

# Self-Organizing Databases: Near-Optimal Query Performance at all Times Using Flexible Views

Rada Chirkova  
North Carolina State University

## Project Summary:

The goal of this project is to develop new effective methods to improve the performance of sets of frequent and important queries on large relational databases at all times, which could improve the efficiency of user interactions with data-management systems. Solving the problem will have the most effect in query optimization, data warehousing, and information integration, which are important research topics with direct practical applications. Moreover, our research program offers a unique test case for a fundamental understanding of optimality in answering queries and, more generally, in database performance. The project focuses on the methodology of evaluating queries using *views*; views are relations that are defined by auxiliary queries and can be used to rewrite and answer user queries. One way to improve query performance is precompute and store (i.e., *materialize*) views.

To truly optimize query performance, it is critical to materialize the "right" views. The current focus of the project is on demonstrating that, by designing and materializing views, it is possible to ensure optimal or near-optimal performance of frequent and important queries, for common and important query types. We consider this problem in the broader context of designing *self-organizing* databases: A self-organizing database periodically determines, without human intervention, a representative set of frequent and important queries on the data, and incrementally designs and precomputes the optimal (or near-optimal) views for that representative query workload. As the representative query workload and the stored data change over time, self-organizing databases adapt to the changes by changing the set of materialized views that are used to improve the query-answering performance in the database. This approach has a potential to lead to dramatic improvements in the efficiency of user interactions with many types of data-management systems. Solving the problem of building self-organizing databases will have the most effect in query optimization, data warehousing, and information integration.

For building self-organizing databases, we consider an *end-to-end solution* – that is, we consider all aspects of handling and using views, including:

- designing and materializing views and indexes to improve query performance;
- exploring the effects of materialized views on the process of query optimization;
- adapting view design to the changing query workload, including the process of retiring views that are no longer useful;
- developing methods for automatically updating existing materialized views over time, to reflect the changes in the stored data;
- developing methods to collect database statistics to reliably estimate the sizes of the views the system considers for materialization;
- analyzing the use of system resources and allocating an appropriate amount of resources to view management in the system.

In our research in self-organizing databases, we adopt an approach that combines theoretical work (for some aspects of the project) and extensive implementation (C++) and experimentation based on an open-source database system called PostgreSQL, <http://www.postgresql.org>. The students involved in the project will get hands-on experience in conducting research in databases, will develop implementation skills in database systems, and will learn to set up, conduct, analyze, and report experiments related to database performance on large amounts of data.

In the current stage of the project, we are setting up PostgreSQL, and are also concentrating on developing efficient and scalable heuristic algorithms that design (near-) optimal sets of views for the given queries. This project has two parts: (1) theoretical analysis and design of algorithms and heuristics for view design, and (2) implementation and experiments on large databases, to evaluate the performance improvements caused by using the views. Funding for research in self-organizing databases is likely in the near future; success in obtaining funding is contingent on the success of the ongoing stages of the project.

## Project Impact:

This project offers a unique test case for a fundamental understanding of optimality in answering queries and, more generally, in database performance costs. The project will provide educational and research experience opportunities for graduate and undergraduate students. We anticipate collaboration with owners and users of large databases, to

improve the efficiency of the queries the users ask on their data, while at the same time testing the techniques proposed in the project. For instance, we plan to collaborate with other departments - primarily in the sciences - at North Carolina State University, with other academic institutions and with companies in the private sector of the North Carolina Research Triangle. The techniques resulting from this project could have application in commercial and experimental database systems, where they will provide new ways to lower query-processing costs. The research results will be accessible via the project web site <http://research.csc.ncsu.edu/selftune/>, publications, and freely disseminated software.

