Manage Data with Delta Lake

Module 03

Module Agenda

Manage Data with Delta Lake

What is Delta Lake

DE 3.1 – Schemas and Tables

DE 3.2 – Version and Optimize Delta Tables

DE 3.3L – Manipulate Delta Tables Lab

DE 3.4 – Set Up Delta Tables

DE 3.5 – Load Data into Delta Lake

DE 3.6L – Load Data Lab

What is Delta Lake?

Delta Lake is an open-source project that enables building a data lakehouse on top of existing cloud storage

Delta Lake Is Not…

- Proprietary technology
- Storage format
- Storage medium
- Database service or data warehouse

Delta Lake Is…

- Open source
- Builds upon standard data formats
- Optimized for cloud object storage
- Built for scalable metadata handling

Delta Lake brings ACID to object storage

Atomicity means all transactions either succeed or fail completely

Consistency guarantees relate to how a given state of the data is observed by simultaneous operations

Isolation refers to how simultaneous operations conflict with one another. The isolation guarantees that Delta Lake provides do differ from other systems

Durability means that committed changes are permanent

Problems solved by ACID

- Hard to append data
- Modification of existing data difficult
- Jobs failing mid way
- Real-time operations hard
- Costly to keep historical data versions

Delta Lake is the default format for tables created in Databricks

```
CREATE TABLE foo
~~USING DELTA~~
```

```
df.write
  .~~format(“delta”)~~
```

DE 3.1: Schemas and Tables

Create schema (database) as repository for your tables/views

Create managed and external Delta tables

Insert records in Delta Lake tables

Dropping Delta Lake tables

DE 3.2: Version and Optimize Delta Tables

Use OPTIMIZE to compact small files into 1GB size along with ZORDER to sort like values in same file(s)

Describe the directory structure of Delta Lake files

Review a history of table transactions

Query and roll back to previous table version

Delete stale data files via VACUUM and DRY RUN

DE 3.3L - Manipulate Delta Tables Lab

Review table history

Query previous table versions and rollback a table to a specific version

Perform file compaction and Z-order indexing

Preview files marked for permanent deletion and commit these deletes

DE 3.4: Set up Delta Tables

Using CTAS statements to create Delta Lake tables

Creating new tables from existing views or tables

Declaring table schema with generated columns and descriptive Comments

Setting options for location, constraints, and partitions

Creating deep and shallow clones

DE 3.5: Load Data into Delta Tables

CREATE OR REPLACE TABLE

Overwrite data tables using INSERT OVERWRITE

Append to a table using INSERT INTO

Append, update, and delete using MERGE INTO

Ingest data incrementally into tables using COPY INTO

DE 3.6L: Load Data Lab

Create an empty Delta table with a provided schema

INSERT INTO from an existing JSON table into a Delta table

Use a CTAS statement to create a Delta table from files databricks