credativ

Iceberg Ahead!

PostgreSQL Sailing Confidently in Data Lakehouse Waters

Josef Machytka <josef.machytka@credativ.de>

2025-10-15 - PostgreSQL Day Ibiza 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, 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
- 33+ 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
- 3+ years of practical experience with different LLMs / AI including their architecture and principles
- From Czechia, living now 12 years in Berlin
- Often presenting at PostgreSQL Conferences and MeetUps (15+ talks)

- 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

In 20 years…

- Simon Riggs: The Next 20 Years (PGConf.EU 2023)

> What's going to happen over the course of 20 years is that Postgres will provide options and extensions to fully and successfully address every use case.
>
> In the future everything will be Postgres in much the same way as Linux has replaced almost every other operating system.

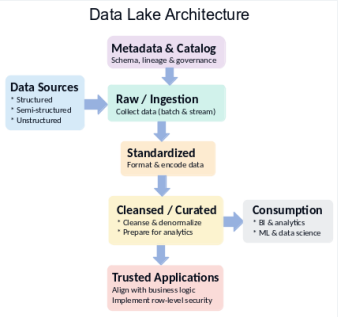
Image (c) Katerina Machytkova

credativ

Data LakeHouses: This Is the Way

Unflexible Schemas vs Unmanageable Data Swamps

- **Data Warehouses**: Optimized for structured, stable data
- Rigid schemas limit flexibility and adaptability
- Schema changes are complex and resource-intensive
- Often use proprietary, engine-specific data formats
- Scaling is costly, usually only vertically
- **Data Lakes**: Store all data types at massive scale
- Without Governance can turn into unmanageable ”data swamps”
- Lack of structure and catalog makes data discovery difficult
- Query performance suffers with complex or large datasets



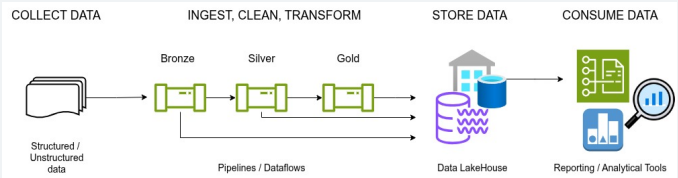
Hitchhiker's Guide to the Data LakeHouse

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

AI images created by the author
using DeepDreamGenerator AI tool

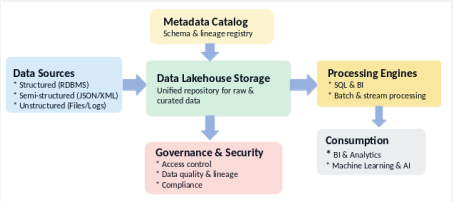
Data LakeHouse Architecture – Lower Decks

- Modern storage formats, cheap and scalable object storage for all data
- 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



Data LakeHouse Architecture - Upper Decks

- Metadata and catalog services for schema management
- Robust data governance, quality checks
- Fine grained access control layers
- Consumption: advanced analytics, BI
- Integration with AI and ML tools

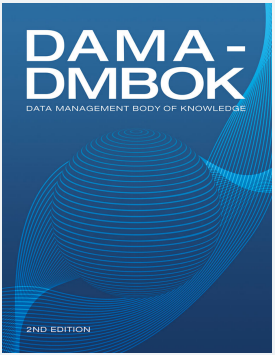


credativ

A Quick Guide to Upper Decks of Data LakeHouses

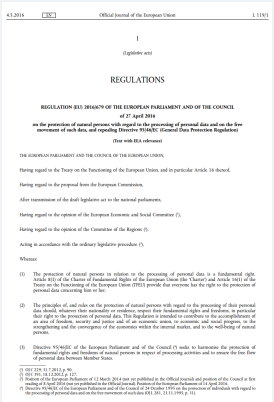
Data Governance: Rules of the Order

- **Best Practices**: DAMA-DMBOK2 framework adoption
- Data Management Association - Data Management Body of Knowledge
- **Start Simple**: Begin with essential, clear governance rules
- **Data as a Strategic Asset**: Assign ownership and stewardship
- Ensure data quality via catalogs, definitions, and profiling
- **Discipline and Consistency**: Data lifecycles and lineage
- Catalog and track data to prevent chaos and technical debt
- **Continuous Process**: Governance is ongoing and organization-wide



Security and Privacy are crucial

- Follow **GDPR**, **CCPA**, **HIPAA** - base line regulations
- **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
- 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).

