# Relative Information Capacity of Simple Relational Database Schemata[1]

(Extended Abstract)

Richard Hull

Department of Computer Science
University of Southern California
Los Angeles, California 90089-0782
USA

## Abstract

Fundamental notions of relative information capacity between database structures are studied in the context of the relational model. Four progressively less restrictive formal definitions of "dominance" between pairs of relational database schemata are given. Each of these is shown to capture intuitively appealing, semantically meaningful properties which are natural for measures of relative information capacity between schemata. Relational schemata, both with and without key dependencies, are studied using these notions. A significant intuitive conclusion concerns the informal notion of relative information capacity often suggested in the conceptual database literature, which is based on accessability of data via queries. Results here indicate that this notion is too general to accurately measure whether an underlying semantic connection exists between database schemata. Another important result of the paper shows that under any natural notion of information capacity equivalence, two relational schemata (with no dependencies) are equivalent if and only if they are identical (up to re-ordering of the attributes and relations). The approach and definitions used here can form part of the foundation for a rigorous investigation of a variety of important database problems involving data relativism, including those of schema integration and schema translation.

[1]This work supported in part by NSF grants IST-81-07480 and IST-83-06517. This is an extended abstract of [19]



## 1. Introduction

A central issue in the area of databases is that of data "relativism", that is, the general activity of structuring the same data in different ways. Considerable effort has been directed at understanding data relativism as it arises in the areas of user view construction [10], view integration [18, 21, 31, 32, 39], "derived" data [17, 24, 33], schema "simplification" [8, 9, 24], translation between data models [7, 11, 12, 22, 25, 26, 27], and relational database normalization theory [4, 6, 13, 28, 40]. A predominant theme in much of this work has been to build new schemata from existing ones using various structural manipulations [8, 18, 24, 31, 32, 39]. The new schemata are intended to have equivalent information capacity with the original schema, or to "subsume" the information capacity of the original schemata in some sense. In these investigations there is typically no formal definition of the notions of equivalent or dominant information capacity. The intuitively appealing approach usually taken is to say the one schema is dominated by another if any query directed at the first can be translated into an equivalent query of the second [7, 8, 13, 17, 24, 31, 32, 39]. (In addition, it is often assumed implicitly that data structured according to the first schema can be transformed into the second schema by some "nice" mapping, for instance, a fixed query which maps instances of the first schema into instances of the second.) We informally call this[2] "query-dominance".

[2]This notion has its roots in the notion of *Θ-equivalence* introduced by Codd [13], and a variant of query-dominance was used to formally investigate horizontal and vertical decomposition in [2].

The objective of the current paper is to introduce and use simple but rigorous theoretical tools for studying this and related measures of relative information capacity in the very simple context of schemata from the relational database model.[3]

The present investigation makes two fundamental contributions. The first consists in several theoretical results which yield conceptual insights into the area of relative information capacity. For example, one result (Theorem 5.1) indicates that the notion of query-dominance described above does not correspond to a natural, semantically meaningful type of information capacity dominance. In particular, it appears that the notion of query-dominance is too broad to accurately measure whether an underlying semantic connection exists between database schemata. A second result (Corollary 6.2) shows that with virtually any reasonable measure of relative information capacity, two relational schemata without dependencies are equivalent if and only if they are identical (up to re-ordering of the attributes and relations). This substantiates the intuition that the relational model in the absence of dependencies does not provide enough data structuring mechanisms to represent a given data set in more than one way.

The second fundamental contribution of the paper is to develop a solid mathematical foundation upon which to base an extensive theoretical investigation of relative information capacity between database schemata. This foundation finds its roots in some early work on query-dominance (usually in connection with relational database normalization [2, 3, 13]), and in the more recent, abstract work of [20]. In the present work, notions from these earlier works (along with one new one) are presented in a simple, rigorous manner and shown to correspond to intuitive and significant properties of natural measures of relative information capacity. The approach here provides mechanisms for studying relative information capacity using a variety of different mathematical techniques, including mathematical logic, combinatorics, and finite permutation group theory. Although the scope of the current paper is somewhat limited, it is clear that the definitions for measuring relative information capacity presented here can be generalized to other contexts within the relational model, and also to other database models. Thus, the approach here can serve as part of the foundation for theoretical investigations of many aspects of data relativism.

In the paper, four progressively less restrictive formal measures of relative information capacity are defined.[4] Suppose that $\mathbf{P}$ and $\mathbf{Q}$ are two relational database schemata. Speaking informally, we say that $\mathbf{Q}$ *dominates* $\mathbf{P}$ if there are functions $\sigma$ and $\tau$ such that (i) $\sigma$ maps the family of instances of $\mathbf{P}$ into the family of instances of $\mathbf{Q}$, (ii) $\tau$ maps the family of instances of $\mathbf{Q}$ into the family of instances of $\mathbf{P}$, and (iii) the composition of $\sigma$ followed by $\tau$ is the identity on the family of instances of $\mathbf{P}$. Three of the measures of information capacity are based directly on this fundamental notion, and are obtained by making certain restrictions on the maps $\sigma$ and $\tau$. The first, called[5] *calculous dominance*, is the measure which arises if $\sigma$ and $\tau$ are required to be (essentially) expressions of the relational calculus. (It is known [3] that this notion is equivalent to query-dominance, although easier to work with.) The second notion, called *generic dominance*, is less restrictive than calculous dominance and captures the notion that "natural" database transformations treat domain elements as "essentially uninterpreted objects" [1, 20]. (This is accomplished by requiring that $\sigma$ and $\tau$ commute with essentially all permutations of the

[3]The investigation here is fundamentally different than investigations such as [5, 30] and [15] into the equivalence of relational database schemes. The basic concern in [5, 30] is the equivalence of two views of an underlying universal relation, where the views are defined simply by projections. In [15] the focus is the equivalence and dominance between relational views constructed from a given underlying relational scheme using projection and join. In the current paper we do not restrict ourselves to views of of a fixed underlying schema, nor to schema manipulation via projection and join alone.

[4]The first of these, calculous dominance, has its roots in the notion of "query equivalence" as described in [13], and a variant of calculous dominance has been formally studied in [2, 3]. Another two of these measures, namely absolute and generic dominance, were originally introduced in the more general Format Model [20]. Generic dominance was also studied in [34].

[5]We choose to call this type of dominance "calculous" rather than "algebraic", because it appears that a definition based on the first order predicate calculus is easier to generalize to other data models than one based on the relationally-based algebraic operators.

underlying set of basic domain elements.) The third measure, *internal domainance*, (which is even less restrictive) captures the intuitive notion that (at a logical or conceptual level) "natural" database transformations are not based on numeric computations or string manipulations.[6] This is accomplished by requiring that $\sigma$ and $\tau$ do not "invent" or "construct" new domain elements (or data values) from the set of domain elements already occurring in an instance (aside from a finite set of data values, which corresponds to the set of constants occurring in a relational expression). The fourth measure, *absolute dominance*, is based on a family of cardinality conditions implied by internal dominance, and is relatively easy to work with.

