DATABASE MANIPULATIONS AS RELATION-TYPE OPERATIONS

Ivan Mezník
Faculty of Business and Management,
Institute of Informatics, Brno University of Technology, Brno, Czech Republic

Abstract

The paper points to highly useful symbiosis of manipulations in databases and operation on relation-type mathematical structures. The common context of *n*-ary relation tools and database structures means is discussed and the correspondence between *n*-ary relation operations and standard database constructions is examined. Also some relevant mathematical problems are pointed out.

Keywords: N-ary relation, relation data model, projection, selection, join, *E*-join, θ-join

Introduction

Relationships among elements of a number of sets often arise in real-life events and lead to investigation of important properties of discrete algebraic structures. For instance, there is a relationship among the name of the passenger, the name of the aircraft carrier, flight number, departure point, destination, departure time, and arrival time. Such relationships may be expressed in terms of *n-ary relations.* In [5] and [6] the authors dealt with the basic aspects how they can be used to represent in a unified way computer databases. These representations evidently help answer the questions about the information stored in databases. Also they point to the correspondence between some database information manipulations and *n*-ary relation operations. In the sequel most of standard database information manipulations is described using *n*- ary relation tools. The paper is organized as follows. First preliminary concepts on relations and relevant database terminology are reviewed. In the following main part database manipulations and corresponding relation operations are investigated.

Preliminaries

The *ordered n-tuple* (shortly *n-tuple*)**,** denoted by $(a_1,...,a_n)$, is the ordered collection of elements that has a_1 as its first element, a_2 as its second element,…, and a_n as its *n*th element.

Let $A_1,...A_n$ be finite sets. The *Cartesian product* of the sets $A_1,...,A_n$, denoted by $A_1xA_2x...xA_n$, is the set of *n*-tuples $(a_1,...,a_n)$, where a_i belongs to A_i for $i=1,...,n$. In symbols,

$$A_1xA_2x...xA_n = \{(a_1,...,a_n) | a_i \in A_i \text{ for } i=1,...,n\}.$$

An *n-ary relation* on sets $A_1,...A_n$, denoted by *R,* is any subset of their Cartesian product, ie.

$$R \subseteq A_1x...xA_n.$$

The sets $A_1,...A_n$ are the *domains* of *R* and *n* is its *degree*. In a special case $n = 2$, denoting $A_1 = A, A_2 = B$, we speak about a *binary relation from A to B* and if moreover $A = B$ about a *relation on A*; in case *n=3* also the word *ternary* is used.

In order to manipulate information in a database efectively (the time is the most decisive factor), various methods for representing databases have been proposed. One of the most important methods, based on the concept of an *n*-ary relation, is said to be the *relation data model.* In the relevant terminology ([2], [4] among others) a database consists of *records*, which are *n*-tuples. The entries of the *n*-tuples are called *fields.* In this manner the relational data model represents a database of records as an *n*-ary relation. Since relations representing databases are often displayed in a table form, they are said to be *tables***.** With a view to the definition of a relation, records are elements of the relation and fields are its domains.

Information Manipulations in Databases as n-ary Relation Operations

There are essentially two types of operations with *n*-ary relations useful to describe information manipulations in databases. The first type concerns operations based on standard set operations with fruitful applications in construction of new databases (*union, intersection, diference, Cartesian product*). The second type may be characterized as operations that are virtually motivated by the aspects of desirable information manipulations (*projection, join, selection*). Besides the mentioned operations there are a variety of further special operations (special *join-type operations* among others) utilized in database theory.

U n i o n, i n t e r s e c t i o n, d i f f e r e n c e

Let R, S be *n*-ary relations on $A_1,..., A_n$. Since both are subsets of $A_1 x A_2 x...x A_n$, they can be combined in any way two sets are traditionally treated. Apparently, the resulting set will be again an *n*-ary relation on $A_1,..., A_n$. The *union* of *R* and S is the *n*-ary relation *T* = $R \cup S$. The *intersection* of *R* and S is the *n*-ary relation $I = R \cap S$. The *difference* of *R* and S is the *n*-ary relation $D = R - S$.

