The Pastwatch: On the usability of provenance data in relational databases

Omar AlOmeir*, Eugenie Yujing Lai*, Mostafa Milani*, Rachel Pottinger*
*University of British Columbia
{oomeir, eugenie.lai, mkmilani, rap}@cs.ubc.ca

***Abstract*—Provenance information can be large and overwhelming to users. We present a set of criteria that any provenance exploration tool must have and introduce *Pastwatch*, a provenance exploration system that adheres to those criteria. We also address the issues associated with provenance of aggregation queries, including the creation of a summarization method that makes provenance of aggregation queries manageable for users. Finally, we conduct a quantitative user study to show statistically significant results that *Pastwatch* makes provenance information more efficient and easier to use than standard approaches.**

***Index Terms*—provenance, relational databases, summarization, visualization, usability**

I. Introduction

Data provenance is any information about the origin of a piece of data and the process that led to its creation. Provenance has been researched in a number of different areas. In the scientific work-flow context, provenance is used to show the processes that data went through. Thus, provenance work in work-flow management has included work on visualizing and presenting provenance information to users. In contrast, in database research most of the focus has been on finding the sources that contributed to the results of a query. This has included a lot of work on developing comprehensive models and representations of data provenance. However, there is very little work on presenting database provenance information in a way that is not overwhelming to users. Using existing database provenance systems can be overwhelming for those who do not have prior knowledge of data provenance models or the data itself. Furthermore, provenance information can increase the data size exponentially. This can have serious implications on both storage and usability. We argue that data provenance needs to be visualized and in some cases summarized to support and facilitate broad exploration. Our system, *Pastwatch*, is a first step toward making provenance data widely usable without the need for deep technical knowledge.

We identify two main challenges: 1) Provenance information needs to be visualized in a way that facilitates exploration. Visualizations should include query results, the processes that led to the creation of the query results, the source tables. 2) Provenance information for aggregate query results can be large and prohibitive for exploration. An aggregation query that counts the number of tuples in a relation returns a single number. The provenance of this single number can be the whole relation. This creates a barrier for users to look through provenance information and find meaningful information. *Pastwatch* addresses these two challenges.

In this paper, we review provenance research and use our findings to create *Pastwatch*, a first step toward comprehensible database provenance. In particular, our contributions are:

- We survey provenance research and create a concise set of desirable features for a provenance exploration system.
- We introduce the *Pastwatch* provenance exploration system and describe its novel visualization features.
- We validate *Pastwatch* with a user study that shows how it improves accuracy and efficiency.
- We introduce a new way to summarize provenance data for aggregation query results.

II. System Design Principles

In this section we lay out a set of design principles that should be followed by future database provenance exploration systems and are adhered to by our *Pastwatch* system.

In particular, in this paper we seek to solve the problem of provenance exploration, which is to present provenance information without overwhelming size and complexity. Provenance exploration in turn allows non-experts to understand the provenance of their relational data. Insightful visualizations of carefully curated provenance information give users the ability to use provenance information to explore the data.

We define a data provenance exploration system to support data provenance exploration through two components: 1) A back-end DBMS with support for provenance, 2) A front end user interface with visualizations. In this section, we characterize design principles for these two components which are considered in the design of *Pastwatch*.

A. Provenance System Principles

What sets an exploration system apart from traditional systems is that the traditional approaches rely on the user to query and explore the results via data manipulation languages. In traditional approaches, the models can be complex and some query languages can be unintuitive, making things difficult.

Several data provenance papers have looked at the desirable features for a data provenance system [1], [2]. Authors in such surveys list desirable features for provenance in data management systems, including user interfaces and visualizations. In this section, we combine these features and augment them with our own to detail the principles for a data provenance exploration system for which *Pastwatch* is a first attempt.

