



UNISTORE UNITES TRANSACTIONAL AND ANALYTICAL DATA

A SINGLE PLATFORM FOR ALL YOUR DATA
TO TRANSFORM YOUR BUSINESS

TABLE OF CONTENTS

- 3 Executive Summary**
- 4 The architectural logic behind OLTP and OLAP**
- 5 Introducing Unistore**
- 6 Unlocking new use cases with Unistore**
 - 6 Application state**
 - 6 Data serving**
 - 6 Transactional applications**
- 6 Unistore: Powered by Hybrid Tables**
- 7 Practicum: Using Hybrid Tables**
 - 7 Primary keys**
 - 7 Foreign keys**
 - 8 Secondary indexes**
 - 8 Join tables**
- 8 Ready to knock down your transactional data silo?**
- 9 About Snowflake**

Executive Summary

For far too long, organizations have used two separate systems to handle transactional and analytical workloads. While transactional workloads demand millisecond speeds, analytical workloads focus on complex queries on large volumes of historical data. The infrastructure requirements to deliver on this have always resulted in the adoption of two different types of databases.

As a result, most teams move transactional data into an analytical database in order to perform the analysis. This process came about with good reason and has been the status quo for many years. However, it's hardly surprising that organizations want to consolidate these two solutions in order to simplify architectures, eliminate data movement, and get real-time value from transactional data.

And now they can. Unistore is the new Snowflake workload that delivers transactional and analytical data together in a single platform. This provides one place to get more value from your data while also benefiting from the many features of Snowflake, including unified data governance and security.

Read on to learn more about:

- [The architectural challenges of working with transactional and analytical data](#)
- [How Unistore delivers the simplicity and unified architecture you need](#)
- [What use cases Unistore supports today and what's next for this new Snowflake workload](#)

THE ARCHITECTURES BEHIND OLTP AND OLAP

For years, transactional and analytical workloads have existed in separate systems that are designed to manage the primary needs of each type of data.

OLTP, or Online Transaction Processing, is ideal as a database for transactional data because it's designed to capture and record time-sensitive events. OLTP systems excel at millisecond response times and simple queries such as single-record selects, inserts, updates, and deletes.

In comparison, **OLAP**, or Online Analytical Processing, is an analytical database that manages complex queries and large-scale scans. Users can trade longer response times since the system is querying across larger volumes of historical data to discover and interpret findings.

While these two databases have served their purpose for a long time, some common challenges arise when working across separate systems:

1 Too many disparate systems lead to inconsistencies. Whenever data silos exist, this heavy burden requires managing these silos, making it especially difficult to maintain consistent data security and governance.

2 Moving data is slow and painful. Teams must build and manage pipelines to move data between systems in order to unite data sources. This requirement introduces massive complexity and adds latency as to when users can extract insights from data.

3 Delayed access stunts development. Teams need data for downstream use cases, but latency impacts what can be done with that data and introduces complexity into incorporating real-time insights into analytics, applications, data science, etc. Simply put, not having data together hurts innovation.

OLTP vs OLAP



TRANSACTIONAL DATABASE

Online Transaction Processing (OLTP)

- Fast, simple queries like single-record, selects, inserts, updates, deletes
- Millisecond response times



ANALYTICAL DATABASE

Online Analytical Processing (OLAP)

- Complex queries and large-scale scans
- Longer processing depending on data volumes

INTRODUCING UNISTORE

Snowflake is built on the belief that any data silo is one too many. That's why Snowflake developed Unistore as its latest workload to unite data on a single platform.

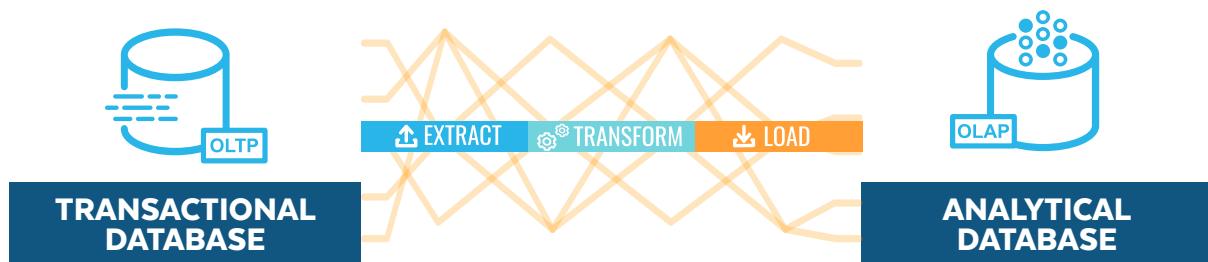
Unistore is a modern approach to working with transactional data and analytical data that brings both data types together in a single platform—Snowflake. Three major benefits arise from Unistore:

- 1 One single dataset:** You can work with your transactional data and analytical data in a single logical dataset, thus unlocking new insights and business opportunities.
- 2 Transactional capabilities:** Because Snowflake now supports your transactional workloads, you can build enterprise or internal transactional applications directly on Snowflake.
- 3 All on Snowflake:** With all of your data in Snowflake, you can dramatically simplify your architecture, unify your data governance and security policies, and accomplish things you couldn't do before, or even imagine.

What do these benefits look like on a day-to-day basis for your organization? Here are some examples of how Unistore empowers your teams and propels your business forward.

- **Data movement, eliminated.** It's no longer necessary to move transactional data from an external source into Snowflake. Unistore enables your teams to consolidate all data in Snowflake without building or maintaining complex pipelines.
- **Architecture, unified.** Unistore supports both operational and analytical queries, wrapped up in a unified data experience. No more managing multiple systems or time spent trying to maintain consistency across your data.
- **Insights, activated.** Unistore can deliver immediate analytics on transactional data, thus eliminating constraints and delays. With a single data set, application developers, data scientists, analysts, and others can uncover insights in real time using all of your data.
- **Application development, disrupted.** Unistore enables fast, single-row operations, which can power streamlined transactional application development directly on Snowflake.

THE CHALLENGES OF TWO SYSTEMS



UNLOCKING NEW USE CASES WITH UNISTORE

With transactional data capabilities and a single dataset approach, Unistore unlocks a new set of use cases directly in Snowflake.

Application state

Unistore can handle data related to the current state of an application. ETL control flow is one example for storing information as to whether ETL jobs have started or failed. This can help determine crucial next steps for troubleshooting or provide inputs into alerting systems. It can also be used to back client applications—storing the state of customer sessions such as their activity.

Data serving

Unistore can serve processed, read-only data to other systems and applications with low latency. An example could be an airline web portal with all the ticket and flight information of those planning to fly the next day and pre-computed so it's quickly accessible for customers as well as airline employees working ticket counters and departure gates.

Transactional applications

Unistore also expands the types of transactional applications Snowflake can support. It is especially suited for enterprise or internal applications, requiring low-latency responses and supporting large amounts of concurrent end users. For example, a logistics company can now track shipments and monitor supply-chain incidents using all transactional, analytical, and historical data together and supply that information directly to its employees via applications.

UNISTORE: POWERED BY HYBRID TABLES

With Unistore, Snowflake delivers one system, one product, and one experience for both analytical and transactional data needs.

Hybrid Tables (in private preview) is the new feature that powers Unistore by enabling fast, single-row reads and writes in order to support transactional workloads. Hybrid Tables are just another table type in Snowflake, making them familiar to use while maintaining the same simplicity to ensure everything just works. A key design principle was the need to support the most common transactional capabilities that application developers have come to rely on. Performance is especially critical for any transactional application, especially for fast single-row operations. To help support that, Snowflake developed entirely new row-based storage. What's especially powerful is that this storage layer leverages the exact same client, cloud services architecture, authorization and authentication, metadata, query processing layer, and more, as does the existing analytic store.

In addition, it's not just about providing the transactional capabilities, but also delivering the query performance required. In order to provide excellent analytical query performance, Snowflake can continuously replicate data into the analytical store and as well in the background. Hybrid Tables provide a single logical table, obfuscating this so you never need to worry about the type of query you're running or what data store is the right fit. This is the magic of Hybrid Tables. Bringing this together as one single product and experience is a truly modern approach.

PRACTICUM: USING HYBRID TABLES

With Unistore, Snowflake delivers one system, one product, and one experience for both analytical and transactional data needs.

- **Primary keys** to uniquely identify each row in a Hybrid Table, with indexes to support concise lookups. Every row must have a primary key value, and each table can have one primary key only.
- **Foreign keys** for linking and cross-referencing data between tables.
- **Referential integrity constraints** to validate data before it's loaded into tables, which enforce the relationship between two tables and prevent actions that might destroy these links.
- **Secondary indexes** to speed up searches and queries, as well as retrieve data from the database along dimensions that are not the primary key.
- **Join tables** to enable Hybrid Tables to be combined with other Snowflake tables for a holistic view of transactional and analytical data, together. Data can also be loaded between tables with a single SQL statement, and it's possible to atomically transact across different table types.

