Exercises for Principles of Data Integration

AnHai Doan Alon Halevy Zachary Ives

1 Introduction

1. What distinguishes data integration from standard keyword search, as with a Web search engine?

2. How might data integration as described in this chapter help reduce concerns about release of a particular source’s private information, e.g., a person’s criminal record?

3. Many people advocate the creation of community standards to “solve” the data integration problem. Which of the “hard” aspects of the problem does this address?

4. How does a single data model and format, such as XML, help with respect to the challenges of data integration? List three examples of problems that it would not solve.

5. Data integration across multiple organizations is harder than integration within an organization. What are the additional difficulties that are introduced when we integrate across organizations?

6. Rewrite the following SQL query as a conjunctive query using the syntax of Section 2.1.4, given the relations Interview(candidate, recruiter, hireDecision) and EmployeePerformance(employee, grade):

   SELECT candidate, grade, hireDecision
   FROM Interview, EmployeePerformance
   WHERE recruiter = "Annette Young" AND Interview.
   candidate = EmployeePerformance.employee

7. Describe the difference in roles between the wrapper and source description in converting data into a form suitable for a mediated schema.
8. Beyond relational and HTML data, give three examples of data source formats.

9. Explain the difference between a data warehouse and a virtual data integration system.

10. Of the various components in a virtual data integration system, which are responsible for doing the computation that translates data from the data source’s schema and format, to answers that are in the output schema of the user’s query?

2 Manipulating Query Expressions

1. For each of the following pairs of queries $Q_1$ and $Q_2$, does $Q_1 \subseteq Q_2$ or $Q_2 \subseteq Q_1$. When containment holds, explain why and when containment does not hold, find a counter-example database.

   (a) $Q_1(A, B, D) :- P(A, B, C), Q(C, D), P(A, B, D), Q(D, C)$
   $Q_2(X, Y, Z) :- P(X, Y, Z), Q(Z, Z)$

   (b) $Q_1(E, F) :- P(A, B, C, E), S(E, F), R(C, E, F)$
   $Q_2(X, Y) :- P(Z, T, U, X), R(U, Y, X)$

   (c) $Q_1(A, B) :- P(A, B), R(C, D)$
   $Q_2(X, Y) :- P(X, Y), R(X, X), R(X_1, Y_1), X_1 < Y_1$

   (d) $Q_1(A, C) :- Parent(A, B), Brother(B, C)$
   $Q_2(X, Z) :- Father(X, Y), Brother(Y, Z)$

   (e) $Q_1(A, C) :- P(A, B), P(B, C)$
   $Q_2(X, Y) :- P(X, X_1), P(X_1, X_2), P(X_2, X_3), P(X_3, Y)$

   (f) $Q_1(A, C) :- P(A, B), P(B, C)$
   $Q_2(X, Y) :- P(X, X_1), P(X_1, X_2), P(X_2, Y)$

   (g) $Q_1() :- r(X_1, X_2), r(X_2, X_3), r(X_3, X_4), r(X_4, X_5), r(X_5, X_1), X_1 < X_2$
   $Q_2() :- r(X_1, X_2), r(X_2, X_3), r(X_3, X_4), r(X_4, X_5), r(X_5, X_1), X_1 < X_3$

   (h) $Q_1() :- p(A, 4), A < 4$
   $Q_2() :- p(X, 4), p(Y, X), X \leq 4, Y < 4$

2. Consider the following restricted versions of the problem of answering queries using views. Assume that all relations have $n$ attributes, and all views and queries are conjunctions, i.e., of the form

$$q(X_1, \ldots, X_n) : \neg R_1(X_1, \ldots, X_n), \ldots, R_k(X_1, \ldots, X_n)$$
Only the choice of relations in the bodies and the number of conjunct can differ, but nothing else.

(a) Assume that all the views are defined with the open-world assumption. Describe an algorithm for answering queries using views. The algorithm needs to be tailored to this special case (and hence simpler than the ones presented in the chapter). What is the computational complexity of the algorithm?