Our principles for a data provenance exploration system are as follows:

1) **Support multiple types of provenance.** As argued in [1], a comprehensive provenance exploration system should support multiple types of provenance. While provenance semantics may not affect the visualization, the user could need different semantics depending on the scenario.
2) **The back-end system should allow for provenance data to be queried.** [1] Without this feature, it would be impossible to show provenance information to the user, let alone visualize it in any meaningful way.
3) **Provenance exploration systems should support provenance information at different granularities.** [3] Provenance exploration systems must support both tuple and table-level granularities in order to make decisions.
4) **Store provenance data in a way that lets provenance information be decoupled from the data.** The way provenance is stored should allow for querying and isolating of provenance information while retaining the link to the data.
5) **Make provenance simpler to use with different dissemination techniques.** As seen in work-flow provenance systems, provenance exploration needs visualizations that give the user the ability to freely explore.
6) **Provenance of large size results of aggregation queries should be summarized.** Provenance of aggregation query results can be large and daunting. A summary of such provenance information can help users make the most of this information. This problem is quite challenging, as we discuss in detail in Section III-B.

We designed *Pastwatch* in accordance with the above design principles. *Pastwatch* supports multiple types of provenance; the back-end supports both why- and where-provenance. The back-end is a relational DBMS, which means it is efficient and simple to query, and provenance information can be queried. Finally, visualization and summarization of data and provenance are first class components of *Pastwatch*.

B. Visualization principles

The design of the visualization and exploration component is guided by the following set of visualization principles:

- **Overview first, zoom and filter, then details on demand** [4]. This principle is essential for dealing with large scale information. The main goal of our visualizations is to show the user a summarized view of the data (usually an aggregation). The user can browse through a custom overview by specifying an aggregation or filtering query. The user can then zoom in and look at specific tuples and look at their provenance information.
- **Recognition over recall** [5]. Navigating through data tables and drilling down can confuse the user as to where they are in the system or the provenance of query results. Keeping track of where the user is via visual elements beats having to remember where they are. Hence we create a provenance graph to show a persistent overview that highlights which level the user is currently browsing.
- **Appropriate encoding should be used for the underlying data**. We use a graph to show the provenance information sources and how they relate to the results. We use a horizontal treemap to represent the summarization rules. Using this visualization allows nesting rules within rules. The score of a rule is mapped to the area of the rectangle that contains the rule.
- **Multiple views are most effective when explicitly linked** [6]. The user is given the option to utilize multiple views at once. Highlighting one element in a certain view (the bar chart for example) would highlight the same item on a different view (country on a map for example).
- **We use aggregation and filtering to reduce the data**. Visual idioms have limitations in terms of the number of items they can display [5]. Reducing guarantees better scalability for large datasets and more efficient visualizations.

III. *Pastwatch* OVERVIEW

Our goal in implementing *Pastwatch* was to provide users with all the facilities they need to use provenance information. *Pastwatch* has two main components: 1) A data and provenance visualization component. 2) A data summarization component that summarizes the provenance of tuples produced by aggregation queries. In this section we go into detail on each component.

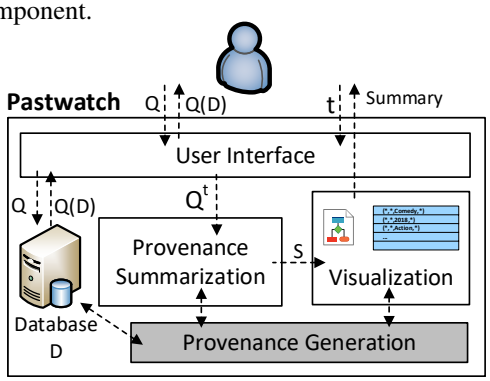


Fig. 1: An overview of *Pastwatch*. The system takes a query and returns the query results to the user, including a visualization of provenance information. The user asks for the provenance of a tuple in the query results. The summarization component takes the provenance, summarizes it and returns a ranked list of rules of size k.

A. Visualization components

Pastwatch visualizes provenance information that is stored in relational tables side by side with their respective tuples. The data alone can be large and overwhelming, and the provenance information increases the size and complexity. The user needs the information to be divided into smaller subsets. To maximize comprehensibility, provenance information should be hidden until the user asks for it.

