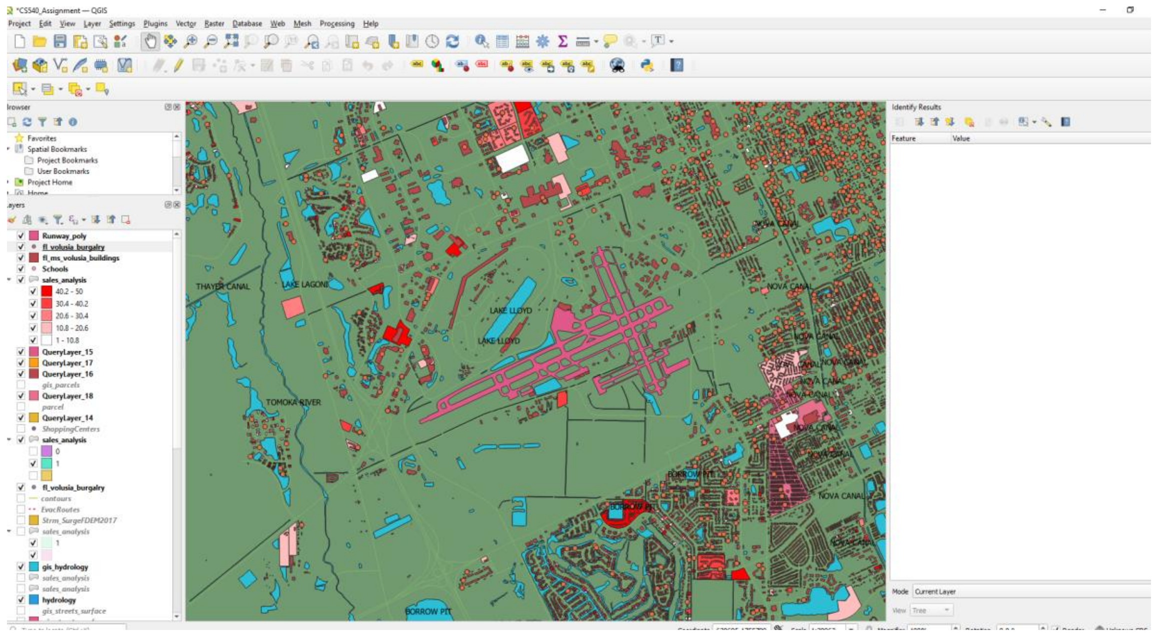
Adding Column Feature and Price Prediction

Shlok Misra, Xi Luo, Matthew Getter

I will be finding distances from any runway. My hypothesis is that there is a relationship between distance of a home from the runway and the price of a house.

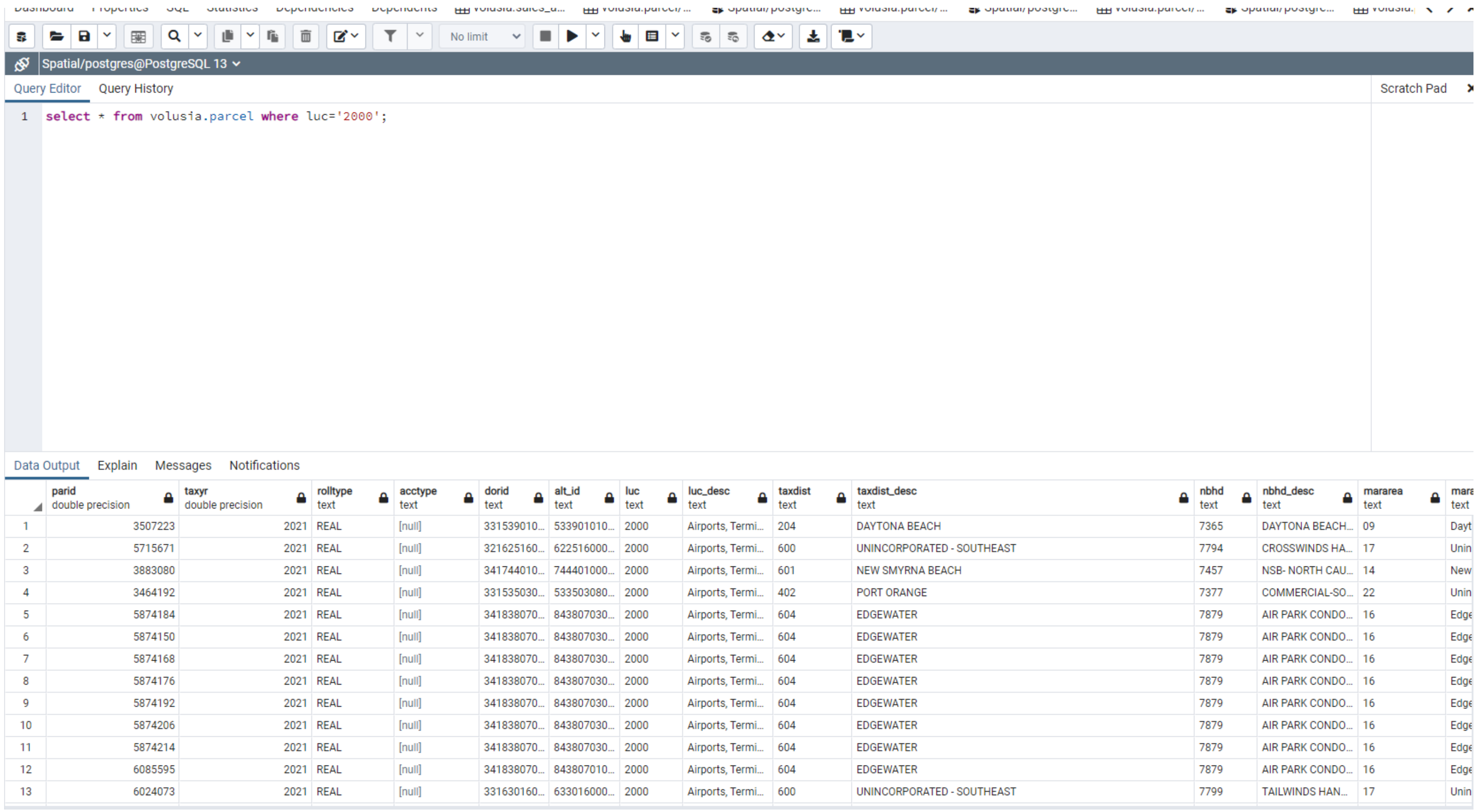
- I have loaded the runway polygon from Volusia.org/gis.



Note

- I have only loaded the runway at Daytona Beach Airport on QGIS. However, the properties on my pgadmin or my analysis will include distances from all runways in my area such as Daytona beach, Ormond Beach, New Smyrna and Edgewater/Massey.

Retrieving Properties

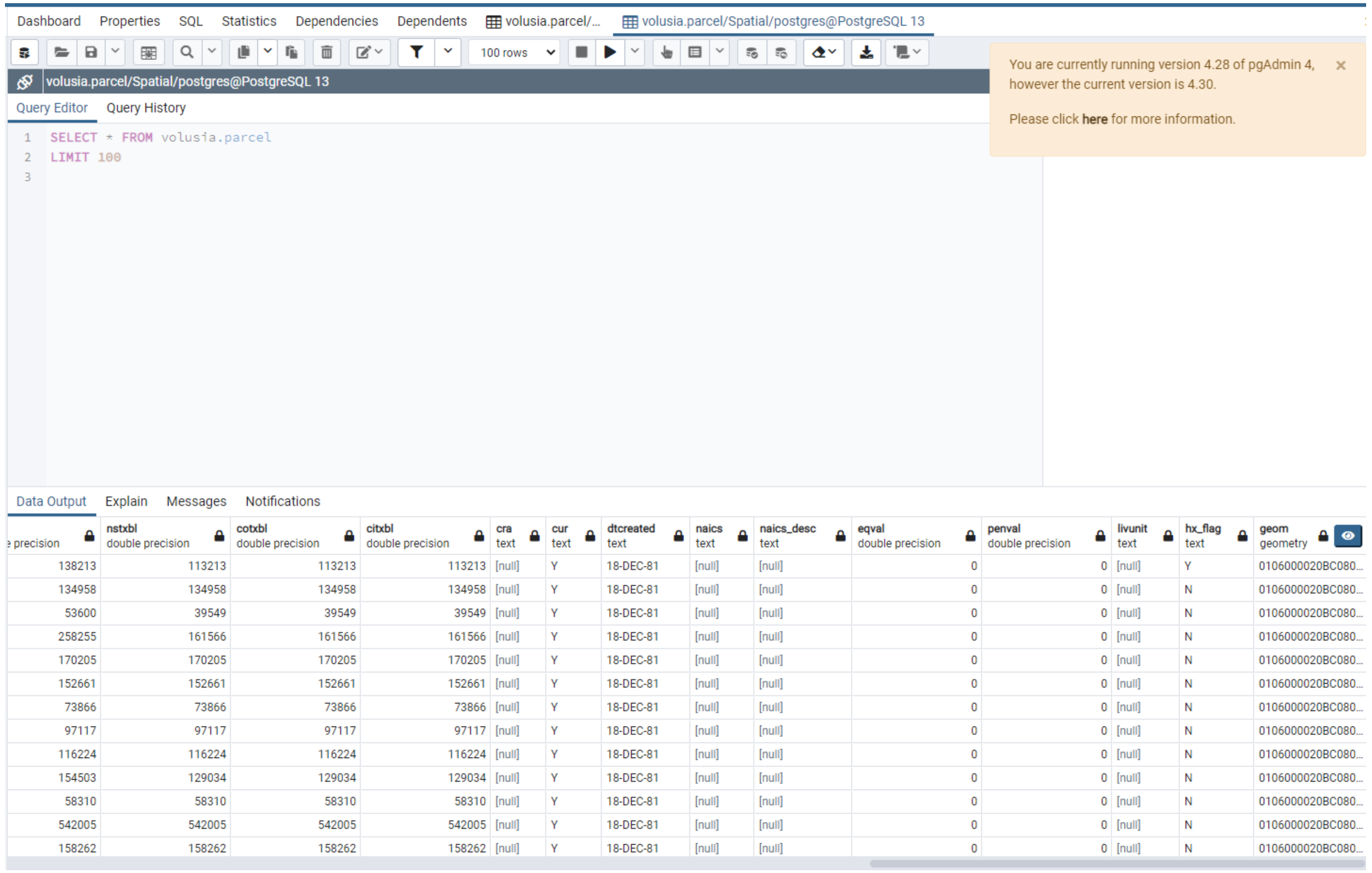


Spatial/postgres@PostgreSQL 13

```
select * from volusia.parcel where luc='2000';
```

	parid double precision	taxyr double precision	rolltype text	acctype text	dorid text	alt_id text	luc text	luc_desc text	taxdist text	taxdist_desc text	nbhd text	nbhd_desc text	mararea text	mara text
1	3507223	2021	REAL	[null]	331539010...	533901010...	2000	Airports, Termi...	204	DAYTONA BEACH	7365	DAYTONA BEACH...	09	Dayt
2	5715671	2021	REAL	[null]	321625160...	622516000...	2000	Airports, Termi...	600	UNINCORPORATED - SOUTHEAST	7794	CROSSWINDS HA...	17	Unin
3	3883080	2021	REAL	[null]	341744010...	744401000...	2000	Airports, Termi...	601	NEW SMYRNA BEACH	7457	NSB- NORTH CAU...	14	New
4	3464192	2021	REAL	[null]	331535030...	533503080...	2000	Airports, Termi...	402	PORT ORANGE	7377	COMMERCIAL-SO...	22	Unin
5	5874184	2021	REAL	[null]	341838070...	843807030...	2000	Airports, Termi...	604	EDGEWATER	7879	AIR PARK CONDO...	16	Edge
6	5874150	2021	REAL	[null]	341838070...	843807030...	2000	Airports, Termi...	604	EDGEWATER	7879	AIR PARK CONDO...	16	Edge
7	5874168	2021	REAL	[null]	341838070...	843807030...	2000	Airports, Termi...	604	EDGEWATER	7879	AIR PARK CONDO...	16	Edge
8	5874176	2021	REAL	[null]	341838070...	843807030...	2000	Airports, Termi...	604	EDGEWATER	7879	AIR PARK CONDO...	16	Edge
9	5874192	2021	REAL	[null]	341838070...	843807030...	2000	Airports, Termi...	604	EDGEWATER	7879	AIR PARK CONDO...	16	Edge
10	5874206	2021	REAL	[null]	341838070...	843807030...	2000	Airports, Termi...	604	EDGEWATER	7879	AIR PARK CONDO...	16	Edge
11	5874214	2021	REAL	[null]	341838070...	843807030...	2000	Airports, Termi...	604	EDGEWATER	7879	AIR PARK CONDO...	16	Edge
12	6085595	2021	REAL	[null]	341838070...	843807010...	2000	Airports, Termi...	604	EDGEWATER	7879	AIR PARK CONDO...	16	Edge
13	6024073	2021	REAL	[null]	331630160...	633016000...	2000	Airports, Termi...	600	UNINCORPORATED - SOUTHEAST	7799	TAILWINDS HAN...	17	Unin

The GEOM column was added



Dashboard Properties SQL Statistics Dependencies Dependents volusia.parcel/... volusia.parcel/Spatial/postgres@PostgreSQL 13

volusia.parcel/Spatial/postgres@PostgreSQL 13

You are currently running version 4.28 of pgAdmin 4, however the current version is 4.30.

