

A NON-NORMAL FORM DATABASE INTERFACE

by

GABRIELE A. HECKER

BS., Fachhochschule Furtwangen, West-Germany

A MASTER'S REPORT

**submitted in partial fulfillment of the
requirements for the degree**

MASTER OF SCIENCE

Department of Computer Science

**KANSAS STATE UNIVERSITY
Manhattan, Kansas**

1986

Approved by:

A handwritten signature in black ink, appearing to be "J. Wang", written over a horizontal line.

Major Professor

Table of Contents

Chapter 1: Introduction	1
1.1 Overview	1
1.2 Relevant research	2
1.2.1 The relational model	2
1.2.2 Normal Forms for databases	3
1.2.3 The extended relational model	3
1.3 The problem	12
Chapter 2: The set-relational interface	14
2.1 Description of a DBMS using the set-relational interface	17
2.2 Definition of the set-relation	18
2.2.1 Definition of the summary-table scheme on the external level	18
2.2.2 Definition of the graphical scheme information of the summary-table scheme	24
2.2.3 Definition of the set-relation on the conceptual level	30
2.3 Mappings for the set-relation between the levels of architecture	34
2.3.1 Mapping from the summary-table scheme on the external level to the summary-table scheme information in the data dictionary	34
2.3.2 Mapping from the summary-table scheme on the external level to the set-relation on the conceptual level	38
2.3.3 Mapping from the graphical scheme information in the data dictionary to the summary-table scheme on the external level	43
Chapter 3: Future aspects	47
3.1 Summary	47
3.2 Queries in the set-relational interface	47
3.3 Input data for the summary-table schemes	49
3.4 Functional dependencies and summary-table schemes	49
3.5 Correspondence between the set-relation and the relations of the relational model	52

Chapter 4: Conclusion	54
Bibliography	55

Table of Figures

Figure 1.2.1a	2
Figure 1.2.2.1a	5
Figure 1.2.2.1b	6
Figure 1.2.2.1c	7
Figure 1.2.2.1d	7
Figure 1.2.2.1e	8
Figure 1.2.2.2a	10
Figure 1.2.2.2b	11
Figure 2a	15
Figure 2b	16
Figure 2.1a	17
Figure 2.1.1.1a	20
Figure 2.1.1.1b	20
Figure 2.1.1.1c	20
Figure 2.1.1.1d	21
Figure 2.1.1.1e	22
Figure 2.1.1.1f	22
Figure 2.1.1.1g	23
Figure 2.1.1.1h	23
Figure 2.2.2a	25
Figure 2.2.2.1a	26
Figure 2.2.2.1b	27

Figure 2.2.2.1c	27
Figure 2.2.2.1d	27
Figure 2.2.2.1e	28
Figure 2.2.2.1f	28
Figure 2.2.2.1g	29
Figure 2.2.2.1h	29
Figure 2.2.2.1i	30
Figure 2.2.2.1j	30
Figure 2.2.3.1a	32
Figure 2.2.3.1b	32
Figure 2.2.3.1c	32
Figure 2.2.3.1d	33
Figure 2.2.3.1e	33
Figure 2.2.3.1f	33
Figure 2.2.3.1g	34
Figure 2.2.3.1h	34
Figure 2.2.3.1i	34
Figure 2.3.1.1a	36
Figure 2.3.2.1a	40
Figure 2.3.2.1b	41
Figure 2.3.2.1c	42
Figure 2.3.2.1d	42
Figure 2.3.2.1e	42
Figure 2.3.3.1a	45

Figure 2.3.3.1b	46
Figure 3.4a	50
Figure 3.4b	50
Figure 3.4c	51
Figure 3.4d	51
Figure 3.5a	52
Figure 3.5b	53

Chapter 1: Introduction

1.1 Overview

The relational model was developed by E. F. Codd in the early seventies as an attempt to bring more data independence into databases. The representation of data in the relational model uses data structures that can be visualized as tables. In these tables the columns, also called attributes, are implicitly related. These relationships can be expressed by dependencies between the data elements. The tables are called relations. If the relation is normalized it contains only atomic values in the domains. There are many applications in which the natural expression of the data occurs not only in atomic units but also in collections, e.g., text in an office form. An extended relational model sometimes, called non-first normal form, has been designed where the relations meet the normal form conditions except for the first normal form condition [JAES82a]. A domain in the extended relational model can contain sets or collections.

There are two relevant major research efforts, one by Ozsoyoglu and Ozsoyoglu and the other by Jaeschke and Schek. Ozsoyoglu and Ozsoyoglu developed a query language based on the extended relational model using summary-tables. The research work done by Jaeschke and Schek describes a new algebra for non-first normal form relations.

With the advent of spreadsheet based systems it seems reasonable to allow a user to interact with the database system through a table for retrieval, creation and maintenance. This paper presents the definition of such an interface. It is based on an extension of the relational and extended relational model using sets of collections.

The relevant research for this paper is described in section 1.2. The problem is defined in chapter 1.3. Chapter 2 presents the set-relational interface on the external and the conceptual level as well as the mappings between these levels of architecture. Future aspects in chapter 3 point out some of the potentials of this interface.

1.2 Relevant research

1.2.1 The relational model

The underlying mathematical concept for the relational model is the set theoretic relation. A relation is defined as any subset of the Cartesian product of one or more domains [CODD72a]. In this definition the domains consist of single-valued attributes which means that a cell can contain one element. Ozsoyoglu and Ozsoyoglu define a relation as a table where each column is labeled by a distinct attribute and each row (tuple) is unique. Figure 1.2.1a depicts a relation with single-valued attributes [OZSO85].

r	A	B	C
	a1	b1	c1
	a2	b2	c2
	a3	b3	c3
	a4	b4	c4

Figure 1.2.1a : Relation with single-valued attributes

A relational database is a collection of relation instances. It contains the current values of the corresponding relation schemes where a relation scheme is defined as the set of attributes labeling the columns of a particular relation [OZSO85].

1.2.2 Normal Forms for databases

For some relations adding, deleting or updating data can have unexpected consequences which are called *modification anomalies*. Normalization rules have been designed to cause database relations or files to be in a form such that maintenance-anomalies and data inconsistencies can be minimized. The normal forms defined in relational database theory represent the guidelines for record design [KENT83]. Each of the normal forms include their predecessor ,e.g., a relation in third normal form is also in second normal form and first normal form. That means that each of the normal forms though projection-join normal form (PJ/NF) is strictly stronger than its predecessor. Each successively stronger normal form tends to eliminate more anomalous situations.

1.2.3 The extended relational model

First normal form requires that each attribute in a relation must be based on a domain of atomic values. The imposition of the first normal form condition, especially at the user interface level, has been an impediment to user acceptance of the model. Many of our common data structures, e.g., forms and statistical tables violate this first normal form condition. Therefore the extended relational model was developed which removes the restriction that the relations must be in first normal form although they can be seen to be in higher normal forms (Boyce-Codd normal form, third normal form) [JAES82a].

Until recently the relational model did not include non-first normal form relations. In 1982 Klug [KLUG82] extended the relational algebra and the relational calculus by incorporating aggregate functions and Jacob [JACOB82] developed a database logic that allows defining and manipulating relations which may have relations as tuple components. Jaeschke and Schek [JAES82a] extended the relational algebra for summary-

tables incorporating set-valued attributes and aggregate functions. Ozsoyoglu, Matos and Ozsoyoglu [OZSO83c] extended tuple relational calculus with set-valued attributes and aggregate functions.

1.2.2.1 Previous work on the extended relational model by Ozsoyoglu and Ozsoyoglu

STBE (Summary-Table-By-Example) is a high level screen-oriented query language for statistical databases introduced by Z. M. Ozsoyoglu and G. Ozsoyoglu in 1984 [OZSO84a] and is part of a project called the "The System for Statistical Databases". The research is supported by the National Science Foundation.

Ozsoyoglu and Ozsoyoglu define a statistical database system (SDB) as a database system that supports statistical data analysis. Statistical analysis ranges from simple summary statistics like sum, average, median to advanced statistical techniques like hypothesis testing. One of the basic functions of such a system is to obtain, maintain and manipulate summary-data from the raw-data or other summary-data in the database [OZSO85]. Summary data is represented by summary-tables. For the extraction of data from a statistical database and the formation of summary-data into tabular form, STBE uses:

- i) Aggregate functions like, e.g., median, sum, average, etc.
- ii) Relations with set-valued attributes (non-first normal form relation)
- iii) Summary-tables

Ozsoyoglu and Ozsoyoglu define a non-first normal form relation relation (a relation with set-valued attributes) as a relation scheme where a tuple component in a column is labeled by a set-valued attribute. A set-valued attribute is a set of elements

where an element may be an integer, a real number or a string. On the other side, a tuple component in a column labeled by a simple-valued attribute is an element. Figure 1.2.2.1a shows a relation with set-valued attributes and in Figure 1.2.2.1b a relation with a single-valued attribute is presented. The set-valued attribute is labeled with an asterisk [OSZO85].

A relation is not in first normal form if it contain set-valued attributes which consist of a set of atomic elements. That restricts the model to have one depth level within the set-valued attribute and, e.g., no sets or sets of sets, may appear in set-valued attributes.

r	A	*B	C
	a1	{ b1, b2 }	c1
	a2	{ b3, b1 }	c2
	a3	{ b2 }	c3

Figure 1.2.2.1a : Relation with set-valued attribute *B and the simple-valued attributes A and C

r	A	B	C
	a1	b1	c1
	a2	b2	c2
	a3	b3	c3
	a4	b4	c4

Figure 1.2.2.1b : Relation with simple-valued attributes A, B and C

Informally expressed, a summary-table scheme is a two-dimensional table of cells. In figure 1.2.2.1c an instance of a summary-table is shown and figure 1.2.2.1d represents the summary-table scheme of that summary-table. The rows and columns of a summary-table have some attributes called category attributes. The category attributes are structured as a forest of trees whose nodes are attributes. Those attributes appearing in a row are called row-category attributes and those appearing in a column are called column-category attributes. A cell has an attribute named cell-attribute. In this example the column-category attribute forest is empty. Actually either the row-category attribute forest or the column category attribute forest can be empty but not both. In figure 1.2.2.1e the row-category attribute forest for the the example summary-table scheme is shown. It consists of the nodes STATE, COUNTY, SEX. Ozsoyoglu and Ozsoyoglu define a summary-table as a tuple $Rs(Fr, Fc, Ac)$ where Fr are row-category attribute forests, Fc are the column- category attribute forests and Ac is an ordered set of cells [OZSO85].