(b) How would the algorithm above change if the views are defined under the closed-world assumption and you’re required to find an equivalent-rewriting of the query using the views.

(c) How would the algorithm of Part 2 change if you consider multi-set semantics (that is, we count the number of times each tuple appears in the answer, and the rewriting needs to produce the same number). Assume that the views are defined under the closed-world assumption.

3. Prove that query unfolding may result in a number of subgoals exponential in the number of applications of the unfolding step.

4. Find an equivalent rewriting of the query

\[ Q_1() \leftarrow r(X_1, X_2), r(X_2, X_3), r(X_3, X_4), r(X_4, X_5), r(X_5, X_1), X_1 < X_2 \]

using the views

\[ v_1(X_1, X_3) \leftarrow r(X_1, X_2), r(X_2, X_3) \]
\[ v_2(X_1, X_3) \leftarrow r(X_3, X_4), r(X_4, X_5), r(X_5, X_1) \]

5. Find the maximally-contained rewriting of \( Q \) using the views \( v_1-v_3 \):

\[ Q() \leftarrow e(X, Z), e(Z, Y), X > 6, Y < 8 \]
\[ v_1(X, Y) \leftarrow e(X, Z), e(Z, Y), Z > 6 \]
\[ v_2(X, Y) \leftarrow e(X, Z), e(Z, Y), Z < 8 \]
\[ v_3(X, Y) \leftarrow e(X, Z_1), e(Z_1, Z_2), e(Z_2, Z_3), e(Z_3, Y) \]

3 Describing Data Sources

1. Consider the relation schema for Source S:

\[
\text{AllUsers}(
\text{userID : int, name : string, email : string, age : int, city : string,}
\text{partner : string, partnerTypeCode : int})
\]

where partnerTypeCode is one of:
• 1 if “Single”
• 2 if “In a Relationship”
• 3 if “Engaged”
• 4 if “Married”
• 5 if “It’s Complicated”
• 6 if “In an Open Relationship”
• 7 if “Widowed”

The corresponding mediated schema relation is normalized slightly differently:

UserAccount(userID : int, name : string, email : string, age : int, city : string)
UserPartner(userID : int, partnerID : int, partnershipDesc : string)

where the partnershipDesc is a string whose value may be one of the above types of partnerships (“Single,” “In a Relationship, ...). Define a correspondence table codes to map the partnership type descriptions to type codes, and treat it as an additional source.

(a) Write a global-as-view mapping between Source S and the relations of the mediated schema.
(b) Write a local-as-view mapping between Source S and the relations of the mediated schema.

2. Given the mappings from Question 1, write a datalog query, posed over the mediated schema, that retrieves the user names and partner status (as a string description) for all users.

(a) Show how to reformulate the query using global-as-view mappings.
(b) Show how to reformulate the query using local-as-view mappings and your choice of algorithms (bucket, inverse rules, or MiniCon).

4 String Matching

5 Schema Matching and Mapping

6 General Schema Manipulation Operators

1. Explain how Object-Relational Mappings, provided for many programming languages such as Java or Python, are instances of the modelGen operator.

2. Define a mapping that computes the merge of the following schemas:
   Taxonomy(organism, name, extinct?, class) with Organism(id, name) and Class(organismId, class).
7 Data Matching

8 Query Processing

1. For two SQL queries of the same complexity, one in a database scenario and one in a database scenario, which (if any) should take longer to optimize, and why (or why not)?

2. Compare a two-way semijoin and an index nested loops join. How are they similar and how are they different?

3. Why is the pipelined hash join not commonly used in a conventional DBMS?

4. What are the differences and relative benefits of horizontal partitioning versus vertical partitioning?

5. What is the difference between a ship operator and a rehash operator?

6. What are the main differences between distributed database techniques and parallel database techniques? Which are more relevant to today’s cloud computing environments?