This report is an extended abstract of a full paper by the same name [19]. Complete proofs and additional motivation for the results stated here are presented in [19]. This abstract is organized as follows. In Section 2 the slightly modified version of the relational model used for this investigation is described. (The modification allows us to easily express the fact that some attributes of a relation share the same set of possible domain values, while other attributes have fundamentally distinct sets of possible domain values.) In Section 3 the four measures of relative information capacity are formally defined and motivated. Section 4 presents some basic results concerning the four measures. Results indicating that query-dominance does not accurately measure the presence of semantic correspondence between schemata are given in Section 5, and the result concerning equivalent schemata is given in Section 6. Concluding remarks are made in Section 7.

## 2. Relation Specifiers and Schemata

The purpose of this short section is to introduce and motivate the slightly modified version of the relational model that will be used in this investigation. It is assumed here that the reader is familiar with the fundamental concepts of the relational model [29, 38]. Since the focus here is different than that of most investigations of this model, the reader is warned that we shall use some symbols here in a manner different than found elsewhere.

Speaking informally, a fundamental premise of our investigation is that the structure of relations is determined primarily by two things: the number of "columns" that a given relation has, and the sets of possible values which can appear in each of these columns. For example, if a column is intended to contain salary data, then we would expect that only positive integers are permitted as entries in that column, whereas in a column for person-names we would expect only names (or perhaps, alphabetic strings).[7] To formally capture these ideas we establish a set of "basic types" (or domain designators), along with a fixed domain of possible values associated with each basic type:

**Notation:** Let $\mathcal{B}$ be a fixed countable set of *basic types*.[8] Let the function Dom be defined on $\mathcal{B}$ such that

a. Dom(B), the *domain* of B, is a countably infinite set of abstract symbols for each $B \in \mathcal{B}$; and

b. $Dom(B) \cap Dom(C) = \emptyset$ whenever $B \neq C$.

The set **DOM** of all *domain elements* is the set $\bigcup_{B \in \mathcal{B}} Dom(B)$.

In many cases, two or more columns of a relation may have exactly the same domain (e.g., START-DATE and END-DATE). For this reason, relations are specified using (finite) sets of basic types, where each basic type may occur more than once:

**Definition:** A *non-keyed (relation) specifier* is a[9] finite multiset over $\mathcal{B}$. The *support* of a non-keyed specifier R is the set $supp(R) = \{B \in \mathcal{B} \mid R(B) > 0\}$.

---

[6] This notion highlights the fact that the current investigation is concerned with "pure" database access and transforamtion lanuguages, i.e., those which focus primarily on the data structures provided by the database model.

[7] R. Fagin studied this notion using "domain dependencies" [16]. Also, R. Reiter has studied this using the logic-based formalism of "typed" databases [35, 36].

[8] We assume that there is a fixed, unspecified total order on $\mathcal{B}$.

[9] A *multiset over* a set X is a total function $M: X \rightarrow \mathbf{N}$ (the natural numbers). A multiset M is *finite* if $\{x \in X \mid M(x) > 0\}$ is finite. If $x \in X$ then x is an *element* of M, denoted $x \in M$, if $M(x) > 0$.

While not all possible real-world relations can be modeled within the framework that is being developed here, the results obtained in this limited context are of sufficient interest to warrant investigation; furthermore, in this new area it is important to resolve simple problems before tackling the more complicated ones.

We generally denote a non-keyed specifier by listing its elements, with multiple occurrences where appropriate. For instance, if R is a specifier with support {A,B,C} and R(A)=2, R(B)=1, and R(C)=3, we denote R by AABCCC or $A^2BC^3$.

Formally, (relational) instances are associated with non-keyed specifiers as follows:

**Definition:** If R is a non-keyed specifier, an *instance* of R is a finite subset of[10] $X_{B \in supp(R)}(X_{i=1}^{R(B)}(Dom(B)))$. The family of instances of R is denoted **I**(R).

We now extend our notation to include one key dependency per relation. Key dependencies, especially in the case of one key dependency per releation, are fundamental to many semantic data models, including the functional data model [23, 37] and the entity-relationship model [11]. Key dependencies are incorporated into our notation in the following convenient manner:

**Definition:** A *keyed (relation) specifier* is an ordered pair (R,S) of multisets over $\mathcal{B}$, usually written as R:S. The *support* of R:S, supp(R:S), is[11] supp(RS). An *instance* of R:S is a total function from an instance I of R to $X_{B \in supp(S)}(X_{i=1}^{S(B)}(Dom(B)))$. (We typically view such instances as if they are members of **I**(RS), i.e., as finite sets of ordered tuples.) The family of instances of a keyed specifier R:S is denoted **I**(R:S).

Note that a non-keyed specifier R can be viewed as the keyed specifier R:$\emptyset$.

[10]We view Cartesian products such as this one as "flattened", that is, we view this product as a single product of subsets of **DOM** rather than as a product of products of subsets of **DOM**. Furthermore, we view the columns of this product to be ordered by the context of the discussion. If no order is specified by that context, then the underlying ordering on $\mathcal{B}$ is used.

[11]If L and M are multisets over X then their *union*, L ∪ M, is the multiset K such that K(x) = L(x) + M(x) for each x ∈ X. Following relational tradition, if R and S are non-keyed specifiers, we denote their union R ∪ S by RS.

Speaking informally, a relation scheme consists of one or more relations, some of which may share the same underlying (column) structure. For this reason, we formally define a relation schema to be a multiset of relation specifiers:

**Definition:**[12] A *(relational) schema* is a finite multiset[13] $\mathbf{P} = P_1,P_2,...,P_n$ of relation specifiers. The *support* of **P** is the set $supp(\mathbf{P}) = \cup_{j=1}^{n} supp(P_j)$. An *instance* of **P** is an element of[14] $\mathbf{I}(\mathbf{P}) = X_{j=1}^{n} \mathbf{I}(P_j)$. If $P_j$ is non-keyed for $1 \leq j \leq n$, then **P** is a *non-keyed* relational schema.

As an aside, we note that the collection of non-keyed relational schemata as defined here corresponds precisely, in the terminology of the Format Model [20], to the collection of formats which are constructed using a composition of one or more subformats, each of which is a collection of a composition of one or more basic types (except that here we do not associate "tokens" with the various components of our relational schemata).

Finally, we (briefly) mention the version of the relational calculus used here. (Further details are given in [19].) We assume that the reader is familiar with the calculus as described in [14, 29, 38]). In the current investigation we use, in the terminology of [38], the *domain relational calculus* in the sense that the variables and constants in our calculus range over individual domain elements. In keeping with our definition of basic types and the fact that their associated domains are disjoint, we assume that each (domain-value) variable occurring in our calculus expressions is associated with a given basic type. Finally, given relational schemata $\mathbf{P} = P_1,...,P_m$ and $\mathbf{Q} = Q_1,...,Q_n$, a (relational) *calculus expression* from **P** to **Q** is an n-tuple $\xi = (\xi_1,...,\xi_n)$ where $\xi_j$ is a

[12]We use the term 'schema' here to distinquish it from the usual notion of a relational 'scheme', where a set of attributes as opposed to a multiset of basic types is specified for each relation [29, 38].

