**Case Study: Migrating from Parquet to Delta & Enabling GDPR Compliance with Time Travel**

**Background**

A company stores customer data in Parquet format. With increasing regulatory requirements (GDPR), they need to:

* Support **deletion of user data** upon request.
* Be able to **restore** previous versions of the dataset for audit or compliance.
* Leverage **Delta Lake** features for schema evolution and time travel.

**Step-by-Step Objectives**

1. **Convert existing Parquet data to Delta format.**
2. **Perform GDPR-compliant deletion** of a user’s data.
3. **Prove data deletion by querying the current state.**
4. **Restore the dataset to a previous version** before deletion using Delta Time Travel.

**Step-by-Step Lab**

**Step 1: Prepare a sample Parquet dataset**

python

data = [

(1, 'Alice', 'UK'),

(2, 'Bob', 'Germany'),

(3, 'Carlos', 'Spain')

]

df = spark.createDataFrame(data, ['user\_id', 'name', 'country'])

df.write.parquet('/tmp/customer\_parquet')

**Step 2: Convert Parquet to Delta**

python

# Load existing Parquet data

parquet\_df = spark.read.parquet('/tmp/customer\_parquet')

# Write it as a Delta table

parquet\_df.write.format("delta").save("/tmp/customer\_delta")

# Register it as a managed table (optional)

spark.sql("CREATE TABLE customer\_delta USING DELTA LOCATION '/tmp/customer\_delta'")

**Step 3: View original data**

sql

SELECT \* FROM delta.`/tmp/customer\_delta`;

**Step 4: GDPR Delete a Specific User**

sql

DELETE FROM delta.`/tmp/customer\_delta` WHERE user\_id = 2;

**Step 5: Prove Deletion**

sql

SELECT \* FROM delta.`/tmp/customer\_delta`; -- Bob’s record will be gone

**Step 6: Describe History**

sql

DESCRIBE HISTORY delta.`/tmp/customer\_delta`;

* Note down the **version number** before deletion (e.g., version 0).

**Step 7: Restore Previous Version using Time Travel**

sql

SELECT \* FROM delta.`/tmp/customer\_delta` VERSION AS OF 0;

You will see Bob’s record again.

**Optional: Restore the Entire Table**

sql

RESTORE TABLE customer\_delta TO VERSION AS OF 0;

**Business Value Achieved**

| **Feature** | **Benefit** |
| --- | --- |
| **Conversion to Delta** | Enables ACID transactions, schema enforcement |
| **GDPR Delete** | Easy compliance with regulation via DELETE + VACUUM |
| **Time Travel** | Point-in-time audit and recovery |
| **Delta Logs** | Transparent data change tracking for governance |

**Extensions**

* Add schema evolution (mergeSchema=true) during new data ingestion.
* Use Delta Table HISTORY and table\_changes() for full audit lineage.
* Integrate VACUUM for permanent deletion post-GDPR compliance.