Apache Iceberg's Best Secret

A Guide to Metadata Tables

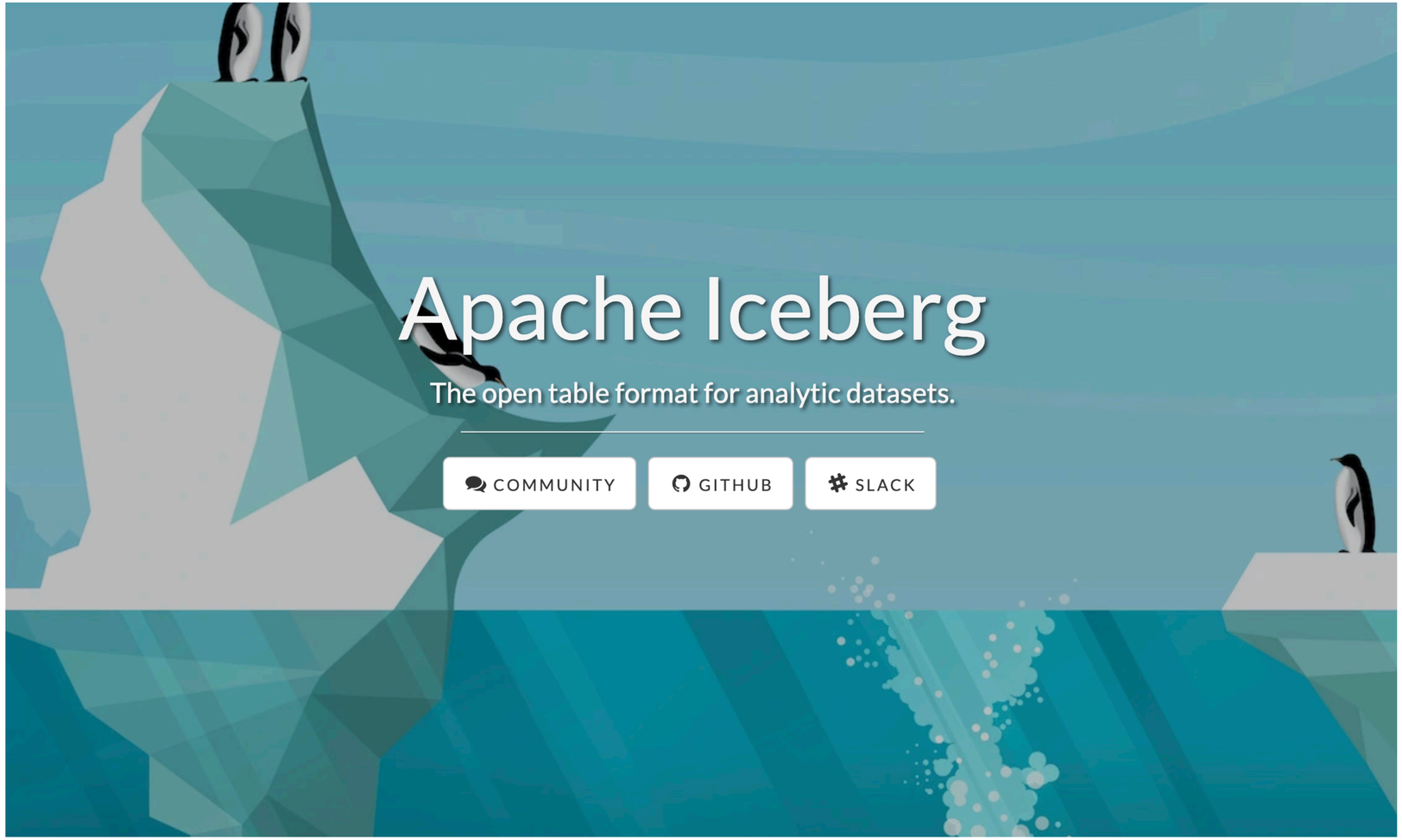
Szehon Ho, October 4 2022

Apache Iceberg Project

- Developed to address Hive shortcomings
- Apache Incubator 2018-2020
- 295 contributors from many companies
- Collaboration with Spark/Flink/Trino communities
- Wide adoption in 2022

What is Apache Iceberg?