Please click **here** for more information.

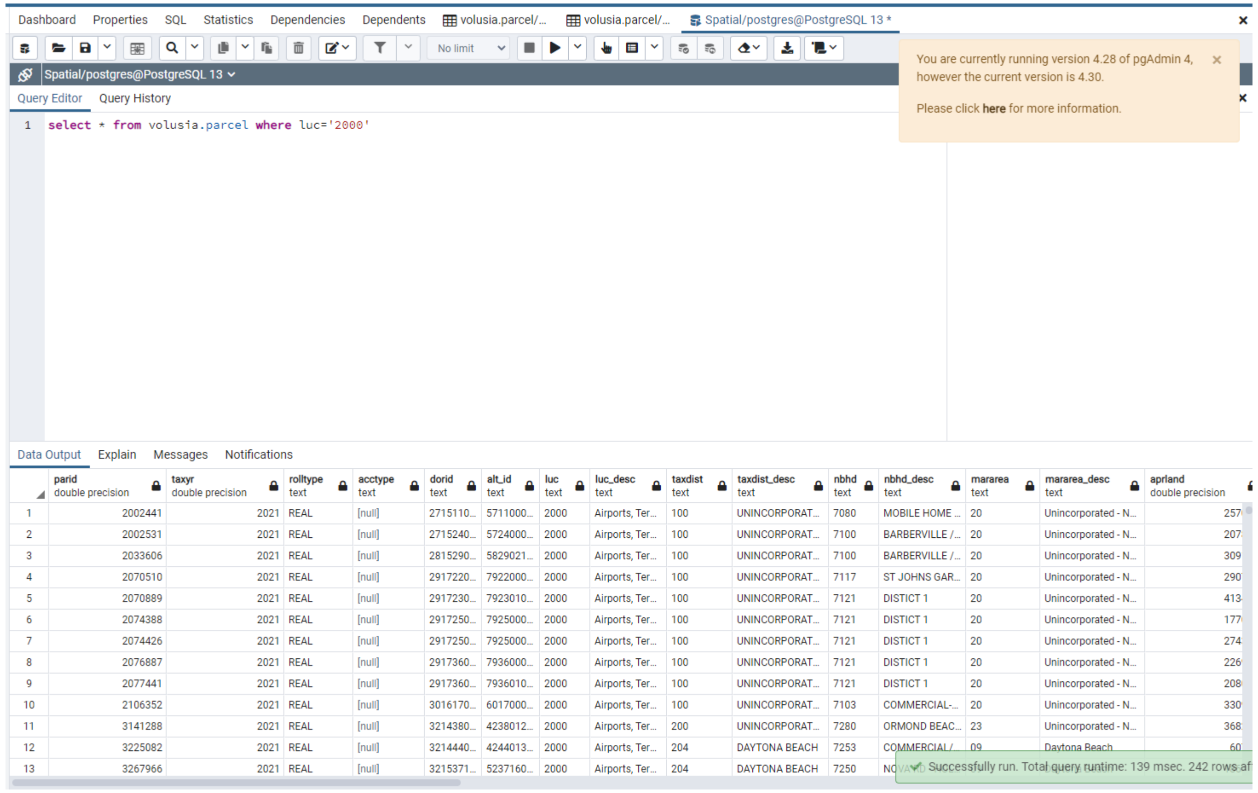
Query Editor Query History

```
SELECT * FROM volusia.parcel
LIMIT 100
```

Data Output Explain Messages Notifications

precision	nstxbl double precision	cotxbl double precision	citxbl double precision	cra text	cur text	dtcreated text	naics text	naics_desc text	eqval double precision	penval double precision	livunit text	hx_flag text	geom geometry
138213	113213	113213	113213	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	Y	0106000020BC080...
134958	134958	134958	134958	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	N	0106000020BC080...
53600	39549	39549	39549	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	N	0106000020BC080...
258255	161566	161566	161566	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	N	0106000020BC080...
170205	170205	170205	170205	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	N	0106000020BC080...
152661	152661	152661	152661	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	N	0106000020BC080...
73866	73866	73866	73866	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	N	0106000020BC080...
97117	97117	97117	97117	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	N	0106000020BC080...
116224	116224	116224	116224	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	N	0106000020BC080...
154503	129034	129034	129034	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	N	0106000020BC080...
58310	58310	58310	58310	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	N	0106000020BC080...
542005	542005	542005	542005	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	N	0106000020BC080...
158262	158262	158262	158262	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	N	0106000020BC080...

We now want Distance between every parcel and the nearest runway. These are all the airport parcels in the parcel table



Dashboard | Properties | SQL | Statistics | Dependencies | Dependents | volusia.parcel/... | volusia.parcel/... | Spatial/postgres@PostgreSQL 13 *

Spatial/postgres@PostgreSQL 13

Query Editor | Query History

```
select * from volusia.parcel where luc='2000'
```

You are currently running version 4.28 of pgAdmin 4, however the current version is 4.30.

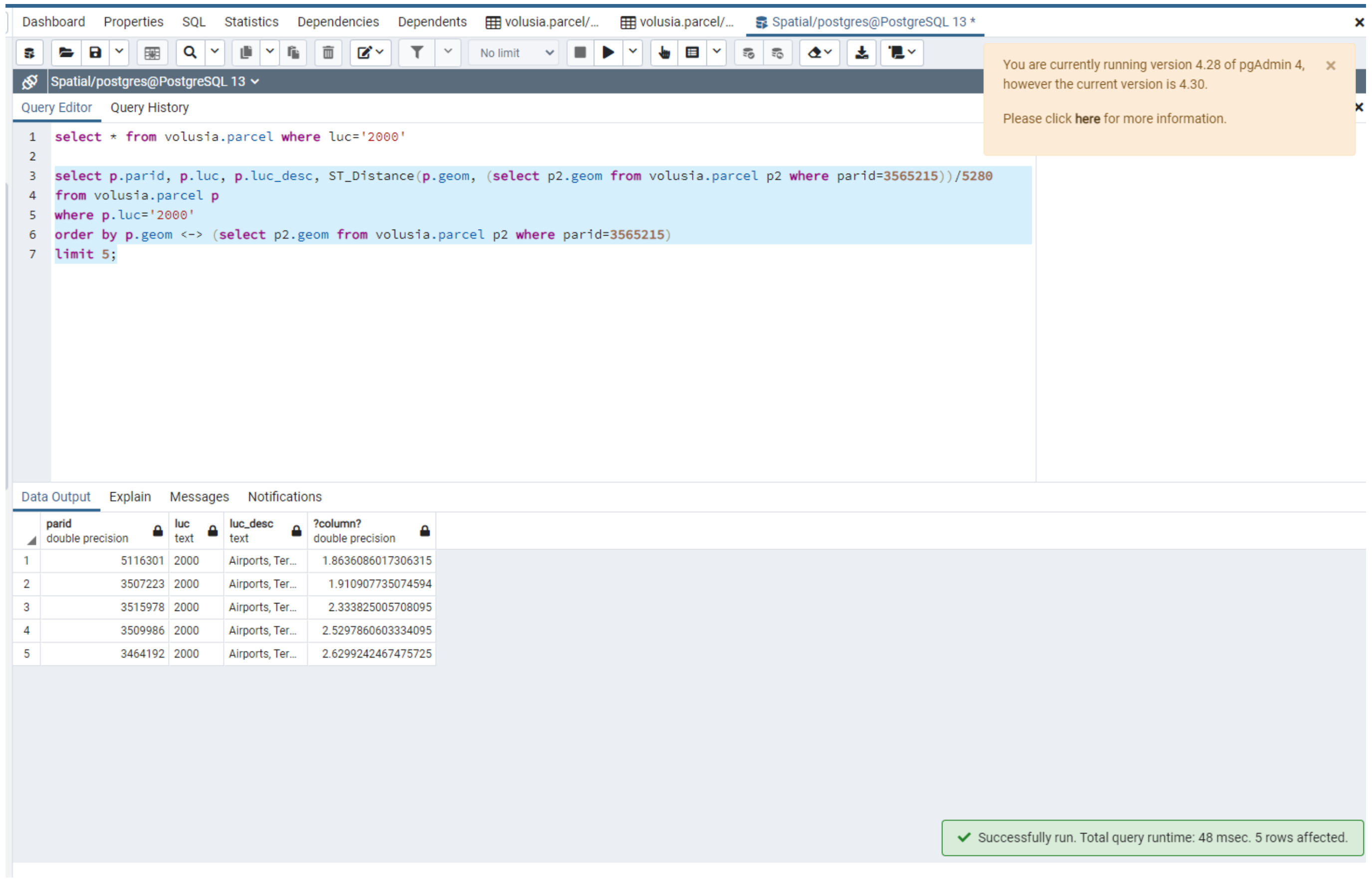
Please click **here** for more information.

Data Output | Explain | Messages | Notifications

	parid double precision	taxyr double precision	rolltype text	acctype text	dorid text	alt_id text	luc text	luc_desc text	taxdist text	taxdist_desc text	nbhd text	nbhd_desc text	mararea text	mararea_desc text	aprland double precision
1	2002441	2021	REAL	[null]	2715110...	5711000...	2000	Airports, Ter...	100	UNINCORPORAT...	7080	MOBILE HOME ...	20	Unincorporated - N...	257
2	2002531	2021	REAL	[null]	2715240...	5724000...	2000	Airports, Ter...	100	UNINCORPORAT...	7100	BARBERVILLE /...	20	Unincorporated - N...	207
3	2033606	2021	REAL	[null]	2815290...	5829021...	2000	Airports, Ter...	100	UNINCORPORAT...	7100	BARBERVILLE /...	20	Unincorporated - N...	309
4	2070510	2021	REAL	[null]	2917220...	7922000...	2000	Airports, Ter...	100	UNINCORPORAT...	7117	ST JOHNS GAR...	20	Unincorporated - N...	290
5	2070889	2021	REAL	[null]	2917230...	7923010...	2000	Airports, Ter...	100	UNINCORPORAT...	7121	DISTICT 1	20	Unincorporated - N...	413
6	2074388	2021	REAL	[null]	2917250...	7925000...	2000	Airports, Ter...	100	UNINCORPORAT...	7121	DISTICT 1	20	Unincorporated - N...	177
7	2074426	2021	REAL	[null]	2917250...	7925000...	2000	Airports, Ter...	100	UNINCORPORAT...	7121	DISTICT 1	20	Unincorporated - N...	274
8	2076887	2021	REAL	[null]	2917360...	7936000...	2000	Airports, Ter...	100	UNINCORPORAT...	7121	DISTICT 1	20	Unincorporated - N...	226
9	2077441	2021	REAL	[null]	2917360...	7936010...	2000	Airports, Ter...	100	UNINCORPORAT...	7121	DISTICT 1	20	Unincorporated - N...	208
10	2106352	2021	REAL	[null]	3016170...	6017000...	2000	Airports, Ter...	100	UNINCORPORAT...	7103	COMMERCIAL-...	20	Unincorporated - N...	330
11	3141288	2021	REAL	[null]	3214380...	4238012...	2000	Airports, Ter...	200	UNINCORPORAT...	7280	ORMOND BEAC...	23	Unincorporated - N...	368
12	3225082	2021	REAL	[null]	3214440...	4244013...	2000	Airports, Ter...	204	DAYTONA BEACH	7253	COMMERCIAL /...	09	Daytona Beach	60
13	3267966	2021	REAL	[null]	3215371...	5237160...	2000	Airports, Ter...	204	DAYTONA BEACH	7250	NO[illegible]	[illegible]	[illegible]	[illegible]

Successfully run. Total query runtime: 139 msec. 242 rows af

These are the 5 closest airports to the parcel=3565215



You are currently running version 4.28 of pgAdmin 4, however the current version is 4.30.