AI in Data LakeHouse: Current Reality

- AI hype is fading; users now recognize real-world limits
- AI effectiveness in Data LakeHouse depends on training data
- Use case selection is critical for successful AI outcomes
- Excels with mainstream topics, struggles in niche domains
- LLMs generate probabilistic answers, not guaranteed facts
- Prompt engineering and validation help avoid hallucinations
- Commercial AIs raise privacy and security concerns
- Open-source/local AIs offer control but need strong hardware

AI Tools, Challenges, and Opportunities

- Common issues: overgeneralization and misinterpretation
- Overfitting, underfitting, and hallucinations are real risks
- RAG can improve results with relevant, specific, up to date data
- AI-agents can drastically improve productivity and quality
- ”Vibe programming” and code analysis can boost developer output
- Use AI for automation, brainstorming, and repetitive tasks
- Always validate AI outputs, especially for specialized topics

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

credativ

Data Formats: From Relational Rigidness to Flexible Dance on Parquet

Golden Rules of Data Storage

- Collect with a clear purpose
- Only data essential for your business operations
- Avoid collecting data only for potential future use
- Use most efficient storage formats & methods
- Implement data retention policies as required by law
- Retain only aggregated or summarized data long-term
- Delete raw data after processing if not needed anymore
- Evaluate ROI (Return on Investment) for stored data

Relational Data are not Predominant Anymore

- 15-20 years ago, data stored mainly in relational databases
- Storage was tightly coupled with compute resources
- Mostly row-oriented storage, proprietary formats
- Data silos: many isolated databases in companies
- Data Warehouses consolidated data but changes were hard
- Scaling required more HW resources on database server
- Horizontal scaling very difficult

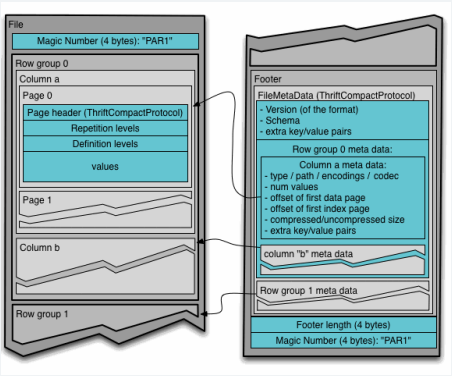
Data Lake Era – New Formats and Architecture

credativ

- New technologies in 2000s-2010s required new storage formats and architectures
- Decoupled computation and storage enable flexible, independent scaling
- Data typically immutable and append-only - simpler consistency model and scaling
- PostgreSQL supports these formats via Foreign Data Wrappers (FDW) or import tools
- **JSON**: text-based, supports nested and hierarchical data structures
- **Parquet**: compressed, columnar format, optimized for analytics workloads
- **Avro**: row-oriented, schema-based, binary serialization format
- **ORC**: columnar format, optimized for high-throughput read operations

Parquet Data Format

- De facto standard for Data Lakes/LakeHouses
- Columnar storage format optimized for reading
- Very efficient for numeric data types
- INT32, INT64, FLOAT, DOUBLE, BOOLEAN
- Strings are less efficient, stored as BYTE_ARRAY
- Supports various compression algorithms
- Optimized for read-heavy workloads
- Format is self-describing - schema in metadata
- Metadata include min/max values for columns / blocks



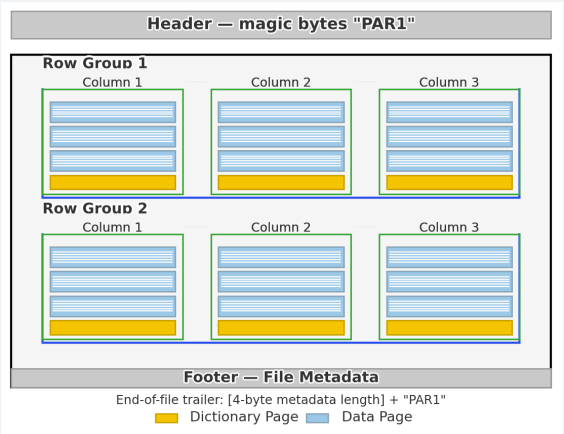
(Image from the apache/parquet-format repository)

Parquet Data Format – Structure

- File is divided into row groups
- Each row group contains column chunks
- Each column chunk is divided into pages
- Pages are the smallest units of data
- PAR1 - 4 bytes of characters 'P', 'A', 'R', '1'
- File starts and ends with PAR1 sequence

