APPLICATION OF THE RELATIONAL MODEL TO RECORD INDEXING AND RETRIEVAL

by

Jenn-Hann Liou

Memorandum No. ERL-M469

2 October 1974

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

APPLICATION OF THE RELATIONAL MODEL TO RECORD INDEXING AND RETRIEVAL

by

Jenn-Hann Liou

Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California, Berkeley, California 94720

ABSTRACT

Relational models are finding an increasing number of applications in data base management and information retrieval. In this report, a concrete application of the relational model to medical record management is analyzed in detail, and a special-purpose language for expressing requests for modular records or subrecords at an on-line video terminal is discussed. This model, then, may be used as a paradigm for application in other problem areas in which large volumes of information have to be searched efficiently for needed data.

Research supported by the Naval Electronics Systems Command, Contract N00039-71-C-0255/0034

[Medical Record Indexes]

In hospitals, the patient medical records are usually filed in the sequence of medical record numbers. Each patient is assigned a unique medical record number. Given a patient name the name index is searched to find his medical record number, which is then used to find the record out of the file. However, a well-managed medical record department maintains a few more indices, namely, the disease index, the operation index, the physician index, etc. These indices provide different access paths to the medical records. Through this facility, a physician or a medical staff committee can conveniently retrieve medical records for purposes like the following [1]:

1. To review previous cases of a given disease to provide insight into the management of a current patient's problem.
2. To locate a record when the physician remembers only the diagnosis and/or operation but not the patient's name.
3. To compare data on certain diseases and treatments to prepare medical reports.
4. To evaluate quality of care in the hospital.

On an index card the following items may be entered:

1. patient's medical record number
2. patient's name
3. patient's sex
4. patient's race
5. patient's birth date
6. date of admission
7. date of discharge
8. disease
9. operation
10. physician's name
11. surgeon's name
12. discharge condition

In the disease index, the disease name is printed at the top of the card. The cards are sorted in the alphabetical order of the disease names. Similar work has to be done to construct the name index, the physician index, and the operation index.

How many items should be put on an index card depends on how the index will be used. A physician may request a list of all cases of tuberculosis in black patients. If race is not on the disease index card, all records which record tuberculosis will be pulled. Thus increasing the number of data items on the index card reduces the retrieval cost that increases the indexing cost. Careful investigation should be made to find out the most frequently used items in order to minimize the overall cost.

These card indexes are expensive to maintain and time-consuming to

search. Just as computers have been used to aid bibliographical search and retrieval, so can they be used for medical record search and retrieval. The relational model introduced by E. F. Codd [2,3] has been found to provide a convenient way to manipulate the index data.

[Data Base]

The index data is stored in three tables T_1, T_2 and T_3 (Fig. 1), where:

a. Each patient is assigned a unique medical record number, and for each admission to the hospital he is given an admission number, which is unique for any patient and any admission.

b. AGE is the age at admission and is therefore functionally dependent on ADMISSION # instead of MR #.

c. For each admission there may be more than one disease diagnosed. There is a row in T_3 for each of the diseases.

d. If an operation was performed for a disease, they are put on the same row in T_3.

e. The DISCHARGE-CONDITION indicates the condition of each disease at discharge. It can be: 0(cured), 1(improved), 2(not improved), 3(worse), 4(the patient died of another disease), and 5(the patient died of this disease).

f. To facilitate computer processing, the disease names and operation names are stored in coded forms. In this paper, the SNDO [4] code is used. The code has the form ***-*** where each * is a symbol from the set {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, x, y}. For example, the disease code 680-160 represents "hepatitis due to virus". The part left of the hyphen indicates the site of disease, and the part right of the hyphen indicates the cause of disease. And in each part the digits, going from left to right, show greater specificity.

