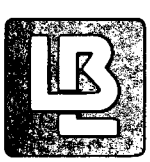

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

Computing Division

REQUIREMENTS FOR A DATABASE MANAGEMENT SYSTEM

J.D. Lawrence and J. McCarthy

September 1984

Prepared for the U.S. Department of Energy under Contract DE-AC03-76SF00098

DISCLAIMER

LBL-18504

REQUIREMENTS FOR A DATABASE MANAGEMENT SYSTEM

J. Dennis Lawrence and John McCarthy

Computer Science Research Department
Computing Division
University of California
Lawrence Berkeley Laboratory
Berkeley, California 94720

September, 1984

This work was supported by the Office of Scientific and Technical Information, U.S. Department of Energy under contract DE-AC03-76SF00098.

Introduction

This document discusses the requirements for a database management system that would satisfy the scientific needs of the Scientific Database Project. We give the major requirements of scientific data management, based on a system developed by Deutsch [1].[1] Actual requirements, for each category, are identified as mandatory, important, and optional.

- Mandatory. We should not consider a DBMS unless it satisfies all mandatory requirements.

- Important. These requirements, while not as crucial as the mandatory ones, are important to the easy and convenient implementation and operation of a scientific database.

- Optional. Such features are "nice extras".

We expect that the scientific database project can be implemented and operated in any DBMS that meets all of the mandatory and most of the important requirements.

This report, of course, reflects only our personal opinions. We welcome responses and discussion, particularly of aspects we have overlooked.

Terminology

Terminology for data structures varies widely. We use the following terms:

- (Data) Elements. The smallest named unit of data. Alternate names given in the literature: Data item, Field, Domain, and Attribute. Elements may be repeating or non-repeating.

- (Item) Group. A named collection of elements, contained within a record. Alternate names are: Structure, Segment, Group of Fields, Attribute group. Groups may be repeating or non-repeating.

- Record. A collection of elements and groups that describe an entity. Usually, a record is the primary unit of storage--and, thus, of I/O, updating, concurrency, etc. Alternate names are: Segment, Entity, Relation Row, and Table Row.

- File. A Collection of record instances, each of which has the same definition. Not all systems make use of this concept. In relational systems, this is the relation, or table

- Database. The complete collection of files (or record instances) that describe an application.

We have avoided implementation considerations here, since we do not believe that the

[1]Requirements for a Database Management System is based in large part on a previous document by Lawrence and Litton [4].

means used by the vendor to provide the services are directly relevant to our process of choosing a system. (Of course, indirectly they are crucial since performance and usability are critical concerns,) Also, we take no stand on the hierarchical/network/inverted/relational model controversy--any of these that satisfy our needs is deemed acceptable.

General Requirements

It is becoming apparent that the requirements of a DBMS for scientific uses are quite different from the requirements for business users [2,5]. In many ways, scientific requirements are more severe, because of the nature of the user community and the variety of data used.

- User groups tend to be many, autonomous, and frequently small.

- User experience with DBMS varies widely from very naive to very sophisticated, both among groups and within groups.

- Queries tend to range from simple to sophisticated, and are frequently unpredictable.

- Processing requirements also vary from the simple to the complex, and are also frequently unpredictable.

- Data may have an elaborate inherent structure.

- Output requirements include short-answer, reports (both pre-defined and ad hoc), plotting and machine readable.

- Scientific databases tend to be in constant change, both in contents and definition, to meet continually changing research environments.

- Usage is primarily interactive query (rather than update), since data tend not to change once they are in the database correctly.

- It may be necessary to "freeze" and archive the state of the database at various points in time, to allow search results to be reproduced and consistently extended.

- Scientific measurements have many attributes that must frequently be recorded in the database: value, uncertainty, units of measurement, normalization, domain of validity, method of measurement, conditions and constraints, type of data, source of data, bibliographic reference, proprietary status, security classification, and comments [3].

Some aspects of DBMS selection that affect both administrative and scientific users are not discussed in detail here, but they are discussed in a recent report from MITRE Corporation [6]. These general areas may be summarized as follows:

- Hardware and software environment.
- Cost--acquisition, rental, operational.
- Availability of documentation.
- Installation, testing.
- Vendor reputation: number of installations, years in business, reputation among customers.
- System limitations: number of record types, number of indices, elements per record, searchable fields per record, number of record occurrences per file, number of concurrent users, etc.
- Performance, for both update and retrieval.
- Space requirements, both central memory and secondary storage, for the DBMS, working sets for users, and the database itself.

I. DATA STRUCTURING MECHANISMS

1.1. Data Types

Data types describe the kinds of elements the DBMS is prepared to recognize and process.

Mandatory: Integer; real; character (both fixed-length and varying-length); text (i.e. very long character string); logical; personal name; date; double precision real.

Important: Time; dollars & cents; code (coded representation of a string), graphic images; complex.

Optional: Fixed decimal; bit string; map coordinates.

1.2. Intra-Record Structures

If a record is a collection of data elements that describe an entity, then an intra-record structure describes a relationship among some of these elements. Such a relationship could be structured physically by placing all of the constituent elements in a physical record, or in several records, or in several relational tables, as long as the user can view them and manipulate them collectively.

Mandatory: Vector; repeating element; group; repeating group; matrix.

Important: Derived (an element produced on retrieval by a calculation on one or more actual elements).

Optional: List; set.

1.3. Inter-Record Structures

These are structures maintained by the DBMS connecting different records.

Mandatory: Static network (the ability to navigate along pre-determined paths); thesaurus (the ability to build a thesaurus); ability to treat an index to a record as a regular record type.

Important: Dynamic network (The ability to navigate along ad-hoc paths), index based on sub-field values.

