JUNE 4, 2018

Legaltech news ONLINE

An ALM Publication

Nervous System: The Small Start of Big Data

'Nervous System,' which approaches data privacy and cybersecurity issues from the context of history, tells the story of modern database systems and the evolution of storing data.

BY DAVID KALAT, BERKELEY RESEARCH GROUP

With the aggressive pace of technological change and the onslaught of news regarding data breaches, cyber-attacks, and technological threats to privacy and security, it is easy to assume these are fundamentally new threats. The pace of technological change is slower than it feels, and many seemingly new categories of threats have actually been with us longer than we remember. ***Nervous System*** *is a monthly blog that approaches issues of data privacy and cybersecurity from the context of history—to look to the past for clues about how to interpret the present and prepare for the future.*

In the early days of computers, inefficient data storage posed a significant barrier to realizing benefits of the information revolution. Early computer scientists tended to organize data in forms that mimicked the familiar organization of information in the human world. For example, a bank's database might list a customer's name, account number, home address, telephone number, and account balance, and then follow that with the *next* customer's name and associated information, and so on. If a given customer had more than one account, their data file would redundantly include their name and address information on each one. If the bank wanted to calculate the total value of the accounts, the computer would need to read through the entire data file, sequentially tallying each account balance. Most early computer systems stored data on reel-to-reel magnetic tape or stacks of punch cards, meaning that scanning an entire data file to calculate a bank balance could entail a long and tedious process.

Computer hardware and data storage media was expensive and bulky. (Users with deep pockets and lots of room might partake of the earliest hard-disk drives—for a few thousand dollars you could buy a disk drive, the size of a small chest freezer, that could hold upward of 10 MB). Also, each organization used a different data system, with no industrywide standards. It was rare for two databases to share the same design, meaning that data was all but impossible to port between systems. To use an organization's data, a specialist programmer had to be trained in the idiosyncrasies specific to that data set. To use a different organization's data, the user would essentially start from scratch.

For businesses to start realizing the value of storing and managing their data in computer systems, the costs and burdens had to come

down. To do that, systems needed to become more efficient.

Enter Edgar Codd, the pioneer behind modern database systems. Codd was an Oxford-educated mathematician who moved to the US after World War II. America was hungrily snapping up programmers and mathematicians to staff the burgeoning computer science field, but at the same time, ugly Cold War prejudices were also at play. Disgusted by the anti-Communist witch hunts led by Senator Joseph McCarthy, Codd left the US not long after arriving. Eventually he was beckoned back, and as an employee of IBM he developed a visionary idea.

Codd's breakthrough was the concept of storing data as a collection of smaller tables, each of which would be connected to the others through a relationship—Codd called the tables "relations" because the design would encode their relationship to one another. In this model, instead of a bank listing each account's data sequentially in a so-called "flat file," customer names would be listed in one relational table, addresses in another, account balances in a third, and so on.

Codd published his proposal in a landmark 1970 paper, *A Relational Model of Data for Large Shared Data Banks*. The paper opened with a bold statement: "Future users of large data banks must be protected from having to know how the data is organized in the machine."

Ironically, one could also argue the future users of large data banks needed some protection from having to know Codd's convoluted and off-putting mathematical jargon. Codd was every bit the dedicated mathematician. When he composed his "Twelve Rules" for ensuring database integrity, the list actually includes *thirteen* rules, because it was numbered 0 to 12. When he articulated a proposal for how users would interact with and query his relational database model, it was based on a language of mathematical symbols he called "Relational Algebra."

Codd's employer, IBM, was unsure of the commercial viability, but tasked a couple of math geeks with developing a working prototype as an academic exercise. Raymond Boyce and Don Chamberlin's first step was to take Codd's Relational Algebra and rework it as something ordinary users could type in using a regular keyboard. They created a language they called the Structured English Query Language, or "SEQUEL," to map Codd's Relational Algebra onto familiar everyday words. IBM allowed Chamberlin and Boyce to publish a paper on their SEQUEL language in 1974, but the company still had no plans to pursue the idea commercially.

Chamberlin continued his work at IBM on trying to engineer a functional relational database prototype (shortly after the 1974 paper's release, Boyce died of a brain aneurysm at the age of 25). A separate competitive team was working in parallel at UC Berkeley on the same goal, using a language they called QUEL. By the late 1970s and early 1980s, competing commercial versions of SEQUEL (now renamed SQL) were available alongside versions of Berkeley's QUEL. History would choose one over the other.

The tipping point came with entrepreneur Larry Ellison, whose startup Software Development Laboratories implemented a SQL-based system under the name "Oracle." The product was so outrageously successful, the company soon assumed the same name. After Oracle became a government contractor, the US government formally adopted SQL in Federal Information Processing Standard 127. This was followed by SQL's adoption as an American National Standard (ANSI) and International Standard (ISO). Making SQL the default language of relational databases finally realized the longstanding ambition of making data truly portable between systems.

Thanks to this common platform, users of computers could now store more data and calculate faster; users trained on one system could port that knowledge to other systems easily; and a new industry of third-party software developers could evolve to provide specialized tools that did not need to be uniquely customized to each individual system. SQL fundamentally transformed the way people interact with computers by fundamentally transforming the way computers interact with data. Today the relational database model is the backbone of a $28 billion-a-year industry.

David Kalat *is Director, Global Investigations + Strategic Intelligence at Berkeley Research Group. David is a computer forensic investigator and e-discovery project manager. Disclaimer for commentary: The views and opinions expressed in this article are those of the author and do not necessarily reflect the opinions, position, or policy of Berkeley Research Group, LLC or its other employees and affiliates.*

Originally published on Corporate Counsel. Reprinted with permission from the June 4, 2018 edition of Legaltech News. © 2018 ALM Media Properties, LLC. All rights reserved. Further duplication without permission is prohibited. For information, contact 877-257-3382 or reprints@alm.com. #010-06-18-01