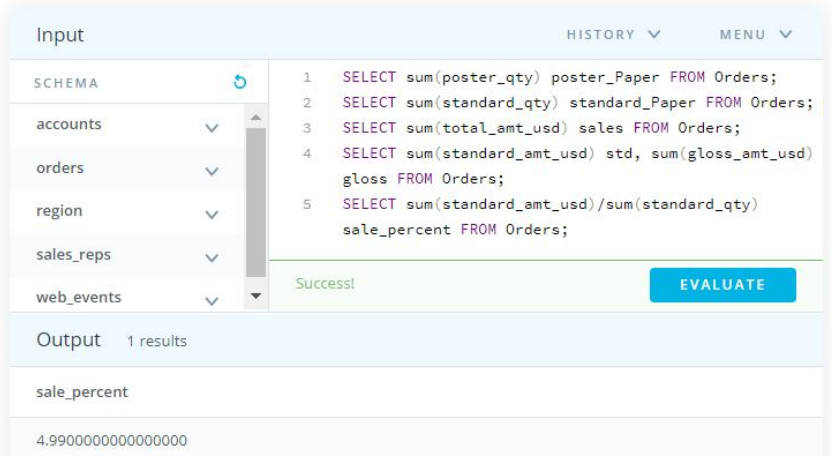
Aggregation Questions

Use the **SQL** environment below to find the solution for each of the following questions. If you get stuck or want to check your answers, you can find the answers at the top of the next concept.

1. Find the total amount of **poster_qty** paper ordered in the **orders** table.
2. Find the total amount of **standard_qty** paper ordered in the **orders** table.
3. Find the total dollar amount of sales using the **total_amt_usd** in the **orders** table.
4. Find the total amount spent on **standard_amt_usd** and **gloss_amt_usd** paper for each order in the orders table. This should give a dollar amount for each order in the table.
5. Find the **standard_amt_usd** per unit of **standard_qty** paper. Your solution should use both an aggregation and a mathematical operator.



Questions: MIN, MAX, & AVERAGE

Use the **SQL** environment below to assist with answering the following questions. Whether you get stuck or you just want to double check your solutions, my answers can be found at the top of the next concept.

1. When was the earliest order ever placed? You only need to return the date.

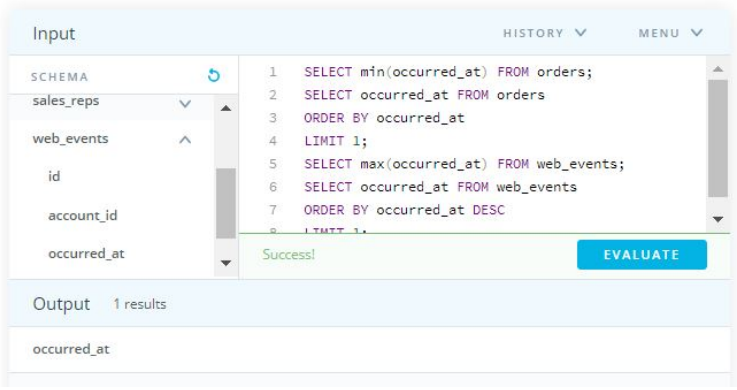
2. Try performing the same query as in question 1 without using an aggregation function.

