Boston 311 Service Request GCP Data Warehousing Project

Introduction

The Boston 311 Service Request system provides a centralized platform for handling non-emergency city service requests. This project focuses on preparing raw 311 request data from Boston for advanced analytics by building a robust ETL (Extract, Transform, Load) pipeline. The goal is to process historical and real-time service request data efficiently, enabling seamless reporting and analysis.

The pipeline processes millions of records and leverages **Apache Beam** for distributed data processing, **Google Cloud Dataflow** for execution, and **BigQuery** as the **Data Warehouse**. This ensures structured, high-performance analytics capabilities.

Links

Dataset: Boston 311 | Boston.gov

GitHub for Previous Implementation on Local: amey379/311_Request_Analysis

GitHub for GCP: amey379/GCPDataWarehousingProject

Tools

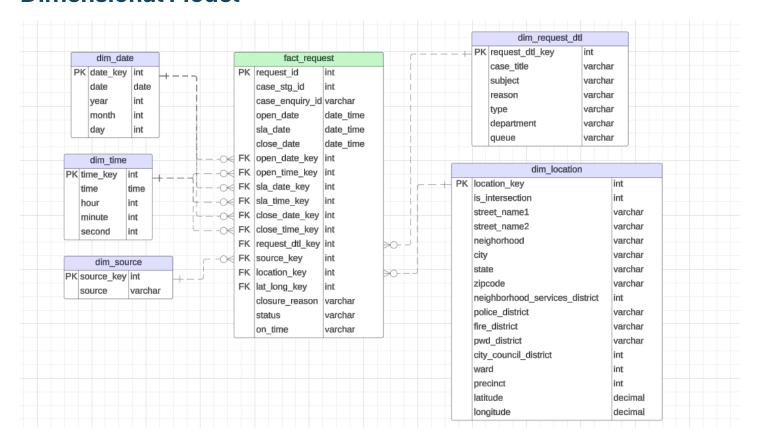
Storage:	Development & System:
BigQuery – Data Warehouse for structured storage	Apache Avro & Parquet – Data serialization and
Google Cloud Storage (GCS) – Staging and	storage formats
intermediate data storage	Java – Programming language for ETL processing
	GCP VM (Unix) – Environment setup & scripting
Orchestration & Workflow Management:	Visualization & Reporting:
Google DataFlow – ETL pipelines and	Power BI – Dashboarding & data visualization
Orchestration	
Orchestration Apache Beam – ETL pipelines and Orchestration	

Skills

Dimensional Modeling & Data Warehousing BigQuery Optimization & Performance Tuning

ETL Development (Extract, Transform, Load) Apache Beam & Google Dataflow

Dimensional Model



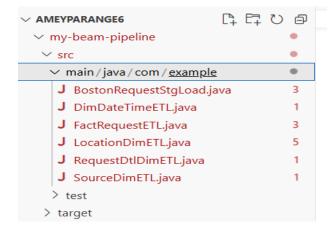
Fact Table:

Fact_311_Requests – Stores key metrics like resolution time, case status, request type

Dimension Tables:

- Dim_Date Dim_Time Date and time details for trend analysis
- Dim_Location Geographic information of incidents
- Dim_Request_Details Categories and types of 311 requests
- Dim_Source Channel through which requests were received

Files:



Technical Implementation:

Data Ingestion Layer (Bronze Layer - Raw Data Storage)

- Source: Boston 311 Service Requests (Parquet in GCS)
- Bronze Layer: Store raw CSV files in Google Cloud Storage (GCS) without modifications
- Extract raw 311 data from GCS (Bronze Layer) and load it into Apache Beam (Dataflow) for processing

Data Processing & Transformation Layer (Silver Layer - Cleaned Data)

- Silver Layer: Apply cleaning, deduplication, and standardization in Apache Beam
- ETL Pipelines (Apache Beam Dataflow) process and enrich data
- Handle missing values, data type inconsistencies, and data standardization
- Apply deduplication and ensure incremental processing
- Stored Data in Parquet Files.

Data Storage & Query Layer (Gold Layer - BigQuery Data Warehouse)

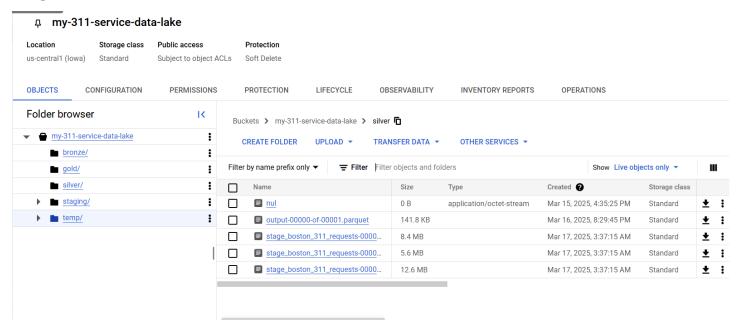
- Gold Layer: Load cleaned and structured data into BigQuery for analytics
- Convert raw data into structured dimension tables (dim_date, dim_time, dim_location, dim_source, dim_request_dtl)
- Optimize table schema with indexing and partitioning for query performance
- Store Fact Table (fact_311_requests) to support analytical queries

Performance Optimization

- Used Apache Avro & Parquet for optimized storage.
- Implemented batch processing for efficient large-scale data ingestion.
- Applied partitioning and clustering in BigQuery for improved query performance.

