DSO552 Project

**Group2: Teresa Chen, Eason (Yu) Tsao, Bella Kuo, Yuan Wang,
Yoav Gillath, Ethan Liu, Calvin Huang**

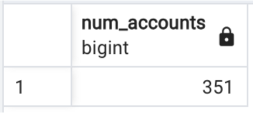
After a bumpy ride in a time machine, you suddenly land somewhere in January 2017. You find yourself in Dunder Mifflin, a big paper company, where you are working as a data analyst. The company is considering to acquire Parch and Posey, a smaller paper company. Your manager, Michael Scott, asked you to run an exploratory data analysis (aka EDA), to help educate the leadership team before they make their final decision. He gave you a list of questions that you need to look into while exploring the data.

Question 1-4

1. How big is the customer base of Parch and Posey (i.e. how many customers / accounts does the company have?) (1 point)
   Query:

   ```
   SELECT COUNT(*) AS num_accounts FROM accounts
   ```

   Output:

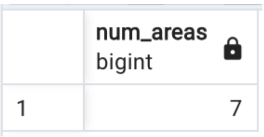
   

	num_accounts bigint
1	351

2. How many areas do they sell at? (1 point)
   Query:

   ```
   SELECT COUNT(DISTINCT id) AS num_areas FROM region
   ```

   Output:

	num_areas bigint
1	7

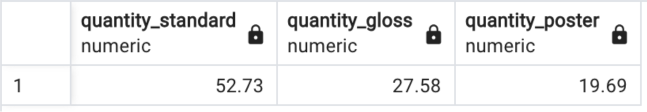
3. Look into the revenue streams:
   a. How many types of paper do they sell and what percentage each one of them makes out of the total quantity sold? Provide a visualization that illustrates the results (e.g. pie chart, bar plot, or any chart of your choice) (1.5 point)

      3 types of paper: standard, gloss, poster

      Query:

      ```
      SELECT
         ROUND(1.0*SUM(standard_qty)/SUM(total)*100,2) AS
      quantity_standard,
         ROUND(1.0*SUM(gloss_qty)/SUM(total)*100,2) AS quantity_gloss,
         ROUND(1.0*SUM(poster_qty)/SUM(total)*100,2) AS quantity_poster
      FROM orders;
      ```

Output:



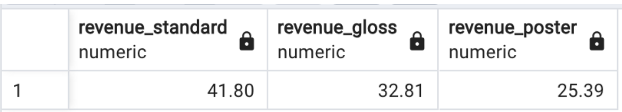
	quantity_standard numeric	quantity_gloss numeric	quantity_poster numeric
1	52.73	27.58	19.69

b. What percentage of revenues comes from which type of paper? Provide a visualization that illustrates the results (e.g. pie chart, bar plot, or any chart of your choice) (1.5 point)

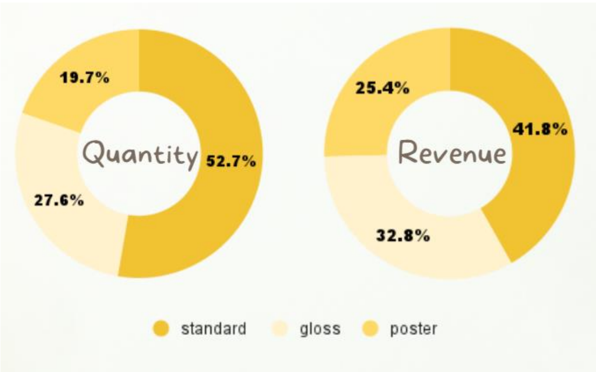
Query:

```
SELECT
   ROUND(SUM(standard_amt_usd)/SUM(total_amt_usd)*100,2) AS
revenue_standard,
   ROUND(SUM(gloss_amt_usd)/SUM(total_amt_usd)*100,2) AS
revenue_gloss,
   ROUND(SUM(poster_amt_usd)/SUM(total_amt_usd)*100,2) AS
revenue_poster
FROM orders;
```

Output:



	revenue_standard numeric	revenue_gloss numeric	revenue_poster numeric
1	41.80	32.81	25.39



