How Uber Leveraged RAG and AI Agents to Revolutionize SQL Query Generation

Savleen Kaur · Follow
4 min read · Oct 14, 2024

Listen Share ••• More

In a world where data-driven decisions are at the heart of business operations, companies need efficient ways to extract meaningful insights from vast data stores. SQL queries are a staple of data analysis, but the manual process of writing complex queries can be time-consuming. Recognizing this challenge, **Uber** turned to cutting-edge AI technologies, using **Retrieval-Augmented Generation (RAG)** and intelligent agents to build their in-house tool called **QueryGPT**. The result? A significant **140,000 hours saved annually** in query writing time.

Here's a glimpse into how Uber built this system, and how it has streamlined their data workflows:

What is RAG?

Before diving into Uber's solution, let's clarify the key technology that underpins it: **RAG (Retrieval-Augmented Generation)**. RAG combines the best of two worlds: **retrieval-based models** and **generative models**. It allows AI systems to fetch relevant information from external databases and then generate responses based on that data. For Uber, this meant pairing AI's natural language understanding capabilities with SQL generation that draws on real-time, accurate data sources.

Image source: https://www.nomidl.com/generative-ai

Introducing QueryGPT

At the core of Uber's efficiency gains is **QueryGPT**, a tool built using multiple agents that work together to translate natural language into SQL queries. Each agent in the pipeline plays a crucial role in ensuring that the generated query is accurate, relevant, and optimized for performance. Here's how the process works:

1. The Intent Agent: Understanding User Needs

The journey begins with the **Intent Agent**, which interprets the user's natural language query. Whether it's a request related to **Mobility**, **Billing**, or any other domain, the Intent Agent accurately identifies the user's intent and maps it to the relevant data workspace. This is akin to a smart assistant that knows which part of the database to query based on the question asked.

2. The Table Agent: Selecting Relevant Tables

Next, the **Table Agent** steps in to choose the appropriate tables from the database. Using a **Large Language Model (LLM)**, this agent ensures that the right data sources are being queried, thus reducing the manual effort users would otherwise invest in determining table relevance. Users are also given the option to review and adjust the selected tables, making the process both efficient and flexible.

3. The Column Prune Agent: Optimizing Data Selection

Uber's datasets are massive, so trimming down unnecessary columns is vital for efficiency. Enter the **Column Prune Agent**, which applies **RAG** to filter out columns that aren't needed for the query. This pruning helps the generated query stay within the **token limits** of language models, ensuring smooth execution while maintaining relevant data.

4. QueryGPT: Generating the SQL Query

The final step is the **QueryGPT** agent. Using **Few-Shot Prompting**, QueryGPT references a small set of SQL examples and the current schema to generate a precise, executable SQL query. By using only a handful of prompts, this approach significantly reduces the time required to produce the query, cutting it down from 10 minutes to just 3 minutes.

The Results: Time Savings and Efficiency Gains

By automating the query generation process with QueryGPT, Uber has drastically reduced the time and effort required to turn data questions into SQL queries. With **over 140,000 hours saved annually**, the benefits are clear:

- **Speed**: QueryGPT brings down query creation time by **70%**, from 10 minutes to 3.
- **Accuracy**: RAG and intelligent agents ensure the generated queries are both relevant and optimized.
- **Flexibility**: Users retain control over the query process, allowing them to review and refine selections, while still reaping the time-saving benefits of automation.

Why RAG and AI Agents Matter

Uber's use of **RAG** and AI agents represents a powerful shift in how companies can use artificial intelligence to optimize internal processes. By blending retrieval-based AI (which finds the most relevant data) with generative AI (which constructs meaningful outputs like SQL queries), Uber has achieved an innovative solution that other businesses can learn from.

As AI continues to evolve, expect to see more tools like QueryGPT that leverage these hybrid models to automate complex tasks, reduce inefficiencies, and free up human talent for more strategic work.

Uber's **QueryGPT** showcases the power of AI to transform how businesses interact with data. By utilizing RAG and a carefully orchestrated series of agents, the company has set a new standard for efficiency in query generation — saving time, AI Data Engineering Analytics Uber Genai

