LIMA: Denial Constraint Discovery in Large Dynamic Databases

Albert Martin
Universitat Politècnica de Catalunya
Barcelona, Spain
albert.martin.g@upc.edu

Eduardo C. de Almeida
Federal University of Paraná
Curitiba, Brazil
eduardo@inf.ufpr.br

Oscar Romero
Universitat Politècnica de Catalunya
Barcelona, Spain
oscar.romero@upc.edu

Anna Queralt
Universitat Politècnica de Catalunya
Barcelona, Spain
anna.queralt@upc.edu

ABSTRACT

Modern Database Management Systems rely on metadata, such as data profiles and attribute associations, to function effectively. However, generating and maintaining this metadata on very large databases often incurs significant computational and storage costs. In this demonstration paper, we present a system capable of efficiently inferring and maintaining metadata in the form of Denial Constraints (DCs). This formalism has attracted considerable interest due to its expressiveness, as it can capture a wide range of data rules, including Functional Dependencies, Keys, and Order Dependencies. Rather than repeatedly extracting DCs whenever they are needed, our system maintains this metadata continuously and updates it seamlessly as the underlying data evolves, with minimal computational and memory overhead. This is the first system that enables the practical discovery of DCs on very large datasets, thereby opening the door to numerous applications that were previously infeasible due to the high cost of obtaining DCs.

PVLDB Reference Format:
Albert Martin, Eduardo C. de Almeida, Oscar Romero, and Anna Queralt. LIMA: Denial Constraint Discovery in Large Dynamic Databases. PVLDB, 19(12): 4762 - 4765, 2026.
doi:10.14778/3827998.3828116

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/NoSocAlgroc/LIMA.

1 INTRODUCTION

Modern Database Management Systems rely on accurate and comprehensive metadata to operate effectively. Such metadata can take multiple forms, including domain profiles, integrity constraints, and attribute associations. This information plays a fundamental role in a wide range of tasks, including data quality, query optimization, and database normalization. Managing such a diverse set of metadata is inherently challenging, which has motivated research into general formalisms capable of capturing richer metadata. One prominent example is Denial Constraints (DCs), a class of integrity rules that subsumes several well-known dependency languages, including Unique Column Combinations (UCCs), Functional Dependencies (FDs), Order Dependencies (ODs), and their conditional variants, such as Conditional Functional Dependencies (CFDs) and Conditional Order Dependencies (CODs)[1].

Denial Constraints (DCs) assert the impossibility of any pair of tuples in a dataset $t_x, t_y \in D$ simultaneously satisfying a given set of predicates $\{P_1, P_2, \dots\}$. Formally, a DC can be expressed as the negation of a conjunction of predicates over tuple pairs. For example, a Functional Dependency (FD) $A \rightarrow B$ can be represented as $\neg(t_x.A = t_y.A \wedge t_x.B \neq t_y.B)$, indicating that no two distinct tuples can agree on attribute A while differing on attribute B. Similarly, a Unique Column Combination (UCC) on attribute ID can be expressed as $\neg(t_x.ID = t_y.ID)$, ensuring that no two tuples share the same ID value. An Order Dependency (OD) between attributes $Salary$ and Tax can be formulated as: $\neg(t_x.Salary < t_y.Salary \wedge t_x.Tax > t_y.Tax)$, which enforces that tuples with lower salary values cannot have higher tax values. This expressiveness enables the representation of a wide range of data rules and relationships, making the identification of DCs for a given dataset immensely valuable. Furthermore, it would be ideal for such metadata to be maintained alongside the database, rather than recomputed each time it is needed.

In this demonstration paper, we present the first system capable of discovering and maintaining DCs for large and dynamic databases, with minimal computational and memory overhead. The backbone of this system is our novel DC discovery algorithm Lattice Information Maximization Algorithm (LIMA)[3]. This algorithm has been specifically designed for the task of discovering DCs on large and dynamic environments, and tackles the main challenges faced by the state of the art of DC discovery:

- **Quality**: A substantial proportion of DCs discovered by existing algorithms do not correspond to meaningful or semantically valid constraints[1, 2, 4]. Our system addresses this limitation by ensuring that discovered DCs capture statistically significant relationships among attributes.
- **Row Scalability**: Since DCs are defined over pairs of tuples, traditional discovery algorithms exhibit quadratic complexity in the number of tuples, limiting their applicability to large datasets[1, 2, 4, 6, 7]. Our system relies on representative sampling, enabling the use of statistical inference to assess DC validity without exhaustively evaluating all tuple pairs.

