Overview of Data Exploration Techniques

Stratos Idreos
Harvard University
stratos@seas.harvard.edu

Olga Papaemmanouil
Brandeis University
olga@cs.brandeis.edu

Surajit Chaudhuri
Microsoft Research
surajitc@microsoft.com

ABSTRACT

Data exploration is about efficiently extracting knowledge from data even if we do not know exactly what we are looking for. In this tutorial, we survey recent developments in the emerging area of database systems tailored for data exploration. We discuss new ideas on how to store and access data as well as new ideas on how to interact with a data system to enable users and applications to quickly figure out which data parts are of interest. In addition, we discuss how to exploit lessons-learned from past research, the new challenges data exploration crafts, emerging applications and future research directions.

1. INTRODUCTION

Assumptions in Traditional Systems. Traditional data management systems assume that when users pose a query a) they have good knowledge of the schema, meaning and contents of the database and b) they are certain that this particular query is the one they wanted to pose. In short, we assume that users know what they are looking for. In response, the system always tries to produce correct and complete results.

Traditional DBMSs are designed for static scenarios with numerous assumptions about the workload. For example, state-of-the-art systems assume that there will be a tuning phase where a database administrator tunes the system for the expected workload. This assumes that we know the workload, we know that it will be stable and we have enough idle time and resources to devote to tuning.

Modern Exploration-driven Applications. The above assumptions were valid for the static applications of the past and they are still valid for numerous applications today. However, as we create and collect increasing amount of data, we are building more dynamic data-driven applications that do not always have the same requirements that database systems have tried to address during the past five decades. Indeed, managing an employee or an inventory database is a drastically different setting than looking for interesting patterns over a scientific database.

Consider an astronomer looking for interesting parts in a continuous stream of data (possibly several TBs per day): they do not know what they are looking for, they only wish to find interesting patterns; they will know that something is interesting only after they find it. In this setting, there are no clear indications about how to tune a database system or how the astronomer should formulate their queries. Typically, an exploration session will include several queries where the results of each query trigger the formulation of the next one. This data exploration paradigm is the key ingredient for a number of discovery-oriented applications, e.g., in the medical domain, genomics and financial analysis.

Database Systems for Data Exploration. Such novel requirements of modern exploration driven interfaces have led to rethinking of database systems across the whole stack, from storage to user interaction. Visualization tools for data exploration (e.g., [38, 49, 66]) are receiving growing interest while new exploration interfaces emerged (e.g., [18, 32, 45, 57]) aiming to facilitate the user's interactions with the underlying database. In parallel, numerous novel optimizations have been proposed for offering interactive exploration times (e.g., [6, 36, 37]) while the database architecture has been re-examined to match the characteristics of the new exploration workloads (e.g., [8, 27, 28, 39]). Together, these pieces of work contribute towards providing data exploration capabilities that enable users to extract knowledge out of data with ease and efficiently.

Tutorial Outline. This tutorial gives a comprehensive introduction to the topic of data exploration, discussing state-of-the-art in the industry and in the academic world. Specifically, it includes the following sections.

1. Introduction: We start with an introduction of the concept of data exploration and an overview of the new challenges presented in the era of "Big Data" which make data exploration a first class citizen for query processing techniques. In this part, we also discuss the support available in today's products and services for data exploration techniques and what is still missing.

2. User Interaction: We take an in-depth look the advanced visualization tools and alternative exploration interfaces for big data exploration tasks. We further divide this last topic into three sub-categories: a) systems that assist SQL query formulation, b) systems that automate the data exploration process by identifying and presenting relevant data items and c) novel query interfaces such as keyword search queries over databases and gestural queries.

SIGMOD'15, May 31–June 4, 2015, Melbourne, Victoria, Australia.

http://dx.doi.org/10.1145/2723372.2731084
This work is partially supported by NSF grants IIS-1253196 and IIS-1452595.

3. Middleware: The second part of the tutorial discusses research that aims at improving the performance of data exploration by building various optimizations on top of the database engine. This category includes research that aims to offer interactive query latencies by exploiting data pre-fetching and background execution of likely similar exploratory queries as well as research on approximate query results.

