

SQL KONFERENZ 2026

SQL
KONFERENZ
2026



Unlocking Synergy

Exploring Snowflake and Fabric Integration with Iceberg Tables & Mirroring

Matthias Nohl

SQL
KONFERENZ
2026



Your Speaker

- **Strategic Partner Manager DACH & Community Engagement Lead at b.telligent,** a technology independent consulting company which has specialized in analytics and data management.
- **15+ Year** experience in Cloud Platform & Data Analytics.
- **Specialist** for Cloud Data Platform on Azure.





Matthias Nohl
#Passion for #technology & #data

linkedin.com/in/matthiasnohl

Agenda

- Introduction
- Integration with Snowflake Mirroring
- Accessing Data from Snowflake in Fabric
- Accessing Data from Fabric in Snowflake
- What should be used and how?
- Q&A



SQL KONFERENZ 2026



Introduction

Why Now: The Data & AI Imperative



- Fragmented stacks drive redundant pipelines, egress cost, and inconsistency.
- GenAI and BI need fresh, governed, organization-wide access.
- Openness & interoperability are operating principles, not add-ons.

Modern Data Formats – Iceberg & Delta Lake



Why do we need new data formats?

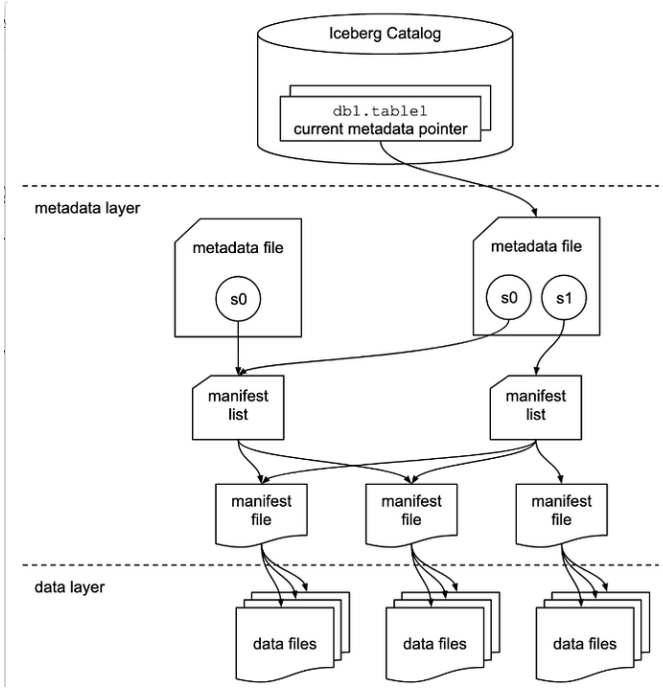
- Traditional data lakes (e.g., Parquet, CSV) struggle with scalability, consistency, and governance.
- Modern formats like **Apache Iceberg** and **Delta Lake** address these challenges with ACID transactions, versioning, and schema evolution.

Feature	Iceberg	Delta
ACID Transactions	Yes	Yes
Schema Evolution	Yes	Yes
Time Travel	Yes	Yes
Hidden Partitioning	Yes	No
Multi-Engine Support	Spark, Trino, Flink, Snowflake, Fabric	Spark, Fabric, Databricks, Snowflake (via virtualization)
Open Source	Yes	Yes

Deep Dive – Apache Iceberg

Key Capabilities

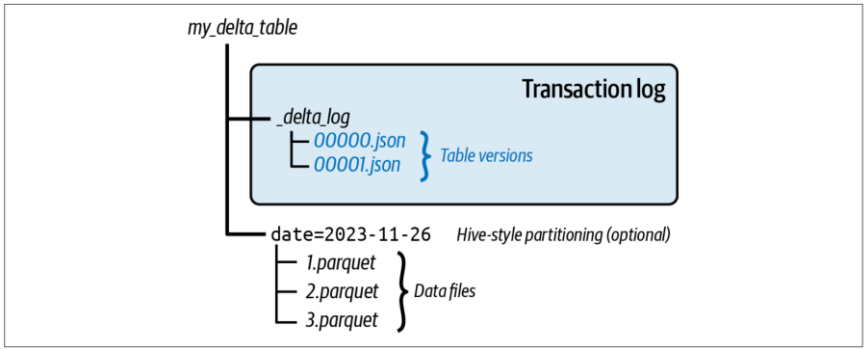
- Table format for large, distributed datasets with a focus on performance and scalability.
- **ACID transactions:** Reliable data consistency for reads and writes.
- **Schema evolution & time travel:** Access historical versions and flexibly adapt table structures.
- **Hidden partitioning:** Optimizes queries without manual partition management.
- **Snapshots:** Every change creates a new snapshot, ideal for audits and rollbacks.



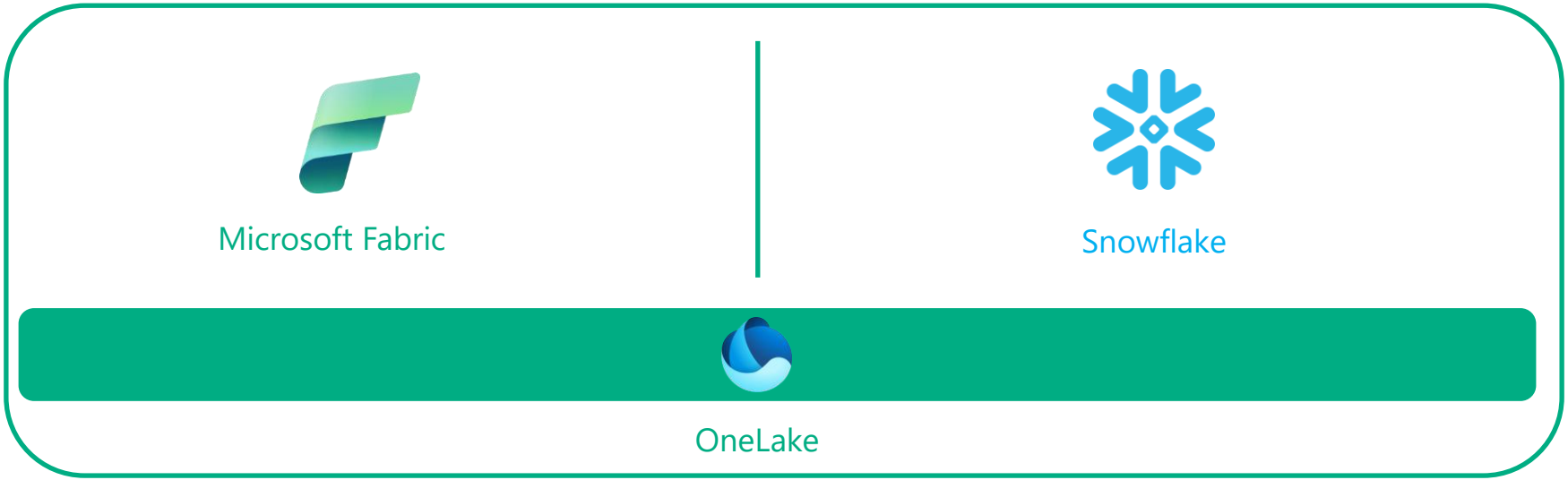
Deep Dive – Delta

Key Capabilities

- **ACID transactions:** Ensures data integrity in the data lake.
- **Schema evolution:** Flexible table definition changes.
- **Time travel:** Query previous versions of the data.
- **Optimized performance:** Efficient upserts and merges.
- **Integration:** Native support in Databricks, Fabric; Snowflake can consume Delta tables via virtualization as Iceberg.



