Learning Notes – Holiday & Weather Data Pipeline

# 1. Pipeline Overview

- Inputs:  
 • Open-Meteo API (weather by lat/lon).  
 • Nager.Date API (public holidays by country/year).  
  
- Process:  
 • Geocode city → lat/lon/country/region.  
 • Fetch 5 years of daily weather (chunked per year).  
 • Fetch 5 years of public holidays.  
 • Normalize into DataFrames → load into SQLite.  
 • Tables/views allow analysis of weather on holidays vs non-holidays.  
  
- Outputs:  
 • SQLite `.db` file with weather\_data, public\_holidays, and analysis views.

# 2. Issues Encountered & Fixes

a) Handling Dates  
Issue: Needed “last 5 years” from yesterday, not today.  
Fix: Used date.today() - timedelta(days=1) and relativedelta(years=5).  
  
b) ISO Region Codes  
Issue: Nominatim sometimes returned None for state\_code or address.  
Fix: Requested addressdetails=True; fallback using ISO3166-2 or state name → USPS mapping.  
  
c) Holiday API Fetching  
Issue: Only got last year (2025).  
Fix: Ensure loop encloses request; each year fetched separately.  
  
d) SQLite Schema  
Issue: Error: expressions prohibited in PRIMARY KEY and UNIQUE constraints.  
Fix: Made region\_code TEXT NOT NULL DEFAULT '', used directly in UNIQUE.  
  
e) Pandas NA Booleans  
Issue: TypeError: boolean value of NA is ambiguous.  
Fix: Added \_is\_public() helper to normalize global/types safely.  
  
f) Logging  
Issue: Log file not written where expected.  
Fix: Use absolute path with os.path.dirname(\_\_file\_\_), add force=True to basicConfig.

# 3. Improvements We Made

- Modularized code into separate files (city\_time\_interval.py, publicholiday.py, weather\_data.py, ds\_utils.py, main.py).  
- Added structured logging to pipeline.log.  
- Input validation for lat/lon ranges, years, date window.  
- Created SQLite indexes for performance.  
- Documented warehouse-style star schema.

# 4. Lessons Learned

- Design warehouse schema first to guide transforms.  
- Handle pd.NA explicitly, never bool(pd.NA).  
- SQLite constraints can't use expressions—normalize first.  
- Always use addressdetails=True with Nominatim.  
- Use absolute log file paths and force=True in logging.basicConfig.  
- Build idempotent scripts with UPSERT/IGNORE.

# 5. Next Steps

- Add unit tests for helper functions and data validation.  
- Migrate from SQLite → Delta/Parquet for scale.  
- Schedule with Airflow/cron, add monitoring and alerts.  
- Enhance logs with row counts, timings, error alerts.