Introduction

Michael L. Brodie, Computer Systems and Artificial Intelligence Laboratory,
Massachusetts Institute of Technology

Our story begins at University of California, Berkeley in 1971, in the heat of the Vietnam War. A tall, ambitious, newly minted assistant professor with an EE Ph.D. in Markov chains asks a seasoned colleague for direction in shaping his nascent career for more impact than he could imagine possible with Markov chains.[1] Professor Eugene Wong suggests that Michael Stonebraker read Ted Codd's just-published paper on a striking new idea: *relational databases*. Upon reading the paper, Mike is immediately convinced of the potential, even though he knows essentially nothing about databases. He sets his sights on that pristine relational mountaintop. The rest is history. Or rather, it is the topic of this book. Ted's simple, elegant, relational data model and Mike's contributions to *making databases work* in practice helped forge what is today a $55 billion industry. But, wait, I'm getting ahead of myself.

A Brief History of Databases

What's a database? What is data management? How did they evolve?

Imagine accelerating by orders of magnitude the discovery of cancer causes and the most effective treatments to radically reduce the 10M annual deaths worldwide. Or enabling autonomous vehicles to profoundly reduce the 1M annual traffic deaths worldwide, while reducing pollution, traffic congestion, and real estate wasted on vehicles that on average are parked 97% of the time. These are merely two examples of the future potential positive impacts of using Big Data. As with all technical advances, there is also potential for negative impacts, both unintentional—in error—and intentional such as undermining modern democracies (allegedly well under way). Using data means managing data efficiently at scale; that's what data management is for.

In its May 6, 2017 issue, *The Economist* declared data to be the *world's most valuable resource*. In 2012, data science—often AI-driven analysis of data at scale—exploded on the world stage. This new, data-driven discovery paradigm may be one of the most significant advances of the early 21st century. Non-practitioners are always surprised to find that 80% of the resources required for a data science project are devoted to data management. The surprise stems from the fact that data management is an infrastructure technology: basically, unseen plumbing. In business, data management "just gets done" by mature, robust database management systems (DBMSs). But data science poses new, significant data management challenges that have yet to be understood let alone addressed.

The above potential benefits, risks, and challenges herald a new era in the development of data management technology, the topic of this book. *How did it develop in the first place*? *What were the previous eras*? *What challenges lie ahead*? This introduction briefly sketches answers to those questions.

This book is about the development and ascendency of data management technology enabled by the contributions of Michael Stonebraker and his collaborators. Novel when created in the 1960s, data management became the key enabling technology for businesses of all sizes worldwide, leading to today's $55B[2],[3] DBMS market and tens of millions of operational databases. The average Fortune 100 company has more than 5,000 operational databases, supported by tens of DBMS products.

---

[1] "I had to publish, and my thesis topic was going nowhere."—Mike Stonebraker

[2] https://www.statista.com/statistics/724611/worldwide-database-market/

[3] Numbers quoted in this chapter are as of mid-2018.

Databases support your daily activities, such as securing your banking and your credit card transactions. So that you can buy that Starbucks latte, Visa, the leading global payments processor, must be able to simultaneously process 50,000 credit card transactions per second. These "database" transactions update not just your account and those of 49,999 other Visa cardholders, but also those of 50,000 creditors like Starbucks, while simultaneously validating you, Starbucks, and 99,998 others for fraud, no matter where your card was swiped on the planet. Another slice of your financial world may involve one of the more than 3.8B trade transactions that occur daily on major U.S. market exchanges. DBMSs support such critical functions not just for financial systems, but also for systems that manage inventory, air traffic control, supply chains, and all daily functions that depend on data and data transactions. Databases managed by DBMSs are even on your wrist and in your pocket if you, like 2.3B others on the planet, use an iPhone or Android smartphone. If databases stopped, much of our world would stop with them.

