



# AI Reality Checkpoint

## The Good, the Bad, and the Overhyped

Fall 2024





Maxime Beauchemin

mistercrunch

creator of Apache Airflow and Apache Superset - founder at Preset

Edit profile

1k followers · 11 following · 139

preset-io

San Mateo, CA

maximebeauchemin@gmail.com

mistercrunch.blogspot.com

#### Organizations



- Passionate about building data tools!
- Started Apache **Airflow** at Airbnb in 2014
- Started Apache **Superset** at Airbnb in 2015
- Started **Preset** - The Apache Superset company in 2019



# AI Reality Checkpoint: The Good, the Bad, and the Overhyped

- How did we get here!?
- The ASK-AI first reflex
- Use cases
  - Data-related workflows (data eng, data analysis, ...)
  - Programming
  - Founder / information worker-related

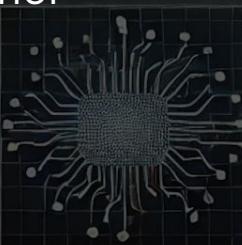
Open source maintainer

- Grips
- What's next!?



MOTOROLA

AGENDA



AI

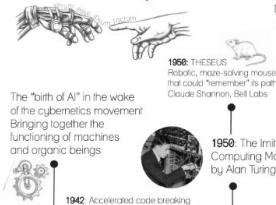


# A brief mostly complete history of Artificial Intelligence

starting around the cybernetics movement

## From cybernetics to AI

The idea that humans and machines are essentially the same gives rise to a project where scientists began to seriously consider what it would take to develop machines with human-like intelligence.



The "birth of AI" in the wake of the cybernetics movement. Bringing together the functioning of machines and organic beings.

1942: Accelerated code breaking Alan Turing used the Bombe machine to decode messages encrypted using the Enigma machine at an accelerated pace during WWII

"The Manchester Baby"

runs its first program

1948:

"Cybernetics"

the study of control and communication in the animal and the machine

by Norbert Wiener

1949:

"Brain: Or Man's Way To Think"

Mark Weingarten compares machines to human brains if we're made of

and hardware instead of flesh and nerves."

1943: Machines and behavior

"Behavior, Purpose, and Teleology"

by Rosenblueth, Wiener, & Bigelow

1943: "Artificial Neurons"

A Logical Calculus of the Ideas Immanent in Nervous Activity

by McCulloch & Pitts

The birth of neural networks

1949:

"Artificial Intelligence"

Introduced into the nomenclature



1970: "From three to eight years we will have a machine with the general intelligence of an average human being" -Marvin Minsky



1956: "AI will beat a human at chess within the next 10 years" -Herbert A. Simon



1955: First AI Workshop Proposed to be held at Dartmouth in 1956 McCarthy, Minsky, Rochester, Shannon



1966: SHAKY First robot to perceive its surroundings, navigate, plan a route, adjust for errors, and improve its abilities using English language communication Stanford Research Institute



1965: DENDRAL Export System: hypothesis formation and science Inferences & Logical Conclusions Feigenbaum & Lebedev



1963: RAND First tablet and stylus created RAND Corporation



1957: The Recurrent Network to execute shrewd, inexact form of machine learning Created by Frank Rosenblatt



1958: LISP (List Processing) First Programming Language Created by John McCarthy



1959: UNIMATE First Industrial Robot commercially available Created by George Devol



1959: "Machine Learning" Introduced into the nomenclature by Arthur Samuel at IBM



1966: The first AI program Logic Theorist - proves theorems in symbolic logic, propositional calculus, and Russell's Principia Mathematica Created by Newell, Shaw, & Simon



1957: General Problem Solver Means-End Problem Solving Newell, Simon, & Shaw



1968: The concept now known as "deep learning" is first introduced by Alexey Ivakhnenko in "Group method of data handling"



1971: AARON Autonomous Drawing Program creates the first "Art" - paints a "turle" Designed by Harold Cohen



1966: SHAKEY

First robot to perceive its surroundings, navigate, plan a route, adjust for errors, and improve its abilities using English language communication Stanford Research Institute

