

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

Received: 18 March 2025

Accepted: 10 November 2025

Published online: 22 November 2025

Cite this article as: Becsei Á., Stéger J., Visontai D. *et al.* A case study using sewage metagenomic data for assessment of text-to-SQL capabilities in large language models. *Sci Rep* (2025). <https://doi.org/10.1038/s41598-025-28341-7>

Ágnes Becsei, József Stéger, Dávid Visontai, Patrick Munk, Frank M. Aarestrup, István Csabai & Krisztián Papp

We are providing an unedited version of this manuscript to give early access to its findings. Before final publication, the manuscript will undergo further editing. Please note there may be errors present which affect the content, and all legal disclaimers apply.

If this paper is publishing under a Transparent Peer Review model then Peer Review reports will publish with the final article.

1 **A Case Study Using Sewage Metagenomic Data for Assessment of Text-** 2 **to-SQL Capabilities in Large Language Models**

3

4 Ágnes Becsei^{1#}, József Stéger^{1#}, Dávid Visontai¹, Patrick Munk², Frank M.5 Aarestrup², István Csabai^{1†}, Krisztián Papp^{1†*}

6

7 1 Department of Physics of Complex Systems, ELTE Eötvös Loránd University,

8 Budapest, Hungary.

9 2 National Food Institute, Technical University of Denmark, Lyngby, Denmark

10

11 # These authors contributed equally

12 †These authors share senior authorship

13 * Corresponding author: Krisztián Papp (krisztian.papp@ttk.elte.hu)

14

15 **Abstract**

16 Relational databases offer an efficient solution for storing and retrieving complex
17 data sets, yet the requirement for SQL programming expertise presents a
18 significant challenge for many life science users. We explore whether a cutting-
19 edge large language model can effectively translate plain English queries into
20 SQL scripts (Text-to-SQL), thereby simplifying database interaction and
21 eliminating the typical usage barriers. A complex database comprising 19
22 interconnected tables of metagenomic analyses from 239 sewage samples
23 across five European cities was available. A large language model was provided
24 with details of the database's structure and background information on its
25 contents. We evaluated the functionalities of this "SewageGPT" tool and
26 assessed the accuracy of its responses to complex questions and visualisation of
27 results. Providing a detailed description of the database enabled SewageGPT to

1 accurately respond to complex inquiries, accelerating the database querying
2 process. Knowledge of the database content proved beneficial, as it minimized
3 the risk of ambiguities in queries; however, ambiguities can lead to incorrect
4 responses. Therefore, human oversight remains crucial, particularly for questions
5 that lack detail or involve ambiguities. The integration of state-of-the-art large
6 language models with direct database connectivity substantially enhances the
7 efficiency of query generation, statistical analysis and visualization of the results.

8 **Keyword**

9 Text-to-SQL, Large Language Model, metagenomics

10

11 **Introduction**

12

13 The FAIR Guiding Principles promote that scientific data should be Findable,
14 Accessible, Interoperable, and Reusable, making them reusable for both
15 machines and humans. ¹. When the size of a structured biology-related dataset
16 increases and reaches a certain limit, then relational databases provide an
17 efficient way to store and retrieve data. Databases are valuable, yet the need for
18 SQL programming skills to execute complex queries can pose a challenge for
19 many users. An easy-to-use dashboard web interface can provide solutions for
20 predefined, often-used queries but is not flexible enough to answer the multiple
21 specific questions a broader user community might have. Comprehending a
22 complex database with several interconnected tables can be challenging and
23 time-consuming, especially for the first time or if it is not used regularly, thus
24 excluding several potential users. An attractive solution would be to use natural
25 language questions - plain English (or another language) - translated
26 automatically to SQL query in the background (text-to-SQL). This approach would
27 allow users to access information from complex databases without SQL

1 programming skills or in-depth knowledge of database schemas. The text-to-SQL
2 field has seen substantial advancements over the past few years. Thus, in just
3 three years, execution accuracy on the Spider ² large-scale cross-domain
4 dataset—a widely used benchmark—rose from 53.5 ³ to 85.3 ⁴. While these
5 benchmarks are valuable, they can be somewhat artificial, as the databases
6 often contain data that aligns more closely with everyday concepts. This makes
7 them easier for large language models (LLMs) to comprehend, leveraging their
8 strengths in processing familiar contexts.

