



OPEN A case study using sewage metagenomic data for assessment of text-to-SQL capabilities in large language models

Ágnes Becsei^{1,3}, József Stéger^{1,3}, Dávid Visontai¹, Patrick Munk², Frank M. Aarestrup², István Csabai^{1,4} & Krisztián Papp^{1,4}✉

Relational databases offer an efficient solution for storing and retrieving complex data sets, yet the requirement for SQL programming expertise presents a significant challenge for many life science users. We explore whether a cutting-edge large language model can effectively translate plain English queries into SQL scripts (Text-to-SQL), thereby simplifying database interaction and eliminating the typical usage barriers. A complex database comprising 19 interconnected tables of metagenomic analyses from 239 sewage samples across five European cities was available. A large language model was provided with details of the database's structure and background information on its contents. We evaluated the functionalities of this "SewageGPT" tool and assessed the accuracy of its responses to complex questions and visualisation of results. Providing a detailed description of the database enabled SewageGPT to accurately respond to complex inquiries, accelerating the database querying process. Knowledge of the database content proved beneficial, as it minimized the risk of ambiguities in queries; however, ambiguities can lead to incorrect responses. Therefore, human oversight remains crucial, particularly for questions that lack detail or involve ambiguities. The integration of state-of-the-art large language models with direct database connectivity substantially enhances the efficiency of query generation, statistical analysis and visualization of the results.

Keywords Text-to-SQL, Large language model, Metagenomics

The FAIR Guiding Principles promote that scientific data should be Findable, Accessible, Interoperable, and Reusable, making them reusable for both machines and humans¹. When the size of a structured biology-related dataset increases and reaches a certain limit, then relational databases provide an efficient way to store and retrieve data. Databases are valuable, yet the need for SQL programming skills to execute complex queries can pose a challenge for many users. An easy-to-use dashboard web interface can provide solutions for predefined, often-used queries but is not flexible enough to answer the multiple specific questions a broader user community might have. Comprehending a complex database with several interconnected tables can be challenging and time-consuming, especially for the first time or if it is not used regularly, thus excluding several potential users. An attractive solution would be to use natural language questions—plain English (or another language)—translated automatically to SQL query in the background (text-to-SQL). This approach would allow users to access information from complex databases without SQL programming skills or in-depth knowledge of database schemas. The text-to-SQL field has seen substantial advancements over the past few years. Thus, in just three years, execution accuracy on the Spider² large-scale cross-domain dataset—a widely used benchmark—rose from 53.5³ to 85.3⁴. While these benchmarks are valuable, they can be somewhat artificial, as the databases often contain data that aligns more closely with everyday concepts. This makes them easier for large language models (LLMs) to comprehend, leveraging their strengths in processing familiar contexts.

We aimed to make a real-life evaluation of the usability of a state-of-the-art large language model for querying a database containing complex multi-level data from metagenomic analysis of sewage samples. For this specific database, the access barrier becomes even more significant when not only researchers but also public

¹Department of Physics of Complex Systems, ELTE Eötvös Loránd University, Budapest, Hungary. ²National Food Institute, Technical University of Denmark, Lyngby, Denmark. ³Ágnes Becsei and József Stéger contributed equally. ⁴István Csabai and Krisztián Papp share senior authorship ✉email: krisztian.papp@ttk.elte.hu

health professionals need to access and analyse pathogen-related data, as they may have limited experience with programmatic access.

We provide a detailed, step-by-step explanation of how to establish a connection between an in-house SQL database and a web-based LLM, accompanied by example files for fine-tuning the model. We also highlight common pitfalls when working with LLMs and propose strategies to avoid them—insights that can enhance both performance and usability. Furthermore, we compare the performance of models developed by two leading companies (OpenAI and Anthropic). Our findings illustrate the potential for extending additional databases with an easy-to-use LLM interface, thereby substantially broadening their user base. Nevertheless, the inherent limitations of LLMs—particularly in addressing highly complex, ambiguous, or unanswerable queries—must be carefully considered.

Methods

Metagenomics analysis

DNA was extracted and sequenced for all sewage samples. Quality-trimmed reads were used for read-level identification of ARGs using kma (v1.2.8)⁵ and the ResFinder database (commit = 3eedbde)⁶. The resulting read count per gene values were transformed into CLR (centered log-ratio) values to quantify the relative abundance of each ARG in each sample. ARGs were grouped by the antibiotic they confer resistance against, and the relative abundance of each class was also quantified.