In 680, 6 means "digestive system", and 80 (when following '6')

T_1 =

MR#	NAME	RACE	BIRTH-DATE	SEX
01-56-213	John K. Doe	W	1/ 9/20	M
25-17-164	Geo M. Blank	W	3/16/35	M
08-94-549	Robert J. Brown	B	10/ 3/22	M
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

T_2 =

ADMISSION #	MR #	ADMISSION - DATE	DISCHARGE - DATE	AGE
527384	25-17-164	9/19/69	10/18/69	34
637926	39-07-615	3/ 2/70	4/31/70	55
674291	01-56-213	6/28/70	8/20/71	50
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

T_3 =

ADMISSION #	DISEASE	PHYSICIAN	OPERATION	SURGEON	DISCHARGE-CONDITION
674291	680-160	06-19	-	-	2
674291	640-915	06-19	640-10	16-32	0
674291	460-782	04-07	-	-	1
527384	361-186	03-02	-	-	0
688745	661-100	06-24	661-12	16-11	0
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

FIG. 1

means 'liver' (an organ of the digestive system) where 0 can be substituted by other symbols to indicate a specific region of liver.

In 160, 1 means "lower plant or animal", and 60 (when following '1') means "virus", where again 0 can be substituted by other symbols to mean a specific kind of virus.

In the operation code, the part left of the hyphen indicates the site of operation, while the part right of the hyphen indicates the operative procedure.

For the same purpose, each physician and surgeon is also given a code.

g. At each discharge of a patient, a medical record clerk extracts data from the discharge summary in the medical record, encodes the diseases and operations if necessary, and adds one or more rows to T_2 and T_3 (and T_1 if this is the first admission of the patient).

Using the natural-join operation [5], the three tables T_1, T_2 and T_3 can be combined into a large table T (Fig. 2), which contains exactly the same amount of information as T_1, T_2 and T_3 together.

MR#	NAME	RACE	BIRTH-DATE	SEX	ADMISSION #	ADMISSION-DATE	DISCHARGE-DATE	AGE	DISEASE	PHYSICIAN	OPERATION	SURGEON	DISCHARG CONDITIO
01-56-213	John K. Doe	W	1/ 9/20	M	674291	6/28/70	8/20/71	50	680-160	06-19	-	-	2
01-56-213	John K. Doe	W	1/ 9/20	M	674291	6/28/70	8/20/71	50	640-951	06-19	640-10	16-32	0
01-56-213	John K. Doe	W	1/ 9/20	M	674291	6/28/70	8/20/71	50	460-782	04-07	-	-	1
25-17-164	Geo M. Blank	W	3/16/35	M	527384	9/19/69	10/18/69	34	361-186	03-02	-	-	0
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.

FIG. 2

[Query Language]

A series of papers by Codd, et al. have defined languages [6,7] for expressing queries on a relational data base. In the case described in this report, tables and attributes (the titles of the columns in the tables) are fixed, and most of the time, the user works at an on-line video terminal. To take full advantages of these and to meet the special objectives of medical record department services, the following examples illustrate a special-purpose language for retrieving information out of the medical record index data base.

EX. 1. A physician wants to find previous cases of female patients between ages of 30 and 40 on whom a gastrectomy was performed.

In Fig. 3, the user input is underlined to distinguish it from the response of the computer. He first types in "START SEARCH", and the computer displays the attributes immediately. He then enters his request (Fig. 3(a)). After 'END' is entered, the computer responds to the request by displaying "SEARCHING" to tell the user that it understands the request. The computer selects those rows in T that contain 'F' for SEX, AGE between 30 and 40, and '640-10' (code of GASTRECTOMY) for OPERATION. The MR #'s on these rows are collected and counted. Then the computer tells the user there are 153 MR #'s satisfying the requirement (Fig. 3(b)). The user decides that this is too many. He types in "BACK UP" and the original request is displayed (Fig. 3(c)). He adds the restriction "$\geq$ 1/1/72" to ADMISSION-DATE, and is told this time 9 MR #'s satisfy the search requirement (Fig. 3(d)). He types "GO" and the list of MR #'s is displayed. He could have used "PRINT MR #" instead of "DISPLAY MR #" to request for a hard copy output.

```
START SEARCH

(SELECTION)
     MR #                  :
     NAME                  :
     RACE                  :
     BIRTH-DATE            :
     SEX                   :  F
     ADMISSION #           :
     ADMISSION-DATE        :
     DISCHARGE-DATE        :
     AGE                   :  (30, 40)
     DISEASE               :
     PHYSICIAN             :
     OPERATION             :  GASTRECTOMY
     SURGEON               :
     DISCHARGE-CONDITION   :

(OUTPUT)
DISPLAY MR #
END
```