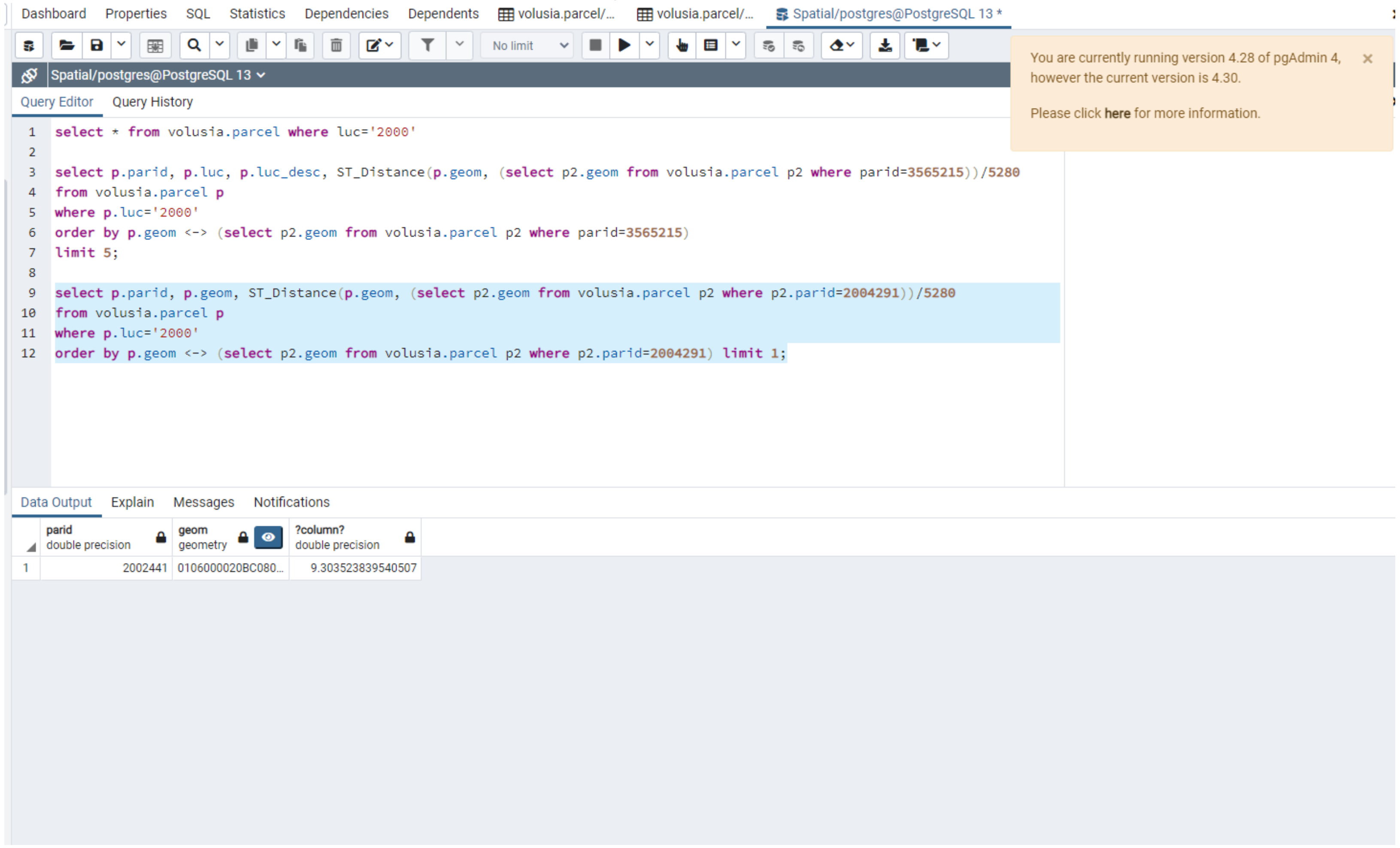
Please click **here** for more information.

```sql
select * from volusia.parcel where luc='2000'

select p.parid, p.luc, p.luc_desc, ST_Distance(p.geom, (select p2.geom from volusia.parcel p2 where parid=3565215))/5280
from volusia.parcel p
where p.luc='2000'
order by p.geom <-> (select p2.geom from volusia.parcel p2 where parid=3565215)
limit 5;
```

	parid double precision	luc text	luc_desc text	?column? double precision
1	5116301	2000	Airports, Ter...	1.8636086017306315
2	3507223	2000	Airports, Ter...	1.910907735074594
3	3515978	2000	Airports, Ter...	2.333825005708095
4	3509986	2000	Airports, Ter...	2.5297860603334095
5	3464192	2000	Airports, Ter...	2.6299242467475725

✓ Successfully run. Total query runtime: 48 msec. 5 rows affected.

Similar code, but now I find the closest airport to a parcel. We can see that the closest airport from the parcel (2004291) is parcel 2002441



Spatial/postgres@PostgreSQL 13

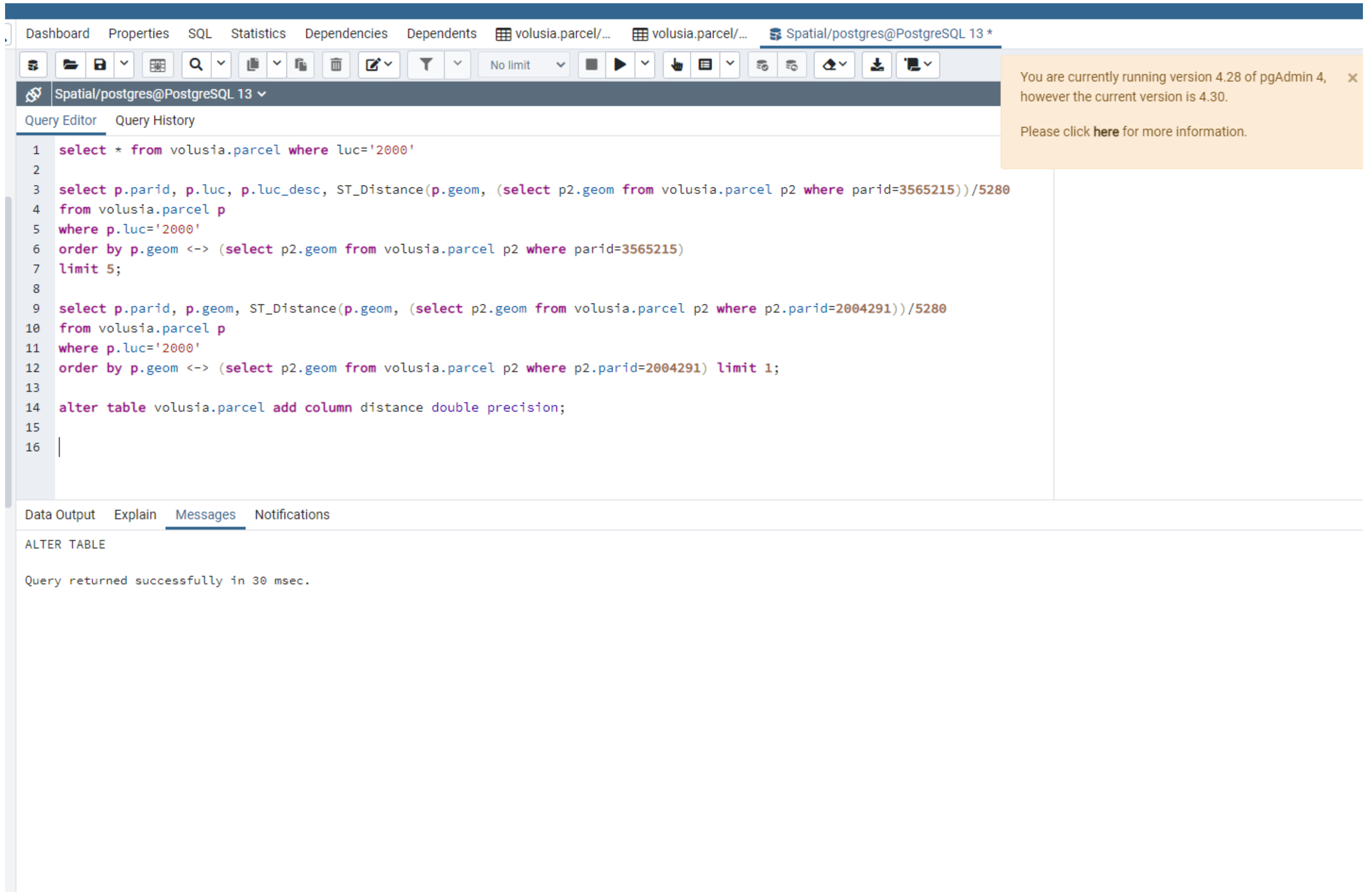
```
select * from volusia.parcel where luc='2000'

select p.parid, p.luc, p.luc_desc, ST_Distance(p.geom, (select p2.geom from volusia.parcel p2 where parid=3565215))/5280
from volusia.parcel p
where p.luc='2000'
order by p.geom <-> (select p2.geom from volusia.parcel p2 where parid=3565215)
limit 5;

select p.parid, p.geom, ST_Distance(p.geom, (select p2.geom from volusia.parcel p2 where p2.parid=2004291))/5280
from volusia.parcel p
where p.luc='2000'
order by p.geom <-> (select p2.geom from volusia.parcel p2 where p2.parid=2004291) limit 1;
```

You are currently running version 4.28 of pgAdmin 4, however the current version is 4.30.

