CHAPTER 1

The Roots

The first five papers in this book represent the roots of relational database systems. Of course, Ted Codd gets most of the credit for focusing attention on the relational model of data with his pioneering paper in *CACM* in June 1970, which we have chosen to be the kickoff article in this volume. This paper started a heated controversy in all ACM SIGFIDET (now SIGMOD) meetings from 1971 onward between two groups of people. On one side were members of the COBOL/CODASYL camp who had recently written their proposal for a standard network database system. This document, referred to as the "DBTG Report" [DBTG71], suggested a network data model containing records and sets with a syntax closely aligned with COBOL and a low-level navigational interface. Perhaps the best nontechnical exposition of the point of view of the DBTG camp was the 1973 Turing Award paper [BACH73] written by Charlie Bachman, the main developer of IDS, a Honeywell DBMS written in the 1960s from which the DBTG Report borrowed heavily. On the other side were Ted Codd and virtually all academic researchers lauding the merits of the relational model.

The positions of the two camps divided approximately along the following lines.

COBOL/CODASYL camp:

1. The relational model is too mathematical. No mere mortal programmer will be able to understand your newfangled languages.
2. Even if you can get programmers to learn your new languages, you won't be able to build an efficient implementation of them.
3. On-line transaction processing applications want to do record-oriented operations.

Relational camp:

1. Nothing as complicated as the DBTG proposal can possibly be the right way to do data management.
2. Any set-oriented query is too hard to program using the DBTG data manipulation language.
3. The CODASYL model has no formal underpinning with which to define the semantics of the complex operations in the model.

This argument came to a head at the 1975 ACM/SIGMOD Conference in Ann Arbor, Michigan, where Ted Codd and two seconds squared off against Charlie Bachman and two seconds in what was publicized as "The Great Debate."

The debate was significant in that it highlighted yet once more that the two camps could not talk to each other in terms the other could understand. Codd gave a formal treatment of how the CODASYL people could best make the semantics of their model less arbitrary. Bachman then argued that the CODASYL model was almost no different from the relational model. Both talks and the resulting discussion left the audience more confused at the end of the debate than at the beginning.

In the latter half of the 1970s, the two camps began to understand each other and the discussion became more focused. At the same time, however, interest in CODASYL systems began to decline based, in our opinion, on two events of the late 1970s. First, easier-to-use relational languages such as QUEL [HELD75] and SQL [CHAM76] were devised to blunt the criticism that earlier relational languages (specifically, Codd's relational calculus [CODD71] and algebra [CODD72]) were too mathematical. Second, prototype relational systems began to appear and prove that implementations could be done with reasonable efficiency.

The two most widely used prototypes were System R and INGRES, and they helped shape a fair amount of the history that followed. Under the able direction of Frank King, a group of about 15 researchers at the IBM Research Laboratory in San Jose, California, built System R from about 1974 to 1978. Although it was a significant effort by a large group of people, the influence of Jim Gray, Franco Putzolu, and Irv Traiger on the lower half of the system (the RSS level) is noteworthy, whereas the RDS level shows the influence of Mort Astrahan, Don Chamberlin, and Pat Selinger. Fifty miles away, a pickup team of Berkeley students built INGRES under the direction of Mike Stonebraker and Eugene Wong from 1973 to 1977. The next two papers in this chapter present the status of these two systems in 1976. After both systems were more or less finished, excellent retrospectives were written by both groups on the good and bad points of their designs. The final two papers in this chapter present these retrospections.

We wish to use the rest of this introduction to make a collection of comments about these prototypes. The reader of the "before" and "after" versions should carefully note what each group admitted to screwing up (e.g., poor integration between the RDS and RSS in System R and the compile time structure in INGRES). The INGRES system was constructed as an interpreter because of inexperience on the part of the designers. In System R, links and images exist at the RSS level, but the RDS level does not allow users to take advantage of them. The apparent reason is that System R development was organized into two teams. The RSS team was farther ahead, and its facilities were defined first. According to Jim Gray, links were put into the RSS in case it would be required at some future time to support a network (DBTG) or hierarchical (IMS) interface. The RDS group decided later on to ignore links and images in their initial implementation because they made the query optimization problem harder.

In addition, both retrospections are less than completely candid about their failures. For example, the recovery management scheme in System R based on shadow pages was declared a failure in another paper [GRAY81]. The absence of hashing is also widely regarded as a substantial mistake. In addition, achieving good performance seemed to require wizardry in the tuning of system parameters [DEWI87]. Lastly, System R paid little attention to presentation services and gave the end user only a very primitive query capability called the user friendly interface (UFI).

On the other hand, the INGRES designers do not point out several shortcuts they took. They used the UNIX file system even though there is no way to guarantee crash recovery services in this environment. The System R designers elected to write their own file system when faced with an unusable file manager. Moreover, they chose simplistic implementations for both locking and crash recovery that were clearly naive.

The reader should also observe what facilities are discussed in the "before" paper that the "after" paper is notably silent about (e.g., triggers in System R). In addition, readers should notice the dominance of the 16-bit architecture of the PDP-11 on the INGRES system and the amount of effort that was expended to deal with it. This pain and suffering has completely vanished from the computer scene a scant 15 years later.

Another comment is the tremendous commercial significance that these systems (especially System R) have had over the years. Kapali Eswaren left the System R project to form his own company, ESVAL, which built a commercial version of System R. Later, the ESVAL code became the basis for the Hewlett-Packard ALLBASE system as well as for IDMS/SQL from Cullinet. In addition, Larry Ellison started Oracle Corporation and independently implemented the published external specifications for System R. Lastly, with some rewriting, DB2 and SQL/DS are derivatives of the original System R prototype.

