Chapter 7.3
Consistent Queries over Databases with Integrity Constraints

Luciano Caroprese
DEIS Universitá della Calabria, Italy

Cristian Molinaro
DEIS Universitá della Calabria, Italy

Irina Trubitsyna
DEIS Universitá della Calabria, Italy

Ester Zumpano
DEIS Universitá della Calabria, Italy

INTRODUCTION

Integrating data from different sources consists of two main steps, the first in which the various relations are merged together, and the second in which some tuples are *removed* (or *inserted*) from the resulting database in order to satisfy integrity constraints. There are several ways to integrate databases or possibly distributed information sources, but whatever integration architecture we choose, the heterogeneity of the sources to be integrated causes subtle problems. In particular, the database obtained from the integration process may be inconsistent with respect to integrity constraints, that is, one or more integrity constraints are not satisfied. Integrity constraints represent an important source of information about the real world. They are usually used to define constraints on data (functional dependencies, inclusion dependencies, etc.) and have, nowadays, a wide applicability in several contexts such as semantic query optimization, cooperative query answering, database integration, and view update.

Since the satisfaction of integrity constraints cannot generally be guaranteed, if the database is obtained from the integration of different information sources, in the evaluation of queries, we must compute answers that are consistent with the integrity constraints. The following example shows a case of inconsistency.

Example 1: Consider the following database schema consisting of the single binary relation *Teaches (Course, Professor)* where the attribute *Course* is a key for the relation. Assume there are two different instances for the relations *Teaches*, D1={(c1,p1),(c2,p2)} and D2={(c1,p1),(c2,p3)}. The two instances satisfy the constraint that *Course* is a key, but from their union we derive a relation that does not satisfy the constraint since there are two distinct tuples with the same value for the attribute *Course*.

In the integration of two conflicting databases simple solutions could be based on the definition of preference criteria such as a partial order on the source information or a majority criterion (Lin & Mendelzon, 1996). However, these solutions are not generally satisfactory, and more useful solutions are those based on (1) the computation of "repairs" for the database, and (2) the computation of consistent answers (Arenas, Bertossi, & Chomicki, 1999).

The computation of repairs is based on the definition of minimal sets of insertion and deletion operations so that the resulting database satisfies all constraints. The computation of consistent answers is based on the identification of tuples satisfying integrity constraints and on the selection of tuples matching the goal. For instance, for the integrated database of *Example 1*, we have two alternative repairs consisting in the deletion of one of the tuples *(c2,p2)* and *(c2,p3)*. The consistent answer to a query over the relation *Teaches* contains the unique tuple *(c1,p1)* so that we do not know which professor teaches course *c2*. Therefore, it is very important, in the presence of inconsistent data, not only to compute the set of consistent answers, but also to know which facts are unknown and if there are possible repairs for the database.

BACKGROUND

Several proposals considering the integration of databases as well as the computation of queries over inconsistent databases have been provided in the literature (Agarwal, Keller, Wiederhold, & Saraswat, 1995; Arenas et al., 1999; Arenas, Bertossi, & Chomicki, 2000; Bry, 1997; Dung, 1996; Greco & Zumpano, 2000; Lin, 1996; Lin & Mendelzon, 1996; Lembo, Lenzerini, & Rosati, 2002; Lenzerini, 2002; Wijsen, 2003). Most of the techniques for computing queries over inconsistent databases work for restricted cases, and only recently have there been proposals to consider more general constraints. This section provides an informal description of the main techniques proposed in the literature.

- Lin and Mendelzon (1996) proposed an approach taking into account the majority view of the knowledge bases in order to obtain a new relation that is consistent with the integrity constraints. The technique proposes a formal semantics to merge first order theories under a set of constraints.

  However, the "merging by majority" technique does not resolve conflicts in all cases since information is not always present in the majority of the databases, and, therefore, it is not always possible to choose between alternative values. Moreover, the use of the majority criteria involves discarding inconsistent data and hence the loss of potentially useful information.
- Arenas et al. (1999) introduced a logical characterization of the notion of consistent answer in a possibly inconsistent database. The technique is based on the computation of an equivalent query $T_\omega(Q)$ derived from the source query Q. The definition of $T_\omega(Q)$ is based on the notion of residue developed in the context of semantic query optimization.