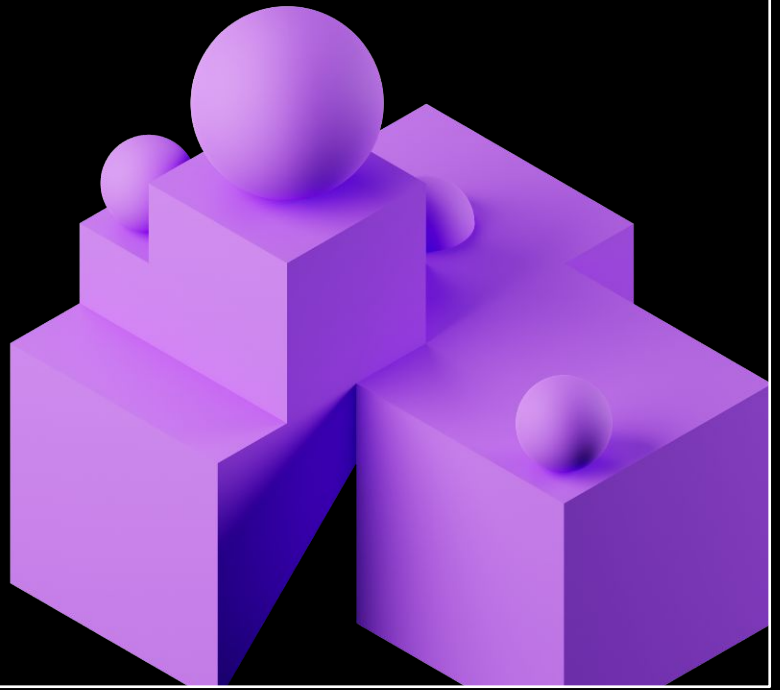
DATA+AI SUMMIT
BY databricks

Building a Minimalistic Open Lakehouse

Using open source: Apache Spark, Project Nessie & Apache Iceberg



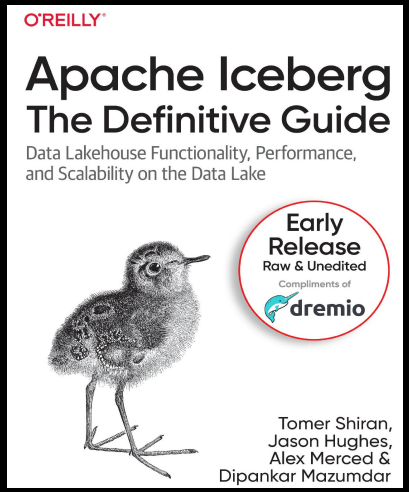
Databricks
2023

About Me

Current: Developer Advocate, *Dremio*

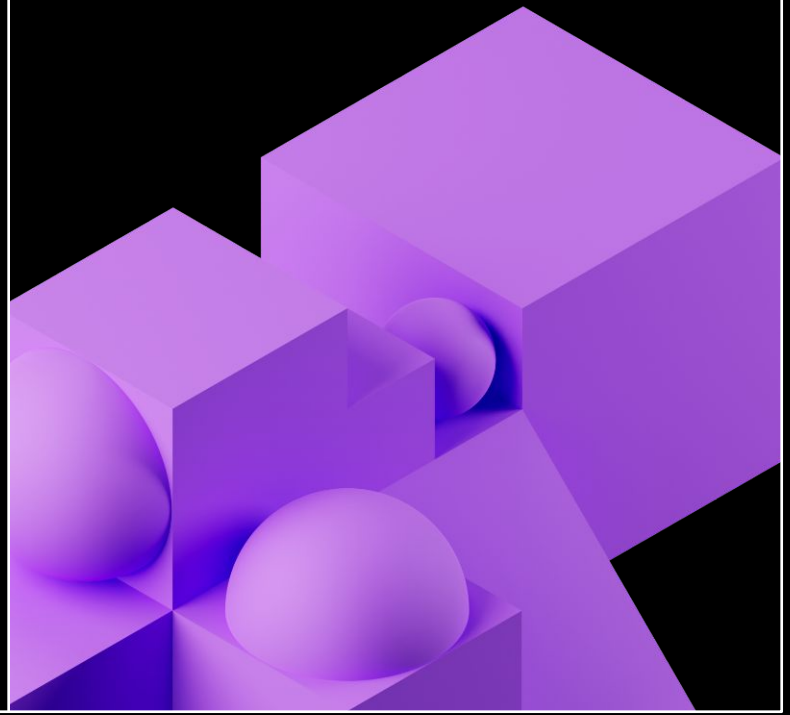
Open source: Apache Iceberg, Arrow, Project Nessie

Past: Software Eng, Data Viz/ML, Data Infra



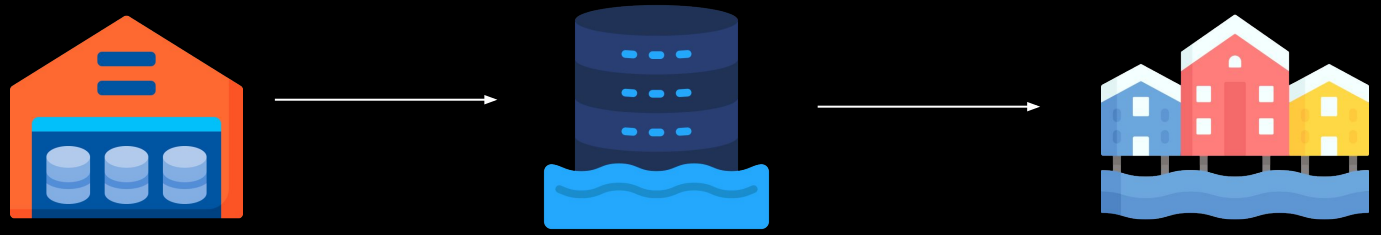
Agenda

- Evolution of Data Architecture
- Components of an Open Lakehouse
- Building a Lakehouse using Open source
- CRUD operations, Metadata, Time-travel, Optimizations