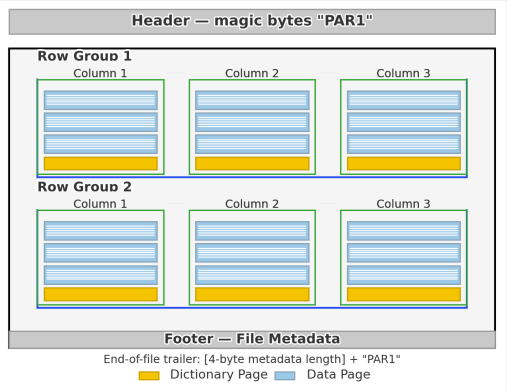
	Column 1 Product	Column 2 Customer	Column 3 Country	Column 4 Date	Column 5 Sales Amount
Row Group 1	Ball	John Doe	USA	2023-01-01	100
	T-Shirt	John Doe	USA	2023-01-02	200
Row Group 2	Socks	Maria Adams	UK	2023-01-01	300
	Socks	Antonio Grant	USA	2023-01-03	100
Row Group 3	T-Shirt	Maria Adams	UK	2023-01-02	500
	Socks	John Doe	USA	2023-01-05	200

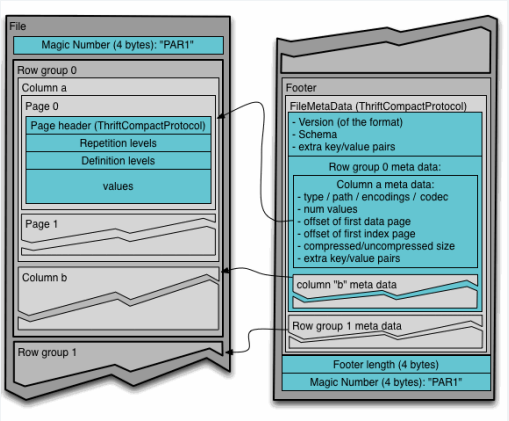
(Image from the article Data Mozart: Parquet file format)



(Image created by ChatGPT Pro)

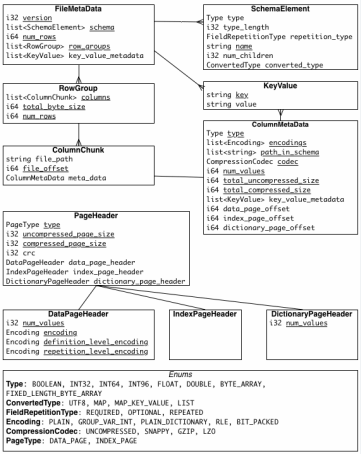
Parquet Data Format - Structure





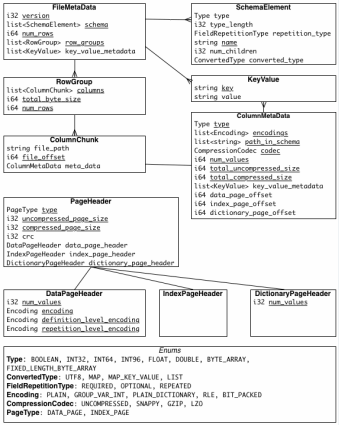
Metadata in Parquet File

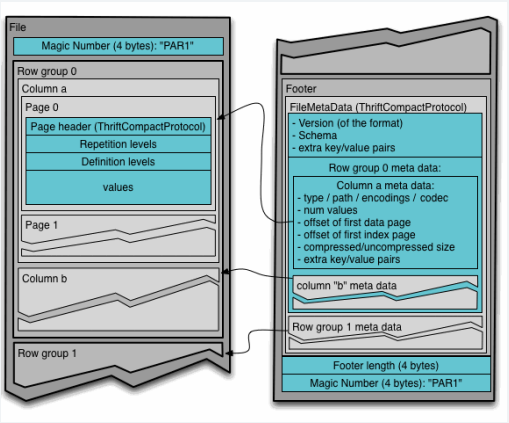
- Metadata stored at file, row group, and column levels
- Include column statistics: min, max, null, distinct counts
- Compression algorithms are recorded per column chunk
- Schema metadata - field names, data types
- Column statistics - skipping irrelevant row groups or pages
- Predicate pushdown and query optimization



(Image from the article
Understanding Parquet File Format)

Metadata in Parquet File





Metadata & Optimization of Queries

- Metadata allow multiple levels of optimization
- Clustering & partitioning can improve pruning efficiency
- Partitions group data files by column values
- Clustering sorts data within files by column values
- **Partition pruning** - use only specific group of files
- **File pruning** - use only specific files in partition
- **Row group pruning** - use only specific row groups
- **Column pruning** - read only necessary columns
- **Page filtering** - read only relevant pages in column chunk

