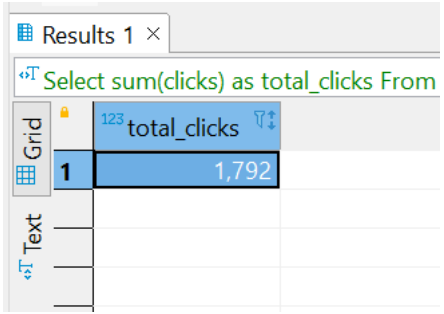
PMG SQL Test

#Question 1

```
Select sum(clicks) as total_clicks
From marketing_data
```



#Question 2

```sql
select store_location, sum(revenue) as revenue
from store_revenue
group by store_location
```

	store_location	revenue
1	United States-CA	235,237
2	United States-TX	9,629
3	United States-NY	51,984

#Question 3

```sql
with cte as(
select *,right(store_location,2) as bridge
from store_revenue)

select geo,cte.date, avg(impressions) as impressions, avg(clicks) as clicks, sum(revenue) as total_revenue
from cte left join marketing_data md on cte.bridge=md.geo and cte.date=md.date
where cte.date is not null and geo is not null
group by geo,cte.date
```

Grid | Text | Record

	geo	date	impressions	clicks	total_revenue
1	CA	2016-01-01	3,425	63	334
2	CA	2016-01-02	1,354	53	465
3	CA	2016-01-03	5,258	36	234,334
4	CA	2016-01-04	7,854	85	36
5	CA	2016-01-05	4,678	73	68
6	NY	2016-01-01	3,532	25	284
7	NY	2016-01-02	4,643	85	2,574
8	NY	2016-01-03	4,735	63	3,479
9	NY	2016-01-04	4,754	36	45,289
10	NY	2016-01-05	2,364	33	358
11	TX	2016-01-01	2,532	45	654
12	TX	2016-01-02	3,643	23	5,765
13	TX	2016-01-03	2,353	57	423
14	TX	2016-01-04	5,783	47	2,357
15	TX	2016-01-05	2,535	63	427

#Question 4

- Since there is no background description, I assume we are analyzing marketing data for E-commerce. In addition, there is no cost of impressions or clicks, so I assume the cost per click and cost per impression is the same for different states, or we are marketing on social media like Instagram at no expense.

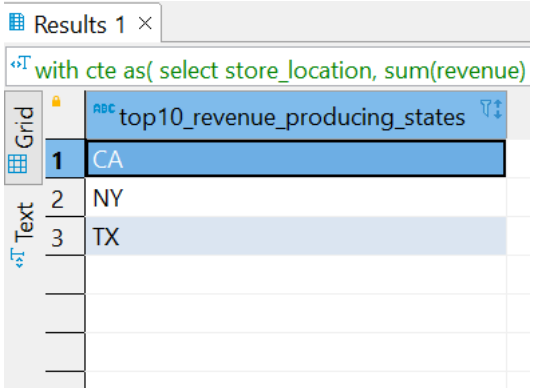
- With these assumptions, I can use revenue per click and revenue per impression as metrics for calculating the efficiency of each store. I get revenue per impression for the store in CA is 10.42, which is much higher than in NY(2.59) and TX(0.57). And revenue per click for the store in CA is 758, which is also much higher than in NY(214) and TX(40). In conclusion, the store in CA is the most efficient.

```
with cte as(
select *,right(store_location,2) as bridge
from store_revenue),
cte2 as(
select geo,cte.date, avg(impressions) as impressions, avg(clicks) as clicks, sum(revenue) as total_revenue
from cte left join marketing_data md on cte.bridge=md.geo and cte.date=md.date
where cte.date is not null and geo is not null
group by geo,cte.date
)
select geo,sum(total_revenue)/sum(impressions) as revenue_per_impression, sum(total_revenue)/sum(clicks) as revenue_per_clicks
from cte2
group by geo
```

	geo	revenue_per_impression	revenue_per_clicks
1	CA	10.4230138686	758.8290322581
2	NY	2.5955662073	214.8099173554
3	TX	0.5714116111	40.9617021277

