A Table Driven User-Friendly Interface for Relational Databases

Yuji Wada Nozomi Nakakawaji Takuro Kanamori
Takaya Ishida
Information Systems & Electronics Development Laboratory
MITSUBISHI ELECTRIC CORPORATION
5-1-1 Ofuna, Kamakura-City, Kanagawa Prefecture 247 JAPAN
(0467)44-6136(ext.,4533)
(0467)44-9269

ABSTRACT

In this paper, we present a user-friendly visual interface called MELQUERY for performing table driven manipulation of relational databases on ME1000 multimedia engineering workstations.

MELQUERY converts the user inputs to SQL, an equivalent international-standard database query language, which is then used to access the databases.

KEYWORDS

User-Friendly Interface, Visual User Interface, Table Driven User Interface, Workstation Database

1 INTRODUCTION

We consider one of the most important facilities of database management systems to be user-friendliness of the database management system.

In the field of database user interfaces, most conventional database management systems provide the database end users with the command typing oriented user interface such that the end users input the database manipulation commands by stroking the keyboard.

As the above user interface requires the users a lot of keyboard operations, it is apt to cause the incorrect input operations and it has the disadvantage that the users must remember the syntax of the database manipulation commands as well as the relation names and the data types.

In the field of database models, a relational model is widely spread [CODD70]. This is because the relational model is easy to understand and is theoretically well-formalized on mathematical background.

In the field of database languages, a database language called SQL has become an international-standard database query language lately [ISO87].

Under these circumstances on the database technology, we have implemented a table driven user-friendly interface called MELQUERY which enables database end users to access the relational database visually and interactively with a mouse and icons under the multiwindow environment of the engineering workstation.

2 RESEARCH BACKGROUND

In this chapter, we show the technological trend of database end user interface and the requirements for the database user interface.

2.1 Technological Trend of the Database User Interface

Lately, the theme on the database user interface has been actively researched and new interfaces with improved user-friendliness have been developed [KOJI88].

The typical examples of the database user interface are the visual interface, the query interface by examples, the natural language, the AI (artificial intelligence) interface and the fourth generation language [MAKI87].

The visual interface has the characteristics that the database users can recognize the data manipulation objects as icons and windows of a workstation, and the users directly manipulate the recognized visual objects with the pointing device of the workstation [DAVI86, FRAS86, GOLD85, HEIL85, KATO87, WATS88]. [GOLD85] and [DAVI86] present graphically manipulationg systems ISIS and ISIS-V, respectively, based on a semantic database model. [FRAS86] describes the Icon-Based System (IBS) consisting of an object interface and an icons-based set of commands. [HEIL85] shows a QBE-style interface G-WHIZ for the functional data model. [KATO87] presents a visual interface OBAQ for an object-oriented query in relational databases. [WATS88] describes a user interface QUERY MANAGER for both novice and sophisticated database users of the OS/2 Database Services.

The query interface by examples has the characteristics that the users can manipulate the database by inputting the database query examples in a terminal full-screen mode [CHAN80, CHAN81, WHAN87, ZLOO77]. [ZLOO77] proposes a graphical user interface QBE (Query by Example) for relational databases. [CHAN80] and [CHAN81] presents a picture query language QPE (Query by Pictorial Example) based on QBE for pictorial databases. [WHAN87] shows an integrated office information system OBE (Office by Example), an extension of QBE, supporting database, word processing, electric mail, graphics and image.

The natural language has the characteristics of being so flexible for the users as to absorb the ambiguity of the user requirements [MAKI87]. In [NAKA86], the technology is researched that the database query with the natural language is transformed into the relational database query language.

In the field of AI interfaces, the intelligent guidance technology and the intelligent information retrieval technology have been developed recently [MAKI87].

The fourth generation language is said to be a programless software development tool and it has the characteristics of the integration of database access functions, spreadsheet manipulation functions, graphics processing functions and software development functions [MART86].

Many commercial products of the fourth generation language have been already implemented and released to the markets.

Summarizing the above-mentioned trend on the database user interface, we consider as follows:

(1) the natural language and the AI interface are on the stage of the basic research,

(2) the query interface by examples and the fourth generation language have already reached the stage of the practical level,

(3) the visual interface is on the intermediate stage between the basic research and the practical level product.

On the other hand, considering the recently spread multimedia workstations equipped with bit-map display units and mouses, we can predict that the above visual interface will become the most important user interface technology among the above-mentioned five kinds of the database user interface.

So, we are researching and developing the new visual interface technology.

2.2 Requirements for the Database User Interface

The database application jobs are classified into the regularly processed jobs and the irregularly processed jobs.

The former corresponds to the usage style of the batch processing jobs such as the database application programs written in a procedural language (for example, COBOL or FORTRAN), and the latter corresponds to that of the interactive processing jobs such as at workstations or personal computers.

The processing style of the database access is categorized into the centralized processing and the distributed processing. The former is realized with the centralized database system which is constructed on a large-scaled host computer, and the latter is realized with the distributed database system which is constructed on workstations and/or large-scaled host computers.

