# Real-World Data Large Language Model Assistive SQL Coding System

Vladimir Turzhitsky, PhD[1]; Varun K. Nomula, MS[1]; Yezhou Sun, MS[1]; Tesfagabir Meharizghi, MS[2]; Henry Wang, MS[2]; Aude Genevay, PhD[2]; Shinan Zhang, MS[2]; Tim Shear, MS[2], Andy Mitchell, AA[2]

[1]Merck & Co., Inc., Rahway, NJ, USA; [2]Amazon Web Services, Seattle, WA, USA

## Objectives

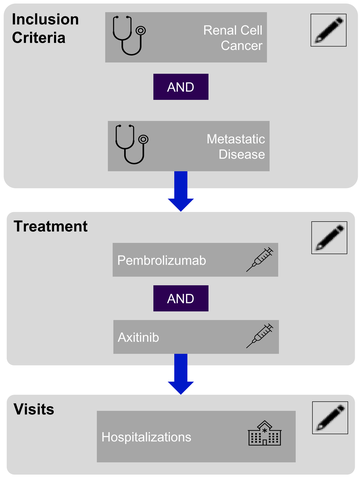
- To develop and evaluate a Large Language Model (LLM)-enabled text-to-SQL assistive programming system that accelerates and standardizes SQL generation for real-world data (RWD) analysis
- To characterize its methodological components and early performance on representative RWD tasks

**Takeaway**
An analyst-facing, metadata-aware SQL assistant can accelerate query drafting for claims and EHR data, but robust evaluation and human review remain essential for decision-grade evidence generation.

## Background

### The Challenge of Preserving Scientific Intent Across RWD Sources

- Real-world databases are heterogeneous, high-dimensional, and vendor-specific
- Common data models can simplify structure but may omit useful source-specific variables
- Cohort and treatment logic often depends on validated code lists and careful event timing
- Analysts need a faster path from natural-language questions to executable, reviewable SQL




**Typical analytic building blocks**

- Visits and care settings
- Inclusion / exclusion criteria
- Diagnoses, procedures, and treatment exposure
- Temporal relationships (before, after, during)
- Reusable validated code lists

## Methods

- Web-based assistant integrates foundation LLMs with retrieval-augmented generation (RAG) for including few-shot examples as question-SQL code pairs
- Retrieves verified few-shot “Golden Examples” based on semantic similarity to the user prompt.
- Prompts combine user question, metadata (see schematic), retrieved examples, and session context to generate SQL plus explanation; early deployment includes multiple commercial claims and EHR datasets
- Benchmark: 40 DE-SynPUF Medicare claims questions composed by reviewing SQL from historical RWE studies
- Benchmark SQL answers were developed with a combinations of two data scientists and LLM outputs. Evaluations were produced from a data scientist with LLM assistance.

**HEOR Efficiency Gains**

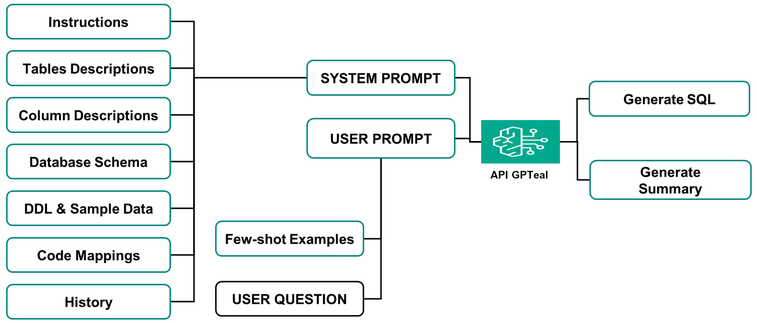
- Reduces time from analytic question to reviewable SQL draft
- Promotes more standardized use of metadata and prior validated query patterns
- Supports reproducible analyst-in-the-loop RWD workflows.

**Human oversight remains essential**

- Natural-language prompts can be ambiguous
- Correct SQL does not guarantee correct HEOR interpretation
- Final analyst review is still required for cohort validity, temporal logic, and code-list appropriateness.

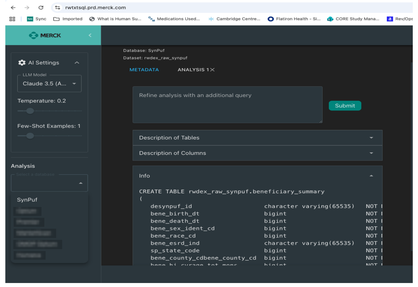
## Methods (continued)

### RWDEx AI SQL Assistant Architecture

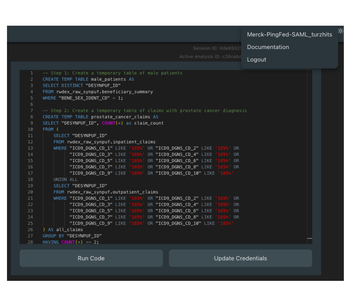



### Software Interface

**Settings and Database Metadata**




**Output SQL example**




### Initial benchmark results reported in abstract (Claude Sonnet 3.5)

- DE-SynPUF benchmark reported in abstract: 82.5% first-attempt accuracy and 97.5% within two attempts
- These results preceded the standardized difficulty and evaluation framework shown below, evaluating on a single difficulty dimension.

