n-ary Relation Operations on Databases

Miloš Koch, Ivan Mezník
Faculty of Business and Management,
Institute of Informatics, Brno University of Technology, Kolejní 2906/4, 61200 Brno, Czech Republic koch@fbm.vutbr.cz, mezník@fbm.vutbr.cz

Abstract
The motivations possess a crucial importance in teaching of mathematics. An instructor should permanently come up with new ideas and invent further sources. For this purpose the applications of mathematics in other disciplines play a key role. Besides traditional natural science subjects, a strong partner has appeared in the last decades, namely informatics. The paper points to highly influencing role of manipulations with databases in utilizing and investigating relation-type mathematical structures. In the paper 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.

Introduction
Relations between two sets and specially on a set are traditionally studied leading to investigation of important properties of discrete structures(e.g. ordering, equivalence, tolerance). They provide a lot of interpretations and applications for modelling of real-life situations and offer useful advanced material for mathematics secondary school lecturing. But also relationships among elements of more than two sets often arise. There is a relationship involving the name of the scholar, the scholar s address, the scholar s date of birth and the scholar s place of birth.. Similarly, 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. An example of such relationship in mathematics involves three sets where the first set is a subset of the second set, which is a subset of the third one. These relationships may be expressed in terms of so called *n-ary relations.* We will deal with a subject how they can be used to represent computer databases. These representations evidently help answer the questions about the information stored in databases. Firstly, we present a comparative view on *n*-ary relations tools and database terms and then we will examine the correspondence between database information manipulations and *n*-ary relation operations.

***n*-ary Relations and Databases**
First some preliminary definitions:
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. Two *n*-tuples are **equal**, if each corresponding pair of their elements is equal.
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,

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

Now, the definition of an *n*-ary relation may follow:
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. For our next consideration the generalization of the concept of *functionality* of a binary relation plays a key role. A binary relation *R* from *A* to *B* is **functional**, if for every $a \in A$ there is at most one $b \in B$ such that $(a,b) \in R$.

Example 1 Let *R* be the relation on sets A_1, A_2, A_3, where $A_1 = A_2 = A_3 = N$, N is the set of natural numbers, given as follows: $(a_1, a_2, a_s) \in R$ if $a_1 \geq a_2 \geq a_3$. Then, for instance, it holds $(10,7,5) \in R$, whereas $(9,10,3) \notin R$.

Example 2 Let *R* be the relation of degree 4 consisting of 4-tuples (N, B, F, U), where *N* is the name(of a person), *B* is the branch of study, *F* is the faculty, *U* is the university defined as follows: $(n,b,f,u) \in R$ if student *a* studies branch *b* at faculty *f* of university *u.*

Example 3 Let *A* be the set of men, *B* the set of women. Define a relation *R* from *A* to *B* : $(a,b) \in R$ if *a* is husband of *b.* Then *R* is functional only in monogamy countries.

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, a database consists of **records**, which are *n*-tuples. The entries of the *n*-tuples are so called **fields.** In this manner the relational data model represents a database of records as an *n*-ary relation. With a view to the definition of a relation, records are elements of the relation and fields are its domains.

Example 4 Person records are represented as 4-tuples (*N,C,P,A*), where *N* is the name, *C* is the citizenship, *P* is the passport number, *A* is the age. A database of five such records is:

(Nowak, Poland, AB02668, 51)
(Powel, U.S.A., 456XG768, 40)
(Heinz, Germany, 567925MK, 35)
(Armstrong, U.S.A., 332HK862, 37)
(Klas, Czech, D985556K, 40).

Such a sample database may be viewed as a 4-ary relation *R* on sets N, C, P, A, ie.

$$R \subseteq NxCxPxA..$$

Since relations representing databases are often displayed in a table form, they are said to be **tables.** The database of persons from Example 3 will be then displayed as Table 1.

Person name	*Citizenship*	*Passport number*	*Age*
Nowak	Poland	AB02668	51
Powel	U.S.A.	456XG768	40
Heinz	Germany	567925MK	35
Armstrong	U.S.A	332HK862	37
Klas	Czech	D985556K	40

Table 1

To identify uniquely elements of a database or an *n*-ary relation, the concepts of *primary* or a *composite keys* are employed.
A domain A_i of an *n*-ary relation *R* is a **primary key**, when no two different *n*-tuples of *R* have the same element as its *i*th element. This concept may be naturally extended to a set of domains. If a combination of domains uniquely identifies n-tuples in an *n*- ary relation, the Cartesian product of these domains is called a **composite key.**
Example 5 A 4-ary relation from Example 4 has a primary key *N* (*Person name*) or *P*(*Passport number*). *Citizenship*(*C*) and *Age* (*A*) are not primary keys. Composite keys are for instance $NxC, NxA, PxCxA$.
Since records are currently added to or deleted from databases, the property that a set is a primary or a composite key is in general time-dependent. To avoid this, a primary(or a composite) key should be chosen in such a way that remains one whenever the database is changed. For this purpose, a primary(or a composite) key of so called **intension** of the database is used, containing all the *n*-tuples that can ever be included in an *n*-ary relation representing the given database.
Example 6 Refering database given in Example 3, the only *Passport numer* has a chance to be a primary key of the intension (supposing that no two persons have the same passport number). Apparently, a composite key of the intension is for instance PxC.

