SEPTEMBER 22, 2023

DATA SCIENCE

ASSIGNMENT 3

ABDULLAH MEHMOOD MINHAS
20F-0204
B(SE)-7A

Assignment 3

Problem 1:

Use the web_events table to find all information regarding individuals who were contacted via the organic
or adwords channels, and started their account at any point in 2016, sorted from newest to oldest.

Query

```
--Question 1
SELECT a.id AS ID, a.name AS Name
FROM  web_events AS w
INNER JOIN accounts AS a
ON a.id = w.account_id AND
w.channel IN ('organic', 'adwords') AND
(DATE_TRUNC('year', w.occurred_at) BETWEEN '2016-01-01' AND '2016-12-31')
GROUP BY a.id, a.name
ORDER BY MAX(w.occurred_at) DESC;
```

Results

Data Output | Messages | Notifications

	id integer	name character
1	4141	Univar
2	1851	Gilead Sciences
3	1791	United Continental Holdings
4	3351	Principal Financial
5	3521	Sherwin-Williams
6	2401	Eli Lilly
7	1151	Fannie Mae
8	3781	Sempra Energy
9	2281	Union Pacific
10	2801	Whole Foods Market

Total rows: 263 of 263 | Query complete 00:00:00.101

Report: List accounts contacted via 'organic' or 'adwords' channels in 2016, ordered by the most recent contact date.

Problem 2

Provide the name for each region for every order, as well as the account name and the unit price they paid (total_amt_usd/total) for the order. Your final table should have 3 columns: region name, account name, and unit price. A few accounts have 0 for total, so divide it by (total + 0.01) to assure not dividing by zero.

Query

```
--Question 2
SELECT r.name AS "region name", sa.name AS "account name",  o.total_amt_usd/ (CASE WHEN o.total = 0 THEN (o.total +0.01)
                    ELSE o.total END) AS "unit Price"
FROM region AS r
RIGHT JOIN sales_reps AS sa
ON r.id = sa.region_id
LEFT JOIN accounts AS a
ON sa.id = a.sales_rep_id
LEFT JOIN orders AS o
ON a.id = o.account_id
```

Results

Data Output | Messages | Notifications

	region name character	account name character	unit Price numeric
1	Northeast	Samuel Racine	5.7599408284023669
2	Northeast	Samuel Racine	5.9653819444444444
3	Northeast	Samuel Racine	5.8801515151515152
4	Northeast	Samuel Racine	5.4445454545454545
5	Northeast	Samuel Racine	5.9605454545454545
6	Northeast	Samuel Racine	6.1690751445086705
7	Northeast	Samuel Racine	6.6292035398230088
8	Northeast	Samuel Racine	7.0040955631399317
9	Northeast	Samuel Racine	5.8338759689922481
10	Northeast	Samuel Racine	5.9362162162162162

Total rows: 1000 of 6913 | Query complete 00:00:00.074

Report: Calculate unit prices for orders associated with sales reps and regions.

Problem 3:

Provide a table that provides the region for each sales_rep along with their associated accounts. This time only for the Midwest region. Your final table should include three columns: the region name, the sales rep name, and the account name. Sort the accounts alphabetically (A-Z) according to account name.

Query

```
--Question 3
WITH mytable AS (
    SELECT r.name AS "region name", sa.name AS "sales rep name", a.name AS "Account name"
    FROM region AS r
    LEFT JOIN sales_reps AS sa
    ON r.id = sa.region_id
    AND r.name = 'Midwest'
    INNER JOIN accounts AS a
    ON sa.id = a.sales_rep_id
    ORDER BY a.name
)
SELECT * FROM mytable
```

Results

Data Output | Messages | Notifications

	region name character	sales rep name character	Account name character
1	Midwest	Chau Rowles	Abbott Laboratories
2	Midwest	Julie Starr	AbbVie
3	Midwest	Cliff Meints	Aflac
4	Midwest	Chau Rowles	Alcoa
5	Midwest	Charles Bidwell	Altria Group
6	Midwest	Delilah Krum	Amgen
7	Midwest	Charles Bidwell	Arrow Electronics
8	Midwest	Delilah Krum	AutoNation
9	Midwest	Delilah Krum	Capital One Financial
10	Midwest	Cordell Rieder	Centene

Total rows: 48 of 48 | Query complete 00:00:00.064

Report: Retrieve accounts in the 'Midwest' region along with sales reps, sorted by account name**.**

Problem 4

Provide a table that provides the region for each sales_rep along with their associated accounts. This time only for accounts where the sales rep has a first name starting with S and in the Midwest region. Your final table should include three columns: the region name, the sales rep name, and the account name. Sort the accounts alphabetically (A-Z) according to account name.

Results

```sql
--Question 4
WITH mytable AS (
    SELECT r.name AS "region name", sa.name AS "sales rep name", a.name AS "Account name"
    FROM region AS r
    LEFT JOIN sales_reps AS sa
    ON r.id = sa.region_id
    AND r.name = 'Midwest'
    AND sa.name LIKE 'S%'
    INNER JOIN accounts AS a
    ON sa.id = a.sales_rep_id
    ORDER BY a.name
)
SELECT * FROM mytable
```

Results

Data Output | Messages | Notifications

	region name character	sales rep name character	Account name character
1	Midwest	Sherlene Wetherington	Community Health Systems
2	Midwest	Sherlene Wetherington	Progressive
3	Midwest	Sherlene Wetherington	Rite Aid
4	Midwest	Sherlene Wetherington	Time Warner Cable
5	Midwest	Sherlene Wetherington	U.S. Bancorp

Report: Retrieve accounts in the 'Midwest' region with 'S' starting sales reps, sorted by account name.

Problem 5

Provide a table that provides the region for each sales_rep along with their associated accounts. This time only for accounts where the sales rep has a last name starting with K and in the Midwest region. Your final table should include three columns: the region name, the sales rep name, and the account name. Sort the accounts alphabetically (A-Z) according to account name.