9 We aimed to make a real-life evaluation of the usability of a state-of-the-art large
10 language model for querying a database containing complex multi-level data
11 from metagenomic analysis of sewage samples. For this specific database, the
12 access barrier becomes even more significant when not only researchers but
13 also public health professionals need to access and analyse pathogen-related
14 data, as they may have limited experience with programmatic access.

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

26 **Methods**

1 *Metagenomics analysis*

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

9 The following analysis and data are partly derived from the study by Becsei and
10 Fuschi et al. ⁷. For contig level analysis trimmed reads were assembled into
11 contigs for each sample (single-sample assembly) and for each sample collection
12 site (co-assembly). Then, the resulting contigs were classified according to their
13 potential origin using the PPR-meta (v. 1.1) tool ⁸. Potential genes were identified
14 within them using Prokka (v1.14.6) ⁹ and subsequently, the contigs were used to
15 reconstruct metagenome-assembled genomes (MAGs). The quality of these
16 MAGs was evaluated using the CheckM2 (v1.0.1) tool ¹⁰, and for further steps, we
17 retained only medium- and high-quality MAGs based on the MIMAG (Minimum
18 Information for Metagenome-Assembled Genomes) criteria ¹¹. The remaining
19 MAGs were dereplicated using the dRep (v3.4.2) tool ¹² and taxonomically
20 classified with GTDB-tk (v2.4.0 with GTDB r220 as reference) ¹³. Quality-filtered
21 reads were aligned to the taxonomically classified bacterial MAG set, and the
22 average alignment depth for each MAG in each sample was calculated. These
23 depth values were then transformed into CLR values to quantify the relative
24 abundance of MAGs in each sample. More details on assembly and MAG-level
25 analysis steps can be found in the study by Becsei and Fuschi et al. ⁷.
26 Additionally, the presence or absence of certain bacteria or bacterial pathogens

1 was assessed using two methods. We conducted qPCR analysis targeting 21
2 selected bacterial pathogens, and a bacterium was considered present in a given
3 sample or sampling site if its MAG could be reconstructed with at least medium
4 quality in that sample or sampling site.

5

6 *Configuration of the Web-Based Customized ChatGPT Interface*

7

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

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

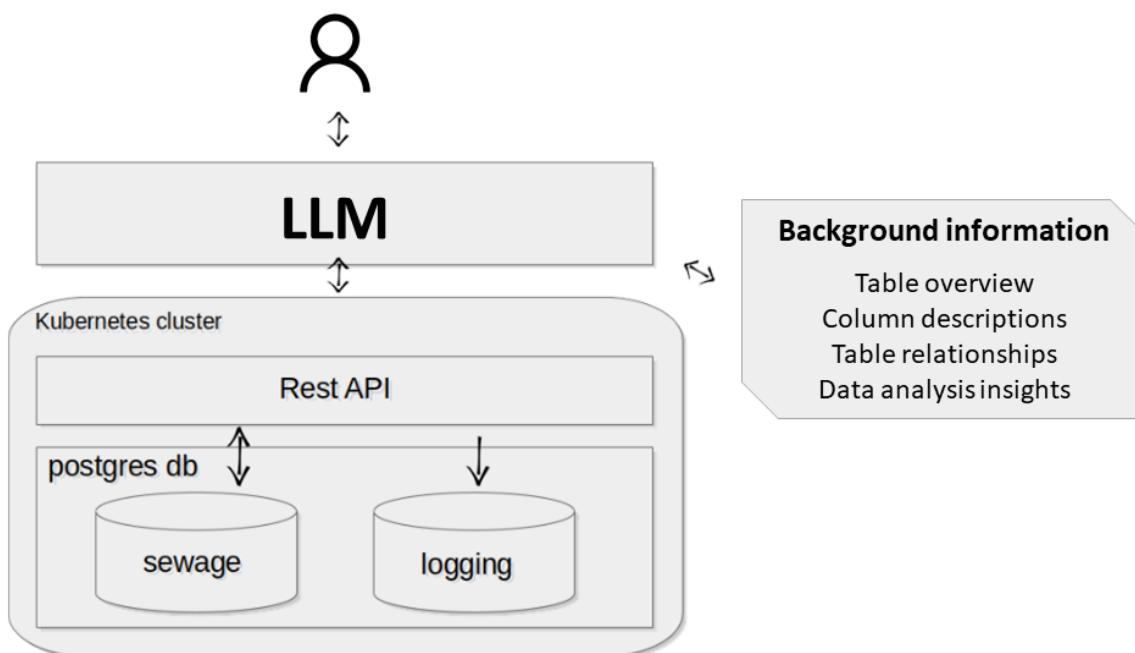
26 In the KNOWLEDGE section, the uploaded files contained the database schema
27 (see Supplementary Data S3), detailed descriptions of the tables and columns

