Extending DuckDB ETL Capabilities with Python

Josef Machytka

3 min read · 2 hours ago

DuckDB has recently become my go-to solution for small ETL tasks. It is an exceptional database created by people who quite obviously prioritize productivity and automation. Compared to traditional databases, DuckDB feels like a fresh take, free from some unnecessary constraints of 1970s and 1980s academic legacies that sometimes make database management unnecessarily complex.

In my previous articles, I covered how DuckDB can serve as a smart ETL tool for PostgreSQL and checked its ability to manage data sets that doesn't fit into memory. Recently, I encountered additional use cases where DuckDB, combined with other programming languages, proved invaluable. In this article I explore how we can easily extend DuckDB's ETL capabilities using simple Python code.

DuckDB was right from the beginning conceived as a modern replacement for SQLite. Created in academia by the National Research Institute for

Mathematics and Computer Science in Amsterdam, it embraces simplicity and effectiveness — features that suit today's data analytics demands far better than long existing legacy systems.

One of DuckDB's strong features is its deep integration with Python, reflecting the language's prominent place in both academia and industry. One of the great advantages of this integration is DuckDB direct support of Pandas and NumPy data frames, which enables seamless manipulation of data.

A Practical Example

To illustrate DuckDB's flexibility, let us look at an example inspired by my conversation with a scientist from a statistical research department. I learned they still have some old but important data stored in DBF format, a legacy data storage of dBase/XBase relational databases. They face challenge ensuring continued accessibility of this data, as support for importing DBF files in major databases is being removed over time.

DuckDB's built-in extensions currently do not support this format. However,


```
import pandas as pd
import duckdb
from dbfread import DBF

table = DBF('dbase_sample_data.dbf', load=True)
dataframe = pd.DataFrame(iter(table))

conn = duckdb.connect('testdb.duckdb')

conn.execute("""
```

```
    ATTACH 'dbname=duckdb_test user=postgres password=postgres host=localhost port=5
    AS pg (TYPE POSTGRES, SCHEMA 'public')
    """)

    conn.execute("CREATE TABLE IF NOT EXISTS pg.dbase_sample_data AS SELECT * FROM d

    conn.close()
```

In this example, I used a random DBF file from GitHub. Without knowing anything about the file's structure or contents, I was able to load it into PostgreSQL in seconds — avoiding the usual steps like defining foreign data wrapper objects or creating tables manually.

Expanding Possibilities

This approach can easily be adapted to other data formats by utilizing Python's extensive library ecosystem. With a corresponding Python library, we could also use similar code to import data from other databases currently unsupported by DuckDB extensions. However, it's worth noting that connecting Python to certain databases can pose challenges, which is beyond the scope of this article. I intend to cover these issues in future articles.

You might of course also rightly point out Pandas library is not the best tool for handling very large datasets. I completely agree, and in future articles, I will address efficient solutions for working with big data in DuckDB.

Summary

DuckDB, with its modern design and great Python integration, offers remarkable flexibility for ETL tasks. In this article I demonstrated how Python can extend DuckDB's capabilities. While this example was very trivial, focused on small data and simplicity, the potential for handling

complex data scenarios is vast, making DuckDB a powerful tool for modern data workflows.

