Understanding Delta Lake

DATA MANAGEMENT IN DATABRICKS

Smriti Mishra
Founder, NordData Insight

My Databricks journey

databricks

python™

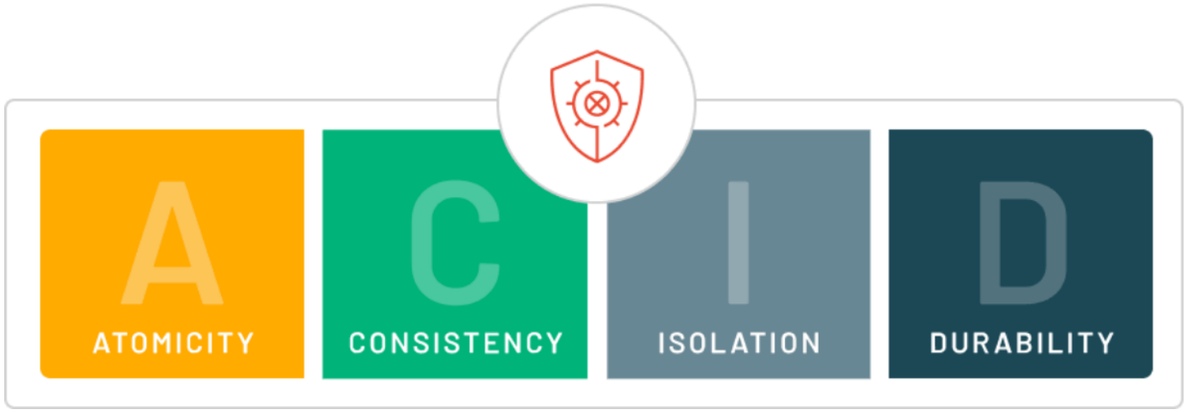
Healthcare scenario

The Delta Lake in Healthcare

- Delta Lake ensures organized, accurate, accessible data.
- Outperforms traditional data warehouses and lakes.
- Simplifies healthcare data management.
- Maintains data integrity and accessibility.

ACID Transactions

ACID Transactions



- **ACID**: Ensures data integrity and reliable, consistent processing

ACID Transactions

- **ACID**: Ensures data integrity and reliable, consistent processing
- Delta Lake ensures accurate handling of patient updates.
- It prevents incomplete records and data errors.
- Supports reliable data for patient safety and care.

Schema enforcement and evolution

- Ensures data integrity in patient care.
- Adapts as patient records change.
- Prevents workflow disruptions and data corruption.

