

[← Back to Articles](#)

Introducing the SQL Console on Datasets

Published September 17, 2024

[Update on GitHub](#)

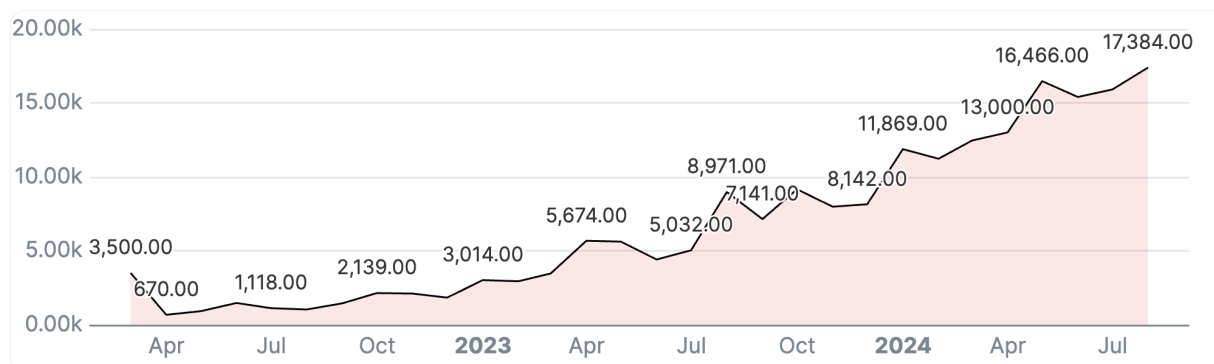
▲ Upvote **10**



[cfahlgren1](#)

Caleb Fahlgren

Datasets use has been exploding and Hugging Face has become the default home for many datasets. Each month, as the amount of datasets uploaded to the Hub increases, so does the need to query, filter and discover them.

