Using Domain-Derived Constraints to Bound the Cardinality of Aggregate Views *

Paolo Ciaccia, Matteo Golfarelli, and Stefano Rizzi

University of Bologna, DEIS, Viale Risorgimento 2, Bologna, Italy
{pciaccia, mgolfarelli, srizzi}@deis.unibo.it

(Extended Abstract)

Abstract. Accurately estimating the cardinality of aggregate views is crucial for logical and physical design of data warehouses. While the warehouse is under development and data are not available yet, the approaches based on accessing data cannot be adopted. This paper reports on the progress of an ongoing research aimed at devising a comprehensive approach to estimate the cardinality of views based on a-priori information derived from the application domain. The results we present concern the computation of upper and lower bounds for the cardinality of a view considering the functional dependencies between attributes of the multidimensional scheme and a set of cardinality constraints expressed on some other views. We propose a bounding strategy which achieves an effective trade-off between the tightness of the bounds produced and the computational complexity, and outline a branch-and-bound approach to compute it. Finally, we discuss some open issues and sketch our future research.

1 Introduction and Motivation

The multidimensional model is the foundation for data representation and querying in multidimensional databases and data warehouses [AGS97]. It represents facts of interest for the decision process into *cubes* in which each cell contains numerical *measures* which quantify the fact from different points of view, while each axis represents an interesting *dimension* for analysis. For instance, within a 4-dimensional cube modeling the phone calls supported by a telecommunication company, the dimensions might be the calling number, the number called, the date, and the time segment in which the call is placed; each cube cell could be associated to a measure of the total duration of the calls made from a given number to another on a given time segment and date.

The basic mechanism to extract significant information from the huge quantity of fine-grained data stored in base cubes is aggregation according to hierarchies of attributes rooted in dimensions [GL97]. In most application cases, cubes are significantly sparse (for instance, most couples of telephone numbers are never connected by a call in a given date), and so are the aggregate views.

Accurately estimating the actual cardinality of views is crucial for logical and physical design as well as for query processing and optimization. In particular, after the data warehouse has been loaded, this estimation activity can be carried out by using well-known techniques based for instance on histograms [MD88] and sampling [HO91]. However, such techniques cannot be applied at all if the data warehouse is still under development, and the estimation of view cardinalities is needed for design purposes. For instance, consider the view materialization problem, where the

* This work was partially supported by the D2I MURST project.

aggregate views which are the most useful in answering the workload queries have to be selected for materialization (see [TB00] for a survey). Since the number of possible views which can be derived by aggregating a cube is exponential in the number of attributes, most approaches assume that a constraint on the total disk space occupied by materialization is posed, and attempt to find the subset of views which contemporarily satisfies this constraint and minimizes the workload cost [GR00,Gup97,HRU96]. Another case where estimation of view cardinalities is relevant is index selection [GHRU97].

In the data warehouse literature, the problem of cardinality estimation when data are not available is commonly addressed by assuming that data are uniformly distributed and by relying only on the cardinalities of the base cube and of the single attributes [RS97,SDNR96]. Unfortunately, when data are skewed, the resulting cardinalities turn out to be significantly overestimated.

This paper reports on the progress of an ongoing research aimed at devising a comprehensive approach to estimate the cardinality of views based on a-priori information derived from the application domain. The results we present here concern the computation of upper and lower bounds for the cardinality of a view considering the functional dependencies between attributes of the multidimensional scheme and a set of cardinality constraints expressed on some other views. In particular, we propose a bounding strategy which achieves an effective trade-off between the tightness of the bounds produced and the computational complexity, and outline a branch-and-bound approach to compute it.