Please click here for more information.

	parid double precision	geom geometry	?column? double precision
1	2002441	0106000020BC080...	9.303523839540507

Adding a Distance column



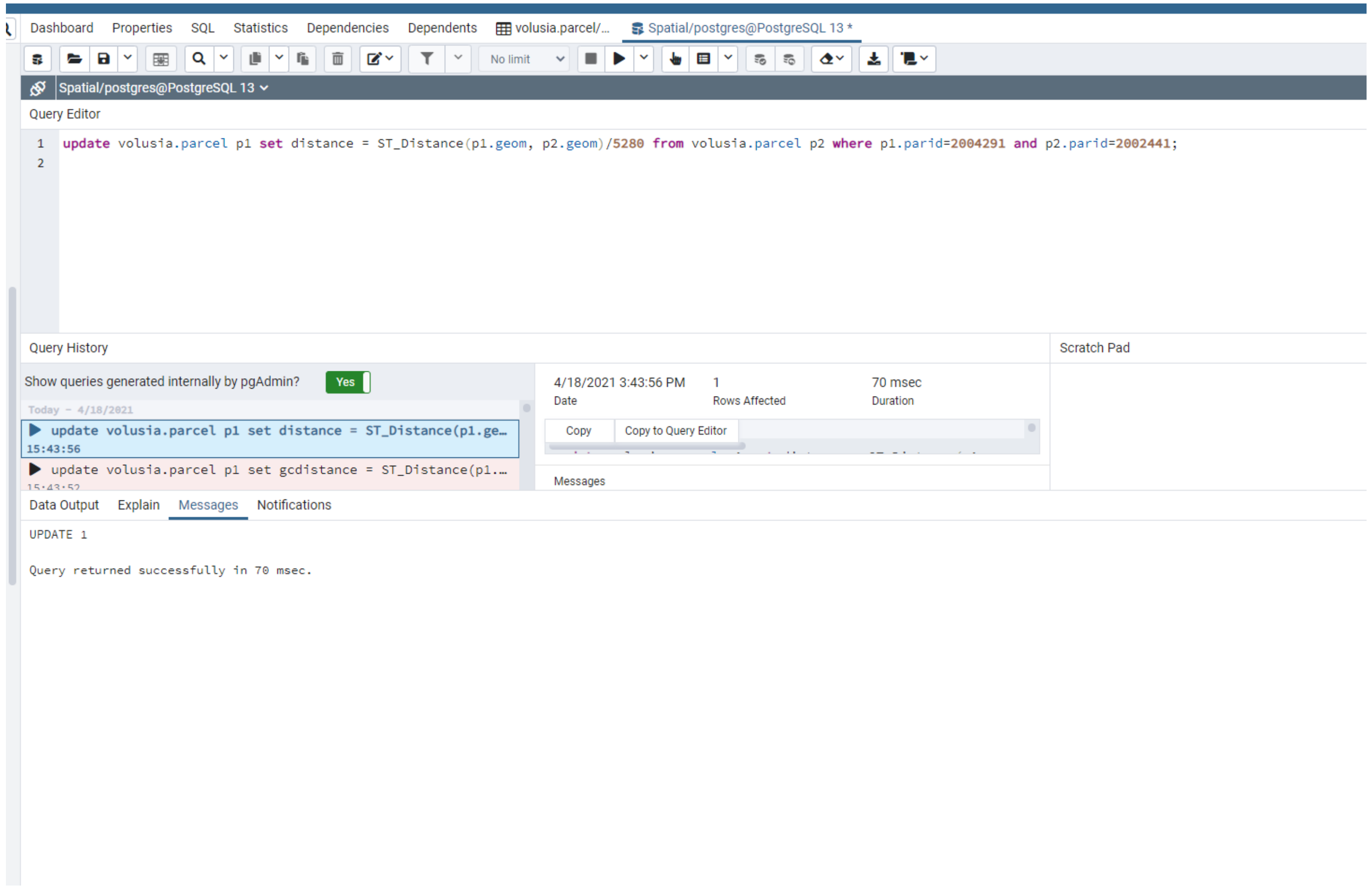
You are currently running version 4.28 of pgAdmin 4, however the current version is 4.30.

Please click **here** for more information.

```
select * from volusia.parcel where luc='2000'

select p.parid, p.luc, p.luc_desc, ST_Distance(p.geom, (select p2.geom from volusia.parcel p2 where parid=3565215))/5280
from volusia.parcel p
where p.luc='2000'
order by p.geom <-> (select p2.geom from volusia.parcel p2 where parid=3565215)
limit 5;

select p.parid, p.geom, ST_Distance(p.geom, (select p2.geom from volusia.parcel p2 where p2.parid=2004291))/5280
from volusia.parcel p
where p.luc='2000'
order by p.geom <-> (select p2.geom from volusia.parcel p2 where p2.parid=2004291) limit 1;

alter table volusia.parcel add column distance double precision;

```

```
ALTER TABLE

Query returned successfully in 30 msec.
```

We find the distance between our parcel (2004291) and the airport (2002441)



Dashboard Properties SQL Statistics Dependencies Dependents volusia.parcel/... Spatial/postgres@PostgreSQL 13 *

Spatial/postgres@PostgreSQL 13

Query Editor

```
update volusia.parcel p1 set distance = ST_Distance(p1.geom, p2.geom)/5280 from volusia.parcel p2 where p1.parid=2004291 and p2.parid=2002441;
```

Query History

Show queries generated internally by pgAdmin? Yes

Today - 4/18/2021

```
update volusia.parcel p1 set distance = ST_Distance(p1.ge…
15:43:56
update volusia.parcel p1 set gcdistance = ST_Distance(p1.…
15:43:52
```

Date	Rows Affected	Duration
4/18/2021 3:43:56 PM	1	70 msec

Copy Copy to Query Editor

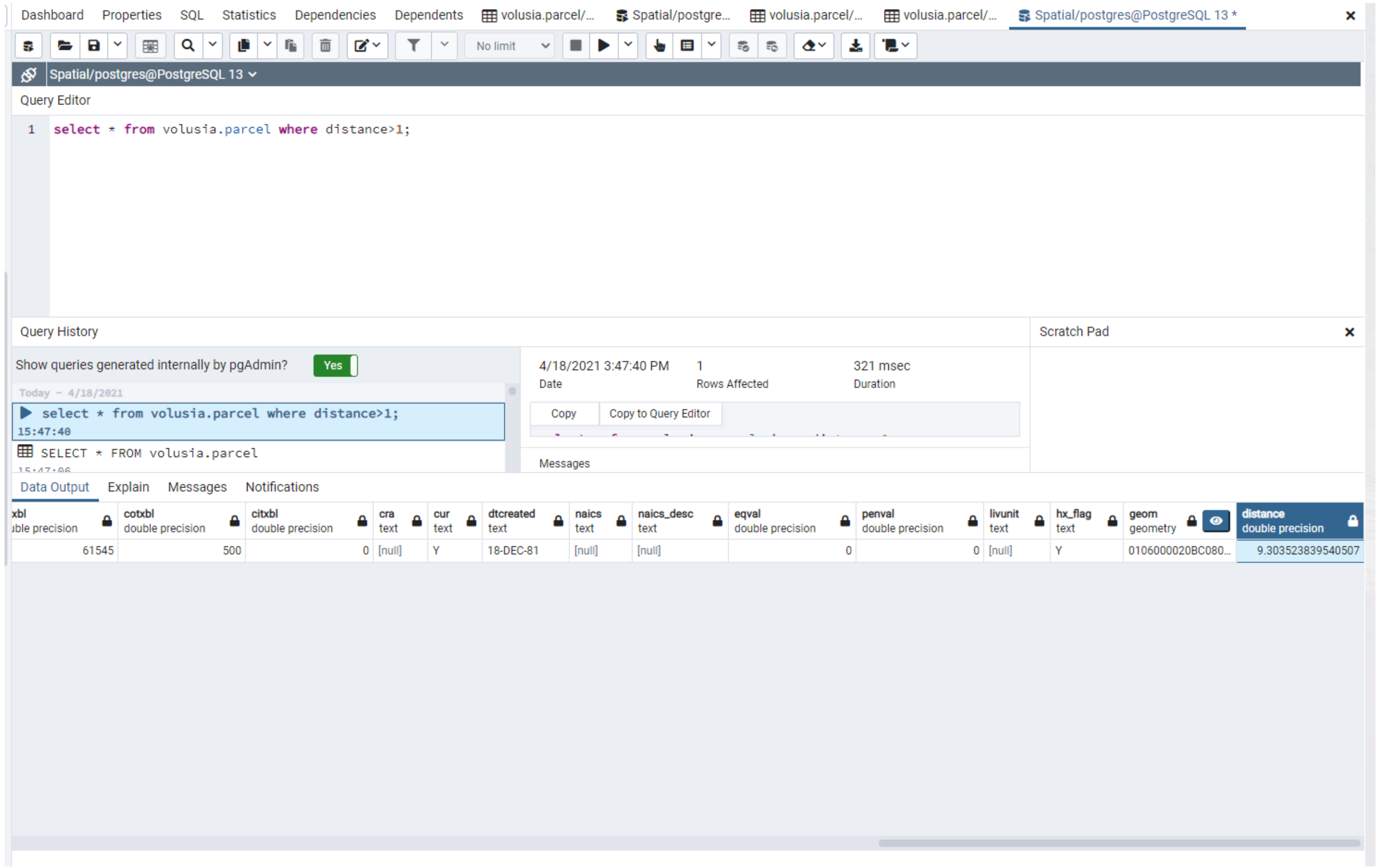
Messages

Scratch Pad

Data Output Explain Messages Notifications

```
UPDATE 1

Query returned successfully in 70 msec.
```

Distance=9.30352



```
select * from volusia.parcel where distance>1;
```

Show queries generated internally by pgAdmin? Yes

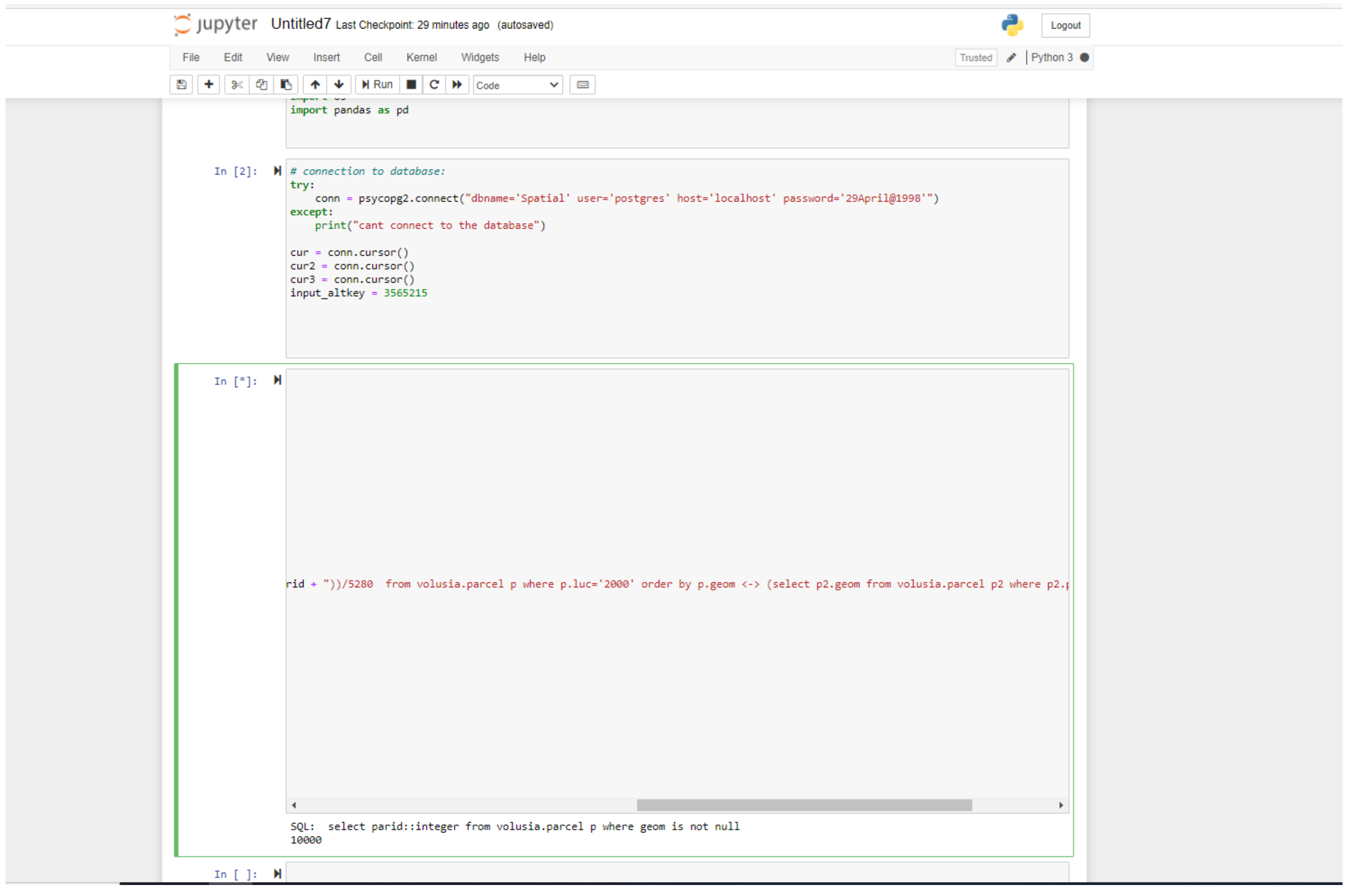
Today - 4/18/2021

```
select * from volusia.parcel where distance>1;
15:47:40
SELECT * FROM volusia.parcel
15:47:06
```

