

APACHE ICEBERG



CONCEPTS AND EXPLANATIONS



Shwetank Singh
GritSetGrow – GSGLearn.com

*follow for
more content* >>

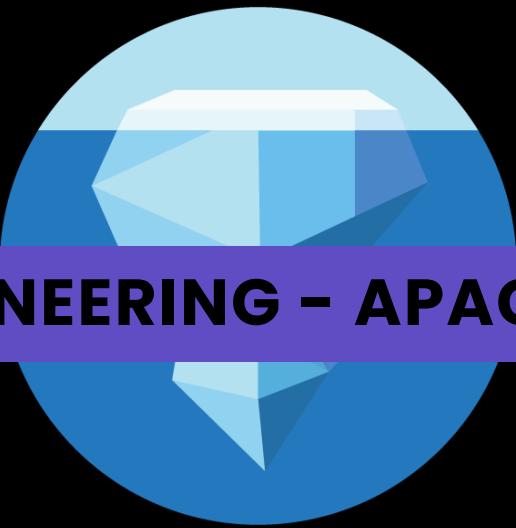


Shwetank Singh
GritSetGrow - GSGLearn.com

DATA LAKEHOUSE

A data architecture pattern that combines the flexibility and scalability of data lakes with the data management and performance capabilities of data warehouses.

Integrating Apache Iceberg with a data lake to enable ACID transactions and schema evolution while maintaining the low-cost storage of a data lake.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

APACHE ICEBERG

An open table format designed for large analytic datasets, providing features like schema evolution, partitioning, and time travel queries.

Using Apache Iceberg to manage petabyte-scale data stored in Amazon S3, enabling efficient queries and data management.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

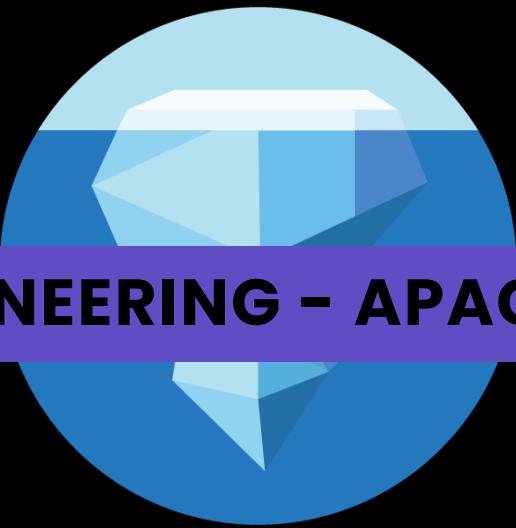


Shwetank Singh
GritSetGrow - GSGLearn.com

ICEBERG ARCHITECTURE

The structural design of Apache Iceberg, including data files, metadata files, manifest files, and the catalog. These components work together to enable efficient data management and querying.

Iceberg's architecture separates metadata from data, allowing for efficient query planning and execution.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

DATA FILES

The actual files where the data is stored, typically in columnar formats like Parquet or ORC.

A Parquet file storing sales data in an Iceberg table.

DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

METADATA FILES

Files that store metadata information such as schema, partitioning, and snapshots. These files help manage and query the data efficiently.

An Iceberg metadata file might include the schema definition and partitioning information for a table.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

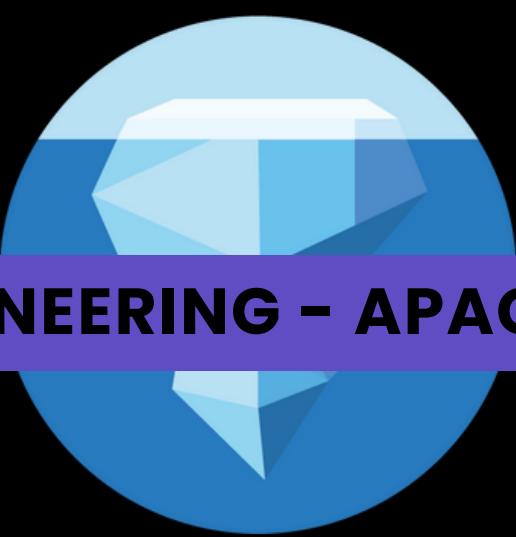


Shwetank Singh
GritSetGrow - GSGLearn.com

MANIFEST FILES

Metadata files that list data files, their statistics, and partition information. They are used to manage data efficiently and support operations like time travel and rollback.

A manifest file in Iceberg listing all the Parquet files and their row counts for a table.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

CATALOG

A service that tracks the metadata of tables, such as schemas, versions, and locations, allowing users to find and interact with tables.

Using the Hive Metastore as the catalog for Iceberg tables, enabling integration with existing Hive infrastructure.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content*



Shwetank Singh
GritSetGrow - GSGLearn.com

HOW ICEBERG WORKS

Iceberg manages large datasets by using a combination of metadata, manifest files, and a catalog to provide efficient data management and querying. It supports features like ACID transactions, schema evolution, and partitioning.

