Small Database Answers for Small Mobile Resources *

A. Lubinski

Computer Science Dept., University of Rostock, Germany

Abstract. *Useful mobile database access depends on available resources of handhelds and wireless networks. Smaller database answers cope better with this situation. The benefits of smaller database responses are faster transfer times and therefore smaller transfer costs, and decreased power and memory consumption of mobile devices.*
Existing research approaches for data reduction do not exactly meet reduction requirements of mobile environments, or just on restricting conditions, respectively. Approximate answers are sometimes insufficient and fast transfers of answers over wireless links are more important than short execution times on fixed sites. In this paper, we study database techniques that issue smaller answers by restricting the quality or the quantity of data to meet resource requirements. We outline various reduction mechanisms and present a hybrid technique of a designed and a progressive technique for supporting mobile users.

Keywords. Mobile Database Systems, Data Reduction, Text Retrieval, Histograms.

1 Introduction

Mobile computing and communication are of growing interest. In particular, portable computers coupled with wireless networks open opportunities for a wide variety of information processing on the move. Low power devices and wireless links make nomadic computing possible but their limited resources like small power capacities, limited storage, restricted and expensive transfer can be very restricting

*This work is part of the Mobile Visualization Project (MoVi) and has been supported by the German Research Association under contract Schu 887/3-2

features. Besides location awarenesses, adaptation to available resources is therefore a main issue in mobile environments.

There are two possible strategies to handle limited device resources: The first strategy is to reserve the necessary resources if available, but to the debit of any other application or information. The second way is to restrict the amount of all data fairly by deteriorating the information quality. If this problem is solved, the challenge remains to achieve infrequent and short transfers of data over wireless links. Therefore, adaptation issues often require reducing techniques to support sensible computing and communication, despite limitations. Reduction is an adaptation concept in mobile environments, and it is a context sensitive process, where we define the mobile context (more detailed in [HL96]) describing relevant environment conditions as consisting of

- persons (roles, communicating persons, etc.),
- resources (available hardware and software resources of the mobile host and the communication network),
- the information (for example type, size, including relevant meta information) and
- the central context: location/time, which other contexts depend on.

In other words, adaptation optimizes data against the mobile context and especially reductions realize adaptations to restricted contexts.

Example: A mobile-working sales representative requires information on the mobile device about the customers he visits. The context *persons* covers location dependent tasks (context *location* covers the current user's location) of representatives and communicating roles of customers for preparation and execution of an advice.

Transferring whole files including images for offers for the customers visited one day would exceed capabilities of representative's PDA. The context *information* covers type, required storage and alternative files whereas the context *resources* contains the PDA's available storage, and transfer time or costs for transferring required data. Resource context and information context are disproportionate in this example. Bringing both of them in accordance requires reduction of requested data.

There are three entries in the query evaluation process for reducing data. Reductions are transformations of the query to get less data than required, accessing smaller stored data, or reducing the query result. Moreover, we argue that reduc-

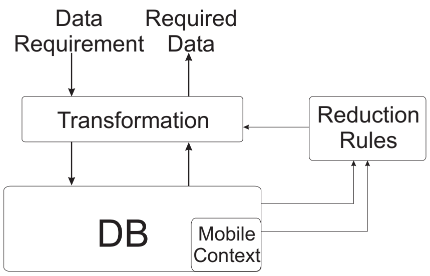

Figure 1: Adaptive Query Evaluation Process

tion techniques consists of not only an abstraction, a projection, and a selection (in reference to the basic database operations), but also a sorting and replacing technique.

Compression techniques as well as caching strategies are topics of many discussion whereas reduction based on information access is not a focus of research. We will consider semantic e.g. content-based reduction.

The rest of the paper is organized as follows. In Section 2, we discuss the possible transformations and outline the described characteristics of reduction. Section 3 is a review of related work. In section 4, we present a hybrid and gradual working approach for reducing data. Section 5 concludes the paper with final remarks.