Ohio	Ashtabula	female	63.5
		male	77.2
	Cuyahoga	female	81.5
		male	56.2
	Ashtabula		68.9
	Cuyahoga		60.4
	Medina		62.0

Figure 1.2.2.1c : Example summary-table instance

AVERAGE-HOUSE-PRICES			
STATE	COUNTY	SEX	PRICE1
	COUNTY		PRICE2

Figure 1.2.2.1d : Example summary-table scheme

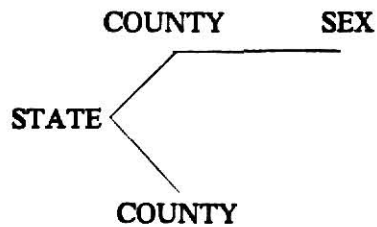


Figure 1.2.2.1e : Row-category attribute forest of the example summary-table scheme

1.2.2.1.1 Queries in STBE

STBE uses the concepts of QBE (Query-By-Example) introduced by M. M. Zloof in 1977 [ZLOO77] and ABE (Aggregates-By-Example) which is another graphical query language introduced by A. Klug in 1981 [KLUG81].

The user fills a skeleton on the screen with an example of a possible answer. The skeletons can be empty relations and summary-tables in hierarchically arranged windows. The user constructs output summary-tables or output-relation skeletons or opens windows. The example elements typed in by the user accomplish the task of matching and retrieving of data in the database and binding of relations and summary-tables in different windows.

In the windows single-underline and double-underline variables can occur. The single-underline indicates a free variable and the double-underline indicates a bound or fixed variable. A particular free variable X can only be specified in one window and a fixed variable which is bound to X can only appear in descendent windows of that one where X is specified.

If for example there is a variable *X bound to *X then it is the same, *X matches the components of only those tuples where the value of the tuple-element is equal to the current value of *X. But a fixed variable X can also be bound to *X. In this case X matches the component of those tuples where the value of the tuple-element is equal to an element of the current value of *X.

1.2.2.1.2 Representation of summary-tables by relations

Ozsoyoglu and Ozsoyoglu define a primitive summary-table as a summary-table where each of the row attribute and column attribute forests consists of a single chain of attributes. That could be a tree with one leaf. A primitive summary-table has exactly one cell.

A relation possibly a non-first normal form relation can be used to represent a primitive summary-table excluding the order row and the type (i.e. column or row) of category attributes.

1.2.2.1.3 Physical storage aspects

Summary tables and set-valued attribute relations require new storage structures and access procedures. The data dictionary of the DBMS has to contain schema information for both summary-tables and set-valued attribute relations [OSZO85].

For the storage structure of set-valued attributes Ozsoyoglu and Ozsoyoglu propose two possibilities. The first is the use of an inverted index, the second is a bit matrix. Presently Ozsoyoglu and Ozsoyoglu are developing the display manager and the access path selection for STBE [OSZO85].

1.2.2.2 Previous work on the extended relational model by Jaeschke and Schek

In their algebra for non-first normal form relations Jaeschke and Schek introduced new operations, called **nest** and **unnest**. These operations transform first normal form relations into non-first normal form relations and vice versa. Given a relation $R(A, B, C)$ (see Figure 1.2.2.2a) in 1NF, the transformation into a non-first normal form relation is achieved by moving along column A of R, abbreviated by $v(R)$. Sets of A values are formed, when the tuples in R agree in the remaining components. The non-first normal form relation is shown in figure 1.2.2.2b. The unnest operation is the inverse to nest. [SCHE82].

r	A	B	C
	a	b	c
	a'	b	c
	a	b	c'
	a''	b'	c'

Figure 1.2.2.2a : First normal form relation $R(A, B, C)$

v (R)	* A	B	C
	a, a' a, a''	b b'	c c'

Figure 1.2.2.2b : Non-first normal form relation after nest-operation

1.3 The problem

Most current database management systems require some programmer level knowledge of the user to be able to interact with the system. Query languages like QBE (Query-By-Example) or STBE (Summary-Table-By-Example) use the form of tables to enable the user to formulate retrieval requests. The user creates and fills a table with an example of the data to be retrieved. Ozsoyoglu and Ozsoyoglu introduce summary-tables in order to produce the formation of summary-data into tabular form, as a means for output formatting. They suggest a restricted summary-table called a primitive summary-table to be represented by a relation. For future work Ozsoyoglu and Ozsoyoglu suggest that schema information of summary-tables should be present in the data dictionary [OSZO85].

In this work an interface is studied where summary-table schemes are represented by relations. This representation allows the user to introduce new relations to the system via summary-table schemes. To serve this purpose a new relation is introduced that has the capability to represent summary-table schemes on the conceptual level. The definition of a summary-table scheme varies from the one presented in the research work of Ozsoyoglu and Ozsoyoglu. The main emphasis of the study has been taken on the scheme information of a summary-table on the external and conceptual level.

For the set-relational interface the following results have been obtained:

- i) Definition of a new relation called set-relation on the external and conceptual level
 - a) Definition of the summary-table scheme on the external level
 - b) Definition of the graphical scheme information of a summary-table scheme
 - c) Definition of the set-relation on the conceptual level
- ii) Definition of the mapping functions for the summary-table scheme and the set-relation on different levels of architecture
 - a) Definition of the mapping from the summary-table scheme on the external level to the graphical scheme information on the conceptual level
 - b) Definition of the mapping from the graphical scheme information on the conceptual level to the summary-table scheme on the external level
 - c) Definition of the mapping from the summary-table scheme on the external level to the set-relation on the conceptual level

Chapter 2: The set-relational interface

In this chapter the set-relational interface is presented as a solution to the problem of providing the user with a tool that allows him to interact with the system through tables on the screen.

The set-relational interface encompasses two levels of architecture, the external level and the conceptual level. Summary-tables defined on the screen by the user are the features of the set-relational interface on the external level. The conceptual level consists of two data structures to capture the relationship of the table labels and the physical positioning of these labels (see topology of figure 2a). One is a relation called the set-relation on the conceptual level which is based on the extended relational model and represents the relationship among data elements, that occur as forests of trees, on the conceptual level. The other data structure represents the physical arrangement of the table on the screen.

Mapping functions defined in the set-relational interface provide the tools for traversing the borderlines between the levels of data base architecture. Those functions perform the transformation of the summary-table scheme on the external level to the scheme information in the data dictionary and the set-relation on the conceptual level. In figure 2b the mapping functions which allow the transformation from one level of architecture to the other and vice versa are depicted.

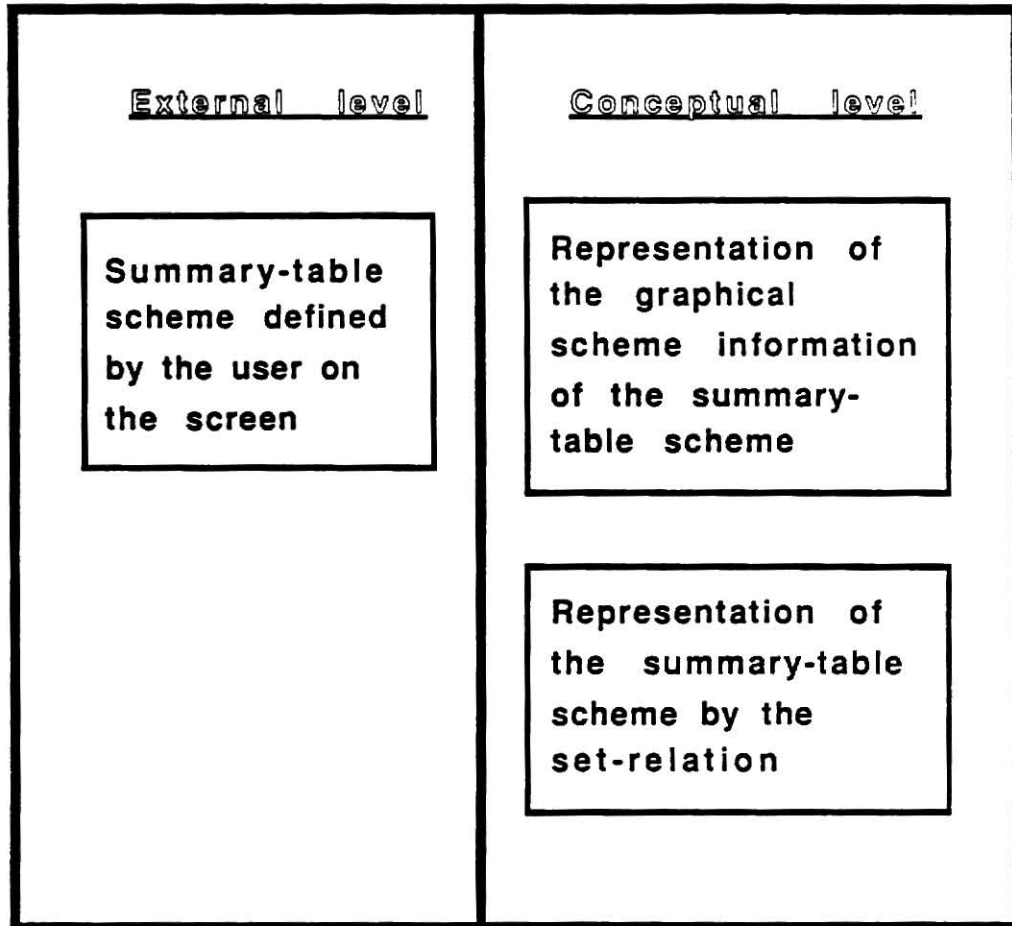


Figure 2a : Features of the set-relational model on the external level and on the conceptual level

