTD inferring algorithm would do, can you think of a couple of difficulties that would be faced by such an algorithm?

3. [4pt] Given the data of type above, suppose I am interested in finding the list of all students who got more than 98 points in some exam or project. Write down an XML-QL query for this. Make it as compact as possible.

4. [2pt] Show the SQL query that is equivalent to the query in 3.
IV. [20pt] We are trying to integrate a set of book sellers. The schema we want to export to the user involves just a single relation:

$$\textbf{Book}(\text{author}, \text{title}, \text{subject}, \text{price}, \text{numpages})$$

We have the following two online bookstore sources:

\begin{itemize}
  \item \textbf{A: AliensRaComing:} Only sells sci-fi books. For each book it sells, it can give the information about author, title and price.
  \item \textbf{E: EndIsNear:} Primarily sells religious books supposed to knock some fear of God into people. However, it also stocks a few books on all other subjects (just in case the end is not quite as near as Pat Robertson says, and the owner needs to make a living in the meantime). For each book it sells, it has information about author, title, subject and price.
\end{itemize}

1.[3pt] Suppose we are integrating these sources using Global as View approach. Write down the view corresponding to the \textbf{Book} relation.

2.[2pt] I have the following SQL query. Show how it gets rewritten in terms of the sources in the GAV approach:

```
Select author
From Book
Where subject="sci-fi"
```

3.[3pt] Now suppose we decide to go with Local as View approach. Show how the two sources will be modeled.
4. [5pt] Consider the following candidate plan for the query in Qn 2:

*Compute the results by calling the source A and selecting the authors from the returned tuples.*

Show, using the idea of containment mappings, that this plan is in fact a “sound” plan for the query. Is it also a complete plan? If not, what additional information about sources could make it a complete plan?

5. [5pt] Show how the inverse rule algorithm rewrites the query in qn 2 into a maximally contained plan.

6. [2pt] Are either of the two approaches, LAV/GAV, able to answer every possible query that can be posed on the mediated schema?
Qn V. [12pt] A search engine returned 5 documents in response to a query. The
documents are completely described in terms of the weights and type of keywords they
have in the following table:

<table>
<thead>
<tr>
<th></th>
<th>Computer</th>
<th>Repair</th>
<th>Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>6</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>D2</td>
<td>3</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>D3</td>
<td>1</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>D4</td>
<td>1</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>D5</td>
<td>6</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Suppose you are trying to cluster these results using K-means clustering algorithm into 2
clusters. Assume you start the clusters off with D3 and D4 as the initial cluster centers.
Assume that you are using a bag-based similarity measure discussed in the LSH paper:

\[
sim(d_1, d_2) = \frac{|B_1 \cap B_2|}{|B_1 \cup B_2|}.
\]

Show the operation of the K-means algorithm on this data. For each iteration of the K-
means, show the cluster dissimilarity measure (which is defined as the sum of the
similarities of docs from their respective cluster centers). You are allowed to use
calculators. Show all details of your work if you expect partial credit.