CS 513 – FINAL PROJECT – PHASE II

TEAM 169 - CHIRANJEEVI (CHIRU) KOILOTH, CHUN CHIEH (WILLY) CHANG, AARON BARRIE

TEAM COMPOSITION

1. Chiranjeevi (Chiru) Koiloth (koiloth2@illinois.edu)
2. Chun Chieh (Willy) Chang (cc132@illinois.edu)
3. Aaron Barrie (abarrie2@illinois.edu)

USE CASE DESCRIPTION

The following section reiterates the Main Use Case described in the team's Phase I submission. As the U1 definition is built on top of the U0 definition, both are included below.

U0 – ZERO CLEANING

The team wishes to build a web-based application to deliver the contents of this dataset for an embedded system where the image references indicated in the MenuPage entity of this dataset are unavailable for distribution. Each menu can be identified from a navigable list that provides search and filtering based on:

- Menu Name
- Location
- Special Occasion
- Date of Publication
- Keyword Search for Notes

Clicking on a menu will open a navigable document that has been reconstructed based on the information from the Menu Page, Menu Item, and Dish entities.

The Menu Page entity will provide the Menu Page ID and page dimensions, which will be scaled to fit the current browser window.

All Menu Items associated with a given Menu Page will be plotted using a textbox with an origin point located at the relative X/Y described by xpos/ypos. This textbox will include:

- Dish Name (from the Dish entity)
- Dish Price (from the MenuItem entity)
- Dish Alternate Price (from the MenuItem entity, when available)
- Dish Prices will be decorated with a Currency Symbol from the Menu entity

Additionally, this application will support the ability to filter the list of Menu's based on a search for Dish names. If a menu is selected when this filter is used, then the application will automatically navigate the user to the page on which that Dish is found, with a highlight placed around the location of the Menu Item textbox on that page.

Any data points presented in the original data set will be done as-is. Any data points that are inaccessible due to integrity constraint violations will result in a "Missing Page" notice to the end user in the even to of a missing page

reference, or "Missing Data" notice to the end user in the event of a missing dish reference. When currency markers are unavailable, the prices will be listed as-is without currency designation.

U1 – MAIN USE CASE

The team wishes to extend the functionality of the application described in U0 by adding the ability to offer additional Organization- and Dish-based analysis.

To offer Organization-based analysis, the data points for Name/Sponsor on the Menu entity need to be appropriately clustered to accurately present a comprehensive view of how the menu characteristics for a single organization change over time.

To offer Dish-based analysis, the data points for the Dish Names from Dish.csv and prices from MenuItem.csv need to be cleaned to provide an accurate view for how the popularity of specific dishes changed over time, and how price trends for dishes change over time.

1. DESCRIPTION OF DATA CLEANING PERFORMED

The general approach to the data cleanup process ended up being a three-step process when viewed from a high-level.

1. Load each source file in SQLite and perform baseline IC data consistency checks.
    a. Inputs for this step are the four raw data files acquired from the course website.
    b. Outputs for this step are IC exception metrics across all four input files.
2. Load each source file in OpenRefine and perform internal file cleanup.
    a. Inputs for this step are the four raw data files acquired from the course website.
    b. This work involves performing cleanup operations that do not require any awareness of relational consistency between each of the input files.
    c. Outputs for this step are four cleaned data files corresponding to the four raw data files provided at input.
3. Load each OpenRefine cleaned file in SQLite and perform relational data cleanup.
    a. Inputs for this step are the four cleaned data files produced as an output of step 1.
    b. This work involves performing cleanup operations that are primarily focused on relational consistency between each of the four input data sets.
    c. Outputs for this step are four final-state cleaned data files corresponding to the four input files.
    d. Additional outputs for this step are IC exception metrics across all four output files.

The rationale for this approach is straightforward: OpenRefine excels at operations that only require awareness of operations within the single data file being inspected. SQLite excels at operations that require awareness of relationships between each of the input files in the source data set.

OPENREFINE ACTIONS

MENU

1. Sponsor, event, venue, place, occasion, and location
    1. Using text transformations in Edit cells
        1. Trim leading and trailing whitespaces.

Text transform on 14 cells in column sponsor: value.trim() Undo

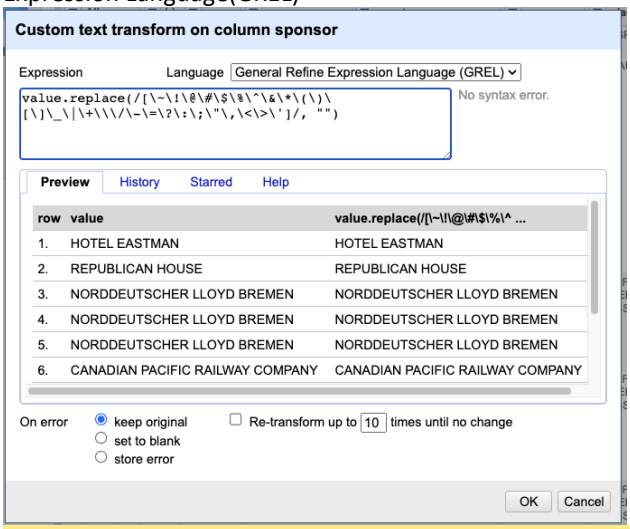
2. Collapse consecutive whitespaces.

Text transform on 127 cells in column sponsor: value.replace(/[\p{Zs}\s]+/,' ') Undo

3. Convert the whole column to uppercase.

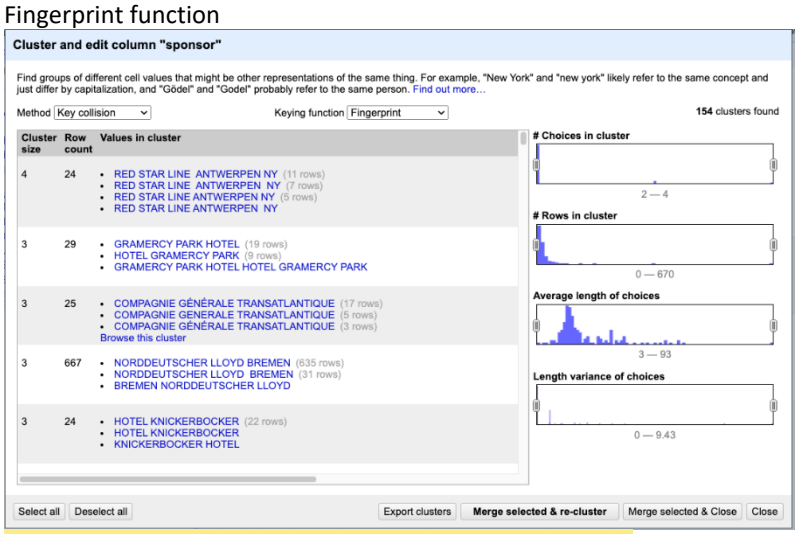
Text transform on 8,535 cells in column sponsor: value.toUppercase() Undo

4. Remove the special characters '~!@#$%^&*()[]_|+\/-=?:;",<>' using General Refine Expression Language(GREL)



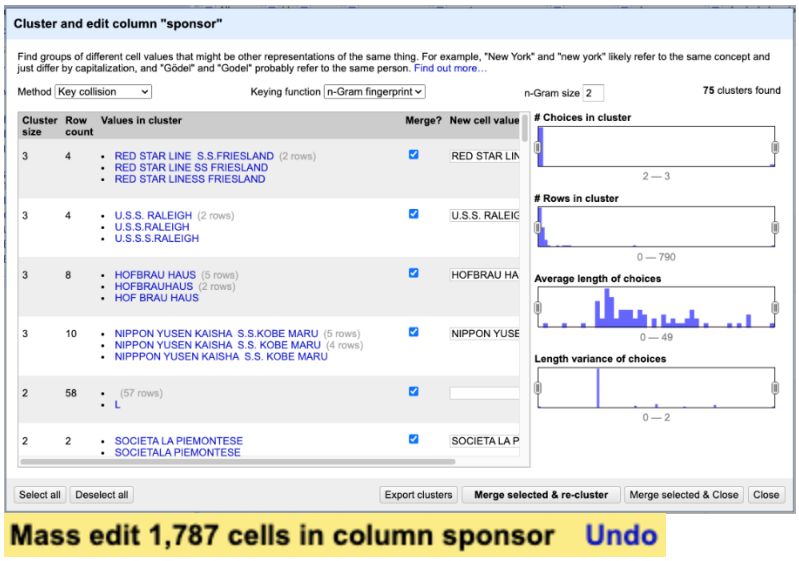
Text transform on 3,564 cells in column sponsor: grel:value.replace(/[\~\!\@\#\$\%\^\&*\(\)\[\]_\|\+\\\/\-\=\?\:\;\"\,\<\>\']/, "") Undo

2. Create a text facet and implement the cluster operation
    1. Using the key-collision method with different keying functions
        1. Fingerprint function