Follow

Written by Savleen Kaur

28 Followers · 15 Following

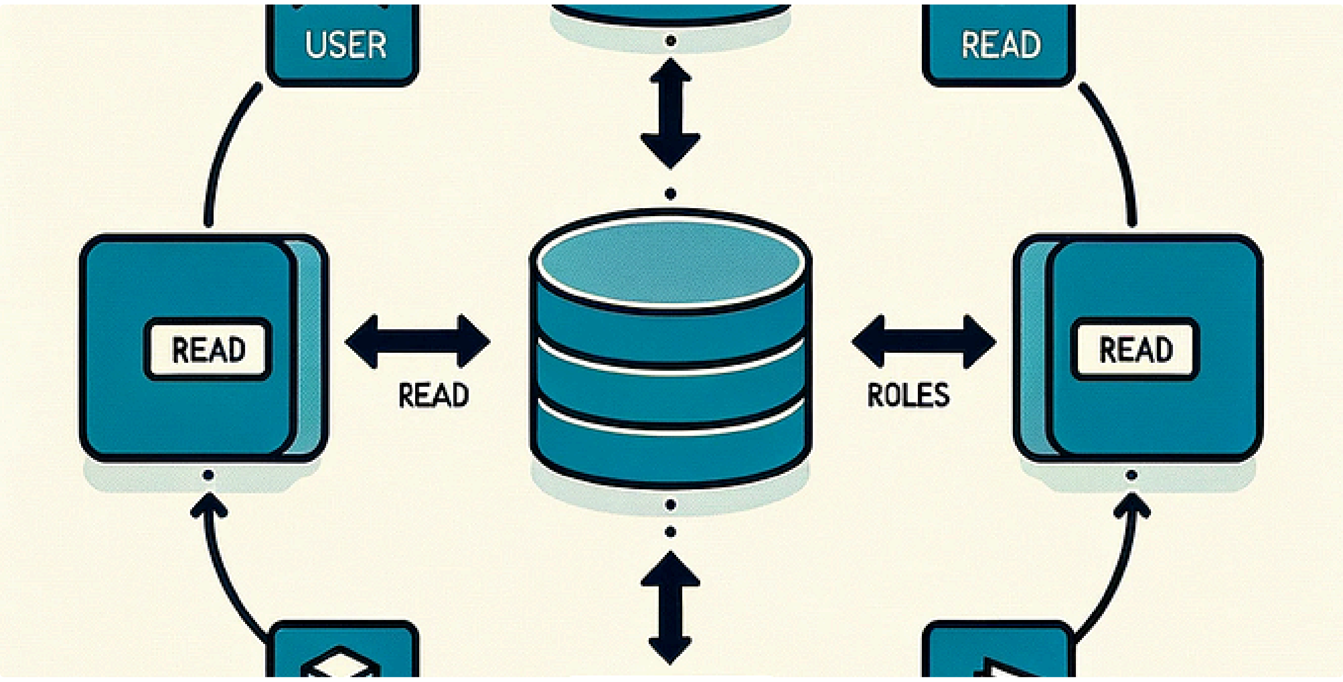
Sr Analytics Engineer at MongoDB

No responses yet

What are your thoughts?

Respond

More from Savleen Kaur



Savleen Kaur

Row-Based Access Control(RBAC) in Snowflake and its Implementation

Data security and privacy are dominant in today’s digital world, especially when dealing with large-scale data processing and analytics...