Query

```sql
--Question 5
WITH mytable AS (
    SELECT r.name AS "region name", sa.name AS "sales rep name", a.name AS "Account name"
    FROM region AS r
    LEFT JOIN sales_reps AS sa
    ON r.id = sa.region_id
    AND r.name = 'Midwest'
    AND SUBSTRING(sa.name, POSITION(' ' IN sa.name) + 1) LIKE 'K%'
    INNER JOIN accounts AS a
    ON sa.id = a.sales_rep_id
    ORDER BY a.name
)
SELECT * FROM mytable
```

Results

Data Output | Messages | Notifications

	region name character	sales rep name character	Account name character
1	Midwest	Delilah Krum	Amgen
2	Midwest	Delilah Krum	AutoNation
3	Midwest	Delilah Krum	Capital One Financial
4	Midwest	Delilah Krum	Cummins
5	Midwest	Carletta Kosinski	Danaher
6	Midwest	Carletta Kosinski	Dollar General
7	Midwest	Delilah Krum	Hartford Financial Services Group
8	Midwest	Carletta Kosinski	International Paper
9	Midwest	Delilah Krum	Kimberly-Clark
10	Midwest	Carletta Kosinski	McDonald's

Total rows: 13 of 13 | Query complete 00:00:00.045

Report: Retrieve accounts in the 'Midwest' region with sales reps having 'K' starting last names, sorted by account name.

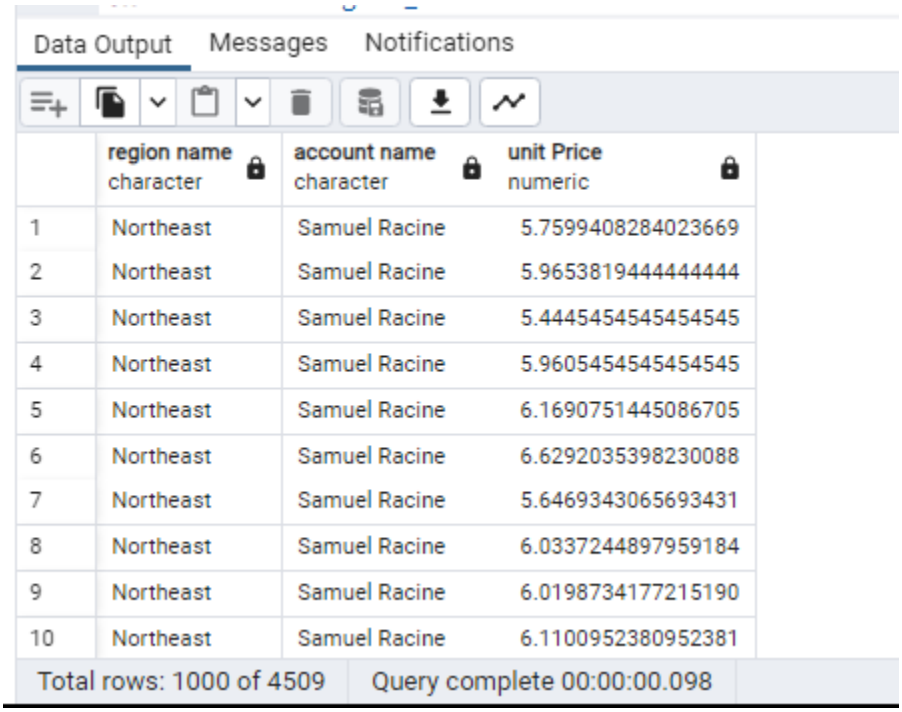
Problem 6

Provide the name for each region for every order, as well as the account name and the unit price they paid (total_amt_usd/total) for the order. However, you should only provide the results if the standard order quantity exceeds 100. Your final table should have 3 columns: region name, account name, and unit price. In order to avoid a division by zero error, adding .01 to the denominator here is helpful total_amt_usd/(total+0.01).

Query

```
--Question 6
SELECT r.name AS "region name", sa.name AS "account name",  o.total_amt_usd/ (CASE WHEN o.total = 0 THEN (o.total +0.01)
                    ELSE o.total END) AS "unit Price"
FROM region AS r
INNER JOIN sales_reps AS sa
ON r.id = sa.region_id
INNER JOIN accounts AS a
ON sa.id = a.sales_rep_id
INNER JOIN orders AS o
ON a.id = o.account_id
AND o.standard_qty > 100
```

Results



Data Output | Messages | Notifications

	region name character	account name character	unit Price numeric
1	Northeast	Samuel Racine	5.7599408284023669
2	Northeast	Samuel Racine	5.9653819444444444
3	Northeast	Samuel Racine	5.4445454545454545
4	Northeast	Samuel Racine	5.9605454545454545
5	Northeast	Samuel Racine	6.1690751445086705
6	Northeast	Samuel Racine	6.6292035398230088
7	Northeast	Samuel Racine	5.6469343065693431
8	Northeast	Samuel Racine	6.0337244897959184
9	Northeast	Samuel Racine	6.0198734177215190
10	Northeast	Samuel Racine	6.1100952380952381

Total rows: 1000 of 4509 | Query complete 00:00:00.098

Report: Calculate unit prices for orders linked to sales reps and regions with quantity criteria.

Problem 7

Provide the name for each region for every order, as well as the account name and the unit price they paid (total_amt_usd/total) for the order. However, you should only provide the results if the standard order quantity exceeds 100 and the poster order quantity exceeds 50. Your final table should have 3 columns: region name, account name, and unit price. Sort for the smallest unit price first.

Query:

```sql
--Question 7
SELECT r.name AS "region name", sa.name AS "account name",  o.total_amt_usd/ (CASE WHEN o.total = 0 THEN (o.total +0.01)
                    ELSE o.total END) AS "unit Price"
FROM region AS r
INNER JOIN sales_reps AS sa
ON r.id = sa.region_id
INNER JOIN accounts AS a
ON sa.id = a.sales_rep_id
INNER JOIN orders AS o
ON a.id = o.account_id
AND o.standard_qty > 100
AND o.poster_qty > 50
```

Results:

Data Output | Messages | Notifications

	region name character	account name character	unit Price numeric
1	Northeast	Samuel Racine	5.9653819444444444
2	Northeast	Samuel Racine	6.6292035398230088
3	Northeast	Renetta Carew	5.4837270726338958
4	Northeast	Cara Clarke	5.7139746835443038
5	Northeast	Sibyl Lauria	5.7202313624678663
6	Northeast	Sibyl Lauria	5.5119274809160305
7	Northeast	Sibyl Lauria	6.2834317343173432
8	Northeast	Sibyl Lauria	5.9117849462365591
9	Northeast	Necole Victory	6.2663137254901961
10	Northeast	Necole Victory	5.9628378378378378

Total rows: 835 of 835 | Query complete 00:00:00.070

Report: Calculate unit prices for orders linked to sales reps and regions with quantity and poster quantity criteria.