4/18/2021 3:47:40 PM	1	321 msec
Date	Rows Affected	Duration

xbl uble precision	cotxbl double precision	citxbl double precision	cra text	cur text	dtcreated text	naics text	naics_desc text	eqval double precision	penval double precision	livunit text	hx_flag text	geom geometry	distance double precision
61545	500	0	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	Y	0106000020BC080...	9.303523839540507

I ran the code for the loop.



```
import pandas as pd
```

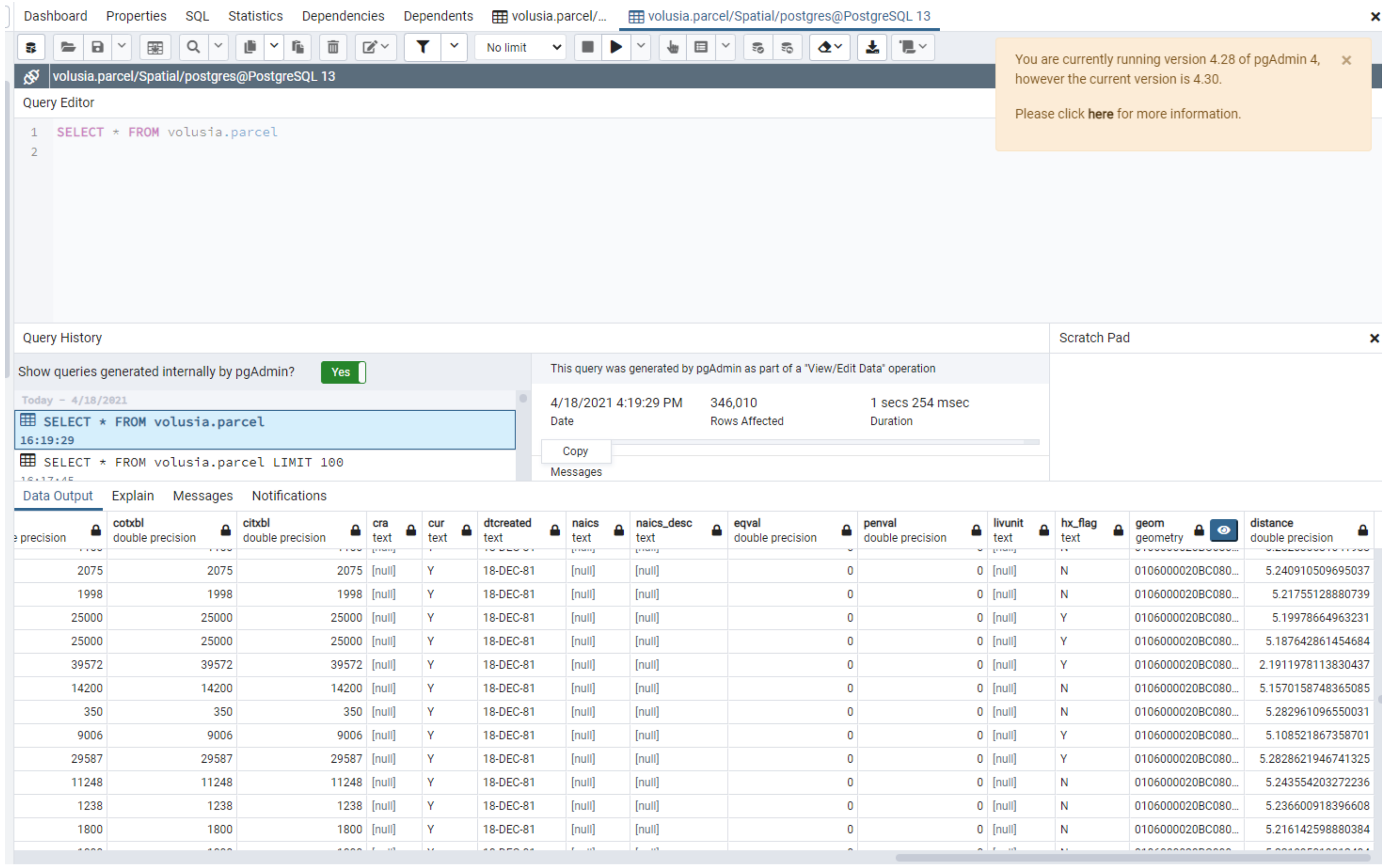
In [2]:
```
# connection to database:
try:
    conn = psycopg2.connect("dbname='Spatial' user='postgres' host='localhost' password='29April@1998'")
except:
    print("cant connect to the database")

cur = conn.cursor()
cur2 = conn.cursor()
cur3 = conn.cursor()
input_altkey = 3565215
```

In [*]:
```
rid + "))/5280  from volusia.parcel p where p.luc='2000' order by p.geom <-> (select p2.geom from volusia.parcel p2 where p2.
```

```
SQL:  select parid::integer from volusia.parcel p where geom is not null
10000
```

In [ ]:

We have the distances from every parcel to the closest airport.



Dashboard Properties SQL Statistics Dependencies Dependents volusia.parcel/... volusia.parcel/Spatial/postgres@PostgreSQL 13

volusia.parcel/Spatial/postgres@PostgreSQL 13

You are currently running version 4.28 of pgAdmin 4, however the current version is 4.30.

Please click here for more information.

Query Editor

```sql
SELECT * FROM volusia.parcel
```

Query History

Show queries generated internally by pgAdmin? Yes

Today - 4/18/2021

```
SELECT * FROM volusia.parcel
16:19:29
SELECT * FROM volusia.parcel LIMIT 100
```

This query was generated by pgAdmin as part of a "View/Edit Data" operation

Date	Rows Affected	Duration
4/18/2021 4:19:29 PM	346,010	1 secs 254 msec

Scratch Pad

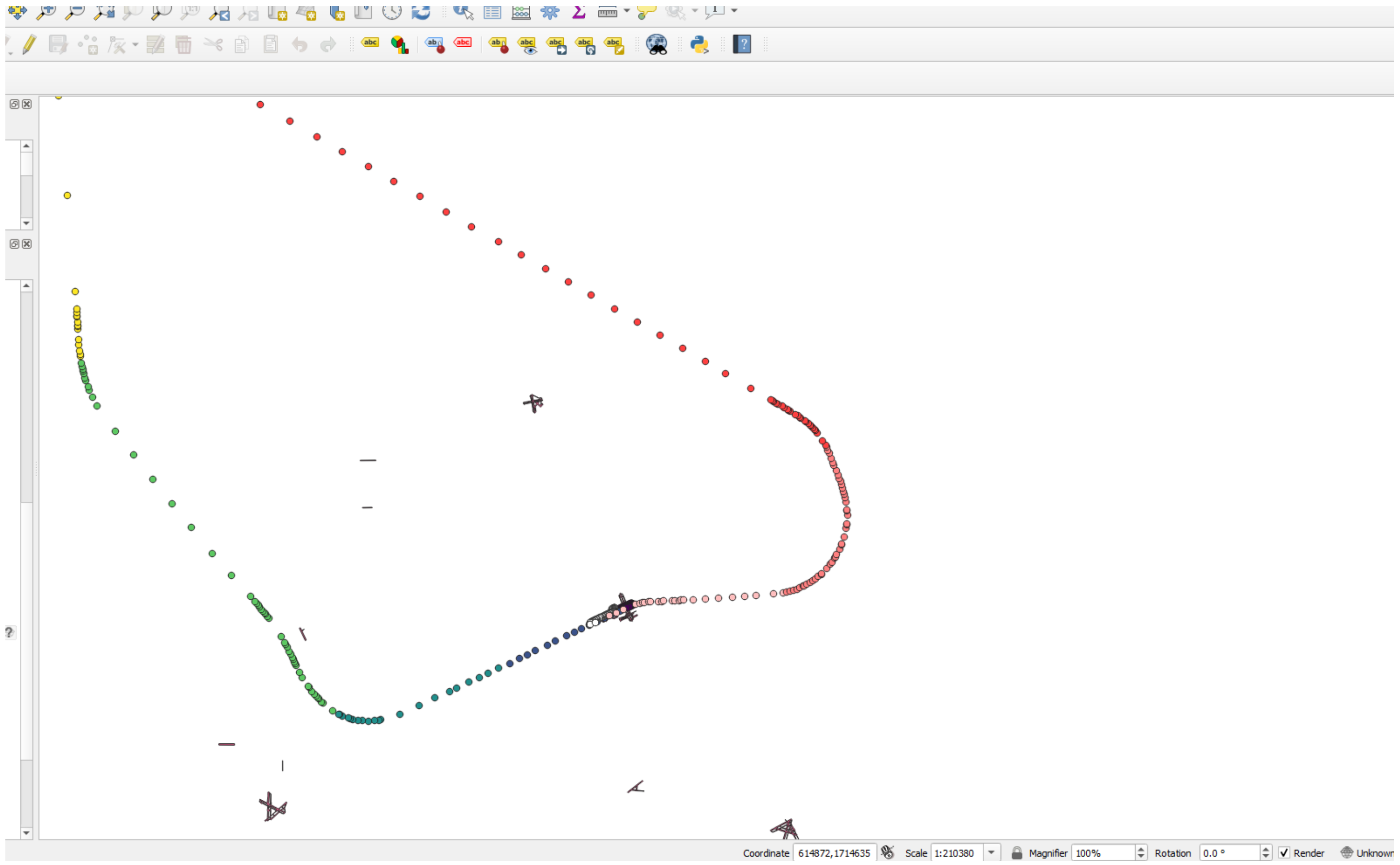
Data Output Explain Messages Notifications

… precision	cotxbl double precision	citxbl double precision	cra text	cur text	dtcreated text	naics text	naics_desc text	eqval double precision	penval double precision	livunit text	hx_flag text	geom geometry	distance double precision
2075	2075	2075	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	N	0106000020BC080...	5.240910509695037
1998	1998	1998	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	N	0106000020BC080...	5.21755128880739
25000	25000	25000	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	Y	0106000020BC080...	5.19978664963231
25000	25000	25000	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	Y	0106000020BC080...	5.187642861454684
39572	39572	39572	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	Y	0106000020BC080...	2.1911978113830437
14200	14200	14200	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	N	0106000020BC080...	5.1570158748365085
350	350	350	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	N	0106000020BC080...	5.282961096550031
9006	9006	9006	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	Y	0106000020BC080...	5.108521867358701
29587	29587	29587	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	Y	0106000020BC080...	5.2828621946741325
11248	11248	11248	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	N	0106000020BC080...	5.243554203272236
1238	1238	1238	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	N	0106000020BC080...	5.236600918396608
1800	1800	1800	[null]	Y	18-DEC-81	[null]	[null]	0	0	[null]	N	0106000020BC080...	5.216142598880384

Now I will add flight tracking information

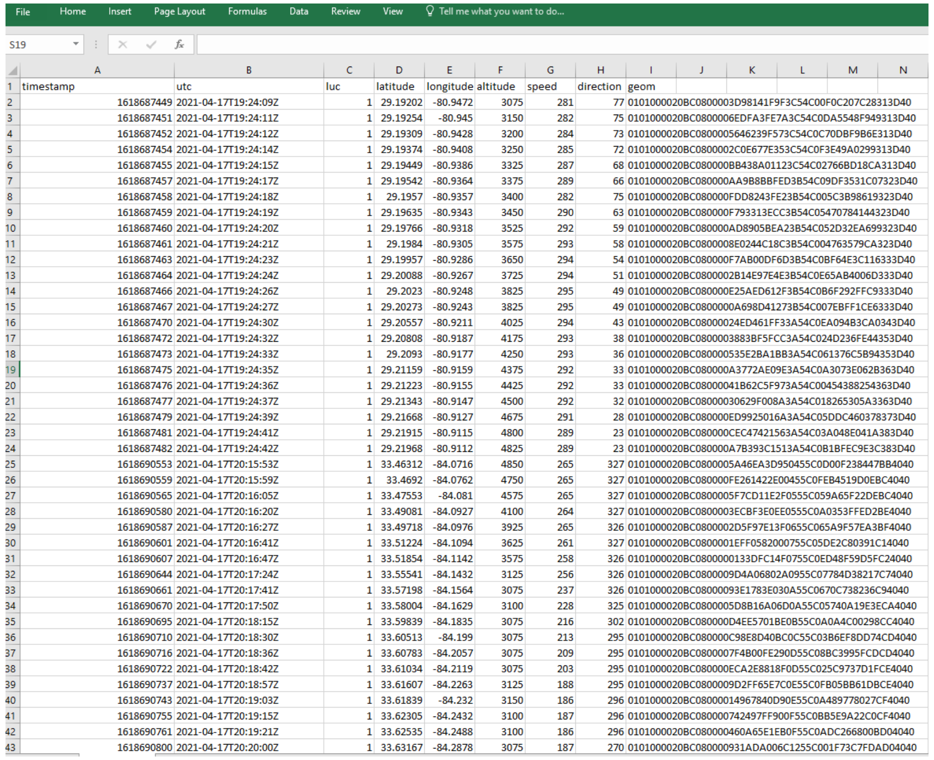
- The Daytona Beach area has a lot of flight activity, but we are only interested in commercial jets as they are the aircraft that produce the most noise.
- New Symrna, Deland, Ormond, and Massey do not have commercial flights. We will only analyze flights from Daytona.
  - Also, I was only able to retrieve flight data for Daytona Beach for free.
- Daytona Beach uses take-offs from two runways for most purposes. Runway 7L and Runway 25R. Both commercial services (American and Delta) take off to the north (Charlotte and Atlanta). So I retrieved two historical flights tracks for Delta 1701. One took off from Runway 7L and one from Runway 25R.