In order for the database users to perform the irregularly processed jobs with the distributed or centralized database system constructed on the workstations which are the major processing units at present, the following functions are required for their database user interface:

(1) the data manipulation objects are recognized visually,

(2) the data manipulation objects are directly selected before the database manipulation functions are selected,

(3) the user interfaces of the data manipulation functions are unified among the same data manipulation functions, regardless of the contents or the locations of the data manipulation objects,

(4) the database access functions are integrated with the data processing functions and the report generation functions.

3 TABLE DRIVEN USER-FRIENDLY INTERFACE MELQUERY

In this chapter, we describe a table driven user-friendly interface called MELQUERY which we have implemented on our engineering workstation called ME1000 series, lately.

3.1 Overview

In order to satisfy the database user interface requirements mentioned in the previous chapter, we have adopted the visual interface approach because this approach has the matured technology and reaches to near the practical level, and we have implemented a table driven user-friendly interface MELQUERY.

MELQUERY is a database service under control of the engineering workstation desktop environment called DESKTOPMANAGER, which provides the document processing service, the electronic file service, the electronic mail service, the geometric service and the business graph processing service as well.

3.2 Configuration

Figure 1 shows the software configuration of MELQUERY of which components have the following functions:

(1) Man Machine Interface (abbreviated MMI)

MMI receives the user inputs via the mouse, menus, and icons under the multi-window environment of the engineering workstation. MMI displays the contents of the data manipulation objects and displays the data manipulation results.

(2) SQL Interface (abbreviated SQLi/f)

SQLi/f converts the user inputs into the equivalent SQL sentences and then transmits the SQL sentences to RDBASEi/f (mentioned later). SQLi/f returns the execution results of RDBASEi/f to MMI.

(3) RDBASE Interface (abbreviated RDBASEi/f)

RDBASEi/f transforms the SQL setences into the equivalent data manipulation commands supported by RDBASE (mentioned later) and transmits the RDBASE data manipulation commands to RDBASE. RDBASEi/f returns the execution results of RDBASE to SQLi/f.

(4) Relational Database Management System (called RDBASE)

RDBASE is a relational database management system supported on our engineering workstation.

(5) Personal Table Management System (abbreviated PTMS)

In this paper, we define a personal table to be a spreadsheet which an individual can manipulate only for himself. PTMS receives the user inputs from MMI and executes the data processing of the personal tables such as the table data calculation of the personal table.

3.3 Functions

3.3.1 Major Characteristics

MELQUERY has the first major characteristics of being a table driven user-friendly interface for databases and personal tables.

MELQUERY has the second major characteristics of providing the database users with the visual user-friendly interface which unifies the relational database manipulation with the personal table data processing. With this interface, the MELQUERY users take an advantage that they do not need to input the names, the contents, the data items, the data types of the relational databases and the personal tables by stroking the keyboard, because the MELQUERY users can point the desired data manipulation object from among the displayed objects with the mouse before the MELQUERY users point the desired data manipulation function. In other words, the MELQUERY users do not need to remember the names, the contents, the data items, the data types of the relational databases and the personal tables in order to manipulate the databases and process the personal table data.

MELQUERY has the third major characteristics that MMI is so independent of RDBASE that MMI can be easily interfaced with another database management system product which supports SQL language, because the contents of the data manipulation specification which the MELQUERY user inputs are converted into the equivalent sentences of the international-standard database query language SQL.

3.3.2 Objects System

The MELQUERY data manipulation objects are classified into a DBBOX object which means an entire set of the relations in a relational database, a PTBOX object which means an entire set of the personal tables, DB objects each of which means an individual relation, and PT objects each of which means an individual personal table. These objects are displayed as the icons, as shown in Figure 2.

(1) DB Object

The DB object has the same concept with the RDBASE relation. Every attribute of the relation is allowed to be specified one of data types such as real number, integer, alpha-numeric character string, and Japanese character string.

Every DB object is specified the access right such as the ownership, the read/write permission for the owner and the nonowner.

(2) PT Object

The PT object means the results (such as the derived data) of the relational operations for the DB object, which can be stored on the secondary storage.

The table data calculation can be executed among either the table columns or the table rows of the PT object. The PT object has the data structure of two -dimensional tabular form, as depicted in Figure 3. A data type such as display specification can be defined to every data item of the PT object.

The ordering of the columns and the rows of the DB object is meaningless because the DB object is a relation, on the other hand that of the PT object is meaningful because the PT object requires to be designated the specific columns and rows where the summation or average values are inserted. The n-th column and the m-th row of the PT object are uniquely identified with a column number Cn and a row number Lm, respectively.

As MELQUERY users are required only to point every MELQUERY object such as its relation name and its attribute names, or its personal table name and its data item names with the mouse operation based on the interwindow copy function of the engineering workstation, they take the advantage that they have the lower probability of incorrectly manipulating the databases or the personal tables.

3.3.3 Commands System

MELQUERY provides the commands of the relational database manipulation functions, the personal table data calculation function, and the basic data manipulation functions.