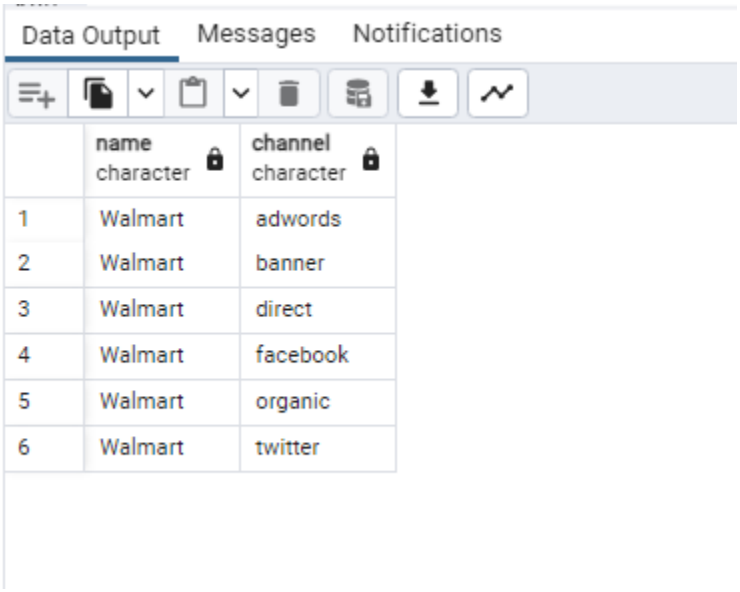
Problem 8

What are the different channels used by account id 1001? Your final table should have only 2 columns: account name and the different channels.

Query:

```sql
--Question 8
SELECT DISTINCT ac.name,  we.channel
FROM accounts AS ac
INNER JOIN web_events AS we
ON ac.id = we.account_id
AND ac.id = 1001
```

Results:



Data Output | Messages | Notifications

	name character	channel character
1	Walmart	adwords
2	Walmart	banner
3	Walmart	direct
4	Walmart	facebook
5	Walmart	organic
6	Walmart	twitter

Report: List accounts with 'facebook' channel for Account ID 1001.

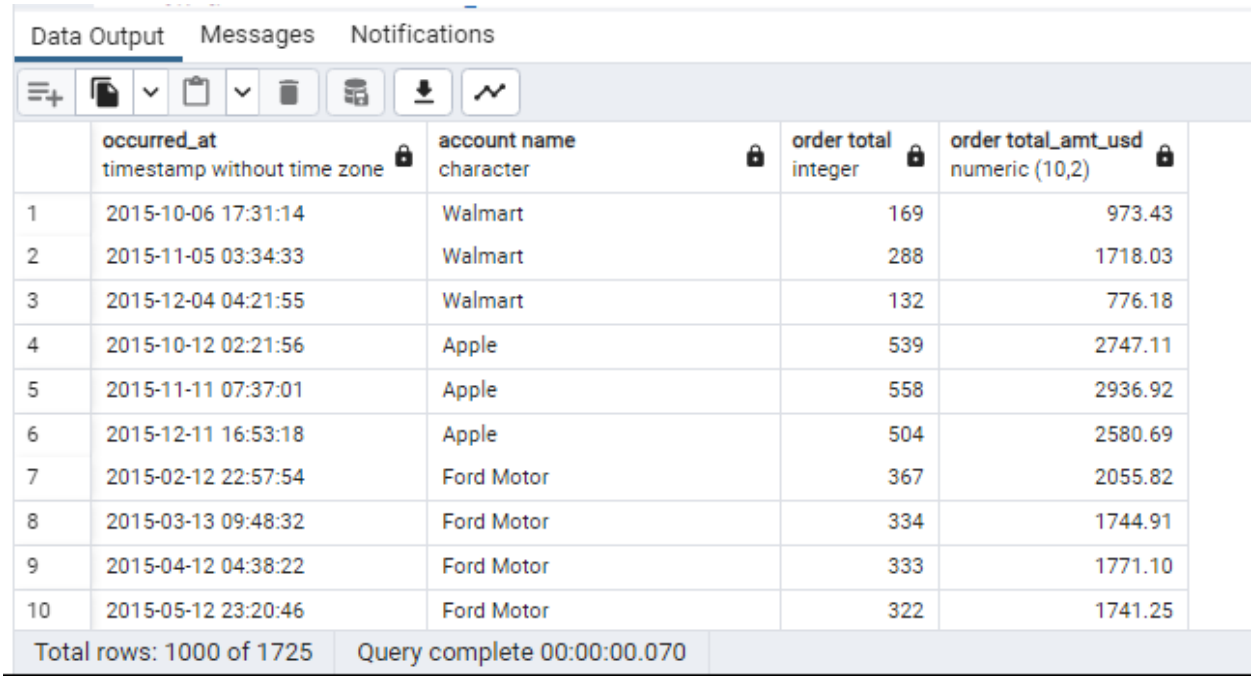
Problem 9

Find all the orders that occurred in 2015. Your final table should have 4 columns: occurred_at, accountname, order total, and order total_amt_usd.

Query:

```sql
--Question 9
SELECT o.occurred_at, a.name AS "account name", o.total AS "order total", o.total_amt_usd AS "order total_amt_usd"
FROM  orders AS o
INNER JOIN accounts AS a
ON a.id = o.account_id AND
(DATE_TRUNC('year', o.occurred_at) BETWEEN '2015-01-01' AND '2015-12-31')
```

Results:



Data Output | Messages | Notifications

	occurred_at timestamp without time zone	account name character	order total integer	order total_amt_usd numeric (10,2)
1	2015-10-06 17:31:14	Walmart	169	973.43
2	2015-11-05 03:34:33	Walmart	288	1718.03
3	2015-12-04 04:21:55	Walmart	132	776.18
4	2015-10-12 02:21:56	Apple	539	2747.11
5	2015-11-11 07:37:01	Apple	558	2936.92
6	2015-12-11 16:53:18	Apple	504	2580.69
7	2015-02-12 22:57:54	Ford Motor	367	2055.82
8	2015-03-13 09:48:32	Ford Motor	334	1744.91
9	2015-04-12 04:38:22	Ford Motor	333	1771.10
10	2015-05-12 23:20:46	Ford Motor	322	1741.25

Total rows: 1000 of 1725 | Query complete 00:00:00.070

Report: Retrieve orders placed by accounts in 2015 with account names, order total, and total amount in USD.

