

Open in app ↗

Medium

Search

54



How Uber is Saving 140,000 Hours Each Month Using Text-to-SQL — And How You Can Harness the Same Power



Howard Chi · Follow

Published in Wren AI

12 min read · Jan 3, 2025



Listen



Share

... More

In a world where data-driven decision-making is critical, businesses are scrambling to find the most efficient ways to extract actionable insights from massive datasets. Uber, a global leader in real-time logistics and transportation, recently shared how their internal Text-to-SQL platform — **QueryGPT** (If you haven't checked out the post, [check it out here](#))— is revolutionizing the way their teams interact with data. By enabling employees to simply ask questions in natural language and receive SQL queries in return, Uber has cut query authoring time by 70%. Considering they run about 1.2 million queries per month, this translates into an astonishing 140,000 hours saved monthly.

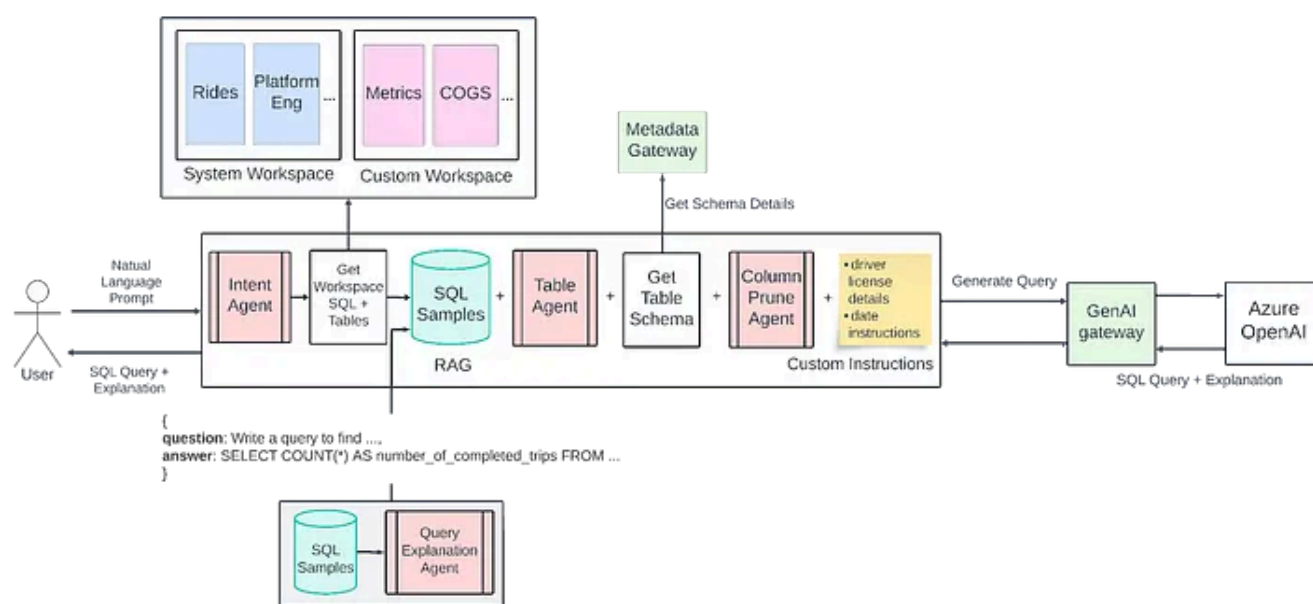


ROI of QueryGPT, image source from the Uber: QueryGPT — Natural Language to SQL Using Generative AI [article](#)

Yet, this remarkable efficiency isn't only for tech giants with vast engineering resources. With the advent of open-source solutions like **Wren AI**, the Text-to-SQL advantage can be democratized. Wren AI Cloud aims to give businesses of all sizes the power to use natural language queries, powered by generative AI, to seamlessly access their data. In this post, we'll dissect Uber's QueryGPT from a technical standpoint and highlight how Wren AI Cloud mirrors (and in some cases simplifies) these complex features. The goal is to help you understand how to implement Text-to-SQL in your own operation — no matter your scale.

Understanding Uber's Technical Approach to Text-to-SQL

Uber's data platform is a behemoth: it handles trillions of rows, petabytes of data, and millions of queries each month. Traditional SQL authoring is time-consuming and requires users to have strong query-building skills, understand the underlying data models, and know where to find the right tables and columns. QueryGPT removes these roadblocks by using large language models (LLMs) and clever integration into Uber's existing data ecosystem.



Current design of [QueryGPT](#) (Image source [QueryGPT](#))

Key Technical Components of designing QueryGPT at Uber, shared in the post of [QueryGPT](#) article

1. Workspaces

Workspaces are curated collections of SQL samples and table schemas aligned with specific business domains, such as *Mobility*, *Ads*, *Core Services*, and more. By creating domain-oriented clusters of relevant tables and query templates, QueryGPT narrows

the scope of possible data sources, which significantly improves the model's accuracy when generating SQL.

How It Works: When a user interacts with QueryGPT, the system will first identify which business domain (e.g., *Mobility* for trips and drivers) is relevant to the query. Within that workspace, QueryGPT will reference a smaller, more focused set of tables and SQL patterns, rather than scanning through Uber's entire database ecosystem.

Advantages:

- *Precision:* By limiting the search space to domain-specific references, QueryGPT is more likely to pick the correct tables and columns.
- *Reduced Complexity:* Users dealing with the *Mobility* domain, for example, only see tables related to trips, drivers, or documents, which simplifies the data exploration process.
- *Customizability:* Beyond the *System Workspaces* that Uber provides by default, users can create their own *Custom Workspaces* for niche use cases or novel projects that are not covered by standard domains.

2. Intent Agent

After a user inputs a question in natural language, QueryGPT employs an **Intent Agent** to interpret the user's intent and determine the most appropriate workspace(s). This step is vital in ensuring the system directs queries to the correct domain and, by extension, the correct subset of tables.

- **Intent Detection:** Using a large language model, the Intent Agent analyzes the user's query — looking for keywords, context, and semantics — to map it to one or more domain workspaces. For instance, if the question involves trip data, driver details, or vehicle attributes, the system might route the query to the *Mobility* workspace.
- **Multiple Mappings:** Some queries may span multiple domains (e.g., an analysis that touches both *Mobility* and *Ads* data), in which case the Intent Agent can map the question to more than one workspace. This ensures that cross-domain queries are also supported.

- **Efficiency Gains:** Because only the relevant workspaces are considered, QueryGPT reduces the computational overhead of rummaging through irrelevant schemas. This not only boosts accuracy but also shortens the overall query generation time.

3. Table Agent

Once QueryGPT knows which business domain(s) to focus on, the **Table Agent** proposes a list of specific tables that are most relevant to the user's request. This step is crucial in large organizations like Uber, where a single domain might contain dozens — or even hundreds — of tables with overlapping or complementary data.