4. Database Engine: The third part of our presentation includes work that aims to rethink how the architecture of database systems can be redesigned to aid data exploration tasks. We will discuss work on adaptive storage layouts, adaptive data loading and adaptive indexing as well as work that aims to provide efficient support for interactive interfaces.

5. Future Work: Finally, we discuss forward looking ideas on the topic of data exploration, discussing open problems and providing possible directions for future work.

2. FACETS OF DATA EXPLORATION

As explained above, an understanding of data exploration has three key facets. Our presentation follows a top-down approach, starting from user interaction layer, proceeding with optimizations on top the database engine and concluding with research that reexamines the architecture of database management systems. Table 1 shows the clustering of the papers we discuss.

2.1 User Interaction

The tutorial will start by introducing research at the user interface layer where the goal is to assist users (often not database experts) to explore big data sets. Here, a large body of research has recently focused on supporting exploratory tasks by providing alternative interfaces for interacting with the underlying database.

Query Result Visualization. We first highlight the role of visualization for data analytics [1, 38]. We introduce visualization tools that assist users in navigating the underlying data structures [61, 62]. We also cover tools that incorporate new types of interactions such as collaborative annotations and searches [48] as well as recommendations of visualizations [40]. We then proceed to describe optimization techniques that aim to support large-scale visual analytics. These approaches include query result reduction allowing for interactive visualization [11], rapid order preserving sampling [12] and search for automatically identifying interesting data visualizations [49]. We describe these optimizations as well as vision work that proposes a novel declarative visualization language [66] to bridge the gap between traditional database optimizations and visualization.

Exploration Interfaces. In this part, we discuss new exploration interfaces that help users navigate the underlying data space. We start with systems that automate the data exploration process by discovering data objects [14, 18, 20] that are relevant to the user. These systems rely on user modeling approaches, such as online classification based on relevance-feedback [18] and models built by variations of facet search techniques [20]. We compare these modeling solutions and highlight their advantages and drawbacks.

We proceed with solutions aiming to assist users formulate their exploratory queries [4, 67]. We cover systems designed for users who are not aware of the exact query predicates but who are often aware of data items relevant to their exploration task or tuples that should be present in their query output [64]. For example, in [13] the authors infer join queries via labeling relevant objects, while in [58] and [51] the focus is on discovering queries given example output tuples. We also cover recent work on query learning based on example tuples [3] as well as solutions for tuning imprecise queries, where the relevance of query predicates is uncertain to the user [52]. Finally, we cover techniques for recommending SQL queries [21] as well as systems that identify the "best" segmentations of the data space to propose to the user [57].

This part of the tutorial concludes with a discussion of a series of novel query interfaces proposed recently. Specifically, we describe dbTouch [32, 44] and GestureDB [45, 47], two user-guided visual tools for specifying relational queries and for processing data directly in an exploratory way.

2.2 Middleware

The second part of the tutorial includes work that focuses on improving the efficiency of data exploration tasks but without changing the underlying data engine. All techniques covered in this section are implemented in the middleware between the user interaction layer and the database engine. This brings flexibility in applying such ideas across several systems, effectively improving the exploratory properties of existing systems without requiring changes in the underlying architecture.

Data Prefetching. Exploratory tasks can be computationally heavy, as users often execute long sequences of unoptimized queries. The tutorial covers research that aims to reduce the overall exploration time through result prefetching techniques. Faster exploration times are achieved by caching data sets which are likely to be used by a user's follow up exploratory query and the main challenge is identifying the data set with the highest utility.

Data prefetching has been studied within the context of data exploration for a number of query types such as multidimensional windows [36], data cubes [37, 55, 54] and spatial queries [63]. In this part of the tutorial we discuss these types of exploration queries and present alternative techniques for identifying promising data sets for pre-fetching, such as background execution of similar speculative queries [36, 37] as well as indexing and searching past users' exploration trajectories [63].

Finally, we discuss the interplay between result diversification [41] and data caching. While returning diverse but relevant data items can assist users in quickly navigating the data space, it also adds significant computational overhead. We discuss techniques that explore the trade-off between introducing new results and re-using cached ones [41] as well as optimization methods for diversifying query results [65].