Example Let *R* and *T* be 3-ary (ternary) relations on *N*(*Student Number*), *S*(*Student Surname*), *M*(*Major*) given as databases of records by the following Tables 1 and 2:

N	*S*	*M*
1	Novak	History
2	Vrana	Physics
3	Thomas	Maths
5	Barta	Economy

Table 1 Ternary relation *R*

N	*S*	*M*
1	Novak	History
2	Vrana	Economy
4	Brown	Maths
2	Vrana	Physics
6	Kabat	Music

Table 2 Ternary relation *T*

Then the ternary relations $R \cup T, R \cap T, R - T, T - R$ are given as databases of records by the following Tables 3, 4, 5, 6.

	S	*M*
1	Novak	History
2	Vrana	Physics
2	Vrana	Economy
5	Barta	Economy
4	Brown	Maths
6	Kabat	Music

Table 3 Ternary relation $R \cup T$

N	*S*	*M*
1	Novak	History
2	Vrana	Physics

Table 4 Ternary relation $R \cap T$

	S	*M*
3	Thomas	Maths
5	Barta	Economy

Table 5 Ternary relation $R-T$

N	*S*	*M*
2	Vrana	Economy
4	Brown	Maths
6	Kabat	Music

Table 6 Ternary relation $T-R$

In words, database corresponding to $R\cup T$ contains records that are in Table 1 or Table 2 (in case that a record is contained in both, in the resulting database appears only ones), database corresponding to $R\cap T$ records that are simultaneously in both Tables 1 and 2, database corresponding to $R-T$ records that are in Table 1 but not in Table 2, database corresponding to $T-R$ records that are in Table 2 but not in Table 1. Notice that in all cases the resulting tables are of the same structure.

Cartesian product

Let *R* be an *m*-ary relation on $A_1,...,A_m$, *S* an *n*-ary relation on $B_1,...,B_n$. The *Cartesian product* of relations *R* and *S* , denoted by RxS, is an (*m*+*n*)-ary relation on sets $A_1,...,A_m,B_1,...,B_n$ such that $(a_1,...,a_m,b_1,...,b_n)\in RxS$ if $(a_1,....a_m)\in R$ and $(b_1,...,b_n)\in S$. It is apparent, that database corresponding to RxT contains records made up by connecting every row of the table corresponding to *R* with every row of the table corresponding to *S*. The resulting table corresponding to RxS will contain the number of columns that equals the sum of columns in corresponding databases and the number of the records in the table corresponding to RxS equals the product of the number of records in both databases.

Projection

Let *R* be an *n*-ary relation on sets $A_1,...,A_n$ and $k\le n$. The ($i_1,...,i_k$)-*projection* of *R*, denoted by $R_{i_1,...i_k}$, is a *k*-ary relation on sets $A_{i_1},...,A_{i_k}$ defined by if ($a_1,...,a_n)\in R$ then ($a_{i_1},...,a_{i_k})\in R_{i_1,...i_k}$.

Verbally, the $R_{i_1,...i_k}$ projection is obtained by deleting ($n-k$) components of each *n*-tuple ($a_1,...,a_n)\in R$ leaving the i_1th,i_2th,…,i_kth components. When the relation *R* is given by the database of records in a table form(with *n* columns), then the resulting table of $R_{i_1,...i_k}$ projection will have *k* columns. Notice that fewer rows may result-this happens when some of the *n*-tuples in the relation *R* have identical values in each of the *k* components of the projection and only disagree in components deleted by the projection.

Example Let *R* be a 5-ary relation on sets *N*(*Student number*), *S*(*Student surname*), *M*(*Major*), *P*(*Professor*) , *L*(*Lecture room*) given as database of records by the following Table 7. Then its projection $R_{3,4}$ is the binary relation shown in Table 8.

N	*S*	*M*	*P*	*L*
1	Novak	History	Kren	384
2	Vrana	Physics	Moor	384
3	Thomas	Maths	Ross	384
5	Barta	Economy	Dale	384
3	Thomas	History	Kren	381
1	Novak	Physics	Moor	381

5	Barta	Maths	Roos	381
1	Novak	Economy	Dale	381

Table 7 5-ary relation *R*

M	*P*
History	Kren
Physics	Moor
Maths	Ross
Economy	Dale

Table 8 Binary relation $R_{3,4}$

Selection

Let *R* be an *n*-ary relation on $A_1,...,A_n$ and θ boolean condition containing sets $A_1,...,A_n$ or their elements respectively. The *selection* R_θ of *R* is an *n*-ary relation on $A_1,...,A_n$ that consists of all *n*-tuples of *R* for which the condition θ holds true. Verbally, the result of selection R_θ is the restriction of *R* in the sense that some rows of the table are omitted according to condition θ.

