SHORT-PAPER

DataDazzle: Intelligent Data Exploration through Natural Language

MIKE XYDAS, National and Kapodistrian University of Athens, Athens, Attica, Greece

ANNA MITSOPOULOU, National and Kapodistrian University of Athens, Athens, Attica, Greece

GEORGE KATSOGIANNIS-MEIMARAKIS, Grenoble Alpes University, Saint Martin d'Heres, Auvergne-Rhone-Alpes, France

CHRIS TSAPELAS, National and Kapodistrian University of Athens, Athens, Attica, Greece

STAVROULA ELEFTHERAKIS, Grenoble Alpes University, Saint Martin d'Heres, Auvergne-Rhone-Alpes, France

ANTONIS MANDAMADIOTIS, Grenoble Alpes University, Saint Martin d'Heres, Auvergne-Rhone-Alpes, France

View all

Published: 22 June 2025

Citation in BibTeX format

SIGMOD/PODS '25: International Conference on Management of Data
June 22 - 27, 2025
Berlin, Germany

Conference Sponsors:
SIGMOD

SIGMOD/PODS '25: Companion of the 2025 International Conference on Management of Data (June 2025)
https://doi.org/10.1145/3722212.3725094
ISBN: 9798400715648

✨ DataDazzle: Intelligent Data Exploration through Natural Language

Mike Xydas
mxydas@di.uoa.gr
Athena Research Center
Athens, Greece
Department of Informatics and Telecommunications, National and Kapodistrian University of Athens
Athens, Greece

Anna Mitsopoulou
amitsop@di.uoa.gr
Archimedes, Athena Research Center
Athens, Greece
Department of Informatics and Telecommunications, National and Kapodistrian University of Athens
Athens, Greece

George Katsogiannis-Meimarakis
katso@athenarc.gr
Athena Research Center
Athens, Greece
Université Grenoble Alpes
Grenoble, France

Chris Tsapelas
ctsapelas@di.uoa.gr
Archimedes, Athena Research Center
Athens, Greece
Department of Informatics and Telecommunications, National and Kapodistrian University of Athens
Athens, Greece

Stavroula Eleftherakis
seleftheraki@athenarc.gr
Athena Research Center
Athens, Greece
Université Grenoble Alpes
Grenoble, France

Antonis Mandamadiotis
antonismand@athenarc.gr
Athena Research Center
Athens, Greece
Université Grenoble Alpes
Grenoble, France

Georgia Koutrika
georgia@athenarc.gr
Athena Research Center
Athens, Greece

Abstract

For over five decades, enabling non-technical users to access databases without SQL has been a major research focus. Despite recent advances in Text-to-SQL, the primary focus remains on accuracy, often overlooking real-world deployment challenges. This demo presents DataDazzle, a natural language data exploration system. It allows users to query data, review and verify answers, and benefit from recommendations, offering a comprehensive data exploration experience.

CCS Concepts

• **Information systems → Search interfaces**.

Keywords

Natural Language Interfaces for Databases, Text-to-SQL, SQL-to-Text, Recommenders

ACM Reference Format:
Mike Xydas, Anna Mitsopoulou, George Katsogiannis-Meimarakis, Chris Tsapelas, Stavroula Eleftherakis, Antonis Mandamadiotis, and Georgia Koutrika. 2025. ✨ DataDazzle: Intelligent Data Exploration through Natural Language. In *Companion of the 2025 International Conference on Management of Data (SIGMOD-Companion '25), June 22–27, 2025, Berlin, Germany*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3722212.3725094

ACM ISBN 979-8-4007-1564-8/2025/06
https://doi.org/10.1145/3722212.3725094

1 Introduction

As data fuels a wide range of human activities, the capacity to access and extract meaningful insights from it has emerged as an essential proficiency. However, data is often stored in large and complex data stores that require technical expertise and proficiency in query languages like SQL to access. For this reason, researchers have long sought solutions to help non-technical users to benefit from data. Early efforts provided keyword search tools, but their limited expressivity hindered users from posing complex queries and retrieving the desired data [7].

Recently, substantial efforts have been directed towards developing Text-to-SQL systems, which often leverage LLMs to translate natural language (NL) questions into SQL [6]. Despite advancements in their performance, deploying text-to-SQL systems in real-world scenarios and integrating them into data exploration pipelines involves several key challenges [3].

