Final Project

End-To-End Data Cleaning Workflow – NYPL Data

by

Sudhir Behani

Bollam Raja Shekar

Table of Contents

[**Introduction and Overview** 3](#_Toc15405887)

[1. Dish.csv 3](#_Toc15405888)

[2. Menu.csv 3](#_Toc15405889)

[3. MenuItem.csv 3](#_Toc15405890)

[4. MenuPage.csv 3](#_Toc15405891)

[5. Missing Data 3](#_Toc15405892)

[Other Information 3](#_Toc15405893)

[**Initial Assessment of the dataset** 5](#_Toc15405894)

[**Data Cleaning methods and Process** 6](#_Toc15405895)

[Open Refine 6](#_Toc15405896)

[**Data Cleaning Results** 10](#_Toc15405897)

[Relation Database Schema 10](#_Toc15405898)

[ER Diagrams 10](#_Toc15405899)

[Integrity Constraint Checks 11](#_Toc15405900)

[Workflow Model 11](#_Toc15405901)

[Or2yw overall workflow 11](#_Toc15405902)

[Or2yw open Refine csv 11](#_Toc15405903)

[Overview of changes 12](#_Toc15405904)

[**Conclusion and Future Work** 13](#_Toc15405905)

# **Introduction and Overview**

## Dish.csv

Information about all the dishes from all the menus transcribed by the project are stored in dish.csv. In this data file, a dish is represented by a row of values. Columns identify attributes of a dish. One of these attributes is an identifier, which identifies the dish. However, the identity of a dish appears to be based on the exact form of the string labeled "name." Thus, dishes with variant orthographic forms of their names, e.g. “half chicken”, “Half Chicken” and “chicken [half]”) are treated as separate entries with different identifiers.

## Menu.csv

Information about the menus as physical objects, including historical information about their origins, uses, and formats are stored in menu.csv

## MenuItem.csv

This is the largest data file. A "MenuItem" represents a single instance of a dish appearing somewhere on a menu page image. “MenuItem.csv” is useful as a mapping between multiple other data files.

## MenuPage.csv

Information about individual pages of the menus is stored in "Menu.csv". Pages are modeled here as digital images produced as a result of digitization by the NYPL. Menus often have multiple pages.

## Missing Data

We note the presence of missing data points in the column “missing values.” However, there are also strings in the present data that point to missing information (e.g. “unknown” or “?”), which do not formally appear as null values.

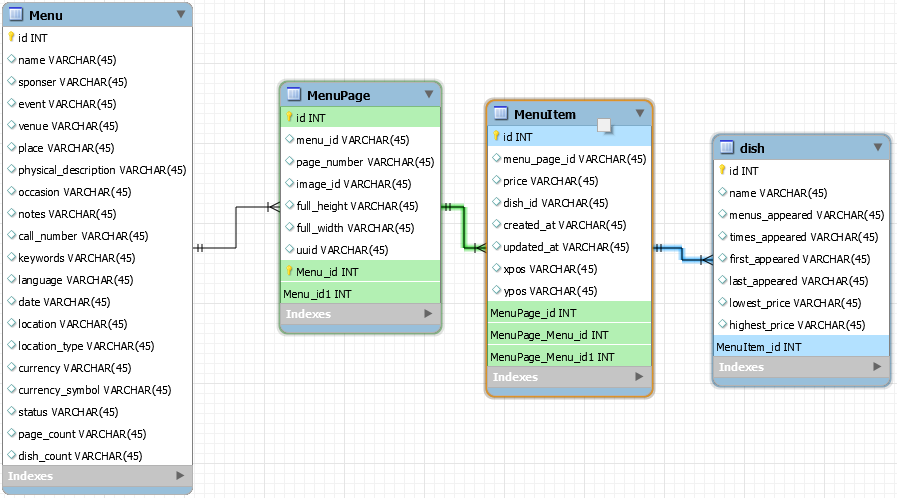
## Source of data

The data in these files comes from several different sources. Some of the data is supplied by “volunteers,” by which we mean people who have participated in the project through the What’s On the Menu? site. Some of the information is generated by “web application.” This means that some of this information was automatically created as the database supporting the application was constructed and populated (e.g., various ids); some is created as the web application runs (e.g. timestamps as data values are updated). Finally, a lot of information is generated from “NYPL metadata.” This metadata comes from many places and reflects the long history of the project and the many parts of New York Public Library involved in it. Much of the data “supplied by NYPL metadata” in the menu spreadsheet is from the catalog cards made by Frank E. Buttolph in the early twentieth century.