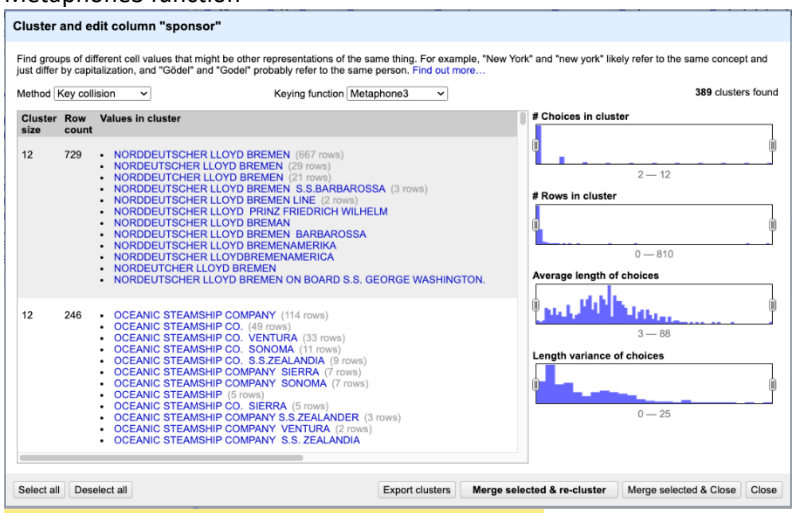


Mass edit 2,933 cells in column sponsor Undo

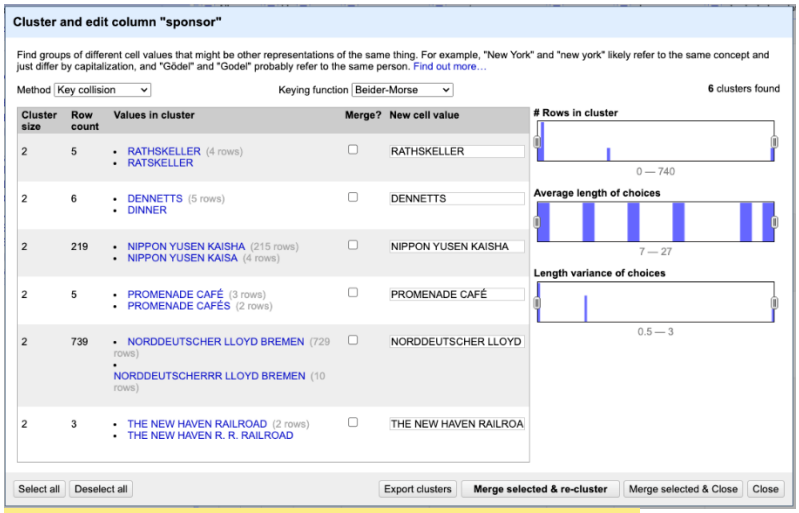
        2. n-Gram fingerprint function with n-Gram size 2



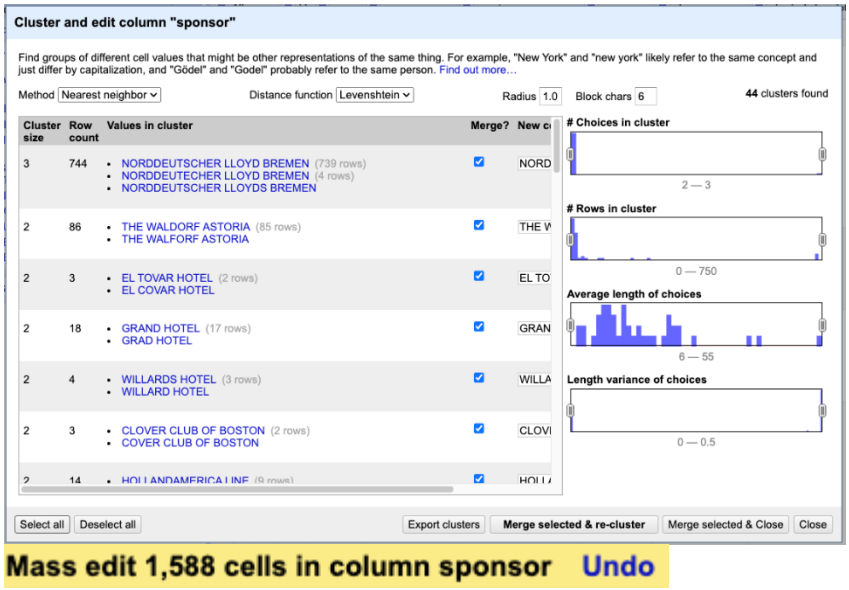
3. Metaphone3 function



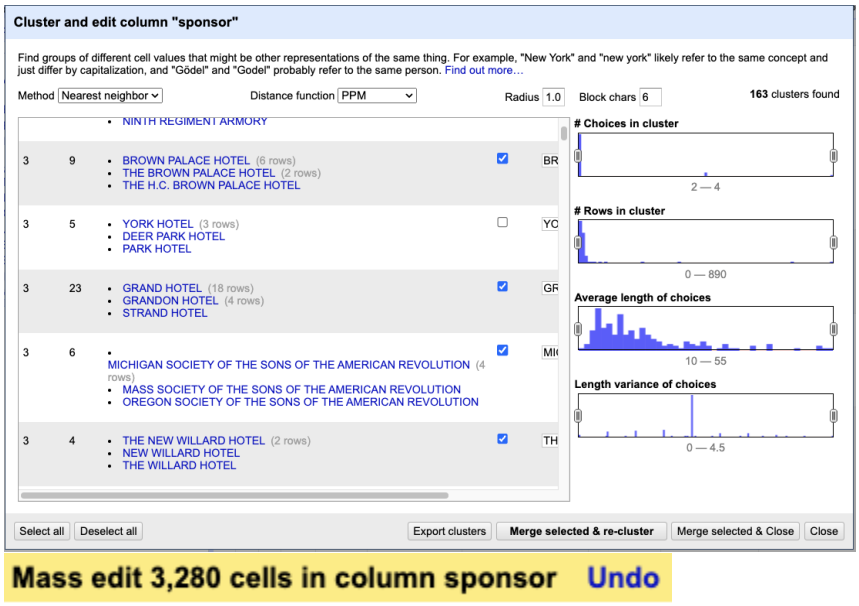
4. Beider-Morse functions



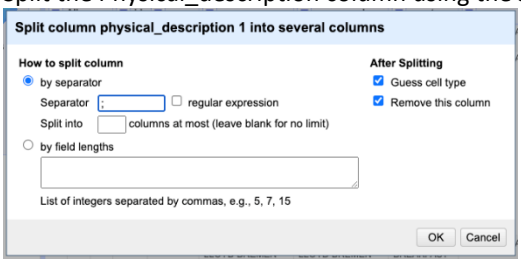
2. Using Nearest Neighbor method with different distance functions
    1. Levenshtein distance function with radius 1.0 and block chars 6



    2. PPM distance function with radius 1.0 and block chars 6



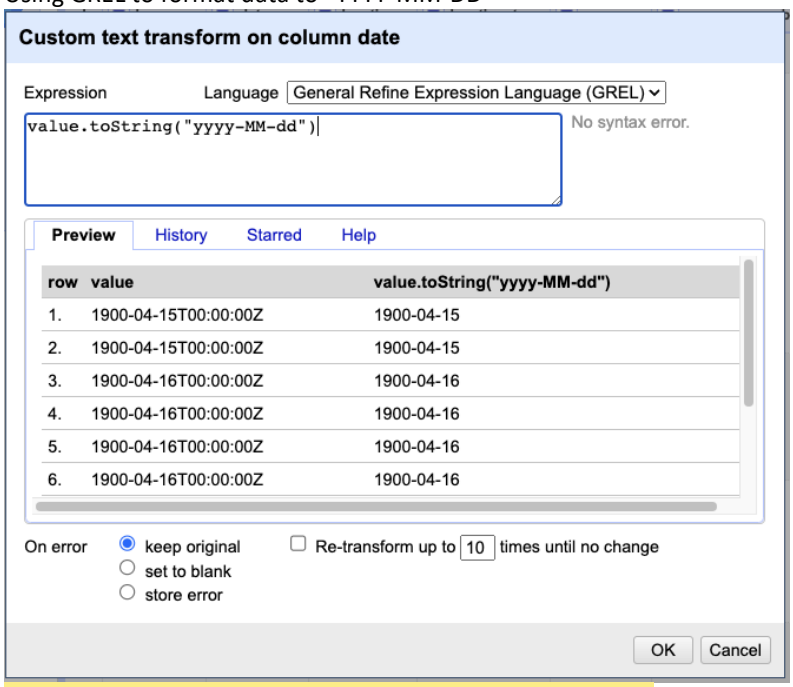
3. Physical_description
    1. Split the Physical_description column using the semicolon ‘;’.