1980-1987  
AI Boom

1972: WABOT-1

The first "android"

Walked, communicated in Japanese and gripped objects Waseda University

1976: Computer Physical Symbol System Hypothesis "Computer science as empirical inquiry: Symbols and Search" by Newell & Simon

1973: Reduced funding for AI due to lack of promised progress in Britain

1979: Garronoid First computer champion of backgammon Created by Hans Berliner

1983: The problem of "knowledge acquisition" acknowledged as a barrier to AI progress

1984: "Winter is Coming" Schank & Minsky

1989: XCON First expert system on the commercial market Used to configure customer orders and reduce shipping times

1990: SAM Program meant to understand stories run on scripts Created by Roger Schank

1991: JABBERWACKY Amusing chatbot Created by Roger Schank

1992: Deep Blue beats Grand Master of Chess



1987: VPL Data Glove

AVR glove that controlled a virtual hand

1989: CYBERFACE VR glasses created LeapVR

1990: 1st National Conference on Artificial Intelligence Held at Stanford University, August 18-21

1991: 5th Generation Computer Project Japan spends \$650 million to develop computers that could translate and use human language + express human-level reasoning

1992: machine learning shifts from knowledge driven approaches to data-driven approaches Linear Algebra is used to advance facial recognition programs Sirosh & Kirby

1993: The "web" is invented by Tim Berners-Lee Shift from rule-based to probabilistic methods in information extraction IJCAI Watson Research Center

1994: GPU 3D graphics introduced with the PlayStation 1 home gaming console

1995: ALICE First chatbot that also collected data 1997: 90x90 pixel First social media platform

1996: ROOMBA iRobot

2002: Digital Information storage surpasses non-digital information storage WorldNet program that converts language into machine readable code is created by Google In 2016, it was found that the program had produced biased results such as referring to categories women as "decent" gender stereotypes about biased learning data

2007: UX Algorithms Used to enhance advertising and user experience

2009: WATSON created & wins on Jeopardy!

2011: SIRI Apple releases virtual assistant



2006: "Deep Learning"

coined by Geoffrey Hinton to explain new algorithm that can be trained to recognize objects and test in images and videos

2007: LSTM RNN used for handwriting and speech recognition Created by Hochreiter & Schmidhuber

2009: ROOMBA iRobot

2011: Video Recognition Google X uses a neural network of 16,000 computer processors with 1 billion connections to analyze YouTube. The identities cats as a common image.

2013: GEMINI First AI to outperform humans on massive multi-task language understanding tasks. Google DeepMind

2020: GPT-3 Uses deep learning to create codes, poetry, and other writing tasks. OpenAI

2023: GPT-4 The next iteration of OpenAI's GPT system. A large multimodal model that accepts images and text as inputs. Passed the Bar Exam scoring in the top 10%



2014: Generative Adversarial Networks produce completely new images for the first time rather than working on pre-existing images. Developed by Ian Goodfellow

2017: Google Deep Mind AI avatar teaches itself how to walk

2019: 2010: XBOX 360 Kinect First gaming hardware to track body movements and translate it into gaming directions

2021: AlphaGO beats Lee Sedol at Go

2023: GEMINI First AI to outperform humans on massive multi-task language understanding tasks. Google DeepMind

2020: SWARMET OpenAI's deep learning system oriented towards three-dimensional structures. OpenAI

2021: AlphaZero Beats Grandmaster of StarCraft II

2023: AlphaGo Beats Grandmaster of StarCraft II

2018: BERT First application of unsupervised learning model for natural language tasks. Created by GOOGLE

2021: DALL-E OpenAI's diffusion model generative AI that can process images to create accurate captions. OpenAI

2016: SOPHIA Humanoid Robot becomes the first "robot citizen". Created by David Hanson Jr.