The commands of the basic data manipulation functions are classified into LIST command, DISPLAY command, DROP command and STORE command. Pointing the DBBOX object or the PTBOX object results in displaying the listing of the relation names and the relation access rights or the personal table names, respectively. DISPLAY command displays the contents of the DB object or the PT object. DROP command drops the DB object or the PT object. STORE command accommodates the DB object or the PT object into the DBBOX or the PTBOX, respectively.

The commands of the relational database manipulation functions for the DB object are classified into SELECT command, PROJECT command, JOIN command, CREATE command, UPDATE command, INSERT command, DELETE command, SORT command, CREATE INDEX command and DROP INDEX command.

The personal table data calculation function provides CALCU command.

MELQUERY users can point each of all the above commands with the mouse from the command menus or the forms which are displayed depending on both the context of the number of the pointed objects and the kinds of the pointed objects.

MELQUERY provides the stepwise retrieval function as STEPWISE command, the access authorization function as ACCESS command, the database recovery function as RECOVER command, the print function of a relation as PRINT command and the conversion function of a relation into a personal table as P/TCONVERT command, too.

3.3.4 Relationships of Objects with Commands

We summarize the relationships of objects with commands, as shown in Figure 4.

In order to create a new relation, the MELQUERY user needs to point the empty DB object icon before pointing CREATE command. Then the new DB object icon is displayed on the MELQUERY window before it is available for the MELQUERY user to manipulate the relational database.

Pointing the DBBOX object causes the name and access right listing of all the existing relations to be displayed on the MELQUERY window. Pointing the desired relation names from among the name listing causes the new DB object icons to be displayed on the MELQUERY window before they are available for the MELQUERY user to manipulate the relational database with SELECT, PROJECT, JOIN, UPDATE, INSERT, DELETE and SORT commands.

Pointing the DB object before pointing SELECT command enables the MELQUERY user to retrieve the relational database. Then the retrieval results are displayed on the MELQUERY window.

Pointing the DB object before pointing ACCESS commands enables the MELQUERY user to change the access autorization of the relation.

Pointing the DB object before pointing RECOVER command enables the MELQUERY user to recover the relation.

Pointing the button menu of P/TCONVERT command displayed on the desired relation window generates a new personal table and causes a new PT object icon to be displayed on the MELQUERY window. Pointing the PT object before pointing STORE command causes the PT object icon to be erased from on the MELQUERY window and causes the PT object to be accommodated into the PTBOX object.

Pointing the PT object before pointing CALCU command enables the MELQUERY user to calculate the table data among the columns or the rows of the personal table. The calculation results can be another personal table.

Pointing the PTBOX object causes the name listing of all the existing personal tables within the PTBOX to be displayed on the MELQUERY window. Pointing the desired personal table names from among the name listing causes the new PT object icons to be displayed on the MELQUERY window before they are available for the MELQUERY user to calculate the table data.

4 EXAMPLES

In this chapter, we present some typical examples of the relational database manipulation and the personal table data calculation with MELQUERY under the control of DESKTOPMANAGER, as follows:

(1) Create

Figure 5 shows an example of creating a new relation which is composed of three attributes such as "name", "address", and "balance". In this example, the empty DB object icon is pointed with the mouse before inputting the relation name, the number of the attributes, the attribute names, the data types and the data lengths.

(2) Insert

Figure 6 shows an example of inserting a new tuple, of which attribute called name has the value of "Field, W.", of which attribute called address has the value of "43 Cherry La.", and of which attribute called balance has the value of "0.0", into the relation called members. In this example, the DB object icon is pointed with the mouse before inputting the new tuple value.

(3) Update

Figure 7 shows an example of searching the tuples of the relation called members which satisfiy the condition that the attribute called balance has the negative value, and of adding twenty to the value of the attribute called balance in the searched tuples. In this example, the DB object icon "members" is pointed with the mouse before inputting the search condition equation such as balance<0 and the updation equation such as balance=balance+20.

(4) Retrieve

Figure 8 shows an example of retrieving the tuples of the relation called publish which satisfy the condition that the attribute called zipcode is equal to "101". In this example, the DB object icon "publish" is pointed with the mouse before inputting the retrieval condition equation such as publish.zipcode="101".

(5) Join and Project

Figure 9 shows an example of joining two relations called library and publish with the common publishing company names between the two relations of library and publish. In this example, the two DB object icons "library" and "publish" are pointed with the mouse before inputting the equi-join condition equation such as publish.companyname=library.comapny and the projected attribute names such as library.bookname, publish.companyname, publish.zipcode, and publish.address.

(6) Calculate

Figure 10 shows an example of the personal tabale data calculation executed among the table colums. In this example, the PT object icon "nouhin" is pointed with the mouse before inputting the calculation equation such as C4=C2*C3.

5 CONCLUSIONS

We have shown that the table driven user-friendly interface called MELQUERY for database end users to access the relational databases. MELQUERY has been implemented on the UNIX operating system of the ME1000 series. MELQUERY is written in C language and the number of the source statements is about 60K steps. MELQUERY has the advantage of being a visual user interface and enables the database end users to access casually the relational databases.