4. Date
    1. Using Common transforms data to date

Text transform on 16,959 cells in column date: value.toDate()
Undo

2. Change the date less than 1851 and more than 2015 to Null
3. Using GREL to format data to "YYYY-MM-DD"



Text transform on 16,954 cells in column date: grel:value.toString("yyyy-MM-dd") Undo

MENUPAGE

1. We only uppercased the UUID to make sure the format was consistent.

Text transform on 66,936 cells in column uuid: value.toUppercase() Undo

MENUITEM

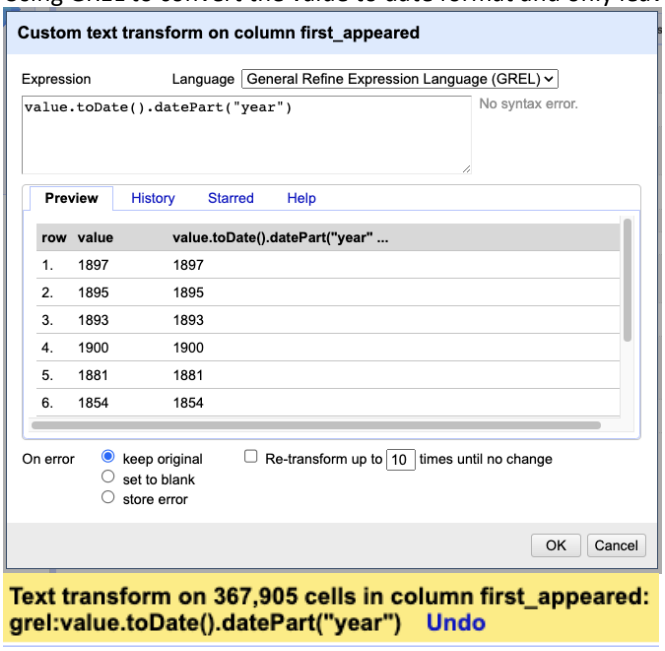
No OpenRefine operations were performed on this dataset. Cleanup operations were attempted on the date fields but no changes resulted as no date format inconsistencies were found.

DISH

1. To resolve the MenuItem -> Dish relationships after the clustering took place. We have to retain the original name column and create a new column called clean_name (Duplicated from the name column)
    1. Using text transformations in Edit cells
        1. Trim leading and trailing whitespace
        2. Collapse consecutive whitespace
        3. Remove the special characters '~!@#$%^&*()[]_|+\/-=?:;",<>' using General Refine Expression Language(GREL)
        4. Transform the text to lowercase

2. Create a text facet and implement the cluster operation using key-collision methods with different keying functions.
    1. Fingerprint function
    2. n-Gram fingerprint function with n-Gram size 2
    3. Metaphone3 function

2. first_appeared, last_appeared
    1. Using GREL to convert the value to date format and only leave the year part.



    2. Change the illegal year which is less than 1851 or greater than 2928 to 0.

SQLITE ACTIONS

BASELINE MEASUREMENTS

The first steps performed in SQLite were to use the raw (uncleaned) data files and load them into the Initial SQLite Schema (see additional details below in the SQLite Schemas section). A set of count queries were run validating each of set of predefined IC check queries to determine the number of records in violation of each IC.

The baseline metrics are represented in the section titled “Document Data Quality Changes” as well as included in the supplementary file named: CS513-PhaseII-QualityMeasures.xlsx

INTERMEDIATE CLEANUP

The data cleanup steps performed in OpenRefine were valuable for performing cleanup that was sensible within the boundaries of each input file. All relational cleanup work was performed in SQLite. The following cleanup operations were performed in SQLite.

1. Remove Duplicate (Clustered) Dish Records While Preserving Statistics

The OpenRefine cleanup operation saved a new name_clean property that represented the output of the clustering operation on the Dish name property. It is important that we preserve only a single row representing the cluster. However, in order to achieve the desired output, we need to update all MenuItem references to Dish.id to reference the updated Dish.clustered_id.

2. Recalculate Menu Dish Counts
3. Recalculate Dish Aggregate Data
    a. Rather than rely on supplied data (which was often incorrect), this process recalculates the following Dish record attributes to represent accurate data based on MenuItem, MenuPage, and Menu datasets:
        i. Menus_appeared
        ii. Times_appeared
        iii. First_appeared
        iv. Last_appeared
        v. Lowest_price
        vi. Highest_price
4. Remove Unreferenced Records
    a. Remove Menu Pages that reference a non-existent menu_id.
    b. Remove Menu Items that reference a non-existent menu_page_id.
    c. Remove Menu Items that reference a non-existent dish_id.
    d. Remove Dishes that are not referenced by any Menu Items.
5. Extract categorized properties from physical_description and push to the following attributes on each Menu record:
    a. Misc Properties (extracted_misc)
    b. Damage Descriptions (extracted_damage)
    c. Dimension Descriptions (extracted_dimensions)
    d. Document Format Descriptions (extracted_format)
6. Remove unneeded attributes:
    a. Attributes that were always empty and do not contribute to any Use Case operations should be removed.
        i. Menu.keywords
        ii. Menu.language
        iii. Menu.location_type
        iv. Dish.description

FINAL MEASUREMENTS

The final steps performed in SQLite were to run the same predefined IC check queries to determine the number of records in violation of each IC in order to quantify the data quality improvements achieved.

The cleansed metrics are represented in the section titled “Document Data Quality Changes” as well as included in the supplementary file named: CS513-PhaseII-QualityMeasures.xlsx

DETAILS STEPS

1. Add clustered_id to Dish, seed with existing id value;
2. Populated DishIntermediate with starting data from OpenRefine Dish export, only generate one row per unique name_clean, and assign the lowest (first occurring) Dish.id value as DishIntermediate.clustered_id.

3. Populate MenuIntermediate with starting data from OpenRefine Menu export.
4. Override Dish.name with the cleaned version from name_clean (**423397 records affected**).
5. Update all MenuItem records to reference the clustered_id that corresponds to the original Dish.id value that was stored in MenuItem.dish_id (**236813 records effected**).
6. Delete all Dish records that were "clustered" into another record (i.e. remove all rows where Dish.id is not used as any instance of Dish.clustered_id) (**236813 records effected**).
7. Calculate DishIntermediate.menus_appeared based on relationship from Dish through MenuItem through MenuPage to Menu.
8. Calculate DishIntermediate.times_appeared based on relationship from Dish through to MenuItem.
9. Calculate DishIntermediate.first_appeared by extracting the year from the earliest date for a Dish as it relates to Menu through MenuItem and MenuPage (using Menu.date_clean).
10. Calculate DishIntermediate.last_appeared by extracting the year from the latest date for a Dish as it relates to Menu through MenuItem and MenuPage (using Menu.date_clean).
11. Calculate DishIntermediate.lowest_price by extracting the minimum value from price and higher_price for a Dish as it relates to MenuItem.
12. Calculate DishIntermediate.highest_price by extracting the maximum value from price and higher_price for a Dish as it relates to MenuItem.
13. Push recalculated menus_appeared from DishIntermediate back on to corresponding Dish records. (**57582 records effected**)
14. Push recalculated times_appeared from DishIntermediate back on to corresponding Dish records. (**58224 records effected**)
15. Push recalculated first_appeared from DishIntermediate back on to corresponding Dish records. (**29035 records effected**)
16. Push recalculated last_appeared from DishIntermediate back on to corresponding Dish records. (**32186 records effected**)
17. Push recalculated lowest_price from DishIntermediate back on to corresponding Dish records. (**70989 records effected**)
18. Push recalculated highest_price from DishIntermediate back on to corresponding Dish records. (**71774 records effected**)
19. Remove MenuPage records that do not reference an existing menu.id value. (**5803 records effected**)
20. Remove MenuItem records that do not reference an existing menupage.id value. (**5373 records effected**)
21. Remove MenuItem records that do not reference an existing dish.id value. (**244 records effected**)
22. Remove Dish records that are not referenced by any MenuItem.dish_id values. (**2285 records effected**)
23. Calculate MenuIntermediate.dish_count based on the number of MenuItem records that reference each menu through MenuPage.
24. Push recalculated dish_count from MenuIntermediate back on to corresponding Menu records. (**4775 records effected**)
25. Trim leading/trailing whitespace in OpenRefine extracted physical_description1, physical_description2, physical_description3, physical_description4, physical_description5, physical_description6, physical_description7 fields.
26. Populate Menu attributes: extracted_misc, extracted_dimensions, extracted_damage, extracted_format based on physical_description# flags manually categorized in an Excel spreadsheet (see: **property-cleanup.xlsx** submitted with this assignment).
27. Drop physical_description1 through physical_description7 fields from Menu.
28. Drop DishIntermediate table;
29. Drop MenuIntermediate table;

