

Database Theory in Action: Making Provenance and Probabilistic Database Theory Work in Practice: Invited Talk

Silviu Maniu   

Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, Grenoble, France
CNRS@CREATE LTD, Singapore

Pierre Senellart   

DI ENS, ENS, CNRS, PSL University, Inria, Paris, France
Institut Universitaire de France
CNRS@CREATE LTD, Singapore
IPAL, CNRS, Singapore

Abstract

There has been a rich literature in database theory on how to model and manage the provenance of data (for instance using the semiring framework) and its uncertainty (in particular via probabilistic databases). In this article, we explain how these results have been used as the basis for practical implementations, notably in the ProvsQL system, and how these implementations need to be adapted for the efficient management of provenance and probability for real-world data.

2012 ACM Subject Classification Theory of computation → Data provenance; Theory of computation → Incomplete, inconsistent, and uncertain databases; Information systems → Database management system engines

Keywords and phrases provenance, probabilistic data, ProvsQL

Digital Object Identifier 10.4230/LIPIcs.ICDT.2025.35

Category Database Theory in Action, Invited Talk

Supplementary Material *Software (Source Code)*: <https://github.com/PierreSenellart/provsql>
archived at `swh:1:dir:28237b2e7a78f7ae65b0035d3bf352ce3ddd010b`

Funding This work is part of the program DesCartes and is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) program.

Acknowledgements ProvsQL is a collective effort; we acknowledge the contributions of Belkis Djeflal, Louis Jachiet, Pratik Karmakar, Baptiste Lafosse, Aryak Sen, Albert Ariel Widiatmaja.

1 Introduction

Real-world data is often incomplete, uncertain, and possibly biased. Database theory has long recognized the importance of properly managing incompleteness and uncertainty in data, as well as maintaining *provenance* information about *where* the output of a query comes from and *why* and *how* it was obtained.

Very early in SQL history, NULLs were introduced as a basic way of modeling incompleteness in the relational model [10]. Limitations of NULL values and incompleteness of real-world data led to *conditional tables* [19] (which would be seen in modern terms as a form of *Boolean provenance* [26]). Later on, the explosion of potentially unreliable data as found on the Web motivated the exploration of the notion of *provenance* [8], metadata keeping track of input data throughout a complex process. In particular, two important categories of provenance, namely *where*- and *why*-provenance, were identified [9]. A few years later,



© Silviu Maniu and Pierre Senellart;
licensed under Creative Commons License CC-BY 4.0
28th International Conference on Database Theory (ICDT 2025).

Editors: Sudeepa Roy and Ahmet Kara; Article No. 35; pp. 35:1–35:7
Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

probabilistic databases [12] emerged as a model for managing uncertain data. Connections with provenance (also called *lineage* at the time) were identified in various works and systems such as Trio [2] and a probabilistic version of conditional tables [17]. A clean mathematical framework for provenance (encompassing in particular why-provenance and Trio’s lineage) was put forward by Green, Karvounarakis, and Tannen: *provenance semirings* [16]. Many further works expanded on these seminal ideas, on both provenance (see within [26, 27] for some additional references) and probabilistic databases (see [29, 13]).

Most of these works were anchored in database theory: they were developing the necessary foundations in terms of modeling, complexity, or algorithms. Though some came with prototype implementations, they were often limited in scope or are no longer maintained.

In this article, we leverage our experience in building ProvSQL [28], a PostgreSQL extension that adds support for provenance management and probabilistic query evaluation to a standard database management system, to discuss practical concerns arising when implementing such database theory concepts. We also show that this may result in efficient processing of real-world data, despite the high theoretical complexity of some operations.

In Section 2, we give a high-level overview of the theory of data provenance, and how this has been implemented in the ProvSQL system and extensions thereof. Similarly, in Section 3 we present basic theoretical results about probabilistic databases and explain how we can achieve efficient probabilistic query evaluation in practice, despite high theoretical complexity.

2 Data Provenance: Foundations and Practice

Theoretical Background A notable and fruitful form of provenance framework is that of *provenance semirings* [16]. A semiring is an algebraic structure formed of a set \mathbb{K} with two binary operations (\oplus and \otimes) and two distinguished elements ($\mathbf{0}$ and $\mathbf{1}$), such that: \oplus is commutative and associative; \otimes is associative; \otimes distributes over \oplus ; $\mathbf{0}$ is neutral for \oplus and annihilator for \otimes ; and $\mathbf{1}$ is neutral for \otimes . Many common algebraic structures are semirings, such as:

1. the set \mathbb{N} of natural numbers with the usual addition and multiplication;
2. the *tropical semiring* consisting of $\mathbb{R}^+ \cup \{+\infty\}$ with minimum as \oplus and addition as \otimes ;
3. the set $\mathcal{B}[X]$ of Boolean functions over a finite set X of propositional variables with Boolean \vee and \wedge operations;
4. the set $\mathbb{N}[X]$ of polynomials over a finite set X of variables with natural integer coefficients, with the usual polynomial addition and multiplication.