Primary keys

Primary keys are unique identifiers within a table to enable fast analysis. **With Hybrid Tables, primary keys are always enforced.** This behavior is different from other Snowflake tables where primary keys are provided but not enforced.

Create a primary key

- The constraint to build a Hybrid Table is that you must provide a primary key.
- The primary key must have a value. It cannot be null.
- To input the primary key, you select a key column.
- You can create a composite primary key composed of multiple columns.
- A primary key can also be defined using autoincrement. When you start with a default value, the table will assign each new record an incremental value.

Check your Hybrid Table

- The “describe table” command indicates with a “Y” the primary key column you selected.
- If you build a Hybrid Table, the “show table” command displays a “Y” in the hybrid column.

Insert records

- Every record you insert must have a unique primary key value.
- If you attempt to insert a record with a primary key value that already exists in the table, the query will fail. The record will not be inserted.
- If you attempt to insert two records with the same primary key value in one transaction, neither record will be inserted.

Foreign keys

Foreign keys provide a link between data in tables by referencing primary keys.

Create a foreign key

- When you build a new hybrid table, add a constraint for a foreign key and reference the primary key from another table.
- Foreign keys can be used to connect hybrid and regular tables.

Referential integrity is enforced

- Data is validated before it's loaded to enforce the foreign key relationship and prevent actions that might destroy this link.
- New records can only be inserted if they have the appropriate foreign key reference.
- If you attempt to insert a value that has a foreign key constraint, and that value doesn't exist as a primary key in the linked table, then the record will not be inserted.

Secondary indexes

Secondary indexes are used as a faster means to query data by an attribute other than the primary key. In Snowflake, creating a secondary index is easy. When you build a Hybrid Table, simply create an index on a non-primary key column.

The value of secondary indexes is demonstrated when you compare a query run on a non-primary key column without an index against a query run on one with an index.

- **No index:** A query on a Hybrid Table without an index results in a full table scan with a filter applied. The time to run the query is dependent on the size of the table.
- **Secondary index:** If you run the exact same query on the exact same data set with a secondary index, the scan comes back in milliseconds because there is no table scan. The index is used to filter the data, which means the results are delivered immediately.

Join tables

Hybrid Tables can be joined with any other table in Snowflake. A simple command enables you to join two or more tables. Curious how it works? Here's how you can see what's happening behind the scenes:

- When the join is created, look at the Hybrid Table profile. You will see that there's no reference to micro partitions because it's querying from the new row storage engine. Snowflake is doing a full table scan without filters.
- When you look at the regular table profile, you will see micro partitions. In this table, Snowflake's storage and query engine are joining the data together and serving up the results set.

The end result is a joined table, regardless of which tables you start with or what process happens behind the scenes.

READY TO KNOCK DOWN YOUR TRANSACTIONAL DATA SILOS?

Everything we do at Snowflake is focused on helping our customers mobilize their data, so they can take their organizations to new frontiers. We continue to evolve what's possible with data and the technology to advance and transform your business, so you can address long-standing challenges, deliver amazing customer experiences, and reveal new business opportunities.

With Unistore, Snowflake removes some of the biggest obstacles facing organizations today. Snowflake brings transactional and analytical data together in the Data Cloud to ensure a single solution and experience. Snowflake has revolutionized the way to access, integrate, analyze, collaborate, and build applications with data.

With unified data, anything is possible. Isn't it time to explore the possibilities? To learn more about Unistore, view this [webinar](#). To speak someone from Snowflake about how Unistore can help transform your business, click [here](#).

ABOUT SNOWFLAKE

Snowflake enables every organization to mobilize their data with Snowflake's Data Cloud. Customers use the Data Cloud to unite siloed data, discover and securely share data, and execute diverse analytic workloads. Wherever data or users live, Snowflake delivers a single data experience that spans multiple clouds and geographies. Thousands of customers across many industries, including 510 of the 2022 Forbes Global 2000 (G2K) as of July 31, 2022, use Snowflake Data Cloud to power their businesses.

Learn more at snowflake.com



© 2022 Snowflake Inc. All rights reserved. Snowflake, the Snowflake logo, and all other Snowflake product, feature and service names mentioned herein are registered trademarks or trademarks of Snowflake Inc. in the United States and other countries. All other brand names or logos mentioned or used herein are for identification purposes only and may be the trademarks of their respective holder(s). Snowflake may not be associated with, or be sponsored or endorsed by, any such holder(s).