CS 513 Data Cleaning Project Phase 1

Team 135: Jess Fan (jefan2@illinois.edu), Monika Janas (janas3@illinois.edu), David Parthun (parthun2@illinois.edu)

1. Dataset Chosen

For this project we chose to use the New York Public Library (NYPL) historical menu collection. The NYPL's collection has been open to the public for nearly 180 years and their menu has evolved tremendously over that time. With so much history this menu provides a unique insight into past cultural and economic trends as well as historic influences in the culinary arts. However, the downside is that countless people have been involved with generating this data and it's fair to assume that most of it was created through manual and inconsistent processes. So, this project is a good opportunity to perform a thorough data clearing in order to see what new insights we can uncover.

2. Description of Dataset

2.1 Narrative Description of the Dataset

The New York Public Library's restaurant menu collection includes over 45,000 menus spanning from the 1840's to present day. This data set is one of the largest of its kind in the world. The data was collected by transcribing menu dishes by hand, not using OCR (optical character recognition). The data set consists of four csv files, MenuPage.csv, Menu.csv, MenuItem.csv, and Dish.csv.

2.1.1 Menu

The Menu data contains information about the location name and address, venue, meal type, and occasion the menu describes. This data also contains a physical description of the menu appearance and other metadata such as page count and number of dishes.

- id – The unique identifier of the menu
- name – The name of the restaurant
- sponsor – Who sponsored the meal (organizations, people, name of restaurant)
- event – The category (e.g. lunch, annual dinner)
- venue – The type of place (e.g. commercial, social, professional)
- place – Where the meal took place (often a geographic location)
- physical_description – The dimension and material description of the menu
- occasion – The occasion of the meal (holidays, anniversaries, daily)
- notes – The notes by librarians about the original material
- call_number – The call number of the menu

- keywords – The keywords of the menu
- language – The language of the menu
- date – The date of the menu
- location – The organization or business who produced the menu
- location_type – The type of the location
- currency – The system of money the menu uses (dollars, etc.)
- currency_symbol – The symbol for the currency ($, etc.)
- status – The completeness of the menu transcription (transcribed, under review, etc.)
- page_count – How many pages the menu has
- dish_count – How many dishes the menu has each menu is associated with some number of MenuPage values.

2.1.2 MenuPage

The MenuPage data contains general metadata such as page number, image id, and size of menu pages. There are multiple ids that are used to link rows in the other relations.

- id – The unique identifier of the menu page
- menu_id – The unique identifier of the menu, corresponds to Menu id
- page_number – The number representing sequence of page in the menu
- image_id – The unique identifier of the page image
- full_height – The height of the page image in pixels
- full_width – The width of the page image in pixels
- uuid – The universally unique identifier for the highest resolution version of the image

2.1.3 MenuItem

The MenuItem data contains metadata used to link menu data to dish data, such as ids and creation and update timestamps. It also includes price information and whether the price of an item is considered high.

