CS513-Project Phase 2

By: Bhushan Bathani(bbath2@illinois.edu) and Zoheb Satta(satta2@illinois.edu)

Introduction

Farmers Markets dataset contains the data of all the markets , their timings, location, inventory, payment methods etc. This data is useful to locate any nearby market which provides customers what they want. e.g. Search for a market which is in 10 miles of radius and provides vegetables and seafood and is currently open.

We used OpenRefine, YesWorkflow and SQLLite tools to clean up these data and extract the meaningful data.

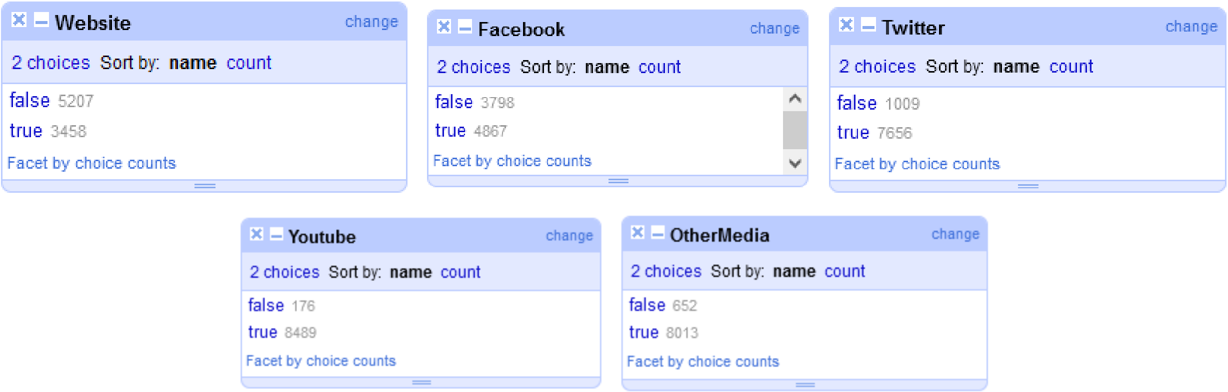
Use Case:

As for the Use cases, we came up with 3 use cases representing:

U0: The use case that we could have implemented without any data cleaning on the data set was an app that could give you the names of the restaurants in your city

U1: The use case we decided to clean our dataset for was: an app that can match consumers with specialty stores by product in their city of choice

U2: The use case that could never be implemented is an app that sends you a link to all of a restaurant's social media when you are within 1 mile of the location. This is because all of the columns involving social media are significantly lacking in data as can be seen in these facets highlighting the null/blank values



Website change
2 choices Sort by: name count
false 5207
true 3458
Facet by choice counts

Facebook change
2 choices Sort by: name count
false 3798
true 4867
Facet by choice counts

Twitter change
2 choices Sort by: name count
false 1009
true 7656
Facet by choice counts

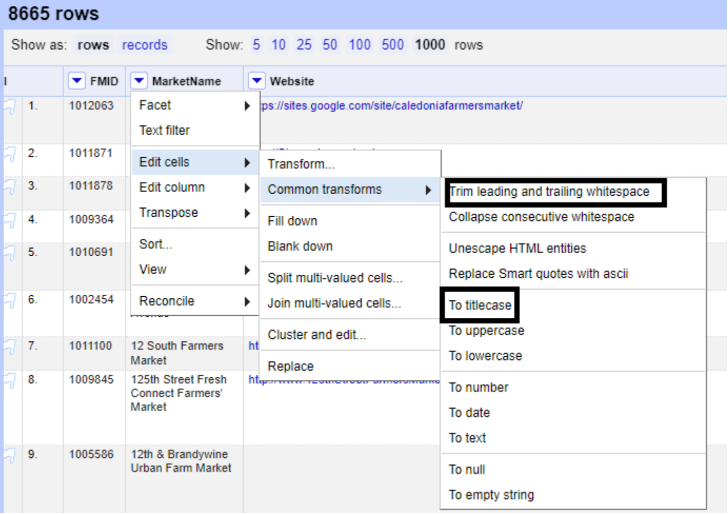
Youtube change
2 choices Sort by: name count
false 176
true 8489
Facet by choice counts

OtherMedia change
2 choices Sort by: name count
false 652
true 8013
Facet by choice counts

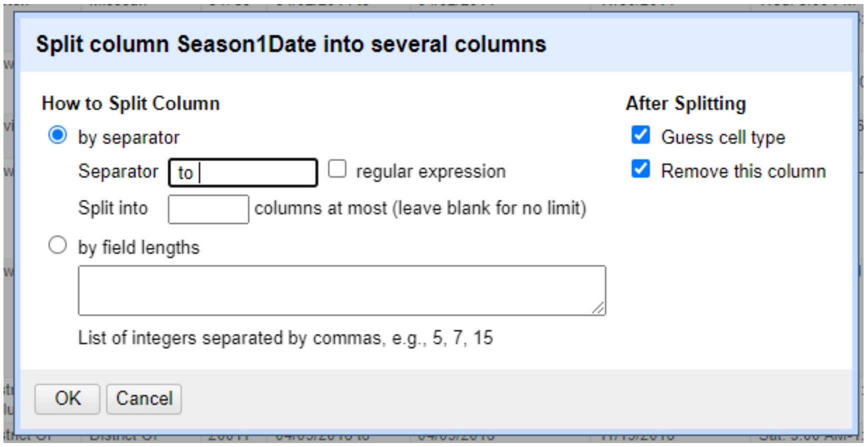
Data Cleaning Workflow:

OpenRefine

All the columns in the data were trimmed and the whitespace and MarketName, Website, Facebook, Twitter, Youtube, OtherMedia, Street, City, County and State columns were converted to TileCase. This was done so that these operations would not need to be done on the client side by the application.


8665 rows
Show as: rows records
Show: 5 10 25 50 100 500 1000 rows
FMID
MarketName
Website
1. 1012063
2. 1011871
3. 1011878
4. 1009364
5. 1010691
6. 1002454
7. 1011100 12 South Farmers Market
8. 1009845 125th Street Fresh Connect Farmers' Market
9. 1005586 12th & Brandywine Urban Farm Market
ps://sites.google.com/site/caledoniafarmersmarket/
Facet
Text filter
Edit cells
Edit column
Transpose
Sort...
View
Reconcile
Transform...
Common transforms
Fill down
Blank down
Split multi-valued cells...
Join multi-valued cells...
Cluster and edit...
Replace
Trim leading and trailing whitespace
Collapse consecutive whitespace
Unescape HTML entities
Replace Smart quotes with ascii
To titlecase
To uppercase
To lowercase
To number
To date
To text
To null
To empty string

Season1Date and Season1Time were then each split into multiple columns so that it would be easier to compare start and end dates/times later once it is dumped into SQLLite. We split Seaseon1Date, Season2Date, Season3Date and Season4Date with " to " and splitted Season1Time, Season2Time, Season3Time and Season4Time with ";" to create multiple time slots which created 8 different columns for each Time column.


