CS 513 – DATA CLEANING FINAL PROJECT

PROJECT TITLE	CS 513 SUMMER 2020 – DATA CLEANING PROJECT
TEAM	TEAM SPIRIT
COLLABORATORS	BALAJI SATHYAMURTHY (**balajis2**) GOWRI SHANKAR RAMANAN (**gsr2**)

PROJECT OBJECTIVE

The objective of this data cleaning project is to refine the publicly available dataset "US Farmer market" dataset available in the website to derive valuable business insights.

https://www.ams.usda.gov/local-food-directories/farmersmarkets

We used OpenRefine v3.3 to perform the initial cleaning of the raw dataset and then SQLite3 to perform integrity constraint checks and violation clean-ups on the dataset. The final cleaned dataset along with the original raw dataset was uploaded in the Box folder at https://uofi.box.com/s/gkictao5xrnrdp3va3qwt6fkoxuv63gz for review. They were further validated for special use-cases. The YesWorkflow model is used to annotate the various steps followed in the data cleaning process. We used the OpenRefine to YesWorkflow model toolkit (OR2YW v0.0.16) to render visual representations of workflows and provenance using Graph Viz.

PART 1: OVERVIEW AND INITIAL ASSESSMENT ON THE DATASET

1.1 DATASET OVERVIEW

- The US Farmers market dataset contains a total of 59 data elements.
- FMID is a numeric field with a unique identifier to identify each Farmers market. There seems to be no duplicate values in FMID.
- Market name is a text field containing the name of the farmer's market.
- Website, Facebook, Twitter, YouTube, and Other Media
  are all text fields containing either website or social media contacts such as twitter ID, channel name etc.,
- Street, City, County, State and Zip contains address information about the farmer's market.
- Season1Date, Season1Time, Season2Date, Season2Time, Season3Date, Season3Time, Season4Date, Season4Time contains various season date and time the respective Farmer's market are open during a year.

- Columns X and Y looks like to represent the geographic latitude and longitude of the of the farmer's market
- Column Location provides additional location details such as nearest landmark of the Farmer's market. This seems to be a text field.
- Credit is a single character field with either Y or N denoting whether the payment via credit card is accepted or not. On initial assessment it seems all the records have this value populated in the dataset.
- Columns WIC, WICcash, SFMNP, SNAP are all single character fields with either Y or N values denoting whether payment through food assistance programs is either accepted or not
- A list of 30 different indicator fields with respect to the characteristics of various food products sold in the farmers market. On initial assessment all these columns look good with no discrepancies.
- Updatetime denotes date and time when the data about farmer's market is captured

1.2 LIST OF DATA QUALITY ISSUES

- Market name seems to contain lot of text variations for same Farmer market name.
- Most of the values for the social media columns are blank
- County and city names contain variations in text for the same value.
- zip contains some invalid zip codes.
- The date format for season dates seems to be not consistent across the records. Some records contain only month name, and some contains only from date etc.,
- Most of the farmer's market have season1 and season 2 values and only very less records contain season three and four values.
- Columns X and Y names are not clear, and we assume it represents the geographic location of the latitude and longitude of the farmers market.

1.3 IS THE DATA CLEAN ENOUGH FOR THE USE CASES?

Though dataset have the above noted quality issues, we could use the dataset for the below list of use cases:

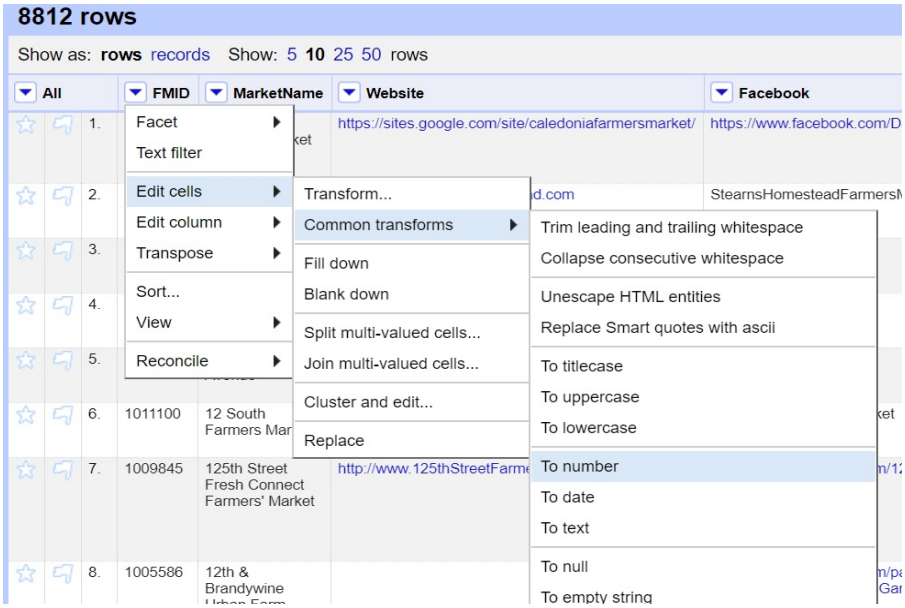
- To identify the list of Farmer's markets nearest based on the geographic location.
- To identify the list of Farmer's markets which are open based on seasonality.
- To identify list of Farmer's markets which accept credit card
- To identify list of Farmer's markets which accept food coupons such as WIC, WICcash etc.,
- To identify list of Famer's markets based on the various sold products such as seafood, pet food, vegetables etc.,

PART 2: DATA CLEANING STEPS USING OpenRefine

The following steps have been performed to clean the data via OpenRefine.

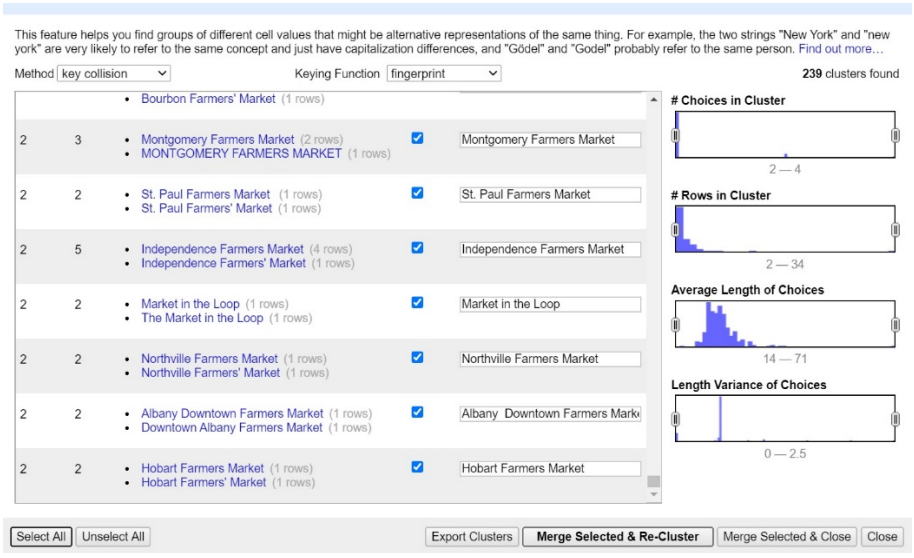
1. FMID

The FMID field is converted to numeric field as shown below:



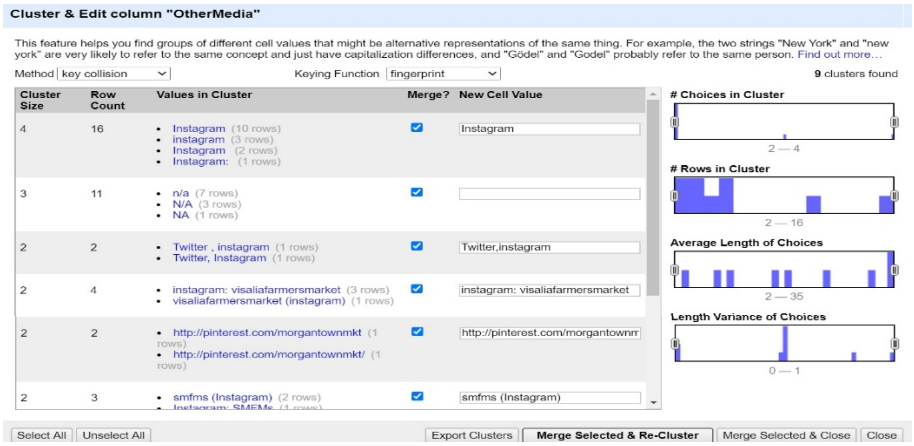
2. MARKET NAME

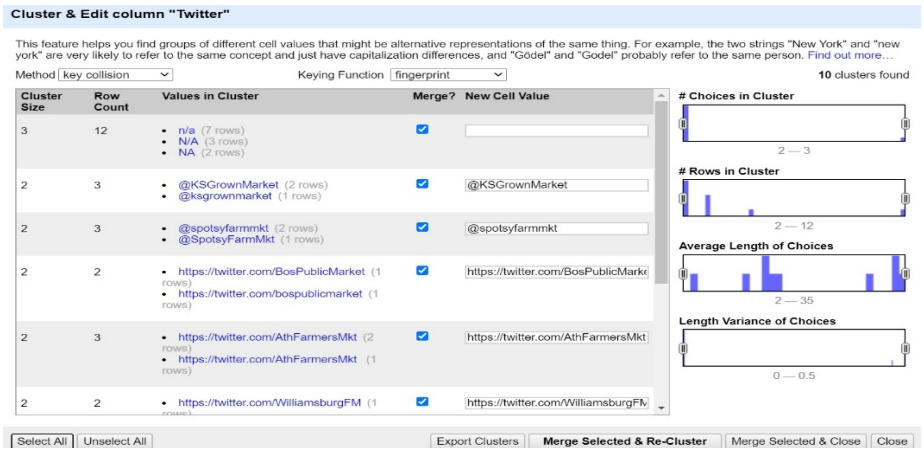
To canonicalize the same market name with different text variations, the text Facets and cluster feature available in OpenRefine is used as shown below:

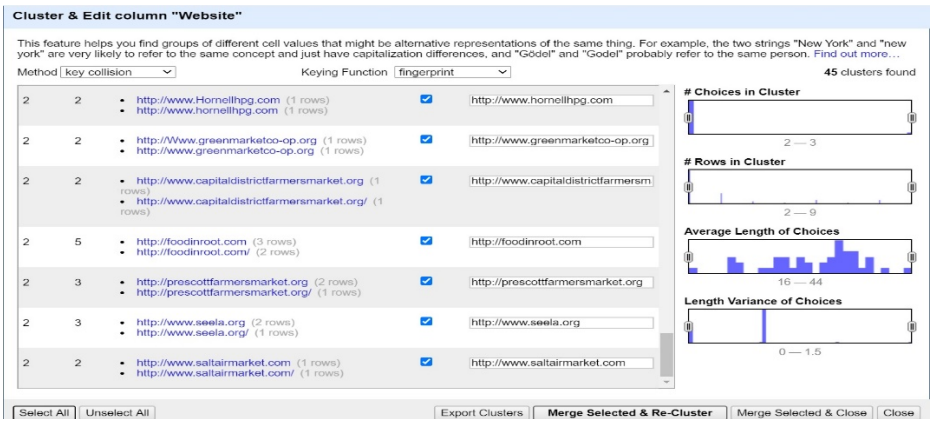


3. WEBSITE, TWITTER, FACEBOOK, OTHER MEDIA and YOUTUBE

To canonicalize the social media columns text variations, the text Facets and cluster feature available in OpenRefine is used as shown below:

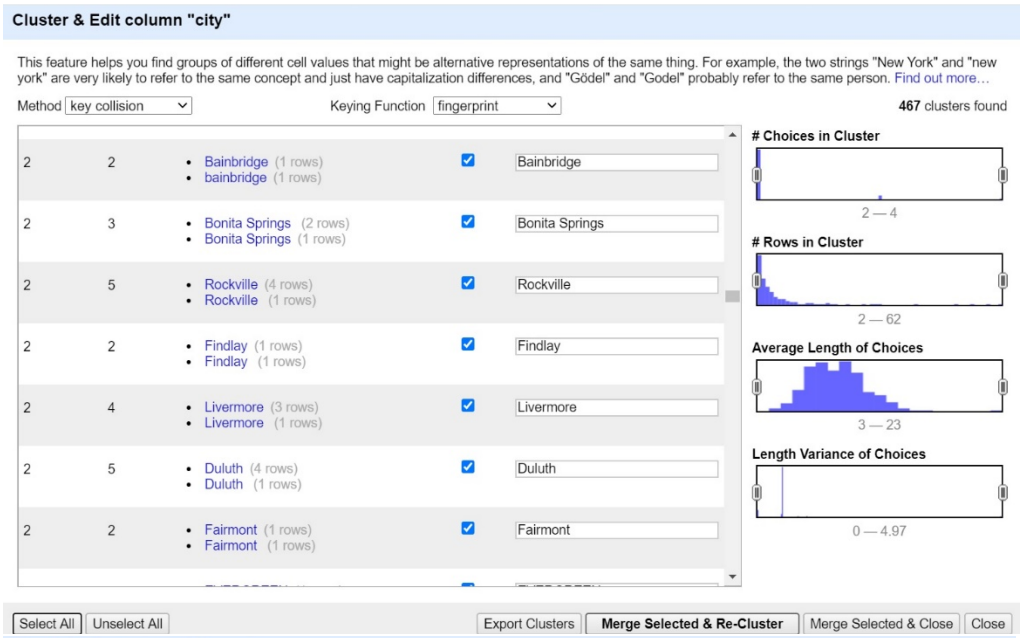


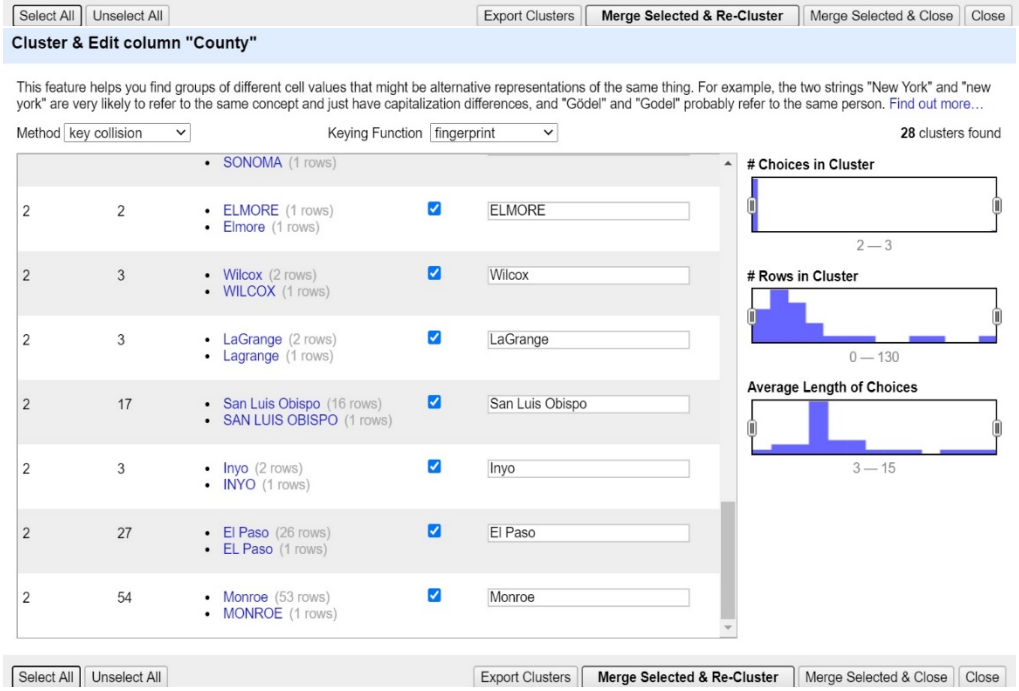




4. CITY AND COUNTY NAME

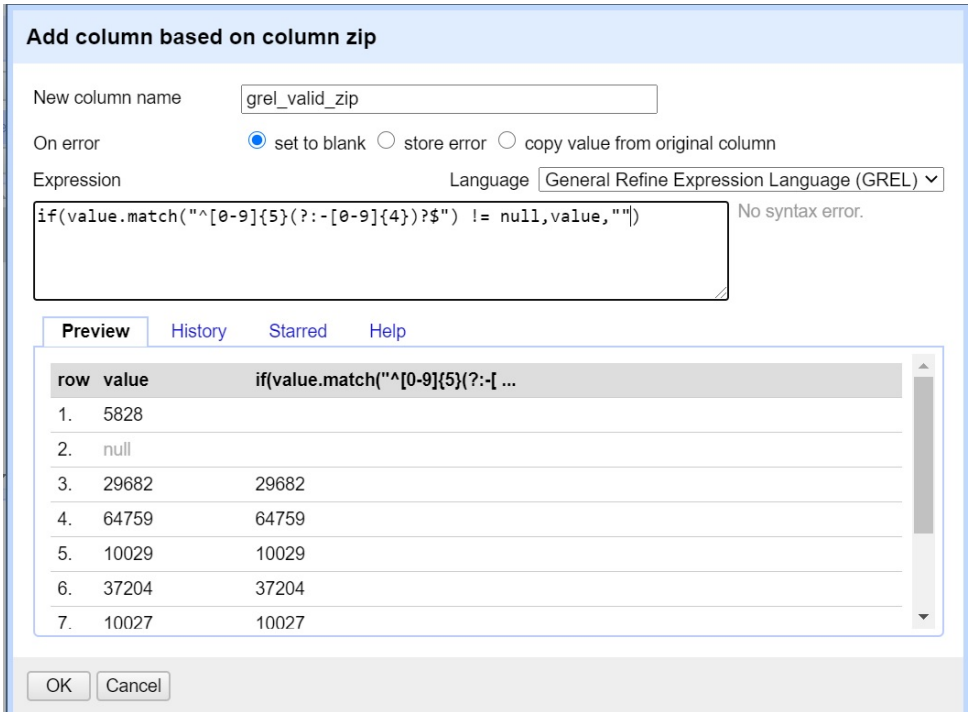
To canonicalize the city and county name text variations, the text Facets and cluster feature available in OpenRefine is used as shown below:





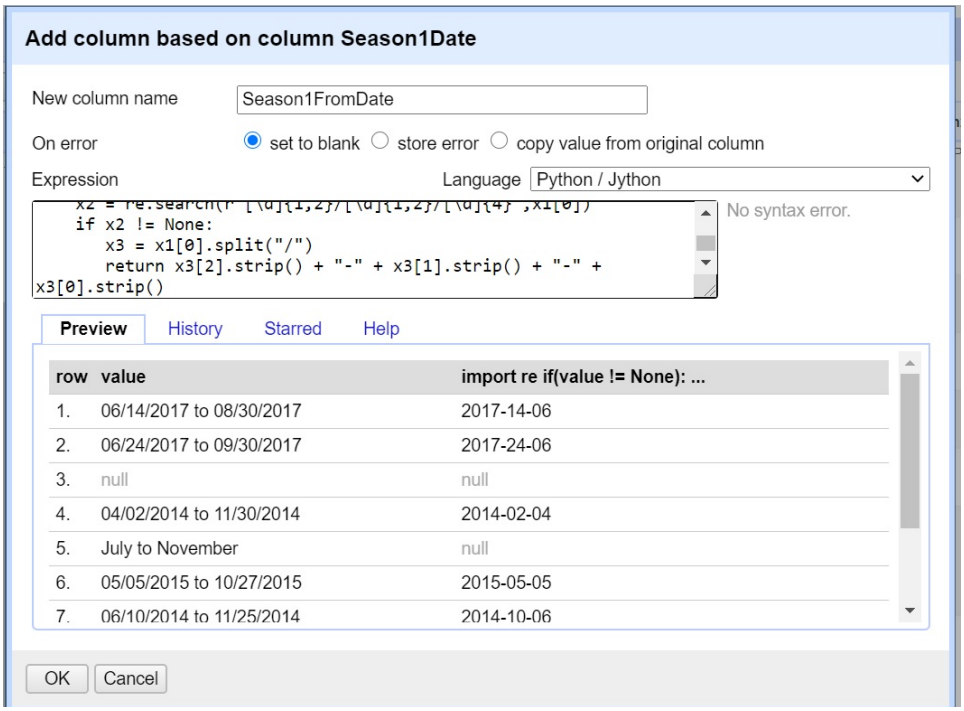
5. GREL – VALID ZIP CODE

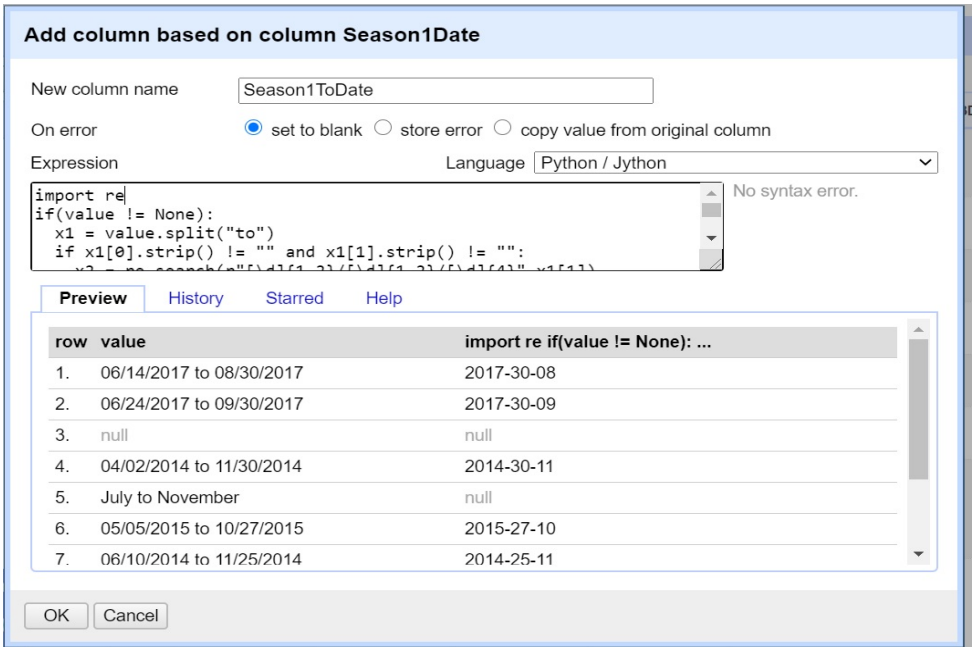
The regular expression is used to identify the valid zip code formats (XXXXX and XXXXX-XXXX) and a new column “grel_valid_zip” is added to the dataset as shown below:



6. SEASON DATE – SPLIT AS FROM AND TO DATES

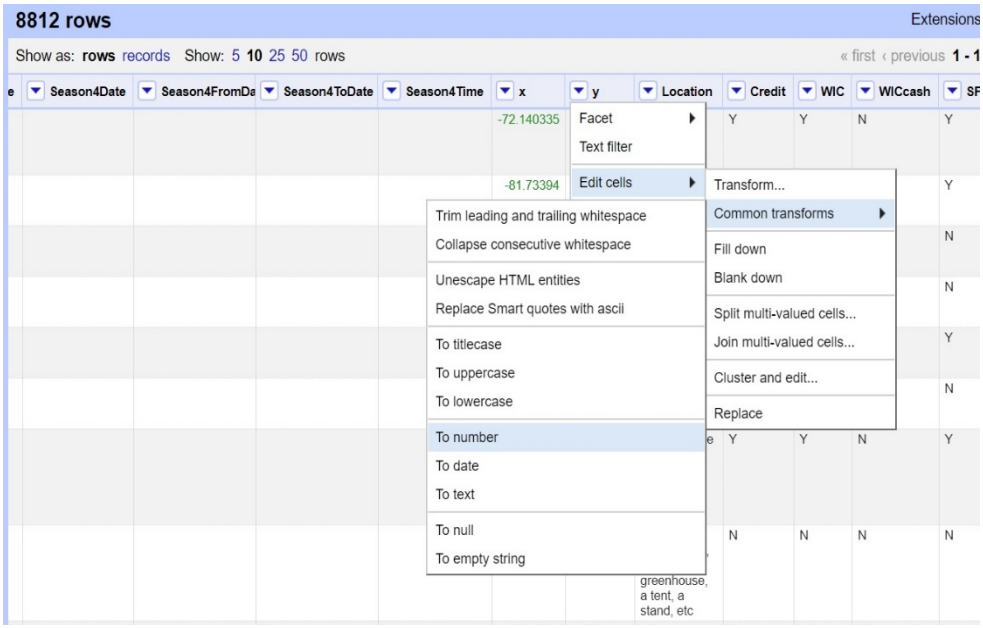
Python is used to identify the season dates in MM/DD/YYYY format and split them in two dates date columns as "From date" and "To Date" so that seasonality of the market can be identified easily. All the season date columns – season1, season2, season 3 and season 4 have been split into two date columns as shown below:





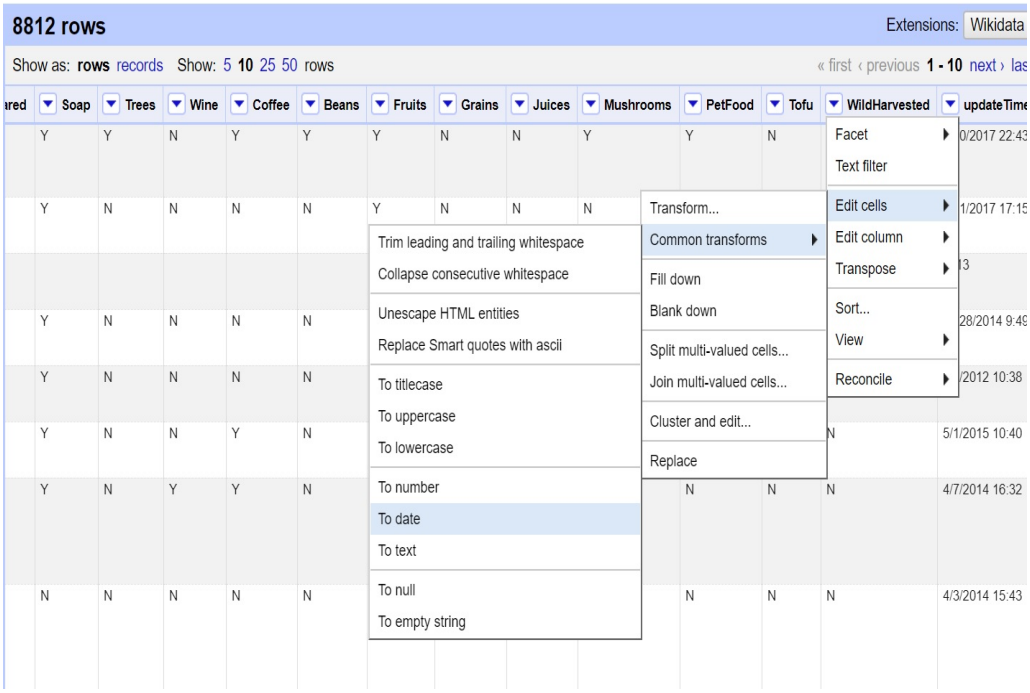
7. COLUMNS X AND Y

The X and Y columns are converted to numeric field as shown below:



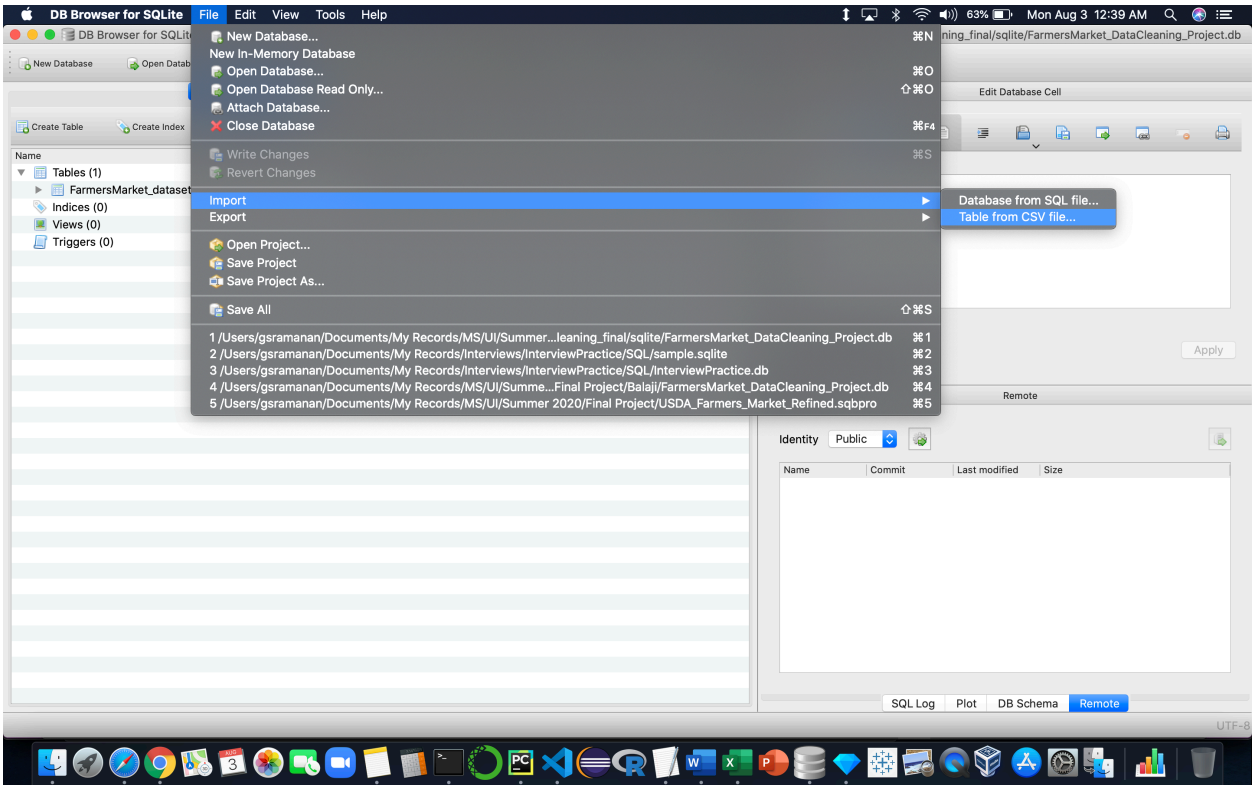
8. UPDATE TIME

Update time column is converted to standard date column as shown below:



PART 3: DEVELOPING A RELATIONAL SCHEMA

The refined data from OpenRefine is exported as csv file and imported into the below table in SQLite to perform further SQL data profiling and Integrity Check queries.

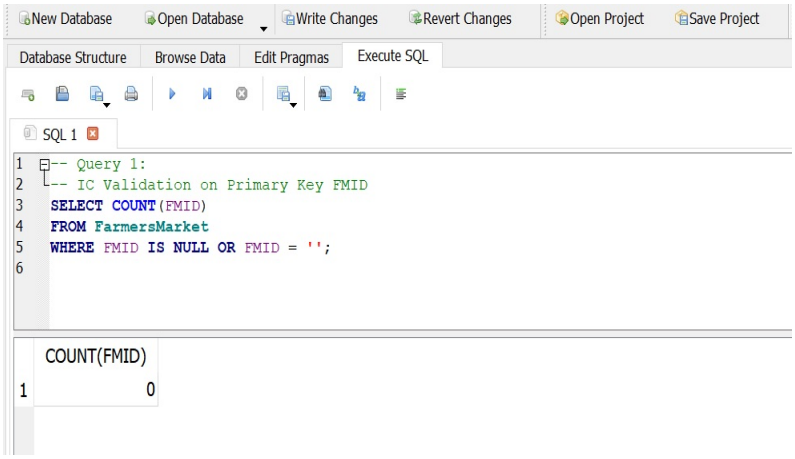


COLUMN NAME	DATA TYPE	Primary Key
FMID	INTEGER	Yes
MarketName	TEXT	No
Website	TEXT	No
Facebook	TEXT	No
Twitter	TEXT	No
Youtube	TEXT	No
OtherMedia	TEXT	No
street	TEXT	No
city	TEXT	No
County	TEXT	No
State	TEXT	No
zip	TEXT	No
grel_valid_zip	TEXT	No
Season1Date	TEXT	No
Season1FromDate	TEXT	No
Season1ToDate	TEXT	No
Season1Time	TEXT	No
Season2Date	TEXT	No
Season2FromDate	TEXT	No
Season2ToDate	TEXT	No
Season2Time	TEXT	No
Season3Date	TEXT	No
Season3FromDate	TEXT	No
Season3ToDate	TEXT	No
Season3Time	TEXT	No
Season4Date	TEXT	No
Season4FromDate	TEXT	No
Season4ToDate	TEXT	No
Season4Time	TEXT	No
x	REAL	No
y	REAL	No
Location	TEXT	No
Credit	TEXT	No
WIC	TEXT	No
WICcash	TEXT	No
SFMNP	TEXT	No
SNAP	TEXT	No
Organic	TEXT	No
Bakedgoods	TEXT	No
Cheese	TEXT	No
Crafts	TEXT	No

Flowers	TEXT	No
Eggs	TEXT	No
Seafood	TEXT	No
Herbs	TEXT	No
Vegetables	TEXT	No
Honey	TEXT	No
Jams	TEXT	No
Maple	TEXT	No
Meat	TEXT	No
Nursery	TEXT	No
Nuts	TEXT	No
Plants	TEXT	No
Poultry	TEXT	No
Prepared	TEXT	No
Soap	TEXT	No
Trees	TEXT	No
Wine	TEXT	No
Coffee	TEXT	No
Beans	TEXT	No
Fruits	TEXT	No
Grains	TEXT	No
Juices	TEXT	No
Mushrooms	TEXT	No
PetFood	TEXT	No
Tofu	TEXT	No
WildHarvested	TEXT	No
updateTime	TEXT	No

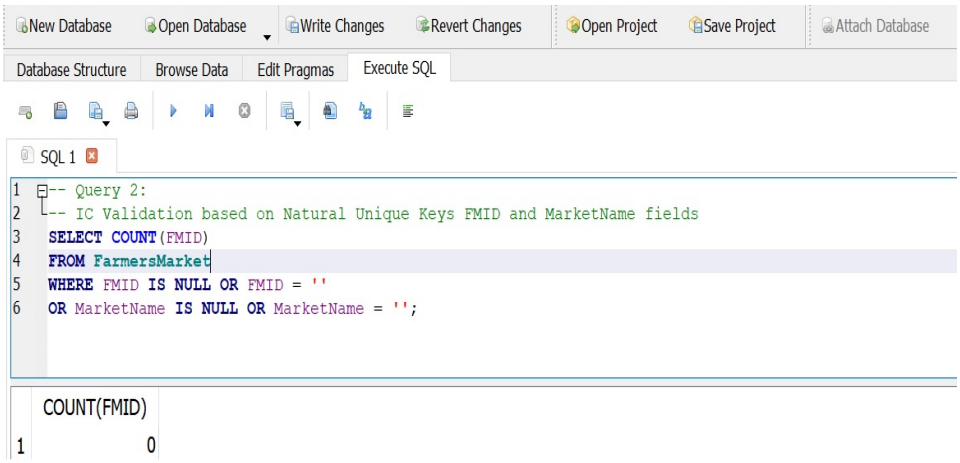
1. DUPLICATE FMID

On query analysis, there is no duplicate value for FMID field in the dataset.



