This short paper was chosen for publication by the editors, who hope that the comments and references will be helpful to readers who desire to strengthen their background in the literature of data base systems. Beginning with early data base concepts, this paper cites a selection of key references on hierarchical, network, and relational data bases. The authors of this selection are R. Ashany, Associate Editor of the IBM Journal of Research and Development, *and Michael Adamowicz, Associate Professor, Polytechnic Institute of New York, Brooklyn, New York.*

Data base systems

Readings

It may never be known when the term *data base* was first used in its present context, but the turning point, which came in the early 1960s, is reviewed by Buchholtz in a paper on file organization and addressing.[1] That paper, which deals with the organization and addressing of files stored on Direct Access Storage Devices (DASD) primarily, calls upon punched card terminology (file, record, identifier, key, sorting, etc.) and upon electronic data processing terms (scan, addressing, sequential storage, random access storage, overflow, etc.) to express the principles and methods of storing and extracting information from direct access files.

The field of data base systems is concerned with the analysis, interpretation, organization, classification, structure, storage, updating, searching, and retrieval of information. Conventional approaches in existing data base systems have their origins in storage structures and access strategies of early digital computers. The first sequential-access and random-access storage devices required certain data structures that, in turn, motivated the development of storing, searching, and retrieving techniques that have been perpetuated and are clearly reflected in today's state of the art. These techniques were developed to solve a problem that is known as the *addressing problem*. This condition occurs whenever a file of records is stored in a data processing system and some procedure has to be devised for deciding where to store each record and how to locate each record, given its identification attribute.

A question often asked is: What was the evolutionary process from which this new field of data base technology emerged? One of the intentions of this article is to select and summarize infor-

mation that the authors have found useful from the mass of data available in the technical literature. This paper does not represent an exhaustive list, nor does it indicate that papers not included are of lesser significance. Rather, we present a limited literature survey that Ashany has used in his research.[2,3] These readings are intended as a reference for those who are working in the field of data base systems and as a guide for those who are just entering the field.

Research in data base systems has been extensive during the last two decades. Large data files stored in random access storage systems were among the major factors triggering interest in the investigation and development of new techniques for data structuring and the use of index tables. As early as 1956, Dumey[4] recognized that when names are used for identification of records in business files, some searching is required to locate the record in the storage system. In his paper, Dumey describes hash coding for randomizing addresses. About a year later, Peterson published a paper, now considered a classic, that deals with the problem of data structuring in large files.[5] Peterson indicates that the dictionary for language translation by a computer, the symbol table for an assembly program or compiler, and many other problems that are solved by table look-up, require techniques like those described in his paper. Peterson defines *open addressing* and analyzes the performance of uniform hashing.

Buchholz, in the paper mentioned above,[1] very clearly presents basic considerations of sequential and random access approaches, key-to-address-transformation techniques, and table look-up methods. It is interesting to note that the use of mathematical techniques for solving the problems of addressing and data structuring have been used by other authors. Abraham et al. published a paper in 1968 on file organization schemes based on finite geometries, which describes new schemes for organizing records with binary valued attributes.[6] During the same period, a few other papers that describe data structuring and key transformations were published. A paper by Morris, entitled "Scatter Storage Techniques," has had great impact.[7] Morris is primarily concerned with the application of scatter storage techniques such as compiler and assembler symbol tables, but, as indicated previously,[5] the same techniques can be applied to any table or file in which access is made in an unpredictable order. Morris also discusses such problems as the handling of collisions, random probing, linear probing, and direct chaining. It is safe to say that Peterson's and Morris' papers are the most cited references in publications dealing with hash addressing.

