

Snowflake BDT

Jan Kozák

6 November 2024



What is Snowflake



THE SNOWFLAKE PLATFORM

“A single, fully managed platform that powers the AI Data Cloud. Snowflake securely connects businesses globally across any type or scale of data to productize AI, applications and more in the enterprise.”

<https://www.snowflake.com/en/data-cloud/platform/>



Brief History



- Founded in 2012 by former Oracle engineers – Benoit Dageville, Thierry Cruanes
- Relational DB in cloud – initially focused on DWH
 - Resource elasticity (data, compute)
 - Fault isolation
 - Performance isolation
- Currently
 - 10,000+ customers
 - 5 billion queries processed daily
 - 2,500 Marketplace listings

Core concepts

> Cloud agnostic

- AWS, Azure, GCP
- Interoperability
 - Between SNFL *accounts*
 - Between SNFL account and various cloud providers

> Cloud oriented

- Data ingestion – primary sources are data stored in cloud services (Azure Blob Storage, AWS S3, Google Cloud Storage)
- Cloud only, infrastructure provisioned by Snowflake
 - You cannot deploy Snowflake on your local device
 - Neither BYOC

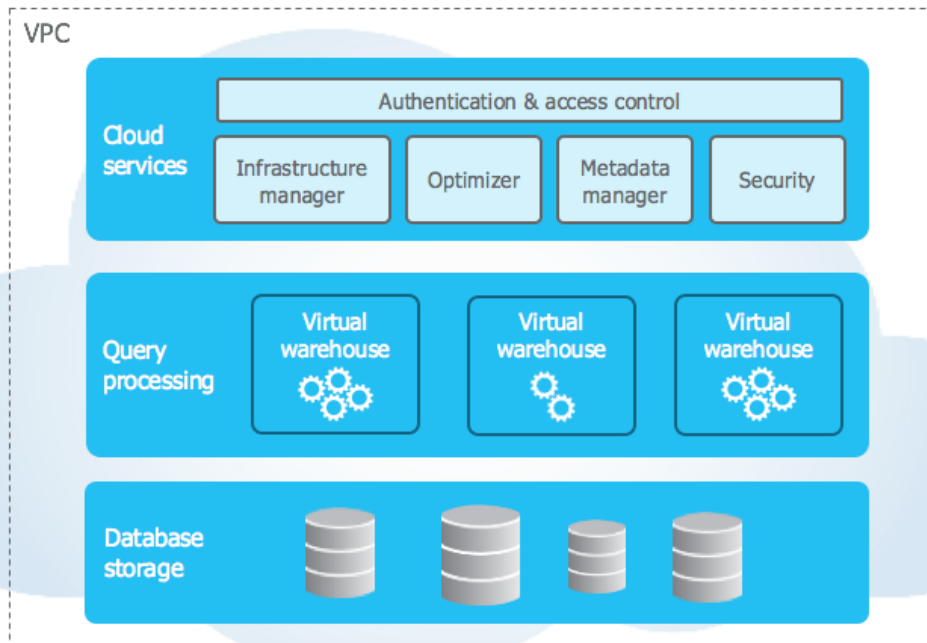
- Organization – company or similar
 - Accounts – linked with organization
- Account
 - Snowflake account ~ DB instance/machine
 - Cloud provider, region – cannot change later
 - Edition (Standard, Enterprise, Business Critical, *Virtual Private Snowflake*)
- Want to try it out?
 - <https://signup.snowflake.com/>
 - 30 days for free, 400 USD credit
 - Select at least Enterprise edition to unlock most features
 - *So far no restrictions on number of trial accounts – you may create more than one at time or register a new one when the trial period expires*



Snowflake - Architecture

- Self-managed Data Platform
 - Snowflake handles all infrastructure (HW, SW, maintenance)
- Data storage and compute resources separation
 - Compute Layer (Virtual Warehouse, Serverless)
 - Data Storage
 - Cloud Services
- Hybrid of *shared-disk* and *shared-nothing* architectures
 - Shared-disk – data stored at a central data repository accessible from all compute nodes
 - Shared-nothing – MPP – each node in the cluster stores a portion of data set locally during computation