3. When did the most recent (latest) **web_event** occur?

4. Try to perform the result of the previous query without using an aggregation function.

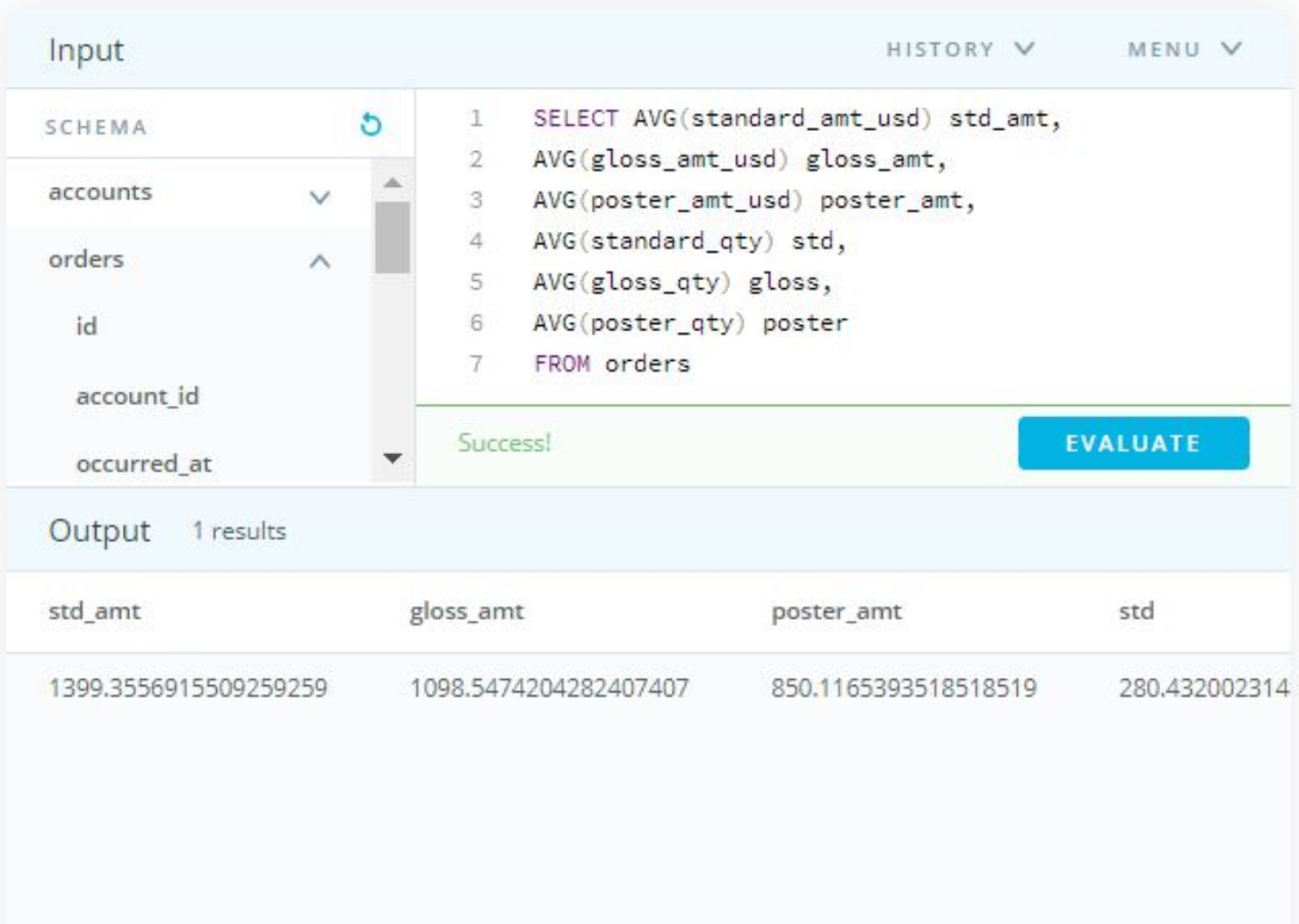
5. Find the mean (**AVERAGE**) amount spent per order on each paper type, as well as the mean amount of each paper type purchased per order. Your final answer should have 6 values - one for each paper type for the average number of sales, as well as the average amount.

6. Via the video, you might be interested in how to calculate the MEDIAN. Though this is more advanced than what we have covered so far try finding - what is the MEDIAN **total_usd** spent on all **orders**?

