Faculteit der Economische Wetenschappen en Econometrie

Serie Research Memorandum

The Necessity and Contours of a Catalog Standard for RM/V2

E.R.K. Spoor
R.J. Veldwijk
R.B. Buitendijk
M. Boogaard

Research Memorandum 1991-67.

vrije Universiteit *amsterdam*

The Necessity and Contours of a Catalog Standard for RM/V2

E.R.K. Spoor
R.J. Veldwijk
R.B. Buitendijk
M. Boogaard

This paper considers the definition of the Relational Model Version 2 (RM/V2) from a database designer's point of view. As RM/V2 offers concepts to express formal models in terms of relational tables and integrity constraints, it can very well be itself a subject for database design. Such a design act would yield a result that can be applied as a catalog standard. The paper argues that a relational catalog standard is needed for many purposes, which include the obvious demand for a RDBMS catalog standard, the assessment of RM/V2 and its proposed enhancements, the evaluation of RDBMSs and CASE-tools that claim to support the Relational Model, the assessment of application database designs, and educational purposes. As the complete design of the catalog standard requires a lengthy and rather technical discussion, this paper presents the contours of the overall design and treats only the core of RM/V2 in detail.

Keywords: catalog, data dictionary, data model, meta data, meta modelling, relational model

1. INTRODUCTION

The Relational Model of Data is a powerful tool for describing the structural and integrity aspects of formal models of reality. The Relational Model itself, though a formal model, appears to be rarely expressed by means other than detailed and careful verbal descriptions or mathematical notations. Codd's latest book on version 2 of the model [CODD90], which forms the foundation of this paper, is a good example of such a practice. The discussions in his book about the concepts of the model remain primarily verbal. An attempt to formalize the Relational Model can be found in [DATE83]. Date utilises a mathematical notation to define the model's concepts.

Both Codd and Date consider the Relational Model only from a definitial point of view. They persist in introducing a number a interrelated relational concepts and discussing their relevance. However, it would be very interesting to go beyond this definitial level and consider the Relational Model itself as a domain for database design purposes. The design of the (meta-)data structures and integrity constraints of this domain would enhance the understanding of the Relational Model, in particular by the relational database designer. Moreover, since there is no essential difference between data and meta-data, such a design is self-evident [CURT81] [ROSS81] [VELD91b].
All the way, the design of the structural and integrity aspects of the Relational Model serves many purposes, as will become apparent in the remainder of this paper. The most important function, though, is that of a catalog standard for Relational Database Management Systems.

An application of a self-modelling procedure with respect to a data model can be found in [NIJS89]. We aim to do the same for the Relational Model, more specifically Codd's latest version of this model (RM/V2). The result is a catalog model that describes RM/V2: a proposal for catalog standard.

Because a full and formal description of the catalog model gains the size of a booklet to cover all thirteen aspects of RM/V2, this paper only describes the details of one aspect (i.e. the Basic Model, see Table 1), while the other aspects are treated very briefly. Moreover, only the result of the design process is presented and explained, its derivation is not discussed. A description of the complete catalog model, including the SQL expressions of all constraints that are expressible in a Relational Language (RL) can be found in [VELD91c].

1	The Basic Model
2	Composite Columns and Composite Domains
3	Archives, Snapshots and Relational Assignments
4	Views
5	Indexes
6	Constraints and Triggered Actions
7	Built-in and User-Defined Functions
8	Synonym Names and Naming Rules
9	Database Users
10	Authorization
11	Audit Logging
12	Distributed Database
13	Database Statistics

Table 1: Aspects of an RM/V2 catalog model

The subdivision of the paper is as follows. In Section 2 several arguments are given that illuminate the need for a catalog standard. Section 3 describes the structure and constraints of the Basic Model. Only those relations, attributes and constraints that are relevant to this Basic Model are included. The extensions due to the treatment of the other aspects of the catalog model are excluded. In Section 4 some statements are made about the design complexity of the complete model. Appendix A of the paper shows three diagrams that together constitute the complete data structure of the catalog model. A list containing all attributes and constraints is not included because it would exceed the paper's limits.

2. WHY A CATALOG MODEL?

The proper design of the structure and constraints of RM/V2 by means of its own mathematically-defined primitives results in a description that can be used for a number of interrelated purposes. We can identify at least nine such purposes.

First, support of Data Base Management Systems for a data model requires formalization of that model in the DBMS software. Modelling the data model is thus a prerequisite for successful DBMS implementation. More specifically, RM/V2 demands that an RDBMS has a self-descriptive catalog, accessible to the user by means of a RL. The catalog is a special purpose relational database used by the RDBMS to store among others the structural and integrity aspects of application databases. The representation in the catalog of the catalog itself is equivalent to a relational representation of the Relational Model, at least of its non-manipulative aspects. The RM/V2 description to be presented can thus be used as input for RDBMS design.

Second, a RM/V2 catalog model should be used as a basis for a Relational Catalog standard. The lack of catalog standardization presently limits application portability and meta-data accessibility, but in the future may very well become a major bottleneck for achieving heterogeneously distributed DBMSs.

