DIGGIBYTE

CREATING VALUE WITH DATA

DIGGIBYTE
CREATING VALUE WITH DATA

Delta Lake

Presented by,
Amit Kundu

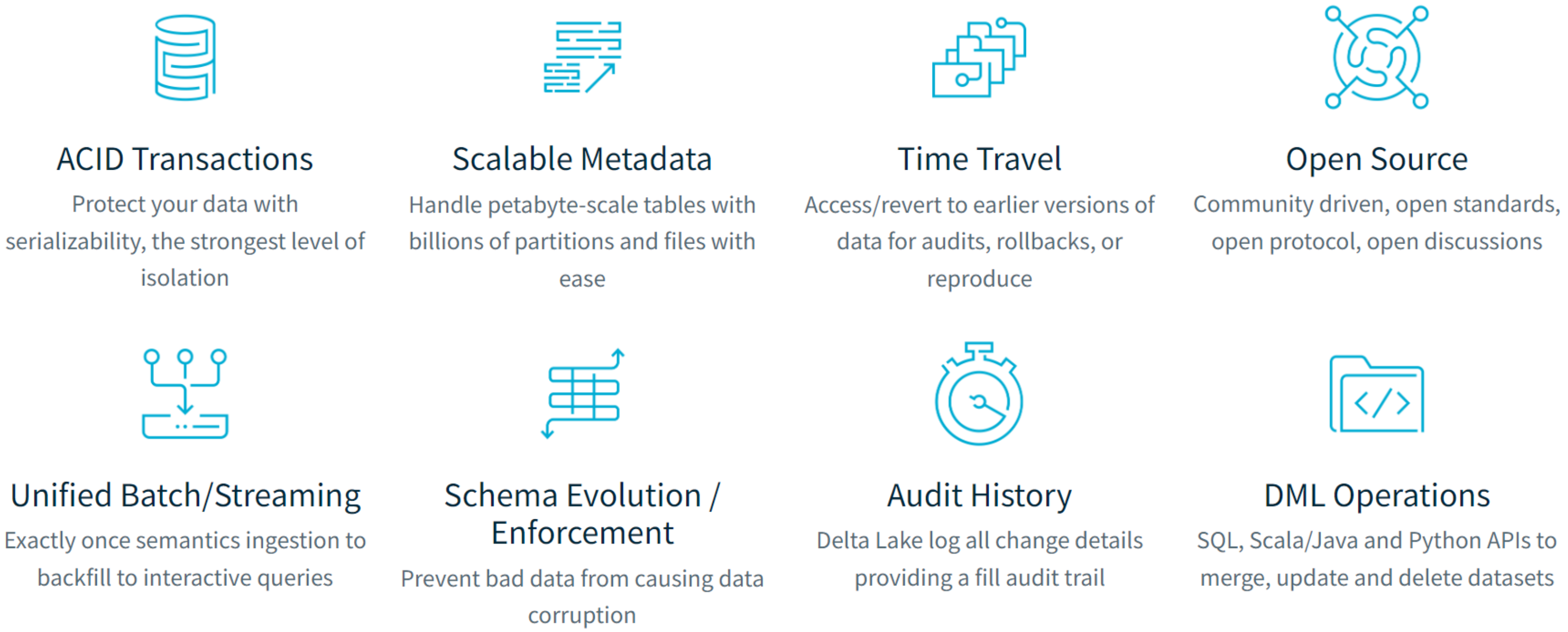
Agenda

- Introduction
- Delta Table Optimization
- Z-Order Commands
- ACID Transactions
- Summary
- Q&A

DIGGIBYTE
CREATING VALUE WITH DATA

Introduction to Delta Lake

- Delta Lake is an open-source storage layer that brings reliability to data lakes.
- It provides ACID transactions, scalable metadata handling, and unifies streaming and batch data processing.