I wanted to visualize this on QGIS. So I added the two files as Delimited Text Layers



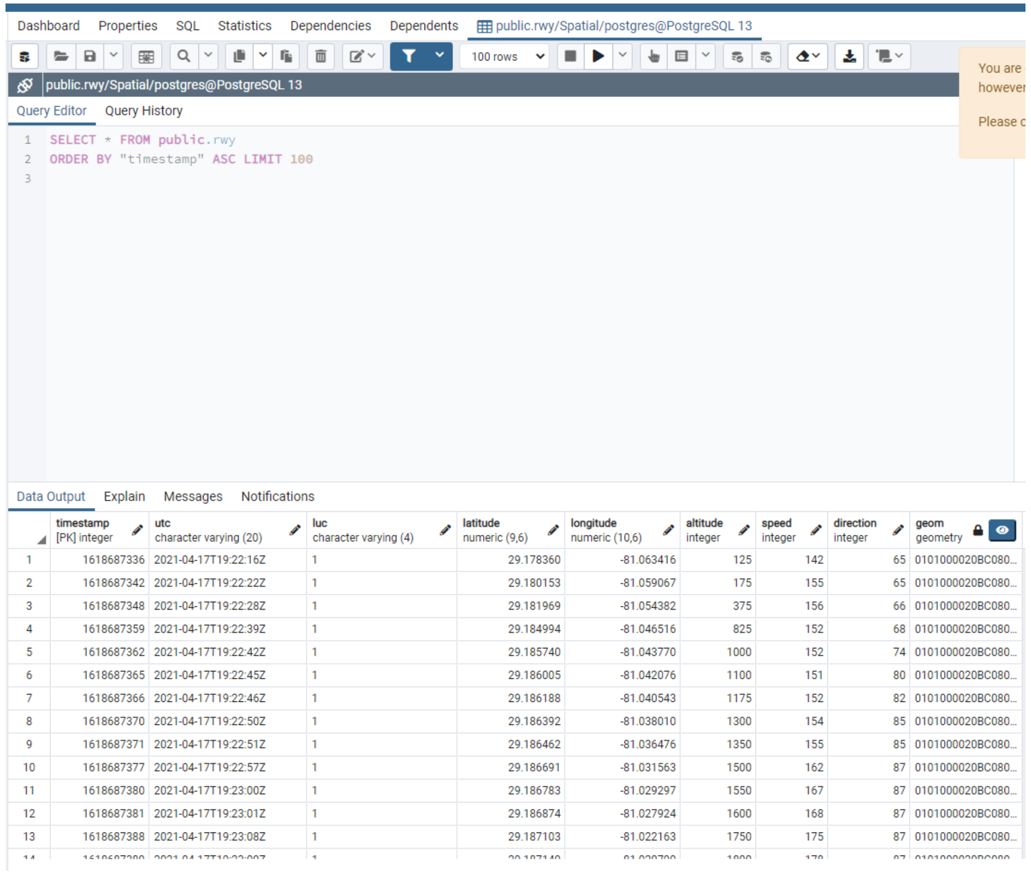
Data Preprocessing

- We combined the flight track for Rwy 7L and Rwy 25R and assigned the flight track an luc=1



timestamp	utc	luc	latitude	longitude	altitude	speed	direction	geom
1618687449	2021-04-17T19:24:09Z	1	29.19202	-80.9472	3075	281	77	0101000020BC0800003D98141F9F3C54C00F0C207C28313D40
1618687451	2021-04-17T19:24:11Z	1	29.19254	-80.945	3150	282	75	0101000020BC0800006EDFA3FE7A3C54C0DA5548F949313D40
1618687452	2021-04-17T19:24:12Z	1	29.19309	-80.9428	3200	284	73	0101000020BC0800005646239F573C54C0C70DBF9B6E313D40
1618687454	2021-04-17T19:24:14Z	1	29.19374	-80.9408	3250	285	72	0101000020BC0800002C0E677E353C54C0F3E49A0299313D40
1618687455	2021-04-17T19:24:15Z	1	29.19449	-80.9386	3325	287	68	0101000020BC080000BB438A01123C54C02766BD18CA313D40
1618687457	2021-04-17T19:24:17Z	1	29.19542	-80.9364	3375	289	66	0101000020BC080000AA9B8BBFED3B54C09DF3531C07323D40
1618687458	2021-04-17T19:24:18Z	1	29.1957	-80.9357	3400	282	75	0101000020BC080000FDD8243FE23B54C005C3B98619323D40
1618687459	2021-04-17T19:24:19Z	1	29.19635	-80.9343	3450	290	63	0101000020BC080000F793313ECC3B54C05470784144323D40
1618687460	2021-04-17T19:24:20Z	1	29.19766	-80.9318	3525	292	59	0101000020BC080000AD8905BEA23B54C052D32EA699323D40
1618687461	2021-04-17T19:24:21Z	1	29.1984	-80.9305	3575	293	58	0101000020BC0800008E0244C18C3B54C004763579CA323D40
1618687463	2021-04-17T19:24:23Z	1	29.19957	-80.9286	3650	294	54	0101000020BC080000F7AB00DF6D3B54C0BF64E3C116333D40
1618687464	2021-04-17T19:24:24Z	1	29.20088	-80.9267	3725	294	51	0101000020BC0800002B14E97E4E3B54C0E65AB4006D333D40
1618687466	2021-04-17T19:24:26Z	1	29.2023	-80.9248	3825	295	49	0101000020BC080000E25AED612F3B54C0B6F292FFC9333D40
1618687467	2021-04-17T19:24:27Z	1	29.20273	-80.9243	3825	295	49	0101000020BC0800000A698D41273B54C007EBFF1CE6333D40
1618687470	2021-04-17T19:24:30Z	1	29.20557	-80.9211	4025	294	43	0101000020BC08000024ED461FF33A54C0EA094B3CA0343D40
1618687472	2021-04-17T19:24:32Z	1	29.20808	-80.9187	4175	293	38	0101000020BC0800003883BF5FCC3A54C024D236FE44353D40
1618687473	2021-04-17T19:24:33Z	1	29.2093	-80.9177	4250	293	36	0101000020BC080000535E2BA1BB3A54C061376C5B94353D40
1618687475	2021-04-17T19:24:35Z	1	29.21159	-80.9159	4375	292	33	0101000020BC080000A3772AE09E3A54C0A3073E062B363D40
1618687476	2021-04-17T19:24:36Z	1	29.21223	-80.9155	4425	292	33	0101000020BC08000041B62C5F973A54C00454388254363D40
1618687477	2021-04-17T19:24:37Z	1	29.21343	-80.9147	4500	292	32	0101000020BC08000030629F008A3A54C018265305A3363D40
1618687479	2021-04-17T19:24:39Z	1	29.21668	-80.9127	4675	291	28	0101000020BC080000ED9925016A3A54C05DDC460378373D40
1618687481	2021-04-17T19:24:41Z	1	29.21915	-80.9115	4800	289	23	0101000020BC080000CEC47421563A54C03A048E041A383D40
1618687482	2021-04-17T19:24:42Z	1	29.21968	-80.9112	4825	289	23	0101000020BC080000A7B393C1513A54C0B1BFEC9E3C383D40
1618690553	2021-04-17T20:15:53Z	1	33.46312	-84.0716	4850	265	327	0101000020BC0800005A46EA3D950455C0D00F238447BB4040
1618690559	2021-04-17T20:15:59Z	1	33.4692	-84.0762	4750	265	327	0101000020BC080000FE261422E00455C0FEB4519D0EBC4040
1618690565	2021-04-17T20:16:05Z	1	33.47553	-84.081	4575	265	327	0101000020BC0800005F7CD11E2F0555C059A65F22DEBC4040
1618690580	2021-04-17T20:16:20Z	1	33.49081	-84.0927	4100	264	327	0101000020BC0800003ECBF3E0EE0555C0A0353FFED2BE4040
1618690587	2021-04-17T20:16:27Z	1	33.49718	-84.0976	3925	265	326	0101000020BC0800002D5F97E13F0655C065A9F57EA3BF4040
1618690601	2021-04-17T20:16:41Z	1	33.51224	-84.1094	3625	261	327	0101000020BC0800001EFF0582000755C05DE2C80391C14040
1618690607	2021-04-17T20:16:47Z	1	33.51854	-84.1142	3575	258	326	0101000020BC0800000133DFC14F0755C0ED48F59D5FC24040
1618690644	2021-04-17T20:17:24Z	1	33.55541	-84.1432	3125	256	326	0101000020BC0800009D4A06802A0955C07784D38217C74040
1618690661	2021-04-17T20:17:41Z	1	33.57198	-84.1564	3075	237	326	0101000020BC08000093E1783E030A55C0670C738236C94040
1618690670	2021-04-17T20:17:50Z	1	33.58004	-84.1629	3100	228	325	0101000020BC0800005D8B16A06D0A55C05740A19E3ECA4040
1618690695	2021-04-17T20:18:15Z	1	33.59839	-84.1835	3075	216	302	0101000020BC080000D4EE5701BE0B55C0A0A4C00298CC4040
1618690710	2021-04-17T20:18:30Z	1	33.60513	-84.199	3075	213	295	0101000020BC080000C98E8D40BC0C55C03B6EF8DD74CD4040
1618690716	2021-04-17T20:18:36Z	1	33.60783	-84.2057	3075	209	295	0101000020BC0800007F4B00FE290D55C08BC3995FCDCD4040
1618690722	2021-04-17T20:18:42Z	1	33.61034	-84.2119	3075	203	295	0101000020BC080000ECA2E8818F0D55C025C9737D1FCE4040
1618690737	2021-04-17T20:18:57Z	1	33.61607	-84.2263	3125	188	295	0101000020BC0800009D2FF65E7C0E55C0FB05BB61DBCE4040
1618690743	2021-04-17T20:19:03Z	1	33.61839	-84.232	3150	186	296	0101000020BC08000014967840D90E55C0A489778027CF4040
1618690755	2021-04-17T20:19:15Z	1	33.62305	-84.2432	3100	187	296	0101000020BC080000742497FF900F55C0BB5E9A22C0CF4040
1618690761	2021-04-17T20:19:21Z	1	33.62535	-84.2488	3100	186	296	0101000020BC080000460A65E1EB0F55C0ADC266800BD04040
1618690800	2021-04-17T20:20:00Z	1	33.63167	-84.2878	3075	187	270	0101000020BC080000931ADA006C1255C001F73C7FDAD04040

I added my csv data into a PostgreSQL table. We used a SQL query to convert the CSV to SQL. To create the Geometry Table, I used the following codes