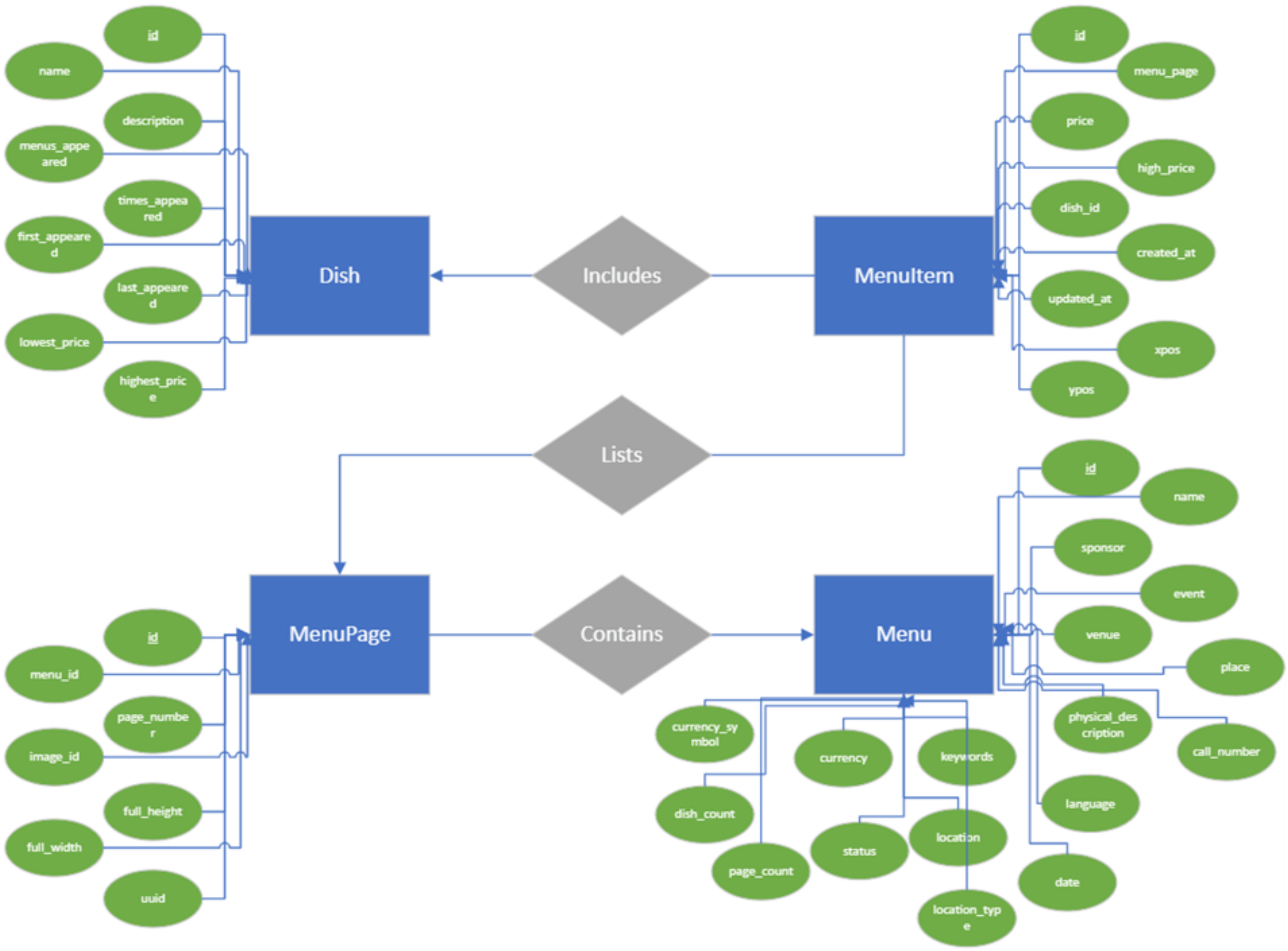
- id – The unique identifier of the menu item
- menu_page_id – The unique identifier of the page the menu item is on, corresponds to MenuPage id
- price – The first price of menu item
- high_price – If the item has more than one price on a single menu, the highest price. If there are more than two values for price, the web application instructs volunteers to enter the lowest and highest prices rather than all values.
- dish_id – The unique identifier of the dish, corresponds to Dish id
- created_at – The date/time of the first transcription
- updated_at – The date/time of the last edit to the value
- xpos – The horizontal coordinate on the page for the upper left point where menu item is on the page
- ypos – The vertical coordinate on the page for the upper left point where the menu item is on the page