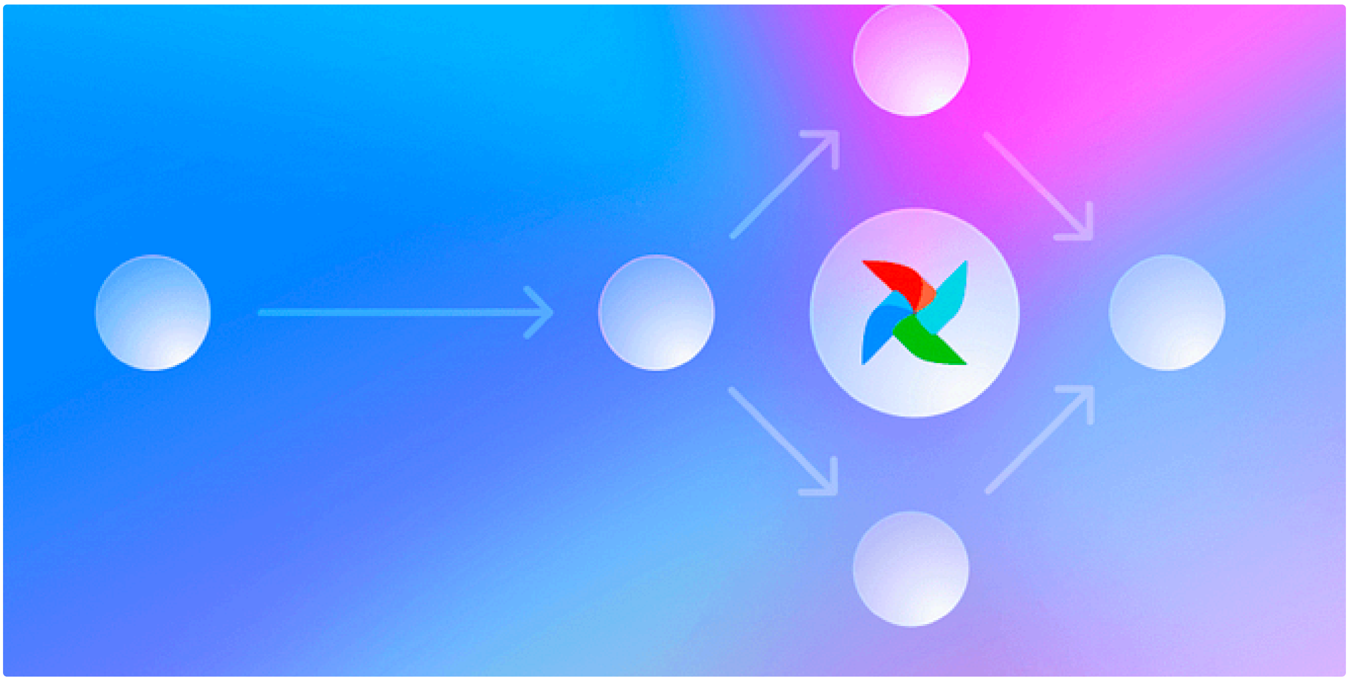
Jan 2, 2024 1

Savleen Kaur

Leveraging DBT Macros for Enhanced Data Engineering Practices

In the evolving world of data engineering, efficiency and scalability are key. That’s where DBT (Data Build Tool) comes in, offering a...

Dec 25, 2023 10 1



Savleen Kaur

Unleashing the Power of Custom Operators in Apache Airflow

Apache Airflow, an open-source platform to programmatically author, schedule, and monitor workflows, is renowned for its flexibility and…

Jan 6, 2024

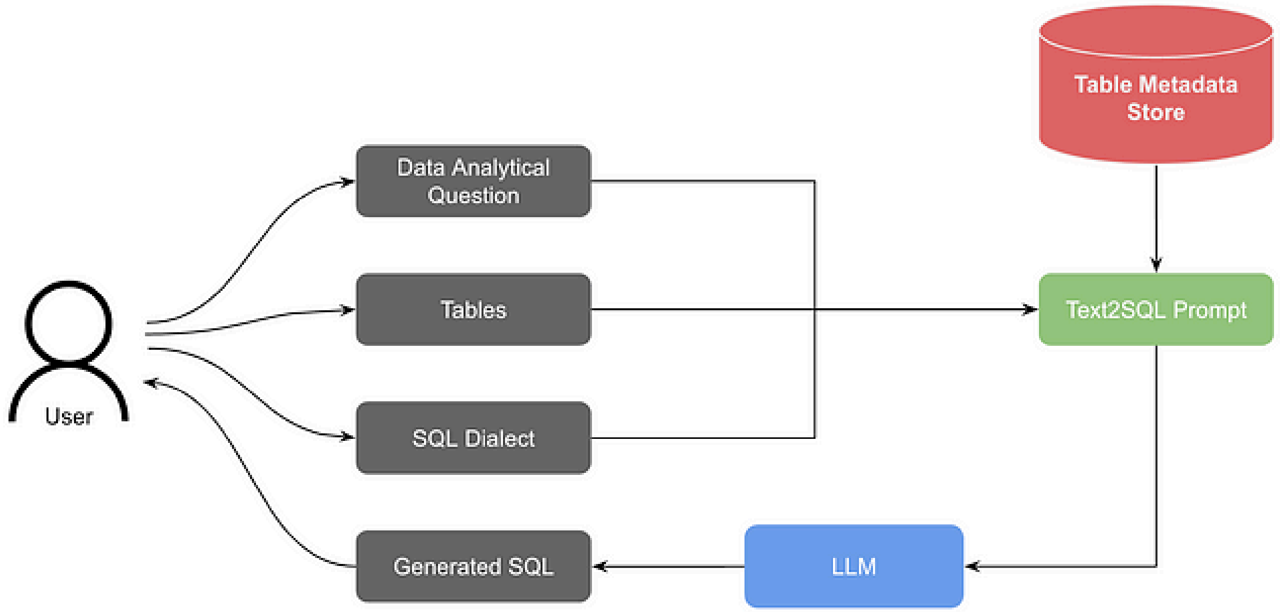
Savleen Kaur

What is Zero ETL?