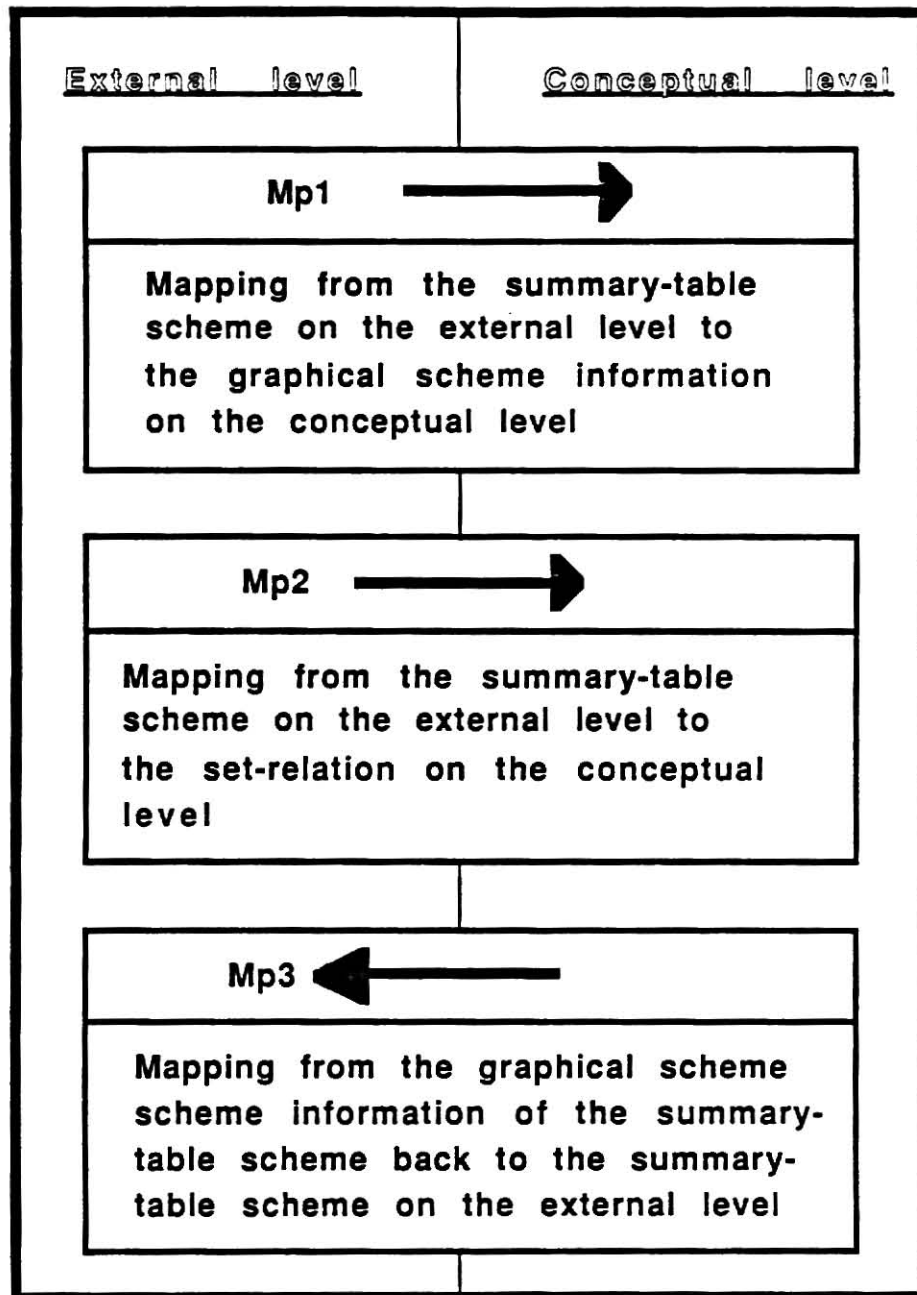


Figure 2b : Mappings from the external level to the conceptual level and vice versa

2.1 Description of the set-relational interface

This section describes how the set-relational model interacts with the user on the external level and how the representation of the user designed summary-table scheme appears on the conceptual level.

i) The user's view

With the set-relational interface the database user is given a system interaction tool that enables him/her to introduce new relations to a relations via summary-table schemes. The user creates summary-table schemes by opening rectangles and filling them with attribute names. The example summary-table scheme in figure 2.1a could be such a user defined table. The summary-table scheme represents facts about the population, capital and size of a state. It consists of four windows or rectangles that are filled with attribute names. After the creation of the summary-table scheme is finished a new relation has been introduced to the system.

STATE	CAPITAL
	POP
	SIZE

Figure 2.1a : Summary-table scheme

ii) The conceptual view

The representation of the summary-table scheme by the set-relation on the conceptual level requires two considerations. First some graphical information has to be kept to reconstruct the user's view on request and second the summary-table scheme has to be transformed into the relation scheme on the conceptual level which will be needed to create the actual relations at the physical level. For the representation of the summary-table scheme as a relation on the conceptual level a new relation called the set-relation is introduced.

The set-relation can be described as a set of atomic values or a set of sets of atomic values. The formal definition of the set-relation is given in the following chapter.

2.2 Definition of the set-relation

2.2.1 Definition of the summary-table scheme on the external level

The definition of a summary-table introduced by Oszoyoglu and Oszoyoglu restricts the nodes of the row-category attribute and the column category attribute forests in their summary-table scheme to attributes. In the set-relational interface the nodes of a tree of a summary-table scheme consists of sets containing elements, that are attributes.

In order to define a summary-table scheme represented by forests of trees the definitions of a tree and a forest are requested. Knuth defines a tree and a forest in the following way. [KNUTH69].

i) Definition of a tree:

A tree is a finite set T of one or more nodes such that

- a) There is one specially designated node called the root of the tree, $\text{root}(T)$;

- b) The remaining nodes (excluding the root) are partitioned into $m \geq 0$ disjoint sets T_1, \dots, T_m , and each of these sets in turn is a tree. The trees T_1, \dots, T_m are called the subtrees of the root, [KNUT69a].

ii) Definition of a forest:

A forest of trees is a set (usually an ordered set) of zero or more disjoint trees, [KNUT69b].

After the definition of a tree and a forest the definition of the a summary-table scheme can be formulated.

Definition of a summary-table scheme:

A summary-table scheme is a triple $Rs(Fc, Fr, As)$ where

- i) As is a set of elements where each element is a set containing exactly one attribute.
- ii) Fc is a forest of trees whose nodes are elements from the set AS . The forest Fc labels the columns of the table and is called column category attribute forest.
- iii) Fr is a forest of trees whose vertex are elements from the set As . The forest Fr labels the rows of the table. The forest Fr is called row category attribute forest.

The root node of each tree of a row or column category attribute forest is formed by the attribute most remote from the data, e.g., for a tree of Fc it is at the top most level of the tree and for a tree of Fr it is the leftmost. Each level of the tree is formed by the attributes at the next level of the scheme structure.

2.2.1.1 Examples for summary-tables schemes

- i) **Example for a summary-table scheme with more than one tree in each category attribute forest**

Figure 2.1.1.1a shows a summary-table scheme that describes the number of male and female employees and the average salaries of each of this group in a company. The summary-table scheme has one tree in the row category attribute forest and one tree in

the column category attribute forest. The trees of figure 2.1.1.1a are shown in figure 2.1.1.1b and figure 2.1.1.1c

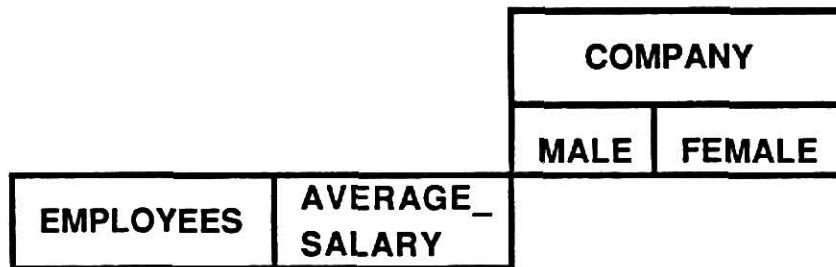


Figure 2.1.1.1a : Summary-table scheme

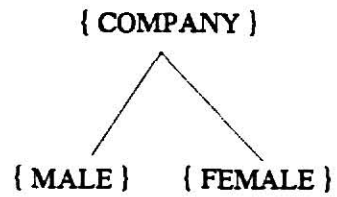


Figure 2.1.1.1b : Column category attribute forest Tr

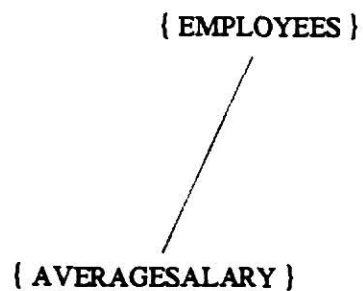


Figure 2.1.1.1c : Row category attribute forest Fr

ii) **Example of a summary-table scheme with more than one tree in a category attribute forest**

In a case where one of the forest consists of more than one tree an example summary-table scheme could have the form of figure 2.1.1.1d where two different areas are described. One is the number of male and female car owners in a state that either have a FORD or a ROLLS_ROYCE or a MERCEDES. The other is the number of employees in a carcompany that are car owners and who have either a FORD or a ROLLS_ROYCE or a MERCEDES. In this example summary-table scheme the row category attribute forest consists of one tree and the column category attribute forest consists of two trees.

		STATE		CARCOMPANY
		MALE	FEMALE	EMPLOYEES
CAR_OWNER	FORD			
	ROLLS_ROYCE			
	MERCEDES			

Figure 2.1.1.1d : Summary-table scheme where the column category attribute forest consists of two trees and the row category attribute forest consists of one tree.

The tree representation of the column category attribute trees and the row category attribute tree is illustrated below in figure 2.1.1.1e.

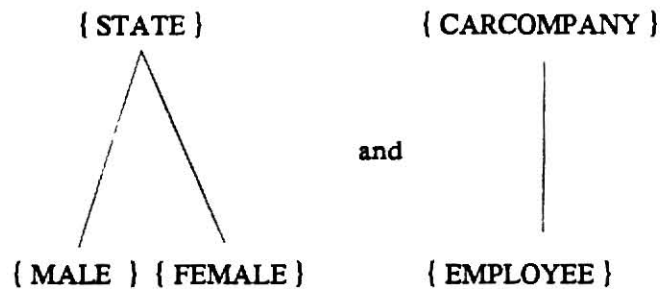


Figure 2.1.1.1e : Representation of the column category attributes trees

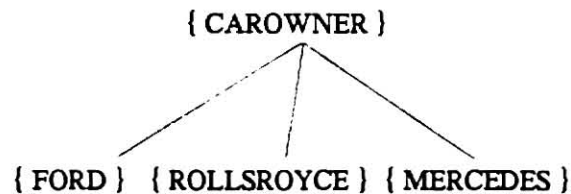


Figure 2.1.1.1f : Representation of the row category attribute tree

2.2.1.2 Limitations imposed on the design of summary-table schemes

While opening windows and creating summary-table schemes the user is bound to certain restrictions considering the construction of those schemes. The shape of a tree in a summary-table scheme has to be a rectangle constructed out of smaller rectangles that represent the nodes and levels of the tree. In the case of an unbalanced tree the shape of the tree in the summary-table scheme could be different from a rectangle. In this case the rectangles that represent this particular tree on the screen have to be reshaped such that the graphical arrangement of the tree forms a rectangle again. Figure 2.1.1.1g shows the case of an unbalanced column category attribute tree. The graphical representation of this tree as illustrated in this figure is not valid. The user could change the size of the attribute rectangle FEMALE in order to gain a rectangular graphical structure for the tree. Figure 2.1.1.1h shows the corrected column category attribute tree of figure 2.1.1.1g.

