BIBLIOGRAPHIC RETRIEVAL USING A RELATIONAL DATABASE SYSTEM

R. G. Crawford
Associate Professor
Department of Computing & Information Science
Queen.s University
Kingston, Ontario K7L 3N6

ABSTRACT

Previous work has described how the types of queries used for bibliographic retrieval could be expressed in a relational query language. This report focuses on the experience of using an existing relational database system for bibliographic retrieval. In particular, the facilities of the INGRES system are examined. Aspects that are of special interest for bibliographic retrieval are highlighted. As with most database systems, there are difficulties involving the use of textual data. These include the problems of setting up and querying a textual database.

LA RECHERCHE BIBLIOGRAPHIQUE SUR UN SYSTEME DE BASES DE DONNEES RELATIONNELLES.

RESUME

Dans des travaux précédents, l'auteur a décrit comment les différents types de recherches utilisées pour le repérage bibliographique pouvaient être exprimées dans un langage d'interrogation relationnelle. Le présent exposé étudie comment on peut utiliser un système existant de base de données relationnelle pour la recherche bibliographique. On présente notamment les caractéristiques du système INGRES. On en souligne les aspects d'intérêt particulier pour la recherche bibliographique. A l'instar de la plupart des systèmes de gestion de bases de données, ce système présente quelques difficultés avec l'utilisation de données textuelles, notamment la question de la création et de l'interrogation des bases de données textuelles.

BIBLIOGRAPHIC RETRIEVAL VIEWED RELATIONALLY

In general, on-line bibliographic retrieval systems have not been developed from such well defined models as have data retrieval systems. And yet, much work that has been done in the area of data base management systems can be easily and constructively applied to document retrieval (Schek, 1982). In particular, the relational data model presents a useful tool for bibliographic retrieval systems design (Macleod, 1979) (Crawford, 1981).

Viewing data relationally is equivalent to viewing it as a set of tables. Such an approach is suitable for several reasons. First, tabular structures are natural in the context of document retrieval. Such structures as document collections, indexes and dictionaries are generally seen as being two dimensional. Second, the relational model provides for simplicity. The user is able to view all aspects of the system in a clear, simple, coherent way. Third, there is consistency of access to all information from the user point of view. Finally, a number of approaches to relational query language design have been developed which provide a powerful and flexible mechanism with which to retrieve data (Chamberlin, 1976) (Crawford & Macleod, 1978).

In an ongoing research project, various aspects of the relational view as applied to bibliographic data are being considered. One piece of work involves the development of user-friendly interfaces. Another involves efficient implementation of primitive relational operations. The work reported here involves the evaluation of existing relational systems for bibliographic retrieval. In particular, one system, INGRES, is considered in some detail. The advantages and disadvantages found in using this system for bibliographic retrieval are described.

THE INGRES SYSTEM

INGRES (Interactive Graphics and Retrieval System) (Stonebraker et al, 1976) is a relational database system that is implemented on top of the UNIX operating system developed at Bell Telephone Laboratories (Ritchie and Thompson, 1974). The implementation of INGRES is primarily programmed in C, a high level language in which UNIX itself is written.

The primary query language supported by INGRES is QUEL (QUEry Language). It has points in common with Data Language/ALPHA (Codd, 1971) and SEQUEL (Chamberlin and Boyce, 1974) in that it is a complete query language which frees the programmer from concern for how data structures are implemented and what algorithms are operating on stored data. As such it facilitates a considerable degree of data independence. QUEL was not developed as a casual user language. Thus, it provides a powerful tool for testing the capabilities though not the user friendliness of this relational system.

THE BIBLIOGRAPHIC DATA BASE

Testing of the INGRES system was done using a collection consisting of 3183 documents from Communications of the Association for Computing Machinery covering a period of twenty-two years, 1958-1979. This data was available in machine-readable form, with each article consisting of the following fields:

TITLE--Title of article
ABSTRACT--Abstract as in journal
JOURNAL--Journal name and month, year of issue
AUTHOR--Author names, inverted, separated by ;
KEYS--Keywords included with the article
CATEGORIES--Computing Reviews categories
END--Termination line

To produce an INGRES data base with this data several issues had to be resolved.

Normalization

Intuitively, a relation is an unordered two-dimensional table in which each row represents a tuple and no two rows are identical. The columns of the table are called attributes. Since a relation represents a "flat table", it is clear that there can be no repeating fields in a relation. Thus, we could not construct our bibliographic data base using a single relation with the attributes listed above as in our description of an article. It is necessary to decompose this into relational form. This decomposition process is called normalization.