7. Adding decision points about re-optimization after every pipeline end gives greater flexibility. What does one give up in return, and how do event-condition-action rules somewhat mitigate this?

8. Show a physical query plan for a join between the relations Patient\textsuperscript{bf}(id, name, ssn) (recall the access-pattern notation of Chapter 3) and PersonalPhysician\textsuperscript{ff}(patientid, name). Are there other feasible plans?

9. Given a lottery scheduling eddy with operators $o_1, o_2$, which have equal execution costs and which have selectivities $s_1, s_2$ where $s_2 > s_1$: What is (a) the closest to optimal performance the eddy can attain? (b) the furthest from optimal performance?

10. For the lottery scheduling eddy, give an example scenario where the lottery scheduling heuristic will result in poor performance for a select-join query.

11. In Corrective Query Processing, discuss the merits of computing the stitch-up at the end, versus always “porting” sub-results from the “old” plan to the current one as we switch from one plan to the next.
9 Wrappers

10 Data Warehousing and Caching

1. Give two examples of operations that can be performed using ETL tools, but not declarative schema mappings.

2. Give two examples of operations that can be performed using ETL tools or declarative schema mappings.

3. List two benefits of materialization versus virtual integration.

4. List two drawbacks of materialization versus virtual integration.

5. Let us consider the cost of materialize views in a very simple setting. Assume that there are $n$ virtual views $V_1, V_2, \ldots, V_n$. For each view $V_i$, there will be a space consumption $S_i$ and a speed gain $G_i$ if $V_i$ is materialized. We also have a total space budget $S$. Given $\{V_i\}, \{S_i\}, \{G_i\}$ and $S$, design an efficient algorithm to maximize the total summation of speed gains for materialized views with respect to the space constraint.

6. Using pseudocode, show the map and reduce functions for the following query:

   ```
   SELECT count(candidate), grade
   FROM Interview, EmployeePerformance
   GROUP BY grade
   ```

7. Explain how the relational join operation can be implemented using map and reduce.

11 XML

1. What do the following XPath queries return when evaluated over the data of Figure 1?

   (a) /dblp/mastersthesis/title

   (b) /dblp/*/editor

   (c) //title

   (d) /*/@key
Figure 1: Sample XML data from the DBLP Web site

```
(e) //title/text()
(f) /descendant::year
(g) //year/..
(h) //ee[position()=first()]
```

2. Given the following relation:

```
Student(userID : int, year : int, major : string, GPA : double)
```

where major is nullable:

(a) Define a corresponding DTD where the root element is named Student and each tuple is a subelement. You may use XML attributes or elements to capture the relation attributes.

(b) Define a corresponding XML Schema where the root element is named Student and each tuple is a subelement. You may use XML attributes or elements to capture the relation attributes.
3. For each paper (inproceedings element) from 1996 onwards, return a year tag, with the following contents: the year, each paper title (in the appropriate element) and set of authors.

4. Using XQuery, find the year with the most papers, according to this file from the previous question. Return the result in a year tag, with all of the papers (title + authors) from that year.

5. Using XQuery, find the key (mentioned in the cite entry) for the paper (inproceedings) that is most cited. Return this within the most − cited tag.

6. Using an XQuery, return, for each paper in proceedings published in 1996, the title and the last author. The result should be nested within a paper tag.

7. Write an XML query that maps information for the year 1996 from inproc.xml and proc.xml into a schema that follows the pattern:

   <proceedings>*
   <title> ... Proceedings title ... </title>
   <paper>*
   <title>... Paper title ...</title>
   <author> *
   <name> .. </name>
   </author>
   </paper>
   </proceedings>

   (The output does not need to be an XML tree here; it can be a forest.)