1 (see Supplementary Tables S1-S2), and background information on data
 2 collection (Becsei et al., submitted 2025).

3
 4 *SQL database-LLM integration architecture*

5
 6 The architecture of the components is sketched in Figure 1. For cybersecurity
 7 reasons, in the PostgreSQL database where public sewage data reside, native
 8 database connections are not exposed. A REST API is deployed to handle proxy
 9 requests, equipped with a read-only database access role and preset resource
 10 limits to prevent system overloads. Query running times are limited. Additionally
 11 in the database, a logging table has been created in a separate schema, where
 12 every query flowing through the REST API is logged with metadata, including
 13 when the query started, how much time it took to complete and if it had been
 14 erroneous what the error message was. This table is utilized for further
 15 enhancements to the system and may become a valuable resource for fine-
 16 tuning of the text-to-SQL module.

17



18

1
2
3
4
5
6
7
8

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

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 Figure 2.

```

{
  "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": {}
  }
}

```

9
10

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

11
12
13
14
15

Results

1 *Dataset and database*

2

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

11

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

21

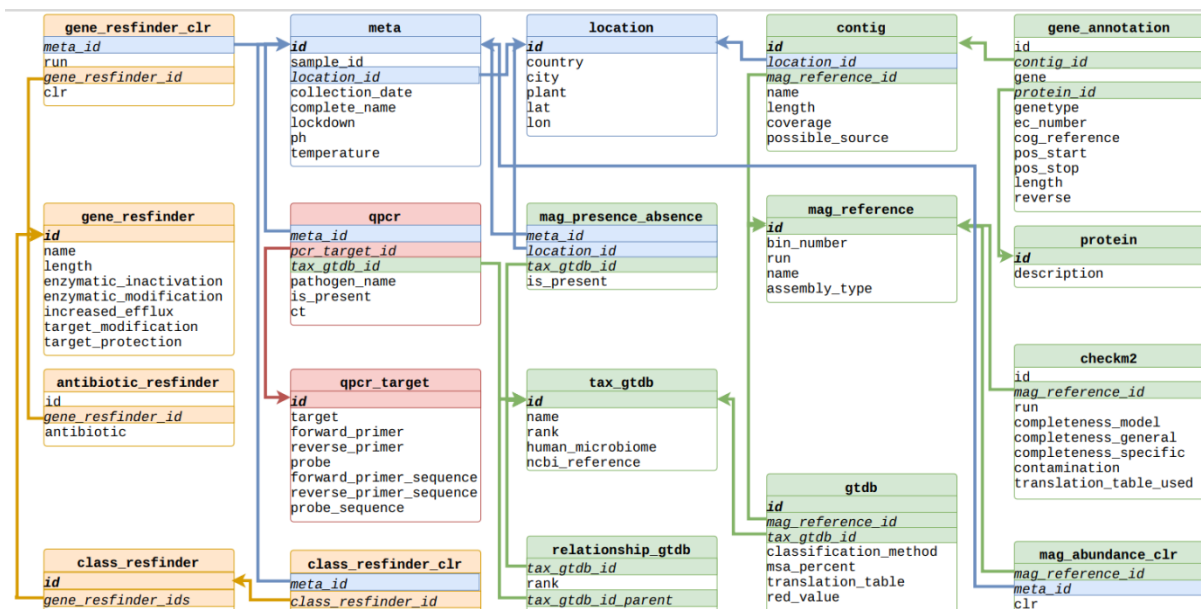


Figure 3: Schema of the database. Rectangles represent the tables in the database with the column names and their connection to each other.

