IBM Db2

Datalake Tables in Db2: Open, Flexible Access to Your Data Wherever It Lives

Dominic So
Senior Software Developer, Db2 Big SQL Development

May 13, 2026

IBM

Agenda

- Modern Data Challenges
- What are Datalake Tables?
- Why this matters
- Architecture Overview
- Use Cases and Applications
- Using Datalake Tables
- Putting it into practice
- To learn more

Modern Data Challenges

- Modern data lives everywhere:
  - Object Stores
  - Data lakes
  - Lakehouses
  - Traditional warehouse
- Challenges:
  - Query data with copying
  - Vendor lock in
  - Waiting before data can be used

Modern Data Challenges

- Introduce Datalake Tables into Db2
  - Leave your data on Object Storage
  - Query your open-format data directly in Db2
  - Leverage the existing and familiar Db2 features and rich SQL on your open format data
  - Data remains in open format, no lock in, free to use the best of breed tools to process and access the data

What are Datalake Tables?

- Db2 tables whose data resides outside Db2 Engine, i.e. on Object Storage, either on Cloud or on-premises
- Db2 manages the table definition and metadata
- Data stored in open table formats (Apache Hive and Apache Iceberg)
- Stored using open data formats (Parquet, ORC, Avro, JSON, Text/CSV)

What are Datalake Tables?

Data is not owned by Db2

- Data is not owned by Db2 and remains on your object storage environment
- Supported object stores include IBM Cloud Object Storage, AWS S3, Azure Blob Storage, or on-premises S3-compatible storage
- Db2 is granted access to the objects and maintains their metadata
- Never lose autonomy over your data, so you can always use the best engines, frameworks, cloud service you prefer

What are Datalake Tables?

Open table formats (Apache Hive and Apache Iceberg)

- 2 types of Datalake tables: Hive and Iceberg
- Hive-style Datalake table organize data in a traditional folder-based layout on object storage
- Hive Tables ideal for append-only scenarios such as logs, historical records, and regulatory archives
- Iceberg tables have ACID properties for transactions, schema evolution, partition pruning, and snapshot isolation
- Iceberg tables suitable for workloads that require updates or rely on lakehouse-style data modeling

What are Datalake Tables?

Open data formats (Parquet, ORC, Avro, JSON, Text)

- Supported open data formats include Parquet, ORC, Avro, JSON, and Text
- These formats are industry standards that allow multiple engines (Spark, Prestro/Trino, etc.) to process the same data without conversion
- Allow interoperability and reduces data duplication across teams and tools

Why this matters

No vendor lock-in

- Data remains portable and accessible from a wide range of engines because it’s stored in open formats on object storage
- Db2 participates as one of many consumers, ensuring your platform choices remain flexible

Why this matters

Separation of compute and storage

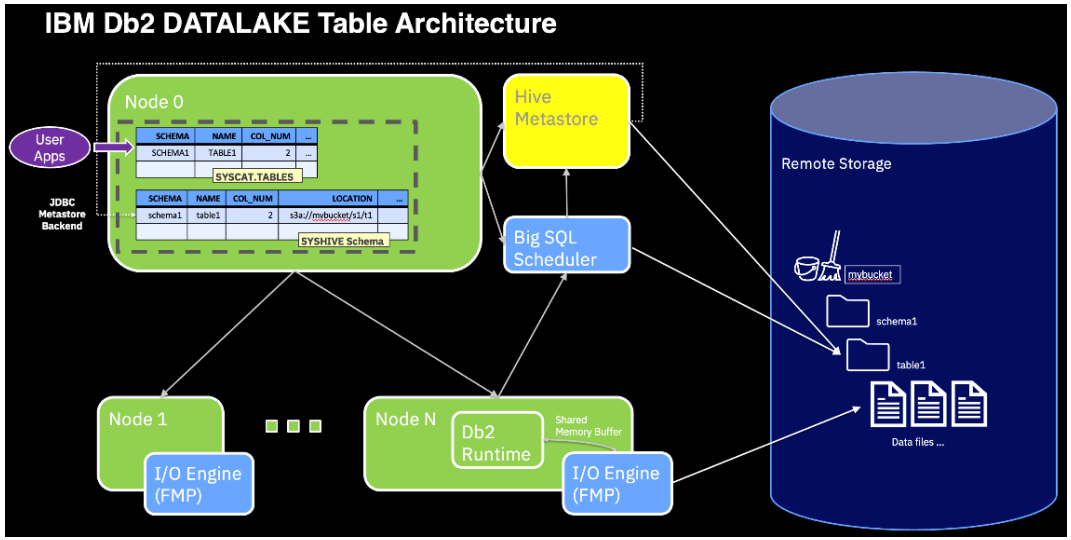
- Datalake tables allow you to separate the compute and storage
- Object storage provides virtually unlimited capacity and durability at low cost
- Db2 compute can be scaled independently from storage
- Allows you to optimize workloads for price or performance without migration overhead