Example Let *R* be a ternary relation on *N*(*Student Number*), *S*(*Student Surname*), *M*(*Major*) given as databases of records by the following Table 9 and θ: *S* = VRANA. Then the resulting relation R_θ is given by the Table 10.

N	S	M
1	Novak	History
2	Vrana	Economy
4	Brown	Maths
2	Vrana	Physics
6	Kabat	Music

Table 9 Ternary relation *R*

N	*S*	*M*
2	Vrana	Economy
2	Vrana	Physics

Table 10 Ternary relation R_θ

Join

Let *R* be an *m*-ary relation on $A_1,...,A_m$, *S* an *n*-ary relation on $B_1,...,B_n$. The *join* of *R, S*, denoted by $J_p(R,S)$, where $p \le m, p \le n,$ is a $(m+n-p)$ -ary relation that consists of all $(m+n-p)-$tuples for which there exist *m*-tuple $(a_1,...,a_{m-p},c_1,...,c_p) \in R$ and *n*-tuple $(c_1,...,c_p,b_1,...,b_{n-p}) \in S.$

Verbally, the result of the join operation is a new relation from two given relations by combining all *m*-tuples of the first relation with all *n*-tuples of the second relation, where the last *p* components of the *m*-tuples coincide with the first *p* components of the *n*-tuples. This operation is used to put together two tables that share some identical fields.

Example Let *R* be a 5-ary relation given by Table 7 and S be a 4-ary relation on sets $M(Major), P(Professor), L(Lecture\ room), C(Credits)$ given by the following Table 11. Then the join of *R,S*, $J_3(R,S)$ is shown in Table 12.

N	*S*	*M*	*P*	*L*	*C*
1	Novak	History	Kren	384	6
2	Vrana	Physics	Moor	384	8
3	Thomas	History	Kren	381	6

M	*P*	*L*	*C*
History	Kren	384	6
Physics	Moor	384	8
History	Kren	381	6
Maths	Ross	381	8
Economy	Dale	381	6

5	Barta	Maths	Roos	381	8
1	Novak	Economy	Dale	381	6

Table 11 4-ary relation *S* Table 12 6-ary relation $J_3(R,S)$

E-Join

Let *R* be an *m*-ary relation on $A_1,...,A_m$, *S* an *n*-ary relation on $B_1,...,B_n$ and suppose that $A_i = B_j$ = *A* for some *i*=1,…,*m*, *j*=1,…,*n*. The *E-join* of *R*, *S* with respect *A*, denoted by $J_E(R,S)$ is a $(m+n)$-ary relation on $A_1,...,A_m,B_1,...,B_n$ that consists of all (*m*+*n*)-tuples for which the values of the common set *A* are equal. Verbally, the result of *E*-join is a new relation constructed by combining all *m*-tuples of the first relation with all *n*-tuples of the second relation, for which the values of common set (attribute) *A* are equal (this motivates the concept and its notation).

Example Let *R* be a 4-ary relation on sets $M(Major), P(Professor)$, $L(Lecture\ room)$, $C(Credits)$ given by the following Table 13, *S* a 6-ary relation on sets *N*(*Student number*), *S*(*Student surname*), *M*(*Major*), *P*(*Professor*) , *L*(*Lecture room*), *C*(*Credits*) given by Table 14. Then the *E*-join of *R*,*S* with respect to *Professor*(from *R*)=*Professor*(from *S*), $J_E(R,S)$ is shown in Table 15.

M	*P*	*L*	*C*
History	Kren	384	6
Physics	Moor	384	8
History	Kren	381	6
Maths	Ross	381	8
Economy	Dale	381	6

Table 13 4-ary relation *R*

N	*S*	*M*	*P*	*L*	*C*
1	Novak	History	Kren	384	6
2	Vrana	History	Moor	384	8
3	Thomas	History	Smith	381	6
5	Barta	Maths	Moor	381	8
1	Novak	Economy	Lear	381	6

Table 14 6-ary relation *S*

M	P	L	C	N	S	M	P	L	C
History	Kren	384	6	1	Novak	History	Kren	384	6
Physics	Moor	384	8	2	Vrana	History	Moor	384	8
Physics	Moor	384	8	5	Barta	Maths	Moor	381	8
History	Kren	381	6	1	Novak	History	Kren	384	6

