



DELTA LAKE VS SNOWFLAKE TABLES – WHICH TO USE WHEN?

Both support **ACID transactions & analytics-ready tables**, but they're designed differently. Let's compare Delta Lake (Databricks) and Snowflake Tables side by side



Abhishek Agarwal
Data Engineer

WHAT IS DELTA LAKE?

- ✓ Open-source table format on top of **Parquet**
- ✓ Native to Databricks & Spark
- ✓ Key features:
 - **ACID transactions**
 - **Schema enforcement + evolution**
 - **Time Travel**
 - **Efficient upserts & deletes (MERGE INTO)**



WHAT ARE SNOWFLAKE TABLES?

- ✓ Fully managed tables inside **Snowflake's platform**
- ✓ Cloud-native, SQL-based
- ✓ Key features:
 - **Micro-partitions (automatic clustering)**
 - **Zero-maintenance performance tuning**
 - **Time Travel & Fail-safe**
 - **Fast cloning & data sharing**



CODE SNIPPETS – DELTA LAKE

- **Create Delta Table**

```
df.write.format("delta").save("/mnt/delta/sales")
```

- **Upsert Data (MERGE INTO)**

```
MERGE INTO sales AS s
USING updates AS u
ON s.id = u.id
WHEN MATCHED THEN UPDATE SET s.amount = u.amount
WHEN NOT MATCHED THEN INSERT (id, amount) VALUES (u.id,
u.amount)
```

- **Time Travel Query**

```
spark.read.format("delta").option("versionAsOf",
5).load("/mnt/delta/sales")
```



CODE SNIPPETS – SNOWFLAKE TABLES

- **Create Table**

```
CREATE TABLE sales (
    id INT,
    product STRING,
    amount FLOAT
);
```



- **Upsert Data (MERGE)**

```
MERGE INTO sales s
USING updates u
ON s.id = u.id
WHEN MATCHED THEN UPDATE SET amount = u.amount
WHEN NOT MATCHED THEN INSERT (id, amount) VALUES (u.id,
u.amount);
```

- **Time Travel Query**

```
SELECT * FROM sales AT (OFFSET => -60*5);
```



PERFORMANCE & STORAGE

Feature	Delta Lake	Snowflake Tables
Storage Format	Parquet + Delta log	Proprietary (columnar)
Partitioning	Manual	Auto (micro-partitions)
Clustering	Manual (Z-Order)	Automatic clustering
Caching	Databricks Delta Cache	Result caching



WHEN TO CHOOSE DELTA LAKE

- ✓ You're using **Databricks or Spark**
- ✓ Need **open format** (Parquet-compatible)
- ✓ Use cases:
 - **Streaming & batch pipelines**
 - **Fine-grained control over file structure**
 - **Hybrid cloud or on-prem deployments**



WHEN TO CHOOSE SNOWFLAKE TABLES

- ✓ You're fully in **Snowflake ecosystem**
- ✓ Want **zero infrastructure management**
- ✓ Use cases:
 - Fast BI dashboards
 - Multi-tenant SaaS apps
 - Need **built-in scaling & tuning**



REAL-WORLD SCENARIOS

- **Delta Lake:**

IoT company managing petabytes of semi-structured data with fast upserts, integrating ML models in Databricks.

- **Snowflake Tables:**

Fintech company running BI dashboards with minimal ops, leveraging auto-clustering & sharing data with partners.



FINAL TAKEAWAY

Choose Delta Lake if...	Choose Snowflake if...
You want open format flexibility	You want fully managed ease
You need deep Spark integration	You prioritize zero tuning
You work in hybrid environments	You're 100% Snowflake-native

Pro Tip: Some teams **combine both:** Transform data in Delta Lake, then load into Snowflake for BI.



**Follow for more
content like this**



Abhishek Agarwal
Data Engineer



Abhishek Agarwal | Data Engineer