The book *Automatic Information Organization and Retrieval* by Salton,[8] published in 1968, deals with the computer processing of large information files and encompassss a large variety of useful techniques. Salton treats extensively such subjects as associations and relations among data items and the use of mathematical and statistical techniques for searching and retrieving information. The research of the problem indicates that relational analysis of the stored data items can provide useful information. Levien and Maron published a paper in 1967 in which they describe a system called the *relational data file*, which is used for the logical analysis of data.[9] Childs, in a paper published in 1968, uses a set-theory approach for data structuring and relational analysis.[10] Childs indicates that his set-theoretic data structure approach relies on set operations to do the work usually allocated to pointers or hash coding as in list structures, ring structures, associative structures, and relational files.

Minker, in his paper "Performing inferences over relational data base,"[11] presents an interesting algorithm that permits questions to be answered where the answer is implicit within the data base. The approach used by Minker, as related to developments in the field of artificial intelligence, should certainly provoke researchers and developers to investigate further this extremely important subject of inferential operations in large data base systems.

Another classic description of a relational data model is given by Codd in a paper published in 1970.[12] A model based on *n*-ary relations introduces a normal form for data base relations and the concept of a universal data sublanguage. In the same paper, certain operations on relations are discussed and applied to the problems of redundancy and consistency. The problem of removing data dependencies—such as ordering dependence, indexing dependence, and access path dependence—is also treated in great detail. In 1971 and 1972, Codd published four additional papers that explore further the relational approach. The idea of relational analysis and key transformations can also be found in the book *A Programming Language* by Iverson, published in 1962.[13] In this book, which has the basic description of the programming language APL, Iverson discusses several types of key transformations and array manipulations. He describes the elements of APL, which is a powerful language for relational analysis of data structured in array forms.

Optimal data structuring and data management are of utmost importance in achieving cost/performance specifications in large integrated information systems. In a book published in 1969, Lefkovitz deals with the problem of performance evaluation of such systems.[14] Lefkovitz suggests formulas for the estimation of update costs for different file structures, but he ignores other

significant parameters that have to be taken into consideration. Senko et al., on the other hand, in their *File Design Handbook*, published in 1969,[15] describe in great detail the relationship between hardware and software parameters and how they affect the performance of information systems. That handbook contains detailed analyses of the sequential access method, direct access method, indexed sequential access method, and secondary indexes. Basic types of data management techniques and interactions between the hardware and software parameters are presented by Dodd in a paper published in 1969.[16]

In 1969 Blier and Vorhaus described a general-purpose data management system designed to operate under the control of a time sharing executive on the IBM System/360.[17] Called the Time Shared Data Management System (TDMS), it was developed by SDC (the System Development Corporation). The data structuring and the performance of the system appear to be quite efficient for the type of queries for which the system is tailored. TDMS provides indexing for all the attributes. Another general-purpose data base system, called IDS (Integrated Data Store), was developed in the late 1960s by the General Electric Company. As described in its reference manual,[18] IDS provides a convenient method of describing complex information structures through the association of the data contents. The association of the records is achieved through the use of chains, which provide cross-reference linkages between records. IDS permits the selection of specific attributes and the incorporation of indexes into the data structures by means of additional chains. The performance of the system is improved when application programs refer to those chains by name. In the years 1968–1970, other information systems such as IMS[19] and GIS[20] were produced by IBM, where the goal of the data structuring was to provide data independence.

A paper on interactive graphics in data processing, published in 1968 by Symonds,[21] describes an approach to representing relations among entities in graphics data structures as triples in the form Attribute(Object) = Value. The paper describes an associative technique for holding a universe of triples on auxiliary storage and then accessing a triple in response to a query. Symonds' paper describes in detail the problems of storage allocation, data compression, hashing, collisions, and associative languages. Symonds also mentions some contributions made by Rovner[22] and Johnson[23] between 1966 and 1968 toward the development of an associative language. The paper published by Feldman and Rovner in 1969 describes a high-level programming language named LEAP.[24] The language is a version of ALGOL that is extended to include associations, sets, and a number of auxiliary constructs. LEAP was designed and implemented for large complex associative structures. The underlying idea for

data structuring was implemented using a hash coding technique. Crick and Symonds published in 1970 a report entitled *A Software Associative Memory for Complex Data Structures*.[25] They describe the implementation of the content addressability concept using Feldman's and Rovner's ideas.[24] Crick and Symonds indicate that the implementation of inter-entity relationships is generalized to be consistent with a formulation based on mathematical relations as discussed by Levien and Maron[9] and Codd.[12] It is worth noting that the content addressable approach to auxiliary storage as conceived by Feldman and Rovner has been implemented at the MIT Lincoln Laboratories and at Stanford University. Ash and Sibley built a system based on the same concepts to run on an IBM System/360 Model 67.[26]