On the INGRES side of the ledger, INGRES Corporation (now part of Computer Associates) and Computer Associates both commercially exploited the public domain University of California prototype. In addition, Bob Epstein left the INGRES project in 1979 to join Britton-Lee (now part of NCR), helping to build the IDM software. Then he formed a second generation relational start-up, Sybase, to focus on the transaction processing marketplace.

As a result, much of the current commercial landscape shows the influence of these systems. In general, this influence is very positive. For example, the query optimization architecture and optimization techniques of System R are generally lauded and form the basis for the algorithms in most commercial systems. The cleanliness of QUEL and the query modification algorithms for views, protection, and integrity control get good marks for INGRES.

However, some of the legacy is less exemplary. For example, because of the position of IBM, the programming-level interface to SQL will be an intergalactic standard for a very long time. SQL and its embedding are not

very elegant, and Date clearly explains in [DATE85] the language mistakes that were made. Second, neither INGRES nor System R was particularly faithful to the relational model. Both systems allow you to perpetrate the cardinal sin—to create a relation with duplicate tuples in it. Moreover, System R would carefully retain the correct number of duplicates during join processing, so a willing user could build semantics into the number of duplicate records. This was one of the "features" of the DBTG data model that relational advocates most despised. Both systems failed to implement the notion of "domains" or even primary keys. Hence, commercial systems have been slow to construct facilities in these areas.

We want to comment on two little-known facts that might have dramatically altered the events of the last several years. First, IBM initially attempted to build an SQL interface on top of IMS. This project, code-named Eagle, would have allowed DL/1 and SQL to be used interchangeably to express DBMS commands for a single database. This effort was abandoned in the late 1970s because of semantic difficulties in building an SQL-to-DL/1 translator. If Eagle had been successful, then the resulting DBMS landscape might have been significantly different. Then IBM decided to build a separate relational system. However, they still had a choice of which approach to convert into a production system. Besides System R, prototypes were available for QBE [ZLOO75] and UDL [DATE76]. Although QBE was designed as an end-user interface and would be very difficult to call from PL/1, UDL offered a number of advantages over SQL, including a clean coupling with PL/1. A very different collection of events would have unfolded if IBM had chosen to exploit one of the other competitors.

We close this chapter with another suggestion for additional reading. It appears that Ted Codd has been blessed as "the keeper of the faith" and has the individual initiative to redefine the relational model whenever appropriate. Hence, you can think of four different versions of the model:

- Version 1—defined by the 1970 CACM paper
- Version 2—defined by Codd's 1981 Turing Award paper [CODD82]
- Version 3—defined by Codd's 12 rules and scoring system [CODD85]
- Version 4—defined by Codd's book [CODD90]

The interested reader is advised to read all four and consider the evolution of the model over time.

At the current time (1997), the relational model is considered the traditional mainstream data model, and the network and hierarchical models have fallen completely from favor. Moreover, the relational model is widely criticized for its inability to meet the needs of users outside of business data processing applications. Business requirements in this area have spawned both object-oriented and object-relational DBMSs. In Chapter 6, we will consider both of these new approaches. Here, we merely observe that object-relational DBMSs are simply an extension of the relational model to better manage complex data. As such, it is an example of how the relational model has further evolved during the 1990s.

REFERENCES

[BACH73] Bachman, C., "The Programmer as Navigator," *CACM* 16(11): 635–658 (1973).

[CHAM76] Chamberlin, D., et al., "SEQUEL 2: A Unified Approach to Data Definition, Manipulation and Control," *IBM Journal of Research and Development*, 20(6): 560–575 (1976).

[CODD71] Codd, E., "A Database Sublanguage Founded on the Relational Calculus," in *Proceedings of 1971 ACM-SIGFIDET Workshop on Data Description, Access and Control*, San Diego, CA, November 1971.

[CODD72] Codd, E., "Relational Completeness of Database Sublanguages," in *Courant Computer Science Symposium 6*, Englewood Cliffs, NJ: Prentice Hall, 1972.

[CODD82] Codd, E., "Relational Database: A Practical Foundation for Productivity," *CACM* 25(2): 109–117 (1982).

[CODD85] Codd, E., "Is Your DBMS Really Relational," *Computer World*, October 14, 1985.

[CODD90] Codd, E., *The Relational Model for Database Management—Version 2*, Reading, MA: Addison-Wesley, 1990.

[DATE76] Date, C., "An Architecture for High-Level Language Database Extensions," in *Proceedings of the 1976 ACM-SIGMOD Conference on Management of Data*, San Jose, CA, June 1976.

[DATE85] Date, C., "A Critique of SQL," *SIGMOD RECORD* 14(3): 8–54 (1985).

[DBTG71] Database Task Group, "April 1971 Report" ACM, New York 1971.

[DEWI87] Dewitt, D., et al., "A Single-User Performance Evaluation of the Teradata Database Machine," MCC Technical Report DB-081-87, MCC, Austin, TX (1987).

[GRAY81] Gray, J., et al., "The Recovery Manager of the System R Database Manager," *ACM Computing Surveys* 13(2) 223–243 (1981).

[HELD75] Held, G., et al., "INGRES—A Relational Database System," in *Proceedings of 1975 National Computer Conference*, Anaheim, CA, June 1975.

[ZLOO75] Zloof, M., "Query by Example," in *Proceedings of 1975 National Computer Conference*, Anaheim, CA, June 1975.