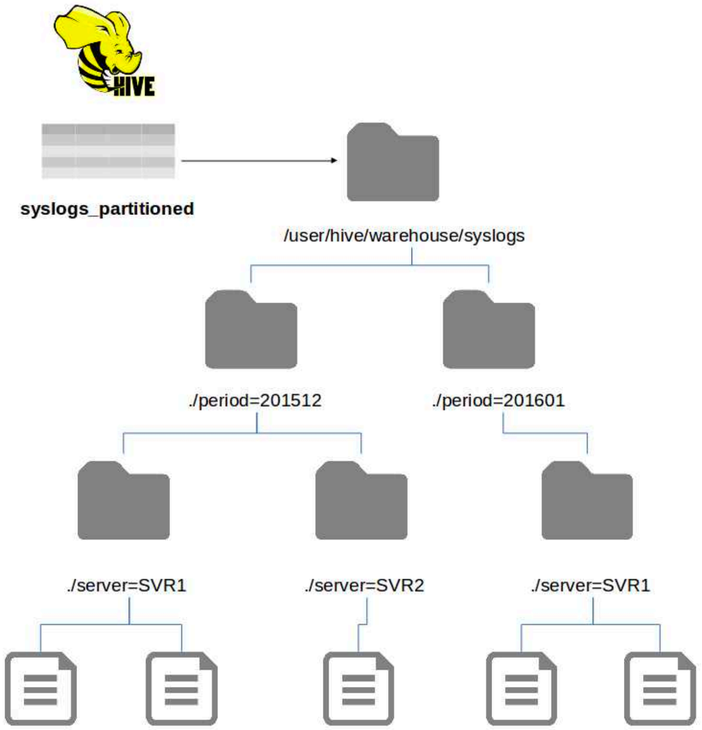
In its own Words



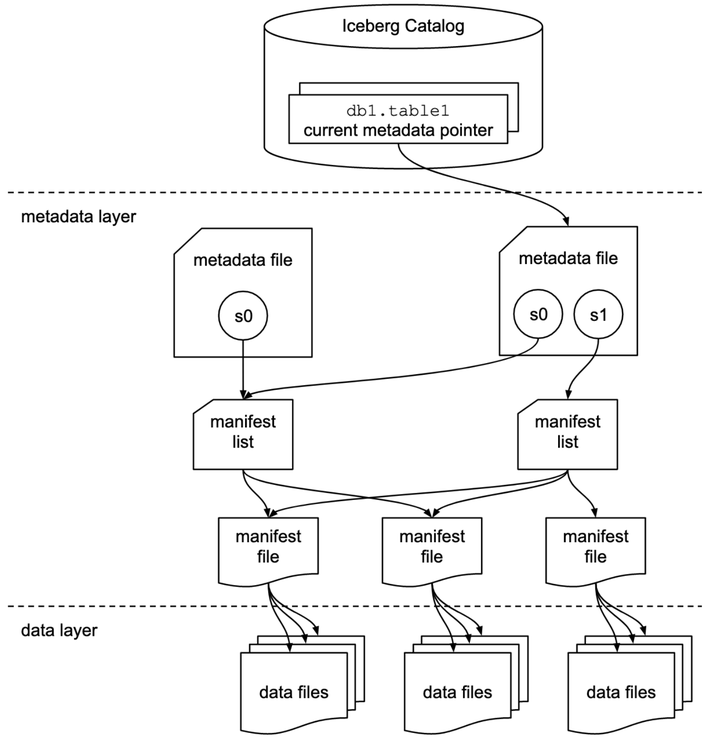
What is Apache Iceberg?