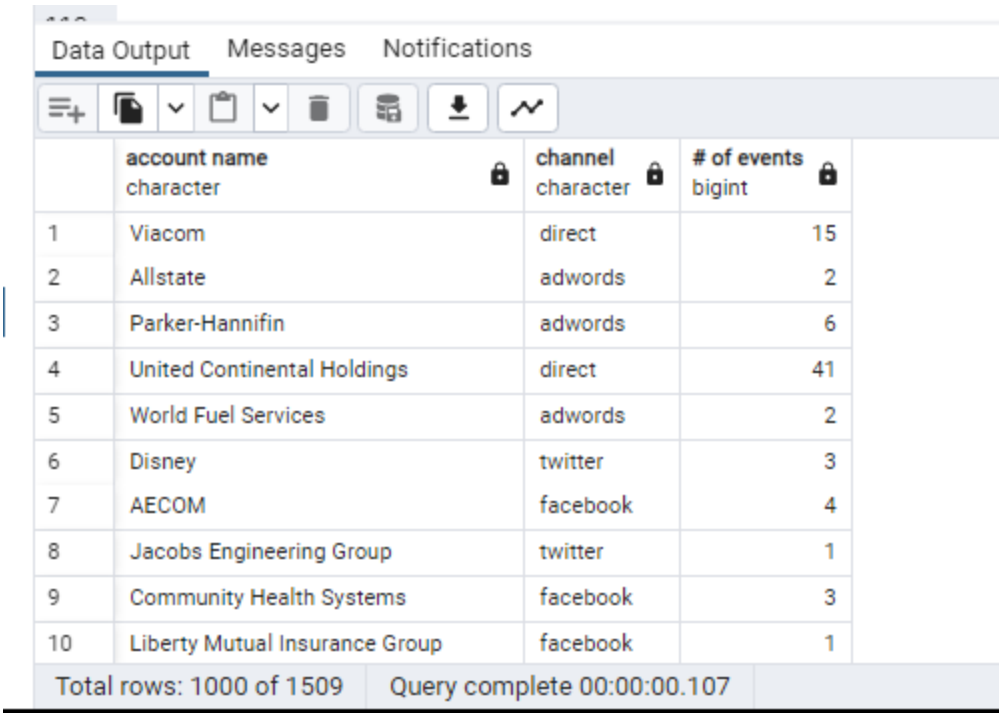
Problem 10

Find the total number of events happened for web for all the accounts. Your result set should have three columns account name, channel, and # of events.

Query:

```
--Question 10
SELECT ac.name AS "account name", we.channel, count(*) AS "# of events"
FROM web_events AS we
INNER JOIN accounts AS ac
ON we.account_id = ac.id
GROUP BY we.account_id, we.channel, ac.name
```

Results:



Data Output | Messages | Notifications

	account name character	channel character	# of events bigint
1	Viacom	direct	15
2	Allstate	adwords	2
3	Parker-Hannifin	adwords	6
4	United Continental Holdings	direct	41
5	World Fuel Services	adwords	2
6	Disney	twitter	3
7	AECOM	facebook	4
8	Jacobs Engineering Group	twitter	1
9	Community Health Systems	facebook	3
10	Liberty Mutual Insurance Group	facebook	1

Total rows: 1000 of 1509 | Query complete 00:00:00.107

Report: Count the number of events for each account in various channels.

Problem 11

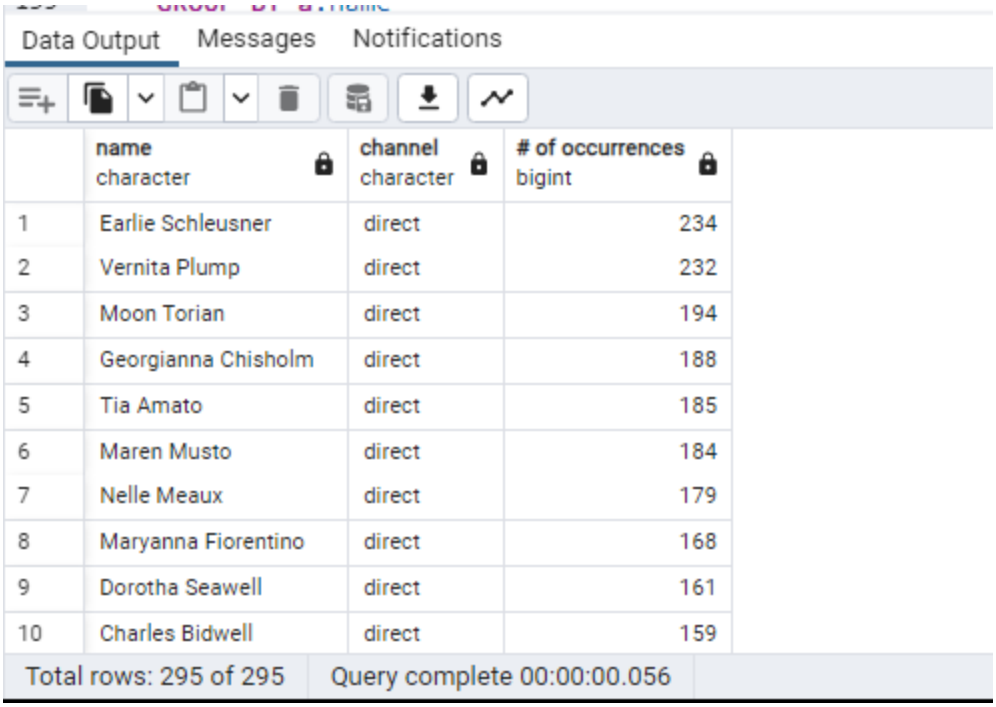
Determine the number of times a particular channel was used in the web_events table for each sales rep. Your final table should have three columns - the name of the sales rep, the channel, and the number of occurrences. Order your table with the highest number of occurrences first.