Creating an Iceberg table and performing queries using Apache Spark.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

EFFICIENCY OF ICEBERG

Iceberg is efficient due to its use of metadata and manifest files, which allow for optimized query planning and execution. It reduces the need for expensive operations like full table scans by using partitioning and data skipping techniques. Using Iceberg's time travel feature to query historical data without scanning the entire table.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

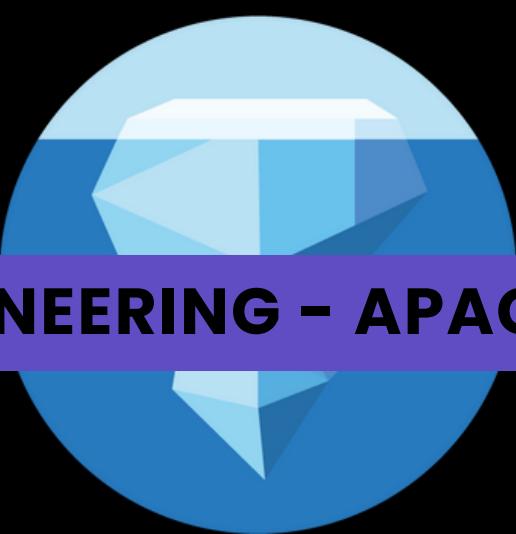


Shwetank Singh
GritSetGrow - GSGLearn.com

SCHEMA EVOLUTION

The ability to change the schema of a table (e.g., add or remove columns) without affecting existing data.

```
ALTER TABLE my_table ADD COLUMN  
new_column STRING;
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

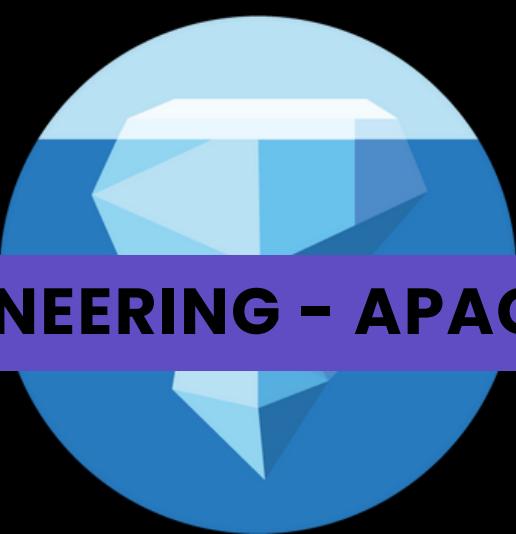


Shwetank Singh
GritSetGrow - GSGLearn.com

PARTITIONING

Dividing a table into smaller, more manageable pieces based on a specified column or set of columns.

```
CREATE TABLE sales (id INT, amount  
DOUBLE, date STRING) PARTITIONED BY  
(date);
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

TIME TRAVEL QUERIES

Queries that allow you to access data as it existed at a specific point in time.

```
SELECT * FROM my_table  
FOR SYSTEM_TIME AS OF '2022-01-01  
00:00:00';
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

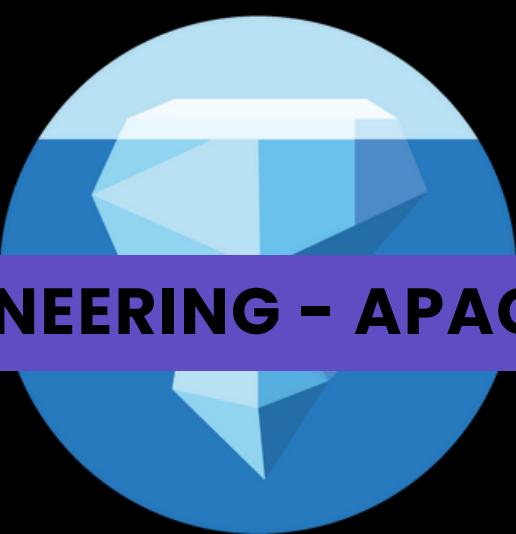


Shwetank Singh
GritSetGrow - GSGLearn.com

ACID TRANSACTIONS

A set of properties that ensure reliable processing of database transactions:
Atomicity, Consistency, Isolation,
Durability.

Ensuring that a batch update operation on an Iceberg table is completed fully and consistently, even in the case of failures.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>>



Shwetank Singh
GritSetGrow - GSGLearn.com

METADATA LAYER

A layer that manages metadata about the data files, such as schema, partitioning, and file locations, enabling efficient query planning and execution.

Apache Iceberg's metadata layer includes manifest files that list all data files and their statistics.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

MANIFEST LISTS

Files that contain lists of manifest files, providing a way to efficiently manage large numbers of data files.

A manifest list in Iceberg might include references to several manifest files for a large table.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

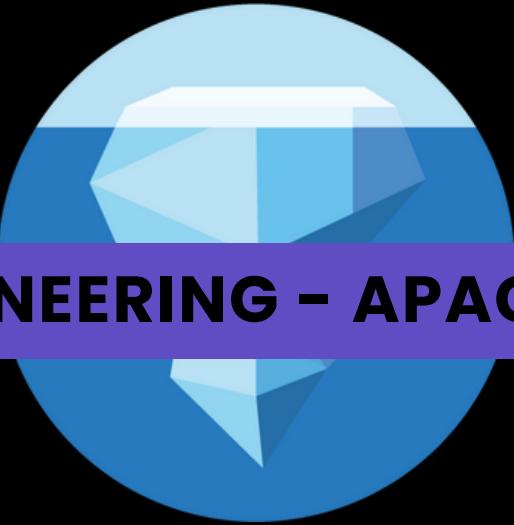


Shwetank Singh
GritSetGrow - GSGLearn.com

SNAPSHOTS

Point-in-time representations of a table's state, allowing for time travel queries and rollback operations.