The following analysis and data are partly derived from the study by Becsei and Fuschi et al.⁷. For contig level analysis trimmed reads were assembled into contigs for each sample (single-sample assembly) and for each sample collection site (co-assembly). Then, the resulting contigs were classified according to their potential origin using the PPR-meta (v. 1.1) tool⁸. Potential genes were identified within them using Prokka (v1.14.6)⁹ and subsequently, the contigs were used to reconstruct metagenome-assembled genomes (MAGs). The quality of these MAGs was evaluated using the CheckM2 (v1.0.1) tool¹⁰, and for further steps, we retained only medium- and high-quality MAGs based on the MIMAG (Minimum Information for Metagenome-Assembled Genomes) criteria¹¹. The remaining MAGs were dereplicated using the dRep (v3.4.2) tool¹² and taxonomically classified with GTDB-tk (v2.4.0 with GTDB r220 as reference)¹³. Quality-filtered reads were aligned to the taxonomically classified bacterial MAG set, and the average alignment depth for each MAG in each sample was calculated. These depth values were then transformed into CLR values to quantify the relative abundance of MAGs in each sample. More details on assembly and MAG-level analysis steps can be found in the study by Becsei and Fuschi et al.⁷. Additionally, the presence or absence of certain bacteria or bacterial pathogens was assessed using two methods. We conducted qPCR analysis targeting 21 selected bacterial pathogens, and a bacterium was considered present in a given sample or sampling site if its MAG could be reconstructed with at least medium quality in that sample or sampling site.

Configuration of the web-based customized ChatGPT interface

The web-based customized ChatGPT was configured on <https://chatgpt.com/>. The following text was included during the customization process:

- DESCRIPTION: “Sewage metagenomics is a powerful tool for the proactive surveillance of potential pathogens and disease outbreaks. Sewage samples from Copenhagen, Rotterdam, Budapest, Rome, and Bologna underwent metagenomic sequencing and results were stored in a database that is reachable with this GPT”.
- INSTRUCTION: “The GPT should construct SQL queries that answer the user’s questions and also need to run it. The “Knowledge” part contains 5 files. The “table_column_description.tsv” file includes a description of columns. The “table_description.tsv” file contains a description of tables. The “schema.sql” is the schema of SQL database. The “sewage_data_descriptor.txt” contains background information about the dataset. For the construction of SQL query be careful, each table is in the “distilled” schema. The “extra_info.txt” file contains supplementary information that can be useful to construct correct SQL queries. Whenever you connect to the database and send an SQL query, then always include exactly the original question as a comment at the end of the SQL query.”

In the KNOWLEDGE section, the uploaded files contained the database schema (see Supplementary Data S3), detailed descriptions of the tables and columns (see Supplementary Tables S1–S2), and background information on data collection (Becsei et al., submitted 2025).

SQL database–LLM integration architecture

The architecture of the components is sketched in Fig. 1. For cybersecurity reasons, in the PostgreSQL database where public sewage data reside, native database connections are not exposed. A REST API is deployed to handle proxy requests, equipped with a read-only database access role and preset resource limits to prevent system overloads. Query running times are limited. Additionally in the database, a logging table has been created in a separate schema, where every query flowing through the REST API is logged with metadata, including when the query started, how much time it took to complete and if it had been erroneous what the error message was. This table is utilized for further enhancements to the system and may become a valuable resource for fine-tuning of the text-to-SQL module.

Among the tested LLMs, only the customized ChatGPT web interface includes an action feature that enables a direct connection to a database. The programming of this action is shown in Fig. 2.