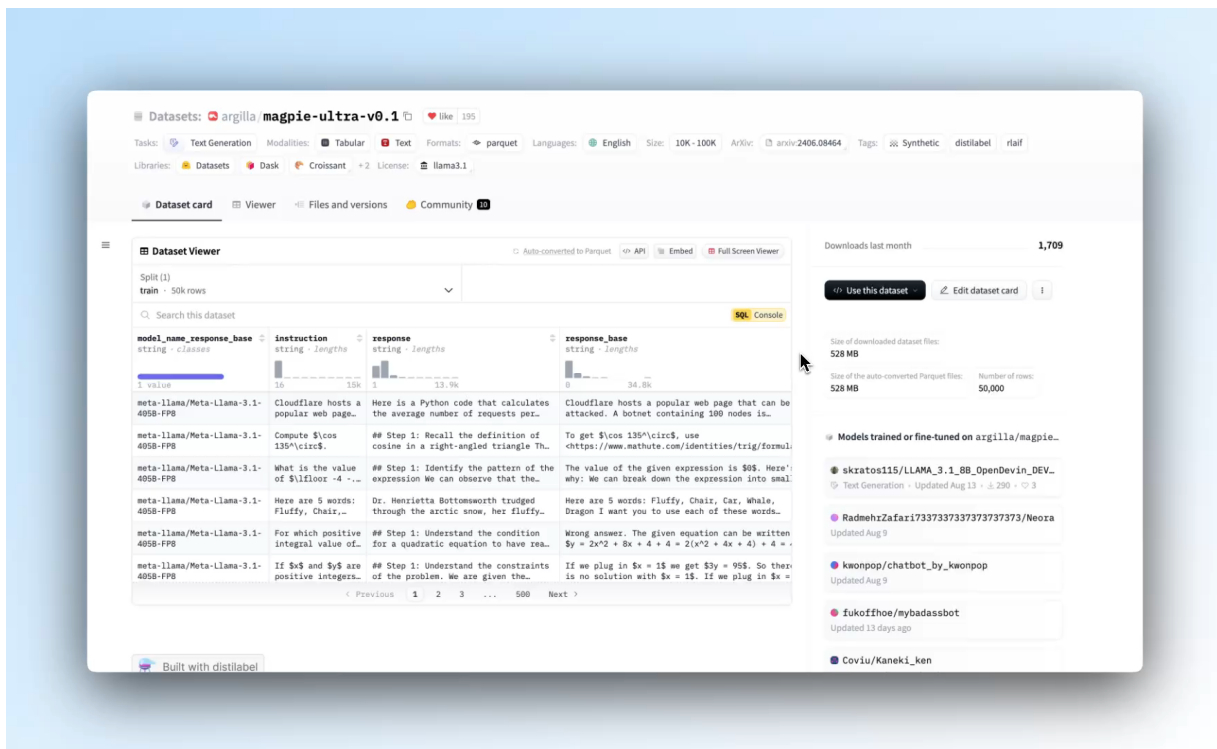


Datasets created on Hugging Face Hub each month

We are very excited to announce that you can now run SQL queries on your datasets directly in the Hugging Face Hub!

🔗 [Introducing the SQL Console for Datasets](#)

On every public dataset you should see a new **SQL Console** badge. With just one click you can open a SQL Console to query that dataset.



Querying the Magpie-Ultra dataset for excellent, high quality reasoning instructions.

All the work is done in the browser and the console comes with a few neat features:

- **100% Local:** The SQL Console is powered by DuckDB WASM, so you can query your dataset without any dependencies.
- **Full DuckDB Syntax:** DuckDB has full SQL syntax support, along with many built in functions for regex, lists, JSON, embeddings and more. You'll find DuckDB syntax to be very similar to PostgreSQL.
- **Export Results:** You can export the results of your query to parquet.
- **Shareable:** You can share your query results of public datasets with a link.

🔗 How it works

🔗 Parquet Conversion

Most datasets on Hugging Face are stored in Parquet, a columnar data format that is

optimized for performance and storage efficiency. The Dataset Viewer on Hugging Face and the SQL Console load the data directly from the datasets Parquet files. And if the dataset is in another format, the first 5GB is auto-converted to Parquet. You can find more information about the Parquet conversion process in the [Dataset Viewer Parquet API documentation](#).

Using the Parquet files, the SQL Console creates views for you to query based on your dataset splits and configs.

[🔗 DuckDB WASM](#) 🦆

[DuckDB WASM](#) is the engine that powers the SQL Console. It is an in-process database engine that runs on Web Assembly in the browser. No server or backend needed.

By running solely in the browser, it gives the user the upmost flexibility to query data as they please without any dependencies. It also makes it really simple to share reproducible results with a simple link.

You may be wondering, *"Will it work for big datasets?"* and the answer is, "Yes!".

Here's a query of the [OpenCo7/UpVoteWeb](#) dataset which has 12.6M rows in the Parquet conversion.

The screenshot shows the SQL Console interface for the OpenCo7/UpVoteWeb dataset. The query entered is:

```
select * from train
where subreddit = 'MovieSuggestions'
limit 10
```

The results are displayed in a table with 8 columns: id, parent_id, post_id, text, url, date, author, and subreddit. The first 10 rows of results are shown.

id	parent_id	post_id	text	url	date	author	sub
string	string	string	string	string	string	string	string
kfrxfca	t3_18vl1uf	18vl1uf	Dude, Where's My...	https://www.reddit.com/r/MovieSuggestions/comments/18vl1uf/movie_for_lonely_gal_on_nye/kfrxfca/	2024-01-01T09:59:58	Silent1900	Mov
kfrxfch	t3_18vihu	18vihu	Freddy Got Fingered...	https://www.reddit.com/r/MovieSuggestions/comments/18vihu/bad_movies_that_you_enjoy/kfrxfch/	2024-01-01T09:59:58	the_stubbozn_bee	Mov
kfrxrdr	t3_18vh47q	18vh47q	Showgirls (1995)	https://www.reddit.com/r/MovieSuggestions/comments/18vh47q/movies_that_are_fun_to_watch_with_friend...	2024-01-01T09:59:38	StillhasaWiiU	Mov
kfrxcch	t3_18vihu	18vihu	Anything Roland...	https://www.reddit.com/r/MovieSuggestions/comments/18vihu/bad_movies_that_you_enjoy/kfrxcch/	2024-01-01T09:59:25	Critical_Town_7724	Mov
kfrxc73	t1_kfrdm61	18vihu	This is the one with...	https://www.reddit.com/r/MovieSuggestions/comments/18vihu/bad_movies_that_you_enjoy/kfrxc73/	2024-01-01T09:59:10	Huhndiddy	Mov

The interface also shows the dataset name, a search bar, and a "Run Query" button. The results table has a "Download Results" link at the bottom right.

