**Snowflake Introduction:** Snowflake is a fully-managed, cloud-native data warehouse built on AWS, Azure, and GCP. It separates storage and compute, which allows independent scaling and cost optimization. It supports structured and semi-structured data, and offers advanced features like Time Travel, Fail-safe, Zero-Copy Cloning, and Secure Data Sharing. Database built in the cloud for the cloud and founded in 2012, as per current records snowfalke managing 350PB data under management. Snowflake act as saas (software as a service)

Snowflake is 1. Simple - Software as a service, plug and pay. 2. Scalable - data, compute ~ users/loads 3. Elasticity - scale up, down 4. Cost effective - pay for what you use 5. Data diversity - can store different types of data ~structured/semi-structured

**Snowflake Architecture**: Snowflake has a multi-cluster, shared data architecture built specifically for the cloud. It has 3 main layers:

1. **Database Storage Layer:** Acts as body of the system

* All structured and semi-structured data (JSON, Avro, Parquet, ORC, XML, etc.) is stored in compressed, columnar format inside Snowflake.
* Data is broken into micro-partitions and optimized automatically.
* Storage is centralized and shared across all compute clusters.
* Users don’t manage indexes or partitions — Snowflake handles it.

2. **Compute Layer:** Query procesing layer - Acts as Muscle of the system

* Compute happens through Virtual Warehouses (independent clusters).
* Warehouses can be scaled up (bigger size) or out (multi-cluster) depending on workload.
* Multiple warehouses can access the same data without contention → supports concurrency.
* Best part: we pay only when a warehouse is running (on-demand).

**3. Cloud Services Layer:** Meta data

* Acts as the brain of Snowflake.
* Handles authentication, metadata, query parsing, query optimization, security, and access control.
* Provides features like Time Travel, Fail-safe, Zero-Copy Cloning, Data Sharing.

This layer is always running (but doesn’t charge for compute).