

Letter from the Special Issue Editors

The longevity of the database systems is truly something to marvel at: it has been more than half a century since the introduction of the relational model in 1970. The increasingly important role played by data in modern-day endeavors has further contributed to growing interest in data management. More students—from not only computer science and data science but also other disciplines—see data management skills as an essential part of their training. Nonetheless, this is no time for complacency by the data engineering community. First, our methods for teaching data management skills are becoming woefully inadequate in the presence of growing number of students and diversity in their backgrounds. Second, the exploding number and variety of new data-driven applications demand constant innovations in data management that go far beyond the interfaces and features of traditional database systems. This special issue is devoted to sampling ongoing research from our community to address the above challenges.

The issue begins with two projects focused on improving teaching and learning relational database technology. **Chandra and Sudarshan** present the work on the *XData* system at IIT Bombay, which automatically grades student queries given a set of correct queries. *XData* generates multiple datasets to catch different errors ensuring better test coverage, it suggests multiple correct query structures that are helpful for partial grading, and also provides individualized feedback on what changes are needed to correct a query. **Bhowmick and Li** describe the *TRUSS* project at the Nanyang Technological University, which helps students learn relational query processing. Its goals include helping users understand executing plans, as well as how database optimizers choose among various alternative plans. Guided by the motivation theories of learning and informed by data collected throughout the learning process, *TRUSS* employs a variety of modes to assist learners, such as natural language explanations, visualizations, and an interactive chatbot. Both papers share their experience of using these two systems in database courses at their respective universities.

The third paper, by **Gatterbauer et al.**, presents the principles of query visualization, which aims at helping humans understand the meaning of a query written in SQL. This visualization paradigm has not only applications in education, but with a graphical representation of queries that abstract away unnecessary syntactical details, it also enables interesting uses such as identifying code reuse opportunities and clustering query patterns.

The subsequent papers in this issue seek to extend database technology for novel data interfaces and applications. The paper from UC San Diego by **Shao et al.** tackles a key challenge in building natural language interfaces for data: how to learn representations of structured data, for text- and speech-to-SQL translation, as well as for building dialog systems for tasks such as helping users produce plots from structure data. The paper from Ohio State by **Burley et al.** is about support for developing augmented reality applications. These applications virtually augment real-world objects with additional information by performing real-time “joins” between data extracted from the physical environment with data from remote sources. The *Quill* framework they present simplifies the development process using declarative specifications, and uses automatic optimization to achieve better performance than code developed with standard tools. The paper from NYU Abu Dhabi and UMass Amherst by **Abouzieed et al.** discusses opportunities for in-database decision support and the challenges related to usability, scalability, data uncertainty, dynamic environments with changing data and models, and robust policy making. Their *PackageBuilder* system and its extensions transform a declarative specification of a package query into an integer linear program, and solve it on deterministic or uncertain data to obtain the desired package of input tuples with minimum cost or maximum profit while satisfying specified constraints.

The last two papers in this issue are about exploratory data analysis (EDA). **Peng et al.** present a survey of several open-source and commercial interfaces for exploratory data analysis. They focus on user exploration activities in three stages of EDA: generating initial questions to get some initial questions to explore, creating visualizations to answer questions, and examining the visualizations to produce answers or ask more questions. **Kennedy et al.** describe *Vizier*, an extensible multi-modal platform for data-centric workflows. The main idea behind *Vizier* is that, rather than alternating between task-specific tools such as spreadsheets and notebooks for data exploration, discovery, and analysis, a better approach is to build multiple user interfaces on top of a single

incremental workflow/dataflow platform with built-in support for versioning, provenance, error discovery, output tracking, and data cleaning.

Overall, we hope these eight articles together offer a sample of the ongoing work as well as exciting challenges in widening the impact of database engineering community — both through education and through novel interfaces and features. We would like to thank the authors of this issue for their contributions, and welcome more from our community to join them.

Sudeepa Roy and Jun Yang
Duke University, USA