[13]When listing the occurrences of elements in a schema we separate them by commas to avoid ambiguities. (For example, $A^2,B^2$ denotes a schema with two specifiers, while $A^2B^2$ denotes a single specifier (or a schema with one specifier in it).)

[14]Elements of **I**(**P**) are n-tuples (thus, we do not view this product as "flattened"); and as before the order of the coordinates in this product are given by the context of the discussion, or if no such order is determined, by some unspecified but fixed ordering.

(conventional) calculus expression which takes as input an instance of **P** and yields as output an instance of $Q_j$ ($1 \leq j \leq n$). In this case we write $\xi$:**P** $\rightarrow$ **Q**.

## 3. Four Measures of Relative Information Capacity

In this section the four measures of relative information capacity are introduced and motivated. As noted in the Introduction, the first of these, calculous dominance, has its roots in the notion of "query equivalence" as described in [13], and a variant of it has been formally studied in [2, 3]. As will be seen, this notion is equivalent to the notion of query-dominance described in the Introduction, but is easier to work with. Two other measures of relative information capacity, namely absolute and generic dominance, were originally introduced in the more general Format Model [20], and generic dominance was also studied in [34]. Finally, the notion of internal dominance is a new notion which is based solely on the intuition that natural database transformations do not "invent" or "construct" new domain elements from old ones. The section concludes with a result stating that the four measures of information capacity are progressively less restrictive.

To begin the formal discussion, we present a notion fundamental to our approach:

**Definition:** Let **P** and **Q** be relational schemata. A (*schema*) *transformation from* **P** *to* **Q** is a map $\sigma$:**I**(**P**) $\rightarrow$ **I**(**Q**). In this case we write $\sigma$:**P** $\rightarrow$ **Q**.

Note that a calculus expression $\xi$:**P** $\rightarrow$ **Q** can be viewed as a transformation.

In the spirit of [2, 3, 13, 20], we define relative information capacity using a pair of transformations, the composition of which forms the identity on the dominated family of instances:

**Definition:** Let **P** and **Q** be schemata, and let $\sigma$:**P** $\rightarrow$ **Q** and $\tau$:**Q** $\rightarrow$ **P**. Then **Q** *dominates* **P** *via* $(\sigma,\tau)$, denoted **P** $\preceq$ **Q** via $(\sigma,\tau)$, if $\tau\circ\sigma$ (i.e., the composition of $\sigma$ followed by $\tau$) is the identity on **I**(**P**).

Suppose that **P** $\preceq$ **Q** via $(\sigma,\tau)$. This means that information structured according to **P** can be restructured (via $\sigma$) to "fit" into **Q**, and restructured again (via $\tau$) to "fit" into **P**, in such a way that the result is the same as the original. This suggests that **Q** has at least as much capacity for storing information as does **P**.

The first of our measures is based on restricting the class of permissable database transformations to be calculus expressions. We note that the notion in [2] of one schema being *included* in a second one is the same as our notion here of calculous dominance, except that in their formal investigation the query language used includes only the operations of projection, selection, join and union, which does not have the full power of the relational calculus. (For example, set difference cannot be realized using these operators).

**Definition:** Let **P** and **Q** be schemata. Then **Q** *dominates* **P** *calculously*, denoted **P** $\preceq$ **Q** (calc), if there is a pair of calculus expressions $\xi$:**P** $\rightarrow$ **Q** and $\omega$:**Q** $\rightarrow$ **P** such that **P** $\preceq$ **Q** via $(\xi,\omega)$. **P** and **Q** are *calculously equivalent*, denoted **P** $\sim$ **Q** (calc), if **P** $\preceq$ **Q** (calc) and **Q** $\preceq$ **P** (calc).

It is easily verified that calculous dominance is transitive and reflexive, and that calculous equivalence is an equivalence relation.

To compare the notion of calculous dominance as just defined with the notion of query-dominance described in the Introduction, we present a formal definition of query-dominance for the current setting:

**Definition:** Let **P** and **Q** be schemata. Then **Q** query-dominates **P** if there is a calculus expression $\mu$:**P** $\rightarrow$ **Q** such that for each relation specifier R and each calculus expression $\alpha$:**P** $\rightarrow$ R, there is a calculus expression $\beta$:**Q** $\rightarrow$ R such that $\alpha = \beta\circ\mu$.

The next result formally states the equivalence of calculous dominance and query-dominance.

**Proposition 3.1** [3, 13]: Let **P** and **Q** be schemata. Then **Q** query-dominates **P** iff **P** $\preceq$ **Q** (calc).

There are two advantages to the definition here of calculous dominance over the definition of query-dominance. First, calculous dominance is easier to work with because it involves only two calculus expressions (as

opposed to infinitely many). Second, as will be seen shortly, the form of the definition of calculous dominance is easily generalized to provide a variety of techniques for studying it.

The second measure of relative information capacity, called "generic dominance," is somewhat more general than calculous dominance, and is useful in showing that one schema does not calculously dominate another (see Proposition 4.5 below). Generic dominance formally captures an intuitively natural restriction on database transformations, namely that any transformation used should "... treat data values as essentially uninterpreted objects ..." [1, 20]. (However, we do allow our transformations to use a bounded number of domain elements as "constants", which correspond intuitively to the constants occurring in calculus expressions.) As in [20], to define generic dominance we first formalize this property of genericity:

**Definition:** Let $Z \subseteq \mathbf{DOM}$. A *Z-permutation (of* **DOM**) is a function $\pi$:**DOM** $\rightarrow$ **DOM** such that

a. $\pi(z) = z$ for each $z \in Z$, and

b. the restriction of $\pi$ to Dom(B) is a 1:1 onto function from Dom(B) to Dom(B) for each $B \in \mathcal{B}$.

A transformation $\sigma$:$\mathbf{P} \rightarrow \mathbf{Q}$ is *Z-generic* if for each Z-permutation $\pi$ and each[15] instance I of **P**, $\pi \circ \sigma(I) = \sigma \circ \pi(I)$ (i.e., $\sigma$ and $\pi$ commute on $\mathbf{I}(\mathbf{P})$).

Speaking informally, a Z-permutation $\pi$ leaves Z fixed, and the restriction of $\pi$ to Dom(B) is a permutation of Dom(B) for each basic type B. And, again speaking informally, a transformation is Z-generic if for each $B \in \mathcal{B}$ it treats all elements of Dom(B) – Z as "equals".

It is easily verified that

**Lemma 3.2:** Let **P** and **Q** be schemata, $\xi$:$\mathbf{P} \rightarrow \mathbf{Q}$ be a calculus expression, and let Z be the set of constants occurring in $\xi$. Then $\xi$ is a Z-generic transformation.