We consider that the further research and development subjects are such that MELQUERY is extended to support firstly the remote database access mechanism and finally the distributed database access mechanism constructed on the micro-mainframe linkage environment.

UNIX is the trademark of AT&T Bell Laboratories.
OS/2 is the trademark of IBM.

REFERENCES

[CHAN80] N. S. Chang and K. S. Fu: "Query by Pictorial Example," IEEE Trans. Soft. Eng., Vol. SE-6, No. 6, pp.519-524 (1980).

[CHAN81] N. S. Chang and K. S. Fu: "Picture Query Languages for Pictorial Data-Base Systems," IEEE Computer Vol. 14, pp.23-33 (Nov. 1981).

[CODD70] E. F. Codd: "A Relational Model for Large Shared Data Banks," Comm. ACM, Vol. 13, No. 6, pp.377-387 (1970).

[DAVI86] J. W. Davison and S. B. Zdonik: "A Visual Interface for a Database with Version Management," ACM TODS, Vol. 4, No. 3, pp.226-256 (July 1986).

[FRAS86] C. Frasson and M. Er-radi: "Principles of an Icon- Based Command Langiage," Proc. of ACM SIGMOD'86, pp.144-152 (May 1986).

[GOLD85] K. J. Goldman, et al.: "ISIS: Information for a Semantic Information System," Proc. of ACM SIGMOD'85, pp.328-342 (May 1985).

[HEIL85] S. Heiler and A. Rosenthal: "G-WHIZ, a Visual Interface for the Functional Model with Recursion," Proc. of the 11th VLDB, pp.209-218 (August 1985).

[ISO87] ISO: "Information Processing Systems-Database Language SQL," INTERNATIONAL STANDARD, ISO 9075- 1987(E) First Edition (1987-06-15).

[KATO87] A. Kato, et al.: "A Visual Interface for an Object-Oriented Query in Relational Data Bases," Trans. of IEICE, Vol. J70-D, No.11, pp.2236-2242 (1987-11).

[KOJI88] I. Kojima and H. Tanuma:"Workstation Databases, Visual Interface, Distributed Processing, Object- Oriented, Local Area Networks," JAACE SYSTEMS AND CONTROL, Vol. 32, No. 5, pp.294-301 (1988).

[MAKI87] A. Makinouchi and B. D. Yoon: "A Survey of Multimedia Database Interfaces," Information Processing Society of Japan, Vol. 28, No. 6, pp.694-704 (June 1987).

[MART86] J. Martin and J. Leben: "Fourth-Generation Language," Vol. 2, Prentice-Hall, Englewood Cliffs, N.J., (1986).

[NAKA86] M. Nakagawa and T. Kato: "Methods of the Semantic Analysis in a Natural Language Interface for Relational Database," Trans. of Information Processing Society of Japan, Vol. 27, No. 11, pp.1069-1076 (Nov. 1986).

[WATS88] S. L. Watson: "OS/2 Query Manager Overview and Prompted Interface," IBM Systems Journal, Vol. 27, No. 2, pp.119-133 (1988).

[WHAN87] K.-Y. Whang, et al.: "Office-by-Example: An Integrated Office System and Database Manager," ACM TODS, Vol. 5, No. 4, pp.393-427 (Oct. 1987).

[ZLOO77] M. M. Zloof: "Query-by-Example: A Data Base Language," IBM Systems Journal, Vol. 16, No. 4, pp.324-343 (1977).

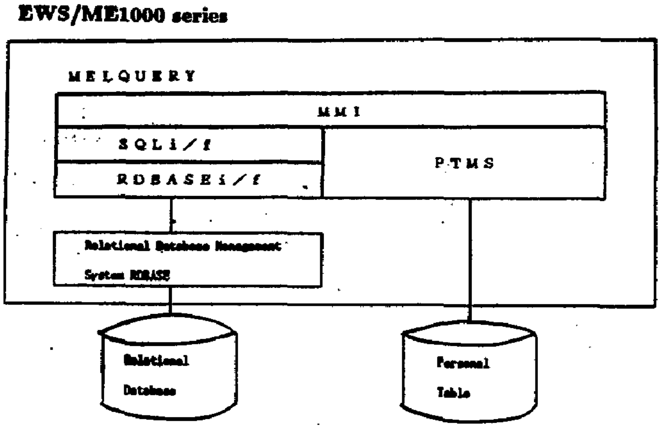
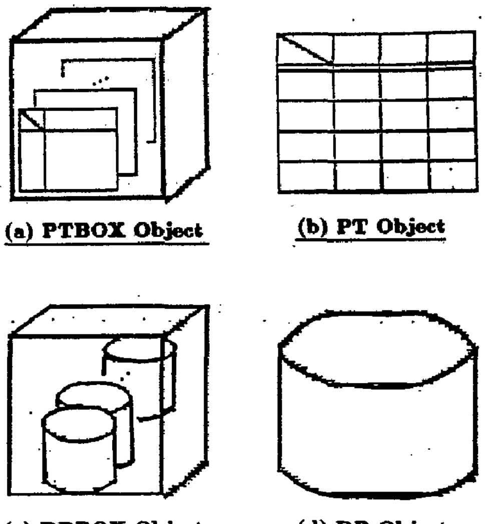


Figure 1 MELQUERY Software Configuration.