Third, a catalog model can be used as a means to measure the faithfulness to RM/V2 of both RDBMSs and Dictionary tools that claim to be useful in the relational database design process. A viable procedure to achieve this is to implement the catalog together with a database description that exploits every construct allowed by RM/V2. If it is not possible to transfer the catalog contents to the RDBMS or Dictionary product, this product does enable the exploitation of all the features of RM/V2. If the contents of the RDBMS or Dictionary product cannot be transferred to the RM/V2 catalog this product allows non-relational constructs, resulting in poor and non-portable database designs. In either case, the RDBMS or Dictionary product is not fully relational as far as the non-manipulative aspects of RM/V2 are concerned.

Fourth, the process of deriving a catalog model may reveal omissions, inconsistencies and ambiguities that have been overlooked by readers or writers of verbal descriptions of RM/V2.

Fifth, a catalog model is useful for assessing and discussing the merits of proposed enhancements of RM/V2, because the model combines formality with compactness and understandability. Furthermore, the effects of proposed enhancements on the complexity and the elegance of the catalog model itself provide a yardstick against which proposed enhancements can be measured. For example, the design of the catalog model appeared to be quite complicated due to the inability of RM/V2 to deal with generalization problems [VELD91c]. The promised introduction of generalization support in RM/V3 [CODD90,p.480] should thus have a profound and positive influence on the RM/V3 catalog model.

Sixth, a catalog model is a powerful tool for the education of database designers. A catalog model that is implemented in RDBMSs and CASE tools not only prevents the designer from devising 'forbidden' database models, but also offers insight in the Relational Model itself. Studying the model and querying the catalog self-description may be the easiest and most efficient way to gain insight in the Relational Model.

Seventh, besides preventing database designers to devise non-relational databases, a catalog model is a powerful tool for assessing (1) whether all RM/V2-concepts are covered by a database design and (2) whether this database design is internally consistent. The first criterion is satisfied if all appropriate catalog tables are filled, while satisfaction of the second criterion requires all constraints not to be violated. During the (probably long) period in which available RDBMS products will not fully support RM/V2, an implementation of the catalog model provides organizations with a means to manage the quality and portability of their application database designs. The catalog model thus becomes a temporary means to protect investments in non-fully relational DBMS environments.

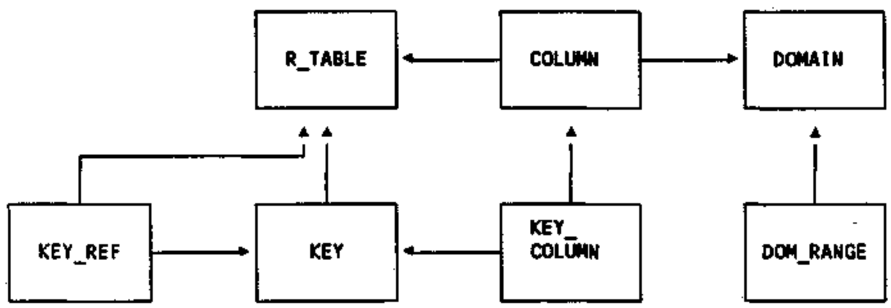
Eighth, a catalog model permits the experienced database designer to design his application database as an extension to the catalog model. This blurring of data and meta-data makes it possible to design extremely flexible applications that are cost-effective in environments characterized by unpredictable change, like management information systems [CURT81] [VELD90] [BOOG91]. For these implementations catalog standardization (see item 2) is exceptionally important.

Finally, a catalog model can serve as an aid in research directed at developing software tools and models that enable organizations to cope with evolutional change of their applications [SHNE82] [VELD91a]. The basis for this research is the observation that the extent to which the Relational Model supports logical data independence is quite poor.

Many of the arguments presented above are reminiscent of Codd's description of the use of data models in database management [CODD81]. In this respect it is unfortunate that Codd seems to be opposed to "*casting as much as possible of the system's behaviour into data structure*" [CODD90,p.244]. This is in contrast with both database design theory (normalization theory) and practice, but it also obstructs crucial activities like catalog standardization [CODD90,p.424] and catalog extension [CODD90,p.278]. It would be very interesting to compare a proposed catalog standard according to Codd's design criteria to the catalog model presented in this paper, which is based on a conventional database design approach.

3. DESIGN OF THE BASIC MODEL

Basic to the Relational Model are concepts like 'Relation', 'Domain', 'Attribute', 'Key', 'Entity Integrity' and 'Referential Integrity'. Figure 1 depicts a relational schema that expresses these concepts. The boxes represent relations, the arrows represent foreign key to primary key references between these relations.




```
R_TABLE(RT_NM, RT_TYPE, RL_EXPR)
COLUMN(RT_NM, COL_NM, COL_SEQ#, DOM_NM, A_MARK_ALL, I_MARK_ALL)
DOMAIN(DOM_NM, DOM_TYPE, COMP_ALL, BD_TYPE_NM, LENGTH, SCALE)
KEY_REF(RT_NM_PK, RT_NM_FK, KEY_NM)
KEY(RT_NM, KEY_NM, KEY_TYPE)
KEY_COLUMN(RT_NM, KEY_NM, COL_NM, KEY_COL_SEQ#)
DOM_RANGE(DOM_NM, DR_SEQ#, LOVAL, HIVAL)
```

Figure 1: A basic model of the Relational Model

An R_TABLE (or relation) in the Relational Model must have a unique name (RT_NM). An R_Table is either a base relation or a view (RT_TYPE). Views are defined in terms of other views and/or base relations by means of a statement in a relational language (RL_EXPR).

