Detecting Data Inconsistencies with Tuple-Level Constraints

Toon Boeckling [1], Antoon Bronselaer [1], and Guy De Tré [1]

[1] Dep. of Telecommunications and Information Processing, Ghent University, Belgium
{toon.boeckling,antoon.bronselaer,guy.detre}@ugent.be

An important aspect in data quality research is the study of constraint-based mechanisms to assess and improve the consistency of data. These mechanisms typically feature sets of constraints that define how data should (not) look in order to be consistent. Recently, it is shown that constraints that are defined on single tuples (or data objects), are, often, very efficient and effective in finding data that are most likely to be inconsistent [1]. Moreover, in many situations, they are shown to be less complex and easier to work with than the more ‘conventional’ and ‘expressive’ types of constraints, such as denial constraints [2].

Although many issues have already been resolved in the past with regard to this topic, there are still many open research questions, particularly with the current challenges brought by big data in mind. First, the autonomous discovery of constraints can be inefficient and the quality of the discovered constraints can be low, especially in the context of large datasets. Therefore, it can be interesting to study techniques to (further) optimize the search for tuple-level constraints (e.g., selection rules) or to combine different discovery methods to improve the quality. Second, although the results of detecting data inconsistencies with tuple-level constraints are promising, other frameworks, which potentially feature more expressive types of constraints (e.g., HoloClean [4]) or which rely on learning approaches (e.g., Raha [3]), are shown to be useful as well. Therefore, studying how these frameworks can be used collaboratively, such that the effectiveness of error detection can be improved without needlessly compromising scalability, can be of particular interest. Third, at the moment, tuple-level constraints are mainly studied in the context of the relational database model. Therefore, with the increasing importance of NoSQL databases, it can be interesting to study how to generalize the concepts and methods of these types of constraints to other database models (e.g., document stores).

References

[1] Boeckling, T., De Tré, G. and Bronselaer, A. (2022). Cleaning Data with Selection Rules. *IEEE Access*, 10, 125212–125229

[2] Chu, X., Ilyas I. F. and Papotti, P. (2013). Discovering Denial Constraints. *Proceedings of the VLDB Endowment*, 6(13), 1498–1509.

[3] Mahdavi, M., Abedjan, Z., Fernandez, R. C., Madden, S., Ouzzani, M., Stonebraker, M. and Tang, N. (2019). Raha: A Configuration-Free Error Detection System. *Proceedings of the 2019 International Conference on Management of Data.* Association for Computing Machinery.

[4] Rekatsinas, T., Chu, X., Ilyas, I. F. and Ré, C. (2017): HoloClean: Holistic Data Repairs with Probabilistic Inference. *Proceedings of the VLDB Endowment*, 10(11), 1190–1201.