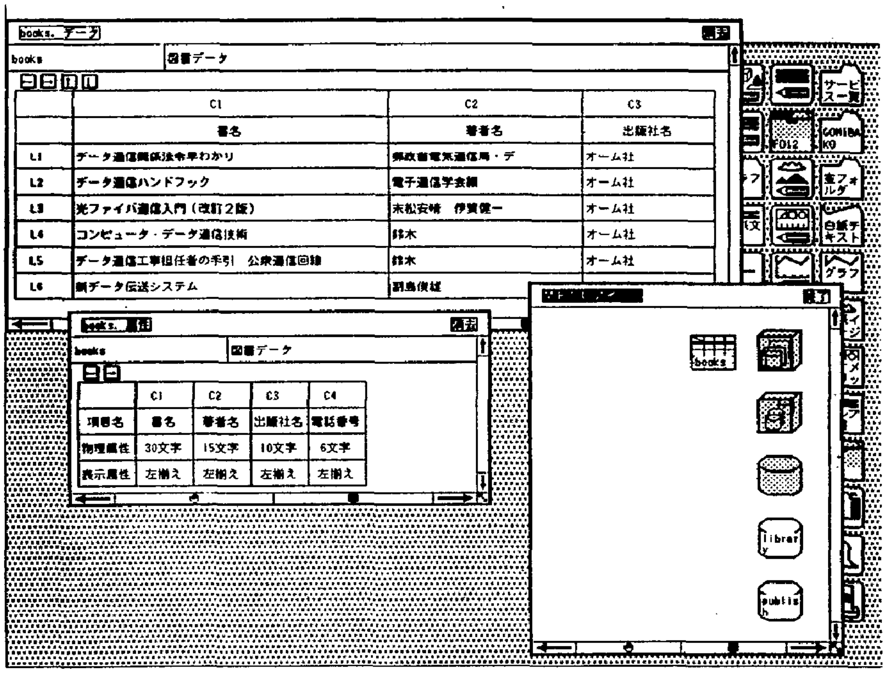
(a) PTBOX Object

(b) PT Object

(c) DBBOX Object

(d) DB Object

Figure 2 Iconic Displays of Objects.



books　図書データ

	C1	C2	C3
	書名	著者名	出版社名
L1	データ通信関係法令早わかり	郵政省電気通信局・デ	オーム社
L2	データ通信ハンドフック	電子通信学会編	オーム社
L3	光ファイバ通信入門（改訂２版）	末松安晴　伊賀健一	オーム社
L4	コンピュータ・データ通信技術	鈴木	オーム社
L5	データ通信工事担任者の手引　公衆通信回線	鈴木	オーム社
L6	新データ伝送システム	副島俊雄	

books　図書データ

	C1	C2	C3	C4
項目名	書名	著者名	出版社名	電話番号
物理属性	30文字	15文字	10文字	6文字
表示属性	左揃え	左揃え	左揃え	左揃え

Figure 3 An Example of Personal Table.

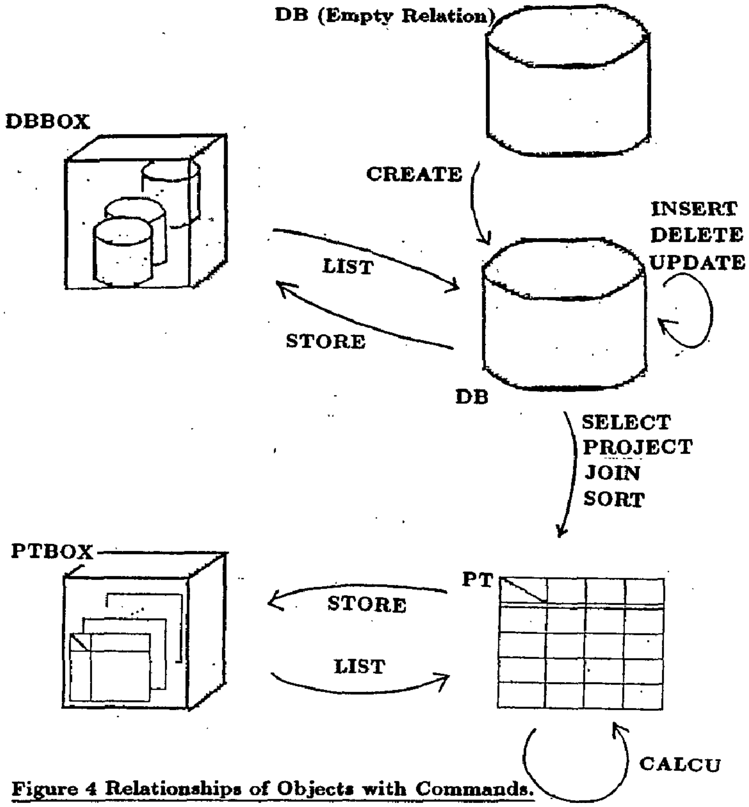
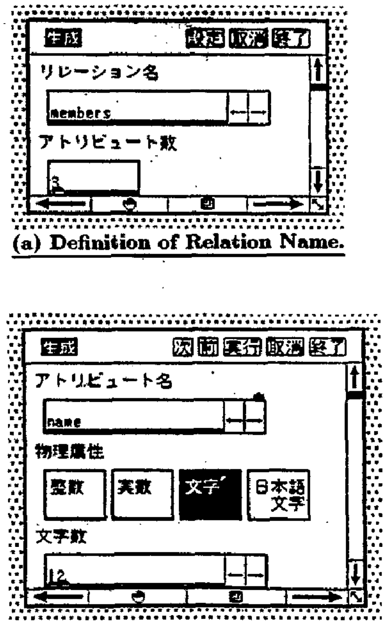