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

1 and database integration, documentation, interface accessibility, and the ability
2 to create figures, and basic statistical calculations.

3

4 *Prompting*

5

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

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

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

- 21 ● If a column includes percentage values, the description should clarify
22 whether 100% corresponds to a value of 100 or 1 in the table.
- 23 ● Multiple methods exist for extracting the maximum value from a column.
24 One option is to sort the column in descending order and select the top
25 entry, or to use the MAX SQL function. If not specified, LLM may choose
26 either method. However, if the column includes empty cells, these
27 approaches may yield different outcomes. Sorting in descending order

1 ('ORDER DESC') treats empty cells as the highest values, leading to an
2 incorrect result when selecting the first entry. To ensure accurate results
3 in such cases, it helps to include a note in the column description, such as:
4 'Some cells are empty; please remove these before ordering this column.'

5 ● Right naming of columns can help LLM models figuring out the connection
6 between tables. Although it may be evident from the formal schema
7 definition, it proved to be a good practice that the column name
8 description contain if it is a primary key that serves as unique identifiers
9 for each row in the table or if it is a foreign key column in a table that
10 points to the column with unique values in another table creating cross-
11 referencing between the tables. We need to avoid cases when multiple
12 columns represent very close semantic meaning as these cause
13 ambiguities.

14 ● Description of date containing columns should contain information about
15 date format

16 ● Typos, misleading names or abbreviations in the names of columns may
17 pose a problem. Even if such an incorrect column name is used
18 consistently in the schema, LLM sometimes uses the grammarly corrected
19 form in SQL query, generating a syntactically wrong query.

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

23
24 Beyond the schema description, additional information can be provided to the
25 LLM to improve accuracy when addressing questions prone to incorrect answers.
26 This information might include detailed explanations of specific data types or
27 example SQL queries for particular tasks. Providing example queries is especially

1 useful when a specific syntax is required for custom-made SQL functions. These
2 examples enable LLM to handle similar tasks in the future.