- **Table Selection:** Drawing on the user's intent and the curated workspace content, the Table Agent pulls up the most likely candidates needed for the SQL query. It leverages example queries, table relationships, and domain-specific knowledge to decide which tables are pertinent.
- **User Verification:** The user is then shown a summary of the chosen tables and asked to confirm if they are correct. If something seems off — maybe the system picked an outdated table or missed a new one — the user can edit the list before moving forward. This *human-in-the-loop* feedback mechanism helps maintain high quality and trustworthiness of the generated SQL.
- **Enhanced Collaboration:** By allowing users to fine-tune table choices, QueryGPT bridges the gap between automated query generation and domain expertise. Data analysts familiar with specific schemas can quickly ensure QueryGPT is referencing the right data sources.

4. Column Prune Agent

Even with the correct tables identified, large enterprise schemas can contain hundreds of columns, each of which must be described to the language model if there's a chance it might appear in the query. Such exhaustive detail can hit or exceed token limits during generation, especially when using models like GPT-4 Turbo with high token capacities.

- **Pruning Logic:** The **Column Prune Agent** uses an LLM call to filter out columns that are unlikely to be relevant to the user's question. By doing so, it dramatically reduces the amount of information passed along to the subsequent query generation step.

- **Cost and Performance Benefits:** With fewer tokens involved, QueryGPT lowers the cost of each LLM call and processes the query faster. Additionally, removing irrelevant columns simplifies the final SQL query, making it more transparent and maintainable.
- **Reduced Errors:** Handling a slimmer, more targeted schema also reduces the chance of the model selecting the wrong fields — improving both the clarity and the accuracy of the final SQL output.

Bringing It All Together

These four components — **Workspaces**, the **Intent Agent**, the **Table Agent**, and the **Column Prune Agent** — orchestrate a streamlined, highly efficient text-to-SQL generation process at Uber. By segmenting the solution into domain-specific Workspaces, filtering queries through an Intent Agent, validating table choices via a Table Agent, and pruning unnecessary columns before the final query is generated, QueryGPT ensures high accuracy, cost savings, and quick turnaround. This approach empowers users to interact with Uber's complex data ecosystem through simple, natural language questions — raising the bar on data accessibility and operational efficiency across the company.

The Direct Business Impacts of QueryGPT at Uber

Technically, QueryGPT is a marvel of LLM integration, prompt engineering, and systems design. The business outcomes reflect this technical mastery:

- **70% Reduction in Query Time:** A decrease from about 10 minutes to 3 minutes per query is a massive gain in analyst efficiency.
- **140,000 Hours Saved per Month:** This doesn't just mean cost savings. It enables analysts to spend more time on value-added tasks: interpreting results, optimizing campaigns, improving rider experiences, and strengthening platform reliability.
- **Faster Feedback Loops:** With quicker access to insights, product managers and data scientists can iterate faster, test hypotheses more frequently, and launch improvements more confidently.
- **Competitive Advantage:** In an industry where responsiveness to market changes is key, the ability to quickly extract insights from data directly translates to better decision-making and improved customer experiences.

Mapping Uber's QueryGPT Features to Wren AI Cloud

Uber's QueryGPT is undoubtedly impressive, but it's tailored for a massive organization with extensive engineering resources. How can a growing startup, a mid-sized enterprise, or even a solo data practitioner tap into similar technologies? This is where **Wren AI** comes into play.



The Wren AI Project

Wren AI is an open-source SQL AI agent designed to democratize Text-to-SQL technology. By offering a cloud-based platform that integrates many of the same features as Uber's QueryGPT, Wren AI aims to level the playing field.

Below is an overview of how Uber's QueryGPT design aligns with and maps to Wren AI's features, illustrating how similar principles of workspace separation, intent detection, table selection, and column pruning are implemented in both systems. These parallels demonstrate a shared commitment to delivering a streamlined, secure, and user-friendly text-to-SQL experience.

1. **Workspaces** → **Projects & Organizations in Wren AI**

QueryGPT at Uber

In QueryGPT, *Workspaces* serve as curated collections of SQL samples and table schemas for specific domains, such as Mobility or Core Services. By narrowing the focus to a particular business domain, QueryGPT can more accurately generate SQL queries and ensure that data analysts only interact with context-relevant tables.

Wren AI Equivalent

Wren AI offers similar functionality through its **project** and **organization** management features. You can read more about creating organizations and projects in the Wren AI Cloud documentation:

- Create a Project
- Create an Organization

Just like QueryGPT's Workspaces, Wren AI's **projects** let you group and isolate specific data models so that only authorized users can access them. Within a single *organization*, you can set up multiple projects for different functions or business domains — similar to how QueryGPT sets up separate Workspaces. Access controls in Wren AI ensure that only the right people can view and manage sensitive data, aligning with the same principles of domain-scoped isolation seen in QueryGPT.

Why This Matters

- **Targeted Context:** By restricting data access to a single workspace or project, the system can better understand user requests and generate more accurate queries.
- **Security & Governance:** Organizations with sensitive or proprietary data benefit from robust access controls and data policies, ensuring compliance and proper data governance.