For the relational model, [16] defines \mathbb{K} -relations for a given \mathbb{K} semiring by associating every tuple of a relation with an element of \mathbb{K} . They show that one can define a semantics for the operators of the positive relational algebra over \mathbb{K} -relations such that:

- semiring homomorphisms *commute* with this provenance semantics;
- the provenance semantics for $\mathcal{B}[X]$ coincides with a *natural semantics for Boolean provenance* where the provenance of a tuple in the query output is a Boolean function that evaluates to true if the query produces this tuple when run over the subdatabase where all tuples whose provenance evaluates to false are removed.

Furthermore, $\mathbb{N}[X]$ is a *universal semiring* for provenance: there exists a unique homomorphism from $\mathbb{N}[X]$ to any semiring K extending a given function from X to K .

Provenance semirings have been extended in various ways, notably by adding a *monus* \ominus operator that captures the semantics of the full relational algebra [15]; and by combining

semiring annotations with aggregation monoid values through a semimodule construction [6] to provide a provenance semantics for aggregate queries.

In ProvSQL The framework of \mathbb{K} -relations (extended to semirings with monus for the whole relational algebra, and to semimodule provenance for aggregate queries) can be readily implemented in a provenance-aware system by adding an extra column to all relations. Before the user query is sent to the planner, it is rewritten to ensure that the provenance semantics is applied to this extra column. ProvSQL exploits the fact that semiring homomorphisms commute with the provenance semantics to store all provenance annotations in the universal semiring, which means provenance computations can be done only once, without having to choose the application semiring at query time. For efficiency, it is very important to store provenance annotations in a compact form. We use *provenance circuits*, as in [14], to avoid duplicating parts of provenance expressions. Unique identifiers are automatically generated for all gates of the circuits, and these identifiers are stored in the extra provenance column. The circuit itself is stored in memory-mapped files accessed by the database management system. As a result of this architecture, provenance computation adds a very reasonable overhead (a small constant factor, in our experiments) to query evaluation.

Extension to graph queries The semiring operators \oplus and \otimes can be thought of as specific instances of the operations required to answer *any* source-to-target query in graphs: the query result is the *aggregation* over all possible paths. Our findings in [24, 25] demonstrate that semiring provenance can effectively handle this. Formally, in a graph $G = (V, E)$, where each edge $e_i \in E$ has a weight $w[e_i]$ – represented as an element of a semiring – the provenance of a source-to-target query is the (potentially infinite) sum over the set of all paths between x and y in G , $P_{xy}(G)$: $\text{prov}_{\mathbb{K}}(G)(x, y) := w[P_{xy}(G)] = \bigoplus_{\pi \in P_{xy}(G)} w[\pi] = \bigoplus_{\pi \in P_{xy}(G)} \bigotimes_{e_i \in \pi} w[e_i]$.

The semantics of the sum varies based on the chosen semiring. In the *tropical semiring*, it corresponds to the shortest distance between nodes x and y in G . In \mathbb{N} , it denotes the number of paths between x and y . In $\mathcal{B}[X]$, it indicates whether y is reachable from x , based on the truth values assigned to the edges.

The choice of provenance semiring influences the computational complexity of provenance algorithms. For semirings where cycles in the graph do not affect the sum (known as 0-closed semirings, such as the tropical semiring) and that define a total order, the provenance can be computed in quasi-linear time using an analogous of Dijkstra’s algorithm. For semirings where the sum is finite, a node elimination algorithm can be used, having a cubic worst-case time complexity. Such algorithms enable efficient practical provenance computation beyond SQL queries.

3 Probabilistic Databases: Foundations of Probabilistic Databases

Theoretical Background Probabilistic databases are compact representations of a probability distribution over a set of *possible databases*. In principle, distributions can be finite, continuous [1], or discrete but infinite [7, 18]. Correlations across data items (tuples in the relational setting) may be disallowed (as in the tuple independent, TID, setting [12]), limited (e.g., block independent databases, BIDs [11]) or fully arbitrary (as in probabilistic c-tables [17]). The main problem studied over probabilistic databases is *probabilistic query evaluation*: computing the (marginal) probability of a tuple in the output of the query; or, when aggregate queries are involved, computing the distribution of data values in the output, or summaries thereof.