Dashboard Properties SQL Statistics Dependencies Dependents public.rwy/Spatial/postgres@PostgreSQL 13

public.rwy/Spatial/postgres@PostgreSQL 13

You are
however

Please c

Query Editor Query History

```sql
SELECT * FROM public.rwy
ORDER BY "timestamp" ASC LIMIT 100
```

Data Output Explain Messages Notifications

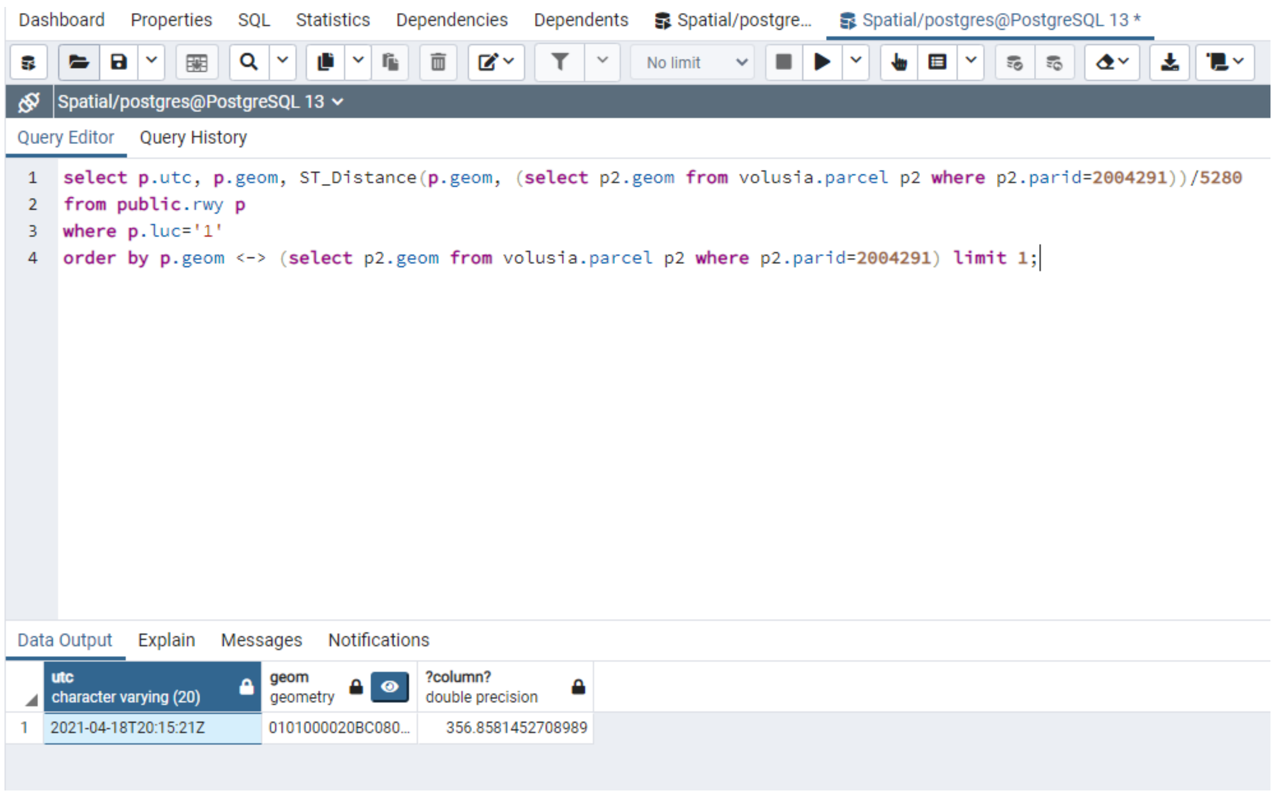
	timestamp [PK] integer	utc character varying (20)	luc character varying (4)	latitude numeric (9,6)	longitude numeric (10,6)	altitude integer	speed integer	direction integer	geom geometry
1	1618687336	2021-04-17T19:22:16Z	1	29.178360	-81.063416	125	142	65	0101000020BC080…
2	1618687342	2021-04-17T19:22:22Z	1	29.180153	-81.059067	175	155	65	0101000020BC080…
3	1618687348	2021-04-17T19:22:28Z	1	29.181969	-81.054382	375	156	66	0101000020BC080…
4	1618687359	2021-04-17T19:22:39Z	1	29.184994	-81.046516	825	152	68	0101000020BC080…
5	1618687362	2021-04-17T19:22:42Z	1	29.185740	-81.043770	1000	152	74	0101000020BC080…
6	1618687365	2021-04-17T19:22:45Z	1	29.186005	-81.042076	1100	151	80	0101000020BC080…
7	1618687366	2021-04-17T19:22:46Z	1	29.186188	-81.040543	1175	152	82	0101000020BC080…
8	1618687370	2021-04-17T19:22:50Z	1	29.186392	-81.038010	1300	154	85	0101000020BC080…
9	1618687371	2021-04-17T19:22:51Z	1	29.186462	-81.036476	1350	155	85	0101000020BC080…
10	1618687377	2021-04-17T19:22:57Z	1	29.186691	-81.031563	1500	162	87	0101000020BC080…
11	1618687380	2021-04-17T19:23:00Z	1	29.186783	-81.029297	1550	167	87	0101000020BC080…
12	1618687381	2021-04-17T19:23:01Z	1	29.186874	-81.027924	1600	168	87	0101000020BC080…
13	1618687388	2021-04-17T19:23:08Z	1	29.187103	-81.022163	1750	175	87	0101000020BC080…
14	1618687389	2021-04-17T19:23:09Z	1	29.187140	-81.020700	1800	178	87	0101000020BC080

- I will now try to find the distance of each parcel in the parcel table to the flight track.

- I will use the same code I used for the runway. The runways had a luc of 2000. I have assigned an LUC of 0001 to the flights tracks.

Finding the closest flight tracking point to a random parcel.

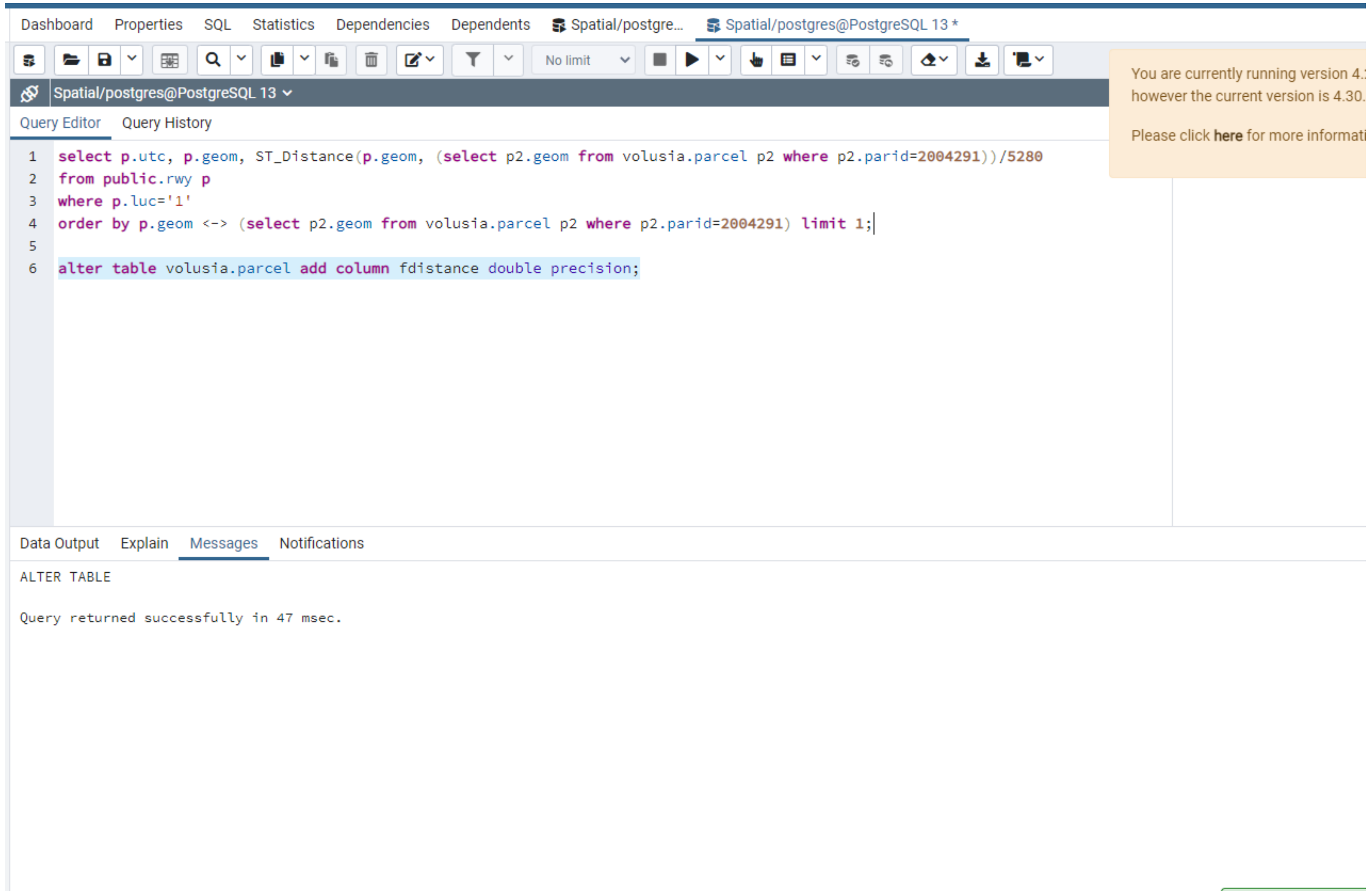
- I have used UTC as an identifier for the flight track information.



```
select p.utc, p.geom, ST_Distance(p.geom, (select p2.geom from volusia.parcel p2 where p2.parid=2004291))/5280
from public.rwy p
where p.luc='1'
order by p.geom <-> (select p2.geom from volusia.parcel p2 where p2.parid=2004291) limit 1;
```

	utc character varying (20)	geom geometry	?column? double precision
1	2021-04-18T20:15:21Z	0101000020BC080...	356.8581452708989

Add column for distance from Flight Track



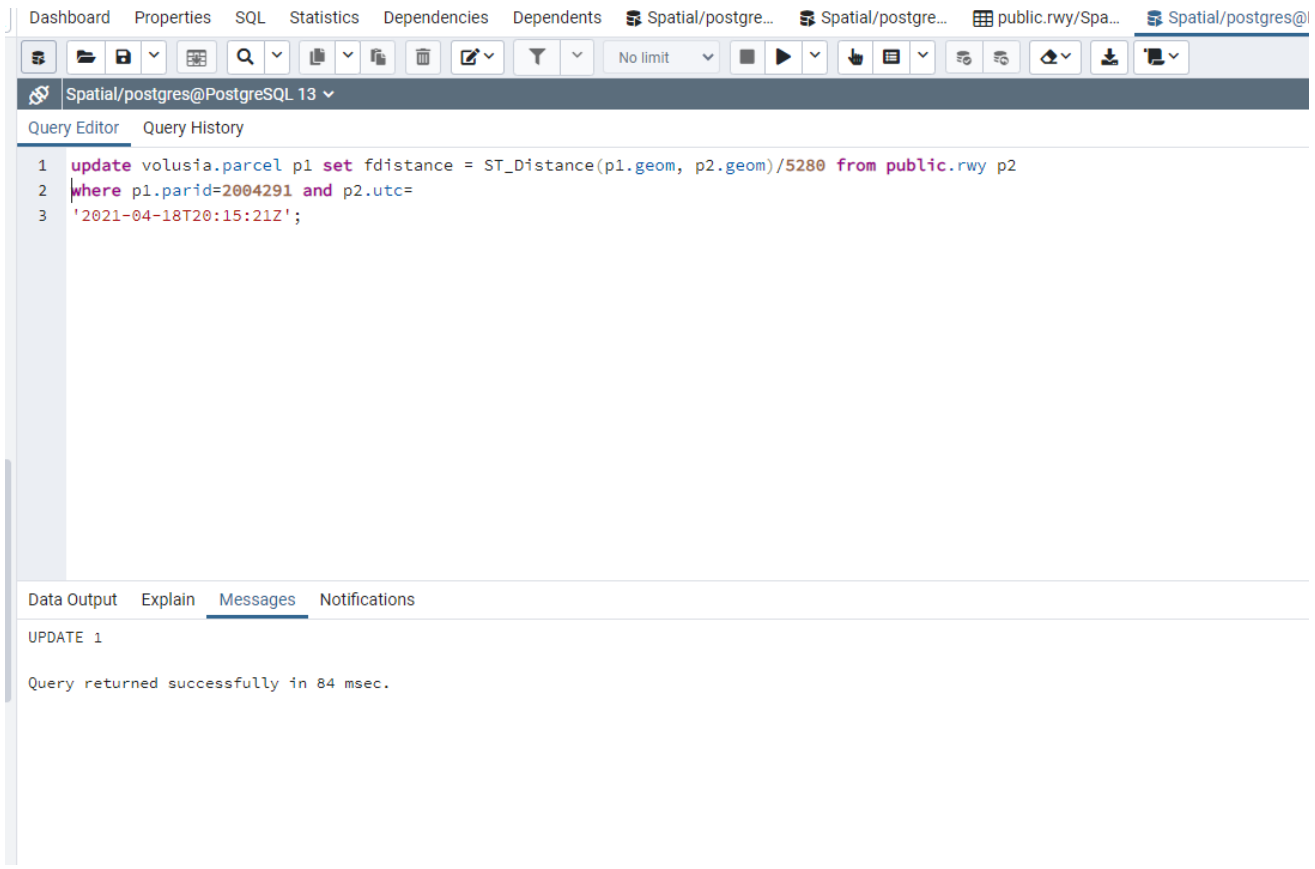
You are currently running version 4.
however the current version is 4.30.

