CS513: Theory & Practice of Data Cleaning

**Final Project Phase 1 Report**

**Team 23**

Avinash Baldeo, Ashley George, Sotheara Chea

[abaldeo2@illinois.edu,](mailto:abaldeo2@illinois.edu) [ageorge8@illinois.edu](mailto:ageorge8@illinois.edu), [chea3@illinois.edu](mailto:chea3@illinois.edu)

**Table of Contents**

[**1. Dataset Overview** 3](#_Toc138694840)

[**2. Dataset Description** 3](#_Toc138694841)

[2.1 Full Data Narrative 3](#_Toc138694842)

[2.2 Database Diagram & Schema 5](#_Toc138694843)

[**3. Use cases** 8](#_Toc138694844)

[3.1 U0: Zero Cleaning Use Case 8](#_Toc138694845)

[3.2 U1: Main Use Case 8](#_Toc138694846)

[3.3 U2: Never Enough Use Case 8](#_Toc138694847)

[**4. Data Quality Problems** 9](#_Toc138694848)

[List obvious data quality problems 9](#_Toc138694849)

[Why data cleaning is necessary for main use case U1 10](#_Toc138694850)

[5. Phase-II Initial Plan 10](#_Toc138694851)

# **1. Dataset Overview**

For this project our group is using the Chicago food inspection dataset which is originally released on Kaggle by the City of Chicago: <https://www.kaggle.com/datasets/chicago/chicago-food-inspections>

The Chicago Department of Public Health's dataset contains information from restaurant inspections since January 1, 2010. As per the description given, the inspections are standardized and conducted by the Food Protection Program staff. The results get input into a database, reviewed, and approved by a Licensed Environmental Health Practitioner. The dataset provided includes a subset of the data elements extracted from the database. A disclaimer is given that the dataset on food inspections may contain duplicates.

# **2. Dataset Description**

## 2.1 Full Data Narrative

The food inspections ensure food safety in licensed establishments such as restaurants, grocery stores, and bakeries. The Chicago Department of Public Health (CDPH) conducts these science-based inspections of food establishments, promoting food safety, sanitation, and preventing food-borne illnesses. The inspections cover food handling, temperatures, hygiene, facility maintenance, and pest control.

Food establishments undergo annual and complaint-based inspections for compliance with City ordinances. Inspections are conducted by the Health Department for sanitation, Buildings Department for structural safety, and Fire Department for fire exits. The City's Dumpster Task Force also checks compliance with sanitation regulations.

The dataset is maintained using Socrata's API (Application Programming Interfaces) and Kaggle's API, and the data source is the City of Chicago Data Portal

Uncompressed, the dataset size is 176 MB. In total there are 17 columns and 153,810 records ranging from 01/04/2010 to 08/28/17.

The table below gives a brief description of each column in the food\_inspection.csv file.

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Column Type** | **Description** |
| Inspection ID | integer | A unique number identifying the inspection occurrence |
| DBA Name | string | Stands for “Doing Business”, it is the legal name of the registered food establishment. |
| AKA Name | string | Stands for “Also Known As”, it is the publicly known name of the food establishment. |
| License # | integer | A unique license number assigned to the establishment for legal purposes by Department of Business Affairs and Consumer Protection |
| Facility Type | string | Describes the type/category of the establishment such as a bakery, restaurant, grocery store, etc. |
| Risk | string | The establishments’ risk level of adversely affecting public health (1 being the highest and 3 the lowest risk). Higher risk is inspected more frequently. |
| Address | string | The full street address of the establishment. |
| City | string | The city where the establishment is located. |
| Zip | integer | The zip code associated with the address. |
| Inspection Date | string | The date when the food inspection occurred. |
| Inspection Type | string | The type of inspection performed such canvass consultation, complaint, etc. |
| Results | string | Indicates whether the inspection passed, passed with conditions, or failed. |
| Violations | string | List of distinct health violations (45 distinct types) with descriptions, found during the inspection |
| Latitude | float | The GPS latitude coordinate of the establishment location |
| Longitude | float | The GPS longitude coordinate of the establishment location |
| Location | string | The GPS point coordinate (latitude, longitude) of the establishment location |

Table 1 – Food Inspection Dataset Schema Description

## 2.2 Database Diagram & Schema

The following database diagram(s) represents a better designed & normalized view of the dataset with foreign key constraints enforcing the referential integrity and maintaining the relationships of the original data.