35:4 Making Provenance and Probabilistic Database Theory Work in Practice

Unfortunately, even in a very simple relational setting where distributions are finite and there are no correlations across tuples, probabilistic query evaluation is $\#P$ -hard, even for simple conjunctive queries. It becomes tractable when:

- its $\mathcal{B}[X]$ provenance is in some tractable form (read-once, deterministic and decomposable circuits, etc.) [20] – it is easy to see that the probability of a query output is the probability of its $\mathcal{B}[X]$ provenance, when variables in \mathcal{X} are assigned the probabilities of input tuples;
- the query is a *safe* UCQ [13] over TIDs (with a fairly complex algorithm to recognize safety);
- the query is a *safe* CQ without self-joins [11] over BIDs;
- the data has bounded treewidth over BIDs [4], for any MSO query;
- the data and correlations have joint bounded treewidth [3], for any MSO query.

In ProVSQL ProVSQL allows representation of finite probability distributions with arbitrary correlations. Probabilistic query evaluation relies on: first, computing its provenance; second, applying a semiring homomorphism to turn this provenance into a $\mathcal{B}[X]$ -provenance circuit; third, computing the probability of this circuit. This approach, sometimes called the *intensional* approach [29] to probabilistic query evaluation, does not permit exploiting specific algorithms for specific query types (in particular, safe queries). However, it decouples the hard problem of probability computation from the query evaluation problem.

To compute the probability of a Boolean circuit, ProVSQL relies on a combination of techniques:

- if the circuit happens to be read-once, the probability is easily computed;
- if the circuit has low treewidth, we use an algorithm from [5] to extract a deterministic and decomposable circuit, on which the probability is easily computed;
- otherwise, we use an external *knowledge compiler* [21] that turns the circuit into a tractable class.

Knowledge compilers are specialized tools that incorporate many heuristics, similarly to SAT solvers. As the Boolean circuits are not arbitrary, they manage to produce tractable small circuits in many cases. Though ProVSQL cannot guarantee efficiency of probabilistic query evaluation, our tests reveal that our approach works in practice for many queries and scale to databases of several GBs. In addition, we also allow standard Monte-Carlo sampling methods for probabilistic query evaluation, for cases where exact computation is not critical.

Exploiting the structure of data in practice A potential avenue for tractable probabilistic query evaluation is to exploit the shape of data. As previously mentioned, this is the case when the data (and correlations) has low treewidth, i.e., if it is structurally close to a tree.

Requiring a graph – or a graph representation of a database – to have bounded treewidth is a strong assumption. Our experimental results in [23] bring evidence that real-world graphs typically have high treewidth, with the possible exception of graphs representing physical networks, such as roads and power grids. However, practical efficiency is possible via *partial decompositions*. In our ProbTree work [22], we introduce algorithms decomposing only the low-treewidth portion of the graph. This can significantly speed up probability computations. Extending this partial decomposition approach to general provenance computation is an area for future research.