The Partnership at a Glance

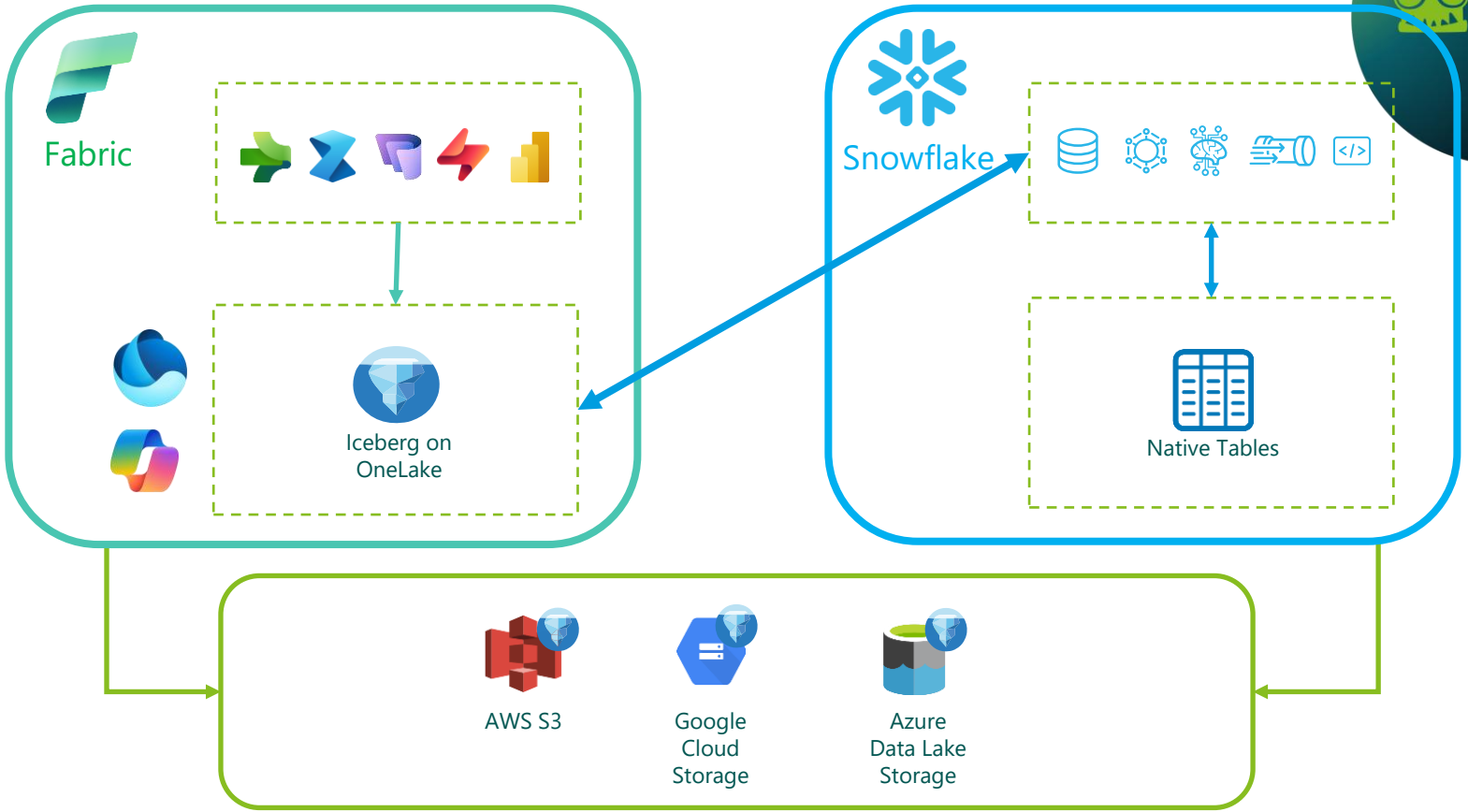


Iceberg support in OneLake

Bi-directional data access

Seamless access from M365 and Copilot

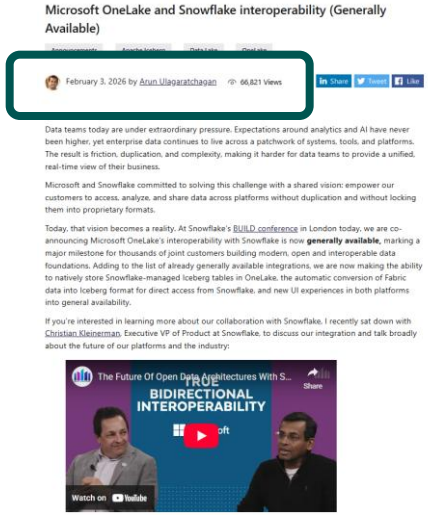
Integration Microsoft Fabric & Snowflake



Microsoft OneLake and Snowflake interoperability

So, what does this GA mean for you?

- The ability to bidirectionally read Iceberg data managed by Snowflake or Microsoft Fabric.
- The ability to natively store Snowflake-managed iceberg tables in Microsoft OneLake.



Microsoft OneLake and Snowflake interoperability (Generally Available)

February 3, 2026 by Arun Ulagaratchagan · 66,821 Views

Data teams today are under extraordinary pressure. Expectations around analytics and AI have never been higher, yet enterprise data continues to live across a patchwork of systems, tools, and platforms. The result is friction, duplication, and complexity, making it harder for data teams to provide a unified, real-time view of their business.

Microsoft and Snowflake committed to solving this challenge with a shared vision: empower our customers to access, analyze, and share data across platforms without duplication and without locking them into proprietary formats.

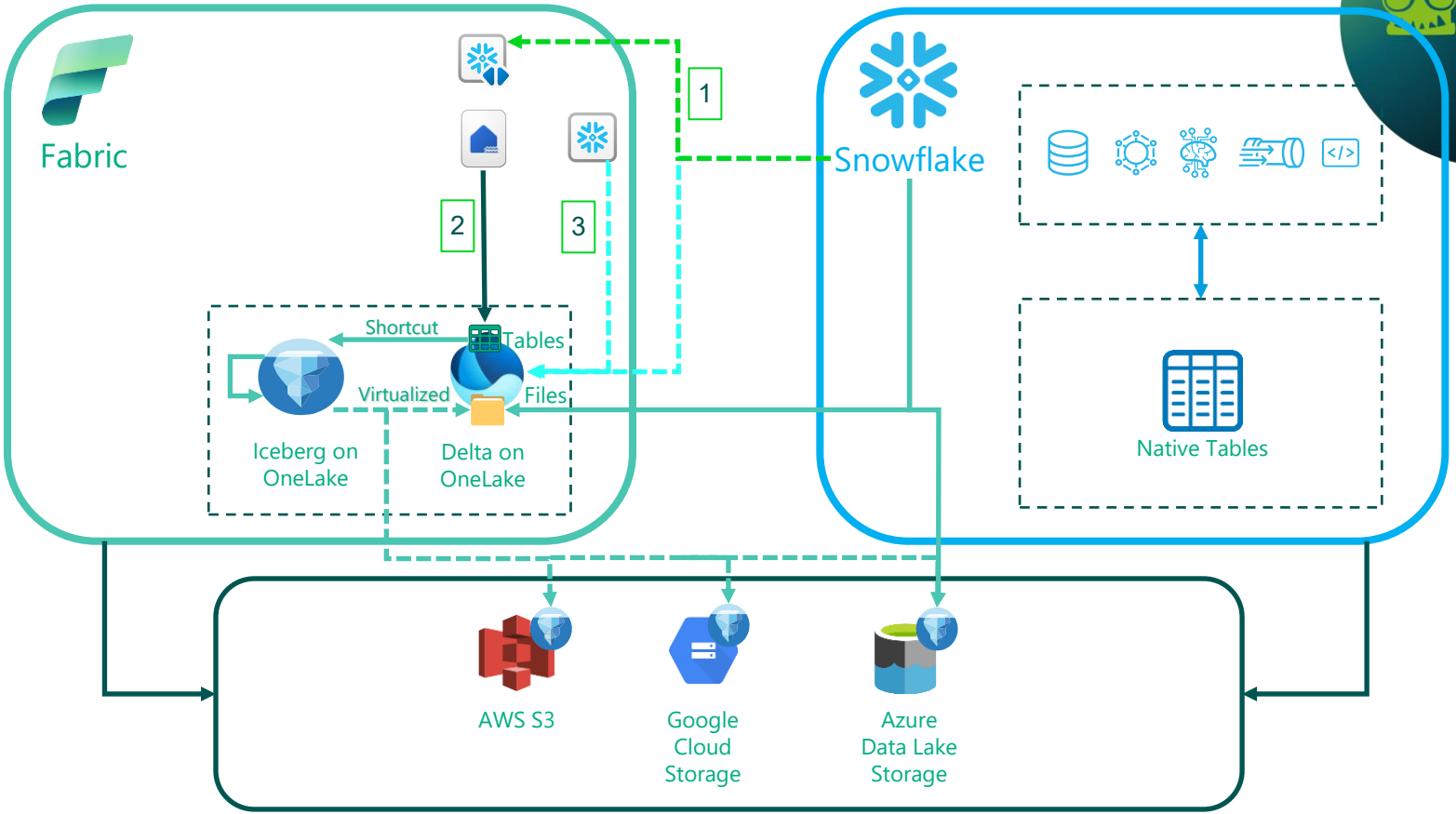
Today, that vision becomes a reality. At Snowflake's BUILD conference in London today, we are co-announcing Microsoft OneLake's interoperability with Snowflake is now **generally available**, marking a major milestone for thousands of joint customers building modern, open and interoperable data foundations. Adding to the list of already generally available integrations, we are now making the ability to natively store Snowflake-managed Iceberg tables in OneLake, the automatic conversion of Fabric data into Iceberg format for direct access from Snowflake, and new UI experiences in both platforms into general availability.