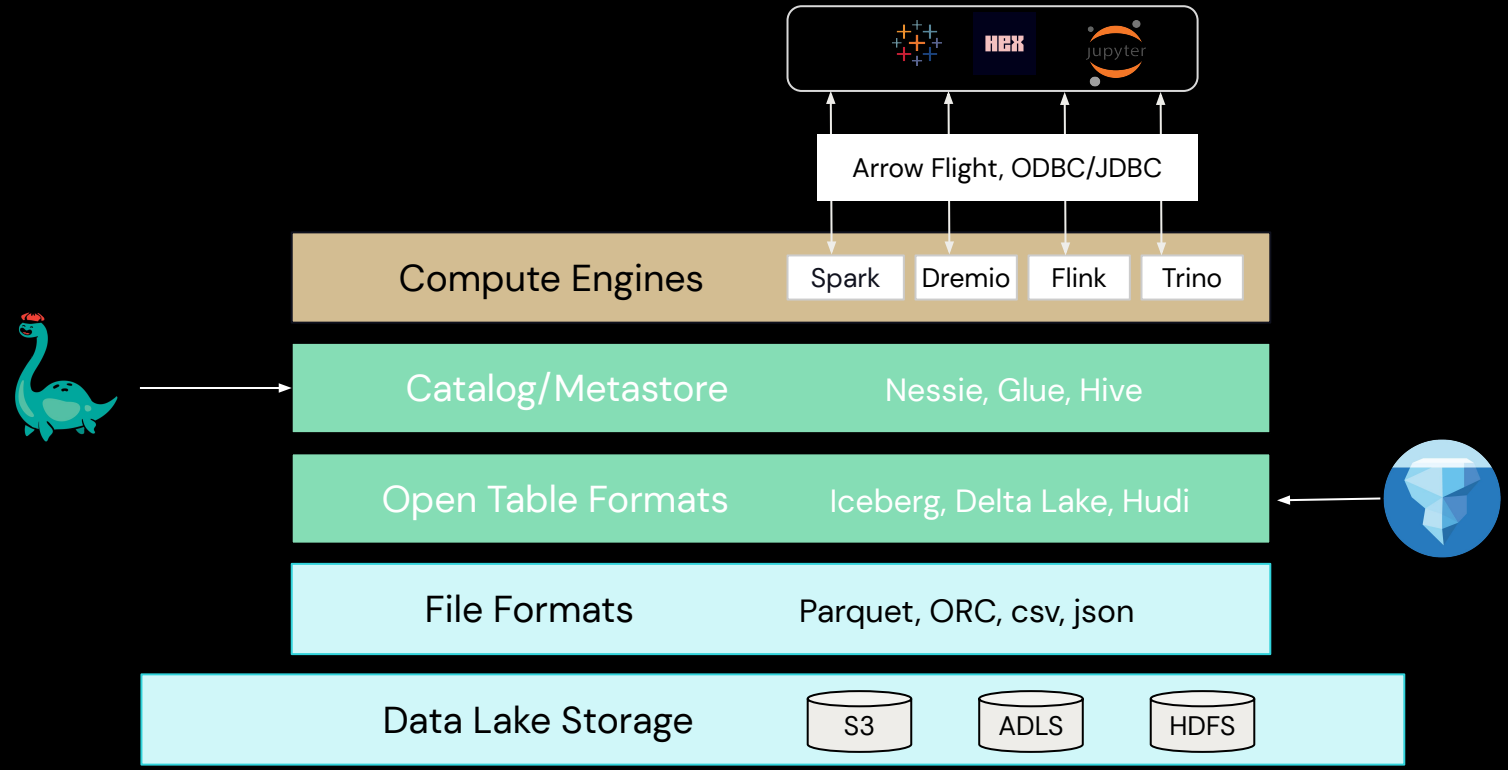
Evolution of Data Architecture

How did we get here?



Centralized, reliable data platform
Democratize data
Data Warehouse -> Data Lakes -> Lakehouse

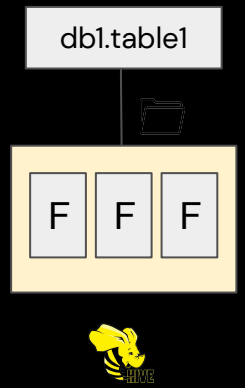
Components of an Open Lakehouse

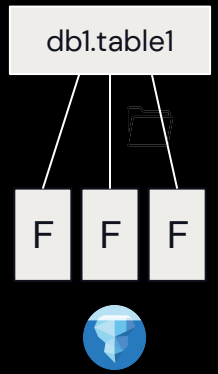


What is a table format?