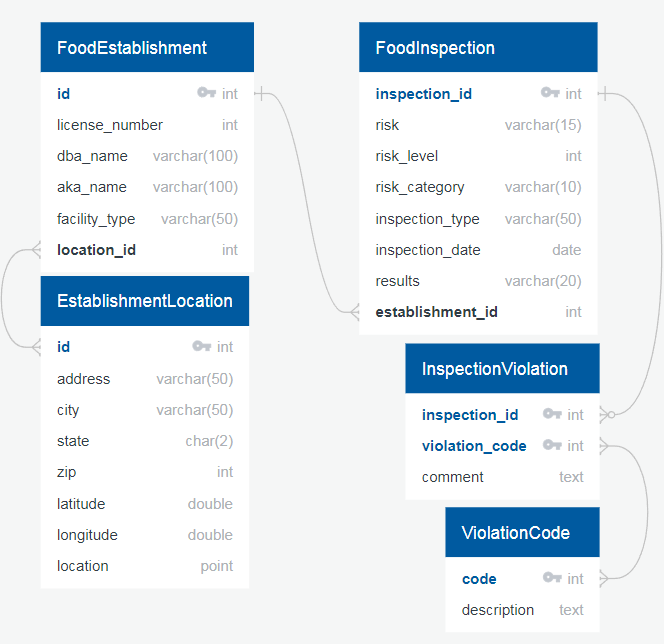


Figure 1A – Entity-Relationship (ER) Diagram of the normalized data

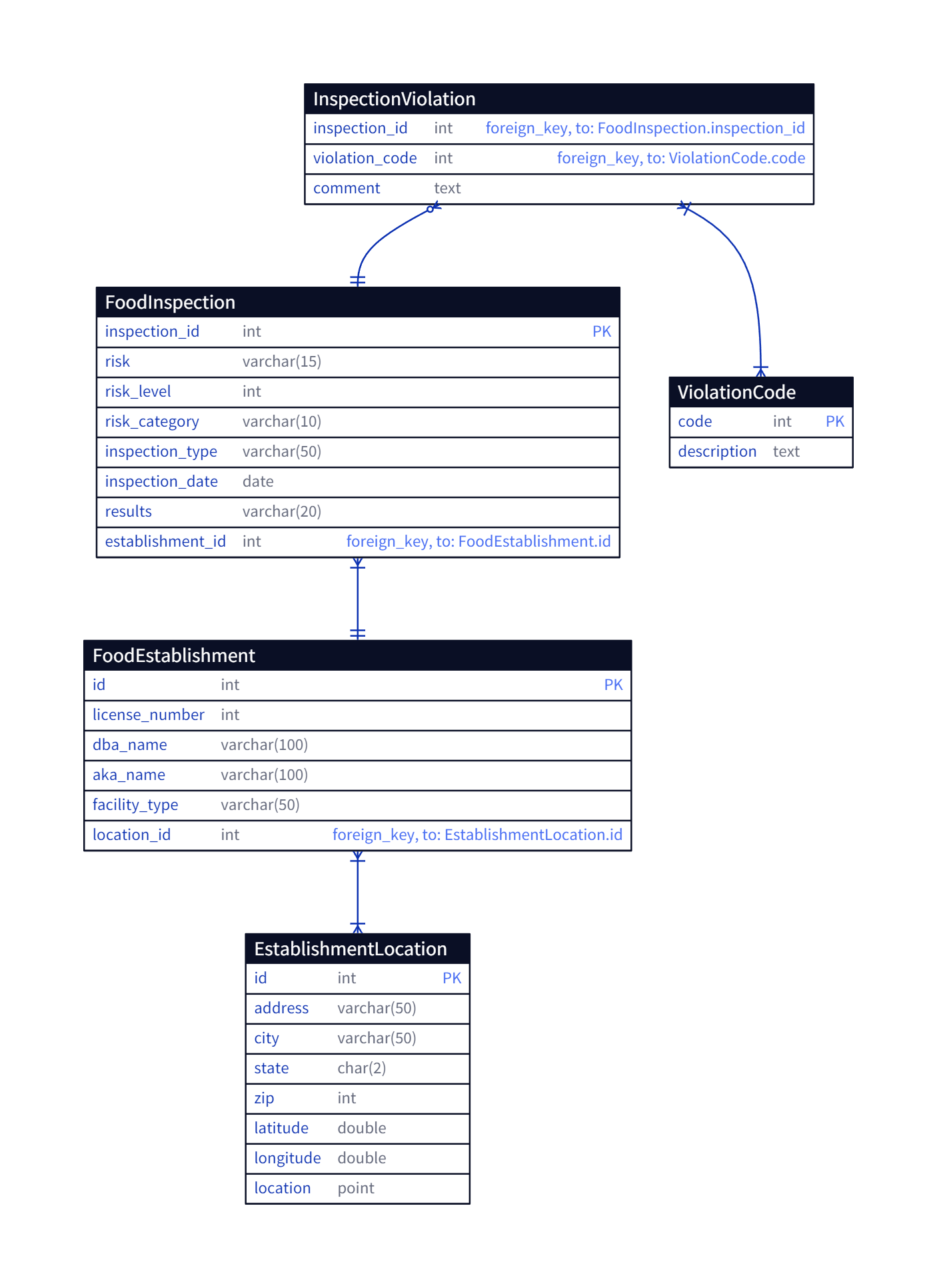


Figure 1B – Entity-Relationship (ER) Diagram with foreign keys

**CREATE TABLE EstablishmentLocation (**  
 **id SERIAL PRIMARY KEY,**  
 **address VARCHAR(50),**  
 **city VARCHAR (50),**  
 **state CHAR(2),**  
 **zip INT,**  
 **latitude DOUBLE PRECISION,**  
 **longitude DOUBLE PRECISION,**  
 **location point**  
 **);**  
  
**CREATE TABLE FoodEstablishment (**  
 **license\_number INT PRIMARY KEY,**  
 **dba\_name VARCHAR(100),**  
 **aka\_name VARCHAR(100),**  
 **facilityType VARCHAR(50),**  
 **location\_id INT REFERENCES EstablishmentLocation(id)**  
 **);**  
  
**CREATE TABLE FoodInspection (**  
 **inspection\_id INT PRIMARY KEY,**  
 **risk VARCHAR(15),**  
 **inspection\_type VARCHAR(50),**  
 **inspection\_date DATE,**  
 **license\_number INT REFERENCES FoodEstablishment(license\_number)**  
 **);**  
  
**CREATE TABLE ViolationCode (**  
 **code INT PRIMARY KEY,**  
 **description TEXT**  
 **);**  
  
**CREATE TABLE InspectionViolation (**  
 **inspection\_id INT REFERENCES FoodInspection(inspection\_id),**  
 **violation\_code INT REFERENCES ViolationCode(code),**  
 **PRIMARY KEY (**  
 **inspection\_id,**  
 **violation\_code**  
 **)**  
 **);**

# **3. Use cases**

## 3.1 U0: Zero Cleaning Use Case

To provide neighborhood-based insights on relative food safety. The data set already contains the name, location, results of the inspections and associated risk are available.

* Geographic distribution by inspection result
* Geographic distribution based on inspection risk
* Summary of inspections per timeframe (year/month) and counts by results.

## 3.2 U1: Main Use Case

Consumers choose food establishments based on several factors. Food safety is one of, if not the most important factor of it. We attempt to provide insights based on past food safety inspection violation types, their frequency, and their severity for consumers. All violations cited are included in the same column now. Comments are provided on the details of the violation too in the same column. The main use case is to provide insights and suggestions to the consumer public based on violations and their trends and distribution. Below are some examples

* Violation type distribution by the facility types
* Violation types by certain popular chains and brands.
* Violation type distribution by neighborhood
* Time-based trends on violation types
* Overall safety insights by facility types