4. Is the business growing?

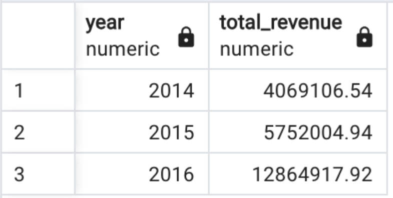
a. How have revenues been year over year? For this, only take into account years with full data (2017 just started, so we don't know how yearly revenues will be and 2013 seems to have dataY only from December). Provide a visualization that illustrates the results (e.g. line chart, bar plot, or any chart of your choice). (1.5 point)

Query:

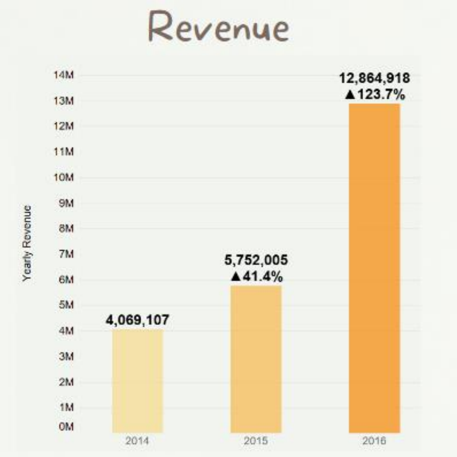
```
SELECT
        EXTRACT(YEAR FROM occurred_at) AS year,
        SUM(total_amt_usd) AS total_revenue
```

```
FROM orders
WHERE EXTRACT(YEAR FROM occurred_at) NOT IN ('2013', '2017')
GROUP BY year
ORDER BY year;
```

Output:



	year numeric	total_revenue numeric
1	2014	4069106.54
2	2015	5752004.94
3	2016	12864917.92

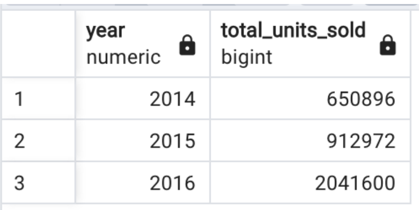


b. How have units sold evolved year over year? Here too, only take into account the past years' data. Provide a visualization that illustrates the results (e.g. line chart, bar plot, or any chart of your choice) (1.5 point)

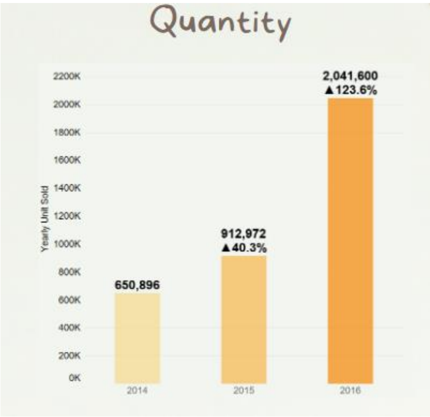
Query:

```
SELECT
        EXTRACT(YEAR FROM occurred_at) AS year,
        SUM(total) AS total_units_sold
FROM orders
WHERE EXTRACT(YEAR FROM occurred_at) NOT IN ('2013', '2017')
GROUP BY year
ORDER BY year;
```

Output:



	year numeric	total_units_sold bigint
1	2014	650896
2	2015	912972
3	2016	2041600

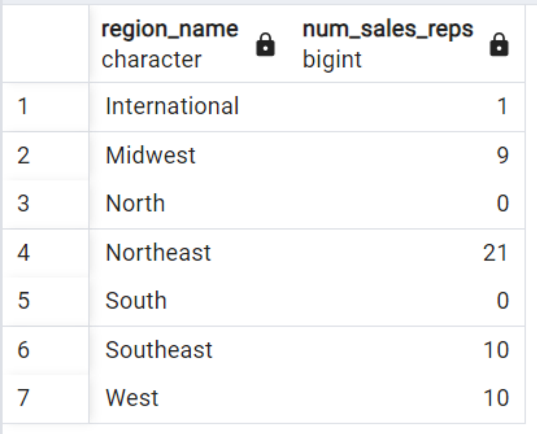


5. How many sales reps do they have in each region? Sort the result by alphabetical order and include the regions that do not have any sales reps