Query:

```
--Question 11
WITH mytable AS (SELECT sr.name AS "name", we.channel AS "channel", count(*) AS "# of occurrences"
    FROM web_events AS we
    INNER JOIN accounts AS a
    ON a.id = we.account_id
    INNER JOIN sales_reps AS sr
    ON a.sales_rep_id = sr.id
    GROUP BY sr.name, we.channel
    ORDER BY "# of occurrences" DESC
)

SELECT * FROM mytable
```

Results



Data Output | Messages | Notifications

	name character	channel character	# of occurrences bigint
1	Earlie Schleusner	direct	234
2	Vernita Plump	direct	232
3	Moon Torian	direct	194
4	Georgianna Chisholm	direct	188
5	Tia Amato	direct	185
6	Maren Musto	direct	184
7	Nelle Meaux	direct	179
8	Maryanna Fiorentino	direct	168
9	Dorotha Seawell	direct	161
10	Charles Bidwell	direct	159

Total rows: 295 of 295 | Query complete 00:00:00.056

Report: Calculate the number of occurrences for sales reps in web events.

Problem 12

Using Having clause with aggregations

a. How many of the sales reps have more than 5 accounts that they manage?
b. Which account has the most orders?
c. Which accounts spent more than 30,000 usd total across all orders?
d. Which account has spent the most with us?
e. Which account has spent the least with us?
f. Which accounts used facebook as a channel to contact customers more than 6 times?
g. Which account used facebook most as a channel?

Queries:

```sql
--Question 12
--a
    (SELECT s.name
    FROM accounts AS a
    RIGHT JOIN sales_reps AS s
    ON a.sales_rep_id = s.id
    GROUP BY s.name
    HAVING COUNT(a.*) > 5)
--b
    (SELECT a.name
    FROM accounts AS a
    LEFT JOIN orders AS o
    ON o.account_id = a.id
    GROUP BY a.name
    HAVING COUNT(o.*) = (SELECT MAX(count)
                         FROM(SELECT COUNT(*) AS "count"
                              FROM orders AS o
                              GROUP BY o.account_id) AS subquery
                        )
    )
--c
    (SELECT a.name
    FROM accounts AS a
    LEFT JOIN orders AS o
    ON o.account_id = a.id
    GROUP BY a.name
    HAVING SUM(o.total_amt_usd) > 30000
    )
--d
    (SELECT a.name
    FROM accounts AS a
    LEFT JOIN orders AS o
    ON o.account_id = a.id
    GROUP BY a.name
    HAVING SUM(o.total_amt_usd) = (SELECT MAX(amount)
                                   FROM (SELECT SUM(total_amt_usd) AS amount
                                         FROM orders
                                         GROUP BY account_id
                                        ) AS subquery
                                  )
    )
```

```sql
--e
    (SELECT a.name
    FROM accounts AS a
    LEFT JOIN orders AS o
    ON o.account_id = a.id
    GROUP BY a.name
    HAVING SUM(o.total_amt_usd) = (SELECT MIN(amount)
                                   FROM (SELECT SUM(total_amt_usd) AS amount
                                         FROM orders
                                         GROUP BY account_id
                                        ) AS subquery
                                  )
    )
--f
    (SELECT a.name, w.channel
    FROM accounts AS a
    LEFT JOIN web_events AS w
    ON a.id = w.account_id
    AND w.channel = 'facebook'
    GROUP BY a.name, w.channel
    HAVING count(a.*) > 6)
--g
    (SELECT a.name
    FROM accounts AS a
    LEFT JOIN web_events AS w
    ON a.id = w.account_id
    AND w.channel = 'facebook'
    GROUP BY a.name
    HAVING count(a.*) = (SELECT MAX(account)
                         FROM (SELECT COUNT(a.*) AS account
                               FROM accounts AS a
                               LEFT JOIN web_events AS w
                               ON a.id = w.account_id
                               AND w.channel = 'facebook'
                               GROUP BY a.name
                         ) AS subquery
                        ))
```

Results:

```
146      ON o.account_id = a.id
```

Data Output | Messages | Notifications

	name character
1	Samuel Racine
2	Elwood Shutt
3	Michel Averette
4	Brandie Riva
5	Elba Felder
6	Nelle Meaux
7	Necole Victory
8	Saran Ram
9	Moon Torian
10	Sibyl Lauria

Total rows: 34 of 34 | Query complete 00:00:00.078

Data Output | Messages | Notifications

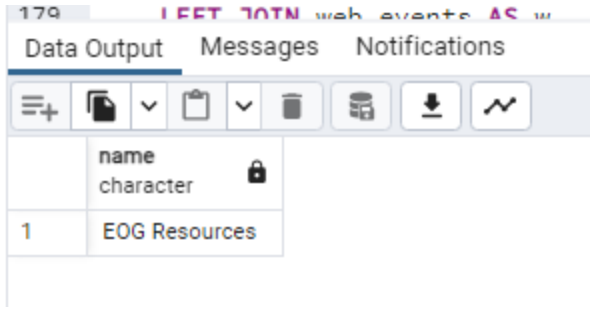
	name character
1	Leucadia National

```
157                                    FROM (SELECT SUM(to
```

Data Output | Messages | Notifications

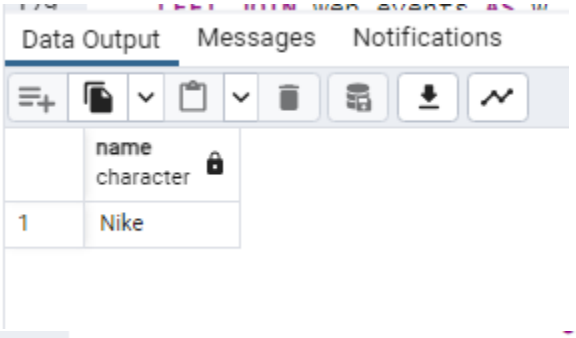
	name character
1	Monsanto
2	KKR
3	Performance Food Group
4	Paccar
5	USAA
6	CST Brands
7	Ally Financial
8	Amgen
9	Broadcom
10	Reynolds American

