CS513: Theory & Practice of Data Cleaning

**Final Project Phase 1 Report**

**Team 23**

Avinash Baldeo, <add Names>

[abaldeo2@illinois.edu](mailto:abaldeo2@illinois.edu), <add Emails>

# 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 01/02/2013 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 event |
| DBA | string | Stands for “Doing Business”, it is the legal name of the registered food establishment. |
| AKA | string | Stands for “Also Known As”, it is the name the public would know the food establishment as. |
| License # | integer | A unique license number assigned to the establishment for legal purposes. |
| Facility Type | string | Describes the type/category of the establishment such as a bakery, restaurant, grocery store, etc. |
| Risk | string | Categorizes the establishments’ risk of adversely affecting public health (1 being the highest and 3 the lowest). |
| Address | string | The full street address of the food establishment. |
| City | string | The city the establishment is registered in. |
| 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 the distinct health violations found during the inspection. |
| Latitude | float | The GPS latitude coordinate of the address |
| Longitude | float | The GPS longitude coordinate of the address |
| Location | string | The GPS point coordinate of the address (latitude, longitude) |

Table 1 – Food Inspection Dataset Schema Description

# 2.2 Database Diagram

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.

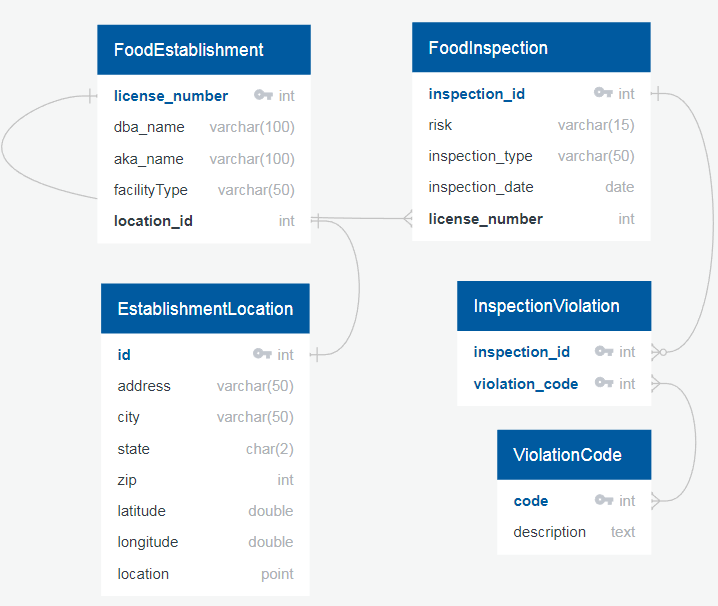


Figure 1A – Entity-Relationship (ER) Diagram of the normalized data

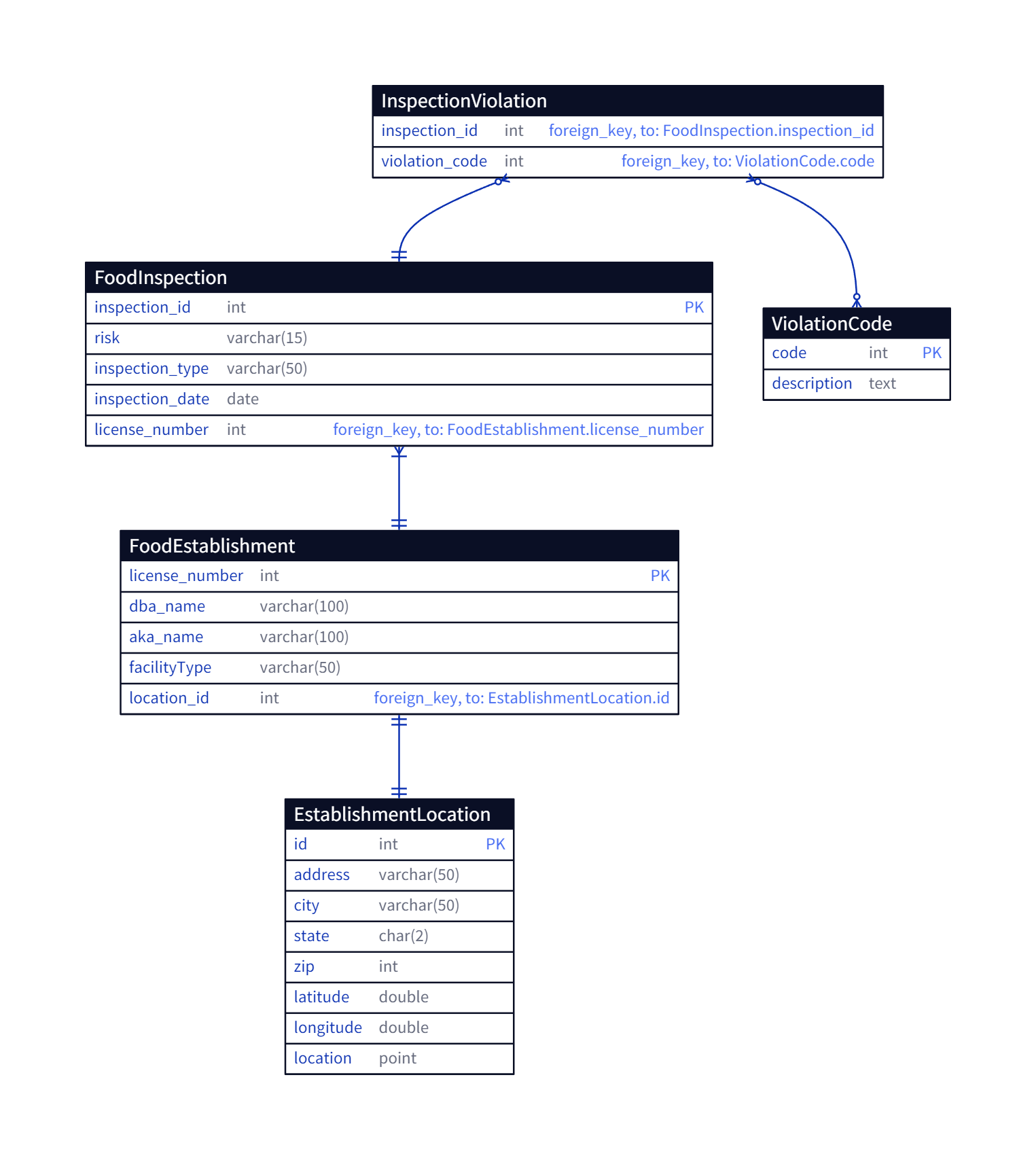


Figure 1B – Entity-Relationship (ER) Diagram with foreign keys

# 2.3 Database Schema

**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

# 3.2 U1: Main Use Case

# 3.3 U2: Never Enough Use Case

# 4. Data Quality Problems

# 5. Phase-II Initial Plan