Data Integration and Data Visualization for the calculation and Analysis of Delivery Costs.

This project involved calculating the shipping costs of thousands of deliveries for one of my clients and displaying them in an easy to read Looker Studio Dashboard that would refresh daily. The big problem that the customer company had was that they had independent information in different data sources and formats and it was difficult for them to integrate the data to be able to calculate the actual shipping cost.. For example, the data about sales and order fulfillments was in a No-SQL database, the data about product warehouse allocation in another No-SQL database and the information about the shipping costs was store in tens of different files in different formats (most of them excel, csv or google sheets) containing the invoice amounts for each delivery.
So to be able to consolidate the data and to calculate the shipping costs for the company.I created a fully comprehensive Business Intelligence solution that handled every aspect from Data Engineering to Data Visualization.

Next a quick summary of the Project.

Image of the final deliverable-Dashboard of Shipping Costs (daily refreshed):

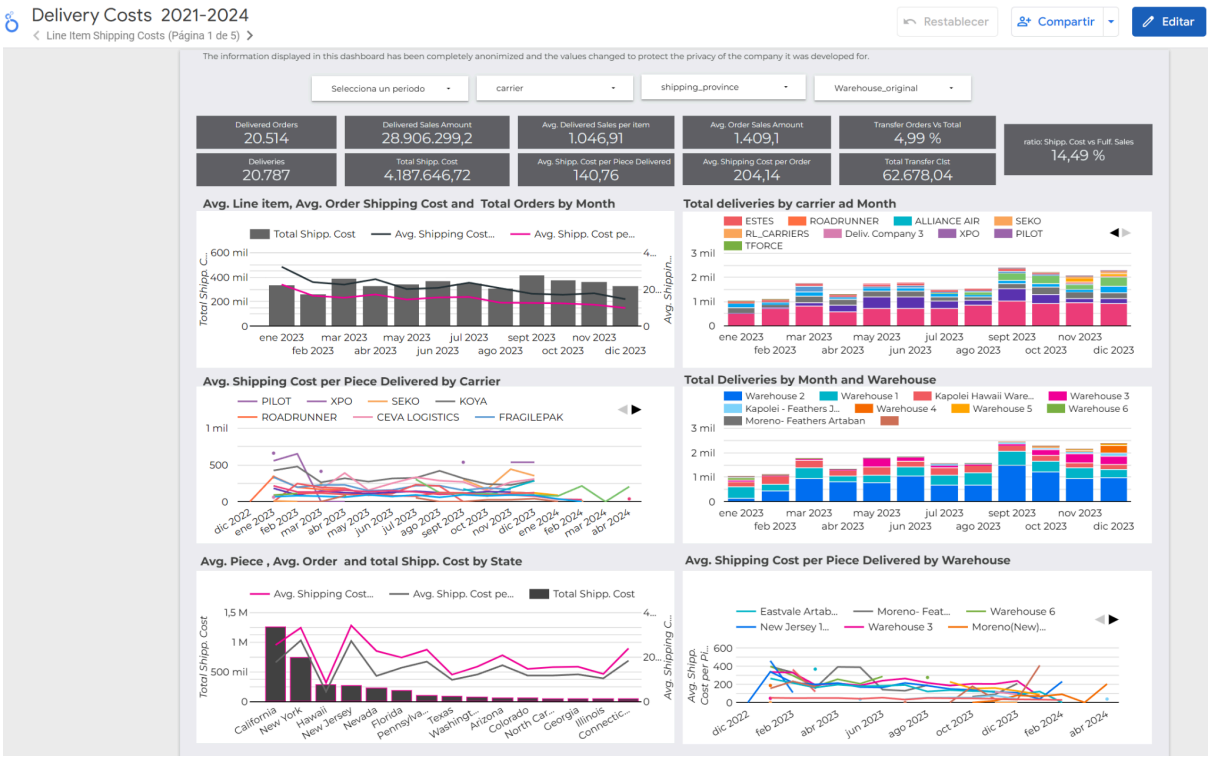
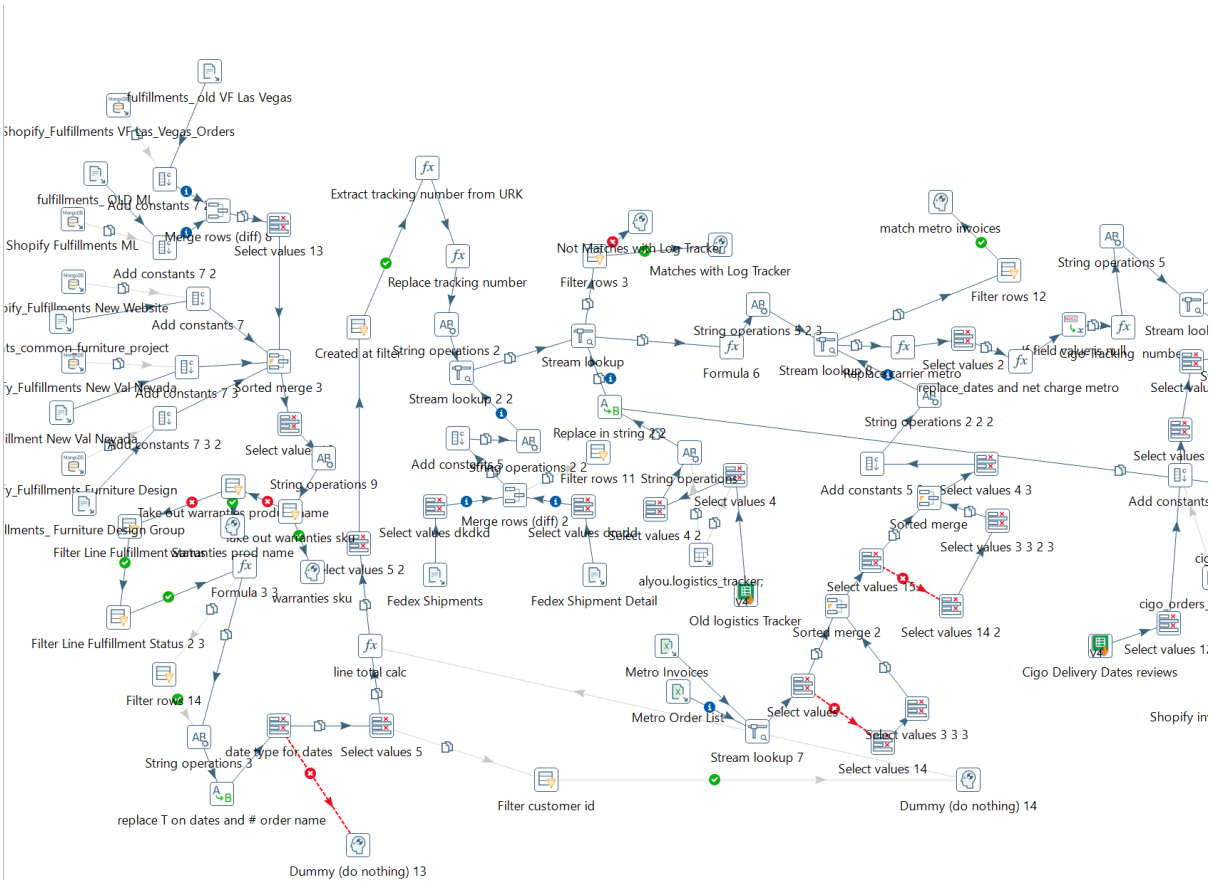