A COLUMN (or attribute) is identified by its name together with the name of the relation to which it belongs (RT_NM, COL_NM). Columns are assigned a sequence number for default presentation purposes. Every column belongs to a domain (DOM_NM). Columns that do not belong to the primary key can be allowed to be unknown but applicable (A_MARK_ALL), unknown but inapplicable (I_MARK_ALL) or both.

A DOMAIN must have a unique name (DOM_NM). If a primary key column draws its values from a domain, it is a primary domain (DOM_TYPE). An indicator (COMP_ALL) designates whether comparison of domain values is meaningful. Domains are extended data types based on DBMS dependent basic data types (BD_TYPE_NM) like 'character', 'number' and 'date'. Depending on the datatype the columns LENGTH en SCALE may be applicable.

Aside: Although Codd only speaks of "*a range of values*" permitted in columns belonging to a domain we believe multiple ranges of values are not forbidden. Admitting only one range of values per domain impairs the interpretation by catalog users of the meaning of the domain and leads to unnecessary domain constraints. Disallowing multiple ranges amounts to abolishing the view of domain as 'a pool of values' [CODD70].

The relation DOM_RANGE permits the specification of multiple ranges of values per domain by means of the columns LOVAL and HIVAL. If no range of values is specified for a domain, the range of values is only limited by the basic data type assigned to the domain.

A KEY is formed by one or more columns of one relation that identify a tuple in some relation. Primary and alternate keys identify tuples in the R-table to which these keys belong themselves. Foreign keys identify tuples in one or more R-tables designated by KEY_REF, if none of the foreign key columns contains an A-mark.

Aside: Codd makes no mention of the term candidate key or alternate key any more. Alternate keys are considered to be user-defined constraints. However, he does specify that for each column the DBA should be able to specify that all values of a column are required to be distinct from one another [CODD90,p.156], which amounts to alternate key support. Because full support for alternate keys leads to a less complicated catalog model such keys have been included.

KEY_COLUMN contains the columns constituting a key. A key column is identified by the combination of the identifiers of the R-tables KEY and COLUMN. The column KEY_COL_SEQ# provides the mechanism to couple a foreign key column to a primary key column. Composite primary/foreign key combinations must have matching sequence numbers for their columns. It is not possible to use the rule that matching primary and foreign keys should have the same domain because more than one column in a key can belong to a given domain. A foreign key references at least one R-table. Every such reference is expressed by a KEY_REF-tuple.

It appears that the core aspect of the Relational Model can be expressed quite concisely and, at least in our opinion, elegantly by means of the Relational Model itself. Nevertheless, there remain a number of constraints that are not enforced by the data structure of Figure 1. These constraints have to be expressed in an ad hoc manner by means of RL and stored in the catalog. Table 2 lists the constraints pertinent to the basic model of Figure 1.

BM01-	Every tuple in R_TABLE is referenced by at least one COLUMN-tuple.
BM02-	No view definition (RL_EXPR) is directly or indirectly recursive.
BM03-	Every tuple in R_TABLE for which RT_TYPE = 'BASE' or 'CHECKOUT' (see section 4) is referenced by a KEY-tuple having KEY_TYPE = 'PRIMARY'.
BM04-	The subset of KEY-tuples for which KEY_TYPE = 'PRIMARY' does not contain duplicate values for RT_NM.
BM05-	Every tuple in KEY having KEY_TYPE = 'FOREIGN' is referenced by at least one KEY_REF-tuple, while other tuples in KEY are not referenced by KEY_REF-tuples.
BM06-	Every tuple in KEY_REF references a tuple in R_TABLE that is referenced by a tuple in KEY having KEY_TYPE = 'PRIMARY'.
BM07-	Every tuple in KEY is referenced by at least one KEY_COLUMN-tuple.
BM08-	KEY-tuples with KEY_TYPE = 'PRIMARY', that are referenced via KEY_REF by a KEY-tuple with KEY_TYPE = 'FOREIGN', are referenced by pairs of KEY_COLUMN-tuples with corresponding values for KEY_COL_SEQ#, referencing COLUMN-tuples with the same value for DOM_NM.
BM09-	No two KEY-tuples, for which KEY_TYPE = 'PRIMARY', 'FOREIGN' or 'ALTERNATE', can be referenced by identical sets of KEY_COLUMN-tuples, except for key combinations in which the values of KEY_TYPE are either 'PRIMARY' and 'FOREIGN' or 'ALTERNATE' and 'FOREIGN'.
BM10-	The column RL_EXPR in R_TABLE contains an I-mark if and only if RT_TYPE = 'BASE'.
BM11-	The values of COL_SEQ# in the set of COLUMN-tuples with the same value for RT_NM are numbered consecutively up from 1 to the number of tuples in the set.
BM12-	The values of KEY_COL_SEQ# in the set of KEY_COLUMN-tuples with the same value for RT_NM and KEY_NM are numbered consecutively up from 1 to the number of tuples in the set.
BM13-	No tuple in COLUMN that is referenced by a tuple in KEY_COLUMN that references a tuple in KEY having KEY_TYPE = 'PRIMARY' has a 'YES' value for either A_MARK_ALL or I_MARK_ALL.
BM14-	No tuple in COLUMN that is referenced by a tuple in KEY_COLUMN that references a tuple in KEY having KEY_TYPE = 'FOREIGN' has a 'YES' value for I_MARK_ALL.
BM15-	Every value of LOVAL is less than or equal to the value of HIVAL in the same DOM_RANGE-tuple.
BM16-	The values of DR_SEQ# in the set of DOM_RANGE-tuples with the same value for DOM_NM are numbered consecutively up from 1 to the number of tuples in the set.
BM17-	Tuples in DOM_RANGE with the same value for DOM_NM do not have overlapping ranges.
BM18-	Every value of LOVAL or HIVAL belongs to the basic data type of the DOMAIN-tuple referenced by the DOM_RANGE-tuple.