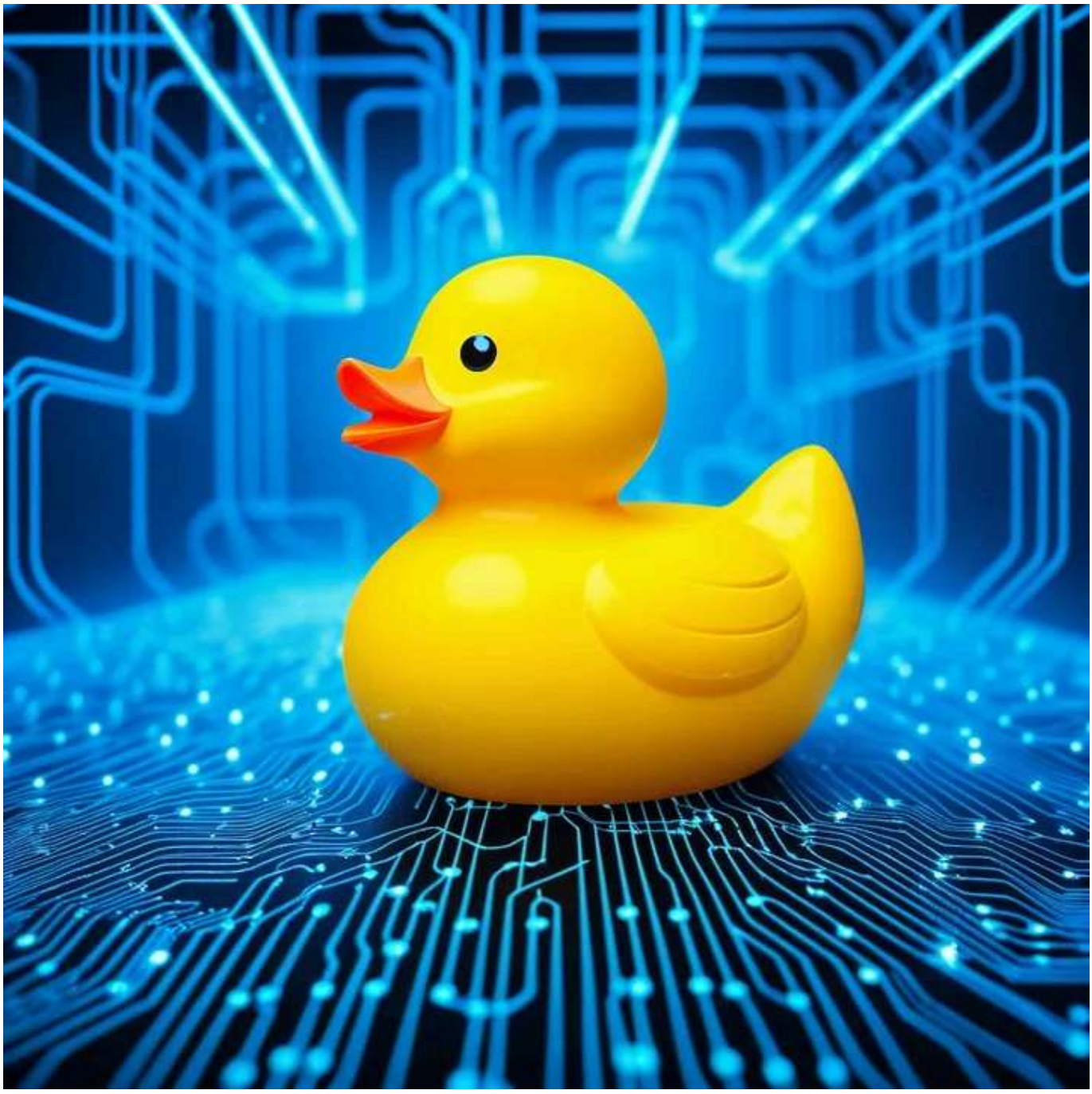


Image created by the author using DeepDreamGenerator

Duckdb Etl Pipeline Python

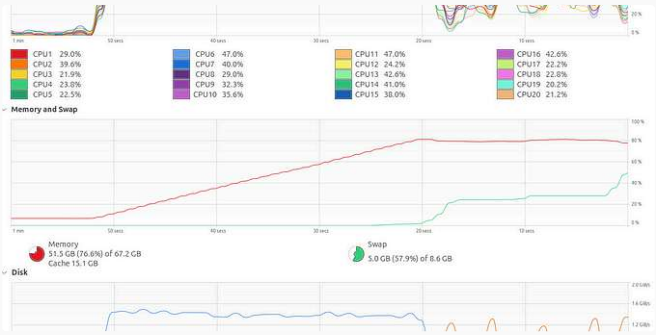
Written by Josef Machytka

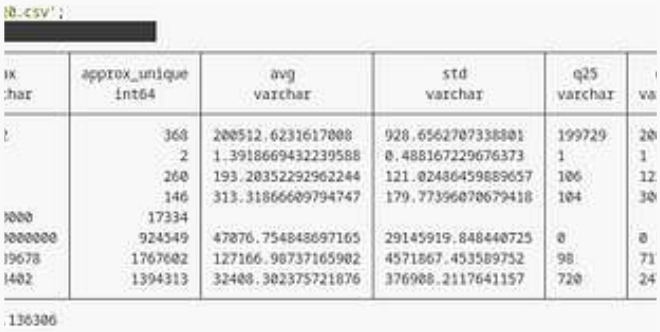
6 Followers · 2 Following

Edit profile

I work as Professional Service Consultant - PostgreSQL specialist in NetApp Deutschland GmbH, Open Source Services division.

More from Josef Machytka





	approx_unique int64	avg varchar	std varchar	q25 varchar	
	368	200512.6231617008	928.6562707338801	199729	20
	2	1.3918669432239588	0.488167229676373	1	1
	260	193.20352292962244	121.02486459889657	106	12
	146	313.31866609794747	179.77396070679418	104	30
	17334				