### **Area Background:**

In a number of applications of modern databases, many users pose complex queries on the data at the same time. Processing numerous complex queries simultaneously and efficiently is a nontrivial task for a database-management system; as a result, some or most users may experience slower-than-desired response to their queries. At the same time, if the database system receives some queries over and over again, it is typically possible to significantly improve their response time, by precomputing and storing in the database auxiliary data, called *views*, and by using the views in the computation of the queries. For instance, in a relational database where all data is stored in tables, a precomputed view becomes just another stored table, which may be used, alongside the original stored data, to answer queries on the database.

The best way to improve query performance is, of course, to precompute and store the answers to all frequent and important queries. Unfortunately, this trivial and optimal solution is often unacceptable, because the new stored data has to satisfy pre-existing constraints on the database system. A common constraint is the amount of disk space allocated for storing the new data; large answers to complex queries typically do not satisfy that constraint.

We explore the approach of finding the “right” (optimal) views, that is, views that satisfy the given database constraints and reduce, as much as possible, the response time for most or all frequent and important database queries. This approach has a potential to lead to dramatic improvements in the efficiency of user interactions with many types of data-management systems. Solving the problem will have the most effect in query optimization, data warehousing, and information integration, which are important research topics with direct practical applications. Moreover, our research program offers a unique test case for a fundamental understanding of optimality in answering queries and, more generally, in database performance.

### **Project Website:**

<http://research.csc.ncsu.edu/selftune/>  
(under construction)

### **Publications and Products:**

- Rada Chirkova and Chen Li. Materializing Views with Minimal Size to Answer Queries. *Proceedings of the 22nd ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (PODS-2003).
- Rada Chirkova, Alon Y. Halevy, and Dan Suciu. A Formal Perspective on the View Selection Problem. *The VLDB Journal* (invited paper), 11(3):216-237, 2002.
- Rada Chirkova. The View-Selection Problem Has an Exponential-Time Lower Bound for Conjunctive Queries and Views. *Proceedings of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (PODS-2002).
- Rada Chirkova and Michael R. Genesereth. Linearly Bounded Reformulations of Conjunctive Databases. *Proceedings of the First International Conference on Computational Logic*, July 2000.

### **Specific objectives of the current stage of the project**

In this project, we plan to initiate research in self-organizing databases. The specific objectives of the project are as follows.

The first specific objective of the project is to set up an experimentation testbed for the project, based on an open-source database system called PostgreSQL, <http://www.postgresql.org>. We plan to analyze and document existing PostgreSQL code and to enhance it with basic modules for using views in query optimization.

The second specific objective is to theoretically analyze and characterize *optimal views* – that is, views that globally minimize the response time of the given frequent and important queries under the given database constraints. We expect the main outcome of this stage of research to be a constructive characterization of optimal views for common and important types of database and query scenarios.

The third specific objective is to develop and test algorithms to design optimal or *near-optimal* sets of views, for common and important types of database and query scenarios. The outcome here will be a set of algorithms, with one algorithm for each common and important problem setup, with emphasis on problems under the storage-limit constraint (i.e., the constraint on the amount of disk space available for storing the auxiliary views). Given appropriate problem inputs, each algorithm will produce a set of views that is optimal or near-optimal for the given input.

Achieving these objectives will allow us to gain a solid understanding of the structure of optimality in query execution and will prepare the research group for next stages of research on self-organizing databases. We will incorporate the algorithms, see the third specific objective above, in the research prototype of a database system based on PostgreSQL, and will use the prototype to rigorously test the algorithms.

### **Relevance to the field of study**

This section and the next section point to references to relevant publications. Both the references and a short survey of the current state of knowledge in the field of designing and using views are available at <http://www4.ncsu.edu/~rychirko/literatureOnViews.pdf>.

This project will build on past work in designing, precomputing, and storing (i.e., *materializing*) views to compute queries more efficiently. The original motivation for view design comes from information-integration applications [19]. From several independent databases or other sources of information, an information-integration application builds a database that combines the data from the sources. One approach to information integration, called data warehousing [52], uses materialized views. In data warehousing, the source data is stored in a central location – a warehouse – after possibly being processed in some way (e.g., joined, filtered, or aggregated). The problem of data-warehouse design is to decide which views to store in the warehouse to obtain optimal query performance [3, 24, 27, 31, 46, 47, 53].