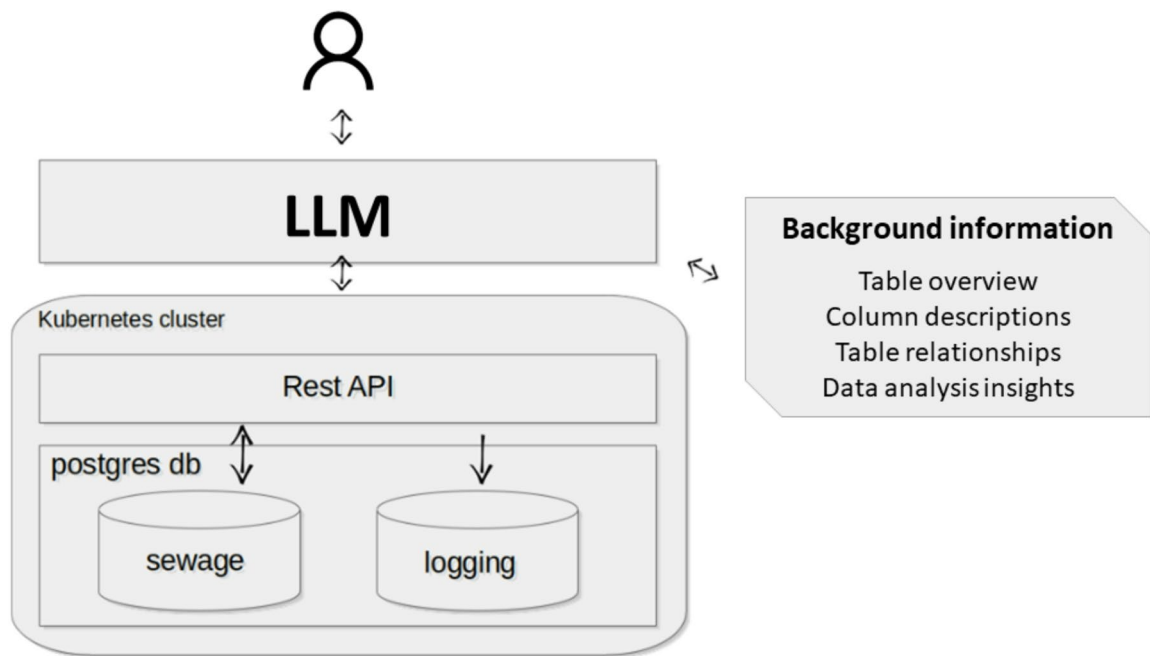


Fig. 1. The architecture of the components and list of provided background information.

```
{
  "openapi": "3.1.0",
  "info": {
    "title": "Sewage database",
    "description": "Retrieves data from Sewage database",
    "version": "v1.0.0"
  },
  "servers": [
    {
      "url": "https://k8plex-veo.vo.elte.hu/notebook/report/ezc9v9-api"
    }
  ],
  "paths": {
    "/sewage": {
      "get": {
        "description": "GET Sewage data using SQL query",
        "operationId": "getData",
        "parameters": [
          {
            "name": "sql",
            "in": "query",
            "description": "SQL query",
            "required": true,
            "schema": {
              "type": "string"
            }
          }
        ],
        "deprecated": false
      }
    }
  },
  "components": {
    "schemas": {}
  }
}
```

Fig. 2. Implementation of the action feature enabling database connectivity in the customized ChatGPT interface.

Results

Dataset and database

Sewage surveillance has emerged as a powerful tool for public health monitoring, providing a non-invasive, cost-effective means to study microbial communities, track pathogens, and monitor antimicrobial resistance (AMR) at the population level. This approach has proven particularly valuable in detecting antimicrobial resistance genes (ARGs)^{14–16} and tracking the spread of infectious diseases, such as SARS-CoV-2^{17,18}. Although the raw sequence data from these studies are publicly accessible, further exploration and reuse of these datasets require skilled bioinformaticians and substantial computational resources.

Our dataset includes analytical results for 239 sewage samples collected across five European cities – Copenhagen, Rotterdam, Budapest, Rome and Bologna – between 2019 and 2021. Metagenomics analysis is detailed in the methods section. To accommodate and organize the data generated from these analyses, we designed a PostgreSQL database, enabling efficient data storage, management, and retrieval. During the schema design of the database, we balanced between usability and the dogma of canonical data representation (Fig. 3). Detailed descriptions of the tables and columns are in Supplementary Table S1, S2.

Large language models (LLM)

A wide range of large language models (LLMs) are available, and this number continues to grow as new, improved, or specialised models are released. Selecting the most suitable model is challenging due to varying comparison outcomes and the rapid pace of advancements in the field, which often results in comparisons being outdated by the time they are published. Our goal was not to conduct a systematic comparison of all available models but rather to assess the performance of leading models from OpenAI and Anthropic on a complex database at the time of our analysis. ChatGPT4o model was used for demonstration of the web-based model functionalities; the choice of ChatGPT4o was based on several factors beyond performance alone, including ease of use and database integration, documentation, interface accessibility, and the ability to create figures, and basic statistical calculations.

Prompting

The input text or instructions, known as a prompt, provided to the model play a crucial role in the accuracy of its responses. To guide the model effectively, we included the database schema (see Supplementary Data S3), along with detailed descriptions of tables and columns (see Supplementary Table S1, S2), and background information (Becsei et al. submitted 2025.) on data collection.

The schema definition for the database was created using the standard tool `pg_dump`, which was subsequently truncated to include only the table and type definitions. This was done to shorten the prompt, reducing the cost of running the LLM.

Clear and precise column and table descriptions are essential. Including these descriptions in the prompt can serve as a validation measure: if incorrect answers are generated, it may indicate that the instructions were unclear. Below, we present key suggestions gathered during the iterative fine-tuning of SewageGPT, which helped reduce ambiguities and improve the success rate and accuracy of LLM-generated queries:

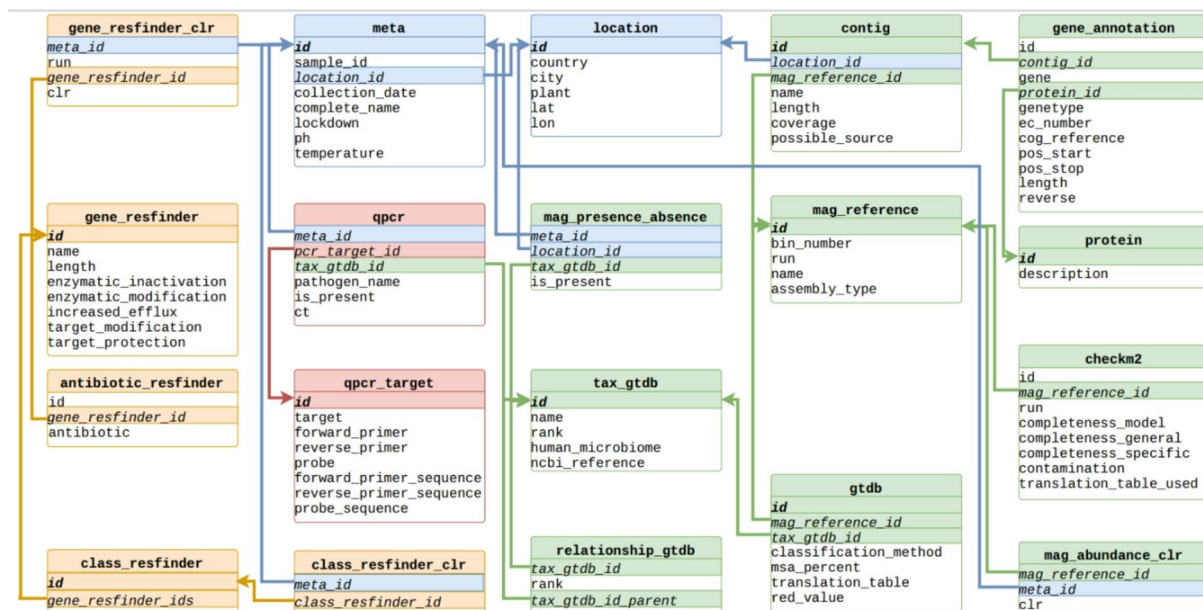


Fig. 3. Schema of the database. Rectangles represent the tables in the database with the column names and their connection to each other.

- If a column includes percentage values, the description should clarify whether 100% corresponds to a value of 100 or 1 in the table.
- Multiple methods exist for extracting the maximum value from a column. One option is to sort the column in descending order and select the top entry, or to use the MAX SQL function. If not specified, LLM may choose either method. However, if the column includes empty cells, these approaches may yield different outcomes. Sorting in descending order ('ORDER DESC') treats empty cells as the highest values, leading to an incorrect result when selecting the first entry. To ensure accurate results in such cases, it helps to include a note in the column description, such as: 'Some cells are empty; please remove these before ordering this column.'
- Right naming of columns can help LLM models figuring out the connection between tables. Although it may be evident from the formal schema definition, it proved to be a good practice that the column name description contain if it is a primary key that serves as unique identifiers for each row in the table or if it is a foreign key column in a table that points to the column with unique values in another table creating cross-referencing between the tables. We need to avoid cases when multiple columns represent very close semantic meaning as these cause ambiguities.
- Description of date containing columns should contain information about date format.
- Typos, misleading names or abbreviations in the names of columns may pose a problem. Even if such an incorrect column name is used consistently in the schema, LLM sometimes uses the grammatically corrected form in SQL query, generating a syntactically wrong query.

The clear column description is important for every database, even if no LLMs are used for querying. LLMs can be useful tools to point out ambiguous descriptions and cleanse the data structure for further human access.

Beyond the schema description, additional information can be provided to the LLM to improve accuracy when addressing questions prone to incorrect answers. This information might include detailed explanations of specific data types or example SQL queries for particular tasks. Providing example queries is especially useful when a specific syntax is required for custom-made SQL functions. These examples enable LLM to handle similar tasks in the future.

Functionalities

The tested models can be accessed either through its web interface or via an OpenAI/Anthropic API. Only the customized ChatGPT web interface includes an action feature that enables direct connection with a database, whereas Anthropic's web interface does not support communication with SQL databases and can therefore be tested only via the Anthropic API. The customized ChatGPT user interface and the OpenAI API reach the same background information yet the handling of the information and their capabilities are different. ChatGPT interface can automatically correct invalid SQL queries, whereas the basic OpenAI API setup lacks this functionality. The web interface is user-friendly, even for non-experts, and is directly connected to the sewage database. User queries are translated into SQL statements, submitted to the database through its REST API, and the results are presented in various formats: as text (Fig. 4a), tables (Fig. 4b), downloadable tables (Fig. 4c), or visualised as graphs (Fig. 4d). In all cases, the executed SQL queries are accessible and readable, allowing experts