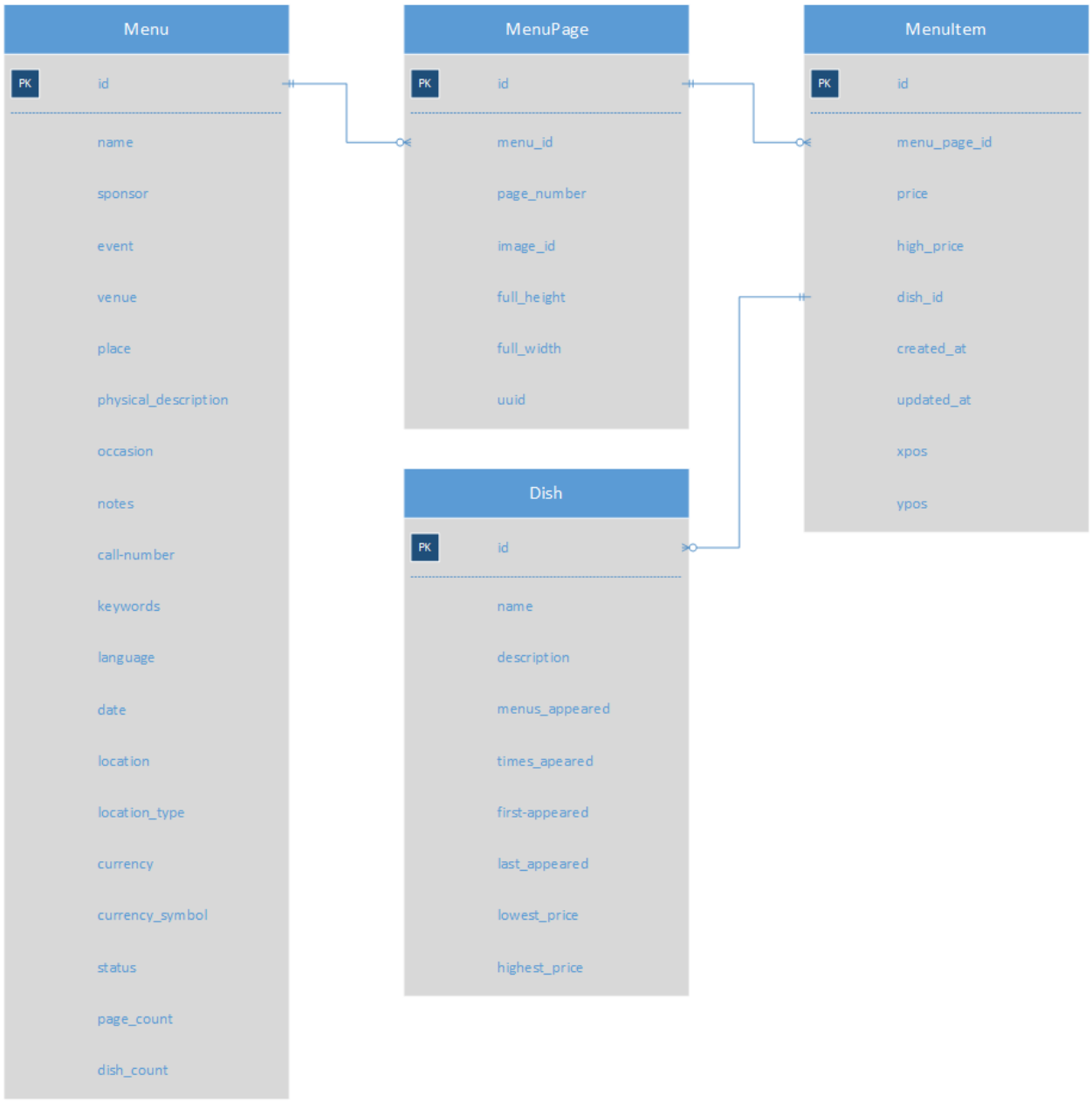
30. Drop fields that are completely empty:
    a. Menu:
        i. keywords
        ii. language
        iii. location_type
    b. Dish
        i. Description

SQLITE SCHEMA ITERATIONS

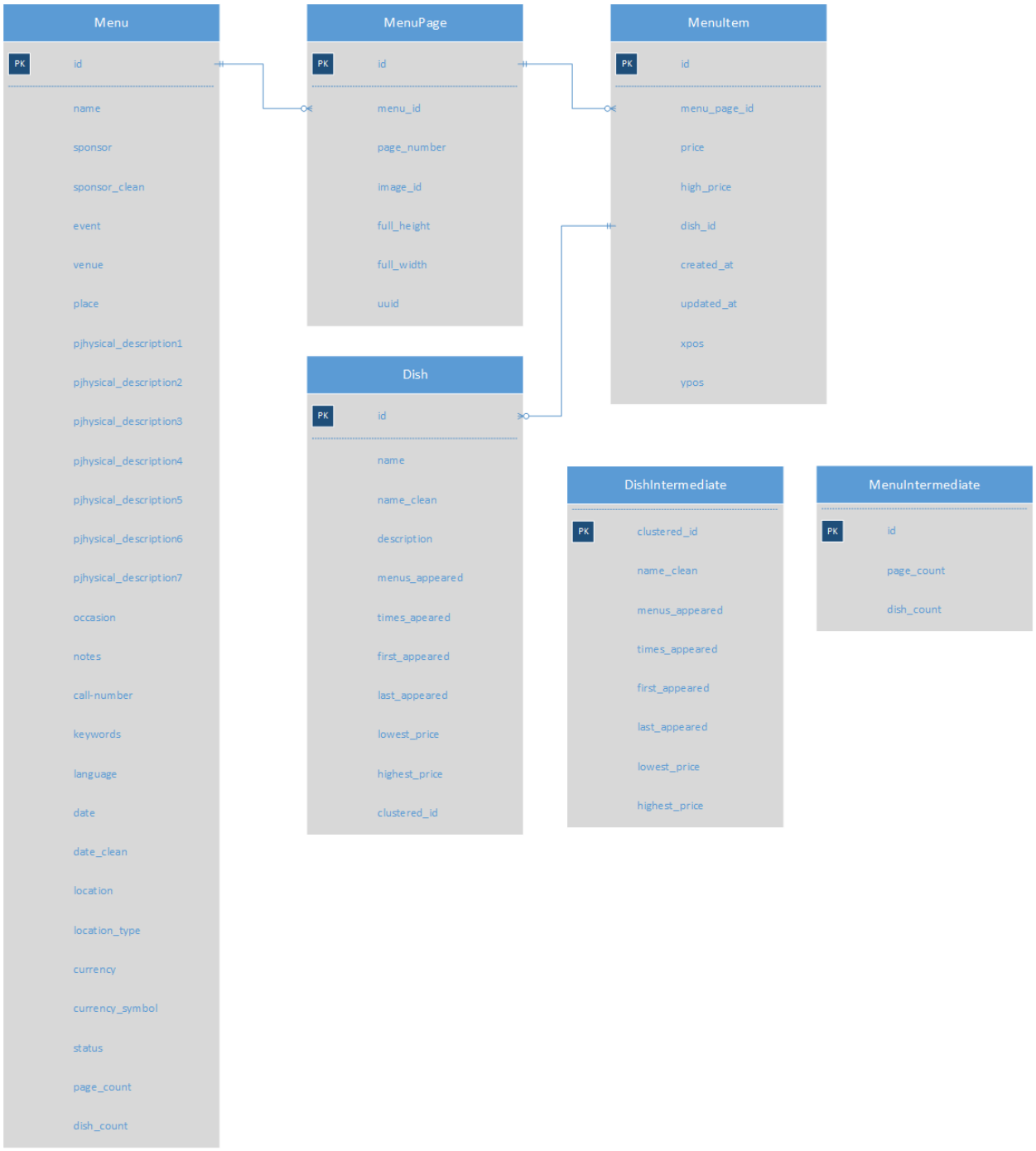
A set of three SQLite data model iterations were used during the cleanup activity for this project.

1. Initial Schema
    a. This schema represents the schema of the original data set and is identical to the schema included in our Phase I report.
2. Intermediate Schema
    a. This schema represents the schema of the data as it was emitted from OpenRefine after the OpenRefine-based cleanup operations were completed.
    b. For any notably transformative cleanup operations in OpenRefine (e.g., Clustering, or Flag-Expansion for the Physical Description attributes), the original attribute was preserved, and the new/cleansed attribute was added. This was necessary in order to preserve MenuItem to Dish relationships when deduplicating.
    c. This schema also contains intermediate tables (MenuIntermediate and DishIntermediate) whose sole purpose is to support performant relational statistic recalculations for aggregate values on Menu and Dish tables.
3. Final Schema
    a. This schema represents the final state of the cleansed data set.
    b. This includes the elimination of both unwanted attributes (e.g., attributes with no valid data) as well as intermediate attributes (e.g., attributes we created to support FK-preservation during deduplication).

INITIAL SCHEMA



INTERMEDIATE SCHEMA



FINAL SCHEMA

SQLITE CONSTRAINTS

MENU TABLE

MISSING ID

SELECT id FROM Menu WHERE id is null or id = ''

MISSING NAME

SELECT id FROM Menu WHERE name is null or name = ''

MISSING SPONSOR

SELECT id FROM Menu WHERE sponsor is null or sponsor = ''

MISSING EVENT

SELECT id FROM Menu WHERE event is null or event = ''

MISSING VENUE

SELECT id FROM Menu WHERE venue is null or venue = ''

MISSING PHYSICAL DESCRIPTION

SELECT id FROM Menu WHERE physical_description is null or physical_description = ''

MISSING PLACE

SELECT id FROM Menu WHERE place is null or place = ''

MISSING OCCASION

SELECT id FROM Menu WHERE occasion is null or occasion = ''

MISSING NOTES

SELECT id FROM Menu WHERE notes is null or notes = ''

MISSING CALL NUMBER

SELECT id FROM Menu WHERE call_number is null or call_number = ''

MISSING KEYWORDS

SELECT id FROM Menu WHERE keywords is null or keywords = ''

MISSING LANGUAGE

SELECT id FROM Menu WHERE language is null or language = ''

MISSING DATE

SELECT id FROM Menu WHERE date is null or date = ''

MISSING LOCATION

SELECT id FROM Menu WHERE location is null or location = ''