The wide interest in generalized data base management systems, and the development of such systems by several companies in the late 1960s, revealed a need for standardization. In May 1969, the CODASYL System Committee completed a report entitled *A Survey of Generalized Data Base Management Systems*.[27] A goal of the committee was to develop the specifications of a standard data base language and functions for a unified data base system. The final objective was to submit the common system to the American National Standards Institute (ANSI) as a candidate for standardization following a path similar to that of COBOL, which had been developed by CODASYL between 1959 and 1961. An introduction to the main CODASYL data base report, published separately in 1971,[28] analyzes the features of generalized data base management systems and describes technical problems that face future designers. It discusses such problems as handling existing stored data and providing more complex data structures than those already available in conventional programming languages. The paper deals with problems of data independence and binding, and it discusses differences and similarities between capabilities in host language systems and in self-contained systems. This CODASYL data base introduction presents a good exposition of problems to be solved through careful research.

McGee presents a useful approach to the specification of file-level operations on network data structures.[29] The author uses a logical network data structure class similar to that developed by the CODASYL Data Description Language Committee. He demonstrates that the advantages of using file-level operations are not limited to hierarchical structures and indicates how they can be extended to network structures.

In the area of Key to Address Transformation (KAT), a paper by Lum et al., published in 1971, presents the results of a study of eight different KAT methods as applied to a set of existing files.[30] The performance of each method is summarized in terms

of the number of accesses required to fetch a record and the number of overflows created by each transformation, when the load factor and bucket size are varied over a wide range. Wong and Chiang, in a paper published in 1971,[31] treat KAT techniques for data structuring in a system that allows queries involving arbitrary Boolean functions of properties to be processed without taking intersections of lists. That method seems to improve performance. Another paper that deals with KAT techniques and their performance was published in 1972 by van der Pool.[32] His analysis is based on the assumption that, for a given set of keys, a transformation exists that gives a uniform probability distribution over the available addresses. He derives formulas in which the costs are expressed as functions of quantity of storage used, number of accesses, cost per unit storage, and cost per access. Van der Pool published another paper in 1973 in which file organizations using KAT and open addressing are studied through the use of simulation and a Markov model.[33] The results obtained are compared with the results reported by Peterson.[5]

A comprehensive study of data structuring and accessing methods in large data base systems was completed and published in 1973 by Senko et al.[34] The authors present a descriptive analysis of data base informations systems. The paper reviews the evolution of data base systems, discusses the structuring of information, and introduces a new fundamental approach to data structuring. In the same paper, the authors introduce a Data Independence Accessing Model (DIAM) for describing information and its stored representations. Although the model is rather complex and abstract, DIAM has some useful features.

Knuth, in his book *The Art of Computer Programming—Fundamental Algorithms*, published in 1968,[35] devotes a chapter to information structures in which he presents in great detail structures as they may appear inside the computer. Knuth's approach is a good introduction to the notions of physical data structures. Another book relevant to large integrated information systems is Knuth's *The Art of Computer Programming—Sorting and Searching*, published in 1973.[36] His chapter on searching is the most comprehensive study in this area known to the present authors. In a section on hashing, Knuth elaborates descriptions of the several techniques and evaluates their performance. In a paper published by Lum in 1973,[37] a new approach to general performance analysis of KAT methods is presented, and the results obtained in an earlier experiment by Lum[28] are substantiated and explained analytically.

