APPLYING UNIVERSAL RELATION THEORY TO BIBLIOGRAPHIC DATA

R.G. Crawford
Associate Professor
Department of Computing & Information Science
Queen.s University
Kingston Ontario
Canada

ABSTRACT

It is appropriate that a bibliographic retrieval system should be designed based on a coherent data model. In particular, the relational view of data is easily applied to bibliographic data. However, some aspects of this model that are advantageous for systems design present difficulties for the user. In particular, the advantages of normalized relations may be lost on the user who must navigate an apparent maze of relations to express a query. Universal relation theory offers a potential solution to this difficulty. Examples using a universal relation in the Ingres relational database system are discussed.

Théorie de la relation universelle appliquée à la conception bibliographique.

Il est essentiel qu'un système de recherche bibliographique soit conçu sur la base d'un modèle cohérent. En particulier, l'interrelation des données est facilement applicable à la conception bibliographique. Cependant, certains aspects de ce modèle qui sont à l'avantage du design des systèmes présentent des difficultés pour l'utilisateur. En particulier, les avantages de relations normalisées peuvent confondre l'utilisateur qui doit alors naviguer dans un labyrinthe apparent de relations pour exprimer une demande. La théorie de la relation universelle offre une solution potentielle à cette difficulté. Des exemples utilisant une relation universelle dans le système de banque de données relationnelle de Ingres sont discutés.

UNIVERSAL RELATION

INTRODUCTION

The advantages of the relational view of data for bibliographic data have been previously described (Crawford 1981). The relational model provides a simple, coherent, consistent, and generally natural approach in this area. The design of a normalized set of relations for bibliographic data is conceptually easy. Nevetheless, there are obstacles to the use of such a database system. While it may be possible to express very powerful queries, some of the more mundane queries may be somewhat cumbersome.

In the following section, this problem is shown explicitly in the INGRES relational database system. The next section introduces universal relation theory, and this is followed by a section in which the application of this theory is demonstrated.

BIBLIOGRAPHIC RETRIVAL USING INGRES

INGRES, a relational database management system, has been used for bibliographic retrieval. Some results have been reported previously (Crawford 1983). One of the conclusions of that work was that it is necessary to permit use of a bibliographic retrieval system in a way that users find natural, and that such an approach was not apparently facilitated by INGRES.

This current report focuses on the use of views in INGRES to facilitate retrieval of bibliographic data. In particular, viewing the data as a universal relation is considered.

INGRES (Interactive Graphics and Retrieval System) (Stonebraker et al 1976) (Stonebraker 1980) is a relational database system that is implemented on top of the UNIX operating system. INGRES is written in C and runs in UNIX as a user job.

Bibliograhic Database

The database consists of five normalized relations. These are listed below with the relation name followed by a list of attributes in parentheses. The names are largely self-explanatory; docno (document number) is a unique integer value assigned to each document.

CITATION (docno, title, journal, month, year)
AUTHOR (docno, name)
KEYS (docno, keyword)
CATEGORIES (docno, category)
ABSTRACT (docno, text)

A careful description of normalization as applied to bibliographic data may be found in (Crawford et al 1983a)

Retrieval in INGRES

The data manipulation and query language of INGRES is QUEL. QUEL is a tuple relational calculus language. This means that range statements and range variables are an important part of the language. It is non-procedural and includes a comprehensive set of arithmetic and

aggregation operators.

For the QUEL examples sown in this paper we assume the following range statements.

```
RANGE of C is CITATION
RANGE of A is AUTHOR
RANGE of K is KEYS
RANGE of T is CATEGORIES
RANGE of B is ABSTRACT
```

As an example of retrieval using QUEL, consider the following approach to retrieving title and year for all documents having "compiler" listed as a keyword.

```
RETRIEVE (C.title, C.year)
WHERE C.docno = K.docno
AND K.keyword = "compiler"
```

This query is a fairly clear expression of what is desired. However, notice that it is necessary to specify the join (C.docno = K.docno) of the CITATION and KEYS relations.

Consider an even more complicated example.. Again, retrieve all references to "compiler", but in this case retrieve more information, including title, author, journal, year and abstract. The expression of this query in QUEL is:

```
RETRIEVE (C.title, A.author, C.journal, C.year, B.abstract)
WHERE C.docno = A.docno
AND   C.docno = B.docno
AND   C.docno = K.docno
AND   K.keyword = "compiler"
```

It is unacceptable to require a user to express in such a complex way what is a fairly simple request. Users generally have in view an entire document, and are not interested in knowing that the document.s author, title, and keywords are in three different relations, each of which must be named. An approach to overcoming this particular problem may be found in universal relation theory.

UNIVERSAL RELATION THEORY

The concept of a universal relation for a relational database has been a subject of discussion recently (Ullman 1982) (Maier & Ullman 1983) (Kent 1981). A universal relation is a single relation whose schema consists of all the attributes in any of the relational schemas of the database. (Assume attributes representing the same thing in different relations are given the same name and attributes representing different things are given different names.) The purpose of a universal relation is to provide the database user with a simplified model in which he can compose quieries without regard to the underlying structure of the relations in the database.