A database is a logical collection of data, like your credit card account information, often stored in records that are organized under some data model, such as tables in a relational database. A DBMS is a software system that manages databases and ensures persistence—so data is not lost—with languages to insert, update, and query data, often at very high data volumes and low latencies, as illustrated above.

Data management technologies have evolved through four eras over six decades, as is illustrated in the Data Management Technology Kairometer.

In the inaugural ***navigational era*** (1960s), the first DBMSs emerged. In them, data, such as your mortgage information, was structured in hierarchies or networks and accessed using record-at-a-time navigation query languages. The navigational era gained a database Turing Award in 1973 for Charlie Bachman's "*outstanding contributions to database technology*."

In the second, ***relational era*** *(*1970s-1990s), data was stored in tables accessed using a declarative, set-at-a-time query language, SQL: for example, "Select name, grade From students in Engineering with a B average." The relational era ended with approximately 30 commercial DBMSs dominated by Oracle's Oracle, IBM's DB2, and Microsoft's SQL Server. The relational era gained two database Turing Awards: one in 1981 for Ted Codd's "*fundamental and continuing contributions to the theory and practice of database management systems, esp. relational databases*" and one in 1998 for Jim Gray's "*seminal contributions to database and transaction processing research and technical leadership in system implementation*."

The start of Mike Stonebraker's database career coincided with the launch of the relational era. Serendipitously, Mike was directed by his colleague, UC Berkeley professor Eugene Wong, to the topic of data management and to the relational model via Ted's paper. Attracted by the model's simplicity compared to that of navigational DBMSs, Mike set his sights, and ultimately built his career, on *making relational databases work*. His initial contributions, Ingres and Postgres, did more to make relational databases work in practice than those of any other individual. After more than 30 years, Postgres—via PostgreSQL and other derivatives—continues to have a significant impact. PostgreSQL is the third[4] or fourth[5] most popular (used) of hundreds of DBMSs, and all relational DBMSs implement the object-relational data model and features introduced in Postgres.

In the first relational decade, researchers developed core RDBMS capabilities including query optimization, transactions, and distribution. In the second relational decade, researchers focused on high-performance queries for data warehouses (using column stores) and high-performance transactions and real time analysis (using in-memory databases); extended RDBMSs to handle additional data types and processing (using abstract data types); and tested the "one-size-fits-all" principle by fitting myriad application types into RDBMSs. But complex data structures and operations—such as those in geographic information, graphs, and scientific processing over sparse matrices—just didn't fit. Mike knew because he pragmatically exhausted that premise using real, complex database applications to push RDBMS limits, eventually concluding that "one-size-does-*not*-fit-all" and moving to special-purpose databases.

With the rest of the database research community, Mike turned his attention to special-purpose databases, launching the third, ***non-relational era*** (2000-2010). Researchers in the non-relational era

[4] https://www.eversql.com/most-popular-databases-in-2018-according-to-stackoverflow-survey/

[5] https://db-engines.com/en/ranking

developed data management solutions for non-relational data and associated processing (e.g., time series, semi- and un-structured data, key-value data, graphs, documents) in which data was stored in special-purpose forms and accessed using non-SQL query languages, called NoSQL, of which Hadoop is the best-known example. Mike pursued non-relational challenges in the data manager but, claiming inefficiency of NoSQL, continued to leverage SQL's declarative power to access data using SQL-like languages, including an extension called NewSQL.

New DBMSs proliferated, due to the research push for and application pull of specialized DBMSs, and the growth of open source software. The non-relational era ended with more than 350 DBMSs, split evenly between commercial and open source and supporting specialized data and related processing including (in order of utilization): relational, key-values, documents, graphs, time series, RDF, objects (object-oriented), search, wide columns, multi-value/dimensional, native XML, content (e.g., digital, text, image), events, and navigational. Despite the choice and diversity of DBMSs, the market remained dominated by five relational DBMSs: the original three plus Microsoft Access and Teradata, which Mike came to call collectively "*the Elephants*." Although the Elephants all supported Mike's object-relational model, they had become "conventional wisdom" that lagged new data management capabilities. A hallmark of Mike's career is to perpetually question conventional wisdom, even of his own making. At the end of the non-relational era, there was a significant shift in the DBMS market away from the RDBMS Elephants and to less-expensive open source DBMSs.