2018: Language processing AI beats human intellect on a Stanford reading and comprehension test for the first time. Created by e-commerce giant Alibaba

2016: The Next Rembrandt Deep learning algorithm creates realistic portrait style painting using 3D printing. Facebook AI Research Lab

2017: Dialog Agents Chatbots programmed in English and designed to negotiate with each other in their own language. Facebook AI Research Lab

2015: SWARM AI A real-time online tool that makes predictions based on converging information from



2006: "BIG DATA"

Introduced into the nomenclature

2007: FERET Program Developed to create algorithms for automatic face recognition software and to create a database of faces

2009: dViDa First robot developed to handle objects by hand. robot over 3 million images across 5k categories for computer vision tasks

2010: Mechanical Turk Service developed to handle tasks over the Internet. 2011: Watson beats Ken Jennings on Jeopardy! Watson is the first computer program to win Jeopardy! ever. Watson is removed from the database

2011: dViDa First robot developed to handle objects by hand. robot over 3 million images across 5k categories for computer vision tasks

2012: Mechanical Turk Service developed to handle tasks over the Internet. 2011: Watson beats Ken Jennings on Jeopardy! Watson is the first computer program to win Jeopardy! ever. Watson is removed from the database

2013: Mechanical Turk Service developed to handle tasks over the Internet. 2011: Watson beats Ken Jennings on Jeopardy! Watson is the first computer program to win Jeopardy! ever. Watson is removed from the database

2014: Mechanical Turk Service developed to handle tasks over the Internet. 2011: Watson beats Ken Jennings on Jeopardy! Watson is the first computer program to win Jeopardy! ever. Watson is removed from the database

2015: Mechanical Turk Service developed to handle tasks over the Internet. 2011: Watson beats Ken Jennings on Jeopardy! Watson is the first computer program to win Jeopardy! ever. Watson is removed from the database

2016: Mechanical Turk Service developed to handle tasks over the Internet. 2011: Watson beats Ken Jennings on Jeopardy! Watson is the first computer program to win Jeopardy! ever. Watson is removed from the database

2017: Mechanical Turk Service developed to handle tasks over the Internet. 2011: Watson beats Ken Jennings on Jeopardy! Watson is the first computer program to win Jeopardy! ever. Watson is removed from the database

2018: Mechanical Turk Service developed to handle tasks over the Internet. 2011: Watson beats Ken Jennings on Jeopardy! Watson is the first computer program to win Jeopardy! ever. Watson is removed from the database

2019: Mechanical Turk Service developed to handle tasks over the Internet. 2011: Watson beats Ken Jennings on Jeopardy! Watson is the first computer program to win Jeopardy! ever. Watson is removed from the database



2020: SWARM AI A real-time online tool that makes predictions based on converging information from



# The Perfect Storm - How Generative AI Took Off

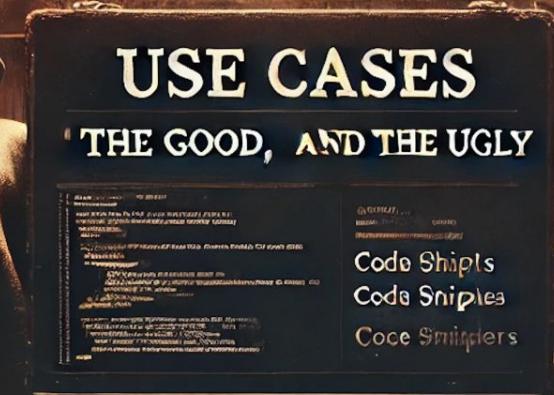
- Advances in Neural Network Topologies
- The Internet as a Massive Public Dataset
- Compute Power Explosion & Big Data
- The Transformer Architecture (2017)
- Unsupervised Learning Breakthroughs
- The "Next Word" Eureka Moment



THE CODE OF THE UGLY

... open source as a leader right out of the gate ...