Table 2: User-defined constraints on the basic model

The majority of these constraints require no elaboration and are easily expressed in RL (see [VELD91c]). Exceptions are constraint BM08 and BM09, which are quite complex, and constraints BM02 and BM18, which cannot be expressed in RL, but must be implemented in a host language. An example of constraint formulation can be found in figure 2. This figure shows the SQL representations of the constraints BM01 and BM09.

4. ABOUT THE COMPLETE DESIGN AND BEYOND

Though with different degrees of complexity, the design of the structure and constraints of the other twelve aspects mentioned in table 1 proceeds in quite the same way as the design of the first aspect. To give an idea of the complexity, consider table 3, which shows, for each of the thirteen aspects of RM/V2, (1) the number of involved R_TABLEs, (2) the number of foreign key to primary key references (FK-refs), and (3) the number of user-defined constraints. Appendix A shows the complete data structure of the catalog model. We restate that a manifest of the complete catalog model, including all attributes and the SQL expressions of all RL-expressible constraints can be found in [VELD91c].

Aspect	R_TABLEs	PK-refs	Constraints
1	7	8	18
2	4	5	9
3	3	2	3
4	3	3	5
5	7	8	8
6	7	11	27
7	7	9	10
8	8	7	6
9	6	6	3
10	11	14	16
11	8	10	4
12	26	34	17
13	6	7	3

Table 3: Design complexity of RM/V2 aspects

The form in which the catalog model is expressed is more important than it may seem. Summarizing the major arguments given in Section 2, we state that if the model is looked upon as *just another relational database*, the artificial and arbitrary distinction between data and meta-data is dropped. This invites database designers to design their databases as an extension to the catalog model, thereby creating the opportunity to design applications that are highly reusable and flexible. Another consequence is that the catalog model becomes a database design standard at the same time, because it implicitly shows what a complete relational database design looks like. Viewing the catalog as just a database may also serve to convince RDBMS vendors that Data Definition Language (DDL) and Data Control Language (DCL) are in fact redundant, because DDL- or DCL-statement can be translated to one or more DML-statements on the catalog database [BUIT91] [VELD91a] [WARD90].

Baring in mind the complexity of Relational Model Version 2, we are aware of the fact that our proposal can be no more than a first step on the road to the design of a real catalog standard for RM/V2. In this paper, we therefore payed specific attention to the motives underlying the creation of a catalog model. In doing so, we hope to initiate a discussion that triggers other and possibly improved designs and that eventually leads to a consensus for a RM/V2 catalog standard.

BM01- Every tuple in R_TABLE is referenced by at least one COLUMN-tuple.

```
SELECT DISTINCT 'BM01: R_TABLE-TUPLE NOT REFERENCED BY COLUMN-TUPLE' FROM R_TABLE
WHERE (USER_NM, RT_NM) NOT IN
        (SELECT USER_NM, RT_NM FROM COLUMN)
```