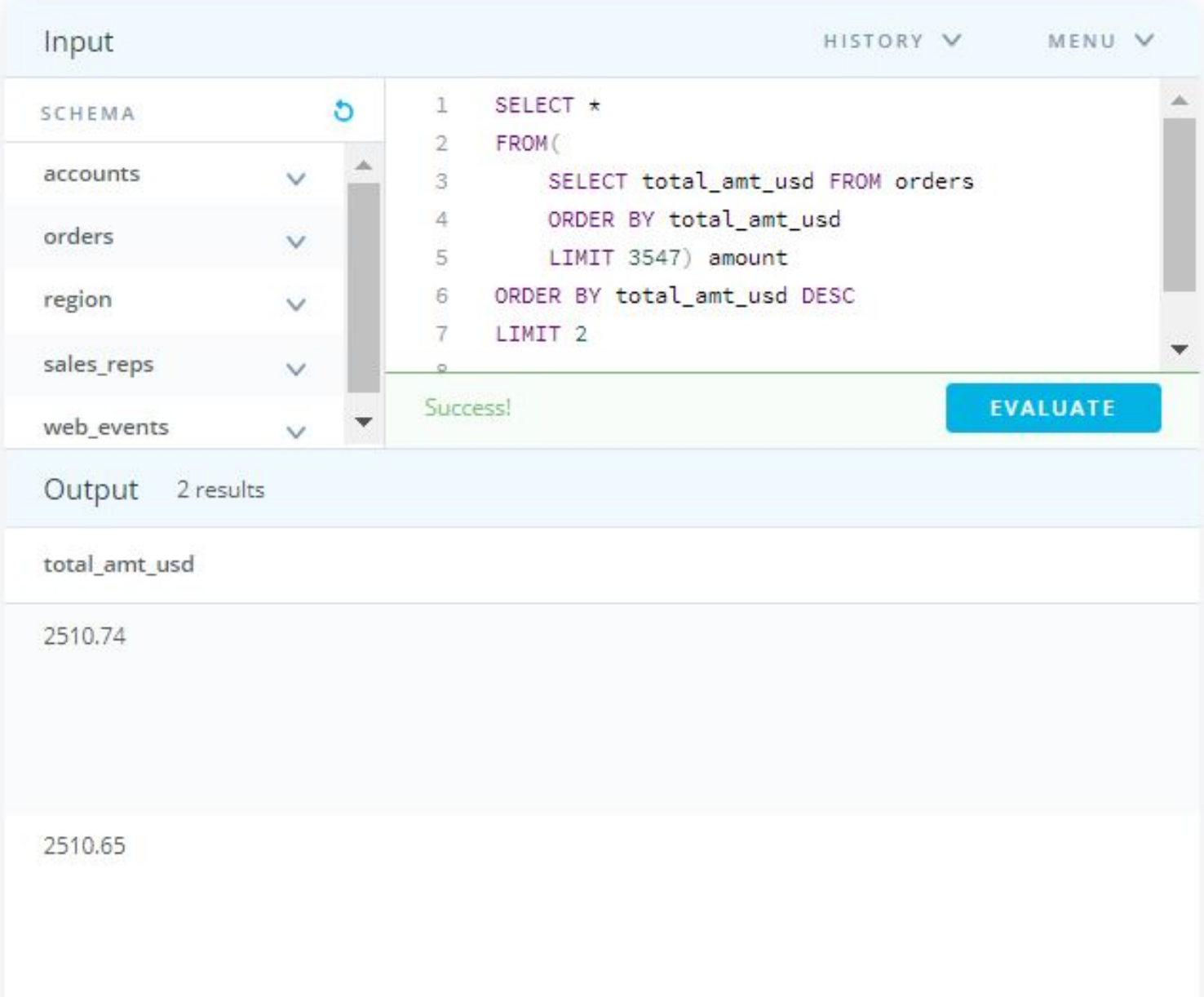


5. Find the mean (**AVERAGE**) amount spent per order on each paper type, as well as the mean amount of each paper type purchased per order. Your final answer should have 6 values - one for each paper type for the average number of sales, as well as the average amount.

6. Via the video, you might be interested in how to calculate the MEDIAN. Though this is more advanced than what we have covered so far try finding - what is the MEDIAN **total_usd** spent on all **orders**?



6. Via the video, you might be interested in how to calculate the MEDIAN. Though this is more advanced than what we have covered so far try finding - what is the MEDIAN **total_usd** spent on all **orders**?



1. Which **account** (by name) placed the earliest order? Your solution should have the **account name** and the **date** of the order.

2. Find the total sales in **usd** for each account. You should include two columns - the total sales for each company's orders in **usd** and the company **name**.