If you're interested in learning more about our collaboration with Snowflake, I recently sat down with Christian Kleinerman, Executive VP of Product at Snowflake, to discuss our integration and talk broadly about the future of our platforms and the industry:

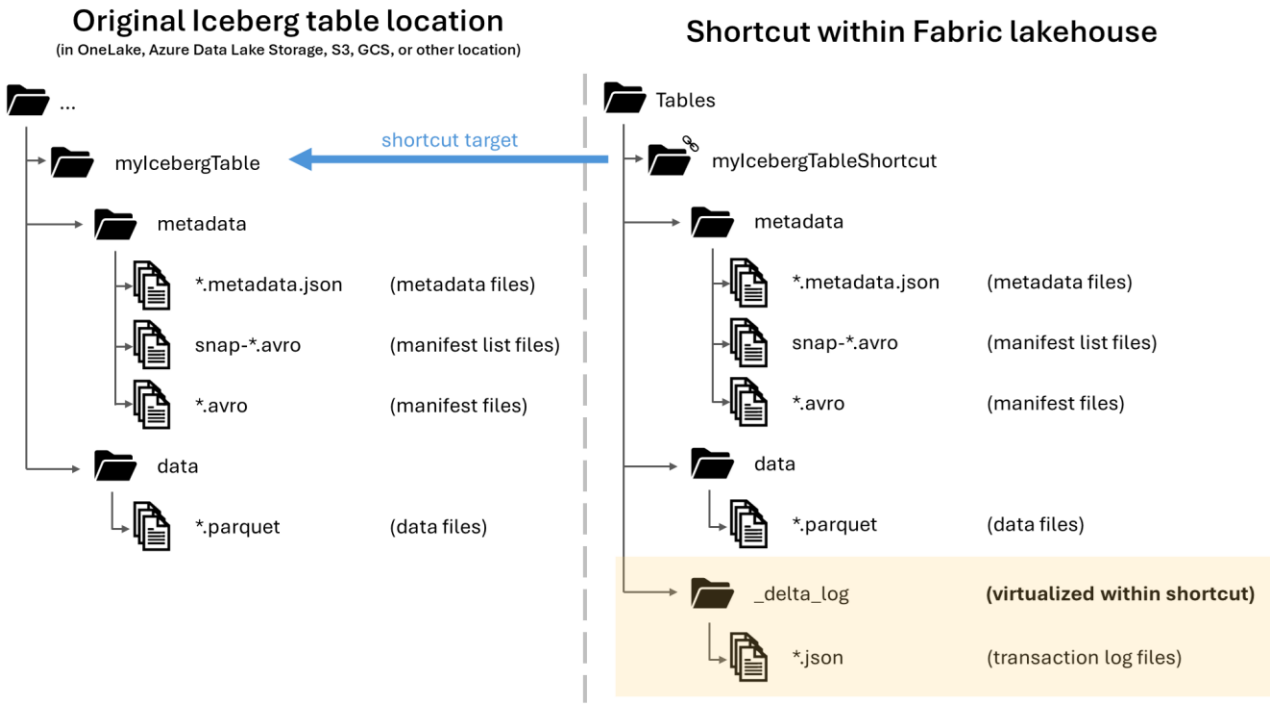
Source: Microsoft OneLake and Snowflake interoperability (Generally Available) | Microsoft Fabric Blog | Microsoft Fabric