Split column Season1Date into several columns

How to Split Column

● by separator

Separator: to

□ regular expression

Split into: columns at most (leave blank for no limit)

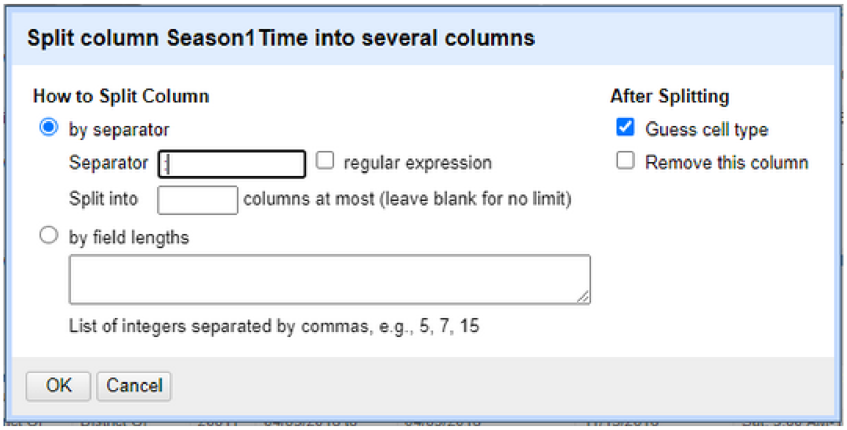
○ by field lengths

List of integers separated by commas, e.g., 5, 7, 15

After Splitting

☑ Guess cell type

☑ Remove this column



Split column Season1Time into several columns

How to Split Column

● by separator

Separator: : ☐ regular expression

Split into [ ] columns at most (leave blank for no limit)

○ by field lengths

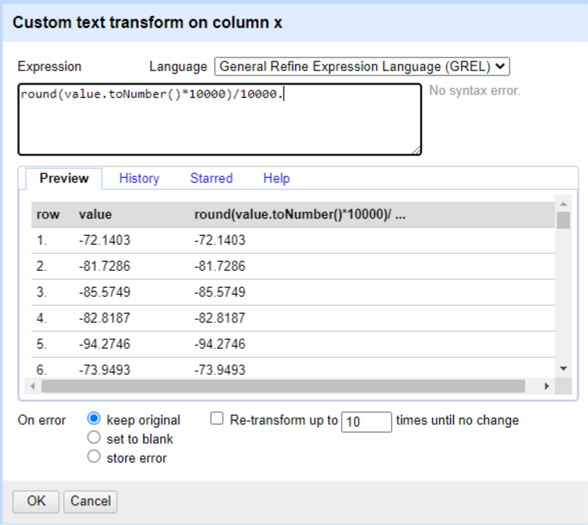
List of integers separated by commas, e.g., 5, 7, 15

After Splitting

☑ Guess cell type

☐ Remove this column

X and Y columns were normalized to 4 decimal places for consistency's sake



Custom text transform on column x

Expression — Language: General Refine Expression Language (GREL)

```
round(value.toNumber()*10000)/10000.
```

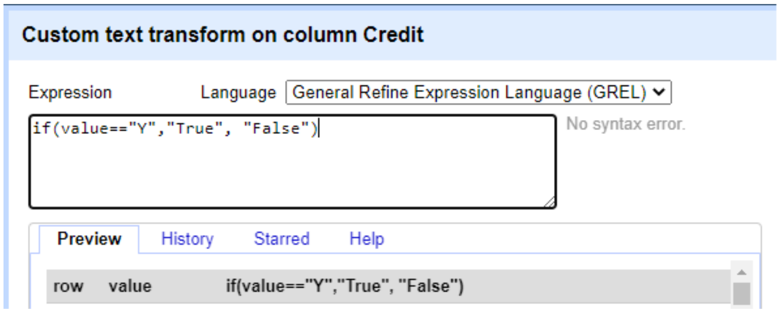
No syntax error.

row	value	round(value.toNumber()*10000)/ ...
1.	-72.1403	-72.1403
2.	-81.7286	-81.7286
3.	-85.5749	-85.5749
4.	-82.8187	-82.8187
5.	-94.2746	-94.2746
6.	-73.9493	-73.9493

On error: ● keep original ○ set to blank ○ store error

☐ Re-transform up to 10 times until no change

All the remaining columns were converted from "Y" / "N" to true/false Boolean values, and any "-" values were converted to Null. This was so the use case could just directly compare the results to see if a product was available rather than needing to convert to Boolean in the client. Here we had to decide between choosing Null and defaulting to false, but OpenRefine defaults null values to false



Custom text transform on column Credit

Expression Language General Refine Expression Language (GREL)

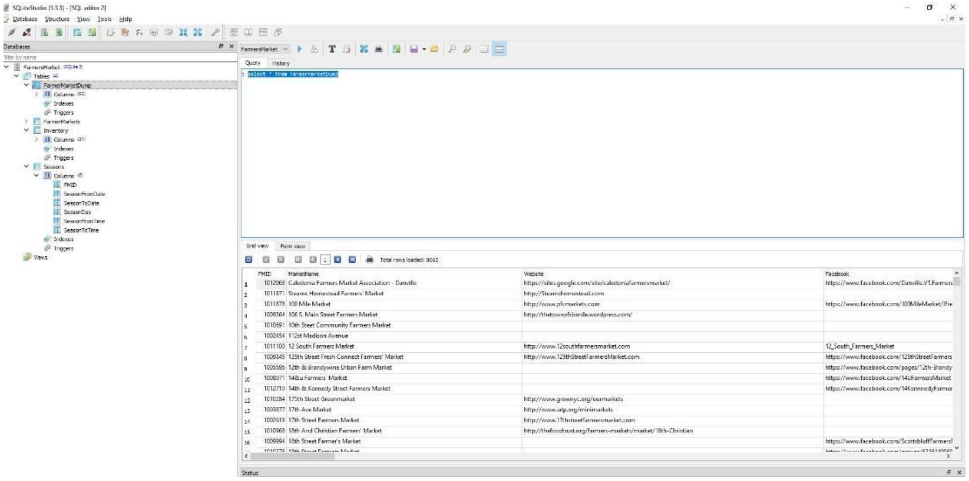
```
if(value=="Y","True", "False")
```

No syntax error.

if(value=="Y","True", "False")

SQLLite

Once we finish the initial data cleaning in OpenRefine, we extracted the clean data from OpenRefine and load the data into SQLite (FarmerMarketDump). This table would represent the entirety of the cleaned data.



