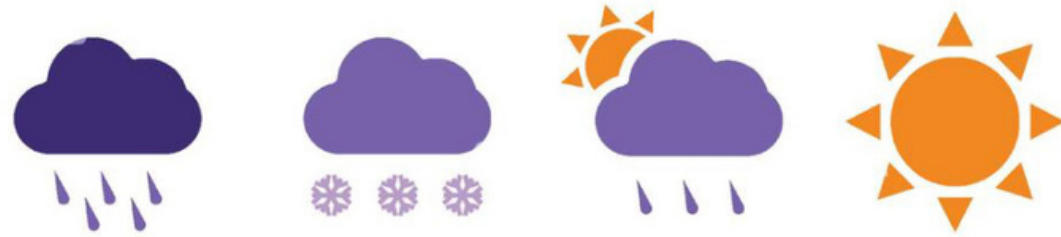


CLEARING ROADBLOCKS

Automating Washington Department of
Transportation (WDOT) Travel Alerts with
Data Engineering

The Problem

Understanding Traffic Challenges



Problem Statement:

- Traffic congestion, accidents, and bad weather make travel unpredictable
- People need real-time updates to make better travel decisions.



Current issues:

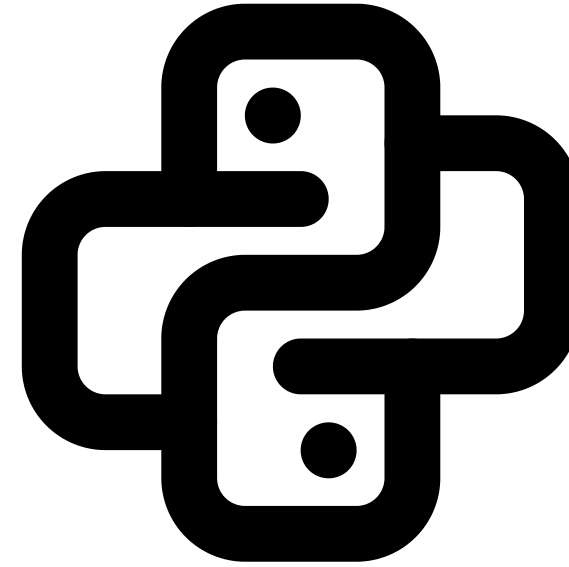
- Information is scattered across different sources.
- No automated system to combine and analyze this information.
- Inconsistent data formats & missing values made it difficult to analyze trends accurately.

System Architecture



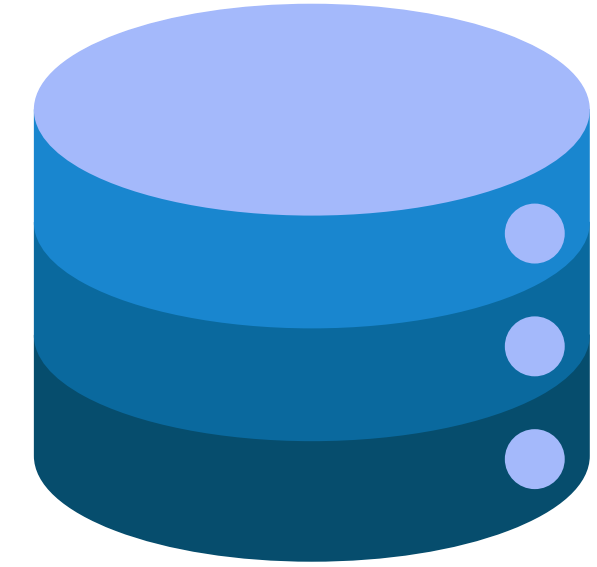
Data Sources

Public RESTful APIs from the Washington State DOT provide real-time traffic data in JSON format. Authentication is managed using API keys.



Extract

A Python script, scheduled via CRON every four hours, fetches and processes the API data into a Pandas DataFrame, handling errors as needed.



Load

The processed data is stored in a MySQL database on Azure for structured storage and future analysis.