Table of contents

Data Engineering:

ETL Program with Pentaho Data Integration:

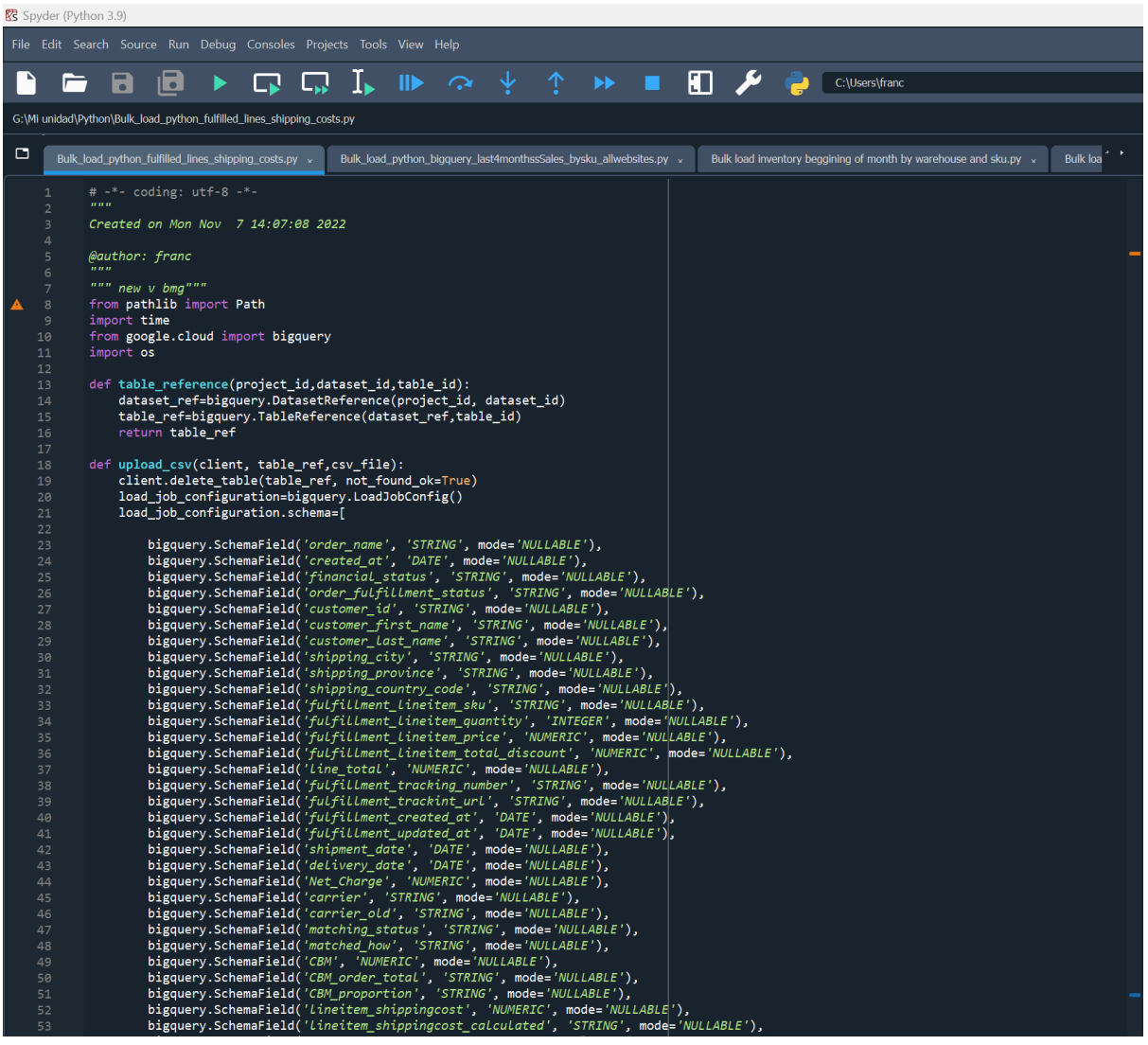
This program would allow to extract the information about the orders from different websites that was stored in MongoDB (Non structured database) and integrate it with the information about the specific cost of each delivery from different invoice files (excel, CSV Files) as well with other complementary information stored in several Google Sheet Files.



After integrating and manipulating the data as needed, the program calculates the shipping costs for each one of the items of thousands of orders.
The resulting data is then stored in a final CSV file that will later be used to copy the data and upload it into a Google Big Query DataBase that the company was using as a Data Warehouse.

Uploading of Resulting Data to Google Big Query:

The resulting CSV file from the last step is then uploaded to a specific table within a Google Big Query DataBase through the use of Python programming.




```python
# -*- coding: utf-8 -*-
"""
Created on Mon Nov  7 14:07:08 2022

@author: franc
"""
""" new v bmg"""
from pathlib import Path
import time
from google.cloud import bigquery
import os

def table_reference(project_id,dataset_id,table_id):
    dataset_ref=bigquery.DatasetReference(project_id, dataset_id)
    table_ref=bigquery.TableReference(dataset_ref,table_id)
    return table_ref

def upload_csv(client, table_ref,csv_file):
    client.delete_table(table_ref, not_found_ok=True)
    load_job_configuration=bigquery.LoadJobConfig()
    load_job_configuration.schema=[

        bigquery.SchemaField('order_name', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('created_at', 'DATE', mode='NULLABLE'),
        bigquery.SchemaField('financial_status', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('order_fulfillment_status', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('customer_id', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('customer_first_name', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('customer_last_name', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('shipping_city', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('shipping_province', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('shipping_country_code', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('fulfillment_lineitem_sku', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('fulfillment_lineitem_quantity', 'INTEGER', mode='NULLABLE'),
        bigquery.SchemaField('fulfillment_lineitem_price', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('fulfillment_lineitem_total_discount', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('line_total', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('fulfillment_tracking_number', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('fulfillment_trackint_url', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('fulfillment_created_at', 'DATE', mode='NULLABLE'),
        bigquery.SchemaField('fulfillment_updated_at', 'DATE', mode='NULLABLE'),
        bigquery.SchemaField('shipment_date', 'DATE', mode='NULLABLE'),
        bigquery.SchemaField('delivery_date', 'DATE', mode='NULLABLE'),
        bigquery.SchemaField('Net_Charge', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('carrier', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('carrier_old', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('matching_status', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('matched_how', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('CBM', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('CBM_order_total', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('CBM_proportion', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('lineitem_shippingcost', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('lineitem_shippingcost_calculated', 'STRING', mode='NULLABLE'),
```

