LUMENDATA

Data Sheet

An Overview of Hybrid Tables in Snowflake

The One-Stop Guide

5201 GREAT AMERICAN PARKWAY, SUITE 320
SANTA CLARA, CA 95054
Tel: (855) 695-8636
E-mail: info@lumendata.com
Website: www.lumendata.com

Overview:

Hybrid Tables are a type of Snowflake table optimized for hybrid transactional and operational workloads that require low latency and high throughput. The data sheet will cover some of the critical concepts about Hybrid Tables like:

- Features of Hybrid Tables
- How to create a Hybrid Table
- When to use a Hybrid Table
- Cost and Limitations of Hybrid Table

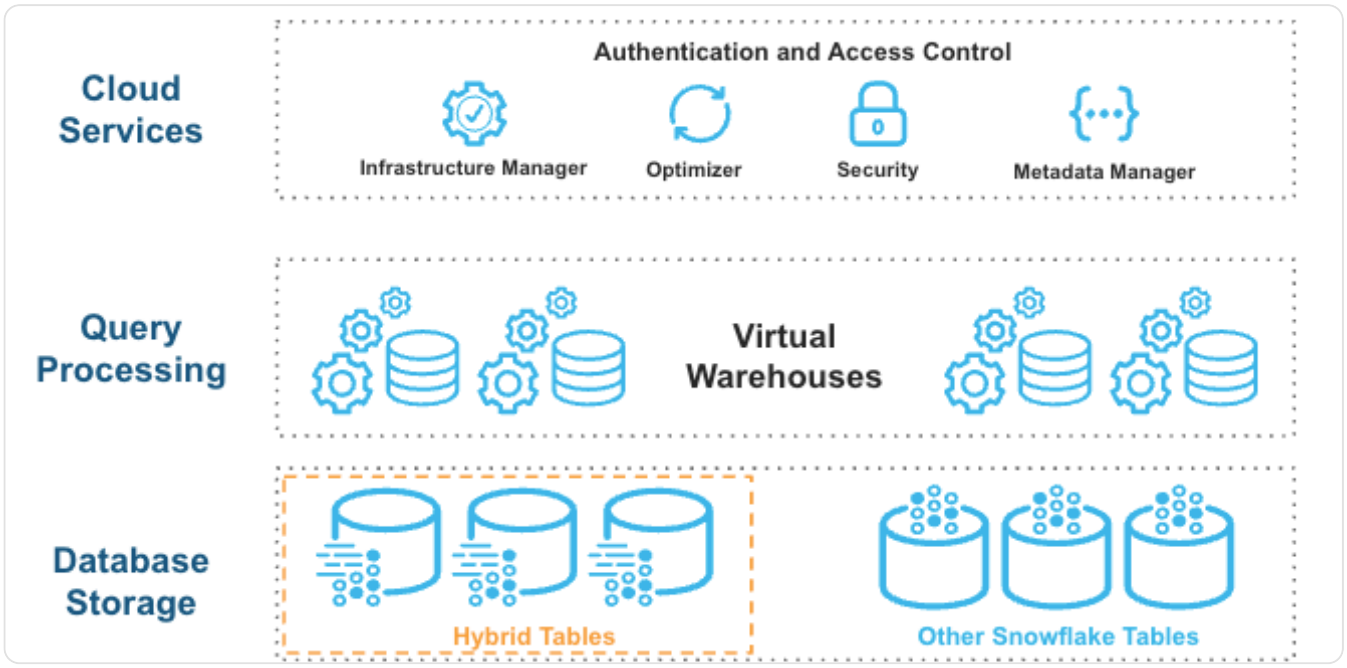

Image Source: Snowflake

Features of Hybrid Tables

Hybrid tables provide additionally the below ones compared to other table types:

- **Rowstore:** Data is written directly into rowstore, which will copy the data into storage asynchronously without impacting the workloads and large scans.

- **Data Governance:** Snowflake provides data governance for the transactional data with Masking Policies like Dynamic Data Masking, External Tokenization, Tag-based Masking, and Row Access Policies.

- **Primary and Foreign Keys:** It requires a primary key with enforced uniqueness and an optional foreign key with enforced referential integrity.
- **Indexes:** Indexes are updated synchronously for performance.
- **Constraints:** Supports the enforcement of unique and referential integrity constraints.

Creating a Hybrid Table

The below syntax helps you to create hybrid tables:

```
CREATE [ OR REPLACE ] HYBRID TABLE [ IF NOT EXISTS ] <table_name>
 ( <col_name> <col_type>
   [
    {
     DEFAULT <expr>
      /* AUTOINCREMENT (or IDENTITY) is supported only for numeric data types
(NUMBER, INT, FLOAT, etc.) */
     | { AUTOINCREMENT | IDENTITY }
       [
        {
         ( <start_num> , <step_num> )
         | START <num> INCREMENT <num>
        }
       ]
       [ { ORDER | NOORDER } ]
    }
   ]
   [ NOT NULL ]
   [ inlineConstraint ]
   [ , <col_name> <col_type> [ ... ] ]
   [ , outoflineIndex ]
   [ , ... ]
 )
 [ COMMENT = '<string_literal>' ]
```

NOTE:

- **The creation of a hybrid table requires a primary key constraint.**
- **Indexes can be defined only on columns that are not semi-structured.**
- **Foreign key constraints are enforced on hybrid tables. Unique and foreign key constraints each build their underlying index.**

- **Creating a Hybrid Table with an Index**

```
CREATE HYBRID TABLE student (
  student_id INT PRIMARY KEY,
  student_name STRING,
  student_department STRING,
  INDEX idx_department (student_department) INCLUDE (student_name)
);
```

- **Inserting sample data into the table.**

```
INSERT INTO student VALUES
(1, 'Sai', 'Computers'),
(2, 'Vijay', 'Electronics'),
(3, 'Hushal', 'Finance'),
(4, 'Nandini', 'Mechanical');
```

- **Querying the data with help of "student_department" index.**

SELECT student_name FROM student WHERE student_department = 'Electronics';
SELECT student_name FROM student WHERE student_department in ('Electronics', 'Computers');

When to use a Hybrid Table

- Concurrently reading data from a range of datasheets.
- Concurrently writing the data to datasheets (Bulk Loading).
- Retrieving some group of data from the entire dataset via some aggregations or GROUP BY.

Limitations of Hybrid Tables

Hybrid tables will not support following features:

- Cloning
- Clustering Keys
- Collations
- Data Retention Period
- Data Sharing
- Dynamic Tables
- Fail-safe
- Materialized Views
- Periodic rekeying
- Query Acceleration Service
- Replication
- Search Optimization Service
- Snowpipe
- Streams
- Tri-secret secure encryption
- Time Travel
- UNDROP

Understanding Cost of Hybrid Tables

- **Database Storage:** Storage cost is based on a flat monthly rate per gigabyte (GB), which is a bit more expensive than normal table storage.
- **Virtual Warehouse:** When we query the hybrid tables, the consumption rate is the same as the other table types.
- **Hybrid Table Requests:** It consumes an additional cost as it uses serverless resources for even small read or write operations and incurs a minimum of 4 KB of storage for a hybrid table request.

Authors

Sai Bharadwaja

Senior Consultant

About LumenData

LumenData is a leading provider of **Enterprise Data Management, Cloud & Analytics** solutions. We help businesses navigate their data visualization and analytics anxieties and enable them to accelerate their innovation journeys.

Founded in 2008, with locations in multiple countries, LumenData is privileged to serve over 100 leading companies. LumenData is **SOC2 certified** and has instituted extensive controls to protect client data, including adherence to GDPR and CCPA regulations.