Time travel feature

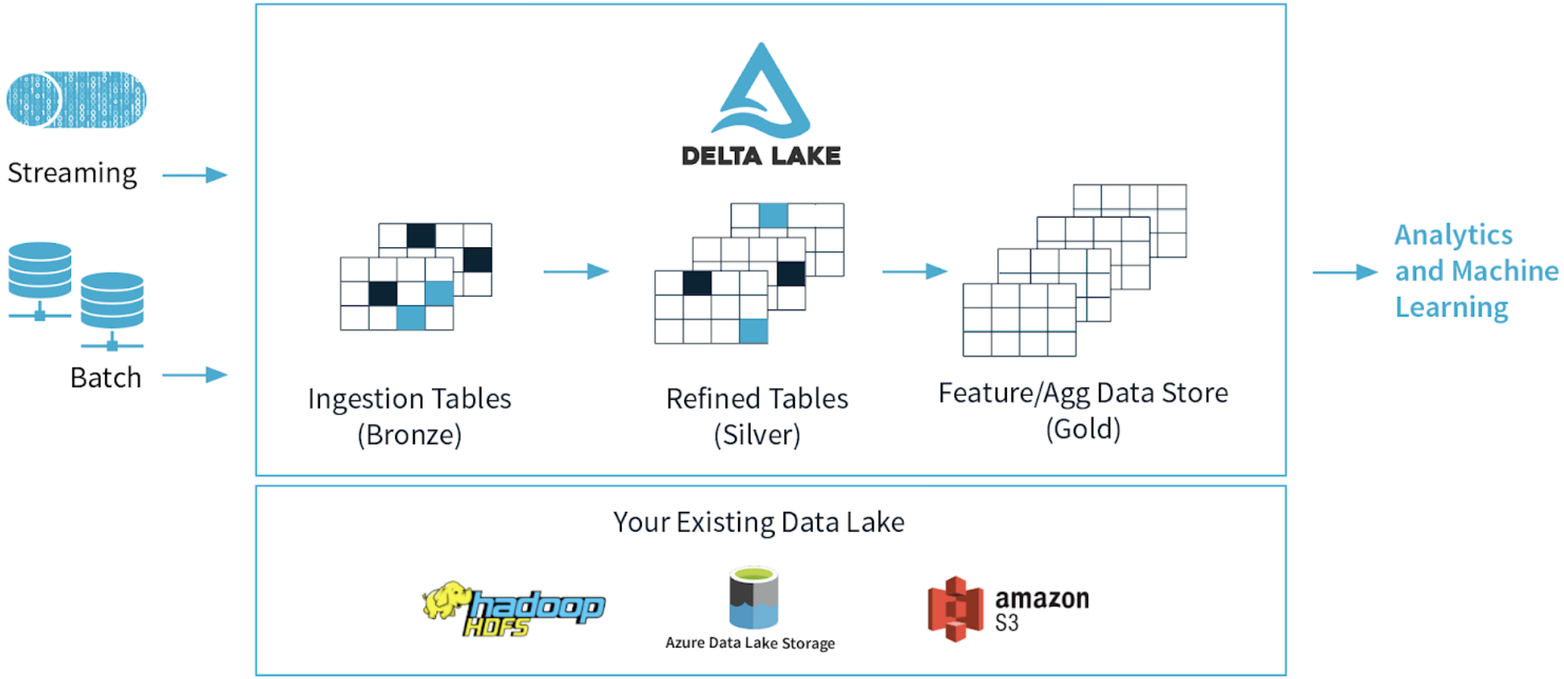
- Access previous versions of data for historical review
- Ensures accurate tracking of past treatments
- Importance of maintaining a comprehensive patient care history

Unified batch and streaming processing

- Delta Lake's support for both real-time monitoring and batch processing
- Reduces redundancy and processing complexity

Delta Lake architecture

- Overview of Delta Tables as secure electronic health records
- Transaction log ensures reliability and time travel capabilities
- Parquet files for quick access and analysis



Comparisons

- **Data warehousing** struggles with unstructured data
- **Data lakes** lack structure without management
- **Delta Lake** combines structure with flexibility, ideal for healthcare

[1] https://www.databricks.com/product/delta-sharing

Let's practice!