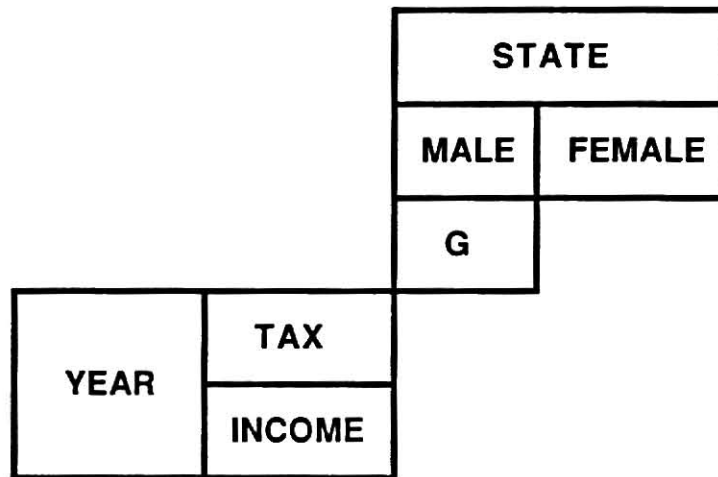


Figure 2.1.1.1g : Invalid summary-table scheme where the graphical arrangement of the column category attribute forest is not valid

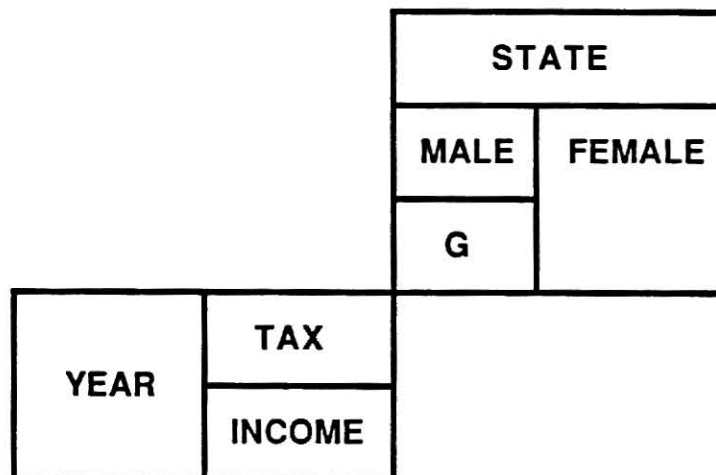


Figure 2.1.1.1h : Valid summary-table scheme with the corrected graphical arrangement of the column category attribute tree

2.2.2 Definition of the graphical scheme information of the summary-table scheme

A representation of the graphical summary-table scheme in the data dictionary is necessary to be able to reconstruct the scheme the user originally designed. This requires that information about the size and shape of the summary-table scheme is kept.

The structure of the summary-table scheme on the screen is represented through trees where the rectangles correspond to the nodes. Each rectangle has a certain size and a position depending on the hierarchy level within the tree. A rectangle represents a node of a tree within a summary-table scheme. To produce exactly the same summary-table scheme the user defined information about the size of the rectangles is needed. For the definition of the graphical scheme information of the summary-table scheme it is necessary to define a rectangle first.

Let cl be the length of a particular rectangle Q and ch the height of the same rectangle. The rectangle contains an attribute which determines the name of the rectangle.

A rectangle can be defined as a triple $RT(AN, CL, CH)$, where

AN is the attribute name contained in the rectangle. It determines also the name of the rectangle

CL is the length of the rectangle (expressed in cursor positions on the screen)

CH is the height of the rectangle (expressed in cursor positions on the screen)

The summary-table scheme node shown in figure 2.2.2a can be represented by the following set where the rectangle name is POPULATION, the cell length is sixteen pixel and the cell height is five pixel: { POPULATION, 16, 5 }.

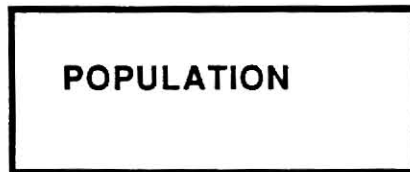


Figure 2.2.2a : Rectangle representing a node in a summary-table scheme

After the definition of a rectangle the graphical scheme information of a summary-table scheme can be defined in the following way:

The graphical scheme information is a tuple $GI(I_c, I_r)$ where

- I_c is an ordered set of sets that contain scheme information for each of the trees of the column category attribute forest of the summary-table scheme. The nodes of the trees are represented by rectangles.
- I_r is an ordered set of sets that contain scheme information for each of the trees of the row category attribute forest of the summary-table scheme. The nodes of the trees are represented by rectangles.

In order to illustrate the representation of the descriptive part of a summary-table scheme two example summary-table schemes $G1$ and $G2$ are shown and their graphical scheme information is formed. The example summary-tables schemes and their graphical scheme information are presented in the next section.

2.2.2.1 Examples of graphical scheme information representation

- i) Graphical scheme information for an example summary-table scheme with one tree in each category attribute forest

Example summary-table scheme $G1$ in figure 2.2.2.1a shows a table that describes

what the average fees and credit hours of American and foreign students are at a university in a year. The row and column category attribute forests consist of a tree each.

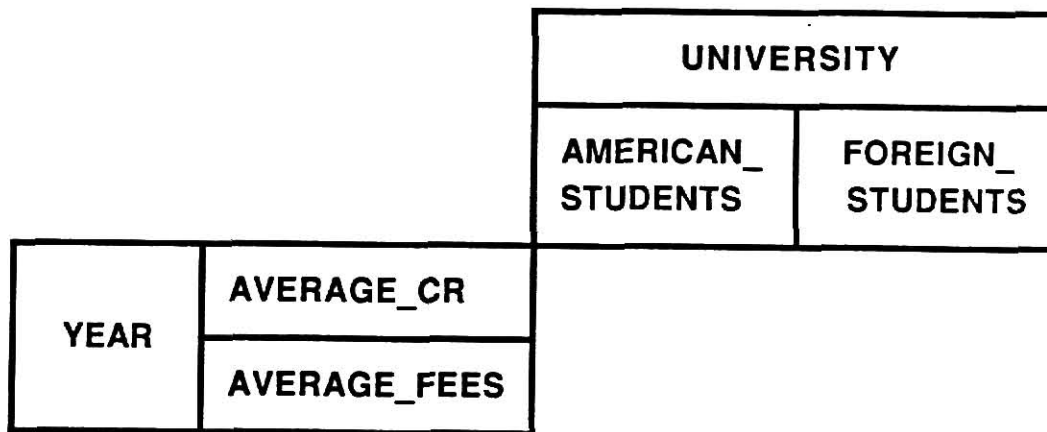


Figure 2.2.2.1a : Example summary-table scheme G1 where the row and column category attribute forests consist of a tree each

In order to represent the graphical structure of the scheme, the nodes of the trees that are sets containing an attribute are extended to two more elements containing the size information of the rectangles on the screen. In figure 2.2.2.1b and 2.2.2.1c the trees of the column category attribute forest and the row category attribute forest of the summary-table scheme G1 are represented where the nodes of the trees contain the graphical scheme information.

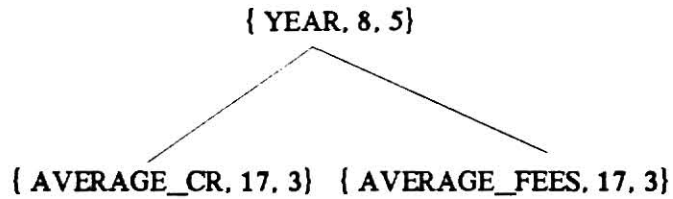


Figure 2.2.2.1b : Graphical information scheme of the row category attribute forest of the summary-table scheme G1

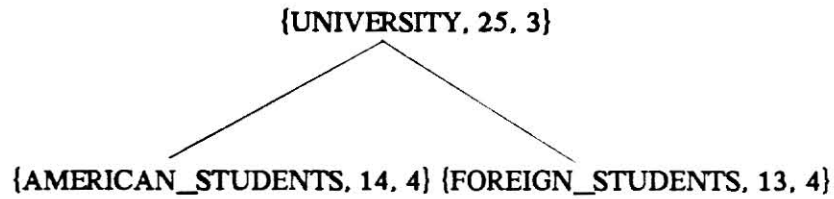


Figure 2.2.2.1c : Graphical information scheme of the column category attribute forest G1

The tree structures shown in figure 2.2.2.1b and 2.2.2.1c can be represented as sets of sets. The sets, Ic1 and Ir1, containing the graphical scheme information for the column category attribute forest and the row category attribute forest of G1 are represented in figure 2.2.2.1d and 2.2.2.1e. The rectangles (nodes) are represented by sets.

Ic1 = { {UNIVERSITY, 25, 3},
 { {AMERICAN_STUDENTS, 14, 4}
 {FOREIGN_STUDENTS, 13, 4} } }

Figure 2.2.2.1d : Set Ic1 containing the graphical information of the column category attribute tree of G1

$Ir1 = \{ \{YEAR, 13, 5\},$
 $\quad \{ \{AVERAGE_CR, 17, 3\}$
 $\quad \{AVERAGE_FEES, 17, 3\} \}$

Figure 2.2.2.1e : Set Ir1 containing the graphical information of the row category attribute forest G1

- ii) Graphical scheme information for an example summary-table scheme with more than one trees in a category attribute forest

Example summary-table scheme G2, depicted in figure 2.2.2.1f describes the same facts as G1 but it contains additional information about the student board at that particular university and how many American students and foreign students are on that board. The row category attribute forest of G2 consists of two trees and the column category attribute forest consists of one tree.

		UNIVERSITY	
		AMERICAN_STUDENTS	FOREIGN_STUDENTS
YEAR	AVERAGE_CR		
	AVERAGE_FEES		
STUDENTBOARD			

Figure 2.2.2.1f : Example summary-table scheme G2 where the row category attribute forest consists of two trees and the column category attribute forest consists of one tree

In figure 2.2.2.1g the graphical scheme information of the two trees of the row category attribute forest are shown and figure 2.2.2.1h depicts the graphical scheme information of the column category attribute forest of G2. The nodes consist of sets that contain the graphical information, e.g., the rectangle that represents the root of the column category attribute forest has the attribute name YEAR, the length of the rectangle is eight pixel and its height is five pixel.