2 Background and Working Example

Definition 1 (Dimensional Scheme). *We call* dimensional scheme $\mathcal{D}$ *a couple* $(U, \mathcal{F})$ *where* U *is a set of attributes and* $\mathcal{F} = \{A_i \rightarrow A_j | A_i, A_j \in U\}$ *is a set of functional dependencies which relate the attributes of* U *into a set of pairwise disjoint directed trees. We call* dimensions *the attributes* $A_k \in U$ *in which the trees are rooted, i.e., such that* $\forall A_i \in U$ $(A_i \rightarrow A_k) \notin \mathcal{F}$*; let* $dim(\mathcal{D}) \subseteq U$ *denote the set of dimensions of* $\mathcal{D}$*.*

Definition 2 (View). *Let* $\mathcal{D} = (U, \mathcal{F})$ *be a dimensional scheme. We call* view *on* $\mathcal{D}$ *any subset of attributes* $V \subseteq U$ *such that* $\forall A_i, A_j \in V$ $(A_i \rightarrow A_j) \notin \mathcal{F}^+$*, where* $\mathcal{F}^+$ *denotes the set of all functional dependencies logically implied by* $\mathcal{F}$*.*

It should be noted that we are using the term *view* to denote the set of grouping attributes used for aggregation, while the "actual" views will typically include also one or more measures. This slight abuse in terminology is possible since we are only interested in determining the *cardinality* of views, which only depends on the grouping attributes.

Definition 3 (Roll-up). *Given the set* $\mathcal{V}_\mathcal{D}$ *of all possible views on* $\mathcal{D}$*, we define on* $\mathcal{V}_\mathcal{D}$ *the* roll-up *partial order* $\preceq$ *as follows:* $V \preceq W$ *iff* $\forall A_i \in V$ $\exists A_j \in W | (A_j \rightarrow A_i) \in \mathcal{F}^+$*, i.e., iff* $W \rightarrow V$*. We call* multidimensional lattice *for* $\mathcal{D}$ *the corresponding lattice, whose top and bottom elements are* $dim(\mathcal{D})$ *and the empty view* $\{\}$*, respectively. We will denote with* $V \oplus W$ *the least upper bound (lub) of* V *and* W*; given a set of views* S*, we will briefly denote their lub with* $\oplus(S)$*.*

Example 1. A dimensional scheme *Calls* modeling the phone calls supported by a telecommunication company might include:

$$U = \{\mathsf{date}, \mathsf{week}, \mathsf{month}, \mathsf{year}, \mathsf{sourceNumber}, \mathsf{sourceDistrict}, \mathsf{sourceState}, \mathsf{destNumber}, \mathsf{destDistrict}, \mathsf{destState}, \mathsf{timeSegment}\}$$

$$\mathcal{F} = \{\mathsf{date} \rightarrow \mathsf{week}, \mathsf{date} \rightarrow \mathsf{month}, \mathsf{month} \rightarrow \mathsf{year}, \mathsf{sourceNumber} \rightarrow \mathsf{sourceDistrict}, \mathsf{sourceDistrict} \rightarrow \mathsf{sourceState}, \mathsf{destNumber} \rightarrow \mathsf{destDistrict}, \mathsf{destDistrict} \rightarrow \mathsf{destState}\}$$

thus having $dim(\mathcal{D}) = \{\mathsf{date}, \mathsf{sourceNumber}, \mathsf{destNumber}, \mathsf{timeSegment}\}$ as dimensions. Examples of views on the *Calls* scheme are $V = \{\mathsf{month}, \mathsf{sourceNumber}, \mathsf{destState}\}$, $W = \{\mathsf{month}, \mathsf{sourceState}\}$ and $Z = \{\mathsf{year}, \mathsf{sourceDistrict}\}$. It is $W \oplus Z = \{\mathsf{month}, \mathsf{sourceDistrict}\}$. The roll-up relationships between these views are the following: $W \preceq W \oplus Z$, $Z \preceq W \oplus Z$, and $W \oplus Z \preceq V \preceq dim(\mathcal{D})$. □

3 Approach Overview

The framework for this work is the logical design of multidimensional databases carried out off-line, i.e., assuming that the source data cannot be directly queried to estimate the cardinality of multidimensional views. For simplicity, in the following we consider that estimates are needed for the purpose of view materialization, thus reliable information on the size of the candidate views has to be supplied to the materialization algorithm.

