The Expert's Opinion

Guy L'Heureux is database administrator at Cessna Aircraft Company working in an IMS and DB2 environment. Guy's over 15 years of data processing experience has included appointment as Assistant Director of Technical Services at Cap Gemini America.

Interview by Mohammad Dadashzadeh

JDA: What is the role and what are the major responsibilities of the Database Administrator at Cessna?

L'Heureux: Data modeling, database performance, database design and implementation, database security (through the program specification block), and database maintenance (tuning, reorganizations, etc.).

JDA: What do you consider to be the most important contributions of the DBA to an organization?

L'Heureux: Naturally, that depends on the company. I feel that it is important to have the foresight to plan for what is going to happen one month down the road, as well as five years down the road. You have to be constantly aware of the business needs of the company, and steer your decisions (or recommendations) in that direction. Other contributions would be meeting users' information requirements, database availability and stability, and database performance.

JDA: What is the computing environment at Cessna? Is computing decentralized? Is development decentralized?

L'Heureux: There are two major systems for the financial and business applications. The system that the database systems are on has DB2 and IMS DB/DC. The other system has CICS/VSAM and TOTAL. I would say that currently, the majority of the computing and development are centralized.

JDA: What are some of the major computer-based information systems at Cessna? What are the volumes of transactions? Are they implemented using a DBMS?

L'Heureux: Currently, our manufacturing system in running under IMS DB/DC. This system has over 1100 transactions defined, and it has 93 databases spanning 12 IBM 3390 disk drives. We are also moving towards implementing a paperless environment for manufacturing that uses DB2 tables with IMS/DC as the transaction processor. It has 31 DB2 tables using approximately 250 cylinders of DASD. Our plans for next year (1991) include integration of financial applications (Accounts Receivable, Billing, Accounts Payable, General Ledger, etc.) that will add roughly 65 new DL/I databases, and about 250 new IMS/DC transactions.

JDA: What is the reporting structure of the DBA and the MIS function as a whole at Cessna?

L'Heureux: The database administrators report to the manager of information systems planning, who reports to the director of information management.

JDA: Does Cessna have an integrated database? Are there any plans to move in that direction?

L'Heureux: Currently our engineering information and our manufacturing information are almost fully integrated. The direction that I would like to go is to integrate as much as feasible. There are some plans for 1991 and 1992 to integrate more of the engineering data, and also to integrate some portion of the financial components.

JDA: What tools would you consider or have at your disposal if you were to integrate the separate databases?

L'Heureux: None at the present time.

JDA: Is Cessna planning for migration to a relational DBMS such as DB2?

L'Heureux: We have DB2 in use for some applications, however we are not planning to migrate existing systems to DB2 in the foreseeable future. We are using DB2 on some new development work, and may consider migration of TOTAL databases to DB2 in the future.

JDA: Do you consider the need to migrate to a relational DBMS as an opportune time to push for an organization-wide integrated database?

L'Heureux: We already have in existence some organization-wide databases, and are already planning for more. However, relational DBMS is not regarded as a prerequi-

site for an organization-wide (global) database.

JDA: Do you consider the concept of an organization-wide integrated database a mirage?

L'Heureux: Absolutely NOT. I believe that it is very real, and that it is the direction that a company should be positioned for.

JDA: What tools do you like to see become available for your database environment?

L'Heureux: A tool that would extract records from a database and create a smaller test database based on control cards. It would have to have the ability to extract records and their related components (those defined through implicit semantic integrity relationships, or extremely complex IMS-type logical relationships).

JDA: What features of your DBMS do you most appreciate?

L'Heureux: With DL/I, it has to be that it has been proven over time. With DB2, it has to be its ease of use.

JDA: What features of your DBMS do you consider most annoying?

L'Heureux: With DL/I, it is cumbersome to use. With DB2, it is the way the optimizer will choose an access path. At times. you have a difficult time making a program behave the same way in production as it behaved in test. At least with IMS, you can control the physical path that will be used to access the data.

JDA: What do you consider to be the most important roles of a Data Dictionary System in your environment?

L'Heureux: We currently use a manual system. We have several systems under consideration for an automated data dictionary. The greatest benefit will come from standardization through the common use of items, labels and data elements.

JDA: What methodology or diagrammatic technique do you use for logical database design?

L'Heureux: We are currently using Yourdon Structured Methodology. By the time we begin designing a database, the Entity-Relationship diagrams, dataflow diagrams and entity state transition diagrams (along with a data dictionary summary of the data elements) have been done. When the design is completed, a data structure diagram is provided.

JDA: How do you deal with specification and enforcement of integrity constraints?

L'Heureux: Currently, it is handled by applications. For example, none of the applications that are currently in-house, or being planned for the near future are having the DBMS enforce referential integrity automatically. If a field must relate to a record in another database or table, then the application program will take care of checking to see that the appropriate value exists.