References

- 1 Serge Abiteboul, T.-H. Hubert Chan, Evgeny Kharlamov, Werner Nutt, and Pierre Senellart. Capturing continuous data and answering aggregate queries in probabilistic XML. *ACM Trans. Database Syst.*, 36(4):25:1–25:45, 2011. doi:10.1145/2043652.2043658.
- 2 Parag Agrawal, Omar Benjelloun, Anish Das Sarma, Chris Hayworth, Shubha U. Nabar, Tomoe Sugihara, and Jennifer Widom. Trio: A system for data, uncertainty, and lineage. In Umeshwar Dayal, Kyu-Young Whang, David B. Lomet, Gustavo Alonso, Guy M. Lohman, Martin L. Kersten, Sang Kyun Cha, and Young-Kuk Kim, editors, *Proceedings of the 32nd International Conference on Very Large Data Bases, Seoul, Korea, September 12-15, 2006*, pages 1151–1154. ACM, 2006. URL: <http://dl.acm.org/citation.cfm?id=1164231>.
- 3 Antoine Amarilli. *Leveraging the Structure of Uncertain Data. (Tirer parti de la structure des données incertaines)*. PhD thesis, Télécom ParisTech, France, 2016. URL: <https://tel.archives-ouvertes.fr/tel-01345836>.
- 4 Antoine Amarilli, Pierre Bourhis, and Pierre Senellart. Provenance circuits for trees and treelike instances. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*, volume 9135 of *Lecture Notes in Computer Science*, pages 56–68. Springer, 2015. doi:10.1007/978-3-662-47666-6_5.
- 5 Antoine Amarilli, Florent Capelli, Mikaël Monet, and Pierre Senellart. Connecting knowledge compilation classes and width parameters. *Theory Comput. Syst.*, 64(5):861–914, 2020. URL: <https://doi.org/10.1007/s00224-019-09930-2>, doi:10.1007/S00224-019-09930-2.
- 6 Yael Amsterdamer, Daniel Deutch, and Val Tannen. Provenance for aggregate queries. In Maurizio Lenzerini and Thomas Schwentick, editors, *Proceedings of the 30th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2011, June 12-16, 2011, Athens, Greece*, pages 153–164. ACM, 2011. doi:10.1145/1989284.1989302.
- 7 Michael Benedikt, Evgeny Kharlamov, Dan Olteanu, and Pierre Senellart. Probabilistic XML via Markov chains. *Proc. VLDB Endow.*, 3(1):770–781, 2010. URL: http://www.vldb.org/pvldb/vldb2010/pvldb_vol13/R69.pdf, doi:10.14778/1920841.1920939.
- 8 Peter Buneman, Sanjeev Khanna, and Wang Chiew Tan. Data provenance: Some basic issues. In Sanjiv Kapoor and Sanjiva Prasad, editors, *Foundations of Software Technology and Theoretical Computer Science, 20th Conference, FST TCS 2000 New Delhi, India, December 13-15, 2000, Proceedings*, volume 1974 of *Lecture Notes in Computer Science*, pages 87–93. Springer, 2000. doi:10.1007/3-540-44450-5_6.
- 9 Peter Buneman, Sanjeev Khanna, and Wang Chiew Tan. Why and where: A characterization of data provenance. In Jan Van den Bussche and Victor Vianu, editors, *Database Theory - ICDT 2001, 8th International Conference, London, UK, January 4-6, 2001, Proceedings*, volume 1973 of *Lecture Notes in Computer Science*, pages 316–330. Springer, 2001. doi:10.1007/3-540-44503-X_20.
- 10 E. F. Codd. Understanding relations (installment #7). *FDT Bull. ACM SIGFIDET SIGMOD*, 7(3):23–28, 1975.
- 11 Nilesh N. Dalvi, Christopher Ré, and Dan Suciu. Queries and materialized views on probabilistic databases. *J. Comput. Syst. Sci.*, 77(3):473–490, 2011. URL: <https://doi.org/10.1016/j.jcss.2010.04.006>, doi:10.1016/J.JCSS.2010.04.006.
- 12 Nilesh N. Dalvi and Dan Suciu. Efficient query evaluation on probabilistic databases. *VLDB J.*, 16(4):523–544, 2007. URL: <https://doi.org/10.1007/s00778-006-0004-3>, doi:10.1007/S00778-006-0004-3.
- 13 Nilesh N. Dalvi and Dan Suciu. The dichotomy of probabilistic inference for unions of conjunctive queries. *J. ACM*, 59(6):30:1–30:87, 2012. doi:10.1145/2395116.2395119.
- 14 Daniel Deutch, Tova Milo, Sudeepa Roy, and Val Tannen. Circuits for Datalog provenance. In Nicole Schweikardt, Vassilis Christophides, and Vincent Leroy, editors, *Proc. 17th International*

