

# Nearest Grocery Store

Fawzia Al-Bluwi

ERAU - CS540

# Objectives

- Collecting Volusia groceries data.
- Calculating the nearest grocery distance for each parcel in Volusia county.
- Adding an attribute for the distance of the nearest grocery in the parcel table.
- Showing the groceries dataset as a layer in QGIS.
- Usage of a groceries distance dataset.

# 1. Collecting Groceries Dataset

- Groceries are not just listed under one LUC, but several LUCs, and not limited, of 1100 (Stores, One Story), 1400 (Supermarkets), and 1600 (Community Shopping Centers).
- Therefore, groceries were compiled depending on **Owner** attribute, **own1**, and by searching manually for all possible known groceries.
- The following where clause has shortened to conditions that get the equivalent result after searching for all possible groceries have compiled in this database.

```
1 select p.parid, p.luc, p.luc_desc, o.own1, p.geom
2 from volusia.owner o join volusia.parcel p on o.parid=p.parid
3 where p.geom is not null
4 and p.luc in ('1100', '1400', '1600')
5 and (
6 (o.own1 ilike '%publix%')
7 or (o.own1 ilike '%walmart%')
8 or (o.own1 ilike '%7-Eleven%')
9 or (o.own1 ilike '%LOVE WHOLE FOODS%')
10 or (o.own1 ilike '%food mart%')
11 or (o.own1 ilike '%FLORIDAS CITRUS WORLD INC%')
12 )
13 order by p.luc
```

	parid double precision	luc text	luc_desc text	own1 text	geom geometry	
1	3508742	1100	Stores, One Sto...	7-ELEVEN INC	0106000020BC0...	
2	3086449	1100	Stores, One Sto...	7-ELEVEN INC	0106000020BC0...	
3	3183746	1100	Stores, One Sto...	7-ELEVEN INC	0106000020BC0...	
4	3119088	1100	Stores, One Sto...	7-ELEVEN INC	0106000020BC0...	
5	3395409	1100	Stores, One Sto...	7-ELEVEN INC	0106000020BC0...	

# 1. Collecting Groceries Dataset

- Then extracted these records into a ***Groceries*** table.
- You can find these dataset in **groceries\_data.txt**
- **Note:** This dataset does not include all actual groceries in the county. Thus, further data compiling and manual searching are needed for getting all actual groceries.

```
1 select p.parid, p.luc, p.luc_desc, o.ownl, p.geom into volusia.groceries
2 from select * from volusia.owner o join volusia.parcel p on o.parid=p.parid
3 where p.geom is not null
4 and p.luc in ('1100', '1400', '1600')
5 and (
6 (o.ownl ilike '%publix%')
7 or (o.ownl ilike '%walmart%')
8 or (o.ownl ilike '%7-Eleven%')
9 or (o.ownl ilike '%LOVE WHOLE FOODS%')
10 or (o.ownl ilike '%food mart%')
11 or (o.ownl ilike '%FLORIDAS CITRUS WORLD INC%')
12 )
13 order by p.luc
```

## 2. Nearest Grocery Distance Calculation

- Groceries distance is measured in miles and calculated by this query.
- You can find these distance dataset in **groceries\_distance.txt**

```
1  -- the closet grocery to a random parcel (divided by 5280 for distances in miles)
2  select p.parid, p.geom, p.luc, p.luc_desc, ST_Distance(p.geom, (select p2.geom
3                                                                    from volusia.parcel p2
4                                                                    where parid=3565215)
5                                                                    )/5280 as groc_distance
6  from volusia.parcel p join volusia.groceries g on p.parid=g.parid
7  order by p.geom <-> (select p2.geom from volusia.parcel p2 where parid=3565215)
8  limit 1;
```

### 3. Adding Nearest Grocery Distance to Parcels

- Add a ***groc\_distance*** column to the ***parcel*** table to store nearest grocery distance of all parcel.

```
1  -- add a column to the parcel table
2  alter table volusia.parcel add column groc_distance double precision;
```