- A way to organize a dataset's files to present them as a single "table"
- A way to answer the question "what data is in this table?"

directories

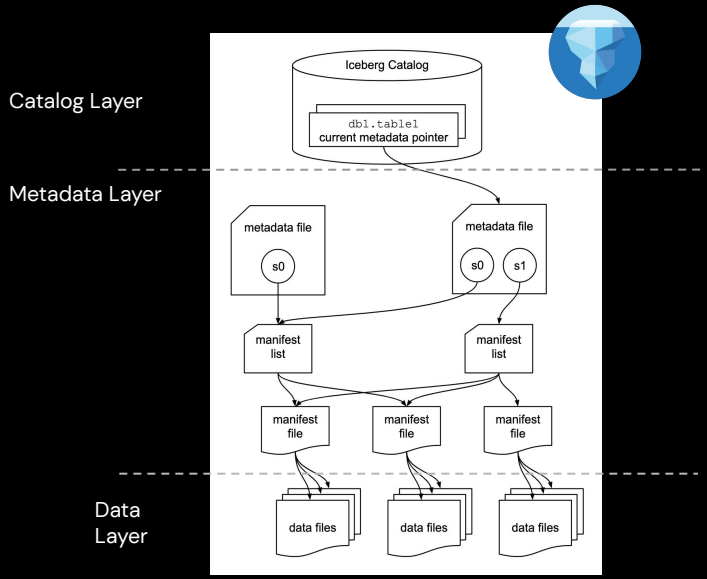




a canonical list of files

Apache Iceberg

Table format



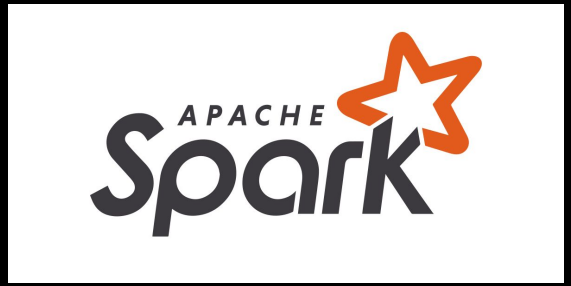
- an ‘open’ table format for analytical datasets in data lake
- Capabilities such as Expressive SQL, schema/partition evolution, time travel
- Each write creates a Snapshot
- Alternatives – Delta Lake, Apache Hudi

DELTA LAKE

Apache hudi

Apache Spark

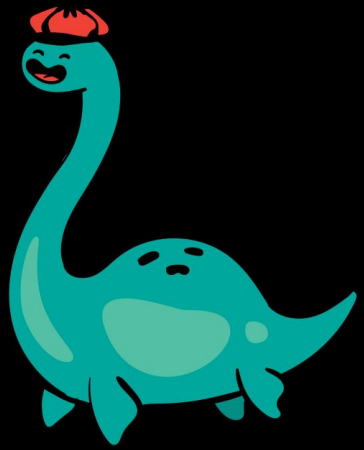
Compute Engine



- Apache Spark is one of the most feature-rich compute engine
- DDL, Read, Writes, Structured Streaming
- Spark Procedures - snapshot, metadata, CDC, Table migration
- Choice - Dremio, Apache Flink, Trino

Project Nessie

Catalog/Metastore



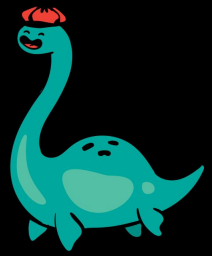
- Transactional Catalog for Data Lakes
- Brings openness to a Lakehouse; Consistency to Iceberg tables
- Nessie is to Data Lake WHAT Git is to Code Repo
- Works with Apache Iceberg & Delta Lake**
- Alternatives - AWS Glue, Hive Metastore, Dremio Arctic, REST

Project Nessie

Data Branching

- Multi-table consistency/transactions
- Experimentation (isolated dev/test)
- Pre-prod verification (stage->prod)

```
CREATE BRANCH etl
[Kafka] Data Ingestion
[Spark] Transformation 1
[Spark] Transformation 2
USE BRANCH main
MERGE BRANCH etl
```