As sketched in Figure 1, whenever the materialization algorithm requires information about a candidate view V, our approach works in two steps. First, the *bounder* uses the set $\mathcal{C}$ of *cardinality constraints* supplied by the user to determine effective bounds for the cardinalities of a proper set of views; then, the *estimator* uses these bounds to derive a probabilistic estimate for the cardinality of V. Note that this two-steps approach generalizes well-known *parametric* models for the estimation of the cardinality of relational queries [MCS88], and in particular those for projection size estimation [CM95], for which bounds are typically given as input parameters. Relevant examples of cardinality constraints that may be considered are the cardinality w of a view W, a lower and/or an upper bound (w^- and w^+, respectively) to the cardinality of W, and the ratio between the cardinalities of two views W_1 and W_2.

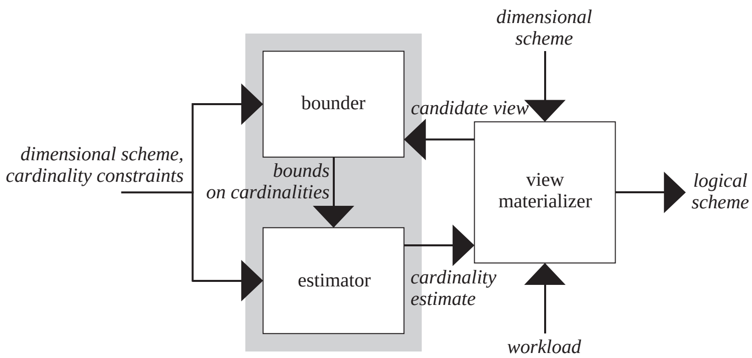

Fig. 1. Overall architecture for logical design

The set $\mathcal{C}$, together with the dimensional scheme $\mathcal{D}$, univocally determines two bounds for the cardinality of V, which are called the *greatest lower bound* and the *least upper bound*, denoted as v^- and v^+, respectively.[1] The interpretation of such bounds is as follows: (1) in each instance I of $\mathcal{D}$ that does not violate any constraint in $\mathcal{C}$, the cardinality v of V is such that $v \in [v^-, v^+]$; and (2) there exists an instance I^- (I^+) compatible with $\mathcal{C}$ where v equals v^- (v^+).

Computing the bounds implied by $\mathcal{C}$ turns out to be a challenging combinatorial problem, even for "simple" forms of cardinality constraints. For instance, it is known that the problem is NP-hard for arbitrary patterns of functional dependencies [CM92]. Furthermore, the actual computational effort needed to compute these bounds might limit applicability in real-world cases. For this reason, the bounder is built around the concept of *bounding strategy.* A bounding strategy s is characterized by a couple of bounding functions that, given $\mathcal{C}, \mathcal{D}$, and V, compute bounds v_s^- and v_s^+ such that $v_\mathsf{s}^- \leq v^-$ and $v^+ \leq v_\mathsf{s}^+$ both hold. In other terms, a bounding strategy never computes bounds which are more restrictive than the ones logically implied by the input constraints, trading-off accuracy for speed of evaluation. We say that a strategy s is *decoupled* iff computing v_s^+ for an arbitrary view V only requires the knowledge of upper bounds w_s^+ of other views W, but no knowledge of lower bounds w_s^-, and vice versa.

Example 2. Consider again the telecommunication company domain, which serves 10^7 telephone numbers during a 10^3 days period on 5 daily time segments, and let $V = \{\mathsf{sourceNumber}, \mathsf{destNumber}, \mathsf{timeSegment}\}$. Using a simple bounding strategy, it is derived that $v \geq 10^7$, since at least one call is made from each number, and $v \leq 5 \times 10^{14}$, since at most each number calls each other number on each time segment. If the cardinality of the base cube is known, for instance it is 10^{11}, a bounding strategy could improve the upper bound of V to 10^{11}, since the cardinality of any aggregate view cannot exceed that of the base cube. Suppose now that the expert of the application domain is capable of providing an additional information: the number of distinct source-destination couples is at most 10^9. From this, we can infer that $v \leq 5 \times 10^9$. □

Turning to the estimator, our framework supports different *probabilistic models.* A probabilistic model is a function m that, given $\mathcal{C}, \mathcal{D}$, V, as well as bounds computed by the bounder, provides an estimate for the cardinality of V. In general, this step can use information from $\mathcal{C}$ that is not suitable to derive bounds. Typically this is the case where a cardinality constraint represents an average value (e.g., the number of calls originating on the average from a given number on each day is 10). For lack of space, in this paper we do not discuss probabilistic models.

