Natural Query Interface

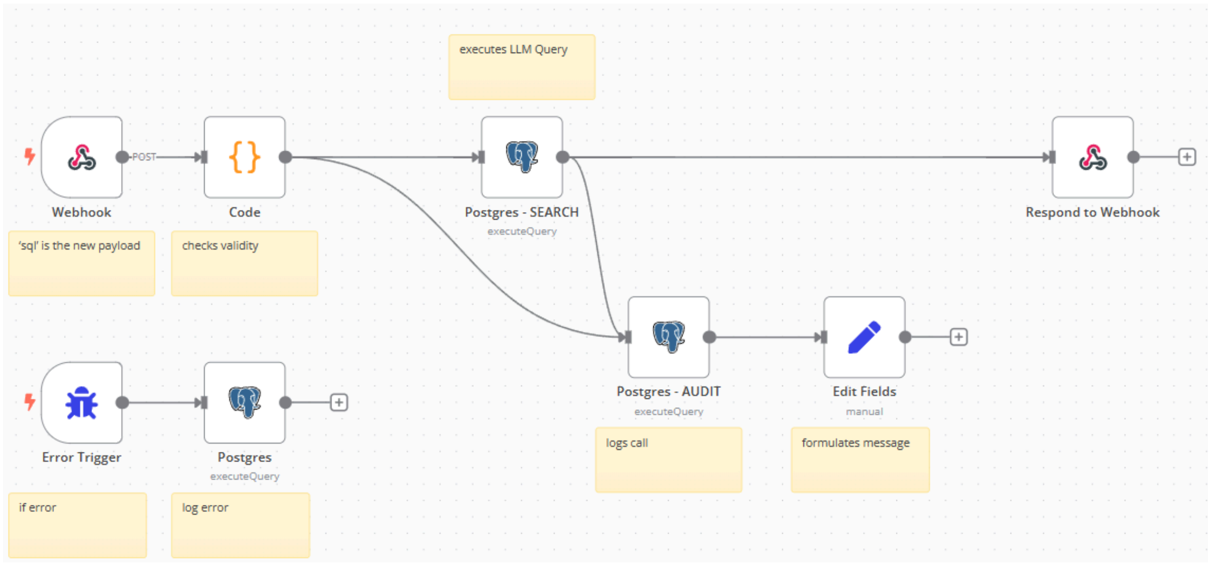
Mihai Constantinescu

The application accepts a natural language query, and through a first LLM call it converts this query and recent conversation context into a structured JSON object that specifies SQL parameters.

This JSON is then validated against a cached Postgres schema and used to build a SQL query. The query is sent to an n8n webhook, which executes it on the Postgres database.

Finally, a second LLM call refines the SQL results into a clear, natural language response that is returned to the user.