2 Data Reduction and Database Techniques

The three transforming or selecting entries in the query evaluation process realize adaptation and namely reduction. Figure 1 illustrates the meta data of the mobile context which control these transformations or access to reduced data based on reduction rules. Consider a relation R with n attributes $A_1, ..., A_n$, where a tuple is defined as $t = (a_1, ..., a_n)$ with a_i is a value in the Domain of attribute $A_i : a_i \in A_i$. The type of attributes define the size of tuples (we consider fixed lengths of types), so that a_i of one tuple has the same length as b_i of another tuple in the same relation where $a_i, b_i \in A_i$. We define the size s of tuple t:

$$s(t) = \sum_{i=1}^{n} s(a_i)$$

The size S of a query result contains n tuples with size s: $S = n \times s(t)$. Let l be the location. It is not important for the explanations whether the current location is estimated or determined (by GPS, e.g.). Furthermore, we assume that location dependencies of data are known. Locations can build hierarchies.

2.1 Reduction Entries for Database Systems

Transformation of the Query. There is a simple transformation of the database request to issue reduced database result by extending the WHERE-clause by location descriptions, for example.

Example: The sales representative requests the necessary customer data for two days (p qualifies the customers):

SELECT * FROM customer c WHERE p

The available resources and the amount of data are disproportionate. Therefore, a qualification is added selecting customers on the base of in his or her local environment:

SELECT * FROM customer c WHERE p AND c.address=location

Early transformation reduces access time but only estimates the size of the result. Provided and required sizes can differ. Moreover, query transformations are unaware of changing mobile contexts during process time and can lead to unforseeable discrepancies.

Reduction of Database Data. Another possible reduction consists of decreasing the amount of data in the database itself. That is, we build database segments with the help of views or use database statistics like histograms in order to reduce data. An access to such data reduces the amount of query results.

Let R_1 be a base relation and R_2 a reduced view of $R_1 : R_1 = (A_1, ..., A_n)$ and $R_2 = (A_1, ..., A_m)$ with $m \leq n$. Then, for any projection or selection operation $o \in \{\pi, \sigma\} : S(o(R_1)) \leq S(o(R_2))$

The result is reduced in relation to the current context. Managing only a reduced view on data does not provide any access getting unreduced answers. Therefore, it is important to decide about reduction instead of or additionally to the original data.

Transformation of Query Results. The transformation of results is more flexible than both of the other reduction entries and notices actual context better than query modifications, but this has consequences for the answer time. Query modification and access to reduced data generally need query evaluation times with $T_1 < T$, where T represents the evaluation time for the original query and T_1 the evaluation time of the transformed request. Reduction of results need $T + T_2$, where T_2 represents the time for the result reduction. As previously discussed, transfer time is usually more important than evaluation time because of disconnections. But if a mobile client is waiting online for an answer, this strategy is inefficient.

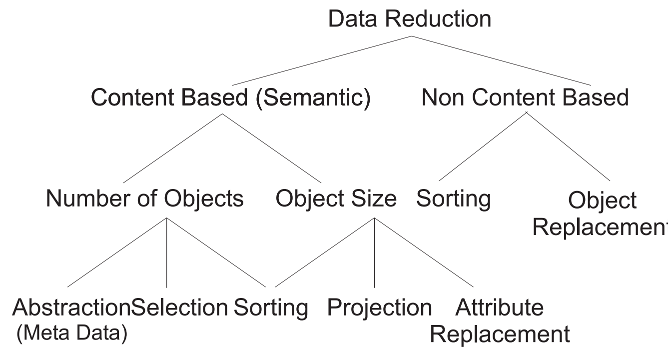

Figure 2: Data Reduction Techniques

Therefore, the particular situation should dedicate the applicable strategy. Various filtering and sorting techniques for query results are imaginable for fulfilling resource requirements.