Proceedings of the VLDB Endowment, Vol. 19, No. 12 ISSN 2150-8097.
doi:10.14778/3827998.3828116

- **Column Scalability**: The combinatorial explosion of predicate sets results in exponential complexity with respect to the number of attributes. Our system mitigates this issue by pruning the search space, avoiding combinations of statistically independent predicates and thereby improving efficiency in high-dimensional settings [1, 2, 4–8].
- **Dynamic Data**: Most existing approaches assume static data, and those allowing updates incur high costs on large datasets [6, 7]. In contrast, our statistical framework naturally accommodates dynamic data, allowing updates in the dataset to be efficiently propagated to the validity of the DCs discovered by our system.

Our demonstration provides users with the opportunity to observe the advantages of LIMA over state-of-the-art DC discovery algorithms, particularly in terms of the quality of the discovered constraints and the efficiency of the process. Furthermore, the demonstration allows users to dynamically modify the underlying dataset and observe how the metadata evolves accordingly. We also present several use cases that leverage this metadata, including the visualization of Functional Dependencies and Order Dependencies, as well as the analysis of attribute associations. All these functionalities are built on top of the dynamically maintained DCs, ensuring that the results remain consistent with the current state of the data.

In summary, our demonstration showcases the novel capability of maintaining and exploiting sets of DCs on very large and dynamic data. While there are numerous systems designed to leverage metadata in the form of DCs for a variety of tasks, their applicability is often constrained by the cost of discovering the DCs. Our system eliminates this limitation by ensuring DCs are always up to date and freely available, remaining valid through all states of the data. This significantly reduces the overhead traditionally associated with metadata extraction and enables a wide range of applications that depend on such information, which were previously unusable due to computational constraints.

2 SYSTEM DESIGN

In this section, we present the details of our system for both discovering DCs in large datasets and updating them efficiently as the data evolves.

2.1 Statistical Inference for DC Validity

For any DC $\varphi = \neg(P_1 \wedge P_2 \wedge \cdots \wedge P_k)$, its validity depends on the joint probability that all predicates are simultaneously satisfied, denoted as $p(\varphi)$. Traditional DC discovery systems estimate this probability through point estimates derived from the evaluation of φ over all pairs of tuples. While prior research has significantly optimized this process, the overall computational complexity remains quadratic in the number of tuples. This limitation motivates the need for methods capable of discovering DCs in large-scale datasets without requiring a full scan of all tuple pairs.

Instead of relying on point estimates for $p(\varphi)$, we model the uncertainty of this quantity by considering the distribution of its possible values. This distribution is inferred by evaluating φ over a sample of uniformly drawn tuple pairs, where n is the size of the sample and k is the number of tuple pairs fulfilling all predicates φ. This statistical modeling is the foundation of our system, as it enables reasoning about the properties of DCs without requiring their evaluation over all tuple pairs. Consequently, the method is not only more efficient, but also more robust, as it explicitly accounts for the uncertainty associated with the inferred properties of the constraints. Next, we detail how this statistical framework can be used to tackle each of the challenges described in Section 1.

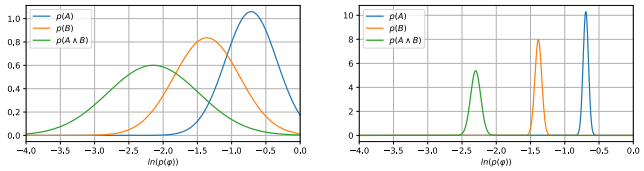
2.2 Quality

The main challenge regarding the quality of DC sets produced by existing methods lies in their low precision. In practice, classic DC discovery algorithms typically generate a number of constraints that exceeds the true number of meaningful attribute relationships in the data by several orders of magnitude [1, 2, 4]. This phenomenon is largely attributed to the fact that most approaches define DC validity solely in terms of the joint predicate probability being small enough, i.e., $p(\varphi) \leq \epsilon$. Under this criterion, the aggregation of statistically independent predicates becomes a viable strategy for generating "valid" DCs which lack semantic significance.

To address this limitation, we use a more restrictive definition of DC validity that requires constraints to capture genuine relationships among predicates. Specifically, the system ensure that the inclusion of each predicate uniquely alters the conditional distribution of another predicate [2]. This prevents the inclusion of independent predicates, thereby significantly improving the semantic quality of the DCs discovered by our system.

2.3 Row scalability

As described in Section 2.1, classical DC discovery algorithms require precise estimates of $p(\varphi)$ to determine DC validity. Obtaining such estimates requires evaluating φ over all pairs of tuples in the dataset, which becomes prohibitively expensive on large databases. In contrast, our statistical framework eliminates the need for exact estimates of $p(\varphi)$. Instead, we maintain a probability distribution over $ln(p(\varphi))$ for each candidate DC φ, inferred from its evaluation on a uniform sample of tuple pairs.