Query Approximation. An alternative approach for improving the response time of exploratory queries is to present approximate results. We first discuss online processing techniques [25] and the related CONTROL project [24]. These techniques offer approximate answers and their goal is to allow users to get a quick sense of whether a particular query reveals anything interesting about the data.

We then proceed with solutions that process queries on sampled data sets to provide fast query response times. We discuss the trade-off between results accuracy and query performance and we present architectures that allow for query execution over subsets of data. In such a setting, meeting

User Interaction	**Data Visualization** [38]	Visual Optimizations [11, 12, 49, 66]	Visualization Tools [40, 48, 61, 62]	
	Exploration Interfaces [14]	Automatic Exploration [18, 20]	Assisted Query Formulation [3, 4, 13, 21, 52, 57, 58, 64, 51]	Novel Query Interfaces [32, 44, 45, 47]
Middleware	**Interactive Performance Optimizations**	Data Prefetching [36, 37, 41, 63]	Query Approximation [16, 5, 6, 7, 24, 25]	
Database Layer [27, 39]	**Indexes**	Adaptive Indexing [26, 29, 30, 31, 33, 22, 23, 50]	Time Series [68]	Flexible Engines [17, 42, 43, 34]
	Data Storage	Adaptive Loading [28, 8, 2, 15]	Adaptive Storage [9, 19]	Sampling [59, 60, 35]

Table 1: Clustering of current work on data management research for data exploration. (Some papers may span multiple clusters; we discuss this in detail in the tutorial content but for simplicity we list them under their primary area only in this table.)

user-defined error bounds is necessary to ensure reliable exploration results. The tutorial covers sampling techniques for controlling the quality of the query results while bounding their execution time [5, 6, 7, 59, 60].

2.3 Database Layer

The third research cluster includes work that aims at rethinking database architectures at their core [27]. Work in this area reconsiders the fundamental methods to store and access data to match exploration patterns. We organize this work into four areas: adaptive indexing, adaptive loading, adaptive storage and sampling based architectures.

Adaptive Indexing. In a data exploration scenario we are searching for interesting data patterns without knowledge of what we are looking for. Yet traditional database systems rely heavily on tuning actions and accurate workload knowledge to achieve good performance. One of the most critical tuning actions is that of choosing the proper set of indexes. Making strict a priori choices means that a system is not well prepared for an exploratory scenario where users may focus on arbitrary data parts at different times. Research on adaptive indexing introduces the idea of creating indexes incrementally and adaptively during query processing based on the columns, tables and value ranges that queries request. Indexes are built gradually; as more queries arrive indexes are continuously fine-tuned [26]. Adaptive indexing has been studied to improve selections in column-stores [29], and has been shown to work in late materialization architectures [31], to allow for incremental and partial projections [31], to be robust in workload changes [23], to absorb updates efficiently and adaptively [30] and to enable multi-query processing via concurrency control [22]. In addition, the basic algorithms have been studied in depth in the face of trade-offs such as adaptation speed and initialization costs [33, 56] as well as optimized for modern hardware [50, 10]. In addition, adaptive indexing has been studied for supporting exploration in time-series processing [68] and in Hadoop [53].

Adaptive Loading. During data exploration not all data is needed. Adaptive loading exploits this fact and introduces the notion that users can start querying a database system (with efficient response times) even before all data is loaded or even leaving some parts of the data unloaded, effectively enabling efficient raw data access [28, 8, 2, 15].

Adaptive Storage. The way we store data defines the best possible ways to access it. There is no perfect storage layout; instead there is a perfect layout for each individual data access pattern. Modern systems rely on static layouts and build the whole architecture around a single layout. In a data exploration scenario we cannot a priori decide what is a good layout as we do not know the exact query patterns up front, leading to sub-optimal performance for traditional static systems. In this part of the tutorial, we discuss recent work that aims at removing this problem through adaptive storage [19, 9].

Flexible Architectures. Furthermore, there has been a significant push towards flexible database architectures where we can tune the architecture to the task at hand, for example by having a declarative interface for the data layouts [17] or for the whole engine [42, 43]. In another vision, organic databases are proposed to continuously match incoming data and queries [34].