We created two components to browse the provenance information: the overview visualization and the provenance graph. The overview visualization summarizes the data and presents the user with a manageable overview. In this overview, the user can click on any data item to see its provenance. The provenance graph presents an interactive overview of the provenance of a piece data.

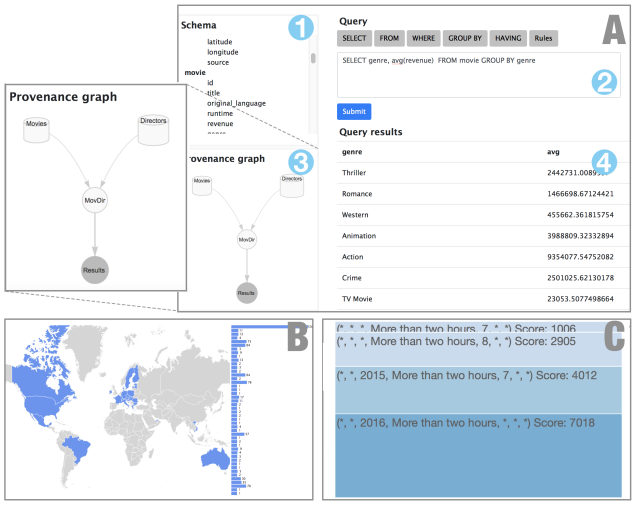


Fig. 2: *Pastwatch* interface. (A) Main interface with: 1) Schema list, 2) Query input, 3) Provenance graph, and 4) Query results. (B) The overview visualization in the form of a a linked view of a world map and bar-chart. (C) The summarization rules.

B. Summarization of provenance of aggregate query results

ID	Title	Director	Gender	Year	Genre	Rev (M$)
t_1	Lincoln	Steven Spielberg	M	2012	Drama	182
t_2	Bonjour Anne	Eleanor Coppola	F	2016	Comedy	13
t_3	Sicario	Denis Villeneuve	M	2015	Action	47
t_4	Mamma Mia	Phyllida Lloyd	F	2008	Comedy	144
t_5	Pitch Perfect 2	Elizabeth Banks	F	2015	Comedy	184

TABLE I: Movies table

1) Data summarization rules: The work in [7] introduces summarization rules to summarize "interesting" aspects of a table. In this section we use these rules to summarize and explore the provenance of aggregate queries. Provenance of an aggregate query is all the tuples that contributed to the aggregate value. Therefore, it is usually a much larger set than the results of an aggregation query.

Example: $s_1 = (\star, \star, \star, 2015, \mathit{Action}, \star)$ is a summarization rule over Table I that matches every Action movie from 2015.

In [7] the score of a list of rules $S = (s_1, s_2, ...)$ is defined as follows:

$$\mathit{Score}(S) = \sum_{s_i \in S} \mathit{MCount}(s_i, S) \times \mathit{Weight}(s_i). \quad (1)$$

where the score of a set of rules is the maximum score between all the possible lists containing the rules in the set. *Weight* is a monotone function that returns a non-negative real number. The weight function conveys how well a rule summarizes the values in a table. We use the common weight function from [7], $w =$ the number of non-$\star$ values.

The summarization problem is defined as follows: Given a relation R and a fixed value k, the summarization problem is to find a set of rules S with $|S| = k$ and maximum $\mathit{Score}(S)$. It is an NP-hard problem [7]. The authors of [7] present a greedy algorithm called Best Rule Set (BRS) that finds a sub-optimal set of rules efficiently. The approximation guarantee in the algorithm is based on the fact that the *Score* function is a sub-modular set-function [7, Lemma 3].