Shneiderman and Scheuermann, in a paper entitled "Structured data structures," published in 1974,[38] proposed a Data Structure Description and Manipulation Language (DSMDL) which provides for the creation of a restricted class of data structures, but

ensures the correctness and compatibility of existing programs. The four papers on the topic of Data Base Management (DMB) that were published in *Datamation* in September 1974[39] succinctly describe the problems that have to be solved in the area of large integrated data bases.

Relational data bases have been in existence for a long time. A relational data base is any data base that is represented in tabular form and that enables one to perform set operations (regardless of the means used: matrix manipulation, relational calculus, algebraic operations, etc.). Many data bases have used tabular forms—for decision making, in graph theory, in network analysis—that were not termed relational data bases. As stated previously, Levien and Maron[9] used the name *relational data file*, but Codd[12] has to be credited with the wide acceptance of the term *relational data base*. There has developed over the past several years a considerable literature on relational data bases, but many of the concepts described have not gone beyond academic interest. In a paper presented at the 1974 IFIPS Congress in Stockholm, Codd discussed some aspects of most urgently needed investigations.[40] He discussed the paramount significance of ascertaining the performance that is attainable when the relational approach is applied to a large-scale data base with concurrent access. The development of storage, access, and modification theory for collections of nonhierarchical *n*-ary relations is another aspect of considerable significance.

Many researchers and designers of relational data base systems have investigated specialized aspects of such systems. Wang and Wedekind published a paper in 1975 entitled "Segment synthesis in logical data base design"[41] in which they indicate that, traditionally, logical segments in a data base are defined on an ad hoc basis, intermixing logical representations and other performance-oriented considerations. The paper describes how, after the removal of redundant relations and the reduction of the number of relations, one can derive an optimal set from the original set of functional relations. The authors use the notion of minimal cover, in which a closure can be derived from a minimal set of relations. Bernstein analyzes similar approaches in his PhD thesis, *Normalization and Functional Dependencies in the Relational Data Base Model* (University of Toronto, 1975).[42]

In a paper entitled "Further normalization of the data base relational model," Codd[43] discusses in great detail the reasons for normalization and provides insight into the intricacies of functional dependencies that exist among relations in a data base system. This paper is recommended as a clear presentation of the normalization process and other relational data base aspects.

An excellent book by Date, *An Introduction to Database Systems*, was published in 1975.[44] The author analyzes and compares, in great detail, the concepts of the relational approach, the hierarchical approach, and the network approach. Date indicates that, despite his bias in favor of the relational approach, the hierarchical and network approaches are extremely important, and they possess the significant advantage that implementations exist. The hierarchical system analyzed by Date is the IBM Information Management System (IMS).[19] His analysis of the network approach is based on the proposals of the Data Base Task Group (DBTG).[27, 28] Some modest implementations of the relational approach have emerged in the last few years, but they are still experimental.[45, 46, 47]

Another excellent book, *Computer Data-Base Organization* by Martin, was published in 1975.[48] Martin introduces basic concepts and definitions, presents design criteria, and explains some of the difficulties encountered in the development of data base systems. A thorough survey of data base research and development activities was published by Blaser and Schmutz in 1975.[49] The survey includes a bibliography that contains 198 entries, and it draws conclusions with respect to established and potentially emerging principles in data base architecture and design, as well as potential future trends in data base research.

Concluding remarks

In general, it can be said that there exists a large body of technical literature that covers a wide spectrum of problems related to the new and very important field of data base systems. There is no doubt that the relational data base approach provides a scientific foundation and a sound engineering discipline. In addition, it provides a theory for data analysis and data structuring as well as a badly needed formalism in this highly significant area of computer science. A considerable literature has been generated by the very fact that the relational approach provides a scientific discipline in a field that has been governed by a haphazard approach for a long time. However, too many papers present concepts that may never go beyond the realm of academic interest. Despite the existence of some small-scale (experimental) implementations, additional evidence is needed to indicate the practicality and feasibility of such an approach in a large-scale data base environment.