Total rows: 204 of 204 | Query complete 00:00:00.086



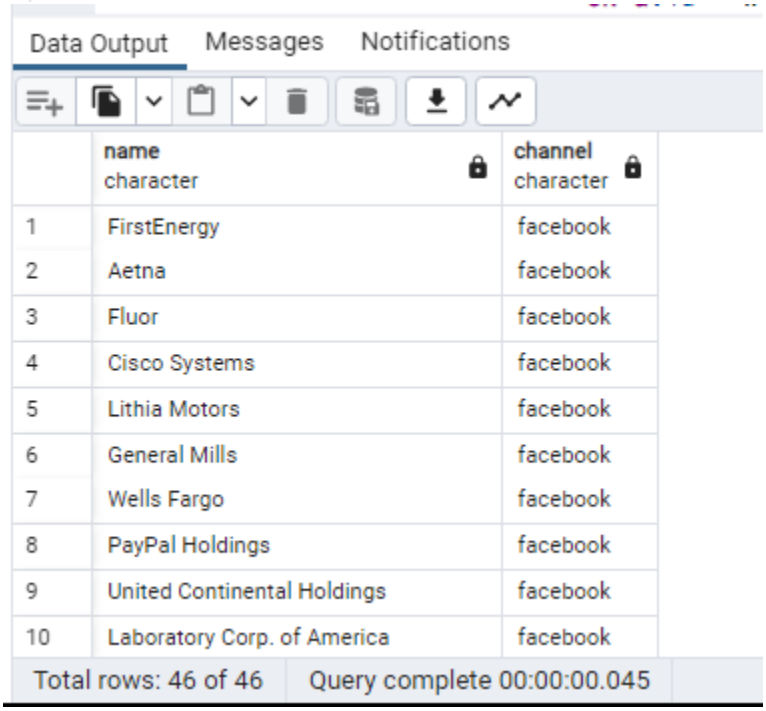
Data Output | Messages | Notifications

	name character
1	EOG Resources



Data Output | Messages | Notifications

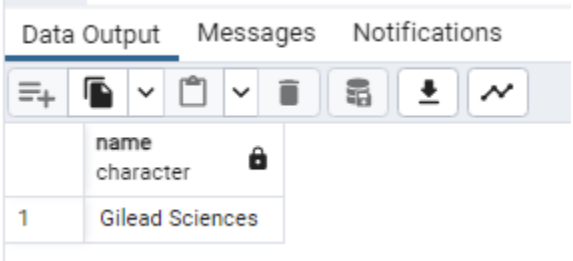
	name character
1	Nike



Data Output | Messages | Notifications

	name character	channel character
1	FirstEnergy	facebook
2	Aetna	facebook
3	Fluor	facebook
4	Cisco Systems	facebook
5	Lithia Motors	facebook
6	General Mills	facebook
7	Wells Fargo	facebook
8	PayPal Holdings	facebook
9	United Continental Holdings	facebook
10	Laboratory Corp. of America	facebook

Total rows: 46 of 46 | Query complete 00:00:00.045



Data Output | Messages | Notifications

	name character
1	Gilead Sciences

Report: Retrieve accounts based on different conditions: (a) Sales reps with more than 5 associated accounts, (b) Accounts with the maximum number of orders, (c) Accounts with total spending over $30,000, (d) Accounts with the maximum order amount, (e) Accounts with the minimum order amount, (f) Accounts with more than 6 Facebook events, (g) Accounts with the maximum number of Facebook events.

Problem 13

Write a query to display for each order, the account ID, total amount of the order, and the level of the order - ‘Large’ or ’Small’ - depending on if the order is $300 or more, or smaller than $3000.

Query:

```
--Question 13
SELECT account_id, total_amt_usd AS "Total amount of order", (CASE WHEN total_amt_usd <= 300 THEN 'Small'
                                                               WHEN total_amt_usd > 300 AND total_amt_usd < 3000 THEN 'Medium'
                                                               ELSE 'Large' END) AS "level"
FROM orders
```

Results

Data Output | Messages | Notifications

	account_id integer	Total amount of order numeric (10,2)	level text
1	1001	973.43	Medium
2	1001	1718.03	Medium
3	1001	776.18	Medium
4	1001	958.24	Medium
5	1001	983.49	Medium
6	1001	1067.25	Medium
7	1001	1498.20	Medium
8	1001	2052.20	Medium
9	1001	752.57	Medium
10	1001	878.56	Medium

Total rows: 1000 of 6912 | Query complete 00:00:00.078

Report: Categorize orders into 'Small,' 'Medium,' or 'Large' based on their total amount.

Problem 14

Write a query to display the number of orders in each of three categories, based on the total number of items in each order. The three categories are: 'At Least 2000', 'Between 1000 and 2000' and 'Less than 1000'

Query:

```
--Question 14
SELECT *, (CASE WHEN total< 1000 THEN 'Less then 1000'
       WHEN total >= 2000 THEN 'Atleast 2000'
        ELSE 'Between 1000 and 2000' END) AS "category"
FROM orders
```

Results

Data Output | Messages | Notifications

	id integer	account_id integer	occurred_at timestamp without time zone	standard_qty integer	gloss_qty integer	poster_qty integer	total integer	standard_amt_usd numeric (10,2)	gloss_amt_usd numeric (10,2)	poster_amt_usd numeric (10,2)	total_amt_usd numeric (10,2)	category text
1	1	1001	2015-10-06 17:31:14	123	22	24	169	613.77	164.78	194.88	973.43	Less then 1000
2	2	1001	2015-11-05 03:34:33	190	41	57	288	948.10	307.09	462.84	1718.03	Less then 1000
3	3	1001	2015-12-04 04:21:55	85	47	0	132	424.15	352.03	0.00	776.18	Less then 1000
4	4	1001	2016-01-02 01:18:24	144	32	0	176	718.56	239.68	0.00	958.24	Less then 1000
5	5	1001	2016-02-01 19:27:27	108	29	28	165	538.92	217.21	227.36	983.49	Less then 1000
6	6	1001	2016-03-02 15:29:32	103	24	46	173	513.97	179.76	373.52	1067.25	Less then 1000
7	7	1001	2016-04-01 11:20:18	101	33	92	226	503.99	247.17	747.04	1498.20	Less then 1000
8	8	1001	2016-05-01 15:55:51	95	47	151	293	474.05	352.03	1226.12	2052.20	Less then 1000
9	9	1001	2016-05-31 21:22:48	91	16	22	129	454.09	119.84	178.64	752.57	Less then 1000
10	10	1001	2016-06-30 12:32:05	94	46	8	148	469.06	344.54	64.96		

Successfully run. Total query runti

Total rows: 1000 of 6912 | Query complete 00:00:00.126

Report: Categorize orders into 'Less than 1000,' 'Atleast 2000,' or 'Between 1000 and 2000' based on their total amount.