BM09- No two KEY-tuples, for which KEY_TYPE = 'PRIMARY', 'FOREIGN' or 'ALTERNATE', can be referenced by identical sets of KEY_COLUMN-tuples, except for key combinations in which the values of KEY_TYPE are either 'PRIMARY' and 'FOREIGN' or 'ALTERNATE' and 'FOREIGN'.

```
SELECT DISTINCT 'BM09: R-TABLE HAS TWO IDENTICAL KEYS (FK-FK, AK-AK, PK-AK)'
FROM KEY K1, KEY K2
WHERE K1.USER_NM = K2.USER_NM
  AND K1.RT_NM   = K2.RT_NM
  AND K1.KEY_NM <> K2.KEY_NM
  AND (
        K1.KEY_TYPE = K2.KEY_TYPE
      OR (
             K1.KEY_TYPE = 'PRIMARY'
         AND K2.KEY_TYPE = 'ALTERNATE'
        )
      )
  AND NOT EXISTS
     (SELECT * FROM KEY_COLUMN KC1
      WHERE KC1.USER_NM = K1.USER_NM
        AND KC1.RT_NM   = K1.RT_NM
        AND KC1.KEY_NM  = K1.KEY_NM
        AND NOT EXISTS
           (SELECT * FROM KEY_COLUMN KC2
            WHERE KC2.USER_NM = K2.USER_NM
              AND KC2.RT_NM   = K2.RT_NM
              AND KC2.KEY_NM  = K2.KEY_NM
              AND KC2.COL_NM  = KC1.COL_NM
           )
     )
  AND NOT EXISTS
     (SELECT * FROM KEY_COLUMN KC2
      WHERE KC2.USER_NM = K2.USER_NM
        AND KC2.RT_NM   = K2.RT_NM
        AND KC2.KEY_NM  = K2.KEY_NM
        AND NOT EXISTS
           (SELECT * FROM KEY_COLUMN KC1
            WHERE KC1.USER_NM = K1.USER_NM
              AND KC1.RT_NM   = K1.RT_NM
              AND KC1.KEY_NM  = K1.KEY_NM
              AND KC1.COL_NM  = KC2.COL_NM
           )
     )
```

Figure 2: The constraints BM01 and BM09

References:

[BOOG91] **Boogaard M, Veldwijk R J, Spoor E.R.K. and Dijk M V van,** 'On Generalization in the Relational Model' *Research memorandum 1991-37 Vrije Universiteit, Amsterdam* (1991)

[BUIT91] **Buitendijk R B and Lek H van der,** 'Direct Manipulation of a Data Dictionary with SQL' *Information Systems* Vol 16 No 3 (1991) pp. 323-333

[CODD70] **Codd E F** 'A Relational Model of Data for Large Shared Data Banks' *Communications of the ACM* Vol 13 No 6 (1970)

[CODD81] **Codd E F** 'Data Models in Database Management' *ACM SIGMOD* Record 11 No 2 (february 1981)

[CODD90] **Codd E F** *The Relational Model for Database Management, Version 2* Reading, Massachusetts: Addison-Wesley Publishing Company (1990)

[CURT81] **Curtice R M** 'DATA DICTIONARIES: an assessment of current practice and problems' *Proceedings of the 7th International Conference on Very Large Databases* pp. 564-570 (1981)

[DATE83] **Date C J** 'A Formal Definition of the Relational Model' in *Relational database, Selected Writings* Reading, Massachusetts: Addison-Wesley Publishing Company (1989)

[NIJS89] **Nijssen G M and Halpin T A** *Conceptual Schema and Relational Database Design, A Fact Oriented Approach* Sydney, Australia: Prentice Hall (1989)

[ROSS81] **Ross R G** *Data Dictionaries and Data Administration* AMACOM (1981)

[SHNE82] **Shneiderman B and Thomas G** 'An Architecture for Automatic Relational Database System Conversion' *ACM Transaction on Database Systems* Vol 7 No 2 (1982)

[VELD90] **Veldwijk R J, Dijk M V van, Boogaard M and Spoor E R K** 'Generalization in a Relational Environment: An approach to reduce Maintenance', In Dutch: 'Generalisatie in een Relationele Omgeving: Een benadering gericht op het terugdringen van onderhoud' *Research memorandum 1990-57 Vrije Universiteit, Amsterdam* (1990)

[VELD91a] **Veldwijk R J, Boogaard M, Dijk M V van and Spoor E R K** 'EDSOs, Implosion and Explosion: Concepts to Automate a Part of Application Maintenance in Relational Databases' *Information and Software Technology* Vol 33 No 5 (june 1991)

[VELD91b] **Veldwijk R J, Spoor E R K, Dijk M V van and Boogaard M** 'On The Expressive Power of the Relational Model, a Database Designers Point of View' *Research memorandum 1991-59 Vrije Universiteit, Amsterdam* (1991). To appear in the proceedings of the 12th International Conference on Information systems New York December 1991

[VELD91c] **Veldwijk R J, Buitendijk R B, Spoor E R K and Boogaard M** 'Towards a Catalog Standard for the Relational Model Version 2: A Manifesto' *Research memorandum 1991-50 Vrije Universiteit, Amsterdam* (1991).

[WARD90] **Warden A** 'Chivalry', In C.J. Date *Relational Database, Writings 1985-1989* Reading, Massachusetts: Addison-Wesley Publishing Company, 1990.

APPENDIX A

This appendix shows the complete data structure of the Catalog Model. The structure has been deliberately portioned into three, mutually overlapping diagrams (i.e. the figures A1, A2, and A3) to benefit understanding.