Another motivation for view design is provided by recent versions of several commercial database systems, which support incremental updates of materialized views and now use materialized views to speed up query evaluation [4, 21, 54]. Choosing an appropriate set of views to materialize in a database is crucial to obtain performance benefits from these new features [2]. Finding the “right” views to materialize will also play a role in contexts where data needs to be placed intelligently over a wide-area network: Materialized views can be used to reduce the amount of inter-node communication in processing distributed queries [22, 35].

Because computing queries efficiently using materialized views is important in many data-management applications, the importance of designing the “right” views is well recognized in the database community. The algorithms and techniques resulting from this project could enable or enhance information access and analysis for broad categories of users, thereby augmenting human intelligence in problem-solving tasks in a variety of applications. Moreover, the research area offers a unique test case for a fundamental understanding of the nature of optimality in query execution and, more generally, in database performance.

### **Previous work and status of other investigations**

As was discussed in the previous section, computing queries efficiently using materialized views is important in many data-management applications. At the same time, in commercial database applications views are commonly designed in an ad hoc manner by database administrators; see, for example, [32]. For these reasons, the problem of automatically finding useful views to materialize has received a lot of attention in the database community. Traditionally, the problem has been studied using the *view-selection approach*; both theoretical analysis and practical studies have been conducted [2, 3, 24-27, 31, 36, 39-41, 43, 44, 46, 47, 51, 53, 55]. An alternative to view selection is a research direction that studies how cached answers to past queries can facilitate answering current database queries [15, 16, 37].

Research in traditional view selection has been conducted under a limiting assumption: The views that are considered for materialization are assumed to be given in the problem input. However, in general, to truly optimize query performance, it may be necessary to come up with completely new view definitions, in other words, to *invent new views*. Papers [8-13] have introduced a unique perspective of inventing new views to optimize query performance; they also provide a theoretical analysis of the process of designing optimal views for queries in a simple (conjunctive) format under the storage-limit constraint.

### **Outline of the approach in the current stage of the project**

Past work described in papers [8-13] is going to be a foundation of the current stage of this project. The goal is to develop theoretically rigorous yet practically applicable techniques, which are needed to design views that would allow optimal or near-optimal speedup of frequent and important queries on large databases. To achieve this goal via the specific objectives described above, we plan the following activities for the research group:

1. Use theoretical analysis of queries and views to discover and describe structural characteristics of optimal views for common and important problem setups. Begin by analyzing the problem under the simplest assumptions (conjunctive queries, views, and rewritings, and the storage-limit constraint); then gradually progress toward more realistic assumptions, including popular queries with aggregation.
2. Develop algorithms that produce optimal views, in the expected sense. In developing the algorithms, use the characterizations of optimal views and the methods for estimating view sizes that were obtained at the earlier stages of the project (see 1 above). Test the algorithms, by generating queries of various types and by comparing, on databases with varying statistics, the performance gains obtained by using the optimal views predicted analytically, with the performance gains obtained by using the outputs of the algorithms. Analyze and characterize the differences between the optimal views predicted analytically and the views produced by the algorithms. Project the results of the experiments to more complex query types.
3. Develop efficient algorithms and heuristics to approximate optimal views obtained using the techniques described in step 2 above. Test the algorithms, by generating queries of various types and by comparing, on databases with varying statistics, the performance gains obtained by using the optimal views predicted analytically and obtained using the techniques described in step 2, with the performance gains obtained by using the outputs of the efficient approximation algorithms. Analyze and characterize the differences between the views produced using these methods. Project the results of the experiments to more complex query types.

We plan to use our PostgreSQL-based prototype of a database system to implement and rigorously test all outcomes of the project. We are confident we will achieve our goals, by producing an effective methodology for improving performance of frequent and important queries on large databases, and by thus making the first step in developing self-organizing database systems.