Elizabeth Fong: Creating the SQL Database Standards

Charles Severance, *University of Michigan*

Elizabeth Fong describes the standards behind SQL.

The rotating mass storage that enabled random access to large amounts of data is one of the key technologies that helped computing evolve from long-running batch processes to something more in the moment and interactive. However, these mass storage devices were limited by seek time and rotational latency, so computer scientists and vendors spent decades optimizing how applications could best utilize them. My first database application back in the early 1980s was on a dual-floppy IBM PC, with the database software and my application on one floppy disk and the data itself on another. You could actually hear seek time and rotational delay in action as the application executed.

By the end of the 1980s, the market for database management systems was starting to mature and converge around the Structured Query Language (SQL) that we still use today to develop database applications. I spoke with Elizabeth Fong of the National Institute of Standards and Technology (NIST) about the early standardization efforts around SQL. To hear more of our conversation, visit www.computer.org/computingconversations.

BEFORE STANDARDIZATION

In the late 1970s, companies like IBM and Oracle were building increasingly sophisticated database management engines. Everyone wanted to maximize the performance of the applications built on top of these proprietary systems:

> Initially, we managed data with file systems. IBM's IMS—Information Management System—had a hierarchical tree structure for its files. There was a continuous debate about whether one should use trees, networks, or flat files as a base structure. We're still debating whether the data has self-describing tags or not.

The US federal government was a large consumer of these expensive and proprietary technologies. To help make sense of the emerging market in the late 1970s, the National Bureau of Standards (NBS, which is now NIST) was charged with building consensus around common approaches to simplify procurements. According to Fong,

> Many products were coming up in the wild. Government agencies would ask whether they should buy Oracle, IBM, or something cheaper. We wanted to build a wide variety of applications on top of database management systems, so we needed a standard—applications needed to work on multiple platforms. We had about a half-dozen products, so the standardization challenge was really about building consensus.

NBS wasn't allowed to simply make the standard by fiat: the standard needed to emerge from a consensus process that involved representatives from the technology industry as well the end users who would be purchasing the systems. It could neither lead the effort nor force participation in the process, but with the promise that government procurements would someday require compliance to the standard, there was broad interest and participation.

BUILDING A MODEL

The first working group simply tried to come up with a sensible model and some terminology to lay the foundation for later technical standards:

Published by the IEEE Computer Society

> The Database System Study Group [DBSSG] came up with a reference model for minimal functionality. To be a database management system [DBMS], it had to be able to store data, retrieve data, modify data, organize data, and delete data.

The next step was to begin the actual development of technical specifications in the American National Standards Institute. ANSI develops proposed specifications and represents the US in the International Organization for Standardization (ISO) process. The overall goal was ultimately to produce an ISO standard, but the technical work started in the ANSI organization:

> During that time, we initiated the creation of an ANSI group called X3H2, which is now called ANSI INCITS H2. Folks like Don Deutsch and Len Gallagher were two of the many participants and leaders of that effort, which came to be called the Data Management Language Group.

This group focused on the development of a language to communicate between an application and the DBMS:

> The only thing that you standardize is communication across an interface, where both sides need to understand a common vocabulary. So the only way to standardize a software system isn't through capabilities but through a language.

While some in the early 1980s were still debating the best model for data storage and retrieval, the relational model was starting to represent the broadest consensus in the marketplace:

> Relational databases were being promoted by IBM and Chris Date. He talked about normalization and flat files and tables. Relational was a very easy concept, and everybody understood it. To retrieve data from a table, you would "SELECT" columns from a table for rows containing the specific value. And that was the birth of a simple query language.

By the late 1980s, the ANSI X3H2 committee had completed and published the first SQL specifications. The standard specified the syntax and semantics of two database languages: a schema language definition language (SQL-DDL) and a data manipulation language (SQL-DML). While the language was simple and straightforward, it took a significant amount of research and development to build highly optimized DBMS implementations to support the relational model.

COMPLIANCE

Once standards were in place and the industry was providing RDMSs, NIST (renamed in 1988) turned its focus to building test suites and certification processes for products claiming to comply with the new standards:

> Conformance testing is a very important aspect of standardization. When you adopt a standard, you want to ensure that the product conforms to a particular version of the standard so they are interoperable. Companies develop products to meet the standard. Products are then tested for conformance to the requirements of the standard often by a third-party testing laboratory, accredited and deemed competent to perform the testing by an accrediting body such as NIST's National Voluntary Laboratory Accreditation Program [NVLAP]. If the product meets the standard, the results are publicized either by the company or by user organizations.

Even though NIST was an important force in creating SQL standards and the conformance testing processes around those standards, it leaves all final purchase decisions to the government agencies doing the procurements:

> Using standards and certifications is strictly up to purchasers because they're paying the money, not NIST.

I asked Fong why the SQL standard turned out to be so successful and continues to be so relevant:

> Timing is everything. You can't standardize a technology too early, or else you drive innovative concepts away. If you standardize prematurely, companies think there's no way to enter the market because everything is already decided. Even if the resulting standards and implementations aren't very good, there's no point in developing a better product because it will be "non-standard." This kills innovation. If you start the process too late, you miss the opportunity and end up with too much variety in the marketplace; too many different approaches become well established and resist any kind of market-wide standardization.

SQL brought the right standard forward at the right time. Its initial development took over 10 years, during which the standards process was both informed by and contributed to research and development efforts. Companies also had enough time to adjust their strategies and embrace the relational approach before the standard was finalized. Today, we greatly benefit from the visionary leadership of NIST and those involved in the ANSI X3H2 (now INCITS) effort all those years ago.

Charles Severance, *Computing Conversations column editor and Computer's multimedia editor, is a clinical associate professor and teaches in the School of Information at the University of Michigan. Follow him on Twitter @drchuck or contact him at csev@umich.edu.*