Chapter 5

TRIO: A SYSTEM FOR DATA, UNCERTAINTY, AND LINEAGE

Jennifer Widom
Dept. of Computer Science
Stanford University
widom@cs.stanford.edu

Abstract

This chapter covers the Trio database management system. Trio is a robust prototype that supports uncertain data and data lineage, along with the standard features of a relational DBMS. Trio’s new ULDB data model is an extension to the relational model capturing various types of uncertainty along with data lineage, and its TriQL query language extends SQL with a new semantics for uncertain data and new constructs for querying uncertainty and lineage. Trio’s data model and query language are implemented as a translation-based layer on top of a conventional relational DBMS, with some stored procedures for functionality and increased efficiency. Trio provides both an API and a full-featured graphical user interface.

Acknowledgments. Contributors to the Trio project over the years include (alphabetically) Parag Agrawal, Omar Benjelloun, Ashok Chandra, Julien Chaumond, Anish Das Sarma, Alon Halevy, Chris Hayworth, Ander de Keijzer, Raghotham Murthy, Michi Mutsuzaki, Tomoe Sugihara, Martin Theobald, and Jeffrey Ullman. Funding has been provided by the National Science Foundation and the Boeing and Hewlett-Packard Corporations.

Keywords: Uncertainty, Trio, ULDB, Lineage

Introduction

Trio is a new kind of database management system (DBMS): one in which data, uncertainty of the data, and data lineage are all first-class citizens. Com-
bining data, uncertainty, and lineage yields a data management platform that is useful for data integration, data cleaning, information extraction systems, scientific and sensor data management, approximate and hypothetical query processing, and other modern applications.

The databases managed by Trio are called ULDBs, for Uncertainty-Lineage Databases. ULDBs extend the standard relational model. Queries are expressed using TriQL (pronounced “treacle”), a strict extension to SQL. We have built a robust prototype system that supports a substantial fraction of the TriQL language over arbitrary ULDBs. The remainder of this Introduction briefly motivates the ULDB data model, the TriQL language, and the prototype system. Details are then elaborated in the rest of the chapter.

Examples in this chapter are based on a highly simplified “crime-solver” application, starting with two base tables:

- **Saw(witness, color, car)** contains (possibly uncertain) crime vehicle sightings.
- **Drives(driver, color, car)** contains (possibly uncertain) information about cars driven.

We will derive additional tables by posing queries over these tables.

**The ULDB Data Model.** Uncertainty is captured by tuples that may include several alternative possible values for some (or all) of their attributes, with optional confidence values associated with each alternative. For example, if a witness saw a vehicle that was a blue Honda with confidence 0.5, a red Toyota with confidence 0.3, or a blue Mazda with confidence 0.2, the sighting yields one tuple in table Saw with three alternative values for attributes color, car. Furthermore, the presence of tuples may be uncertain, again with optionally specified confidence. For example, another witness may have 0.6 confidence that she saw a crime vehicle, but if she saw one it was definitely a red Mazda. Based on alternative values and confidences, each ULDB represents multiple possible-instances (sometimes called possible-worlds), where a possible-instance is a regular relational database.

Lineage, sometimes called provenance, associates with a data item information about its derivation. Broadly, lineage may be internal, referring to data within the ULDB, or external, referring to data outside the ULDB, or to other data-producing entities such as programs or devices. As a simple example of internal lineage, we may generate a table Suspects by joining tables Saw and Drives on attributes color, car. Lineage associated with a value in Suspects identifies the Saw and Drives values from which it was derived. A useful feature of internal lineage is that the confidence of a value in Suspects can be computed from the confidence of the data in its lineage (Section 4). If we generate further tables—HighSuspects, say—by issu-