2.2 Database Techniques for Data Reduction

Figure 2 classifies characteristics of these reduction techniques. Leaf nodes represent a special database technique or the database information for implementing the reduction. Reductions are able to observe semantics of data and are in relation to database operations

- abstraction
- projection
- selection,
- sorting
- replacing, or
- hybrid techniques.

As it relates to these characteristics, there is the question whether the original type should be kept or modified. Moreover, in reference to the database life cycle, it is interesting to ask who is the initiator of reduction and who maintains the reduction rules. Designed reduction requires static environments rather than user driven reduction.

We now provide an overview of the techniques itemized above and the database means for their realization considering questions of type modifications and influences to reduction rules. More detailed explanations are provided in [HL98].

[BDF+97] realized that data reduction is often regarded as being closely associated with aggregation. Aggregation is an abstraction concept that calculates meta-data about a collection of objects like sum and average or the number of resulting tuples. But, database systems support more opportunities for reduction than aggregations based on histograms. In database systems, the knowledge of explicitly designed or implicitly identified priorities of attributes or attribute values are selective factors that could be used to extend user queries. *Selections* reduce a collection of objects to a smaller ones because of selection predicates and the number of objects is decreased. The resulting object type is unchanged.

Projections shrink the number of attributes for a collection, i.e. in contrast to selections, the object size and the resulting type are modified.

Besides aggregations, *abstractions* are based on statistics (histograms for aggregations) or sampling techniques for decreasing the number of tuples, or they use the metadata of the database system stored in the data dictionary or supplementing relations (see [Mot94]). This kind of reduction modifies the resulting type.

Replacements work mainly data type (medium) dependent. The more complex the types are, the more resources are necessary for their access, transfer, and presentation. Reduction of multimedia objects such as videos and images, are not discussed in this paper. We focus on replacing one object or attribute of an expensive type with one object of a less expensive data type (e.g. replacing images by their description).

Techniques for qualitative deterioration can profit from various attribute relationships like functional dependencies or integrity rules. Natural-key attributes could not be removed without appearing misinterpretations, for example.

Reduction is processed by context-dependent reduction rules. It is difficult for mobile users to influence and understand these rules when they are statically designed and managed, especially with strong reductions. On the one hand, the inadequacy of requirements of resources to query result decreases with strong reductions, whereas on the other hand, the inadequacy of required and expected to the automatically reduced result increases.

Progressively provided results serve as a direct control by the user remedying the inadequacy in understanding of the result. Users stop the transfer and presentation of results, and induce access for more detailed information explicitly. In the database environment, this is achievable by *ordered* transfer and output of the results.

Example A sales representative wishes to load article data:

SELECT * FROM article a WHERE p

We assume that answers include expensive data. The query is modified to issue the most popular (most ordered) articles first and the user is able to stop the transfer after any time:

SELECT * FROM article a WHERE p ORDER BY a.order DESC

Ordering is a technique reducing the number of objects as well as the object size. Projected attributes can be removed, or substituted, or they can be ordered. Sorting because of projections exchanges the presentation dimensions. Important attributes are placed at the beginning of the resulting data stream and marginal attributes are ordered at its end.

Note that users can participate in reduction by progressive access as well as by selecting the level of reduction (see section 4).

There are research approaches not primarily supporting mobile environments, but which can be used for this purpose.

3 Related Work

In this section, we outline related work. There are no approaches meeting our reduction goals for mobile environments, but the diversity of reduction methods mentioned above makes it necessary to consider various research approaches.

[ZD95] suggested a filtering mechanism for retaining information in restricted environments. The authors presented the idea of positioning an intermediary (proxy) between client and server. The proxy served as "reachability point" for the mobile client. However, a fixed host communicating with a mobile host would be unaware of the intermediary. The filtering methods implemented by these proxies were only listed in general and exemplary. Our focus is on these filtering and other reducing techniques and their possible implementation. However, the proposed placing of filtering methods is useful.

