credativ

Building a Data Lakehouse with PostgreSQL

Dive into Formats, Tools, Techniques, and Strategies (version 3)

Josef Machytka <josef.machytka@credativ.de>

2025-06-27 - Swiss PostgreSQL Day 2025

credativ GmbH

- Founded 1999 in Jülich, Germany
- Close ties to Open-Source Community
- More than 40 Open-Source experts
- Consulting, development, training, support (3rd-level / 24x7)
- Open-Source infrastructure with Linux, Kubernetes and Proxmox
- Open-Source databases with PostgreSQL
- DevSecOps with Ansible, Puppet, Terraform and others
- Since 2025 independent owner-managed company again

Josef Machytka

credativ

- Professional Service Consultant - PostgreSQL specialist at credativ GmbH
- 30+ years of experience with different databases
- PostgreSQL (13y), BigQuery (7y), Oracle (15y), MySQL (12y), Elasticsearch (5y), MS SQL (5y)
- 10+ years of experience with Data Ingestion pipelines, Data Analysis, Data Lake and Data Warehouse
- 2+ years of practical experience with different LLMs / AI including their architecture and principles
- From Czechia, living now 11 years in Berlin

  - LinkedIn: linkedin.com/in/josef-machytka
  - ResearchGate: researchgate.net/profile/Josef-Machytka
  - Academia.edu: academia.edu/JosefMachytka
  - Medium: medium.com/@josef.machytka
  - sessionize: sessionize.com/josefmachytka

Table of contents

All AI images without credits
were created by the author of this talk
using DeepDreamGenerator

What is Data LakeHouse?

- Data LakeHouse is a modern data platform
- Merges the best of Data Warehouses and Data Lakes
- Stores all types of data - structured, semi-structured, and unstructured
- Stores them in a single low-cost data store like cloud object storage
- Separates storage from compute, each can scale independently
- Ensures strong data consistency and integrity
- Adds the structure and management features of a warehouse
- Provides catalog layer for access control and lineage

Which components has a Data LakeHouse?

- Big variety of opinions around this term

- Modern storage format like Apache Iceberg, Hudi, Delta Lake
- Object storage with structured and unstructured data
- Data pipelines processing structured and unstructured data
- Mesh of all existing data sources in the organization
- All of it and Data Governance
- All of it and AI and ML models

credativ

War of Data Formats ended

Store only what you really need

- Store only data necessary for your operations
- And store them in efficient way

- Avoid storing data just because it ”might be useful in future”
- Law regulations require Data Retention Policies
- Most companies need in long run only aggregated data
- Some types of raw data can or must be deleted after processing
- You pay for collecting, storing, and processing data
- What about Return Of Investment from this data?

Data Warehouses vs Data Lakes Formats

credativ

- Data Warehouses - specialized relational databases with engine specific formats
- Computation and storage tightly coupled, difficult to scale horizontally
- Almost exclusively row-oriented formats, but ACID compliant, easily updatable, structured

- New online technologies required new formats - JSON, Avro, ProtoBuf, Parquet
- JSON - key-value pairs, supports nested structures
- Parquet - compressed columnar storage format, optimized for data analysis
- Avro - row oriented, schema-based, binary format
- ORC - Optimized Row Columnar - columnar storage format, for read-heavy workloads
- Data in these formats is hard to update - usually append-only, immutable
- PostgreSQL has FDW for some of these formats or can import them

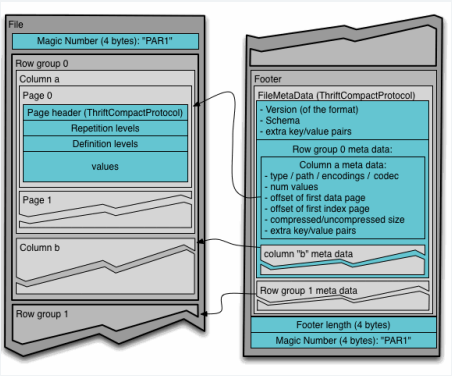
{json}

Avro

Apache orc

Parquet Data Format

- Columnar storage format optimized for reading
- Repository: github.com/apache/parquet-format
- Very efficient for numeric data types
- INT32, INT64, FLOAT, DOUBLE, BOOLEAN
- Strings are less efficient, stored as BYTE_ARRAY
- Optimized for read-heavy workloads
- Metadata include min/max values for columns / blocks
- Used in all modern Data LakeHouse solutions



(Image from the apache/parquet-format repository)

