Chapter XXXIX
Merging, Repairing, and Querying Inconsistent Databases

Luciano Caroprese
University of Calabria, Italy

Ester Zumpano
University of Calabria, Italy

INTRODUCTION

Data integration aims to provide a uniform integrated access to multiple heterogeneous information sources designed independently and having strictly related contents. However, the integrated view, constructed by integrating the information provided by the different data sources by means of a specified integration strategy could potentially contain inconsistent data; that is, it can violate some of the constraints defined on the data. In the presence of an inconsistent integrated database, in other words, a database that does not satisfy some integrity constraints, two possible solutions have been investigated in the literature (Agarwal, Keller, Wiederhold, & Saraswat, 1995; Bry, 1997; Cali, Calvanese, De Giacomo, & Lenzerini, 2002; Dung, 1996; Grant & Subrahmanian, 1995; S. Greco & Zumpano, 2000; Lin & Mendelzon, 1999): repairing the database or computing consistent answers over the inconsistent database. Intuitively, a repair of the database consists of deleting or inserting a minimal number of tuples so that the resulting database is consistent, whereas the computation of the consistent answer consists of selecting the set of certain tuples (i.e., those belonging to all repaired databases) and the set of uncertain tuples (i.e., those belonging to a proper subset of repaired databases).

Example 1. Consider the database consisting of the relation Employee(Name, Age, Salary) where the attribute Name is a key for the relation, and suppose we have the integrated database DB = {Employee(Mary, 28, 20), Employee(Mary, 31, 30), Employee(Peter, 47, 50)}. DB is inconsistent and there are two possible repaired databases each obtained by deleting one of the two tuples whose value of the attribute Name is Mary. The answer to the query asking for the age of Peter is constituted by the set of certain tuples \(<47>\), whereas the answer to the query asking for the age of Mary produces the set of uncertain values \(<28>, <31>\).
This work proposes a framework for merging, repairing, and querying inconsistent databases. To this aim, the problem of the satisfaction of integrity constraints in the presence of null values is investigated, inspired by the approach presented in Bravo and Bertossi (2006), and a new semantics for constraints satisfaction is proposed. The present work focuses on the inconsistencies of a database instance with respect to particular types of integrity constraints implemented and maintained in a commercial DBMS (database management system) such as primary keys, general functional dependencies, and foreign-key constraints.

The framework for merging, repairing, and querying inconsistent databases with functional dependencies restricted to primary-key constraints and foreign-key constraints has been implemented in a system prototype, called RAINBOW, developed at the University of Calabria.

**DATABASE MERGING**

Once the logical conflicts owing to the schema heterogeneity have been resolved, conflicts may arise during the integration process among data provided by different sources. In particular, the same real-world object may correspond to many tuples that may have the same value for the key attributes but different values for some nonkey attribute.

The database integration problem consists of the merging of $n$ databases $DB_i = \{R_{i,1}, \ldots, R_{i,n_i}\}, \ldots, DB_k = \{R_{k,1}, \ldots, R_{k,n_k}\}$. In the following we assume that relations corresponding to the same concept and furnished by different sources are homogenized with respect to a common ontology so that attributes denoting the same property have the same name (Yan & Oszu, 1999). We say that two homogenized relations $R$ and $S$, associated with the same concept, are overlapping if they have the same value for the key attributes. Given a set of overlapping relations, an important feature of the integration process is related to the way conflicting tuples are combined. Before performing the database integration, the relations to be merged must be first reconciled so that they have the same schema.

**Definition 1.** Given a set of overlapping relations $\{S_1, \ldots, S_n\}$, a reconciled relation $R$ is $\text{attr}(R) = \bigcup_{i=1}^{n} \text{attr}(S_i) \cup \{\text{Src}\}$, and contains all tuples $t \in S_i, 1 \leq i \leq n$. All attributes belonging to $\text{attr}(R) - \text{attr}(S_i)$ are fixed to $\perp$; $R[\text{Src}] = i$, where $i$ is the unique index of the source database.

**Example 2.** Consider the following two overlapping relations $S_1$ and $S_2$.

<table>
<thead>
<tr>
<th>K</th>
<th>Title</th>
<th>Author</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Moon</td>
<td>Greg</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Money</td>
<td>Jones</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Sky</td>
<td>Jones</td>
<td></td>
</tr>
</tbody>
</table>

$S_1$

<table>
<thead>
<tr>
<th>K</th>
<th>Title</th>
<th>Author</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Flower</td>
<td>Smith</td>
<td>1965</td>
</tr>
<tr>
<td>4</td>
<td>Sea</td>
<td>Taylor</td>
<td>1971</td>
</tr>
<tr>
<td>7</td>
<td>Sun</td>
<td>Steven</td>
<td>1980</td>
</tr>
</tbody>
</table>

$S_2$
Related Content

Integrated Functional and Executional Modeling of Software Using Web-Based Databases
www.igi-global.com/article/integrated-functional-executional-modeling-software/51206?camid=4v1a

A Paradigm For Natural Language Explanation Of Database Queries: A Semantic Data Model Approach
www.igi-global.com/article/paradigm-natural-language-explanation-database/51129?camid=4v1a

Information Privacy: Understanding How Firms Behave Online
Gerald C. Kane, Kathy Stewart Schwaig and Veda C. Storey (2011). Theoretical and Practical Advances in Information Systems Development: Emerging Trends and Approaches (pp. 81-100).
www.igi-global.com/chapter/information-privacy-understanding-firms-behave/52953?camid=4v1a

INDUSTRY AND PRACTICE: Object-Oriented Database: How Wide is the Application Range?
www.igi-global.com/article/industry-practice-object-oriented-database/51153?camid=4v1a