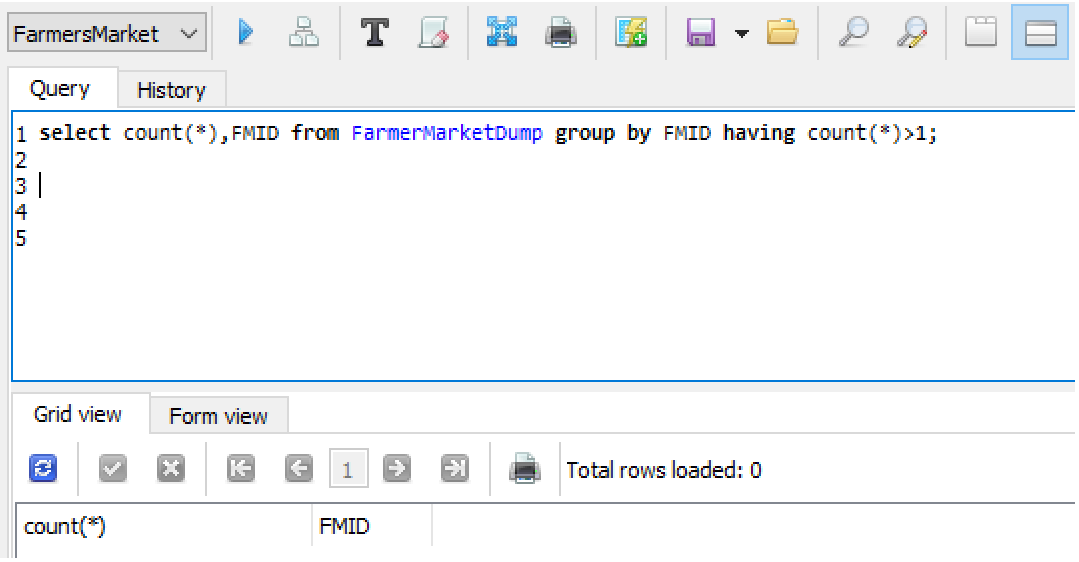
```
select * from FarmerMarketDump
```

Total rows loaded: 8665

	FMID	MarketName	Website	Facebook
1	1012063	Caledonia Farmers Market Association - Danville	https://sites.google.com/site/caledoniafarmersmarket/	https://www.facebook.com/Danville.VT.Farmers
2	1011871	Stearns Homestead Farmers' Market	http://Stearnshomestead.com	
3	1011878	100 Mile Market	http://www.pfcmarkets.com	https://www.facebook.com/100MileMarket/?fre
4	1009364	106 S. Main Street Farmers Market	http://thetownofsixmile.wordpress.com/	
5	1010691	10th Steet Community Farmers Market		
6	1002454	112st Madison Avenue		
7	1011100	12 South Farmers Market	http://www.12southfarmersmarket.com	12_South_Farmers_Market
8	1009845	125th Street Fresh Connect Farmers' Market	http://www.125thStreetFarmersMarket.com	https://www.facebook.com/125thStreetFarmers
9	1005586	12th & Brandywine Urban Farm Market		https://www.facebook.com/pages/12th-Brandy
10	1008071	14&u Farmers' Market		https://www.facebook.com/14UFarmersMarket
11	1012710	14th & Kennedy Street Farmers Market		https://www.facebook.com/14KennnedyFarmer
12	1010284	175th Street Greenmarket	http://www.grownyc.org/ourmarkets	
13	1003877	17th Ave Market	http://www.iatp.org/minimarkets	
14	1002619	17th Street Farmers Market	http://www.17thstreetfarmersmarket.com	
15	1010968	18th And Christian Farmers' Market	http://thefoodtrust.org/farmers-markets/market/18th-Christian	
16	1009994	18th Street Farmer's Market		https://www.facebook.com/ScottsbluffFarmers!

Integrity Constraint Check

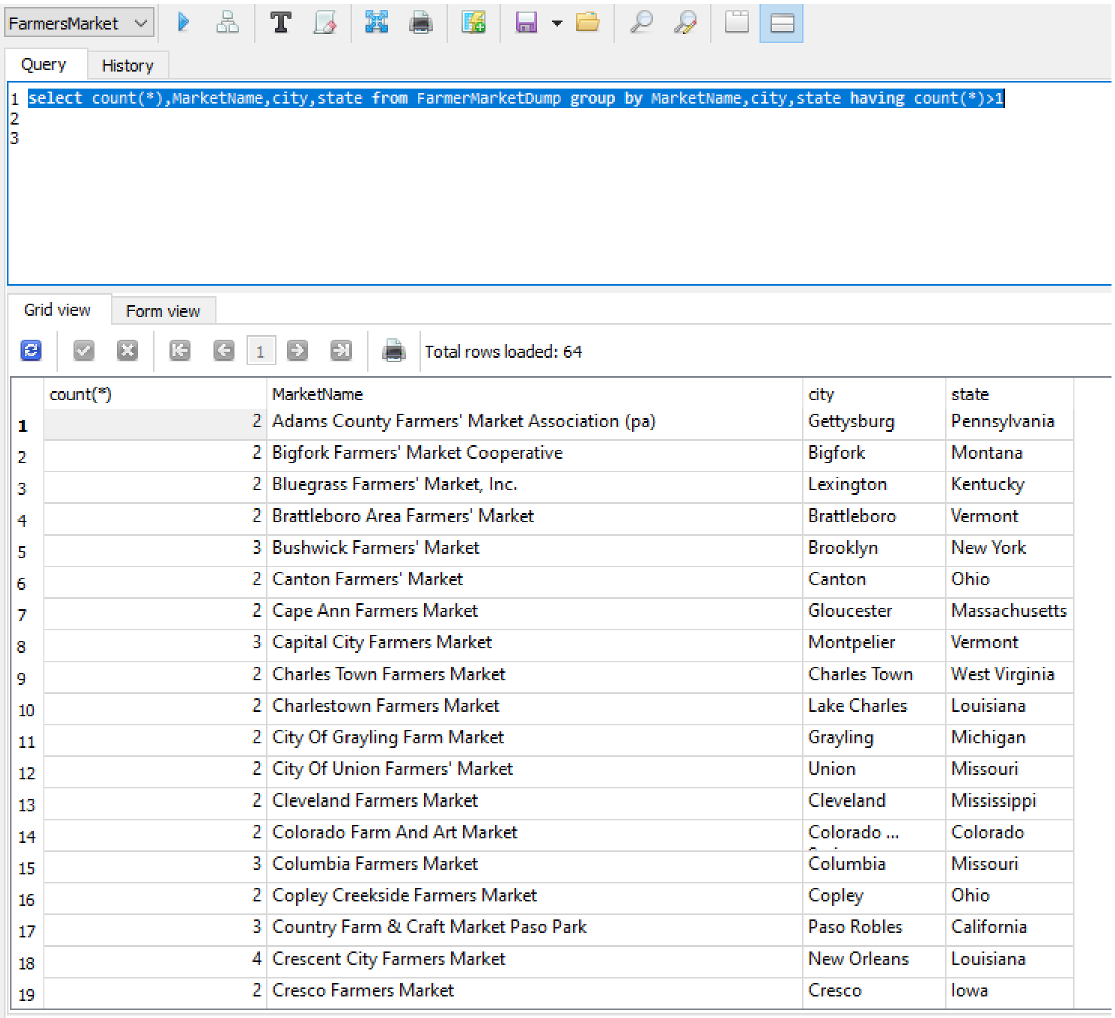
We checked Integrity Constraint violation by grouping by FMID which is unique as below. But we found multiple FMID data for the same Market under same state and city which we cleaned by by deleting duplicates as below.



```
select count(*),FMID from FarmerMarketDump group by FMID having count(*)>1;
```

Total rows loaded: 0

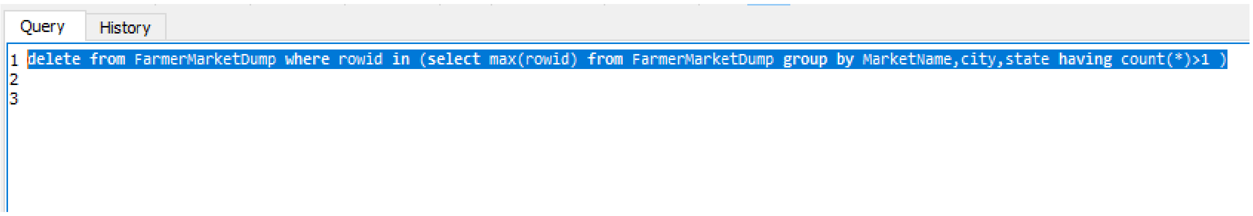
count(*)	FMID



```
select count(*),MarketName,city,state from FarmerMarketDump group by MarketName,city,state having count(*)>1
```