#Question 5

```sql
with cte as(
select store_location, sum(revenue) as total_revenue
from store_revenue
group by store_location
order by total_revenue desc
limit 10
)
select right(store_location,2) as Top10_revenue_producing_states
from cte
```



Appendix

Upload table into database

Use Python to link Postgre SQL

jupyter Untitled Last Checkpoint: an hour ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [1]:
```
!pip install psycopg2-binary
```

```
Collecting psycopg2-binary
  Downloading psycopg2_binary-2.9.3-cp39-cp39-win_amd64.whl (1.2 MB)
Installing collected packages: psycopg2-binary
Successfully installed psycopg2-binary-2.9.3
```

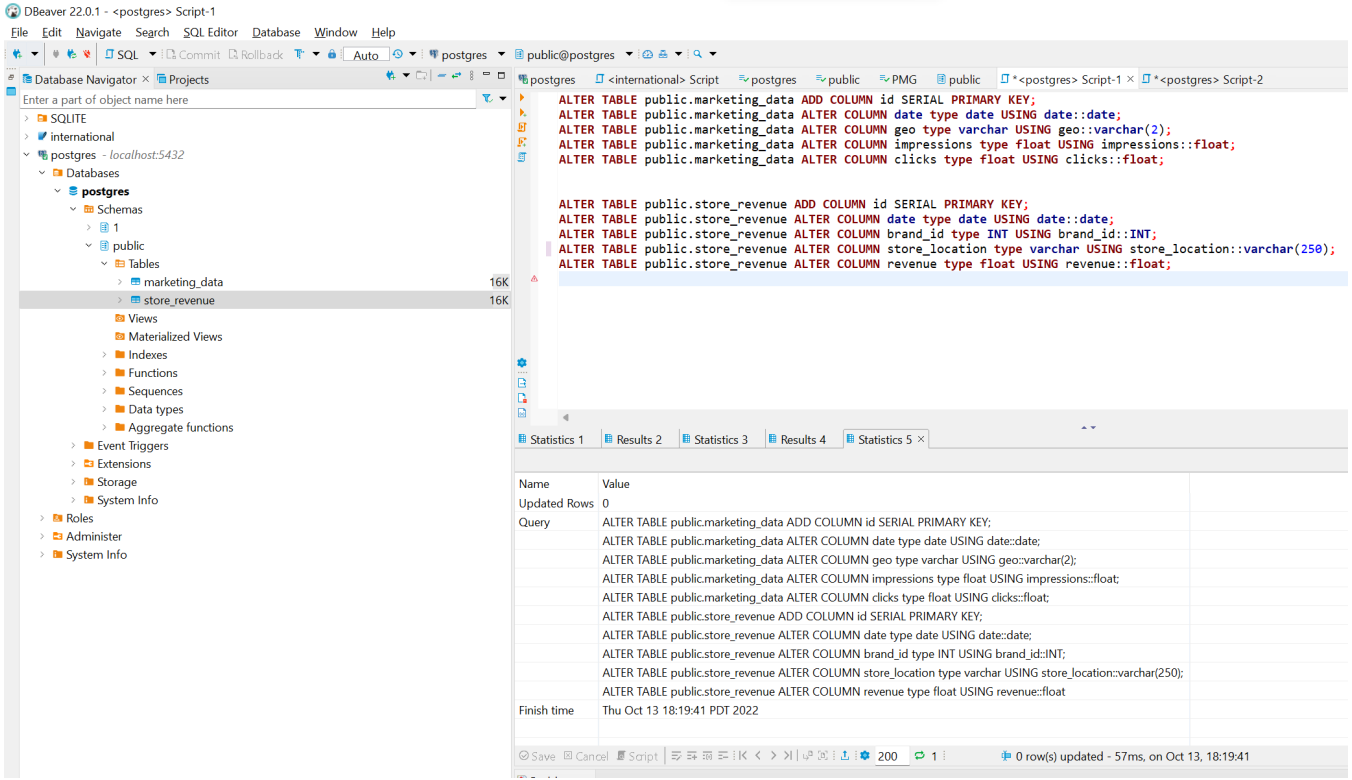
In [1]:
```
import pandas as pd
from sqlalchemy import create_engine
engine=create_engine('postgresql://postgres:woshizhu@localhost:5432/postgres')
```