“Table Format” = Layout of Files in Table

- Hive: Directory contains all files in tables and partitions

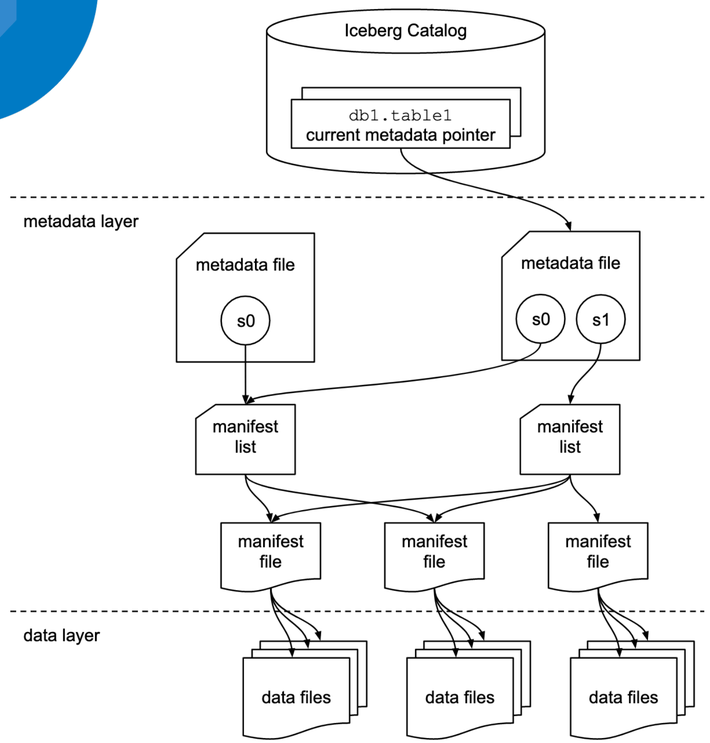


- Iceberg: Follow a tree of “Metadata Files” that track data of Tables and Partitions



Metadata Files

Unlocking many new features: only some shown here



Category	Hive Behavior	Iceberg Metadata Feature
Atomicity on Object Store (S3)	Inconsistent Listing Non-Atomic	Data File Listings in metadata file
Time Travel/ Rollback	Not supported	Snapshot File
Isolation Level	Need Explicit Directory Lock	Snapshot Info on each Data File, Check only conflicts
Performance (Predicate Pruning)	Partition (Directory) level filter only	1. Partition stats at multiple layers 2. Min/Max Column Stats

“Open” Table Format

- Metadata Files are the basis for all of Iceberg’s advance feature-set
- Metadata Tables: Exposes all Metadata Files in user-friendly way
  - Interface: Exposed as SQL as system tables
  - Performance: Queries are much faster than data queries
- Full Transparency: Users/Systems can easily self-explore Metadata Tables to know how the system works, and how to improve it
  - Most tough problems can be debugged (at least partially) by Iceberg metadata tables
  - Decide how to optimize the table pre-emptively
  - Build monitoring, auditing, data quality checks beyond Iceberg

My First Metadata Table

Partitions Table

```
spark-sql> show partitions iceberg.default.sales;
Error in query: Table iceberg.default.sales does not support partition management.;
ShowPartitions [partition#0]
+- ResolvedTable org.apache.iceberg.spark.SparkCatalog@4536a09a, default.sales, iceberg.default.sales, [data#1, day#2, hour#3]
```

Partition table = “db.table.partitions”

```
spark-sql> select * from iceberg.default.sales.partitions order by partition.day, partition.hour;
partition       record_count    file_count
{"day":"2022-10-04","hour":0}   1       1
{"day":"2022-10-04","hour":1}   1       1
{"day":"2022-10-04","hour":2}   1       1
{"day":"2022-10-04","hour":3}   1       1
{"day":"2022-10-04","hour":4}   1       1
{"day":"2022-10-04","hour":5}   1       1
{"day":"2022-10-04","hour":6}   1       1
{"day":"2022-10-04","hour":7}   2       2
{"day":"2022-10-04","hour":8}   1       1
{"day":"2022-10-04","hour":9}   1       1
{"day":"2022-10-04","hour":10}  1       1
{"day":"2022-10-04","hour":11}  1       1
{"day":"2022-10-04","hour":12}  1       1
{"day":"2022-10-04","hour":13}  1       1
{"day":"2022-10-04","hour":14}  1       1
```