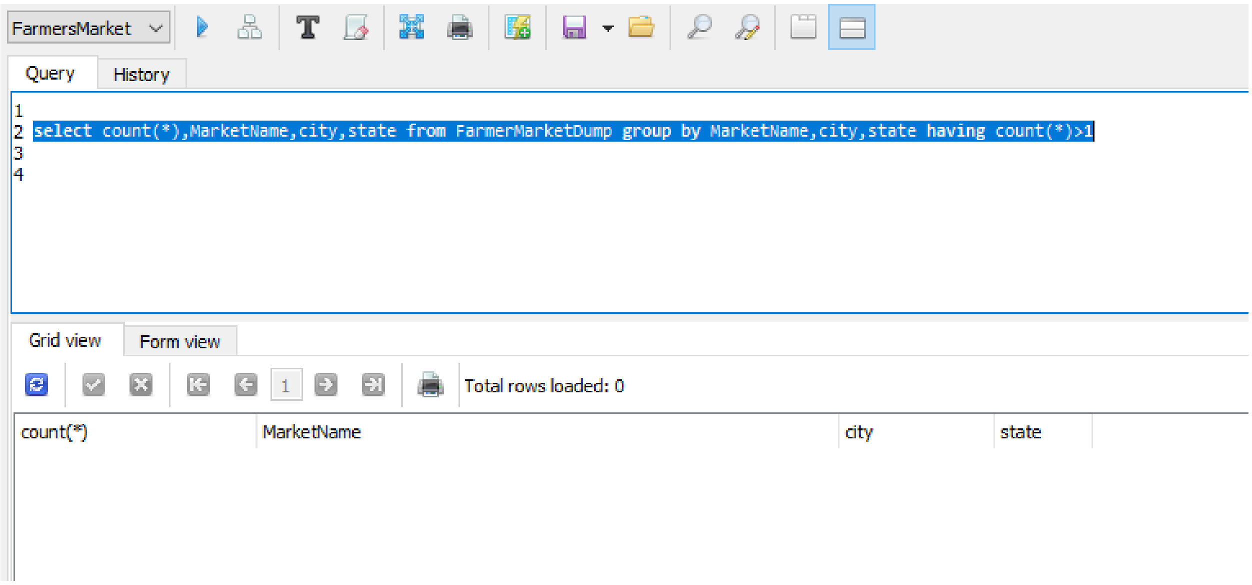
Total rows loaded: 64

	count(*)	MarketName	city	state
1	2	Adams County Farmers' Market Association (pa)	Gettysburg	Pennsylvania
2	2	Bigfork Farmers' Market Cooperative	Bigfork	Montana
3	2	Bluegrass Farmers' Market, Inc.	Lexington	Kentucky
4	2	Brattleboro Area Farmers' Market	Brattleboro	Vermont
5	3	Bushwick Farmers' Market	Brooklyn	New York
6	2	Canton Farmers' Market	Canton	Ohio
7	2	Cape Ann Farmers Market	Gloucester	Massachusetts
8	3	Capital City Farmers Market	Montpelier	Vermont
9	2	Charles Town Farmers Market	Charles Town	West Virginia
10	2	Charlestown Farmers Market	Lake Charles	Louisiana
11	2	City Of Grayling Farm Market	Grayling	Michigan
12	2	City Of Union Farmers' Market	Union	Missouri
13	2	Cleveland Farmers Market	Cleveland	Mississippi
14	2	Colorado Farm And Art Market	Colorado ...	Colorado
15	3	Columbia Farmers Market	Columbia	Missouri
16	2	Copley Creekside Farmers Market	Copley	Ohio
17	3	Country Farm & Craft Market Paso Park	Paso Robles	California
18	4	Crescent City Farmers Market	New Orleans	Louisiana
19	2	Cresco Farmers Market	Cresco	Iowa



```
delete from FarmerMarketDump where rowid in (select max(rowid) from FarmerMarketDump group by MarketName,city,state having count(*)>1 )
```

After clean up, we can see that it does not consist of any duplicate rows.



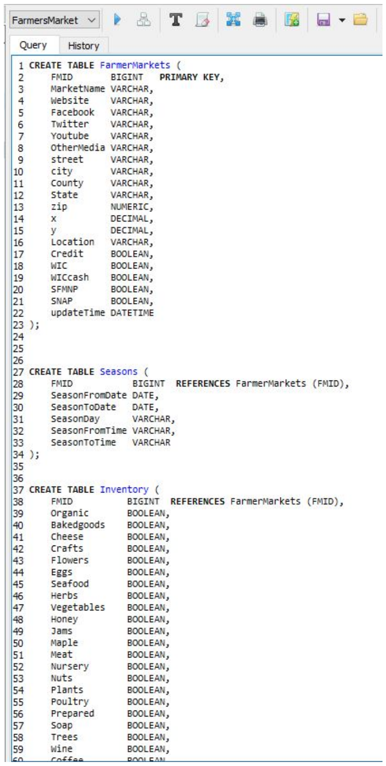
```

select count(*),MarketName,city,state from FarmerMarketDump group by MarketName,city,state having count(*)>1
```

Total rows loaded: 0

count(*)	MarketName	city	state

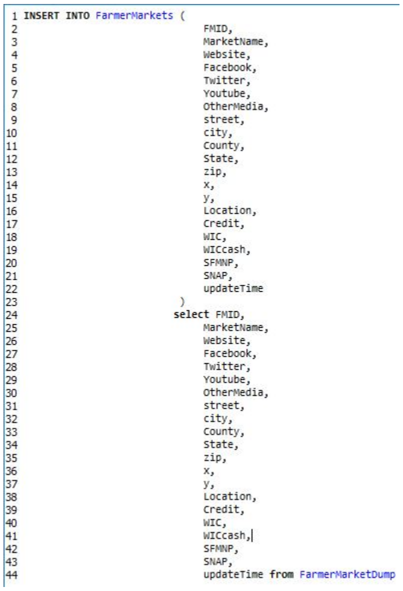
Split Data in Respected Tables and Data Quality Changes

Then we created 3 separate tables to organize the data. **FarmerMarkets** holds the info about the Market itself, including the Name, and Social Networking values, as well as its geographic location. **Seasons** stores the open/close dates and times for seasonal markets. **Inventory** holds Boolean data for what products a market does or does not provide

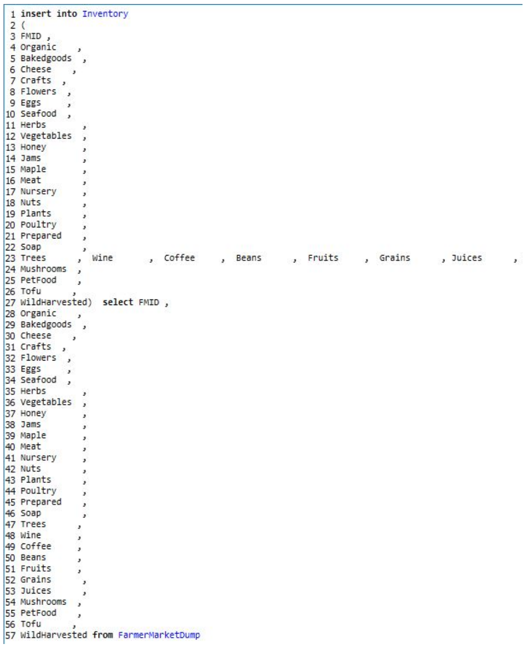


```sql
CREATE TABLE FarmerMarkets (
    FMID       BIGINT   PRIMARY KEY,
    MarketName VARCHAR,
    Website    VARCHAR,
    Facebook   VARCHAR,
    Twitter    VARCHAR,
    Youtube    VARCHAR,
    OtherMedia VARCHAR,
    street     VARCHAR,
    city       VARCHAR,
    County     VARCHAR,
    State      VARCHAR,
    zip        NUMERIC,
    x          DECIMAL,
    y          DECIMAL,
    Location   VARCHAR,
    Credit     BOOLEAN,
    WIC        BOOLEAN,
    WICcash    BOOLEAN,
    SFMNP      BOOLEAN,
    SNAP       BOOLEAN,
    updateTime DATETIME
);



CREATE TABLE Seasons (
    FMID           BIGINT  REFERENCES FarmerMarkets (FMID),
    SeasonFromDate DATE,
    SeasonToDate   DATE,
    SeasonDay      VARCHAR,
    SeasonFromTime VARCHAR,
    SeasonToTime   VARCHAR
);


CREATE TABLE Inventory (
    FMID          BIGINT  REFERENCES FarmerMarkets (FMID),
    Organic       BOOLEAN,
    Bakedgoods    BOOLEAN,
    Cheese        BOOLEAN,
    Crafts        BOOLEAN,
    Flowers       BOOLEAN,
    Eggs          BOOLEAN,
    Seafood       BOOLEAN,
    Herbs         BOOLEAN,
    Vegetables    BOOLEAN,
    Honey         BOOLEAN,
    Jams          BOOLEAN,
    Maple         BOOLEAN,
    Meat          BOOLEAN,
    Nursery       BOOLEAN,
    Nuts          BOOLEAN,
    Plants        BOOLEAN,
    Poultry       BOOLEAN,
    Prepared      BOOLEAN,
    Soap          BOOLEAN,
    Trees         BOOLEAN,
    Wine          BOOLEAN,
    Coffee        BOOLEAN
```