It has been observed that certain collections of relations have better properties in updating than do other collections containing the same data. The theory of normalization provides a rigorous discipline for the design of relations that have favorable update properties. The theory is based on a series of normal forms which provide successive improvements in the update properties of a database.

For our data, we could design a data base consisting of the following five normalized relations. Here we specify the relation name followed by a list of attributes in parentheses. The relation and attribute names are largely self-explanatory. An additional attribute, docno (document number) has been included for convenience.

CITATION (docno, title, journal, month, year)

AUTHOR (docno, name)

KEYS (docno, keyword)

CATEGORIES (docno, category)

ABSTRACT (docno, text)

It should be clear how these relations apply to our original data. For example, there will be exactly one entry (tuple) in the CITATION relation for every document in the data base, but there may be more than one entry in the AUTHOR relation for any given document, in the case of multiple authorship of documents.

Once the design of the relational data base is complete, it should be possible to load relations of the specified form into the relational data base system. Here further difficulties with INGRES are encountered.

Constraints Involving Textual Data

There are properties of bibliographic data that make it more difficult to work with than many other types of very well structured data. In particular, it is necessary to store a lot of textual data. And, in general, it is neither easy nor desireable to specify a limit on the number of characters that can be in, say, a title, or an author's name. Yet in INGRES the size, in characters, of all attributes, must be precisely specified. Thus, it is necessary to determine limits for the size of an author's name (35 characters) and a title (101 characters).

A further constraint was imposed by INGRES. A single attribute can be specified as a maximum of 255 characters, and the tuple width cannot exceed 498 characters. This places intolerable constraints on the handling of abstracts. The decision taken was to implement, for experimental purposes, an ABSTRACT relation containing two fields of text, as follows:

ABSTRACT (docno, text1, text2)

Each text field consists of 245 characters, permitting the storage of a total of about seven lines for each abstract.

Summary

The difficulties encountered in building the bibliographic data base do not seem to be inherent in the relational model. Normalization is a fairly intuitive process for bibliographic data, with the resulting relations being quite natural. However, some of the specific design features of INGRES were the cause of problems.

RETRIEVAL IN INGRES

In this section, a few QUEL queries will be shown. This will be described in more detail at the conference. The general form of the statement in QUEL to perform retrieval is:

```
RETRIEVE (list of desired attributes)
WHERE one or more specified conditions
```

Because an INGRES data base is, in general, comprised of several relations, it is necessary to qualify each attribute name that is used in a query so that the system will know to which relation each attribute is being applied. To do this we use the RANGE statement. Every QUEL interaction includes at least one RANGE statement of the form:

```
RANGE OF variable-list IS relation-name
```

Assume, for our examples, that we have performed the following, based on the relations defined above:

```
RANGE OF C IS CITATION
RANGE OF A IS AUTHOR
RANGE OF K IS KEYS
```

As a first very simple example of RETRIEVE, assume that we want to list the title for a specified document number, say, docno 12967. In QUEL, this is expressed as:

```
RETRIEVE (C.title)
WHERE C.docno = 12967
```

In fact, while this may not be a very attractive way to state this request, it is an important point to recognize that all requests involving single relations will be of this same form. The user no longer needs different commands to investigate authors, keyword lists, etc. Rather, the names of the relations and attributes must be known. As a further example, the query to list all occurrencs of authors named "Holt" is:

```
RETRIEVE (A.docno, A.name)
WHERE A.name >= "Holt"
      AND A.name < "Holta"
```

The response to this query by the system is:

```
12967   Holt,A.
13172   Holt,A.W.
12810   Holt,A.W.
12132   Holt,J.F.
11048   Holt,R.C.
10258   Holt,R.C.
```

This query illustrates another difficulty with the way text is handled in INGRES. The author's entire name, last name followed by initials, is simply treated as a string of characters. Thus we have to ask for all strings in a particular alphabetically ordered range. This can be avoided by specifying last names and initials as distinct attributes, but there is akwardness associated with that approach as well.

