Executive Summary – Holiday & Weather Data Pipeline

This mini-capstone project demonstrates the design and implementation of a data pipeline that integrates weather data with public holiday calendars. The pipeline enables analysis of weather patterns on holidays versus non-holidays, and was implemented using Python, SQLite, and external APIs.

# Objectives

- Build a modular data pipeline to extract, transform, and load (ETL) daily weather data and public holidays.  
- Create a warehouse-style schema (fact and dimension tables) to support analytical queries.  
- Implement resiliency measures for API downtime, data inconsistencies, and logging.  
- Prototype with SQLite, with a path to scale into Delta/Parquet on Databricks.

# Pipeline Overview

1. Geocode the primary city to obtain latitude, longitude, and country code.  
2. Fetch daily weather data for the last 5 years using the Open-Meteo API.  
3. Fetch public holidays for the same window using the Nager.Date API.  
4. Normalize and validate the datasets, ensuring referential integrity.  
5. Load into SQLite tables: weather\_data, public\_holidays, and analytical views.

# Key Challenges and Fixes

- \*\*Date handling\*\*: Ensured the end date was always 'yesterday' to avoid partial data.  
- \*\*Region codes\*\*: Used Nominatim's addressdetails and ISO3166-2 fallback.  
- \*\*Holiday API loop\*\*: Moved request inside loop to fetch all years.  
- \*\*SQLite schema\*\*: Removed expressions from UNIQUE constraints; normalized region\_code.  
- \*\*Boolean NA\*\*: Replaced `bool(pd.NA)` with a helper `\_is\_public()` to handle missing values.  
- \*\*Logging\*\*: Directed logs to `pipeline.log` using absolute paths and `force=True`.

# Results

The pipeline successfully produced a SQLite database with structured tables and indexes. Analysts can now run SQL queries to compare weather on holidays versus non-holidays, and aggregate results by year, city, or country.

# Next Steps

- Add automated quality checks and unit tests.  
- Schedule the pipeline using Airflow or cron with monitoring alerts.  
- Scale data storage to Delta Lake/Parquet for enterprise readiness.  
- Extend analysis with visualizations (e.g., Power BI, Tableau).