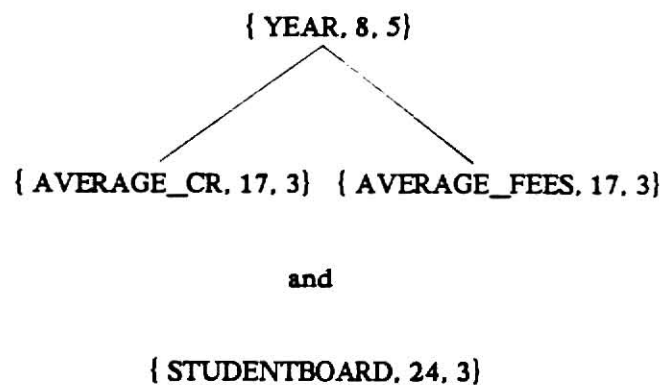


Figure 2.2.2.1g : Graphical information scheme of the row category attribute forest of the summary-table scheme G2

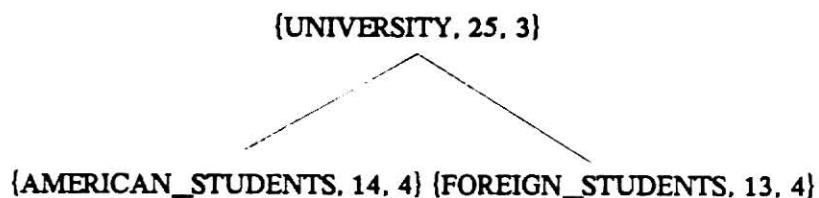


Figure 2.2.2.1h : Graphical information scheme of the column category attribute forest G2 in form of a tree

In figure 2.2.2.1i and figure 2.2.2.1j, the sets Ic2 and Ir2 of the summary-table

scheme G2 are shown. Ir2 is a set that contains two other sets as elements. Those elements are the graphical scheme representation of the two trees of the row category attribute forest. Ic2 is the graphical scheme information of the tree of the column category attribute forest.

$$\begin{aligned} \text{Ic1} = & \{ \{ \text{UNIVERSITY}, 25, 3 \}, \\ & \{ \{ \text{AMERICAN_STUDENTS}, 14, 4 \} \\ & \{ \text{FOREIGN_STUDENTS}, 13, 4 \} \} \end{aligned}$$

Figure 2.2.2.1i : Set Ic1 containing the graphical information of the column category attribute forest of G2

$$\begin{aligned} \text{Ir1} = & \{ \\ & \{ \{ \text{YEAR}, 13, 5 \}, \\ & \{ \{ \text{AVERAGE_CR}, 17, 3 \} \\ & \{ \text{AVERAGE_FEES}, 17, 3 \} \} \\ & \{ \text{STUDENTBOARD}, 24, 3 \} \\ & \} \end{aligned}$$

Figure 2.2.2.1j : Set Ir1 containing the graphical information of the row category attribute forest G2

2.2.3 Definition of the set-relation on the conceptual level

To be able to represent the summary-table scheme by a relation on the conceptual level, a new relation called set-relation is introduced. A set-relation can be described as a set or a set of sets of attributes. In the following paragraph the formal definition of the set-relation is presented.

Definition of a set-relation

A set-relation can be defined as a tuple $Rs(Ts, Ss, As)$ where

- i) As is a set of attributes where an attribute can be an integer, a real number or a string
- ii) Ss is a set of elements where an element can be an attribute or a set of attributes etc, where the attributes are elements of As .
- iii) Ts is a tree whose vertices are sets whose elements are elements of Ss such that $Root(Ts)$ can occur in three different cases:

Case I: The table scheme on the external level consists of one forest, the other forest is empty, then $Root(T)$ is a set containing a single attribute. Note: The forest is not a set of only roots

Case II: The forests of the table scheme on the external level are both non-empty, then $Root(Ts)$ is a set containing 2 attributes

Case III: One of the forest, row category attribute or column category attribute forest is empty. The non-empty forest consists only of roots, then $Root(Ts)$ is the union of the roots of the forest.

2.2.3.1 Examples of set-relations on the conceptual level and the corresponding summary-table schemes on the external level

In the following paragraph for each of the cases I through III in the definition of the set-relation, an example set-relation and the corresponding summary-table scheme from the external level is illustrated.

i) Example set-relation for case I

In figure 2.2.3.1a the set-relation SR1 is shown where the corresponding summary-table scheme on the external level consists of a column category attribute forest consisting of one tree and an empty row category attribute forest. SR1 represents an example for case I in the definition of the set-relation in chapter 2.2.3. In this case $Root(T)$ contains one attribute (STUDENT). The tree structure of the set-relations SR1 is depicted in figure 2.2.3.1b. Figure 2.2.3.1c represents the corresponding summary-table scheme to

SR1 on the external level.

$$SR1 = \{ \{STUDENT\} \{ \{ID\} \{SOC_NO\} \{NAME\} \} \}$$

Figure 2.2.3.1a : Set-relation SR1 (Case I of the definition of the set-relation in chapter 2.2.3)

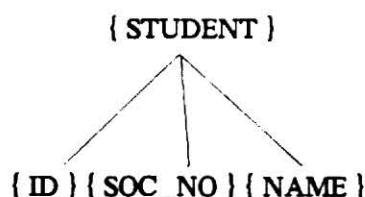


Figure 2.2.3.1b : Tree representation of the set-relation SR1

STUDENT		
ID	SOC_NO	NAME

Figure 2.2.3.1c : Corresponding summary-table scheme to SR1 on the external level

ii) Example set-relation for case II

As an example for case II of the definition of the set-relation in chapter 2.2.3 figure 2.2.3.1d presents the the set-relation SR2 where the column category attribute forest and the row category attribute forest in the summary-table scheme on the external level consist of one tree each. For this case the Root(T) of the set-relation contains the two roots of those trees. In figure 2.2.3.1d the tree representation of the set-relation SR2 is shown.

```

SR2 = { {STATE DEPT}
        { {UNIVERSITY}
          { { HEAD
              BUDGET_SIZE
              CURR_STD_NO } } } }

```

Figure 2.2.3.1d : Set-relation SR2 (Case II of the definition of the set-relation in chapter 2.2.3)

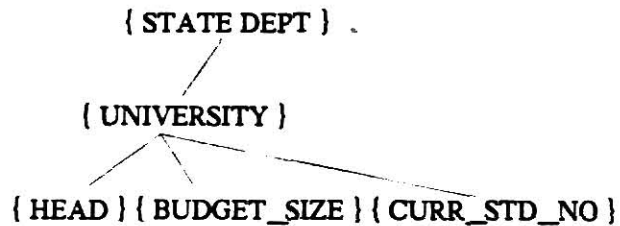


Figure 2.2.3.1e : Tree representation of the set-relation SR2

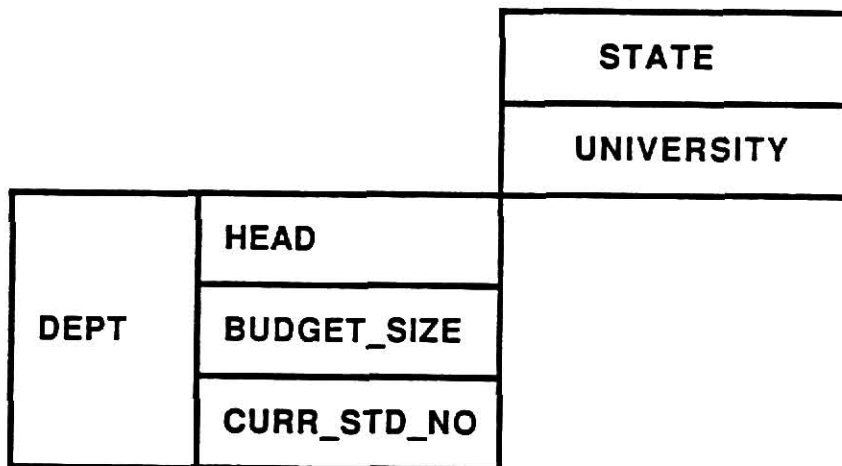


Figure 2.2.3.1f: Corresponding summary-table scheme of the external level to the set-relation SR2

iii) Example set-relation for case III

The next example consists of a set-relation where the row category attribute forest is empty and the column category attribute forest is built of three trees that have only

roots. In figure 2.2.3.1g the set-relation SR3 is shown and figure 2.2.3.1h depicts the tree representation of SR3. Figure 2.2.3.1i represents the corresponding summary-table scheme on the external level.

$SR3 = \{ \{ CAR_TYPE \ YEAR \ ENGINE \} \}$

Figure 2.2.3.1g : The set-relation SR3 (Case III of the definition of the set-relation in chapter 2.2.3)

$SR3 = \{ CAR_TYPE \ YEAR \ ENGINE \}$

Figure 2.2.3.1h : Tree representation of the set-relation SR3

CAR_TYPE	YEAR	ENGINE
----------	------	--------

Figure 2.2.3.1i : Corresponding summary-table scheme on the external level for set-relation SR3

2.3 Mappings between the levels of architecture in the set-relational interface

Within the architecture of a DBMS two kinds of mappings can be found. These mappings define the correspondence between the view of a relation on one level and the view on the other level. One mapping exists between the external and the conceptual level and the other is between the conceptual and the internal level. [DATE86a].

2.3.1 Mapping from the summary-table scheme on the external level to the graphical summary-table scheme information

To achieve the transformation of the summary-table scheme on the external level to

its structure representation on the internal level the summary-table scheme on the screen has to be analyzed. This will be done at the creation time of the summary table scheme.

The summary-table scheme on the screen consists, as previously defined, of two forests called the column category attribute forest and the row category attribute forest. The nodes of the a tree of one of the forests are represented by rectangles containing an attribute name.

For the mapping from the summary-table scheme on the external level to the graphical scheme information in the data dictionary it is necessary to analysis the structure of the forests. The graphical structure of a category forest of a summary-table scheme is determined by the size and the arrangement of the rectangles on the screen. Therefore the mapping function $Mp1$ can be defined as follows:

- i) For each tree that is a member of the column category attribute forest perform the following steps:
 - a) Perform a depth-first search through the structure of the column category attribute tree on the screen. Collect size information for each of the rectangles as well as the attribute names within the rectangle. Form an information set for each rectangle, e.g., { attribute_name, cell_length, cell_height }.
 - b) Represent the tree as an ordered set of sets
 - c) Represent the structure of the column category attribute forest as an ordered set of trees (Ic).
- ii) For each tree that is a member of the row category attribute forest perform the following steps:
 - a) Perform a depth-first, left to right search through the structure of the row category attribute tree on the screen. For this tree a depth-first search can be regarded as a most-right-first search and the left to right search can be interpreted as bottom up. Collect size information for each of the rectangles as well as the attribute names within the rectangle. Form an information set for each rectangle, e.g., { attribute_name, cell_length, cell_height }.
 - b) Represent the tree as an ordered set of sets
 - c) Represent the structure of the row category attribute forest as an ordered set of trees (I_r).

- iii) Form an ordered set of I_c and I_r

2.3.1.1 Example for the mapping function $Mp1$

In the following paragraph an example showing how the mapping function $Mp1$ works is illustrated. The starting point is an example summary-table scheme on the external level, depicted in figure 2.3.1a, that consists of a row and column category attribute forest with one tree each. The row category attribute tree consists of three rectangles or nodes; this is also true for the column category attribute tree. The steps of the mapping function are performed on the example summary-table scheme and the graphical scheme information that will be represented on the conceptual level in the data dictionary is produced.