Output:

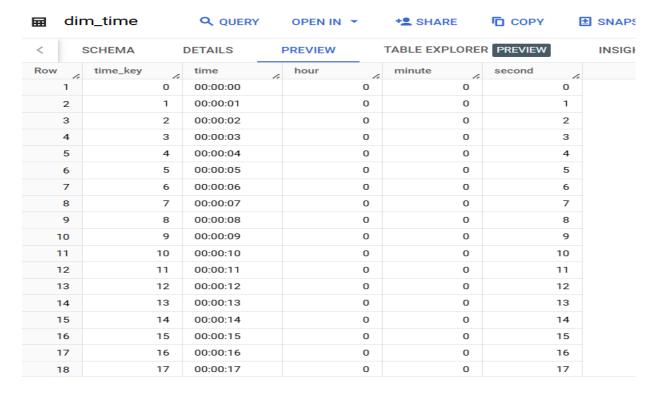
Stage_load



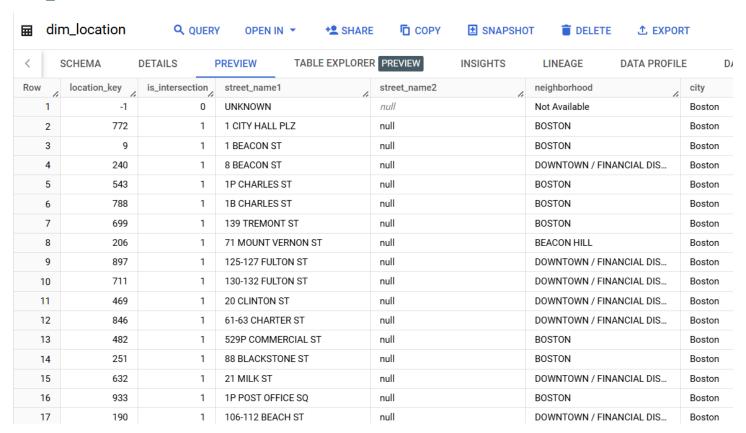
Dim_date

= (dim_date	Q QUERY	OPEN IN 🕶	+ SHARE	СОРУ	SNAF
<	SCHEMA	DETAILS	PREVIEW	TABLE EXPLORER	PREVIEW	INSIC
Row	date_key	date	year	month //	day	
1		2015-01-01	2015	1	1	
2	20150102	2015-01-02	2015	1	2	
3	20150103	2015-01-03	2015	1	3	
4	20150104	2015-01-04	2015	1	4	
5	20150105	2015-01-05	2015	1	5	
6	20150106	2015-01-06	2015	1	6	
7	20150107	2015-01-07	2015	1	7	
8	20150108	2015-01-08	2015	1	8	
9	20150109	2015-01-09	2015	1	9	
10	20150110	2015-01-10	2015	1	10	
11	20150111	2015-01-11	2015	1	11	
12	20150112	2015-01-12	2015	1	12	
13	20150113	2015-01-13	2015	1	13	
14	20150114	2015-01-14	2015	1	14	
15	20150115	2015-01-15	2015	1	15	
16	20150116	2015-01-16	2015	1	16	
17	20150117	2015-01-17	2015	1	17	
18	20150118	2015-01-18	2015	1	18	