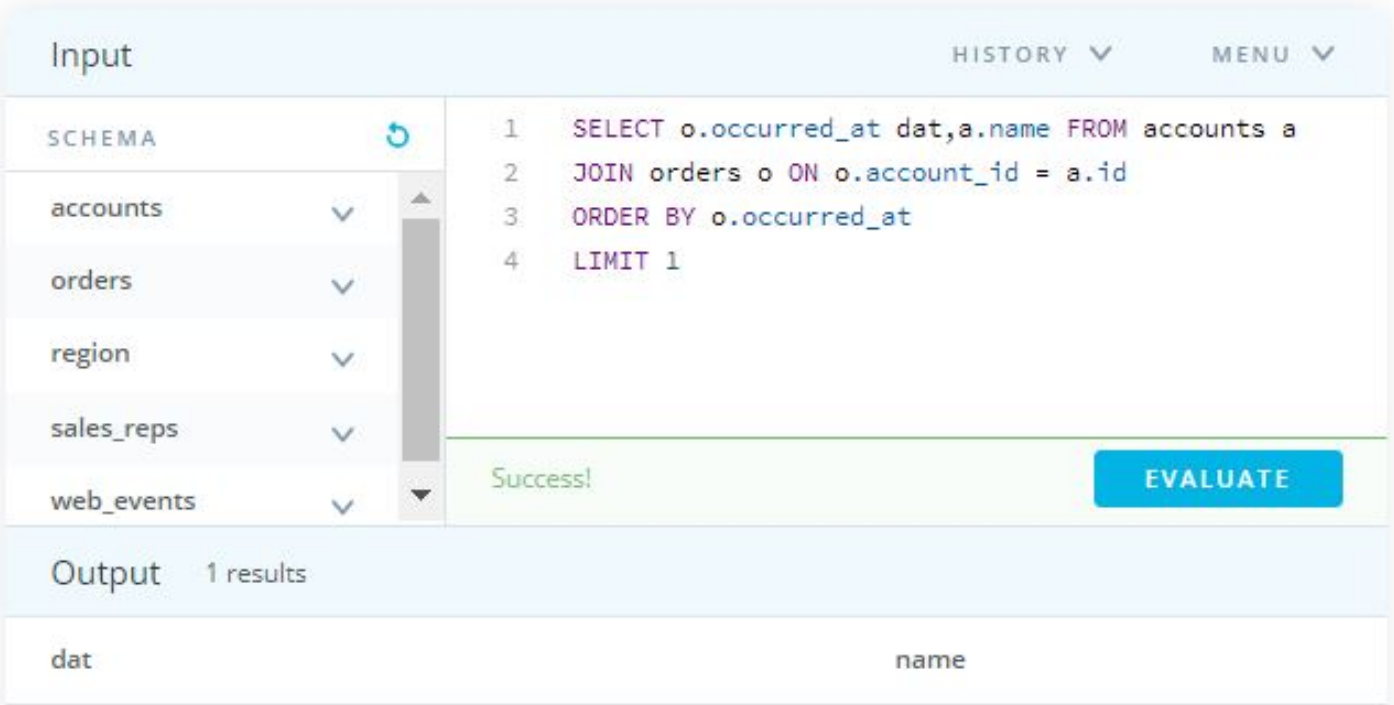
3. Via what **channel** did the most recent (latest) **web_event** occur, which **account** was associated with this **web_event**? Your query should return only three values - the **date**, **channel**, and **account name**.

4. Find the total number of times each type of **channel** from the **web_events** was used. Your final table should have two columns - the **channel** and the number of times the channel was used.

5. Who was the **primary contact** associated with the earliest **web_event**?

6. What was the smallest order placed by each **account** in terms of **total usd**. Provide only two columns - the account **name** and the **total usd**. Order from smallest dollar amounts to largest.

7. Find the number of **sales reps** in each region. Your final table should have two columns - the **region** and the number of **sales_reps**. Order from fewest reps to most reps.



2. Find the total sales in **usd** for each account. You should include two columns - the total sales for each company's orders in **usd** and the company **name**.

