Development of an Emergency Response Dashboard and Associated Infrastructure - Final Report (Group 2)

Antoun, Fadi[1], Fishman, David[2], Haughton, Laurie[3], Jung, Myunghee[4], and McLennan, Dan[5]

Emails: [1]fadi.antoun.95@gmail.com, [2]davjfish@gmail.com, [3]fadi.antoun.95@gmail.com, [4]mundlejung@gmail.com, [5]DanJMcLennan@gmail.com

December 2025

Contents

1 Objectives

1.1 Rationale behind the analysis

Effective emergency response is paramount to public safety, yet emergency services are often collected and monitored by disparate systems. Understanding trends and patterns across different jurisdictions and agencies can help decision-makers and public administrators make better decision about resource allocation. The challenge that this project is addressing is converting a high volume of raw, unstructured call data, such as that depicted in [1], into meaningful, operational intelligence.

In this project, we will discuss different approaches for data aggregation and storage and will outline some examples of ways these data can be shared with stakeholders and administrators, alike.

1.2 Structured Data in a SQL Database

Storing the emergency response data in a relational database offers several critical advantages for building an effective and reliable dashboard. The primary rationale is centered on data integrity, efficient analysis, and compatibility with standard business intelligence tools, specifically, web and mobile applications that stakeholders will want to use to view and interact with the data.

There are several key reasons why a SQL database is an appropriate choice for this project. First, most call data will have a predicable and well-structured format, e.g., incident type, timestamp, location (zip code, township), latitude and longitude. Data without these basic attributes are going to be less useful to stakeholders. Next, data organized into a relational database can be stored much more effectively than a two-dimensional dataframe (assuming a well-designed schema and sufficient normalization). A relational database management system (RDBMS) will also give administrators the ability to implement constraints in order to maximize data integrity. This includes adding lookup tables (i.e, foreign key constraints) and enforcing uniqueness, either within a column (e.g., should only have a single entry for a zipcode in a ZipCode table) or even between columns (e.g., combinations of 'city', 'state' and 'country' should be unique in an AdministrativeArea table).

Finally, a SQL database is highly compatible with REST APIs, and the two technologies are commonly used together in modern application architecture. A REST API would allow us to implement reactive tools for viewing and interacting with the data across a variety of platforms, for example,

browsers, alert-systems or mobile applications.

1.3 Unstructured Data in a NoSQL Database

NEED SOME HELP HERE!

Figure 1: An example image for our project.

2 Data Preparation

2.1 Emergency - 911 Calls from Kaggle

The principle dataset used for the project was sourced from Montgomery County, Pennsylvania and is accessible from kaggle [1]. This structured, tabular dataset is provided as a flat CSV file and consists of over 600,000 records of emergency calls from December 2015 to April 2020.

After an initial inspection of the data using exploratory tools from Pandas [**?**], a SQL schema of five tables was devised. The five tables and their descriptions are as follows:

- Category - categorical descriptions of the types of calls received (e.g., car accident)
- Township - township name and state (e.g., Kings Township, PA)
- ResponseUnit - complete list of units responding to emergency calls (e.g., Station 123, EMS)
- ResponseType - the response unit type (e.g., EMS, Traffic, or Fire)
- EmergencyCall - this is the primary data table and used to store information about emergency calls received. It has several links to the above tables.

2.2 Data quality and procurement?

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

3 Analysis

3.1 Methodology

4 Conclusions

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

5 Appendix 1: SQL Schema of Emergency Response Database

```
CREATE TABLE IF NOT EXISTS "EmergencyCall" (
    "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
    "datetime" datetime NOT NULL,
    "address" text NULL,
    "latitude" real NOT NULL,
    "longitude" real NOT NULL,
    "category_id" bigint NOT NULL REFERENCES
        "Category" ("id") DEFERRABLE INITIALLY DEFERRED,
    "response_unit_id" bigint NOT NULL REFERENCES
        "ResponseUnit" ("id") DEFERRABLE INITIALLY DEFERRED,
    "zip_code" smallint NULL, "township_id" bigint NULL REFERENCES
        "Township" ("id") DEFERRABLE INITIALLY DEFERRED
);

CREATE INDEX "emergencycall_category_id_28afcc20"
    ON "EmergencyCall" ("category_id");

CREATE INDEX "emergencycall_response_unit_id_f0a9566e"
    ON "EmergencyCall" ("response_unit_id");

CREATE INDEX "emergencycall_township_id_c7779d84"
    ON "EmergencyCall" ("township_id");

CREATE TABLE IF NOT EXISTS "Township" (
    "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
    "state" varchar(10) NOT NULL,
    "name" varchar(255) NOT NULL
);

CREATE UNIQUE INDEX "township_state_name_a30a5e69_uniq"
    ON "Township" ("state", "name");
```

```
CREATE TABLE IF NOT EXISTS "ResponseUnit" (
    "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
    "station_name" varchar(255) NOT NULL,
    "response_type_id" bigint NOT NULL REFERENCES
        "ResponseType" ("id") DEFERRABLE INITIALLY DEFERRED
);

CREATE UNIQUE INDEX "responseunit_response_type_id_station_name_25efee89_uniq"
    ON "ResponseUnit" ("response_type_id", "station_name");

CREATE INDEX "responseunit_response_type_id_21e2bd85"
    ON "ResponseUnit" ("response_type_id");

CREATE TABLE IF NOT EXISTS "Category" (
    "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
    "name" varchar(255) NOT NULL UNIQUE
);

CREATE TABLE IF NOT EXISTS "ResponseType" (
    "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
    "name" varchar(255) NOT NULL UNIQUE
);
```

References

[1] Mike Chirico. Emergency - 911 calls, 2020.