2.1.4 Dish

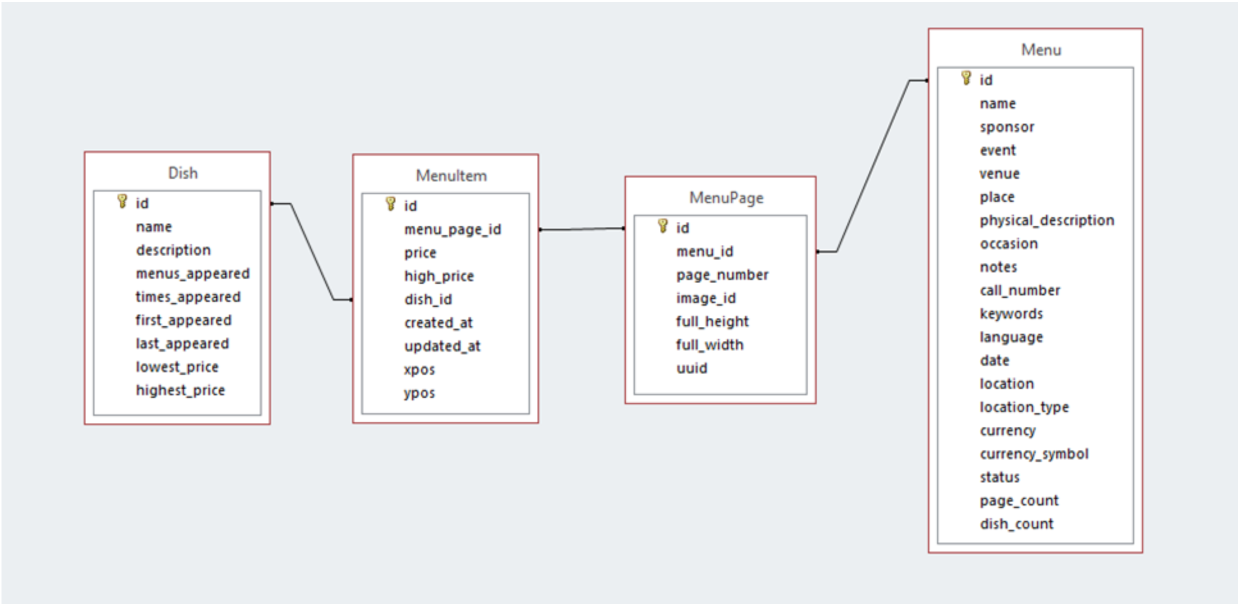
The Dish data contains information about the dishes offered on menus, such as name and description. It also includes some aggregated statistics like how many times a dish appears on menus in the collection, the dates (by year) when an item first and last appeared on a menu, and the lowest and highest prices the dish was offered at.

- id – The unique identifier of the dish
- name – The name of dish
- description – The description of the dish
- menus_appeared – The total count of menus on which dish with this id appears
- times_appeared – The total count of appearances of the dish with this id across all menus
- first_appeared – The earliest year of a menu on which a dish with this id appears
- last_appeared – The latest year of a menu on which a dish with this id appears
- lowest_price – The lowest price associated with a dish with a given id
- highest_price – The highest price associated with a dish with a given id

2.2 ER diagram (Conceptual Model)



2.3 Database Schema (Physical Model)



3. Use Cases

3.1 “Zero cleaning” use case (U0)

The following examples of use cases of data cleaning are not necessary.

ID	Name	Question/Query	Source Data	Description
Q001	Unique identifier of menu	Can the menu be uniquely identified?	Menu | ID	The ‘id’ column in Menu has all unique numeric values. Thus we can assume the menu can be uniquely identified.
Q002	Unique identifier of menu page	How many menu pages does each menu offer on average?	Menupage | ID,menu_page_id Menu | ID	The ‘id’ column in MenuPage has all unique numeric values per each menu; and is linked correctly to the Menu table. Therefore, querying the Menu table with joined MenuPage can get the corresponding MenuPage information.
Q003	Unique identifier of Dish	Calculating the number of times “Chicken soup with Rice” appears as a dish on the menus in the collection.	Menu | ID Dish | menu_id, name	The Dish table contains two columns that would enable this query: name and times_appeared
Q004	Dish name	List of dish names that first appeared on menus in 1912?	Dish | Name	The Dish table contains two columns that would enable this query: name and first_appeared. Dish name is clean and does not need further cleaning.
Q005	Menu Status	What are menu status? How many menus are still under review	Menu |Status	The ‘status’ column has all values available, very concise, only having two categories: complete or under review.
Q006	Menu page number, full height and full width	What is the average pages, height and width for each menu?	Menupage | page_number, full_height and full_width	The page_number, full_height, and full_width columns all have missing entries (1202, 329, and 329, respectively) but seem to be otherwise clean. Both full_height and full_width are missing entries in the exact same rows.
Q007	Menu item id, menu_page_id, dish_id, xpos, and ypos	What are the average items per menu page or per dish? What are average position (x and y) are located per menu item	Menu Item | id, menu_page_id, dish_id, xpos, and ypos	In menuitem.csv, id, menu_page_id, dish_id, xpos, and ypos are clean, can be used to query to answer the use case questions without cleaning.