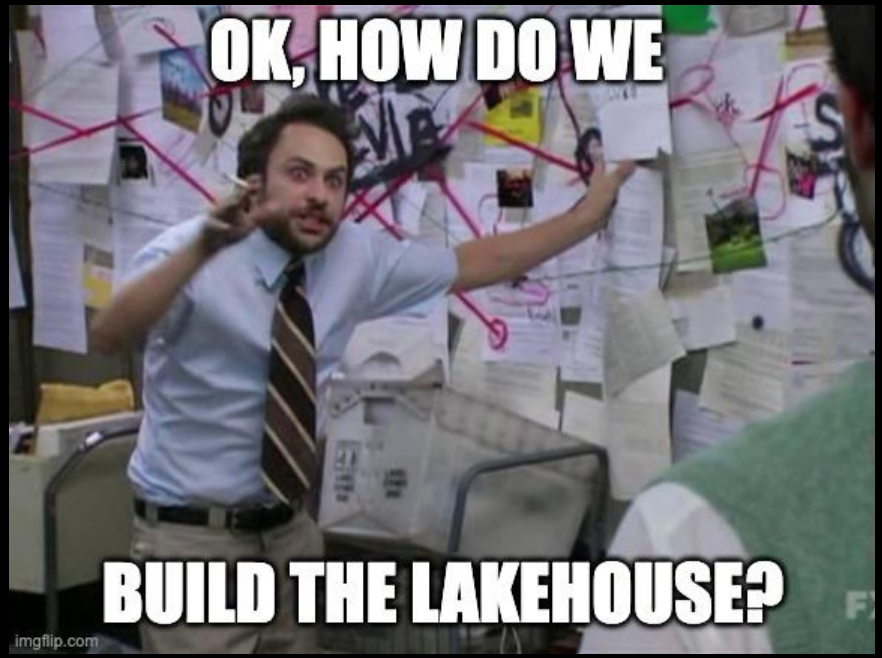


Data Version Control

- Reproducibility
- Compliance
- Historical comparisons

```
USE BRANCH 'main'
SELECT * FROM t1 // main implicit
SELECT * FROM t1@etl
SELECT * FROM t1 AT '2020-10-26'
SELECT * FROM t1@etl AT '2020-10-26'
```

Building Lakehouse



Bring the different components together

Building Lakehouse

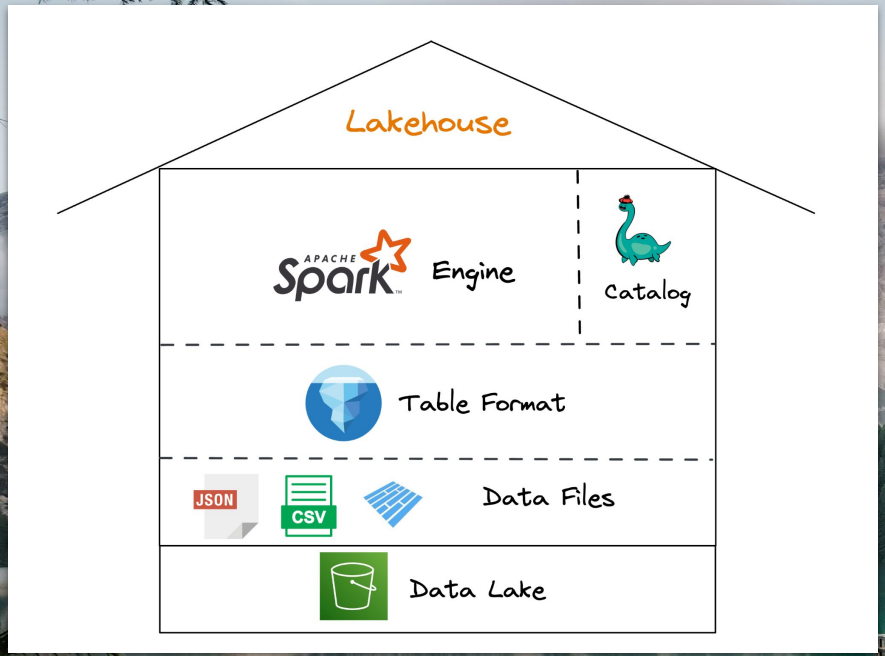


Table format: Apache Iceberg
Compute engine: Apache Spark
Catalog: Project Nessie

Configurations

```
conf.set(
    "spark.jars.packages",
    f"org.apache.iceberg:iceberg-spark-runtime-3.3_2.12:1.2.0,org.projectnessie:nessie-spark-extensions-3.3_2.12:0.54.0
)
# ensure python <-> java interactions are w/ pyarrow
conf.set("spark.sql.execution.pyarrow.enabled", "true")

# Config to change the IO implementation of target catalog; write to object store
conf.set("spark.sql.catalog.nessie.io-impl","org.apache.iceberg.aws.s3.S3FileIO")

# create catalog named arctic as an iceberg catalog
conf.set("spark.sql.catalog.nessie", "org.apache.iceberg.spark.SparkCatalog")

# tell the catalog that its a Nessie catalog
conf.set("spark.sql.catalog.nessie.catalog-impl", "org.apache.iceberg.nessie.NessieCatalog")

# set the location for the catalog to store data. Spark writes to this directory
conf.set("spark.sql.catalog.nessie.warehouse", "s3a://dm-iceberg/nessie/")

# set the location of the Arctic/Nessie server.
conf.set("spark.sql.catalog.nessie.uri", "http://nessie:19120/api/v1")

# default branch for Arctic catalog to work on
conf.set("spark.sql.catalog.nessie.ref", "main")

# Authentication mechanism. Here, we use AWS with BEARER
conf.set("spark.sql.catalog.nessie.authentication.type", "NONE")


# enable the extensions for both Nessie and Iceberg
conf.set(
    "spark.sql.extensions",
    "org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions,org.projectnessie.spark.extensions.NessieSparkSe
)

# finally, start up the Spark server
spark = SparkSession.builder.config(conf=conf).getOrCreate()
print("Spark Running")
```

Iceberg-spark-runtime &
nessie-spark-extensions

Create a catalog ‘nessie’

Tell Spark it’s a Custom catalog

Catalog warehouse Location

Nessie Server URI

Spark SQL extensions: Nessie, Iceberg

Create a Local Branch

Git-branches anyone?

```
spark.sql("CREATE BRANCH dev IN nessie").toPandas()
```

	refType	name	hash
0	Branch	dev	70ba3d3af51237457da071c3875829795b8a98e713034d...

```
spark.sql("LIST REFERENCES IN nessie").toPandas()
```

	refType	name	hash
0	Branch	main	70ba3d3af51237457da071c3875829795b8a98e713034d...
1	Branch	work	70ba3d3af51237457da071c3875829795b8a98e713034d...
2	Branch	dev	70ba3d3af51237457da071c3875829795b8a98e713034d...