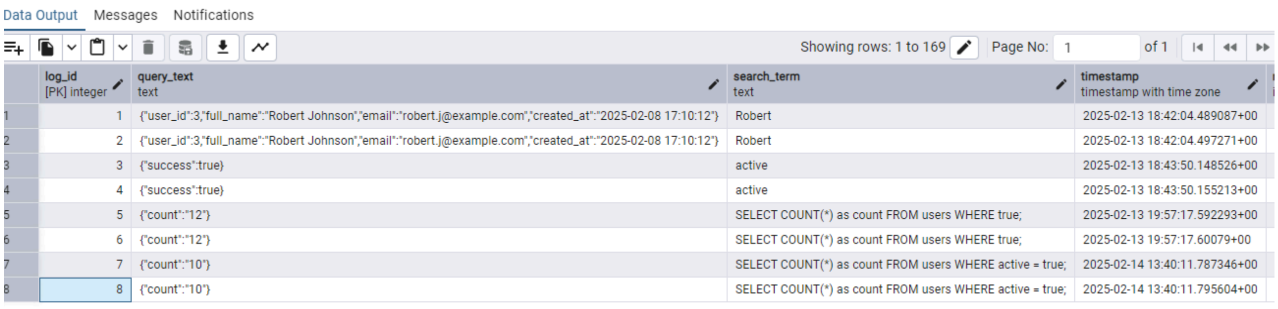
N8N



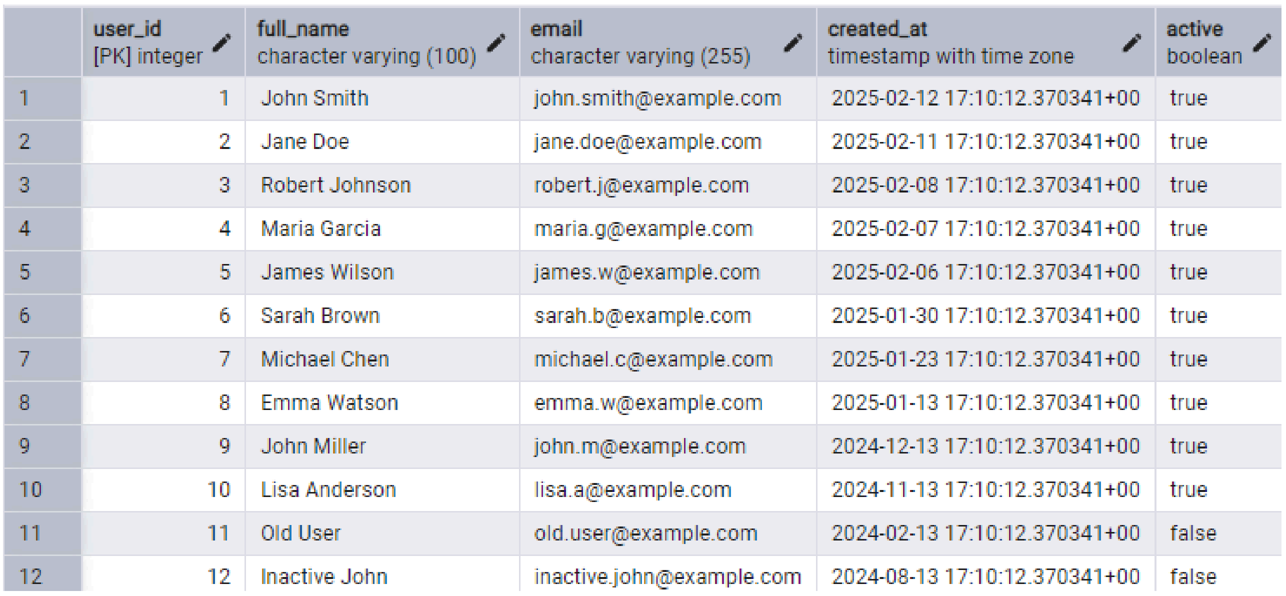
A WebHook accepting POST, a verification node, an execution node and an audit node. The execution node output is sanitized by a JS Code and sent as a response. Error triggers make sure errors are logged no matter what.

Postgres

Chosen for ease of integration, speed and simplicity. Hosts a schema containing 2 tables: 1 user and 1 query log.



	log_id [PK] integer	query_text text	search_term text	timestamp timestamp with time zone
1	1	{"user_id":3,"full_name":"Robert Johnson","email":"robert.j@example.com","created_at":"2025-02-08 17:10:12"}	Robert	2025-02-13 18:42:04.489087+00
2	2	{"user_id":3,"full_name":"Robert Johnson","email":"robert.j@example.com","created_at":"2025-02-08 17:10:12"}	Robert	2025-02-13 18:42:04.497271+00
3	3	{"success":true}	active	2025-02-13 18:43:50.148526+00
4	4	{"success":true}	active	2025-02-13 18:43:50.155213+00
5	5	{"count":"12"}	SELECT COUNT(*) as count FROM users WHERE true;	2025-02-13 19:57:17.592293+00
6	6	{"count":"12"}	SELECT COUNT(*) as count FROM users WHERE true;	2025-02-13 19:57:17.60079+00
7	7	{"count":"10"}	SELECT COUNT(*) as count FROM users WHERE active = true;	2025-02-14 13:40:11.787346+00
8	8	{"count":"10"}	SELECT COUNT(*) as count FROM users WHERE active = true;	2025-02-14 13:40:11.795604+00