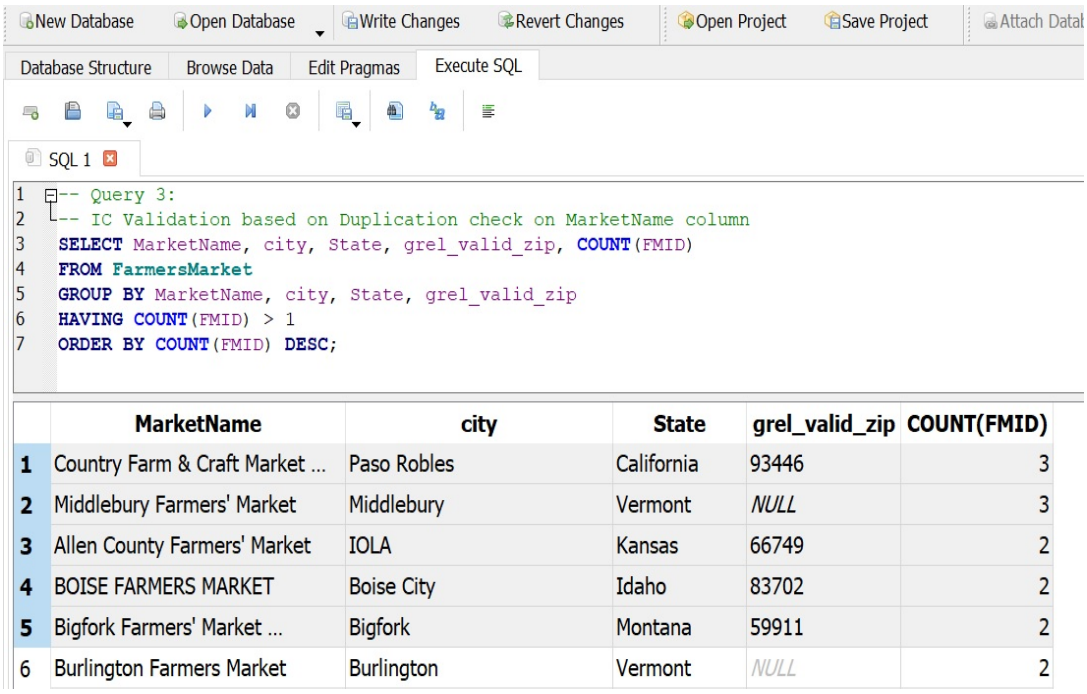
2. MARKETS WITH FMID OR MARKETNAME IS BLANK OR NULL

On analysis there seems to be no FMID or Market name having blank or null values in the dataset



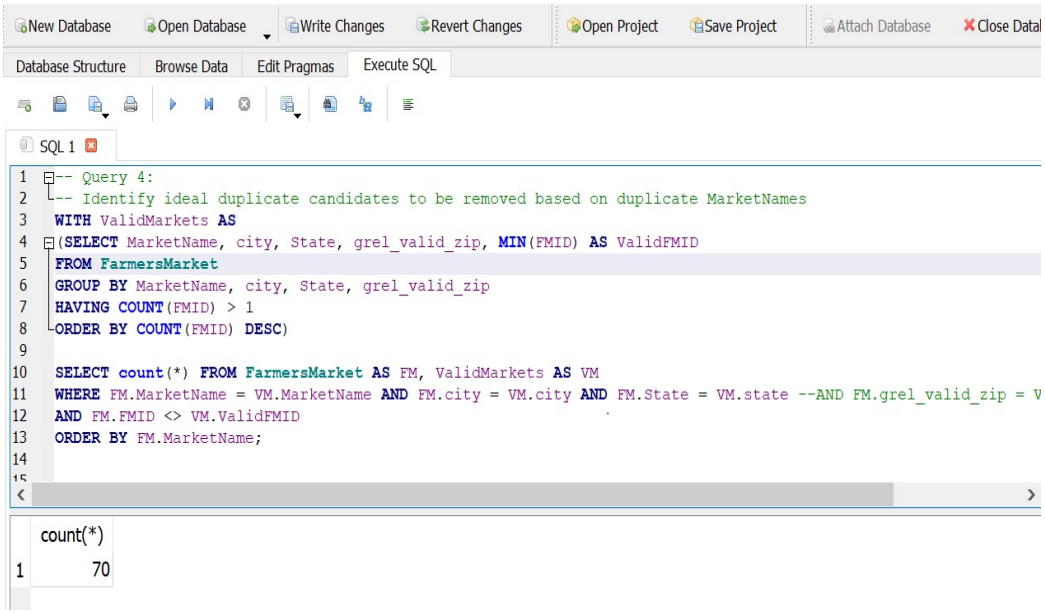
3. CHECKING FOR SAME MARKET NAME WITHIN SAME CITY / STATE / ZIP BUT DIFFERENT FMID.

The below sql shows there are same market name within the same city and state but with different FMID.

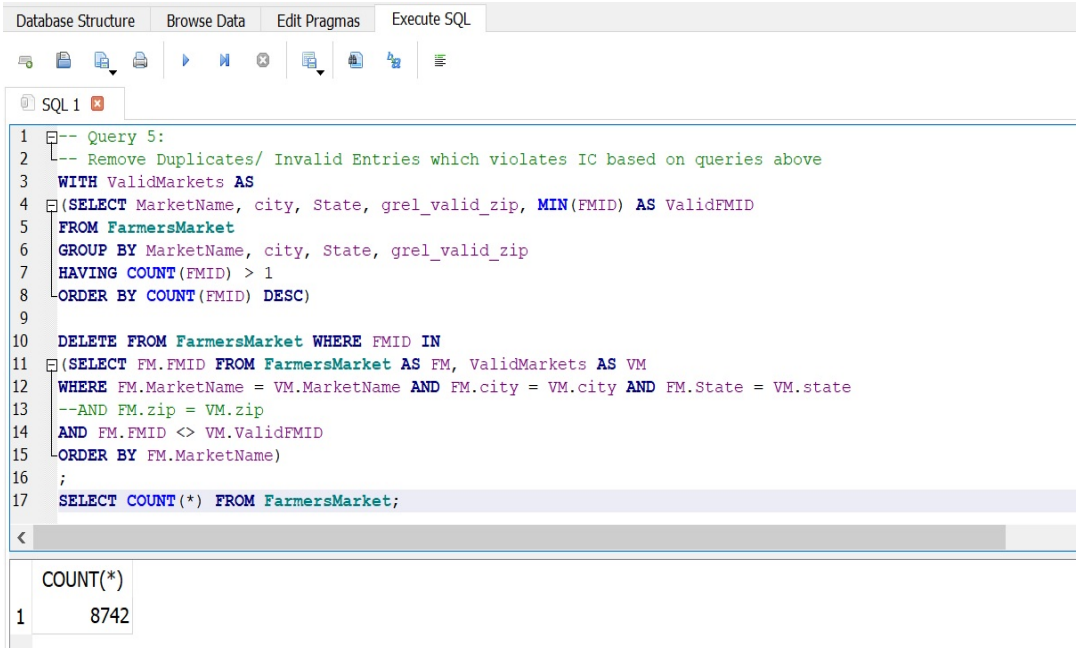


	MarketName	city	State	grel_valid_zip	COUNT(FMID)
1	Country Farm & Craft Market ...	Paso Robles	California	93446	3
2	Middlebury Farmers' Market	Middlebury	Vermont	*NULL*	3
3	Allen County Farmers' Market	IOLA	Kansas	66749	2
4	BOISE FARMERS MARKET	Boise City	Idaho	83702	2
5	Bigfork Farmers' Market ...	Bigfork	Montana	59911	2
6	Burlington Farmers Market	Burlington	Vermont	*NULL*	2

On further analysis, we identify around 70 markets to be removed due to the same FMID:

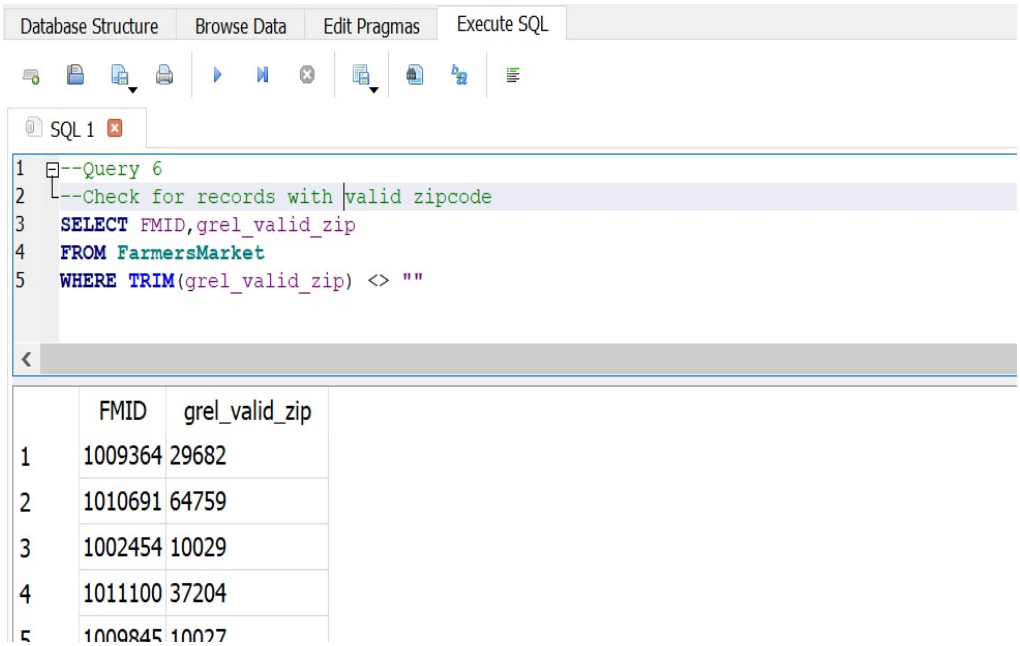


Deleting the identified duplicate market entries:



4. MARKETS WITH VALID ZIP CODE FOR GEOGRAPHIC POSITIONING

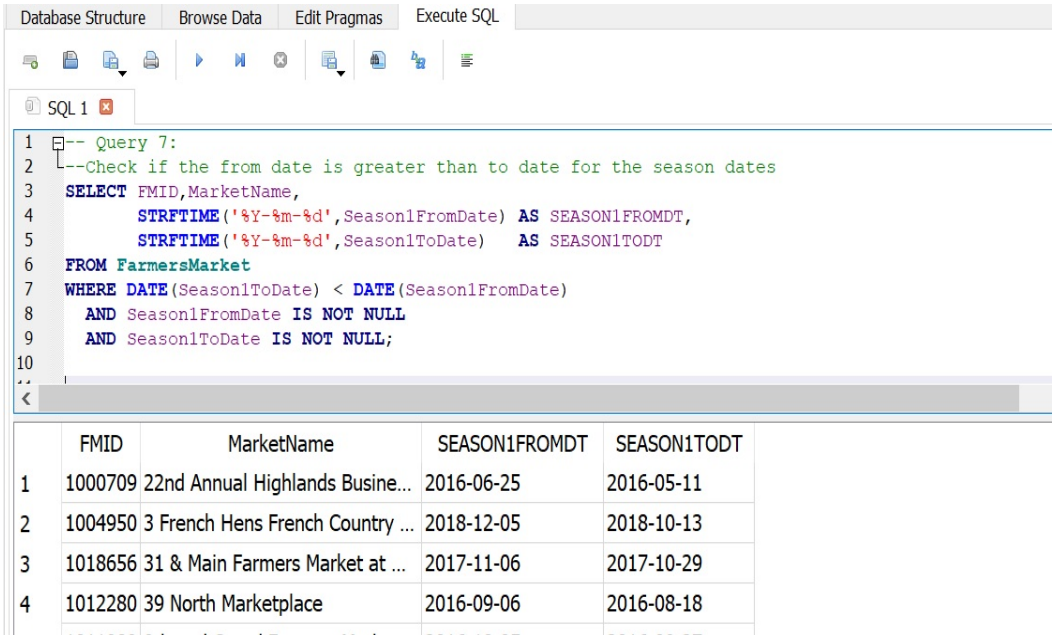
For the use case to identify the nearby markets around a geographic position, the grel_valid_zip field is selecting accurate date on zip codes.



	FMID	grel_valid_zip
1	1009364	29682
2	1010691	64759
3	1002454	10029
4	1011100	37204
5	1009845	10027

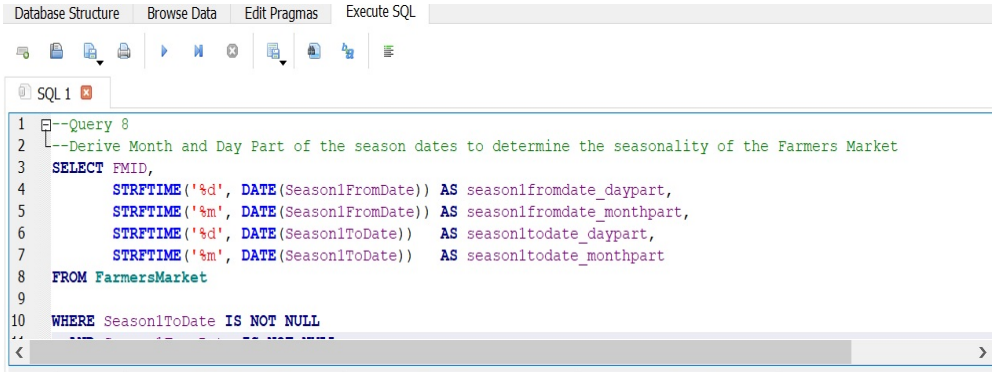
5. MARKETS WITH VALID SEASON DATES

For the use case to identify markets open for specific seasonality, the season from and to dates column fetch the appropriate values.



	FMID	MarketName	SEASON1FROMDT	SEASON1TODT
1	1000709	22nd Annual Highlands Busine...	2016-06-25	2016-05-11
2	1004950	3 French Hens French Country ...	2018-12-05	2018-10-13
3	1018656	31 & Main Farmers Market at ...	2017-11-06	2017-10-29
4	1012280	39 North Marketplace	2016-09-06	2016-08-18

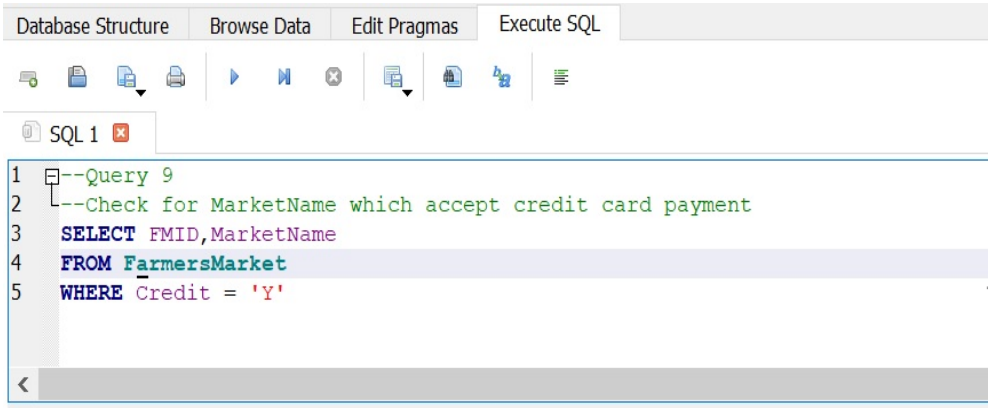
Extracting the month and date part of the season dates:



	FMID	season1fromdate_daypart	season1fromdate_monthpart	season1todate_daypart	season1todate_monthpart
1	1018261	14	06	30	08
2	1018318	24	06	30	09
3	1010691	04	02	30	11

6. MARKETS WITH CREDIT CARD PAYMENT

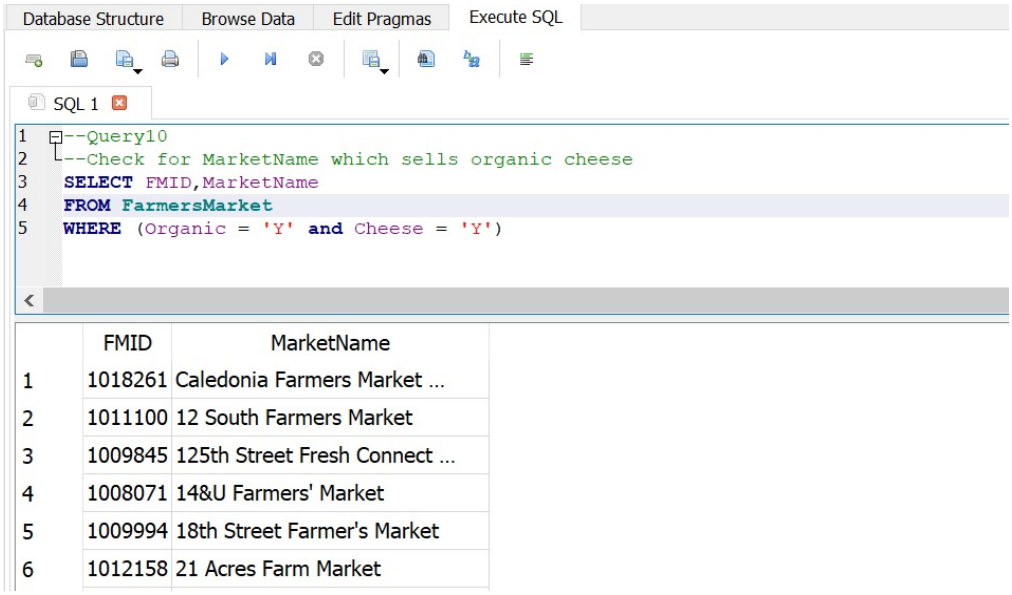
The data column for credit card payment is fetching the appropriate market names.