Figure A1 presents the catalog structure of R-tables that are required for the support of the first 6 aspects of RM/V2 (see Section 1), i.e., (1) the Basic Model, (2) Composite Columns and Composite Domains, (3) Archives, Snapshots and Relational Assignments, (4) Views, (5) Indexes, and (6) Constraints and Triggered Actions.

The R-table structure given in Figure A2 models the aspects 7 - 11, i.e. (7) Built-in and User-Defined Functions, (8) Synonym Names and Naming Rules, (9) Database Users, (10) Authorization, and (11) Audit Logging.

Figure A3, finally, shows an extension of the previous two diagrams with the purpose to include the aspects 12 and 13, i.e. Distributed Databases and Database Statistics.

The abbreviations used in the diagrams have to be interpreted as follows (Table A):

R_TABLE	Relational Table (i.e. Base Table, View, etc.)
KEY_REF	Foreign Key Reference
KEY_COLUMN	Column in Key
DOM_RANGE	Domain Range
COLUMN_STRUCT	Column Composite Structure
DOMAIN_STRUCT	Domain Composite Structure
DB_INDEX	Domain Based Index
DB_INDEX_COL	Column of DB Index
COND_ACT	Conditional Action (Triggered Action or Constraint)
RT_CA	R-table Conditional Action association
COL_CA	Column Conditional Action association
RT_FUNC	Function for R-table
RT_SYN	Synonym for R-table
RL_KEYWORD	Relational Language (reserved) Keyword
RT_AUTH	R-table Authorization
COL_AUTH	Column Authorization
TERM_DFLT	Terminal Default
PROG_DFLT	Program Default
LOG_PK_COLUMN	Logging of Primary Key Column values
RT_SITE	R-table Site
COL_SITE	Column Site
CA_SITE	Conditional Action Site