Metadata Tables

The Full List

Partitions is just an aggregate view of files table

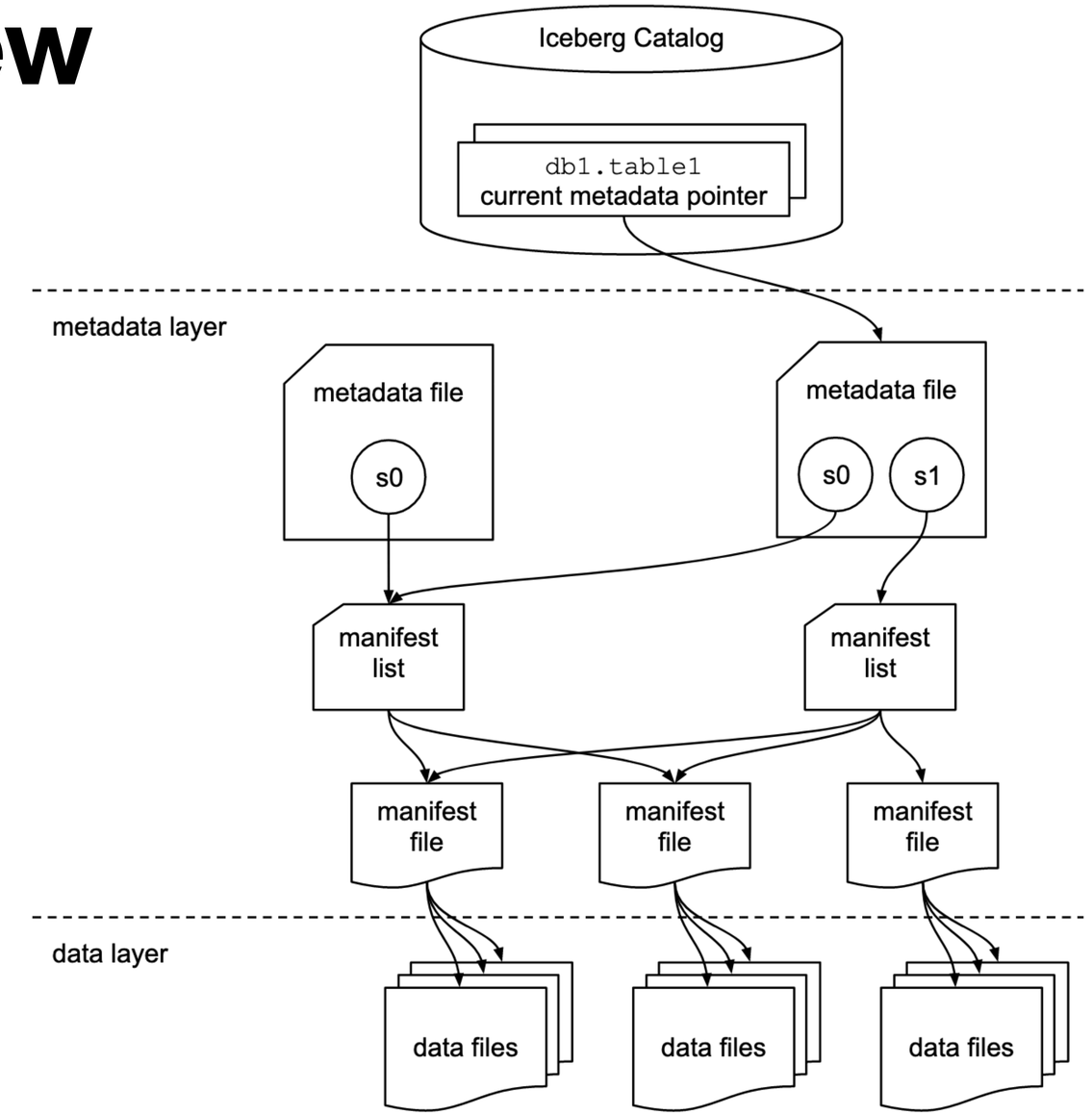
Iceberg Metadata Tables:

- history
- metadata_logs
- snapshots
- manifests
- all_manifests
- entries
- all_entries
- files
- data_files
- delete_files
- all_files
- all_data_files
- all_delete_files

Metadata Files Review

Hierarchical Structure

- Catalog (atomic pointer to Root Metadata)
- Metadata File (Root Metadata)
- Snapshot Files (Manifest List)
- Manifest Files
- Data Files

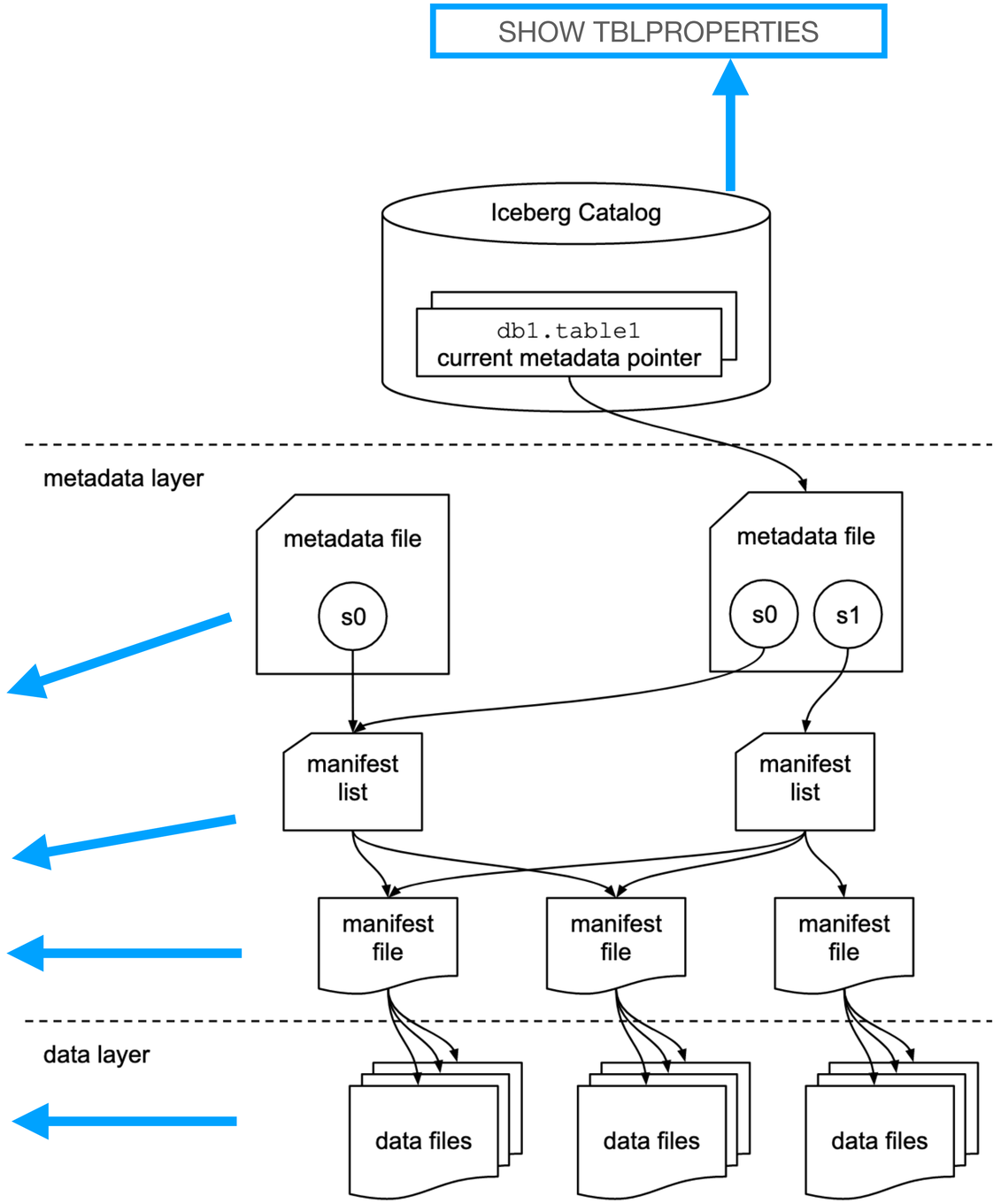


Metadata Tables

Mapping to Metadata Files

- Each Metadata Table has information about all or a subset of one layer of “Metadata File”
- Table for a Metadata File doesn’t read that layer metadata file, rather from the layer above it

Metadata Table	Queries	About
metadata_logs	Last Metadata File	Metadata File
snapshots	Last Metadata File	Snaphot Files (Manifest Lists)
manifests	Snapshot Files (Manifest Lists)	Manifests
Files/Entries (see next slide)	Manifests	Data Files



Files/Entries Tables

Various Views of “Data Files” for User Convenience

- Partitions table is just an aggregate view of Files table
- Files/Entries: Equivalent. Manifest File Entry = metadata about a data file
  - Files = “Files” part of Manifest Entry, only physical attributes of a file
  - Entries = Complete row, including snapshot information of the file
- All_ tables: All_Manifests, All_Files, All_Entries
  - all_x = All Metadata Files of X Layer
  - x = Metadata Files of X layer that are pointed to by current snapshot
- Data/Delete: Data_Files, Delete_Files
  - Delete Files a V2 concept for Merge-on-Read
  - “Files” table selects both types of files