	FMID	MarketName
1	1018261	Caledonia Farmers Market ...
2	1018318	Stearns Homestead Farmers' ...
3	1009364	106 S. Main Street Farmers ...
4	1010691	10th Steet Community Farmer...

7. MARKETS BASED ON VARIOUS SOLD PRODUCT CATEGORIES

The data column for various sold product categories is fetching the appropriate market names.



	FMID	MarketName
1	1018261	Caledonia Farmers Market ...
2	1011100	12 South Farmers Market
3	1009845	125th Street Fresh Connect ...
4	1008071	14&U Farmers' Market
5	1009994	18th Street Farmer's Market
6	1012158	21 Acres Farm Market

PART 4: CREATING A WORKFLOW MODEL

YesWorkflow model/ prototype has been used to annotate the data cleaning procedure followed, and to generate visual model representations using graphviz. The workflows have been created for both overall data cleaning process as well as the open refine tool specific operation history to represent the data provenance better.

4.1 Overall Workflow:

The binaries available at yw-prototypes repository (https://github.com/yesworkflow-org/yw-prototypes) have been employed to generate these workflows. The scripts (Overall_Workflow.txt and OpenRefine_Workflow.txt) containing special YesWorkflow (YW) comments will be interpreted by the prototype tool with simple commands, to render these graphical representations in DOT format (.gv).

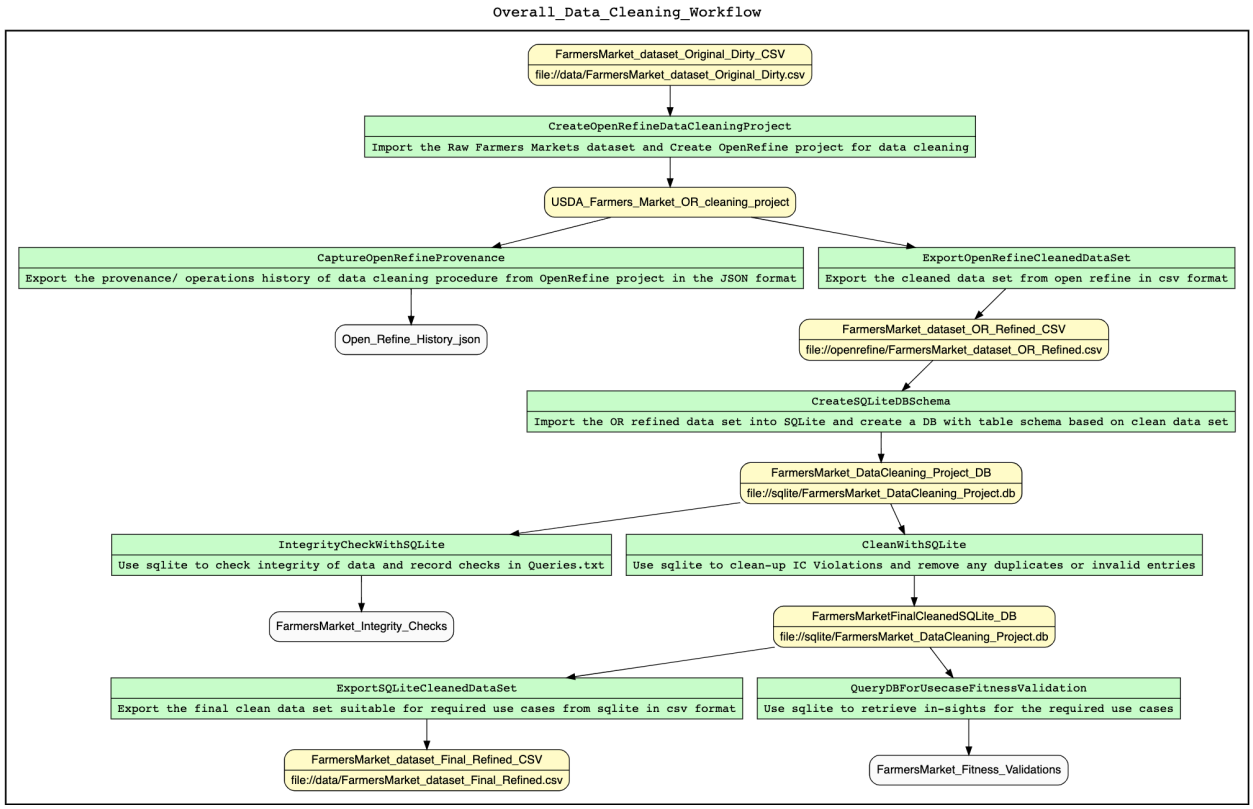
Overall_Workflow.txt (Annotations as python script comments explaining the overall workflow)

```
'''
@begin Overall_Data_Cleaning_Workflow @desc Overall Workflow for USDA Farmers Markets dataset cleaning project
@in FarmersMarket_dataset_Original_Dirty_CSV @uri file://data/FarmersMarket_dataset_Original_Dirty.csv

   @begin CreateOpenRefineDataCleaningProject @desc Import the Raw Farmers Markets dataset and Create OpenRefine project for data cleaning
   @in FarmersMarket_dataset_Original_Dirty_CSV @uri file://data/FarmersMarket_dataset_Original_Dirty.csv
   @out USDA_Farmers_Market_OR_cleaning_project
   @end CreateOpenRefineDataCleaningProject

   @begin CaptureOpenRefineProvenance @desc Export the provenance/ operations history of data cleaning procedure from OpenRefine project in the
JSON format
   @in USDA_Farmers_Market_OR_cleaning_project
   @out Open_Refine_History_json @uri file://openrefine/Open_Refine_History.json
   @end CaptureOpenRefineProvenance

   @begin ExportOpenRefineCleanedDataSet @desc Export the cleaned data set from open refine in csv format
   @in USDA_Farmers_Market_OR_cleaning_project
   @out FarmersMarket_dataset_OR_Refined_CSV @uri file://openrefine/FarmersMarket_dataset_OR_Refined.csv
   @end ExportOpenRefineCleanedDataSet

   @begin CreateSQLiteDBSchema @desc Import the OR refined data set into SQLite and create a DB with table schema based on clean data set
   @in FarmersMarket_dataset_OR_Refined_CSV @uri file://openrefine/FarmersMarket_dataset_OR_Refined.csv
   @out FarmersMarket_DataCleaning_Project_DB @uri file://sqlite/FarmersMarket_DataCleaning_Project.db
   @end CreateSQLiteDBSchema
```

```
@begin IntegrityCheckWithSQLite @desc Use sqlite to check integrity of data and record checks in Queries.txt
@in FarmersMarket_DataCleaning_Project_DB @uri file://sqlite/FarmersMarket_DataCleaning_Project.db
@out FarmersMarket_Integrity_Checks @uri file://sqlite/Queries.txt
@end IntegrityCheckWithSQLite

@begin CleanWithSQLite @desc Use sqlite to clean-up IC Violations and remove any duplicates or invalid entries
@in FarmersMarket_DataCleaning_Project_DB @uri file://sqlite/FarmersMarket_DataCleaning_Project.db
@out FarmersMarketFinalCleanedSQLite_DB @uri file://sqlite/FarmersMarket_DataCleaning_Project.db
@end CleanWithSQLite

@begin ExportSQLiteCleanedDataSet @desc Export the final clean data set suitable for required use cases from sqlite in csv format
@in FarmersMarketFinalCleanedSQLite_DB @uri file://sqlite/FarmersMarket_DataCleaning_Project.db
@out FarmersMarket_dataset_Final_Refined_CSV @uri file://data/FarmersMarket_dataset_Final_Refined.csv
@end ExportSQLiteCleanedDataSet

@begin QueryDBForUsecaseFitnessValidation @desc Use sqlite to retrieve in-sights for the required use cases
@in FarmersMarketFinalCleanedSQLite_DB @uri file://sqlite/FarmersMarket_DataCleaning_Project.db
@out FarmersMarket_Fitness_Validations @uri file://sqlite/Queries.txt
@end QueryDBForUsecaseFitnessValidation

@out FarmersMarket_dataset_Final_Refined_CSV @uri file://data/FarmersMarket_dataset_Final_Refined.csv
@end Overall_Data_Cleaning_Workflow
'''
```

The annotations were converted into graphviz representations in DOT format (.gv file) using the following simple commands:

```
$ alias yw='java -jar ~/bin/yesworkflow-0.2.0-jar-with-dependencies.jar'
$ yw graph Overall_Workflow.txt > Overall_Workflow.gv
$ dot -Tpdf Overall_Workflow.gv -o Overall_Workflow.pdf
```



The above overall workflow explains clearly that the data set was cleaned initially with Open Refine to remove data anomalies and then checked for integrity and cleaned up the violations with deeper analysis using SQLite database before producing the final refined dataset in CSV format.

4.2 Open Refine Workflow:

The operations history/ provenance collected from Open Refine tool after the completion of necessary cleaning, is exported in JSON format. This JSON file is further converted into YesWorkflow models and visual representations using OR2YW tool available at https://pypi.org/project/or2ywtool/. This tool aims to provide an auto-parsing method from the Open Refine Operation Recipe JSON file to YesWorkflow model which classifies the operations into two levels: schema level and column level.