MISSING LOCATION TYPE

SELECT id FROM Menu WHERE location_type is null or location_type = ''

MISSING CURRENCY

SELECT id FROM Menu WHERE currency is null or currency = ''

MISSING CURRENCY SYMBOL

SELECT id FROM Menu WHERE currency_symbol is null or currency_symbol = ''

MISSING STATUS

SELECT id FROM Menu WHERE status is null or status = ''

MISISNG PAGE COUNT

SELECT id FROM Menu WHERE page_count is null or page_count = ''

MISSING DISH COUNT

SELECT id FROM Menu WHERE dish_count is null or dish_count = '')

NON-UNIQUE ID

SELECT COUNT(*), id FROM Menu GROUP BY id HAVING COUNT(*) >= 2)

NO PAGES

SELECT id FROM Menu WHERE page_count = 0 or page_count IS NULL)

NO DISHES

SELECT id FROM Menu WHERE dish_count = 0 or dish_count IS NULL)

UNREALISTICALLY EARLY DATE

SELECT id FROM Menu WHERE datetime(date) < datetime('1850-01-01'))

UNREALISTICALLY LATE DATE

SELECT id FROM Menu WHERE datetime(date) > datetime('2023-08-01'))

MENU NOT REFERENCED BY ANY MENU PAGES

SELECT id FROM Menu WHERE id NOT IN (select menu_id from MenuPage))

DISH TABLE

MISSING ID

SELECT id FROM Dish WHERE id is null or id = ''

MISSING NAME

SELECT id FROM Dish WHERE name is null or name = ''

MISSING DESCRIPTION

SELECT id FROM Dish WHERE description is null or description = ''

MISSING MENUS APPEARED

SELECT id FROM Dish WHERE menus_appeared is null or menus_appeared = ''

MISSING TIMES APPEARED

SELECT id FROM Dish WHERE times_appeared is null or times_appeared = ''

MISSING FIRST APPEARED

SELECT id FROM Dish WHERE first_appeared is null or first_appeared = ''

MISSING LAST APPEARED

SELECT id FROM Dish WHERE last_appeared is null or last_appeared = ''

MISSING LOWEST PRICE

SELECT id FROM Dish WHERE lowest_price is null or lowest_price = ''

MISSING HIGHEST PRICE

SELECT id FROM Dish WHERE highest_price is null or highest_price = '')

NON-UNIQUE ID

SELECT COUNT(*), id FROM Dish GROUP BY id HAVING COUNT(*) >= 2)

ZERO MENUS APPEARED

SELECT id FROM Dish WHERE menus_appeared = 0 or menus_appeared IS NULL)

ZERO TIMES APPEARED

SELECT id FROM Dish WHERE times_appeared = 0 or times_appeared IS NULL)

ZERO FIRST YEAR APPEARED

SELECT id FROM Dish WHERE first_appeared = 0 or first_appeared IS NULL)

ZERO LAST YEAR APPEARED

SELECT id FROM Dish WHERE last_appeared = 0 or last_appeared IS NULL)

NO LOW PRICE

SELECT id FROM Dish WHERE lowest_price = 0 or lowest_price IS NULL)

NO HIGH PRICE

SELECT id FROM Dish WHERE highest_price = 0 or highest_price IS NULL)

DOES NOT APPEAR IN ANY MENUITEM RECORDS

SELECT id FROM Dish WHERE id NOT IN (select dish_id from MenuItem))

HIGH PRICE IS LESS THAN LOW PRICE

SELECT id FROM Dish WHERE highest_price < lowest_price)

UNREALISTICALLY HIGH LOW-PRICE

SELECT id FROM Dish WHERE lowest_price > 500)

UNREALISTICALLY HIGH HIGH-PRICE

SELECT id FROM Dish WHERE highest_price > 500)

NEGATIVE LOW PRICE

SELECT id FROM Dish WHERE lowest_price < 0)

NEGATIVE HIGH PRICE

SELECT id FROM Dish WHERE highest_price < 0)

TIMES APPEARED NOT IN SYNC WITH RELATIONSHIPS

SELECT id FROM Dish WHERE id NOT IN (select dish_id from MenuItem) and times_appeared > 0);

MENUPAGE TABLE

MISSING ID

SELECT id FROM MenuPage WHERE id is null or id = ''

MISSING MENU ID

SELECT id FROM MenuPage WHERE menu_id is null or menu_id = ''

MISSING PAGE NUMBER

SELECT id FROM MenuPage WHERE page_number is null or page_number = ''

MISSING IMAGE ID

SELECT id FROM MenuPage WHERE image_id is null or image_id = ''

MISSING FULL HEIGHT

SELECT id FROM MenuPage WHERE full_height is null or full_height = ''

MISSING FULL WIDTH

SELECT id FROM MenuPage WHERE full_width is null or full_width = '') UNION

MISSING UUID

SELECT id FROM MenuPage WHERE uuid is null or uuid = '')

NON-UNIQUE ID

SELECT COUNT(*), id FROM MenuPage GROUP BY id HAVING COUNT(*) >= 2)

ZERO MENU ID

SELECT id FROM MenuPage WHERE menu_id = 0 or menu_id IS NULL)

INVALID MENU ID (FK VIOLATION)

SELECT id FROM MenuPage WHERE menu_id NOT IN (SELECT ID FROM Menu))

ZERO OR NEGATIVE PAGE NUMBER

SELECT id FROM MenuPage WHERE page_number <= 0 or page_number IS NULL)

PAGE NUMBER OUT OF MENU PAGE RANGE

SELECT mp.id FROM MenuPage mp LEFT JOIN Menu m ON mp.menu_id = m.id WHERE mp.page_number > m.page_count)

MORE PAGES FOR THIS MENU IN MENUPAGE THAN REPRESENTED IN MENU.PAGE_COUNT

SELECT mp.id FROM MenuPage mp LEFT JOIN Menu m ON mp.menu_id = m.id WHERE mp.page_number > m.page_count)

NOT REFERENCED BY ANY MENUITEM

SELECT id FROM MenuPage WHERE id NOT IN (select menu_page_id from MenuItem))

MORE PAGES IN MENU THAN REPRESENTED IN MENUPAGE

SELECT m.id FROM Menu m LEFT JOIN (SELECT COUNT(*) as page_count, menu_id FROM MenuPage GROUP BY menu_id) mp on m.id = mp.menu_id WHERE mp.page_count < m.page_count)

FEWER PAGES IN MENU THAN REPRESENTED IN MENUPAGE

SELECT m.id FROM Menu m LEFT JOIN (SELECT COUNT(*) as page_count, menu_id FROM MenuPage GROUP BY menu_id) mp on m.id = mp.menu_id WHERE mp.page_count > m.page_count);

MENUITEM TABLE

MISSING ID

SELECT id FROM MenuItem WHERE id is null or id = ''

MISSING MENU PAGE ID

SELECT id FROM MenuItem WHERE menu_page_id is null or menu_page_id = ''

MISSING PRICE

SELECT id FROM MenuItem WHERE price is null or price = ''

MISSING HIGH PRICE

SELECT id FROM MenuItem WHERE high_price is null or high_price = ''

MISSING DISH ID

SELECT id FROM MenuItem WHERE dish_id is null or dish_id = ''

MISSING CREATED AT

SELECT id FROM MenuItem WHERE created_at is null or created_at = ''

MISSING UPDATED AT

SELECT id FROM MenuItem WHERE updated_at is null or updated_at = ''

MISSING X POSITINO

SELECT id FROM MenuItem WHERE xpos is null or xpos = ''

MISSING Y POSITION

SELECT id FROM MenuItem WHERE ypos is null or ypos = '')

NON-UNIQUE ID

SELECT COUNT(*), id FROM MenuItem GROUP BY id HAVING COUNT(*) >= 2)

ZERO MENU PAGE ID

SELECT id FROM MenuItem WHERE menu_page_id = 0 or menu_page_id IS NULL)

ZERO DISH ID

SELECT id FROM MenuItem WHERE dish_id = 0 or dish_id IS NULL)

INVALID MENU PAGE ID (FK CONSTRAINT)

SELECT id FROM MenuItem WHERE menu_page_id NOT IN (SELECT ID FROM MenuPage))

INVALID DISH ID (FK CONSTRAINT)

SELECT id FROM MenuItem WHERE dish_id NOT IN (SELECT ID FROM Dish))

UPDATED TIMESTAMP EARLIER THAN CREATION TIMESTAMP

SELECT id FROM MenuItem WHERE datetime(updated_at) < datetime(created_at))

PRICE ON ITEM IS LOWER THAN DISH LOWEST PRICE

SELECT mi.id from MenuItem mi INNER JOIN Dish d on mi.dish_id = d.id WHERE mi.price < d.lowest_price and mi.price != '' and mi.price is not null)

PRICE ON ITEM IS HIGHER THAN DISH HIGHEST PRICE

SELECT mi.id from MenuItem mi INNER JOIN Dish d on mi.dish_id = d.id WHERE mi.price > d.highest_price and mi.price != '' and mi.price is not null)

HIGH PRICE ON ITEM IS LOWER THAN DISH LOWEST PRICE

SELECT mi.id from MenuItem mi INNER JOIN Dish d on mi.dish_id = d.id WHERE mi.high_price is not null and mi.high_price != '' and mi.high_price < d.lowest_price)

HIGH PRICE ON ITEM IS HIGHER THAN DISH HIGHEST PRICE

SELECT mi.id from MenuItem mi INNER JOIN Dish d on mi.dish_id = d.id WHERE mi.high_price is not null and mi.high_price != '' and mi.high_price > d.highest_price);

2. DOCUMENT DATA QUALITY CHANGES

All IC checks were chained together in a series of queries to be executed in the SQLite environment enabling us to compare the values produced by these checks prior to the data cleanup being executed and after the data cleanup being executed. Please review the table below for a summary of outcomes. Each constraint will correspond to a constraint defined in the previous section.

