DuckDB: Crunching data anywhere, from laptops to servers

Dr. Gábor Szárnyas

goto;

2024-06-12

DuckDB Labs





About me



PhD 2014-2019



postdoc 2020-2023

graph databases

database benchmarks



developer relations 2023–

DuckDB database



What is DuckDB?





DuckDB = an analytical database management system for processing 100 GB+ data sets on end-user devices



Apple MacBook Pro up to 16 cores



Microsoft Copilot+ up to 12 cores





DuckDB's key values





Feature-rich



Demo data set



Open railway datasets:

- train services
- stations
- distances





km/h)

1



Demo



What we did not do



Simple

We did not

Register an account

Start a server / Configure a client



What we did not do



Simple

We did not

Register an account

Start a server / Configure a client



We also did not

Enter credit card details

Ask for a quota increase



What we did



Fast

Analyze 15GB data in seconds



Performance



Fast

Boost from 5 years \rightarrow 200 years of services

- 600 GB CSV files = 5.4 billion rows
- Load time: 16 minutes
- Aggregation: 22 seconds



Features



Fast



Feature-rich



Features



Fast

Full SQL support: PostgreSQL dialect Plus "friendly" SQL extensions

...back to the demo



Feature-rich



More characteristics of DuckDB





In-process

Portable





Ergonomic

Extensible



In-process architecture





Client-server architecture

Client application









In-process architecture





Truly serverless!



In-process architecture



Single-file database format



Database systems

In-process



Client-server





Transactional



Internals: Storage and execution





Storage





Storage





Storage





Execution





Execution









2 row groups

L1 cache

L1 cache





2 row groups

L1 cache

L1 cache





2 row groups



L1 cache







2 row groups





Indexing: Zonemaps

Zonemaps (min-max indexes) are created for each column in each row group

row group 1



row group 2

	date	id	type	station
	June 1	5	Sprinter	Schiphol
	June 1	6	Sprinter	Utrecht
	June 1	7	Intercity	Utrecht
	June 2	8	Intercity	Ams C
min	June 1	5	Intercity	Ams C
max	June 2	8	Sprinter	Utrecht





Portability





Portability

DuckDB is written in C++11

No external dependencies

Instead: a few inlined dependencies

→ DuckDB is **very portable**





DuckDB clients





install.packages("duckdb")



brew install duckdb





npm install duckdb



org.duckdb:duckdb_jdbc



cargo add duckdb















WebAssembly: DuckDB in the browser







Ergonomic





Supported formats and protocols





Partial reading on Parquet

Parquet files also have zonemaps These can be used for HTTP range requests (https://, s3://, etc.)

row group 1



row group 2

	date	id	type	station
	June 1	5	Sprinter	Schiphol
	June 1	6	Sprinter	Utrecht
	June 1	7	Intercity	Utrecht
	June 2	8	Intercity	Ams C
min	June 1	5	Intercity	Ams C
max	June 2	8	Sprinter	Utrecht





Supported formats and protocols











Pandas integration







IDEs for DuckDB

😨 DBeaver 22.0.1 - <memory> Script</memory>	-		×
<u>Eile Edit Navigate Search SQL Editor</u>	Database <u>W</u> indow <u>H</u> elp		
👯 🕶 🗰 🏀 💥 🛄 SQL 💌 📴 Cor	nmit 🗋 Rollback 🏋 🔻 🔒 🔛 Auto 🕙 👻 🔿 memory 🔻 🔢 < N/A > 💌 🙆 📇 👻 🔍 💌	Q	
😒 Databa × 🛅 Projects 🗖 🗆	I * <memory> Script ×</memory>		- 8
	SELECT 'Ready to fly' as first_duckdb_query		^
 ✓ Omemory 			
> 📑 information_schema			
> 📑 main	Pt.		
> iii temp			
	*		
	B		
			~
	Results 1 ×		
	🐨 SELECT 'Ready to fly' as first duckdb guery 👯 Enter a SQL expression to filter results 🕨 💌 🖉 🌾 🖪 🕅) - i (= 1	• • •
	U 1 Ready to fly		Par
			lels
Project - General X			88
Name			
> E Bookmarks			in in iteration in the second
 ER Diagrams Scripts 			E 2
	oo		
	□ O Save 🗵 Cancel 🖉 Script 🔜 🚎 🚍 🗄 🗶 🗸 > > 💷 🗇 : , ↑, 🗄 😫 200 🗭 1 🕴 Rows: 1		-
< >			
	DRoover		_
	DDeaver		