4 A Decoupled Bounding Strategy

In this section we focus on issues related to computing upper and lower bounds by means of a decoupled bounding strategy and assuming that the set $\mathcal{C}$ just consists of a set of view cardinalities. In particular, for each $A_i \in U$ we assume that $card(A_i) = a_i \in \mathcal{C}$. This assumption, which is perfectly reasonable in all application domains, is necessary in order to guarantee that at least one upper/lower bound can be determined for each view. In addition, $\mathcal{C}$ may also include the cardinality w_j of some other view W_j. We say that a view is *ground* iff its cardinality is in $\mathcal{C}$.

The basic observation underlying the determination of effective bounds for view cardinalities is that the multidimensional lattice induces an isomorphic structure over such cardinalities. Let I be

[1] For simplicity of notation, we omit the dependence of bounds on $\mathcal{D}$ and $\mathcal{C}$.

an instance of $\mathcal{D}$, and let v stand for the cardinality of view V in I. From Definition 3 it follows that $V \preceq W$ implies $v \leq w \ \forall I$, since $W \rightarrow V$ holds. This inequality also applies to bounds.

Lemma 1. *If $V \preceq W$, then $v^- \leq w^-$ and $v^+ \leq w^+$.*

4.1 Upper Bounding

The strategy we present strongly relies on the concept of *cover* of a view. Let $\mathcal{P}(\mathcal{V}_\mathcal{D})$ be the set of all subsets of $\mathcal{V}_\mathcal{D}$.

Definition 4 (Cover). *Let $V \in \mathcal{V}_\mathcal{D}$ be a view on $\mathcal{D}$ and $S = \{W_1, \ldots W_m\} \in \mathcal{P}(\mathcal{V}_\mathcal{D})$ be a set of views. S is called a V-*cover *iff $V \preceq \oplus(S)$. A V-cover is said to be* ground *when all the views it includes are ground.*

Example 3. In the *Calls* scheme, let $V = \{\mathsf{sourceDistrict}, \mathsf{destDistrict}, \mathsf{timeSegment}, \mathsf{month}\}$ and $\mathcal{C} = \{w_0, w_1, w_2, w_3, w_4, w_5\}$ where

$W_0 = \{\mathsf{sourceNumber}, \mathsf{destNumber}, \mathsf{timeSegment}, \mathsf{date}\}$ $\quad W_3 = \{\mathsf{timeSegment}, \mathsf{month}\}$

$W_1 = \{\mathsf{sourceNumber}, \mathsf{destDistrict}, \mathsf{timeSegment}, \mathsf{date}\}$ $\quad W_4 = \{\mathsf{sourceDistrict}, \mathsf{destDistrict}, \mathsf{date}\}$

$W_2 = \{\mathsf{sourceState}, \mathsf{destDistrict}, \mathsf{timeSegment}\}$ $\quad W_5 = \{\mathsf{sourceNumber}, \mathsf{date}\}$

Some examples of ground V-covers are $S_1 = \{W_1\}$, $S_2 = \{W_2, W_4\}$, $S_3 = \{W_3, W_4\}$, $S_4 = \{W_2, W_5\}$; in fact, it is $V \preceq \oplus(S_1) = \oplus(S_4) = W_1$, $V \preceq \oplus(S_2) = \oplus(S_3) = \{\mathsf{sourceDistrict}, \mathsf{destDistrict}, \mathsf{date}, \mathsf{timeSegment}\}$. In Figure 2 the roll-up relationships between these views are depicted. □

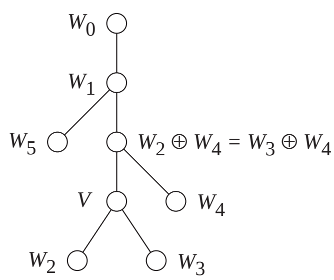

Fig. 2. Roll-up relationships between views in Example 3

The notion of cover leads to generalize Lemma 1 to the case where several views at a time are used to bound from above the cardinality of V. Intuitively, this corresponds to viewing the problem as the one of determining the upper bound of the size of the (natural) join of the views in S.