SQL KONFERENZ 2026

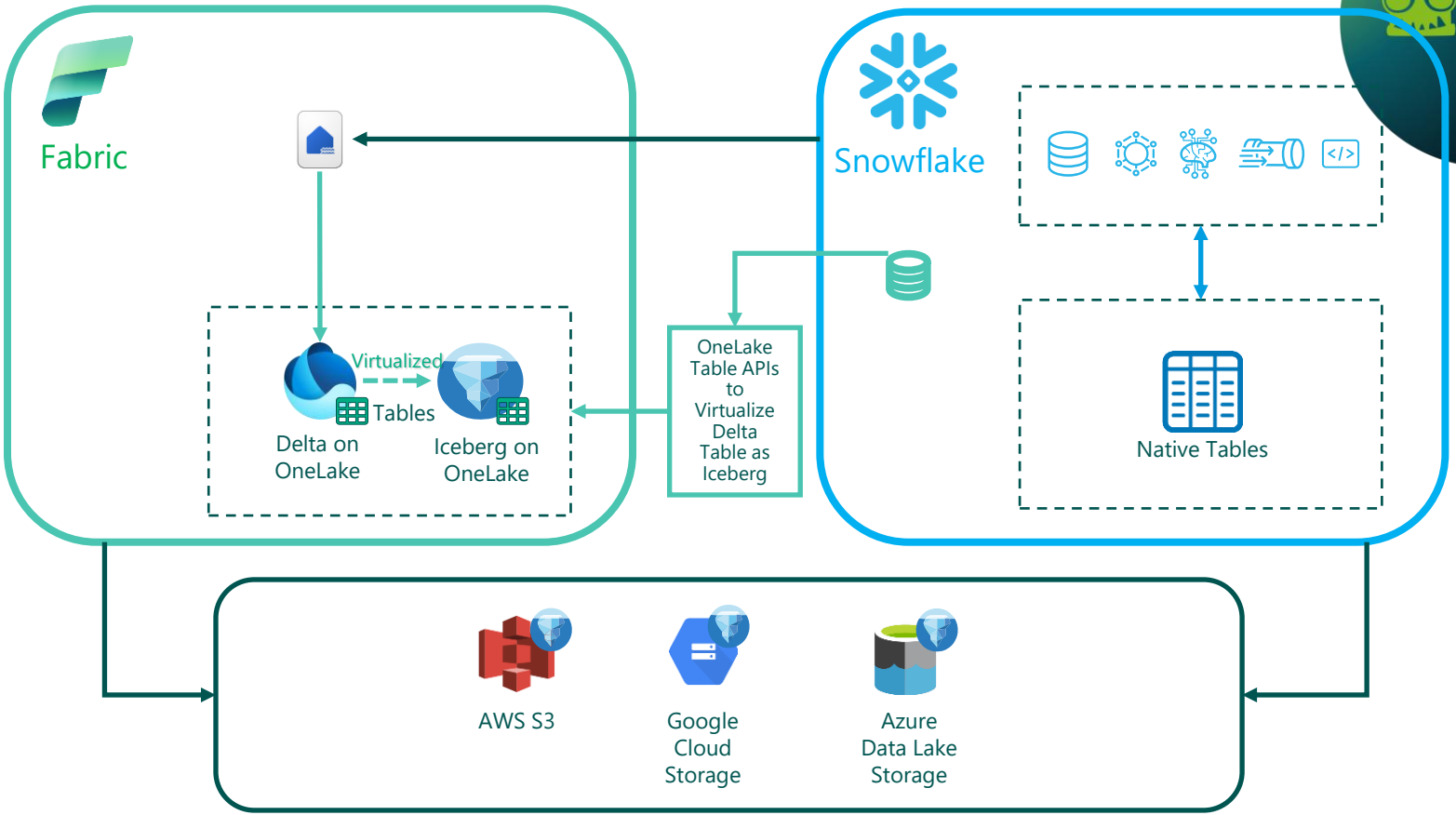
Accessing Data from Snowflake in Fabric



Accessing Data from Snowflake in Fabric



Accessing Data from Fabric in Snowflake





Integration with Snowflake Mirroring

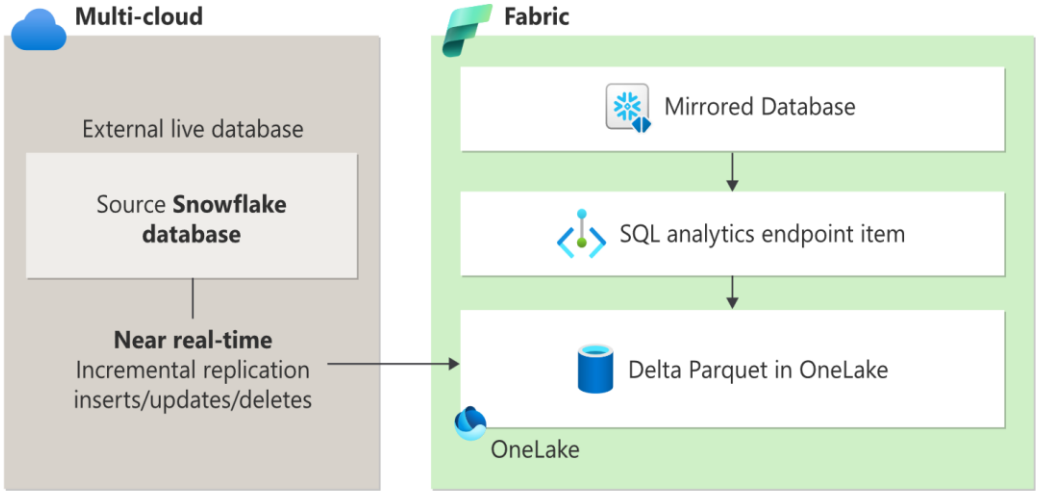
Mirroring Snowflake in Microsoft Fabric

Fabric items:

- Mirrored database
- A SQL analytics endpoint

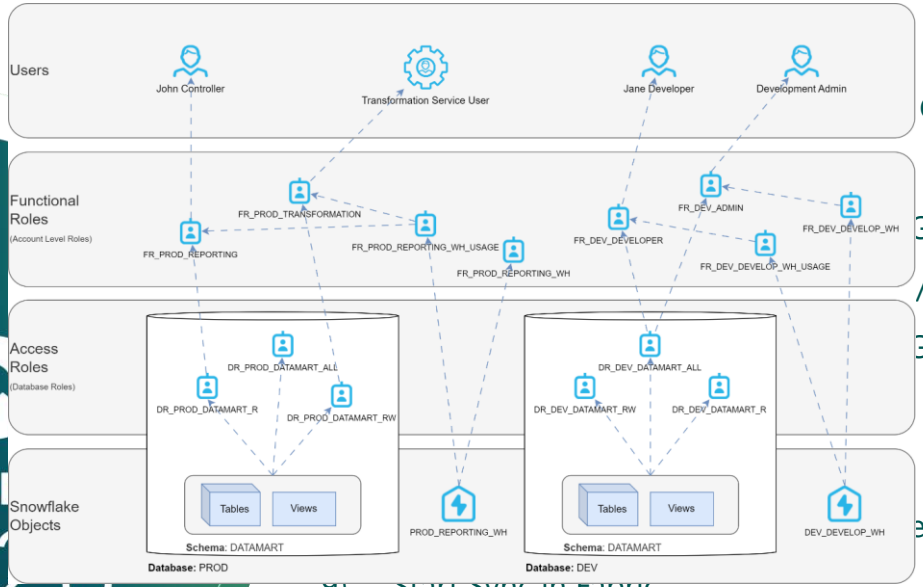
Cost considerations:

- Fabric compute used to replicate your data into Fabric OneLake is free. The Mirroring storage cost is free up to a limit based on capacity.
- Snowflake virtual warehouse compute charges
- Snowflake services compute charges



Source: Microsoft Fabric Mirrored Databases From Snowflake - Microsoft Fabric | Microsoft Learn

Step-by-Step (1)



Compute in Snowflake

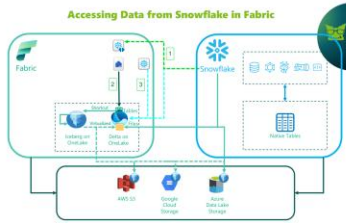
Grant Permissions

/ Grant Permissions

Grant Permissions

em

9) Start Sync in Fabric
10) Build Semantic Model
11) Build Report



Limitations



Database level limitations

- If there are no updates in a source table, the replicator engine starts to back off with an exponentially increasing duration for that table, up to an hour. The same can occur if there's a transient error, preventing data refresh. The replicator engine will automatically resume regular polling after updated data is detected.
- Only replicating native tables are supported. Currently, External, Transient, Temporary, Dynamic tables aren't supported.
- Source schema hierarchy is replicated to the mirrored database. For mirrored databases created before this feature enabled, the source schema is flattened, and schema name is encoded into the table name. If you want to reorganize tables with schemas, recreate your mirrored database.
- Mirroring supports replicating columns containing spaces or special characters in names (such as , ; { } ( ) \n \t =). For tables under replication before this feature enabled, you need to update the mirrored database settings or restart mirroring to include those columns.
- The maximum number of tables that can be mirrored into Fabric is 500 tables. Any tables above the 500 limit currently can't be replicated.
    - If you select Mirror all data when configuring Mirroring, the tables to be mirrored over will be determined by taking the first 500 tables when all tables are sorted alphabetically based on the schema name and then the table name. The remaining set of tables at the bottom of the alphabetical list will not be mirrored over.
    - If you unselect Mirror all data and select individual tables, you're prevented from selecting more than 500 tables.

~~Network and firewall~~

- ~~Mirroring does not support Snowflake instances behind a virtual network or private networking~~

Security

- Authentication via username/password and Entra
- Sharing recipients must be added to the workspace. To share a dataset or report, first add access to the workspace with a role of admin, member, reader, or contributor.