Modern Data LakeHouse Format Frameworks

credativ

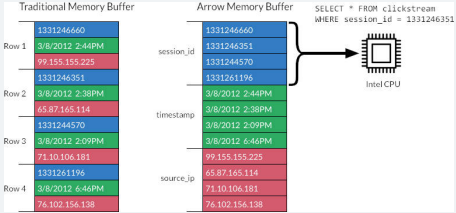
- Apache Arrow - platform for in-memory analytics, defines columnar data format

- In 2024 we had competition between three main frameworks:
- Apache Iceberg, Delta Lake, Apache Hudi
- All were based on Parquet format, designed for managing and processing large data sets
- Optimized for analytical workload, all allow limited updates and deletes with ACID transactions
- But Apache Iceberg effectively won the competition
- Thanks to its openness, advanced features, and wide adoption in the industry

Apache Arrow - platform for in-memory analytics

credativ

- Cross-language platform for in-memory processing of large data sets
- Repository: github.com/apache/arrow
- Standardized, language-independent columnar in-memory format
- Enables zero-copy reads across multiple processes
- Closely integrated with Python for data analysis



(Image from the article Apache Arrow Overview)

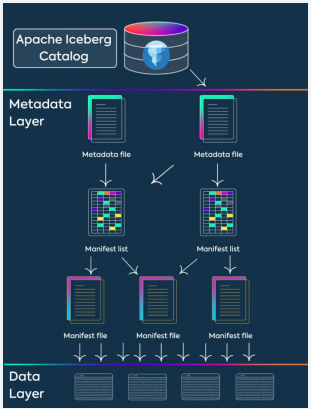
Apache Iceberg Table Format - Overview

- From Netflix, repo: github.com/apache/iceberg
- Open table format for large analytic datasets
- Supports versioning, including version rollbacks
- Features ACID transactions and schema evolution
- Implements hidden partitioning, and partition evolution
- Time travel to query historical data using snapshots
- For mostly immutable data, append-only, minimal updates/deletes

Apache Iceberg Table Format - Architecture

- Iceberg is a table format, not a database
- Designed to work with various data processing engines
- Allows multiple applications to work on the same data
- Core libraries in Java, separate projects for Python, C++, Rust, Go
- Supported by multiple engines: Spark, Trino, Flink, Presto, Hive, Dremio

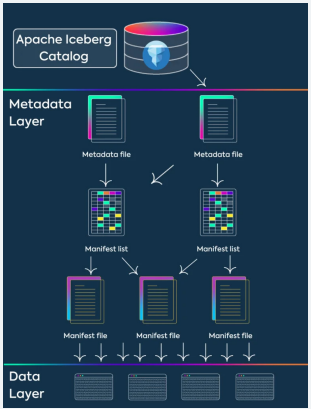
- 3 main layers:
- Catalog layer, metadata layer, and data layer
- Catalog layer: Hive, AWS Glue, JSON or custom
- Catalog tells us where to find the data



(Image from the article
What Is Apache Iceberg?)

Apache Iceberg Table Format - Architecture

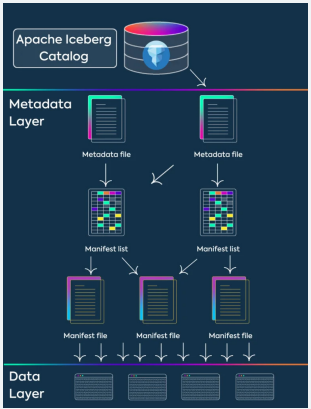
- Metadata layer: metadata files, manifests lists, manifests files
- Metadata files: hierarchical, immutable JSON files
- Tells state of each table - current schema, partitions, sort order
- Every change creates a new metadata file and snapshot
- Snapshot provides full isolation and consistency
- Snapshot is defined by a manifest list and a set of manifests files
- Manifest list and manifests files contain references to data files
- Use Avro format - also immutable like every metadata file



(Image from the article
What Is Apache Iceberg?)

Apache Iceberg Table Format - Architecture

- Snapshot - complete state of a table at a specific point in time
- Each snapshot is defined by exactly one metadata file
- Snapshots allow time travels, version rollbacks, and isolation
- Data layer - Parquet files, stored in cloud storage
- Data files are immutable, append-only
- Uses deletion vectors for row level deletes
- Stored as compressed bitmap - position of deleted row in data file
- ”Aggressive data pruning” to avoid reading unnecessary data
- Partition pruning using bounds, file pruning using min/max values

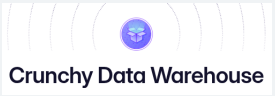


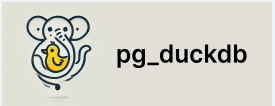
(Image from the article What Is Apache Iceberg?)