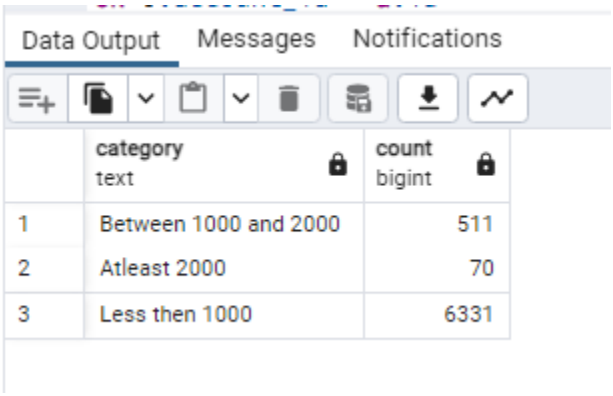
Problem 15

Count the orders based on the categories you defined in previous question. Your table must have tow column, category and count of orders for each category

Query:

```
--Question 15
SELECT (CASE WHEN total< 1000 THEN 'Less then 1000'
       WHEN total >= 2000 THEN 'Atleast 2000'
        ELSE 'Between 1000 and 2000' END) AS "category", count(*)
FROM orders
GROUP BY (CASE WHEN total< 1000 THEN 'Less then 1000'
       WHEN total >= 2000 THEN 'Atleast 2000'
        ELSE 'Between 1000 and 2000' END)
```

Results



Data Output | Messages | Notifications

	category text	count bigint
1	Between 1000 and 2000	511
2	Atleast 2000	70
3	Less then 1000	6331

Report: Count the number of orders in 'Less than 1000,' 'Atleast 2000,' or 'Between 1000 and 2000' categories.

Problem 16

We would now like to perform a similar calculation to the first, but we want to obtain the total amount spent by customers only in 2016 and 2017. Keep the same levels as in the previous question. Order with the top spending customers listed first.

Query:

```
--Question 16
SELECT a.name, SUM(o.total_amt_usd) AS "Total Spendings", (CASE WHEN SUM(o.total_amt_usd)< 1000 THEN 'Less then 1000'
       WHEN SUM(o.total_amt_usd) >= 2000 THEN 'Atleast 2000'
        ELSE 'Between 1000 and 2000' END) AS "category"
FROM orders AS o
INNER JOIN accounts AS a
ON o.account_id = a.id
AND (DATE_TRUNC('year', o.occurred_at) BETWEEN '2016-01-01' AND '2017-12-31')
GROUP BY a.name
ORDER BY "Total Spendings" DESC
```

Results

Data Output | Messages | Notifications

	name character	Total Spendings numeric	category text
1	Pacific Life	255319.18	Atleast 2000
2	Mosaic	172180.04	Atleast 2000
3	CHS	163471.78	Atleast 2000
4	Core-Mark Holding	148105.93	Atleast 2000
5	Disney	129157.38	Atleast 2000
6	National Oilwell Varco	121873.16	Atleast 2000
7	Sears Holdings	114003.21	Atleast 2000
8	State Farm Insurance Cos.	111810.55	Atleast 2000
9	Fidelity National Financial	110027.29	Atleast 2000
10	BB&T Corp.	107300.05	Atleast 2000

Total rows: 322 of 322 Query complete 00:00:00.121

Report: Calculate total spendings for accounts in 2016-2017 and categorize them into different groups.

Problem 17

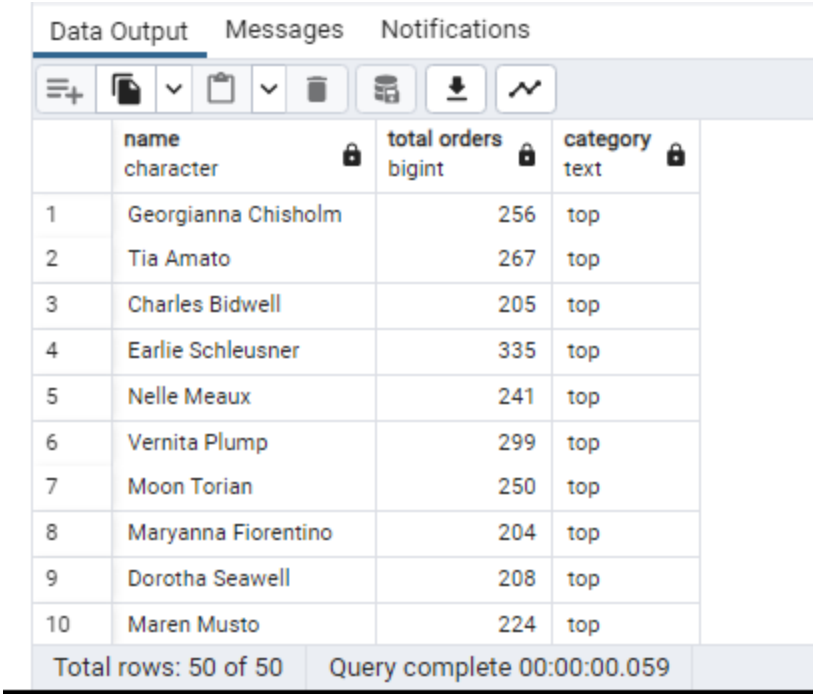
We would like to identify top performing sales reps, which are sales reps associated with more than 200 orders. Create a table with the sales rep name, the total number of orders, and a column with top or not depending on if they have more than 200 orders. Place the top sales people first in your final table.