Create a Local Branch ‘dev’ to start with our analysis

List all Branches

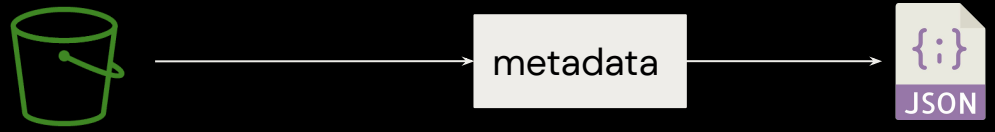
Create Iceberg Table

```
spark.sql("USE REFERENCE dev IN nessie")

spark.sql(
    """CREATE TABLE IF NOT EXISTS nessie.db1.sales
            (id STRING, name STRING, product STRING, price STRING, date STRING) USING iceberg"""
)
```

```
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
```

```
DataFrame[]
```

- CREATE statement, CTAS
- Create Partitions



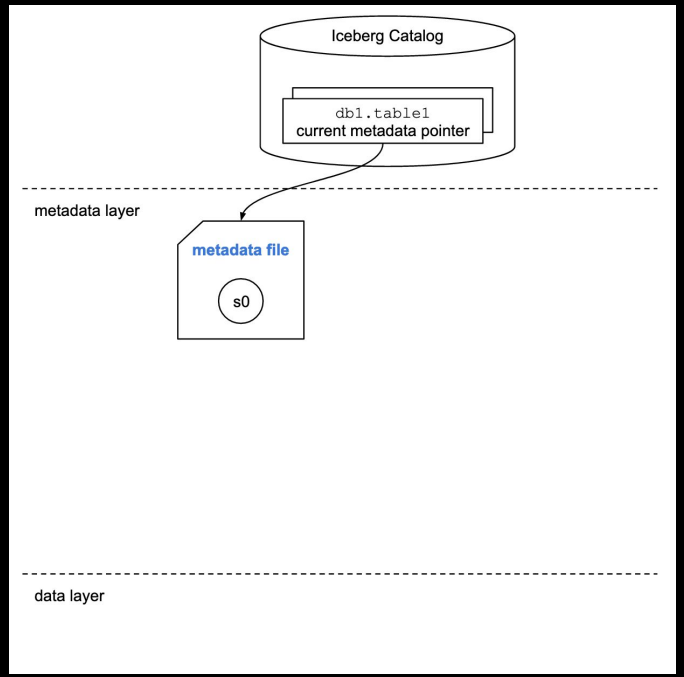
Warehouse Location:
`mybucket/nessie/db1/sales`

File: `metadata.json`

CREATE: Under the Hood

```
CREATE TABLE db1.table1 (
      order_id bigint,
      customer_id bigint,
      order_amount DECIMAL(10, 2),
      order_ts TIMESTAMP
)
USING iceberg
PARTITIONED BY ( hour(order_ts) );
```

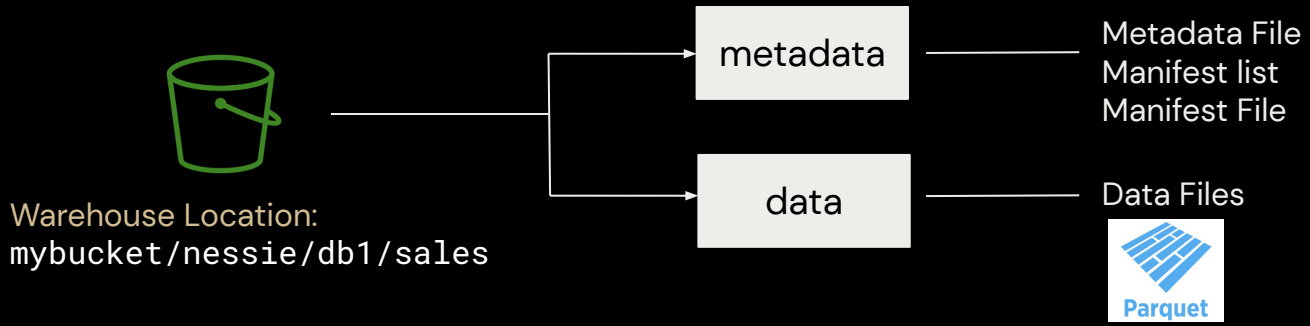

```
table1/
 |- metadata/
 |    |- v1.metadata.json
 |- data/
```



Insert Into Iceberg Table

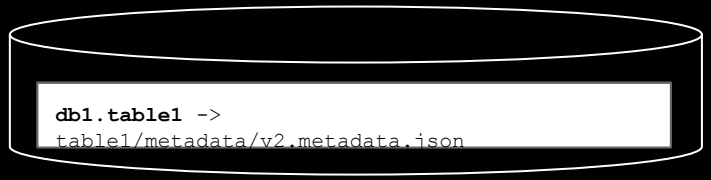
```
spark.sql("CREATE OR REPLACE TEMPORARY VIEW my_iceberg_table USING csv OPTIONS "
          "(PATH 's3a://dm-iceberg/datasets/salesdata.csv', HEADER 'true', INFERSCHEMA 'true')")

spark.sql("INSERT INTO nessie.db1.sales SELECT * FROM my_iceberg_table")
```