2. Intent Agent → Wren AI's Intent Detection

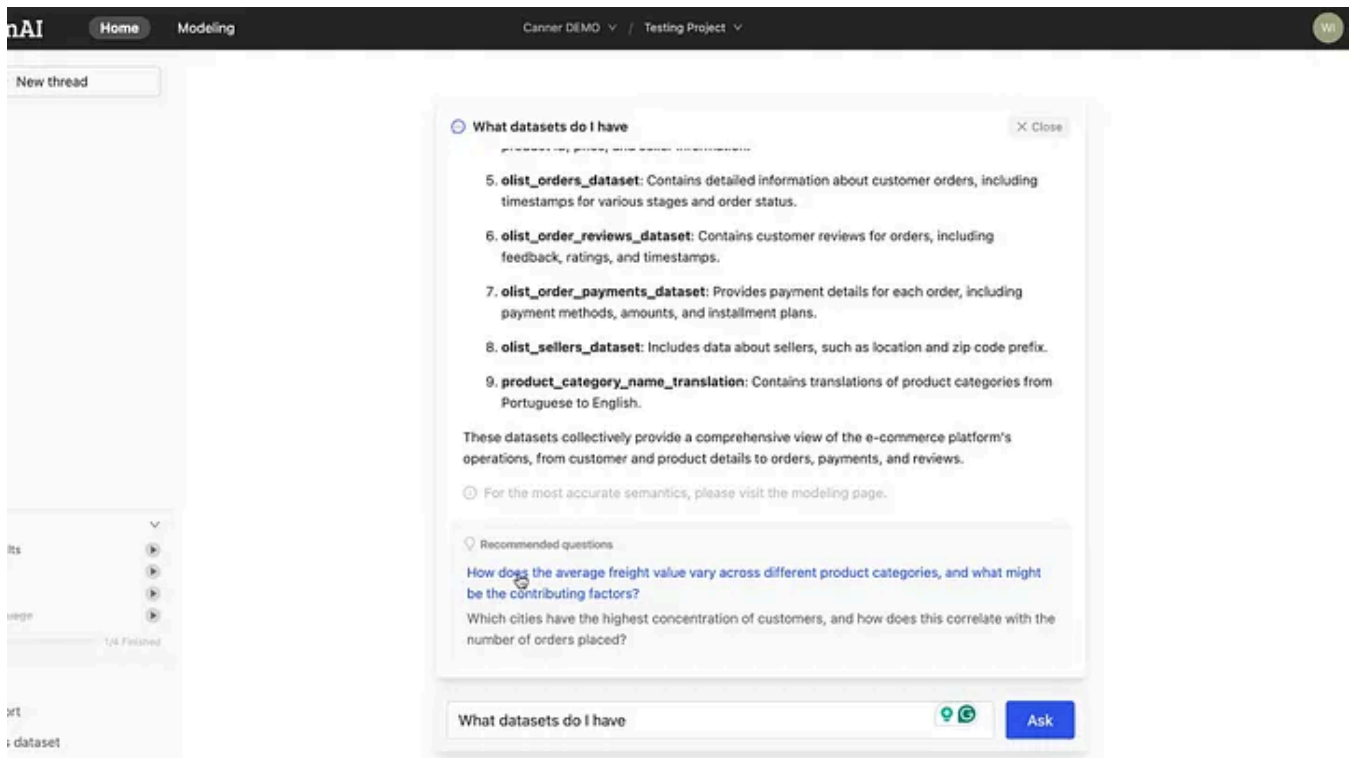
QueryGPT at Uber

When a user inputs a question, QueryGPT's *Intent Agent* identifies which business domain the question belongs to — Mobility, Ads, etc. — and routes the query to the corresponding workspace. This step dramatically narrows the search space for relevant tables and schemas, improving accuracy and speed.

Wren AI Equivalent

Wren AI's approach to intent detection is described in the [Ask documentation](#).

When you ask a question like “How many tables do I have?” or “Explain the customer table to me,” Wren AI automatically discerns whether you're requesting data retrieval, schema exploration, or if your question falls out of scope (e.g., a casual greeting).



AI-powered Data Exploration Features

- **Data Retrieval Requests:** These queries prompt Wren AI to generate SQL, mapping the request to the underlying data schema.
- **Schema Exploration:** When questions are more about the structure — like listing available tables, describing a table's columns, or explaining relationships — Wren AI provides in-depth metadata and recommended queries.

Why This Matters

- **Automatic Domain Routing:** Similar to QueryGPT, Wren AI's *Intent Detection* ensures that your request is processed accurately, either leading to SQL generation or a schema exploration response.
- **User Guidance:** If a request is out-of-scope (e.g., small talk), Wren AI prompts the user to clarify, maintaining a clear focus on data and schema queries.

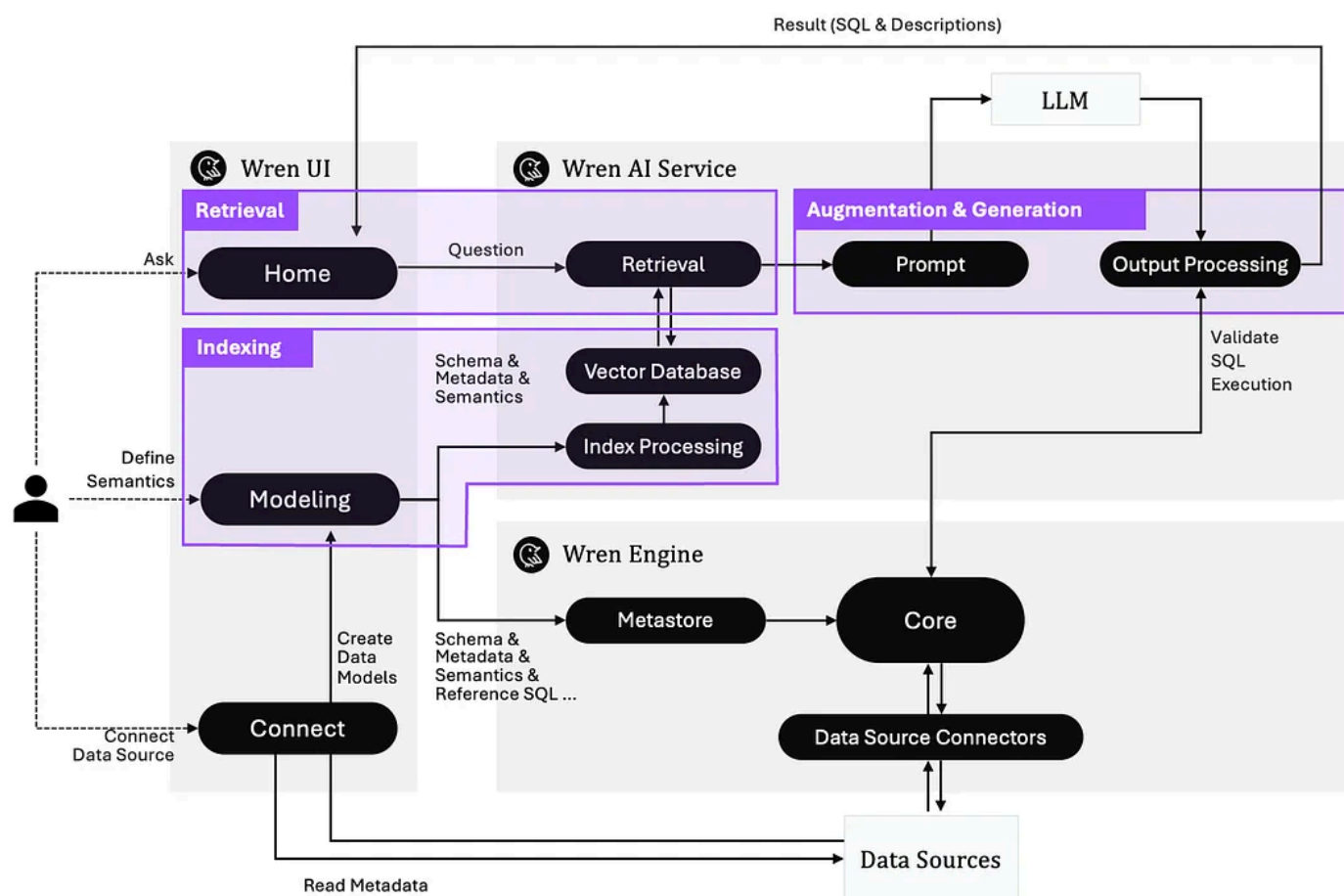
3. Table Agent → Wren AI's Table Retrieval Agent

QueryGPT at Uber

In QueryGPT, once the correct domain is identified, the Table Agent proposes which specific tables are necessary to construct the SQL query. Users can either confirm these suggestions or edit them to ensure alignment with their real-world data expertise.