DATA MANAGEMENT IN DATABRICKS

Persistence and scope of tables

DATA MANAGEMENT IN DATABRICKS

Smriti Mishra
Founder, NordData Insight

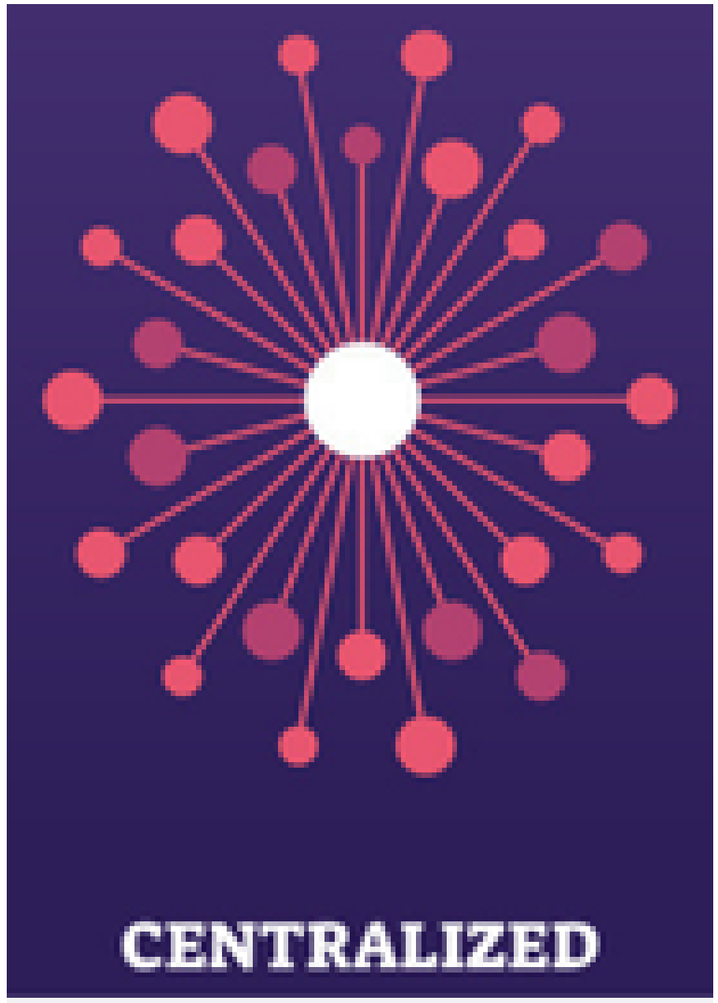
datacamp

What is table persistence?

- Table persistence controls data storage and retention
- It affects storage, access, and maintenance
- Databricks supports managed and unmanaged tables

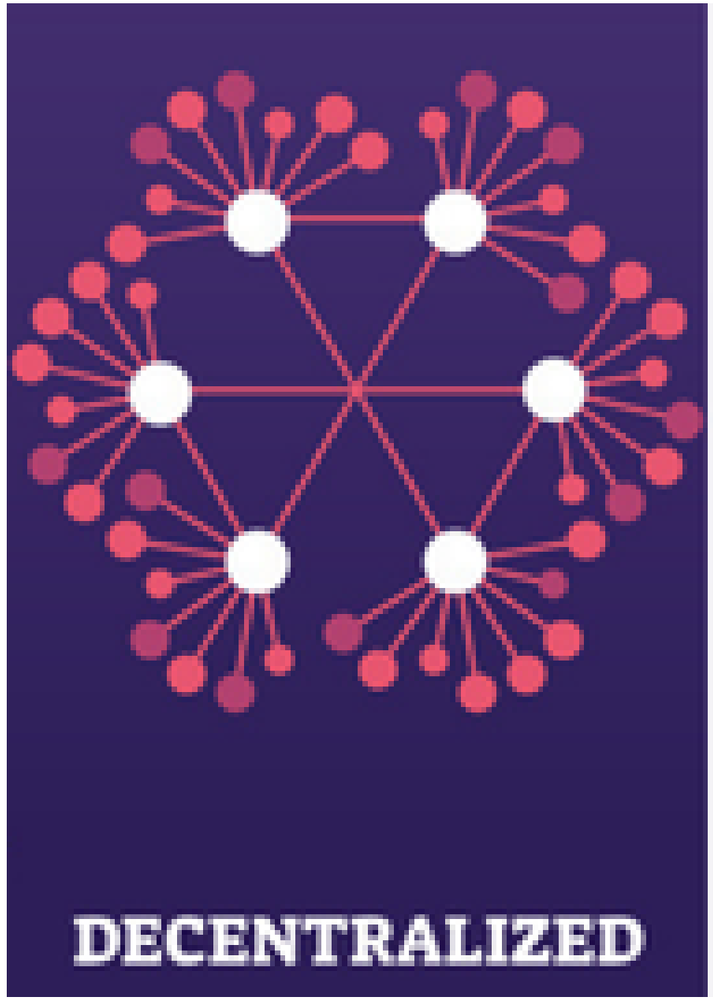
Managed tables in Databricks

- Fully managed by Databricks, including data location and lifecycle.
- Automatically deletes data when the table is deleted.
- Suitable for simple, centralized data management.



Unmanaged tables in Databricks

- Decentralized approach
- Control the data storage location and lifecycle
- Deleting an unmanaged table doesn't delete the data
- Useful for custom storage or compliance requirements



Managed or unmanaged tables?