COWBOY



USE CASES



'The good, the bad, and the ugly' and 'good' and the 'ugly'

# As a Data Practitioner

# As a Data Practitioner

Good



- AI assist - SQL authoring
- Scripting acceleration
- As an advisor on approach
- Suggesting code snippet
- Ideating / advising on KPIs, industry standards
- Abstract modeling questions

Bad



- Anything that requires deep or wide context

Overhyped



- the "Analyst Agent" idea
- RAG - making progress...
- Code Interpreter (hit or miss)
- Anything that requires executive skills

# Example: authoring SQL with AI assist in Preset.io

The screenshot shows the Preset.io interface with the following details:

- Untitled Query 1**: The current query tab.
- DATABASE**: Set to **bigrquery Google BigQuery**.
- SCHEMA**: Set to **githubarchive**.
- SEE TABLE SCHEMA**: A dropdown menu to select a table or search for tables.
- AI-Assist**: A text input field containing the query: `when was the last time yousoph commented or reacted to an issue or pr in the apache |`. The text "when was the last time yousoph commented or reacted to an issue or pr in the apache |" is highlighted.
- SELECT ...**: A placeholder text for the generated SQL query.
- RUN**: A blue button to execute the query.
- LIMIT: 1000**: A dropdown menu to set the result limit.
- SAVE**: A button to save the query.
- COPY LINK**: A button to copy the query link.
- RESULTS**: The active tab, showing an empty results area.
- QUERY HISTORY**: A tab to view previous queries.

# Example: Cloning table structures into another schema

in bigquery, I'd like to create/clone all the table structures from a dataset to another, what's the easiest way to do this?



To clone the table structures (without the data) from one dataset to another in BigQuery, you can follow these steps:

## 1. List all tables from the source dataset:

You can get the list of tables in the source dataset using the `bq` command-line tool:

bash

Copy code

```
bq ls --format=prettyjson [project_id]:[source_dataset_id]
```

This will list all the tables in the source dataset.

## 2. Use `bq mk` to create the new tables in the target dataset:

For each table, you can copy the schema and create an empty table in the new dataset.

Here's a simple example using `bq`:

bash

Copy code

```
bq mk --table [project_id]:[target_dataset_id].[table_name] [project_id]:[source
```

Example: Assisted me in writing “entity-centric data modeling” blog



DATA ENGINEERING

# Introducing Entity-Centric Data Modeling for Analytics

Maxime Beauchemin April 06, 2023 [LinkedIn](#) [Twitter](#) [Reddit](#) [RSS](#)

[Subscribe](#)

## Context

In this blog post, I am introducing **entity-centric data modeling (ECM)**, a new approach to data modeling for analytics that elevates the core idea of an “entity”(ie: user, customer, product, business unit, ad campaign, etc) at the very top of things. Many of the ideas here are not new and are inspired by combining concepts from [dimensional modeling](#) and [feature engineering](#) – pushing to **anchor on “entities” and to actively bring metrics and more complex data structures inside dimensions.**



THE GOOD, THE BAD THE UGLY

# As a Software Engineer

THE GOOD, THE BAD THE UGLY

# As a Software Engineer



## Good

- Autocomplete! GH copilot & Codeium plugins
- Functional programming (over object orient)
- Refactoring! js -> ts, type hints, migrations
- Unit tests generation
- Ninja templates for prompts
- Naming things



## Bad

- Anything that requires deep or wide context
- API usage - lots of hallucinations - unpredictable
- OOP - inheritance



## Overhyped

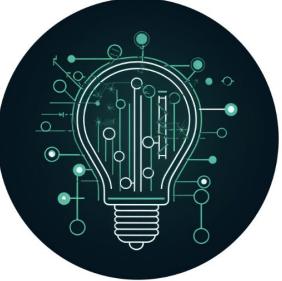
- the "Programmer Agent" idea
- Anything that requires executive skills

