Delta Lake Introduction

DAY 04

DATABRICKS LEARNING CHALLENGE

Master the fundamentals of Delta Lake and transform how you manage data in modern data engineering pipelines.

In collaboration with:

Databricks | Indian Data Club | Codebasics

Presented by: Ruchi Wange

FOUNDATIO
N

What is Delta Lake?

Delta Lake is an **open-source storage layer** that sits on top of your data lake, bringing reliability and performance to cloud storage systems.

Think of it as an upgrade that transforms simple file storage into a powerful database-like system while keeping your data in standard formats.

Storage Layer

Built on Parquet files

ACID Support

Database reliability

COMPARISON

CSV vs Parquet vs Delta

CSV Files

Simple text format

- No schema enforcement
- Slow to read
- No transactions

1

Parquet Files

Columnar binary format

- Fast reads
- Compressed storage
- Still no ACID

2

Delta Format

Parquet + transaction log

- ACID transactions
- Schema enforcement
- Time travel enabled

3

Delta Lake combines the performance of Parquet with database-level reliability, making it the ideal choice for production data pipelines.

PROBLEMS & SOLUTIONS

Why Delta Lake Matters

Traditional Data Lake Problems

- No ACID guarantees
- Duplicate records
- Schema inconsistencies
- Failed writes leave corrupt data

Delta Lake Solutions

- Full ACID transactions
- MERGE for deduplication
- Automatic schema validation
- Atomic operations

Real-World Impact: Delta Lake prevents data quality issues that cost companies millions in bad analytics and failed pipelines.

CORE FEATURES

Delta Lake Superpowers

ACID Transactions

All operations are **Atomic, Consistent, Isolated, and Durable**. Multiple users can read/write simultaneously without conflicts or corruption.

Schema Enforcement

Delta automatically validates data against the table schema. Invalid data is rejected, preventing quality issues downstream.

Time Travel

Access previous versions of your data using timestamps or version numbers. Perfect for audits and recovering from mistakes.

MERGE Operations

Upsert data efficiently by updating existing records and inserting new ones in a single atomic operation.

UPDATE & DELETE

Modify or remove records directly, unlike traditional data lakes where you'd need to rewrite entire partitions.

DATASET
OVERVIEW

E-Commerce Dataset

Monthly Data Ingestion

Our dataset contains e-commerce events from **October and November 2019**, representing typical monthly data loads in production systems.

01

October 2019

Initial data load from CSV

02

November 2019

Incremental monthly append

Key Columns

- **event_time:** Timestamp of user action
- **event_type:** view, cart, purchase
- **product_id:** Unique product identifier
- **category_code:** Product category
- **brand:** Product brand name
- **price:** Product price
- **user_id:** Customer identifier
- **user_session:** Session tracking ID

TRANSFORMATION

CSV to Delta Conversion

Load CSV Files

Read October and November data from raw CSV format using Spark DataFrames.

Write as Delta

Convert to Delta format using .write.format("delta") with overwrite and append modes.

Create Managed Tables

Store as Delta tables in the workspace catalog for easy querying and management.

Benefits Gained

- ACID transaction support
- Schema validation on writes
- 10-100x faster queries
- Automatic data optimization
- Version history tracking

DATA
QUALITY

Handling NULL Values

The NULL Problem

NULL values represent missing or unknown data. In analytics, NULLs can cause:

- Incorrect aggregations
- Failed joins
- Confusing reports
- Application errors

In our dataset, category_code and brand fields often contain NULLs when data isn't available.

The Solution

Use COALESCE to replace NULLs with meaningful defaults like "Not Available":

```
COALESCE(brand, 'Not Available') AS brand
```

This ensures:

- Consistent data for analytics
- Clear indication of missing values
- Reliable downstream processing

DEDUPLICATION

Handling Duplicate Inserts

1

The Duplicate Problem

When loading data multiple times, the same records can be inserted repeatedly, causing inflated counts and incorrect analytics.

2

MERGE to the Rescue

The MERGE command performs an "upsert" - it updates existing records or inserts new ones based on matching keys.

```
MERGE INTO target t
USING source s
ON t.event_time = s.event_time
  AND t.user_id = s.user_id
WHEN NOT MATCHED THEN INSERT *
```

3

Why MERGE Beats INSERT

- Prevents duplicates automatically
- Single atomic operation
- Can update existing records
- Maintains data integrity

GOVERNANCE & PRODUCTION

Delta Table History & Real-World Use

Audit Trail with DESCRIBE HISTORY

Every Delta table maintains a complete transaction log showing:

- Who made changes (user email)
- When changes occurred
- What operation was performed
- Version numbers for time travel

This audit trail is **critical for data governance** and compliance requirements.

Production Use Cases

Incremental Loads

Daily/hourly data ingestion with MERGE

Data Versioning

Rollback to previous states instantly

Concurrent Access

Multiple teams reading/writing safely

Key Takeaway: Delta Lake transforms data lakes from simple storage into reliable, production-grade data platforms that power modern analytics and AI applications.

Presented by: Ruchi Wange