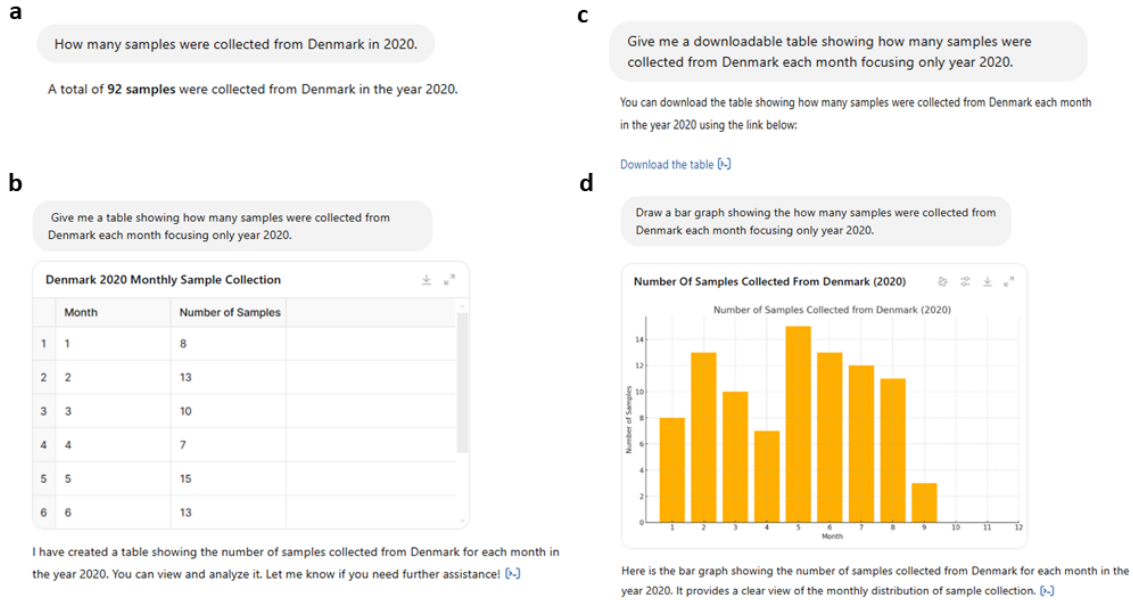
3

4 Functionalities

5

6 The tested models can be accessed either through its web interface or via an
7 OpenAI/Anthropic API. Only the customized ChatGPT web interface includes an
8 action feature that enables direct connection with a database, whereas
9 Anthropic's web interface does not support communication with SQL databases
10 and can therefore be tested only via the Anthropic API. The customized ChatGPT
11 user interface and the OpenAI API reach the same background information yet
12 the handling of the information and their capabilities are different. ChatGPT
13 interface can automatically correct invalid SQL queries, whereas the basic
14 OpenAI API setup lacks this functionality. The web interface is user-friendly, even
15 for non-experts, and is directly connected to the sewage database. User queries
16 are translated into SQL statements, submitted to the database through its REST
17 API, and the results are presented in various formats: as text (Fig4.a), tables
18 (Fig4.b), downloadable tables (Fig4.c), or visualised as graphs (Fig4.d). In all
19 cases, the executed SQL queries are accessible and readable, allowing experts to
20 review and verify their accuracy if needed. In addition to collecting data from the
21 database, the web interface can also modify the collected data and generate and
22 execute Python code, enabling the performance of statistical analyses.

23



1

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

5

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

12

13

14 *Evaluation of answers*

15

16 The OpenAI/Anthropic API was utilised for programmatic evaluation, enabling
 17 precise specification of the LLM models (gpt-3.5-turbo, gpt-4o-2024-05-13,
 18 *claude-opus-4-20250514* and *claude-sonnet-4-20250514*) and ensuring that each
 19 question is addressed independently, without interference from previous queries.

1 The temperature parameter by LLM's influences the randomness and diversity of
 2 the answer, the higher value encourages creativity. In our case, it was set to zero
 3 to get more precise and predictable answers. Questions of varying difficulty were
 4 selected, and a list of these questions and the correct answers can be found in
 5 Table 1. The real-life questions were chosen such that each table in the database
 6 is required for at least one of the questions. Each question was asked 10 times to
 7 test the reproducibility of the answers (Fig. 5).

8

gpt-4o-2024-05-13	1	1	1	1	1	1	0.8	1	1	1	1	1	1	1	1
gpt-3.5-turbo	1	1	1	0	0	0	0.1	1	1	1	1	0	1	0.7	0.7
claude-sonnet-4-20250514	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1
claude-opus-4-20250514	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Questions														

9

10 *Figure 5. Accuracy performance of LLM models. Fifteen questions (Table 1.) were*
 11 *posed to four LLM models (gpt-4o-2024-05-13, gpt-3.5-turbo, claude-opus-4-*
 12 *20250514 and claude-sonnet-4-20250514). The figures on the tiles represent the*
 13 *accuracy, which was calculated after each question was asked ten times.*

14

15 Execution accuracy, which measures if the model-generated SQL query produces
 16 the same correct result, was used to evaluate the performance of the LLM. This
 17 metric is better than the exact set match accuracy that matches the queries
 18 lexically ¹⁹, as multiple valid SQL queries with minor differences can exist for a
 19 given question. Even the execution accuracy can be misleading for example
 20 when the query asks for the top rows and there is a tie in the top position (e.g.
 21 two samples have the same high value). If it is not specified exactly how to
 22 handle these ties then the model may return all ties or only one so in these
 23 situations also human evaluation of the SQL query is necessary.

1

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

9

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

Which bacteria were detected by ELISA method?

The ELISA method is not used in this dataset. Detection of bacteria was performed using:

1. **qPCR** – for detecting specific pathogens (see `qpcr` table).
2. **Metagenomic sequencing** – for identifying bacterial species from DNA sequences and reconstructing MAGs (see `mag_presence_absence` table).

Therefore, **no bacteria were detected using the ELISA method** in this dataset.

Correct answer. ELISA method was not used. The LLM did not search in the database, answer derived from provided background information.