Architectures Tailored for Approximate Processing. Finally, approximate processing is an important tool to support data exploration. In addition to the ideas discussed for sampling in previous sections, another line of work aims to push sampling inside the core of the engine, creating an architecture where storage and access patterns are tailored to support sampling-based query processing efficiently [59, 60]. This allows for efficient access and updates of sampled data sets. More recently, DICE combines sampling with speculative execution in the same engine [35]. Approximation and sampling has also been coined as a key approach to support interactive visual analytics at the core of a DB engine [32]. Of course, approximate processing is a rich topic in DB literature [16, 5]; Exploiting this knowledge to design new engines is one of the open challenges.

2.4 Open Problems and Challenges

The final part of the tutorial presents open problems. Some of these topics were introduced by recent vision papers [46, 39, 14]. The overall vision is to achieve data navigation systems that automatically steer users towards interesting data. Data system architectures should inherently support exploration with storage and access patterns being fully driven by the exploration paths taken by the users. A system should be able to provide answers instantly even if they are not complete, but it should also be able to eventually lead users towards interesting data patterns. Users should be able to interact with the database at various levels, ranging from manual steering using a declarative "steering" language to automatic steering with minimum query formulation expectations from the user. Some of the ideas include architectures that inherently support sampling at the lower level (e.g., inside operators) as well as opportunistically answering similar queries depending on where data resides (e.g., data in L1 cache is cheap to show to the user even if it is not exactly what the user requested). In addition, work on user profiles and interaction histories can play a crucial role in optimizing exploration tasks.

Finally, we summarize the progress made towards the directions introduced by these papers and highlight the po-

tential challenges both at the user interaction layer and the database architecture layer. At the user interaction layer we still lack declarative "exploration" languages to present and reason about popular navigational idioms. Such languages could facilitate custom optimizations, such as user-driven prefetching, reusing past or in-progress query results and customizing visualization tools. Other future directions include processing past user interaction histories to predict exploration trajectories and identify interesting exploration patterns. Similarly, at the database system layer there are numerous opportunities to reconsider fundamental assumptions about data and storage patterns and how they can be driven dynamically by high level requests.

Finally, we discuss the importance of interconnecting research from both the user interaction and the database architecture layers in order to provide a complete stack of "exploration-ready" database systems.

3. BIOGRAPHIES

Stratos Idreos is an assistant professor of Computer Science at Harvard University where he leads DASlab, the Data Systems Laboratory@Harvard SEAS. Stratos works on data systems architectures with emphasis on designing systems for big data exploration. For his doctoral work on Database Cracking, Stratos won the 2011 ACM SIGMOD Jim Gray Doctoral Dissertation award and the 2011 ERCIM Cor Baayen award as <most promising European young researcher in computer science and applied mathematics> from the European Research Council on Informatics and Mathematics. In 2010 he was awarded the IBM zEnterpise System Recognition Award by IBM Research, and in 2011 he won the VLDB Challenges and Visions best paper award. In 2015 he received an NSF CAREER award and was awarded the 2015 IEEE TCDE Early Career Award from the IEEE Technical Committee on Data Engineering.

Olga Papaemmanouil is an assistant professor of Computer Science at Brandeis University since 2009. She received her undergraduate degree from the University of Patras, Greece and completed her Ph.D at Brown University. Her research interests are in data management and distributed systems with a recent focus on performance management for cloud databases and interactive data exploration. She is the recipient of an NSF CAREER Award (2013) and a Paris Kanellakis Fellow (2002).

Surajit Chaudhuri is a Distinguished Scientist at Microsoft Research and leads the Data Management, Exploration and Mining group. In addition, as a Deputy Managing Director of Microsoft Research Lab at Redmond, he also has oversight of Distributed Systems, Networking, Security, Programming languages and Software Engineering groups. He serves on the Senior Leadership Team of Microsoft's Cloud and Enterprises division. His current areas of interest are enterprise data analytics, data discovery, self-manageability and cloud database services. Working with his colleagues in Microsoft Research, he helped incorporate the Index Tuning Wizard (and subsequently Database Engine Tuning Advisor) and data cleaning technology into Microsoft SQL Server. Surajit is an ACM Fellow, a recipient of the ACM SIGMOD Edgar F. Codd Innovations Award, ACM SIGMOD Contributions Award, a VLDB 10 year Best Paper Award, and an IEEE Data Engineering Influential Paper Award. Surajit received his Ph.D. from Stanford University in 1992.