- Append new data
- INSERT, INSERT OVERWRITE

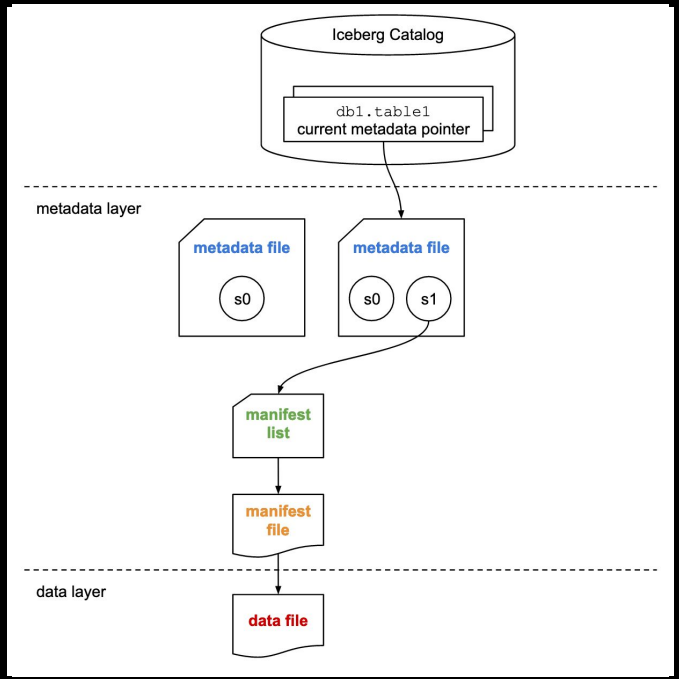


INSERT: Under the Hood

```
INSERT INTO db1.sales VALUES (
      123,
      456,
      36.17,
      '2021-01-26 08:10:23'
);
```



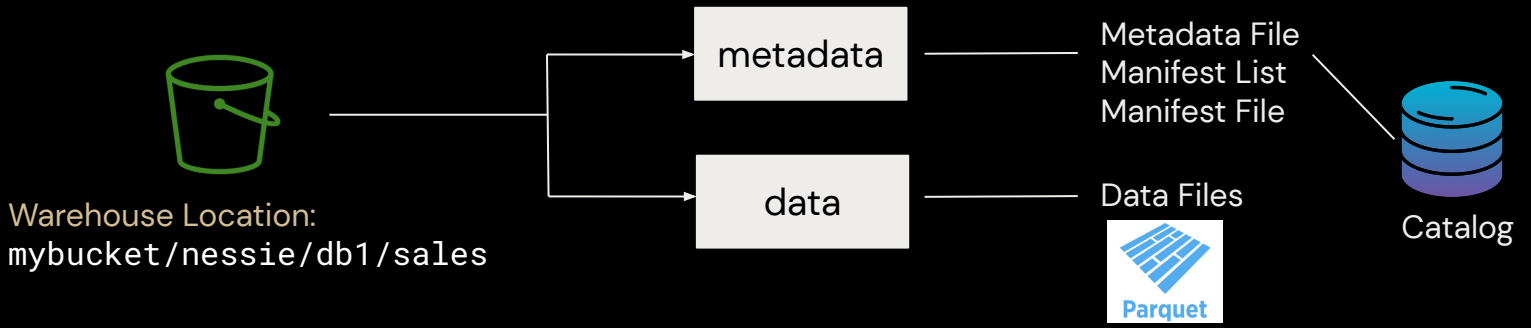

```
table1/
 |- metadata/
 |    |- v1.metadata.json
 |    |- v2.metadata.json
 |    |- snap-2938-1-4103.avro
 |    |- d8f9-ad19-4e.avro
 |- data/
 |    |- order_ts_hour=2021-01-26-08/
 |    |    |-  00000-5-cae2d.parquet
```



Read from Iceberg

```
spark.sql("SELECT * FROM nessie.db1.sales LIMIT 5").toPandas()
```