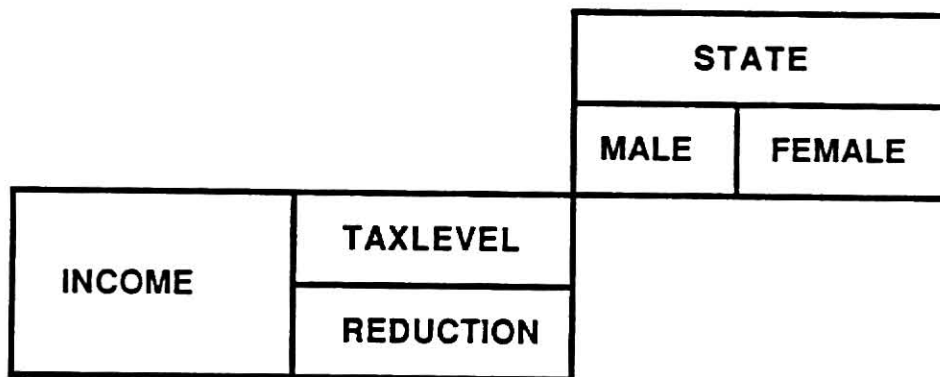
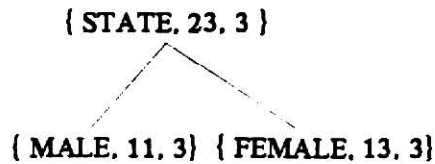


Figure 2.3.1.1a : Example summary-table scheme

- i) For the column category attribute tree the results of the steps are:
 - a) The size information of the rectangles of the column category attribute forest can be described through the following sets:
 For the rectangle STATE : { STATE, 23, 3 }
 for the rectangle MALE : { MALE, 11, 3 }
 for the rectangle FEMALE: { FEMALE, 13,3 }

The information tree that represents the column category attribute tree has the following outlook:



- b) The set-representation I_c of the tree has the following form:

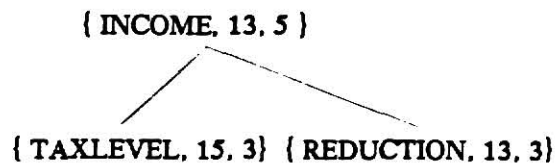
$$I_c = \{ \{ \text{STATE}, 23, 3 \} \{ \text{MALE}, 11, 3 \} \{ \text{FEMALE}, 13, 3 \} \}$$

- ii) For the row category attribute tree the results of the steps are:

- a) The size information of the rectangles of the column category attribute forest can be described through the following sets:

For the rectangle INCOME : {INCOME,13,5}
 for the rectangle TAXLEVEL : {TAXLEVEL,15,3}
 for the rectangle REDUCTION: {REDUCTION,15,3}

The tree that represents the row category attribute tree has the following outlook:



- b) The set-representation I_r of the tree has the following form:

$$I_c = \{ \{ \text{INCOME}, 13, 5 \} \{ \text{TAXLEVEL}, 15, 3 \} \{ \text{REDUCTION}, 15, 3 \} \}$$

- iii) Finally the graphical scheme information of the example summary-table scheme has the following form:

```

GI(Ic Ir) = {{{ STATE, 23, 3} {
                { MALE, 11, 3}
                { FEMALE, 13, 3} } }
              {{ INCOME, 13, 5} {
                { TAXLEVEL, 15, 3}
                { REDUCTION, 15, 3} } }
              }

```

2.3.2 Mapping from the summary table on the external level to the set-relation on the conceptual level

The structure of the summary-table on the external level is represented on the conceptual level as a set-relation. The set-relation consists, as described earlier, of sets whose elements are attributes. The mapping function Mp2 describes how a summary-table scheme can be transformed to the set-relation.

Again the column category attribute forest and row category attribute forests of the summary-table scheme are considered. For this mapping function the nodes of the trees are no longer regarded as rectangles of the summary-table scheme but as sets where each set contains exactly one attribute. The mapping function between the summary-table and the set-relation is defined in the following way:

- i) For each tree of the row-category attribute forest perform the following steps
 - a) Perform a depth-first, left to right search through the structure of the column category attribute tree on the screen. For this forest a depth-first search can be regarded as a most-right-first search and left to right can be interpreted as bottom up.
 - b) Represent the tree as an ordered set of sets but exclude the root(T)

Before the second step ii) of the mapping function Mp2 is performed, another fact has to be considered. For building the set-relations that are represented through a summary-table scheme, each tree of the column category attribute forest has to be combined with each tree of the row category attributes forest. A summary-table will be

represented by as many set-relations as there are binary combinations between the column category attribute trees and the row category attribute trees. Step ii) of the mapping function Mp2 is defined as follows.

- ii) For all combinations of trees of the column category attribute forest and the row category attribute forest perform the following step
 - a) Append the set representation of the row category attribute forest to each of the leaves of the column category attribute forest.
 - b) Perform a depth-first, left to right search through the structure of the column category attribute tree.
 - c) Represent the tree as a ordered set of sets and exclude the root(T)
 - d) Form an ordered set of the union of the roots of the combination of trees that is considered currently (column category attribute tree and row category attribute tree).
 - e) Form an ordered set that contains as elements the set of the roots and the set of the combined trees.

2.3.2.1 Examples for the mapping function Mp2

- i) **Example mapping for a summary-table scheme with one tree in each category forest**

In this example for the mapping Mp2 the summary-table scheme on the external level consists of one tree in the row and column category attribute forests. The summary-table scheme describes the price and ISBN of a book title written by a certain author. The example summary-table scheme is presented in figure 2.3.2.1a.

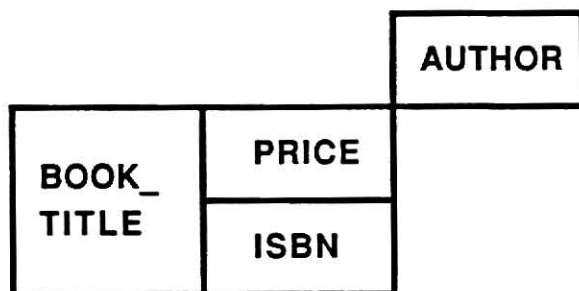


Figure 2.3.2.1a : Example summary-table with one tree in each category attribute forest

The steps i) a) through b) of the definition of the mapping function $Mp2$ result in the set $Sr1$.

$$Sr1 = \{ PRICE \text{ ISBN } \}$$

The steps ii) a) through b) are not applicable fully in this case for the column category attribute forest consists only of the root and the result is the following set $Src1$.

$$Src1 = \{ \{ PRICE \text{ ISBN } \} \}$$

Then steps ii) c through e) are performed. Step ii) c) performs the following result, where a set $Rcr1$ is formed of the roots of the row and column category attribute trees.

$$Rcr1 = \{ AUTHOR \text{ TITLE } \}$$

Finally the set-relation $S1$ is formed.

$$S1 = \{ \{ AUTHOR \text{ TITLE } \} \{ PRICE \text{ ISBN } \} \}$$

ii) **Example mapping for a summary-table scheme with more than one tree in a category forest**

The example summary-table scheme that is to be mapped into the set-relation consists of two trees in the row category attribute forest and one tree in the column category attribute forest. For there exists a forest where there more than one trees the mapping will result in two set-relations (there are two combinations of row category attribute trees with the category attribute forest possible). The example summary-table scheme is shown in figure 2.3.2.1b. The trees Tr1, Tr2 and Tc1 of the category attribute forests are depicted in figure 2.3.2.1c, 2.3.2.1d and 2.3.2.1e.

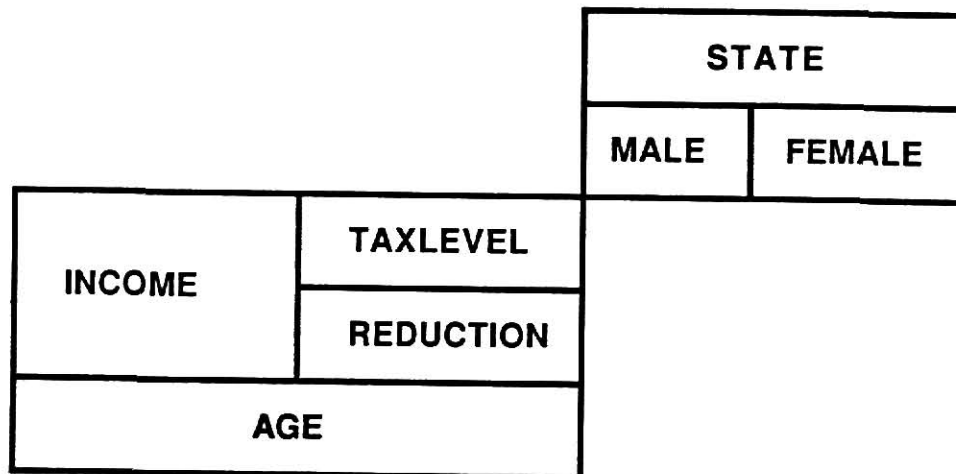


Figure 2.3.2.1b : Example summary-table scheme

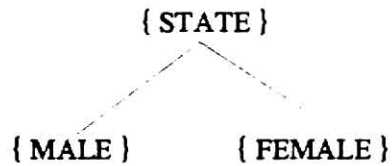


Figure 2.3.2.1c : Column category attribute tree Tc1

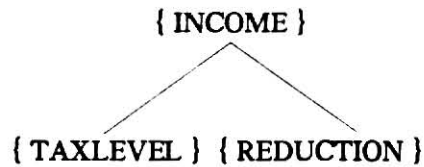


Figure 2.3.2.1d : Row category attribute forest Tr1



Figure 2.3.2.1e : Row category attribute forest Tr2

For this summary-table scheme consists of two trees in the column category attribute forest there will be two combinations of row and category attribute trees and hence, two resulting set-relations.

a) For the combination of Tr1 and Tc1 the mapping function Mp2 is performed in the following way:

Performing the steps i) a) through b) the resulting set Sr1 is:

$$Sr1 = \{ TAXLEVEL REDUCTION \}$$

The steps ii) a) through c) form the set Src1:

$$Src1 = \{ \{ MALE \{ TAXLEVEL REDUCTION \} \} \\ \{ FEMALE \{ TAXLEVEL REDUCTION \} \} \}$$

Through the step d) the set Rcr1 of the roots is produced:

$$Rcr1 = \{ \text{STATE INCOME} \}$$

Finally the set-relation S1 is achieved through step ii) f).

$$S1 = \{ \{ \text{STATE INCOME} \} \{ \begin{array}{l} \text{MALE} \\ \{ \text{TAXLEVEL REDUCTION} \} \\ \text{FEMALE} \\ \{ \text{TAXLEVEL REDUCTION} \} \} \} \}$$

b) For the combination of Tr2 and Tc1 the mapping function Mp2 is performed in the following way:

Performing the steps i) a) through c) the resulting set Sr2 is empty for Tr2 consists only of the root. The steps ii) a) through c) form the set Src2:

$$Src2 = \{ \{ \text{MALE FEMALE} \} \}$$

Through the step d) the set Rcr1 of the roots is produced:

$$Rcr1 = \{ \text{STATE INCOME} \}$$

Finally the set-relation S1 is achieved through step ii) f).

$$S1 = \{ \{ \text{STATE INCOME} \} \{ \begin{array}{l} \text{MALE FEMALE} \} \} \}$$

2.3.3 Mapping from the graphical scheme information on the conceptual level to the summary-table scheme on the external level

The graphical scheme information in the data dictionary represents the structure of the summary-table scheme from the screen. The information is represented through a set

whose elements consist of two sets that describe the appearance of the column category attribute forest and the row category attribute forest. Through the mapping function the summary-table scheme is recreated on the screen. The mapping function Mp3 that transforms the graphical scheme information in the data dictionary to the summary-table scheme on the external level is defined in the following way:

- i) For the set that represents the row category attribute forest through sets perform the following steps:
 - a) Traverse the row category attribute forest set from left to right, consider each element. The elements represent a jar trees to each other on the screen.
 - b) For each tree set of the forest set, consider the graphical information and form the rectangle levels on the screen according to the levels within the tree set. After drawing a rectangle it has to be labeled with the attribute name present in the information set that represents the nodes of the tree, e.g., { attribute_name, cell-length, cell_height }. The row category attribute tree structures are drawn one after the other from bottom to top for the row category attribute
- ii) For the set that represents the column category attribute forest through sets perform the following steps:
 - a) Traverse the column category attribute forest set from left to right, consider each element. The elements represent a jar trees to each other on the screen.
 - b) For each tree set of the forest set, consider the graphical information and form the rectangle levels on the screen according to the levels within the tree set. After drawing a rectangle it has to be labeled with the attribute name present in the information set that represents the nodes of the tree, e.g., { attribute_name, cell-length, cell_height }. The column category attribute tree structures are drawn one after the other from left to right for the column category attribute forest.
- iii) Perform the graphical connection of the row category attribute forest and the row category attribute forest by connecting the left most leaf of the left most tree of the column category attribute forest with the up most leaf of the up most tree of the row category attribute forest.

2.3.3.1 Examples for the mapping function Mp3

For the graphical scheme information is an ordered set, consisting of two elements that contain the column and row category attribute forest scheme information. For the graphical scheme information in this example the two sets representing the information for the column and row category attribute forests consist of one set each which can be equalized with the fact that the summary-table scheme must have one tree in each of its category attribute forests. Figure 2.3.3.1a illustrates the example graphical scheme information GI1.

```
GI1 = {  
  { {NATION, 16, 3}  
    { {SOCCERTEAM, 16, 3} } }  
  { {SCORES, 11, 5}  
    { {CURRENTSEASON, 18, 3}  
      {PREV_SEASON, 18, 3} } } }
```

Figure 2.3.3.1a : Example graphical scheme information GI1

From the definition of the graphical scheme information is known that GI1 appears in the form $GI(I_c, I_r)$ where I_c represents the graphical scheme information for the column category attribute forest and I_r represents the graphical scheme information of the row category attribute forest. With regard to the example I_{c1} and I_{r1} look as follows:

```
Ic1 = { {NATION, 16, 3}  
        { {SOCCERTEAM, 16, 3} } }  
  
Ir1 = { {SCORES, 11, 5}  
        { {CURRENTSEASON, 18, 3}  
          {PREV_SEASON, 18, 3} } }
```

Following the steps of the definition of the mapping the graphical scheme information that is represented by I_{r1} is considered first. In this case there is only one set in I_{r1} , which means the row category attribute forest of the summary-table scheme has exactly one tree. Then the levels of the set are considered, they form the levels of the tree on the screen. With the definition of the rectangles as a triple $RT(AN, CL, CH)$ the rectangles can be formed on the screen according to the information in the particular set, e.g., the rectangle that form the root of the summary-table scheme has the attribute name NATION, its cell length is sixteen pixels and its cell height is three pixels. The same steps have to be taken for I_{c1} except for the drawing direction. I_{r1} is drawn bottom-up and from right to left and I_{c1} is drawn bottom-up (if it had more branches from left to right). Figure 2.3.3.1b shows the final summary-table scheme on the screen.

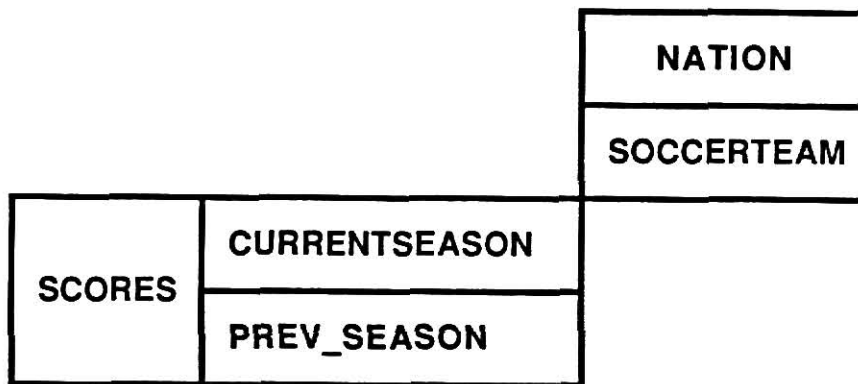


Figure 2.3.3.1b : Reproduced graphical scheme information on the screen

Chapter 3: Future aspects

3.1 Summary

The set-relational interface operates on the external level as well as on the conceptual level. On the conceptual level it provides the user with a tool to interact with a database system via summary-table schemes. The user defines summary-table schemes in order to introduce new relations to the system. The summary-tables schemes themselves can be viewed as two-dimensional arrays of cells. The rows and columns are labeled by attribute forests.

When a user introduces a new summary-table scheme to the system, two mappings from the external to the conceptual level are performed. One is the mapping of the summary-table scheme on the external level to the graphical scheme information in the data dictionary. The representation of the graphical scheme information in the data dictionary guarantees that the user defined summary-table scheme can be reproduced on the screen. The other is the mapping from the summary-table scheme to the set-relation on the conceptual level. The set-relation is based on the extended relational model and represents the summary-table scheme as a set of sets.

3.2 Queries in the set-relational interface

For the future work the retrieval and instantiation of set-relations has to be defined. If the user wants to retrieve a particular in the system existing summary-table scheme with the purpose to retrieve data or to introduce a new instance of a relation, a mapping function is performed that uses the graphical scheme information from the data dictionary previously defined, and rebuilds the summary-table scheme on the table. To retrieve data from the database another mapping function has to be performed that also uses the scheme information of the data dictionary to fill the cells of the summary-table

with the data. In future research a query request in the set-relational interface could be posed either by requiring a predefined and in the system present summary-table scheme or defining a new summary-table scheme combining the attributes represented in the system. If a query concerns an existing summary-table scheme the user can reproduce the structure on the screen by using the name of the particular summary-table scheme. Then the mapping functions are applied between the levels of architecture and the summary-table scheme filled with data can be produced on the screen.

For the future work the mapping between the conceptual level and internal level has to be defined as well as the representation of the set-relation at this level. For a query language the features introduced by Oszoyoglu and Oszoyoglu in STBE (Summary-Table-By-Example) and QBE (Query-By-Example) introduced by Zloof could be applied [OZSO85], [ZLOO72]. The user should be able to fill the tables with logical operators depending which values are to be queried on

e.g., $>$, $<$, $=$, $>=$, $<=$, that could be done in the attribute cells. If the user wants to perform a query involving for example statistical functions like average, sum, median this could be specified within an extra cell, which would appear outside the summary-table scheme itself.

3.3 Input data for the summary-table schemes

The input data for summary-tables schemes has been singled valued in the definition of Oszoyoglu and Oszoyoglu. But with regard to the set-relational interface also sets or sets of sets could be allowed as an input and output data structure to the data cells in a summary-table scheme. The set-relation defined in the set-relational interface seems quite capable to handle sets of input and output values because of its set

structure.

Regarding a particular summary-table scheme there could be an occurrence of input data that has no value for a particular data input cell because it is only a subset of the data described by the summary-table scheme. In a system where there are no null values allowed, another summary-table scheme that covers this data subset has to be introduced.

The allowance of null values in the set-relational interface would allow the user to cover different data spaces with one summary-table scheme. In other words not only exactly the data space that is described by the summary-table scheme is covered but also all subsets of this data space because it is allowed to leave certain cells empty or fill in an empty set {}.

3.4 Functional dependencies and summary-table schemes

A functional dependency is defined as follows:

Given a relation R, attribute Y of R is functionally dependent on attribute X of R - in Symbols $R.X \rightarrow R.Y$ - if and only if each X-value in R has associated with it precisely one Y - value (at any one time). Attributes X and Y may be composites, [DATE86b].

A functional dependency can be represented by a tree. Therefore the structure of a functional dependency could be defined in the following way:

Definition of the structure of a functional dependency:

A functional dependency can be described as a tree where

- i) $Root(T)$ is a set containing the determinant(s) of the functional dependency
- ii) The leaves of $Root(T)$ are sets of attributes being determined by $Root(T)$. These sets contain single attributes.

The following example shows a set F of functional dependencies and the representation of the dependency as a tree. For a functional dependency can be represented by a tree one can also think of as a summary-table scheme. This summary-table scheme can be transformed to a set-relation. This could lead to the conclusion that summary-table schemes are able to capture functional dependencies. In figure 3.4b the set of functional dependencies from figure 3.4a is represented by a summary-table scheme and figure 3.4c depicts the set-relation representation of this summary-table scheme.

$$F = \{ AB \rightarrow CDE \}$$

Figure 3.4a : Set of functional dependencies

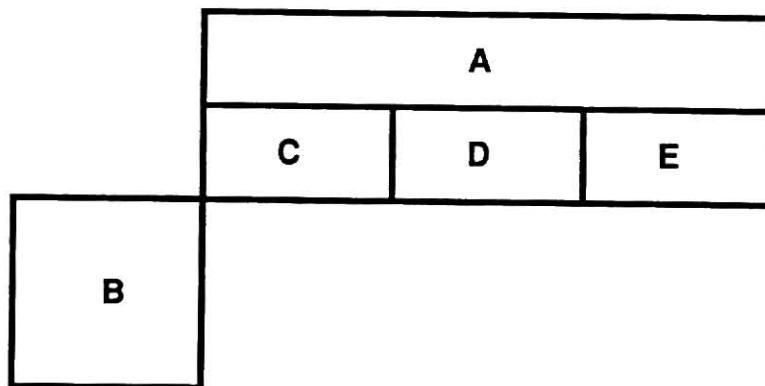


Figure 3.4b : Representation of the set of functional dependencies of figure 3.4a as a summary-table scheme

$$F_Set_relation = \{ \{A\ B\} \{ \{C\ D\ E\} \} \}$$

Figure 3.4c : Set-relation of the summary-table scheme of figure 3.4c

If sets of values are considered as a possible input for the summary-tables, this implies that also functional dependencies could be an input being represented as a set. Allowing sets of values the summary-table can represent functional dependencies in two dimensions and in an additional third dimension. Figure 3.4d shows the three dimensions that can be represented by a summary-table.