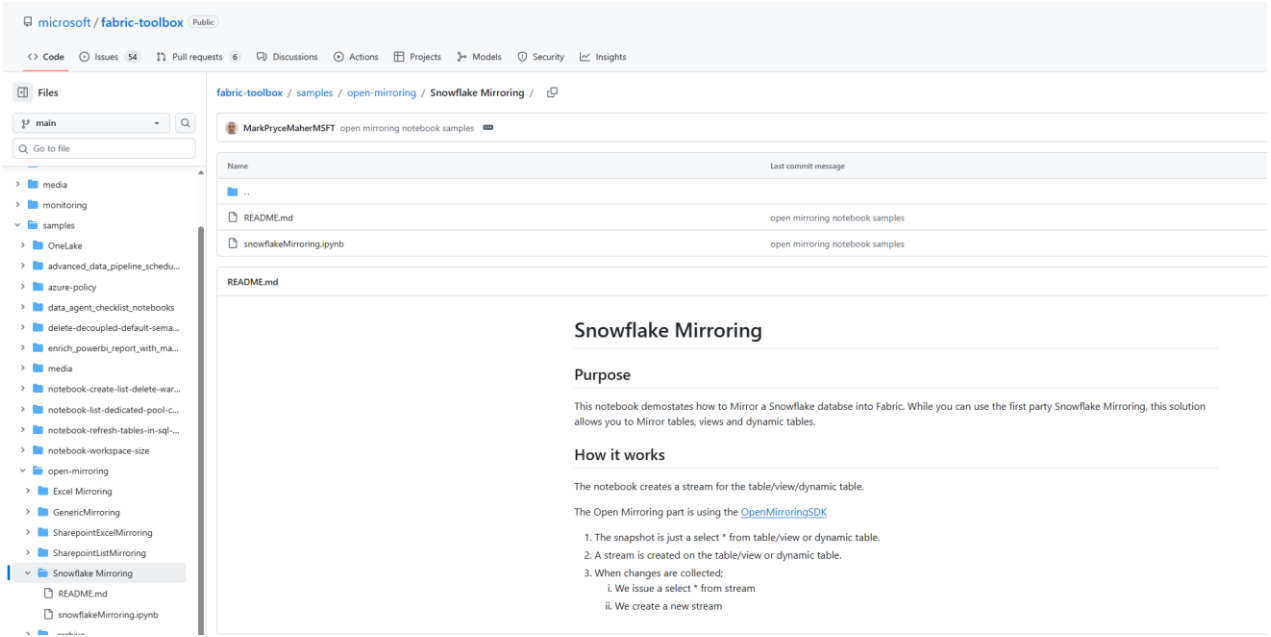
Performance

- If you're changing most the data in a large table, it's more efficient to stop and restart Mirroring. Inserting or updating billions of records can take a long time.
- Some schema changes aren't reflected immediately. Some schema changes need a data change (insert/update/delete) before schema changes are replicated to Fabric.
- When mirroring data from Snowflake to a customer's OneLake, we usually stage the data to improve performance. However, if data exfiltration from Snowflake through inline URL is disabled via PREVENT_UNLOAD_TO_INLINE_URL, direct read from Snowflake could be required. This approach can lead to slower replication times and an increased risk of connection timeouts, particularly for large datasets.

Region availability limitation

Source: Limitations for Microsoft Fabric Mirrored Databases From Snowflake - Microsoft Fabric | Microsoft Learn

Snowflake Mirroring



Source: fabric-toolbox/samples/open-mirroring/Snowflake Mirroring at main · microsoft/fabric-toolbox · GitHub



Accessing Data from Snowflake in Fabric

Accessing Data from Snowflake in Fabric



SQL KONFERENZ 2026

✅ Prerequisites

Microsoft Fabric Environment:

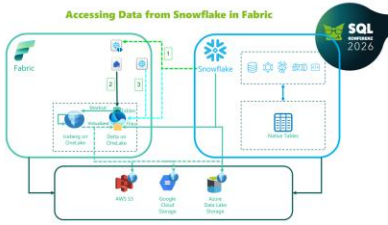
- You must have access to a Fabric workspace with appropriate permissions.
- A Fabric capacity should be enabled (Same region as Snowflake)
- Be aware of Tenant Settings!
- Service Principal

Snowflake Account:

- A valid Snowflake account with access to the database, schema, and tables you want to query + External Volume creation
- Ensure your Snowflake user has the necessary roles and privileges (e.g., USAGE on database/schema and SELECT on tables).

Step-by-Step (2)

1) Create External Volume (ADLS Gen2 / OneLake) in Snowflake
2) Create Service Principal in Azure via Consent URL
3) Add Service Principal as „Storage Blob Data Contributor“ / Fabric Workspace „Contributor“
   OneLake: Allow API access for this SP (Security Group)
4) Write Iceberg Tables within Snowflake
5) Create Shortcut to ADLS Gen2 / OneLake
6) Build Semantic Model
7) Build Report



Accessing Data from Snowflake in Fabric

- Create External Volume (ADLS Gen2 / OneLake) in Snowflake &
- Create Service Principal in Azure via Consent URL

```
-- 1 – Set up connection to external volume (ADLS Gen 2) in Snowflake
CREATE OR REPLACE EXTERNAL VOLUME adlsgen2_write_exvol
  STORAGE_LOCATIONS =  (
      ( NAME = 'adlsgen2_write'
        STORAGE_PROVIDER = 'AZURE'
        STORAGE_BASE_URL = 'azure://<ADLSGen2BlobPath>/icebergtables'
        AZURE_TENANT_ID = '<Tenant_ID>'  )  );

-- Get AZURE_CONSENT_URL
DESC EXTERNAL VOLUME adlsgen2_write_exvol;

-- Verify Volume
SELECT SYSTEM$VERIFY_EXTERNAL_VOLUME('adlsgen2_write_exvol');
```

```
-- 1 – Set up connection to external volume (OneLake) in Snowflake
CREATE OR REPLACE EXTERNAL VOLUME onelake_write_exvol
  STORAGE_LOCATIONS = (
    ( NAME='onelake_write'
      STORAGE_PROVIDER='AZURE'
      STORAGE_BASE_URL='azure://<OneLakeLakehousePath>/Files/icebergtables'
      AZURE_TENANT_ID='<Tenant_ID>' ) );

-- Get AZURE_CONSENT_URL
DESC EXTERNAL VOLUME onelake_write_exvol;

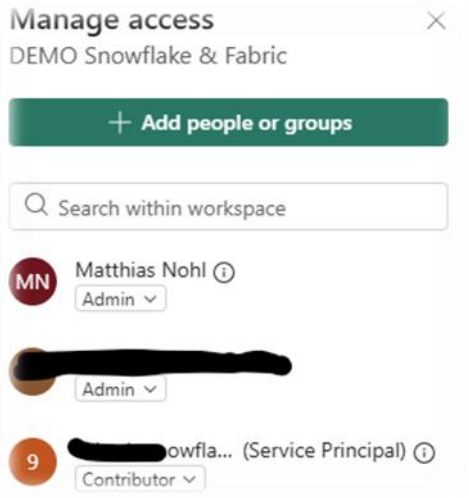
-- Verify Volume
SELECT SYSTEM$VERIFY_EXTERNAL_VOLUME('onelake_write_exvol');
```