The relational and post-relational eras gained a database Turing Award for Stonebraker's "*concepts and practices underlying modern database systems*." It is the stories of these relational and non-relational developments that fill these pages. You will find stories of their inception, evolution, experimentation, demonstration, and realization by Mike and his collaborators through the projects, products, and companies that brought them to life.

This brings us to the fourth and current ***Big Data era*** (2010-present), characterized by data at volumes, velocities, and variety (heterogeneity) that cannot be handled adequately by existing data management technology. Notice that, oddly, "Big Data" is defined in terms of data management technology rather than in terms of the typically-real-world phenomena that the data represents. Following the previous eras, one might imagine that the Big Data era was launched by the database research community's pursuit of addressing future data management requirements. That is not what happened. The database community's focus remained on relational and non-relational for three decades with little concern for a grander data management challenge—namely managing *all* data as the name "data management" suggests. The 2012 annual IDC/EMC Digital Universe study[6] estimated that of all data in the expanding digital universe, less than 15% was amenable to existing DBMSs. Large enterprises like Yahoo! and Google faced massive data management challenges for which there were no data management solutions in products or research prototypes. Consequently, the problem owners built their own solutions, thus the genesis of Hadoop, MapReduce, and myriad NoSQL, Big Data managers. In 2009, Mike famously criticized MapReduce to the chagrin of the Big Data community, only to be vindicated five years later when its creators disclosed that Mike's criticism coincided with their abandonment of MapReduce and Hadoop for yet another round of Big Data management solutions of their own making. This demonstrates the challenges of the Big Data era and that data management at scale is hard to address without the data management underpinnings established over six decades. Retrospectively, it illustrates the challenges faced in the previous database eras and validates that the solutions warranted database Turing Awards.

We appear to be entering a golden age of data, largely due to our expectations for Big Data: that data will fuel and accelerate advances in every field for which adequate data is available. Although this era started in 2010, there has been little progress in corresponding data management solutions. Conventional DBMS "wisdom," as Stonebraker loves to say, and architectures do not seem to apply. Progress, at least progress Stonebraker-style (as we will see throughout the book), is hindered by a paucity of effective use cases. There are almost no (1) reasonably well-understood Big Data management

[6] Gantz, J. and Reinsel, D., The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East — United States, IDC, February 2013

applications that are (2) owned by someone with a pain that no one else has addressed, with (3) the willingness to provide access to their data and to be open to exploring new methods to resolve the pain.

The bright side of this is the explosion of data-intensive applications leading to, as Michael Cafarella and Chris Ré of Stanford University say in [1], a "blindingly bright future" for database research. The dominant class of data-intensive processing is data science, itself just emerging as a domain; thus, its use above as an example of the need for effective new database management technologies. As the data management technology evolution story continues, the overwhelming interest in data requires a distinction between core data management technology, the topic of this book, and its development to support activities in every human endeavor.

If you are a young researcher or engineer contemplating your future in data management or in computer systems, you now find yourself at the dawn of the Big Data era, much as Mike found himself at the beginning of the relational era. Just as Mike was new to the then-new idea of relational databases, so too you must be new to the as-yet-undefined notion of a Big Data system and data manager. These pages tell the stories of Mike's career seen from the many different perspectives of his primary collaborators. These stories may give you an historical perspective and may provide you guidance on your path: what issues to pursue, how to select them, how to pursue them, how to collaborate with people who complement your knowledge, and more. These stories recount not only challenges, technology, methods, and collaboration styles, but also people's attitudes that, perhaps more so than technology, contributed to the state of data management today, and specifically to the achievements of Mike Stonebraker and his collaborators. However, the world now is not as it was in 1971 as Mike launched his career, as Mike discusses in the chapter "Where we have failed."