FIG. 3(a)

```
SEARCHING
NUMBER OF ITEMS  :  153
BACK UP
```

FIG. 3(b)

```
START SEARCH
(SELECTION)
     MR #                  :
     NAME                  :
     RACE                  :
     BIRTH-DATE            :
     SEX                   :  F
     ADMISSION #           :
     ADMISSION-DATE        :  ≥ 1/1/72
     DISCHARGE-DATE        :
     AGE                   :  (30, 40)
     DISEASE               :
     PHYSICIAN             :
     OPERATION             :  GASTRECTOMY
     SURGEON               :
     DISCHARGE-CONDITION   :

(OUTPUT)
DISPLAY MR #
END
END
```

FIG. 3(c)

```
SEARCHING
NUMBER OF ITEM  :  9
GO
```

FIG. 3(d)

```
(SELECTION)
     SEX              :  F
     ADMISSION-DATE   :  ≥ 1/1/72
     AGE              :  (30, 40)
     OPERATION        :  GASTRECTOMY

(OUTPUT)
          MR #
     14-06-053
     28-96-502
     01-45-102
     27-69-453
     00-91-287
     34-21-319
     18-24-275
     13-05-076
     05-47-418

END OF OUTPUT
```

FIG. 3(e)

EX. 2. A medical record clerk wants to tabulate the age (in decades) against number of admissions of male patients after 1/1/71 for any digestive system problem in the range of admission age between 11 and 60.

Figure 4(a) shows his request. The ! in 6!! - !!! denotes "don't care". All rows in T with SEX = M, ADMISSION-DATE $\geq$ 1/1/71 and disease code starting with 6 are selected. EACH-OF (11, 60; 10) further discards the selected rows with AGE <11 or >60 and divides the rest of rows into five groups each corresponding to a decade of admission age. (11 and 60 are the lower bound and upper bound of the range and 10 is the interval length.) NUMBER-OF (ADMISSION #) counts the number of ADMISSION #'s in each group of rows and the result is printed (Fig. 4(b)).

```
START TABULATION

(SELECTION)
     MR #                 :
     NAME                 :
     RACE                 :
     BIRTH-DATE           :
     SEX                  :  M
     ADMISSION #          :
     ADMISSION-DATE       :  ≥ 1/1/71
     DISCHARGE-DATE       :
     AGE                  :  EACH-OF (11, 60; 10)
     DISEASE              :  6!! - !!!
     PHYSICIAN            :
     OPERATION            :
     SURGEON              :
     DISCHARGE-CONDITION  :

(OUTPUT)
PRINT AGE, NUMBER-OF (ADMISSION #)
END
SEARCHING
```

FIG. 4(a)

```
(SELECTION)
     SEX              :  M
     ADMISSION-DATE   :  ≥ 1/1/71
     DISEASE          :  6!! - !!!

(TABLE)

        AGE          NUMBER OF (ADMISSION #)
       11-20                  109
       21-30                  251
       31-40                  342
       41-50                  169
       51-60                  238

END OF OUTPUT
```

FIG. 4(b)