```
SELECT * FROM my_table.snapshots;
```



DATA ENGINEERING – APACHE ICEBERG



follow for
more content ➤



Shwetank Singh
GritSetGrow - GSGLearn.com

MERGE-ON-READ

A data writing strategy where updates are written to delta files, which are merged with base files at read time.

```
MERGE INTO target_table USING source_table
ON target_table.id = source_table.id WHEN
MATCHED
THEN UPDATE
SET target_table.value = source_table.value
WHEN NOT MATCHED
THEN INSERT (id, value) VALUES
(source_table.id, source_table.value);
```



DATA ENGINEERING – APACHE ICEBERG



follow for
more content ➤

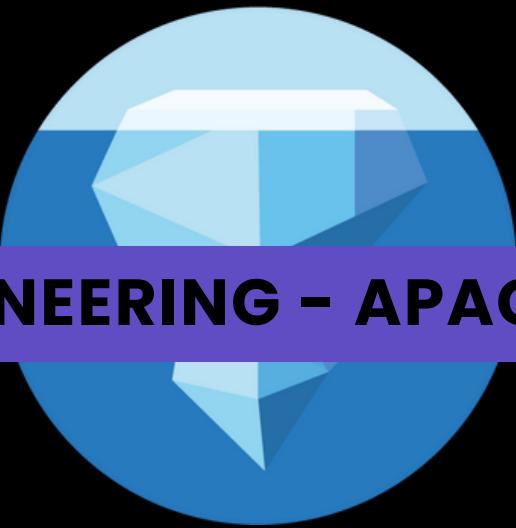


Shwetank Singh
GritSetGrow - GSGLearn.com

COPY-ON-WRITE

A data writing strategy where updates are written by creating new versions of data files, replacing the old files.

Using copy-on-write to handle batch updates to a fact table in Iceberg, ensuring consistent and optimized query performance.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>>



Shwetank Singh
GritSetGrow - GSGLearn.com

COMPACTON

The process of combining smaller data files into larger ones to improve read performance.

CALL

```
iceberg.system.compact_table('my_table');
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

Z-ORDERING

A data sorting technique that improves query performance by clustering related data together.

```
ALTER TABLE my_table ORDER BY ZORDER  
(column1, column2);
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

HIDDEN PARTITIONING

A technique where partitions are defined internally by the system rather than explicitly by the user, improving flexibility and performance.

Iceberg's hidden partitioning automatically manages partitions based on data characteristics, simplifying schema management.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

DATA SKIPPING

A query optimization technique that avoids reading irrelevant data based on metadata information.

Iceberg uses data skipping to avoid reading files that do not match query predicates, reducing I/O and improving query performance.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

OBJECT STORAGE

Storage systems designed to handle large amounts of unstructured data, often used in cloud environments.

Using Amazon S3 as the underlying storage for Iceberg tables to leverage scalable and cost-effective data storage.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

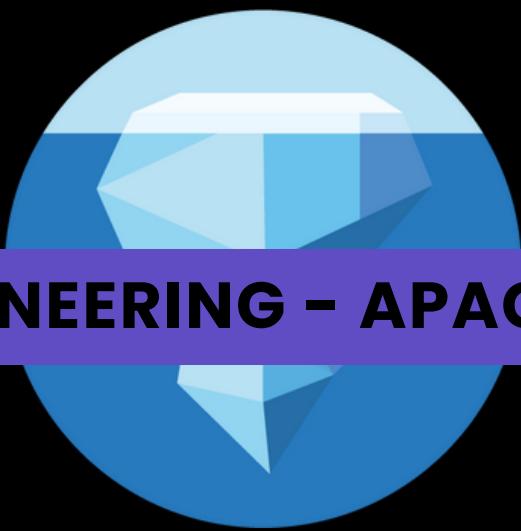


Shwetank Singh
GritSetGrow - GSGLearn.com

BLOOM FILTERS

A space-efficient probabilistic data structure used to test whether an element is present in a set. It helps to optimize data skipping by quickly identifying relevant data files.

```
CREATE TABLE my_table USING iceberg  
OPTIONS (bloom_filter_columns = 'id');
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

DATAFILE BLOOM FILTERS

Bloom filters applied to data files in Iceberg to quickly identify whether a file contains relevant data for a query.

A Bloom filter in an Iceberg data file helps skip irrelevant files during a query, reducing I/O.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

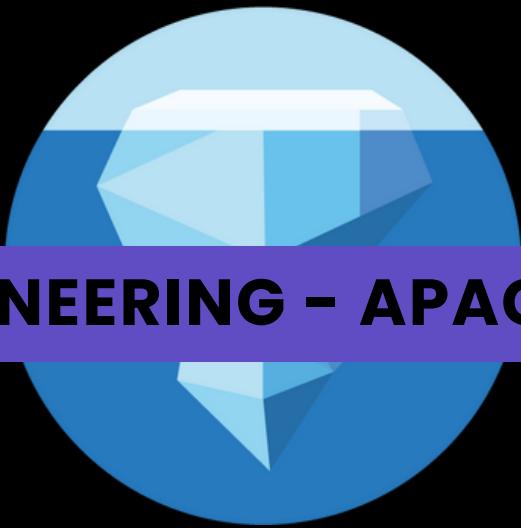


Shwetank Singh
GritSetGrow - GSGLearn.com

HADOOP CATALOG

An Iceberg catalog implementation that stores metadata in HDFS.

```
CREATE TABLE my_table USING iceberg  
OPTIONS (catalog = 'hadoop', location =  
'hdfs:///path/to/table');
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

HIVE CATALOG

An Iceberg catalog implementation that stores metadata in Hive Metastore.