Table 15 10-ary relation $J_E(R,S)$

θ-Join

Let *R* be an *m*-ary relation on $A_1,...,A_m$, *S* an *n*-ary relation on $B_1,...,B_n$. Further, let $A_i = B_j$ =*A* for some *i*=1,…,*m*, *j*=1,…, *n* and *θ* be a boolean condition containing values of the common attribut *A*. The *θ-join* of *R*, *S* with respect to *A* and *θ*, denoted by $J_\theta(R,S)$ is a $(m+n)$-ary relation on $A_1,...,A_m,B_1,...,B_n$ that consists of all (*m*+*n*)-tuples for which the values of the common set *A* satisfy condition *θ*. Verbally, the result of *θ*-join is a new relation constructed by combining all *m*-tuples of the first relation with all *n*-tuples of the second relation, for which the values of common set (attribute) *A* satisfy condition *θ*.

Example Let *R*,*S* be relations given by Tables 13, 14.. Then their *θ-join*, $J_\theta(R,S)$, with respect to *L* and *θ*: *Lecture room*(from *R*) ≠ *Lecture room*(from *S*) is shown in Table 16.

M	P	L	C	N	S	M	P	L	C
History	Kren	384	6	5	Barta	Maths	Moor	381	8
History	Kren	384	6	3	Thomas	History	Smith	381	6
History	Kren	384	6	1	Novak	Economy	Lear	381	6
Physics	Moor	384	8	3	Thomas	History	Smith	381	6
Physics	Moor	384	8	1	Novak	Economy	Lear	381	6
Physics	Moor	384	8	5	Barta	Maths	Moor	381	8
History	Kren	381	6	1	Novak	History	Kren	384	6
History	Kren	381	6	2	Vrana	History	Moor	384	8
Maths	Ross	381	8	1	Novak	History	Kren	384	6
Maths	Ross	381	8	2	Vrana	History	Moor	384	8
Economy	Dale	381	6	1	Novak	History	Kren	384	6
Economy	Dale	381	6	2	Vrana	History	Moor	384	8

Table 16 10-ary relation $J_\theta(R,S)$

Remark Obviously E-join operation is as a special case of θ-join operation. With a view to very frequent use of E-join in applications it is usually treated separately

Conclusions

Informatics besides mathematics plays undoubtedly an integrating role in all with real-life occupying disciplines. The progress in informatics is primarily determined by new technologies and particularly by the development of software engineering. The symbiosis between mathematics and informatics initiated historically computing processes. The present total influence of computers to all spheres of life together with free access of all individuals to computers, information nets and sources shifts the essence of such symbiosis strongly to logical processes. From the viewpoint of a current user the logic is naturally (sometimes unknowingly) employed when manipulating and browsing in databases. For more sophisticated approach mathematical tools to perform operations on databases are advisable. It may evidently help to answer queries about the information stored in databases. The use of extensive relation algebra tools may be beneficial to solve important problems in database theory. For instance, the testing procedures for composite keys, the properties of composite keys with respect to database operations and optimization problems. It may also set problems concerning algebraic properties of the operations on the special types of n-ary relations motivated by database manipulations.

References:

Chaffey, D.: *Business Information Systems.* Pearson Education Limited, Essen 2003

Date, C.J.: *An Introduction to Database systems.* Addison-Wesley, New York 1990

Hawryszkiewycz, I.T.: *Database Analysis and Design.* Maxwell-Macmillan, London 1991

Koch, M.: *Datové a funkční modelování.* Akademické nakladatelství CERM, s.r.o., Brno 2006 (in Czech)

Koch, M., Mezník, I.: *n-ary Relation Operations on Databases.* In: Proc. of the Ninth International Conference "Mathematics Education in a Global Community" (Ed.D.K. Pugalee, A.Rogerson and A. Schinck), The University of North Carolina , Charlotte, September 7-12, 2007, 370-376

Koch, M., Mezník, I.: *Special Database Constructions as Relation Operations.* In: Proc. of the 9th Annual International Conference on Business, Honolulu, Hawaii, June 12-15, 2009

Mesarovic, M. D., Takahara, Y.: *General Systems Theory: Mathematical Foundations*. Academic Press, New York 1975

Riordan, R.M.: *Vytváříme relační databázové systémy.* Computer Press, Praha 2000 (in Czech)
Rosen, K.H.: *Discrete Mathematics and Its Applications.* McGraw/Hill, New York 1999
Wymore, A. W.: *A Mathematical Theory of Systems Engineering- The Elements.*Wiley, New York 1969