05.01.2025

OMOPSYNC Abstract

A Modular Prompt Powered OMOP ETL & Analytical Software

Team 2.1 B OMOPSYNC

LOF TA - Merlin Simoes

Aum Sathwara (A20543213)

Bjoern Sagstad (A20557181)

Harneet Kaur (A20548613)

Vishnu Shanmugavel (A20561323)

Abstract

OMOPSYNC streamlines OMOP CDM use by pairing a language model with the Model Context Protocol Server. Natural language prompts are converted into ETL and query actions, while backend scripts handle data processing and analysis. Its modular design separates reasoning from execution, reduces errors, and enables easy updates for a clear path from raw data to insights.

Introduction

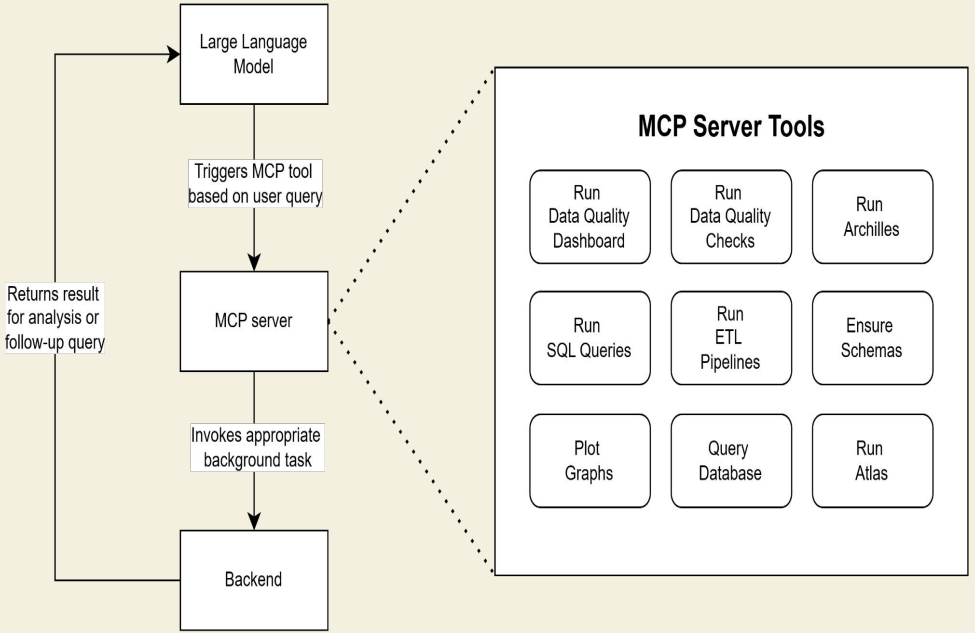
- Healthcare data heterogeneity poses challenges for scalable research.
- The OMOP CDM standardizes data structure, but transforming raw data remains labor-intensive.
- OMOPSYNC allows data loading, query execution and analysis through simple prompts
- It leverages LLMs and the Model Context Protocol (MCP) server architecture.

Architecture & Methodology

Three-tier modular design that can be hot-swapped or extended with minimal refactoring :

1. **LLM layer:** Large language model parses natural language queries and selects a Model Context Protocol (MCP) tool.
2. **MCP Server:** Exposes asynchronous tools (Run SQL, Plot Graphs, Run ETL, etc.); communicates with tools via **STDIO**; no network surface, mitigating web-based attacks.
3. **Backend:** Python wrappers trigger vetted R ETL scripts that load Synthea records, OMOP vocabularies into the OMOP CDM and execute Data Quality Dashboard checks; R can be swapped for Python modules without refactoring upper layers.

End-to-end flow: *query → LLM → MCP tool → backend → result → LLM response.*



Results

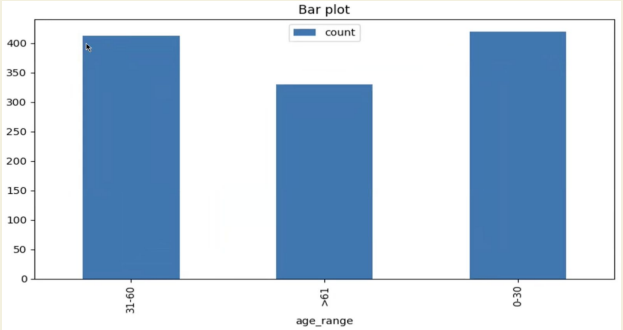
```
Query: Create a bar chart by querying the PostgreSQL database to calculate each person's age from the birth_datetime in the cdm54.person table, group the individuals into age ranges 0-30, 31-60, and 61+, count how many fall into each group, and display the result as a bar chart.

[Calling tool plot_query with args {'sql': "\nSELECT CASE WHEN age BETWEEN 0 AND 30 THEN '0-30' WHEN age BETWEEN 31 AND 60 THEN '31-60' ELSE '>61' END AS age_range, COUNT(*) AS count FROM (SELECT EXTRACT(YEAR FROM AGE(CURRENT_DATE, birth_datetime)) AS age FROM cdm54.person) AS ages GROUP BY age_range;\n", 'y_field': 'count', 'x_field': 'age_range', 'chart_type': 'bar'}]

Results: {"type": "text", "text": "Plot saved to 'analysis\\plot_bar_1745882969.png'."}
```

a) LLM calls post-query tool based on natural language query

- OMOPSYNC enables task execution via natural language prompts.
- Reduces token usage and cost vs. full code generation.
- Mitigates LLM hallucinations and improves task accuracy.



b) Generated bar chart

Data Output | Messages | Noti

	age_range text	count bigint
1	31-60	413
2	>61	330
3	0-30	419

c) Query validated with PostgreSQL

Conclusion

- **Accessibility With Data Quality:** OMOPSYNC helps researchers convert and analyze healthcare data using natural language commands.
- **AI as Enhancer, Not Replacement:** The key distinction of our approach is that AI does not generate or replace ETL pipelines; rather, it serves as a flexible interface that builds on predefined workflows to improve their accessibility and usability.
- **Balanced Approach:** This combination of user-friendly AI and solid scientific methods points to a practical way forward for healthcare data standardization.

Thank You!

OMOPSYNC