Motro [Mot89, Mot92, Mot94] proposed supplementing answers to database queries with their properties. These extensional answers including, for example, statements of soundness and completeness should help users assess the value and meaning of the information they receive. Queries were evaluated on relations as well as on additional meta-relations containing explicitly defined properties. Extensional data by themselves are not expressive enough to represent intensional data because of their annotating characteristics. Thus, Motro's techniques can be applied to reduce answers, but only in connection with other techniques or data, respectively.

Other reduction approaches for database systems are concerned with fast and ranked answers and take into consideration approximate answers that are based often on histograms for increasing performance. The Data Reduction Report [BDF+97] summarizes techniques for fast, approximate answers used for data warehouse applications. Various papers by Poosala and Ioannidis [Poo97, PGI99] described approximate answers using histograms. Such answers can be meaningful data-reduction techniques in mobile environments but do not lead in principle to smaller answers. Small response time is necessary to decrease mobile connection time but are very irrelevant in case of frequent disconnections. Therefore, it is important to reduce the data transfer through a reduced amount of answers.

Approximations are also a drawback if the applications require exact data. Unfortunately, histograms need to be created and maintained, which is impossible when dynamic data resources should be managed in histograms on mobile units. In this paper, we present further techniques for correct data support.

Top-k queries provide correct data but do not exactly match the query. The k closest tuples are the query result. The cardinality of a query result is restricted by a user specified k. Therefore, query answers are *ordered* sets of tuples. Carey and Kossmann [CK97, CK98] proposed an extension of SQL using *STOP AFTER k*. Computation of the query stopped after k results with the help of a stop operator in the execution plan. Termination requires at least partial sorting or computing of the distance (score) which can be very expensive.

[CG99, DR99] proposed top-k queries based on histograms thereby avoiding expensive preprocessing. A minimal "'score" of tuples arranged in a histogram according to the query is determined. The frequency of the data is important for the estimation. Then the query is modified for resulting tuples with a score equal or greater than this score. If the query issues less than k tuples, then a lower score (less selective filter) must be chosen and the process must be restarted. Two strategies are possible to choose a score. Selecting a very low score may lead to more tuples than necessary and increases query evaluation time, but it avoids restarts. [DR99] proposed a probabilistic approach to quantify the risk of restarts that can appear in the second strategy.

Top-k queries are applicable in mobile environments. Our assumption is that there are typical queries, even in mobile environments i.e. queries expecting exact answers. However, the mobile context and especially the user location, requires an ordering and limited resources demand the restriction of query results. Applying top-k queries requires finding out another ordering criterion beside neighborhood of tuples (score). Moreover, k is well suited for being controlled by mobile contexts.

4 Gradual Data Reduction

We are convinced of that the user's focus is dependent on the mobile context. *Resource context* is assumed to be restricted. A data reduction is necessary because of the complete database reply is in contrast to the available resources. All focused data are relevant but their importance to the user decreases with increasing distance to the focus. This makes an increasing reduction with extended distance of the user's location possible.

Example: In *location* dependent applications like tourist guides, customers wish to be informed about the sights near at hand and less detailed about more remote sights. Data for a current task in an *application* are more interesting than such of the last or the next step. These examples are applicable to the context *person*.
For the rest of the paper, we focus on the location, but statements are transferable to other contexts.

This assumption of diminishing importance lead consequently to the concept of "fisheye views" in the computer graphics where the quality of presentation decreases uniformly with the distance from a central point. We will map these idea to database systems with the help of the reduction techniques mentioned above. We define three main domains of data precision in relation to the importance of data to the focus which are attained by at least three reduction layers enforcing different layers of data reduction.