Lemma 2. *Let V be a view and $S = \{W_1, \ldots, W_m\}$ be a V-cover. Then: $v \leq s^+ \stackrel{\text{def}}{=} \prod_{j=1}^{m} w_j^+$.*

Coherently with Lemma 2, the *cover-based* strategy cb computes upper bounds as:

$$v^+_{\mathsf{cb}} = \begin{cases} v & \text{if } v \in \mathcal{C}, \\ \min\{s^+_{\mathsf{cb}} \mid S \in \mathcal{P}(\mathcal{V}_\mathcal{D}), V \preceq \oplus(S)\} & \text{if } v \notin \mathcal{C}. \end{cases} \tag{1}$$

where $s^+_{\mathsf{cb}} = \prod_{j=1}^{m} w^+_{j,\mathsf{cb}}$.

Even for schemes with only a few attributes, computing v^+_{cb} by directly using Equation 1 is not practically feasible, since the size of $\mathcal{P}(\mathcal{V}_\mathcal{D})$ is $O(2^{2^N})$, where N is the number of attributes in $\mathcal{D}$. Fortunately, we can limit ourselves to consider only a restricted set of V-covers, which are called *minimal ground* V-covers and provide useful, non redundant, bounds. To see how minimal ground V-covers are determined, we need to consider two orthogonal aspects: a *domination* relationship between sets of views and the input information, $\mathcal{C}$. While the former induces a partial order on the bounds obtainable from V-covers, *regardless* of the specific input $\mathcal{C}$, the latter can be used to restrict the set of useful V-covers to only those consisting of ground views.

Definition 5 (Domination). *Let $S_1 = \{W_{1,1}, \ldots W_{1,m}\}$ and $S_2 = \{W_{2,1}, \ldots W_{2,n}\}$ be two sets of views in $\mathcal{P}(\mathcal{V}_\mathcal{D})$. We say that S_1* dominates *S_2, written $S_1 \sqsubseteq S_2$, iff S_2 can be partitioned into m subsets $S_{2,1}, \ldots S_{2,m}$ such that $W_{1,i} \preceq \oplus(S_{2,i})$ $\forall i = 1, \ldots, m$.*

For instance, in Example 3 it is $S_1 \sqsubseteq S_4$, since $W_1 \preceq \oplus(S_4)$. Note that if $S_i \sqsubseteq S_j$ then $\oplus(S_i) \preceq \oplus(S_j)$ necessarily holds, whereas the opposite is not true in general (e.g., $S_3 \not\sqsubseteq S_4$ though $\oplus(S_3) \preceq \oplus(S_4)$).

Lemma 3. *Let S_1 and S_2 be two sets of views. If $S_1 \sqsubseteq S_2$ then $s^+_{1,\mathsf{cb}} \leq s^+_{2,\mathsf{cb}}$.*

Lemma 4. *Let S be a non-ground V-cover. Then there exists a ground V-cover S_1 such that $s^+_{1,\mathsf{cb}} \leq s^+_{\mathsf{cb}}$.*

Definition 6 (Minimal Ground Cover). *A ground V-cover S_1 is* minimal *iff there is no other ground V-cover S_2 such that $S_2 \sqsubseteq S_1$ holds.*

Theorem 1 (Sufficiency of Minimal Ground Covers). *Let $\mathcal{G}_\mathcal{C}(V)$ be the set of minimal ground V-covers. Then $\min\{s^+_{\mathsf{cb}} \mid S \in \mathcal{P}(\mathcal{V}_\mathcal{D}), V \preceq \oplus(S)\} = \min\{s^+_{\mathsf{cb}} \mid S \in \mathcal{G}_\mathcal{C}(V)\}$.*

