Logical Data Expiration

(Invited Talk)

David Toman

Department of Computer Science, University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
E-mail: david@uwaterloo.ca

Abstract

Data expiration is an essential component of data warehousing solutions: whenever large amounts of data are repeatedly collected over a period of time, it is essential to have a clear approach to identifying parts of the data no-longer needed and a policy that allows disposing and/or archiving these parts of the data. Such policies are necessary even if adding storage to accommodate an ever- growing collection of data were possible, since the growing amount of data needs to be examined during querying and turn leads to deterioration of query performance over time.

The approaches to data expiration range from ad-hoc administrative policies or regulations to sophisticated data analysis-based techniques. The approaches have, however, one thing in common: intuitively, they try to identify the parts of the data collection that are not needed in the future. The key to deciding if a piece of information will be needed in the future lies in identifying what queries can be asked over the collection of data and how the collection can evolve from its current state. The various techniques proposed in the literature differ in the way they identify no longer needed parts of data.

The talk formalizes the notion of data expiration in terms of how the data is used to answer queries. We survey existing approaches to the problem in a unified framework and discusses their features and limits and the limits of data expiration based techniques in general. Particular focus of the talk is on comparing the *performance* of various data expiration methods.

Outline

1. What is Data Expiration?
   - flow of time and aging of data
   - updates and database histories
2. Ad-hoc Expiration Techniques
3. Query-driven Expiration
   - for temporal logic queries
   - for first-order logic queries
4. How do we Compare Performance of Expiration Methods?
   - query expressiveness vs. storage requirements
   - known upper bounds
   - information-theoretical lower bounds
5. Temporal Integrity Constraints
6. Extensions
   - Retroactive Updates
   - Real-time Expiration
7. Implementation Issues
   - immediate vs. delayed expiration
   - garbage collection
   - expiration of interval-encoded data
8. Conclusion, Open Issues, and Future Directions

Full version of this talk will appear in *Logics for Emerging Applications of Databases*, Chomicki, van der Meyden, and Saake, eds. Springer Edited Monograph, 2003.

Acknowledgments. The author gratefully acknowledges support by the Natural Sciences and Engineering Research Council of Canada. Part of this work was done while visiting BRICS[1] at the University of Aarhus.

References

[1] S. Abiteboul, L. Herr, and J. V. den Bussche. Temporal versus first-order logic to query temporal databases. In *Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 49–57, 1996.

[2] J. Chomicki. History-less checking of dynamic integrity constraints. In *Proceedings of the 8th IEEE Conference on Data Engineering*, pages 557–564, 1992.

[3] J. Chomicki. Real-Time Integrity Constraints. In *Proceedings of the 11th Symposium on Principles of Database Systems, PODS 1992*, pages 274–281, 1992.

[4] J. Chomicki. Efficient checking of temporal integrity constraints using bounded history encoding. *TODS*, 20(2):149–186, 1995.

[5] J. Chomicki and D. Niwiński. On the feasibility of checking temporal integrity constraints. *Journal of Computer and System Sciences*, 51(3):523–535, 1995.

[6] J. Chomicki and G. Saake, editors. *Logics for Databases and Information Systems*. Kluwer, 1998.

[7] J. Chomicki and D. Toman. Implementing Temporal Integrity Constraints Using an Active DBMS. *IEEE Transactions on Knowledge and Data Engineering*, 7(4):566–582, 1995.

[8] J. Chomicki and D. Toman. Temporal Logic in Information Systems. In Chomicki and Saake [6], pages 31–70.

[9] L. S. Colby, T. Griffin, L. Libkin, I. S. Mumick, and H. Trickey. Algorithms for deferred view maintenance. In *ACM SIGMOD International Conference on Management of Data*, pages 469–480. ACM Press, 1996.

[10] H. Garcia-Molina, W. Labio, and J. Yang. Expiring data in a warehouse. In A. Gupta, O. Shmueli, and J. Widom, editors, *24rd International Conference on Very Large Data Bases, VLDB'98*, pages 500–511. Morgan Kaufmann, 1998.

[11] M. Gertz and U. Lipeck. Deriving optimized integrity monitoring triggers from dynamic integrity constraints. *Data and Knowledge Engineering*, 20(2):163–193, 1996.

[1] Basic Research in Computer Science (`www.brics.dk`) funded by the Danish National Research Foundation.

[12] A. Gupta and I. S. Mumick. Maintenance of materialized views: Problems, techniques, and applications. *Data Engineering Bulletin*, 18(2):3–18, 1995.

[13] N. Huyn. Multiple-view self-maintenance in data warehousing environments. In *VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases*, pages 26–35, 1997.

[14] C. S. Jensen. Vacuuming. In Snodgrass [18], chapter 19, pages 451–462.

[15] N. D. Jones, C. K. Gomard, and P. Sestoft. *Partial Evaluation and Automatic Program Generation*. Prentice-Hall International, 1993. Also available online at `http://www.dina.kvl.dk/ ~sestoft/pebook/pebook.html`.

[16] U. Lipeck and G. Saake. Monitoring Dynamic Integrity Constraints Based on Temporal Logic. *Information Systems*, 12:255–269, 1987.

[17] D. Quass, A. Gupta, I. S. Mumick, and J. Widom. Making views self-maintainable for data warehousing. In *Proceedings of the Fourth International Conference on Parallel and Distributed Information Systems*, pages 158–169, 1996.

[18] R. T. Snodgrass, editor. *The TSQL2 Temporal Query Language*. Kluwer, 1995.

[19] D. Toman. Point vs. Interval-based Query Languages for Temporal Databases. In *Proceedings of 15th ACM Symposium on Principles of Database Systems*, pages 58–67, 1996.

[20] D. Toman. Point-Based Temporal Extensions of SQL and Their Efficient Implementation. In O. Etzion, S. Jajodia, and S. Sripada, editors, *Temporal Databases: Research and Practice*, volume 1399, pages 211–237. Springer-Verlag, 1998.

[21] D. Toman. Expiration of historical databases (extended abstract). In *Proceedings of 8th International Symposium on Temporal Representation and Reasoning, TIME'01*, pages 128–135, 2001.

[22] D. Toman and D. Niwinski. First-order queries over temporal databases inexpressible in temporal logic. In *proceedings of Advances in Database Technology, EDBT 1996*, volume 1057, pages 307–324. Springer, 1996.

[23] F. W. Tompa and J. A. Blakeley. Maintaining materialized views without accessing base data. *Information Systems*, 13(4):393–406, 1988.

[24] J. Yang and J. Widom. Maintaining temporal views over non-temporal information sources for data warehousing. In *Proceedings of Advances in Database Technology, EDBT 1998*, pages 389–403, 1998.

[25] J. Yang and J. Widom. Temporal view self-maintenance. In *Proceedings of Advances in Database Technology, EDBT 2000*, pages 395–412, 2000.