3.2 “Main” use case (U1)

The main use case would be a restaurant entrepreneur or a consulting group that is evaluating menu options and/or offerings as a service to a new restaurant set to open in New York City. They would need a clean dataset with no duplicates to conduct their research, but some minor discrepancies in the data are acceptable. Data cleaning is valuable in this context because in the raw data form, the dataset contains blank values, inconsistency in naming convention and data types and duplicate values that can be merged. Completing these steps will improve the quality of the data enough that it is a valuable resource to the owner or the consulting firm to make proper business decisions or recommendations to a new dining establishment on what type of menus they should offer. Or before a restaurant offers a dish, it could look at this dataset to see if a similar dish has been in the menus and how popular it is, as well as the price at which it’s been offered throughout the years.

Other use cases would be in the space of journalists or other researchers who are doing research on some dishes. It would be interesting to see if there are any forgotten dishes that perhaps could be revived.

From customers or travelers’ point of view, the clean dataset would be valuable to see what are choices of menus, most popular dishes, restaurants and average prices per menu, etc. based on their preferences in New York city.

The following are only a few examples of use cases of data cleaning that are necessary and sufficient. The additional use cases are listed under section 4 data quality problems section.

ID	Name	Question	Source Data	Description
Q101	Menu name deduplication	How many restaurants contain the same menu?	Menu | Name	There are a lot of menu names that are exactly the same but due to extra spaces, punctuations, different order of words, and typos they don’t match exactly. Through necessary and sufficient data cleaning, the question can be answered.
Q102	Sponsor name deduplication	Which sponsor sponsored the most of the menus?	Menu | Sponsor	There are a lot of sponsor names that are exactly the same but due to extra spaces, punctuations, different order of words, and typos they don’t match exactly. Through necessary and sufficient data cleaning, the question can be answered.
Q103	Uniquely identify dish	Calculate the number of distinct	Dish | name	We can use OpenRefine to categorize multiple dishes with

	name	dishes		similar values in the name column as one item. This cleaned data can then be used to calculate an accurate number of distinct dishes in the dataset.
Q104	Classify category or event of the Menu	Calculate the number of menus corresponding to “Dinner”	Menu | event	We can use OpenRefine to categorize menus from the Menu table with similar values in the event column to “Dinner” as one item. This cleaned data can then be used to calculate an accurate number of menus for Dinner in the dataset.

3.3 “Never Enough” use case (U2)

The following examples of use cases of data cleaning are not sufficient

ID	Name	Question	Source Data	Description
Q201	Missing value of menu name	What is the name of each menu?	Menu | Name	The ‘name’ column has only 3197 non-empty values. There are also placeholders for missing value, e.g., ‘[Restaurant name and/or location not given]’ or ‘[Not given]’. Due to limitation of missing value in data source.There is no way to make each menu contain a valid name.
Q202	Missing value of menu sponsor	Who sponsored the meal (organizations, people, name of restaurant) ?	Menu | Sponsor	The ‘sponsor’ column has 15,984 non-empty values, and these values have similar issues to the ‘name’ column. Also, some of the values are just question marks. Due to limitation of missing value in data source, there is no way to make each menu contain a valid sponsor.
Q203	Missing value and classification of Menu notes	Classify menu notes	Menu | Notes	The ‘notes’ column has 10,613 non-empty values. The values in this column are mostly represented by paragraphs of free text, mostly unstructured. Deriving and classifying from this column may not be possible.

4. Data Quality Problems

4.1 Menu

The first inspection of the data shows us that this file has 17,545 entries and 20 columns. The following are the list of data quality problems identified and their related actions and reasons why data clean up will help support use case 1.

Field(s)	**Data Quality Problems (with screenshots/examples)**	**Actions and Reasons to support U1**
Keywords, language, location_type	● These three columns do not have any values	No information and missing all values, can be deleted to save database space.
name	● Has only 3197 non-empty values. ● There are also placeholders for missing value, e.g., '[Restaurant name and/or location not given]' or '[Not given]'. ● There are a lot of names that are exactly the same but due to extra spaces, punctuations, different order of words, and typos they don't match exactly.	Use OpenRefine to clean up extra spaces, punctuations, different order of words, typos.
sponsor	● Has 15984 non-empty values ● These values have similar issues as the 'name' column. ● Some of the values are just question marks.	Use OpenRefine to clean up extra spaces, punctuations, different order of words, typos.
event	● Has 8154 non-empty values. ● The values for this column can be grouped into different buckets such as 'breakfast', 'lunch', 'dinner' etc. ● Some of these values are written in different languages e.g., French or German, and it depends on the use case whether this can be grouped together. ● The values such as '107th, 108th ... anniversary dinner' can be grouped together as just 'anniversary dinner'.	Clean up categories and expand rows with multiple values. Consistent category values will make it easier to aggregate and query menus by the event.