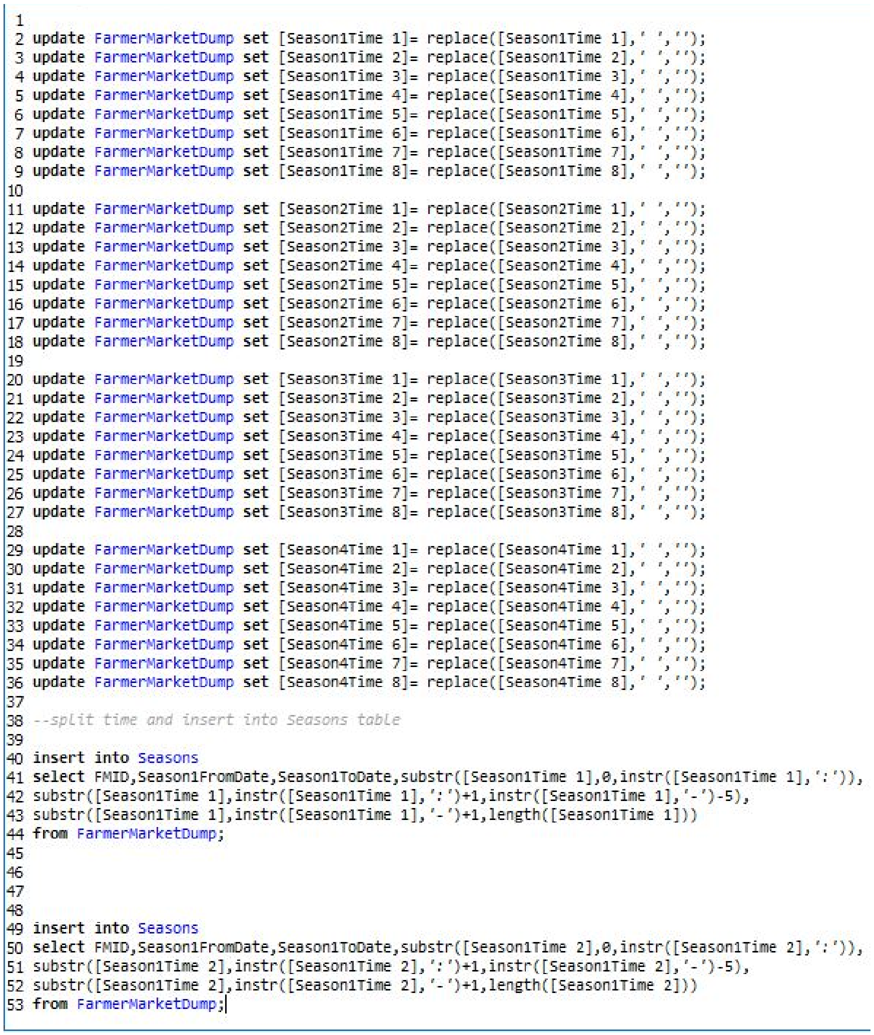
The Data was split and distributed into their respective tables to organize the data so our app would be able to search along all 3 axes (By Store, By Time, By Products) with ease.



```sql
INSERT INTO FarmerMarkets (
                              FMID,
                              MarketName,
                              Website,
                              Facebook,
                              Twitter,
                              Youtube,
                              OtherMedia,
                              street,
                              city,
                              County,
                              State,
                              zip,
                              x,
                              y,
                              Location,
                              Credit,
                              WIC,
                              WICcash,
                              SFMNP,
                              SNAP,
                              updateTime
                          )
                         select FMID,
                              MarketName,
                              Website,
                              Facebook,
                              Twitter,
                              Youtube,
                              OtherMedia,
                              street,
                              city,
                              County,
                              State,
                              zip,
                              x,
                              y,
                              Location,
                              Credit,
                              WIC,
                              WICcash,
                              SFMNP,
                              SNAP,
                              updateTime from FarmerMarketDump
```