Aspect	Managed Table	Unmanaged Table
Data Storage	Stored in Databricks' default path, usually `/user/hive/warehouse/`	Stored in user-defined locations like S3, ADLS, or Azure Blob
Data Management	Databricks manages the data and metadata	Databricks manages the metadata only; data stays external
Data Lifecycle	Dropping deletes both data and metadata	Dropping removes metadata; data stays in place
Use Cases	Best for temporary tables needing auto-management	Ideal for sharing or using external data
Automatic Cleanup	Data deletes automatically when dropped	Data remains after the table is dropped
Data Sharing	Sharing across systems requires exporting	Easier to share via accessible storage

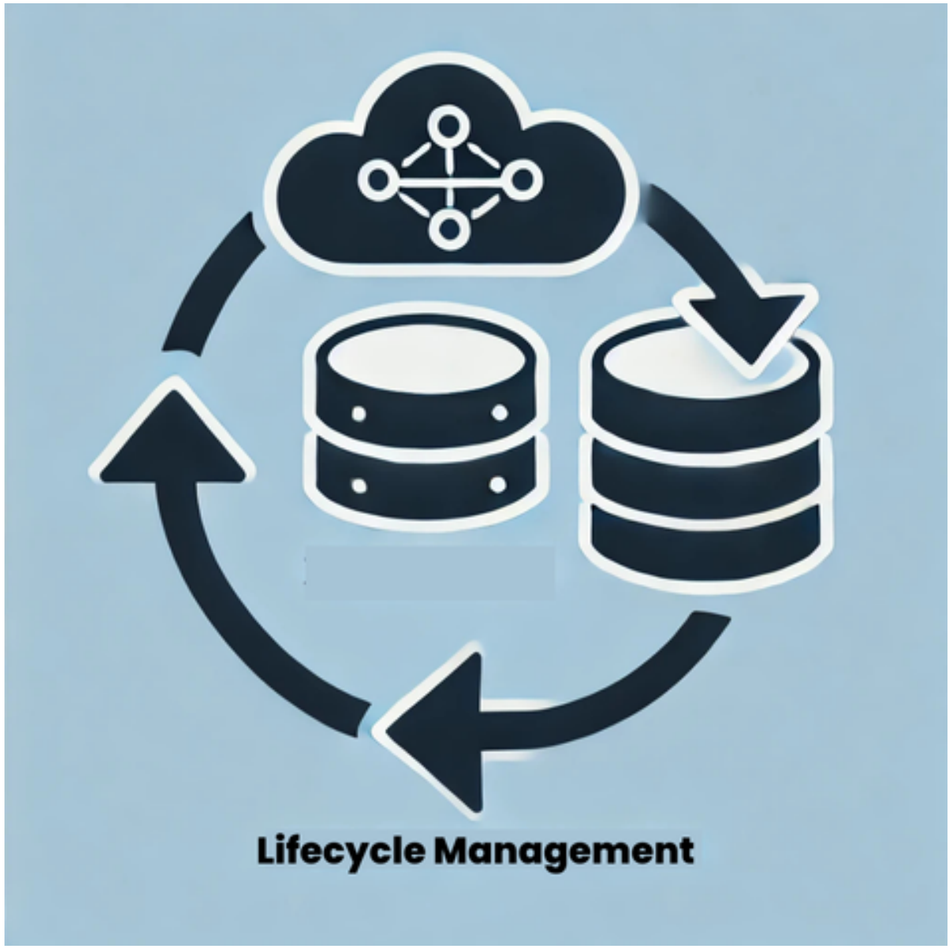
The LOCATION keyword

- Essential for setting data storage in unmanaged tables.
- Storage location impacts cost, retrieval times, and retention policies.

```
CREATE TABLE table_name (
    column_name data_type,
    ...
)
USING file_format
LOCATION 'path/to/data';
```

Key takeaways

- Managed tables centralize storage and lifecycle within Databricks.
- Unmanaged tables offer flexibility for storage and data lifecycle.
- Choose based on data storage, control, and management needs.



Let's practice!

DATA MANAGEMENT IN DATABRICKS

Table showdown

DATA MANAGEMENT IN DATABRICKS

Smriti Mishra
Founder, NordData Insight

Let's practice!

DATA MANAGEMENT IN DATABRICKS