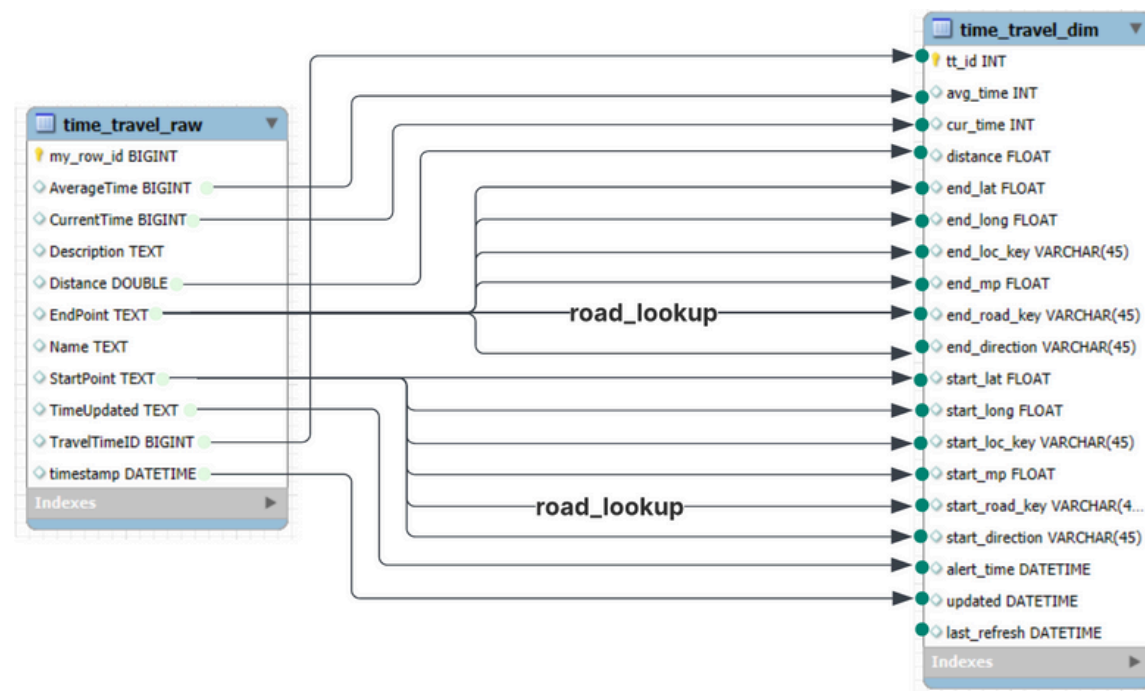
Transformation

Transformation

Created stored procedure in the mysql database. Automatically called when API scripts refresh.

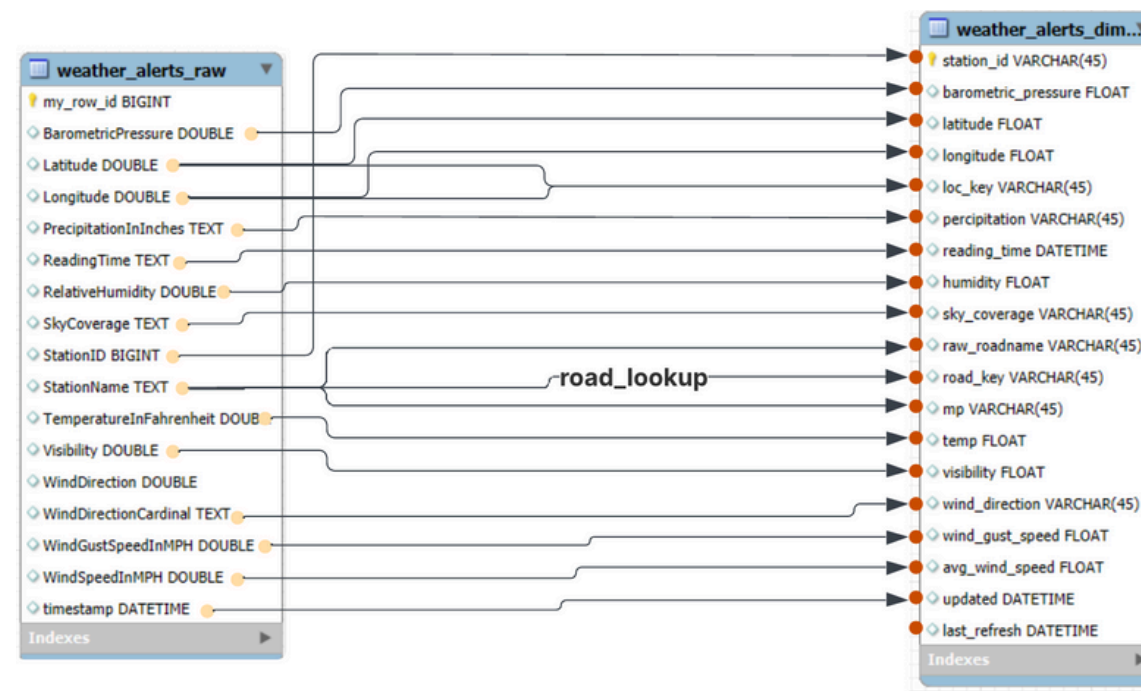
call TransformTravelTime()