	• Each value can have multiple categories e.g., ‘lunch and dinner’, which also can be post-processed based on the use case.	
venue	• The column has 8119 non-empty values. • The values in this column have the most of common issues, including question marks, extra punctuations, etc., • new unique issues with abbreviations e.g., ‘SOC’ and ‘SOCIAL’, ‘COM’ and ‘COMMERCIAL’. • In addition, this column can also have multiple categories within one value.	Use OpenRefine to clean up extra spaces, punctuations, different order of words, typos.
place	• The ‘place’ column has 8123 non-empty values. • And again, besides common issues, this column has an issue with partial values.	The value can be cleaned up and classified to represent just the name of the place or place and city or address line, city and state, etc.
physical_description	• The column 14763 non-empty values. There as some ‘#N/A’ values. • Each value in this column has multiple sub-values such as type of menu e.g. ‘booklet’, ‘card’, ‘folder’ and physical dimensions of the menu e.g. ‘5.75 X 7.25’, ‘5 X 8’ and some unique features of the menu e.g. with or without illustration, regular or column layout, folded or open. • This column can have multiple variations of such properties within one value.	This column can be cleaned up and possibly split into multiple values for easy query.
occasion	• The column has 3791 non-empty values. • The values of this column also can be grouped into multiple buckets.	The column can be classified into multiple buckets for easy querying.
call_number	• The column has 15983 non-empty values. • The majority of values in this column are numeric with some OCR-like issue e.g. we see ‘o’ instead of ‘0’, or ‘l’ instead ‘1’. Some	The column can be classified into multiple buckets for easy querying.

	of them have postfixes such as ‘item’, ‘_wotm’, ‘copy’. And some of them starting from the word and continuing with a number, e.g. ‘Zander 645’, ‘Soete 162’, ‘Baratta 35’.	
date	● Inconsistent and inaccurate dates. Some are dates while other are years and others are seem to have been mistyped	Only three values where there are some issues with the year and can be easily detected using timeline facet from OpenRefine.
location	● The ‘location’ column does not have empty values. However, there are values such as question mark. The issues are similar to the issues with ‘name’ or ‘sponsor’ columns.	Use OpenRefine to clean up extra spaces, punctuations, different order of words, typos.
currency, currency_symbol	● Both have 6456 non-empty values, and they look good. ● Some preprocessing can be done for cents because it can be cents of different currency.	Use OpenRefine to clean up to make currency consistent.
Page_count, dish count	● There are some extreme values that need to be analyzed.	Possible to use OpenRefine to clean up or mark the extreme value.

4.2 MenuPage

The first inspection of the data shows us that this file has 66937 entries and seven columns.. The following are the list of data quality problems identified and their related actions and reasons why data clean up will help support use case 1.

Field(s)	**Data Quality Problems (with screenshots/examples)**	**Actions and Reasons to support U1**
image_id	The values in this column are using three different formats. About half of the entries are using 7-digit numeric IDs, another half are using 10-digit numeric IDs, and a few (23) of the values are using alpha-numeric IDs.	Update OpenRefine to make format consistent.
uuid	The column was almost entirely clean, only one entry needed to be updated to use lower-case letters. It is worth noting that some uuids are duplicated.	Use OpenRefine to update one uuid to make data consistent with all lower cases.

4.3 MenuItem

The first inspection of the data shows us that this file has 1332726 entries and nine columns. The following are the list of data quality problems identified and their related actions and reasons why data clean up will help support use case 1.