Architecture



Ref: <https://docs.snowflake.com/en/user-guide/intro-key-concepts>

- Using data storage services of underlying cloud providers
 - VPS – storage account dedicated only for the account
 - Others – all accounts in one region share one centralized storage account
- Data Format (for internal tables)
 - Column-oriented
 - Proprietary and private
- Micro-partitioning
 - Automatically for all Snowflake internal tables
 - Compressed data file (disk segment), typically 50-500 MB (*pre-compression*)
 - Metadata about value ranges, selectivity etc. -> efficient filtering and pruning
 - Snowflake claims it is efficient for tables up to 5 TB large

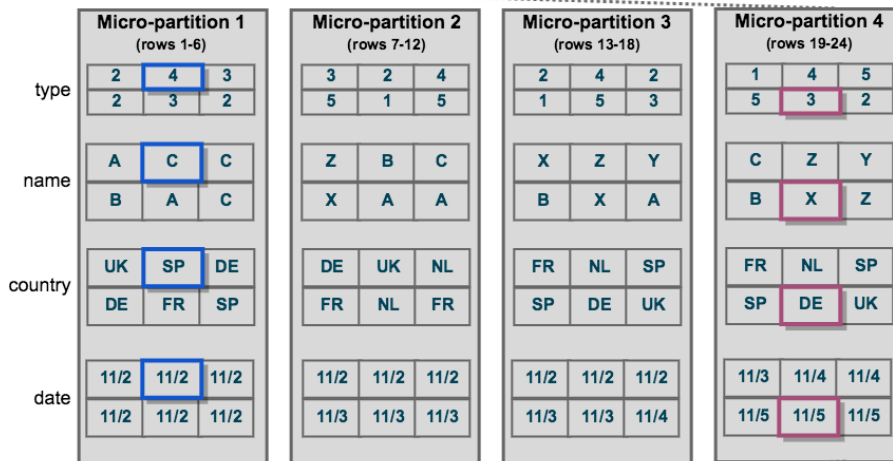
Micro-partitions

Table: t1

Logical Structure

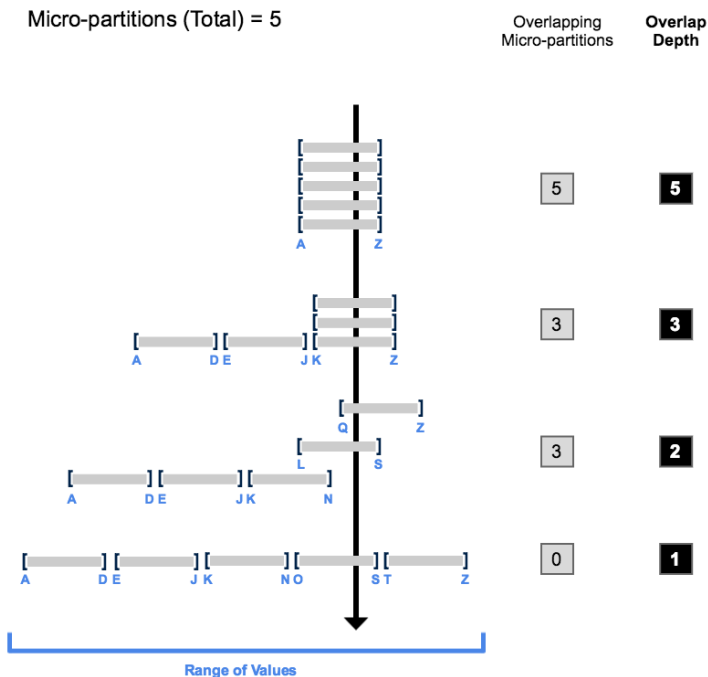
type	name	country	date
2	A	UK	11/2
4	C	SP	11/2
3	C	DE	11/2
2	B	DE	11/2
3	A	FR	11/2
2	C	SP	11/2
3	Z	DE	11/2
2	B	UK	11/2
4	C	NL	11/2
5	X	FR	11/3
1	A	NL	11/3
5	A	FR	11/3
2	X	FR	11/2
4	Z	NL	11/2
2	Y	SP	11/2
1	B	SP	11/3
5	X	DE	11/3
3	A	UK	11/4
1	C	FR	11/3
4	Z	NL	11/4
5	Y	SP	11/4
5	B	SP	11/5
3	X	DE	11/5
2	Z	UK	11/5

