Quiz 4: Data Models
6170: Software Studio
Fall 2014

Answers (only for those who are unable to enter their solutions online)

Name:
MIT email:

For questions 1 to 4, enter a letter from A through H

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></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>
</tbody>
</table>

For questions 5 to 9, enter a letter from A through C

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<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>
</tbody>
</table>
The data model shown below attempts to describe the state of an application for editing slide decks (such as Powerpoint or Keynote). Only one kind of elementary element -- a text box -- is allowed in this rather simple app. Elements can be grouped, and a slide can contain groups and/or basic elements.

Each of the following questions is about the markings at the ends of relations; the answer is one of the letters from A to H labeling the ends in the diagram. Throughout, "stronger" means more constraining.

1. Which of the multiplicity markings is too strong, and rules out some functionality that is obviously intended?  
   *D*, because it permits each group to contain only one element.

2. Which relation end needs a stronger multiplicity marking in order to eliminate an unnecessary complication?  
   *A*, to prevent a component from belonging to more than one slide.
3. Suppose you were required to add an immutability marking to just one end. Which one would damage the functionality least?

A: This would prevent you from moving components between slides. But that’s unlikely to be a major restriction since there may not even be a way to drag a component across slide boundaries in the user interface. And copy/paste is ubiquitous, so you could always make fresh copies.

D is not such a good answer, but we gave credit for it too. It would cause slightly more damage, since it would prevent you from adding or removing elements in a group. In fact, this restriction used to be common: in early version of Keynote you could not even select an element of a group, so there was no way to add or delete; this was fixed so you can now adjust members of a group (and delete them but not add them). In OpenOffice, you can add and delete within a group.

4. Which end should be moved to a more general set in order to provide richer functionality?

D: a group should be able to contain a group.

E was a common answer; this would suggest that both groups and elements have x and y coordinates. It’s not clear, however, why a group should have coordinates in addition to the elements it contains.
Consider building a web application that handles applications for job postings. The three diagrams below show the same data model, transformed (by reversal of some relations and addition of contours) to three different data designs.

The following code sketches suggest Mongo implementations of these three designs:

D.  
collection 1: [{description: ..., applications: [{date: ..., by: 123}]}

collection 2: [ {_id: 123, resume: ...}, ... ]

E.  
collection 1: [{date: ..., by: 123, for: 25}, ...]

collection 2: [ {_id: 123, resume: ...}, ... ]

collection 3: [ {_id: 25, description: ...}, ... ]

F.  
collection 1: [ {_id: 123, resume: ..., applications: [{date: ..., for: 25}]}], ...]

collection 2: [ {_id: 25, description: ...}, ... ]

Match the diagram (A through C) to the implementation (D through F):

5. Which diagram matches E? A
6. Which diagram matches D? B
Identify a design by its diagram (A through C) that:

7. May lead to contention if there are many users applying for the same job.
   
   B, because the job document would need to be updated whenever a new application is added.

8. Would make it hardest to gather data for an employer to view all applications for a specific job.
   
   C, because even if the right document is found in the application collection, it would require searching within the array of applications to find the one matching a particular job.

9. Is likely to require the largest number of joins in queries.
   
   A, because there’s no nesting at all, so any query involving more than one of {applications, users and jobs} will require a join.