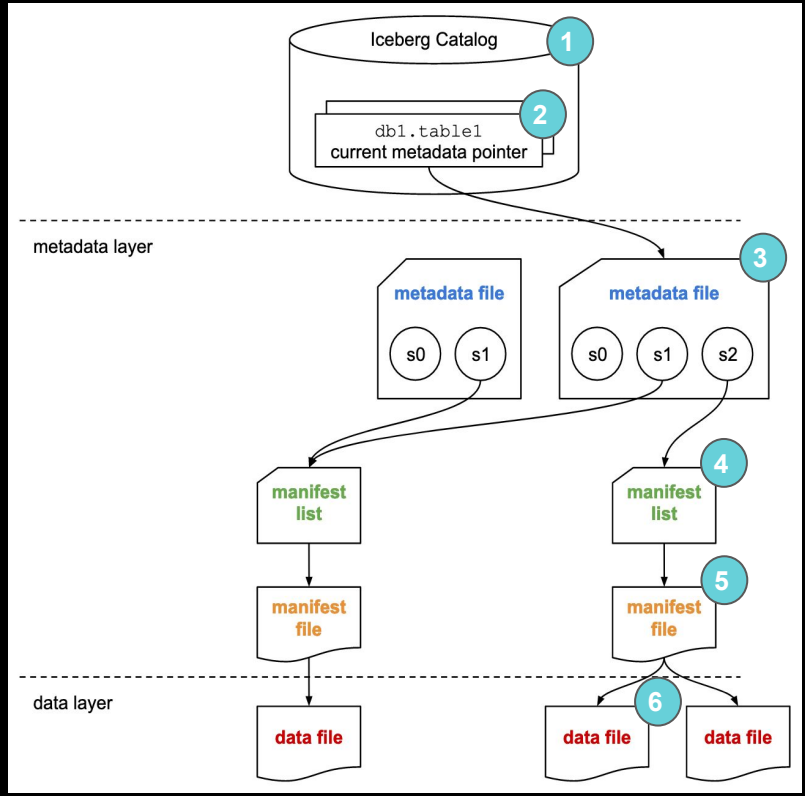
	id	name	product	price	date
0	1	Selinda Rheaume	Wine - Prosecco Valdobiaddene	10.31	10/20/2022
1	2	Wynnie Gozzard	Wine - Red, Wolf Blass, Yellow	20.05	10/20/2022
2	3	Patten Whitter	Crackers - Soda / Saltins	39.75	10/20/2022
3	4	Hulda Eslie	Roe - White Fish	37.10	10/20/2022
4	5	Chrystal Haggie	Wine - Delicato Merlot	35.82	10/20/2022



READ: Under the Hood

```
SELECT *
FROM db1.sales
WHERE order_ts = DATE '2021-01-26'
```


```
table1/
 |- metadata/
 |    |- v1.metadata.json
 |    |- v2.metadata.json
 |    3  v3.metadata.json
 |       snap-29c8-1-b103.avro
 |    4  snap-9fa1-3-16c3.avro
 |    |- d8f9-ad19-4e.avro
 |    5  0d9a-98fa-77.avro
 |- data/
 |    |- order_ts_hour=2021-01-26-08/
 |    |    |- 00000-5-cae2d.parquet
 |    |    6  00000-1-aef71.parquet
 |    |- order_ts_hour=2021-01-27-10/
 |    |    |- 00000-3-0fa3a.parquet
```



Update Iceberg Tables

```
spark.sql("SELECT * FROM nessie.db1.sales WHERE id = 4").toPandas()
```

	id	name	product	price	date
0	4	Hulda Eslie	Roe - White Fish	37.10	10/20/2022

```
spark.sql("UPDATE nessie.db1.sales SET price = 100 WHERE id = 4").toPandas()
```

—

```
spark.sql("SELECT * FROM nessie.db1.sales WHERE id = 4").toPandas()
```

	id	name	product	price	date
0	4	Hulda Eslie	Roe - White Fish	100	10/20/2022

2 Strategies for row-level updates:

- Copy-on-Write (COW): rewrite reqd
- Merge-on-Read (MOR): no rewrites

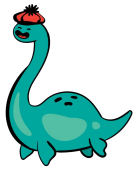
Wait, which Branch did we WRITE to?

```
spark.sql("USE REFERENCE main IN nessie").toPandas()
```

	refType	name	hash
0	Branch	main	70ba3d3af51237457da071c3875829795b8a98e713034d...

```
spark.sql("SHOW TABLES IN nessie").show()
```

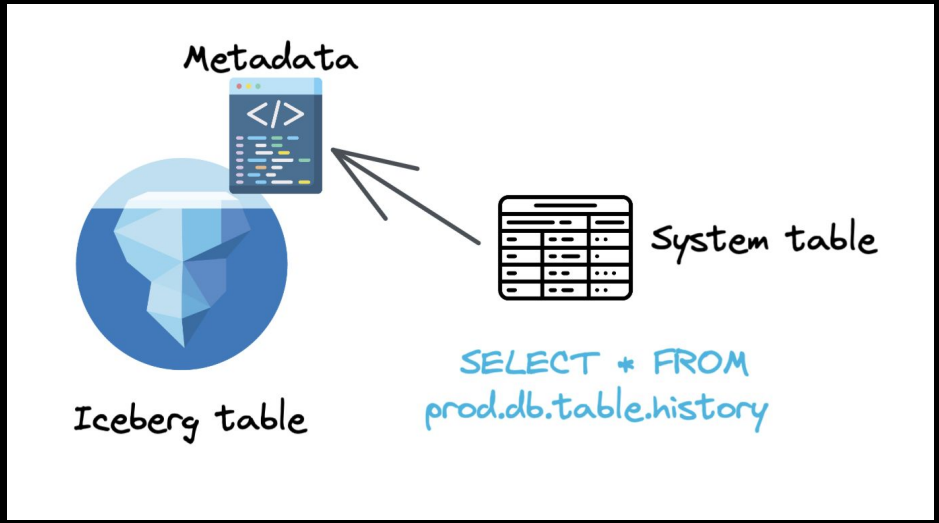
```
+---------+---------+-----------+
|namespace|tableName|isTemporary|
+---------+---------+-----------+
|      db1|  details|      false|
|      db1| personal|      false|
|      db1| nominees|      false|
|      db1|  contact|      false|
|      db1|     info|      false|
|      db1|      new|      false|
|      db1| employee|      false|
+---------+---------+-----------+
```