Table A: Explanation of abbreviations

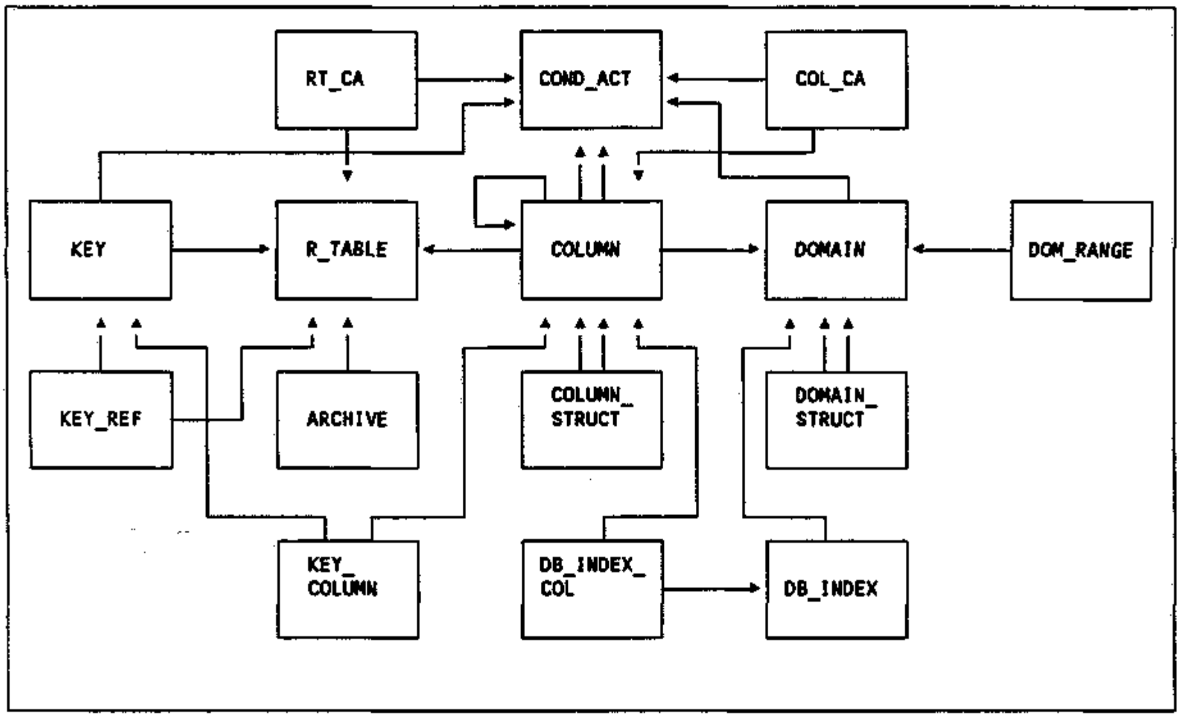


Figure A1: Structure of R-tables for the aspects 1 - 6

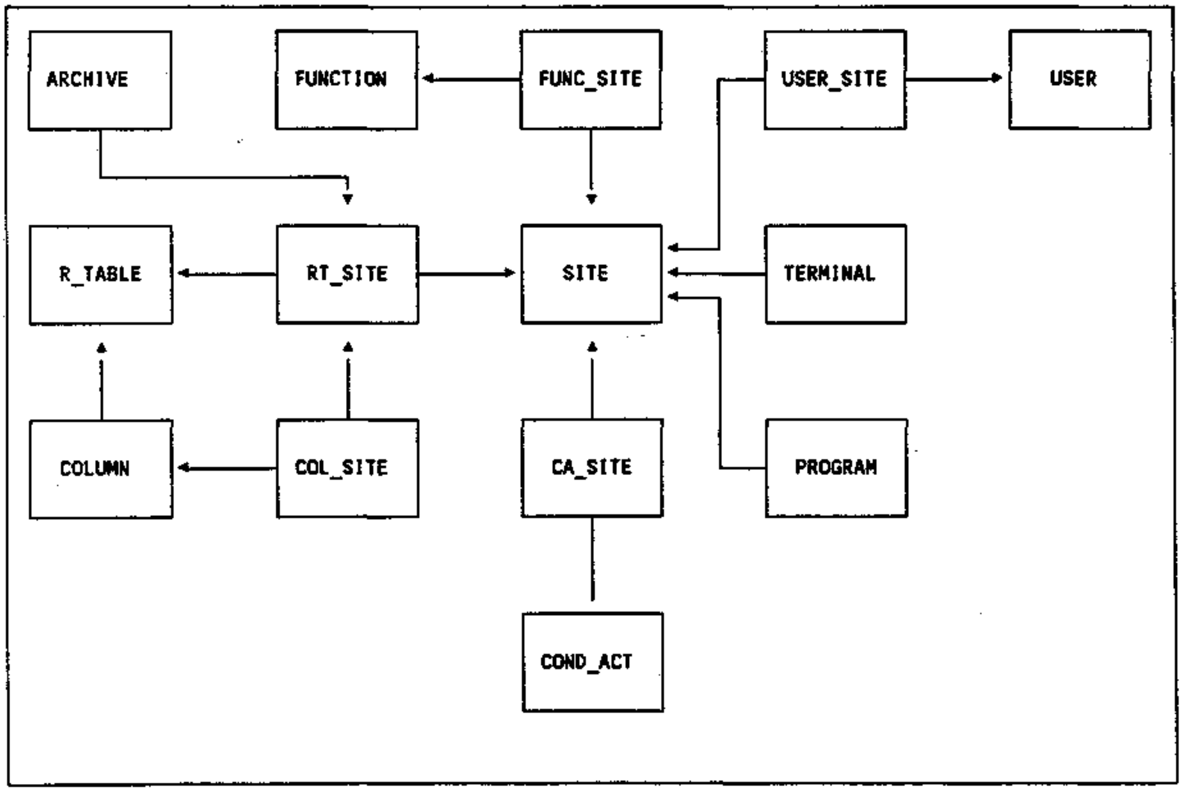


Figure A3: Structure of R-tables for the aspects 12 - 13

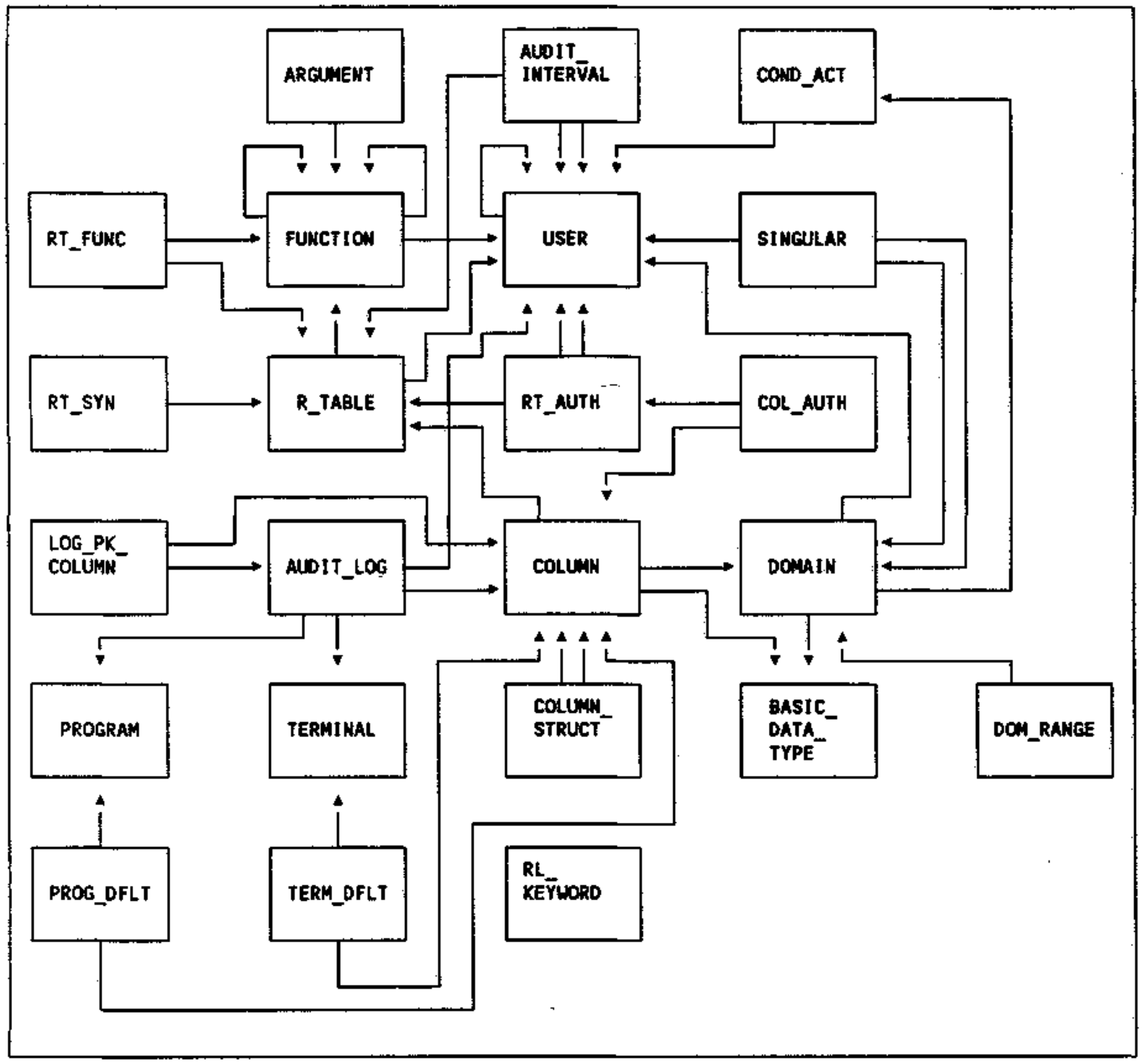


Figure A2: Structure of R-tables for the aspects 7 - 11

INTRODUCTION

The MESDAG project is a joint project endorsed by three organizations in the Netherlands: the N.V. Nederlandse Spoorwegen (The Netherlands Railways Company), RAET N.V. and the Vrije Universiteit of Amsterdam. The MESDAG project originated at RAET N.V. during the second half of 1989 as an outgrowth of research done in the field of active data dictionary models. This research and a prototype of an active data dictionary form the basis for the mission of the MESDAG project that officially started its activities in September 1990.

MESDAG is an abbreviation of:

MEta Systems Design And Generation

MISSION AND OBJECTIVES

The mission of the MESDAG project is to prove the feasibility of developing inherently flexible information systems by introducing higher levels of logical data independence.

Derived from this mission following are the two main objectives:

1. Examine the feasibility and initiate the development of an active, self-referential data dictionary model in which both a description of the database data and a description of all specifiable application design data can be stored. This data dictionary model should contain sufficient semantic aspects (like domains, constraints and time aspects) to assure the integrity, consistency and validity of the stored (meta) data, to avoid maintenance and to support query-formulation independent of current database structure.

2. Examine the feasibility and initiate the development of the possibilities of data dictionaries in general and the described data dictionary in specific. This analysis of possibilities is directed at the embedding in and developing methods, techniques, methodologic guidelines and automated tools for the design, implementation and maintenance of flexible information systems.

MEMBERS OF THE MESDAG RESEARCH GROUP

