

Changing the Paradigm from Dynamic Queries to LLM-generated SQL Queries with Human Intervention

Assor Ambre*
Inria

Hyeon Jeon†
SNU

Sungbok Shin*
Inria

Jean-Daniel Fekete*
Inria

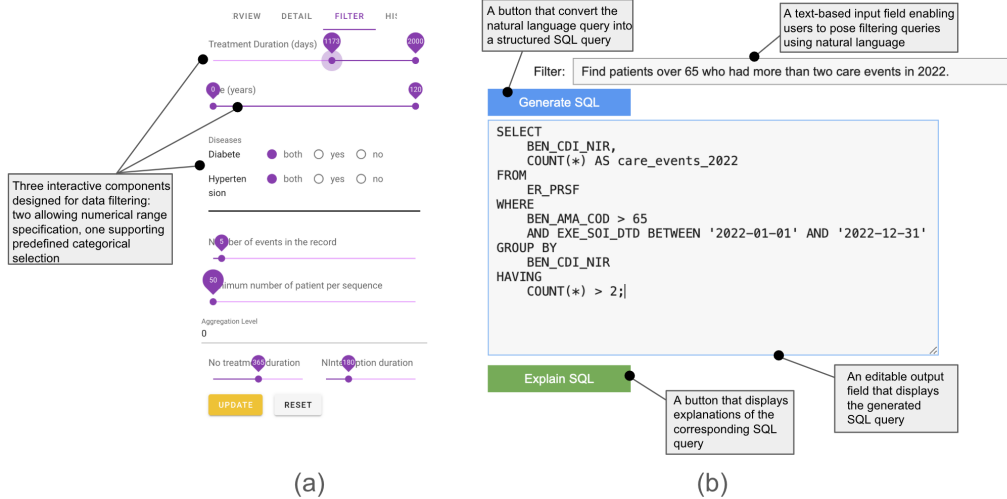


Figure 1: (a): Dynamic query widgets from the current version of the ParcoursVis aggregated EHR sequence visualization system (see the Filter tab on the right panel at parcourvis.lisn.upsaclay.fr). The panel has limited space and cannot show the large number of available attributes and value codes for filtering patients. (b): Prototype of our vision using an LLM to translate a natural language query into an SQL query featuring an editable query box and a natural language paired-explanation.

ABSTRACT

We propose leveraging Large Language Models (LLMs) as an interaction layer for medical visualization systems. In domains like healthcare, where users must navigate high-dimensional, coded, and heterogeneous datasets, LLM-generated queries enable expert medical users to express complex analytical intents in natural language. These intents are then translated into editable and executable queries, replacing the dynamic query interfaces used by traditional visualization systems built around sliders, check boxes, and drop-downs. This interaction model reduces visual clutter and eliminates the need for users to memorize field names or system codes, supporting fluid exploration, with the drawback of not exposing all the filtering criteria. We also reintroduce dynamic queries on demand to better support interactive exploration. We posit that medical users are trained to know the possible filtering options but challenged to remember the details of the attribute names and code values. We demonstrate this paradigm in ParcoursVis, our scalable EventFlow-inspired patient care pathway visualization system powered by the French National Health Data System, one of the largest health data repositories in the world.

Index Terms: LLM, Visualization, Temporal Event Sequences, Electronic Health Records, Interactive Filtering.

*e-mail: first-name.last-name@inria.fr.

†e-mail: hj@hcil.snu.ac.kr

1 PROBLEM STATEMENT

Large-scale Electronic Health Record (EHR) databases are being developed to inform policy-making and facilitate disease analysis [23]. For medical users, such as doctors, nurses, and medical administrative managers, EHRs deliver beneficial context, such as demographics, clinical history, medications, diagnostic results, and coded data linked to external dictionaries. However, EHR data also presents the challenge of containing hundreds of patient attributes. For example, the French National Health Database (SNDS) [22], managing the EHRs of the French population, contains over 3,000 variables related to patient care, organized in tables filled with numbers, text, and codes [21].

Traditional visualization systems rely on the direct manipulation paradigm for interaction [17], where users directly interact with the visualized objects of interest or indirectly with dynamic query widgets [18]. In EHR visualization systems such as EventFlow [14] and our system ParcoursVis [1], visible objects include steps in treatment pathways and event types that users can select and filter out for exploration. Additionally, they display dynamic query widgets to filter on other attributes (Figure 1 (a)). However, dynamic query user interfaces can only accommodate a limited number of widgets.