# **Initial Assessment of the dataset**

Structure

Explaining the structure of the data using an **ER diagram**. This shows connection between all the files/tables using foreign key.



Content

Quality Issues

# **Data Cleaning methods and Process**

## Open Refine

1. Menu

Activities performed:

* Trim leading and trailing white spaces:
* Collapse consecutive white spaces:
* Convert columns to number type:
* Convert columns date to format YYYY-MM-DD :
  + Using expression: toString(toDate(value),"yyyy-MM-dd")
* Remove special characters from the text columns. Also, remove characters at the end of the string like semicolon, comma and hyphen.

Using expression - value.replace(/\\|\(|\)|\"|\[|\]|\?|#|%|!|[;,-]$/,'')

Note: Can’t remove (semicolon ; ) character in the middle of string as it signifies

multiple options- example multiple sponsors, event, venue, place etc.

* Remove continuous hyphen (-) characters from the text columns.
* Convert all values in columns with string data type to upper case so that problem in clustering similar values is reduced considerably.
* After removing special characters again perform white spaces removal for each the string data type columns
* Clusters columns based on method key collision and keying function as fingerprint. Accepted the default values for new cell value. Merge cluster and then perform ngram-fingerprint using ngram size as 2. Metaphore3 and cologne-phonetic were mostly creating clusters with somewhat different names too so were not used.
* Physical\_description seems to contain information about the menu hard copy format and its size. Menu could be in different forms – accordian fold, book, booklet, broadside, card, folder, tri-folder, two cards joined by ribbons etc.

Split the physical\_description column with separator as semicolon. It is split into 7 columns. Rename the first column as “physical\_description\_menu\_type” and join other columns as rename it as “physical\_description\_menu\_others”

Using expression:

if(isBlank(cells["physical\_description 2"].value), "",

cells["physical\_description 2"].value) + ";" +

if(isBlank(cells["physical\_description 3"].value), "",

cells["physical\_description 3"].value) + ";" +

if(isBlank(cells["physical\_description 4"].value), "",

cells["physical\_description 4"].value) + ";" +

if(isBlank(cells["physical\_description 5"].value), "",

cells["physical\_description 5"].value) + ";" +

if(isBlank(cells["physical\_description 6"].value), "",

cells["physical\_description 6"].value) + ";" +

if(isBlank(cells["physical\_description 7"].value), "",

cells["physical\_description 7"].value)

Remove semilon at the end of the string using expression - value.replace(/[;]{1,}/,''). This step will take care of empty semicolons in the end of the string and will remove all of them. In this way no value will be lost.

Deleted Columns:

Columns with no values were deleted – keywords, language, location\_type,

Not updated columns:

Columns which are not updated: id, name, keywords, language, location\_type, currency, currency\_symbol

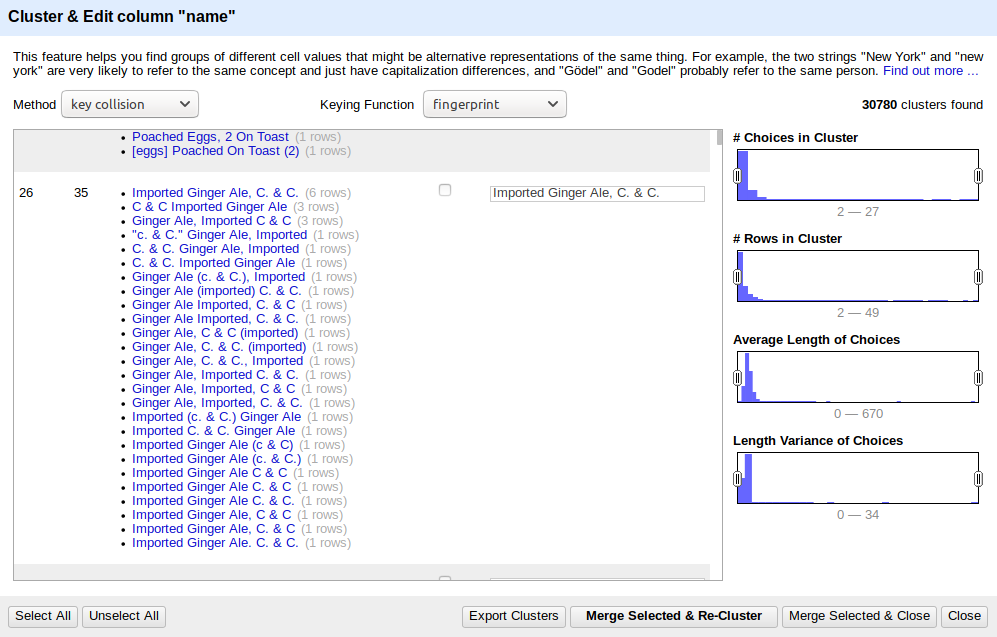
Difficulties:

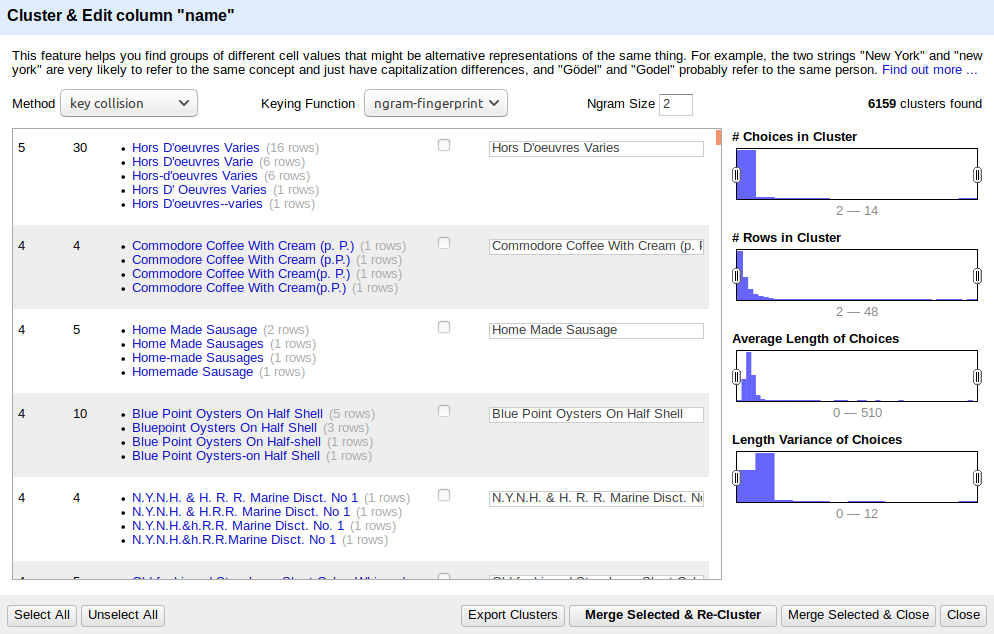
* There occasions when “,” (comma) is interchangeably used for multiple items and additional information after comma. This needs to be taken care separately.
* Unable to perform nearest neighbor with ppm keying function as it was using 100% cpu and openrefine was getting into improper state.

1. Dish

Activities performed:

* Trim and Collapse white spaces in “name” column.
* Convert column “name” to title case.
* Make a facet and perform the cluster operation using the “key-collision” method and “fingerprint” as shown below.



* Repeat step c with “ngram-fingerprint” as shown below. 
* Remove “description” column, since most of the entries are empty.

1. MenuPage

Activities performed:

* Convert columns to number type:

This file does not have any other cleaning activity to be performed.

1. MenuItem

Activities performed:

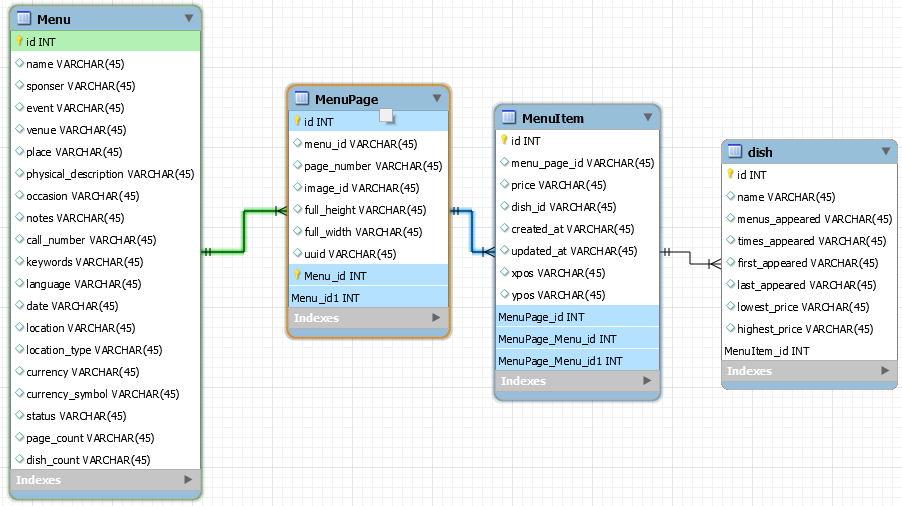
Following data cleaning steps were performed on the Dish dataset.

* “high\_price” column is removed
* “created\_at” is transposed into “toDate()”
* “updated\_at” is transposed into “toDate()”