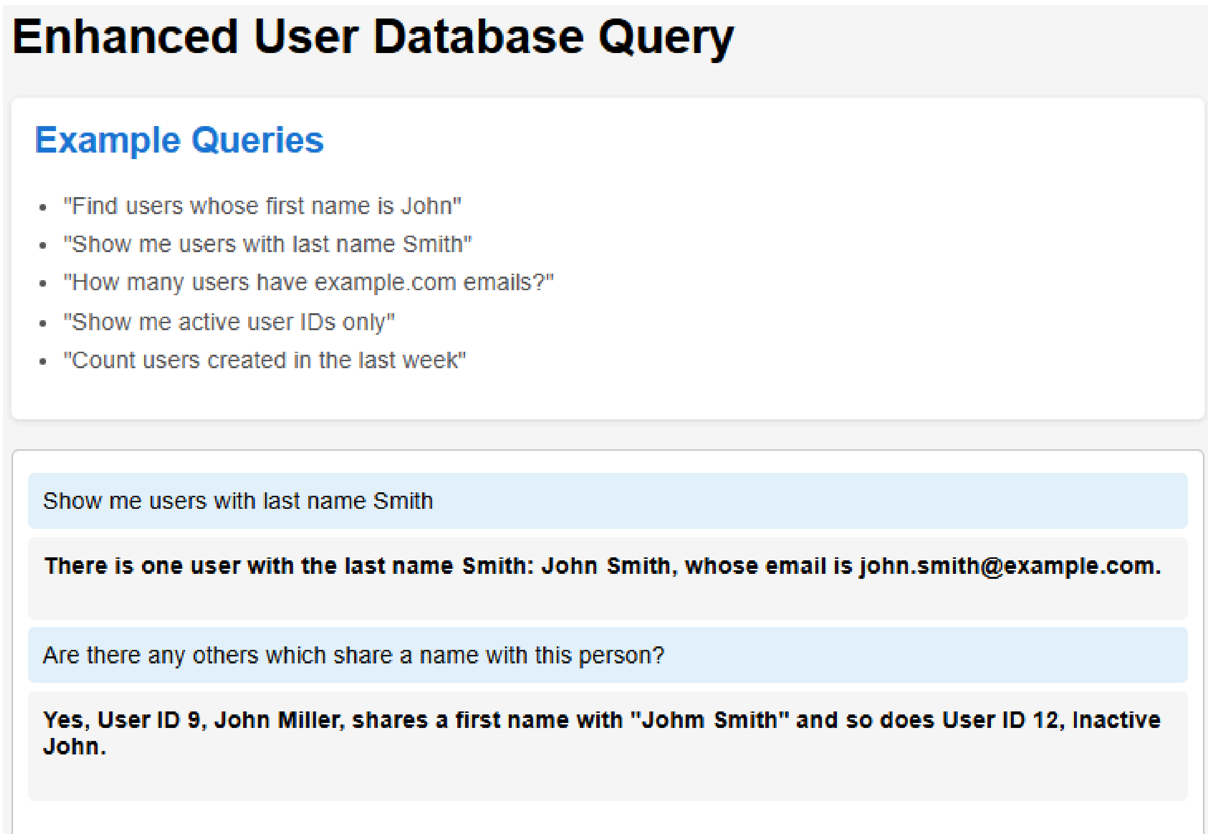


	user_id [PK] integer	full_name character varying (100)	email character varying (255)	created_at timestamp with time zone	active boolean
1	1	John Smith	john.smith@example.com	2025-02-12 17:10:12.370341+00	true
2	2	Jane Doe	jane.doe@example.com	2025-02-11 17:10:12.370341+00	true
3	3	Robert Johnson	robert.j@example.com	2025-02-08 17:10:12.370341+00	true
4	4	Maria Garcia	maria.g@example.com	2025-02-07 17:10:12.370341+00	true
5	5	James Wilson	james.w@example.com	2025-02-06 17:10:12.370341+00	true
6	6	Sarah Brown	sarah.b@example.com	2025-01-30 17:10:12.370341+00	true
7	7	Michael Chen	michael.c@example.com	2025-01-23 17:10:12.370341+00	true
8	8	Emma Watson	emma.w@example.com	2025-01-13 17:10:12.370341+00	true
9	9	John Miller	john.m@example.com	2024-12-13 17:10:12.370341+00	true
10	10	Lisa Anderson	lisa.a@example.com	2024-11-13 17:10:12.370341+00	true
11	11	Old User	old.user@example.com	2024-02-13 17:10:12.370341+00	false
12	12	Inactive John	inactive.john@example.com	2024-08-13 17:10:12.370341+00	false

Backend

The testing backend is a Flask server that processes user queries in three stages. First, it uses an LLM to convert a natural language query—along with recent conversation history and cached PostgreSQL schema details—into a structured JSON specification for a SQL query. Next, this JSON is validated and transformed into an actual SQL query, which is sent to an n8n webhook for execution on the PostgreSQL database. Finally, a second LLM call refines the raw SQL results into a clear natural language response that is returned to the user.

I have also begun and mostly finished an implementation using TypeScript (+ React and Tailwind) to improve maintainability and developer productivity and increase safety, reducing runtime errors.



Enhanced User Database Query

Example Queries

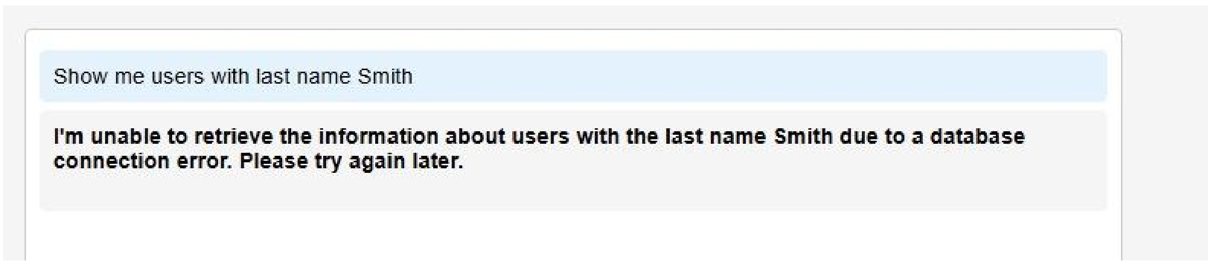
- "Find users whose first name is John"
- "Show me users with last name Smith"
- "How many users have example.com emails?"
- "Show me active user IDs only"
- "Count users created in the last week"

Show me users with last name Smith

There is one user with the last name Smith: John Smith, whose email is john.smith@example.com.

Are there any others which share a name with this person?

Yes, User ID 9, John Miller, shares a first name with "Johm Smith" and so does User ID 12, Inactive John.


Show me users with last name Smith

I'm unable to retrieve the information about users with the last name Smith due to a database connection error. Please try again later.