If a question mark "?" follows EACH-OF, then the selected rows are divided by each instance that can substitute the "?". Slightly change the request in EX. 2, the result is shown in Fig. 5(a) and (b).

```
START TABULATION

(SELECTION)
     MR #                  :
     NAME                  :
     RACE                  :
     BIRTH-DATE            :
     SEX                   :  M
     ADMISSION #           :
     ADMISSION-DATE        :  ≥ 1/1/71
     DISCHARGE-DATE        :
     AGE                   :  EACH-OF (11, 60; 10)
     DISEASE               :  EACH-OF 6?! - !!!
     PHYSICIAN             :
     OPERATION             :
     SURGEON               :
     DISCHARGE-CONDITION   :  EACH-OF ?

(OUTPUT)
PRINT AGE, DISEASE, DISCHARGE-CONDITION, NUMBER-OF (ADMISSION #)
END
END
```

FIG. 5(a)

(SELECTION)

SEX : M

ADMISSION-DATE : ≥ 1/1/71

(TABLE)

AGE	DISEASE	DISCHARGE - CONDITION	NUMBER-OF (ADMISSION #)	
11-20	60!-!!!	0	2	
		1	11	
		2	0	
		3	9	
		4	4	
		5	3	
	61!-!!!	0	0	
		1	0	
		2	0	
		3	7	sum up to 109
		4	1	
		5	2	
	62!-!!!	0	5	
	.	.	.	
	.	.	.	
	.	.	.	
	.	.	.	
	69!-!!!	0	4	
		.	.	
		.	.	
		.	.	
21-30	60!-!!!	.	.	
.	.	.	.	
.	.	.	.	
.	.	.	.	
.	.	.	.	
.		.	.	
.	.	.	.	
51-60	60!-!!!	.	.	
	.	.	.	
	.	.	.	
	.	.	.	
	.	.	.	
		5	12	

END OF OUTPUT

FIG. 5(b)

[Discussion]

Many efforts have been made to organize the data in the patient medical record more logically [8,9], and to store it in the computer [10]. Although the work described in this report is independent of the computerization of the medical record, it should be noted that: After medical records are computerized, at each patient discharge, the discharge summary part of the patient's medical record is examined by the computer; and data items are extracted from the discharge summary and stored into the index data base automatically. This will greatly reduce the indexing cost. Also, if the medical record consists of subrecords, each for an admission, then admission numbers can be used to retrieve subrecords.

Although the indexing of medical records is illustrated in this report, it is easy to find other problem areas to which the same model can be readily applied. For example, an insurance company may want to keep a record of each customer. An index table can be constructed, where the attributes are the customer's age, profession, income, etc. This table can then be used to search for record(s) and to obtain important statistics needed in determining the company policies.

ACKNOWLEDGEMENT

The author wishes to express his gratitude to his research adviser, Professor Lotfi A. Zadeh, for his constant and helpful guidance. Thanks are also due to Mr. Rowland Johnson for his careful reading of the draft of this report.

REFERENCES

1. Huffman, E. K., "Medical Record Management," 1972, Physician's Record Company, Berwyn Illinois.
2. Codd, E. F., "A Relational Model of Data for Large Shared Data Banks," CACM, vol. 13, no. 6, June 1970, 377-387.
3. Codd, E. F., "Further Normalization of the Data Base Relational Model," Courant Science Symposia, vol. 6, Data Base Systems, Prentice-Hall, New York, May 1971.
4. American Medical Association,"SNDO, Standard Nomenclature of Diseases and Operations," 5th edition, 1961, McGraw-Hill.
5. Codd, E. F., "Relational Completeness of Data Base Sublanguages," 1972, IBM Research Laboratory, San Jose.
6. Codd, E. F., "A Data Base Sublanguage Founded on the Relational Calculus," 1971, IBM Research Laboratory, San Jose.
7. Chamberlin, D. D., "SEQUEL: A Structured English Query Language," 1974, IBM Research Laboratory, San Jose.
8. Weed, L. L., "Medical Records, Medical Education, and Patient Care," 1969, the press of Case Western Research University.
9. Walker, Hurst and Woody, "Applying the Problem-Oriented System," 1973, Medcom Press.
10. Davis, L. S., "Prototype for Future Computer Medical Records," 1970, Computers and Biomedical Research, 3, 539-554.