CITED REFERENCES

1. W. Buchholtz, "File organization and addressing," *IBM Systems Journal* **2**, 86–111 (June 1963).

2. Taken from the dissertation by R. Ashany, *SPARCOM: A Sparse Matrix Associative Relational Approach to Dynamic Data Structuring and Data Retrieval*, submitted to the Faculty of the Polytechnic Institute of New York in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Electrical Engineering) June 1976.
3. R. Ashany, *Concepts of Data Manipulation—The Connection Matrix Method*, IBM Technical Report T.R. 00.2200, IBM Corporation, Poughkeepsie, New York (June 1971).
4. A. L. Dumey, "Indexing for rapid random-access memory, *Computers and Automation* **5**, 12, 6–8 (December 1956).
5. W. W. Peterson, "Addressing for random-access storage," *IBM Journal of Research and Development* **1**, 2, 130–146 (April 1957).
6. C. T. Abraham et al., "File organization schemes based on finite geometries," *Information and Control* **12**, 2, 143–163 (February 1968).
7. R. Morris, "Scatter storage techniques," *Communications of the ACM* **11**, 1, 35–38 (January 1968).
8. G. Salton, *Automatic Information Organization and Retrieval*, McGraw-Hill, New York, (1968).
9. R. E. Levien and M. E. Maron, "A computer system for inference execution and data retrieval," *Communications of the ACM* **10**, 11, 715–721 (November 1967).
10. D. L. Childs, "Description of set-theoretic data structure," *Proceedings of the Fall Joint Computer Conference* **33**, 557–564, Thompson, Washington (1968).
11. J. Minker, "Performing inferences over relational data bases," *ACM SIGMOD International Conference on Management of Data*, 79–91 (May 1975).
12. E. F. Codd, "A relational model for data for large shared data banks," *Communications of the ACM* **13**, 6, 377–387 (June 1970).
13. K. E. Iverson, *A Programming Language*, John Wiley, New York, (1962).
14. D. Lefkovitz, *File Structures for On-Line Systems*, Spartan Press, New York (1969).
15. M. E. Senko et al., *File Design Handbook*, Information Sciences Department, IBM Research Laboratory, San Jose, California (November 1969).
16. G. D. Dodd, "Elements of data management systems," *Computing Surveys* **1**, 2, 117–133 (June 1969).
17. R . E. Bleier and A. H. Vorhaus, "File organization in the SDC Time-Shared Data Management System (TSMS)," *Proceedings of the IFIP Congress*, 1968, 1245–1252, North Holland, Amsterdam (1969).
18. *IDS Reference Manual* GE625/635, GE Information Systems Division, Phoenix, Arizona, CPB1093B (February 1968).
19. *Information Management System IMS/360*, Application Description Manual GH20-0765-1, IBM Corporation, White Plains, New York (1971).
20. *Generalized Information System GIS/360*, Application Description Manual GH20-0892-0, IBM Corporation, White Plains, New York (1970).
21. A. J. Symonds, "Auxiliary-storage associative data structure for PL/I," *IBM Systems Journal* **7**, 3 and 4, 229–245 (1968).
22. P. D. Rovner, *The LEAP Users Manual*, Massachusetts Institute of Technology, Lincoln Laboratory, Lexington, Massachusetts (December 1968).
23. T. E. Johnson, *Mass Storage Relational Data Structure for Computer Graphics and Other Arbitrary Data Stores*, Massachusetts Institute of Technology, Department of Architecture Report, Cambridge, Massachusetts (1967).
24. J. A. Feldman and P. D. Rovner, "An Algol-based associative language," *Communications of the ACM* **12**, 8 (August 1969).
25. M. F. C. Crick and A. J. Symonds, *A software associative memory for complex data structures*, IBM Cambridge Scientific Center Report G320-2060, Cambridge, Massachusetts.