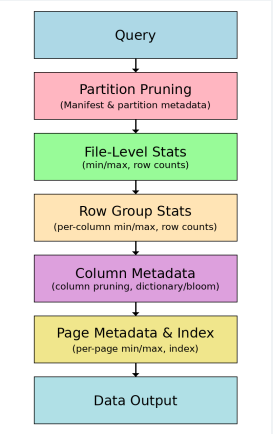


Image created by ChatGPT Pro

credativ

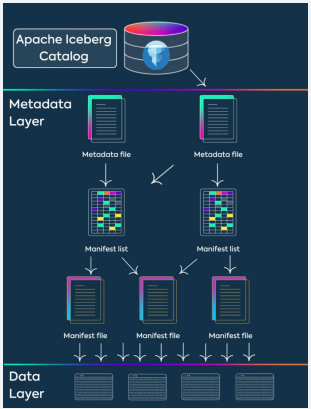
Apache Iceberg: Choreographing Data on a Parquet Stage

Apache Iceberg Table Format - Overview

- Created by Netflix - 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

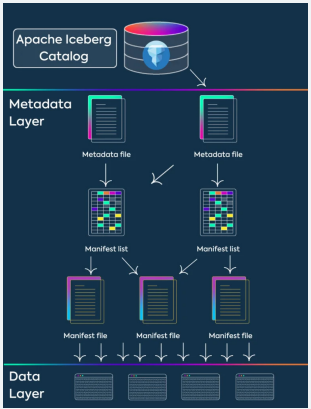
- Iceberg is a rich table format, not a database engine
- Core libraries in Java, separate projects for Python, C++, Rust, Go
- Supported by multiple engines: Spark, Trino, Flink, Presto, Hive, Dremio
- Allows multiple applications to work on the same data
- **Catalog layer** - table location, schema definitions
- **Metadata layer** - table state, snapshots, partitions
- **Data layer** - actual data files



(Image from the article What Is Apache Iceberg?)

Apache Iceberg - Catalog Layer

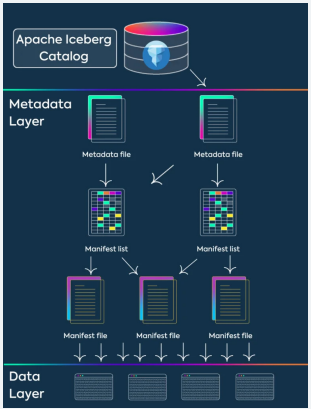
- Front door to any Iceberg dataset
- Source of truth for latest snapshot and schema
- Catalogs can be: Hive, AWS Glue, or simple JSON files
- References to tables, their locations, and metadata
- Hive Metastore is most common catalog on-premises
- RisingWave hosts catalog and metadata in PostgreSQL



(Image from the article
What Is Apache Iceberg?)

Apache Iceberg - Metadata Layer

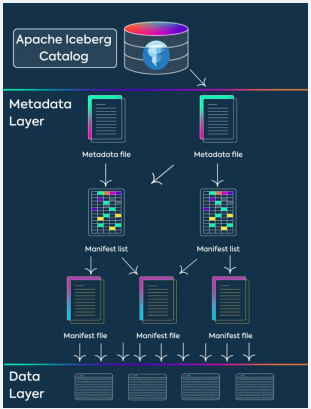
- Metadata files, manifests lists, manifests files
- **Metadata files**: hierarchical, immutable JSON files
- **Manifest list**: Avro file referencing manifests files
- **Manifest files**: Avro files listing data files and stats
- Every change creates a new metadata file and snapshot
- Snapshot provides full isolation and consistency
- Defined by exactly one metadata file
- Supports time travel and rollbacks
- Enables schema and partition evolution without rewriting
- Manifests enable efficient pruning of data files



(Image from the article
What Is Apache Iceberg?)

Apache Iceberg - Data Layer

- **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?)

credativ

PostgreSQL sailing with Icebergs

PostgreSQL-in-Lakehouse solution

credativ

- PostgreSQL is integrated into a larger platform
- Primary platform is an open storage layer
- Primary processing engine is Spark, Flink, Trino, etc.
- PostgreSQL can be used for collecting data
- Can serve as Gold-layer data store / metadata store
- Rich FDW support can make it federated query engine
- Pros: best-of-all-worlds, highly scalable
- Cons: complex multi-system, performance bottlenecks