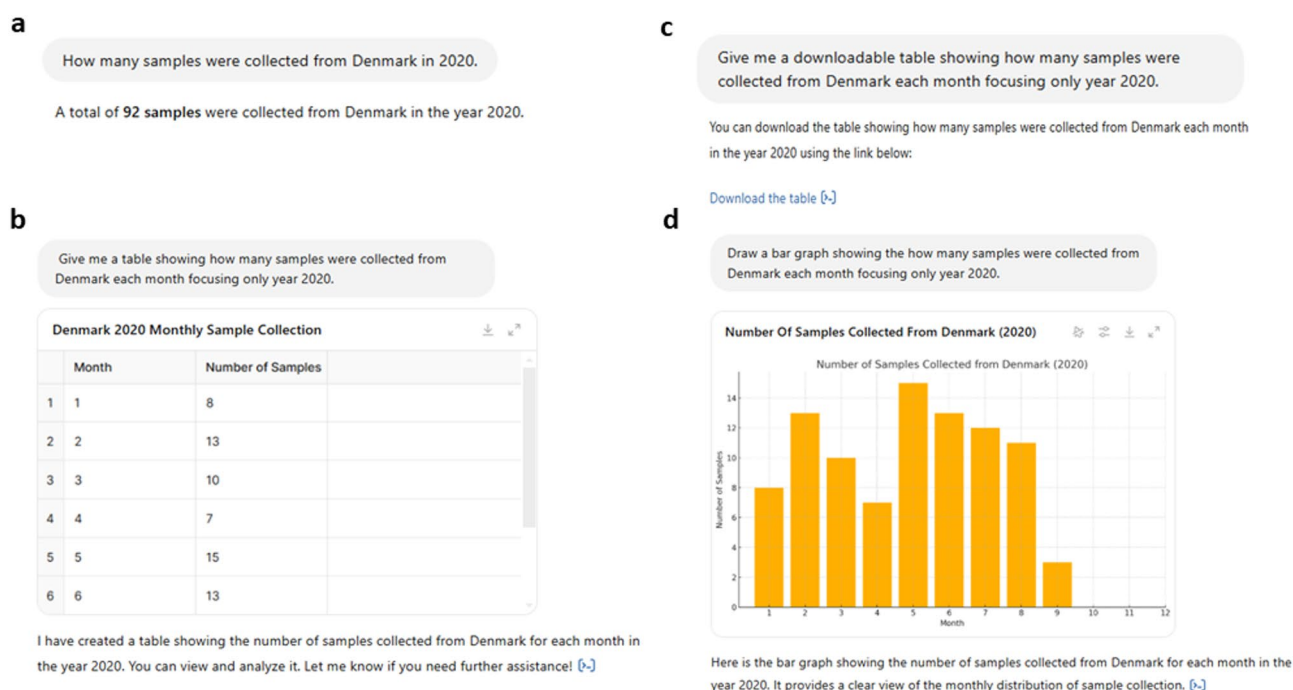


Fig. 4. Various outputs of SewageGPT web interface. The answer to a given question can be a text (a), table (b), downloadable table (c) or even a graph (d) depending on the user's request.

to review and verify their accuracy if needed. In addition to collecting data from the database, the web interface can also modify the collected data and generate and execute Python code, enabling the performance of statistical analyses.

The web interface has limitations regarding the size of tables that can be visualised or used for downstream analysis. In such cases, the question may need to be modified, or SewageGPT can be used solely to generate the SQL query, which can then be executed separately using an interface capable of handling larger datasets. For responses to our test questions generated by the web interface, refer to see Supplementary Data S4.

Evaluation of answers

The OpenAI/Anthropic API was utilised for programmatic evaluation, enabling precise specification of the LLM models (gpt-3.5-turbo, gpt-4o-2024-05-13, claude-opus-4-20250514 and claude-sonnet-4-20250514) and ensuring that each question is addressed independently, without interference from previous queries. The temperature parameter by LLM's influences the randomness and diversity of the answer, the higher value encourages creativity. In our case, it was set to zero to get more precise and predictable answers. Questions of varying difficulty were selected, and a list of these questions and the correct answers can be found in Table 1. The real-life questions were chosen such that each table in the database is required for at least one of the questions. Each question was asked 10 times to test the reproducibility of the answers (Fig. 5).

Execution accuracy, which measures if the model-generated SQL query produces the same correct result, was used to evaluate the performance of the LLM. This metric is better than the exact set match accuracy that matches the queries lexically¹⁹, as multiple valid SQL queries with minor differences can exist for a given question. Even the execution accuracy can be misleading for example when the query asks for the top rows and there is a tie in the top position (e.g. two samples have the same high value). If it is not specified exactly how to handle these ties then the model may return all ties or only one so in these situations also human evaluation of the SQL query is necessary.

There is a possibility to provide example questions and answers in the prompt ensuring more precise answering of LLM for hard to answer questions. We did not apply any helper predefined questions related to the question used for evaluation as this would falsify the results. Only custom-made function examples were utilized as their syntax clarification is inevitable. The clear schema, table and column definitions were enough for the gpt-4o-2024-05-13 model to give correct answers to the test questions and outperformed the other tested models.

Figure 6 illustrates the responses generated by the GPT-4o to a set of example questions for which the relevant content is absent from the database. When provided with detailed background information, the LLM is sometimes able to answer such questions correctly. However, if the background context is insufficient or unhelpful, the LLM may generate queries that lead to partially correct responses -or, in some cases, entirely incorrect or misleading answers. These examples highlight the importance of being aware of the database content or first asking the LLM about the contents of the database. By summarizing the available information, the LLM can help users better understand the database, making it easier to formulate questions that are actually answerable.

Discussion

In this study, we utilised a complex database of metagenomic sequencing data from sewage samples to assess the text-to-SQL capabilities of LLMs. The performance difference between ChatGPT 3.5 and GPT-4o was substantial, clearly showing the improvement in the newer model. Anthropic's models were outperformed by GPT-4o on our test set. Although future models are anticipated to offer even better performance, the current GPT-4o model already proves valuable, correctly answering the test questions in our analysis. Further improvements could be realised by incorporating example queries into the prompts; however, the goal here was to evaluate the model without such assistance.