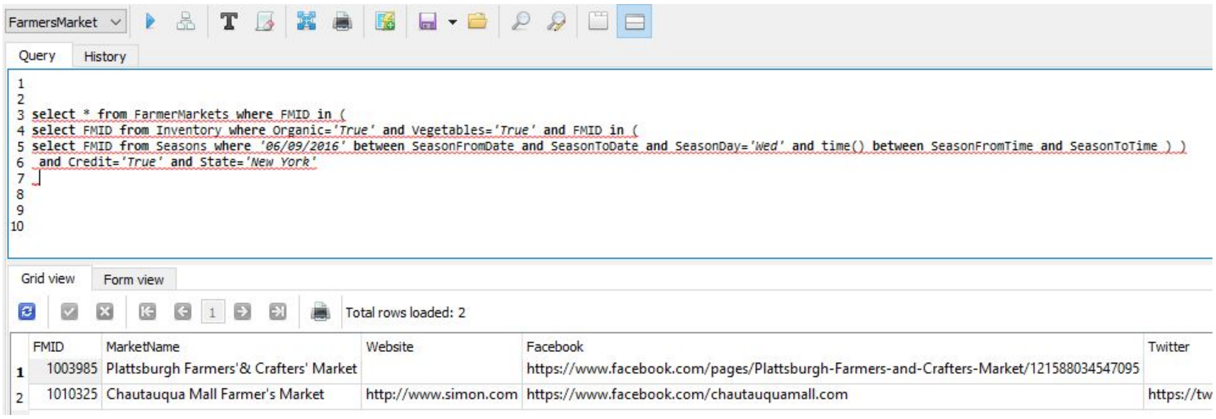


```
insert into Inventory
(
FMID ,
Organic    ,
Bakedgoods  ,
Cheese    ,
Crafts  ,
Flowers  ,
Eggs     ,
Seafood  ,
Herbs        ,
Vegetables  ,
Honey        ,
Jams         ,
Maple        ,
Meat         ,
Nursery      ,
Nuts         ,
Plants       ,
Poultry      ,
Prepared     ,
Soap         ,
Trees       ,  Wine       ,  Coffee     ,  Beans      ,  Fruits     ,  Grains      , Juices      ,
Mushrooms  ,
PetFood    ,
Tofu      ,
WildHarvested)  select FMID ,
Organic    ,
Bakedgoods  ,
Cheese    ,
Crafts  ,
Flowers  ,
Eggs     ,
Seafood  ,
Herbs        ,
Vegetables  ,
Honey        ,
Jams         ,
Maple        ,
Meat         ,
Nursery      ,
Nuts         ,
Plants       ,
Poultry      ,
Prepared     ,
Soap         ,
Trees       ,
Wine        ,
Coffee      ,
Beans       ,
Fruits      ,
Grains       ,
Juices       ,
Mushrooms  ,
PetFood    ,
Tofu      ,
WildHarvested from FarmerMarketDump
```



```

update FarmerMarketDump set [Season1Time 1]= replace([Season1Time 1],’ ’,’’);
update FarmerMarketDump set [Season1Time 2]= replace([Season1Time 2],’ ’,’’);
update FarmerMarketDump set [Season1Time 3]= replace([Season1Time 3],’ ’,’’);
update FarmerMarketDump set [Season1Time 4]= replace([Season1Time 4],’ ’,’’);
update FarmerMarketDump set [Season1Time 5]= replace([Season1Time 5],’ ’,’’);
update FarmerMarketDump set [Season1Time 6]= replace([Season1Time 6],’ ’,’’);
update FarmerMarketDump set [Season1Time 7]= replace([Season1Time 7],’ ’,’’);
update FarmerMarketDump set [Season1Time 8]= replace([Season1Time 8],’ ’,’’);

update FarmerMarketDump set [Season2Time 1]= replace([Season2Time 1],’ ’,’’);
update FarmerMarketDump set [Season2Time 2]= replace([Season2Time 2],’ ’,’’);
update FarmerMarketDump set [Season2Time 3]= replace([Season2Time 3],’ ’,’’);
update FarmerMarketDump set [Season2Time 4]= replace([Season2Time 4],’ ’,’’);
update FarmerMarketDump set [Season2Time 5]= replace([Season2Time 5],’ ’,’’);
update FarmerMarketDump set [Season2Time 6]= replace([Season2Time 6],’ ’,’’);
update FarmerMarketDump set [Season2Time 7]= replace([Season2Time 7],’ ’,’’);
update FarmerMarketDump set [Season2Time 8]= replace([Season2Time 8],’ ’,’’);

update FarmerMarketDump set [Season3Time 1]= replace([Season3Time 1],’ ’,’’);
update FarmerMarketDump set [Season3Time 2]= replace([Season3Time 2],’ ’,’’);
update FarmerMarketDump set [Season3Time 3]= replace([Season3Time 3],’ ’,’’);
update FarmerMarketDump set [Season3Time 4]= replace([Season3Time 4],’ ’,’’);
update FarmerMarketDump set [Season3Time 5]= replace([Season3Time 5],’ ’,’’);
update FarmerMarketDump set [Season3Time 6]= replace([Season3Time 6],’ ’,’’);
update FarmerMarketDump set [Season3Time 7]= replace([Season3Time 7],’ ’,’’);
update FarmerMarketDump set [Season3Time 8]= replace([Season3Time 8],’ ’,’’);

update FarmerMarketDump set [Season4Time 1]= replace([Season4Time 1],’ ’,’’);
update FarmerMarketDump set [Season4Time 2]= replace([Season4Time 2],’ ’,’’);
update FarmerMarketDump set [Season4Time 3]= replace([Season4Time 3],’ ’,’’);
update FarmerMarketDump set [Season4Time 4]= replace([Season4Time 4],’ ’,’’);
update FarmerMarketDump set [Season4Time 5]= replace([Season4Time 5],’ ’,’’);
update FarmerMarketDump set [Season4Time 6]= replace([Season4Time 6],’ ’,’’);
update FarmerMarketDump set [Season4Time 7]= replace([Season4Time 7],’ ’,’’);
update FarmerMarketDump set [Season4Time 8]= replace([Season4Time 8],’ ’,’’);

--split time and insert into Seasons table

insert into Seasons
select FMID,Season1FromDate,Season1ToDate,substr([Season1Time 1],0,instr([Season1Time 1],’:’)),
substr([Season1Time 1],instr([Season1Time 1],’:’)+1,instr([Season1Time 1],’-’)-5),
substr([Season1Time 1],instr([Season1Time 1],’-’)+1,length([Season1Time 1]))
from FarmerMarketDump;




insert into Seasons
select FMID,Season1FromDate,Season1ToDate,substr([Season1Time 2],0,instr([Season1Time 2],’:’)),
substr([Season1Time 2],instr([Season1Time 2],’:’)+1,instr([Season1Time 2],’-’)-5),
substr([Season1Time 2],instr([Season1Time 2],’-’)+1,length([Season1Time 2]))
from FarmerMarketDump;
```

Once the data was separated, we generated the data for the use case. For example: “What are the nearby stores in New York City that accept credit cards and are open today that sells Organics and Vegetables''

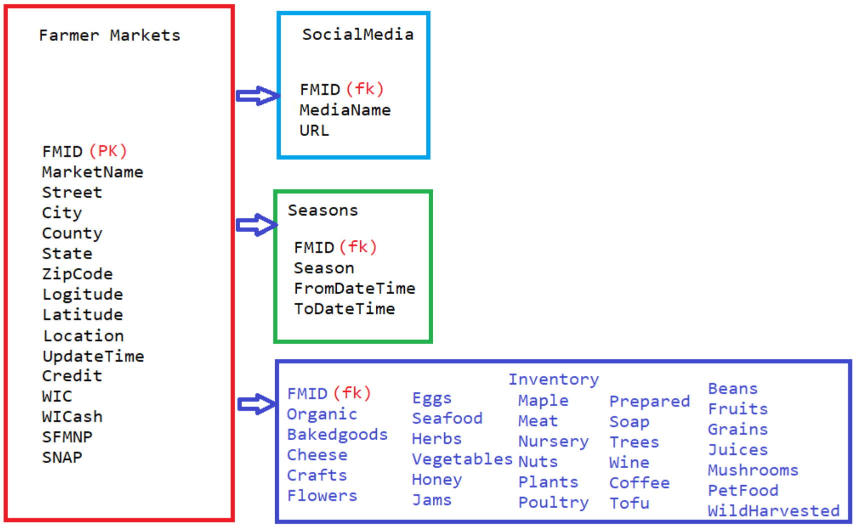


```
select * from FarmerMarkets where FMID in (
select FMID from Inventory where Organic='True' and Vegetables='True' and FMID in (
select FMID from Seasons where '06/09/2016' between SeasonFromDate and SeasonToDate and SeasonDay='Wed' and time() between SeasonFromTime and SeasonToTime ) )
 and Credit='True' and State='New York'
```

Total rows loaded: 2

	FMID	MarketName	Website	Facebook	Twitter
1	1003985	Plattsburgh Farmers'& Crafters' Market		https://www.facebook.com/pages/Plattsburgh-Farmers-and-Crafters-Market/121588034547095	
2	1010325	Chautauqua Mall Farmer's Market	http://www.simon.com	https://www.facebook.com/chautauquamall.com	https://tw

There were some steps from our initial Phase 1 that did not go as planned. For example: unlike our ER diagram in phase one, we decided against creating a SocialMedia table as there wasn’t enough available data for it to be valuable to our Use Case.