In [2]:
```
application_1=pd.read_csv(r'C:\Users\hebuw\OneDrive\PMG\store_revenue.csv')
application_2=pd.read_csv(r'C:\Users\hebuw\OneDrive\PMG\marketing_data.csv')
```

In [3]:
```
application_1.to_sql('store_revenue',engine,index=False)
application_2.to_sql('marketing_data',engine,index=False)
```

Upload table into database

Convert the Data Type



Upload table into database

Overview the data

marketing_data | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text

	date	geo	impressions	clicks	id
1	2016-01-01	TX	2,532	45	1
2	2016-01-01	CA	3,425	63	2
3	2016-01-01	NY	3,532	25	3
4	2016-01-01	MN	1,342	784	4
5	2016-01-02	TX	3,643	23	5
6	2016-01-02	CA	1,354	53	6
7	2016-01-02	NY	4,643	85	7
8	2016-01-02	MN	2,366	85	8
9	2016-01-03	TX	2,353	57	9
10	2016-01-03	CA	5,258	36	10
11	2016-01-03	NY	4,735	63	11
12	2016-01-03	MN	5,783	87	12
13	2016-01-04	TX	5,783	47	13
14	2016-01-04	CA	7,854	85	14
15	2016-01-04	NY	4,754	36	15
16	2016-01-04	MN	9,345	24	16
17	2016-01-05	TX	2,535	63	17
18	2016-01-05	CA	4,678	73	18
19	2016-01-05	NY	2,364	33	19
20	2016-01-05	MN	3,452	25	20

store_revenue | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text | Record

	date	brand_id	store_location	revenue	id
1	2016-01-01	1	United States-CA	100	1
2	2016-01-01	1	United States-TX	420	2
3	2016-01-01	1	United States-NY	142	3
4	2016-01-02	1	United States-CA	231	4
5	2016-01-02	1	United States-TX	2,342	5
6	2016-01-02	1	United States-NY	232	6
7	2016-01-03	1	United States-CA	100	7
8	2016-01-03	1	United States-TX	420	8
9	2016-01-03	1	United States-NY	3,245	9
10	2016-01-04	1	United States-CA	34	10
11	2016-01-04	1	United States-TX	3	11
12	2016-01-04	1	United States-NY	54	12
13	2016-01-05	1	United States-CA	45	13
14	2016-01-05	1	United States-TX	423	14
15	2016-01-05	1	United States-NY	234	15
16	2016-01-01	2	United States-CA	234	16
17	2016-01-01	2	United States-TX	234	17
18	2016-01-01	2	United States-NY	142	18
19	2016-01-02	2	United States-CA	234	19
20	2016-01-02	2	United States-TX	3,423	20
21	2016-01-02	2	United States-NY	2,342	21
22	2016-01-03	2	United States-CA	234,234	22
23	2016-01-06	3	United States-TX	3	23
24	2016-01-03	2	United States-TX	3	24
25	2016-01-03	2	United States-NY	234	25
26	2016-01-04	2	United States-CA	2	26
27	2016-01-04	2	United States-TX	2,354	27
28	2016-01-04	2	United States-NY	45,235	28
29	2016-01-05	2	United States-CA	23	29
30	2016-01-05	2	United States-TX	4	30
31	2016-01-05	2	United States-NY	124	31