Following [20], generic dominance is now defined by requiring that the transformations $\sigma$ and $\tau$ which restructure data be generic.

**Definition:**[16] Let **P** and **Q** be schemata. Then **Q** *dominates* **P** *generically*, denoted $\mathbf{P} \preceq \mathbf{Q}$ (gen), if there is a finite $Z \subseteq \mathbf{DOM}$ and Z-generic transformations $\sigma$:$\mathbf{P} \rightarrow \mathbf{Q}$ and $\tau$:$\mathbf{Q} \rightarrow \mathbf{P}$ such that $\mathbf{P} \preceq \mathbf{Q}$ via $(\sigma,\tau)$. **P** and **Q** are *generically equivalent*, denoted $\mathbf{P} \sim \mathbf{Q}$ (gen), if $\mathbf{P} \preceq \mathbf{Q}$ (gen) and $\mathbf{Q} \preceq \mathbf{P}$ (gen).

As with calculous dominance and equivalence, it is easily verified that generic dominance is reflexive and transitive, and that generic equivalence is an equivalence relation.

Generic dominance is of particular importance because it is independent of any data-access language, but captures a property of all such languages discussed in the literature.[17] (For example, each query in the language consisting of the relational algebra plus the least-fixed point operator is generic although this language is strictly stronger than the relational algebra or calculus [1].) Thus, results stating that one schema is not generically dominated by another can be used to support an intuitive claim that the first schema is not query-dominated by the other, where any natural query language is being used.

Our third measure of relative information capacity is more general than generic dominance, and focuses on the natural property that database transformations (at least, those used for restructuring data sets) are not typically based on numerical computations or string manipulations, and thus do not typically "invent" data values. (For example, a mapping which encodes the pair (i,j) of integers into the single integer $2^i3^j$ is based on a computation, and in essence "invents" the value $2^i3^j$.)

[15] Permutations on **DOM** are extended to families of instances in the natural manner.

[16] While technically different than the original definition of generic dominance given in [20], it can be verified that the notion used here and the original notion are equivalent in the current context.

[17] As noted earlier, this investigation is concerned only with "pure" database query or transformation languages which do not encompass numeric computation or string manipulation.

To formally capture this property of not inventing data elements, we first need:

**Definition:** Let I be an instance of a relation schema. Then the set of *symbols* of I, denoted Sym(I), is the set of elements of **DOM** which occur in I.

In the following we allow each given transformation to "invent" a (finite) set of data elements, these corresponding intuitively to the set of constants that might occur in a calculus expression.[18]

**Definition:** Let $Z \subseteq \mathbf{DOM}$. A transformation $\sigma:\mathbf{P} \rightarrow \mathbf{Q}$ is *Z-internal* if $Sym(\sigma(I)) \subseteq Sym(I) \cup Z$ for each $I \in \mathbf{I}(\mathbf{P})$.

Note that if $\sigma$ is Z-internal for some finite Z and $Sym(I) \supseteq Z$, then $Sym(\sigma(I)) \subseteq Sym(I)$. As implied by the following result, each Z-generic transformation is Z-internal. (And by Lemma 3.2, each calculus expression is Z-internal for some finite Z.)

**Lemma 3.3:** Let **P** and **Q** be schemata, $Z \subseteq \mathbf{DOM}$, and $\sigma:\mathbf{P} \rightarrow \mathbf{Q}$ be Z-generic. Then $\sigma$ is Z-internal.

We now have:

**Definition:** Let **P** and **Q** be schemata. Then **Q** *dominates* **P** *internally*, denoted $\mathbf{P} \preceq \mathbf{Q}$ (int), if there is a finite $Z \subseteq \mathbf{DOM}$ and Z-internal transformations $\sigma:\mathbf{P} \rightarrow \mathbf{Q}$ and $\tau:\mathbf{Q} \rightarrow \mathbf{P}$ such that $\mathbf{P} \preceq \mathbf{Q}$ via $(\sigma,\tau)$. **P** and **Q** are *internally equivalent*, denoted $\mathbf{P} \sim \mathbf{Q}$ (int), if $\mathbf{P} \preceq \mathbf{Q}$ (int) and $\mathbf{Q} \preceq \mathbf{P}$ (int).

As before, internal dominance is reflexive and transitive, and internal equivalence is an equivalence relation.

Our final measure of relative information capacity does not have the form of the other three, and is more general than all of them. The primary advantage of this final measure is that it is easily characterized in terms of the cardinalities of certain families of instances (see Theorem 4.2), and is therefore relatively easy to work with.

[18] As noted in Section 7, it would also be interesting to study ∅-internal transformations, i.e., transformations which do not "invent" any domain elements at all.

To define this type of dominance we need:

**Notation:** Let **P** be a relation schema and $Y \subseteq \mathbf{DOM}$. Then $\mathbf{I}_Y(\mathbf{P}) = \{ I \in \mathbf{I}(\mathbf{P}) \mid Sym(I) \subseteq Y \}$.

Suppose now that $\sigma:\mathbf{P} \rightarrow \mathbf{Q}$ is Z-internal and $Y \supseteq Z$. Then for each $I \in \mathbf{I}_Y(\mathbf{P})$, $\sigma(I) \in \mathbf{I}_Y(\mathbf{Q})$. In other words,[19] $\sigma[\mathbf{I}_Y(\mathbf{P})] \subseteq \mathbf{I}_Y(\mathbf{Q})$. Finally, if $\mathbf{P} \preceq \mathbf{Q}$ via $(\sigma,\tau)$ where $\sigma$ and $\tau$ are Z-internal, then $\sigma$ is 1-1 and so[20] $|\mathbf{I}_Y(\mathbf{P})| \leq |\mathbf{I}_Y(\mathbf{Q})|$ for each $Y \supseteq Z$. This motivates:

**Definition:**[21] Let **P** and **Q** be schemata. Then **Q** *dominates* **P** *absolutely*, denoted $\mathbf{P} \preceq \mathbf{Q}$ (abs), if there is a finite $Z \subseteq \mathbf{DOM}$ such that $|\mathbf{I}_Y(\mathbf{P})| \leq |\mathbf{I}_Y(\mathbf{Q})|$ for each (finite) $Y \supseteq Z$. **P** and **Q** are *absolutely equivalent*, denoted $\mathbf{P} \sim \mathbf{Q}$ (abs), if $\mathbf{P} \preceq \mathbf{Q}$ (abs) and $\mathbf{Q} \preceq \mathbf{P}$ (abs).

We conclude the section by stating that each of the formal measures of relative information capacity introduced above are indeed progressively less restrictive, in the sense that if **P** is dominated by **Q** according to one the the measures, then **P** is dominated by **Q** according to each of the subsequent measures:

**Theorem 3.4:** Let **P** and **Q** be schemata. Then $\mathbf{P} \preceq \mathbf{Q}$ (calc) implies $\mathbf{P} \preceq \mathbf{Q}$ (gen); $\mathbf{P} \preceq \mathbf{Q}$ (gen) implies $\mathbf{P} \preceq \mathbf{Q}$ (int); and $\mathbf{P} \preceq \mathbf{Q}$ (int) implies $\mathbf{P} \preceq \mathbf{Q}$ (abs).