Why this matters

Full Db2 SQL experience on open data

- Benefit from using Db2’s rich SQL support, mature optimizer, and decades of performance innovation
- Can join, filter, aggregate, and analyze Datalake table exactly like native Db2 tables
- Leverage your teams’ existing knowledge of Db2

Architecture Overview



When a SQL statement references a Datalake table is submitted:

- Existing Db2 Compiler is used
- Db2 Compiler sees a Datalake Table and calls out to a Big SQL Scheduler component to request the table's metadata from the Hive Metastore
- Existing Db2 Optimizer uses stats previously collected on the Datalake tables via Analyze command to produce the access plan
- Existing MPP Runtime is used to run the access plan
- Big SQL IO FMPs are used an all Db2 nodes to read/write to the Object Store

Use Cases and Applications

Data sharing and silo reduction

- Centralize raw and curated data once on object storage
- Allow teams access the same open-data files via the tools they prefer
- No need to duplicate data, and a single governance layer can manage permissions, lineage, and quality rules centrally

Use Cases and Applications

Data warehouse augmentation

- Offload historical and low-value data to low-cost object storage while still querying it and joining it with your high-value warehouse data for new insights, all natively from Db2
- Keep performance-sensitive data in Db2 tables while BI/analytics that are less latency-sensitive can read from Datalake tables
- Use Datalake tables to pre-process data in the lake before loading curated tables into the warehouse

Use Cases and Applications

Set up a Datalake / Lakehouse or integrate with an existing one

- Create Datalake tables in Db2 and point them at your chosen object storage
- Ingest raw or semi-structured data directly to storage (via any application) and query it with SQL in Db2
- Integrate at the catalog level with your existing data lake or lakehouse, to create new tables through Db2 or take advantage of Db2 industry leading performance to process existing ones.

Using Datalake Tables

Platform availability

- In Db2 12.1.4, the Datalake table feature becomes available on traditional on-premises installations of Db2
- These same capabilities were previously available only in Db2 Warehouse SaaS and containerized Db2 Warehouse environments
- It is supported on Linux (x86 and PPCLE) across Db2 and Db2 Warehouse deployments on Red Hat OpenShift and Kubernetes.

Using Datalake Tables

Table formats

Db2 supports both Hive and Iceberg Datalake tables.

- Hive Datalake tables are best used for append-only workloads.
- Iceberg Datalake tables provide ACID transactions, making them suitable for mutable analytical datasets.

Using Datalake Tables

Data formats

- Supported data file formats:
  - Parquet
  - ORC
  - Avro
- Hive Datalake Tables also support:
  - JSON or Text.
- Parquet and ORC offer the best query performance due to their columnar layout and binary encoding.

Using Datalake Tables

Object storage providers

- Supported Object Storage providers:
  - IBM Cloud Object Storage,
  - AWS S3
  - Azure Blob Storage
  - Red Hat Ceph Storage,
  - S3-compatible systems

Putting it into practice

Installing additional datalake component

- An additional installation process is required to enable Datalake table functionality on top of an existing Db2 installation
- Download the Db2 Datalake supplementary package from IBM support website
- Run install script

```
/opt/ibm/db2/v12.1.4/install/db2installDatalake --file /tmp/v12.1.4_linuxx64_datalake.tar.gz
```

- Run setup script to make the instance Datalake enabled

```
${DB2_HOME}/bin/datalake-instance-setup
```

Putting it into practice

Installing additional datalake component

You can verify that your instance is enabled for Datalake tables by checking the DATALAKE_INSTANCE DBM CFG parm:

```
$ db2 get dbm cfg | grep -i datalake_instance

 Datalake Instance (DATALAKE_INSTANCE) = YES
```

Putting it into practice

Create a Datalake database

- Add DATALAKE keyword to CREATE DATABASE command to create a datalake database

```
CREATE DATABASE dldb DATALAKE
```

- codeset must UTF-8 and collation must be IDENTITY (either implicitly or specified explicitly)
- To make an existing database a datalake database, use SYSPROC.ENABLE_DATALAKE_DB stored procedure

```
call ENABLE_DATALAKE_DB( 'SYSHIVE' )
```

- Create as many Datalake databases as you like
- Can activate only 1 Datalake database at a time

Putting it into practice

Create a storage alias

1) Create a storage alias using the SYSIBMADM.STORAGE_ACCESS_ALIAS.CATALOG procedure to register the connection information for your object storage container and safely store its credentials:

This example uses IBM Cloud Object Storage:

```
CALL SYSIBMADM.STORAGE_ACCESS_ALIAS.CATALOG(

  'datalakealias', -- The alias name

  's3', -- This storage service is S3 compatible

  's3.us-south.cloud-object-
storage.appdomain.cloud', -- endpoint

  '${user}',

  '${password},

  'mybucket',

  'mypath',

  'U', -- The grantee (next arg) type: User,
Group or Role

  'user1' -- Who can use this alias to create SQL
objects

);
```