Accessing Data from Snowflake in Fabric



Add Service Principal as „Storage Blob Data Contributor“





Fabric Workspace „Contributor“

Write Iceberg Tables within Snowflake

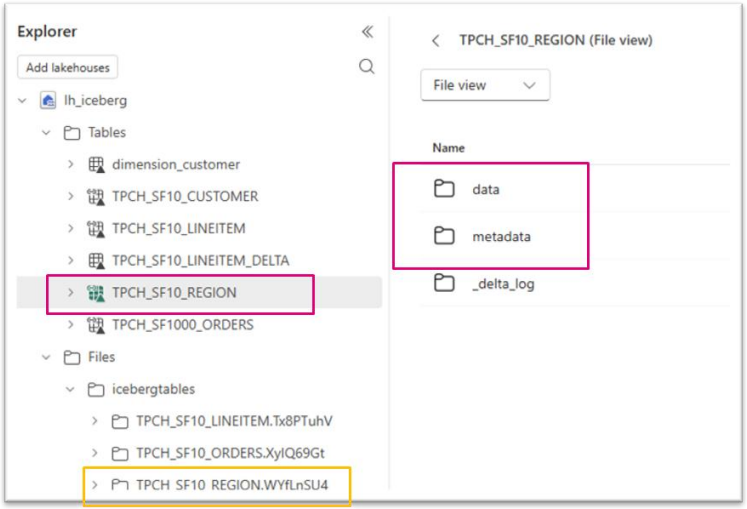
Write Iceberg Tables within Snowflake

```
// 2 - Write table

CREATE OR REPLACE ICEBERG TABLE <TableName> LIKE SNOWFLAKE_SAMPLE_DATA.<TableName>
    CATALOG = 'SNOWFLAKE'
    EXTERNAL_VOLUME = '<extVol>'
    BASE_LOCATION = '<TableName>/';

INSERT INTO <TableName> SELECT * FROM SNOWFLAKE_SAMPLE_DATA.<TableName>;

SELECT COUNT(*) FROM <TableName>
```



Create Shortcut to ADLS Gen2 / OneLake

Limitations

- Supported data types
- Type width issue
- Iceberg table metadata storage isn't portable
- Iceberg table folders must contain only one set of metadata files
- Metadata changes not immediately reflected
- Region availability limitation
- Private links not supported
- OneLake shortcuts must be same-region
- Certain Iceberg partition transform types are not supported



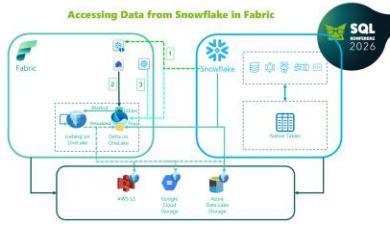
SQL KONFERENZ 2026

Source: https://learn.microsoft.com/en-us/fabric/onelake/onelake-iceberg-tables#limitations-and-considerations

Step-by-Step (3) – Snowflake database item

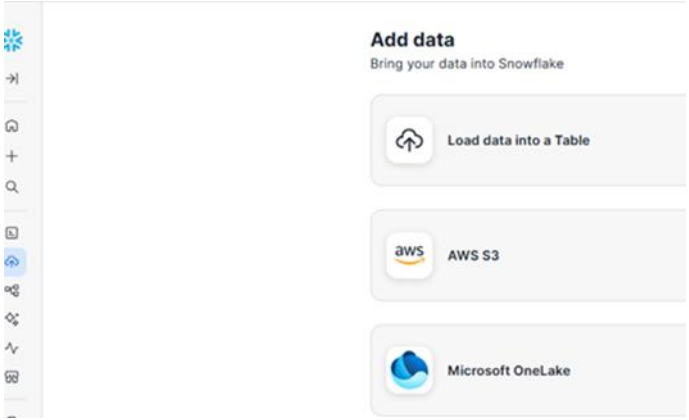
SQL KONFERENZ 2026

1) Create or use existing Warehouse for Compute in Snowflake
2) Create Functional Roles in Snowflake
3) Create Technical User in Snowflake / Grant Permissions
4) Create Fabric Cloud Connection with SNF Technical User
5) Share the connection with the Service Principal
6) Snowflake: Go to Ingestion > Add data > Microsoft OneLake
7) Provide: Workspace ID, Connection ID, SNF Database Name, External Volume Name
8) Grant Permissions for the database to the Functional Role
9) Write Iceberg Tables within Snowflake
10) Read the Iceberg Tables via SQL endpoint



Go to Ingestion > Add data > Microsoft OneLake

SQL KONFERENZ 2026



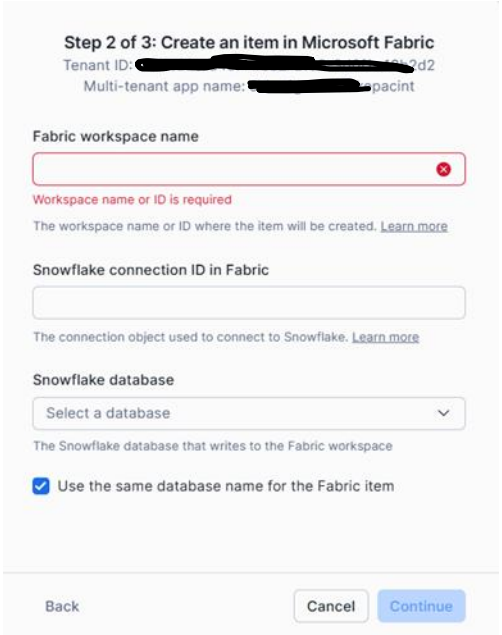


Connect to Microsoft OneLake



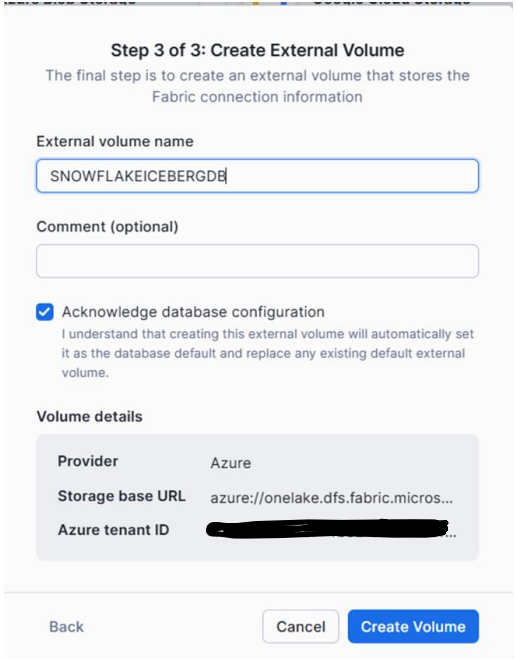
SQL KONFERENZ 2026

Create an item in Microsoft Fabric



SQL KONFERENZ 2026

Create External Volume

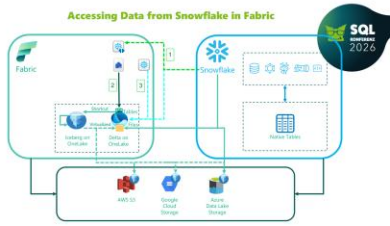


SQL KONFERENZ 2026

Step-by-Step (3) – Snowflake database item - Alternative

SQL KONFERENZ 2026

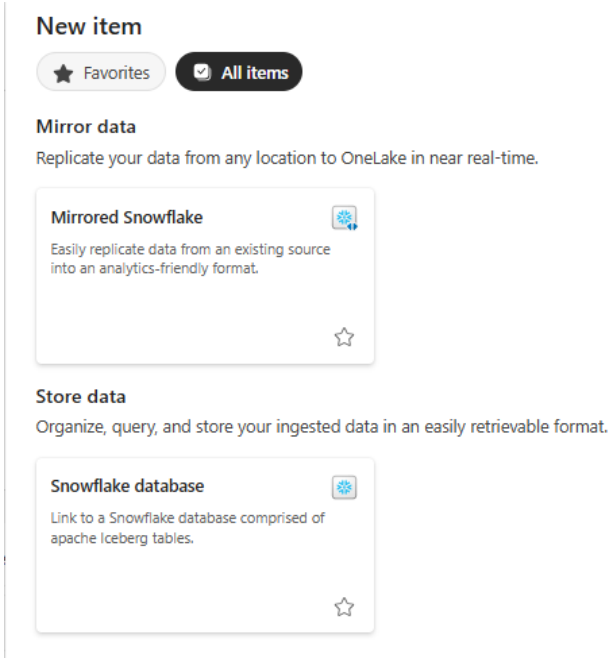
1) Steps 1 – 5
2) Create Database in Snowflake
3) Grant Permissions for the database to the Functional Role
4) Create a new Snowflake database item in Fabric
5) Copy the artifact GUID for the item
6) Create and external volume in Snowflake ([…]onelake.[…]/SnowflakeVolume)
7) Alter Database to use this external volume
8) Grant Permissions on the volume to the Functional Role
9) Write Iceberg Tables within Snowflake
10) Read the Iceberg Tables via SQL endpoint