As we shall see, the converse of each of these implications is also true for non-keyed relational schemata **P** and **Q** where **Q** consists of only one relation specifier (Theorem 5.1), and for non-keyed relational schemata involving only one basic type (Theorem 5.3). However, if **Q** contains more that one specifier, or if key dependencies are incorporated, then at least one of these converse implications fails (Proposition 4.5.)

[19] If $f:M \rightarrow N$ and $K \subseteq M$, then $f[K]$ denotes $\{ f(k) \mid k \in K \}$.

[20] If X is a set then |X| denotes the cardinality of X.

[21] While technically different than the original definition of absolute dominance given in [20], it is easily verified that the notion used here and the original notion are equivalent in the current context.

## 4. Some Basic Results

In this section we present several basic results concerning the notions of information capacity dominance defined above. The first result gives a characterization of absolute dominance in terms of certain functions. A simple application of this result is given to indicate how it can be used to show that calculous dominance does not hold, and this result is also used as the basis for Theorem 6.1. The second major result of the section (Theorem 4.4) gives a characterization of internal dominance. The third major result (Proposition 4.5) shows that absolute and internal dominance are different from generic and calculous dominance, and illustrates a technique for showing that generic dominance does not hold. The section concludes with a number of results giving sufficient conditions for dominance (of one sort or another) to hold. Most important of these is Theorem 4.7, which concerns schemata constructed from other schemata through "renamings" of the basic types used.

The functions needed for the characterization of absolute dominance are now presented.

**Definition:** Let R be a non-keyed relation specifier and let $B_1,...,B_n$ be an enumeration of the basic types in supp(R) (and possibly some other basic types). Then the *cardinality function* of R (relative to this enumeration) is the polynomial $f_R: \mathbf{N}^n \rightarrow \mathbf{N}$ where

$$f_R(\vec{x}) = \Pi_{1 \leq j \leq n}(x_j^{R(B_j)}).$$

Now suppose that R is the keyed specifier S:T and supp(R) $\subseteq \{B_1,...,B_n\}$. Then the *cardinality function* of R is

$$f_R(\vec{x}) = [\log_2((\Pi_{1 \leq j \leq n} x_j^{T(B_j)}) + 1)] \cdot [\Pi_{1 \leq j \leq n}(x_j^{S(B_j)})].$$

Finally, let $\mathbf{P} = P_1,...,P_m$ be a relational schema with supp(**P**) $\subseteq \{B_1,...,B_n\}$. Then the *cardinality function* of **P** is

$$f_{\mathbf{P}}(\vec{x}) = \Sigma_{1 \leq i \leq m}(f_{P_i}(\vec{x})).$$

The significance of the cardinality functions is given by:

**Lemma 4.1:** Let **P** be a relational schema with support contained in $B_1,...,B_n$, and let $Y \subseteq$ **DOM** be a finite set with $|Y \cap \mathrm{Dom}(B_j)| = x_j$, $1 \leq j \leq n$. Then

$$|\mathbf{I}_Y(\mathbf{P})| = 2^{f_{\mathbf{P}}(\vec{x})}.$$

The following characterization of absolute dominance is now immediate:

**Theorem 4.2:** Let **P** and **Q** be relational schemata. Then $\mathbf{P} \preceq \mathbf{Q}$ (abs) iff there is some $t \geq 0$ such that $f_{\mathbf{P}}(\vec{x}) \leq f_{\mathbf{Q}}(\vec{x})$ for each $\vec{x}$ with $x_j \geq t$ $(1 \leq j \leq n)$.

The above result provides an easy mechanism for showing that calculous dominance does not hold in many cases. For example, the following corollary yields the intuitive conclusion that a relation with two NAME columns and one NUMBER column is not query-dominated by a relation with one NAME column and two NUMBER columns.

**Corollary 4.3:** AAB $\not\preceq$ ABB (abs), and hence AAB $\not\preceq$ ABB (calc).

It is clear that the technique of the above proof can be applied in many situations to infer that one schema is not calculously dominated by another one.

We now turn to the second major result of the section, namely a characterization of internal dominance. In particular, this result shows that internal dominance is equivalent to a cardinality condition which is similar in spirit to the definition of absolute dominance. To state the result we need:

**Notation:** Let **P** be a schema, and let Y and Z be finite subsets of **DOM**. Then **P**(Y,Z) denotes $\{ I \in \mathbf{I}(\mathbf{P}) \mid Y - Z \subseteq \mathrm{Sym}(I) \subseteq Y \}$.

Thus, instances in **P**(Y,Z) involve all the symbols of Y - Z, and no symbols outside of Y. Note that for each **P** and Y, $\mathbf{P}(Y,Y) = \mathbf{I}_Y(\mathbf{P})$.

We now have:

**Theorem 4.4:** Let **P** and **Q** be schemata. Then $\mathbf{P} \preceq \mathbf{Q}$ (int) iff there is some finite $Z \subseteq$ **DOM** such that $|\mathbf{P}(Y,Z)| \leq |\mathbf{Q}(Y,Z)|$ for each finite $Y \supseteq Z$.

We now turn to the third major result of the section, which shows that there are examples where

absolute and internal dominance hold, but generic and hence calculous dominance do not. The general proof technique used to show that generic dominance does not hold is of interest, because it provides one of the few known methods for demonstrating that one schema is not calculously dominated by another one, even though absolute dominance holds.

**Proposition 4.5:**

a. AB $\preceq$ AA,BB (abs,int) but[22] AB $\not\preceq$ $[AA]^n$,$[BB]^n$ (gen,calc) for each $n > 0$; and

b. A:BB $\preceq$ AB (abs, int) but A:BB $\not\preceq$ AB (gen, calc).

It remains open whether absolute and internal dominance can be distinquished in the present context, or whether generic and calculous dominance can be distinquished in the present context. With regard to the latter question, it is interesting to recall the result of [1] which states that the generic query operation of transitive closure (of a binary relation) is not realizable by any calculus expression. In other words, the notions of generic and calculous can be distinquished in the context of query operations.

Returning to general results, we now present several sufficient conditions for inferring dominance of one sort or another. The first result presents cases where the existence of one transformation (from **P** to **Q**) rather than two (both from **P** to **Q** and back) are needed. The techniques used to prove this theorem are quite general, and so it appears that these results also hold in more general database models. In particular, both parts of this theorem hold in the Format Model [20] (assuming that the definition of family of instances used there is modified in analogy to the definition used here). It remains open whether the theorem also holds for calculous dominance.

**Theorem 4.6:** Let **P** and **Q** be schemata. Then

a. $\mathbf{P} \preceq \mathbf{Q}$ (gen) iff there is a finite set $Z \subseteq$ **DOM** and a 1-1 Z-generic transformation $\sigma$:$\mathbf{P} \rightarrow \mathbf{Q}$.

b. $\mathbf{P} \preceq \mathbf{Q}$ (int) iff there is a finite set $Z \subseteq$ **DOM** and a 1-1 Z-internal transformation $\sigma$:$\mathbf{P} \rightarrow \mathbf{Q}$.

