Book review

A Guide to DB2, C. J. Date, Addison-Wesley Publishing Company, Reading, MA, 1984. 312 pp. (ISBN 0-201-11317-1).

This is a well-organized, clearly presented introductory text on IBM Database 2 (DB2), IBM's recently announced relational data base management system designed for those users running applications under the MVS operating system. (For additional material on DB2 see Volume 23, Number 2, 1984 of the *IBM Systems Journal*.) It is also a useful guide to the SQL language. As such, the book will be helpful to SQL/DS users, as well as users of DB2. In addition, the book serves well as an introduction to the relational data base *approach*, since it describes what is clearly a state-of-the-art data base management system based on the relational concepts and theory of E. F. Codd. It also contains an appendix that introduces the reader to the basic theory of relational data base and another that contains guidelines for the design of relational data bases that can be processed by any relational data base management system, including, of course, DB2.

Incidentally, the author provides an unequivocal answer to the often-asked question: What are the essential characteristics of a relational data base management system? His answer, simply, is that a relational system is one in which (1) the user perceives the data base as a collection of tables and (2) the language supports the select, project, and join operators as they were originally defined by Codd. Date notes that the join operation must not require predefined access paths.

The book begins with an overview that shows the application processing options available to DB2 users, depending upon which of three transaction managers is supporting their applications, *IMS*, *CICS*, or *TSO*. It is noteworthy that IMS or CICS users can mix requests for data base services, incorporating in a single program both DL/I requests against data stored in an IMS hierarchical data base and SQL requests against data stored in a DB2 relational data base. Transactions can be processed on line with any of these transaction managers. Batch processing can be performed as an IMS or TSO application. TSO users can process a DB2 data base using SQL in either batch or on-line mode. They can also take advantage of DB2 Interactive (DB2I) to execute SQL statements and to prepare and invoke application programs and system utilities. A related product for TSO users, the Query Management Facility (QMF), processes queries or update requests made in either SQL or the Query-By-Example (QBE) language. QMF also includes a comprehensive report writer. Data Extract (DXT), another related product, extracts data from DL/I data bases or from SAM or VSAM files and converts them to a format suitable for loading by the DB2 load utility.

Four chapters of this guide introduce the reader to the SQL language as it can be used from a terminal to perform data definition, query, or update processing. Three additional chapters show how the same language can be embedded in a host language (PL/I, COBOL, FORTRAN, or Assembler) to perform batch or on-line transaction processing. All of the tutorial material on the SQL language is presented using examples that are representative of what a broad range of users, from the novice to the sophisticated, might actually want to do. The author introduces SQL at a beginner's level and graduates carefully to more complex examples.

Readers should not be put off by the apparent difficulty of some of the examples, either those given in the text or those given in the well-conceived exercises. Although they might tax the beginner in SQL, time spent on these query and update examples is time well spent.

As Date points out, it would be a mistake to suppose that the complexity of some of these examples is an inherent characteristic of the language. It is important to notice how much these complicated queries are asking of the system. Usually the complexity is

not attributable to properties of the language, but is due to the richness of the information being provided. I suggest that a simple test of this point can be made by trying to express the same queries and updates using a nonrelational data base language, including nonrelational query languages.

A key feature of a relational data base system is the ability to generate views of the data base. Views are virtual tables; they do not contain real data. They serve several useful purposes. First they can be used to simplify queries or updates. Let us suppose that there is no single table at present in our data base that allows us to formulate, in a convenient way, a particular query we have in mind. Also suppose that the data required for this query already exist in various rows and columns of two of our data base tables, Table A and Table B. (In the jargon of relational data bases, Tables A and B are base tables; that is, they contain real data.) What we would like to do, then, is to "cut and paste" the appropriate rows and columns from Tables A and B to create a new table, Table C. The view mechanism enables us to create a virtual Table C in a simple and direct way. Once this is done we can proceed with our query formulation *as if* there were a real table whose organization and content correspond to that of Table C. Views can be defined and made available for use dynamically. That is, they can be defined and used in the same transaction.

Views are created *without generating redundant data* in Table C. Thus it is best to think of views as "windows" on the base tables on which they are defined.

Of course, this mechanism can also be used to prevent unauthorized use of data, and it can be used to enhance data independence. Date gives the reader instructive examples of each such use of the view mechanism in DB2.

It is fairly well known at this point that although updates to base tables are automatically reflected in any view defined on those tables, the converse is not true. Thus, in some instances, updates to a view cannot be propagated to the base tables by the system. Date's chapter on views spells out under what circumstances such updates will and will not work in DB2. Moreover, Date explains clearly and in detail the rationale for the constraints on such updates. I believe this is the most detailed treatment in the literature of the constraints on view updating that exist in current relational data base systems.

One of the merits of each of Date's books on data base is his willingness to tackle topics and problems that are unusually difficult to explain to the beginner. This book is no exception. For example, SQL does not have a single expression that is the equivalent of the division operator of the relational algebra as defined by Codd. Nonetheless, as we would expect, in SQL there are alternate, indirect ways of expressing this operation. Although these alternates are not easy to explain, Date tackles and performs the job nicely.

I take this opportunity to express my own conviction on how data base languages based on the relational calculus, e.g., SQL, QUEL, and QBE, might be taught to those users who would like to exploit fully their power and richness by becoming fluent in all facets of such languages. I have found that instruction in the notation of the very basic elements of first-order logic along with a brief tutorial on Codd's ALPHA language serves as *an* effective way of introducing these languages to the class of users to which I refer. Note that I do not maintain that such instruction is a necessary requirement for learning SQL, QBE, or QUEL at this level. Nor is this suggestion intended to conflict with my judgment that such languages are much easier to learn and to use, for all levels of users, than nonrelational data base languages.

This book serves well to demonstrate the comprehensive range of function provided by DB2 and its associated languages and products. It also offers very persuasive evidence of the ease with which these functions can be exercised in DB2—and the felicity with which complex information requirements can be specified within the framework of the relational data base approach.

Charles J. Bontempo
IBM Systems Research Institute
New York, New York

Note—The Editor assigns reviews of books that might interest our readers. Reviews are signed, and opinions are those of the reviewers.