	924549	47076.754848697165	29145919.848440725	0	0
	1767602	127166.98737165902	4571867.453589752	98	71
	1394313	32408.302375721876	376988.2117641157	720	24

Josef Machytka

How DuckDB handles data not fitting into memory?

In my previous article about DuckDB I described how to use this database as an...

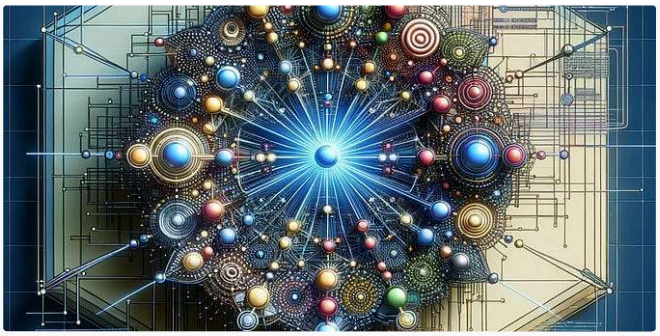
Nov 13 1

Josef Machytka

Using DuckDB as an Intelligent ETL tool for PostgreSQL

There is a lot of hype around DuckDB these days. At one PostgreSQL conference, I even...

Nov 2

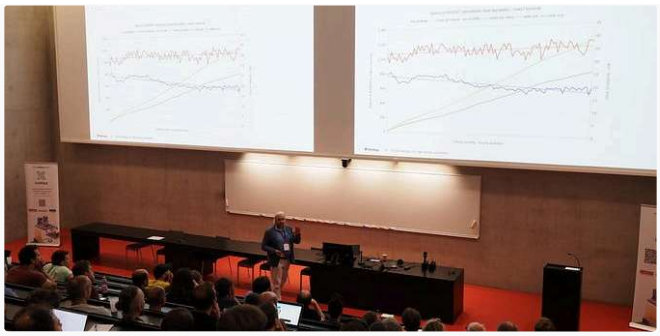


Josef Machytka

AI Hallucinations are caused by Quantum Pigeons Nesting in...

This is not a new discovery in quantum physics—it is a playful deliberate...