The next result examines the impact of changing the basic types occurring in schemata. Speaking informally, the result states that dominance is preserved by re-namings of basic types (even if different basic types are identified by the re-naming.) For this result we use:

**Definition:** A *homomorphism (on $\mathcal{B}$)* is a function h:$\mathcal{B} \rightarrow \mathcal{B}$. If h is a homomorphism and R is a non-keyed specifier, then h(R) denotes the non-keyed specifier U where $U(A) = \Sigma_{h(B)=A} R(B)$ for each basic type A. If R and S are non-keyed specifiers then h(R:S) = h(R):h(S), and if $\mathbf{P} = P_1,...,P_n$ is a schema then $\mathbf{h(P)} = h(P_1),...,h(P_n)$.

**Theorem 4.7:** Let **P** and **Q** be schemata, and h a homomorphism on $\mathcal{B}$. If $\mathbf{P} \preceq \mathbf{Q}$ (xxx) then $\mathbf{h(P)} \preceq$ h(**Q**) (xxx), where 'xxx' ranges over 'calc', 'gen', 'int', and 'abs'.

The converse of this result does not hold for any of the types of dominance, since ABB $\not\preceq$ AAB (xxx) (by Corollary 4.3) but AAA $\preceq$ AAA (xxx), where 'xxx' ranges over each of the four types of dominance.

We conclude the section with three relatively simple results for inferring dominance between two schemata, given dominance by two other schemata. Each demonstrates the preservation of dominance under a kind of "additivity".

**Proposition 4.8:** Let **P**, **Q**, and **R** be schemata. Then

a. $\mathbf{P} \preceq \mathbf{Q}$ (xxx) $\Rightarrow$ $\mathbf{RP} \preceq \mathbf{RQ}$ (xxx), where 'xxx' ranges over 'calc', 'gen', and 'int'; and

b. $\mathbf{P} \preceq \mathbf{Q}$ (abs) $\Leftrightarrow$ $\mathbf{RP} \preceq \mathbf{RQ}$ (abs).

**Corollary 4.9:** Let **P**, **Q**, **R**, and **S** be schemata, with $\mathbf{P} \preceq \mathbf{Q}$ (xxx) and $\mathbf{R} \preceq \mathbf{S}$ (xxx). Then $\mathbf{PR} \preceq \mathbf{QS}$ (xxx), where 'xxx' ranges over 'calc', 'gen', 'int', and 'abs'.

**Proposition 4.10:** Let R, S, U, V and T be non-keyed specifiers. Then

[22] If R is a specifier and $k \geq 0$, then $[R]^k$ denotes the schema with k occurrences of R.

a. R:S $\preceq$ U:V (xxx) $\Rightarrow$ TR:S $\preceq$ TU:V (xxx), where 'xxx' ranges over 'calc', 'gen', and 'int'; and

b. R:S $\preceq$ U:V (abs) $\Leftrightarrow$ TR:S $\preceq$ TU:V (abs).

It remains open whether the converse of part (a) of either of Propositions 4.8 or 4.10 holds for any of calculous, generic or internal dominance.

## 5. Calculous Dominance vs. Semantic Correspondence

In this section we present results which indicate that the notion of calculous dominance (and hence, query-dominance) does not accurately reflect or measure the presence of "semantic correspondence" between schemata. Our first result, Theorem 5.1, characterizes calculous dominance between non-keyed relational schemata, where the dominating schema consists of a single specifier. This result implies that calculous dominance holds in a variety of counter-intuitive situations (see Example 5.2). The section concludes with a result which implies that each of the types of dominance is the same in the context of non-keyed relational schemata which involve only one basic type.

For the first result, we use the natural partial ordering of non-keyed specifiers, considered simply as multisets:

**Notation:** For non-keyed specifiers R and S, write $R \subseteq S$ if $R(B) \leq S(B)$ for each $B \in \mathcal{B}$. Write $R \subset S$ if $R \subseteq S$ and $R(B) < S(B)$ for some $B \in \mathcal{B}$.

We can now state:

**Theorem 5.1:** Let $\mathbf{P} = R_1,...,R_n$ be a non-keyed schema and S a non-keyed specifier. Then the following are equivalent:

a. $\mathbf{P} \preceq S$ (calc);

b. $\mathbf{P} \preceq S$ (gen);

c. $\mathbf{P} \preceq S$ (int);

d. $\mathbf{P} \preceq S$ (abs); and

e. either $n = 1$ and $R_1 \subseteq S$, or $n > 1$ and $R_i \subset S$ for each j, $1 \leq j \leq n$.

The following example illustrates the significance of this result.

**Example 5.2:** Assume that a (large) set of NAME-values and a (large) set of NUMBER-values is fixed (where there is no ordering or other predicate on either of these sets). Suppose further that a relation scheme **R** consists of 50 relations $R_i$, each of which has one column with NAME-values and two columns with NUMBER-values; and also 50 relations $S_j$, each of which has two columns with NAME-values and one column with NUMBER-values. Also, let **T** be a relation scheme with a single relation in it, where that relation has two columns for NAME-values and two columns for NUMBER-values. By Theorem 5.1, $\mathbf{R} \preceq \mathbf{T}$ (calc) and so **R** is query-dominated by **T**. Since it appears that there is no intuitively appealing, semantically meaningful 1-1 mapping of instances of **R** to instances of **T**, this indicates that query-dominance does not accurately measure whether there is a "semantic" connection between pairs of schemata. □

We briefly consider a natural analog of Theorem 5.1 for keyed specifiers. Specifically, suppose that the statement of the theorem were changed to allow keyed specifiers, and that condition (e) were changed to read "either $n = 1$ and $P_1 \preceq R$ (xxx), or $n > 1$ and both $P_i \preceq R$ (xxx) and $P_i \neq R$ for each i, $1 \leq i \leq n$" (where 'xxx' ranges over one (or more) of the four types of dominance). A counterexample to this modified version of the theorem is easily obtained. For example, it is easily verified that A:B $\preceq$ A:BB (calc) (and hence for each of the types of dominance) but $[A:B]^3 \not\preceq$ A:BB (abs) (and hence for none of them).

The final result of this section concerns dominance between non-keyed schemata which involve only one basic type. (Note that many of the early investigations of the relational calculus were essentially based on relational schemata of this type.)

**Theorem 5.3:** Let **P** and **Q** be non-keyed relational schemata over a single basic type B. Then the following are equivalent:

a. $\mathbf{P} \preceq \mathbf{Q}$ (calc);

b. $\mathbf{P} \preceq \mathbf{Q}$ (gen);

c. $\mathbf{P} \preceq \mathbf{Q}$ (int);

d. $\mathbf{P} \preceq \mathbf{Q}$ (abs); and

e. the[23] leading coefficient (as a polynomial) of $f_{\mathbf{Q}}(x) - f_{\mathbf{P}}(x)$ is a non-negative integer.