The main challenge is to effectively deliver the specific patient information that doctors are looking for [4], providing sufficient filtering capabilities to support in-depth data exploration. Given the large number of variables to manage, dynamic query interfaces may not offer the most effective solution.

2 HUMAN-GUIDED LLM-GENERATED SQL QUERIES

To address the challenge of identifying and presenting relevant variables to medical users, we envision design choices that stem from the practical difficulties of delivering patient information in a way

that aligns with their needs. Specifically, we identify two key requirements: *usability* and *transparency*. Indeed, many medical users have a working knowledge of coding and database querying. However, navigating datasets remains difficult, particularly when variables are encoded in cumbersome ways, such as with alphanumeric codes or non-descriptive identifiers. LLMs can generate SQL queries [2, 8–10, 13], but to be effective and provide agency to doctors [3, 20], they should have these two properties:

- **Editable, scriptable queries:** The system should expose the generated SQL and allow users to edit it. This promotes user agency, allowing them to inspect, understand, and own the variable-selection and filtering logic [19].
- **Natural-language paired explanations:** Each generated query should be accompanied by a plain-language description of its effects. This enhances transparency [7], facilitates learning about the database structure (e.g., codes, fields), and helps detect LLM mismatches or hallucinations [6].

To that end, we suggest a two-stage process to support doctors in querying complex attributes effectively on large EHR databases.

Step 1. The first stage is the query translation phase (NL \Rightarrow SQL), where the doctors’ requests are translated into one or more SQL query versions. The translation phase is done by the LLM. The benefits are that the doctors can learn, check, edit, save the queries for later reuse, and share them with colleagues; they are deterministic, whereas the translation process with LLMs is not.

Step 2. Within the second stage (SQL \Rightarrow Vis), the queries are executed and power the visualization, which shows various details about the features the doctors selected.

Yet, this process loses some important properties of the traditional dynamic queries [18]. We discuss that issue in Section 4.

3 USE-CASE SCENARIO

We illustrate our vision using *ParcoursVis*, our web-based system for visualizing large-scale EHR event sequences. Inspired by *EventFlow* [14], *ParcoursVis* scales to datasets 4–5 orders of magnitude larger (tens of millions of patients, billions of events) using a progressive architecture [5]. It is applied to explore specific treatment cohorts extracted from the French SNDS database.

The SNDS data are split across multiple tables (e.g., medications, procedures, hospitalizations) and use opaque alphanumeric codes, making direct querying difficult, even for users familiar with the schema. Retrieving and interpreting these codes often requires external dictionaries from various sources, which can be complex or outdated. For example, the *ER_PRSF* table [22] contains records of reimbursed healthcare acts. In our subset, each care episode includes attributes such as the beneficiary’s age at care (*BEN_AMA_COD*), encrypted social-security ID (*BEN_CDI_NIR*), year-month and exact date of death if applicable (*BEN_DCD_AME*, *BEN_DCD_DTE*), year of birth (*BEN_NAI_ANN*), municipality and department of residence (*BEN_RES_COM*, *BEN_RES_DPT*), and care start and end dates (*EXE_SOI_DTD*, *EXE_SOI_DTF*).

ParcoursVis features a **Filter tab** (Figure 1 (a)) which lets users filter views by patient attributes (e.g., comorbidities, age) and by various attributes (e.g., treatment duration). This could, in principle, extend to all the available attributes from the database, but the limited screen real estate cannot present all the dynamic query widgets to enable extensive filtering of care pathways. Instead, we explored the feasibility of using a pre-trained Mistral LLM [12] to translate natural language queries into SQL. It is open-source, downloadable, and licensed under Apache 2.0, so it can be run on any machine without restrictions. We provided it the *ER_PRSF* schema and prompted it with “*You are helping write SQL queries for a dataset from the SNDS health database. The data is stored in a single flat table called “ER_PRSF”, asking for the following format: 1. SQL query in a code block; 2. Short explanation; 3. Any assumptions you made.*” **Step 1.**’s output is shown in Figure 1 (b),