4.1 Domains of Data Precision

We define for the present three main domains for specifying the precision of the result in relation to the user's focus. The more important data in the specified focus are, the more precise they have to be.
Domain 1 is the domain closest to the user associated with his actual needs. *Domain 2* contains semi-important data which could be interesting in the near future (because of a conceivable move, e.g.). *Domain 3* defines the domain for data of marginal interest. They are necessary to be the context for data in *Domains 1* and *2*.
A further partitioning of *Domain 2* as presented in the figure is imaginable and is explained later on. We represent the three domains as follows:

$$Domain\_number = \begin{cases} 1 & : \quad location \leq l_1 \\ 2 & : \quad location > l_1 \wedge location \leq l_2 \\ 3 & : \quad otherwise \end{cases}$$

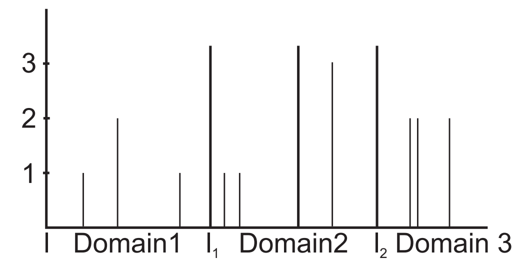

Figure 3: Tuples in three location domains

location represents the current user location. l_i is a domain boundary, limiting the levels of interest.

Example: In the example of representatives, domain boundaries are predefined on the base of local distances. A histogram grouping customers in domains is processed for improving access. There are three interesting domains.

4.2 Layers of Data Reduction

Moreover, we define layers of data reduction. The particular set of reduction rules on each layer determine resulting answer sizes. On the first one (*Layer 1*), the complete database reply without any reduction is provided. The extreme reduction (*Layer n*) issues a strong abstraction. E.g., users receive only the information that there is any result. We implemented providing the number of the resulting objects (or tuples) as the content of this layer, for example. Low semantics without knowledge of any context inhere in such an answer.

Beside these two others, *Layer 2* .. *Layer n-1*, can not be specified generally since several applications and data semantics make differing reduction techniques possible and necessary. Reduction rules will be applied in the previous described manner.

Example: The data of the four customers in *Domain 1* are provided as required. A *Layer 2* reduces tuples of *Domain 2*, and *Domain 3* contains the rest of tuples queried. Users see only the number of customers. The size of resulting tuples is computable.

4.3 Gradual Data Reduction

In the reduction process, now the reduction rules of the reduction layers L are mapped to the precision domains D: $D \rightarrow L$. The complete answer to a query q is $r = r_1 \cup r_2 \cup r_3$ (see Figure 4).

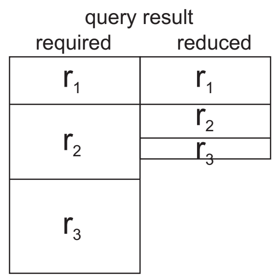

Figure 4: Layered Reduction

The modified query delivers after data reduction the diminished answer illustrated on the right of figure 4. r_i is the query result of domain D_i. r_1 has the same size, because it is not reduced. r_2 illustrates a reduced part compared to the original answer, and r_3 delivers a very small answer like number of tuples. In the example of Figure 3 , it would be reduced data in $r_2(D_2)$ of five tuples and the data of $r_3(D_3)$ = "6" as the reduction up to the number of tuples. The selection of reduction rules in a choice of available reduction layers for the specific domain can be designed or dynamically user driven. We now explain an example reduction more detailed.

The *customer* relation is defined as

customer = (name, address, city-map, preferred-articles, remarks)

Remarks are textual remarks of representatives referred to customers. The representative now requires all customer data:

SELECT * FROM customer c WHERE p

We apply query modifications for reducing query results. The query is split into three queries related to three domains of interest. Customer data described in the first zone are delivered completely with the query:

1. SELECT * FROM customer c WHERE p AND c.address $\leq l_1$

The query provides unreduced data for the current location domain. but there is the condition of the adequacy of available and required resources. Consequently, the location domain possibly have to shrink to an acceptable measure.

PDA's are usually not able to present animations. Assuming, a queried relation *article* includes animations, there are no data which are transferable unreduced, i.e. *Domain 1* would be empty.