1. Dr. E.R.K. Spoor
Dr. E.R.K. Spoor is associate professor at the Vrije Universiteit Amsterdam. He teaches and consults in the area of database systems and database development with a focus on the use of these technologies in organizations. His eighteen years of experience with computer technology includes eight years with NCR and six years with the Vrije Universiteit, first as a systems engineer and later as a computer scientist. He is one of the founders and board members of two automation oriented organizations: PSB (Amsterdam) and VDA (Hilversum).

2. Drs. R.J. Veldwijk
Drs. R.J. Veldwijk graduated from the Vrije Universiteit Amsterdam in 1986. In his quality as consultant at RAET N.V. Utrecht, he is among others responsible for the design and implementation of data models. His main interest lies in developing and implementing self-knowledgeable database models, aimed at reducing maintenance costs and at improving the accessibility of databases by end-users. Furthermore he teaches courses in data modelling.

3. Drs. M. Boogaard
Drs. M. Boogaard is assistant researcher at the Vrije Universiteit Amsterdam. Furthermore, he is part-time involved in projects by the Netherlands Railways Company. He graduated from the Vrije Universiteit Amsterdam, in August 1990. The objective of his research is to develop an approach to achieve higher levels of logical data independence for both end-users and application programs and to analyze the consequences of the level of logical data independence accomplished on the system development life cycle in general and on software maintenance and database inquiry in particular.

Guest co-author dr. R.B. Buitendijk
After completing the study of Computer Science at the Vrije Universiteit Amsterdam in 1987, dr. R.B. Buitendijk received his PhD in April 1991. He is currently employed by the Netherlands Railways Company as a consultant primarily focused on databases, data management, data modelling, and CASE-tools.

ACCOMMODATION ADDRESS

Vrije Universiteit
Faculteit Economie & Econometrie
Vakgroep BIK
De Boelelaan 1105
1081 HV Amsterdam   Phone: +31-20-548708
The Netherlands   Fax : +31-20-6462645