2) Pastwatch *summarization:* In *Pastwatch*, we present *AScore* for a set of rules as the maximum score between every possible list that contains the rules in the set. *AScore* generalizes *Score* by replacing $\mathit{MCount}(s_i, S)$ with $\mathit{MAgg}^t(s_i, S)$. Unlike *Score*, the *AScore* function considers the impact of the tuples covered by rules in S on the aggregate query result $Q^t(R)$. To measure this impact, we use *sensitivity analysis*, a technique that measures the sensitivity of a query to a tuple or a set of tuples [8]. The following function defines MAgg^t based on the sensitivity of Q^t:

$$\mathit{MAgg}^t(s_i, S) = \sum_{r \in \mathit{MCover}(s_i, S)} |Q^t(R) - Q^t(R \setminus \{r\})|. \quad (2)$$

Example: The user asks a query for the average revenue of all movies in Table I. The user proceeds to click on the tuple: $t = (\mathit{Comedy}, 113.6\ \mathit{M\$})$, asking for a summary of the provenance of this tuple. The Why-provenance of t is three tuples: t_2, t_4, and t_5. In this example t_2 is an interesting tuple because $t_2[\mathit{Rev}] = 13$. Which means t_2 has a considerable impact on the average of the revenue of the comedy movies. In this case, we prefer rules that can single out and highlight this tuple. Say there is a list of rules S_1 with only one rule $s_1 = (\star, \star, \mathit{F}, \star, \mathit{Comedy}, \star)$ that covers and explains all tuples t_2, t_4, t_5. *AScore* would assign a higher score to a set of rules $S_2 = (s_2, s_1)$ with $s_2 = (\star, \mathit{Eleanor\ Coppola}, \mathit{F}, \star, \mathit{Comedy}, \star)$ that highlights the movie with the highest impact on the average result.

AScore is sub-modular for aggregate functions Sum and Average which means we can use the greedy algorithm with the same guarantees.

IV. Quantitative User Study

We performed a quantitative user study to validate the visualization and exploration components of *Pastwatch*. In this study, the users interacted with a visualization of provenance of a data-set and its provenance meta-data to answer a set of questions. The users also tried to answer similar questions using a web interface that presented the provenance in HTML tables in the same format of Perm [9]. While users performed these tasks, we measured the time of completion and answer accuracy. The users also evaluated the difficulty of each task and offered subjective feedback at the conclusion of the study.

A. Hypotheses

Our hypotheses for this user study were:

- H1: The users would find it less difficult to answer questions using the provenance explorer than the web interface. This is measured on a 1–5 Likert scale.
- H2: The users would complete the tasks faster using the provenance explorer than the web interface.
- H3: The users would answer the questions more accurately using *Pastwatch* than the web interface.

B. Participants and setup

All participants were graduate and undergraduate students from the Computer Science and Electrical and Computer Engineering departments who had sufficient familiarity with data and relational tables. However, none of the participants had any prior experience with provenance information. The data-set used in the study is a real world financial data-set used in creating Global Legal Entity Identifiers (GLEIs) for financial institutions [10].

The study was conducted with 21 participants: 6 women and 14 men. Participants ranged from 2nd year undergrads to 4th year PhD students.

Study sessions were done with a single user at a time on a Macbook Pro 15" laptop. Participants who were not familiar with the Apple track-pad were offered a mouse.

C. Procedure

Each study session started with a survey for demographic information and to assess the participants' familiarity with the study concepts. The users were shown a demo of the system component that produces the table format output. The demo shows an example which is a smaller scale problem that resemble the four tasks the participants have to perform. Participants were also shown how to access the visualization component and where to input the answers to the tasks' answers. The participants were also provided with details on the data-set and what provenance means. The participants were then given instructions on how to perform their tasks with task questions and subjective questions. At the end of each session, participants were handed a questionnaire to evaluate the tool and provide comments.