Table	Measure	Total Rows	Raw Data	%	Clean Total Rows	Clean Data	%	Delta	% Delta
Menu	Total Rows	17545	17545	100.00%	17545	17545	100.00%	0	
MenuPage	Total Rows	66937	66937	100.00%	61134	61134	100.00%	5803	
Dish	Total Rows	423397	423397	100.00%	184299	184299	100.00%	239098	
MenuItem	Total Rows	1332726	1332726	100.00%	1327109	1327109	100.00%	5617	
Menu	Missing Attribute: Dish Count	17545	0	0.00%	17545	0	0.00%	0	0.00%
Menu	Missing Attribute: ID	17545	0	0.00%	17545	0	0.00%	0	0.00%
Menu	Missing Attribute: Location	17545	0	0.00%	17545	48	0.27%	-48	-0.27%
Menu	Missing Attribute: Page Count	17545	0	0.00%	17545	0	0.00%	0	0.00%
Menu	Missing Attribute: Status	17545	0	0.00%	17545	0	0.00%	0	0.00%
Menu	Missing Attribute: Date	17545	586	3.34%	17545	586	3.34%	0	0.00%
Menu	Missing Attribute: Sponsor	17545	1561	8.90%	17545	1561	8.90%	0	0.00%
Menu	Missing Attribute: Call Number	17545	1562	8.90%	17545	1562	8.90%	0	0.00%
Menu	Missing Attribute: Physical Description	17545	2777	15.83%	17545	0	0.00%	2777	15.83%
Menu	Missing Attribute: Notes	17545	6932	39.51%	17545	6933	39.52%	-1	-0.01%
Menu	Missing Attribute: Event	17545	9391	53.53%	17545	9409	53.63%	-18	-0.10%
Menu	Missing Attribute: Venue	17545	9414	53.66%	17545	9435	53.78%	-21	-0.12%
Menu	Missing Attribute: Place	17545	9422	53.70%	17545	9507	54.19%	-85	-0.48%
Menu	Missing Attribute: Currency	17545	11089	63.20%	17545	11089	63.20%	0	0.00%
Menu	Missing Attribute: Currency Symbol	17545	11089	63.20%	17545	11089	63.20%	0	0.00%

Menu	Missing Attribute: Occasion	17545	13742	78.32%	17545	13793	78.61%	-51	-0.29%
Menu	Missing Attribute: Name	17545	14348	81.78%	17545	14348	81.78%	0	0.00%
Menu	Missing Attribute: Keywords	17545	17545	100.00%	17545	17545	100.00%	0	0.00%
Menu	Missing Attribute: Language	17545	17545	100.00%	17545	17545	100.00%	0	0.00%
Menu	Missing Attribute: Location Type	17545	17545	100.00%	17545	17545	100.00%	0	0.00%
Menu	Non-Unique ID	17545	0	0.00%	17545	0	0.00%	0	0.00%
Menu	NULL ID	17545	0	0.00%	17545	0	0.00%	0	0.00%
Menu	Zero MenuPage Records related to this Menu	17545	0	0.00%	17545	0	0.00%	0	0.00%
Menu	Zero Pages in Page Count Attribute	17545	0	0.00%	17545	0	0.00%	0	0.00%
Menu	Zero Dishes	17545	32	0.18%	17545	28	0.16%	4	0.02%
Menu	Very Early Date	17545	4	0.02%	17545	4	0.02%	0	0.00%
Menu	Very Late Date	17545	1	0.01%	17545	1	0.01%	0	0.00%
Dish	Missing Attribute: First Appeared	423397	0	0.00%	184299	0	0.00%	0	0.00%
Dish	Missing Attribute: Id	423397	0	0.00%	184299	0	0.00%	0	0.00%
Dish	Missing Attribute: Last Appeared	423397	0	0.00%	184299	0	0.00%	0	0.00%
Dish	Missing Attribute: Menus Appeared	423397	0	0.00%	184299	0	0.00%	0	0.00%
Dish	Missing Attribute: Name	423397	0	0.00%	184299	0	0.00%	0	0.00%
Dish	Missing Attribute: Times Appeared	423397	0	0.00%	184299	0	0.00%	0	0.00%
Dish	Missing Attribute: Highest Price	423397	29100	6.87%	184299	9678	5.25%	19422	1.62%
Dish	Missing Attribute: Lowest Price	423397	29100	6.87%	184299	9678	5.25%	19422	1.62%
Dish	Missing Attribute: Description	423397	423397	100.00%	184299	184299	100.00%	239098	0.00%
Dish	Non-Unique ID	423397	0	0.00%	184299	0	0.00%	0	0.00%
Dish	NULL ID	423397	0	0.00%	184299	0	0.00%	0	0.00%
Dish	Relationally Appeared in Zero Menus	423397	9262	2.19%	184299	0	0.00%	9262	2.19%
Dish	Negative High Price	423397	0	0.00%	184299	0	0.00%	0	0.00%
Dish	Negative Low Price	423397	0	0.00%	184299	0	0.00%	0	0.00%
Dish	Dish not referenced but MenuItem, but	423397	9	0.00%	184299	0	0.00%	9	0.00%

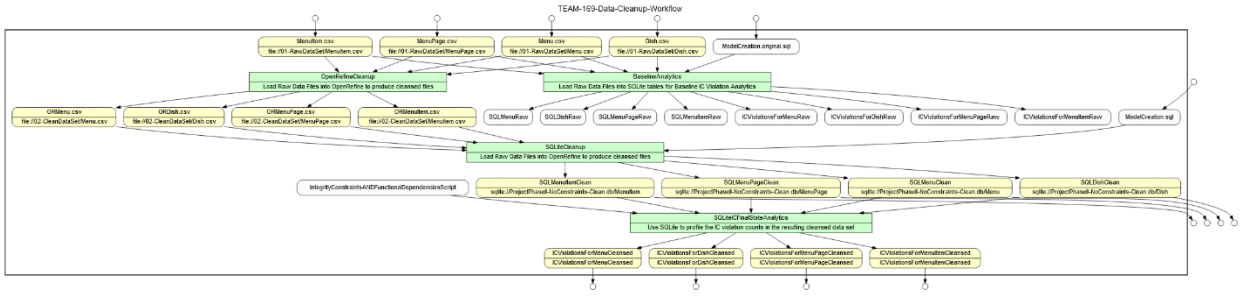
	Times Appeared > 0								
Dish	Zero Menu Appeared	423397	2271	0.54%	184299	0	0.00%	2271	0.54%
Dish	Zero Times Appeared	423397	9248	2.18%	184299	0	0.00%	9248	2.18%
Dish	Zero First Appeared	423397	55284	13.06%	184299	18310	9.93%	36974	3.12%
Dish	Zero Last Appeared	423397	55287	13.06%	184299	18317	9.94%	36970	3.12%
Dish	Zero Highest Price	423397	218014	51.49%	184299	48183	26.14%	169831	25.35%
Dish	Zero Lowest Price	423397	222566	52.57%	184299	48373	26.25%	174193	26.32%
MenuPage	Missing Attribute: Id	66937	0	0.00%	61134	0	0.00%	0	0.00%
MenuPage	Missing Attribute: Image Id	66937	0	0.00%	61134	0	0.00%	0	0.00%
MenuPage	Missing Attribute: Menu Id	66937	0	0.00%	61134	0	0.00%	0	0.00%
MenuPage	Missing Attribute: Uuid	66937	0	0.00%	61134	0	0.00%	0	0.00%
MenuPage	Missing Attribute: Full Height	66937	329	0.49%	61134	0	0.00%	329	0.49%
MenuPage	Missing Attribute: Full Width	66937	329	0.49%	61134	0	0.00%	329	0.49%
MenuPage	Missing Attribute: Page Number	66937	1202	1.80%	61134	945	1.55%	257	0.25%
MenuPage	Non-Unique ID	66937	0	0.00%	61134	0	0.00%	0	0.00%
MenuPage	NULL ID	66937	0	0.00%	61134	0	0.00%	0	0.00%
MenuPage	Relational Page Count Mismatch - Actual Count Higher	66937	0	0.00%	61134	0	0.00%	0	0.00%
MenuPage	Relational Page Count Mismatch - Actual Count Lower	66937	0	0.00%	61134	0	0.00%	0	0.00%
MenuPage	Zero Menu ID	66937	0	0.00%	61134	0	0.00%	0	0.00%
MenuPage	Page Number Out of Range	66937	951	1.42%	61134	951	1.56%	0	-0.13%
MenuPage	Relational - Page Count Aggregate	66937	951	1.42%	61134	951	1.56%	0	-0.13%
MenuPage	Invalid Menu ID	66937	5803	8.67%	61134	0	0.00%	5803	8.67%
MenuPage	Page Not Referenced by Menu Item	66937	40347	60.28%	61134	34656	56.69%	5691	3.59%
MenuItem	Missing Attribute: Created At	1332726	0	0.00%	1327109	0	0.00%	0	0.00%
MenuItem	Missing Attribute: Id	1332726	0	0.00%	1327109	0	0.00%	0	0.00%
MenuItem	Missing Attribute: Menu Page Id	1332726	0	0.00%	1327109	0	0.00%	0	0.00%