You can see we received results for a simple filter query in under 3 seconds.

While queries will take longer based on the size of the dataset and query complexity, you will be surprised about how much you can do with the SQL Console.


As with any technology, there are limitations.

- The SQL Console will work for a lot of queries. However, the memory limit is ~3GB, so it is possible to run out of memory and not be able to process the query (*Tip: try to use filters to reduce the amount of data you are querying along with `LIMIT`*).
- While DuckDB WASM is very powerful, it doesn't have full feature parity with DuckDB. For example, DuckDB WASM does not yet support the [hf:// protocol](#) to query datasets.

🔗 Example: Converting a dataset from Alpaca to conversations

Now that we've introduced the SQL Console, let's explore a practical example. When fine-tuning a Large Language Model (LLM), you often need to work with different data formats. One particularly popular format is the conversational format, where each row represents a multi-turn dialogue between a user and the model. The SQL Console can help us transform data into this format efficiently. Let's see how we can convert an Alpaca dataset to a conversational format using SQL.


Typically, developers would tackle this task with a Python pre-processing step, but we can show how to use the SQL Console to achieve the same in less than 30 seconds.

 yahma / **alpaca-cleaned**

Split (1)
train · 51.8k rows

Search this dataset

SQL Console

output string · lengths	input string · lengths	instruction string · lengths
		
1. Eat a balanced and nutritious diet: Make...		Give three tips for staying healthy.
The three primary colors are red, blue, and...		What are the three primary colors?
An atom is the basic building block of all...		Describe the structure of an atom.
There are several ways to reduce air pollution,...		How can we reduce air pollution?
I had to make a difficult decision when I was...		Pretend you are a project manager of a constructio...
The Commodore 64 was a highly successful 8-bit...		Write a concise summary of the following:...
The fraction 4/16 is equivalent to 1/4 becaus...	4/16	Explain why the following fraction is equivalent t...
Sophie sat at her desk, staring blankly at the...		Write a short story in third person narration...
There are two spelling errors in the sentence....	He finnished his meal and left the resturant	Evaluate this sentence for spelling and grammar...
Julius Caesar, the Roman Military general, and...		How did Julius Caesar die?
The capital city of France is Paris.		What is the capital of France?

[< Previous](#)[Next >](#)

In the dataset above, click on the **SQL Console** badge to open the SQL Console. You should see the query below automatically populated.

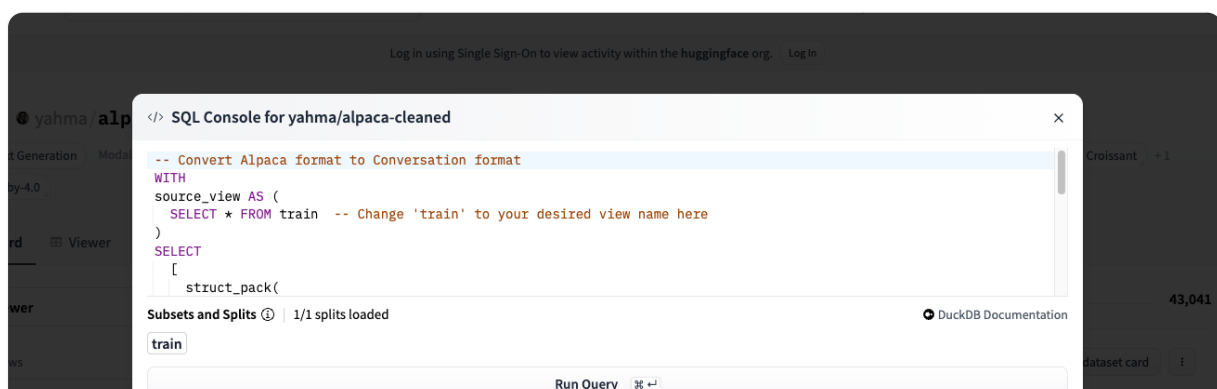
When you are ready, click the **Run Query** button to execute the query.