```
CREATE TABLE my_table USING iceberg
OPTIONS (catalog = 'hive', location =
'hdfs:///path/to/table');
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

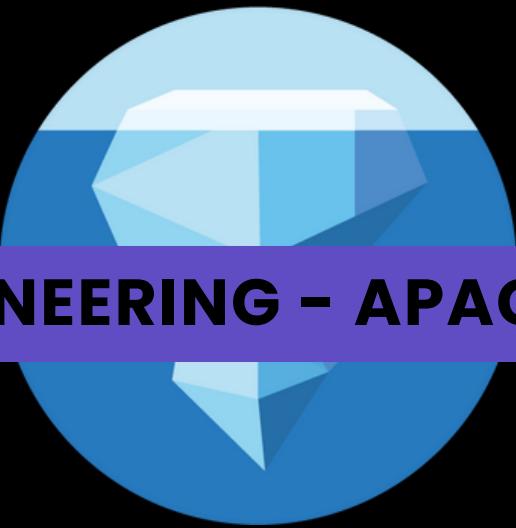


Shwetank Singh
GritSetGrow - GSGLearn.com

AWS GLUE CATALOG

An Iceberg catalog implementation that integrates with AWS Glue for metadata management.

```
CREATE TABLE my_table USING iceberg
OPTIONS (catalog = 'glue', location =
's3://bucket/path/to/table');
```



DATA ENGINEERING – APACHE ICEBERG



follow for
more content ➤



Shwetank Singh
GritSetGrow - GSGLearn.com

NESSIE CATALOG

An Iceberg catalog implementation that supports versioning and branching of table metadata.

```
CREATE TABLE my_table USING iceberg
OPTIONS (catalog = 'nessie', branch =
'main');
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

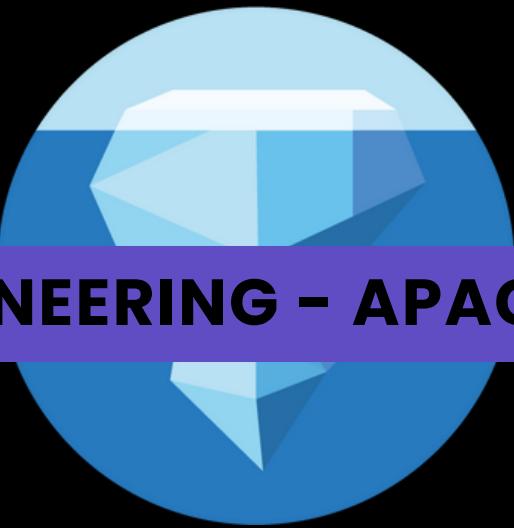


Shwetank Singh
GritSetGrow - GSGLearn.com

REST CATALOG

An Iceberg catalog implementation that uses a RESTful service for metadata management.

Using REST Catalog to manage Iceberg tables with a custom metadata service.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

JDBC CATALOG

An Iceberg catalog implementation that uses JDBC-compatible databases for metadata management.

Using JDBC Catalog to store Iceberg table metadata in a PostgreSQL database.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

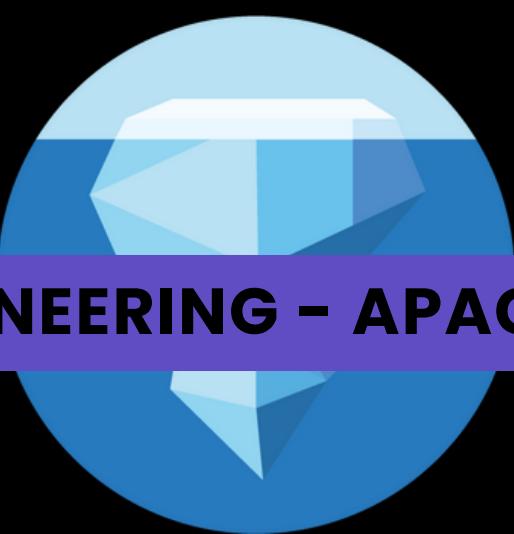


Shwetank Singh
GritSetGrow - GSGLearn.com

CREATE TABLE

The operation to define a new table in Iceberg, specifying its schema and partitioning strategy.

```
CREATE TABLE my_table (id INT, name  
STRING) PARTITIONED BY (name);
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

INSERT DATA

The operation to add new data to an Iceberg table.

```
INSERT INTO my_table  
VALUES (1, 'Alice'), (2, 'Bob');
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

UPDATE DATA

The operation to modify existing data in an Iceberg table.