time_travel_dim



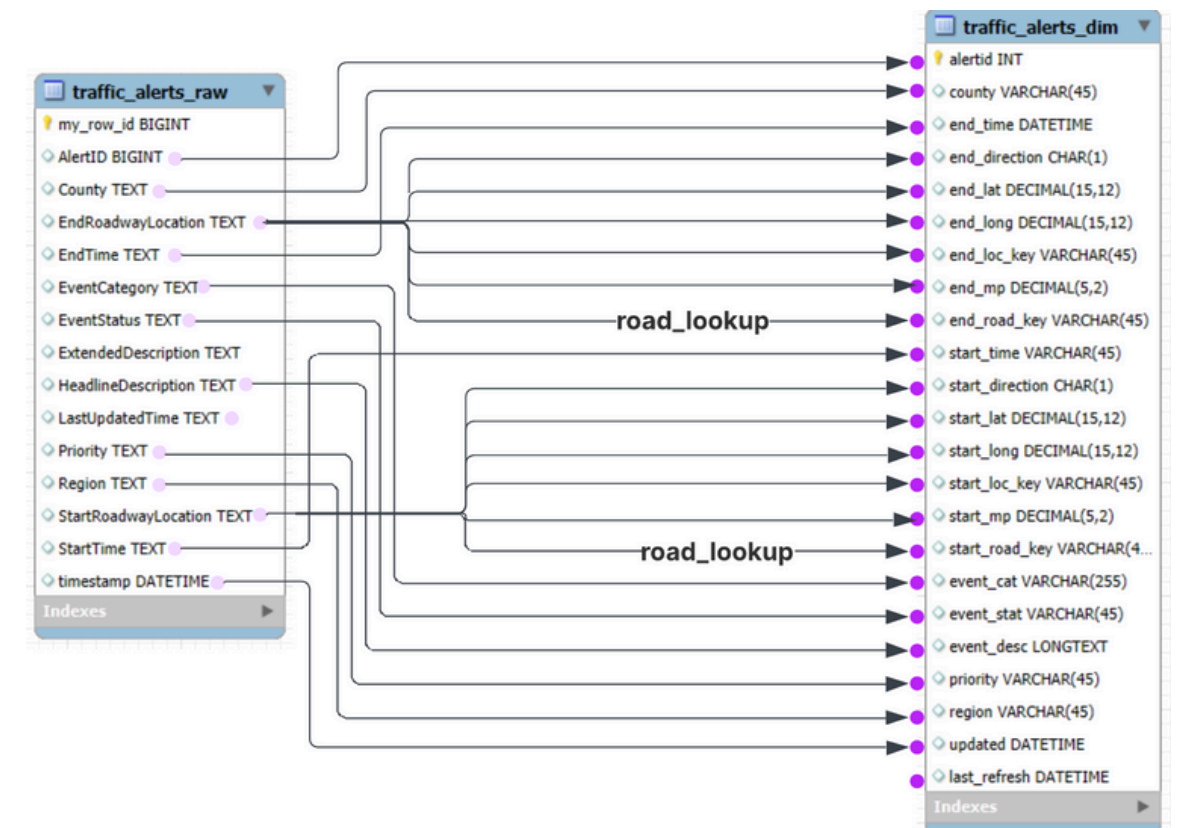
call TransformWeatherAlerts()

weather_alerts_dim



call TransformTrafficAlerts()

traffic_alerts_dim



Our Solutions: A Smarter Travel System



Automated Data Pipeline:

- Collects real-time traffic, weather, and road alerts
- Cleans and organizes data for better accuracy
- Stores structured data in a MySQL database
- Provides real-time insights through dashboards

Key Features:

- Real-time traffic and weather updates
- Structured data storage for reporting and dashboards
- Automated alerts for road incidents and weather impacts

System Monitoring

API Fetch

Logs the latest API call status to confirm successful data retrieval

API Fetch History

Stores historical API response statuses, including timestamps and HTTP response codes, allowing for long-term monitoring.

Application Logs

Tracks ELT execution, capturing timestamps, log levels, and transformation steps.



Design Document



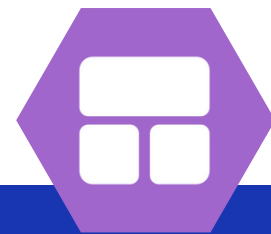
Home

Project overview, important links, basic setup instructions.



Data Dictionary

Defines dataset structure, column names, and data types.



Dashboard

User guide for Tableau dashboard



EER Diagram

Visual representation of the database schema and table relationships in MySQL



ELT Architecture

Overview of Extract, Load, Transform process and stored procedures for data processing



Monitoring

Overview of pipeline monitoring to track ETL failures and ensure data integrity

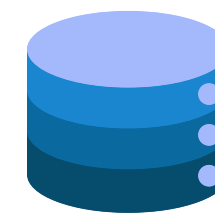
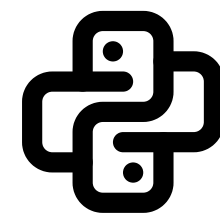


Code Review

Log of code reviews done within project

Design Document in Google Sheets

Deploying the Traffic Data Pipeline



How It Works:

- Fetches real-time traffic, weather, and alert data from the WSDOT APIs
- Processes and stores data in a MySQL database

Deployment Steps:

- Set Up the Environment -> Install dependencies (Conda or Virtual Environment)
- Configure the Database and APIs -> MySQL schema, API credentials in `.env` file.

Monitoring

- Logs track execution and detect failures
- Full setup details available on GitHub

Challenges & Lessons Learned

Challenges:

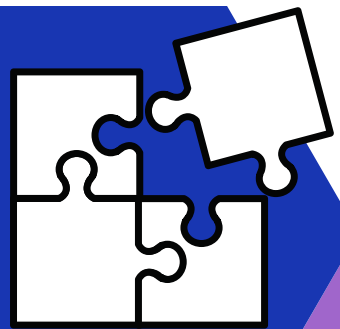
- Requires Tableau Pro or web app
- No automated alerts for failures
- Needs adjustments for major API challenges
- Lacks SSL encryption and access restrictions
- Scalability concerns with growing data

Lessons Learned

- Optimized API calls for better efficiency
- Improved data cleaning for accuracy
- Identified security and performance gaps

Future Considerations & System Gaps

- Add alerts for failures
- Plan for data growth
- Improve weather insights
- Enhance security measure



Thanks for Watching!

The following is a list of references for our work as well as important links for the project.

Project Links

- ◆ **Data Source:** <https://wsdot.wa.gov/traffic/api/>
- ◆ **Github:** <https://github.com/junclemente/ads507-finalproject>
- ◆ **Database:** ads507-finalproject.mysql.database.azure.com
- ◆ **Dashboard:** [Tableau Public](#)
- ◆ [LucidChart Architecture Diagram](#)
- ◆ [Design Doc](#)

References

- ◆ Washington State Department of Transportation (n.d). Traveler Information API. Retrieved from <https://wsdot.wa.gov/traffic/api/>
- ◆ Microsoft (n.d.) Azure. Retrieved from <https://azure.microsoft.com/en-us/>
- ◆ Reis, J., & Housley, M. (2022). Fundamentals of data engineering. O'Reilly Media
- ◆ Beaulie, A. (2020). Learning SQL. O'Reilly Media.