4. REFERENCES

[1] Magic quadrant for business intelligence and analytics platforms. *Gartner Group*, 2015.

[2] A. Abouzied, D. J. Abadi, and A. Silberschatz. Invisible loading: access-driven data transfer from raw files into database systems. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, pages 1–10, 2013.

[3] A. Abouzied, D. Angluin, C. H. Papadimitriou, J. M. Hellerstein, and A. Silberschatz. Learning and verifying quantified boolean queries by example. In *Proceedings of the International Conference on Principles of Database Systems (PODS)*, 2013.

[4] A. Abouzied, J. M. Hellerstein, and A. Silberschatz. Playful query specification with dataplay. *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 5(12):1938–1941, 2012.

[5] S. Acharya, P. B. Gibbons, V. Poosala, and S. Ramaswamy. The Aqua Approximate Query Answering System. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 574–576, 1999.

[6] S. Agarwal, H. Milner, A. Kleiner, A. Talwalkar, M. Jordan, S. Madden, B. Mozafari, and I. Stoica. Knowing when you're wrong: Building fast and reliable approximate query processing systems. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 2014.

[7] S. Agarwal, B. Mozafari, A. Panda, H. Milner, S. Madden, and I. Stoica. BlinkDB: Queries with Bounded Errors and Bounded Response Times on Very Large Data. In *EuroSys*, 2013.

[8] I. Alagiannis, R. Borovica, M. Branco, S. Idreos, and A. Ailamaki. Nodb: efficient query execution on raw data files. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 241–252, 2012.

[9] I. Alagiannis, S. Idreos, and A. Ailamaki. H2O: a hands-free adaptive store. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 1103–1114, 2014.

[10] V. Alvarez, F. M. Schuhknecht, J. Dittrich, and S. Richter. Main memory adaptive indexing for multi-core systems. In *Proceedings of the International Workshop on Data Management on New Hardware (DAMON)*, page 3, 2014.

[11] L. Battle, R. Chang, and M. Stonebraker. Dynamic Reduction of Query Result Sets for Interactive Visualization. In *IEEE Workshop on Big Data Visualization*, 2013.

[12] E. Blais, A. Kim, A. Parameswaran, P. Indyk, S. Madden, , and R. Rubinfeld. Rapid Sampling for Visualizations with Ordering Guarantees. In *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 2015.

[13] A. Bonifati, R. Ciucanu, and S. Staworko. Interactive inference of join queries. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, 2014.

[14] U. Cetintemel, M. Cherniack, J. DeBrabant, Y. Diao, K. Dimitriadou, A. Kalinin, O. Papaemmanouil, and S. Zdonik. Query Steering for Interactive Data Exploration. In *Proceedings of the biennial Conference on Innovative Data Systems Research (CIDR)*, 2013.

[15] Y. Cheng and F. Rusu. Parallel in-situ data processing with speculative loading. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 1287–1298, 2014.

[16] G. Cormode, M. N. Garofalakis, P. J. Haas, and C. Jermaine. Synopses for massive data: Samples, histograms, wavelets, sketches. *Foundations and Trends in Databases*, 4(1-3):1–294, 2012.

[17] P. Cudré-Mauroux, E. Wu, and S. Madden. The case for rodentstore: An adaptive, declarative storage system. In *Proceedings of the biennial Conference on Innovative Data Systems Research (CIDR)*, 2009.

[18] K. Dimitriadou, O. Papaemmanouil, and Y. Diao. Explore-by-Example: An Automatic Query Steering Framework for Interactive Data Exploration. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 2014.

[19] J. Dittrich and A. Jindal. Towards a one size fits all database architecture. In *Proceedings of the biennial Conference on Innovative Data Systems Research (CIDR)*, pages 195–198, 2011.

[20] M. Drosou and E. Pitoura. Ymaldb: exploring relational databases via result-driven recommendations. *VLDB J.*, 22(6):849–874, 2013.

