Reminiscences on Influential Papers

Kenneth A. Ross, editor

I continue to invite unsolicited contributions. See `http://www.acm.org/sigmod/record/author.html` for submission guidelines.

Alfred V. Aho, Columbia University, `aho@cs.columbia.edu`.

[E. F. Codd. A Relational Model for Large Shared Data Banks. Comm. ACM 13(6), pp. 377–387, 1970.]

I've long been fascinated by the question of how to organize information so that it is conceptually simple to grasp its organization, yet at the same time being easy to store and retrieve. I was impressed with simplicity, elegance and power of the relational model at the time it was proposed by Codd, particularly his natural join operator. The simplest way to think about data is in the form of a single universal relation scheme, even though the data may be stored in a collection of smaller schemes. So this naturally led to the question, under what conditions can a relation scheme be decomposed into smaller schemes without losing information – the so-called lossless join decomposition. In the late 1970s, Catriel Beeri, Jeff Ullman and I determined when this is possible, and subsequently Yehoshua Sagiv, Jeff Ullman and I provided a general method for testing for the equivalence of relational expressions. If we look at our understanding of relational data today, the database research community can take great pride in the deep, rich theory that Codd's seminal paper stimulated.

Anastassia Ailamaki, Carnegie Mellon University, `natassa@cmu.edu`.

[Goetz Graefe. Query Evaluation Techniques for Large Databases. ACM Computing Surveys (CSUR) 25(2), June 1993.]

It is extremely difficult to write a really good survey about concepts and algorithms that our community uses every day, such as joins, sorting, and hashing — this stuff is, to a certain extent, in every textbook. During my graduate student years at UW-Madison, I would often get lost in the labyrinths of code writing and would surface back to this paper to place my thinking into the proper perspective, to remember what is important, and to understand the details of query processing all over again. This paper is a celebration of Graefe's charisma to present the intuition behind all the query evaluation techniques used in today's database query processing, explaining not only *how* the algorithms work, but *why* they were designed this way. It covers a broad range of algorithms, from conventional methods to parallel and "non-standard" query processing, in a wonderfully comprehensive and intuitive fashion.

At Carnegie Mellon, I work with several students of non-database background that take an interest in query processing as part of their research. Every single time a student has asked me to provide a reading on how the query engine works, I have recommended parts of this paper, and it has been a quite efficient way to learn. The students read it to understand the concepts, and use it thereafter for frequent reference, just like I did. This work is of immense educational value, regardless of whether the reader is a database expert or not. Its amazing depth and breadth helped shape my ideas as a graduate student, and continue to influence my research as faculty.