Preparing to Read the Stories and What You Might Find There

To explore and understand Lithuania on a trip, few people would consider visiting every city, town, and village. So how do you choose what to see? Lonely Planet's or Eyewitness' travel guides to Estonia, Latvia, and Lithuania are fine, but they are 500 pages each. It's better to be motivated and have worked out the initial questions that you would like answered. Here is a method for visiting Lithuania and for reading the stories in this book.

The book was written by computer systems researchers, engineers, developers, startup managers and funders, and Silicon Valley entrepreneurs, executives, and investors for people like themselves and for the broad computing community. Let's say you aspire to a career in, or are currently in, one of those roles: *What would you like to learn from that perspective*?

Say for example that you are a new professor in Udine, Italy, planning your career in software systems research. As Mike Stonebraker was at UC Berkeley in 1971, you ask: *What do I want to do in my career*? *How do I go about realizing that career*? Develop a list of questions you would like to pursue such as the 20 or so italicized questions in this Introduction and the Preface. Select the stories that appear to offer answers to your questions in a context in which you have some passion. Each story results in significant successes, perhaps on different topics than interest you or in a different culture and a different time. Nonetheless, the methods, the attitudes, and the lessons are generally independent of the specific story. The contributors have attempted to generalize the stories with the hindsight and experience of as much as 40 years.

Choose your role, figure out the questions on which you would like guidance, choose your own perspective (there are 30 in the book); and set off on your journey, bailing when it is not helping. Become your own software systems ciceroni.[7]

A Travel Guide to Software Systems Lessons in Nine Parts

The thirty stories in this book are arranged into nine Parts.

[7] Expert tourist guide, who makes the trip worthwhile, derived from Marcus Tullius Cicero, Roman politician and orator known for guiding through complex political theses, diminished today to restaurants.

Part 1, **2014 ACM A.M. Turing Award Paper and Lecture,** contains the paper in which Turing awardees typically describe the achievements for which the award was conferred. The paper is given as a lecture at ACM conferences during the award year. True to Mike's idiosyncratic nature, he used the paper and lecture as a platform for what he considered his most important message for the community, as opposed to the already published technical achievements. Mike writes of challenges posed by software systems, the theme of this book as explained in the Preface. Mike described the nature of the challenges by analogy with significant physical challenges—something most audiences can understand—from Mike's own personal life. The paper is reproduced from the *Communications of the ACM*. The lecture was given initially at the Federated Computing Research Conference, June 13, 2015 and can be viewed online.[8]

Part 2, Mike Stonebraker's Career, lays out Mike's career in Sam Madden's biography and in two graphic depictions. Chart 1 lists chronologically Mike's Ph.D. students and postdocs. Chart 2 illustrates the academic projects and awards and the creation and acquisition of his companies. On April 12, 2014, Mike Carey (University of California, Irvine), David DeWitt (then at University of Wisconsin), Joe Hellerstein (University of California, Berkeley), Sam Madden (MIT), Andy Pavlo (Carnegie Mellon University), and Margot Seltzer (Harvard University) organized a Festschrift for Mike: a day-long celebration of Mike Stonebraker at 70.[9] More than 200 current and former colleagues, investors, collaborators, rivals, and students attended. It featured speakers and discussion panels on the major projects from Mike's +40-year career. Chart 2 *The Career of Mike Stonebraker* was produced for the Festschrift by Andy Pavlo and his wife.