8. Write an XQuery that sorts papers titles from inproc in ascending alphabetical order of title. Results should be in the following format:

   <papers>
   <title> ... </title> *
   </papers>

9. Design an XML schema for a social network system, corresponding to these relations.
UserAccounts(userID : int, name : string, email : string, age : int, city : string)
UserEducation(userID : int, school : string, startYear : int, finishYear : int)
UserInterests(userID : int, interest : string)
UserPartner(userID : int, partnerUserID : int, marriedToPartner : boolean)
ConnectedTo(userID : int, friendUserID : int, privilegeLevel : int)

Use this as a basic foundation: design an XML representation (either hierarchical or interlinked) for the appropriate entity sets and the connections. You may rename the specific table or attribute names as seems appropriate.

10. Take the schema from the previous exercise. Specify keys and keyrefs for each complexType.

12 Ontologies and Knowledge Representation

1. RDF is often characterized as having more “semantics” than XML. List two aspects of RDF that are “semantic” in nature and not present in XML.

2. What are the relative roles of SPARQL versus OWL?

3. Show a set of at least 10 RDF triples describing data integration research topics and papers.

13 Incorporating Uncertainty into Data Integration

1. Using the block-independent-disjoint model, encode a set of bird sightings. Alice has seen either a blue jay (prob. 75%) or a scrub jay (25%). Bob has seen a cardinal (prob. 85%). Candace has seen a zebra finch (prob. 33%) or a silver bill finch (prob. 67%).

2. If we have a mapping to a data source that has not yet been tested, are we likely to use by table or by tuple semantics for the mapping?

3. Probabilistic schema mappings assume independence among attribute alignments. In general this is not true (as we can see in Chapter 5). Explain why it is still reasonable to work under this assumption.
Figure 2: Example hypergraph of relationships between tuples in a database (and view) instance, where tuples are indicated by rectangular nodes and derivation hyperedges are represented by ellipses with incoming and outgoing arcs.

14 Data Provenance

<table>
<thead>
<tr>
<th>id</th>
<th>from</th>
<th>to</th>
</tr>
</thead>
<tbody>
<tr>
<td>UA724</td>
<td>SFO</td>
<td>IAD</td>
</tr>
<tr>
<td>AS385</td>
<td>SFO</td>
<td>PDX</td>
</tr>
<tr>
<td>UA616</td>
<td>SFO</td>
<td>ORD</td>
</tr>
<tr>
<td>UA5901</td>
<td>ORD</td>
<td>MSN</td>
</tr>
<tr>
<td>US1466</td>
<td>SFO</td>
<td>PHL</td>
</tr>
</tbody>
</table>

Flight:

<table>
<thead>
<tr>
<th>id</th>
<th>from</th>
<th>to</th>
</tr>
</thead>
<tbody>
<tr>
<td>UA724</td>
<td>SFO</td>
<td>IAD</td>
</tr>
<tr>
<td>AS385</td>
<td>SFO</td>
<td>PDX</td>
</tr>
<tr>
<td>UA616</td>
<td>SFO</td>
<td>ORD</td>
</tr>
<tr>
<td>UA5901</td>
<td>ORD</td>
<td>MSN</td>
</tr>
<tr>
<td>US1466</td>
<td>SFO</td>
<td>PHL</td>
</tr>
</tbody>
</table>

Hub:

<table>
<thead>
<tr>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFO</td>
</tr>
<tr>
<td>ORD</td>
</tr>
<tr>
<td>PHL</td>
</tr>
</tbody>
</table>

Figure 3: Example flights database

1. Take the example provenance hypergraph of Figure 2 and show how it would be stored in relations, using the techniques of Section 14.4.

2. Given the view definition:

\[ V_1(X,Y) :- \text{Flight}(X,Z), \text{Hub}(Z), \text{Flight}(Z,Y) \]

and the tables of Figure 3 show the provenance graph for the results.
15 Data Integration on the Web

1. Give three examples of vertical search portals from the Web.

2. For the form of Figure 4 show the corresponding relational schema that an extractor would produce.

3. What are the main bottlenecks in traditional data integration that pay-as-you-go data integration seeks to alleviate? What are the trade-offs?

