o

RELATIONAL DATABASE-CONCEPT,

CONSEQUENCES FOR ORGANIZATION AND MANAGEMENT-STRUCTURES

Uwe Hinrichs

Uwe Hinrichs
Sauer-Informatic
Krokamp 35
2350 Neumünster
West-Germany

Summary of: RELATIONAL DATABASE-CONCEPT, CONSEQUNCES FOR ORGANIZATION
              AND MANAGEMENT-STRUCTURES

1. What is a releational database

If you look at EDP-concepts, you will find in almost all cases an
hierarchically structured file organization. Data sets were organized in
different stand-alone files, which were accessed through SEARCH-ITEMS. Most
of the FILES were designed for a single application. The structure of this
file-system (i.e. the PATHES to the DATA-ENTRIES) has been defined before
implementing the data sets (i.e. in ISAM, HISAM, KSAM files etc.).

EXAMPLE : find "city of Berlin"

            if ZIP-CODE is the search-item to all cities, you could find
            the city of Berlin only, if you knew the zip-code.

Only with SORT/MERGE you could reassemble your file, so that it would
satisfy a different QUERY.

If changes would become more complex, you would have to reorganize your
file-system for this particular application. that means building up
new files with new PATHES and reflecting these changes within the
program/system.

With the appearance of database systems this became much easier. But still
you needed a strong, hierarchical structure of organization at the
beginning. The advantage was that your DATA didn't fit for just one
application but for your application system in total.

EXAMPLE : a whole application system ---▷ COMPANY

            a single application   ---▷ SALES
                                              MATERIALS MANAGEMENT
                                              FINANCE
                                              etc.

The understanding of a database system was originally a mass storage file
system, in which the data sets were application independent.
Still you had to organize the pathes within the database in the beginning,
so that you had a regulated structure, which couldn't be modified without
any problem.

EXAMPLE : implemented references between : SALES ---▷ FINANCE
            postulate references between   : SALES ---▷ MATERIALS MANAGEMENT

To fullfill this postulate, you needed a modification of the database struc-
ture or you could use a QUERY-SYSTEM.
But a disadvantage of QUERY systems is the time used to search items in a large
database through PATHES, which have not been implemented (SCHEMA).

There are different types of such traditional DATABASE-SYSTEMS; for Example :
HIERARCHICAL and NETWORK DATABASES.
Both TYPES OF DATABASE SYSTEMS have the same features of a traditional file
management organization :

- a regulated structure
- modifying and adding of references and programs are very difficult/time consuming
- cycle of lifetime about 5 years, because the requirements are changing (trouble shouting)
- must of top/down approach
- batch approach
- price/performance for modifiying and adding

The relational database system, now, has a completely different approach. The structure of TRADITIONAL FILE-SYSTEM/DATABASE-SYSTEM has to be defined first, before you start solving your detail problem. With RELATIONAL DATABASE SYSTEMS (RDBS) the strcture is variable :
you still might define temporary relations like you could with traditional QUERY SYSTEMS. However, the most important feature of a RDBS is the capability of learning relations (PATHES).

The capability of modifying, deleting and adding new relations at any time is a feature of this system.

EXAMPLE : available relation ---▷ SALES - FINANCE

               postulate relation   ---▷ SALES - PRODUCTION PLANNING

You can define this relation easily with the new variable path information.

The advantage of relational database systems are :

- a variable structure without hierarchical levels
- easy to modify, delete and add relations and programs
- bottom up approach
- lifetime of the systems without a limit, because the system is flexible to adapt new requirements immediately
- full dialogue approach
- high quality of price/performance
- high data quality
- high automation level

2. Organizational Aspects

With a traditional computer organization you need a general concept before you can start programming and building the FILE-/DATABASE-SYSTEM in the field (i.e. manufacturing firms, administrations etc.).
The main chapters of this concept are :

- analysis of given structures
- definition of future requirements
- definition of implementation steps
- description of analysis
- implementation

The general concept - also the analysis - is divided into two parts.

first part : organizational concept of the analysed field

               related to field structure and needs

second part : organizational concept of EDP

               related to Hard- and Software-Systems (Software-tools).

These steps are time consuming with the additional disadvantage that the result is always just a snapshot of an existent system, possibly reflecting some future requirements.

With a traditional concept there is a strong interdependence between the first and the second part and there fore it is important to analyze the complete system in order to get an optimal, general concept which covers future requirements, too. Before starting to implement the application system with a traditional file/or database system it is necessary to have the organizational concept completed. The traditional TOP/DOWN APPROACH is usually used for this type of concept.

EXAMPLE : Organization Analysis

... etc.
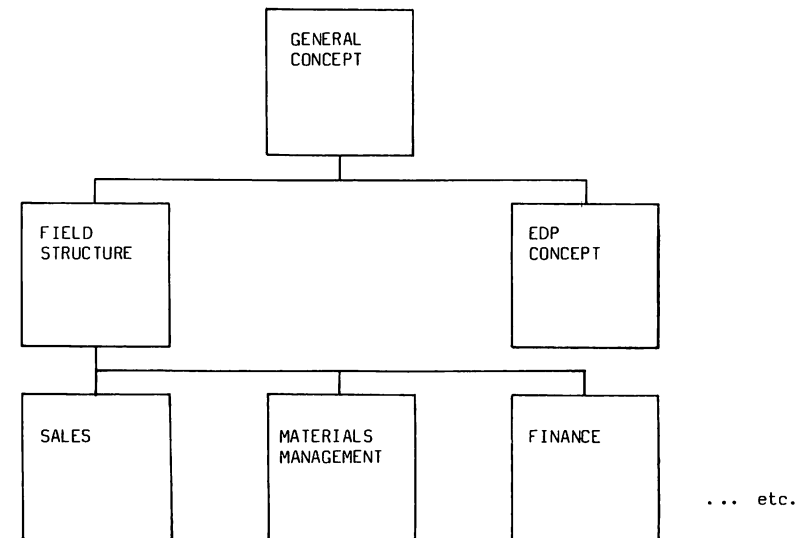
You must modify the whole structure and analyse the whole system again, if you modify or add relations between elements of the defined concept.

The great advantage of a relational database system is the variability in organization structure. The steps of system analysis are flexible, because you don't need a general concept that is structured in a first and second part like the general concept.

The main steps of realization are here :

- description of given structure with future requirements
- implementation

If new requirements arise you will implement them immediately without reflection on previous written Software. The problem is now restricted to generation of new field structure elements.

The system isn't a snapshot any longer, but a living system. Using this technique will now allow you to use the BOTTOM UP APPROACH for your organizational work. This is in fact the most efficient way to proceed because the level of quality of your DECISION-DATA is depending on the level of quality of your BASIS DATA.

The features of this organization method are :

- horizontal structured information --▷ relations between the field
  elements
- vertical structured information   --▷ Management Information System
- high level automation
- all kinds of information available immediately

EXAMPLE : Relational Concept