Figure 4 Relationships of Objects with Commands.



リレーション名

members

アトリビュート数

3

(a) Definition of Relation Name.

アトリビュート名

name

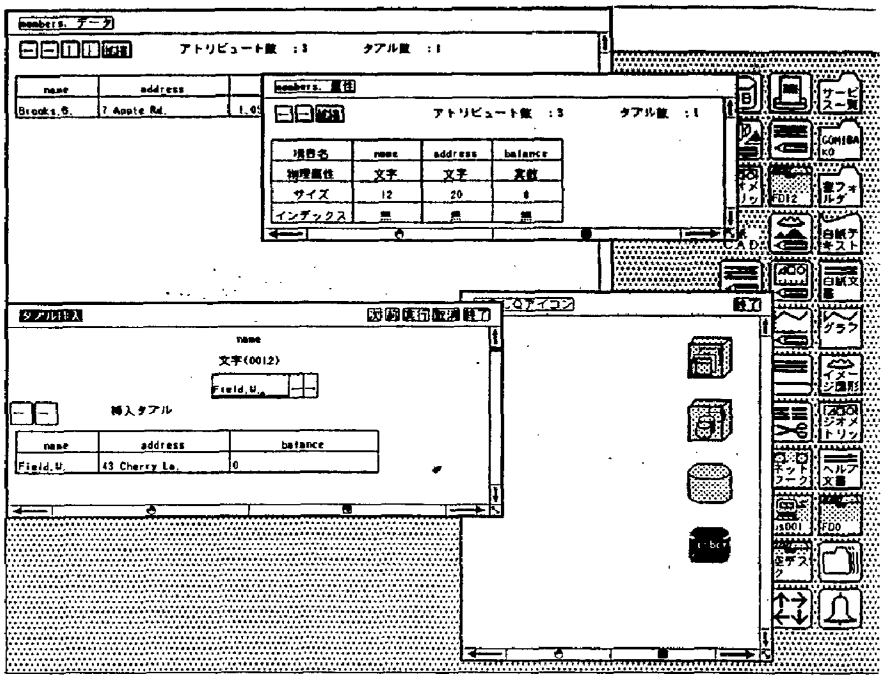
物理属性

文字数

12

(b) Definition of Attribute Name.

Figure 5 Creation of Relation.


members. データ

アトリビュート数 :3 タプル数 :1

name	address	
Brooks,G.	7 Apple Rd.	1.05

members. 属性

アトリビュート数 :3 タプル数 :1

項目名	name	address	balance
物理属性	文字	文字	実数
サイズ	12	20	8
インデックス	無	無	無

タプル挿入

name

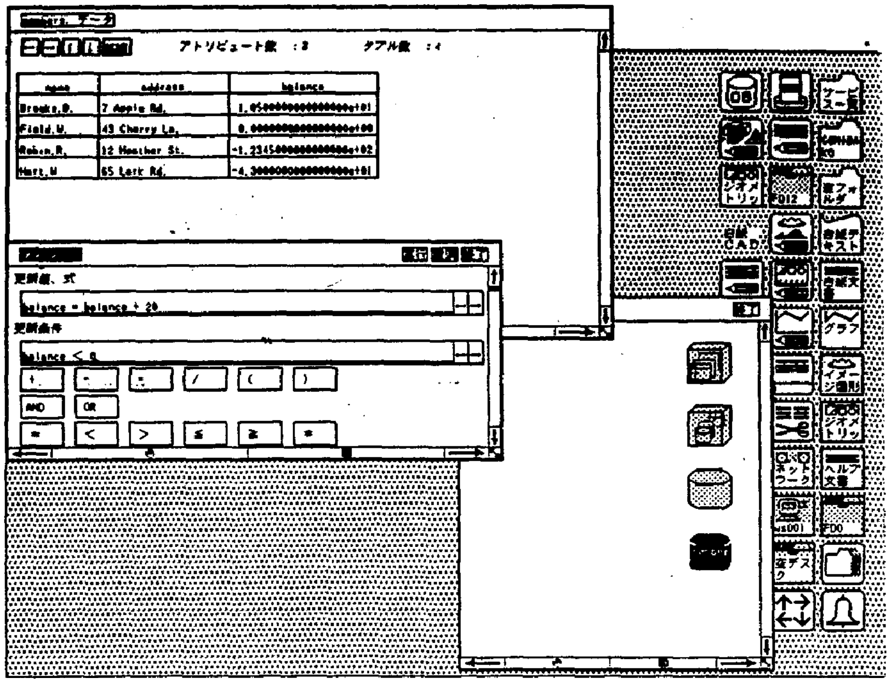
文字(0012)

Field,U.

