**Delta lake with Uniform**

Every uniform Apache Hudi, Delta Lake or Apache Iceberg has json for meta data and parquet for data

Parquet files remain the same but meta data automatically generated to make Delta accessible as Iceberg or Hudi readers.

**Data Ingestion**

How to bring raw data to our Delta Lake. There are different Techniques:

Set the default cataloge and schema, and be sure if tables are existed.

USE CATALOGE ${module\_cataloge};

USE SCHEMA IDENTIFIER( :my\_schema);

SHOW TABLES;

LIST ‘ path ‘;

Remember that the source may be a file and it is possible to read a file and then create a table from it and store it in our storage, it can help also for versioning (time table) since you save the table in create command!

* **CTAS**: create table by selecting data from an existing table or data source. (DATA AND SCHEMA)

Drop the table if already existes

**DROP TABLE IF EXISTS current\_employee\_ctas;**

Create the table using CTAS

**CREATE TABLE mydeltatable AS SELECT**

**AS**

**SELECTED ID, Firstname, Country, Role**

**FROM read\_file(‘path’ , format =>’csv’ , header => true, inferSchema => true);**

Display available tables in your schema

**SHOW TABLES;**

REMEMBER THAT THERE IS AN ALTERNATE METHOD FOR CTAS AND IT IS CREATING THE VIEW

Drop the table if exists

**DROP TABLE IF EXISTS current\_employees\_ctas;**

Create temporary view

**CRAETE OR REPLACE TEMP VIEW vw\_currrent\_employees**

**USING CSV**

**OPTIONS (‘path’, header= ‘true’, delimiter = ‘,’);**

Use temporary view in CTAS statement or create the table

**CRAETE OR REPLACE TABLE current\_employees\_ctas AS**

**SELECT \***

**FROM vw\_cureent \_employees;**

* **UPLOAD UI**: provides a point and click interface to upload files and create tables and it allowes to uplaoad csv,tsv, parquet, avro … it enables you to uplad the file directly to a Unity Cataloge volumes.

1. Select the cataloge icon in the navigation bar.
2. Type the modul’s catalge name getstarted in the search bar.
3. Select the refresh icon to refresh the getstarted cataloge.
4. Expand the getstartted cataloge. Within the cataloge you should se a veraiety of schemas
5. Expand the schem. You can locate the schem in the setup nor=tes

* **COPY INTO**: load **files** from a file location into a delta table or support various files format and storage locations and handle the schema changes. (you can put location of the files not just one file)

The important is **Idempotent** that skipped from the already loaded file in the sources for increasing efficiency.

Create a table from employee.csv file using COPY INTO statement. The copy into statement loads data from a file location onto Delta table. This is a retriable and idempotent operation- file in the source location that have already been loaded are skipped.

Drop the table if exists

**DROP TABLE IF EXISTS current\_employee\_copyinto;**

Create an empty table with the **column data type,** so in **COPY INTO** you must know the **columns type.**

**CREATE TABLE current\_employee\_copyinto (ID INT, FirstName STRING, Country STRING, Role STRING);**

**COPY INTO my\_catalog.my\_schema.my\_table**

**FROM ‘path’**

**FILEFORMAT= CSV**

**FORMAT\_OPTIONS (‘header’ = ‘true’ , ‘inferSchema’ = ‘true’);**

**COPY INTO mydeltatable**

**FROM ‘the path’**

**FILE\_FORMAT = ‘ format’**

**FILE\_OPTION = (‘format\_option)**

**And then select shows the table**

**Key Features of COPY INTO**

✅ **Incremental Loads** – Avoids duplicate data by skipping previously ingested files.  
✅ **Schema Evolution** – Can automatically merge new columns.  
✅ **Efficient** – Faster than INSERT INTO for large-scale ingestion.

* **AUTO LOADER**: incrementally or streaming processes new data files as they arrive in cloud storage. **Automatically infers schema and accommodates schema changes,** and includes a **rescue data** column for data that does not adhere to the schema.