Physical Structure



Clustering depth

- Measure for efficient query pruning
- May be different for each column
- Lower depth ~ faster queries
- Affected by load patterns
 - Random vs bulk load



Ref: docs.snowflake.com/en/user-guide/tables-clustering-micropartitions

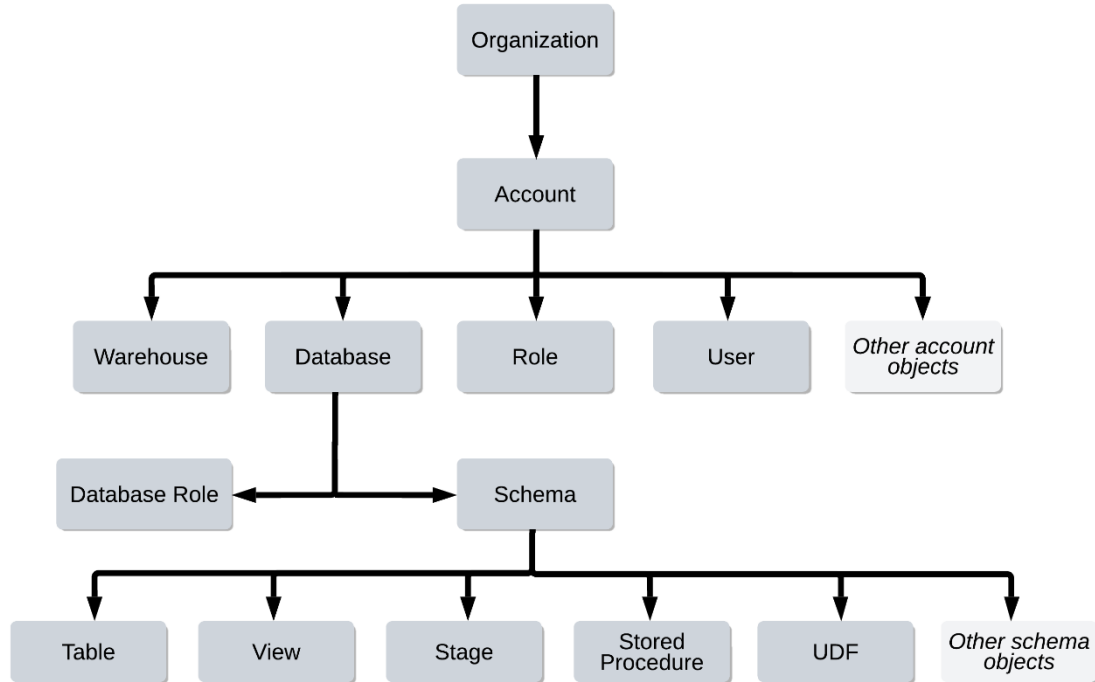
- Micro-partitioning is used in internal tables
- When micro-partitioning is not performing well, you may declare **clustering keys**
 - But no indexing or partitioning as in classical relational DBs
- What about unstructured and semi-structured data?
 - Semi-structured
 - JSON, XML, Avro, Parquet, ORC – directly supported by native functions
 - VARIANT column
 - Unstructured
 - BLOB/CLOB – very limited size in internal tables (16 MB)
 - Directory tables

