cloudquery / pg_gpt Public

Experimental extension that brings OpenAI API to your PostgreSQL to run queries in human language.

Apache-2.0 license

408 stars 15 forks

Star Notifications

Code Issues 1 Pull requests Actions Projects Security Insights

main

14

View code

README.md

Postgres <> ChatGPT

Experimental PostgreSQL extension that enables the use of OpenAI GPT API inside PostgreSQL, allowing for queries to be written using natural language.

Demo

Postgres.ChatGPT.mp4

(This demo uses data from the Hacker News and Azure CloudQuery plugins)

How does it work?

The extension sends a subset of the database schema to ChatGPT and asks it to generate a query based on this and the user input.

Before you start

- **Note**: This plugins sends schema (without the data) to OpenAI GPT API, so it is not recommended to use it on production databases.
- **Note**: This is an experimental plugin and not officially supported by CloudQuery.

Installation

Requires pgx. Install this first:

```
cargo install --locked cargo-pgx
cargo pgx init
```

Now you can install the extension:

```
git clone https://github.com/cloudquery/pg_gpt
cd pg_gpt
export OPENAI_KEY=<YOUR_KEY>
cargo pgx run
# will drop into psql shell
```

```
create extension pg_gpt;
set openai.key = '<YOUR OPENAPI API KEY HERE>'; -- set your key
select gpt('show me all open aws s3 buckets');
-- will output the following query, so you can execute it
-- select * from aws_s3_bucket;
```

Available Functions

- `gpt(text)` - Generates a query based on the user input and the full database schema. This works fine for databases with small schemas.
- `gpt_tables(table_pattern, text)` - Similar to gpt, but only uses the tables that match the pattern. The pattern is passed to a `table_name LIKE` query, so `%` can be used as wildcard.

Installing the extension on an existing Postgres instance

First run:

```
cargo pgx install
```

This places the extension in the postgres extensions directory. Then, in your postgres instance, run:

```
create extension pg_gpt;
set openai.key = '<YOUR OPENAPI API KEY HERE>';
-- proceed to use the extension
```

Limitations

- Schema Size - Currently we use gpt-3.5-turbo, which is limited to 4096 tokens. Use `gpt_tables` to narrow down the set of tables.

Releases

No releases published

Packages

No packages published

Contributors 2

hermanschaaf Herman Schaaf

yevgenypats Yevgeny Pats

Languages

● **Rust** 100.0%