Which sample from a sewage treatment plant in Africa has been confirmed to contain *Bacteroides fragilis* based on metagenomic sequencing?

The dataset only includes sewage samples from **five European cities**: Copenhagen, Rotterdam, Budapest, Rome, and Bologna. There are **no samples from Africa**.

Therefore, **no sample from a sewage treatment plant in Africa** is available in this dataset, and consequently **none can be confirmed to contain *Bacteroides fragilis*** based on metagenomic sequencing.

Correct answer. Samples derived only from European cities. The LLM did not search in the database, answer derived from provided background information.

How many samples have a pH below 7.6 and a yellowish color?

Talked to k8plex-veo.vo.elte.hu ^

The following was shared:

Privacy policy

```
"SELECT COUNT(*) FROM distilled.meta m WHERE m.ph < 7.6 AND m.complete_name
sql: ILIKE ANY (ARRAY['%yellow%', '%yellowish%']); -- How many samples have a pH
below 7.6 and a yellowish color?"
```

There are **no samples** in the dataset that have both a pH below 7.6 and a yellowish color mentioned in their metadata.

Not precise answer. Information about the color of samples were not collected. The LLM searched for „yellow” or „yellowish” words in „complete_name” column, but the names of samples are only ID’s. Correct answer would be that information about the color is not available.

Which sample was collected from an area with a chicken farm?

Talked to k8plex-veo.vo.elte.hu ^

The following was shared:

Privacy policy

```
"SELECT complete_name FROM distilled.meta WHERE complete_name ILIKE "%chicken
sql: farm%"; -- Which sample was collected from an area with a chicken farm?"
```

None of the sample names or metadata in the dataset explicitly mention being collected from an area with a chicken farm.

Not precise answer. Information about the the presence of chicken farms in the area was not collected. The LLM searched for „chicken farm” words in the „complete_name” column, but the names of samples are only ID’s. Correct answer would be that information about the presence of chicken farm are not available.

According to metagenomic sequencing data, were any oral bacteria detected in the samples?

Talked to k8plex-veo.vo.elte.hu ^

The following was shared:

Privacy policy

```
"SELECT DISTINCT t.name AS species_name\nFROM distilled.mag_presence_absence
mpa\nJOIN distilled.tax_gtdb t ON mpa.tax_gtdb_id = t.id\nWHERE mpa.is_present
= TRUE AND t.name ILIKE '%oral%';\n-- According to metagenomic sequencing
data, were any oral bacteria detected in the samples?"
```

Yes, based on the metagenomic sequencing data, the bacterium *Thiothrix litoralis*—which is known to inhabit various environments including oral microbiomes—was detected in the samples. This suggests that at least one oral-associated bacterial species was present.

Wrong answer. There is not any information in our database about if a bacteria present or not in the oral cavity. The LLM searched for bacterial name that contains the word „oral”. This is a complete failure here as the *Thiothrix litoralis* is an environmental bacteria and is not presence in the mouth.

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

6 **Discussion**

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

17 The text-to-SQL functionality of GPT-4o facilitates rapid, low-effort
18 experimentation, laying the groundwork for further refinements and
19 optimizations as needed. This feature is valuable for experts, as it accelerates
20 the process of constructing SQL queries. By building on and refining the initial
21 queries generated by GPT-4o, experts can achieve more precise results more
22 efficiently than if starting from scratch. For highly complex questions that the
23 LLM cannot directly address, breaking the problem into smaller, more
24 manageable components often leads to better outcomes (see Supplementary
25 Data S5). Simplifying the problem allows the LLM to make meaningful
26 contributions, though expert oversight remains crucial for tasks where accuracy

1 and performance are critical. However, we do foresee that the feature will be
2 valuable especially for non-experts enabling them to do data-exploration and
3 simple analyses without the bottlenecks of needing experts. A concern here can
4 naturally be that if the LLM provides incorrect answers, non-experts might draw
5 the wrong conclusions. The true power of the text-to-sql approach is not that it
6 answers simple questions in one shot. Rather in the hands of experts - experts of
7 the discipline from which the data is derived, but not necessarily in programming
8 - it can be used as a programmatically skilled assistant. The LLM excels at
9 brainstorming and testing unconventional ideas that might otherwise go
10 unexplored due to the effort required. This enables users to concentrate on
11 problem-solving rather than query formulation. However, using LLM to ask
12 database-related questions is less effective for users unfamiliar with the
13 database structure, as their questions are prone to ambiguities so the answers
14 are not necessarily correct. Ideally, users should first familiarize themselves with
15 the database content, a task for which LLM is well-suited as it can summarize
16 relevant information, helping to formulate answerable questions.