挿入タプル

name	address	balance
Field,U.	43 Cherry La.	0

Figure 6 Insertion of Tuples.



members. データ

アトリビュート数 :3　　タプル数 :4

name	address	balance
Brooks,B.	7 Apple Rd.	1.050000000000000e+01
Field,W.	43 Cherry La.	0.000000000000000e+00
Robin,R.	12 Heather St.	-1.234500000000000e+02
Hart,W	65 Lark Rd.	-4.300000000000000e+01

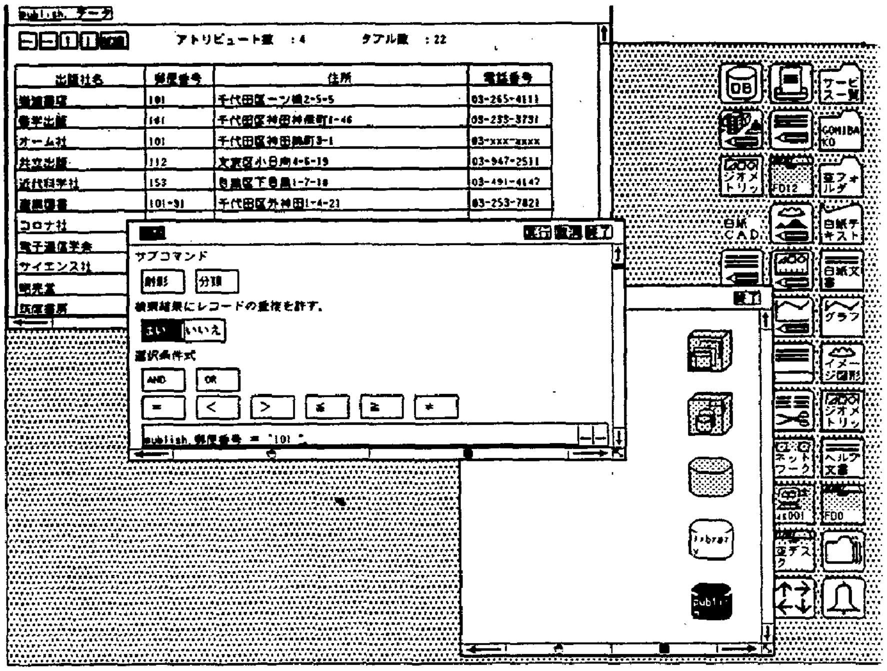
更新値、式

balance = balance + 20

更新条件

balance < 0

Figure 7 Updation of Relation.



publish. データ

アトリビュート数 :4　　タプル数 :22

出版社名	郵便番号	住所	電話番号
岩波書店	101	千代田区一ツ橋2-5-5	03-265-4111
啓学出版	101	千代田区神田神保町1-46	03-233-3731
オーム社	101	千代田区神田錦町3-1	03-xxx-xxxx
共立出版	112	文京区小日向4-6-19	03-947-2511
近代科学社	153	目黒区下目黒1-7-18	03-491-4147
[illegible]	101-91	千代田区外神田1-4-21	03-253-7821
コロナ社			
電子通信学会			
サイエンス社			
昭晃堂			
[illegible]			