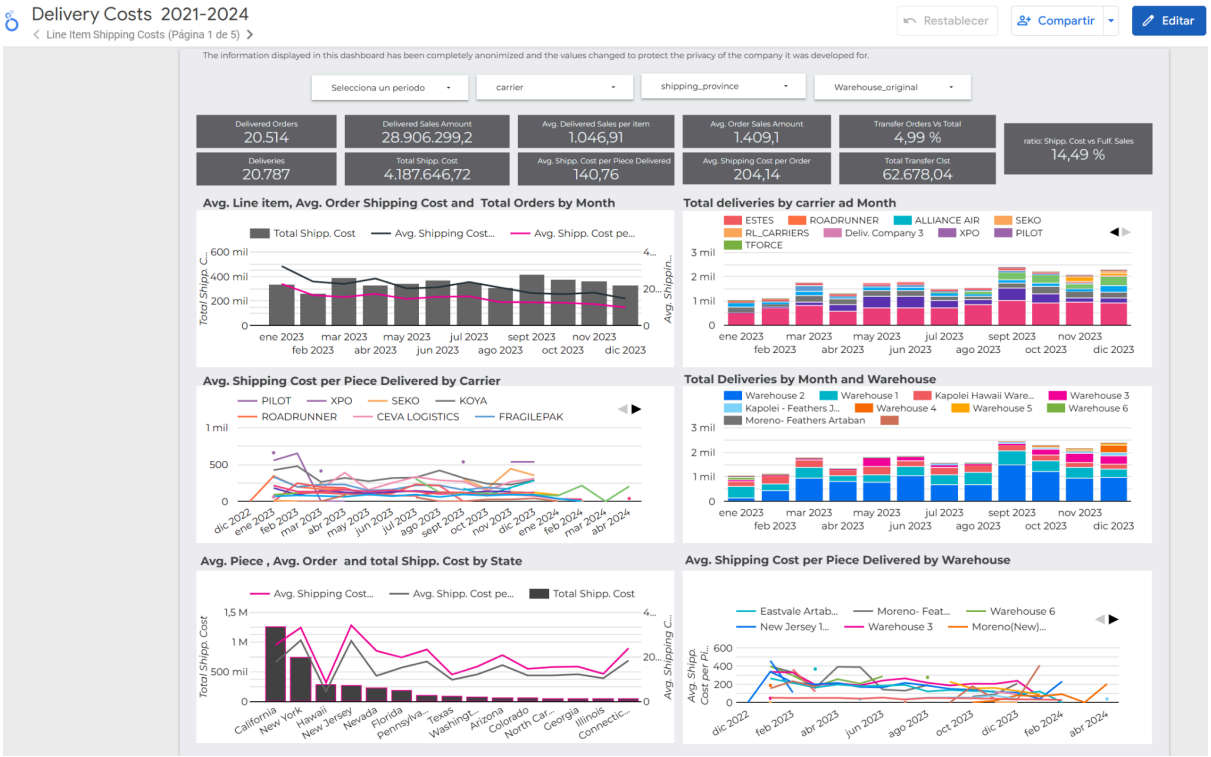
Connection and Uploading of Data into LookerStudio

Looker Studio connects to the BigQuery resulting table and uploads the data,

Data Visualization:

Creation of dashboard and metrics for the analysis of Shipping Costs with Google Looker Studio

After the connection with Big Query and the uploading of data, several metrics within Looker Studio, using its metric creation capabilities.



Impact for the company

As a result of the availability of this information the managers were able to make several informed decisions. For example, in the short term they realized that some companies were significantly more expensive than others. The information of the dashboard allowed them to get rid of very expensive suppliers and in some cases to renegotiate deals.

As another example, with the dashboard, they were able to understand which warehouses were cost effective and which were not, thus allowing them to optimize the use of them, and in some cases closing them to open in new locations. The dashboard also helped as tool for negotiations with new dealers or suppliers.