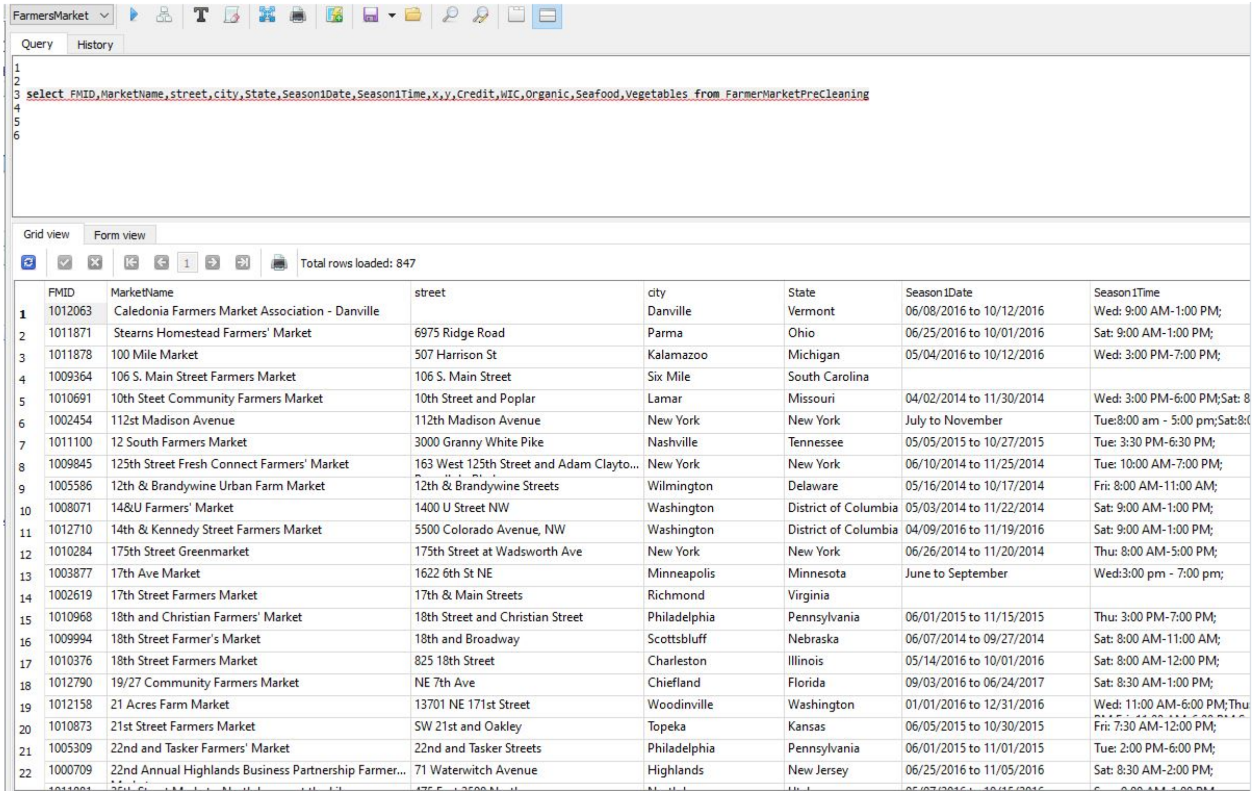


Similarly, we considered using data log however, so we used OpenRefine and SQLLite for the majority of our data cleaning because it was the most efficient tool out of the ones we've been taught for the level of data cleaning that was required for our chosen use case. Similarly, SQLite provided a simple way to organize and retrieve the data for the use case.

Cleaned Data:

Here you can see just some of the rows and columns that were improved by our data cleaning

Before:

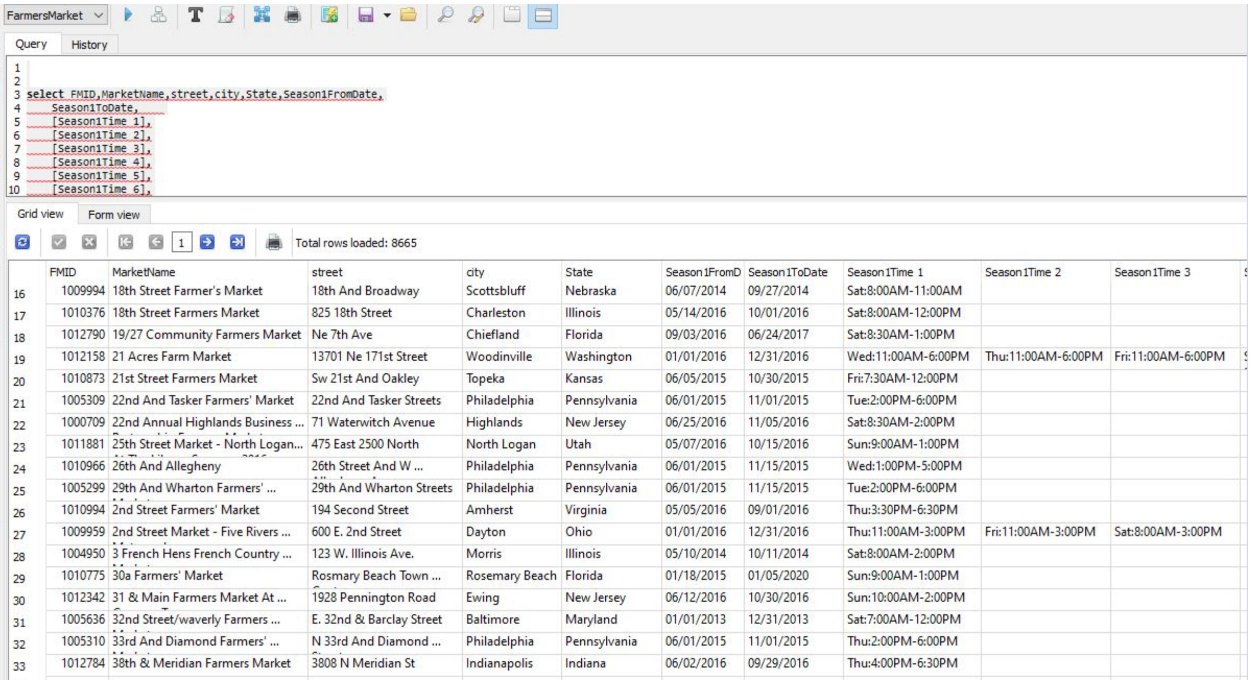


```
select FMID,MarketName,street,city,State,Season1Date,Season1Time,x,y,Credit,WIC,Organic,Seafood,Vegetables from FarmerMarketPreCleaning
```