```
UPDATE my_table  
SET name = 'Charlie' WHERE id = 1;
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>>

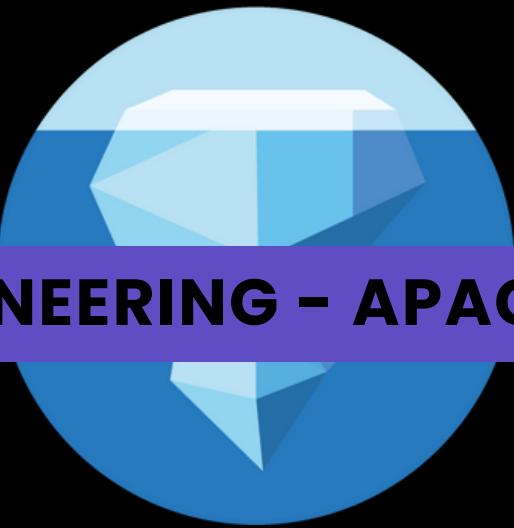


Shwetank Singh
GritSetGrow - GSGLearn.com

DELETE DATA

The operation to remove data from an Iceberg table.

```
DELETE FROM my_table WHERE id = 2;
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

TIME TRAVEL

The ability to query data as it existed at a specific point in time.

```
SELECT * FROM my_table  
FOR SYSTEM_TIME AS OF '2023-01-01';
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

PARTITION EVOLUTION

The ability to change the partitioning strategy of a table over time.

```
ALTER TABLE my_table
DROP PARTITION FIELD old_field
ADD PARTITION FIELD new_field;
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>>



Shwetank Singh
GritSetGrow - GSGLearn.com

COMPACTION STRATEGIES

Techniques used to combine smaller files into larger ones to improve read performance.

CALL
`iceberg.system.compact_table('my_table');`



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>>



Shwetank Singh
GritSetGrow - GSGLearn.com

SNAPSHOT MANAGEMENT

Managing snapshots of a table to enable time travel and rollback operations.

```
CALL iceberg.system.expire_snapshots('my_table',  
older_than = '2023-01-01');
```

DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

MANIFEST MERGING

Combining multiple manifest files into a single one to optimize metadata management.

CALL

```
iceberg.system.merge_manifests('my_table');
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>>



Shwetank Singh
GritSetGrow - GSGLearn.com

TABLE MAINTENANCE

Routine operations to keep Iceberg tables optimized and performant, such as compaction and snapshot expiration.

```
CALL iceberg.system.expire_snapshots('my_table',  
older_than = '2023-01-01');
```

DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

METADATA CACHING

Storing frequently accessed metadata in memory to improve query performance.

Configuring Iceberg to cache metadata in memory for faster query planning.

DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

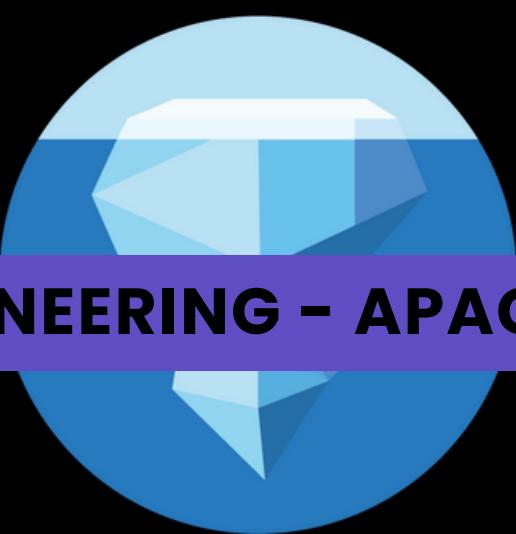


Shwetank Singh
GritSetGrow - GSGLearn.com

READ OPTIMIZATION

Techniques used to improve the performance of read operations in Iceberg tables.

Using partition pruning and data skipping to optimize read performance.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

WRITE OPTIMIZATION

Techniques used to improve the performance of write operations in Iceberg tables.

Using write distribution mode to balance the load across multiple writers.

DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

GARBAGE COLLECTION

Removing obsolete data files and metadata to free up storage space.

CALL

```
iceberg.system.remove_orphan_files('my_table');
```

DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>>



Shwetank Singh
GritSetGrow - GSGLearn.com

METADATA CLEANING

Removing outdated or redundant metadata to keep the catalog clean and efficient.