- Conference on Database Theory (ICDT), Athens, Greece, March 24-28, 2014*, pages 201–212. OpenProceedings.org, 2014. URL: <https://doi.org/10.5441/002/icdt.2014.22>, doi: 10.5441/002/ICDT.2014.22.
- 15 Floris Geerts and Antonella Poggi. On database query languages for K-relations. *J. Appl. Log.*, 8(2):173–185, 2010. URL: <https://doi.org/10.1016/j.jal.2009.09.001>, doi:10.1016/J.JAL.2009.09.001.
 - 16 Todd J. Green, Gregory Karvounarakis, and Val Tannen. Provenance semirings. In Leonid Libkin, editor, *Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 11-13, 2007, Beijing, China*, pages 31–40. ACM, 2007. doi:10.1145/1265530.1265535.
 - 17 Todd J. Green and Val Tannen. Models for incomplete and probabilistic information. In Torsten Grust, Hagen Höpfner, Arantza Illarramendi, Stefan Jablonski, Marco Mesiti, Sascha Müller, Paula-Lavinia Patranjan, Kai-Uwe Sattler, Myra Spiliopoulou, and Jef Wijsen, editors, *Current Trends in Database Technology - EDBT 2006, EDBT 2006 Workshops PhD, DataX, IIDB, IIHA, ICSNW, QLQP, PIM, PaRMA, and Reactivity on the Web, Munich, Germany, March 26-31, 2006, Revised Selected Papers*, volume 4254 of *Lecture Notes in Computer Science*, pages 278–296. Springer, 2006. doi:10.1007/11896548_24.
 - 18 Martin Grohe and Peter Lindner. Independence in infinite probabilistic databases. *J. ACM*, 69(5):37:1–37:42, 2022. doi:10.1145/3549525.
 - 19 Tomasz Imielinski and Witold Lipski Jr. Incomplete information in relational databases. *J. ACM*, 31(4):761–791, 1984. doi:10.1145/1634.1886.
 - 20 Abhay Kumar Jha and Dan Suciu. Knowledge compilation meets database theory: Compiling queries to decision diagrams. *Theory Comput. Syst.*, 52(3):403–440, 2013. URL: <https://doi.org/10.1007/s00224-012-9392-5>, doi:10.1007/S00224-012-9392-5.
 - 21 Jean-Marie Lagniez and Pierre Marquis. An improved decision-DNNF compiler. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 667–673. ijcai.org, 2017. URL: <https://doi.org/10.24963/ijcai.2017/93>, doi:10.24963/IJCAI.2017/93.
 - 22 Silviu Maniu, Reynold Cheng, and Pierre Senellart. An indexing framework for queries on probabilistic graphs. *ACM Trans. Database Syst.*, 42(2):13:1–13:34, 2017. doi:10.1145/3044713.
 - 23 Silviu Maniu, Pierre Senellart, and Suraj Jog. An experimental study of the treewidth of real-world graph data. In Pablo Barceló and Marco Calautti, editors, *22nd International Conference on Database Theory, ICDT 2019, March 26-28, 2019, Lisbon, Portugal*, volume 127 of *LIPICs*, pages 12:1–12:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. URL: <https://doi.org/10.4230/LIPICs.ICDT.2019.12>, doi:10.4230/LIPICs.ICDT.2019.12.
 - 24 Yann Ramusat, Silviu Maniu, and Pierre Senellart. Semiring provenance over graph databases. In Melanie Herschel, editor, *10th USENIX Workshop on the Theory and Practice of Provenance, TaPP 2018, London, UK, July 11-12, 2018*. USENIX Association, 2018. URL: <https://www.usenix.org/conference/tapp2018/presentation/ramusat>.
 - 25 Yann Ramusat, Silviu Maniu, and Pierre Senellart. Provenance-based algorithms for rich queries over graph databases. In Yannis Velegarakis, Demetris Zeinalipour-Yazti, Panos K. Chrysanthis, and Francesco Guerra, editors, *Proceedings of the 24th International Conference on Extending Database Technology, EDBT 2021, Nicosia, Cyprus, March 23 - 26, 2021*, pages 73–84. OpenProceedings.org, 2021. URL: <https://doi.org/10.5441/002/edbt.2021.08>, doi:10.5441/002/EDBT.2021.08.
 - 26 Pierre Senellart. Provenance and probabilities in relational databases. *SIGMOD Rec.*, 46(4):5–15, 2017. doi:10.1145/3186549.3186551.
 - 27 Pierre Senellart. On the impact of provenance semiring theory on the design of a provenance-aware database system. In Antoine Amarilli and Alin Deutsch, editors, *The Provenance of Elegance in Computation - Essays Dedicated to Val Tannen, Tannen's Festschrift, May 24-25, 2024, University of Pennsylvania, Philadelphia, PA, USA*, volume 119 of *OASICs*,

- pages 9:1–9:10. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. URL: <https://doi.org/10.4230/OASICS.Tannen.9>, doi:10.4230/OASICS.TANNEN.9.
- 28 Pierre Senellart, Louis Jachiet, Silviu Maniu, and Yann Ramusat. Provenance and probability management in PostgreSQL. *Proc. VLDB Endow.*, 11(12):2034–2037, 2018. URL: <http://www.vldb.org/pvldb/vol11/p2034-senellart.pdf>, doi:10.14778/3229863.3236253.
- 29 Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. *Probabilistic Databases*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011. doi:10.2200/S00362ED1V01Y201105DTM016.