Total rows loaded: 847

	FMID	MarketName	street	city	State	Season1Date	Season1Time
1	1012063	Caledonia Farmers Market Association - Danville		Danville	Vermont	06/08/2016 to 10/12/2016	Wed: 9:00 AM-1:00 PM;
2	1011871	Stearns Homestead Farmers' Market	6975 Ridge Road	Parma	Ohio	06/25/2016 to 10/01/2016	Sat: 9:00 AM-1:00 PM;
3	1011878	100 Mile Market	507 Harrison St	Kalamazoo	Michigan	05/04/2016 to 10/12/2016	Wed: 3:00 PM-7:00 PM;
4	1009364	106 S. Main Street Farmers Market	106 S. Main Street	Six Mile	South Carolina		
5	1010691	10th Steet Community Farmers Market	10th Street and Poplar	Lamar	Missouri	04/02/2014 to 11/30/2014	Wed: 3:00 PM-6:00 PM;Sat: 8
6	1002454	112st Madison Avenue	112th Madison Avenue	New York	New York	July to November	Tue:8:00 am - 5:00 pm;Sat:8:(
7	1011100	12 South Farmers Market	3000 Granny White Pike	Nashville	Tennessee	05/05/2015 to 10/27/2015	Tue: 3:30 PM-6:30 PM;
8	1009845	125th Street Fresh Connect Farmers' Market	163 West 125th Street and Adam Clayto...	New York	New York	06/10/2014 to 11/25/2014	Tue: 10:00 AM-7:00 PM;
9	1005586	12th & Brandywine Urban Farm Market	12th & Brandywine Streets	Wilmington	Delaware	05/16/2014 to 10/17/2014	Fri: 8:00 AM-11:00 AM;
10	1008071	14&U Farmers' Market	1400 U Street NW	Washington	District of Columbia	05/03/2014 to 11/22/2014	Sat: 9:00 AM-1:00 PM;
11	1012710	14th & Kennedy Street Farmers Market	5500 Colorado Avenue, NW	Washington	District of Columbia	04/09/2016 to 11/19/2016	Sat: 9:00 AM-1:00 PM;
12	1010284	175th Street Greenmarket	175th Street at Wadsworth Ave	New York	New York	06/26/2014 to 11/20/2014	Thu: 8:00 AM-5:00 PM;
13	1003877	17th Ave Market	1622 6th St NE	Minneapolis	Minnesota	June to September	Wed:3:00 pm - 7:00 pm;
14	1002619	17th Street Farmers Market	17th & Main Streets	Richmond	Virginia		
15	1010968	18th and Christian Farmers' Market	18th Street and Christian Street	Philadelphia	Pennsylvania	06/01/2015 to 11/15/2015	Thu: 3:00 PM-7:00 PM;
16	1009994	18th Street Farmer's Market	18th and Broadway	Scottsbluff	Nebraska	06/07/2014 to 09/27/2014	Sat: 8:00 AM-11:00 AM;
17	1010376	18th Street Farmers Market	825 18th Street	Charleston	Illinois	05/14/2016 to 10/01/2016	Sat: 8:00 AM-12:00 PM;
18	1012790	19/27 Community Farmers Market	NE 7th Ave	Chiefland	Florida	09/03/2016 to 06/24/2017	Sat: 8:30 AM-1:00 PM;
19	1012158	21 Acres Farm Market	13701 NE 171st Street	Woodinville	Washington	01/01/2016 to 12/31/2016	Wed: 11:00 AM-6:00 PM;Thu
20	1010873	21st Street Farmers Market	SW 21st and Oakley	Topeka	Kansas	06/05/2015 to 10/30/2015	Fri: 7:30 AM-12:00 PM;
21	1005309	22nd and Tasker Farmers' Market	22nd and Tasker Streets	Philadelphia	Pennsylvania	06/01/2015 to 11/01/2015	Tue: 2:00 PM-6:00 PM;
22	1000709	22nd Annual Highlands Business Partnership Farmer...	71 Waterwitch Avenue	Highlands	New Jersey	06/25/2016 to 11/05/2016	Sat: 8:30 AM-2:00 PM;

After:



```
select FMID,MarketName,street,city,State,Season1FromDate,
    Season1ToDate,
    [Season1Time 1],
    [Season1Time 2],
    [Season1Time 3],
    [Season1Time 4],
    [Season1Time 5],
    [Season1Time 6],
```

Total rows loaded: 8665

	FMID	MarketName	street	city	State	Season1FromD	Season1ToDate	Season1Time 1	Season1Time 2	Season1Time 3
16	1009994	18th Street Farmer's Market	18th And Broadway	Scottsbluff	Nebraska	06/07/2014	09/27/2014	Sat:8:00AM-11:00AM		
17	1010376	18th Street Farmers Market	825 18th Street	Charleston	Illinois	05/14/2016	10/01/2016	Sat:8:00AM-12:00PM		
18	1012790	19/27 Community Farmers Market	Ne 7th Ave	Chiefland	Florida	09/03/2016	06/24/2017	Sat:8:30AM-1:00PM		
19	1012158	21 Acres Farm Market	13701 Ne 171st Street	Woodinville	Washington	01/01/2016	12/31/2016	Wed:11:00AM-6:00PM	Thu:11:00AM-6:00PM	Fri:11:00AM-6:00PM
20	1010873	21st Street Farmers Market	Sw 21st And Oakley	Topeka	Kansas	06/05/2015	10/30/2015	Fri:7:30AM-12:00PM		
21	1005309	22nd And Tasker Farmers' Market	22nd And Tasker Streets	Philadelphia	Pennsylvania	06/01/2015	11/01/2015	Tue:2:00PM-6:00PM		
22	1000709	22nd Annual Highlands Business ...	71 Waterwitch Avenue	Highlands	New Jersey	06/25/2016	11/05/2016	Sat:8:30AM-2:00PM		
23	1011881	25th Street Market - North Logan...	475 East 2500 North	North Logan	Utah	05/07/2016	10/15/2016	Sun:9:00AM-1:00PM		
24	1010966	26th And Allegheny	26th Street And W ...	Philadelphia	Pennsylvania	06/01/2015	11/15/2015	Wed:1:00PM-5:00PM		
25	1005299	29th And Wharton Farmers' ...	29th And Wharton Streets	Philadelphia	Pennsylvania	06/01/2015	11/15/2015	Tue:2:00PM-6:00PM		
26	1010994	2nd Street Farmers' Market	194 Second Street	Amherst	Virginia	05/05/2016	09/01/2016	Thu:3:30PM-6:30PM		
27	1009959	2nd Street Market - Five Rivers ...	600 E. 2nd Street	Dayton	Ohio	01/01/2016	12/31/2016	Thu:11:00AM-3:00PM	Fri:11:00AM-3:00PM	Sat:8:00AM-3:00PM
28	1004950	3 French Hens French Country ...	123 W. Illinois Ave.	Morris	Illinois	05/10/2014	10/11/2014	Sat:8:00AM-2:00PM		
29	1010775	30a Farmers' Market	Rosmary Beach Town ...	Rosemary Beach	Florida	01/18/2015	01/05/2020	Sun:9:00AM-1:00PM		
30	1012342	31 & Main Farmers Market At ...	1928 Pennington Road	Ewing	New Jersey	06/12/2016	10/30/2016	Sun:10:00AM-2:00PM		
31	1005636	32nd Street/waverly Farmers ...	E. 32nd & Barclay Street	Baltimore	Maryland	01/01/2013	12/31/2013	Sat:7:00AM-12:00PM		
32	1005310	33rd And Diamond Farmers' ...	N 33rd And Diamond ...	Philadelphia	Pennsylvania	06/01/2015	11/01/2015	Thu:2:00PM-6:00PM		
33	1012784	38th & Meridian Farmers Market	3808 N Meridian St	Indianapolis	Indiana	06/02/2016	09/29/2016	Thu:4:00PM-6:30PM		

<u>Supplementary Materials:</u>

- **Workflow:**
  1. WorkFlowLinear.pdf
  2. WrokFlowParallel.pdf (please download and zoom)
  3. WorkFlowLinear.gv
  4. WorkFlowLinear.yw
  5. yw.properties

Commands used to generate above files are below.

```
or2yw -i OpenRefineOperations.json  -ot pdf -o WorkFlowLinear.pdf
or2yw -i OpenRefineOperations.json  -ot gv -o WorkFlowLinear.gv
or2yw -i OpenRefineOperations.json  -ot yw -o WorkFlowLinear.yw
or2yw -i OpenRefineOperations.json  -ot pdf -o WorkFlowParallel.pdf -t parallel
```

- **OpenRefine:**
  OpernRefineOperations.json

- **Queries:**
  queries.txt
- **Datasets:**
  1. Clean:
  2. Dirty:

Conclusions:

One of the problems that we encountered was splitting the dates, using OpenRefine, it created a lot of columns because some rows had multiple date/times in one value so we decided to split the date into 2 columns (to and from) and listed the times split by ";" Then in SQLite we split that even further. Some of the date data was also difficult to convert as they were just months rather than day/month/year. Next step we can make this application more intelligent by taking the user's current timezone and use it to check current open markets based on the geolocation of the user from where he is sending the request.

Repository : https://github.com/bkbathani/CS513_FinalProject

References :

https://pypi.org/project/or2ywtool/

Team Member Contribution

Initially both Bhushan and Zoheb researched through all datasets and chose farmers market dataset to use for the final project considering team size and timeline. Zoheb worked through the data to identify the use case and tried to study data, meanwhile Bhushan worked on yesworkflow, openrefine and SQLLite setup and research. Both worked together later to go through the files and apply changes and scripted the data cleaning part on zoom call.