Part 3, Mike Stonebraker Speaks Out: An Interview, is a post-Turing-Award interview in the storied series of interviews of database contributors by Marianne Winslett. A video of the interview can be seen online.[10]

In ***Part 4, The Big Picture***, world leading researchers, engineers, and entrepreneurs reflect on Mike's contributions in the grander scope of things. Phil Bernstein, a leading researcher and big thinker, reflects on Mike's leadership and advocacy. James Hamilton, a world-class engineer and former lead architect of DB2—an Elephant—reflects on the value of Turing contributions. Jerry Held, Mike's first PhD student, now a leading Silicon Valley entrepreneur, recounts experiences collaborating and competing with Mike. Dave DeWitt, comparable to Mike in his data management contributions, reflects on 50 years as a mentee, colleague, and competitor.

Part 5, Startups, tells one of the 21st century's hottest stories: how to create, fund, and run a successful technology startup. As the *Data Management Technology Kairometer* illustrates, Mike has co-founded more than nine startups. The startup story is told from three distinctly different points of view; from that of the technical innovator and Chief Technology Officer (Mike), from that of the CEO, and from that of the prime funder. These are NOT mere *get rich quick* or *get famous quick* stories. Startups and their products are an integral component of Mike Stonebraker's database technology research and development methodology, to ensure that the results have impact. This theme of industry-driven and industry-proven database technology research pervades these pages. If you get only one thing from this book, let this be it.

Part 6, Database Systems Research, takes us into the heart of database systems research. Mike answers: *Where do ideas come from? How to exploit them*? Take a master class with a Turing Award winner on how to do database systems research. Mike's invitation to write the "Failures" chapter was: *So, Mike, what do you really think*? His surprising answers lay out challenges facing database systems research and the database research community. Mike Olson, a Ph.D. student of Mike and an extremely successful co-founder and Chief Strategy Officer of Cloudera (which has a data management product based on Hadoop), lays out Mike's contributions relative to the open source movement that was launched after Ingres was already being freely distributed. Part 6 concludes with Felix Naumann's amazing Relational Database Management Genealogy, which graphically depicts the genealogy of hundreds of RDBMSs from 1970 to the present showing how closely connected RDBMSs are in code, concepts, and/or developers.

---

[8] https://www.youtube.com/watch?v=BbGeKi6T6QI

[9] For Photographs see http://stonebraker70.com

[10] Morgan and Claypool has this video – please put a link to the video where M&C determines to place it

Part 7, Contributions by System, describes the technical contributions for which the Turing award was given. They are presented chronologically in the context of the nine projects, each centered on the software system in which the contributions arose. Each system is described from a research perspective in ***Part 7.1, Research Contributions by System,*** and from a systems perspective, in a companion story in ***Part 7.2, Contributions from Building Systems***. Sam Madden's Overview synopsizes the major technical achievements and offers a map to contributions and their stories in parts 7.1 and 7.2.

The stories of ***Part 7.1, Research Contributions by System*****,** focus on the major technical achievements that arose in each specific project and system. They do not repeat the technical arguments from the already published papers that are all cited in the book. The research chapters explain the technical accomplishments: their significance, especially in the context of the technology and application demands of the time; and their value and impact in the resulting technology, systems, and products that were used to prove the ideas and in those that adopted the concepts. Most stories, like Daniel Abadi's "C-Store: Through the Eyes of a Ph.D. Student," tell of career decisions made in the grips of challenging and rewarding research. The first seven systems—Ingres, Postgres, Aurora, C-Store, H-Store, SciDB, and Data Tamer—span 1972 to 2018, including the relational, non-relational, and now the Big Data eras of data management. All seven projects resulted in successful systems, products, and companies. Not included in the book are two systems that Mike does not consider successful. (But if some didn't fail, he wasn't trying hard enough.) The final two projects, BigDAWG and Data Civilizer, are under way at this writing as two of Mike's visions in the Big Data world.