(a) Loose distributions for $ln(p(\varphi))$ with sample size 10. **(b) Tight distributions for $ln(p(\varphi))$ with sample size 1000.**

Figure 1: Distributions of joint predicate probabilities $ln(p(\varphi))$ estimated with different sample sizes.

Figure 1 illustrates the type of information held by our system through these distributions. In Figure 1a, only 10 uniformly sampled tuple pairs are used to infer the distributions, resulting in high uncertainty and insufficient evidence to assess the validity of the corresponding DCs. In Figure 1b, a larger sample size of 1000 enables more confident inference, as it becomes possible to conclude with high confidence that $p(A \wedge B) < 0.1$. More importantly, the system can also infer that $p(A \mid B) < p(A)$ and $p(B \mid A) < p(B)$ with high confidence, establishing that $\neg(A \wedge B)$ captures a meaningful approximate DC with $\epsilon = 0.1$ between predicates A and B.

It is important to emphasize that these conclusions are reached without requiring exact estimates of joint predicate probabilities. Using a sample of only 1000 tuple pairs, the system is able to derive results that would traditionally require scanning a full database containing potentially billions of tuple pairs. In effect, our approach decouples the cost of DC discovery from the size of the dataset, making it instead dependent on the desired approximation ϵ. This is crucial to discover any kind of metadata in very large databases.

2.4 Column scalability

Another major challenge in DC discovery is the rapid growth in computational cost with respect to the number of attributes. Empirical evidence has shown that the runtime of classical DC discovery algorithms increases exponentially with the number of columns in the dataset [1, 2, 4–8]. This behavior is primarily driven by the combinatorial explosion of predicate sets, largely resulting from the unrestricted combination of independent predicates [2]. By contrast, the more restrictive DC validity definition introduced in Section 2.2 effectively prevents the agglomeration of independent predicates. By incorporating this definition directly into the exploration mechanism of our system, we significantly reduce the search space of candidate DCs, hence improving the column scalability.

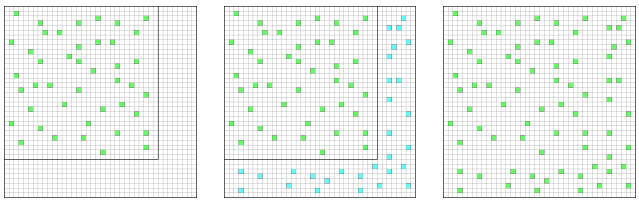
At any point during execution, our system maintains a set of candidate DCs Ω, together with a sample of tuple pairs used to infer the distribution associated with each $\varphi \in \Omega$[1]. Initially, the candidate set contains only the empty constraint, i.e., $\Omega = \{\emptyset\}$, and the sample associated with each $\varphi \in \Omega$ begins empty. At each iteration, the algorithm increases the sample size for each candidate DC in order to refine the estimation of its distribution. This increase is performed proportionally to the variance of the corresponding distributions, ensuring that computational effort is concentrated on uncertain estimates while avoiding unnecessary work for well-characterized ones. Subsequently, the system evaluates whether the updated distributions provide sufficient evidence that a given DC holds some relationship, as described in Section 2.2. When this is the case, supersets of that DC are incorporated into Ω, as the evidence no longer supports the hypothesis that such predicate combinations are independent.

In essence, this exploration mechanism ensures our system allocates computation optimally, both in terms of refining distributions and when deciding which DCs should be considered.

2.5 Dynamic Data

Most existing DC discovery algorithms are designed for static datasets, while those that support dynamic data often incur substantial memory overhead. Both limitations make these approaches unsuitable for large-scale and evolving environments. In contrast, the computational cost and memory requirements of our system are fully decoupled from the size of the database. Consequently, our approach is particularly well-suited for scenarios in which the volume of data renders classical DC discovery methods impractical. Furthermore, as discussed in Section 2.3, our framework naturally accommodates evolving distributions. In the static setting, this is reflected by progressively increasing the sample of tuple pairs, allowing the estimated distribution of joint predicate probabilities to converge to their true values. In dynamic settings, the same principle can be adapted to allow the distributions to adapt as new data is incorporated.

[1]In practice, the tuple pairs themselves are not materialized in memory by our system. It suffices to store the sample size (n) and the number of pairs satisfying φ (k) to fully characterize the distribution.



