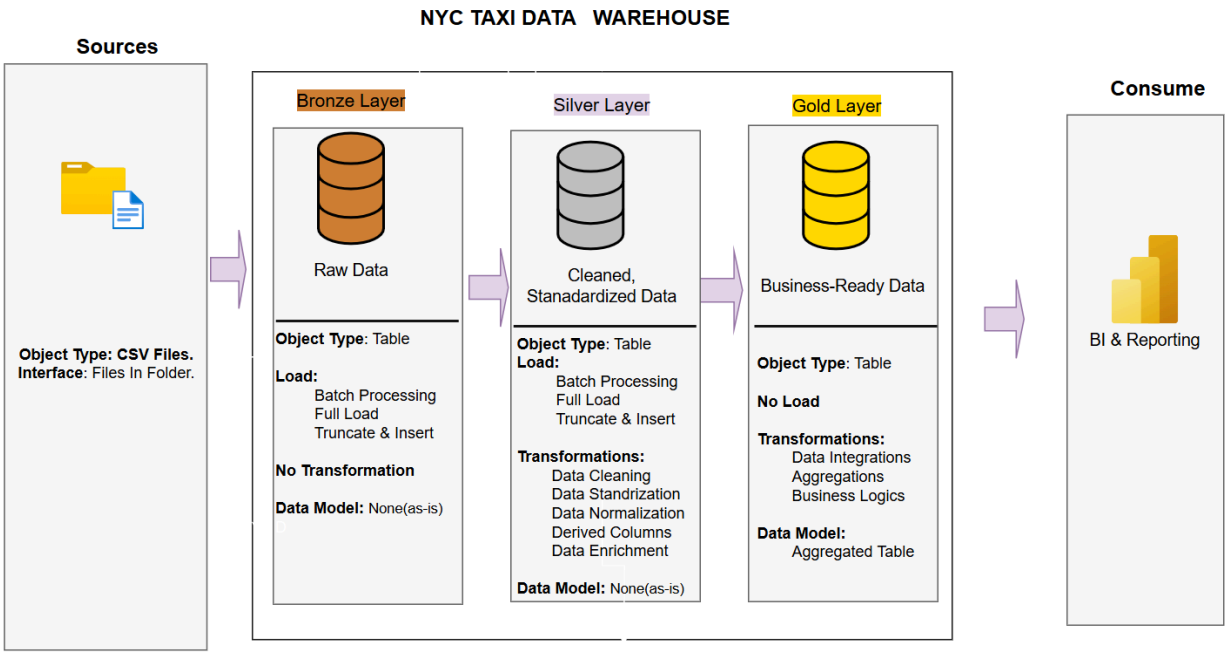
1. 

2. The sql code are available in the github.
3.
    a. DESIGN RATIONALE
    The Bronze layer stored the raw data without any changes, the silver is why the cleaning, transformation, standardization and deriving new columns that will be useful for the business is done and the gold layer is where the aggregation and summarizing of data is done.
    b. FULL VS INCREMENTAL LOAD

The full refresh has a direct solution, I create the table for the bronze layer, and upload all the csv directly with COPY FROM for my folder, i use CTAS(Create Table As) to perform the cleaning, transformation, derived new columns and inserting the data to the silver layer and CTAS also in creating 5 different table for data aggregation and summary.

The incremental load is a little tricky, after much brainstorming, I decided to use triggers on my layers, starting from the bronze layer, and also create a metadata schema this time in order to store them anytime there is any change in my bronze, silver and gold layer, it will save the time the change occur will be useful to process incoming data.

So this how, my code on incremental loading work, I create a trigger on the bronze layer that anytime a new data is inserted into my bronze layer, the silver layer will check the data, if a new data and the silver layer will do all the necessary transformation, cleaning and deriving new columns before inserting, and also another trigger is on that silver.taxi table. Anytime new data is append in the silver layer, the trigger will be triggered and the append the new data to the

gold layer, during the appending the necessary aggregation and summarization into 5 different tables will automatically be done.
The only human interaction that will occur is when you want to upload a new data, the silver and the gold will automatically happen.

c. METADATA MANAGEMENT
The metadata is done by creating the last_loading_period that will automatically update any time that is a new update in the bronze tables, silver tables and even gold tables.

d. SELECT trip_date, total_revenue
FROM gold.daily_summary
ORDER BY trip_date;

SELECT vendor_name, total_revenue, avg_fare
FROM gold.vendor_summary
ORDER BY total_revenue DESC
LIMIT 5;

SELECT payment_description, avg_tip_percent
FROM gold.payment_summary
ORDER BY avg_tip_percent DESC;

SELECT month, total_trips, total_revenue
FROM gold.monthly_summary
ORDER BY month;

SELECT PULocationID, pickups, revenue_from_pickups
FROM gold.zone_summary
ORDER BY pickups DESC
LIMIT 10;

4. The pipeline can run multiple times, without duplicates, because there is CONSTRAINT unique in the bronze layer, so the bronze layer will automatically reject any duplicates data and once the bronze layer rejects duplicates record, silver layer and gold are in safe pipeline and will avoid duplicates record.