Wren AI Equivalent

In Wren AI, this step corresponds to the **table retrieval** phase, where we use semantic search to select the *top 10 tables* most relevant to the user's question. We look at each table name and its metadata — such as descriptions or tags — to determine the best matches before generating the query.



Wren AI Architecture

- **Focused Retrieval:** By narrowing it down to the most relevant tables, Wren AI saves users from sifting through large, complex schemas.
- **Semantic Relevance:** Matching a user's question against the table metadata ensures that the initial query stage is both accurate and efficient.

Why This Matters

- **Accuracy:** Identifying the right tables up front reduces the likelihood of incorrect or irrelevant queries.
- **Efficiency:** Semantic search and a concise top-10 table list streamline the retrieval process, saving time for both technical and non-technical users.

4. Column Prune Agent → Wren AI's Column Pruning

QueryGPT at Uber

Large data schemas in enterprise environments can have hundreds of columns per table, potentially hitting token limits when feeding this data into an LLM. The *Column Prune Agent* filters out unnecessary columns to avoid overloading the model, cutting down on both latency and cost.

Wren AI Equivalent

Wren AI's **Column Prune Agent** serves an identical function: when you connect large numbers of tables and columns, Wren AI prunes those columns that aren't relevant to the query or the user's immediate needs. This keeps the system fast, efficient, and affordable to operate at scale.

Why This Matters

- **Scalability:** Reducing token load ensures that the LLM can handle complex queries without timing out or incurring huge processing costs.
- **Improved Accuracy:** Focusing on the most pertinent columns reduces distractions for the AI, leading to more precise SQL generation.

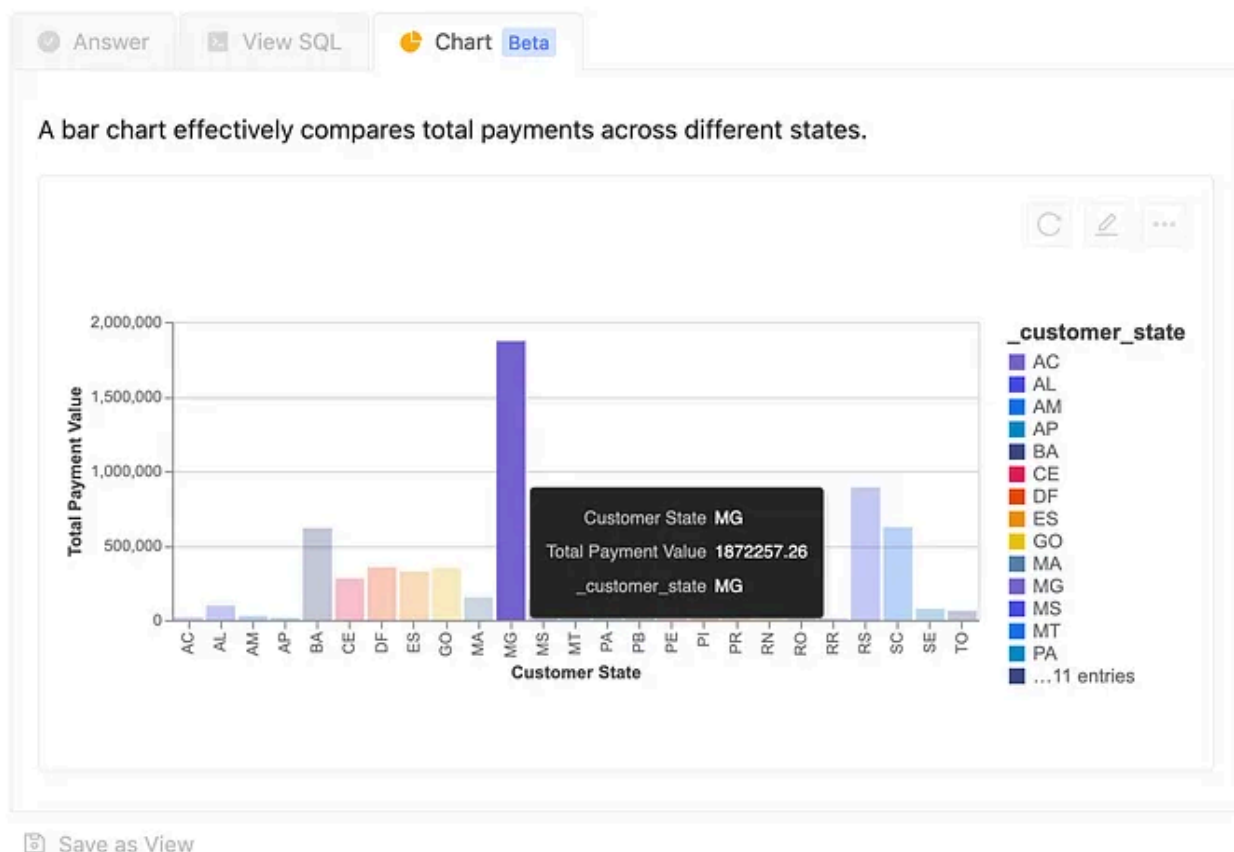
What's Even More with Wren AI

Beyond its robust text-to-SQL functionality, **Wren AI** delivers a host of additional features that make data analysis even more convenient, interactive, and accessible for teams of all technical backgrounds.

1. Text-to-Chart

Wren AI automatically generates insightful charts to visualize your data and uncover meaningful patterns — no additional steps required.

What is the total value of payments made by customers from each state?