[21] J. Fan, G. Li, and L. Zhou. Interactive SQL Query Suggestion: Making Databases User-Friendly. In *Proceedings of the International Conference on Data Endineering (ICDE)*, 2011.

[22] G. Graefe, F. Halim, S. Idreos, H. Kuno, and S. Manegold. Concurrency Control for Adaptive Indexing. *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 5(7):656–667, 2012.

[23] F. Halim, S. Idreos, P. Karras, and R. H. C. Yap. Stochastic Database Cracking: Towards Robust Adaptive Indexing in Main-Memory Column-Stores. *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 5(6):502–513, 2012.

[24] J. M. Hellerstein, R. Avnur, A. Chou, C. Hidber, C. Olston, V. Raman, T. Roth, and P. J. Haas. Interactive data analysis: The control project. *Computer*, 32(8), 1999.

[25] J. M. Hellerstein, P. J. Haas, and H. J. Wang. Online Aggregation. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1997.

[26] S. Idreos. Database Cracking: Towards Auto-tuning Database Kernels. *CWI, PhD Thesis*, 2010.

[27] S. Idreos. *Big Data Exploration*. Taylor and Francis, 2013.

[28] S. Idreos, I. Alagiannis, R. Johnson, and A. Ailamaki. Here are my Data Files. Here are my Queries. Where are my Results? In *Proceedings of the biennial Conference on Innovative Data Systems Research (CIDR)*, 2011.

[29] S. Idreos, M. L. Kersten, and S. Manegold. Database cracking. In *Proceedings of the biennial Conference on Innovative Data Systems Research (CIDR)*, 2007.

[30] S. Idreos, M. L. Kersten, and S. Manegold. Updating a cracked database. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 413–424, 2007.

[31] S. Idreos, M. L. Kersten, and S. Manegold. Self-organizing tuple reconstruction in column stores. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 297–308, 2009.

[32] S. Idreos and E. Liarou. dbtouch: Analytics at your fingertips. In *Proceedings of the biennial Conference on Innovative Data Systems Research (CIDR)*, 2013.

[33] S. Idreos, S. Manegold, H. Kuno, and G. Graefe. Merging What's Cracked, Cracking What's Merged: Adaptive Indexing in Main-Memory Column-Stores. *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 4(9):585–597, 2011.

[34] H. V. Jagadish, A. Nandi, and L. Qian. Organic databases. In *International Workshop Databases in Networked Information Systems*, pages 49–63, 2011.

[35] P. Jayachandran, K. Tunga, N. Kamat, and A. Nandi. Combining user interaction, speculative query execution and sampling in the DICE system. *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 7(13):1697–1700, 2014.

[36] A. Kalinin, U. Cetintemel, and S. Zdonik. Interactive Data Exploration using Semantic Windows. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 2014.

[37] N. Kamat, P. Jayachandran, K. Tunga, and A. Nandi. Distributed Interactive Cube Exploration. In *Proceedings of the International Conference on Data Endineering (ICDE)*, 2014.

[38] D. Keim. Exploring Big Data using Visual Analytics-Keynote. In *Exploratory Search in Databases and the Web. EDBT/ICDT Workshops*, 2014.

[39] M. Kersten, S. Idreos, S. Manegold, and E. Liarou. The Researcher's Guide to the Data Deluge: Querying a Scientific Database in Just a Few Seconds. *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 4(12):1474–1477, 2011.

[40] A. Key, B. Howe, D. Perry, and C. Aragon. VizDeck: Self-Organizing Dashboards for Visual Analytics. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 2012.

[41] H. Khan, M. Sharaf, and A. Albarrak. DivIDE: efficient diversification for interactive data exploration. In *SSDBM*, 2014.

[42] Y. Klonatos, C. Koch, T. Rompf, and H. Chafi. Building efficient query engines in a high-level language. *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 7(10):853–864, 2014.

[43] C. Koch. Abstraction without regret in database systems building: a manifesto. *IEEE Data Eng. Bull.*, 37(1):70–79, 2014.

[44] E. Liarou and S. Idreos. dbTouch in Action: Database kernels for touch-based data exploration. In *Proceedings of the International Conference on Data Endineering (ICDE)*, pages 1262–1265, 2014.