Query:

```
        SELECT r.name AS region_name, COUNT(sr.id) AS num_sales_reps FROM region r
        LEFT JOIN sales_reps sr
        ON r.id=sr.region_id
        GROUP BY r.name
        ORDER BY r.name;
```

Output:



	region_name character	num_sales_reps bigint
1	International	1
2	Midwest	9
3	North	0
4	Northeast	21
5	South	0
6	Southeast	10
7	West	10

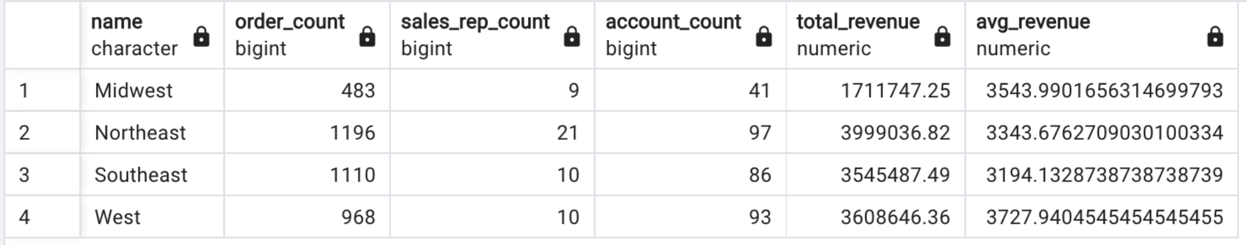
Question 6

6a

Query:

```
SELECT
        r.name,
        COUNT(o.id) order_count,
        COUNT(DISTINCT sr.id) sales_rep_count,
        COUNT(DISTINCT a.id) account_count,
        SUM(total_amt_usd) total_revenue,
        AVG(total_amt_usd) avg_revenue
FROM region r
JOIN sales_reps sr
ON r.id = sr.region_id
JOIN accounts a
ON a.sales_rep_id = sr.id
JOIN orders o
ON o.account_id = a.id
WHERE EXTRACT(YEAR FROM occurred_at) = 2016
GROUP BY r.name
```

Output:



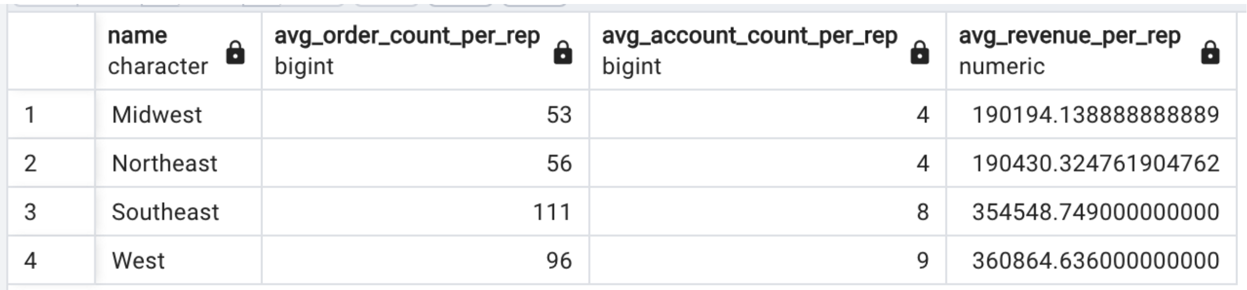
	name character	order_count bigint	sales_rep_count bigint	account_count bigint	total_revenue numeric	avg_revenue numeric
1	Midwest	483	9	41	1711747.25	3543.9901656314699793
2	Northeast	1196	21	97	3999036.82	3343.6762709030100334
3	Southeast	1110	10	86	3545487.49	3194.1328738738738739
4	West	968	10	93	3608646.36	3727.9404545454545455

6b:

Query:

```
SELECT
        r.name,
        COUNT(o.id)/COUNT(DISTINCT sr.id) avg_order_count_per_rep,
        COUNT(DISTINCT a.id)/COUNT(DISTINCT sr.id) avg_account_count_per_rep,
        SUM(total_amt_usd)/COUNT(DISTINCT sr.id) avg_revenue_per_rep
FROM region r
JOIN sales_reps sr
ON r.id = sr.region_id
JOIN accounts a
ON a.sales_rep_id = sr.id
JOIN orders o
ON o.account_id = a.id
WHERE EXTRACT(YEAR FROM occurred_at) = 2016
GROUP BY r.name
```