ACID Transactions

Protect your data with serializability, the strongest level of isolation

Scalable Metadata

Handle petabyte-scale tables with billions of partitions and files with ease

Time Travel

Access/revert to earlier versions of data for audits, rollbacks, or reproduce

Open Source

Community driven, open standards, open protocol, open discussions

Unified Batch/Streaming

Exactly once semantics ingestion to backfill to interactive queries

Schema Evolution / Enforcement

Prevent bad data from causing data corruption

Audit History

Delta Lake log all change details providing a fill audit trail

DML Operations

SQL, Scala/Java and Python APIs to merge, update and delete datasets

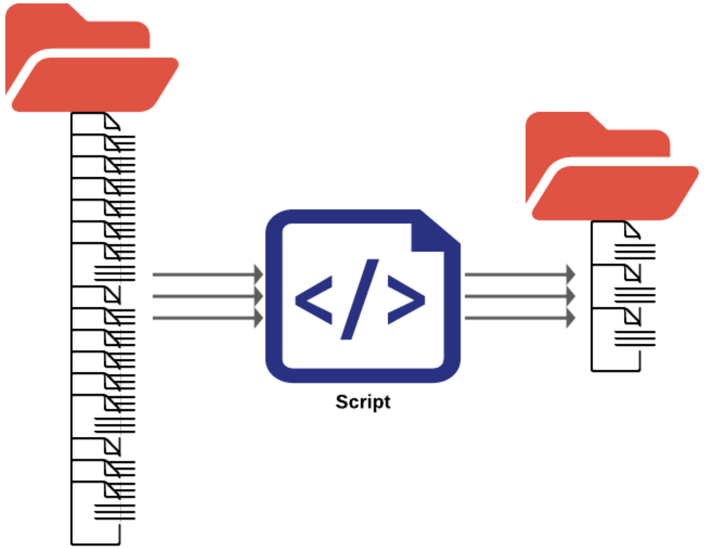
Delta Table Optimization

- Improve query performance and reduce latency by optimizing how data is stored.
- **Key Techniques:**
  - File Compaction: Merging small files into larger ones to reduce overhead.
  - Data Skipping: Using statistics to skip irrelevant data during query execution.

DIGGIBYTE
CREATING VALUE WITH DATA

File Compaction

- Compaction is the process of merging smaller files into larger ones to optimize storage and read performance.
- **Benefits:**
  - Reduces the number of files
  - Decreases query latency

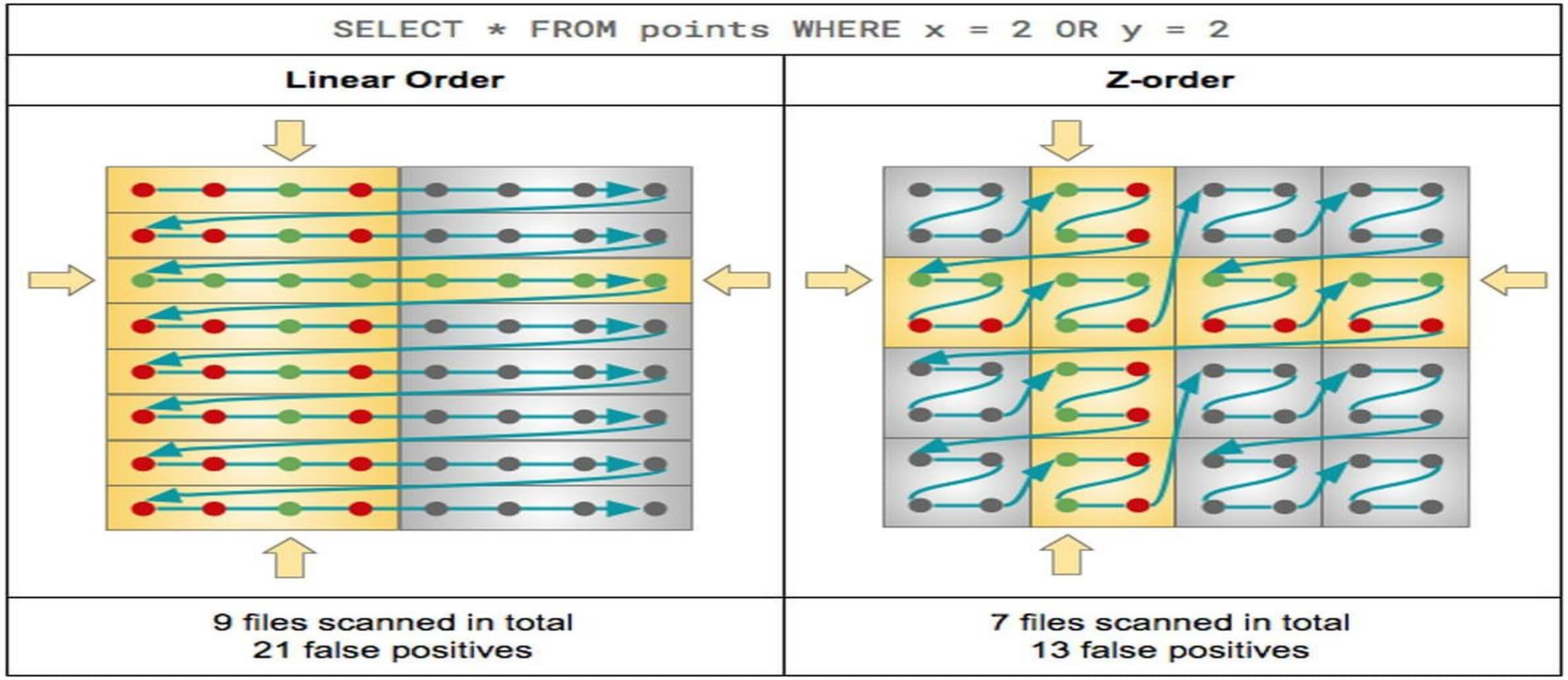


Data skipping

- Data skipping leverages file-level statistics to avoid reading irrelevant data, thereby improving query performance.
- **How It Works:**
    - Statistics such as min and max values are collected for each file.
    - During query execution, files that do not match the query criteria are skipped.