The text-to-SQL functionality of GPT-4o facilitates rapid, low-effort experimentation, laying the groundwork for further refinements and optimizations as needed. This feature is valuable for experts, as it accelerates the process of constructing SQL queries. By building on and refining the initial queries generated by GPT-4o, experts can achieve more precise results more efficiently than if starting from scratch. For highly complex questions that the LLM cannot directly address, breaking the problem into smaller, more manageable components often leads to better outcomes (see Supplementary Data S5). Simplifying the problem allows the LLM to make meaningful contributions, though expert oversight remains crucial for tasks where accuracy and performance are critical. However, we do foresee that the feature will be valuable especially for non-experts enabling them to do data-exploration and simple analyses without the bottlenecks of needing experts. A concern here can naturally be that if the LLM provides incorrect answers, non-experts might draw the wrong conclusions. The true power of the text-to-sql approach is not that it answers simple questions in one shot. Rather in the hands of experts—experts of the discipline from which the data is derived, but not necessarily in programming—it can be used as a programmatically skilled assistant. The LLM excels at brainstorming and testing unconventional ideas that might otherwise go unexplored due to the effort required. This enables users to concentrate on problem-solving rather than query formulation. However, using LLM to ask database-related questions is less effective for users unfamiliar with the database structure, as their questions are prone to ambiguities so the answers are not necessarily correct. Ideally, users should first familiarize themselves with the database content, a task for which LLM is well-suited as it can summarize relevant information, helping to formulate answerable questions.

Data scientists typically do not just write SQL queries; they also execute, analyse the results, and iterate on these queries to refine them until the desired results are achieved¹⁹. Advanced LLM models have the capability to interact with databases and refine queries iteratively if syntactical errors occur. However, human review of the final results is essential, especially if the questions are not very detailed or ambiguous columns are in the database.

No	Question	SQL query	Answer
1	How many samples derived from Budapest?	SELECT COUNT(*) AS num_samples FROM distilled.meta m INNER JOIN distilled.location l ON m.location_id=l.id WHERE l.city = 'Budapest'	26
2	What was the lowest pH of sewage samples?	SELECT MIN(ph) AS lowest_ph FROM distilled.meta WHERE ph IS NOT NULL	6.5
3	Which country does the sewage samples come from with the highest temperature?	SELECT l.country, m.temperature FROM distilled.meta m INNER JOIN distilled.location l ON m.location_id=l.id WHERE m.temperature = (SELECT MAX(temperature) FROM distilled.meta WHERE temperature IS NOT NULL);	Hungary
4	Which antibiotic resistance gene, contributing to resistance through enhanced efflux mechanisms, is most abundant in the earliest sample collected from Budapest?	SELECT gr.name, gr.clr FROM distilled.gene_resfinder gr JOIN distilled.gene_resfinder_clr grc ON gr.id = grc.gene_resfinder_id JOIN distilled.meta m ON grc.meta_id = m.id JOIN distilled.location l ON m.location_id = l.id WHERE l.city = 'Budapest' AND gr.increased_efflux = TRUE AND m.collection_date = (SELECT MIN(collection_date) FROM distilled.meta m JOIN distilled.location l ON m.location_id = l.id WHERE l.city = 'Budapest') ORDER BY gr.clr DESC LIMIT 1;	tet(C)_2_ AY046276
5	How many samples contain antibiotic resistance genes linked to Amoxicillin resistance?	SELECT COUNT(DISTINCT gc.meta_id) AS num_samples FROM distilled.antibiotic_resfinder ar INNER JOIN distilled.gene_resfinder_clr gc ON ar.gene_resfinder_id = gc.gene_resfinder_id WHERE ar.antibiotic = 'Amoxicillin';	238
6	When was the sample with the highest clr value for the "msr(E)_1_FR751518" antimicrobial resistance gene collected in Copenhagen?	SELECT m.collection_date FROM distilled.gene_resfinder_clr grc JOIN distilled.meta m ON grc.meta_id = m.id JOIN distilled.gene_resfinder gr ON grc.gene_resfinder_id = gr.id JOIN distilled.location l ON m.location_id = l.id WHERE l.city = 'Copenhagen' AND gr.name = 'msr(E)_1_FR751518' ORDER BY grc.clr DESC LIMIT 1;	2019-05-27
7	What percentage of the sewage plants submitted samples confirmed to contain <i>Bacteroides fragilis</i> through metagenomic sequencing?	SELECT ROUND((COUNT(DISTINCT l.plant)::decimal / (SELECT COUNT(DISTINCT plant) FROM distilled.location) * 100), 2) AS percentage FROM distilled.mag_presence_absence mpa JOIN distilled.location l ON mpa.location_id = l.id JOIN distilled.tax_gtdb tg ON mpa.tax_gtdb_id = tg.id WHERE tg.name = 'Bacteroides fragilis' AND mpa.is_present = TRUE;	14.29%
8	How many contigs have a coverage greater than 10?	SELECT COUNT(*) AS num_contigs FROM distilled.contig WHERE coverage > 10;	58,636
9	How many contigs from plasmids were found in samples collected from Budapest?	SELECT COUNT(*) AS num_plasmid_contigs FROM distilled.contig c INNER JOIN distilled.location l ON c.location_id = l.id WHERE c.possible_source = 'plasmid' AND l.city = 'Budapest';	67,417
10	How many contigs contain the 'miaB' gene?	SELECT COUNT(*) AS num_contigs FROM distilled.gene_annotation ga WHERE ga.gene = 'miaB'	1488
11	How many species were confirmed by the "Gruppo HERA" sewage plant?	SELECT COUNT(DISTINCT mpa.tax_gtdb_id) FROM distilled.mag_presence_absence AS mpa INNER JOIN distilled.location l ON mpa.location_id = l.id WHERE l.plant = 'Gruppo HERA' AND mpa.is_present = TRUE;	298
12	How many MAGs do we have with contamination below 10% and GC content greater than 0.6?	SELECT COUNT(*) AS num_mags FROM distilled.checkm2 cm WHERE cm.contamination < 10 AND cm.gc_content > 0.6;	1006
13	How many samples from Budapest tested positive for the virulence gene "STp" through qPCR detection?	SELECT COUNT(*) FROM distilled.qpcr qpcr JOIN distilled.meta m ON qpcr.meta_id = m.id JOIN distilled.location l ON m.location_id = l.id JOIN distilled.qpcr_target qt ON qpcr.qpcr_target_id = qt.id WHERE l.city = 'Budapest' AND qt.target = 'STp' AND qpcr.is_present = TRUE;	7
14	How many MAGs are part of the Unified Human Gastrointestinal Genome?	SELECT COUNT(*) AS num_mags FROM distilled.gtdb g INNER JOIN distilled.tax_gtdb t ON g.tax_gtdb_id = t.id WHERE t.human_microbiome = TRUE;	240
15	By which sewage plant was the presence of <i>Bacteroides fragilis</i> confirmed by metagenomic sequencing?	SELECT l.plant FROM distilled.mag_presence_absence mpa JOIN distilled.location l ON mpa.location_id = l.id JOIN distilled.tax_gtdb t ON mpa.tax_gtdb_id = t.id WHERE t.name = 'Bacteroides fragilis' AND mpa.is_present = TRUE;	Gruppo HERA