# Wrote 80%+ of the README and code for Promptimize

📄 README 📄 Apache-2.0 license

## 💡 ipromptimize! 💡

License Apache 2.0 · pypi package 0.2.3

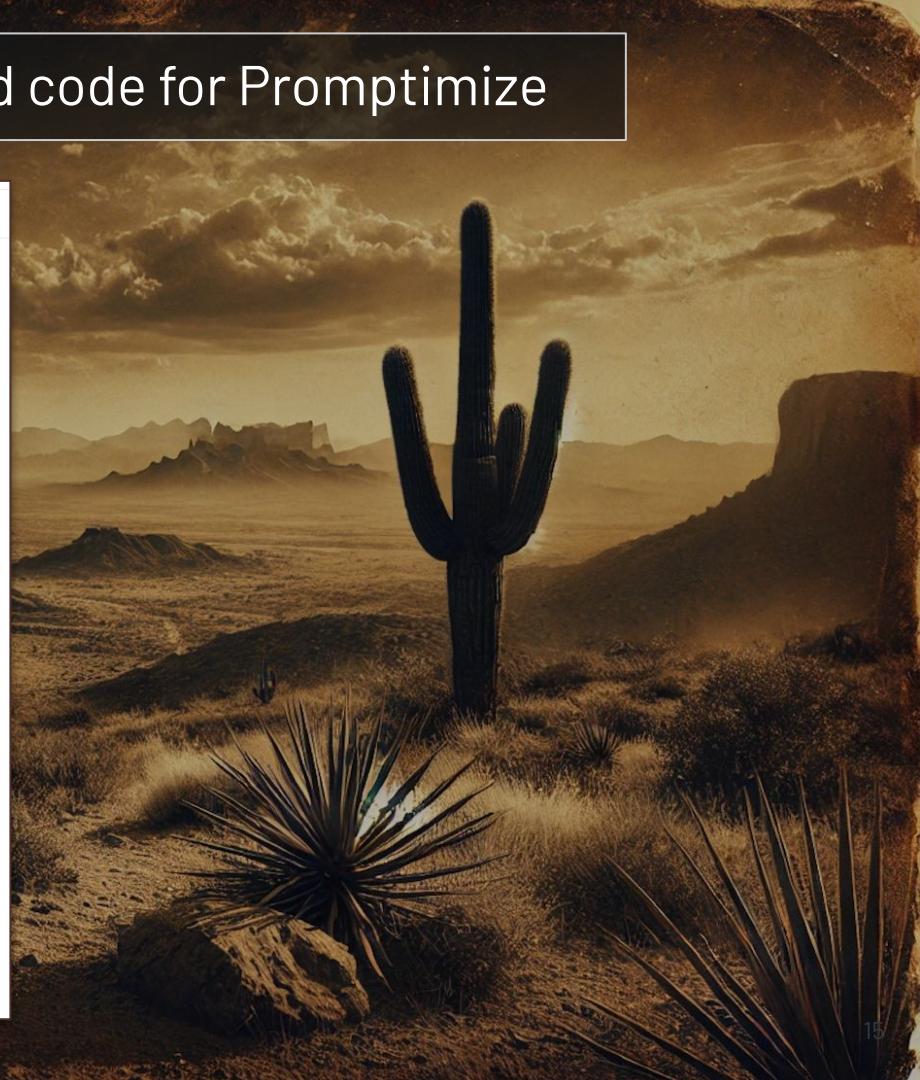


Promptimize is a prompt engineering **evaluation** and **testing** toolkit.

It accelerates and provides structure around prompt engineering at scale with confidence, bringing some of the ideas behind test-driven development (TDD) to engineering prompts.

With promptize, you can:

- Define your "prompt cases" (think "test cases" but specific to evaluating prompts) as code and associate them with evaluation functions
- Generate prompt variations dynamically
- Execute and rank prompts test suites across different engines/models/temperature/settings and compare results, bringing the hyperparameter tuning mindset to prompt engineering
- Get reports on your prompts' performance as you iterate. Answer question around how different prompt suites are performing against one-another. Which individual cases or categories of cases improved? regressed?