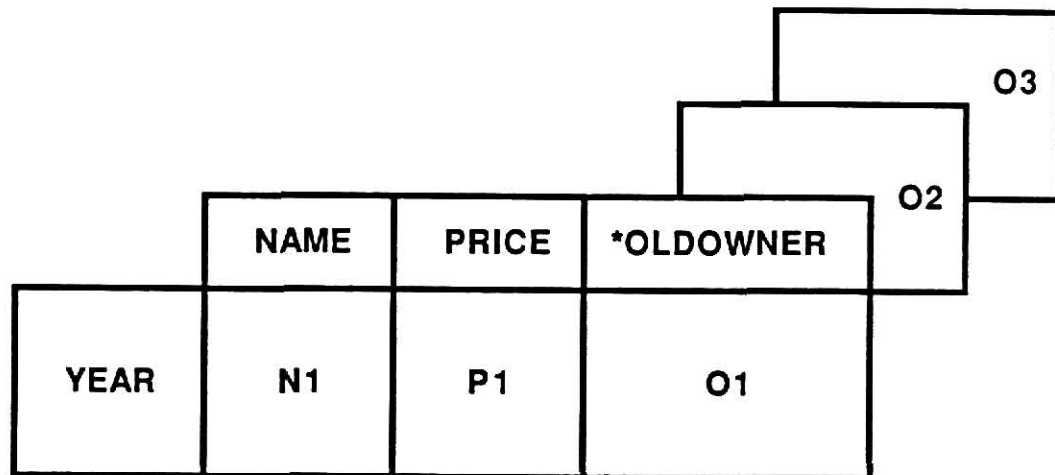


Figure 3.4d : Dimensions within a summary-table where sets are allowed as input values

This shows that functional dependencies cannot only occur within the forests of the table but also within the cells of the table itself. A major drawback for putting functional dependencies into the cells of a table is that there is no description of the functional dependency as it is for the table cell itself. If the cell contains another structure itself it (the structure) has also to be described. The ability to represent functional

dependencies as summary-tables schemes poses the question how far summary-tables schemes created by the user contain those dependencies. There is also a problem concerning the minimality that is required in the database set of functional dependencies. Future studies could be done with regard to the user and how far dependencies can be expressed within a summary-table schemes without the explicitly knowing about them as functional dependencies but merely in data that the user needs to do his work. This could bring new methods into the database design process in a way that the user is provided with a tool that has the capability to capture the information needed for a good design.

3.5 Correspondence between the set-relation and the relations of the relational model

In the relational model a relation is defined as any subset of the Cartesian product of one or more domains, [ULLM83a]. Ullman also describes a relation as a table, where each column corresponds to one attribute. Figure 3.5a shows such a relation.

CITY	STATE	POP
------	-------	-----

Figure 3.5a : Relation of the relational model

This relation of the relational model can also be regarded as a summary-table scheme that has an empty row category attribute forest and the column category attribute forest is build of three trees that consist of roots only. Figure 3.5b shows the

relation of figure as a set relation.

{ {CITY, STATE, POP} }

Figure 3.5b : Set-relation for the table in figure 3.5a

The possibility of representing a relation of the relational model as a set-relation opens the question whether the set-relational interface could be extended to a model that has the relational model as a subset. This question could be considered in future studies on this matter.

Chapter 4: Conclusion

In this paper a set-relational interface has been defined that provides the user with a means to interact with a database via summary-table schemes. For this interface a new relation called the set-relation has been introduced. So far set-relational interface encompasses the external and the conceptual level which is hoped to be extended to the internal level in future research.

The set-relational interface has been developed with the idea of finding new ways for the user to interact with a database system. At the same time it seems that this interface could lead to a new form of database design methods where the information that is provided by the user to the system via summary-table schemes provides a part of the basics needed for the design process.

In future research the capability of the summary-table schemes to capture functional dependencies has to be studied. For the set-relational interface can represent relations of the relational model in a summary-table scheme and also the relations of the extended relational model, provided the fact that sets are allowed as input data, there seems to be a potential for a new model. This model could combine three types of relations: the relation of the relational model, the relation of the extended relational model and the set-relation of the set-relational interface.

Bibliography

- [CODD72a] Codd E.F., "Further normalization of the database relational model", Database Systems, Prentice Hall, Englewood Cliffs, NJ, 1972, p. 33 - 64
- [CODD72b] Codd E.F., "Relational completeness of database sublanguages", Database Systems, Prentice Hall, 1972, 67 - 101
- [DATE85a] Date C.J., "An Introduction to Database Systems", Vol 1, Fourth Edition, Addison-Wesley Publishing Company, 1985, chapt. 2.7, p. 37
In this section Date gives a definition of the mappings between the levels of architecture in a database system.
- [DATE85a] Date C.J., "An Introduction to Database Systems", Vol 1, Fourth Edition, Addison-Wesley Publishing Company, 1985, chapt. 17.3, p. 364 - 366
A definition of functional dependencies is given in this paragraph.
- [FAGI77] Fagin R., "Multivalued Dependencies and a New Normal Form for Relational Databases", ACM Trans. on Database Systems, Vol. 2, No 3, Sept 1977, p.262 - 278
- [FISH83] Fischer P. C., Thomas S. J., "Operators for Non-First-Normal-Form relations", Department of Computer Science, Vanderbilt University, Nashville, Tennessee, IEEE Computer Soc Press, Nov 83, The IEEE Computer Society's Seventh International Software and Application Conference, Chicago Ill, Nov 83, p. 464 - 475
- [JACO82] Jacobs B. E., "On Database logic", Journal of the ACM, 29,2,1982
- [JAES82a] Jaeschke G., Schek H. -J., "Remarks on the algebra of nonfirst normal form relations", Proc. ACM SIGACT-SIGMOD Proceedings, Los Angeles, March 1982
Jaeschke and Schek present a discussion of the relational algebra that has been extended to include transformations between non first normal form relations and first normal relations.
- [JAES82b] Jaeschke G., "An Algebra of Power Set Type Relations", IBM Scientific Center Technical Report, TR 82.12.002, Heidelberg Scientific Center, West-Germany, December 1982

- [KENT83] Kent W., "A simple guide to Five Normal Forms in Relational Database Theory", Communications of the ACM, Feb 1983, p. 120 - 125
In this paper Kent gives the definition of the Five Normal Forms in Relational Database Theory as a guideline for record-design.
- [KITA80] Kitagaw H., Kunii T., "Form Transformer - The Formal Aspects of Table Nests Manipulation", University of Tokyo 1980
A powerful office data modeling tool, the nested table data model, is presented. The manipulation of such tables using four form transformer operators is shown.
- [KLUG82] Klug A., "Equivalence of relational algebra and relational calculus query languages having aggregate functions", Journal of the ACM, 29, 1982 p. 699 - 717
- [KNUT69] Donald E. Knuth, "The Art of Computer Programming", Volume 1 / Fundamental Algorithms, Addison-Wesley Series in Computer Science and Information Processing, 1969, p. 305
- [KROE83] Kroenke David M., "Database Processing: Fundamentals, Design, Implementation", Second Edition, Science Research Associates. Inc. chapter 7, p. 252
Kroenke describes the different categories of relational data manipulation languages.
- [OZSO83a] Ozsoyoglu G., Matos V., Ozsoyoglu Z.M., "Extending relational algebra and relational calculus for set-valued attributes and aggregate functions", Technical Report, CES 83-14, CWRU, 1983
- [OZSO83b] Ozsoyoglu G., Ozsoyoglu Z.M., "Features of SSDB", Proc. 2nd LBL SDB Workshop, 1983
- [OZSO83c] Ozsoyoglu G., Ozsoyoglu Z.M., "An extension of relational algebra for summary-tables", Proc. Second LBL SDB Workshop, 1983

- [OZSO85] Ozsoyoglu G., Ozsoyoglu Z.M., "A Query Language for Statistical Databases", Topics in Information Systems, Query Processing in Database Systems, Springer Verlag 1985, p. 171 - 187.
Ozsoyoglu and Ozsoyoglu present a high-level screen-oriented query-language for statistical databases called Summary-Table-By-Example (STBE). This query language is based on extended relational calculus and allows set-valued attributes and aggregate functions.
- [SCHE82] Schek H.-J., Pistor P., "Data Structures for an Integrated Data Base Management and Information Retrieval System", VLDB 1982
An Interface to a database for office information systems which allows integration of classically formatted data with textual information and its manipulation is presented which uses the extended relational model.
- [ULLM83a] Ullman Jeffrey D., "Principles of Database Systems", Computer Science Press, 1983, chapt. 1.4, p. 19 - 20
In this passage Ullman presents the description of the Relational Data Model .
- [ULLM83b] Ullman Jeffrey D., "Principles of Database Systems", Computer Science Press, 1983, chapt. 6.5, p. 197 - 209
This chapter deals with Query-By-Example (QBE) which is a domain calculus language and was developed at IBM, Yorktown Hts.
- [ULLM83c] Ullman Jeffrey D., "Principles of Database Systems", Computer Science Press, 1983, chapt. 7.4, p. 234 - 235
The Third Normal Form and Boyce-Codd Normal Form definitions are presented.
- [ZLOO77] Zloof M.M., "Query-by-Example: a data base language", IBM Systems Journal, 1977, Vol 16, Number 4, p. 324 - 343 In this paper the high level query language Query-by-Example is discussed. This query language is based on domain calculus. In order to retrieve data the user has to fill a skeleton with a possible answer of a query.

A NON-NORMAL FORM DATABASE INTERFACE

by

GABRIELE A. HECKER

BS., Fachhochschule Furtwangen, West-Germany

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

**KANSAS STATE UNIVERSITY
Manhattan, Kansas**

1986

Abstract

The problem of allowing the user to build and manipulate databases from a graphical screen interface has been approached. A set-relational interface has been defined based upon examination of the tabular structure the user draws on the screen. This tabular structure is called a summary-table scheme. A relational form called the set-relation has been introduced to accomodate sets of values and sets of sets. The definition of these data structures and the mappings between them are given in this work.