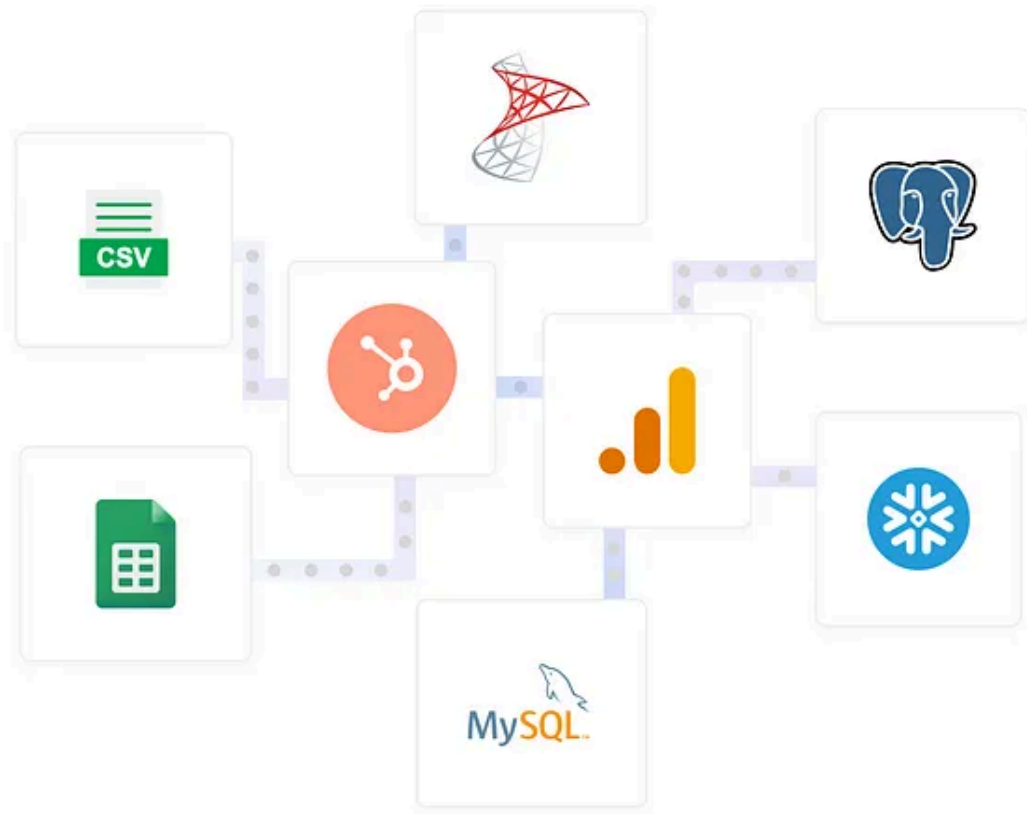


- **Automatic Chart Generation:** When you pose a question, Wren AI analyzes your dataset and selects the most suitable chart type to display the results.
- **Seamless Exploration:** Simply switch to the “Chart” tab to see your query results in a bar chart, line chart, pie chart, or another visually compelling format.
- **Quick Insights:** By representing the data visually, your team can spot trends and correlations faster, making data-driven decisions more intuitive.

Learn more: <https://docs.getwren.ai/oss/guide/home/chart>

2. Data Boilerplates

Boilerplates in Wren AI are pre-defined templates designed to simplify your data analysis journey from start to finish.



- **Streamlined Setup:** Boilerplates come with pre-selected tables and columns, eliminating the need to import large datasets that contain irrelevant information.
- **Jumpstart Exploration:** Curated sets of frequently asked questions help you quickly discover valuable insights without starting from scratch.
- **Supported Integrations:** Wren AI already offers boilerplates for **HubSpot**, **GA4**, and soon **WooCommerce**, with more on the way.

Learn more: <https://docs.getwren.ai/oss/guide/boilerplates/overview>

3. Step-by-step SQL breakdown

This breakdown walks users through how the AI arrives at certain tables and columns, culminating in the final SQL query.

- **User Transparency:** Wren AI reveals which tables and columns it intends to use. Users can see the logic behind these choices, enhancing trust and correctness.
- **Explainability:** Detailed breakdowns help users understand how the AI made its decisions, a critical factor for debugging and compliance in enterprise settings.

The screenshot shows the WrenAI web interface. On the left is a sidebar with a list of queries, including 'Total order count by city'. The main area displays the results for the query 'Question: What are the top 3 value for orders placed by customers in each city?'. The title is 'Total order count by city.' and it includes a 'Summary' section explaining the breakdown process. Below the summary are four numbered steps detailing the SQL logic. At the bottom, there is a table with two columns: 'City' and 'order_count'. The table shows results for Chula Vista (181), San Francisco (181), and Oakland (181). There is also an 'Ask to explore your data' input field and an 'Ask' button.

Question: What are the top 3 value for orders placed by customers in each city?

Total order count by city.

Summary

The breakdown simplifies the process of counting the number of orders for each city, while ensuring case-insensitive matching of customer IDs with order IDs.

1. Selects the city and customer ID from the customers table.
2. Selects the lowercase customer ID and order ID from the orders table.
3. Joins the customer and order data on matching customer IDs, counts the distinct order IDs for each city, and groups the results by city.
4. Retrieves the city and corresponding order count from the aggregated data, ordering the results by order count in descending order.

[Preview Data](#) [View Full SQL](#)

City	order_count
Chula Vista	181
San Francisco	181
Oakland	181

Ask to explore your data [Ask](#)

Learn more: <https://docs.getwren.ai/cloud/guide/home/answer#result-steps>

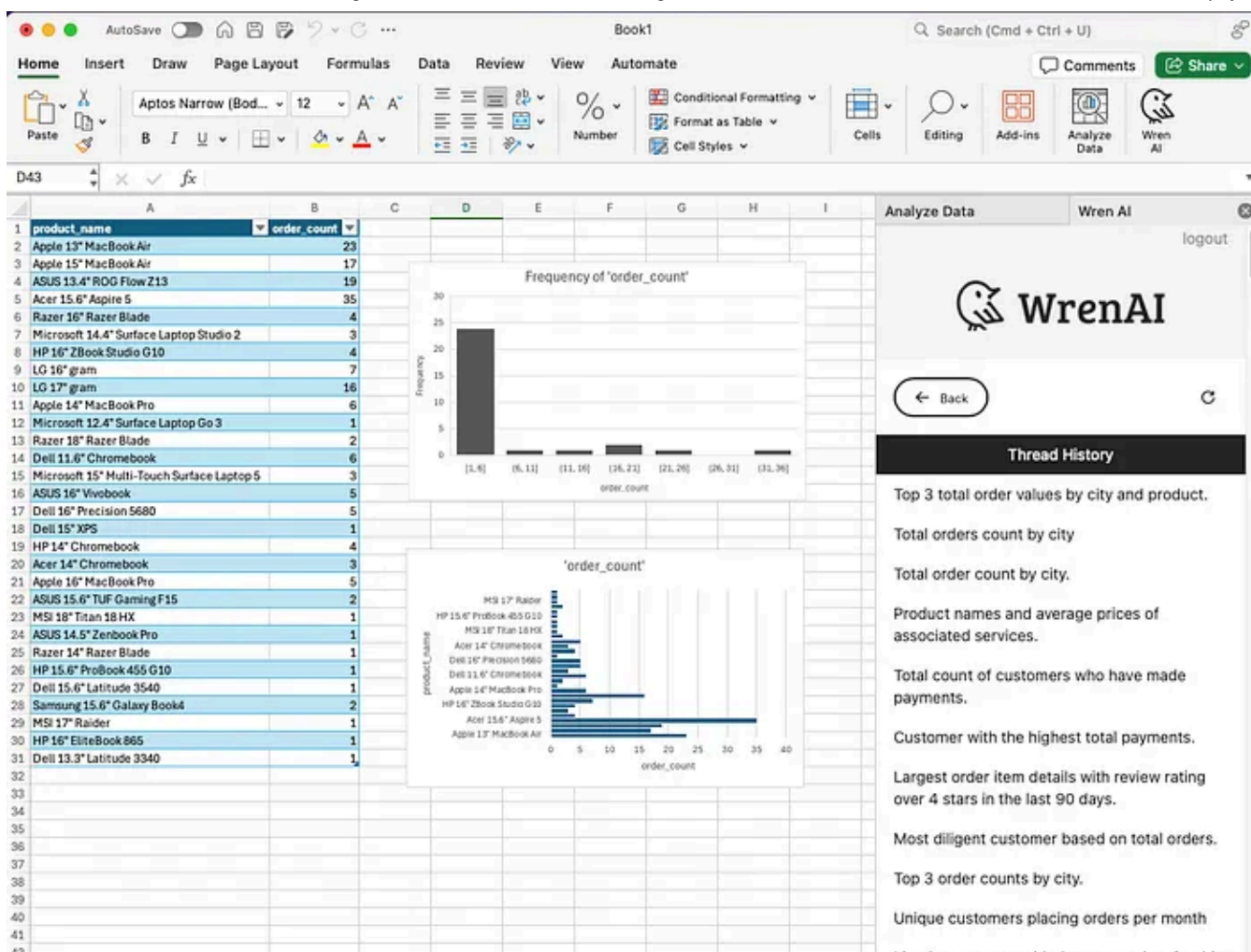
4. Connection with Excel and Google Sheets