[SQL](#)

```
-- Convert Alpaca format to Conversation format
WITH
source_view AS (
  SELECT * FROM train -- Change 'train' to your desired view name here
)
SELECT
[
  struct_pack(
    "from" := 'user',
    "value" := CASE
      WHEN input IS NOT NULL AND input != ''
      THEN instruction || '\n\n' || input
      ELSE instruction
    END
  ),
  struct_pack(
    "from" := 'assistant',
    "value" := output
  )
] AS conversation
FROM source_view
WHERE instruction IS NOT NULL
AND output IS NOT NULL;
```


In the query we use the `struct_pack` function to create a new STRUCT row for each conversation.

DuckDB has great documentation on the STRUCT [Data Type](#) and [Functions](#). You'll find many datasets contain columns with JSON data. DuckDB provides functions to easily parse and query these columns.






Once we have the results, we can download them as a Parquet file. You can see what the final output looks like below.

 cfahlgren1/**alpaca-conversational**

Split (1)
train · 51.8k rows

Search this dataset SQL Console

conversation
list · *lengths*


2 2

[{ "from": "user", "value": "Give three tips for staying healthy." }, { "from": "assistant", "value": "1. Eat a balanced and nutritious diet: Make sure your meal...

[{ "from": "user", "value": "What are the three primary colors?" }, { "from": "assistant", "value": "The three primary colors are red, blue, and yellow. These...

[{ "from": "user", "value": "Describe the structure of an atom." }, { "from": "assistant", "value": "An atom is the basic building block of all matter and is...

[{ "from": "user", "value": "How can we reduce air pollution?" }, { "from": "assistant", "value": "There are several ways to reduce air pollution, including:...

[{ "from": "user", "value": "Pretend you are a project manager of a construction company. Describe a time when you had to make a difficult decision." }, { "from": "...

[{ "from": "user", "value": "Write a concise summary of the following:
\n\n\"Commodore 64 (commonly known as the C64 or CBM 64) was manufactured by...

< Previous Next >

Try it out!

As an another example, you can try a SQL Console query for [SkunkworksAI/](#)

[reasoning-0.01](#) to see instructions with more than 10 reasoning steps.

🔗 SQL Snippets

DuckDB has a ton of use cases that we are still exploring. We created a [SQL Snippets](#) space to showcase what you can do with the SQL Console.

Here are some really interesting use cases we have found:

- [Filtering a function calling dataset for a specific function with regex](#)
- [Finding the most popular base models from open-llm-leaderboard](#)
- [Converting an alpaca dataset to a conversational format](#)
- [Performing similarity search with embeddings](#)
- [Filtering 50k+ rows from a dataset for the highest quality, reasoning instructions](#)

Remember, it's one click to download your SQL results as a Parquet file and use for your dataset!

We would love to hear what you think of the SQL Console and if you have any feedback, please comment in this [post](#)!

🔗 Resources

- [DuckDB WASM](#)
- [DuckDB Syntax](#)
- [DuckDB WASM Paper](#)
- [Intro to Parquet Format](#)
- [Hugging Face + DuckDB](#)
- [SQL Snippets Space](#)

More Articles from our Blog

LeRobotDataset



Scaling robotics datasets with video encoding

By aliberts August 26, 2024 • ▲ 31



The 5 Most Under-Rated Tools on Hugging Face

By derek-thomas August 21, 2024 • ▲ 74



Company

[TOS](#)

[Privacy](#)

[About](#)

[Jobs](#)

Website

[Models](#)

[Datasets](#)

[Spaces](#)

[Pricing](#)

[Docs](#)

© Hugging Face