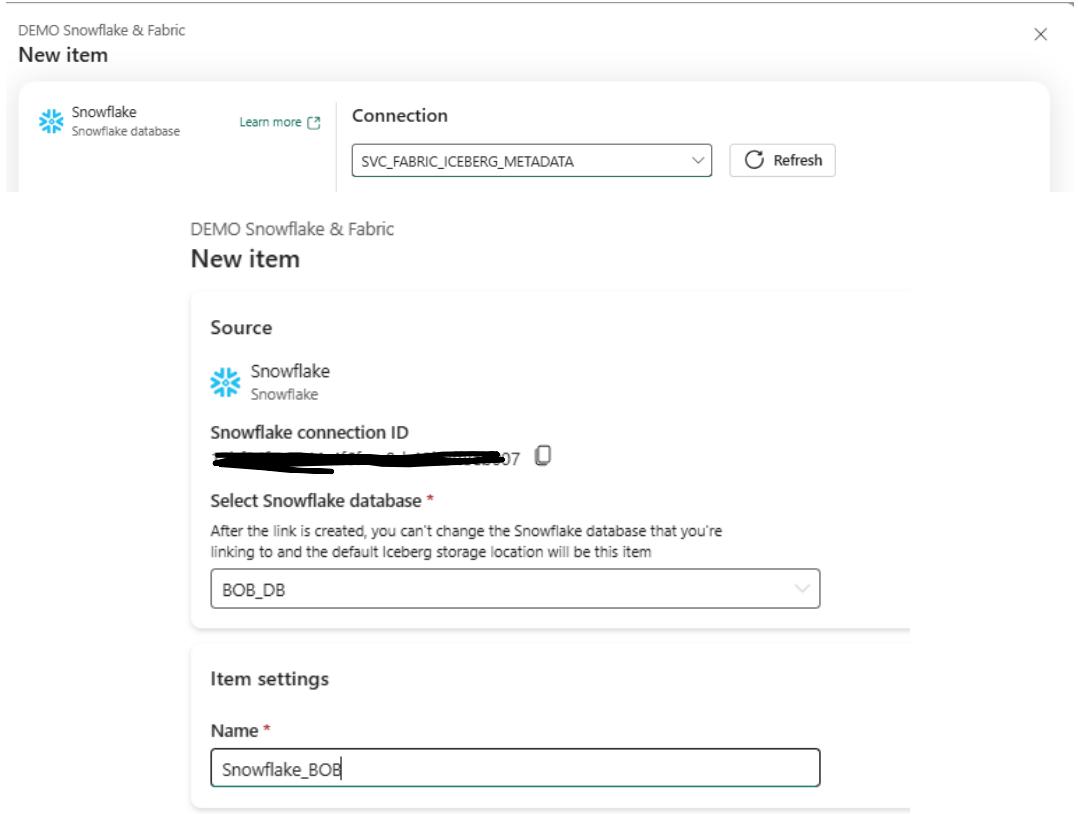
Create a new Snowflake database item in Fabric



SQL KONFERENZ 2026



Create a new Snowflake database item in Fabric



SQL KONFERENZ 2026

Create a new Snowflake database item in Fabric



SQL KONFERENZ 2026

"Create and external volume in Snowflake
([…]onelake.[…]/SnowflakeVolume)
Alter Database to use this external volume
Grant Permissions on the volume to the Functional Role"





Accessing Data from Fabric in Snowflake

Accessing Data from Snowflake in Fabric

✅ Prerequisites

Microsoft Fabric Environment:

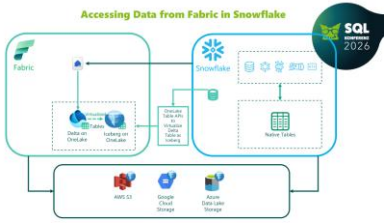
- You must have access to a Fabric workspace with appropriate permissions.
- A Fabric capacity should be enabled (Same region as Snowflake)
- Be aware of Tenant Settings!
- Service Principal



SQL KONFERENZ 2026

Step-by-Step

1) Create a Service Principal in Entra
2) Create Catalog Integration object
3) Create External Volume (OneLake) in Snowflake
4) Add Service Principal as Fabric Workspace „Contributor“
   OneLake: Allow API access for this SP (Security Group)
5) Create a Snowflake catalog linked database



Accessing Data from Fabric in Snowflake

Create catalog integration object

```
-- Create catalog integration object
CREATE OR REPLACE CATALOG INTEGRATION IRC_CATINT
    CATALOG_SOURCE = ICEBERG_REST
    TABLE_FORMAT = ICEBERG
    REST_CONFIG = (
        CATALOG_URI = 'https://onelake.table.fabric.microsoft.com/iceberg' -- Iceberg base URL at the
OneLake table endpoint
        CATALOG_NAME = '12345678-abcd-4fbd-9e50-3937d8eb1915/98765432-dcba-4209-8ac2-0821c7f8bd91' --
Fabric data item scope, in the form `workspaceID/dataItemID`
    )
    REST_AUTHENTICATION = (
        TYPE = OAUTH -- Entra auth
        OAUTH_TOKEN_URI = 'https://login.microsoftonline.com/11122233-1122-4138-8485-
a47dc5d60435/oauth2/v2.0/token' -- Entra tenant ID
        OAUTH_CLIENT_ID = '44332211-aabb-4d12-aef5-de09732c24b1' -- Entra application client ID
        OAUTH_CLIENT_SECRET = '[secret]' -- Entra application client secret value
        OAUTH_ALLOWED_SCOPES = ('https://storage.azure.com/.default') -- Storage token audience
    )
    ENABLED = TRUE
;
```

SQL KONFERENZ 2026

Accessing Data from Fabric in Snowflake

Create External Volume (OneLake) in Snowflake

```
-- Create external volume object
CREATE OR REPLACE EXTERNAL VOLUME IRC_EXVOL
    STORAGE_LOCATIONS =
    (
        (
            NAME = 'IRC_EXVOL'
            STORAGE_PROVIDER = 'AZURE'
            STORAGE_BASE_URL = 'azure://onelake.dfs.fabric.microsoft.com/12345678-abcd-4fbd-9e50-
3937d8eb1915/98765432-dcba-4209-8ac2-0821c7f8bd91'
            AZURE_TENANT_ID='11122233-1122-4138-8485-a47dc5d60435' -- Entra tenant id
        )
    )
    ALLOW_WRITES = FALSE;
;

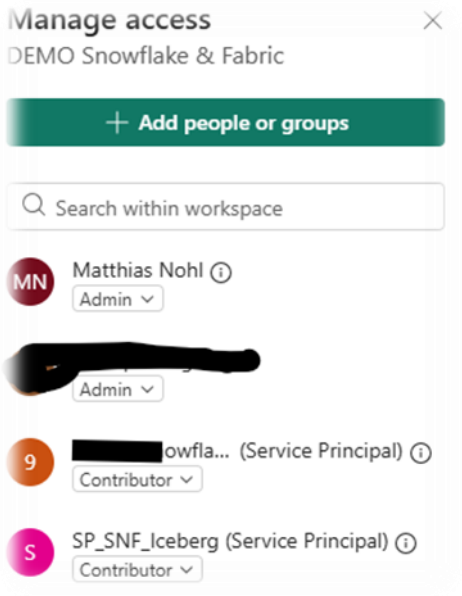
-- Describe the external volume
DESC EXTERNAL VOLUME IRC_EXVOL;
```