Challenges. We focus on the following crucial challenges.
– *Reliability*. Users not familiar with SQL cannot verify if the generated SQL query matches their request or they may not fully understand how the results were obtained resulting in low confidence in the system responses. This may carry significant risks in domains such as healthcare, where undetected errors can have serious consequences (e.g., lead to wrong treatment decisions). To ensure the system is reliable, mechanisms are required to align retrieved data with user expectations by providing meaningful feedback.

– *Response Presentation.* When working with SQL, programmers typically expect to receive unprocessed tabular data as output. This raw data format requires further manipulation and interpretation to derive meaningful insights, posing a challenge for those unfamiliar with SQL or data processing techniques. This is a problem in data exploration settings, where users specify their requests in natural language. Showing raw tables may hinder user understanding and data discovery. Results should be presented in a way that brings out their inherent structure and highlights their unique characteristics.
– *Holistic Data Exploration.* Relying solely on natural language search for data retrieval is insufficient, as users may lack clarity on their specific needs. In a data exploration context, user queries should be complemented by system interventions or guidance, creating a comprehensive and cohesive exploration experience.

Providing clear starting points helps users utilize data effectively. Recommendations tailored to users' goals enhance this process.
– *Portability.* Last but not least, such a system must be portable and easy to adapt to new use cases, even with the unique peculiarities of each database and domain. To truly empower users to access databases freely, the system must support seamless transfer to any new database they wish to explore. Additionally, since such a system will most likely rely on LLMs, the solution should remain model-agnostic, allowing users to integrate the model they prefer.
Our Solution. To address these challenges, we present *DataDazzle*, a natural-language data exploration system for relational data. Our vision is to deliver the next generation of database accessibility through an intuitive and holistic data exploration experience.
– **Reliability via Explanations**. Users can explore data comprehensively through a wide range of queries, from simple lookups to complex analytical ones. When a user poses a query in NL, this is automatically translated into SQL and then executed on the database to retrieve the required data. To help users confidently use the results, the system also generates an explanation in NL.
– **Dynamic Result Presentation**. DataDazzle offers three result presentation formats: textual, entity-based, and tabular. Textual Presentation provides natural language summaries for analytical queries, allowing users to extract insights without interpreting raw tables. Entity-based Presentation structures results around user-defined entity types, enhancing clarity and engagement. Tabular Presentation serves as a fallback for complex queries best represented in traditional tables. This adaptive approach ensures that results are presented in an intuitive and informative way.
– **Multi-level Recommendations**. To further enhance data exploration, DataDazzle offers recommendations that span multiple levels of relationships involving entities: those of the same entity type, those between different entity types, and those between users and entities. This multi-level approach allows DataDazzle to effectively address diverse user needs and exploration goals.
– **A Generalized Solution**. DataDazzle was designed for adaptability, requiring minimal configuration to deploy on new databases. Both of our NL services are powered by an LLM but remain model-agnostic, making them adaptable to various available resources. Developers simply define the entities to expose in query results, optionally provide examples of expected NL questions paired with their corresponding SQL queries to improve performance, and create the frontend interface. This plug-and-play approach eliminates the need for manual prompt engineering, vector storage management, and complex database connection setups. By automating these processes, DataDazzle greatly reduces the effort and time needed to deploy an NL interface for exploring new databases.

Figure 1 shows the system's interaction flows and architecture.
Demonstration Details. Conference participants will explore the OpenAIRE Graph database[1], rich with research entities. They will interact with DataDazzle via a user-friendly web interface and gain insights into its setup, query processing, and presentation strategies.