Mar 19, 2024 👏 1

See all from Savleen Kaur

Recommended from Medium

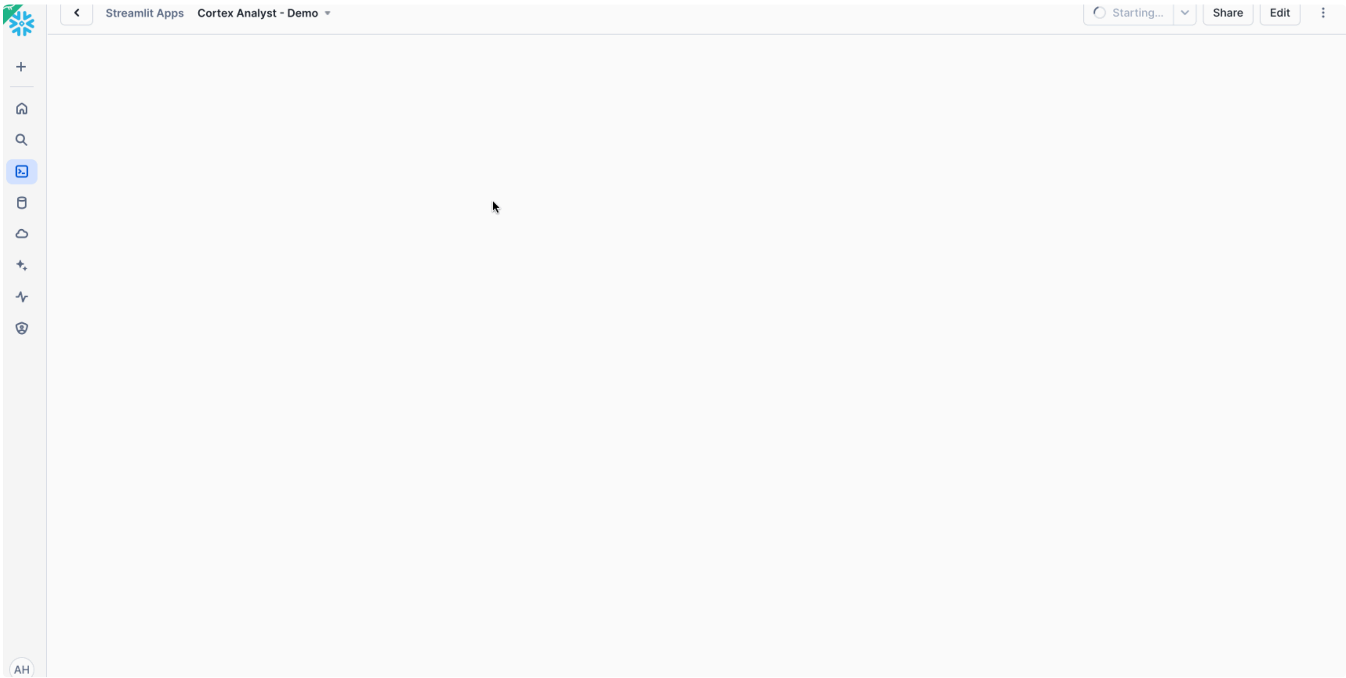


In Pinterest Engineering Blog by Pinterest Engineering

How we built Text-to-SQL at Pinterest

Adam Obeng | Data Scientist, Data Platform Science; J.C. Zhong | Tech Lead, Analytics Platform; Charlie Gu | Sr. Manager, Engineering

Apr 3, 2024 👏 2.5K 22



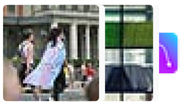
In Snowflake Builders Blog: Data Engineers, App Developers, AI/ML, & Data Science by Ahmed Rachid Hazourli

Self-serve Analytics application in minutes with Snowflake Cortex Analyst ❄️

A Streamlit application that helps you gain instant insights on your data stored in Snowflake in less than 2 minutes

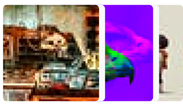
Oct 22, 2024 👏 7

Lists



Generative AI Recommended Reading

52 stories · 1657 saves

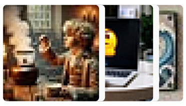


What is ChatGPT?