Lakehouse-in-PostgreSQL solutions

- PostgreSQL functions as a single-site Lakehouse
- Ideal for On-Premises os Simpler Stacks
- Easier to manage, lower TCO (Total Cost of Ownership)
- Basic structure is ”PostgreSQL with Iceberg”
- Users leverage knowledge of PostgreSQL ecosystem
- Attractive for all-in-one OLTP & OLAP solutions
- (HTAP - Hybrid Transactional/Analytical Processing)
- Pros: simplicity, lower costs, familiar tools
- Cons: scalability bottlenecks, performance bottlenecks

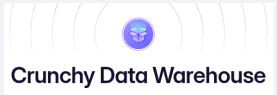
DuckDB – Powerful Analytical Database

- Created by Institute for Mathematics and Computer Science, Amsterdam
- The same institute is credited for Python and MonetDB
- Single-node database, for embedding in applications, like SQLite
- Open-source, column-oriented, in-memory relational database
- Designed for heavy parallel analytical workloads
- Columnar-vectorized query processing engine
- Uses PostgreSQL SQL parser and SQL dialect with improvements
- Direct selects from multiple formats and cloud storages
- See more in my talk PostgreSQL and DuckDB on YouTube

PostgreSQL and Data LakeHouse solutions

credativ

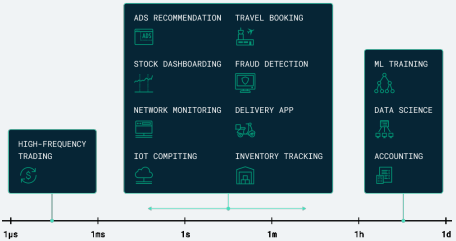
- **Cybertec LakeHouse** - for Scalefield platform
- All-in-one OLTP and OLAP workloads with AI integration
- **Crunchy Data Warehouse** - extends PostgreSQL with DuckDB
- Users can create/query Iceberg tables directly from PG
- **Databricks Lakebase** - uses Lakehouse IQ engine
- PostgreSQL-compatible OLTP database, supports Iceberg tables
- **Google's AlloyDB** - PG clone, columnar, vectorized engine
- Claims to be 4x faster for transactional workloads
- And up to 100x faster for analytical workloads
- *(BTW: Many databases now claim to be "PostgreSQL-compatible")*



RisingWave – Distributed SQL Engine

- Open-source, cloud-native distributed SQL engine
- Streaming database - processes real-time data streams
- Performs incremental computation for incoming data
- Updates results continuously, stores in its storage
- Fraud detection, stock market analysis, monitoring
- Built-in support for Apache Iceberg tables
- Fully compatible with PostgreSQL
- Uses PostgreSQL for catalog and metadata storage

RisingWave



PostgreSQL and Data LakeHouse solutions

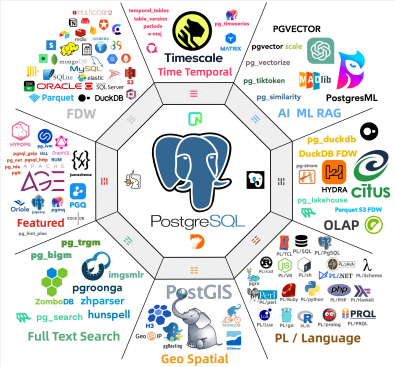
credativ

- **ParadeDB** - PostgreSQL with DuckDB and Iceberg
- **pg_duckdb** - DuckDB integration with PostgreSQL
- **Citus** - distributed columnar PostgreSQL
- **Pg_mooncake** - mirror of PG tables in Iceberg
- **pg_parquet** - read/write Parquet files with COPY
- **parquet_fdw / parquet_s3_fdw** - read Parquet files
- **postgres_fdw** - federated queries between PG servers

In 20 years...

Simon Riggs on PG conf EU 2023:

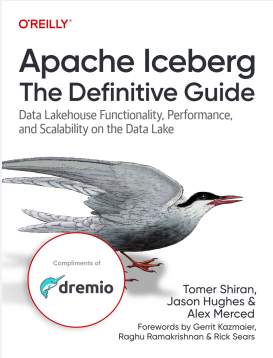
”In 20 years, everything will be done in PostgreSQL...”



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

Other Resources Used for the Talk

- Articles:
  - Parquet official documentation
  - Apache Iceberg main web page
  - Dremio.com: What is Apache Iceberg
  - Sqream.com: What is Apache Iceberg
  - Big Data File Formats
- E-books:
  - T.Shiran, J.Hughes, A.Merced: Apache Iceberg, The Definitive Guide - O'Reilly
  - A.Kaplan, A.Kara: Data Lakehouse for Dummies - Databricks
- ChatGPT Pro Deep Research
- Google Gemini Advanced 2.5 Pro Deep Research



credativ

Questions?