STARTUP FOUNDER



# As a Startup Founder

PITCH DECKS

PRODUCT RADOPS

PRODUCT ROTHHS

STARTUP  
WESTERN

# As a Startup Founder



## Good

- Product marketing
- Judgment-free thought partner / advisor
- Content generation
- Comms / advice
- Legal / advice



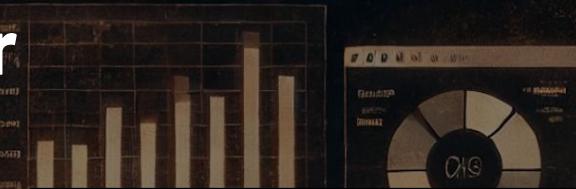
## Bad

- Running your company



## Overhyped

- Anything that requires executive skills
- Staffing - AI won't replace 80% of your opex, at least not just yet



DATA 1 DATA 2 DATA 3 DATA 4 DATA 5 DATA 6 DATA 7 DATA 8 DATA 9

# Example: Crafting a customer email

Sure thing. Here's a draft for your email:

hey I have some evidence that some of our paid customers are user service accounts and therefore breaking our TOS, can you help me write an email to the account owner?

**Subject:** Clarification Needed on Account Usage

Hi [Customer Name],

I hope this email finds you well. After reviewing your account activity, we've noticed some indications that it might be used by service accounts, which falls outside the scope of our terms of service for paid users.

To ensure we're aligned, can you confirm whether any of your seats are being used by non-human or automated accounts? If so, we'll need to discuss how to bring the account into compliance with our policies.

Please let me know if you'd like to set up a quick call to go over any questions or clarify further.

Thanks for your attention to this,

Max



The Good, and the Ugly,

Open Source

OpenSource Maintainer

Open Source

Code

Open Source Maintainer

GitHub 'Open Source Maintainer'

As an Open Source Maintainer / GitHub User

Code Shop

Open Source Maintainer

The Good, and the Ugly,

Open Source

# As an Open Source Maintainer

**Good**

- Bots



**Bad**

- Bots



**Overhyped**

- Bots



# Example: Sensitive / thoughtful community comms

0

hey! I just got CCed as the chair of Apache Superset for what seems to be a geo-political issue in relation to the maps we exposed in Superset. Note that I have no opinion on the matter, and would rather stay out of any political topic as an open source maintainer. I'm worried that our actions, whatever they may be, or inaction may be interpreted as taking a side. I'd love to push on the Apache Software Foundation to provide clear guidelines and handle communications around such topics. Here's the related thread, please provide email draft for me answer to the internal communication:

---

Original message to the apache mailing list:

Subject: Re: Map-related geopolitical concerns in Superset

Hi DW,

Thanks for reaching out and for your thoughtful approach to this issue.

As an open-source project, Apache Superset aims to be a neutral platform that serves a global audience. Given the sensitivity and complexity surrounding geopolitical topics like this, I believe it's crucial for us to avoid taking any particular stance that could be interpreted as political.

This is a space where clear guidance from the Apache Software Foundation (ASF) would be incredibly valuable. I would suggest that ASF consider establishing a set of guidelines or protocols to help projects like Superset navigate these geopolitical challenges. It would help ensure that we remain aligned with the Foundation's values and avoid unintentionally stepping into political disputes.

In this specific case, I think it's important to clarify the source of the maps we are using and ensure they follow some globally recognized standards, such as those provided by the UN. That way, we can stay focused on providing a technical solution while maintaining neutrality.

I'd love to see ASF take the lead on managing communications around these sensitive issues, ensuring that projects like ours can continue to focus on their technical mission without being drawn into political debates.

Happy to discuss further if needed, and let me know if you need anything else from me to help guide the next steps.

Best regards,

Max



# Dosu AI (dosu.dev)

Intricate issue →

Intricate answer →

Success!

apache / superset

Code Issues Pull requests Discussions Actions Projects Wiki Security Insights

flask\_log.current\_user set to None in SQL\_QUERY\_MUTATOR while using SQLLab #30168

Closed 3 tasks done dmarkey opened this issue 4 days ago · 3 comments

dmarkey commented 4 days ago

Bug description

While implementing the correct ...

dosubot bot commented 4 days ago

Hey there, @dmarkey! I'm here to help you with any bugs, questions, or contributions you have for Apache Superset. Let's tackle this issue together!