CALL

```
iceberg.system.remove_orphan_files('my_table');
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

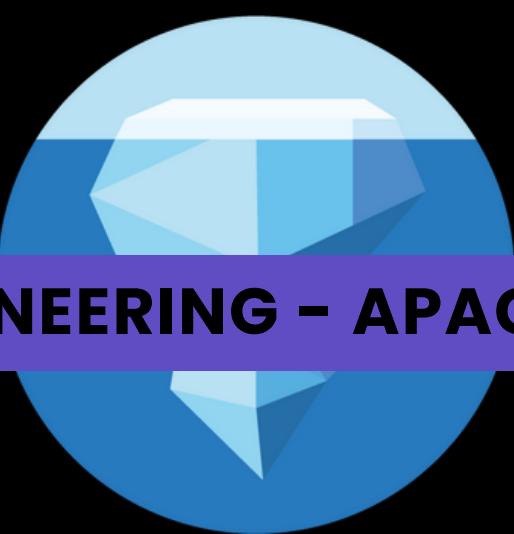


Shwetank Singh
GritSetGrow - GSGLearn.com

TRANSACTION ISOLATION

Ensuring that concurrent transactions do not interfere with each other, maintaining data consistency.

Using snapshot isolation to ensure that each transaction operates on a consistent view of the data.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

DATA VERSIONING

Keeping track of changes to the data over time, enabling rollback and historical queries.

Creating new snapshots for each data update to track versions over time.

DATA ENGINEERING – APACHE ICEBERG



follow for
more content ➤



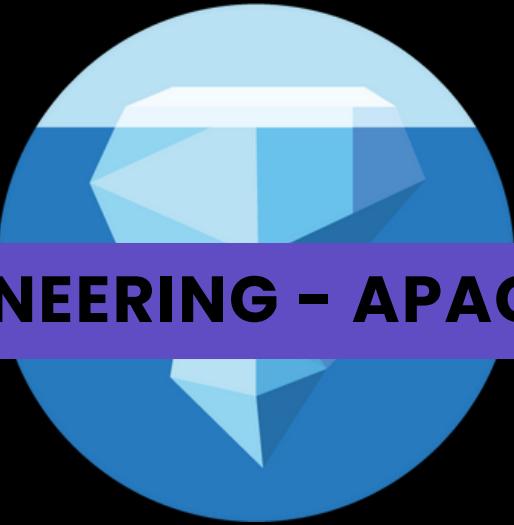
Shwetank Singh
GritSetGrow - GSGLearn.com

BRANCHING

Creating separate branches of the table metadata to support different versions or experiments.

CALL

```
iceberg.system.create_branch('my_table',  
branch = 'test_branch');
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

TAGGING

Assigning tags to specific snapshots for easy identification and access.

```
CALL iceberg.system.tag_snapshot('my_table',  
snapshot_id = 12345, tag = 'release_v1.0');
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

CATALOG MIGRATION

Moving metadata from one catalog implementation to another.

Migrating from a Hive catalog to a REST catalog to take advantage of new features.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

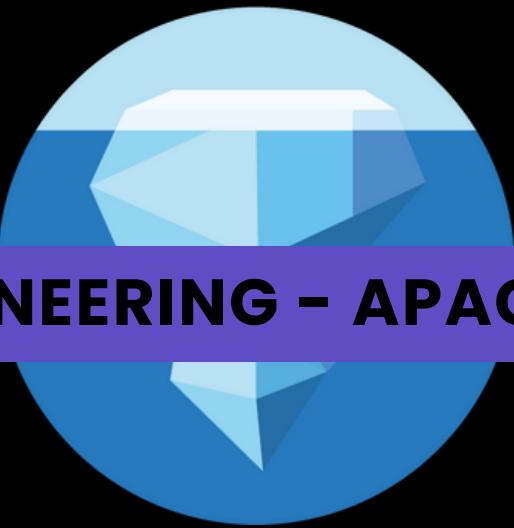


Shwetank Singh
GritSetGrow - GSGLearn.com

INCREMENTAL DATA PROCESSING

Processing only the data that has changed since the last operation.

Using incremental processing to update a dashboard with the latest data changes.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

DATA LINEAGE

Tracking the origin and transformations of data over time.

Using data lineage to understand how data in a report was generated from raw input files.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

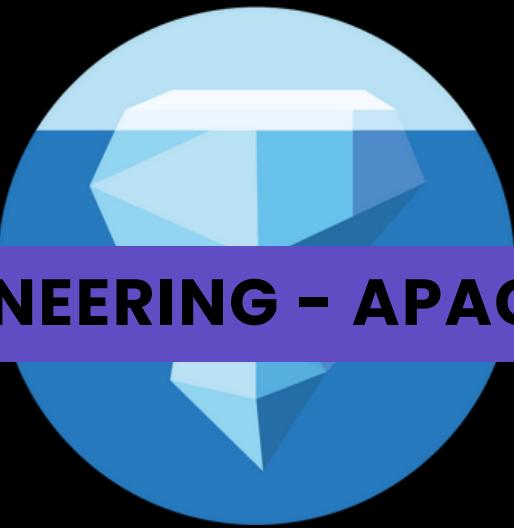


Shwetank Singh
GritSetGrow - GSGLearn.com

AUDITING

Tracking access and changes to data for security and compliance purposes.

Using audit logs to track who accessed or modified data in an Iceberg table.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

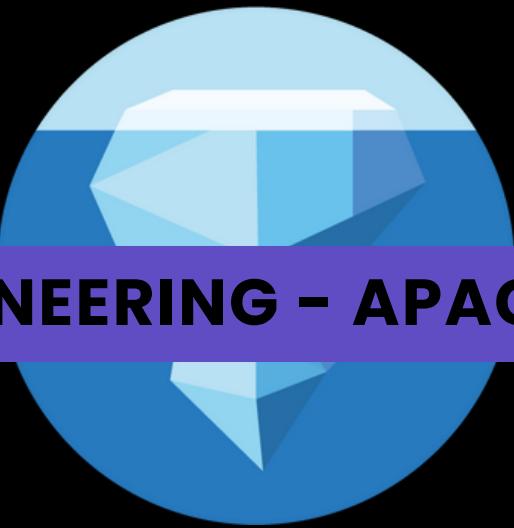


Shwetank Singh
GritSetGrow - GSGLearn.com

METADATA REPLICATION

Copying metadata across different locations or systems to ensure high availability.

Using metadata replication to keep a backup copy of Iceberg metadata in a different data center.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

FEDERATED QUERYING

Querying data across multiple Iceberg tables or catalogs as if they were a single table.

Using federated querying to join data from Iceberg tables in different catalogs.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

SCHEMA VALIDATION

Ensuring that the schema of incoming data matches the expected schema.

Using schema validation to enforce data quality rules when loading new data into an Iceberg table.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

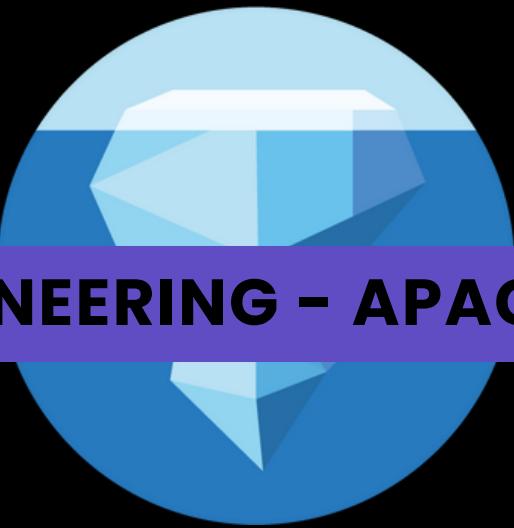


Shwetank Singh
GritSetGrow - GSGLearn.com

DATA ENCRYPTION

Protecting data at rest and in transit using encryption techniques.

Configuring Iceberg to use encryption keys for data files stored in S3.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

DATA MASKING

Hiding sensitive data by obfuscating it in query results.

Using data masking to hide personally identifiable information (PII) in Iceberg table queries.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

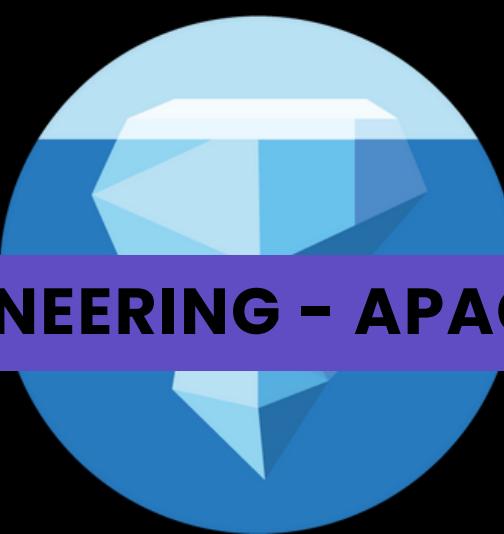


Shwetank Singh
GritSetGrow - GSGLearn.com

ROLE-BASED ACCESS CONTROL

Managing permissions based on user roles to secure access to data.

Using RBAC to grant read-only access to analysts and write access to data engineers for Iceberg tables.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

INTEGRATION WITH SPARK

Using Apache Spark to perform data processing and analytics on Iceberg tables.