Example 4. With reference to Example 3, S_1, S_2, S_3, and $S_5 = \{W_2, \{\mathsf{sourceDistrict}\}, \{\mathsf{month}\}\}$ are minimal ground V-covers. □

Although Theorem 1 states that v^+_{cb} can be determined by considering only minimal ground V-covers, Definition 5 does not directly provide a constructive rule to generate them. Nevertheless, based on the results obtained so far, it is possible to derive some rules aimed at reducing significantly the cardinality of the superset of minimal ground V-covers to be generated:

1. A ground view W such that $V \preceq W$ is a ground V-cover (from Definition 4).
2. A ground view W such that $arity(W) = 1$ and $W \cap V = \emptyset$ does not belong to any minimal ground V-cover[2] (from Definitions 5 and 6).
3. A ground view W such that $arity(W) > 1$ and $\forall W'$ for which $W' \preceq W$ it is $arity(W' \cap V) < 2$ does not belong to any minimal ground V-cover (since $\mathcal{C}$ includes the cardinalities of all the attributes).
4. If S is a ground V-cover, no set S' such that $S \subset S'$ is a minimal ground V-cover (from Definition 5).
5. If a minimal ground V-cover S contains a ground view W, it cannot contain any other ground view W' such that $W \preceq W'$ (from Definitions 5 and 6).

[2] $arity(W)$ denotes the number of attributes in W.

We approach the problem of computing the upper bound for a view V given a set of constraints $\mathcal{C}$ using a branch-and-bound algorithm which generates a superset of the minimal ground V-covers by solving a set of subproblems that repeatedly add new views to the partial solution obtained so far. Each subproblem is associated to: (1) a partial solution S containing all the ground views selected so far to build a ground V-cover; (2) an ordered set $T = \{W_i\}$ of the possible ground views to be added to S and compatible with S with reference to the rules above; (3) a function $lb(S)$ which returns a lower bound of the cardinality of the ground V-covers that can be obtained by extending S. The order on T is obtained by considering first the ground views for which the intersection with V has higher arity. It is remarkable that, if we impose that a partial solution containing a ground view W_i can be extended only with ground views W_j such that $j > i$, inducing a total order on T avoids the same cover to be generated twice; furthermore, the chosen order determines an heuristic criterion for generating the "most promising" covers first.

Example 5. We are interested in estimating $V = \{\mathsf{sourceDistrict}, \mathsf{destDistrict}, \mathsf{timeSegment}, \mathsf{month}\}$ in the *Calls* scheme. Table 1 shows how the upper bound of v improves as additional cardinality constraints are supplied as input (see Example 3). The result when all six views are included in $\mathcal{C}$ (besides the cardinality of the views with arity 1) is obtained by building only 6 complete ground V-covers. Seven partial solutions are abandoned since dominated by the current best solution. □

W_i	W_0	W_1	W_2	W_3	W_4	W_5
w_i	10^{11}	10^{10}	10^3	1.8×10^2	10^4	6×10^9
v^+_{cb}	10^{11}	10^{10}	9.36×10^6	9.36×10^6	1.8×10^6	1.8×10^6

Table 1. Improving upper bounds of v for increasing domain-derived information

4.2 Lower Bounding

For a decoupled bounding strategy, there is a striking asymmetry between computing upper and lower bounds. In fact, while computing v^+_{cb} can be cast as a "bounding-a-join-size" problem, the computation of v^-_{cb} corresponds to a "bounding-a-projection-size" problem, where the relevant difference is that projection is a unary operator. This leads to a much simpler situation to deal with, in which only Lemma 1 can be exploited and v^-_{cb} is computed as:

$$v^-_{\mathsf{cb}} = \max\{w \mid w \in \mathcal{C}, W \preceq V\} \ . \tag{2}$$

Differently from upper bounds, no combinatorial issues arise in computing lower bounds through this strategy; thus, complexity is linear in the cardinality of $\mathcal{C}$.