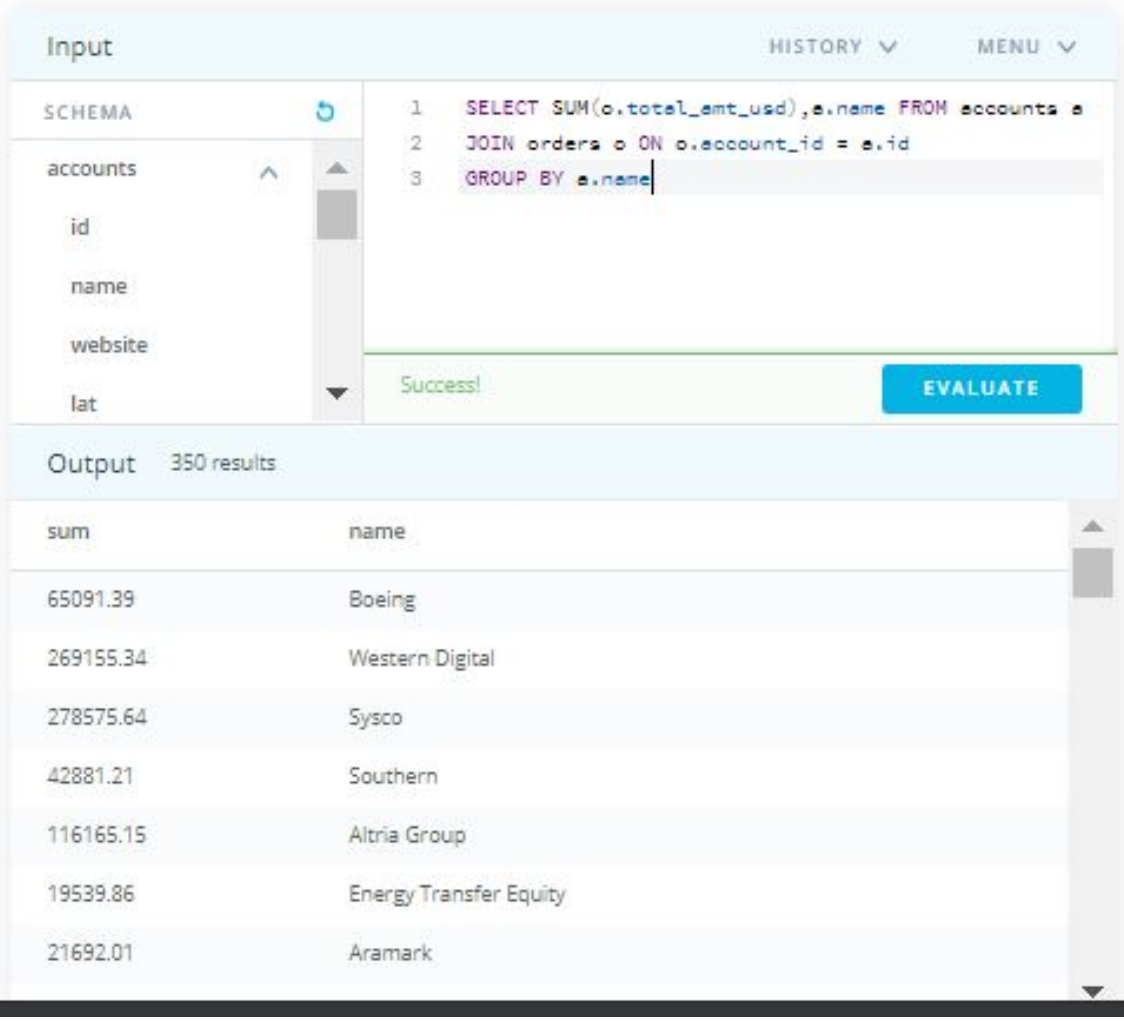
3. Via what **channel** did the most recent (latest) **web_event** occur, which **account** was associated with this **web_event**? Your query should return only three values - the **date**, **channel**, and **account name**.

4. Find the total number of times each type of **channel** from the **web_events** was used. Your final table should have two columns - the **channel** and the number of times the channel was used.

5. Who was the **primary contact** associated with the earliest **web_event**?

6. What was the smallest order placed by each **account** in terms of **total usd**. Provide only two columns - the account **name** and the **total usd**. Order from smallest dollar amounts to largest.

7. Find the number of **sales reps** in each region. Your final table should have two columns - the **region** and the number of **sales_reps**. Order from fewest reps to most reps.