Image: Server Tree Image: Server Tree Image: Server S	temp\duckdb\alldata.duckdb Documents new 3 new 4 duckdb.sql x 14 Iat DECIMAL, 15 Ion DECIMAL 16); 17 18 INSERT INTO weather VALUES ('San Francisco', 46, 50, 0. 19 INSERT INTO cities VALUES ('San Francisco', -194.0, 53.0 20 INSERT INTO weather (city, temp_lo, temp_hi, prcp, date) 21 22 SELECT * FROM weather; 23 SELECT * FROM information_schema.tables;	25, '1994-11-27');); VALUES ('San Franc
Success:::	Chart Count = ? Time =	= 28 ms

CR SE	EATE TABL LECT * FR	E comments AS SELECT OM comments LIMIT 10	* FROM read	d_csv('/U	sers/r/Downloads/comment:	s.csv'); 	
¦ li	ne 2, column	35, location 118				Beautify XI 🗸	Run Current ≇⊷ ∨
	id	comment	from_user_id	to_user_id	created_at	updated_at	item_id is_disabled
41	21217237	* * *	16425		2016-1-26 10:27:22.440000	2016-1-27 2:15:31.764256	19441 †alse
42	21217238	⊕♥ ₩	11704		2016-1-26 10:27:33.575000	2016-1-27 2:15:31.897612	19435 false
43	21217239	Beautiful!	13151		2016-1-26 10:28:8.509000	2016-1-27 2:15:32.29877	21677 false
44	21217240	Thank you!! 🌸💞	16893		2016-1-26 10:48:1.547000	2016-1-27 2:15:32.168225	21629 false
45	21217241	♥♥♥	16893		2016-1-26 10:59:8.976000	2016-1-27 2:15:32.894351	21985 false
46	21217242	<i></i>	9691		2016-1-26 11:1:38.338000	2016-1-27 2:15:33.190972	22033 false
47	21217243	Gorgeous!	16893		2016-1-26 11:1:54.10000	2016-1-27 2:15:33.432769	21985 false
48	21217244	Yay! 👍💞	16109		2016-1-26 11:2:30.756000	2016-1-27 2:15:34.228161	22004 false
49	21217245	Beaut! 🤎	8995		2016-1-26 11:2:52.325000	2016-1-27 2:15:34.391123	22003 false
50	21217246	v	16248		2016-1-26 11:3:2.966000	2016-1-27 2:15:34.513438	21988 false
51	21217247	Gorgeous! 💞 💞	16376		2016-1-26 11:13:10.464000	2016-1-27 2:15:34.659028	22205 false
52	21217248	***	11643		2016-1-26 11:13:18.937000	2016-1-27 2:15:34.913476	22163 false
53	21217249	**	17234		2016-1-26 11:13:26.292000	2016-1-27 2:15:35.158069	22164 false
54	21217250	v" v" v "	16582		2016-1-26 11:13:34.31000	2016-1-27 2:15:35.463182	22183 false
55	21217251	Love this! 💝	9429		2016-1-26 11:13:44.95000	2016-1-27 2:15:35.683366	22165 false
56	21217252	Cool!	16567		2016-1-26 11:13:52.328000	2016-1-27 2:15:35.816546	22166 false
57	21217253	Gorgeous! 💞	13589		2016-1-26 11:14:3.188000	2016-1-27 2:15:35.963616	22168 false
58	21217254	v" v" v "	16893		2016-1-26 11:14:10.190000	2016-1-27 2:15:36.91458	22167 false
59	21217255	**	17112		2016-1-26 11:14:14.952000	2016-1-27 2:15:36.239749	22169 false
Dat	a Messa	ge Chart 182 ms		Tał	olePlus		X Q Export

📓 Č Q 💷 🔲 🔳

SQL LOCAL | DuckDB...db : SQL Query

••••

<pre>Data Catalog</pre>	Query Editor 1 select 2 drivers.surname, 3 drivers.forename, 4 drivers.national 5 avg(driver_stand 6 avg(driver_stand 7 from driver_standing 8 join drivers on driver 10 group by 1, 2, 3 11 order by avg_standin -Query Results (850 Reco surname s Hamilton Prost Stewart Schumacher Verstappen Fangio Pablo Montoya Farina Hulme Fagioli Clark Vettel Senna
CTRL+Q Quit F1 Help	Ha





0	DuckDB_in_Jupyter_Notebooks.ipynb File Edit View Insert Runtime Tools Help <u>Changes will not be saved</u>	Share
:=	+ Code + Text Copy to Drive	Conne
٩	 Querying DuckDB 	
{x]	Single line SQL queries can be run using %sql at the start of a line. Query results will be displayed as a Pa the SQL syntax highlighting!	anda
	[] %sql SELECT 'Off and flying!' as a_duckdb_column	
	 a_duckdb_column Off and flying! An entire Jupyter cell can be used as a SQL cell by placing %sql at the start of the cell. Query results will a Pandas DF. 	ll be o
	<pre>[] %%sql SELECT schema_name, function_name FROM duckdb_functions() ORDER BY ALL DESC LIMIT 5</pre>	
	Schema_name function_name	