Compute layer

- Virtual Warehouse (WH)
 - Main computing resource
 - Active WH is necessary for most operations
- May create multiple WH per account
 - Separate monitoring, access controls, sizing, tagging ...
- Sizing
 - Currently 10 sizes from X-Small (XS) to 6X-Large
 - + *Snowpark Optimized* counterparts for M-6XL sizes – ML/AI tasks
 - Exact technical specifications are not public
- Scaling – both up/down and in/out (even online)
 - Scaling-out – only in Enterprise and higher editions, 1-10 *clusters*

- > Structure fairly similar to common relational DBs
 - Database – schema – tables/views/functions...
- > Differences
 - *Role* (user/account-level) vs *Database Role*
 - RBAC – Snowflake cares about roles, not about users
 - Stage – landing point for data (un)loading
 - Internal – primarily for ad-hoc data uploaded from local machines
 - **External** – connection to cloud data storage accounts
 - AWS S3, Azure Blob Storage, Google Cloud Storage
 - Other special types
 - Integrations, Masking Policies, Data Quality Measures, ...

Objects Hierarchy



Ref: <https://docs.snowflake.com/en/user-guide/security-access-control-overview>

- Snowflake provides various table types
- Common constraints (primary/foreign keys, checks) exist, but generally they are only informative
 - Only NOT NULL constraints are enforced
- Internal tables
 - Main table type; variants – temporary, transient
- External tables – for data stored in stages
- Directory tables – for unstructured data in stages
 - Files accessed by URL, possible partitioning (actually encouraged)
- Hybrid tables – indexes, PKs/FKs enforced, Unistore concept
 - Real-time transactional tables (with random point reads/writes)

> All table types can be queried together

- We can join internal, external, directory and hybrid table together in one query
- We can declare views over any table type

> Views

- Secure Views
 - Suppressing some internal optimizations so the consumer cannot indirectly see or deduce data they aren't privileged to (e.g. through query plan or aggregation queries)
 - Only view owners can see its definition
- Materialized Views
 - For large or complex queries, results refreshed automatically
 - Only in Enterprise and higher editions; can be also Secure Materialized Views

- How do we get data to tables?
- Get data into internal or external stages
- Bulk Loading
 - Tasks (orchestration, schedules)
 - Streams – Change Data Capture (CDC), special table-like Snowflake object
- Continuous Loading
 - Snowpipe – monitoring stages when new file arrives and triggering pipeline
 - Snowpipe Streaming – connecting directly to external stream (Kafka)
- Tasks may be executed on Warehouse or in serverless mode
- Dynamic Tables – declarative ETL framework

- Snowflake provides rich set of predefined functions
 - Window functions
 - Analytic functions (time-series, MATCH)
 - Approximate functions (similarity, percentiles ...)
- User defined functions and procedures
 - Supports multiple languages
 - SQL, Python, Java, JavaScript
- External functions (executed outside Snowflake)
- Scripting – REST API, Python API
- **Snowpark** – for data processing, similar to Apache Spark (based on DataFrame paradigm)

- Zero-copy cloning
- Time-travel
 - Time-travel period configurable per table, may be set to 0-90 days
 - Fail-safe period – 7 days, for exceptional cases only (“best-effort”)
- Data replication
 - Same organization, to accounts in different regions
- **Data sharing**
 - One of the strongest Snowflake features
 - Share data with accounts from other organizations (in the same region)
 - If replicated, may share data even across different regions and cloud providers
 - Consumer – read-only, but no additional costs (data stored only at provider)

- Data Sharing
 - Multiple methods
 - Marketplace listings, Secure Data Sharing, Data Exchange, Clean Rooms
- Masking policies (row-level, column-level)
- Snowflake/Snowpark ML
 - Set of services for ML and MLOps, mimicking Mlflow
 - Model Registry, Feature Store, Datasets
- Applications
 - Streamlit, Snowflake Native App
- Snowpark Container Services
 - For Native Apps or Snowpark ML model serving, currently only in AWS