[45] A. Nandi. Querying Without Keyboards. In *Proceedings of the biennial Conference on Innovative Data Systems Research (CIDR)*, 2013.

[46] A. Nandi and H. V. Jagadish. Guided interaction: Rethinking the query-result paradigm. *PVLDB*, 4(12):1466–1469.

[47] A. Nandi, L. Jiang, and M. Mandel. Gestural Query Specification. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, 2014.

[48] P. Neophytou, R. Gheorghiu, R. Hachey, T. Luciani, D. Bao, A. Labrinidis, G. E. Marai, and P. K. Chrysanthis. AstroShelf: Understanding the Universe Through Scalable Navigation of a Galaxy of Annotations (Demo). In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 2012.

[49] A. Parameswaran, N. Polyzotis, and H. Garcia-Molina. SeeDB: Visualizing Database Queries Efficiently. *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 7(4):325–328, 2013.

[50] H. Pirk, E. Petraki, S. Idreos, S. Manegold, and M. L. Kersten. Database cracking: fancy scan, not poor man's sort! In *Proceedings of the International Workshop on Data Management on New Hardware (DAMON)*, 2014.

[51] F. Psallidas, B. Ding, K. Chakrabarti, and S. Chaudhuri. Top-k Spreadsheet-Style Search for Query Discovery. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 2015.

[52] B. Qarabaqi and M. Riedewald. User-driven refinement of imprecise queries. In *Proceedings of the International Conference on Data Endineering (ICDE)*, 2014.

[53] S. Richter, J. Quiané-Ruiz, S. Schuh, and J. Dittrich. Towards zero-overhead static and adaptive indexing in hadoop. *VLDB J.*, 23(3):469–494, 2014.

[54] S. Sarawagi, R. Agrawwal, and N. Megiddo. Discovery-driven Exploration of OLAP Data Cubes. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, 2008.

[55] S. Sarawagi and G. Sathe. i3: Intelligent, Interactive Investigation of OLAP Data Cubes. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 2000.

[56] F. M. Schuhknecht, A. Jindal, and J. Dittrich. The uncracked pieces in database cracking. *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 7(2):97–108, 2013.

[57] T. Sellam and M. Kersten. Meet Charles, big data query advisor. In *Proceedings of the biennial Conference on Innovative Data Systems Research (CIDR)*, 2013.

[58] Y. Shen, K. Chakrabarti, S. Chaudhuri, B. Ding, and L. Novik. Discovering Queries based on Example Tuples. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 2014.

[59] L. Sidirourgos, M. L. Kersten, and P. A. Boncz. SciBORQ: Scientific data management with Bounds On Runtime and Quality. In *Proceedings of the biennial Conference on Innovative Data Systems Research (CIDR)*, 2011.

[60] L. Sidirourgos, M. L. Kersten, and P. A. Boncz. Scientific discovery through weighted sampling. In *BigData Conference*, 2013.

[61] C. Stolte, D. Tang, and P. Hanrahan. Polaris: A System for Query, Analysis and Visualization of Multi-dimensional Relational Databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1), 2002.

[62] M. Stonebraker and J. Kalash. TIMBER: A Sophisticated Relation Browser. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, 1982.

[63] F. Tauheed, T. Heinis, F. Schurmann, H. Markram, and A. Ailamaki. SCOUT: Prefetching for Latent Structure Following Queries. *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 5(11):1531–1542, 2012.

[64] Q. T. Tran, C.-Y. Chan, and S. Parthasarathy. Query by output. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 2009.

[65] M. R. Vieira, H. L. Razente, M. C. N. Barioni, M. Hadjieleftheriou, D. Srivastava, C. Traina, and V. J. Tsotras. On query result diversification. In *Proceedings of the International Conference on Data Endineering (ICDE)*, 2011.

[66] E. Wu, L. Battle, and S. Madden. The Case for Data Visualization Management Systems. *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 7(10):903–906, 2014.

[67] J. X. Yu, L. Qin, and L. Chang. Keyword search in relational databases: A survey. *IEEE Data Eng. Bull.*, 33(1):67–78, 2010.

[68] K. Zoumpatianos, S. Idreos, and T. Palpanas. Indexing for interactive exploration of big data series. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 1555–1566, 2014.