Nov 5



Josef Machytka

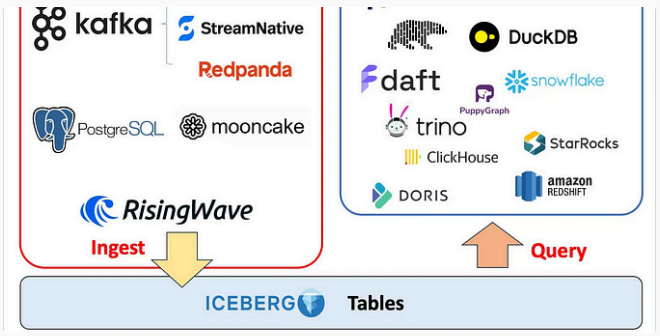
Josef Machytka: Speaker Portfolio

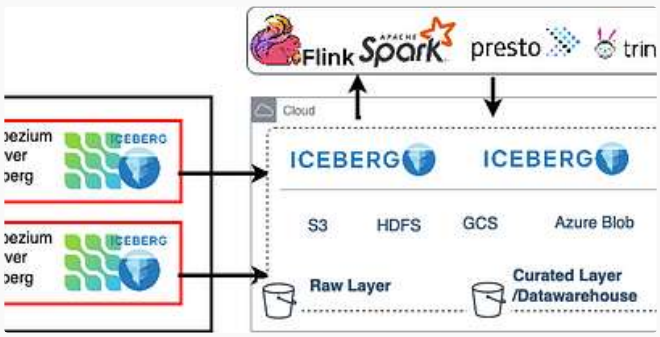
Expert Talks on PostgreSQL, Databases, Data Ingestion and Data Analysis

Nov 3

See all from Josef Machytka

Recommended from Medium





In Data Engineer Things by Yingjun Wu

Apache Iceberg Won the Future — What's Next for 2025?

RBAC, CDC, Materialized Views, and More: Everything You Need to Know About Apache...

6d ago 540 4

Ismail Simsek

Building a Data Lake with Debezium and Apache Iceberg:...

Revolutionize your data analytics with Debezium Server Iceberg! Effortlessly build...

Nov 15 19

Lists

Coding & Development

11 stories · 914 saves

Predictive Modeling w/ Python

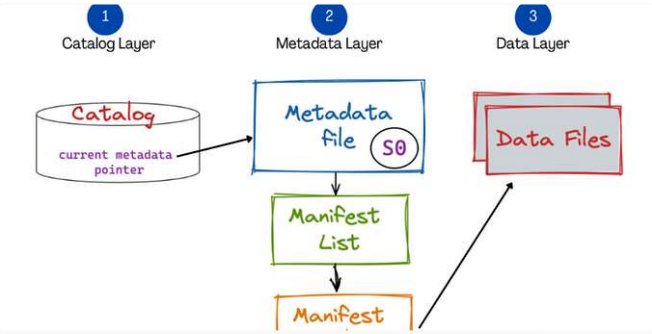
20 stories · 1684 saves

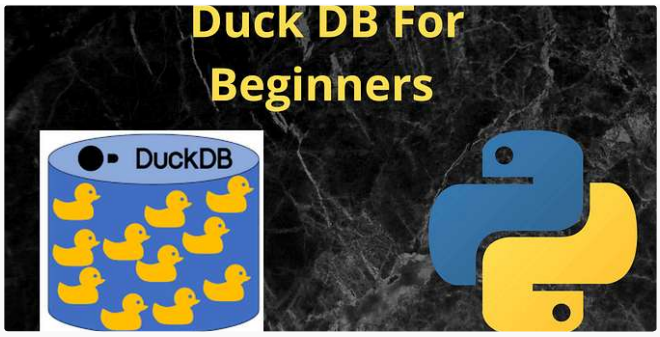
Practical Guides to Machine Learning

10 stories · 2045 saves

ChatGPT

21 stories · 888 saves





Shraddha Shetty

Understanding the Table Structure of Apache Iceberg

Apache Iceberg is a high-performance, open table format designed for large-scale...

Aug 21 2

Kevin Meneses González

Maximize Your Data with DuckDB: A Simple Guide to OLAP vs OLTP...

Introduction

Jul 22 10

Ali Raza

How to Use Python for Data Engineering

Using Python for data engineering has become standard practice in the field due to...

Nov 9 90 1

In Snowflake Builders Blog: Data Engine... by Jon ...

Best Practices for Using QUERY_TAG in Snowflake

When managing complex Snowflake queries and tracking their context, QUERY_TAG is a...

Nov 13 10

See more recommendations