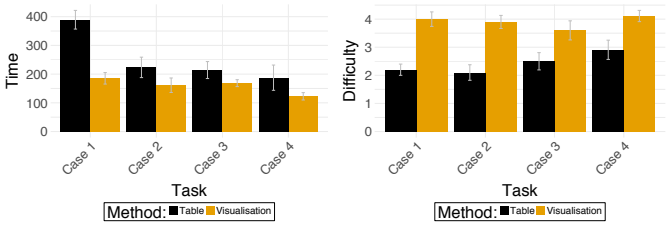
The participants' tasks consisted of two different sets of questions. In the first set, users were asked to locate a data item in the output and then locate its origin point. In the second set, users were asked to choose any data item from a certain source and to find out whether it has changed on update. Participants performed a task using *Pastwatch* or the web interface. 11 participants started with the visualization method, the other 10 started the first task with the web interface. Users did not write any queries to get the results, they simply clicked on a button to generate the query results. Each session lasted about 30-40 minutes.

D. Experimental design and analysis

The study had one within subject factor: the method used to find answers to the questions in each task and the following levels: 1) a web table representation of the results of a provenance generating query, and 2) the *Pastwatch* interface.

The dependent measures were: 1) The accuracy of answer (there was a binary correct or incorrect answer to each question). 2) The time it took to complete a task. 3) The subjective perceived difficulty of a task in Likert scale 1-5.

Because the data does not follow a normal distribution, we used a the Wilcoxon signed rank test which tests the median difference between two sets of observations.



(a) Time to complete (b) Perceived difficulty

Fig. 3: Results for time and perceived difficulty broken down by task for each method. Error bars correspond to standard error.

E. Results and discussion

Efficiency, Perceived difficulty, and Accuracy We saw statistically significant effects of the different methods on the time it took to complete a task, the perceived difficulty, and the accuracy ($P < 0.01$). Participants spent less time to complete tasks using *Pastwatch* (M = 163 seconds) than they did using the web interface (M = 262 seconds). On a scale of 1 being very difficult and 5 very easy, participants rated the task (M = 3.95) using *Pastwatch* and (M = 2.42) using the web interface. Participants, on average, were able to answer 92% of the questions correctly using *Pastwatch* compared to 82% using the web interface.

Discussion Visualization of database provenance is not the end-all solution to all provenance information problems. However, it is a first step toward a comprehensive understanding of provenance. Comparing the table format and visualization may not seem fair. However, there are no other current approaches that can offer the same visualization. Work-flow provenance systems work with different semantics and other database approaches offer little in terms of user interface we could compare against.

REFERENCES

[1] B. Glavic and K. R. Dittrich, "Data provenance: A categorization of existing approaches." in *BTW*, vol. 7, no. 12, 2007, pp. 227–241.

[2] Y. L. Simmhan, B. Plale, and D. Gannon, "A survey of data provenance in e-science," *SIGMOD Record*, vol. 34, no. 3, pp. 31–36, 2005.

[3] S. B. Davidson and J. Freire, "Provenance and scientific workflows: challenges and opportunities," in *SIGMOD*, 2008, pp. 1345–1350.

[4] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *VL/HCC*, 1996.

[5] T. Munzner, *Visualization Analysis and Design*. CRC Press, 2014.

[6] J. C. Roberts, "State of the art: Coordinated & multiple views in exploratory visualization," in *CMV*, 2007, pp. 61–71.

[7] M. Joglekar, H. Garcia-Molina, and A. Parameswaran, "Interactive data exploration with smart drill-down," in *ICDE*, 2016, pp. 906–917.

[8] B. Kanagal, J. Li, and A. Deshpande, "Sensitivity analysis and explanations for robust query evaluation in probabilistic databases," in *SIGMOD*, 2011, pp. 841–852.

[9] B. Glavic, R. J. Miller, and G. Alonso, "Using SQL for efficient generation and querying of provenance information," in *In Search of Elegance in the Theory and Practice of Computation*. Springer, 2013, pp. 291–320.

[10] T. Li, V. L. Lemieux, and R. Pottinger, "Challenges in resolving semantic heterogeneity with the global legal entity identifier system," in *DSMM*, 2014.