16 Keyword Search: Integration on Demand

1. Show an example of a data graph where the two different cost models of Section 16.1 will differ.

2. For a keyword search query with two keywords, a Steiner tree algorithm is not necessary to find the lowest-cost tree. What algorithm(s) can be used instead?

3. Take the dataset of Table 1 and show the steps involved in computing the Threshold Algorithm for the top-4 results for the scoring function $\tau = rating \cdot 2 + (4 - price)$. 

Figure 4: HTML for defining a form.

<form action="http://jobs.com/find" method="get">
  <input type="hidden" name="src" value="hp"/>
  Keywords: <input type="text" name="kw"/>
  State: <select name="st">
    <option value="Any"/>
    <option value="AK"/>
    <option value="AL"/>
    ...
  </select>
  Sort By: <select name="sort"/>
    <option value="salary"/>
    <option value="startdate"/>
    ...
  </select>
  <input type="submit" name="s" value="go"/>
</form>
Table 1: Restaurants, rated and priced

<table>
<thead>
<tr>
<th>name</th>
<th>location</th>
<th>rating</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alma de Cuba</td>
<td>1523 Walnut St.</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Moshulu</td>
<td>401 S. Columbus Blvd.</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Sotto Varalli</td>
<td>231 S. Broad St.</td>
<td>3.5</td>
<td>3</td>
</tr>
<tr>
<td>McGillin’s</td>
<td>1310 Drury St.</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Di Nardo’s Seafood</td>
<td>312 Race St.</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

17 Peer-to-Peer Integration

Figure 5: PDMS for coordinating emergency response in Oregon and Washington states. Arrows indicate that there is (at least a partial) mapping between the relations of the peers.

1. Given the PDMS setting of Figure 5, the peer mappings:

   LH.CritBed(bed, hosp, room, PID, status) ⊆ H.CritBed(bed, hosp, room), H.Patient(PID, bed, status)

   LH.EmergBed(bed, hosp, room, PID, status) ⊆ H.EmergBed(bed, hosp, room), H.Patient(PID, bed, status)
LH.GenBed(bed, hosp, room, PID, status) ⊆ H.GenBed(bed, hosp, room), H.Patient(PID, bed, status)

and the source descriptions:

\[
\begin{align*}
LH.CritBed(\text{bed, hosp, room, PID, status}) & :- \quad \text{Bed(\text{bed, hosp, room, PID, status, 'crit'}))} \\
LH.EmergBed(\text{bed, hosp, room, PID, status}) & :- \quad \text{Bed(\text{bed, hosp, room, PID, status, 'emerg'})} \\
LH.GenBed(\text{bed, hosp, room, PID, status}) & :- \quad \text{Bed(\text{bed, hosp, room, PID, status, 'gen'})}
\end{align*}
\]

show how the following query is reformulated, using the rule-goal tree expansion and only the mapping shown above.

\[
Q(\text{PID, room}) :- \quad \text{H.GenBed(bed, hosp, room), H.Patient(PID, bed, status)}
\]

18 Integration in Support of Collaboration

1. Given the query:

```
SELECT title, startTime 
FROM Movie, Plays 
WHERE Movie.title = Plays.movie AND location = "New York" 
AND director = "Woody Allen"
```

over relations Movie and Plays, show how the different clauses PROPAGATE DEFAULT, PROPAGATE DEFAULT-ALL will affect an annotation on the title attribute.

2. Given the trails:

\[
\begin{align*}
*.tuple.date & \rightarrow //*.tuple.received \\
yesterday & \rightarrow date = yesterday() \\
yesterdaysPhotos & \rightarrow //*.jpeg/yesterday
\end{align*}
\]

Show the expansion of the search for “yesterdaysPhotos” into an extended XPath.

3. Explain how the trust semiring of Chapter 14, along with data provenance, is useful for resolving conflicts in a CDSS.

4. Summarize the key differences between a CDSS and the PDMS (Chapter 17).