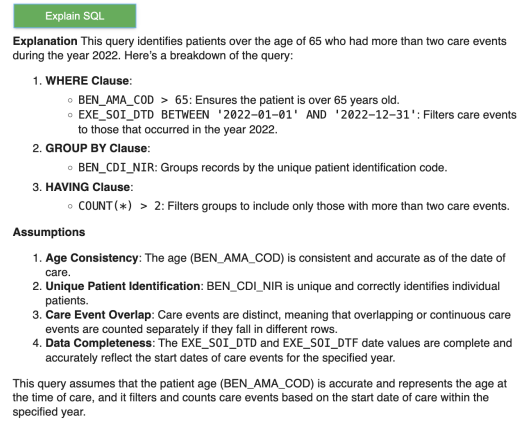


Figure 2: Example of explanations provided by our natural language to SQL LLM query translator.

and Figure 2. This step was part of an initial investigation to determine whether LLM-based query generation could be viable in our context (it is not integrated into the final system yet).

4 CHALLENGES ADDRESSING THE BENEFITS AND LIMITATIONS OF USING LLMs IN HEALTHCARE

Standardization of datasets. Processing complex health-related datasets such as SNDS presents several challenges. Besides navigating multiple tables, a model must detect and resolve duplicates, distinguish them when necessary, determine the shortest join paths, and suggest alternative queries. We think that many of these issues are caused by the lack of standardization, and thus advocate for standardizing schemas. Below, we introduce two examples.

- **Adopt the OMOP data model.** The ongoing effort to align health data with the OMOP model [15] promotes interoperability across clinical and medico-administrative databases. Conforming SNDS and other health datasets to OMOP would supply consistent input, allowing fine-tuned models to train on OMOP data.
- **Flatten complex datasets.** When applications do not require the richness of the full SNDS or equivalent, we propose simplifying medical data by flattening it into smaller single-table formats to reduce ambiguity, simplify analysis, and simplify LLM-based SQL generation. Different analytical needs (e.g., patient, disease, or treatment-based) may benefit from tailored flat tables, suggesting a use-case-specific flattening strategy.

Back to dynamic query interfaces. To revive key properties of dynamic queries, in particular *rapid, incremental, and reversible control; selection by pointing, not typing; and immediate, continuous feedback*, we propose extracting SQL subtrees, based on prior queries or user requests, to generate dynamic query widgets. Such a mechanism, already explored in [16], supports customizable, context-aware queries built from SQL fragments in the original tree. This implementation of the interface logic could be simplified by driving widget generation by an LLM [11] (versus being manually coded).

5 CONCLUSION

We outline our position in proposing an approach to replace pure dynamic query interfaces with LLM-generated, editable, and explained queries. It would promote user agency, allowing users to inspect and understand the variable-selection and filtering logic, and mitigate the unavoidable hallucinations. Our approach preserves the possibility of using regular dynamic queries for an interactively selected subset of queries, but provides the scalability needed to deal with real-world EHR databases in exploratory settings.

ACKNOWLEDGEMENTS

We thank our reviewers for their valuable insights. This work was supported in part by a grant from the Health Data-Hub, and from the URGE AP-HP/Inria project.