Wren AI makes it easy to share and manipulate query results in familiar spreadsheet tools.

- **Excel Add-in:** Export data from Wren AI directly into Excel by selecting specific threads or views. This integration reduces manual data copying and speeds up analysis in the environment your team already knows.
- **Google Sheets Add-on:** Similar functionality for Google's productivity suite, so teams relying on G Suite can also enjoy seamless data exports.

Learn more:

- [Excel Add-in](#)
- [Google Sheets Add-on](#)



By incorporating text-to-chart, boilerplates, and deep spreadsheet integrations, Wren AI goes beyond text-to-SQL to deliver a comprehensive, user-friendly ecosystem for data analytics. Whether you need a quick visualization, ready-to-use templates for common data questions, or seamless export to your preferred spreadsheet tool, **Wren AI** has you covered.

Conclusion

Both Uber's QueryGPT and Wren AI share a modular design that solves text-to-SQL challenges by leveraging key steps — workspace or project segmentation, intent detection, table selection, and column pruning. This structured approach ensures accurate, efficient, and scalable SQL generation while maintaining strict data governance and security.

If you're looking for a powerful, open-source text-to-SQL solution that brings these innovations to your organization, **Wren AI** offers everything you need to transform how your team interacts with data. With **project and organization management**, **intent detection**, **step-by-step SQL breakdowns**, and **column pruning**, Wren AI ensures that data is accessible, accurate, and secure — whether you're a data analyst, engineer, or business leader.

👉 Explore Wren AI's open-source project on GitHub:

<https://github.com/Canner/WrenAI>

👉 Learn more and try Wren AI today: <https://getwren.ai/>

Start simplifying your data workflows and empower your team to get insights faster — no manual SQL writing required.

AI

Analytics

Data

Database

Sql



Follow

Published in Wren AI

608 Followers · Last published 4 days ago

Wren AI is a SQL AI Agent for data teams to get results and insights faster by asking business questions without writing SQL.



Follow

Written by Howard Chi

1.4K Followers · 273 Following

CEO & co-founder @ Canner, share thoughts and inspirations.

Responses (9)



What are your thoughts?

[Respond](#)**Stan Thompson**

Jan 11 (edited)



I'm curious about the practicality of it all. Admittedly a lot of this is over my head...but wouldn't introducing a relatively homegrown solution introduce more costs (architecture, AI/ML engineers, tooling), when it will only replace certain... [more](#)



9

[Reply](#)**Peter Heller**

Jan 9



I thoroughly enjoyed your article as it resonates deeply with my perspectives on MT (Machine Teaching), collaborative pair programming, and the iterative refinement of prompt engineering, akin to the Socratic Method of learning.

Integrating with the... [more](#)



7

[Reply](#)**ElectricLlama**

Jan 9



So 20% of the time you get an incorrect query which gives you the wrong answer but you don't care cause you just want the numbers.




5

[Reply](#)[See all responses](#)

More from Howard Chi and Wren AI

Aspect	Traditional Text-to-SQL (No Semantic Layer)	Semantics-Driven Text-to-SQL (with Wren AI)
Reliance on Model Logic	Relies heavily on model inference and guesswork	Integrates a stable layer of domain definitions
Adaptability to Schema Changes	Easily breaks under schema changes	Remains robust and accurate even as the database evolves
Handling Domain-Specific Metrics	Struggles to represent and maintain domain-specific metrics	Ensures consistent interpretation of business metrics and entities
Maintenance Requirements	Often requires continuous prompt engineering or retraining	Reduces maintenance overhead and minimizes user confusion

 In Wren AI by Howard Chi

Why the Semantic Layer is Essential for Reliable Text-to-SQL and How Wren AI Brings it to Life

Unlocking the Power of Natural Language Queries: How the Semantic Layer Transforms Text-to-SQL with Wren AI

Dec 10, 2024

 40

 1





Recent email campaigns with highest click-through rate?

Změnily se náklady na

sales?

Tranche d'âge la plus rentable de nos clients les mieux payés ?

ch social media platform is driving the most growth?


哪種行銷通路的投資報酬率最高?

ers?

가장 많은 트래픽을 유도하는 소셜 미디어 플랫폼은 무엇입니까?

erkesan untuk kempen Google Ads kami?