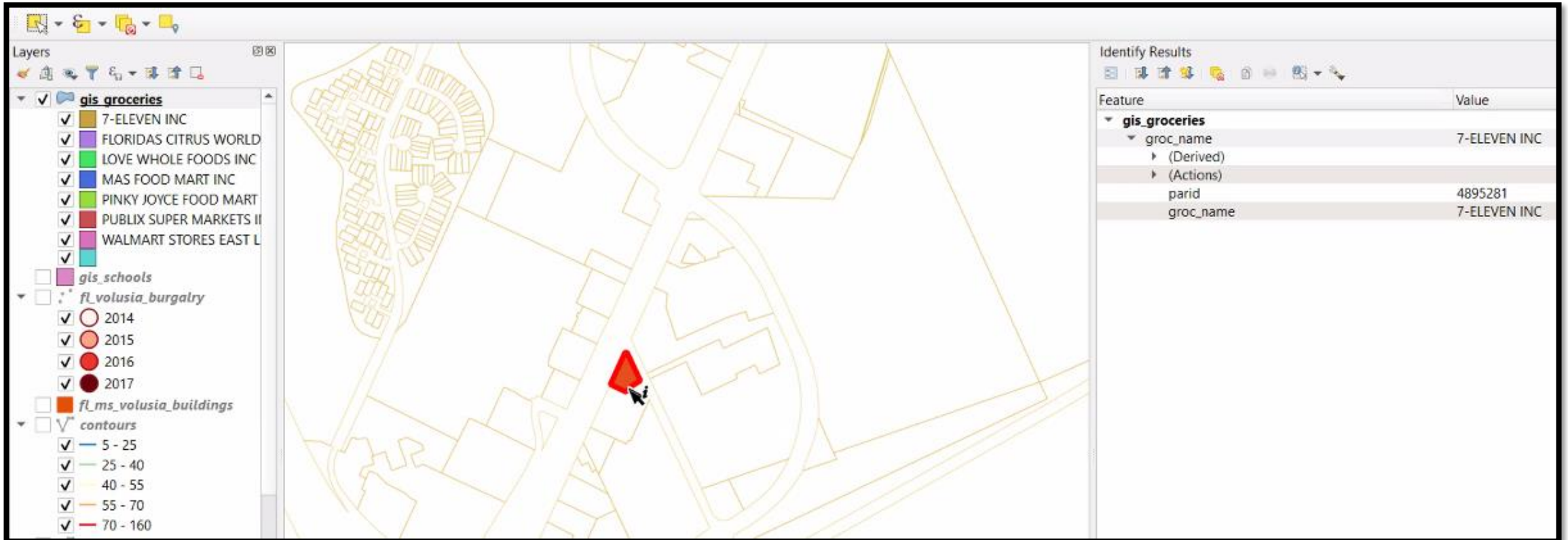
### 3. Adding Nearest Grocery Distance to Parcels

- Then, update every *parcel* record to find the distance to closet grocery. This query works for one parcel at a time.

```
1  -- update to find distance from a parcel to its closet grocery
2  -- this works for one parcel at a time (divided by 5280 for distances in miles)
3  update volusia.parcel p1 set groc_distance = ST_Distance(p1.geom, p2.geom)/5280
4  from volusia.parcel p2
5  where p1.parid=2004291 and p2.parid=2469498;
```

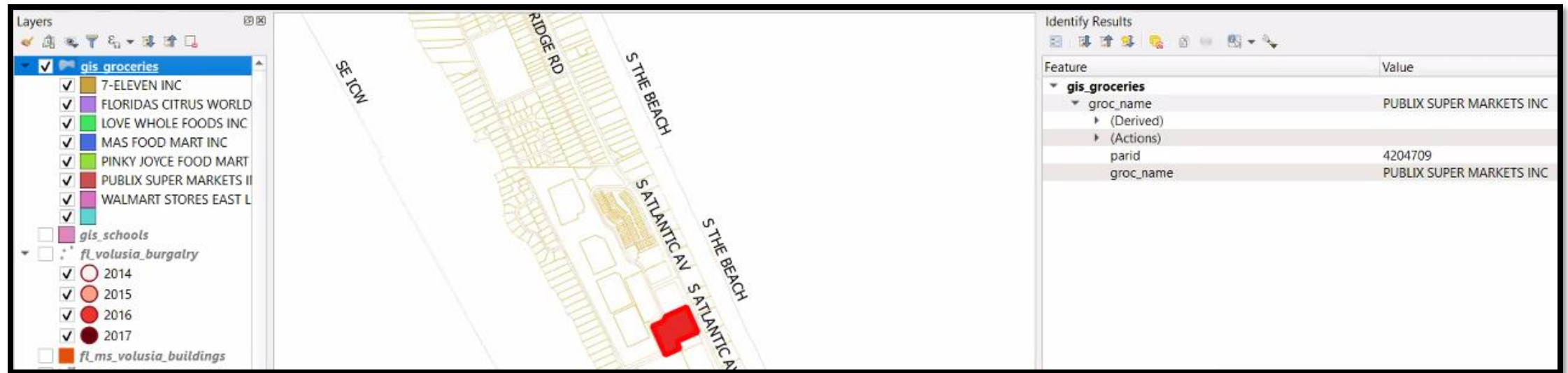
- Using a loop is needed to complete that for each parcel. Run the Python script, **update\_grocery\_distances.py**, to update all parcel records.

