5K-3

関係モデルデータベースの問合わせ用言語に関する一考察
A Study of Relational Data Manipulation Languages

Goro Watanabe, Hidehiko Tanaka and Tohru Moto-oka
(Faculty of Engineering, University of Tokyo)

1. Introduction

Various types of data manipulation languages have been proposed, such as SQUARE, SEQUEL, QBE etc. This paper presents another high-level data manipilation language for relational model databases. The language proposed in this paper aims one which is simple, systematic and capable to be used by both casual users and experts of the system.

The language is designed to be used on relational databases, therefore it is named RDML (Relational Data Manipulation Language). RDML has a layer structure which consists of three levels of different languages, called A, B, C-level language, respectively.

2. RDML language system

A-level language is the outmost level interface, which is for casual users who want to use the system with simple queries only. They need not have the exact knowledge of relation structures, relation names and even the data model itself. The only knowledge required for the user is a set of names of attributes. If he forgets the names, the system will help him by showing the list of the names of the attributes.

B-level is a relational calculus level. There are two classes of relational calculus language. One is the tuple relational calculus and the other the domain relational calculus. B-level language is based on the latter one. This level of the language system is opened for experiensed users who want to use more complicated queries. The user must be conscious of what relations are stored in the database. The queries are grasped as arbitrary functions applied to the relation domains. As a characteristic of the relational calculus, he has only to specify what he wants with the functions. Also the nonquery aspects, such as insertion, deletion and modification of relation tuples and their components are provided.

C-level is a relational algebra level. Experienced users can use C-level language, although this level is intended to be the internal representation of the queries. A relational algebra language is the so-called procedural language. C-level language specifies the order of operations, while B-level language leaves it to the system. The expressions written in B-language only show the result which the user wants. So the B-language expressions must be converted to those of C-language which indicate the order of operations. The conversion is left to the system, which automatically determines an efficient order. However, for users who want to specify the order of operations, C-level feature may be helpful. B and C-level languages are essentially derivable from first order predicate calculus, and they are complete after the manner of Codd.

A simpl example is presented here.
Consider the following relations.

```
EMP(NAME, SAL, DEPT),     LOC(DEPT, FLOOR)
```

then a query:

list the names of employees who work on the second floor.

is illustrated as:

```
A:   LIST  NAME
     WHERE  FLOOR  IS  2

B:   LIST  N
     WHERE  EMP(NAME:N, DEPT:D)  AND  LOC(DEPT:D, FLOOR:2)

C:   PROJ(JOIN(EMP, DEPT, DEPT, SEL(LOC, FLOOR=2)), NAME)
```

The conversion algorithm from A-language to B-language is simpler, when compared with that from B-language to C-language. As mentioned above, B-language is relationally complete, which means that all the necessary information to compute the result is contained in it. But A-language lacks the information as to which relation contains the attributes. The lack is supplemented by the system when the language is converted. The ambiguity which is derived from duplicated attribute names among relations can be settled by dialog between the user and the system.

The conversion from B-language to C-language is rather difficult, because the order of operations must be determined from plain results which the user specified, moreover, the order cannot be uniquely determined. Namely, the system bears the responsibility for optimizing it. The first stage of the algorithm is to guarantee that each atom and each basic operation of B-language can be converted to those of C-language. Then the formula of B-language which represents the query is decomposed to atoms connected with basic operators. Now, they can be converted to the C-level expressions, which can recompose the formula represented in C-language. The algorithm is entirely dominated by the selection of basic operations of the relational algebra.

3. Conclusions

We proposed a language, but a lot of problems are still left for the future. Many of them concern to the implementation of the language.
For example, the optimization of the operation order of the relational algebra. It is determined by the evaluation order of unfolded formula of the relational calculus, and many factors influence the efficiency.

References

Pirotte, A., Fundamental and Secondary Issues in the Design of Non-Procedural Language, Proc. of VLDB, 1979;

Ullman, J. D., Principles of Database Systems, Computer Science press, Inc., 1980;