Query three reduces data in *Domain 3* and delivers only the number of customer tuples in that location domain:

3. SELECT COUNT(*) FROM customer c WHERE p AND c.address $> l_2$

We now define reduction for *Layer 2* assigned to *Domain 2*. We are using a projective reduction:

2. SELECT name, remarks FROM customer c WHERE p AND c.address > l_1 AND c.address $\leq l_2$

Actually, we are testing data reduction by using text retrieval functionality creating abstracts from texts like *remarks* in the relation above. Its application would delivers two reductions for one step, projection of a smaller attribute set and replacing of text with a shorter abstracted text.

Another possible reduction which can be mapped to this domain is the modification of the query result consisting in an ordering of attributes on the base of explicitly defined priorities:

attribute	name	address	city-map	preferred-articles	remarks
priority	1	4	8	3	2

The user is able to cut the output data stream which is ordered by attributes because of their priorities. Additionally, the *city-map* attribute is only issued on demand because of its low priority.

We now assume that despite reduction, the resulting size is still inadequate to the available resources, or the reduction presents data with insufficient semantics. Therefore, *Domain 2* is partitioned into two sub-buckets. Partitioning criterion can differ from the domain building criterion.

The first sub-domain is reduced with query 2. For the second one, we are using a top-k query delivering only one of the customers in an address-domain. The suitable OQL-query would be:

2b. SELECT address, (SELECT * FROM partition fetch first only) as y
FROM customer GROUP BY address

Partitioning into sublayers results in an 1:n relationship between domains and reduction layers. Query number *2* (now *2a.*) reduces projective whereas *2b* diminishes results with the help of a selection. It depends on the particular data if *2a* or *2b* delivers less data sizes. The following query retrieves less or equal data than query *2a*, because of a stronger WHERE clause:

2c. SELECT name, remarks FROM customer c WHERE p AND c.address > l_1 AND c.address $\leq l_2$ AND c.preferred-articles=article

Moreover, giving users more opportunities for influencing data reduction, we predefine reducing views. *2a*, *2b* and *2c* will be created as views. Additional views and view hierarchies (like *2a* and *2c*) are possible. (E.g., *2c* is a subnode of *2a*) in the tree of reducing views.

Let v_i and v_j be views. They are in a view hierarchy if $v_i \subseteq v_j$, e.g. v_i results in a more reduced answer than v_j. An answer is more reduced if its size is decreased compared to other answers.

Summarizing, users have the following opportunities for influencing data reduction and for bridging the gap between queried and received answers. They are required to determine or modify domain boundaries, they select the desired reducing views, and they stop the transfer of ordered data streams.

5 Conclusion

We outlined in this paper database techniques and extensions to database techniques for diminishing query results for fitting in restricted mobile resources. We used query modifications, views, database statistics, and metadata for implementing reduction of database results. Additional functionality is necessary if the query result should be modified.

Limiting answer sizes for mobile environments can be achieved by homogeneous deterioration of number or quality of the resulting objects. An inhomogeneous enforcement of reduction techniques in relation to the user's focus is more satisfying user needs. Therefore, we introduced on the one hand domains of data precision and on the other hand layered reductions, i.e. data in the near environment of an user are generally unreduced because of their actual importance. With increasing distance of the focus, the degree of reduction increases. The domains specify the focus of interest based on the user's location which are assigned sets of reduction rules. These rule sets can be predefined possibly in hierarchies and selected by the user.

This kind of layered reduction is able to meet the requirements of restricted resources gradually, and provide mobile user's information needs adaptive under inclusion of user's interaction, if he wishes.