SQL KONFERENZ 2026

Accessing Data from Fabric in Snowflake

Add Service Principal as Fabric Workspace „Contributor“



SQL KONFERENZ 2026

Accessing Data from Fabric in Snowflake

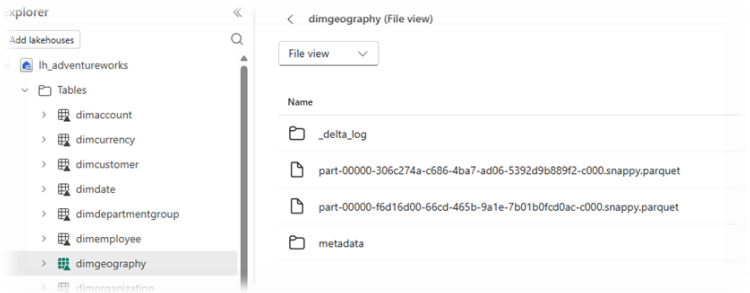
Create a Snowflake catalog linked database

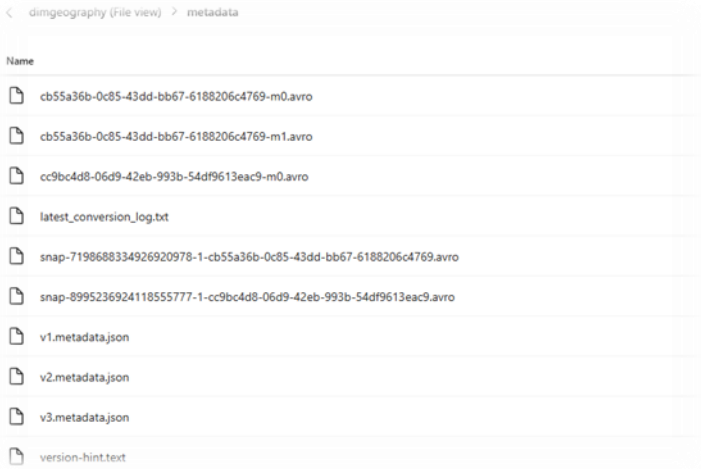
```
-- Create a Snowflake catalog linked database
CREATE OR REPLACE DATABASE IRC_CATALOG_LINKED
    LINKED_CATALOG = (
        CATALOG = 'IRC_CATINT'
    )
    EXTERNAL_VOLUME = 'IRC_EXVOL'
;

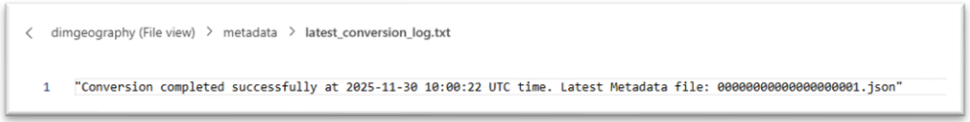
SELECT SYSTEM$CATALOG_LINK_STATUS('IRC_CATALOG_LINKED');
```

SQL KONFERENZ 2026

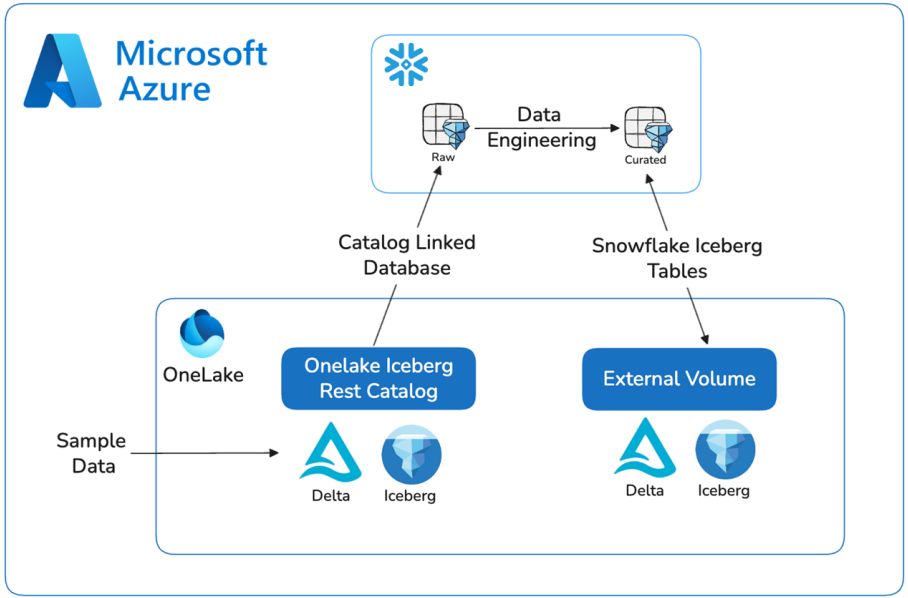
Accessing Data from Fabric in Snowflake







Building an Iceberg Lakehouse with Snowflake and Microsoft OneLake





Source: Building an Iceberg Lakehouse with Snowflake and Microsoft OneLake



What should be used and how?

Decision Slide: Mirroring vs. Virtualize vs. Hub

1. **First decide the "Primary Writer" (System of Record)**

   If Snowflake remains the primary writer, prefer Snowflake → *Fabric patterns*

   If OneLake/Fabric is the primary writer, prefer Fabric → *Snowflake patterns* using Iceberg interoperability.

2. **Need fast BI/AI activation in Fabric (low effort, near real-time)? → Use Mirroring**

   Mirroring continuously replicates Snowflake data into Fabric OneLake and auto-creates a mirrored database + SQL analytics endpoint for immediate analytics

3. **Need "Open Lakehouse" (zero-copy sharing, minimize lock-in)? → Use Iceberg + OneLake virtualization/shortcuts**

   OneLake supports working with Delta + Iceberg via metadata virtualization; Fabric can consume Iceberg tables (e.g., written by Snowflake) without duplicating data.

4. **Want OneLake/Fabric as the enterprise hub, but keep Snowflake workloads? → Use Fabric → Snowflake via Iceberg.**

   With GA interoperability, Snowflake can access data in OneLake by using Iceberg-compatible metadata (Delta ↔ Iceberg), enabling cross-platform use without forcing a single engine everywhere.

5. **Practical guardrails (cost + limits) determine the final choice**.

   Mirroring has documented constraints (e.g., native tables only, max 500 tables, polling backoff up to an hour on unchanged tables) and Snowflake-side compute costs may apply.

6. **"Who uses what?"**

   BI teams → Mirroring for quick Power BI/Fabric consumption;

   Platform/Architecture teams → Iceberg/virtualization for Open Lakehouse + vendor lock-in mitigation;

   Central data teams → Fabric hub pattern with curated Delta in OneLake and Snowflake consuming via Iceberg.

SQL KONFERENZ 2026

Q&A





Feedback

THANKS ❤️