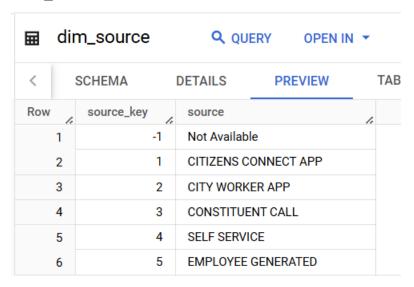
Dim_time



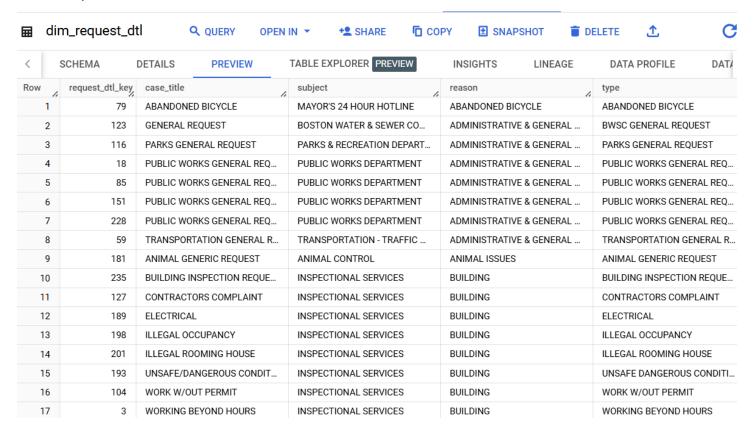
Dim_location



Dim_source



Dim_request_dtl



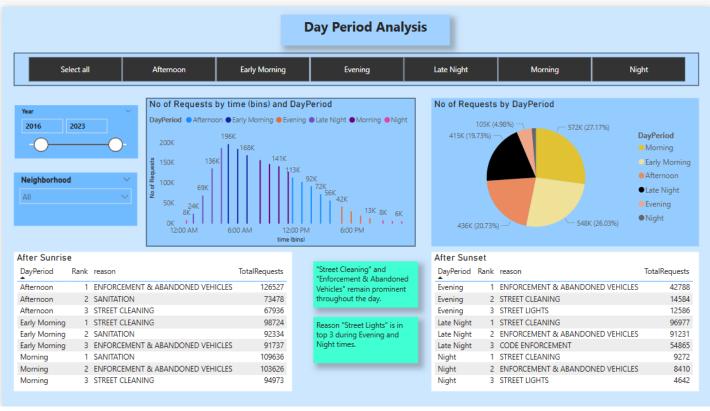
Fact

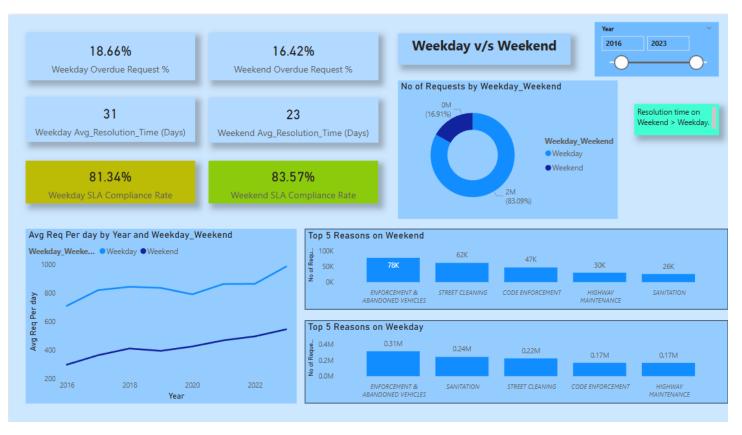
⊞ f	act_request	Q QUERY	OPEN IN 🔻	* SHARE	COPY	■ SNAPSHOT	DELETE	 ≜ EXPORT
<	SCHEMA	DETAILS PR	REVIEW TA	BLE EXPLORER P	REVIEW	INSIGHTS	LINEAGE	DATA PROFILE
Row	request_id	ca case_enquiry_	id	open_date	sla_date		close_date	6
1	728	101005384842	2	1970-01-01	1970-01-01T	07:05:51	null	
2	875	101005654998	8	1970-01-01	1970-01-01T	06:28:39	1970-01-01T07	':32:29
3	521	101005633434	4	1970-01-01	1970-01-01T	11:22:41	null	
4	493	10100567152	5	1970-01-01	1970-01-01T	09:34:56	1970-01-01T10	:24:18
5	732	10100537892	1	1970-01-01	1970-01-01T	09:49:56	null	
6	432	10100535145	8	1970-01-01	1970-01-01T	10:34:23	null	
7	169	101005571512	2	1970-01-01	1970-01-01T	11:44:20	1970-01-01T04	:06:54
8	553	10100562434	0	1970-01-01	1970-01-01T	15:56:54	1970-01-01T05	:07:01

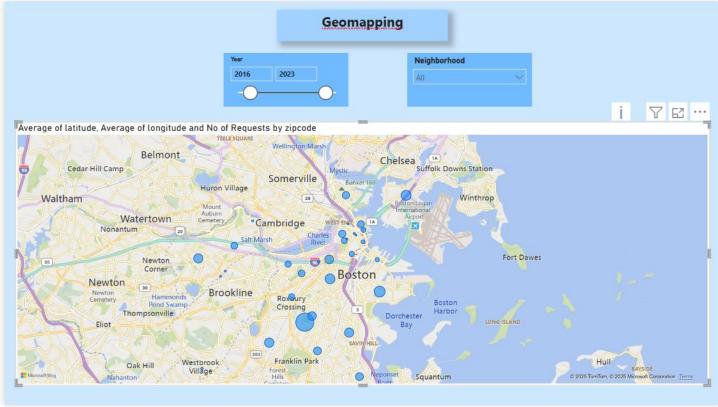
Data Visualization











Future Scope

- Delta Lakehouse Implementation
- Enhanced Airflow Orchestration
- Microservices & Real-Time Processing