2 Overview of DataDazzle

2.1 Natural Language Search

DataDazzle's NL search uses two synergistic components: *Text-to-SQL* generates a SQL query from the user question, and *SQL-to-Text* converts the SQL query into an NL explanation for user verification. Both components are powered by an LLM, currently Llama3.1-70b [1], though any instruction-tuned LLM can be used, depending on the use case. To effectively adapt across diverse databases and domains, DataDazzle incorporates customizable prompts for each component and a few-shot example retriever, both of which can be seamlessly tailored to new databases.
Text-to-SQL Component. To perform Text-to-SQL, we follow the guidelines set from state-of-the-art systems [4] and apply in-context learning to make our solution transferable without fine-tuning. Our goal is to construct an informative input prompt that will help the LLM perform accurate Text-to-SQL translations. The three most important parts of the prompt are: (i) the *task instructions*, (ii) the *database representation*, and (iii) the *few-shot examples*. Firstly, the task instructions define what the model must do and may include database-specific guidelines. The database representation contains important information about the database and its schema, essential for generating accurate SQL queries. Column names, data types, foreign and primary keys, and sampled values from each column, are automatically extracted for several databases (e.g., SQLite, PostgreSQL, MySQL). Finally, an example retriever provides NL-SQL pairs that are semantically similar to the user's question and demonstrate how the model must perform the task. Overall, this customizable design incorporates a wide range of options found in the literature [4] while supporting quick integration of new components (e.g., a new example selector).
SQL-to-Text Component. As with the previous component, DataDazzle uses an LLM-based solution with in-context learning for SQL-to-Text. The input prompt includes similar elements as before, adapted to this inverse task. An additional pre-processing step was implemented for this component that aims to simplify the input SQL query to improve LLM's understanding. More specifically, the developer can optionally provide descriptive names for database attributes (e.g., "publication_id" instead of "pub_id") to make the SQL query more interpretable.
Automatic Configuration. To simplify and streamline adaptation to new databases, DataDazzle features an automated test suite that, given a dataset of NL-SQL pairs for the target database, searches for the optimal configuration for the input prompt construction in

[1]https://graph.openaire.eu/

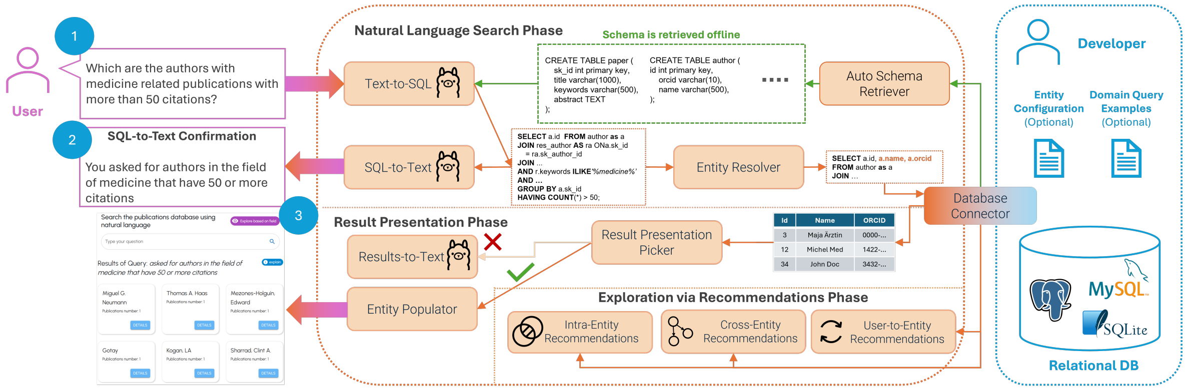

Figure 1: DataDazzle Interaction Flows of Data Exploration via Natural Language and Recommendations.

each component. This feature reduces integration effort, making the system both accessible and effective across diverse use cases. Moreover, the same NL-SQL pairs can be used by the few-shot example retriever, further enhancing the performance of both components.

2.2 Dynamic Result Presentation

To make result presentation engaging and tailored for intuitive exploration, DataDazzle offers three result presentation formats: *textual*, *entity-based*, and *tabular*.

Textual Presentation. For analytical queries, DataDazzle employs a Results-to-Text model powered by an LLM. In our demo, we use Llama3.1-8b [1], though the system supports other models, independent of the LLM used for Text-to-SQL translation. This model generates concise, query-specific NL answers by taking as input the query results (formatted as markdown tables), the user's NL query, and the involved table names. This allows users to gain insights easily, without interpreting raw data tables.

Entity-based Presentation. Entities represent instances of user-defined data types derived from the database schema and domain context. For example, in our demo, publications and authors are entities of different types. Each entity type is characterized by a set of attributes (columns from the relevant table(s) in the database)

The developer deploying DataDazzle can select the most representative ones to define an **entity profile**, forming the foundation for presenting entities in a clear and engaging manner (e.g., as clickable cards). The **entity resolver** acts as a corrective layer for the Text-to-SQL service, ensuring its output aligns with the entity profiles defined by the developer. In particular, it verifies that all attributes specified in an entity profile are included in the SQL query results. If any attributes are missing, the resolver rewrites the SQL query to retrieve them. For instance, if a user asks, "Bring me the publications after 2023", and the generated SQL query returns only publication IDs, the resolver adds attributes like title and publication date, ensuring complete and meaningful results.