17 Data scientists typically do not just write SQL queries; they also execute, analyse
18 the results, and iterate on these queries to refine them until the desired results
19 are achieved ¹⁹. Advanced LLM models have the capability to interact with
20 databases and refine queries iteratively if syntactical errors occur. However,
21 human review of the final results is essential, especially if the questions are not
22 very detailed or ambiguous columns are in the database.

23

24 This sewage dataset consists only of publicly available, non-sensitive data,
25 mitigating data security concerns. However, hosting the largest and most
26 advanced LLMs locally is not yet affordable to many raising privacy issues when
27 analysing sensitive data with ChatGPT. For cases with security concerns, other

1 open-source models that can be run locally such as qwen2.5-coder ²⁰ or,
2 deepseek-coder-v2 ²¹ may be used, and are subject of our further studies.
3 Although their performance is close to that of the more advanced LLMs, their
4 accuracy still lags behind as of now but it is also expected to be improved with
5 ongoing developments.

6

7 **Conclusion**

8 In summary, the current state-of-the-art LLMs with direct database connection
9 capabilities are sufficiently advanced to significantly reduce the time needed to
10 create queries and visualize results. The user-friendly setups also make it
11 relatively easy to fine-tune the LLM for databases containing public data.

12 SewageGPT is accessible online at: [https://chatgpt.com/g/g-
13 679b5d7678e0819181f1da4d0b3b625c-sewagegpt](https://chatgpt.com/g/g-679b5d7678e0819181f1da4d0b3b625c-sewagegpt).

14

15 **Data availability statement**

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

22

23 **References**

24

- 1 1. Wilkinson, M. D. *et al.* The FAIR Guiding Principles for scientific data
2 management and stewardship. *Sci. Data* **3**, 160018 (2016).
- 3 2. Yu, T. *et al.* Spider: A Large-Scale Human-Labeled Dataset for Complex and
4 Cross-Domain Semantic Parsing and Text-to-SQL Task. Preprint at
5 <https://doi.org/10.48550/arXiv.1809.08887> (2019).
- 6 3. Zhong, V., Lewis, M., Wang, S. I. & Zettlemoyer, L. Grounded Adaptation for
7 Zero-shot Executable Semantic Parsing. Preprint at
8 <https://doi.org/10.48550/arXiv.2009.07396> (2021).
- 9 4. Pourreza, M. & Rafiei, D. DIN-SQL: Decomposed In-Context Learning of Text-
10 to-SQL with Self-Correction. Preprint at
11 <https://doi.org/10.48550/arXiv.2304.11015> (2023).
- 12 5. Clausen, P. T. L. C., Aarestrup, F. M. & Lund, O. Rapid and precise alignment
13 of raw reads against redundant databases with KMA. *BMC Bioinformatics* **19**,
14 307 (2018).
- 15 6. Florensa, A. F., Kaas, R. S., Clausen, P. T. L. C., Aytan-Aktug, D. & Aarestrup,
16 F. M. ResFinder – an open online resource for identification of antimicrobial
17 resistance genes in next-generation sequencing data and prediction of
18 phenotypes from genotypes. *Microb. Genomics* **8**, 000748 (2022).
- 19 7. Becsei, Á. *et al.* Time-series sewage metagenomics distinguishes seasonal,
20 human-derived and environmental microbial communities potentially allowing
21 source-attributed surveillance. *Nat. Commun.* **15**, 7551 (2024).
- 22 8. Fang, Z. *et al.* PPR-Meta: a tool for identifying phages and plasmids from
23 metagenomic fragments using deep learning. *GigaScience* **8**, giz066 (2019).
- 24 9. Seemann, T. Prokka: rapid prokaryotic genome annotation. *Bioinformatics* **30**,
25 2068–2069 (2014).