A great deal of flexibility exists in the specification of the WHERE clause. For example, the following produces a list of titles of papers published in the Communications of the ACM in January, 1976.

```
RETRIEVE (C.title)
WHERE C.journal = "Comm. of ACM"
      AND C.month = "Jan"
      AND C.year = "1976"
```

Queries involving attributes from more than one relation require the use of a "join term" in the WHERE clause. This term specifies the attributes from two different relations by means of which tuples from the two relations may be joined together. For example, to get a list showing year of publication and title of articles authored by R.C.Holt we write:

```
RETRIEVE (C.year, C.title)
WHERE A.name = "Holt,R.C."
      AND A.docno = C.docno
```

This produces the response:

```
1971    Comments on Prevention of System Deadlocks
1977    SP/k: A System for Teaching Computer Programming
```

The is clearly some akwardness caused by the requirement that the user know not only attribute names, but also the names of the relations in which those attributes occur. This difficulty may be partially overcome by the definition of macros, permitting essentially the definition of new commands that are tailored to the needs of a specific user group.

As an example of the use of macros, consider the following as a way to facilitate the writing of our last query.

```
{DEFINE; TITLESBY $n; RETRIEVE (C.year, C.title)
WHERE C.docno = A.docno
AND A.name = "$n" }
```

We can now write the following, obtaining the equivalent of our last query:

```
TITLESBY Holt,R.C.
```

Further details of retrieval will be presented at the conference.

CONCLUSIONS

The experiments conducted with the INGRES relational database system have led to the conclusion that the system, while in many ways powerful and flexible, is not convenient to use for bibliographic retrieval. It is clear that certain characteristics of bibliographic data must be taken into account in the design of the system.

In particular, provision must be made for handling text. Attributes such as titles and abstracts must be permitted to be of arbitrary length. As well, the ability to manipulate text is essential, for example for the construction of a list of index terms found in the abstracts.

Further, while it is claimed that bibliographic data may be naturally viewed as tabular, there are certain aspects of the relational structure that the casual user should not need to be aware of. For example, the user cannot be required to assume a new mental image of what a citation includes, just because the relational model does not permit repeating fields. Rather, the user must be able to view a citation in the traditional way, including the possibility of multiple authorship. Codd (1972) addressed this problem to some extent, presenting an operation called factoring, which converts a normalized relation to unnormalized form for presentation purposes. An extension of this notion to permit the user of a bibliographic retrieval system in a way that they find natural is necessary.

Little need be said regarding the query language used. It is not user friendly, but was not intended as such. The INGRES system includes a second user interface, CUPID, which is a graphics oriented, casual user language, which we have not yet evaluated. Of more interest is EQUEL (Embedded QUEL), which may be used to write customized interfaces in place of the QUEL language. This is the direction that would be followed to provide a bibliographic retrieval system based on INGRES.

REFERENCES

CHAMBERLIN, Donald D. "Relational Data-Base Management Systems", in *Computing Surveys*. Vol. 8, No. 1 (March 1976), pp. 43-66.

_______. and BOYCE, R. "SEQUEL: A Structured English Query Language", in *Proceedings of the 1974 ACM-SIGMOD Workshop on Data Description, Access and Control*. Ann Arbor, Michigan, (May 1974), pp. 249-264.

CODD, E.F. "A Data Base Sublanguage Founded on the Relational Calculus", in *Proceedings of the 1971 ACM-SIGFIDET Workshop on Data Description, Access and Control*. San Diego, California (November 1971), pp. 35-68.

_______. "Relational Completeness of Database Sublanguages", in *Data Base Systems*, Courant Computer Science Symposium 6, Rustin, R., Ed. Prentice-Hall, 1972, pp. 65-98.

CRAWFORD, R. G. "The Relational Model in Information Retrieval", in *Journal of the American Society for Information Science*. Vol.

32, No. 1 (January 1981), pp. 51-64.

———. and MACLEOD, I. A. "A Relational Approach to Modular Information Retrieval Systems Design", in *Proceedings of the 41st Conference of the American Society for Information Science*. New York, (November 1978), pp. 83-85.

MACLEOD, I. A. "SEQUEL as a Language for Document Retrieval", in *Journal of the American Society for Information Science*. Vol. 30, No. 5 (September 1979), pp. 243-249.

RITCHIE, D.M. and THOMPSON, K. "The UNIX time-sharing System", in *Communications of the ACM*. Vol. 17, No. 7 (July 1974), pp. 365-375.

SCHEK, H.J. and PISTOR, P. "Data Structures for an Integrated Data Base Management and Information Retrieval System", in *Proceedings of the Eighth International Conference on Very Large Data Bases*. Mexico City, (September 1982), pp.197-207.

STONEBRAKER, Michael, WONG, Eugene, KREPS, Peter, HELD, Gerald "The Design and Implementation of INGRES", in *ACM Transactions on Database Systems*. Vol. 1, No. 3 (September 1976), pp. 189-222.