References

[BDF+97] Barbará, D. ; DuMouchel, W. ; Faloutsos, C. ; Haas, P.J. ; Hellerstein, J. M. ; Ioannidis, Y. ; Jagadish, H.V. ; Johnson, T. ; Ng, R. ; Poosala, V. ; Ross, K.A. ; Sevcik, K. C.: The New Jersey Data Reduction Report. In: *Bulletin of the Technical Commitee on Dara Engineering* 20 (1997), Nr. 4, S. 3–45

[CG99] Chaudhuri, S. ; Gravano, L.: Evaluating Top-k Selection Queries. In: Atkinson, Malcolm P. (Hrsg.) ; Orlowska, Maria E. (Hrsg.) ; Valduriez, Patrick (Hrsg.) ; Zdonik, Stanley B. (Hrsg.) ; Brodie, Michael L. (Hrsg.): *VLDB'99, Proceedings of 25th International Conference on Very Large Data Bases, September 7-10, 1999, Edinburgh, Scotland, UK*, 1999, S. 397–410

[CK97] Carey, M. J. ; Kossmann, D.: On saying "Enough Already!" in SQL. In: Peckham, Joan (Hrsg.): *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona*, 1997, S. 219–230

[CK98] Carey, M. J. ; Kossmann, D.: Reducing the braking distance of an SQL query engine. In: Gupta, Ashish (Hrsg.) ; Shmueli, Oded (Hrsg.) ; Widom, Jennifer (Hrsg.): *VLDB'98, Proceedings of 24th International Conference on Very Large Data Bases, August 24-27, 1998, New York City, New York, USA*, 1998, S. 158–169

[DR99] Donjerkovic, D. ; Ramakrishnan, R.: Probabilistic Optimization of Top N Queries. In: Atkinson, Malcolm P. (Hrsg.) ; Orlowska, Maria E. (Hrsg.) ; Valduriez, Patrick (Hrsg.) ; Zdonik, Stanley B. (Hrsg.) ; Brodie, Michael L. (Hrsg.): *VLDB'99, Proceedings of 25th International Conference on Very Large Data Bases, September 7-10, 1999, Edinburgh, Scotland, UK*, 1999, S. 411–422

[HL96] Heuer, A. ; Lubinski, A.: Database Access in Mobile Environments. In: Wagner, R. R. (Hrsg.) ; Thoma, H. (Hrsg.): *Proceedings of the 7th International Conference on Database and Expert Systems Applications*, 1996, S. 544–553

[HL98] Heuer, A. ; Lubinski, A.: Data Reduction - an Adaptation Technique for Mobile Environments. In: *Proceedings of the 2nd IMC*, 1998, S. 1–2

[LL96] Lee, W.-C. ; Lee, D. L.: Using Signature Techniques for Information Filtering in Wireless and Mobile Environments. In: *Distributed and parallel databases* (1996), Nr. 4, S. 205–227

[Mot89] Motro, A.: Using Integrity Constraints to Provide Intensional Answers to Relational Queries. In: Apers, Peter M. G. (Hrsg.) ; Wiederhold, Gio (Hrsg.): *Proceedings of the Fifteenth International Conference on*

Very Large Data Bases, August 22-25, 1989, Amsterdam, The Netherlands, 1989, S. 237–246

[Mot92] MOTRO, A.: Annontating Answers with Their Properties. In: *SIGMOD RECORD* 21 (1992), Nr. 1, S. 54–57

[Mot94] MOTRO, A.: Intensional Answers to Database Queries. In: *IEEE Transactions on Knowledge and Data Engineering* 6 (1994), Nr. 3, S. 444–454

[PGI99] POOSALA, V. ; GANTI, V. ; IOANNIDIS, Y.E.: Approximate Query Answering using Histograms. In: *Bulletin of the Technical Commitee on Dara Engineeri ng* 22 (1999), Nr. 4, S. 5–14

[Poo97] POOSALA, V.: *Histogram-Based Estimation Techniques in Database Systems*, University of Wisconsin-Madison, Dissertation, 1997

[ZD95] ZENEL, B. ; DUCHAMP, D.: Intelligent Communication Filtering for Limited Environments. In: *Hot Topics in OS*, 1995