FAQ: Partition Information

- How many files per partition?

```
SELECT partition, file_count
FROM db.table.partitions
```

partition	file_count
{"date":"2022-10-04","hour":5}	5

- Total size of each partition?

```
SELECT partition,
sum(file_size_in_bytes) AS partition_size,
FROM db.table.files
GROUP BY partition
```

partition	partition_size
{"date":"2022-10-04","hour":5}	937

- Last update time per partition?

```
SELECT
e.data_file.partition,
MAX(s.committed_at) AS last_modified_time
FROM db.table.snapshots s
JOIN db.table.entries e
WHERE s.snapshot_id = e.snapshot_id
GROUP_BY by e.data_file.partition
```

partition	last_modified_time
{“date":"2022-10-04", "hour":5}	2022-09-07 01:30:52.371

Closer Look at Snapshots

Two Meanings vis-a-vis Files

- Snapshot points to a list of files belonging to table at point in time
- Snapshot is also an operation on files (adding, removing)
- Entries table tracks which snapshot operated on the file
  - entries.snapshot_id
  - entries.status : 0=EXISTING, aka rewrite, 1= ADDED, 2 =DELETED

FAQ: Snapshot Questions

- What files are added by snapshot 8339536322928208593?

```
SELECT data_file.file_path
FROM db.table.entries
WHERE snapshot_id=8339536322928208593
AND status=1;
```

- What files are referenced by snapshot 8339536322928208593?
  - Use time-travel (SQL Syntax)

```
SELECT file_path
FROM db.table.files
VERSION AS OF 8339536322928208593;
```

FAQs: How to Keep Iceberg Maintained

- Expire Snapshots (Cleanup)
- RewriteManifests (Metadata Files Optimization)
- RewriteFiles (Data Files Optimization)