JupySQL / Google Colab

Harlequin





State of DuckDB





Adoption



19k GitHub stars

~30k followers on LinkedIn/Twitter

1M visits per month

4M+ PyPI installs/month

















DuckDB v1.0.0: Snow Duck

2018-07-13: Initial commit 2024-06-03: v1.0.0 release

Stable storage: DuckDB's file format is backwards-compatible





Extensions





Extensions

- Powerful extension mechanism:
 - new types and functions
 - data formats
 - operators
 - SQL syntax
 - memory allocator
 - compression
- We are dogfooding many key features are DuckDB extensions:
 - httpfs
 - JSON
 - Parquet

\equiv README.md

DuckDB Extension Template $\ensuremath{\mathscr{O}}$

This repository contains a template for creating a DuckDB extension. The main goal of this template is to allow users to easily develop, test and distribute their own DuckDB extension. The main branch of the template is always based on the latest stable DuckDB allowing you to try out your extension right away.

Getting started $\ensuremath{\mathscr{O}}$

First step to getting started is to create your own repo from this template by clicking Use this template. Then clone your new repository using

```
git clone --recurse-submodules https://github 🖸 /·
```





Extension ecosystem



DuckDB

VSS

DELTA



Use cases





Reducing costs



- Local data processing
- Avoid egress fees

• Replace proprietary systems: DuckDB is free Replace distributed systems: DuckDB is more efficient



TPC-H experiments on Parquet files







TPC-H experiments on Parquet files





Last mile analytics



- Build 120 FPS dashboards!



Rill Developer	💮 metri
SOURCES +	
\vee all_stream_data	1.06
	1.001
	81.01
	44.91
	66.7%
> all_quarter	
> MODELS +	\$249
> query.sql	
	\$14.9
metric set 0	
_	

• E.g., 100TB log file processed by Spark to 50GB summary The lasts steps can be done interactively using DuckDB



2019/05/08 - 2023/11/29						
<	Ма	y ~	1	2019		>
				Th		
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

🐦 DuckDB pypi stats | Streamlit

Total Download 25M

Monthly Downloads Over Time







Local prototyping



- Experimenting with SQL queries • In a local environment
- ...with quick response time
- ...for free
- Using a popular SQL dialect helps!



Building block



- Not a database use case!

Feature/fix operation stream loading #165

szarnyasg merged 19 commits into main from feature/fix-operation-stream-loading I∽ Merged E Checks 0 Conversation 0 -O- Commits 19 ±) Files changed 102 GLaDAP commented on Jun 23, 2022 • edited -Member) ••• **Reviewers** ื szarnyasg This PR contains the following: QueryEventStreams are merged into 1 class Assignees Operation streams are loaded using DuckDB No one assigned Queries moved to their own namespace Labels GLaDAP added 19 commits last year None yet Move queries to separate namespace 5bf4581 Projects Add DuckDb for CSV parsing 3c6f682 None yet



A single data processing step in a larger system • Parquet \rightarrow CSV conversion, loading data, etc.





Education



- Easy to install
- Open-source
- Uses de facto standards (PostgreSQL dialect, Parquet, ...) Does not need configuration Does not need a DBA to run it



Limitations



Concurrency control

- ACID compliance via multi-version concurrency control (MVCC)
- Recovery using a write-ahead log (WAL)
- But: Not a good fit for write-heavy workloads



ncurrency control (MVCC) L)



No distributed execution

DuckDB only supports **single-node** execution

DuckDB can **scale up**:

- r6id.32xlarge: 1TB RAM, \$10/h
- x1e.32xlarge: 4TB RAM, \$28/h
- u7in-32tb.224xlarge, 36TB, \$408/h :)

Allows scaling for TBs of data

Typical setup:

- Store the data in S3 or S3 Express One
- Run short bursts of workloads







Business model





DuckDB Labs

A company of ~15, based in Amsterdam

Operations funded by revenue (no VC)

Consulting and support for DuckDB



databricks















DuckDB Foundation

A non-profit organization

Owns the intellectual property of the project

- 12 silver supporters
- 3 gold supporters













MotherDuck

Venture capital-funded, HQ in Seattle

Builds a cloud data warehouse using DuckDB

Reached General Availability yesterday!





CLOUD DATABASE STORAGE

SIMPLIFIED DATABASE SHARING







HYBRID QUERY EXECUTION

STRONG DUCKDB ECOSYSTEM



Summary





Spectrum of data processing systems







data size





DuckDB is an old-school analytical database for modern hardware

Open-source MIT license

Portable

Built on open standards Easy to avoid vendor lock-in

Works offline (PDF, ZIP) No need to be permanently connected

No tracking, no cookies





duckdb.org



Stay in touch



discord.duckdb.org





@duckdb



duckdb.org



Why DuckDB?

Hannes