Tabular Presentation. DataDazzle also provides a conventional table format, best suited for queries with large or complex result sets for which the aforementioned formats are less effective.

The **result presentation picker** determines the best presentation format using a set of heuristics. These heuristics take into account factors such as the number of the results and the attributes in the SELECT clause. For example, if the query returns only entity-defining values, it chooses the entity-based format.

2.3 Multi-level Recommendations

DataDazzle provides three recommendation components to enhance data exploration. *Intra-Entity Recommendations* identifies and suggests related entities of the same type, enabling users to explore connections at an entity level. *Cross-Entity Recommendations* uncovers relationships between different entity types, guiding users to discover interlinked data. Lastly, *User-to-Entity Recommendations* generates personalized recommendations based on user interactions with entities, streamlining goal-oriented data exploration.

Intra-Entity Recommendations. This component identifies entities similar to a given entity by computing pairwise similarities based on categorical and textual attributes. Categorical attributes are transformed into one-hot vectors, and their similarity is computed using cosine similarity. Textual attributes are concatenated, and sentence embeddings are generated using SBERT [9]. Pairwise cosine similarities are then computed between sentences, and the final text similarity score is computed using the SDR method [5]. Developers must specify the entity type (e.g., publication) and the relevant attributes to configure the recommendations.

Cross-Entity Recommendations. This component analyzes associations between entities of different types (e.g., citations linking authors and publications) to generate recommendations, helping users uncover valuable cross-entity insights. Given an entity, the *target entity*, our state-of-the-art collaborative filtering recommender establishes neighborhoods of similar entities based on their connections (e.g., co-authored publications) to generate recommendations for the target entity (e.g., finding relevant publications for a given author) [2]. To enable this feature, developers must specify the two entity types, the type of the target entity, and their relationships.

User-to-Entity Recommendations. This component relies on user interactions with entities (e.g., viewing a specific publication). To handle limited user feedback during early deployment, we use an online learning-based recommender that dynamically adapts as users engage with entities [8]. To reduce the recommendation search space, developers can optionally define entity groups (e.g., research fields for publications). The service then recommends both groups and a set of entities within each group. As users interact

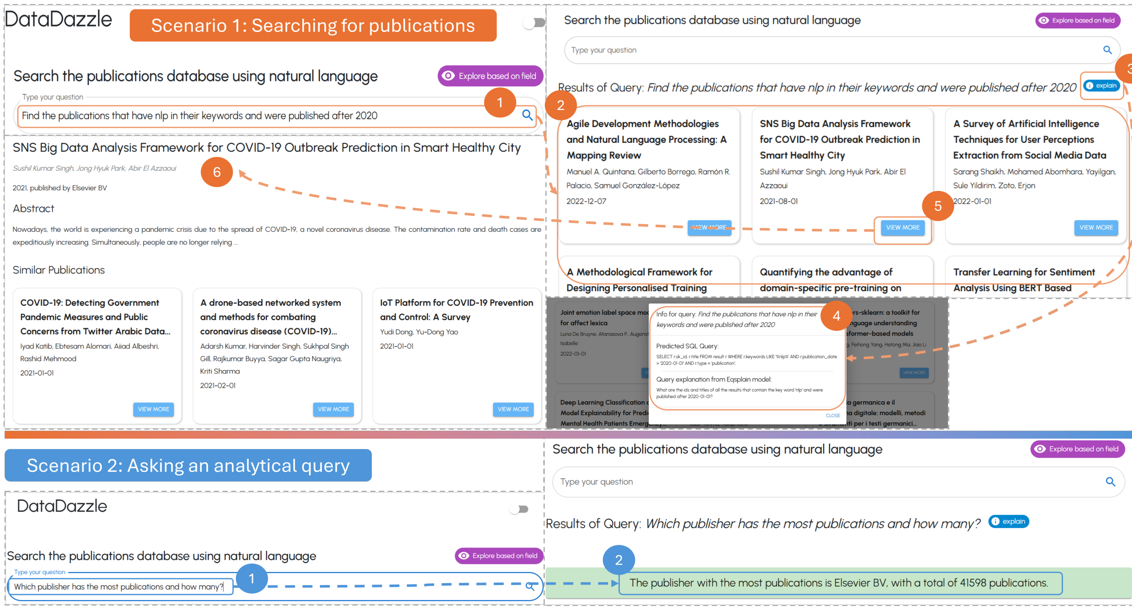

Figure 2: Exploration Scenarios in DataDazzle.