Switch to ‘main’ branch

See the Tables

Metadata Tables



- Metadata is core to Iceberg/ other table formats
- Easy access to metadata via system tables
- Data files, History, Manifests & Snapshots

Metadata Tables

```
spark.sql("SELECT * FROM nessie.db1.sales.history").show()
```

```
+--------------------+-------------------+-------------------+-------------------+
|     made_current_at|        snapshot_id|          parent_id|is_current_ancestor|
+--------------------+-------------------+-------------------+-------------------+
|2023-05-10 19:17:...|3881904599646321029|               null|               true|
|2023-05-10 20:08:...|1123401274700139758|3881904599646321029|               true|
+--------------------+-------------------+-------------------+-------------------+
```

```
spark.sql("SELECT * FROM nessie.db1.sales.snapshots").show()
```

```
+--------------------+-------------------+-------------------+---------+--------------------+--------------------+
|        committed_at|        snapshot_id|          parent_id|operation|       manifest_list|             summary|
+--------------------+-------------------+-------------------+---------+--------------------+--------------------+
|2023-05-10 19:17:...|3881904599646321029|               null|   append|s3a://dm-iceberg/...|{spark.app.id -> ...|
|2023-05-10 20:08:...|1123401274700139758|3881904599646321029|overwrite|s3a://dm-iceberg/...|{spark.app.id -> ...|
+--------------------+-------------------+-------------------+---------+--------------------+--------------------+
```

```
spark.sql("SELECT * FROM nessie.db1.sales.files").toPandas()
```

	content	file_path	file_format	spec_id	record_count	file_size_in_bytes	column_sizes	value_counts	null_value_counts	nan_value_co
0	0	s3a://dm-iceberg/nessie//db1/sales_f5d94466-ad...	PARQUET	0	500	14775	{1: 949, 2: 4912, 3: 5979, 4: 1584, 5: 101}	{1: 500, 2: 500, 3: 500, 4: 500, 5: 500}	{1: 0, 2: 0, 3: 0, 4: 0, 5: 0}	

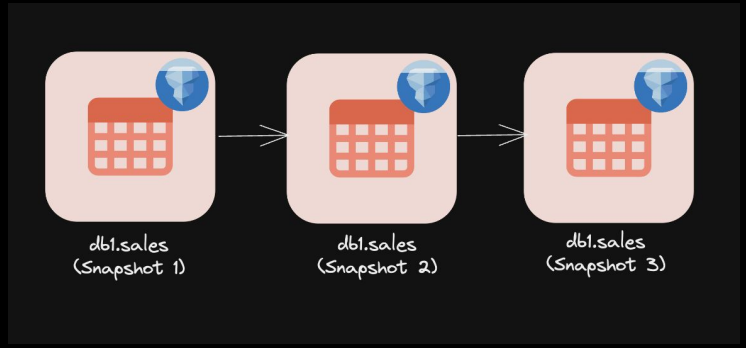
Time Travel in Iceberg

```
spark.sql("SELECT * FROM nessie.db1.sales VERSION AS OF '3881904599646321029' WHERE ID=4 ").toPandas()
```

	id	name	product	price	date
0	4	Hulda Eslie	Roe - White Fish	37.10	10/20/2022

```
spark.sql("SELECT * FROM nessie.db1.sales VERSION AS OF '1123401274700139758' WHERE ID=4").toPandas()
```

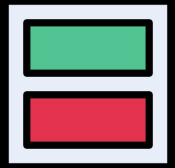
	id	name	product	price	date
0	4	Hulda Eslie	Roe - White Fish	100	10/20/2022



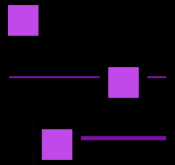
- Iceberg maintains 'snapshots'
- Snapshot: State of a table at a certain time
- Use Snapshots/Timestamp to time travel

Optimization methods

Compaction

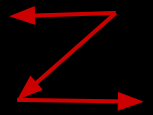


Hidden Partitioning



Min/Max Filtering

Sorting



Z-ordering

Spark Procedures

Table Maintenance

- Expire Snapshots
- Remove Metadata files
- Compact Data files
- Delete Orphan files
- Rewrite Manifests

What's with the Branches?

```
spark.sql("MERGE BRANCH dev INTO main IN nessie").toPandas()
```

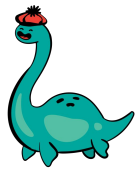
	name	hash
0	main	d7daae2b32908e3c7acd277ce6846d6b32f5cfd3e34649...

```
spark.sql("USE REFERENCE main IN nessie").show()
spark.sql("SHOW TABLES IN nessie").show()
```

```
+-------+----+--------------------+
|refType|name|                hash|
+-------+----+--------------------+
| Branch|main|d7daae2b32908e3c7...|
+-------+----+--------------------+
```

```
[Stage 8:>                                                  (0 + 4) / 4]
```

```
+---------+---------+-----------+
|namespace|tableName|isTemporary|
+---------+---------+-----------+
|      db1|    sales|      false|
|      db1|  details|      false|
|      db1| personal|      false|
|      db1| nominees|      false|
|      db1|  contact|      false|
|      db1|     info|      false|
|      db1|      new|      false|
|      db1| employee|      false|
+---------+---------+-----------+
```



MERGE the 'dev' branch to 'main'

Sales table is in 'main' now

Or DROP it

Thank you

Dipankar Mazumdar

DipankarTnt