Field(s)	**Data Quality Problems (with screenshots/examples)**	**Actions and Reasons to support U1**
price	The column has 445,916 blank rows. It is also worth noting that there are 130 rows with extremely high (over $10,000) prices.	Possible to mark these extremely high value price for further analysis
high_price	The column has 1,240,821 blank rows, which means that the vast majority of the rows are blank.	It may be worth excluding this column to save database space.
created_at & updated_at	Depending on our purpose we may want to drop the UTC string and convert these values to ISO 8601 datetime formats	Consistent timestamp format enables comparisons and aggregations by date

4.4 Dish

The first inspection of the data shows us that this file has 423,397 observations of 9 variables. The following are the list of data quality problems identified and their related actions and reasons why data clean up will help support use case 1.

Field(s)	**Data Quality Problems (with screenshots/examples)**	**Actions and Reasons to support U1**
description	• Over 98% of this column does not contain any values.	Option 1, remove description to save database space. Option 2. There are 9,125 rows where the name column most likely contains the description of the dish because the length of the text is over 100 characters and most of the names are under 100 characters. We can move from the 'name' column into the 'description' column.
times_appeared	• There are several negative numbers going as low as -6 • There are also some '0's in there as well.	Negative values for times_appeared indicates a mistake in data entry, and that the item should be reviewed for correctness.
first_appeared & last_appeared	• There are a number of records with 0s,1s, and inaccurate outlier dates (2928) • some values in the 'first_appeared' column are greater than those in the 'last_appeared' column	Further examine the data correlation, correct the outliers.

lowest_price & highest_price	- Contains both 0s and nulls - Has a number of prices that are way too high	Further examine the data, correct the outliers as needed.

5. Initial Plan for Phase 2

5.1 Data Cleaning Workflow Steps

The following table describes the steps we will perform to clean up this data, who is the owner for completing that task, and what tool will be used.

Task	**Tool**	**Owner**
Import Menu table data into OpenRefine and perform cleaning tasks - Remove unneeded columns - Expand any additional columns - Condense categories - Clean up unnecessary punctuation	OpenRefine	Jess
Import MenuPage table data into OpenRefine and perform cleaning tasks - Remove unneeded columns - Expand any additional columns - Condense categories - Clean up unnecessary punctuation	OpenRefine	Jess
Import MenuItem table data into OpenRefine and perform cleaning tasks - Remove unneeded columns - Expand any additional columns - Condense categories - Clean up unnecessary punctuation	OpenRefine	Jess
Import Dish table data into OpenRefine and perform cleaning tasks - Remove unneeded columns - Expand any additional columns - Condense categories - Clean up unnecessary punctuation	OpenRefine	Jess
Define integrity constraints for Menu table - Key constraints - Extreme values	Datalog	Dave

Define integrity constraints for MenuPage table ● Key constraints ● Extreme values	Datalog	Dave
Define integrity constraints for MenuItem table ● Key constraints ● Extreme values	Datalog	Monika
Define integrity constraints for Dish table ● Key constraints ● Extreme values ● Valid semantic values (first_appeared < last_appeared)	Datalog	Monika
Import Menu table data in SQL and validate U1 scenarios	SQLite	Dave
Import MenuPage table data into SQL and validate U1 scenarios	SQLite	Dave
Import MenuItem table data into SQL and validate U1 scenarios	SQLite	Monika
Import Dish table data into SQL and validate U1 scenarios	SQLite	Dave
Validate U1 scenarios for Menu table	SQLite or Python	Dave
Validate U1 scenarios for MenuPage table	SQLite or Python	Dave
Validate U1 scenarios for MenuItem table	SQLite or Python	Monika
Validate U1 scenarios for Dish table	SQLite or Python	Dave
Document workflow for Menu table	YesWorkflow	Dave
Document workflow for MenuPage table	YesWorkflow	Dave
Document workflow for MenuItem table	YesWorkflow	Monika
Document workflow for Dish table	YesWorkflow	Monika

5.2 Project plan with Timeline

Due Date	Milestone
July 9	Complete and submit Project Phase 1 document
July 15	Finalize workflow and plan, begin implementation of data cleaning ● OpenRefine ● Integrity constraints in Datalog
July 22	Continue implementation of data cleaning, begin writing summary and conclusions of the experience ● Import data into SQL ● Scenario and cleaning validation
July 22	Collect all artifacts of the data cleaning process in Github repository ● Document workflow using YesWorkflow
July 30	Complete and submit Project Phase 2 document