The issue you're encountering with `flask_login.current_user` being `None` during the execution of `SQL_QUERY_MUTATOR` in SQLLab sessions is due to differences in the execution context between SQLLab and navigating graphs. In SQLLab sessions, the user ID is explicitly retrieved and stored in the `SqlJsonExecutionContext` class using the `get_user_id()` function, rather than relying on `flask_login.current_user` directly [1].

To work around this, you can modify your `SQL_QUERY_MUTATOR` to use the `get_user_id()` function to retrieve the user ID. Here's an example:

```
from superset.utils.core import get_user_id
def SQL_QUERY_MUTATOR(sql, str, **kwargs) -> str:
    user_id = get_user_id()
    print(user_id) # This should print the correct user ID during a SQLLab session
    return sql
```

This approach ensures that you get the correct user information even in the SQLLab context.

To continue talking to [Dosu](#), mention `@dosu`.

dmarkey commented 4 days ago

That actually works.. first time I've been impressed by AI.



# Kapa.ai + RunLLM

(as featured on our Slack's #ask-ai)

Here:

- 2 bots offer different answers
- Long conversations ensues

achieve that?



37 replies Last reply 3 days ago



ZHE CHONG LOW 9:03 PM

how to enable CSV upload in superset



9 replies Last reply 4 days ago

## Thread



Manish Thursday at 7:03 AM

what is Other option in supported database?

4 replies



ThreadoAI APP Thursday at 7:03 AM

Hey @Manish

Superset supports a wide range of databases, including any SQL-speaking datastore or data engine that has a Python DB-API driver and a SQLAlchemy dialect. Some of the major supported databases include:

- StarRocks
- Snowflake
- SQLite
- SQL Server
- Teradata

[See more](#)



Tag me @threadoai to ask any follow up questions, always happy to assist.



RunLLM APP Thursday at 7:03 AM

The "Other" option in the supported databases allows users to connect to databases that are not explicitly listed in the predefined options. This option typically uses a SQLAlchemy URI for the connection. It provides flexibility for connecting to various databases by manually specifying the connecti...

[See more](#)



💡 Hint: Mention @RunLLM in the thread for followups.



kapa.ai APP Thursday at 7:03 AM

Hi @Manish. I'm kapa.ai, your friendly support bot. I will try to answer your question by searching through the knowledge base. 🙌

Feel free to ask a follow-up question once I've answered by mentioning me with @kapa.ai

Apache Superset supports a wide range of databases. If your database or data engine isn't on the list but a SQL interface exists, you can file an issue on the [Superset GitHub repo](#), so the team can work on documenting and supporting it.

The main criteria for a database to be supported by Superset is...

[See more](#)

Kapa.ai is still learning and improving, please let me know how I did by reacting below



# Kapa.ai - i18n!

(as featured on our Slack's #ask-ai)



O

Oguzhan Deniz Saturday at 4:15 PM

Jinja template ile iki farklı veri setinde bulunan aynı veriyi tek bir dashboardda nasıl filtreleme işlemi yapabilirim ?

4 replies

t

ThreadAI APP Saturday at 4:15 PM

Hey @Oguzhan Deniz

To filter the same data from two different datasets in a single dashboard using Jinja templates, you can follow these steps:

#### 1. Create Filters in the Dashboard:

- Start by adding filters to your dashboard. You can do this by selecting the right arrow to expand the Filter area and then choosing "+ Add/Edit Filters."
- In the "Add and edit filters" window, enter a name for your filter and select the appropriate dataset and column.

#### 2. Use Jinja Templating:

- Jinja templating allows you to create dynamic queries that can handle complex filtering logic. You can use Jinja to ensure that the filters apply correctly across different datasets.
- For example, you can create a filter that applies an OR condition instead of an AND condition by using Jinja syntax in your query.

#### 3. Map Filters to Charts:

- Ensure that the filters are correctly mapped to the charts. You can do this by selecting the "Scoping" tab in the filter settings and mapping the filter to the relevant charts.
- If you have charts from different data sources, make sure each chart's dataset is configured to accept the filter.

#### 4. Test the Filters:

- After setting up the filters, test them to ensure they work as expected. Apply the filters and check if the charts update correctly with the filtered data.