REFERENCES

- [1] A. Assor, M. Sereno, and J.-D. Fekete. Visualization of electronic health record sequences at scale. *arXiv preprint arXiv:2508.10700*, 2025. 1
- [2] R. C. Basole and T. Major. Generative AI for Visualization: Opportunities and Challenges. *IEEE Computer Graphics and Applications*, 44(2):55–64, 2024. doi: 10.1109/MCG.2024.3362168 2
- [3] D. Ceneda, C. Collins, M. El-Assady, S. Miksch, C. Tominski, and A. Arleo. A heuristic approach for dual expert/end-user evaluation of guidance in visual analytics. *IEEE Trans. Vis. Comput. Graph.*, 30(1):997–1007, 2023. 2
- [4] D. Ceneda, T. Gschwandtner, and S. Miksch. A review of guidance approaches in visual data analysis: A multifocal perspective. *Computer Graphics Forum*, 38(3):861–879, 2019. doi: 10.1111/cgf.13730 1
- [5] J.-D. Fekete, D. Fisher, and M. Sedlmair. *Progressive Data Analysis: Roadmap and Research Agenda*. Eurographics, Nov. 2024. doi: 10.2312/pda.20242707 2
- [6] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2), article no. 42, 55 pages, Jan. 2025. doi: 10.1145/3703155 2
- [7] Y. Huang, L. Sun, H. Wang, S. Wu, Q. Zhang, Y. Li, C. Gao, Y. Huang, W. Lyu, et al. Position: TrustLLM: Trustworthiness in Large Language Models. In *Forty-first International Conference on Machine Learning (ICML)*. OpenReview.net, 2024. 2
- [8] J. Kim, S. Lee, H. Jeon, K.-J. Lee, H.-J. Bae, B. Kim, and J. Seo. Phe-noFlow: A Human-LLM Driven Visual Analytics System for Exploring Large and Complex Stroke Datasets. *IEEE Trans. Vis. Comput. Graph.*, 2024. 2
- [9] S. Lee, H. Song, J.-c. Lee, Y. J. Lee, B. Lee, H.-E. Lim, D. Kim, J. Seo, and B. Kim. MedOrchestra: A Hybrid Cloud-Local LLM Approach for Clinical Data Interpretation. *arXiv preprint arXiv:2505.23806*, 2025. 2
- [10] J. Li, B. Hui, G. Qu, J. Yang, B. Li, B. Li, B. Wang, B. Qin, R. Geng, N. Huo, X. Zhou, C. Ma, G. Li, K. C. Chang, F. Huang, R. Cheng, and Y. Li. Can LLM Already Serve as A Database Interface? A BIG Bench for Large-Scale Database Grounded Text-to-SQLs. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems, (NeurIPS)*, 2023. 2
- [11] D. Masson, S. Malacria, G. Casiez, and D. Vogel. Directgpt: A direct manipulation interface to interact with large language models. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, article no. 975, 16 pages. Association for Computing Machinery, New York, NY, USA, 2024. doi: 10.1145/3613904.3642462 2
- [12] Mistral-small-3.1-24b-instruct-2503. <https://huggingface.co/mistralai/Mistral-Small-3.1-24B-Instruct-2503>. Accessed: 2025-06-30. 2
- [13] A. Mohammadjafari, A. S. Maida, and R. Gottumukkala. From natural language to SQL: Review of LLM-based text-to-SQL systems. *arXiv preprint arXiv:2410.01066*, 2024. 2
- [14] M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman. Temporal Event Sequence Simplification. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2227–2236, 2013. doi: 10.1109/TVCG.2013.200 1, 2
- [15] Standardized Data: The OMOP Common Data Model. <https://www.ohdsi.org/data-standardization/>. Accessed: 2025-06-30. 2
- [16] A. Pister, C. Prieur, and J.-D. Fekete. ComBiNet: Visual Query and Comparison of Bipartite Multivariate Dynamic Social Networks. *Computer Graphics Forum*, 42(1):290–304, 2023. doi: 10.1111/CGF.14731 2
- [17] B. Shneiderman. Direct Manipulation: A Step Beyond Programming Languages. *Computer*, 16(8):57–69, 13 pages, Aug. 1983. doi: 10.1109/MC.1983.1654471 1
- [18] B. Shneiderman. Dynamic queries for visual information seeking. *IEEE Software*, 11(6):70–77, 1994. doi: 10.1109/52.329404 1, 2
- [19] B. Shneiderman. Human responsibility for autonomous agents. *IEEE Intelligent Systems*, 22(2):60–61, March 2007. doi: 10.1109/MIS.2007.32 2
- [20] B. Shneiderman. *Human-centered AI*. Oxford University Press, 2022. 2
- [21] Documentation collaborative du SNDS. <https://documentation-snds.health-data-hub.fr/>. Accessed: 2025-06-30. 1
- [22] P. Tuppin, J. Rudant, P. Constantinou, C. Gastaldi-Ménager, A. Rachas, L. de Roquefeuil, G. Maura, H. Caillol, A. Tajahmady, J. Coste, C. Gissot, A. Weill, and A. Fagot-Campagna. Value of a national administrative database to guide public decisions: From the système national d’information interrégimes de l’Assurance Maladie (SNIIRAM) to the système national des données de santé (SNDS) in France. *Revue d’Épidémiologie et de Santé Publique*, 65:S149–S167, 2017. Réseau REDSIAM. doi: 10.1016/j.respe.2017.05.004 1, 2
- [23] Q. Wang and R. S. Laramée. EHR STAR: the state-of-the-art in interactive EHR visualization. *Computer Graphics Forum*, 41(1):69–105, 2022. doi: 10.1111/CGF.14424 1