It remains open whether the above result can be generalized to apply to pairs of relational schemata involving keyed specifiers. More specifically, while it is clear that $A{:}A \preceq A^2$ (calc) and it can be shown that $A{:}A^2 \preceq A^2$ (calc), it is unknown whether $A{:}A^n \preceq A^2$ (calc) for each $n \geq 0$.

## 6. Equivalence implies Equality

This section presents the result that if $\mathbf{P}$ and $\mathbf{Q}$ are non-keyed schemata and they are equivalent under any of the notions of dominance, then $\mathbf{P} = \mathbf{Q}$ (up to reordering). A corollary of this result (presented at the end of the section) implies that for any "natural" notion of information capacity equivalence, a pair of non-keyed relational schemata are equivalent if and only if they are equal. This supports the intuition that in the relational model (with no dependencies) there is essentially at most one way to represent a given data set. It is interesting to note that a result of [20] concerning the Format Model provides contrast with the results mentioned here. In particular, it is shown there that two formats may be generically equivalent and yet be unequal.[24]

We now have:

**Theorem 6.1:** Let $\mathbf{P}$ and $\mathbf{Q}$ be non-keyed relational schemata. Then the following are equivalent:

a. $\mathbf{P} \sim \mathbf{Q}$ (calc);

b. $\mathbf{P} \sim \mathbf{Q}$ (gen);

c. $\mathbf{P} \sim \mathbf{Q}$ (int);

d. $\mathbf{P} \sim \mathbf{Q}$ (abs); and

e. $\mathbf{P} = \mathbf{Q}$.

[23] The *leading coefficient* of a polynomial of one variable is the coefficient of the term having the highest exponent.

[24] Furthermore, in [20], it is shown that two formats are generically equivalent if and only if one can be formed from the other using a set of six natural, local, structural transformations (called "reductions") and their inverses.

While it appears that the above theorem can be generalized to keyed relational schemata, no proof is currently known.

The following corollary of Theorem 6.1 can be interpreted as a result concerning any "natural" notion of relative information capacity dominance in the following sense: The corollary considers all measures $\ll$ which lie in the "region" between equality (i.e., isomorphism) and absolute dominance. Since we would expect any "natural" measure of information capacity (which does not involve computation) to lie in this region, the corollary applies to all such "natural" measures.

**Corollary 6.2:** Let $\ll$ be any binary relation on non-keyed relational schemata such that for each pair $\mathbf{P}$, $\mathbf{Q}$ of schemata,

a. $\mathbf{P} \ll \mathbf{Q} \Rightarrow \mathbf{P} \preceq \mathbf{Q}$ (abs); and

b. $\mathbf{P} = \mathbf{Q} \Rightarrow \mathbf{P} \ll \mathbf{Q}$.

Also, let $\asymp$ be defined so that $\mathbf{P} \asymp \mathbf{Q}$ iff $\mathbf{P} \ll \mathbf{Q}$ and $\mathbf{Q} \ll \mathbf{P}$. Then for each pair $\mathbf{P}$, $\mathbf{Q}$ of schemata, $\mathbf{P} \asymp \mathbf{Q}$ iff $\mathbf{P} = \mathbf{Q}$.

## 7. Concluding remarks

The model of relative information capacity introduced here can serve as part of the foundation for the theoretical study of data relativism as it arises in a variety of database areas. While several interesting results have been reported above, the investigation also raises a number of important questions. In this section some of these are briefly mentioned.

A major area demanding further investigation is to seek measures of relative information capacity other than the ones presented here. For example, the results of Section 5 indicate that none of the types of dominance studied here correspond closely to the intuitive notion of "semantic correspondence". A more "natural" measure might be obtained by modifying calculous dominance (and the other types as well) so that constants are not allowed. Alternatively, other "abstract" properties of natural database transformations (such as being generic or internal) might be formalized and investigated. For example, a transformation $\sigma{:}\mathbf{P} \rightarrow \mathbf{Q}$ can be called *additive* if for

each I and I', we have[25] $\sigma(I \cup I') = \sigma(I) \cup \sigma(I')$.

Another direction is to consider a broader notion of information capacity which incorporates update capabilities. For example, suppose that some update language UL is fixed (e.g., one consisting of all compositions of the primitive operations of delete one tuple, insert one tuple, and update one tuple). Define $\mathbf{P} \preceq \mathbf{Q}$ (UL) if (i) $\mathbf{P} \preceq \mathbf{Q}$ (calc) via calculus expressions $\xi$ and $\omega$, *and* if (ii) whenever $I \in \mathbf{I}(\mathbf{P})$ and $\mu$ is an update in UL of I then there is an update $\mu'$ in UL of $\xi(I)$ such that $\mu'(\xi(I)) = \xi(\mu(I))$. This approach may give insight into the "view-update" problem.

A third general area is to consider the various complexity issues raised by this investigation. For example, what is the complexity of deciding whether $\mathbf{P} \preceq \mathbf{Q}$ (xxx)? If it is known, say, that $\mathbf{P} \preceq \mathbf{Q}$ (calc), then how hard is it to find a pair $\xi,\omega$ of calculus expressions such that $\mathbf{P} \preceq \mathbf{Q}$ via $(\xi,\omega)$? Finally, it is also important to examine the complexity of actually performing $\xi$ and $\omega$ on instances of **P** and **Q**, that is, to examine the ease of translating between these two schemata.

A variety of other issues can be studied using the model of information capacity presented here. For example, it would be interesting to examine relative information capacity between schemata taken from database models besides the relational one; and to examine relative information capacity between schemata **P** and **Q**, where **P** is taken from one model and **Q** from another (similar to [27]). Within the relational model, it would be useful to examine the impact of null values and new dependencies.

Finally, a number of specific open questions are raised in this report. Most provacative, perhaps, is the question of whether the notions of calculous and generic dominance are actually co-extensive in the context of relational schemata as defined here, or at least in the context of non-keyed relational schemata. The analogous questions for internal and absolute dominance are also of interest. As noted in Section 6, it remains open whether the result on equivalence (Theorem 6.1) can be extended to relational schemata with keys. Finally, simple characterizations of calculous, generic and internal dominance remain unknown. Indeed, there are simple questions about these types of dominance that remain unresolved. For example, if A is a basic type, what is the greatest value of $n > 0$ for which $A:A^n \preceq AA$ (gen)?

[25] By $\cup$ we mean coordinate-wise union. Speaking intuitively, we also view $I \cup I'$ as undefined if any of the coordinates violate one of the key dependencies built into the corresponding specifier.


### Acknowledgement

The author expresses his gratitude to the participants of the USC Seminar on the Theory of Databases, and to Serge Abiteboul, for discussions which lead to several improvements of this paper.


### Bibliography


**1.** Aho, A.V. and J.D. Ullman. Universality of data retrieval languages. Symp. on Principles of Programming Languages, 1979, pp. 110-120.