26. W. Ash and E. Sibley, "TRAMP, an interpretive associative processor with deductive capabilities," *Proceedings of the ACM 23rd National Conference*, 143–156, Brandon Systems Press, Princeton, New Jersey (1968).
27. *A Survey of Generalized Data Base Management Systems*, CODASYL Systems Committee. Available from the Association for Computing Machinery, 1133 Avenue of the Americas, New York.
28. "Introduction to feature analysis of generalized data base management systems," CODASYL Systems Committee, *Communications of the ACM* **14**, 5, 308–318 (May 1971).
29. W. C. McGee, "File-level operations on network data structures," *ACM SIGMOD International Conference on Management of Data*, 32–47 (May 1975).
30. V. Y. Lum et al., "Key-to-address transform techniques: a fundamental performance study on large existing formatted files," *Communications of the ACM* **14**, 4, 228–239 (April 1971).
31. E. Wong and T. C. Chiang, "Canonical structure in attribute based file organization," *Communications of the ACM* **14**, 9, 593–597 (September 1971).
32. J. A. van der Pool, "Optimum storage allocation for initial loading of a file," *IBM Journal of Research and Development* **16**, 6, 579–586 (November 1972).
33. J. A. van der Pool, "Optimum storage allocation for a file with open addressing," *IBM Journal of Research and Development* **17**, 2, 106–114 (March 1973).
34. M. E. Senko, et al., "Data structure and accessing in data base systems," *IBM Systems Journal* **12**, 1, 30–94 (1973).
35. D. E. Knuth, *The Art of Computer Programming—Fundamental Algorithms*, Addison-Wesley, Reading, Massachusetts (1973).
36. D. E. Knuth, *The Art of Computer Programming—Sorting and Searching*, Addison-Wesley, Reading, Massachusetts (1973).
37. V. Y. Lum, "General performance analysis of key-to-address transformation methods using an abstract file concept," *Communications of the ACM* **16**, 10, 603–612 (October 1973).
38. B. Shneiderman and P. Scheuerman, "Structured data structures," *Communications of the ACM* **17**, 10, 566–574 (October 1974).
39. R. F. Schubert, et al., "Data base management," *Datamation* 49–65 (September 1974).
40. E. F. Codd, "Recent investigations in relational data base systems," *Information Processing 74*, North Holland, Amsterdam (1974).
41. C. P. Wang and H. H. Wedekind, "Segment synthesis in logical data base design," *IBM Journal of Research and Development* **19**, 1, 71–76 (January 1975).
42. P. A. Bernstein, *Normalization and Functional Dependencies in the Relational Data Base Model*, Technical Report (PhD Dissertation) CSRG-60, Computer Systems Research Group, University of Toronto (October 1975).
43. E. F. Codd, "Further normalization of the data base relational model," *Data Base Systems*, Current Computer Science Symposia Series, Vol. 6, Prentice-Hall, New York (1972).
44. C. J. Date, *An Introduction to Database Systems*, Addison-Wesley, Reading, Massachusetts (1975).
45. G. Bracchi et al., "A multilevel relational model for data base management systems," *Proceedings of the IFIPS Workshop and Conference*, (Cargese, Corsica) North Holland, Amsterdam (April 1974).
46. N. McDonald, M. Stonebreaker, and E. Wong, *Preliminary Design of INGRES: Part I*, Electronics Research Laboratory, ERL-M435, University of California, Berkeley (April 10, 1974).
47. D. J. McLeod and M. J. Meldman, "RISS: a generalized minicomputer relational data base management system," *Proceedings of the National Computer Conference*, Anaheim, California (1975).

48. J. Martin, *Computer Data-Base Organization*, Prentice-Hall, Englewood Cliffs, New Jersey (1975).
49. A. Blaser and H. Schmutz, *Data Base Research: A Survey*, IBM Technical Report TR75.10.009, Heidelberg Scientific Center, Germany (November 1975).

Reprint Order No. G321-5036.