UNIVERSAL RELATION

The universal relation is not a physical storage structure for the data in the database. If data were stored in a single relation, anomalies could arise when inserting into, deleting from or updating the database. Although the user sees and queries a single relation representing the entire database, the data is in fact being stored in a number of normalized relations. The user is not required to know what the attributes and schemas are for each of the normalized relations. The user need not worry about how to join relations to answer queries. It is up to the database management system to interpret and handle the queries by ensuring that the appropriate relations are joined for a specific query.

A bibliographic database lends itself well to the concept of a universal relation. In many instances, a naive user is querying a bibliographic database to obtain some or all information regarding a particular document, author or subject (category). With a universal relation the user is able to formulate queries based on any of the attributes very easily. The user need not know how joins are made on normalized relations or understand in great detail how queries are interpreted.

In order to present the user with a single universal relation containing all the information stored in the underlying normalized relations, it may be necessary to pad out some of the tuples in the universal relation with null values.

A formal description of a universal relation for bibliographic data is given in (Crawford et al 1983b). Suffice it to say here that, for each of the theoretical problems associated with the universal relation (Kent 1981), either the problem does not apply to our data, or a reasonably simple solution can be found. Here our concern is the practical implementation of the universal relation concept in INGRES.

The Universal Relation in INGRES

INGRES permits the definition of views on the relations. This provides several capabilities. For example, as a privacy provision, a user could be permitted a view of a certain relation that did not include all attributes of the relation. Or, for convenience, a user may wish to view the database in ways that involve joining two or more relations to construct the view.

For our database, the universal relation is that view that involves the join of the 5 constituent relations; and therefore includes the following attributes:

```
UNIVERSAL (udocno, utitle, ujournal, umonth, uyear,
           uname, utext, ucategory, ukeyword)
```

UNIVERSAL RELATION

The appropriate INGRES command for defining the universal relation is:

```
define view universal (udocno = C.docno,
       utitle = C.title, ujournal = C.journal,
       umonth = C.month, uyear = C.year,
       uname = A.name, utext = B.text,
       ucategory = T.category, ukeyword = K.keyword)
       where B.docno = A.docno
       and   C.docno = A.docno
       and   T.docno = A.docno
       and   K.docno = A.docno
```

Now, queries can be stated in terms of this one relation. We must define a range variable on this relation, such as:

```
RANGE of U is UNIVERSAL
```

Consider our previous two examples of queries, restated in terms of the universal relation. The first is:

```
RETRIEVE (U.utitle, U.uyear)
WHERE U.ukeyword = "compiler"
```

The join term is omitted because it already is encompassd by the view.

The second query takes the form:

```
RETRIEVE (U.utitle, U.uauthor, U.ujournal, U.uyear, U.uabstract)
WHERE U.Keyword = "compiler"
```

This further emphasizes the facility with which queries on the Universal relation may be expressed. Note that we are using attribute names with the prefix "u" here to distinguish then from the attribute snares in the original relations. This is not necessary, and in fact we would not want our universal relation attribute names to be so unweildy.

CONCLUSIONS

For bibliographic data, the advantages of the universal relation are apparent. The user can compose queries without regard to the underlying structure of the component relations. Further, none of the difficulties which may attend, in theory, to the universal relation, pertain to its use for bibliographic data. This emphasizes the naturalness of the relational view for bibliographic data.

There are still problems attendant to the display of results that are stored relationally. However, these may be handled by writing additional software to format the output in bibliographic applications.

REFERENCES

CRAWFORD, R.G., "Bibliographic Retrieval Using a Relational Database System", in Proceedings of CAIS Annual Conference, (May 1983), 108-116.

CRAWFORD, R.G., "The Relational Model in Information Retrieval", in Journal of the American Soceity for Information Science, Vol. 32, No. 1 (January 1981), pp. 51-64.

CRAWFORD, R.G., BECKER, H.S., OGILVIE, J.E., "A Relational Bibliographic Database", Technical Report #83-149, Department of Computing & Information Science, Queen.s University, September 1983.

CRAWFORD, R.G., BECKER, H.S., OGILVIE, J.E., "Universal Relation Theory As Applied to Bibliographic Data", Technical Report #83-150, Department of Computing & Information Science, Queen.s University, (1983), 13 pages.

KENT, W., "Consequences of Assuming a Universal Relation", ACM Transactions on Database Systems. 6(4) : 539-556; 1981.

MAIER, D., ULLMAN, J.D., "Maximal Objects and the Semantics of Universal Relation Databases", ACM Transactions on Database Systems. 8(1) : 1-14; 1983.

STONEBRAKER, M. "Retrospection on a Database System", in ACM Transactions on Database Systems Vol. 5, No. 2 (June 1980), pp. 225-240.

STONEBRAKER, M., WONG, E., KREPS, P., HELD, G., "The Design and Implementation of INGRES", in ACM Transactions on Database Systems, Vol. 1, No. 3 (September 1976), pp. 189-222.

ULLMAN, J.D., "Principles of Database Systems", Rockville, Maryland : Computer Science Press Inc. : 1982.