

8 Best Streaming ETL Software Tools for Snowflake Pipelines

TechRounder PDF Edition

Live article: <https://www.techrounder.com/software/best-streaming-etl-software-tools-for-snowflake-pipelines/>

By Vipin PG | Published September 10, 2026 | Updated September 26, 2026 | Format: Article | 8 min read

In brief

Snowflake can process fresh data quickly, but the data reaching it may already be delayed. That delay usually begins upstream.

Snowflake can process fresh data quickly, but the data reaching it may already be delayed. That delay usually begins upstream.

Production databases continue receiving orders, account changes, transactions, inventory updates, application events, and customer activity between scheduled ETL runs. Even pipelines described as "near real time" can introduce meaningful delays when extraction, staging, merging, and transformation run on separate schedules.

Streaming ETL software reduces that delay by continuously moving new and changed data toward Snowflake rather than rebuilding datasets in periodic batches.

8 Streaming ETL Software Tools for Snowflake Pipelines

1. Artie

Artie is well suited to Snowflake teams that need to move changing operational database data into the warehouse continuously with low latency and limited infrastructure overhead.

Artie is a fully managed CDC streaming platform. Teams connect a source database, choose the tables they want replicated, and configure Snowflake as the destination. Artie performs the historical backfill before switching to continuous CDC, keeping subsequent inserts, updates, and deletes aligned with the operational source.

For PostgreSQL, Artie reads changes from the WAL through logical replication. It can maintain source-aligned tables in Snowflake using merge operations. Append-oriented history workloads can use Snowpipe Streaming. The product also includes mechanisms for delivery reliability and recovery.

Artie currently provides exactly-once delivery, automatic schema evolution, initial and online backfills, pipeline observability, SCD Type 1 replication, and a history mode that preserves row versions over time. History tables can record each create, update, and delete along with processing and source timestamps.

Common schema changes, including new destination tables and columns, can be handled automatically. Artie can optionally remove deleted source columns after a verification period. Column inclusion and exclusion controls also let teams keep unnecessary or sensitive fields out of Snowflake.

2. Estuary Flow

Estuary Flow combines CDC, streaming data movement, batch ingestion, and transformations in a streaming-first platform.

For Snowflake pipelines, Estuary can capture changes from sources such as PostgreSQL and materialize those collections into Snowflake. Its Snowflake integration supports standard and delta update patterns, with Snowpipe Streaming available for lower-latency ingestion.

Captured data is stored as a reusable collection that can later be materialized into one or more destinations. This gives Estuary a different architecture from a conventional point-to-point replication pipeline.

3. Striim

Striim has a long-standing focus on enterprise data streaming, CDC, and real-time integration. Its Snowflake offering can continuously move changes from databases such as Oracle, SQL Server, PostgreSQL, MySQL, and MariaDB into Snowflake after an initial synchronization.

Striim also supports in-flight processing. Data can be filtered, enriched, masked, denormalized, or otherwise transformed before reaching Snowflake. Striim describes this as real-time ETL, with SQL-oriented processing logic applied to streaming data before the warehouse layer when needed.

4. Matillion

Matillion combines data ingestion and transformation in a platform closely associated with cloud data warehouses.

For Snowflake, Matillion CDC can continuously capture changes from supported operational databases and load them into Snowflake. Its current documentation includes CDC sources such as Oracle, PostgreSQL, and SQL Server, along with target representations such as copied tables, soft-delete patterns, and change logs.

Matillion is useful when ingestion and warehouse transformation are both significant parts of the pipeline. Teams can bring CDC records into Snowflake and use Matillion's transformation capabilities to build analytics-ready models.

5. Qlik Replicate

Qlik Replicate focuses on log-based CDC and enterprise-scale data replication. It can capture changes from operational databases and continuously apply them to targets such as Snowflake. Qlik provides a Snowflake-optimized warehouse loading mode designed for the way massively parallel analytical systems ingest and merge data.

Source coverage is one of the platform's notable characteristics. Large enterprises may need to move data into Snowflake from databases, enterprise applications, mainframe-related systems, cloud platforms, and streaming infrastructure. Qlik's replication stack supports this wider range of source environments.

6. Hevo

Hevo provides a managed, no-code pipeline environment for database, SaaS, file, and other source types.

Its pipeline architecture continuously ingests source data, applies configured transformations and schema mappings, and loads records into destinations including Snowflake. The platform supports near-real-time transfer, automatic schema management, live monitoring, and built-in or Python-based transformations.

For database replication, Hevo can use source-specific CDC methods. Its MySQL integration, for example, uses binlog-based replication for low-latency data movement into Snowflake. SQL Server pipelines can use Change Tracking. PostgreSQL workflows can use incremental or CDC-based synchronization depending on the configuration.

7. Fivetran

Fivetran provides managed data movement across a broad connector ecosystem and supports Snowflake as both a destination and a source. For Snowflake destinations, Fivetran continuously synchronizes source schemas and incrementally loads changes into the warehouse. The platform uses Snowflake staging and SQL merge operations to maintain destination tables.

Fivetran also provides high-volume database connectors that use log-based CDC for systems such as Oracle and SQL Server. Reading source logs avoids repeated scans of transactional tables and can support high-volume replication with less impact on source systems.

8. Informatica

Informatica supports streaming Snowflake pipelines as part of a broader enterprise data management platform. Its cloud data ingestion and replication capabilities support CDC from transactional systems into cloud analytical platforms such as Snowflake. Informatica describes continuous CDC pipelines for sources including Oracle, SQL Server, SAP-related environments, Salesforce, and other enterprise systems.

