🚀 **Data Engineering in Action!** 🚀

Excited to share my latest project where I built a scalable **Customer-Order Management System** using **Delta Lake** in **Databricks**! 💻

Here’s a quick overview:

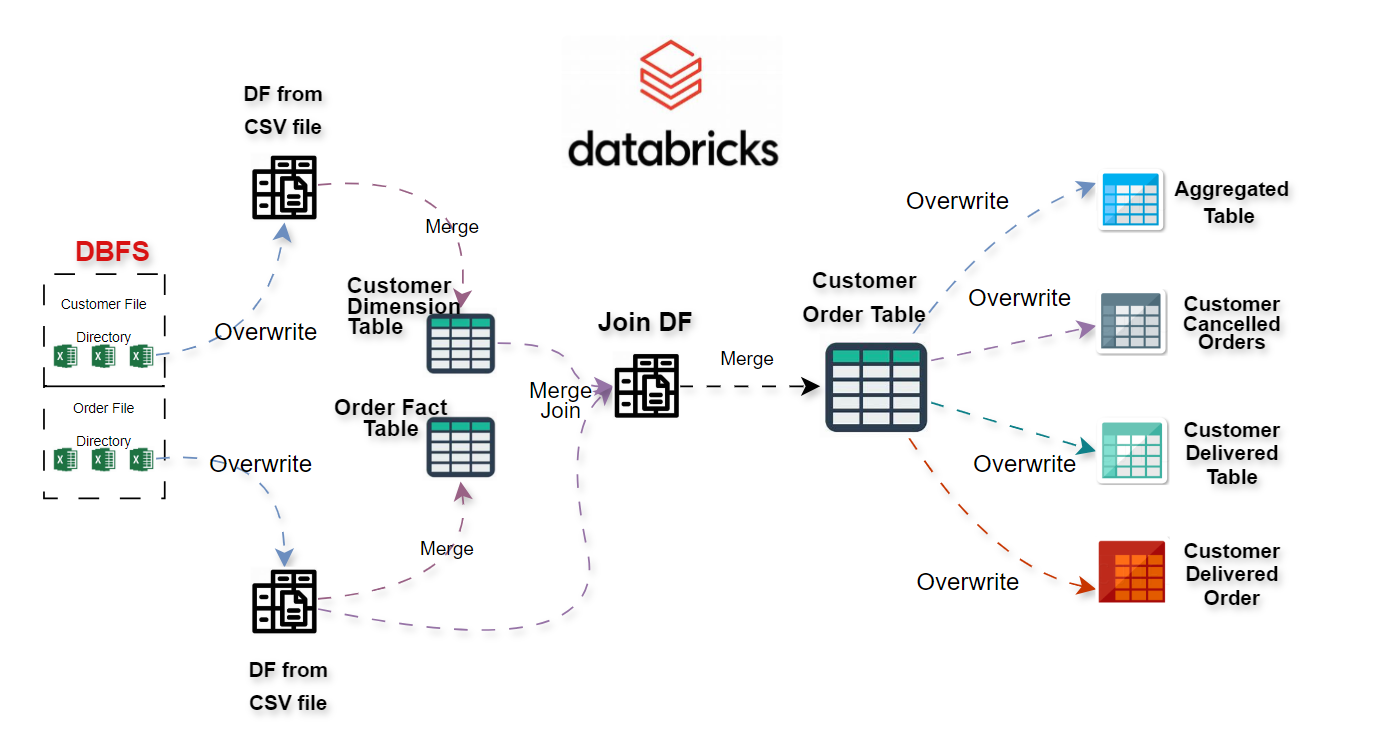
* **Customer and Order Data**: Ingesting multiple datasets from CSV files stored in DBFS. As new data will arrive to the directory our pipeline will get trigger and it will read file only for that day.
* **Dimensional Modeling**: Created **Customer Dimension** and **Order Fact** tables for efficient data management.
* **Merging and Aggregation**: Joined tables to form a **Customer Order Table**, which I further aggregated for different types of analysis. Also, I have segregated data on the basis of **Delivered, Shipped** and **Canceled Orders** to generate different reports.
* **Delta Lake Magic**: Leveraged Delta Lake for seamless updates and merges Delta tables.

This project showcases my expertise in:

✔️ Data pipelines  
✔️ Delta Lake architecture  
✔️ Real-time data updates  
✔️ Data aggregation for business insights

#DataEngineering #Databricks #DeltaLake #ETL #CustomerAnalytics #RealTimeData #BigData #Spark

OverView of this project:



In DBFS we will be dumping file on daily basis with naming convention –

dbfs:/FileStore/Ecommerce\_data\_02\_10\_24/Customer\_details\_01\_10\_24.csv

date=current\_date()

This will help to pick current date below query will pick csv file on the basis of naming convention depends on date.

customer\_df=spark.read.format('csv').option('header',True).option('inferSchema',True).load(f'dbfs:/FileStore/Ecommerce\_data\_02\_10\_24/Customer\_details\_{date}.csv')

Steps:

* From DBFS we will pick present date file and create df on it for both the files.
* We will use merge operation to update the latest data in Order fact table and Customer Dimension table. By this way every time we get latest data.
* To optimize our workflow we will use df of order and table of customer to join only incremental data and reducing load not joining fact table of Order with dimension table of Customer on daily basis.
* Now we will use this join\_df to merge the incremental data into Customer Order table.
* The from Customer Order table we would be using it as a base table to created different different reports.