IRM PRESS
701 E. Chocolate Avenue, Hershey PA 17033-1117, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.irm-press.com
ITB9211

Chapter XVIII

Repairing and Querying Inconsistent Databases

Gianluigi Greco
Università della Calabria, Italy

Sergio Greco
Università della Calabria, Italy

Ester Zumpano
Università della Calabria, Italy

ABSTRACT

The integration of knowledge from multiple sources is an important aspect in several areas such as data warehousing, database integration, automated reasoning systems, active reactive databases and others. Thus a central topic in databases is the construction of integration systems, designed for retrieving and querying uniform data stored in multiple information sources. This chapter illustrates recent techniques for computing repairs as well as consistent answers over inconsistent databases. Often databases may be inconsistent with respect to a set of integrity constraints, that is, one or more integrity constraints are not satisfied. Most of the techniques for computing repairs and queries over inconsistent databases work for restricted cases and only recently there have been proposals to consider more general constraints. In this chapter we give an informal description of the main techniques proposed in the literature.

INTRODUCTION

The problem of integrating heterogeneous sources has been deeply investigated in the fields of multidatabase systems (Breitbart, 1990), federated databases (Wiederhold, 1992) and, more recently, mediated systems (Ullman, 2000; Wiederhold, 1992). A large

variety of approaches has been proposed in the literature for performing data source integration. Many of them are embedded in more complex systems managing the interoperability and the cooperation of data sources characterized by heterogeneous representation formats.

The aim of data integration is to provide uniform integrated access to multiple heterogeneous information sources, which were designed independently for autonomous applications and whose contents are strictly related.

Integrating data from different sources consists of two main steps: first, the various relations are merged together, and second, some tuples are *removed* (or *inserted*) from the resulting database in order to satisfy integrity constraints.

In particular, 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, often designed independently for autonomous applications, 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 which 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 which 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 criteria (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; 2) the computation of consistent answers (Arenas et al., 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 of 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 don't know which professor teaches course *c2*.