Overview:

- Incompleteness: Databases often do not contain all the information that they should, either because of delays in the data insertion process or because information is not entered at all.
- Metadata: Often, information about which data is in a database exists or can be derived from business processes.
- Quality information need: User wants to know which queries over a database are reliable (complete).

Use Case: School Data Management in South Tyrol

Distributed IT infrastructure exists, but schools are largely autonomous in their management. Besides core data, many schools manage themselves using other software or using Excel or paper.

Central database is largely incomplete.

Example: Vocational schools put final grades into the database, others often not.

- Query: How many pupils at high schools have grade A in?
  - Result cannot be trusted, data could be missing.
- Query: How many pupils at vocational schools have grade A in?
  - Result can be trusted.

Goal:

A framework for managing information about database completeness.

Challenges

- How to describe the completeness of parts of a database?
- How to reason whether query answers are reliable (complete)?
- How can the reasoning be efficiently implemented?
- How can completeness information be extracted from business processes?
- How can completeness information be used in business intelligence?

Achievements

- Framework for reasoning about database completeness using conjunctive-query formalisms.
- Boolean completeness (Yes/No).
- Tractable reasoning procedures for such statements and SQL-SELECT-JOIN queries.
- Reasoning for databases with NULL values.

Application Scenarios

- Data of organisations, where data management policies are not very strict or where data is maintained also in other forms (paper, MS Excel, ...).
- Data integration: Information about the completeness of the integrated data based on knowledge about the content of the sources.
- Voluntarily created data such as Wikipedia or Openstreetmap.