Output:

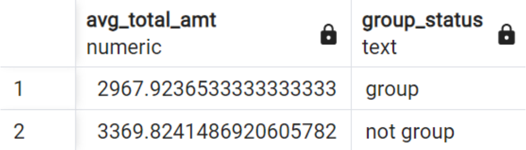


	name character	avg_order_count_per_rep bigint	avg_account_count_per_rep bigint	avg_revenue_per_rep numeric
1	Midwest	53	4	190194.138888888889
2	Northeast	56	4	190430.324761904762
3	Southeast	111	8	354548.749000000000
4	West	96	9	360864.636000000000

6c:
Response: Based on the finding, Midwest and Northeast both have lower average number of orders per sales representative and average number of accounts per sales representative. This suggests that the sales representatives in these two regions are not at their full capacity compared to their colleagues in Southeast and West. Thus, we suggest that some sales representatives from Midwest and Northeast can be relocated to our new areas without negatively impacting our current business.

Question 7:
Response: Our assumption was incorrect, as the accounts without the word group at the end of their name averaged more average total revenue than the accounts with the word group at the end. The prior group averaged about $3370 in total_amt_usd, while the latter group averaged about $2968. As such, group accounts should not receive more attention from the company after acquisition.

```
select avg(o.total_amt_usd) as avg_total_amt,
case
   when a.name ilike '%Group' then 'group'
   else 'not group'
end group_status
from accounts a
join orders o on a.id = o.account_id
group by case
          when a.name ilike '%Group' then 'group'
          else 'not group'
       end
having count(o.id)>0
```

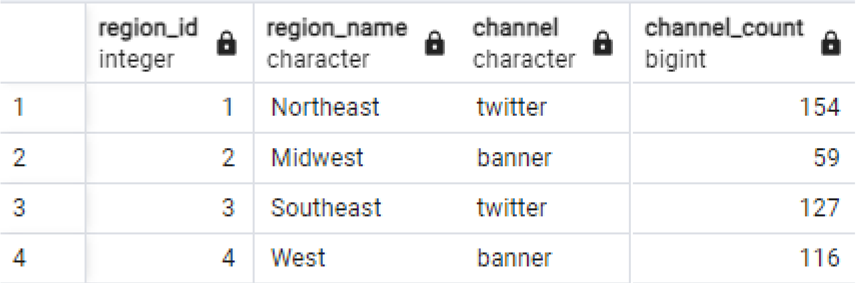


	avg_total_amt numeric	group_status text
1	2967.9236533333333333	group
2	3369.8241486920605782	not group

Question 8: The Marketing team needs to focus on channels for the newly added sales regions, and because of its limited resources, it will have to deprioritize/deactivate temporarily some channels in the old areas. Specifically it decided to deactivate, for every old region, the channel that is used the least for web events in that region. Which channels should they deactivate in each region? Use a window function to give the answer here.

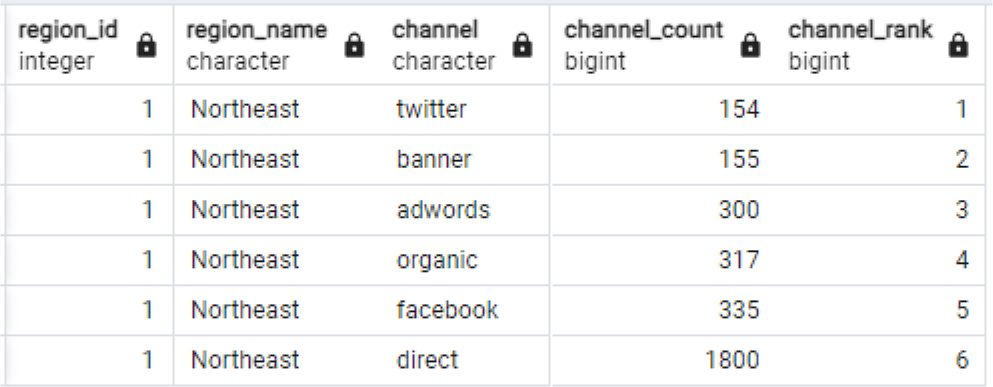
Response: The marketing team should deactivate Twitter in the Northeast and Southeast regions and deprioritize banners in the West and Midwest Regions.

```
SELECT d.region_id, d.region_name, d.channel, d.channel_count
FROM (
  SELECT r.id AS region_id, r.name AS region_name, w.channel, COUNT(w.id) AS
channel_count,
          RANK() OVER (PARTITION BY r.id ORDER BY COUNT(w.id)) AS channel_rank
  FROM web_events AS w
  JOIN accounts AS a ON w.account_id = a.id
  JOIN sales_reps AS s ON a.sales_rep_id = s.id
  JOIN region AS r ON s.region_id = r.id
  GROUP BY r.id, r.name, w.channel
) AS d
WHERE d.channel_rank = 1
ORDER BY d.region_id;
```