The Workflow annotations embedded in python script comments are generated using the following simple command:

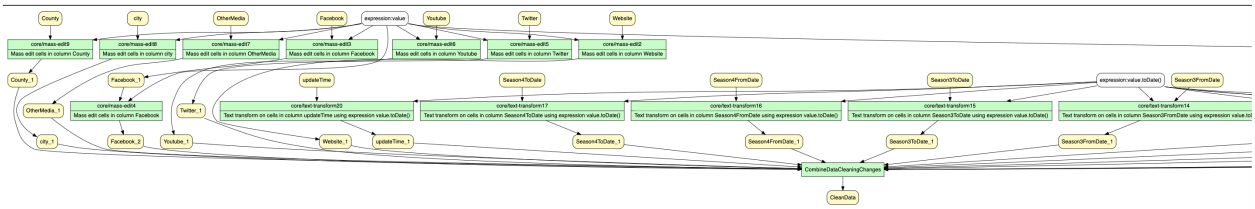
```
$ or2yw -i openrefine/Open_Refine_History.json -o workflows/Open_Refine_Workflow.txt -t parallel
File Open_Refine_Workflow.txt generated.
```

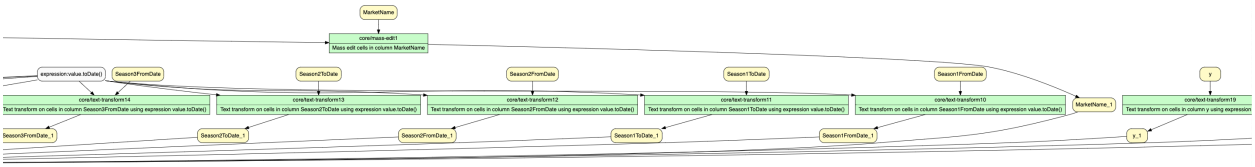
The graphical output is produced in DOT format (.gv file) using the following simple command:

```
$ or2yw -i openrefine/Open_Refine_History.json -o workflows/Open_Refine_Workflow.gv -ot gv -t parallel
java found:  java
File Open_Refine_Workflow.gv generated.
```

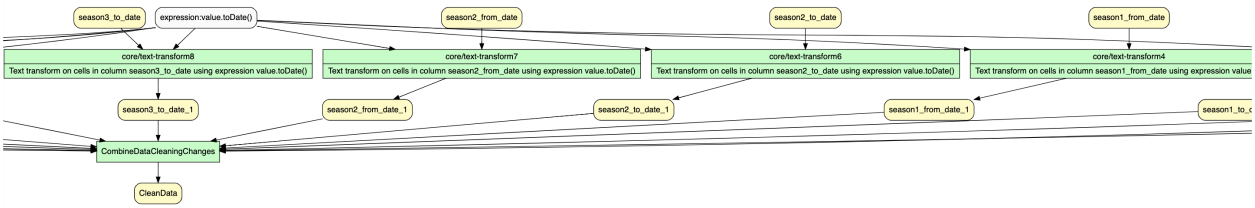
The output gv file has then been tested with GraphViz Online for proper rendering of workflow image. The workflow image has been broken down into multiple chunks as the original output is too large to fit into this report with actual size.

You can see the mass edit normalization of data performed with Open Refine Text Facet and clustering feature below for MarketName, City, County and other text columns.

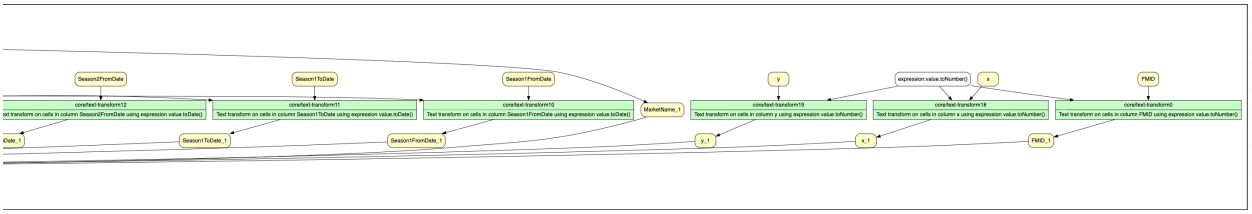




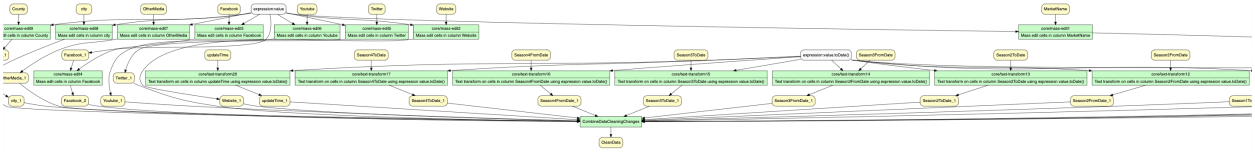
The date fields were also normalized to ISO date format with the transformation feature of Open Refine.



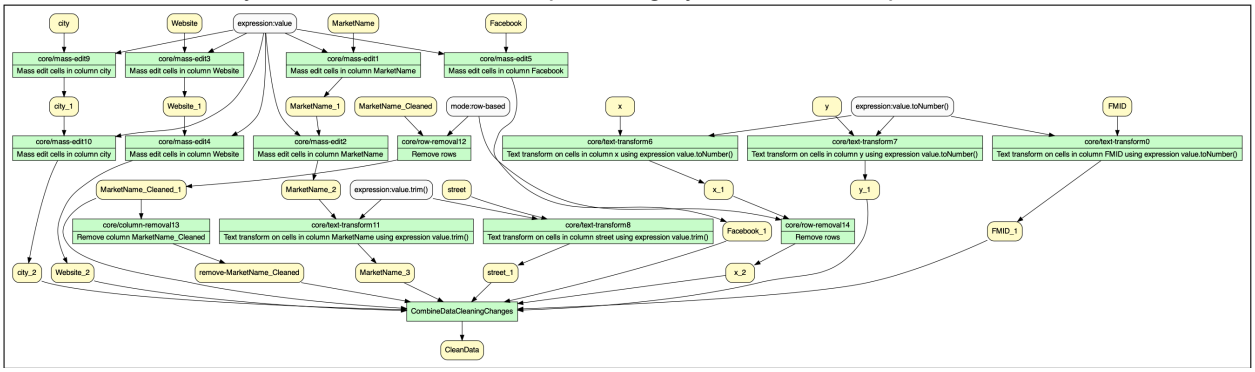
All data columns were trimmed to remove leading and lagging spaces, and numeric data columns were appropriately transformed into numeric fields using transformation features of Open Refine. Invalid data was identified using GREL expressions and cleaned up using filter features in Open Refine.



All normalized and cleaned up data columns were finally combined into a clean data set.



The final data set was further explored for opportunities to identify duplicate rows based on Market Name and other associated columns which could potentially be the same Farmers Market entity record. The refined data set was finally exported from Open Refine in the CSV format. This refined data set was fed into SQLite to identify and eliminate more complex integrity violations as explained in above sections.



CONCLUSION:

The USDA Farmers Markets Dataset is overall a well-structured dataset except that data is incomplete and inconsistent on several columns which required many parts of data to be cleaned and reformatted to improve consistency and integrity. There are few columns in the dataset which becomes completely unusable such as “updateTime”, and “Season1StartDate”, which is very inconsistent and contains either “Month” or “DD/MM/YYYY” in variable text format, and there were overlapping seasonal dates observed, which makes it unusable specifically for use-cases related to seasonal date specific search queries. Lot of anomalies have been observed with respect to invalid date ranges with Season_To_Date being less than the From_Date and those have been cleaned up. With these cleaned up season1 start and end dates, we can search for farmers market for a specific seasonality more accurately.

Duplicate entries of Market information with different FMIDs but with similar Market Name profile information, address etc. have also been identified as Integrity violations through queries done through SQLite database. These potential duplicates have been cleaned up for more accurate representations.

Overall, the dataset was cleaned appropriately for the hypothetical use cases. Zip code data has also been validated for accuracy and the discrepancies were eliminated, thus making it usable for comparison between the number of farmers markets in each city or states and to use with geographic positioning related use cases.

Contributions:

Tasks	Contributors
Raw Data Set Profiling & Analysis	Gowri Shankar Ramanan & Balaji Sathyamurthy
Clean-up using Open Refine & capturing operations history	Balaji Sathyamurthy
Integrity Checks with SQLite	Balaji Sathyamurthy
Clean-up IC violations using SQLite	Gowri Shankar Ramanan & Balaji Sathyamurthy
Create Overall Workflow using YesWorkflow	Gowri Shankar Ramanan
Generate Open Refine Workflow with history using OR2YW tool	Gowri Shankar Ramanan
Create Final Project Report	Gowri Shankar Ramanan & Balaji Sathyamurthy
Submit the artifacts of final project	Gowri Shankar Ramanan