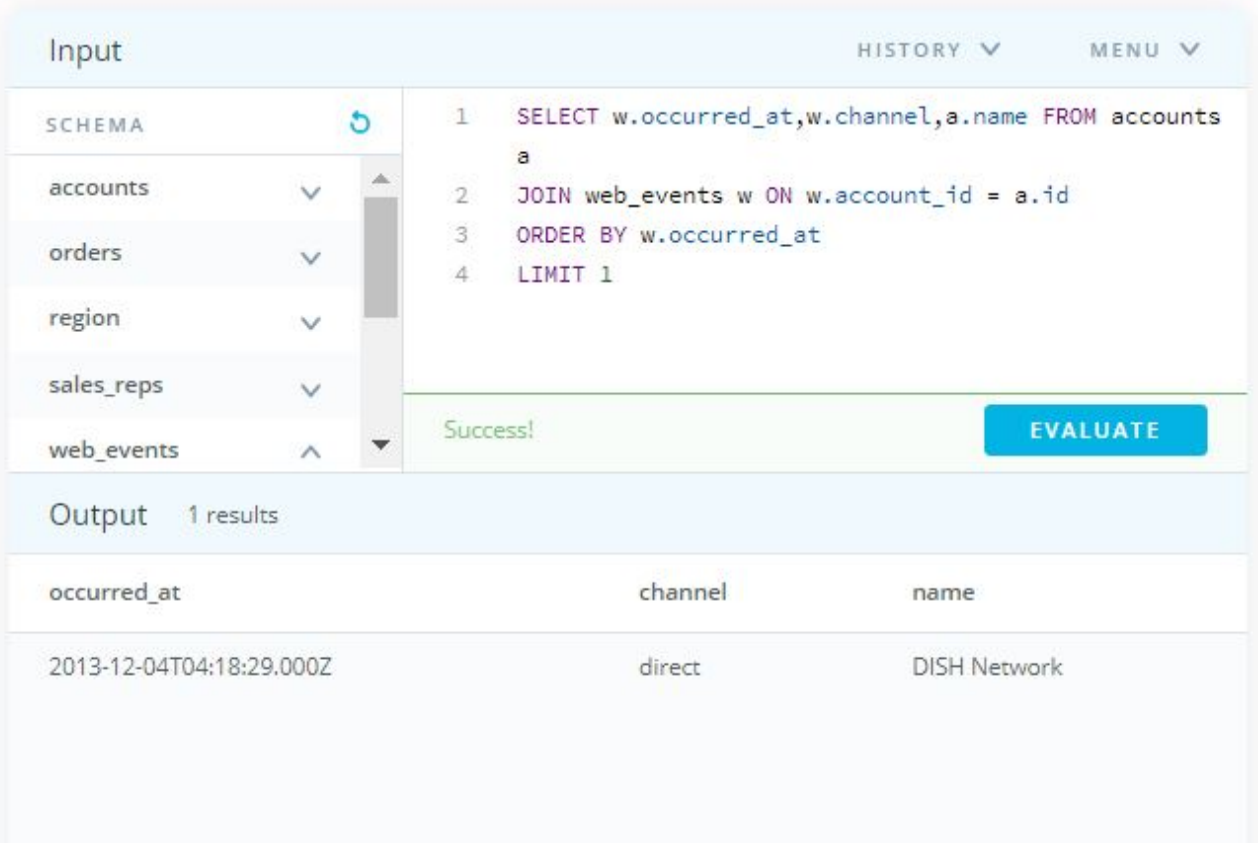
3. Via what **channel** did the most recent (latest) **web_event** occur, which **account** was associated with this **web_event**? Your query should return only three values - the **date**, **channel**, and **account name**.

4. Find the total number of times each type of **channel** from the **web_events** was used. Your final table should have two columns - the **channel** and the number of times the channel was used.

5. Who was the **primary contact** associated with the earliest **web_event**?

6. What was the smallest order placed by each **account** in terms of **total usd**. Provide only two columns - the account **name** and the **total usd**. Order from smallest dollar amounts to largest.

7. Find the number of **sales reps** in each region. Your final table should have two columns - the **region** and the number of **sales_reps**. Order from fewest reps to most reps.



4. Find the total number of times each type of **channel** from the **web_events** was used. Your final table should have two columns - the **channel** and the number of times the channel was used.

5. Who was the **primary contact** associated with the earliest **web_event**?

6. What was the smallest order placed by each **account** in terms of **total usd**. Provide only two columns - the account **name** and the **total usd**. Order from smallest dollar amounts to largest.

7. Find the number of **sales reps** in each region. Your final table should have two columns - the **region** and the number of **sales_reps**. Order from fewest reps to most reps.

Input HISTORY MENU

SCHEMA

- accounts
- orders
- region
- sales_reps
- web_events

```
1  SELECT w.channel, COUNT(*)
2  FROM web_events w
3  GROUP BY w.channel
```