The stories of ***Part 7.2, Contributions from Building Systems,*** are a little unusual in that they tell seldom-told stories of heroism in software systems engineering. The development of a software system, e.g., a DBMS, is a wonderful and scary experience for individuals, the team, and the backers! There is a lot of often-unsung drama: inevitable disasters and unbelievable successes, and always discoveries, sometimes new but often repeated for the umpteenth time. Key team members of some of the best-known DBMSs in the world (DB2, Ingres, Postgres) tell tales of the development of the codelines we have come to know as DB2, Ingres, StreamBase, Vertica, VoltDB, and SciDB, and the data unification system Tamr.

These stories were motivated, in part, by an incidental question that I asked James Hamilton, former lead architect, IBM DB2 UDB, which (with Stonebraker's Ingres) was one of the first relational DBMSs and became an Elephant. I asked James: "Jim Gray told me some fascinating stories about DB2. What really happened?" What unfolded was a remarkable story told in Chapter 31, "IBM Relational Database Code Bases," which made it clear that all the codeline stories must be told. One sentence that got me was: "Instead of being a punishing or an unrewarding 'long march,' the performance improvement project was one of the best experiences of my career." This came from one of the world's best systems architects, currently, Vice President and Distinguished Engineer, Amazon Web Services.

But more importantly, these stories demonstrate the theme of the book and Mike's observation that "*system software is so hard to build, and why good teams screw it up on a regular basis.*"

From the beginning of Mike's database career, the design, development, testing, and adoption of prototype and commercial systems have been fundamental to his research methodology and to his technical contributions. As he says in Chapter 9: Where Good Ideas come from and How to Exploit Them, "Ingres made an impact mostly because we persevered and got a real system to work." Referring to future projects, he says "In every case, we built a prototype to demonstrate the idea. In the early days (Ingres/Postgres), these were full-function systems; in later days (C-Store/H-Store) the prototypes cut a lot of corners." These systems were used to test and prove or disprove research hypotheses, to understand engineering aspects, to explore details of real use case, and to explore the adoption, hence impact, of the solutions. As a result, research proceeded in a virtuous cycle in which research ideas improved systems and, in turn, systems and application challenges posed research challenges.

Part 8, Perspectives, offers five personal stories. James Hamilton recounts developing one of the world's leading DBMSs, which included the highlights of his storied engineering career. Raul Castro Fernandez recounts how, as a Stonebraker postdoc, he learned how to do computer systems research—how he gained research taste. Marti Hearst tells an engaging story of how she matured from a student to a researcher under a seemingly intimidating but actually caring mentor. Don Haderle, a product developer on IBM's Systems R, the alleged sworn enemy of the Ingres project, speaks admiringly of the competitor who became a collaborator and friend. In the final story of the book, I recount meeting Mike Stonebraker for the first time in 1974. I had not realized until I reflected for this story that the 1974 pre-SIGMOD conference that hosted the much anticipated CODASYL-Relational debate marked a changing of the database guard: not just shifting database research leadership from the creators of navigational DBMSs to new leaders, such as Mike Stonebraker, but also foreshadowing the decline of navigational database technology and the rise of the relational and subsequent eras of database technology.

Part 9, Seminal Works of Michael Stonebraker and his Collaborators, reprints the six papers that together with the 2014 ACM A.M. Turing Award Paper in Part 1, constitute the papers that present Mike's most significant technical achievements. Like most of the stories in this book, these seminal works should be read in the context of the technology and challenges at the time of their publication. That context is exactly what the corresponding research and systems stories in Part 7 provide. Until now, those papers lacked that context, now given by contributors who were central to those contributions and told from the perspective of 2018.

Mike's seven seminal papers were chosen from the approximately 340 that he has authored or co-authored in his career. Mike's publications are listed chronologically and grouped by year in **Collected Works of Michael Stonebraker**.

Additional References

[1] Cafarella, M., & Ré, C. The last decade of database research and its blindingly bright future. or Database Research: A love song. DAWN Project, Stanford University. April 11, 2018. https://dawn.cs.stanford.edu/2018/04/11/db-chommunity/