2. USER INTERFACES

2.1. Data Definition User Interface

A separate data dictionary is at the center of a database administrator's (DBA) control of the database--and, indeed, can be considered a pre-requisite to designating a complete system as a DBMS. Here we discuss data dictionary facilities.

Mandatory: Single level schema; ability to specify element names, types, length, and intra-record and inter-record structures; ability to specify element-level integrity; simple reporting (about the data definition); ability to update the data dictionary (e.g., to add new data elements indexes, and record-types) without disturbing unaffected portions of the database; ability to specify security and multiple-element integrity constraints.

Important: Two level schema (schema and subschema).

Optional: Three level schema (schema, subschema, storage schema); ability to specify physical structures.

2.2. External Linkage Interface

A DBMS can be completely self-contained, can be accessed by host language, or both.

Mandatory: Self-contained interactive query and update facilities; ability to transfer a portion of the database to a file that can be accessed by an application (e.g. FORTRAN) program.

Important: Subroutine-call host language interface for application program (e.g. FORTRAN); batch reporting and updating.

Optional: Embedded host language interface for FORTRAN and other languages.

2.3. Database Maintenance Interface

Maintenance involves initial loading; adding, deleting and changing elements and records; and restructuring.

Mandatory: Initial load in DBMS-specified format; interactive on-line updating; record-level updating; integrity checking for type, range, authority list; encode/decode; physical units conversion; unload/reload restructuring; intra-record integrity checking.

Important: Initial load in user-specified format; restructuring with no impact on unaffected portions of the database.

Optional: Update from host language program; item-level updating; inter-record integrity checking.

2.4. Data Retrieval Interface

Retrieval involves the operation of record selection, record sorting, and report writing. Sorting may be omitted and report format may be elementary in simple replies to queries. The "report" may be a program- readable computer file or plot.

2.4.1. Record Selection

This involves the form of query language, specification of selection criteria, and range of query targets.

Mandatory: On-line interactive self-contained query language; selection criteria to include equality, inequality (including greater and lesser), range presence/absence, elementary string searching; Boolean combinations of conditional expressions; ability to search based on the contents of any field; single file searching.

Important: Searching from host language; selection criteria to include variants, data variants, personal name exclusion/inclusion lists.

Optional: Multiple file searching; selection criteria to include pattern matching.

2.4.2. Record Sorting

If more than one record is selected, it may be necessary to sort the records before producing a report.

Mandatory: On-line sort of small sets of records; batch sort of large sets of records; multiple keys; increasing or decreasing order; numerical or character sort.

Important: On-line sorting of moderately large sets of records; sort keys need not be record (i.e. index) keys; sort by time, date, name.

Optional: On-line sorting of large sets of records; user-specified collating sequences.

2.4.3. Report Writing

Response to a query may be simple number, a few element values, a complex report with summaries, a plot, or a computer file.

Mandatory: Output to terminal, printer, or computer files; short answer, tables, reports; full report writer, including titles, headings, computed results, summaries (mean, max, min, total, count), conversion of units; predefined reports.

Important: Output to graphic terminal, plotter or film; ad-hoc reports; summaries include additional statistics; host language reporting (i.e. passing records to host language).

Optional: Reports include maps, equations, statistical analysis.

2.5. User Aids

These are system facilities that aid the DBA in tailoring the system to a user community. They are important attributes for a system that is to be used by inexperienced people.

Mandatory: On-line explanation of commands; browsing over indices or elements; DBA constructed dialogues to control user interaction.

Important: On-line system and application documentation; pre-defined queries; saving queries for re-use.

2.6. DBA Aids

Mandatory: Multiple independent databases on system.

Important: Some performance statistics to aid tuning; archiving versions of database.

Optional: Many performance statistics; multiple on-line versions of database.

3. SECURITY

Security involves protecting the database form unauthorized access, and deliberate or accidental harm. Because of its close involvement with protection and recovery, concurrency is discussed here also.

3.1. Database Protection

Mandatory: Protect against unauthorized writing, at database and record-type (file) level.

Important: Protect against unauthorized reading or writing at element level.

Optional: Protect against unauthorized reading or writing at record or element instance level (i.e., content-dependent access controls).

3.2. Recovery

Mandatory: Equivalent of checkpoint/restart plus roll-ahead. DBA controlled.

Important: Transaction logging; session restart.

Optional: Automatic recovery on error detection.

3.3. Concurrency

Mandatory: Multiple readers, single writer, at record level.

Optional: Element level concurrency.

REFERENCES

(1) D. Deutsch and E. Long. Characteristics of generalized database management systems. *Generalized Data Management Systems*, OECD Nuclear Energy Agency (1978), 27-42.

(2) A. Brooks. The structure of R and D information--some observations. *Generalized Data Management Systems*, OECD Nuclear Energy Agency (1978), 93-105.

(3) V. E. Hampel and D. R. Ries. Requirements for the design of a scientific data base management system. *Generalized Data Management Systems*, OECD Nuclear Energy Agency (1978), 111-131.

(4) J. Dennis Lawrence and Gerry Litton. LBL internal memo, April 31, 1980.

(5) A. Shoshani, F. Olken, H. Wong. Characteristics of Scientific Databases, LBL-17582, April, 1984.

(6) The MITRE Corporation, "A Corporative Analysis of Data Base Management Systems for Use in MEDLARS III" Metrek working paper 81 w00379 July,1981.

(7) J. Dennis Lawrence "List of DBMS Products and Vendors" LBL Internal Memo May 20, 1980.

ACKNOWLEDGEMENT

This work was supported by the Office of Scientific and Technical Information, U.S. Department of Energy under contract DE-AC03-76SF00098.

TECHNICAL INFORMATION DEPARTMENT
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720