- 1 10. Chklovski, A., Parks, D. H., Woodcroft, B. J. & Tyson, G. W. CheckM2: a rapid,
2 scalable and accurate tool for assessing microbial genome quality using
3 machine learning. *Nat. Methods* **20**, 1203–1212 (2023).
- 4 11. Bowers, R. M. *et al.* Minimum information about a single amplified genome
5 (MISAG) and a metagenome-assembled genome (MIMAG) of bacteria and
6 archaea. *Nat. Biotechnol.* **35**, 725–731 (2017).
- 7 12. Olm, M. R., Brown, C. T., Brooks, B. & Banfield, J. F. dRep: a tool for fast and
8 accurate genomic comparisons that enables improved genome recovery from
9 metagenomes through de-replication. *ISME J.* **11**, 2864–2868 (2017).
- 10 13. GTDB-Tk: a toolkit to classify genomes with the Genome Taxonomy Database
11 | Bioinformatics | Oxford Academic.
12 [https://academic.oup.com/bioinformatics/article/36/6/1925/5626182?login=fa](https://academic.oup.com/bioinformatics/article/36/6/1925/5626182?login=false)
13 lse.
- 14 14. Larsson, D. G. J., Flach, C.-F. & Laxminarayan, R. Sewage surveillance of
15 antibiotic resistance holds both opportunities and challenges. *Nat. Rev.*
16 *Microbiol.* **21**, 213–214 (2023).
- 17 15. Hendriksen, R. S. *et al.* Global monitoring of antimicrobial resistance based on
18 metagenomics analyses of urban sewage. *Nat. Commun.* **10**, 1124 (2019).
- 19 16. Munk, P. *et al.* Genomic analysis of sewage from 101 countries reveals global
20 landscape of antimicrobial resistance. *Nat. Commun.* **13**, 7251 (2022).
- 21 17. Izquierdo-Lara, R. *et al.* Monitoring SARS-CoV-2 Circulation and Diversity
22 through Community Wastewater Sequencing, the Netherlands and Belgium.
23 *Emerg. Infect. Dis.* **27**, 1405–1415 (2021).
- 24 18. Izquierdo-Lara, R. W. *et al.* Rise and fall of SARS-CoV-2 variants in Rotterdam:
25 Comparison of wastewater and clinical surveillance. *Sci. Total Environ.* **873**,
26 162209 (2023).

- 1 19. Pourreza, M. Text-to-SQL Systems in the Era of Advanced Large Language
2 Models. (University of Alberta, 2024).
- 3 20. Hui, B. *et al.* Qwen2.5-Coder Technical Report. Preprint at
4 <https://doi.org/10.48550/arXiv.2409.12186> (2024).
- 5 21. DeepSeek-AI *et al.* DeepSeek-Coder-V2: Breaking the Barrier of Closed-Source
6 Models in Code Intelligence. Preprint at
7 <https://doi.org/10.48550/arXiv.2406.11931> (2024).

8

9 **Acknowledgements**

10 We would like to extend our gratitude to the employees of Fővárosi Csatornázási
11 Művek (Budapest, Hungary) for their efforts in conducting the sewage sampling.
12 This work was supported by the European Union's Horizon 2020 research and
13 innovation programme under grant agreements No. 874735 (VEO) National
14 Research, Development, and Innovation Office of Hungary within the framework
15 of the MILAB Artificial Intelligence National Laboratory (RRF-2.3.1-21-2022-
16 00004) and by the National Institute Of Allergy And Infectious Diseases of the
17 National Institutes of Health under Award Number U24AI183840. The content is
18 solely the responsibility of the authors and does not necessarily represent the
19 official views of the National Institutes of Health.

20

21 **Authors' contributions**

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

1

2 **Additional Information**3 *Competing interests*

4 The authors declare no competing interests.

5

6

7 *Table 1. List of questions utilized for assessing LLM performance.*

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, grc.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 grc.clr DESC LIMIT 1;	tet(C)_2_AY0462 76

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 Bacteroides fragilis 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;	58636
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';	67417
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 AS 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

1

2 *The table includes the questions and the corresponding SQL queries that yield*3 *the correct answers.*

4