Success! EVALUATE

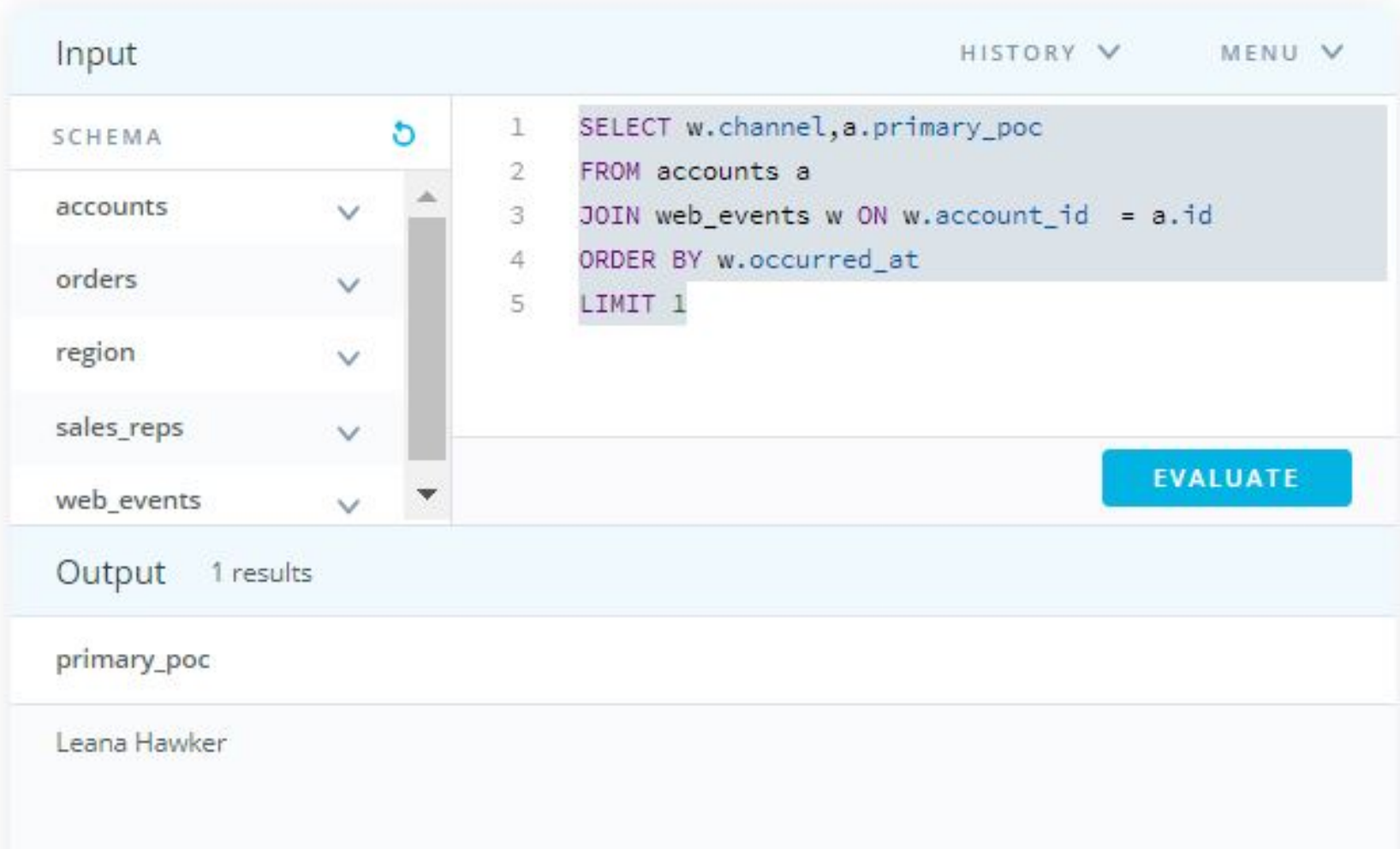
Output 6 results

channel	count
adwords	906
direct	5298
banner	476
facebook	967
organic	952
twitter	474

5. Who was the **primary contact** associated with the earliest **web_event**?

6. What was the smallest order placed by each **account** in terms of **total usd**. Provide only two columns - the account **name** and the **total usd**. Order from smallest dollar amounts to largest.

7. Find the number of **sales reps** in each region. Your final table should have two columns - the **region** and the number of **sales_reps**. Order from fewest reps to most reps.



6. What was the smallest order placed by each **account** in terms of **total usd**. Provide only two columns - the account **name** and the **total usd**. Order from smallest dollar amounts to largest.

7. Find the number of **sales reps** in each region. Your final table should have two columns - the **region** and the number of **sales_reps**. Order from fewest reps to most reps.

Input HISTORY MENU

SCHEMA

orders

id

account_id

occurred_at

```
1  SELECT a.name, sum(o.total_amt_usd) total FROM
   accounts a
2  JOIN orders o ON o.account_id = a.id
3  GROUP BY a.name
4  ORDER BY total
5  LIMIT 1
```

Success! EVALUATE

Output 1 results

name	total
Nike	390.25

7. Find the number of **sales reps** in each region. Your final table should have two columns - the **region** and the number of **sales_reps**. Order from fewest reps to most reps.

Input HISTORY MENU

SCHEMA

- sales_reps
  - id
  - name
  - region_id
- web_events

```
1  SELECT COUNT(s.name) sales_rep_count, r.name FROM
   region r
2  JOIN sales_reps s ON s.region_id = r.id
3  GROUP BY r.name
```