サブコマンド

射影　分類

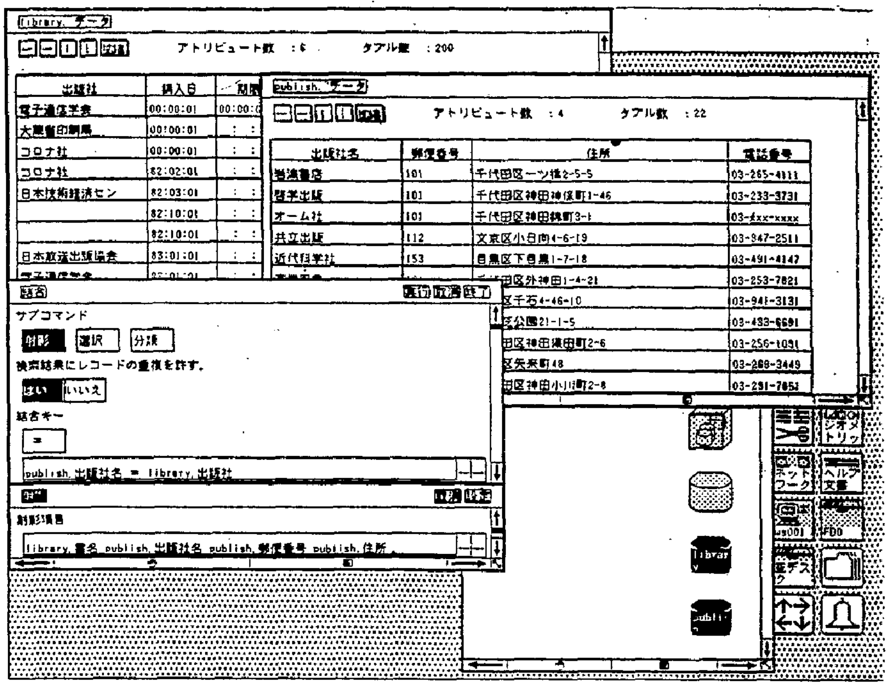
検索結果にレコードの重複を許す。

[✓ はい] いいえ

選択条件式

publish. 郵便番号 = "101 "

Figure 8 Retrieval of Relation.



library. データ

アトリビュート数 : 6　タプル数 : 200

出版社	購入日	期限
電子通信学会	00:00:01	00:00:0
大蔵省印刷局	00:00:01	: :
コロナ社	00:00:01	: :
コロナ社	82:02:01	: :
日本技術経済セン	82:03:01	: :
	82:10:01	: :
	82:10:01	: :
日本放送出版協会	83:01:01	: :

publish. データ

アトリビュート数 : 4　タプル数 : 22

出版社名	郵便番号	住所	電話番号
岩波書店	101	千代田区一ツ橋2-5-5	03-265-4111
啓学出版	101	千代田区神田神保町1-46	03-233-3731
オーム社	101	千代田区神田錦町3-1	03-xxx-xxxx
共立出版	112	文京区小日向4-6-19	03-947-2511
近代科学社	153	目黒区下目黒1-7-18	03-491-4147
		田区外神田1-4-21	03-253-7821
		区千石4-46-10	03-941-3131
		区公園21-1-5	03-433-6691
		田区神田須田町2-6	03-256-1091
		区矢来町48	03-268-3449
		田区神田小川町2-8	03-291-7851

結合　実行 取消 終了

サブコマンド

■射影 □選択 □分類

検索結果にレコードの重複を許す。

■はい □いいえ

結合キー

=

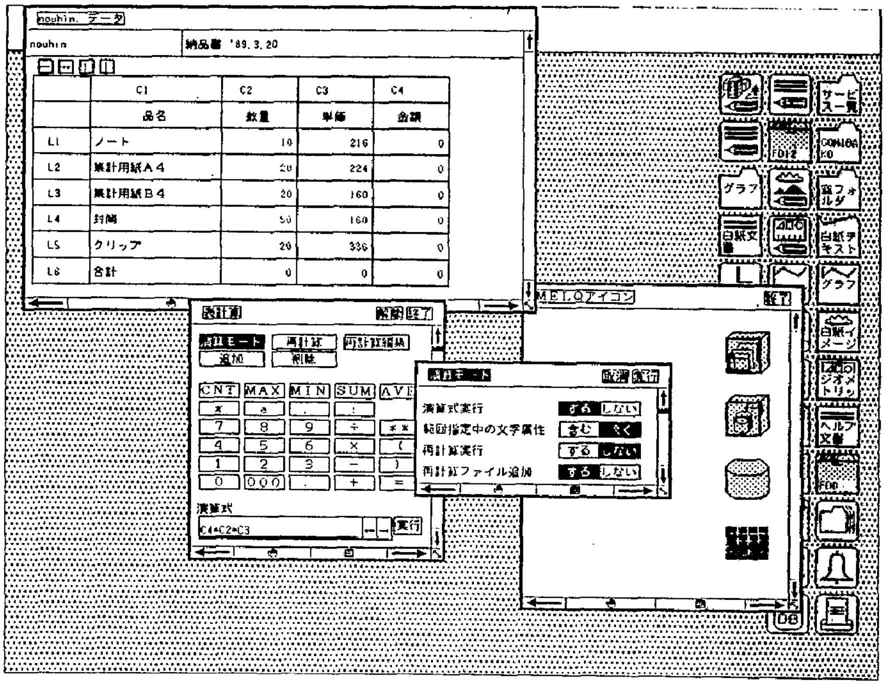
publish.出版社名 = library.出版社

射影　取消 設定

射影項目

library.書名 publish.出版社名 publish.郵便番号 publish.住所

Figure 9 Join of Relations and Projection.



nouhin. データ

nouhin　納品書 '89.3.20

	C1	C2	C3	C4
	品名	数量	単価	金額
L1	ノート	10	216	0
L2	集計用紙Ａ４	20	224	0
L3	集計用紙Ｂ４	20	160	0
L4	封筒	50	160	0
L5	クリップ	20	336	0
L6	合計	0	0	0

表計算　解除 終了

■演算モード □再計算 □再計算編集 □追加 □削除

演算式

C4*C2*C3　実行

演算モード　取消 実行

演算式実行	■する	□しない
範囲指定中の文字属性	□含む	■除く
再計算実行	□する	■しない
再計算ファイル追加	■する	□しない

Figure 10 Calculation of Personal Table.