PostgreSQL and Data LakeHouse Formats

credativ

- ”PostgreSQL-in-Lakehouse” vs ”Lakehouse-in-PostgreSQL”
- Crunchy Data Warehouse
- Cybertec LakeHouse for Scalefield with AI
- Databricks Lakebase
- Pg_duckdb - DuckDB integration with PostgreSQL
- Columnar storage extensions
- Citus - distributed columnar PostgreSQL
- Pg_mooncake - mirror of PG tables in Iceberg
- ParadeDB - PostgreSQL with DuckDB and Iceberg
- Google’s AlloyDB - PG clone, columnar, vectorized engine





DuckDB – Powerful Analytical Database

credativ

- Created at the National Research Institute for Mathematics and Computer Science, Amsterdam
- Open-source, column-oriented, in-memory relational database
- Single-node database, intended for embedding in applications, like SQLite
- Designed for heavy parallel analytical workloads
- Columnar-vectorized query processing engine
- Direct selects from multiple formats and cloud storages
- Extremely portable, runs on all architectures, no dependences
- Uses PostgreSQL SQL dialect with multiple interesting improvements
- See details in my talk PostgreSQL and DuckDB on YouTube

credativ

Examples of Solutions

One Solution does not fit All Cases

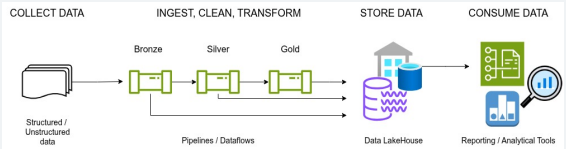
- ”One solution does not fit all cases”
- ”Look for solutions that solves your business problems”
- ”Do not chase every latest sexy buzzword”

- Classical data warehousing theory emphasized centralization
- Also some marketing articles see Data Lakehouse as centralized
- Companies try to sell ”one size fits all” solutions
- Demos with small trivial datasets do not reflect reality
- Some use cases require partial or full decentralization

(Image from the article
One Size Does Not Fit All)

Ideal Data LakeHouse Architecture

- Idealistic Data LakeHouse architecture has layers
- Data Ingestion, Processing, Transformation, Cleaning and Storage
- Described as ”medallion” architecture: Bronze, Silver, Gold layers
- Bronze layer: ingests raw data - files from cloud storage, Kafka, other databases
- Silver layer: filters, validates and cleans data and puts them into a structured format
- Gold layer: dimensional modelling, aggregation, and enrichment
- Stores pre-aggregated data ready for analysis, reporting, ML and AI
- Focus on data quality, consistency, integrity, and lineage



Examples of Data Pipelines

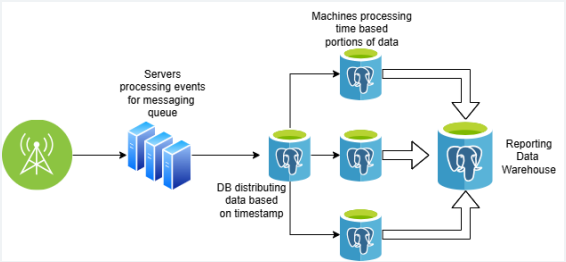
- *Telecommunication software for events from mobile networks*
- *Widget predicting sizes for online stores selling clothes*
- *Software for secure online logins and financial transactions*

- Must calculate output for the clients very quickly
- Collect and process huge amounts of data but in different ways
- PostgreSQL heavily used in all these companies
- Multiple PostgreSQL instances in each company

Telecommunication software for Events from Mobile Network

- Probes collect events from the mobile network
- Hundreds or tens of hundreds GBs per minute
- Provider needs only aggregated summaries

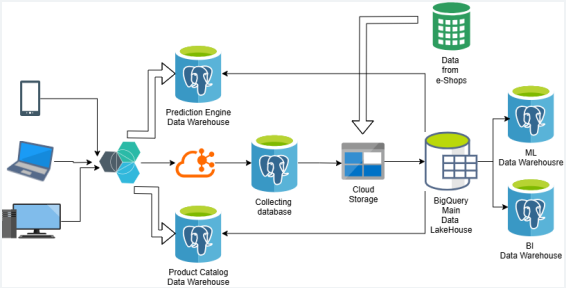
- Centralized model, storing only aggregated data
- Raw data are discarded after processing
- originally multiple PostgreSQL dbs and PL/proxy
- Later rebuilt with Kafka and quicker hardware