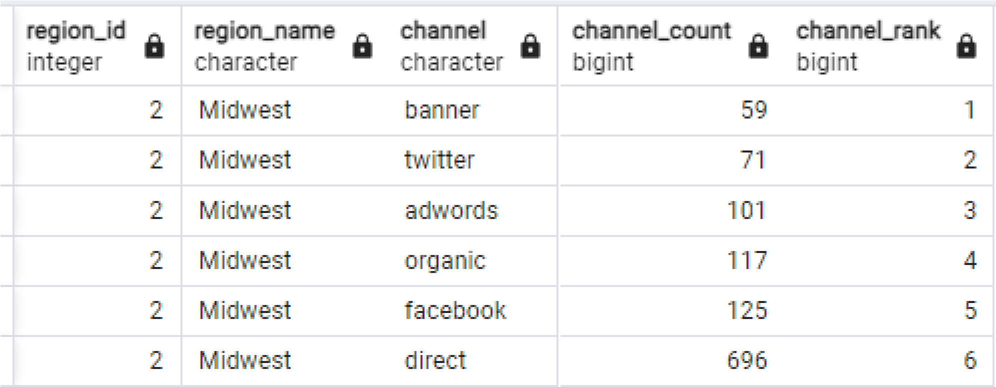
	region_id integer	region_name character	channel character	channel_count bigint
1	1	Northeast	twitter	154
2	2	Midwest	banner	59
3	3	Southeast	twitter	127
4	4	West	banner	116

```
SELECT r.id AS region_id, r.name AS region_name, w.channel, COUNT(w.id) AS
channel_count,
          RANK() OVER (PARTITION BY r.id ORDER BY COUNT(w.id)) AS channel_rank
FROM web_events AS w
JOIN accounts AS a ON w.account_id = a.id
JOIN sales_reps AS s ON a.sales_rep_id = s.id
JOIN region AS r ON s.region_id = r.id
WHERE r.id = 1
GROUP BY r.id, r.name, w.channel
```



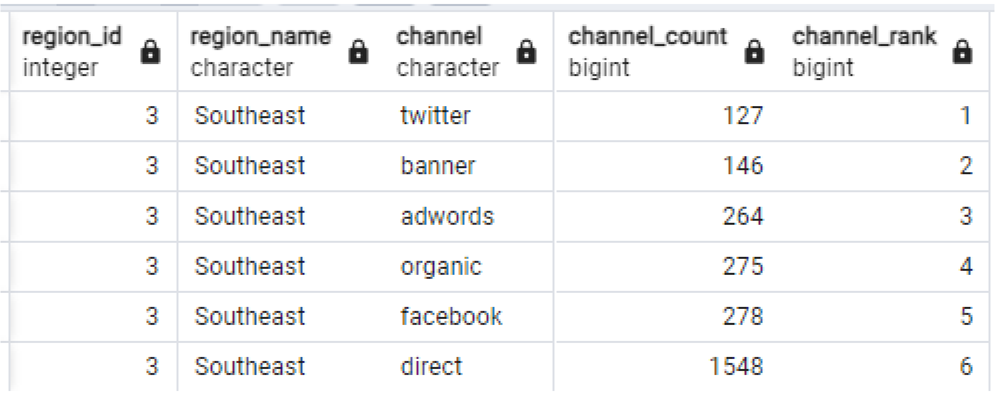
region_id integer	region_name character	channel character	channel_count bigint	channel_rank bigint
1	Northeast	twitter	154	1
1	Northeast	banner	155	2
1	Northeast	adwords	300	3
1	Northeast	organic	317	4
1	Northeast	facebook	335	5
1	Northeast	direct	1800	6

```
SELECT r.id AS region_id, r.name AS region_name, w.channel, COUNT(w.id) AS
channel_count,
          RANK() OVER (PARTITION BY r.id ORDER BY COUNT(w.id)) AS channel_rank
FROM web_events AS w
JOIN accounts AS a ON w.account_id = a.id
JOIN sales_reps AS s ON a.sales_rep_id = s.id
JOIN region AS r ON s.region_id = r.id
WHERE r.id = 2
GROUP BY r.id, r.name, w.channel
```