The platform is particularly relevant when Snowflake is part of a larger enterprise modernization program. Informatica combines ingestion with data quality, governance, lineage, cataloging, transformation, and other data management capabilities. Its support for older enterprise environments also makes it relevant to organizations working across legacy and cloud systems.

Where Streaming Snowflake Pipelines Fail

Getting the first set of rows into Snowflake is usually the easy part.

Operational problems appear later. A source schema changes, traffic spikes, Snowflake slows down, a historical table needs rebuilding, or the team discovers that the source and destination have drifted apart.

These failure modes provide a more useful way to evaluate pipeline architecture than connector checklists alone.

Failure Mode | What Happens | What the Pipeline Needs

Schema drift | Source adds, alters, or removes fields | Schema detection and safe destination evolution

Destination slowdown | Snowflake cannot consume changes at the normal rate | Buffering and backpressure handling

Pipeline restart | Workers fail or deployment restarts | Durable checkpoints and deterministic recovery

Duplicate delivery | Events are replayed after failure | Idempotent or exactly-once processing

Large initial load | Historical data must be copied while production continues changing | Concurrent backfill and CDC

Missing deletes | Destination retains records removed upstream | Delete-aware CDC

Source traffic spike | Transaction volume rises rapidly | Scalable capture and buffering

Transformation lag | Raw tables are fresh but business models are stale | Continuous or sufficiently frequent transformation

Bad source deployment | A new column or type breaks downstream assumptions | Schema monitoring and controlled propagation

Full table rebuild | A table must be refreshed without interrupting downstream users | Online or staged backfill strategy

These operational characteristics can matter more than connector count. A pipeline may support PostgreSQL and Snowflake on paper yet still struggle with a high-volume production workload if it cannot safely manage backfills, replication slot pressure, destination outages, or schema evolution.

How Streaming Design Affects Snowflake Cost

Increasing pipeline frequency can also change Snowflake economics. Applying every individual row change to Snowflake immediately may create inefficient destination activity.

Production pipelines commonly buffer changes and write them efficiently while maintaining the required freshness SLA. The appropriate latency target depends on the workload and the cost of maintaining it.

Teams need to balance:

- Data freshness
- Destination compute
- Write frequency
- Merge efficiency
- Pipeline reliability
- Data volume

A fraud-monitoring workflow may justify aggressive freshness requirements. A finance model used three times per day may tolerate a longer interval. For many Snowflake workloads, sub-minute freshness can provide more practical value than pushing every pipeline toward millisecond latency.

Why Artie Fits Snowflake CDC Pipelines

All eight platforms can help keep Snowflake data current, but they solve different surrounding problems. An enterprise integration suite may suit a company that also needs mainframe modernization, broad application integration, data governance, complex transformations, and CDC within one vendor ecosystem.

A connector platform can make sense when the data stack depends heavily on hundreds of SaaS applications. Streaming processing systems are useful when substantial enrichment must happen before data lands in Snowflake.

Artie is more narrowly aligned with teams whose requirement is:

Keep production database data continuously synchronized with Snowflake without requiring the data team to operate a homegrown CDC stack.

That narrower focus is the main reason Artie takes the first position in this list.

FAQs

What is streaming ETL for Snowflake?

Streaming ETL continuously captures new or changed data, processes it, and delivers it into Snowflake as those changes occur. Database pipelines commonly use change data capture to read transaction logs and detect inserts, updates, and deletes. Transformations may happen during the stream or after the data reaches Snowflake, depending on the architecture.

Is CDC the same as streaming ETL?

No. CDC is a technique for detecting database changes. Streaming ETL covers the wider pipeline that captures, transports, optionally transforms, and loads continuously changing data. CDC is often the extraction mechanism within a streaming ETL or ELT architecture, particularly when PostgreSQL, MySQL, Oracle, or SQL Server is the source.

Should transformations happen before or after data reaches Snowflake?

It depends on the workload. Security filtering, masking, and certain operational transformations may need to happen before loading. Analytics-oriented transformations are often easier to manage after ingestion using Snowflake SQL, dbt, Dynamic Tables, or similar tools. Separating replication from analytical transformation can also simplify pipeline ownership and debugging.

How fresh does Snowflake data need to be?

The appropriate freshness target depends on the business decision using the data. Fraud detection, operational dashboards, AI agents, personalization, and customer-facing analytics may need updates within seconds or minutes. Traditional executive reporting may tolerate longer intervals. Teams should define freshness as an end-to-end SLA based on the workload and its actual latency requirements.

Why use log-based CDC instead of frequent database queries?

Log-based CDC reads changes already recorded in the database's transaction log. This reduces the need for repeated table scans, can lower load on production databases, and makes continuous capture of inserts, updates, and deletes easier. Transaction logs also provide an ordered record of changes for downstream replication systems.

What happens when a Snowflake streaming pipeline falls behind?

A well-designed pipeline buffers changes and continues tracking source progress while the destination recovers. Once Snowflake can accept changes normally again, the pipeline can process the backlog. Teams should evaluate checkpointing, backpressure handling, recovery behavior, source-log retention, observability, and delivery semantics before using a pipeline for critical real-time workloads.

How does schema evolution affect streaming pipelines?

Schema evolution occurs when developers change the structure of a source table by adding or removing columns, changing field types, or making similar modifications. A streaming pipeline needs to detect those changes and determine how to represent them safely in Snowflake. Weak schema-evolution handling can cause ordinary application deployments to break replication or leave downstream data incomplete.

References

1. artie.com - <https://www.artie.com/>