Widget predicting sizes for online stores selling clothes

- Calculates the best fit in dozens of milliseconds
- Prediction uses only pre-aggregated data from ML
- Scripts collect events from the website and devices
- Raw data tens or hundreds of GBs per hour

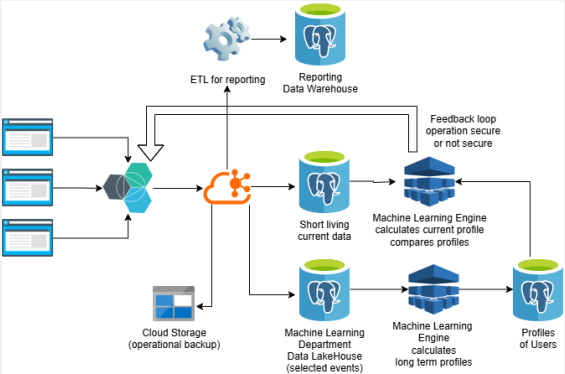
- Mixed model, main Data LakeHouse is BigQuery
- Raw events stored for 2 years for Data Analysis
- Multiple other PostgreSQL instances for other tasks



Software for secure logins and financial transactions

- Software analyzes behavior of users
- Calculates current behavioral profile
- Compares with stored profiles
- Decides if operation is secure
- Response needed in milliseconds

- Strongly decentralized model
- Storing only aggregated data
- PostgreSQL used multiple times
- Raw data tens or hundreds of GBs per minute
- Discarded soon after processing



PostgreSQL as Important Part of Data Pipelines

credativ

- PostgreSQL plays multiple roles in Data Pipelines
- Used both in streaming and batch processing
- Data ingestion / processing tools have PostgreSQL connectors
- Airbyte, Airflow, Apache Nifi, Debezium, Apache Spark, Flink, dbt
- PostgreSQL has powerful partitioning and indexing features
- Integration with DuckDB can speed up data ingestion and processing

credativ

Data Governance & Legal Aspects

Discipline is essential

- ”Discipline is essential, you foolish lads!”
  Without it, you would be climbing trees like monkeys!”

- Every Data Lakehouse requires clear Data Governance
- Properly defined Data Life Cycles are crucial
- Clear Data Catalog and Lineage are necessary
- Without these, we can ”climb” every new technology
- Jumping ”like monkeys” from one technical buzzword to another
- But we will end up in the same mess...

(Image by Mikoláš Aleš
from the book
"The Good Soldier Švejk")

Do not be overwhelmed by Data Governance

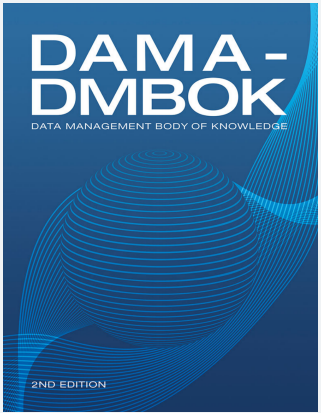
- Data Governance is also full of many buzzwords
- Abundance of articles and books on Data Governance
- Majority of them overly maximalistic and complex
- Marketing articles always sell something
- Some proprietary solution, or consulting services

- Big companies really need complex Data Governance
- But smaller companies shall start with simple rules

(Picture from article Data Governance)

DAMA - Data Management Body of Knowledge

- Source of Data Governance best practices and standards
- DAMA International is a non-profit organization
- Website: data.org
- DAMA-DMBOK2 - guide to Data Management
- Very detailed information on Data Governance



Business will suffer with poor quality data

- If data belong to no one, no one will care about quality
- If no one checks data quality, it will be poor

- **Data Quality** - garbage in, garbage out
- To know what is garbage, you need to know what is good
- Basic **Data Catalog** and **Data Definitions** are needed
- **Quality Checks** and **Data Profiling** based on Data Catalog
- **Data Producers/ Owners/ Stewards** responsible for data quality

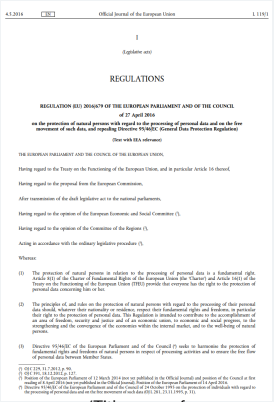
(Title page of
EU Commission Data Governance
Document)