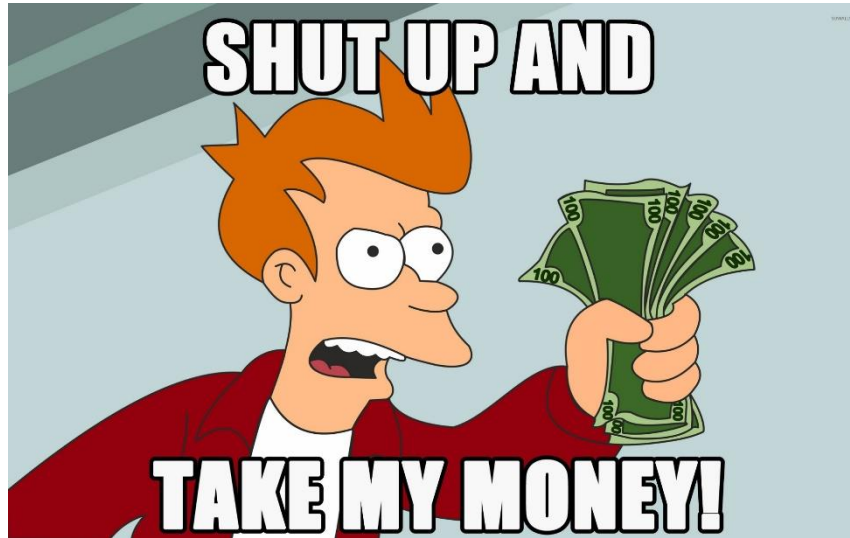
- Rapidly evolving in last years/months
- 2024 – introducing interactive notebooks
- LLM – **Snowflake Cortex**
 - Multiple models supported (mistral, llama, Snowflake Arctic ...)
 - RAG – native support for vector embeddings storage (VECTOR datatype)
 - Cortex Search – out-of-box RAG over stored documents
 - Cortex Analyst – answering nature language questions about data
 - Document AI – extraction and classification of documents

- So what does Snowflake charge you for?
 - Data storage (20-40 USD/TB per month)
 - Compute costs – in term of *credits*
 - 1 Snowflake credit ~ 2-10 USD (varies by provider, region and edition)
 - Virtual warehouses – 1 credit = 1 hour of XS warehouse uptime
 - Serverless – actual compute time
 - Cloud services – usually negligible
 - Data transfer costs
 - *Egress* (outbound traffic) from Snowflake account – both unloading to data storage (e.g. AWS S3) or Snowflake accounts in another regions
 - *Ingress* generally not charged (by Snowflake, cloud providers *will* charge some fee...)

Resources - Control

< PROFINIT >

- Almost unlimited resources
- If you pay for it



- How to control our bills?
- Resource Monitor
 - Formally just another DB object type
 - Controls credit consumption of *virtual warehouses*
 - Actions – Notify, Suspend, Suspend immediately
 - But it doesn't monitor serverless features or cloud services usage
- Internal metadata tables
- Cost attribution
 - Using ACCOUNT_USAGE tables and tags
 - Per organizational unit, user or query

> **Snowflake Horizon Catalog**

- Counterpart to Unity Catalog, Snowflake governance solution
- Governing data, apps and models across whole organization in single place
- Focused on Snowflake, but may integrate (on some level) data from external sources, mainly if they use Apache Iceberg format

> **Polaris Catalog**

- Open source (one of first Snowflake open-sourced projects)
- Using Apache Iceberg REST API for integrations
- Deployment: not necessarily on Snowflake
- Ambition: cross-engine universal catalog, industry standard
 - Integrations with Databricks, Fabric?

Snowflake Ecosystem

< PROFINIT >



Questions

Thank you

Profinit EU, s.r.o.
Tychonova 2, 160 00 Praha 6

Tel.: + 420 224 316 016, web: www.profinit.eu



LINKEDIN
linkedin.com/company/profinit



TWITTER
[@profinit_EU](https://twitter.com/profinit_EU)



FACEBOOK
facebook.com/Profinit.EU



YOUTUBE
Profinit EU, s.r.o.