Query:

```
--Question 17
WITH mytable AS (SELECT sr.name, COUNT(o) AS "total orders", (Case
                                                    WHEN COUNT(o) >= 200 THEN 'top'
                                                    ELSE 'not' END) AS "category"
    FROM orders AS o
    INNER JOIN accounts AS a
    ON o.account_id = a.id
    RIGHT JOIN sales_reps AS sr
    ON sr.id = a.sales_rep_id
    GROUP BY sr.name
    ORDER BY category DESC)

SELECT * FROM mytable
```

Results



Data Output | Messages | Notifications

	name character	total orders bigint	category text
1	Georgianna Chisholm	256	top
2	Tia Amato	267	top
3	Charles Bidwell	205	top
4	Earlie Schleusner	335	top
5	Nelle Meaux	241	top
6	Vernita Plump	299	top
7	Moon Torian	250	top
8	Maryanna Fiorentino	204	top
9	Dorotha Seawell	208	top
10	Maren Musto	224	top

Total rows: 50 of 50 Query complete 00:00:00.059

Report: Calculate the total number of orders for each sales rep and categorize them as 'top' or 'not.'

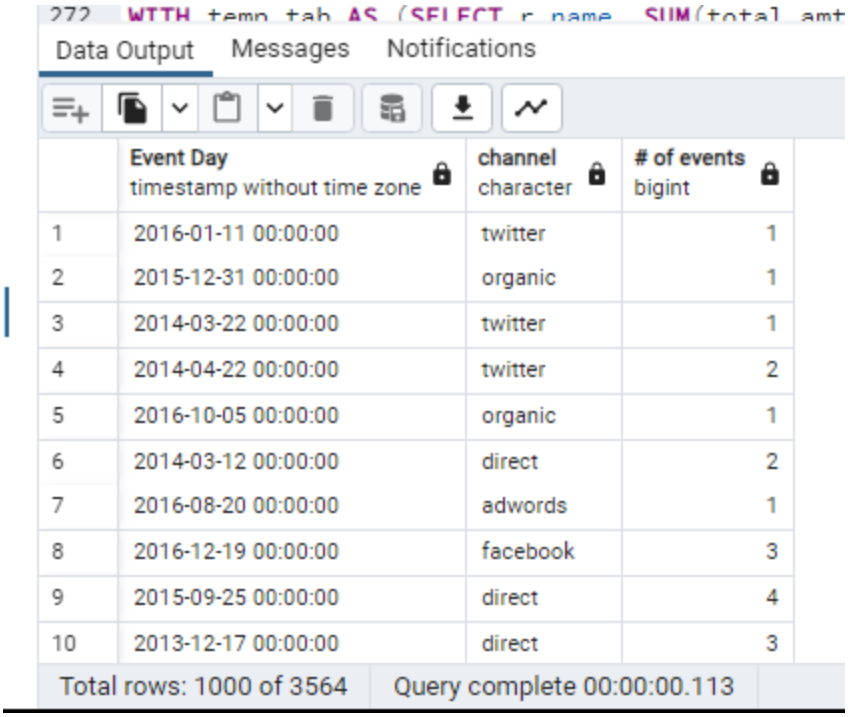
Problem 18

Find the number of events that happened each day for each channel. You query must return event_day, channel, and count in the result set.

Query:

```
--Question 18
SELECT DATE_TRUNC('day', occurred_at) AS "Event Day", channel, count(*) AS "# of events"
FROM web_events
GROUP BY DATE_TRUNC('day', occurred_at), channel
```

Results



	Event Day timestamp without time zone	channel character	# of events bigint
1	2016-01-11 00:00:00	twitter	1
2	2015-12-31 00:00:00	organic	1
3	2014-03-22 00:00:00	twitter	1
4	2014-04-22 00:00:00	twitter	2
5	2016-10-05 00:00:00	organic	1
6	2014-03-12 00:00:00	direct	2
7	2016-08-20 00:00:00	adwords	1
8	2016-12-19 00:00:00	facebook	3
9	2015-09-25 00:00:00	direct	4
10	2013-12-17 00:00:00	direct	3

Total rows: 1000 of 3564 Query complete 00:00:00.113

Report: Count the number of web events that occurred each day for each channel.

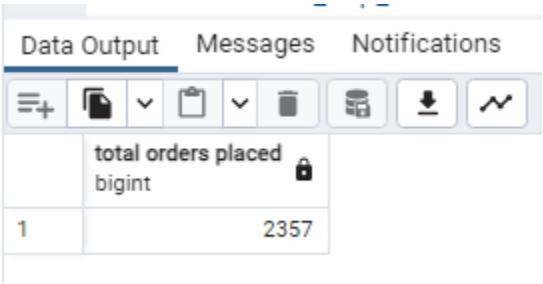
Problem 19

For the region with the largest sales total_amt_usd, how many total orders were placed?

Query:

```sql
--Question 19
SELECT COUNT(o) AS "total orders placed"
FROM orders AS o
INNER JOIN accounts AS a ON o.account_id = a.id
INNER JOIN sales_reps AS sr ON a.sales_rep_id = sr.id
INNER JOIN region AS r ON sr.region_id = r.id
GROUP BY r.name
HAVING SUM(total_amt_usd) = (
    SELECT MAX(total_sales)
    FROM (
        SELECT SUM(total_amt_usd) AS total_sales
        FROM orders AS o
        INNER JOIN accounts AS a ON o.account_id = a.id
        INNER JOIN sales_reps AS sr ON a.sales_rep_id = sr.id
        INNER JOIN region AS r ON sr.region_id = r.id
        GROUP BY r.name
    ) AS subquery
);
```

Results



Report: Calculate the total number of orders placed in regions with the highest total sales.

Problem 20

For the region with the largest sales total_amt_usd, how many total orders were placed? Write query using CTE for this same problem as previous.

Query:

```sql
--Question 20
WITH temp_tab AS (SELECT r.name, SUM(total_amt_usd) AS total_count, COUNT(o)
    FROM orders AS o
    INNER JOIN accounts AS a
    ON o.account_id = a.id
    INNER JOIN sales_reps AS sr
    ON a.sales_rep_id = sr.id
    INNER JOIN region as r
    ON sr.region_id = r.id
    GROUP BY r.name
                )

SELECT count AS "total orders placed"
FROM temp_tab
WHERE total_count = (SELECT MAX(total_count) FROM temp_tab);
```

Results

Data Output | Messages | N

	total orders placed bigint
1	2357

Report: Count the total orders placed in the region with the highest total sales.