Security and Privacy are crucial

- **Data Security** - protect data from unauthorized access
- Security is about safeguarding data
- **Data Privacy** - protect data from unauthorized use
- Privacy is about safeguarding user identity

- **GDPR** - General Data Protection Regulation
- **CCPA** - California Consumer Privacy Act
- **HIPAA** - Health Insurance Portability and Accountability Act

- Principle of Least Privilege & Only for limited time
- Minimize use of customer data, minimize access to them
- Delete or anonymize data at the end of their lifecycle



4.5.2016 EN Official Journal of the European Union L 119/1

I

(Legislative acts)

REGULATIONS

REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

of 27 April 2016

on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)

(Text with EEA relevance)

THE EUROPEAN PARLIAMENT AND THE COUNCIL OF THE EUROPEAN UNION,

Having regard to the Treaty on the Functioning of the European Union, and in particular Article 16 thereof,

Having regard to the proposal from the European Commission,

After transmission of the draft legislative act to the national parliaments,

Having regard to the opinion of the European Economic and Social Committee (1),

Having regard to the opinion of the Committee of the Regions (2),

Acting in accordance with the ordinary legislative procedure (3),

Whereas:

(1) The protection of natural persons in relation to the processing of personal data is a fundamental right. Article 8(1) of the Charter of Fundamental Rights of the European Union (the 'Charter') and Article 16(1) of the Treaty on the Functioning of the European Union (TFEU) provide that everyone has the right to the protection of personal data concerning him or her.

(2) The principles of, and rules on the protection of natural persons with regard to the processing of their personal data should, whatever their nationality or residence, respect their fundamental rights and freedoms, in particular their right to the protection of personal data. This Regulation is intended to contribute to the accomplishment of an area of freedom, security and justice and of an economic union, to economic and social progress, to the strengthening and the convergence of the economies within the internal market, and to the well-being of natural persons.

(3) Directive 95/46/EC of the European Parliament and of the Council (4) seeks to harmonise the protection of fundamental rights and freedoms of natural persons in respect of processing activities and to ensure the free flow of personal data between Member States.

(1) OJ C 229, 31.7.2012, p. 90.
(2) OJ C 391, 18.12.2012, p. 127.
(3) Position of the European Parliament of 12 March 2014 (not yet published in the Official Journal) and position of the Council at first reading of 8 April 2016 (not yet published in the Official Journal). Position of the European Parliament of 14 April 2016.
(4) Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data (OJ L 281, 23.11.1995, p. 31).

(Title page of
GDPR official text)

PostgreSQL and Data Governance

credativ

- Data Governance uses mostly external tools and processes
- Great Expectations / dbt for Data Quality checks
- Apache Atlas / OpenMetadata for Data Catalog/Lineage
- OpenLineage for Data Lineage for AI / ML
- Marquez for open source Metadata Service

- Role based access control (RBAC) - permissions for roles
- Row Level Security (RLS) - permissions for rows
- But causes performance issues in queries with many joins
- Data encryption at rest / in transit - Transparent Data Encryption (TDE)
- Auditing and Accountability - pgAudit, pgBadger extensions
- Improvements necessary to not leak sensitive data in logs

credativ

AI and Data LakeHouse

Over Promising AI Marketing Hype

credativ

- Naive AI hype is already mostly over
- Many people now have a good understanding of AI capabilities
- Signs ”AI powered” or ”AI driven” lost their marketing magic

- Usefulness of AI in Data LakeHouse depends on use cases
- Commercial AIs can lead to privacy and security issues
- Local Open Source AI solutions give more control
- But are usually not as powerful and need good hardware

AI Answers Based on Probability

- Large Language Models (LLMs) with Transformer architecture
- Use ”Attention mechanism” to understand context
- Fully depend on quality of training data or internet data
- ”Garbage in, garbage out” always applies
- Answers are the ”most probable”, not necessarily ”correct”
- For LLMs, there is no ”correct” or ”incorrect” answer
- Answers depend on activation of semantic associations
- Prompt engineering and system prompts are crucial
- ”*Just because it sounds plausible, doesn’t mean it’s true*”

See more in my talks on YouTube
(click on icons above)

Problematic Usability for Niche Topics

- LLMs absolutely depend on the quality of data
- Amount, quality and topic coverage are crucial
- LLMs work amazingly for general topic data
- Like invoicing, financial reports, warehouse management
- If we have mainly these use cases, AI is perfect for us