FAQ: Disk Usage and Expire Snapshots

- User Question: I am hitting HDFS quotas. I ran compact files/deleted partitions, why do I still see quota limit?
- Answer: Expire snapshots
- Metadata Tables:
  - all_manifests, all_files will show you everything reachable even from previous snapshots
  - manifests, files will show everything reachable from current snapshot
- Useful Queries for Dashboards:

```
select sum(file_size_in_bytes) from db.table.all_files;  // all reachable data files size

select sum(length) from db.table.all_manifests;  //all reachable manifest files size

select sum(file_size_in_bytes) from db.table.files;  // current snapshot files size

select sum(length) from db.table.manifests;  // current snapshot manifest files size
```

FAQ: Disk Usage

Snapshots Table Alternative

```
SELECT committed_at, snapshot_id, summary FROM db.table.snapshots;
```

Committed_at	snapshot_id	Summary
2022-08-24 14:01:43.191	4077543616265127980	{“added-data-files":"1", “added-files-size":"904", “added-records":"1", “changed-partition-count":"1", "spark.app.id":"local-1661374186213", “total-data-files":"23", “total-delete-files":"0", “total-equality-deletes":"0", “total-files-size":"20792", “total-position-deletes":"0", "total-records":"23"}

FAQ: When to Optimize Metadata

- Improve query planning time, metadata table query time, by reducing overhead of reading metadata-files

```
// How many manifests?
SELECT count(*)
FROM db.table.manifests;
```

count(1)
200

```
// Which manifests?
SELECT path,
added_data_files_count +
existing_data_files_count +
deleted_data_files_count as files
FROM db.table.manifests;
```

path	files
s3://my_bucket/db/table/…	2
s3://my_bucket/db/table/…	4

```
// Are manifests sorted?
SELECT path, partition_summaries
FROM db.table.manifests;
```

path	partition_summaries
s3://my_bucket/db/table/…	{“lower_bound”:”2022-10-04”, "upper_bound":"2022-10-04"}

FAQ: When to Optimize Data

- Improve query time by minimizing file-read overhead
- Sort to improve selectivity of files, and compression ratio of files

```
// Too many small data files?
SELECT partition, count(*) as file_count,
sum(file_size_in_bytes)/count(*) as avg_size,
FROM db.table.files
GROUP BY partition
```

partition	file_count	avg_size
{"date":"2022-10-04","hour":5}	100	5120000

```
// Are data files sorted?
// Note: Column coming soon
SELECT file_path,
readable_metrics.emp.upper_bound,
readable_metrics.emp.lower_bound,
FROM db.table.files;
```

file_path	col.lower_bound	col.upper_bound
s3://my_bucket/db/table/…	Abigail Adams	Mike Monroe
s3://my_bucket/db/table/…	Nancy Nomura	Zachary Zunich

Beyond Iceberg

Use Case: Ingest Monitoring

- Measuring a system data completeness and latency is typically hard, but becomes do-able in Iceberg
- Incoming Dataset from Flink:
  - (data string, event_time timestamp) partitioned by hour (event_time)

```
// Data Completeness
SELECT record_count AS received, partition
FROM db.table.partitions;
```

```
// Data Latency with custom UDF for calcuating time difference.
// Will be easier with readable_metrics column
SELECT max(diff(entries.data_file.lower_bounds[1], hour(snapshots.committed_at)) AS max_latency
FROM db.table.entries JOIN db.table.snapshots
ON entries.snapshot_id = snapshots.snapshot_id
GROUP BY entries.data_file.partition;
```

Beyond Iceberg

Use Case: Data Quality Alerts

- Iceberg keeps interesting metrics per data file of every column:
  - column_sizes
  - value_counts
  - null_values
  - nan_values
  - lower_bounds
  - upper_bounds

- Can create alerts for partitions with nan_values

```
Select partition, (sum(to_int(files.nan_values[0])) AS nan_values
FROM db.table.files
GROUP BY files.partition
```

Future

Stay Tuned for Puffin Files

- Puffin Files introduced into Iceberg spec
- https://github.com/apache/iceberg/blob/master/format/puffin-spec.md
- For (TBD)
  - Bloom Filters
  - Datasketches
- Apply to data file or set of data files (TBD)
- Can be used for data quality percentiles

Questions?

Thank you for attending!