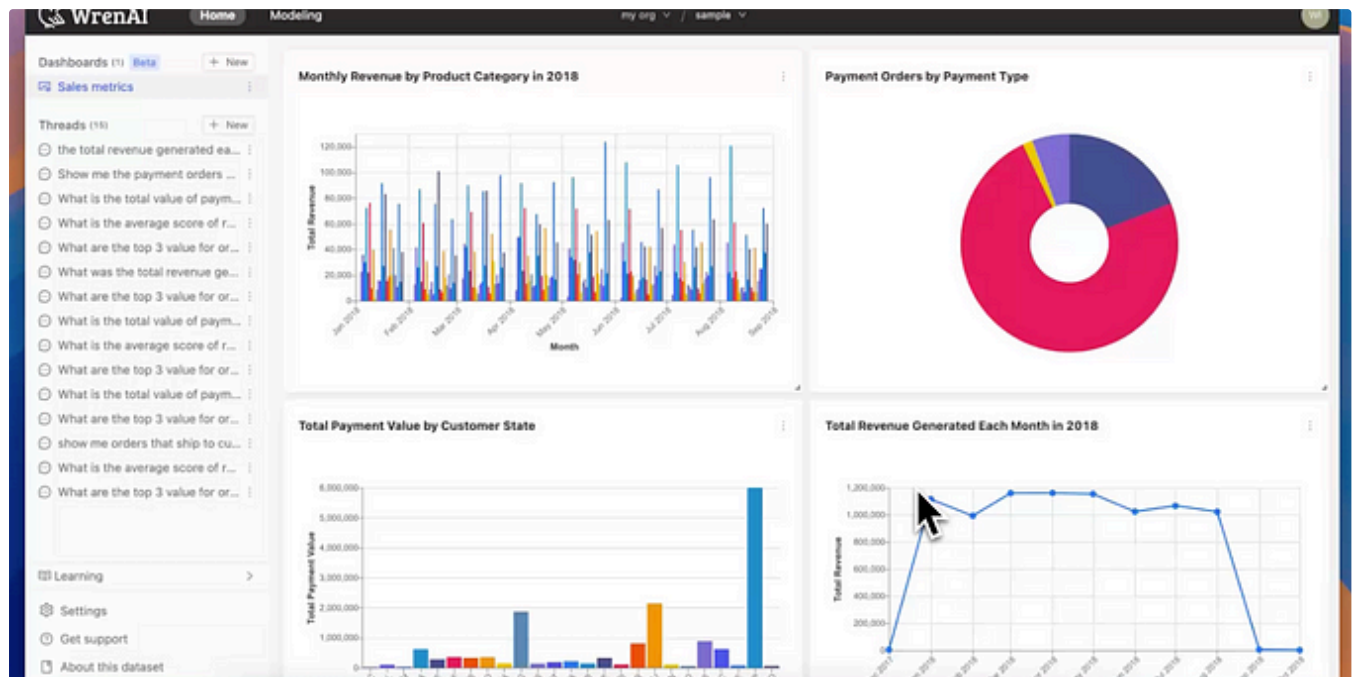
最も売上が伸びているのはどの地域ですか?

 In Wren AI by Howard Chi

Reducing Hallucinations in Text-to-SQL: Building Trust and Accuracy in Data Access

How Schema Grounding, Semantic Layers, and Iterative Validation Can Enhance Text-to-SQL Reliability

Jan 7 🖱️ 113 💬 1



In Wren AI by Howard Chi

The Future of Business Intelligence: The Generative BI (GenBI)

Discover how Generative Business Intelligence (GenBI) powered by Wren AI is transforming data access with conversational AI, real-time...

Jan 22 🖱️ 112



Llama 3



In Wren AI by Howard Chi

How to use Meta Llama 3 to query MySQL database using Ollama and Wren AI

Step-by-step tutorial on hosting private LLM endpoints through Ollama and using the latest open model Meta Llama 3 to query your MySQL

Jul 2, 2024

👏 188

💬 3

[See all from Howard Chi](#)[See all from Wren AI](#)

Recommended from Medium

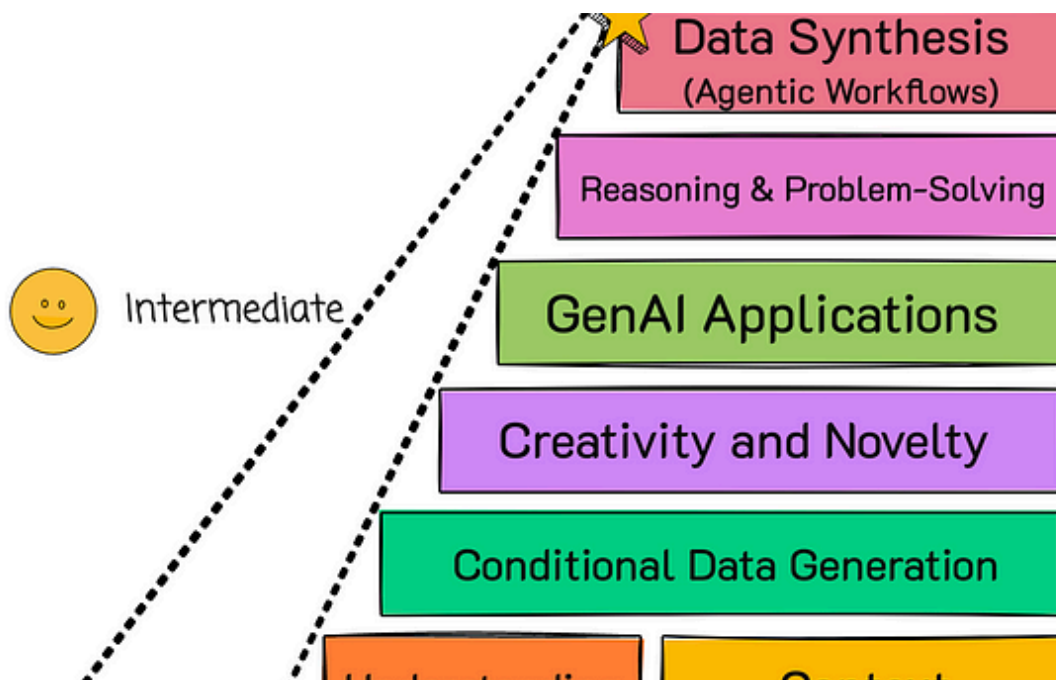


In AI Advances by Manpreet Singh

Goodbye RAG? Gemini 2.0 Flash Have Just Killed It!

Alright!!!

Feb 10 1.8K 64



Cobus Greyling

Why The Focus Has Shifted from AI Agents to Agentic Workflows

We find ourselves on a stairway from where Large Language Models were introduced to AI Agents with human like digital interactions. But...

Feb 5 851 20

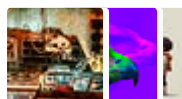


Lists



Generative AI Recommended Reading

52 stories · 1657 saves



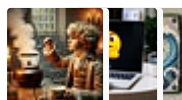
What is ChatGPT?