5 Conclusions and Open Issues

In this paper we have shown how cardinality constraints derived from the application domain may be employed to determine effective bounds on the cardinality of aggregate views. In order to devise a comprehensive approach, several issues still need to be investigated. In the following we briefly discuss those we believe to be crucial:

- *Bounding strategies.* The bounds we derive are not necessarily the tightest possible ones. In fact, more complex and effective bounding strategies can be defined to the detriment of computational speed. Basically, in these strategies the concept of cover is extended by considering more complex patterns of views, where upper and lower bounds are used jointly.
- *Cardinality constraints.* The input knowledge for our technique may be extended by considering different kinds of cardinality constraints which are typically known to the experts of the application domain. For instance, knowing that each telephone number calls in the average 10 other numbers on each day, allows the cardinality of view {sourceNumber,destNumber,date} to be estimated as 10 times the cardinality of view {sourceNumber,date}.
- *Probabilistic estimates.* Assuming that effective bounds have been derived, cardinality estimation will be based on a probabilistic model. The most used model to this end is the one described in [Car75], which bases its estimate for view V on the maximum cardinality of V and on the cardinality of a view W such that $V \preceq W$. Within this topic, we will work on improving this rough estimate by taking all the information collected so far into account.

References

[AGS97] R. Agrawal, A. Gupta, and S. Sarawagi. Modeling Multidimensional Databases. In *Proc. ICDE'97*, pages 232–243, Birmingham, UK, 1997.

[Car75] A.F. Cardenas. Analysis and Performance of Inverted Database Structures. *Communications of the ACM*, 18(5):253–263, 1975.

[CM92] P. Ciaccia and D. Maio. On the Complexity of Finding Bounds for Projection Cardinalities in Relational Databases. *Information Systems*, 17(6):511–515, 1992.

[CM95] P. Ciaccia and D. Maio. Domains and Active Domains: What This Distinction Implies for the Estimation of Projection Sizes in Relational Databases. *IEEE Transactions on Knowledge and Data Engineering*, 7(4):641–655, 1995.

[GHRU97] H. Gupta, V. Harinarayan, A. Rajaraman, and J. Ullman. Index Selection for OLAP. In *Proc. ICDE'97*, pages 208–219, Birmingham, UK, 1997.

[GL97] M. Gyssens and L.V .S. Lakshmanan. A Foundation for Multi-Dimensional Databases. In *Proc. 23rd VLDB*, pages 106–115, Athens, Greece, 1997.

[GR00] M. Golfarelli and S. Rizzi. View Materialization for Nested GPSJ Queries. In *Proc. DMDW'2000*, Stockholm, Sweden, 2000.

[Gup97] H. Gupta. Selection of Views to Materialize in a Data Warehouse. In *Proc. ICDT'97*, pages 98–112, Delphi, Greece, 1997.

[HO91] W. Hou and G. Özsoyoglu. Statistical Estimators for Aggregate Relational Algebra Queries. *ACM Transactions on Database Systems*, 16(4):600–654, 1991.

[HRU96] V. Harinarayan, A. Rajaraman, and J. Ullman. Implementing Data Cubes Efficiently. In *Proc. ACM Sigmod Conf.*, pages 205–216, Montreal, Canada, 1996.

[MCS88] M. V. Mannino, P. Chu, and T. Sager. Statistical Profile Estimation in Database Systems. *ACM Computing Surveys*, 20(3):191–221, 1988.

[MD88] M. Muralikrishna and D.J. DeWitt. Equi-depth Histograms for Estimating Selectivity Factors for Multi-Dimensional Queries. In *Proc. ACM Sigmod Conf.*, pages 28–36, Chicago, IL, 1988.

[RS97] K. Ross and D. Srivastava. Fast Computation of Sparse Datacubes. In *Proc. 23rd VLDB*, pages 116–125, Athens, Greece, 1997.

[SDNR96] A. Shukla, P. Deshpande, J. Naughton, and K. Ramasamy. Storage Estimation for Multidimensional Aggregates in the Presence of Hierarchies. In *Proc. VLDB*, pages 522–531, 1996.

[TB00] D. Theodoratos and M. Bouzeghoub. A General Framework for the View Selection Problem for Data Warehouse Design and Evolution. In *Proc. DOLAP 2000*, pages 1–8, Washington, 2000.