## 4. Showing Groceries QGIS Layer

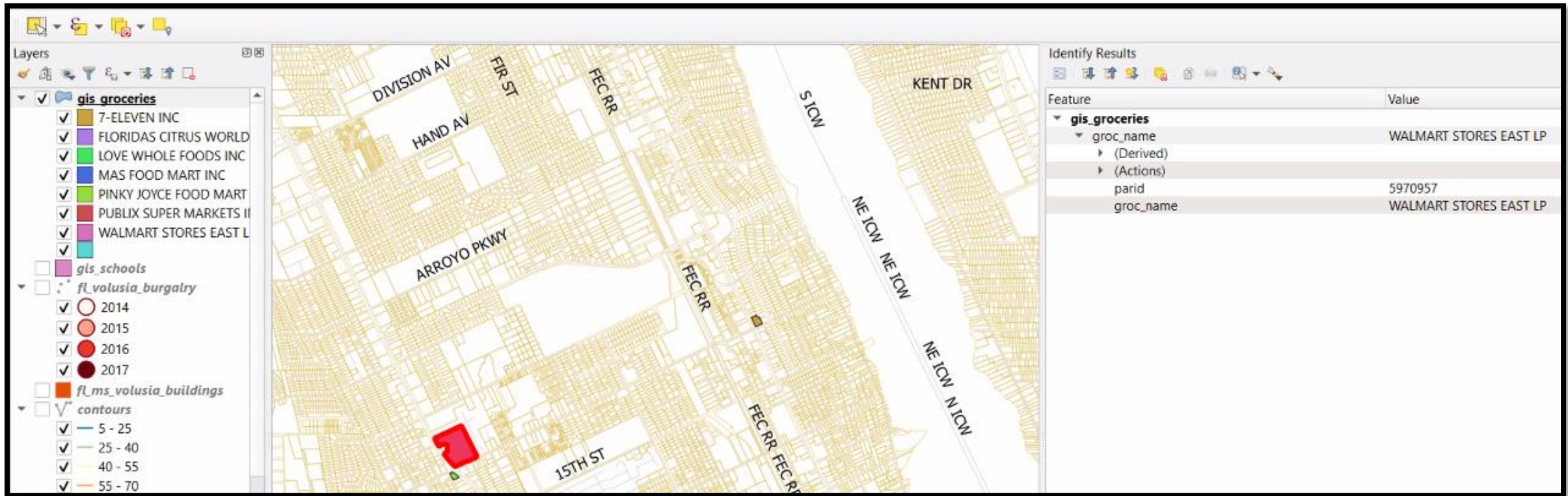




## 4. Showing Groceries QGIS Layer



## 4. Showing Groceries QGIS Layer



## 4. Showing Groceries QGIS Layer



## 5. Usage of Groceries Distance Dataset

- For use the dataset in your database, run these queries  
**load\_groceries\_distances\_data.sql** to create a temp table with *parid* and *groc\_distance*, load the data into, and then update joined with *parcel table* based on *parid*.
- This is a sample of **groceries\_distance.txt** in the zip file.

1	parid	luc	luc_desc	own1	geom
2	3395409	1100	Stores, One Story	7-ELEVEN INC	0106000020BC
3	2234971	1100	Stores, One Story	MAS FOOD MART INC	01060000
4	3246233	1100	Stores, One Story	7-ELEVEN INC	0106000020BC
5	4895281	1100	Stores, One Story	7-ELEVEN INC	0106000020BC
6	2470381	1100	Stores, One Story	FLORIDAS CITRUS WORLD INC	
7	3183746	1100	Stores, One Story	7-ELEVEN INC	0106000020BC
8	3473299	1100	Stores, One Story	7-ELEVEN INC	0106000020BC
9	3051696	1100	Stores, One Story	7-ELEVEN INC	0106000020BC
10	7550539	1100	Stores, One Story	7-ELEVEN INC	0106000020BC
11	3821980	1100	Stores, One Story	7-ELEVEN INC	0106000020BC
12	3194829	1100	Stores, One Story	PINKY JOYCE FOOD MART INC	
13	3300424	1100	Stores, One Story	7-ELEVEN INC	0106000020BC
14	3119088	1100	Stores, One Story	7-ELEVEN INC	0106000020BC
15	3245491	1100	Stores, One Story	7-ELEVEN INC	0106000020BC