9 stories · 508 saves



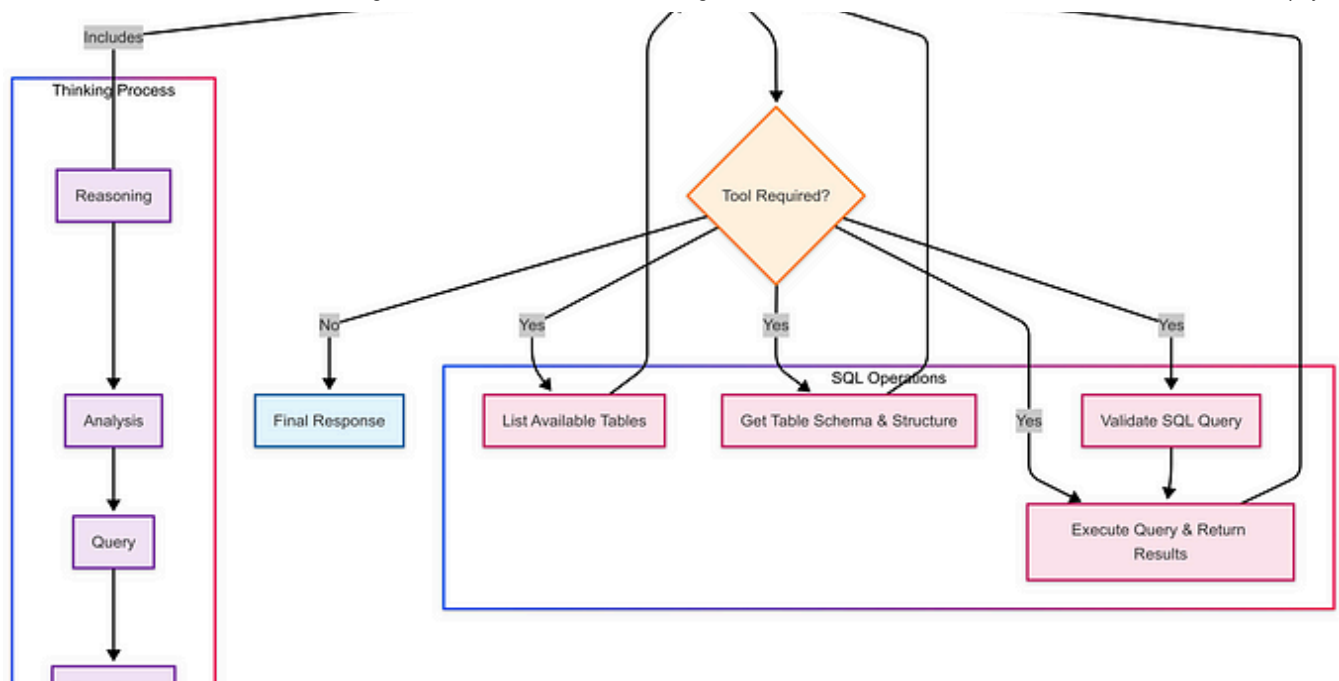
The New Chatbots: ChatGPT, Bard, and Beyond

12 stories · 549 saves



Natural Language Processing

1939 stories · 1594 saves

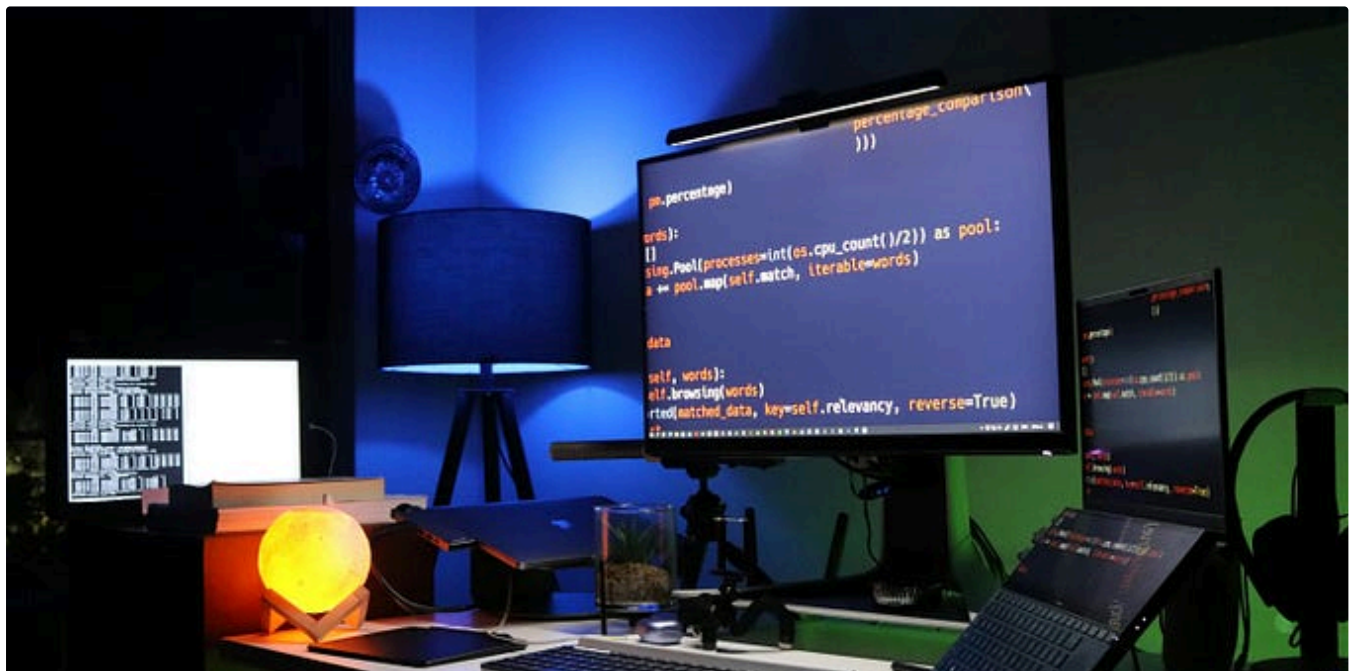


Yi Ai

Implementing Reasoning in Text-to-SQL Agents

Unlock Deeper Thinking in Any LLM

Feb 2 🖱️ 281 💬 4



Vijay Gadhave

When to Use COUNT(*) vs COUNT(1) in SQL Queries

Note: If you're not a medium member, [CLICK HERE](#)



Jan 14



336



23




	LiveCodeBench (Code Generation)	Code generation in Python. Code Generation subset covering more recent examples: 06/01/2024 - 10/05/2024	30.0%	34.3%	35.1%
Factuality	FACTS Grounding	Ability to provide factuality correct responses given documents and diverse user requests. Held out internal dataset	82.9%	80.0%	83.6%
Math	MATH	Challenging math problems (incl. algebra, geometry, pre-calculus, and others)	77.9%	86.5%	89.7%
	HiddenMath	Competition-level math problems. Held out dataset AIME/AMC-like, crafted by experts and not leaked on the web	47.2%	52.0%	63.0%
Reasoning	GPQA (diamond)	Challenging dataset of questions written by domain experts in biology, physics, and chemistry	51.0%	59.1%	62.1%

 In CodeX by Austin Starks

Google just ANNIHILATED DeepSeek and OpenAI with their new Flash 2.0 model

Three weeks ago, when DeepSeek released R1, their inexpensive reasoning model, I thought it was the pinnacle of the AI revolution.

 Feb 6  1.5K  48



20 Advanced Statistical

 Sarowar Jahan Saurav

20 Advanced Statistical Approaches Every Data Scientist Should Know



Data science is a multidisciplinary field that combines mathematics, statistics, computer science, and domain expertise to extract...

Feb 6



1.1K



15



See more recommendations