(a) $T \subset D \times D$ **(b)** $\Delta T \subset D' \times D' - D \times D$ **(c)** $T' \subset D' \times D'$

Figure 2: Diagram displaying how a uniform sample over a domain of tuple pairs (a) may be extended with a uniform sample over pairs of newly added tuples (b) to obtain a uniform new sample over all tuple pairs (c).

The only requirement for the reliability of our statistical approach is that the sample of tuple pairs is uniform over the dataset. Under this condition, the inferred distributions remain representative of the full population, with variances that decrease monotonically as the sample size increases. Figure 2 illustrates how this invariant can be preserved as new data is introduced. Given a uniform sample of tuple pairs $T \subset D \times D$ from a dataset D, it can be extended to remain uniform over an updated dataset $D' = D \cup \Delta D$ by incorporating an additional sample $\Delta T \subset D' \times D' \setminus D \times D$ drawn uniformly from the newly introduced region. This process may be inverted to deal with data deletions, where a distribution drawn from a uniform sample on D' may be updated to become representative of D.

3 DEMONSTRATION

Our demonstration provides users with the opportunity to experience the advantages of our system for DC discovery on large-scale dynamic databases in comparison with state-of-the-art approaches.

The front-end interface has been implemented as a Neo4j web plugin, leveraging its graph database capabilities to visualize and interact with the evolving metadata. An overview of the interface is shown in Figure 3. The interface first allows users to select from a list of datasets, with the option to upload a custom dataset. Subsequently, users can choose among several DC discovery algorithms, including ours (LIMA), to serve as the backend. Finally, users may specify the target approximation factor ϵ for the discovered DCs.

The demonstration focuses on visualizing the evolution of metadata in the form of DCs over a large and dynamic dataset. Internally, the system maintains both the dataset and the set of valid DCs, while users can dynamically insert or remove batches of tuples and observe how the discovered constraints evolve accordingly. This interactive environment enables the audience to directly observe the advantages of our system over the state of the art with respect to the four aforementioned challenges:

- **Quality**: For any dataset and approximation factor, other algorithms produce an unmanageable amount of rules, which in most cases don't correspond to meaningful constraints. In contrast,

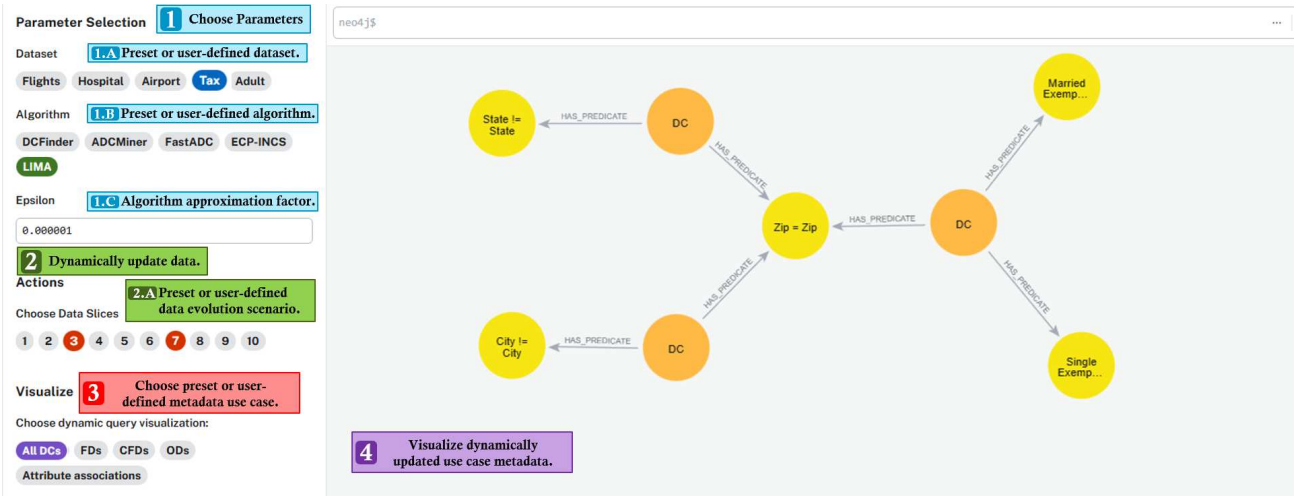


Figure 3: System frontend interface, with controls on the left and a graph on the right for visualizing the evolving results.

the DCs generated by LIMA achieve precisions that are orders of magnitude higher than those of the state of the art [2].