**Abstract-linked benchmark**

- Easy (N=5): 80% first attempt, 100% within two
- Medium (N=14): 86% first attempt, 100% within two
- Hard (N=21): 81% first attempt, 95% within two

### Two Dimensions of SQL Question Difficulty

**Orthogonality Concept**
SQL Difficulty and RWD Reasoning measure separate cognitive challenges and should remain independent.

**SQL Difficulty**
Assumes the intended domain concept is already understood.
Measures only how hard it is to express the logic correctly in SQL

**RWD Reasoning**
Assumes a perfect SQL assistant is available
Measures only how hard it is to identify the correct real-world data concept or event relationship

**Low SQL / High RWD Example**
A simple query that depends on distinguishing admitting diagnosis from principal diagnosis.

**High SQL / Medium RWD Example**
An obvious clinical concept that requires multi-step earliest-event logic across datasets.

### SQL Question Difficulty

**Level 1**
Simple single-table selection, DISTINCT, or COUNT with at most one obvious filter on a clearly indicated field.

**Examples**
Count distinct inpatients; select all outpatients in 2006; select patients with a diagnosis from one table.

**Level 2**
One substantive SQL operation beyond Level 1, such as a join, union, simple group by, or straightforward calculation.

**Examples**
Join claims to beneficiary table; group by patient; calculate treatment duration within one query.

**Level 3**
Combination of Level-2 elements, or multi-step logic such as subqueries, temp tables, earliest/latest comparisons, or complex branching.

**Examples**
Compare earliest diagnosis versus earliest drug exposure; identify patients who died during hospitalization after prior exposure.

### RWD Reasoning

**Low**
Can be answered from prompt wording plus schema / data-dictionary lookup. No substantive claims interpretation is needed.

**Examples**
Count inpatients; select all outpatients; select distinct patients.

**Medium**
Requires one nontrivial domain interpretation, such as selecting among plausible fields or understanding what a coded field operationally represents.

**Examples**
Race code meaning; claim admission date vs claim start date; covered days; BENE_DEATH_DT = 0 means alive.

**High**
Requires multiple domain concepts or interpretation of real-world temporal relationships between events, independent of the SQL used to implement them.

**Examples**
Diagnosis A before diagnosis B; age at diagnosis; admitting vs principal diagnosis; hospitalized because of a disease.

### SQL Evaluation Standard

**Evaluation principle**
A response can be marked Correct if it reflects a reasonable interpretation of the prompt and does not materially change the intended result set, even when it differs from the benchmark SQL. Benchmark SQL is a reference point, not the only acceptable answer.

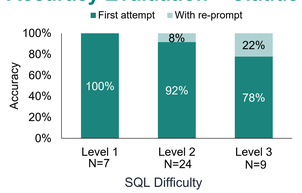
**Mark as Correct when**

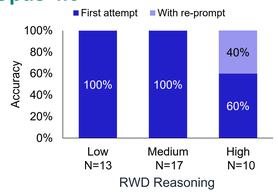
- The SQL gives a defensible interpretation of the prompt
- The chosen tables and fields are appropriate for that interpretation
- Differences from the benchmark are minor or not materially cohort-changing

**Mark as Wrong when**

- The SQL answers a different substantive question
- Required tables, event sources, or logic are missing
- Encoded values or temporal logic are misinterpreted
- Date logic materially changes who enters the cohort

### Accuracy Evaluation – Claude Opus 4.6



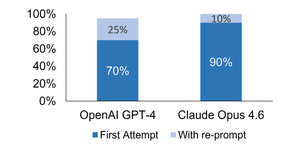




First-pass performance was higher for simpler questions, though a single re-prompt was able to resolve all errors, achieving 100% accuracy for every category.

**Common errors**

- Ambiguous date boundaries
- Event-sequencing errors
- Incomplete diagnosis-field coverage
- Invalid encoded-value or date handling

### Improvement in LLMs




- Newer LLM performs better on both SQL coding and RWD Reasoning metrics, answering 90% of questions on the first attempt and 100% when allowed a re-prompt, as compared to 70% and 95% for GPT-4.
- Data Scientist /informaticist review is important due to inherent ambiguity in natural language questions

**Implications**

- Higher first-pass accuracy can reduce analyst iteration burden.
- Metadata retrieval and Golden Examples help translate model gains into workflow gains.
- Decision-grade use still depends on transparent evaluation and human oversight.

## Discussions and Conclusions

- An LLM-driven, RAG-enhanced text-to-SQL assistant can reliably generate executable SQL for many RWD tasks and support iterative query refinement.
- The value of such a system depends not only on model capability, but also on maintained metadata assets, validated examples, transparent evaluation, and expert review.
- The most difficult problems often involve preserving scientific intent in temporal logic, diagnosis definitions, and cohort construction.
- Future work should expand benchmarking, quantify time-to-correct-query, and evaluate controlled efficiency and usability outcomes.

**Poster takeaway**
A use-specific benchmark and robust evaluation framework is very helpful for optimizing AI tools

**Keywords:** artificial intelligence; large language models; Text-to-SQL; real-world data; and retrieval-augmented generation.

**Acknowledgements**
We acknowledge the support of MRL IT in bringing this tool to production and AWS Professional services in building the front and back end.

Presented at International Society for Pharmacoeconomics and Outcomes Research (ISPOR); May 17 - 20, 2026; Philadelphia, PA, USA

Copyright © 2026 Merck & Co., Inc., Rahway, NJ, USA and its affiliates. All rights reserved.