Z-Order Commands

- Z-Ordering is a technique to optimize the storage of data by ordering it based on the values of one or more columns.
- **Benefits:**
  - Improves query performance for queries filtering on multiple columns.
  - Reduces the amount of data read during query execution.



```
SELECT * FROM points WHERE x = 2 OR y = 2
```

Linear Order

9 files scanned in total
21 false positives

Z-order

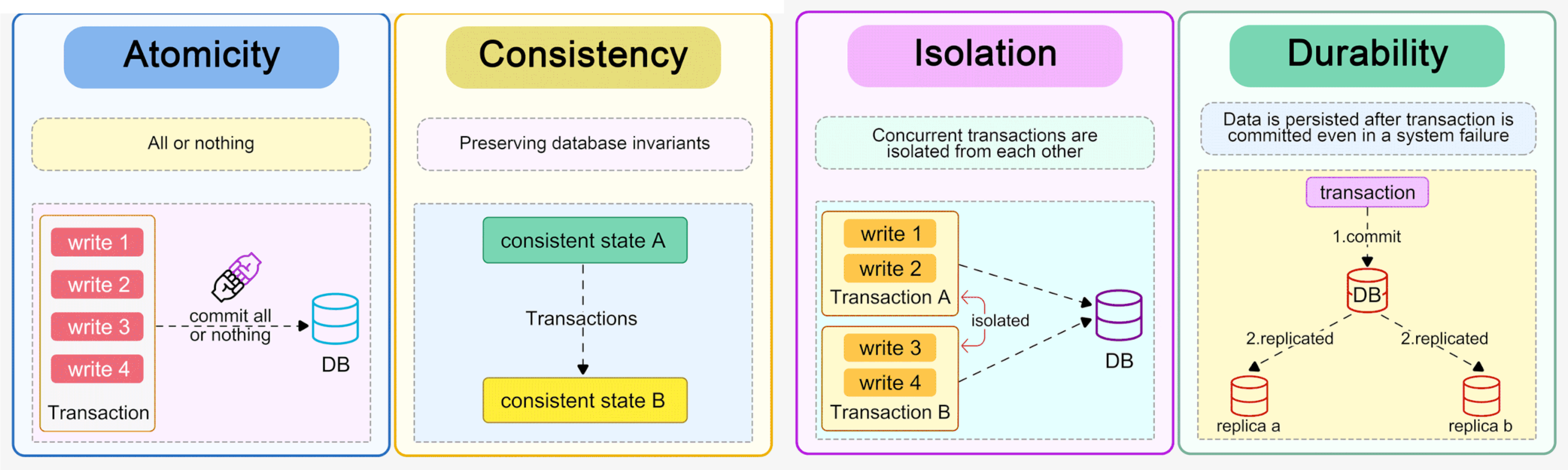
7 files scanned in total
13 false positives

ACID Transactions

- ACID stands for Atomicity, Consistency, Isolation, Durability.
- **Properties:**
    - Atomicity: Ensures all operations within a transaction are completed successfully or none at all.
    - Consistency: Ensures data remains consistent before and after the transaction.
    - Isolation: Ensures transactions are executed in isolation from one another.
    - Durability: Ensures that once a transaction is committed, it remains so, even in the event of a system failure.

DIGGIBYTE
CREATING VALUE WITH DATA

ACID Transactions



Atomicity

All or nothing

write 1
write 2
write 3
write 4
Transaction
commit all or nothing
DB

Consistency

Preserving database invariants

consistent state A
Transactions
consistent state B

Isolation

Concurrent transactions are isolated from each other

write 1
write 2
Transaction A
write 3
write 4
Transaction B
isolated
DB

Durability

Data is persisted after transaction is committed even in a system failure

transaction
1.commit
DB
2.replicated
2.replicated
replica a
replica b

Summary

- **Delta Lake Enhancements:**
  - Delta Table Optimization and Z-Order significantly boost query performance.
  - ACID Transactions ensure data integrity and support concurrent reads and writes.
- **Conclusion:**
  - Delta Lake is a powerful addition to data lakes, providing robust features for efficient and reliable data processing.

THANK YOU !!