MenuItem	Missing Attribute: Updated At	1332726	0	0.00%	1327109	0	0.00%	0	0.00%
MenuItem	Missing Attribute: Xpos	1332726	0	0.00%	1327109	0	0.00%	0	0.00%
MenuItem	Missing Attribute: Ypos	1332726	0	0.00%	1327109	0	0.00%	0	0.00%
MenuItem	Missing Attribute: Dish Id	1332726	241	0.02%	1327109	0	0.00%	241	0.02%
MenuItem	Missing Attribute: Price	1332726	445916	33.46%	1327109	444424	33.49%	1492	-0.03%
MenuItem	Missing Attribute: High Price	1332726	1240821	93.10%	1327109	1235450	93.09%	5371	0.01%
MenuItem	Invalid Menu Page ID	1332726	0	0.00%	1327109	0	0.00%	0	0.00%
MenuItem	Non-Unique ID	1332726	0	0.00%	1327109	0	0.00%	0	0.00%
MenuItem	NULL ID	1332726	0	0.00%	1327109	0	0.00%	0	0.00%
MenuItem	Zero Dish ID	1332726	0	0.00%	1327109	0	0.00%	0	0.00%
MenuItem	Zero Menu Page ID	1332726	0	0.00%	1327109	0	0.00%	0	0.00%
MenuItem	Higher Price Lower than Low Price on Dish (relational)	1332726	58	0.00%	1327109	0	0.00%	58	0.00%
MenuItem	Invalid Dish ID	1332726	244	0.02%	1327109	0	0.00%	244	0.02%
MenuItem	Price Lower Than Lowest Price on Dish (relational)	1332726	1172	0.09%	1327109	0	0.00%	1172	0.09%
MenuItem	High Price Higher than Highest Price on Dish (relational)	1332726	13591	1.02%	1327109	0	0.00%	13591	1.02%
MenuItem	Price Higher than Highest Price on Dish (relational)	1332726	149616	11.23%	1327109	0	0.00%	149616	11.23%

3. WORKFLOW MODEL

All workflow images have a corresponding higher-resolution PNG or JPG file included in the project submission for situations where the readability is difficult due to image scaling. See Supplementary Materials section for more details.