Table 1. List of questions utilized for assessing LLM performance. The table includes the questions and the corresponding SQL queries that yield the correct answers.

This sewage dataset consists only of publicly available, non-sensitive data, mitigating data security concerns. However, hosting the largest and most advanced LLMs locally is not yet affordable to many raising privacy issues when analysing sensitive data with ChatGPT. For cases with security concerns, other open-source models that can be run locally such as qwen2.5-coder²⁰ or, deepseek-coder-v2²¹ may be used, and are subject of our further

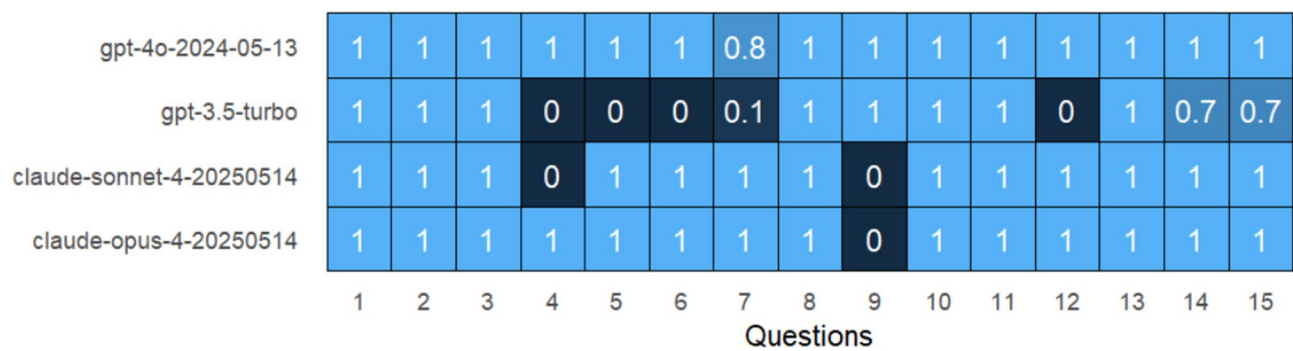


Fig. 5. Accuracy performance of LLM models. Fifteen questions (Table 1.) were posed to four LLM models (gpt-4o-2024-05-13, gpt-3.5-turbo, claude-opus-4-20250514 and claude-sonnet-4-20250514). The figures on the tiles represent the accuracy, which was calculated after each question was asked ten times.

studies. Although their performance is close to that of the more advanced LLMs, their accuracy still lags behind as of now but it is also expected to be improved with ongoing developments.

Conclusion

In summary, the current state-of-the-art LLMs with direct database connection capabilities are sufficiently advanced to significantly reduce the time needed to create queries and visualize results. The user-friendly setups also make it relatively easy to fine-tune the LLM for databases containing public data. SewageGPT is accessible online at: <https://chatgpt.com/g/g-679b5d7678e0819181f1da4d0b3b625c-sewagegpt>.



Fig. 6. Representative questions illustrating issues related to missing data. The displayed answers correspond to queries where the required information is not available in the database. Evaluative comments are provided on the right-hand side.

Data availability

The dataset supporting the conclusions of this article are included within the article (and its Supplementary Information files). The datasets analysed during the current study are available in the European Nucleotide Archive (ENA) repository under accession code PRJEB68319 (<https://www.ebi.ac.uk/ena/browser/view/PRJEB68319>). Additional information is available on GitHub (<https://github.com/pkrisz5/SewageGPT>).

