LEVEL4_Data_Pipeline_Design

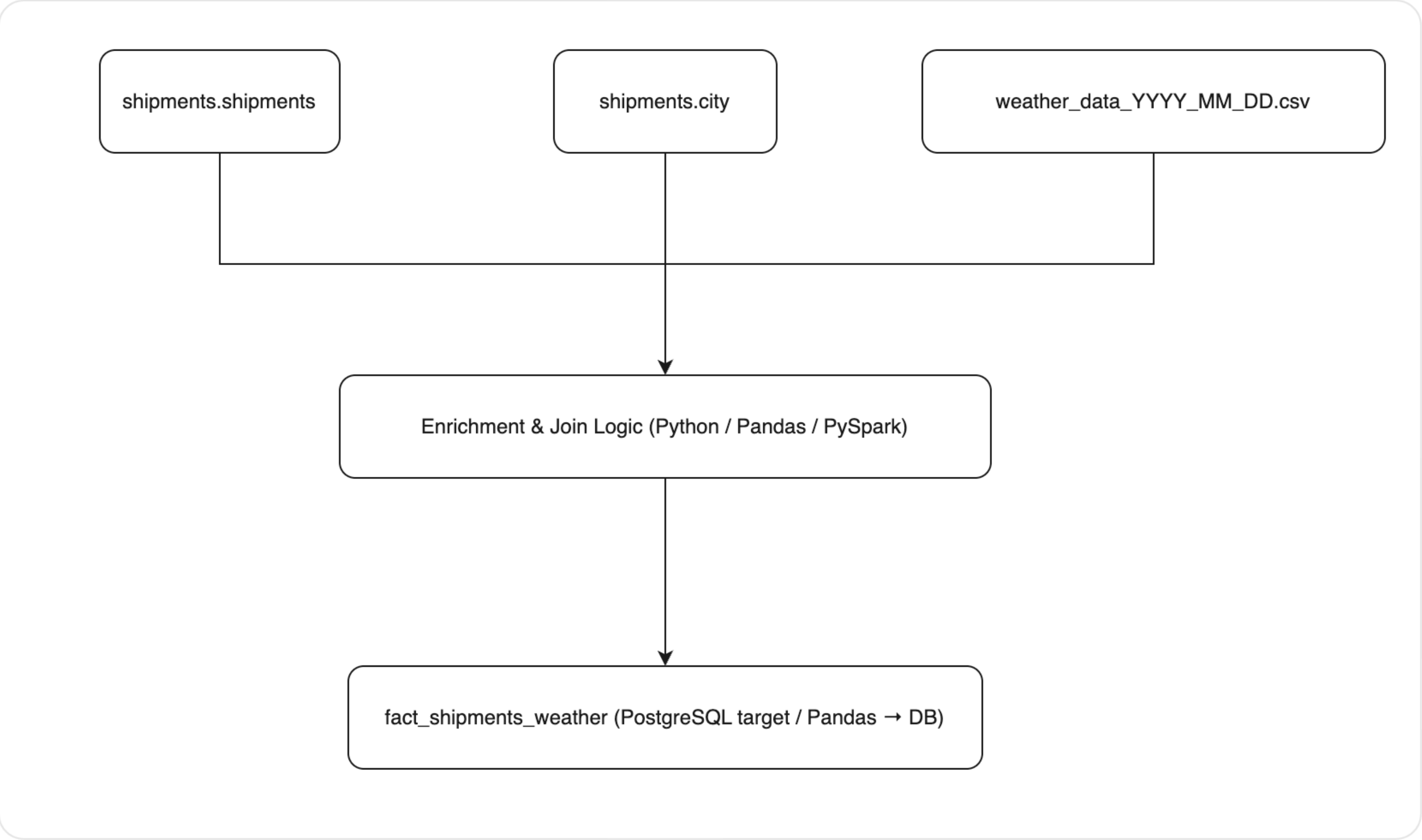
BTTF Data Engineer Assignment

Objective

Design and implement a data pipeline that:

- Joins shipment data from PostgreSQL with weather data from CSV
- Performs timestamp alignment and city-level joins
- Loads the final merged data into the fact table: `fact_shipments_weather`
- Enables downstream analysis (KPI queries)

Pipeline Architecture (Logical)


shipments.shipments
shipments.city
weather_data_YYYY_MM_DD.csv
Enrichment & Join Logic (Python / Pandas / PySpark)
fact_shipments_weather (PostgreSQL target / Pandas → DB)

- Read shipment & city data from PostgreSQL (shipments.shipments, shipments.cities)
- Read weather data from CSV (weather_data_2022_07.csv)
- Normalize city names and convert timestamps to hourly granularity
- Merge weather data with cities to assign city_id
- Join shipments with weather on city + hourly timestamp
- Output merged data to PostgreSQL table: analytics.fact_shipments_weather

---

Steps

1. Read Raw Inputs
    - PostgreSQL tables: shipments, cities
    - Local CSV: weather_data_2022_07.csv
2. Preprocessing
    - Lowercase + trim city names
    - Convert timestamps to hourly using `.dt.floor('H')`
    - Assign city_id via coordinates match
3. Join Logic
    - Merge: weather × cities (on city name, lat/lon)
    - Merge: shipments × weather (on city + hourly timestamp)

4. Output
    - CSV: /data/processed/fact_shipments_weather.csv
    - PostgreSQL: analytics.fact_shipments_weather (automated table creation)

Technical Stack Used

Step	Tool/Language
ETL Logic	Python (Pandas)
DB Reads/Writes	psycopg2 or SQLAlchemy
Logging	Python `logging` module
Output Inspection	DBeaver (PostgreSQL)
Documentation	Obsidian
Query Inspection	DBeaver

Directory Location

```
scripts/
└── processing/
└── build_fact_shipments_weather.py
```

Summary

The pipeline performs the following:

- Extracts shipments and city data from PostgreSQL
- Extracts weather data from local CSVs
- Aligns timestamps to the nearest hour
- Joins weather → city → shipments to create one enriched record per shipment
- Loads final data into fact table

Output Table Schema

```
CREATE TABLE analytics.fact_shipments_weather (
    shipment_id BIGINT,
    city_id BIGINT,
    timestamp TIMESTAMP,
    fuel_consumed_liters FLOAT,
    temperature_2m FLOAT,
    windspeed_10m FLOAT,
    precipitation FLOAT,
    weathercode BIGINT
);
```

> 💡 This fact table is the foundation for all KPI aggregations in Level 5 (Check the last section of LEVEL3_Data Modeling

---

Next Planned Action

→ Proceed to in LEVEL5_Visualization_Approach to gain more insights about the visualization ideas regarding the BTFF project