```
spark.read.format("iceberg").load("my_table")  
.show();
```



DATA ENGINEERING – APACHE ICEBERG



follow for
more content ➤



Shwetank Singh
GritSetGrow - GSGLearn.com

INTEGRATION WITH FLINK

Using Apache Flink to perform stream processing on Iceberg tables.

```
table_env.execute_sql("SELECT * FROM  
iceberg.my_table");
```



DATA ENGINEERING – APACHE ICEBERG



follow for
more content ➤



Shwetank Singh
GritSetGrow - GSGLearn.com

INTEGRATION WITH HIVE

Using Apache Hive to query and manage
Iceberg tables.

```
CREATE EXTERNAL TABLE my_table
STORED BY
'org.apache.iceberg.mr.hive.HiveIcebergStora
geHandler' TBLPROPERTIES ('iceberg.catalog' =
'hive');
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

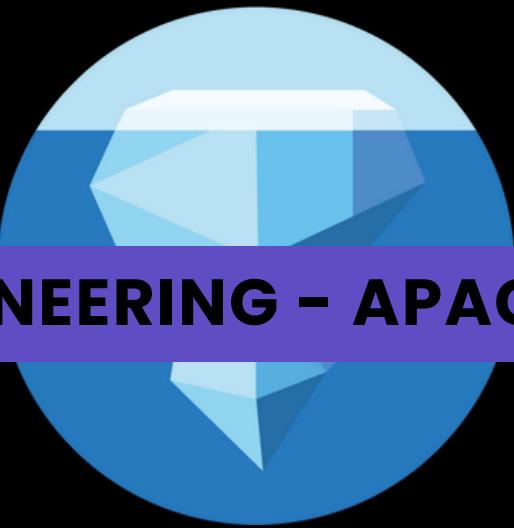


Shwetank Singh
GritSetGrow - GSGLearn.com

INTEGRATION WITH PRESTO

Using Presto to query Iceberg tables.

```
SELECT * FROM iceberg.hive.my_table;
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

INTEGRATION WITH TRINO

Using Trino to query Iceberg tables.