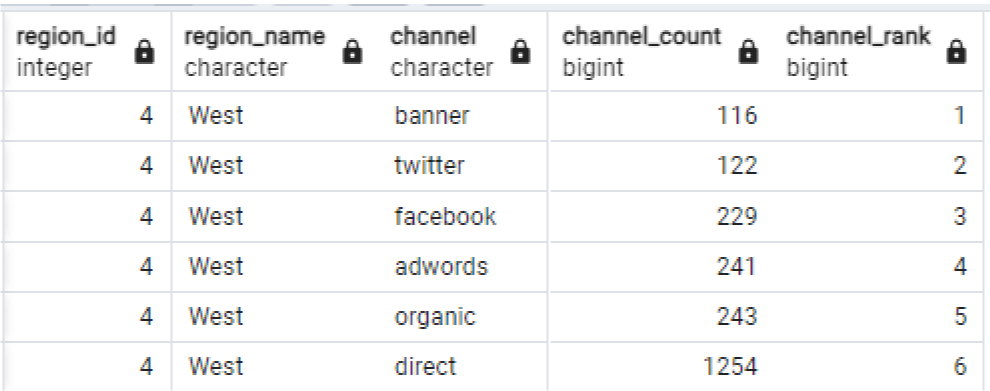
region_id integer	region_name character	channel character	channel_count bigint	channel_rank bigint
2	Midwest	banner	59	1
2	Midwest	twitter	71	2
2	Midwest	adwords	101	3
2	Midwest	organic	117	4
2	Midwest	facebook	125	5
2	Midwest	direct	696	6

```
SELECT r.id AS region_id, r.name AS region_name, w.channel, COUNT(w.id) AS
channel_count,
          RANK() OVER (PARTITION BY r.id ORDER BY COUNT(w.id)) AS channel_rank
FROM web_events AS w
JOIN accounts AS a ON w.account_id = a.id
JOIN sales_reps AS s ON a.sales_rep_id = s.id
JOIN region AS r ON s.region_id = r.id
WHERE r.id = 3
GROUP BY r.id, r.name, w.channel
```



region_id integer	region_name character	channel character	channel_count bigint	channel_rank bigint
3	Southeast	twitter	127	1
3	Southeast	banner	146	2
3	Southeast	adwords	264	3
3	Southeast	organic	275	4
3	Southeast	facebook	278	5
3	Southeast	direct	1548	6

```
SELECT r.id AS region_id, r.name AS region_name, w.channel, COUNT(w.id) AS
channel_count,
          RANK() OVER (PARTITION BY r.id ORDER BY COUNT(w.id)) AS channel_rank
FROM web_events AS w
JOIN accounts AS a ON w.account_id = a.id
JOIN sales_reps AS s ON a.sales_rep_id = s.id
JOIN region AS r ON s.region_id = r.id
WHERE r.id = 4
GROUP BY r.id, r.name, w.channel
```



region_id integer	region_name character	channel character	channel_count bigint	channel_rank bigint
4	West	banner	116	1
4	West	twitter	122	2
4	West	facebook	229	3
4	West	adwords	241	4
4	West	organic	243	5
4	West	direct	1254	6