JDA: What do you see as future trends and issues in database management?

L'Heureux: First trend is distributed processing and cooperative processing. I feel that over the next few years this will be the up-and-coming issue. Second trend is the organization-wide integrated (global) database. This will catch on as distributed processing catches on. Corporate data will live in one place for everyone with a need to know to share, while allowing extract databases that are used by decision support applications to live in another place.

JDA: What would be your advice to your colleagues to help them prepare or deal with challenges of database administration over the next decade?

L'Heureux: Remain open to change. The industry is going to encompass a lot of change. Be aware of how you want to be positioned in a long range plan. Then, try and make decisions and recommendations not only based on today, but also tomorrow.

Software Review

Review by Mohammad Dadashzadeh
The Wichita State University

Alpha FOUR, Version 1.1b
Alpha Software Corporation
One North Avenue
Burlington, MA 01803
(617) 229-2924
List Price: $549.00

Although advertised as The Fully *Relational* Database for Non-Programmers, Alpha FOUR's strengths are neither in its relational capabilities nor in its lack of a programming language interface. Instead, its native dBASE III Plus file formats, combined with its ease of learning and use, along with its powerful report generation capabilities may prove to be its strongest assets in attracting dBASE III Plus users still undecided about the right upgrade path.

Database Definition. Alpha FOUR provides a menu driven interface for defining the base tables in the database. It supports character, numeric, date, Boolean, and memo (long text) data types. In addition, through its facility called *field rules* the domain of each column may be further restricted. Specifically, default, maximum and minimum values, or a list of possible values may be specified for each column. In effect, field rules allow for limited specification of integrity constraints in Alpha FOUR. Field rules can be used to validate the entry in a field by defining a logical expression that tests whether the entered value meets specific requirements. They can also be used to test for the existence of the entered value in columns of another table, thereby implementing referential integrity.

Field rules are also used to implement calculated fields, automatically perform case conversion, design templates to format fields with special characters, define field masks to further restrict the type of data allowed, automatically increment a field's value during data entry or update, require certain fields to be entered before the record is saved, require a field to be entered twice to ensure accuracy, skip specific fields during data entry if certain conditions are met, and automatically fill in the fields in a record with values from a lookup table or another relation.

Alpha FOUR relies heavily on indices for data retrieval in addition to requiring them for such functions as maintaining primary key integrity, looking up tables, and linking relations. As such, the specification of (multi-column) indices is considered an integral part of Alpha FOUR's database definition. And, up to 7 indices per table will automatically be maintained by Alpha FOUR.

Finally, because Alpha FOUR's data retrieval functions are essentially based on a single table, its sets are perhaps the most important aspect of its database definition capabilities. An Alpha FOUR set is fundamentally the join of two or more base relations. Similar to views in SQL, set definitions are maintained but the corresponding tables are only materialized in response to such data retrieval operations on them as browsing or report generation. Alpha FOUR allows set definitions to include other sets but places a limit of 10 on the total number of base relations that may be joined in a set.

Data Manipulation. Two salient features of Alpha FOUR's data manipulation capabilities are the absence of a programming language interface and the lack of a conventional relational query/update language (such as relational algebra, QBE, or SQL). Of course, Alpha FOUR supports, as utility functions, the relational algebra operations of join, set difference, and set intersection which along with its standard support for performing selections and projections on a table accord it the legitimate claim to relational completeness. However, its relational algebra is hardly intended to be its everyday medium of discourse.

Alpha FOUR's approach to data manipulation is menu driven and centered on browsing through (selected) records from base tables and Alpha FOUR's sets (joined tables). Alpha FOUR supports complex search expressions (filters) which may employ any of its over 50 built-in functions and can be saved for reuse.

The principal strength of Alpha FOUR's data retrieval capabilities rests on its sophisticated yet easy to use output layout specifications. Customized output layouts may be specified for browsing, data entry forms, reports, form letters, and mailing labels. Alpha FOUR allows a great deal of flexibility in layout specifications ranging from specifying the display colors of specific fields, to printing long text fields in multi-line windows with word wrapping, to specific support for printing checks, to conditional printing of paragraphs in form letters, to support for calculated fields such as percentage of total that would require multiple passes through the selected records.

Alpha FOUR's global update/delete command allows updating of up to 20 fields, or marking for deletion all, or a subset of, the rows in a base table.

Alpha FOUR provides good support for application generation through creation of password-protected customized menus. Supported menu actions are: invoking a sub-menu, executing a keystroke macro, calling another application, or transferring to another application. Although the macro language allows for variables, it does not supports looping or conditional constructs and thus is limited to replicating rote procedures.

Finally, Alpha FOUR's data dictionary facilities are limited to displaying the structure of the current base table or set. An external utility is provided to print documentation for applications, listing field rules, and showing the structure of tables and Alpha FOUR's sets.