## 3.3 U2: Never Enough Use Case

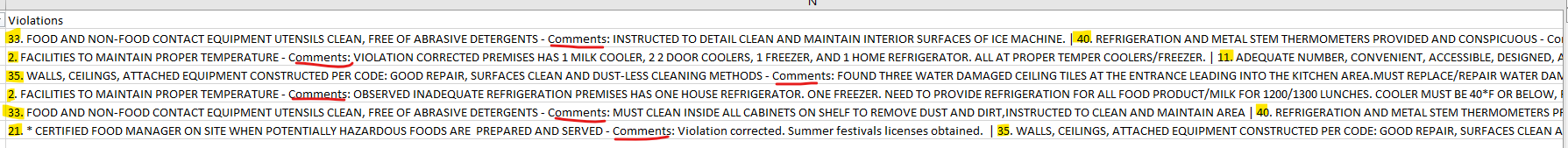
Be able provide overall recommendations to consumers on food establishments as the data available here is only a view of food safety. Does not incorporate customer reviews on taste, customer service, price etc.

# **4. Data Quality Problems**

## 4.1 List obvious data quality problems

1. Empty cells- there are many cells missing in some of the columns in the dataset like the violations, facility type, aka name, and longitude and latitude. The most is the violations column with 30798 null cells. Depending on the use case, these locations with no violation may need to be filtered out. In other use cases, these null cells will show that inspection resulted in no violations.
   1. A picture containing text, screenshot, display, software

      Description automatically generated
2. Inconsistent Naming and Typos- the same facility types are listed in different ways. Many of this will need to be cleaned and combined.
   1. A screenshot of a computer program

      Description automatically generated with medium confidence
3. Violations and comments are all listed in one column
   1. 
4. License # as 0- there are 439 records with a license number of 0.
   1. A screenshot of a computer

      Description automatically generated with medium confidence

## 4.2 Why data cleaning is necessary for main use case U1

Data cleaning is a crucial step to support the main use case of providing insights and suggestions based on food safety inspections. Currently the ‘Violation’ column contains various types of violations and their corresponding comments in a single column. To pave the way for a more detailed analysis and provide reliable insights and suggestions, data cleaning will be needed to extract these violations and comments to their own tables and columns. Next to provide safety insights by facility types, we need a high level of consistency of the data. There are noticeable inconsistencies with the ‘Facility Types’ column like spelling mistakes, inconsistent casing, and lack of standardization.

Without data cleaning of this dataset, we will not be able to ensure integrity and validity of the data, which will not achieve our goal of providing a accurate and reliable insights for our consumers.

# **5. Phase-II Initial Plan**

5.1 List typical data cleaning workflow/steps

1. Describe the dataset
2. Identify the main use case
3. Profile the data and identify data quality problems to support main use case
4. Performing data cleaning \*(check data cleaning plan below)
   1. Tools: Python, OpenRefine, RegEx,
5. Document all changes, the type and what actions was performed

5.2 Data Cleaning Plan

1. General data cleaning using OpenRefine, python, \_\_\_\_\_
   1. Check for missing values
   2. Check for and delete Duplicate records
   3. Fix inconsistent data
   4. Formatting
   5. Text cleaning to remove special characters, punctuations, and unnecessary whitespaces
   6. Check categorical data column like Risk, inspection type, results are inconsistencies or misspellings
   7. Delete unnecessary columns
   8. Remove outliers
2. Fix violations columns using python and RegEx
   1. Extract each violation to its own table
   2. Extract comments out for each violation
3. Fix facility type column using OpenRefine and python
   1. Standardized spelling and facility type name
   2. Combine and merge errors to use standardized name
4. Unique identifier for each location/restaurant
   1. Fix lice Jul 10 n to use as unique identifier

5.3 Who in the team will be responsible for which steps and Timeline

|  |  |  |
| --- | --- | --- |
| **Action** | **Who** | **Deadline** |
| Write detailed description of data cleaning performed |  |  |
| Create data quality changes taken |  |  |
| General data cleaning (trailing spaces, missing values, duplicates, outliers) |  |  |
| Fix violations column |  |  |
| Fix facility type column |  |  |
| Create workflow model using YesWorkflow tool |  |  |
| Work on Conclusions and Summary |  |  |
| Finish Final Report | All | Jul, 30 |