The property of being primary or composite key of a database corresponds with the property of functionality of the related *n*-ary relation:
An *n*-ary relation *R* on sets $A_1,..., A_n$ is A_i **- functional**, if for every $a_i \in A_i$ there is at most one *n*-tuple $(a_1,...,a_i,...,a_n)$ that belongs to *R*.
In the sequel in a Cartesian product $A_{i_1}xA_{i_2}x...xA_{i_k}$ we suppose that $k \leq n$, $i_1,...,i_k \in \{1,…, n\}$ and $i_1<…<i_k$.
An *n*-ary relation *R* on sets $A_1,..., A_n$ is $A_{i_1}xA_{i_2}x...xA_{i_k}$ - **functional**, if for every $(a_{i_1},...,a_{i_k}) \in A_{i_1}xA_{i_2}x...xA_{i_k}$ there is at most one *n*-tuple $(a_1,...,a_{i_1},...,a_{i_2},...,a_{i_k},...,a_n)$ that belongs to *R*.
The following proposition is straightforward:
Proposition If *R* is A_i-functional, then *R* is $A_{i_1}xA_{i_2}x...xA_{i_k}$ - functional for any Cartesian product $A_{i_1}xA_{i_2}x...xA_{i_k}$ such $i = i_k$ for some *k*.
Proof: If *R* is A_i-functional, then for every $a_i \in A_i$ there is at most one *n*-tuple $(a_1,...,a_i,...,a_n) \in$ *R*. Since $i = i_k$ for some *k*, it is just the *n*-tuple required for a relation *R* to be $A_{i_1}xA_{i_2}x...xA_{i_k}$ -functional.
Consequence If a domain A_i is a primary key, then any combination of domains containing A_i is a composite key.

n-ary Relation Operations and Information Manipulations in Databases
There are esentially 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 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 7 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 2 and 3:

N	*S*	*M*
1	NOVAK	HISTORY
2	VRANA	PHYSICS
3	THOMAS	MATHS
5	BARTA	ECONOMY

Table 2 Ternary relation R

N	*S*	*M*
1	NOVAK	HISTORY
2	VRANA	ECONOMY
4	BROWN	MATHS
2	VRANA	PHYSICS
6	KABAT	MUSIC

Table 3 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 4, 5, 6, 7.

N	*S*	*M*
1	NOVAK	HISTORY
2	VRANA	PHYSICS
2	VRANA	ECONOMY
5	BARTA	ECONOMY
4	BROWN	MATHS
6	KABAT	MUSIC

Table 4 Ternary relation $R \cup T$

N	*S*	*M*
1	NOVAK	HISTORY
2	VRANA	PHYSICS

Table 5 Ternary relation $R \cap T$

N	*S*	*M*
3	THOMAS	MATHS
5	BARTA	ECONOMY

Table 6 Ternary relation $R - T$

N	*S*	*M*
2	VRANA	ECONOMY
4	BROWN	MATHS
6	KABAT	MUSIC

Table 7 Ternary relation $T - R$

In words, database corresponding to $R \cup T$ contains records that are in Table 2 or Table 3(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 2 and 3, database corresponding to $R - T$ records that are in Table 2 but not in Table 3, database corresponding to $T - R$ records that are in Table 3 but not in Table 2. Notice that in all cases the resulting tables are of the same structure.

C a r t e s i a n p r o d u c t

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 $(b_1,...,b_n) \in S$.

Example 8 Let R be a ternary relation as given in Example 7 and T a 2-ary(binary) relation on P (*Professor*), L(*Lecture room*) given as database of records by the following Table 8. Then the resulting 5-ary relation RxT on *N, S, M, P, L* is given as database of records by Table 9.

P	*L*
KREN	384
MOOR	381

Table 8 Binary relation T

N	*S*	*M*	*P*	*L*
1	NOVAK	HISTORY	KREN	384
2	VRANA	PHYSICS	KREN	384
3	THOMAS	MATHS	KREN	384
5	BARTA	ECONOMY	KREN	384
1	NOVAK	HISTORY	MOOR	381
2	VRANA	PHYSICS	MOOR	381
3	THOMAS	MATHS	MOOR	381
5	BARTA	ECONOMY	MOOR	381

Table 9 5-ary relation RxT

Database corresponding to RxT contains records made up by connecting every row of Table 7 with every row of Table 8. The resulting Table 9 posseses a new structure with 3+2=5 columns and the number of its records is the product of the number of records in both databases.

P r o j e c t i o n

Let R be an n-ary relation on sets $A_1,...,A_n$ and $k \leq n$. 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

$$\text{if } (a_1,...,a_n) \in R \text{ then } (a_{i_1},...,a_{i_k}) \in R_{i_1,...,i_k}.$$

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 9 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 10. Then its projection $R_{3,4}$ is the binary relation shown in Table 11.

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 10 5-ary relation R

M	*P*
HISTORY	KREN
PHYSICS	MOOR
MATHS	ROSS
ECONOMY	DALE

Table 11 Binary relation $R_{3,4}$

J o i n

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 $(m+n-p)$ 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 10 Let R be a 5-ary relation given by Table 10 and S be a 4-ary relation on sets M(*Major*), P(*Professor*), L(*Lecture room*), C(*Credits*) given by the following Table 12. Then the join of R,S, $J_3(R,S)$ is shown in Table 13.

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 12 4-ary relation S

N	*S*	*M*	*P*	*L*	*C*
1	NOVAK	HISTORY	KREN	384	6
2	VRANA	PHYSICS	MOOR	384	8
3	THOMAS	HISTORY	KREN	381	6
5	BARTA	MATHS	ROOS	381	8
1	NOVAK	ECONOMY	DALE	381	6

Table 13 6-ary relation $J_3(R,S)$

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 in 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. This should be the main aim and contribution of the paper. A number of problems arise to further investigation. For instance, the testing procedures for composite keys, the properties of composite keys with respect to database operations and optimization problems.

References

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

Koch, M.: *Datové a funkční modelování.* FP VUT v Brně v Akademickém nakladatelství CERM, s.r.o., Brno 2006 (in Czech)

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