Schema Modification. Alpha FOUR's RECONFIGURE command is used for schema modification. One can add or delete columns (fields), change the order of the columns, and change column names as well as data types. Re-configuring a table is treated as creating a new table with Alpha FOUR automatically copying records from the original table to the modified one. Alpha FOUR does a fair job in automatically handling the side effects of typical schema modifications. For example, when a column is deleted, Alpha FOUR will invalidate the indices based on, and the expressions (e.g., field rules) involving, the deleted field. It would also remove the field from any mailing label or report layouts on which it had been placed. Alpha FOUR also permits modification of field rules (i.e., integrity constraints) and supplies a command to re-apply the rules to each row in the table and marking for deletion the records violating the modified rules.

Transaction Processing and Concurrency Control. Alpha FOUR does not support the transaction concept. Integrity constraints are verified at the end of each atomic update. And, as a single user DBMS it does not require any mechanisms for concurrency control.

Backup and Recovery from Failure. Alpha FOUR does not buffer updates in memory and writes all updates immediately to disk. However, it does not include any utilities to recover from a failure in the middle of a global (multi-record) update operation. (An Alpha FOUR command provides for re-building indices in case of failure.)

Security and Authorization. Alpha FOUR does not include any mechanism for access control. However, an Alpha FOUR turnkey application can provide password protection.

Utilities. Alpha FOUR's COPY command may be used to materialize the projection or selection of a base table or set, and its SUMMARIZE command serves to perform a grouping operation similar to SQL's GROUP BY clause. Aggregate operations, called summary options, supported by the SUMMARIZE command include Count, Biggest, Smallest, Total, and Average.

Alpha FOUR uses the dBASE III Plus file formats directly and can import and export data in Lotus 1-2-3, delimited ASCII, as well as fixed length ASCII formats. In addition, it can also export data in WordPerfect secondary merge file format.

User Interface. Alpha FOUR's user interface is simple and easy to learn. Its logical menu structure, context sensitive help, and good documentation (including an informative tutorial and an audio cassette tour) accelerate learning and effective use of the package.

In summary, although not quite in the same league as full-featured relational database management systems such as R:BASE 3.1, Alpha FOUR's ease of use and powerful report generation combined with its native dBASE III Plus file formats make it worthy of careful consideration especially by organizations still undecided about the right upgrade path from dBASE III Plus.

Book Review

Reviews by Mohammad Dadashzadeh
The Wichita State University

Database: Step-by-Step, Second Edition
Mark L. Gillenson
John Wiley & Sons
1990, 394 pp.
ISBN 0-471-61759-8

Writing an introductory book on database management concepts that would be at once accessible to data processing professionals, managers, computer users, and students of undergraduate and graduate courses in DBMS is not a simple task. Gillenson does an admirable job in meeting this challenge. He has produced a book that in large part accomplishes his objective of explaining "the *PRINCIPLES* behind the database and the broad range of database subtopics *IN A MANNER THAT LEAVES NOTHING TO THE IMAGINATION*." As such, he approaches the subject in a progressive, step-by-step manner starting with the origins of record keeping and culminating with such evolving topics as object-oriented and expert database systems. In between, the reader is presented with chapters covering file organizations and access methods, the database approach, database management systems characteristics, the relational approach, SQL, QBE, dBASE, the hierarchical and network models, logical database design, data dictionaries, database control (security, concurrency, and backup and recovery), and data administration.

The many virtues of Database: Step-by-Step include a chapter on IBM AS/400 database environment, a brief discussion of the Date-Wilson database design technique, and the very readable narrative writing style. Its basic shortcoming—lack of detailed coverage in some areas—is in fact an intentional product of its design. As acknowledged in the preface, in an attempt "to cut through the maze of database detail and yet hit hard at the basic concepts" while providing a "user-friendly introduction to the entire field of database" for a rather wide audience, "this book may generalize or oversimplify a few points." Overall, I tend to concur with the author that this design tradeoff is worth it.

An Introduction to Structured Query Language
Ron McFadyen and Vijay Kanabar
Wm.C. Brown Publishers
1991, 214 pp.
ISBN 0-697-11853-3

There are several features that make this book a very good choice as a supplementary textbook on SQL for introductory courses in database management systems. Those same features help make the book quite appropriate for information systems professionals who wish to obtain a quick understanding of SQL and SQL-based environments.

First, there is the complete and accurate coverage of ANSI SQL supported by a running example. Second, there is a concise chapter on embedded SQL, including Dynamic SQL, featuring example programs in COBOL and PL/I. Third, there are chapters covering specific features and extensions of ORACLE, dBASE IV, and DB2's SQL-based environments. In particular, readers familiar with dBASE may find the presentation of examples in both dBASE IV SQL and the traditional dBASE language especially helpful.

In short, McFadyen and Kanabar have produced a well-written book that covers SQL in greater depth than is typically found in available DBMS textbooks. As such, it should serve well as a supplementary textbook in introductory database management system courses.