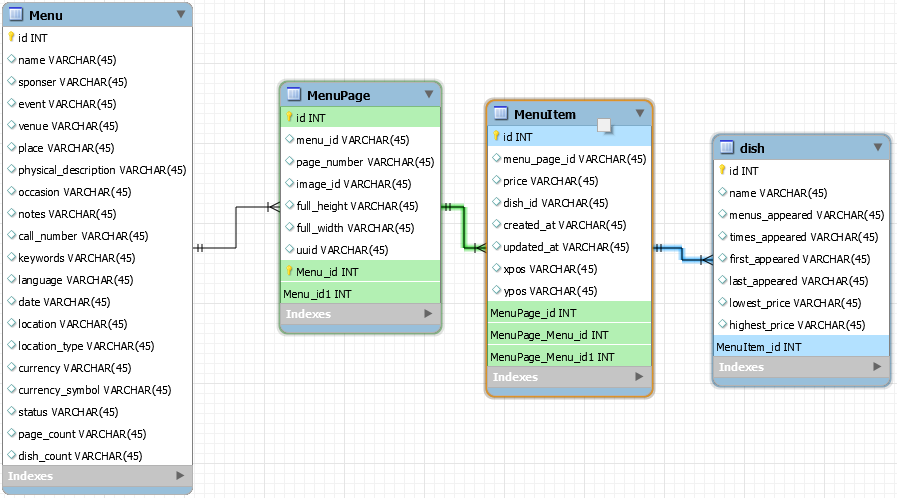
# **Data Cleaning Results**

## Relation Database Schema

### ER Diagrams



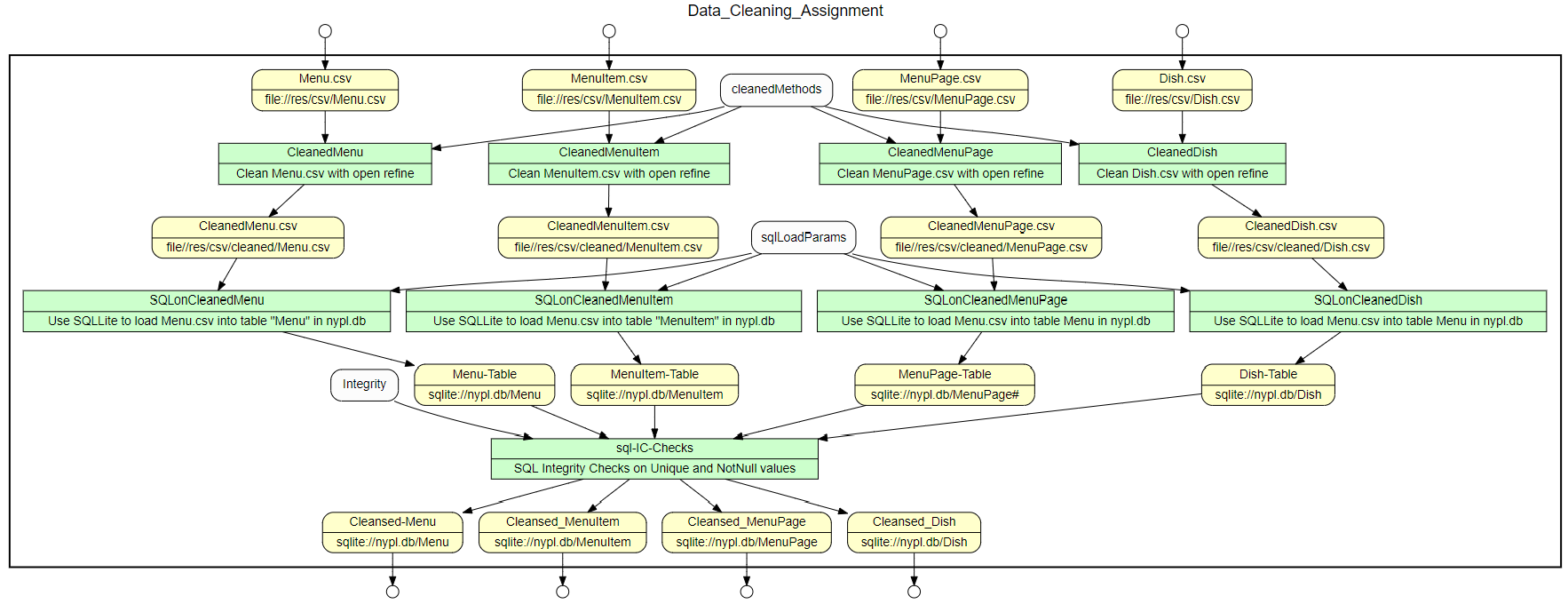
<<<Add text >>>



### Integrity Constraint Checks

## Workflow Model

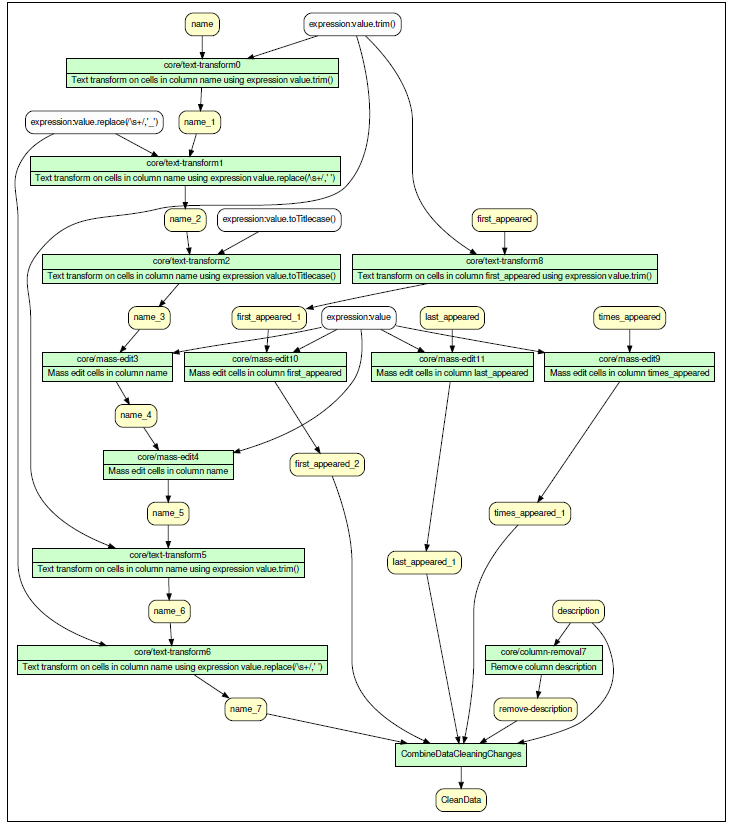
### Or2yw overall workflow



### Or2yw open Refine csv

Following graphs are generated using or2yw tool based on the recipes generated from the openrefine too.

1. Dish.csv



1. Menu.csv

## Overview of changes

# **Conclusion and Future Work**

Main Outcomes:

Key takeaways:

Problems or Challenges:

Additional steps:

Certain foreign letters which needs to be take care by studying the words in more detail. These foreign characters seems to occur in menu table under columns – sponsor and location.

Future additional tasks:

For Menu we could use status column having two values “Complete” and “Under Review” and perform classification task. We could try to predict which features in menu table led to the value Complete and Under Review.