Received: 18 March 2025; Accepted: 10 November 2025

Published online: 22 November 2025

References

1. Wilkinson, M. D. et al. The FAIR guiding principles for scientific data management and stewardship. *Sci. Data*. **3**, 160018 (2016).
2. Yu, T. et al. Spider: a large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL Task. Preprint at <https://doi.org/10.48550/arXiv.1809.08887> (2019).
3. Zhong, V., Lewis, M., Wang, S. I. & Zettlemoyer, L. Grounded adaptation for zero-shot executable semantic parsing. Preprint at (2021). <https://doi.org/10.48550/arXiv.2009.07396>
4. Pourreza, M. & Rafiei, D. D. I. N. S. Q. L. Decomposed in-context learning of text-to-SQL with self-correction. Preprint at. <https://doi.org/10.48550/arXiv.2304.11015> (2023).
5. Clausen, P. T. L. C., Aarestrup, F. M. & Lund, O. Rapid and precise alignment of raw reads against redundant databases with KMA. *BMC Bioinform.* **19**, 307 (2018).
6. Florensa, A. F., Kaas, R. S., Clausen, P. T. L. C., Aytan-Aktug, D. & Aarestrup, F. M. ResFinder: an open online resource for identification of antimicrobial resistance genes in next-generation sequencing data and prediction of phenotypes from genotypes. *Microb. Genomics*. **8**, 000748 (2022).
7. Becsei, Á. et al. Time-series sewage metagenomics distinguishes seasonal, human-derived and environmental microbial communities potentially allowing source-attributed surveillance. *Nat. Commun.* **15**, 7551 (2024).
8. Fang, Z. et al. PPR-Meta: a tool for identifying phages and plasmids from metagenomic fragments using deep learning. *GigaScience* **8**, giz066 (2019).
9. Seemann, T. Prokka: rapid prokaryotic genome annotation. *Bioinformatics* **30**, 2068–2069 (2014).
10. Chklovskii, A., Parks, D. H., Woodcroft, B. J. & Tyson, G. W. CheckM2: a rapid, scalable and accurate tool for assessing microbial genome quality using machine learning. *Nat. Methods*. **20**, 1203–1212 (2023).
11. Bowers, R. M. et al. Minimum information about a single amplified genome (MISAG) and a metagenome-assembled genome (MIMAG) of bacteria and archaea. *Nat. Biotechnol.* **35**, 725–731 (2017).
12. Olm, M. R., Brown, C. T., Brooks, B. & Banfield, J. F. dRep: a tool for fast and accurate genomic comparisons that enables improved genome recovery from metagenomes through de-replication. *ISME J.* **11**, 2864–2868 (2017).
13. GTDB-Tk. A toolkit to classify genomes with the genome taxonomy database | bioinformatics | Oxford academic. <https://academic.oup.com/bioinformatics/article/36/6/1925/5626182?login=false> (2019).
14. Larsson, D. G. J., Flach, C. F. & Laxminarayan, R. Sewage surveillance of antibiotic resistance holds both opportunities and challenges. *Nat. Rev. Microbiol.* **21**, 213–214 (2023).
15. Hendriksen, R. S. et al. Global monitoring of antimicrobial resistance based on metagenomics analyses of urban sewage. *Nat. Commun.* **10**, 1124 (2019).
16. Munk, P. et al. Genomic analysis of sewage from 101 countries reveals global landscape of antimicrobial resistance. *Nat. Commun.* **13**, 7251 (2022).
17. Izquierdo-Lara, R. et al. Monitoring SARS-CoV-2 circulation and diversity through community wastewater Sequencing, the Netherlands and Belgium. *Emerg. Infect. Dis.* **27**, 1405–1415 (2021).
18. Izquierdo-Lara, R. W. et al. Rise and fall of SARS-CoV-2 variants in rotterdam: comparison of wastewater and clinical surveillance. *Sci. Total Environ.* **873**, 162209 (2023).
19. Pourreza, M. *Text-to-SQL Systems in the Era of Advanced Large Language Models* (University of Alberta, 2024).
20. Hui, B. et al. Qwen2.5-Coder Technical Report. Preprint at (2024). <https://doi.org/10.48550/arXiv.2409.12186>
21. DeepSeek-AI et al. DeepSeek-Coder-V2: Breaking the Barrier of Closed-Source Models in Code Intelligence. Preprint at (2024). <https://doi.org/10.48550/arXiv.2406.11931>

Acknowledgements

We would like to extend our gratitude to the employees of Fővárosi Csatornázási Művek (Budapest, Hungary) for their efforts in conducting the sewage sampling. This work was supported by the European Union's Horizon 2020 research and innovation programme under grant agreements No. 874735 (VEO) National Research, Development, and Innovation Office of Hungary within the framework of the MILAB Artificial Intelligence National Laboratory (RRF-2.3.1-21-2022-00004) and by the National Institute Of Allergy And Infectious Diseases of the National Institutes of Health under Award Number U24AI183840. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Author contributions

K.P. constructed and evaluated the GPT interface. Á.B., P.M. performed metagenomic data analyses. J.S. carried out the design of the database schema and the building of the PostgreSQL database. KP, ÁB, JS and DV drafted the manuscript with revisions provided by P.M., F.M.A. and I.Cs. All authors read and approved the final manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-025-28341-7>.

Correspondence and requests for materials should be addressed to K.P.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025