Please click **here** for more informati

```
select p.utc, p.geom, ST_Distance(p.geom, (select p2.geom from volusia.parcel p2 where p2.parid=2004291))/5280
from public.rwy p
where p.luc='1'
order by p.geom <-> (select p2.geom from volusia.parcel p2 where p2.parid=2004291) limit 1;

alter table volusia.parcel add column fdistance double precision;
```

```
ALTER TABLE

Query returned successfully in 47 msec.
```

Find Distance Between parcel and Flight Track

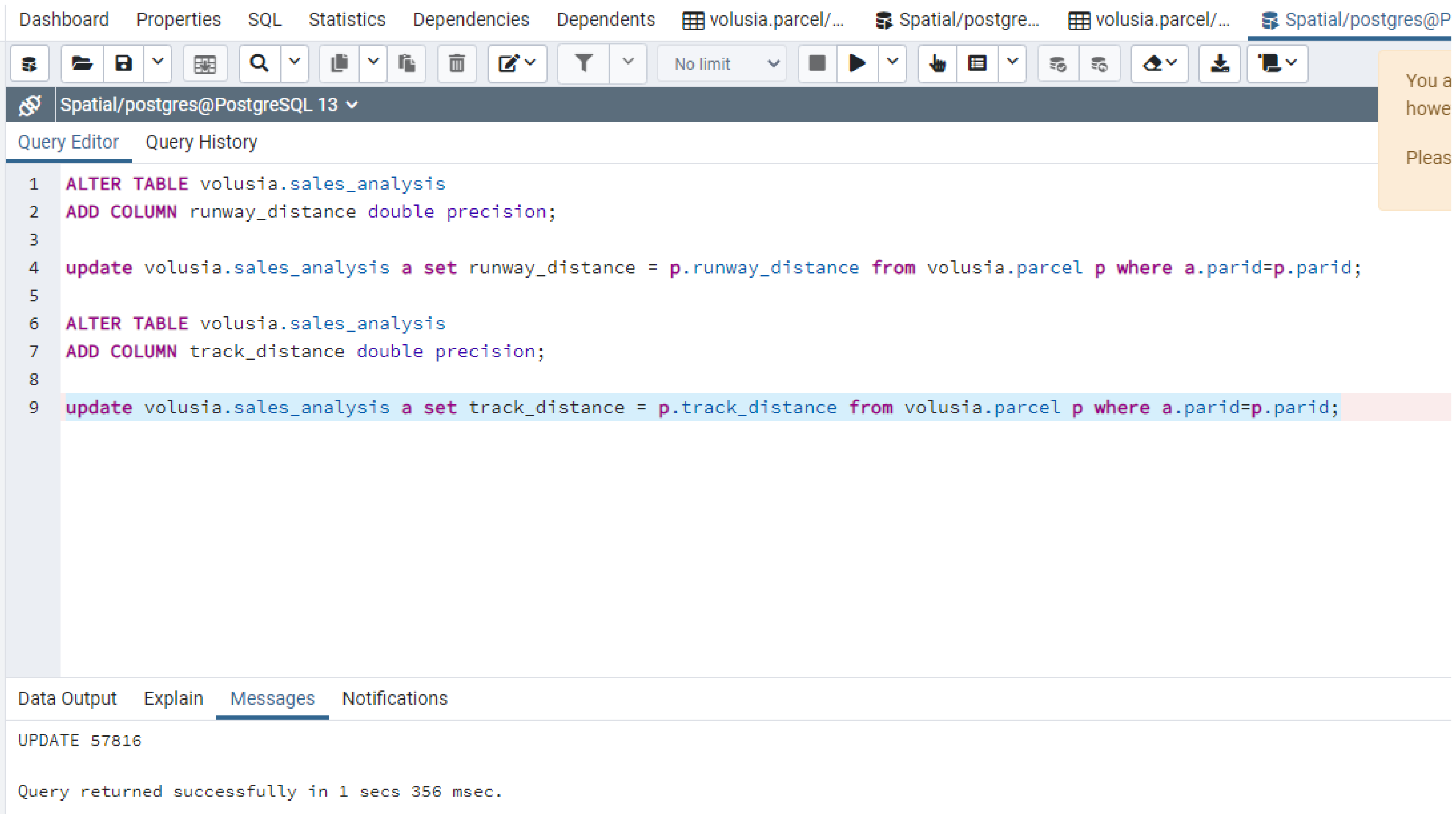


```sql
update volusia.parcel p1 set fdistance = ST_Distance(p1.geom, p2.geom)/5280 from public.rwy p2
where p1.parid=2004291 and p2.utc=
'2021-04-18T20:15:21Z';
```

```
UPDATE 1

Query returned successfully in 84 msec.
```

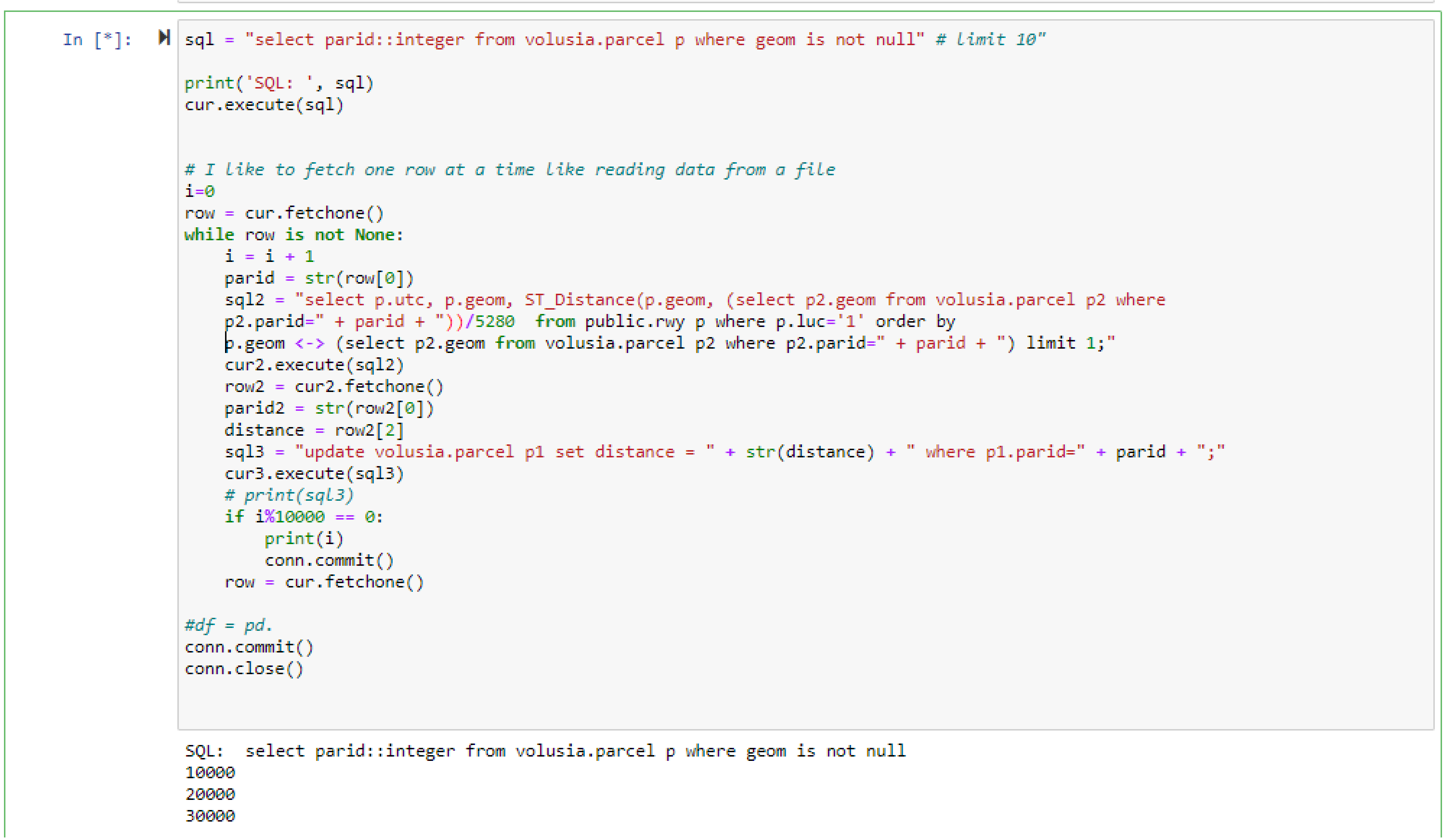
- I will now move on to the python code and adjust it accordingly to populate the remaining values of fdistance

Add runway_distance and track_distance to the sales analysis table.



```sql
ALTER TABLE volusia.sales_analysis
ADD COLUMN runway_distance double precision;

update volusia.sales_analysis a set runway_distance = p.runway_distance from volusia.parcel p where a.parid=p.parid;

ALTER TABLE volusia.sales_analysis
ADD COLUMN track_distance double precision;

update volusia.sales_analysis a set track_distance = p.track_distance from volusia.parcel p where a.parid=p.parid;
```

```
UPDATE 57816

Query returned successfully in 1 secs 356 msec.
```



In [*]:

```
sql = "select parid::integer from volusia.parcel p where geom is not null" # limit 10"

print('SQL: ', sql)
cur.execute(sql)


# I like to fetch one row at a time like reading data from a file
i=0
row = cur.fetchone()
while row is not None:
    i = i + 1
    parid = str(row[0])
    sql2 = "select p.utc, p.geom, ST_Distance(p.geom, (select p2.geom from volusia.parcel p2 where
    p2.parid=" + parid + "))/5280  from public.rwy p where p.luc='1' order by
    p.geom <-> (select p2.geom from volusia.parcel p2 where p2.parid=" + parid + ") limit 1;"
    cur2.execute(sql2)
    row2 = cur2.fetchone()
    parid2 = str(row2[0])
    distance = row2[2]
    sql3 = "update volusia.parcel p1 set distance = " + str(distance) + " where p1.parid=" + parid + ";"
    cur3.execute(sql3)
    # print(sql3)
    if i%10000 == 0:
        print(i)
        conn.commit()
    row = cur.fetchone()

#df = pd.
conn.commit()
conn.close()
```

```
SQL:  select parid::integer from volusia.parcel p where geom is not null
10000
20000
30000
```