For more detailed guidance on using Jinja templating in Apache Superset, you can refer to these resources:

- [Superset SQL Templating] (<https://superset.apache.org/docs/installation/sql-template/>)
- [Intro to Jinja Templating in Apache Superset] (<https://preset.io/blog/intro-jinja-templating-apache-superset/>)

I hope this helps! If you have any further questions, feel free to ask. 😊

Sources :

1. [Intro To Jinja Templating in Apache Superset™ | Preset](#)

[See less](#)

# Dosu

dosubot bot commented on Jul 12

Hey @JatinVenkatesh! I'm here to assist you with any bugs, questions, or contributions you have while we wait for a human maintainer to look into your issue.

The `DISALLOWED_SQL_FUNCTIONS` error in Apache Superset version 4.0.2 is due to the introductory setting that restricts the use of certain SQL functions. This setting is used to enhance security by preventing unsafe SQL functions.

To resolve this issue, you can modify the `DISALLOWED_SQL_FUNCTIONS` setting in your Superset configuration file. You need to add the database names and the specific functions you need. Here is an example of how to adjust this setting:

```
DISALLOWED_SQL_FUNCTIONS: dict[str, set[str]] = {
    "postgresql": {"version", "query_to_xml", "inet_server_addr", "inet_client_addr", "clickhouse": {"url"}, "mysql": {"version"}},
}
```

You can remove or modify the functions listed under "postgresql" to fit your requirements. For instance, if you need to allow the `version` function, you can update the configuration as follows:

```
DISALLOWED_SQL_FUNCTIONS: dict[str, set[str]] = {
    "postgresql": {"query_to_xml", "inet_server_addr", "inet_client_addr", "clickhouse": {"url"}, "mysql": {"version"}},
}
```

Additionally, ensure that your database permissions and configurations are correctly set to allow the execution of the required SQL functions.

For more details, you can refer to the relevant configuration documentation [\[1\]](#).

To continue talking to Dosu, mention @dosu.

Michael S. Molina commented on Jul 17 · edited · Member

Wow. @dosubot nailed! @rusackas

PhilBlackRose commented on Aug 6

Thanks bot, updating did indeed clear the error.

jvines commented 2 weeks ago

Hi, sorry it took me so long to get to this. I'll open a PR to solve this following Dosu's suggestion.

PhilBlackRose closed this as completed on Aug 6

dosubot bot commented on Aug 6

Great to hear that updating resolved the error! Happy coding!

To continue talking to Dosu, mention @dosu.

PhilBlackRose commented on Aug 6

Thanks bot, updating did indeed clear the error.

PhilBlackRose closed this as completed on Aug 6

dosubot bot commented on Aug 6

Great to hear that updating resolved the error! If you encounter any more issues or have further questions, feel free to ask. Happy coding!

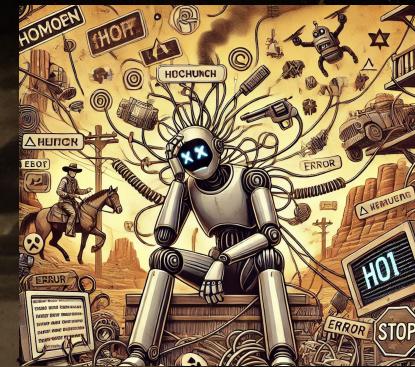
To continue talking to Dosu, mention @dosu.

Habeeb556 commented 5 days ago

Awesome!

# Gripes!

- Is GPT getting stupider over the 4-o cycle?
- GPT is incredibly verbose - STFU!
- AI always one-shots the answer, never asks for clarification
- Hallucination - guesses how APIs should work
- Don't answer from weights, search the damn web!



# Wrapping up with some related thoughts

- This is still the first inning - this talk is already obsolete!
- Intelligence / computation / energy limits? No problem.
- RAG/integration implementation are promising but clunky
- Trust still an issue - work needs expert validation
- Great for creative workflows and ideation - be careful where correctness matters
- Highly encourage people to develop their “first reflex”

*That's All  
Folks!*

WOSLEY  
TUNES