- Specialized topics lead to many hallucinations
- But we can definitely use AI for brainstorming
- It can give us new ideas and perspectives
- But we must always double-check the results

Most Common Issues with AI Outputs

- **Overgeneralization**: wrong conclusions due to biased data
- **Misinterpretation**: wrong conclusions due to wrong context
- **Underfitting**: model too simple to capture details, too general
- **Overfitting**: AI specialized on training data, cannot generalize

- **Overinterpretation of the Input**: wrong conclusions due to incomplete input, hallucinates missing parts of the input

- **Out-of-distribution Generalization**: wrong conclusions due to topic not covered by training data

Some Older AI Promises stayed Unfulfilled

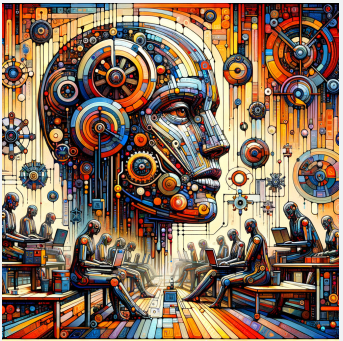
credativ

- **Fine-tuning** on domain specific data can shift performance
- Model specializes on new data
- But struggles with more general data
- Can even lead to ”catastrophic forgetting”

- **Retrieval-Augmented Generation (RAG)**
- Depends fully on the quality of additional data
- Very useful for chat-bots, and help-desk systems
- Not that great for analysis of complex data
- Highly specific data require examples and explanations

Will AI-agents do better?

- **AI-agents** are new hype
- They can run additional tasks like internet browsing
- Could also run machine learning models
- Could use multiple knowledge sources
- Capable of multi-step reasoning
- But still depend on quality of LLMs
- We can expect best performance on well known topics
- Niche topics can lead to multiple levels of hallucinations



Vibe Programming / Vibe Data Analysis

- Vibe Programming means programming with AI
- AI generates code based on user description
- This way AI can generate code or SQL for data analysis
- It works nicely for well documented mainstream topics
- With Python and SQL have AI answers the highest success rate
- But it is still only like 60% - 70% correct
- It can replace boring manual repetitive tasks
- It can help with brainstorming new ideas
- Also deep research of online resources can be very useful
- But it still has problems with niche topics

PostgreSQL and AI + ML

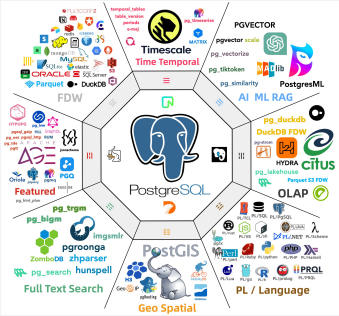
credativ

- AI, ML and PostgreSQL are a great match
- Has multiple extensions for AI and ML
- pgvector for vector similarity search for RAG
- Timescale pgvectorscale improved pgvector extension
- Timescale pgai automates creation of embeddings for RAG
- PostgresML for machine learning in PostgreSQL

Summary

Simon Riggs on PG conf EU 2023:

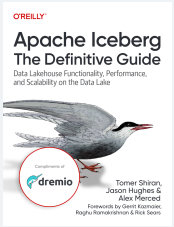
"Maybe in 20 years, everything will be done in PostgreSQL."

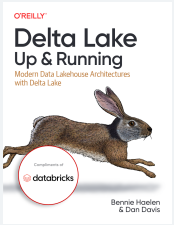


(Image from the article Postgres is eating the database world on PIGSTY (PostgreSQL In Great STYle) blog)

Other Resources Used for the Talk

- Articles:
  - What is a Data Lakehouse?
  - History and evolution of data lakes
  - What is Data as a Product
  - Apache Iceberg main web page
  - Dremio.com: What is Apache Iceberg
  - Sqream.com: What is Apache Iceberg
  - Estuary.dev: Apache Iceberg vs Hudi
- E-books:
  - T.Shiran, J.Hughes, A.Merced: Apache Iceberg, The Definitive Guide - O'Reilly
  - Bennie Haelen, Dan Davis: Delta Lake, Up & Running - O'Reilly
  - Dremio white paper: Optimizing the supply chain with a data lakehouse
  - A.Kaplan, A.Kara: Data Lakehouse for Dummies - Databricks
- Paid tier ChatGPT-4o/ o1, Google Gemini Advanced 2.5 Deep Research





credativ

Questions?