Putting it into practice

Create a datalake schema

2) Next, we create a Db2 schema

```
CREATE SCHEMA myschema;
```

3) Extend the Db2 schema with a Datalake schema and tie it to the storage alias and path

```
CREATE DATALAKE mydlschema ON DB2
SCHEMA myschema

LOCATION
'DB2REMOTE://datalakealias/mybucket/my
path';
```

Putting it into practice

Create Datalake table

4) New Datalake tables created in the Datalake schema will land in the storage alias and path prefix with no need to use the LOCATION clause

```
CREATE DATALAKE TABLE
mydlschema.datalaketbl1 (i INT)

STORED BY ICEBERG;
```

5) Query it like any Db2 table:

```
SELECT * FROM mydlschema.datalaketbl1;
```

Putting it into practice

New DBM CFG parms

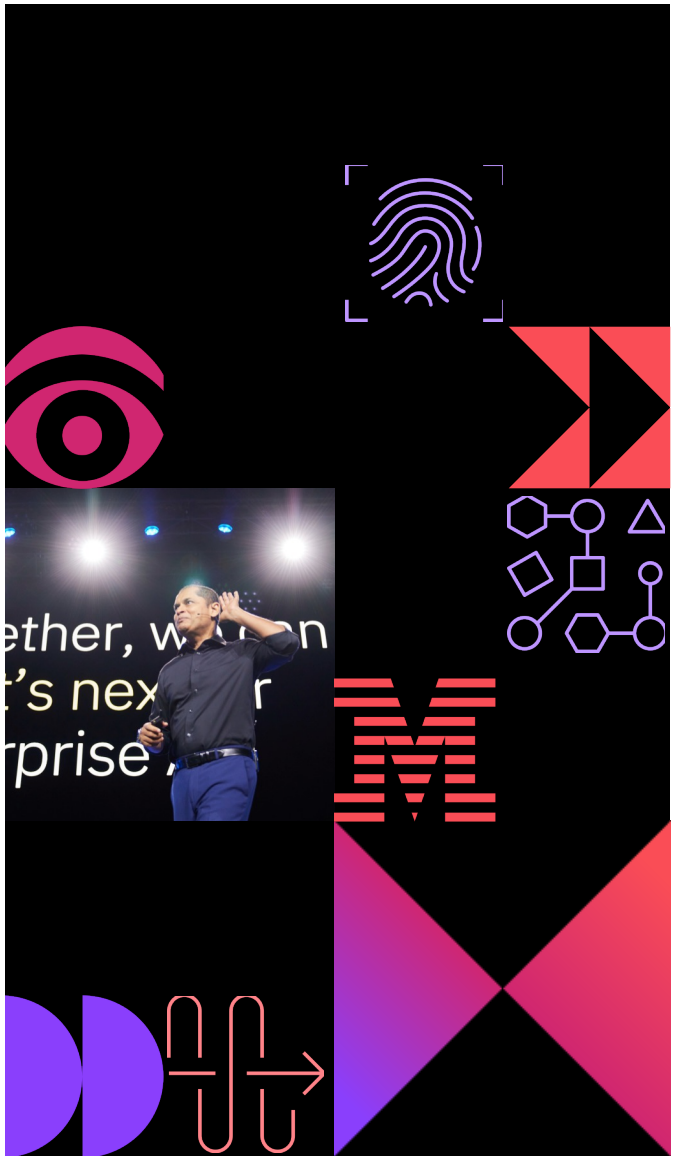
- dl_io_heap_sz - Maximum heap size for Datalake I/O configuration parameter
- Max heap for Big SQL IO FMPs
- Default is 524288 4KB (2 GB)
- sys_java_heap_sz - Maximum Java interpreter heap size for System routines configuration parameter
- Max heap for System FMP, which runs a set of routines in SYSHADOOP schema that is used to process Datalake tables
- Default is 524288 4KB (2 GB)

To learn more

Datalake tables - https://www.ibm.com/docs/en/db2/12.1.x?topic=datalake-tables

Installing the Datalake table feature for Db2 - https://www.ibm.com/docs/en/db2/12.1.x?topic=tables-installing-datalake-table-feature-db2

Thank you



Call for Sessions

Submission deadline: May 22

IBM TechXchange 2026

October 26–29 | Atlanta, GA

Clients, Partners & IBMers are eligible to submit sessions

Share your expertise with thousands of technical professionals!

What we are looking for

- Real-life use cases and implementation stories
- Technical deep dives (non-promotional)
- Demonstrated expertise with IBM products
- Sessions aligned to 2026 priorities: enterprise AI applications, key products, technical skills
- A compelling title and concise, clear abstract

Session types

Technology Breakout (45 minutes)
In-depth technical session featuring architectures, demos, best practices, client/partner stories

Tech Talk (20 minutes)
Fast-paced spotlight on use cases, technical trends, and special-topic insights.

Client & partner submissions only for Tech Talks.

Technology areas

AI Applications | Data Management | Cloud & Platform Engineering | Infrastructure | Security, Risk & Compliance

Special perks

- A complimentary full conference pass (that's a $1599 USD value!)
- Speaker training and content support
- Social tools to promote your session

Submission deadline: May 22

Get started →