

Title

ShopEZ E-Commerce Orders – Delta Lake Data Engineering Pipeline

Objective:

Design and implement a **Delta Lake–based data storage system** on Databricks for the ShopEZ e-commerce platform that processes daily international orders with:

ACID transactions

Time Travel

Schema evolution

Updates & deletes

Optimize / Z-Ordering

Dataset Description

The input contains **40 sample e-commerce orders** with the following fields:

Column	Type	Description
order_id	string	Unique order ID
order_timestamp	timestamp	When the order was placed
customer_id	string	Customer ID
country	string	Country code (US, IN, UK, FR, etc.)
amount	double	Order amount
currency	string	Currency code
status	string	CREATED / PAID / CANCELLED

Tasks Completed

1. Ingest Input Data

- Created a PySpark DataFrame using predefined schema.
- Loaded 40 rows of order data.

2. Derived Column – order_date

- Extracted date from timestamp using:
`withColumn("order_date", to_date("order_timestamp"))`

3. Write as Delta Table with Partitioning

- Stored table at:
dbfs:/mnt/delta/shop_ez_orders
- Partitioned by:
country and **order_date**
- Registered table:
CREATE TABLE shop_ez_orders USING DELTA LOCATION ...

4. Verified Partition Structure

Used DBFS browser and:

```
dbutils.fs.ls(path)
```

5. Demonstrated Partition Pruning

Two queries:

- Filter by country ,Filter by country + order_date
Explained query plan using `.explain(True)`.

6.Delta Time Travel

- Viewed table history using:
DESCRIBE HISTORY shop_ez_orders
- Updated rows to create new versions.
- Queried older versions using `versionAsOf`.

7. Schema Evolution

- Added `payment_method` and `coupon_code`.
- Used:
`.option("mergeSchema", "true")`
- Appended new data **without using partitionBy** again.

8. Updates & Deletes (Delta Lake)

- Updated order statuses (ex: mark high-value orders as CANCELLED).
- Deleted low-value test entries.

9. Optimization (Bonus)

- Compacted files using:
OPTIMIZE shop_ez_orders
- Applied ZORDER on `customer_id`.

10. Small File Problem (Bonus)

- Demonstrated how excessive partitions create many small files.
- Showed how OPTIMIZE resolves it.

Technologies Used

- Apache Spark (PySpark)
- Delta Lake
- Databricks Notebook
- DBFS (Databricks File System)