9 stories · 508 saves

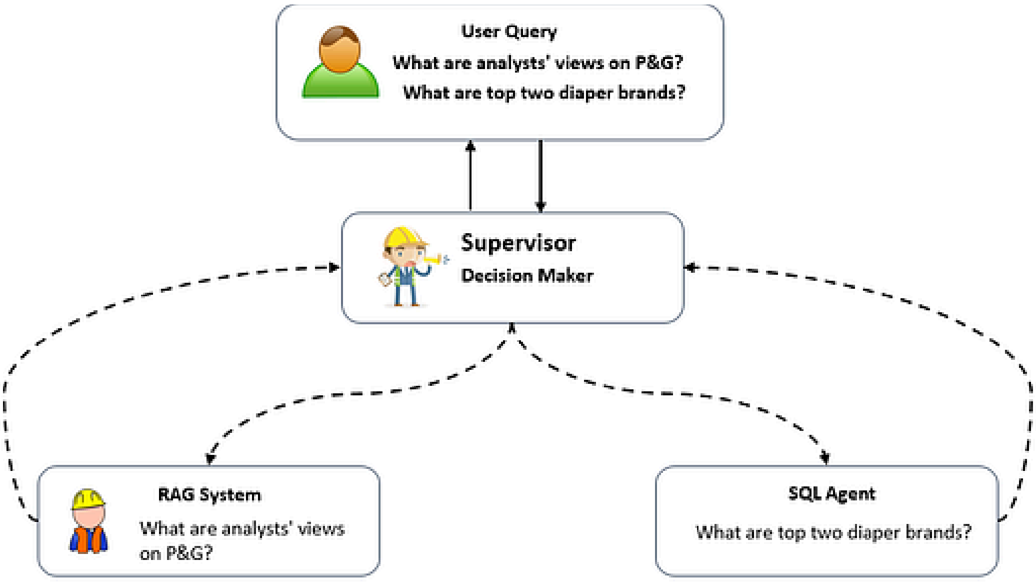
The New Chatbots: ChatGPT, Bard, and Beyond

12 stories · 549 saves



Natural Language Processing

1939 stories · 1595 saves


User Query
What are analysts' views on P&G?
What are top two diaper brands?
Supervisor
Decision Maker
RAG System
What are analysts' views on P&G?
SQL Agent
What are top two diaper brands?

B Rajeshrao

Building a Multi-Agent System using Langgraph involving SQL Agent and RAG model

Introduction

Sep 8, 2024 50

In Software, AI and Marketing by Madhukar Kumar

Why Text-to-SQL is Failing for Agents and How to Fix It?

A proposed solution for a zero inaccuracy data retrieval for Agents

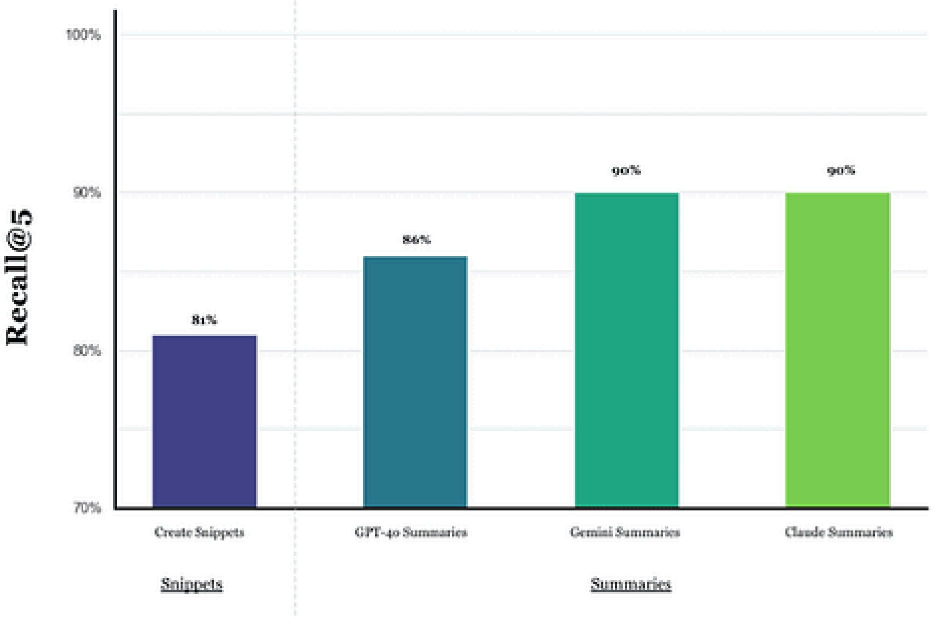
Dec 20, 2024 67

Beinset Hounwanou

Using LangChain and OpenAI to Query SQL Databases: A Practical Example

In this article, we will explore how to use LangChain and OpenAI to interact with an SQL database. We'll walk through a Python script that...

Aug 30, 2024 7



In Timescale by Team Timescale

Enhancing Text-to-SQL With Synthetic Summaries: A Few-Shot Learning Approach

Learn how few-shot learning with synthetic SQL query summaries can improve question-to-query matching.

Jan 3 👏 11

See more recommendations