**2.** Atzeni, P., G. Ausiello, C. Batini and M. Moscarini. "Inclusion and equivalence between relational database schemata." *Theor. Computer Science 19* (1982), 267-285.

**3.** Ausiello, G., C. Batini and M. Moscarini. On the equivalence among database schemata. Proc. International Conference on Data Bases, Aberdeen, 1980.

**4.** Beeri, C., P.A. Bernstein, and N. Goodman. A sophisticate's introduction to database normalization theory. Proc. 4th Int. Conf. Very Large Data Bases, 1978, pp. 113-124.

**5.** Beeri, C., A.O. Mendelzon, Y. Sagiv, and J.D. Ullman. "Equivalence of relational database schemes." *SIAM J. Comput. 10*, 2 (1981), 352-370.

**6.** Bernstein, P.A. "Synthesizing third normal form relations from functional dependencies." *ACM Trans. on Database Systems 1* (1976), 277-298.

**7.** Borkin, S.A.. *Data Models: A Semantic Approach for Database Systems*. MIT Press, Cambridge, Mass., 1980.

**8.** Brown, R. and D.S. Parker. LAURA: A formal data model and her logical design methodology. VLDB, 1983, pp. 206-218.

**9.** Casanova, M.A. and V.M.P. Vidal. Towards a sound view integration methodology. Proc. ACM SIGACT-SIGMOD Symp. on Principles of Database Systems, 1983, pp. 36-47.

**10.** Chamberlin, D.D., J.N. Gray, and I.L. Traiger. Views, authorization and locking in a relational database system. Proc. AFIPS NCC 44, 1975, pp. 425-430.

**11.** Chen, P.P. "The entity-relationship model -- toward a unified view of data." *ACM Trans. on Database Systems 1*, 1 (1976), 9-36.

**12.** Chu, W. and V.T. To. A hierarchical conceptual data model for data translation in a heterogenious database system. In P.P. Chen, Ed., *Entity-Relationship Approach to Systems Analysis and Design*, North-Holland, Amsterdam, 1980.

**13.** Codd, E.F. Further normalization of the data base relational model. In R. Rustin, Ed., *Data Base Systems*, Prentice Hall, Englewood Cliffs, N.J., 1972, pp. 33-64.

**14.** Codd, E.F. Relational completeness of data base sublanguages. In R. Rustin, Ed., *Data Base Systems*, Prentice Hall, Englewood Cliffs, N.J., 1972, pp. 65-98.

**15.** Connors, T. Equivalence of Expression Views by Query Capacity. Ph.D. thesis, Univ. of Southern California, L.A., CA, in preparation

**16.** Fagin, R. "A normal form for relational databases that is based on domains and keys." *ACM Trans. on Database Systems 6*, 3 (1981), 387-415.

**17.** Hammer, M. and D. McLeod. "Database description with SDM: A semantic database model." *ACM Trans. on Database Systems 6*, 3 (1981), 357-386.

**18.** Heimbigner, D.M. A federated architecture for database systems. Tech. Rept. TR-114, Univ. of Southern Calif., Los Angeles, CA, Aug., 1982.

**19.** Hull, R. Relative Information Capacity of Simple Relational Database Schemata. Tech. Rept. TR-84-300, USC, January, 1984.

**20.** Hull, R. and C.K. Yap. The format model: A theory of database organization. Proc. ACM SIGACT-SIGMOD Symp. on Principles of Database Systems, 1982, pp. 205-211. To appear, *J. ACM*

**21.** Jacobs, B.E. Applications of database logic to the view update problem. Tech. Rept. , Univ. of Maryland, College Park, MD, 1980.

**22.** Jacobs, B.E. "On database logic." *J. ACM 29*, 2 (1982), 310-332.

**23.** Kerschberg, L. and J.E.S. Pacheco. A functional data base model. Tech. Rept. , Pontificia Universidade Catolica do Rio de Janeiro, Rio de Janeiro, Brazil, Feb., 1976.

**24.** King, R. and D. McLeod. The event database specification model. Proc. of the 2nd Intl. Conf. on Databases: Improving Usability and Responsiveness, Jerusalem, Isreal, June, 1982, pp. 299-321.

**25.** Klug, A. Entity-relationship views over uninterpreted enterprise schemas. In P.P. Chen, Ed., *Entity-Relationship Approach to Systems Analysis and Design*, North-Holland, Amsterdam, 1980, pp. 52-72.

**26.** Lien, Y.E. On the semantics of the entity-relationship data model. In P.P. Chen, Ed., *Entity-Relationship Approach to Systems Analysis and Design*, North-Holland, Amsterdam, 1980, pp. 131-146.

**27.** Lien, Y.E. "On the equivalence of database models." *J. ACM 29*, 2 (1982), 333-362.

**28.** Ling, T.-W., F.W Tompa, and T. Kameda. "An improved third normal form for relational databases." *ACM Trans. on Database Systems 6*, 2 (1981), 329-346.

**29.** Maier, D.. *The Theory of Relational Databases*. Computer Science Press, Rockville, Maryland, 1983.

**30.** Maier, D., A.O. Mendelzon, F. Sadri, and J.D. Ullman. "Adequacy of decompositions of relational databases." *J. Computer and System Sciences 21*, 3 (1980), 368-379.

**31.** Motro, A. Construction and interrogation of virtual databases. Tech. Rept. TR-83-211, USC, April, 1983.

**32.** Motro, A. and P. Buneman. Constructing superviews. Proc. ACM SIGMOD Int. Conf. on the Management of Data, 1981, pp. 56-64.

**33.** Mylopoulos, J., P.A. Bernstein, and H.K.T. Wong. "A language facility for designing database-intensive applications." *ACM Trans. on Database Systems 5*, 2 (1980), 185-207.

**34.** O'Dunlaing, C. and C.K. Yap. Generic transformation of data structures. Proc. 23rd Ann. IEEE Symp. on Foundations of Computer Science, 1982, pp. 186-195.

**35.** Reiter, R. "Equality and domain closure in first-order databases." *J. ACM 27*, 2 (1980), 235-249.

**36.** Reiter, R. On the integrity of typed first order data bases. In H. Gallair, J. Minker, and J.-M. Nicolas, Ed., *Advances in Database Theory*, Plenum Press, N.Y., 1981, pp. 137-157.

**37.** Shipman, D. "The functional data model and the data language DAPLEX." *ACM Trans. on Database Systems 6*, 1 (1981), 140-173.

**38.** Ullman, J.D.. *Principles of Database Systems, 2nd ed.* Computer Science Press, Potomac, MD, 1982.

**39.** Yao, S.B., V. Waddle, and B.C. Housel. "View modeling and integration using the functional data model." *IEEE Trans. Soft. Eng. SE-8*, 6 (1982), 544-553.

**40.** Zaniolo, C. "A new normal form for the design of relational database schemata." *ACM Trans. on Database Systems 7*, 3 (1982), 489-499.