- **Row Scalability**: The audience can choose datasets with varying number of tuples, up to dozens of millions, and experience how the time required to discover DCs on large datasets is reduced by several orders of magnitude when using LIMA.
- **Column Scalability**: The audience can choose datasets with varying number of attributes, up to several dozens. They can see existing algorithms fail to operate on datasets with more than 25 attributes, whereas LIMA remains effective in such high-dimensional settings.
- **Dynamic Data**: The audience is able to dynamically update the dataset, and when using LIMA, see how these are efficiently propagated to the set of valid DCs.

In addition, the system provides mechanisms to exploit this discovered metadata through the interface. A variety of use cases are supported via pre-defined and user-defined queries on the graph, enabling users to seamlessly interact with the evolving set of DCs. For instance, users can visualize the complete set of constraints or focus on specific subclasses such as Functional Dependencies or Order Dependencies. The system also offers a summarized view of attribute associations derived from the discovered DCs, and allows users to define custom use cases that leverage this metadata.

All visualizations and query results are dynamic and are automatically updated as the dataset evolves through user interaction. This demonstration highlights the versatility of dynamically maintained metadata in the form of DCs, and aims to showcase the great value of making rich metadata inference an integral part of the DBMS.

4 CONCLUSIONS

In this demonstration paper, we present the first system capable of efficiently maintaining a set of Denial Constraints over very large and dynamic databases. Our demonstration enables users to experience the superior scalability of our approach compared to the state of the art in DC discovery, and the improvement in result quality. We further showcase several ways in which this metadata can be exploited, illustrating how its minimal maintenance cost can enable a wide range of applications. Ultimately, the ability to readily access a rich set of previously unobtainable DCs on large and dynamic databases has the potential to enable a wide range of applications exploiting them.

ACKNOWLEDGMENTS

This work is supported by the Horizon Europe Programme under GA.101135513 (CyclOps) and the Spanish Ministerio de Ciencia e Innovación under project PID2024-161707OB-I00 / AEI /10.13039/501100011033 (FeeD). Anna Queralt is a Serra-Húnter fellow. Eduardo Almeida is funded by the CNPQ grants 302909/2022-2 and 444192/2024-7. Albert Martin is funded by the predoctoral program AGAUR-FI grants (2025 FI-1 00967) Joan Oró, which is backed by the Department of Research and Universities of the Generalitat of Catalonia, as well as the European Social Plus Fund.

REFERENCES

[1] Xu Chu, Ihab F. Ilyas, and Paolo Papotti. 2013. Discovering denial constraints. *Proc. VLDB Endow.* 6, 13 (2013), 1498–1509. https://doi.org/10.14778/2536258.2536262 Number: 13.

[2] Albert Martin, Eduardo C. De Almeida, Oscar Romero, and Anna Queralt. 2025. How and Why False Denial Constraints are Discovered. *Proceedings of the VLDB Endowment* 18, 10 (June 2025), 3477–3489. https://doi.org/10.14778/3748191.3748209

[3] Albert Martin, Eduardo C. De Almeida, Oscar Romero, and Anna Queralt. 2026. Discovering Approximate Denial Constraints in Large Databases. *Proceedings of the VLDB Endowment* 19, 9 (2026), 2086 – 2098. https://doi.org/10.14778/3819518.3819536

[4] Eduardo H. M. Pena, Eduardo C. de Almeida, and Felix Naumann. 2019. Discovery of approximate (and exact) denial constraints. *Proc. VLDB Endow.* 13, 3 (Nov. 2019), 266–278. https://doi.org/10.14778/3368289.3368293 Number: 3.

[5] Eduardo H. M. Pena, Fabio Porto, and Felix Naumann. 2022. Fast Algorithms for Denial Constraint Discovery. *Proc. VLDB Endow.* 16, 4 (2022), 684–696. https://doi.org/10.14778/3574245.3574254 Number: 4.

[6] Eduardo H. M. Pena, Fabio Porto, and Felix Naumann. 2024. Discovering Denial Constraints in Dynamic Datasets. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. 3546–3558. https://doi.org/10.1109/ICDE60146.2024.00273 ISSN: 2375-026X.

[7] Chaoqin Qian, Menglu Li, Zijing Tan, Ai Ran, and Shuai Ma. 2023. Incremental discovery of denial constraints. *The VLDB Journal* 32, 6 (Nov. 2023), 1289–1313. https://doi.org/10.1007/s00778-023-00788-y Number: 6.

[8] Renjie Xiao, Zijing Tan, Haojin Wang, and Shuai Ma. 2022. Fast approximate denial constraint discovery. *Proc. VLDB Endow.* 16, 2 (Oct. 2022), 269–281. https://doi.org/10.14778/3565816.3565828 Number: 2.