```
SELECT * FROM iceberg.hive.my_table;
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

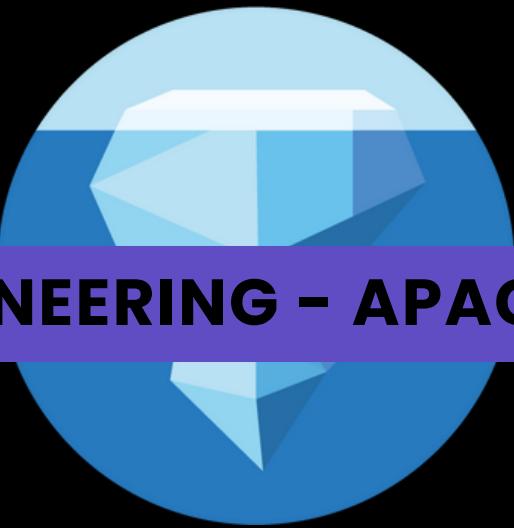


Shwetank Singh
GritSetGrow - GSGLearn.com

INTEGRATION WITH DREMIO

Using Dremio to query and analyze
Iceberg tables.

Adding an Iceberg source in Dremio and
running queries on the Iceberg tables.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

INTEGRATION WITH ATHENA

Using Amazon Athena to query Iceberg tables stored in S3.

Creating an Athena data source pointing to an Iceberg table in S3.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

DATA COMPACTION JOBS

Scheduled jobs to compact data files and improve read performance.

CALL

```
iceberg.system.compact_table('my_table');
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>>



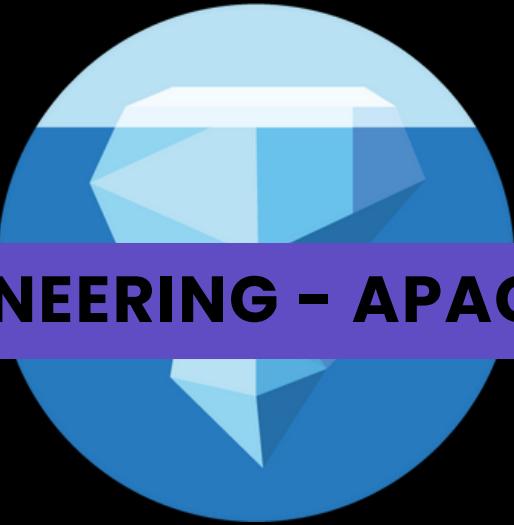
Shwetank Singh
GritSetGrow - GSGLearn.com

EXPIRING SNAPSHOTS

Removing old snapshots to save storage space and keep metadata manageable.

CALL

```
iceberg.system.expire_snapshots('my_table',  
older_than = '2023-01-01');
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

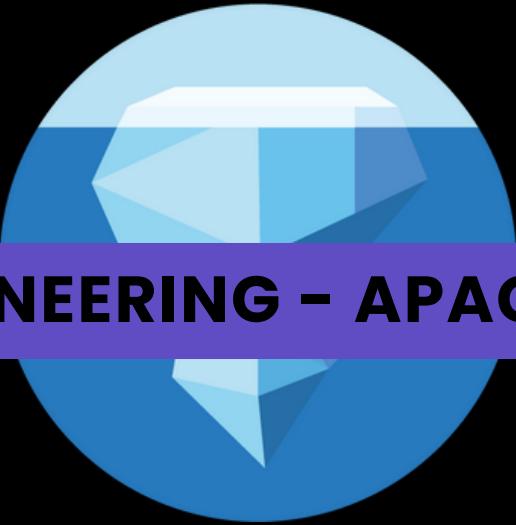


Shwetank Singh
GritSetGrow - GSGLearn.com

ROLLING BACK SNAPSHOTS

Reverting a table to a previous state using a snapshot.

```
CALL iceberg.system.rollback('my_table',  
snapshot_id = 12345);
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

MANAGING PARTITIONS

Creating, altering, and dropping partitions
to optimize query performance.

```
ALTER TABLE my_table  
ADD PARTITION FIELD date_trunc('month',  
event_time);
```



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>

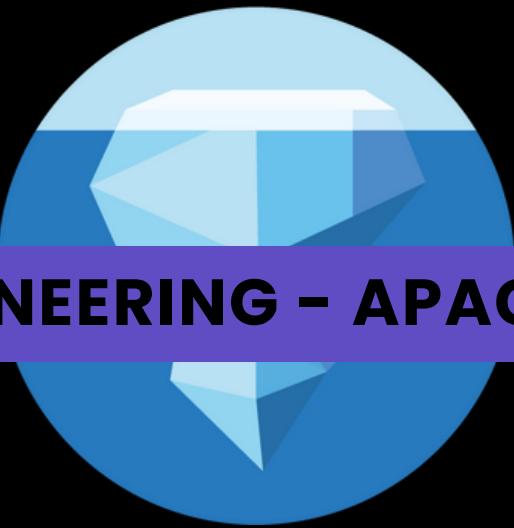


Shwetank Singh
GritSetGrow - GSGLearn.com

DATA QUALITY CHECKS

Ensuring data integrity and consistency through validation rules.

Using data quality checks to enforce that no null values exist in the primary key column.



DATA ENGINEERING – APACHE ICEBERG



*follow for
more content* >>



Shwetank Singh
GritSetGrow - GSGLearn.com

QUERY OPTIMIZATION TECHNIQUES

Strategies to improve query performance, such as predicate pushdown and partition pruning.

Using predicate pushdown to filter data at the storage layer, reducing the amount of data read.



DATA ENGINEERING – APACHE ICEBERG



Thank you!



Shwetank Singh
GritSetGrow - GSGLearn.com

follow for more content