

Extending SQL for Decision Support Applications

Haixun Wang

IBM T. J. Watson Research Center
Hawthorne, NY 10532
haixun@us.ibm.com

Carlo Zaniolo

Department of Computer Science, UCLA
Los Angeles, CA 90095
zaniolo@cs.ucla.edu

Extended Abstract

The challenge of extending database systems for decision support applications has been the topic of much recent research—a very incomplete list of previous work includes [11, 8, 12, 4, 10, 5]. Yet, there is no generally accepted solution for the problem, which remains a critical one, since the inability of current DBMSs to support data mining applications is well-tested and clearly documented [12].

Our research approach in addressing this difficult problem is motivated by the observation that aggregate functions provide the linchpin for most decision support computations; moreover inductive discovery from large data sets can be viewed as the process of aggregating low level data into statistical summaries of semantic significance. Therefore, the ATLaS system designed at UCLA [2] allows end-users to define new powerful aggregate functions by writing them in SQL. The same mechanism can be used to define new table functions in ATLaS, whose name stands for Aggregate & Table Language and System. ATLaS is the successor of the AXL system described in [15].

These SQL-based native extension mechanisms turn ATLaS into a powerful and flexible system for advanced data-intensive applications, including applications from many domains that are not supported well by current Object-Relational database systems, which still suffer from limited extensibility. In fact, the only extensibility mechanism now provided by Object-Relational systems relies on nonnative datablades—i.e., on external functions defined in a procedural language and imported into SQL.

ATLaS is very effective at expressing decision support tasks: we demonstrate this property by the efficient implementation of several functions, such as rollups, datacubes, classifiers, and frequent item sets for association rules [2]. The performance of these functions expressed in ATLaS is typically within 30% of the performance of the same algorithms coded in C/C++. To achieve this level of performance, ATLaS supports various optimization techniques, and the ability of manipulating in-memory tables in SQL. In fact, attributes of reference type in such tables allow the efficient support of data structures, such as tries, that are instrumental in implementing data mining algorithms, such as Apriori [1, 7].

The stream-oriented computation model used by ATLaS contrasts with the computation model based on ‘blocking’ semantics that is normally used for aggregates in current database systems. Thus, online aggregates [9], time-series queries [13], sliding-window aggregates, approximate aggregates, and continuous queries [3] are naturally supported in ATLaS. Furthermore, important properties of an ATLaS program, such as blocking behavior and monotonicity, can be easily inferred from the syntactic structure of the program [14].

References

- [1] R. Agrawal, R. Srikant. “Fast Algorithms for Mining Association Rules”. VLDB 1994: 487-499.
- [2] ATLaS download page: [http://www.cs.ucla.edu /classes/spring02/cs240b/Readme.htm](http://www.cs.ucla.edu/classes/spring02/cs240b/Readme.htm)
- [3] Shivnath Babu and Jennifer Widom. Continuous Queries over Data Streams, SIGMOD Record, Vol. 30 No. 3, pp. 109-120 (September 2001).
- [4] F. Giannotti, G. Manco, D. Pedreschi, F. Turini: Experiences with a Logic-Based Knowledge Discovery Support Environment. AI*IA 1999: 202-213.
- [5] Frédéric Gingras, Laks V. S. Lakshmanan: nD-SQL: A Multi-Dimensional Language for Interoperability and OLAP. VLDB 1998: 134-145
- [6] R. Agrawal, R. Srikant, “Fast Algorithms for Mining Association Rules,” VLDB 1994: 487-499.
- [7] J. Han and M. Kamber: Data Mining, Concepts and Techniques: Morgan Kaufman, 2001
- [8] J. Han, Y. Fu, W. Wang, K. Koperski, and O. R. Zaiane. DMQL:A Data Mining Query Language for Relational Databases. In Proc. 1996 SIGMOD’96 Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD’96), pp. 27-33, Montreal, Canada, June 1996.
- [9] J. M. Hellerstein, P. J. Haas, H. J. Wang. “Online Aggregation”. SIGMOD Conference 1997: 171-182.
- [10] T. Imielinski and A. Virmani. MSQL: a query language for database mining. Data Mining and Knowledge Discovery, 3:373-408, 1999.
- [11] R. Meo, G. Psaila, S. Ceri. A New SQL-like Operator for Mining Association Rules. In Proc. VLDB96, 1996 Int. Conf. Very Large Data Bases, Bombay, India, pp. 122-133, Sept. 1996.
- [12] S. Sarawagi, S. Thomas, R. Agrawal, “Integrating Association Rule Mining with Relational Database Systems: Alternatives and Implications.”SIGMOD Conference 1998: 343-354.
- [13] Reza Sadri, Carlo Zaniolo, Amir M. Zarkesh, Jafar Adibi: A Sequential Pattern Query Language for Supporting Instant Data Mining for e-Services, VLDB 2001: 653-656.
- [14] H. Wang and C. Zaniolo, User-Defined Aggregates in Database Languages. DBPL 1999: 43-60, In Lecture Notes in Computer Science 1949, Springer 2000.
- [15] H. Wang and C. Zaniolo: Using SQL to Build New Aggregates and Extenders for Object-Relational Systems. VLDB 2000: 166-175.