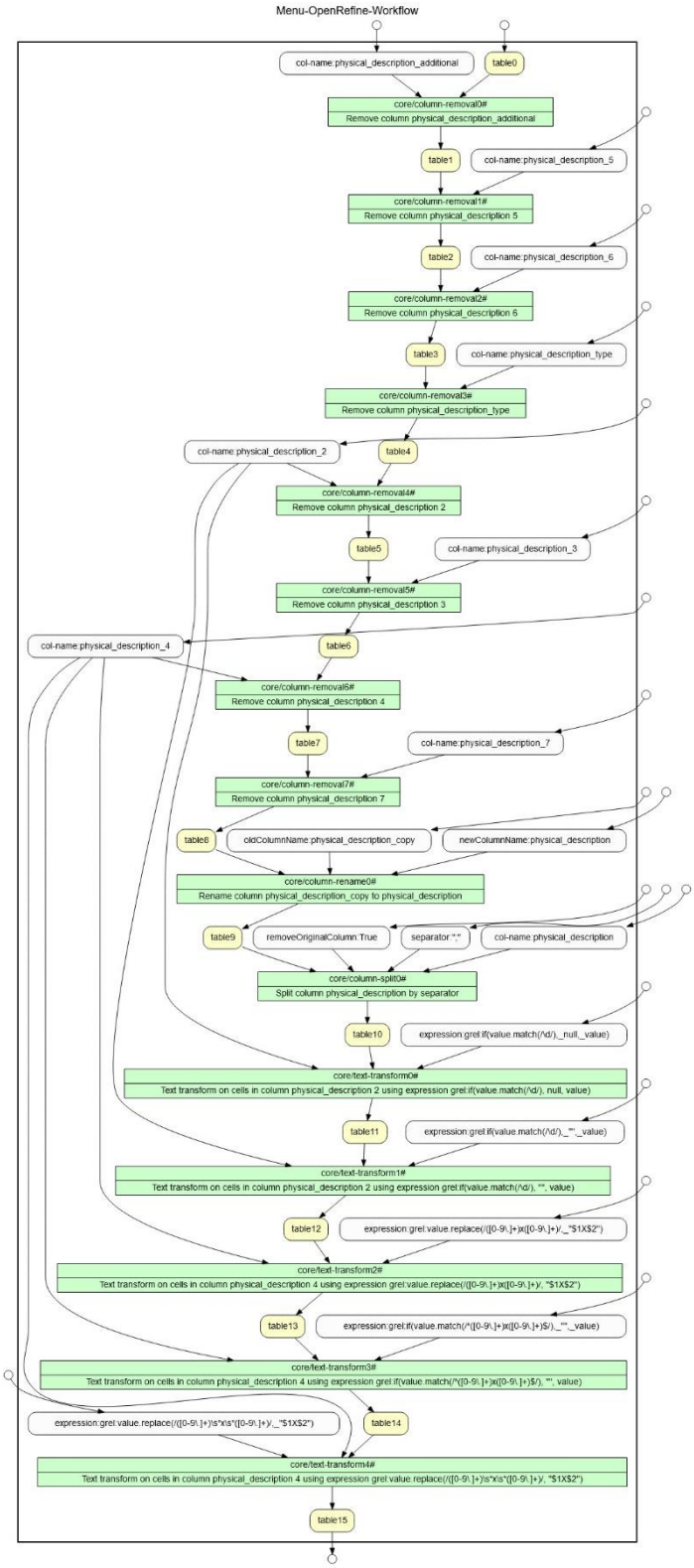
OUTER WORKFLOW



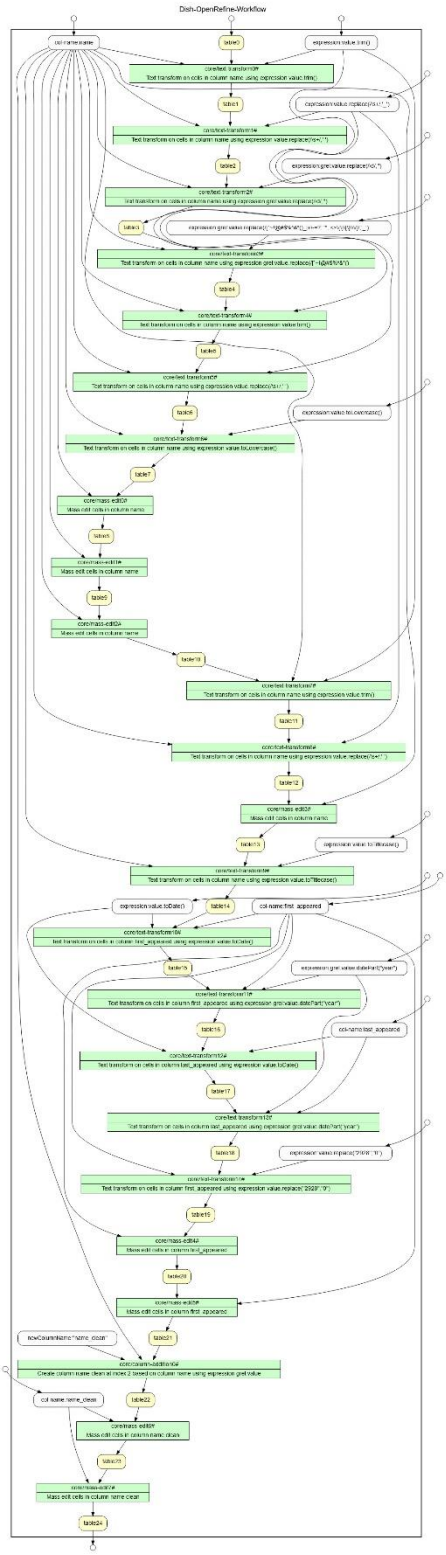
INNER WORKFLOW

OPENREFINE

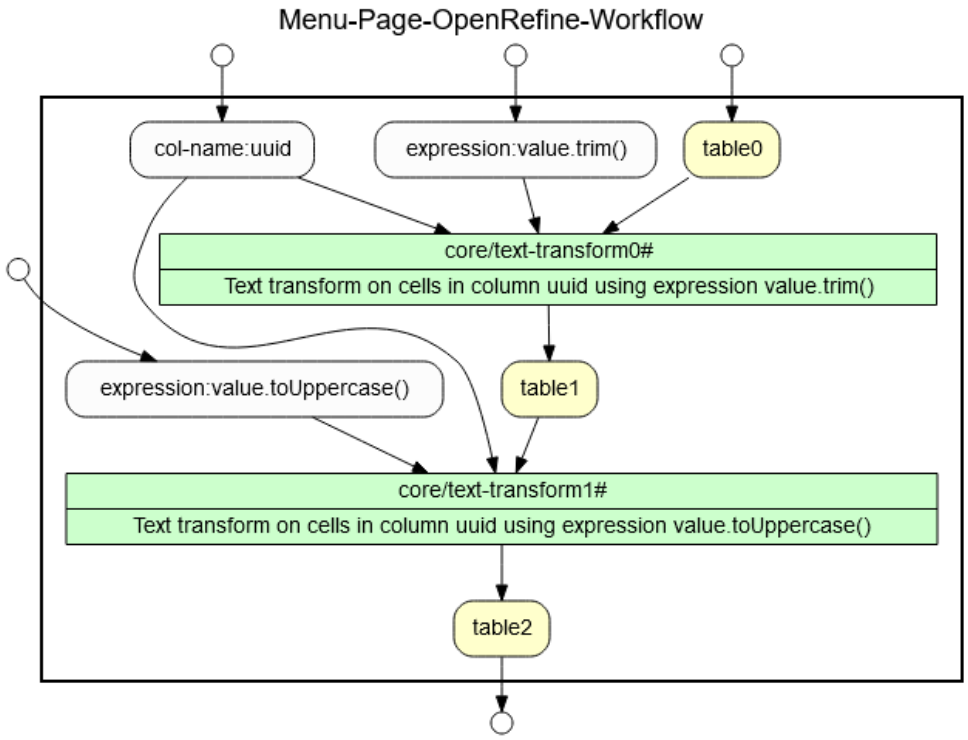
MENU



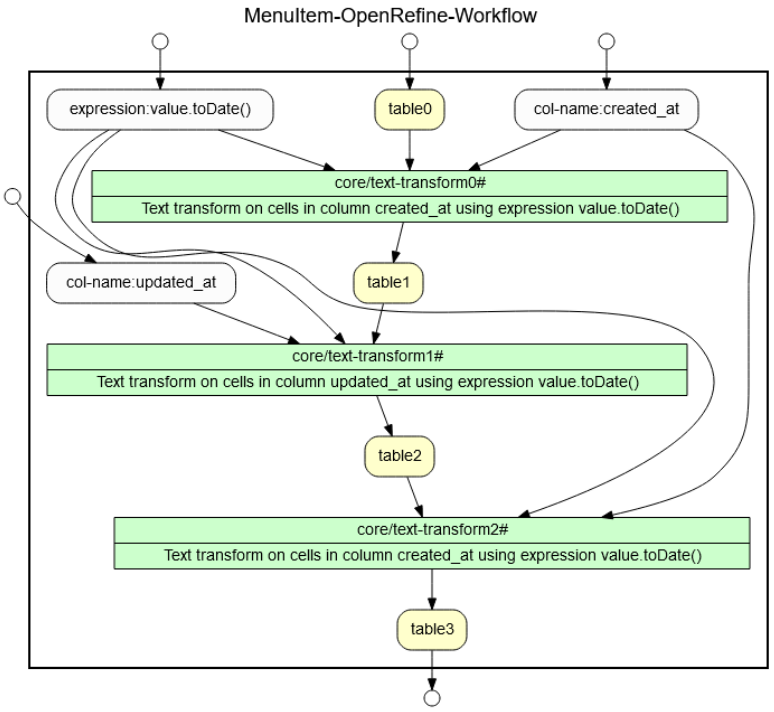
DISH



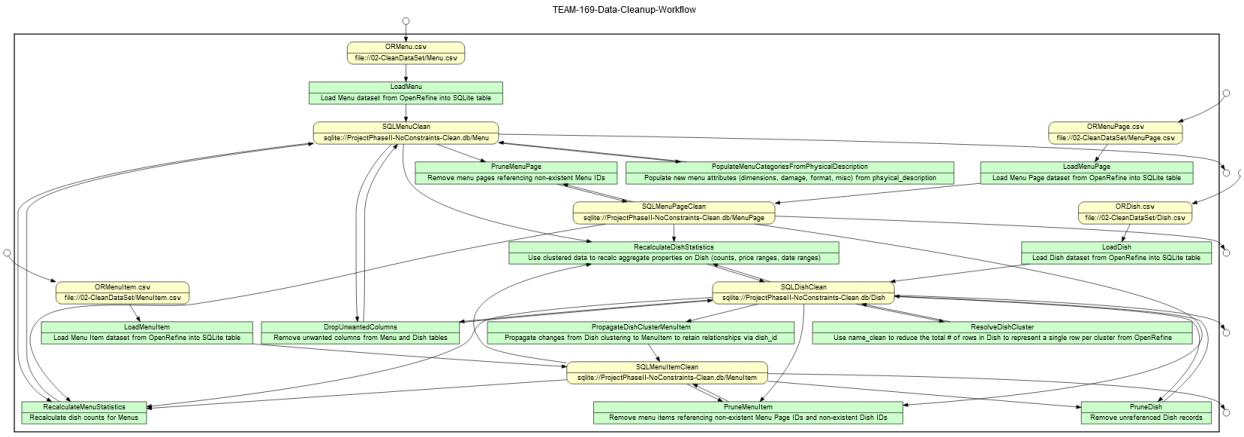
MENUPAGE



MENUITEM



SQLITE



4. CONCLUSIONS & SUMMARY

The processes described in this report have demonstrated a high value in improving the data quality, particularly with aggregate data calculation between data entities, in service of our U1 Main Use Case.

Both the OpenRefine and SQLite processes implemented in service of this cleanup project were valuable in achieving the goals we set out to accomplish.

Some moderately cumbersome SQLite conventions were used (i.e., frequent use of UPSERT operations) to achieve high-speed execution of our SQLite based operations. While these SQL expressions were not as concise as ones that would have been described using a SQL platform that supported UPDATE operations with JOIN conditions, the UPSERT conventions used did enable us to implement a SQL-based cleanup process that was able to execute in a very high-speed fashion.

CHALLENGES

The key challenge experienced in this activity was the late realization that the low and high price aggregate representations produce a moderately inaccurate price range given the inconsistency between currency and currency units across different representations of the same dish.

The team agreed not to pursue any currency normalization (i.e., rebase the aggregate price calculations using a normalized currency representation based on historical foreign exchange rates for the involved currencies). The team made this decision based on the belief that this expectation was not set out by our team in our original Project Plan. Were this activity to be undertaken again, the team would consider either currency normalization or representing low/high price ranges in all available currencies for any Dish that is priced in any Menu using that currency.

TEAM MEMBER CONTRIBUTIONS

The group regularly engaged in working sessions where the full team was present on a call while work was being conducted. The table below summarizes the areas where each members contributions were central.

TEAM MEMBER	PRIMARY AREAS OF CONTRIBUTION
AARON	SQLite Operations, Yes Workfow Definition, Report preparation
CHIRU	OpenRefine Cleanup Process, Yes Workflow Definition
WILLY	OpenRefine Cleanup Process, Yes Workflow Definition, OpenRefine process documentation

5. SUPPLEMENTARY MATERIALS

The following materials were submitted with the report:

- Property-cleanup.xlsx
  - Spreadsheet representing the categorization work performed on flag values extracted from physical_description in the Menu data set.
- CS513-PhaseII-QualityMeasures.xlsx
  - Spreadsheet version of the quantified data quality changes included in section 2.
- OpenRefine-Project.zip
  - OpenRefine project files representing the cleanup activities and provenance for the OpenRefine based cleanup operations described in the “OpenRefine Actions” section.
- OpenRefine-CleansedOutput.zip
  - ZIP containing 4x CSV files representing the intermediate state cleaned data for each of the four tables produced as a result of OpenRefine cleanup operations..
- SQLite-CleansedOutput.zip
  - ZIP containing 4x CSV files representing the final state cleaned data for each of the four tables represented in our initial and final schema.
- ModelCreation.original.sql
  - Script file used to load and operate the baseline data quality assessment on the raw input files in SQLite.
- ModelCreation.sql
  - Script file used to load the OpenRefine-cleansed data files, operate the SQLite data cleanup, and operate the finish-line data quality assessment.
- YesWorkflow-Outer.png
  - Detailed PNG for workflow represented in Workflow Model – Outer Workflow section.
- YesWorkflow-Inner-OpenRefine-Menu.jpg
- YesWorkflow-Inner-OpenRefine-Dish.jpg
- YesWorkflow-Inner-OpenRefine-MenuItem.png
- YesWorkflow-Inner-OpenRefine-MenuPage.png
  - Detailed PNG/JPG files for workflow represented in Workflow Model – Inner Workflow – OpenRefine section.
- YesWorkflow-Inner-SQLite.png
  - Detailed PNG for workflow represented in Workflow Model – Inner Workflow – SQLite section.
- Outer-workflow.yw
  - Outer YesWorkflow workflow file.
- MenuItem-OR.yw
- Menu-OR.yw

- MenuPage-OR.yw
- Dish-OR.yw
- Inner-workflow-sqlite.yw
    - Inner YesWorkflow workflow definition files.