with entities, recommendations are updated in real time using a multi-armed bandit approach, balancing exploring new options with prioritizing those that have received the most interactions. Developers must specify the interaction type (e.g., clicks) and the number of entities to recommend, with groups being optional.

3 Demonstration

This demo shows researchers how NL search integrates into a data exploration pipeline. DataDazzle is configured on a subset of the OpenAIRE Graph with 120K publications, 570K authors, and 320K citations. Participants will use its web interface to explore data, receive recommendations, and observe query processing and presentation strategies. Figure 2 illustrates two usage scenarios.

Scenario 1: Searching for Entities of Interest. In this scenario, the user explores NLP publications by entering the query *"Find the publications that have NLP in their keywords and were published after 2020"* 1. The system translates the query to SQL, and returns a list of relevant publications 2. The user then selects the Explain button 3, which reveals both the SQL-to-Text explanation and the SQL query itself, allowing for validation of the results 4. For demonstration purposes, the SQL query is displayed, though this feature can be toggled based on the expertise of the users. While browsing the results, the user finds a publication about COVID-19 outbreak prediction via social media analysis 5 and opens its detailed view 6. At the bottom of this view, the Intra-Entity recommendations offer similar publications, enabling users to continue their exploration.

This scenario demonstrates a seamless and continuous exploration experience. Unlike conventional NL interfaces that might return static tables of publication titles, DataDazzle enables users to extract and interact with meaningful insights in a single workflow.

Scenario 2: Asking an Analytical Query. In our second scenario, the user wants to ask an analytical question: *Which publisher has the most publications and how many?* 1. The *Result Presentation Picker* component considers text as the most appropriate answer format, returning back the answer in NL: *The publisher with the most publications is Elsevier BV, with a total of 41598 publications.* 2. This scenario highlights DataDazzle's ability to generate concise answers for analytical queries, enabling efficient data exploration without overwhelming the user with raw data.

4 Acknowledgments

This work has been partially supported by DataGEMS, funded by the European Union's Horizon Europe Research and Innovation programme, under grant agreement No 101188416. This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.

References

[1] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. 2024. The llama 3 herd of models. (2024).
[2] S. Eleftherakis, G. Koutrika, and S. Amer-Yahia. 2024. Optimizing Neighborhoods for Fair Top-N Recommendation. In *UMAP 2024*. ACM, 57–66.
[3] J. Fürst, C. Kosten, F. Nooralahzadeh, Y. Zhang, and K. Stockinger. 2025. Evaluating the Data Model Robustness of Text-to-SQL Systems Based on Real User Queries. In *EDBT 2025, Barcelona, Spain, March 25-28, 2025*. 158–170.
[4] D. Gao, H. Wang, Y. Li, X. Sun, Y. Qian, B. Ding, and J. Zhou. 2023. Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. arXiv:2308.15363 [cs.DB] https://arxiv.org/abs/2308.15363
[5] D. Ginzburg, I. Malkiel, O. Barkan, A. Caciularu, and N. Koenigstein. 2021. Self-Supervised Document Similarity Ranking via Contextualized Language Models and Hierarchical Inference. In *ACL-IJCNLP 2021*. Online, 3088–3098.
[6] G. Katsogiannis-Meimarakis and G. Koutrika. 2023. A survey on deep learning approaches for text-to-SQL. *The VLDB Journal* 32, 4 (Jan. 2023), 905–936.
[7] G. Koutrika. 2024. Natural Language Data Interfaces: A Data Access Odyssey (Invited Talk). In *ICDT 2024 (LIPIcs, Vol. 290)*. 1:1–1:22.
[8] A. Mandamadiotis, G. Koutrika, and S. Amer-Yahia. 2024. Guided SQL-Based Data Exploration with User Feedback. In *ICDE 2024*. IEEE, 4884–4896.
[9] N. Reimers and I. Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *EMNLP 2019*.