Success! EVALUATE

Output 4 results

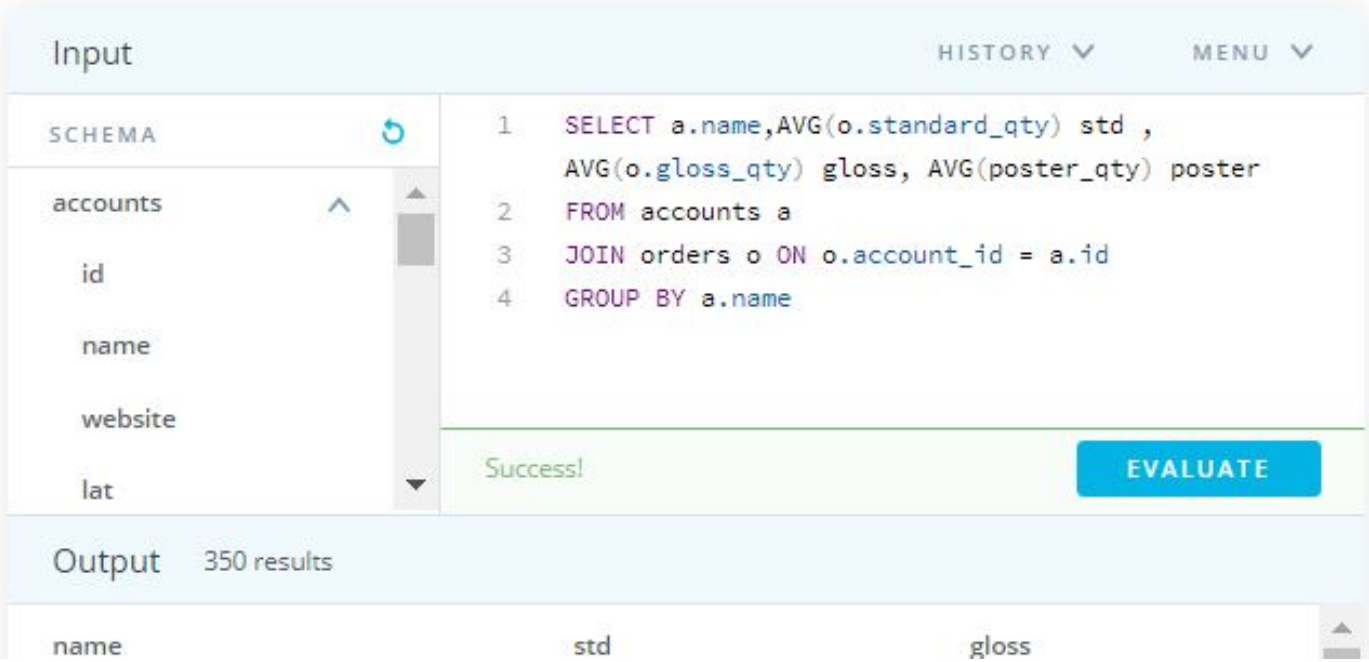
sales_rep_count	name
9	Midwest
10	Southeast
21	Northeast
10	West

1. For each account, determine the average amount of each type of paper they purchased across their orders. Your result should have four columns - one for the account **name** and one for the average quantity purchased for each of the paper types for each account.

2. For each account, determine the average amount spent per order on each paper type. Your result should have four columns - one for the account **name** and one for the average amount spent on each paper type.

3. 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.

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



3. 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.

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

Input HISTORY MENU

SCHEMA

- accounts
- orders
- region
- sales_reps
- web_events

```
1  SELECT s.name, w.channel, COUNT(w.channel) events
2  FROM accounts a
3  JOIN web_events w ON w.account_id =  a.id
4  JOIN sales_reps s ON s.id = a.sales_rep_id
5  GROUP BY w.channel,s.name
6  ORDER BY events DESC
7
```

Success! EVALUATE

Output 295 results

name	channel	events
Earlie Schleusner	direct	234
Vernita Plump	direct	232
Moon Torian	direct	194

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

Input HISTORY MENU

SCHEMA

- accounts
- orders
- region
- sales_reps
- web_events

```
1  SELECT r.name, w.channel, COUNT(w.channel) events
2  FROM accounts a
3  JOIN web_events w ON w.account_id =  a.id
4  JOIN sales_reps s ON s.id = a.sales_rep_id
5  JOIN region r ON r.id  = s.region_id
6  GROUP BY w.channel,r.name
7  ORDER BY events DESC
```

Success! EVALUATE

Output 24 results

name	channel	events
Northeast	direct	1800
Southeast	direct	1548
West	direct	1254
Midwest	direct	696
Northeast	facebook	335
Northeast	organic	317

concept.

1. How many of the **sales reps** have more than 5 accounts that they manage?

2. How many **accounts** have more than 20 orders?

3. Which account has the most orders?

4. Which accounts spent more than 30,000 usd total across all orders?

5. Which accounts spent less than 1,000 usd total across all orders?

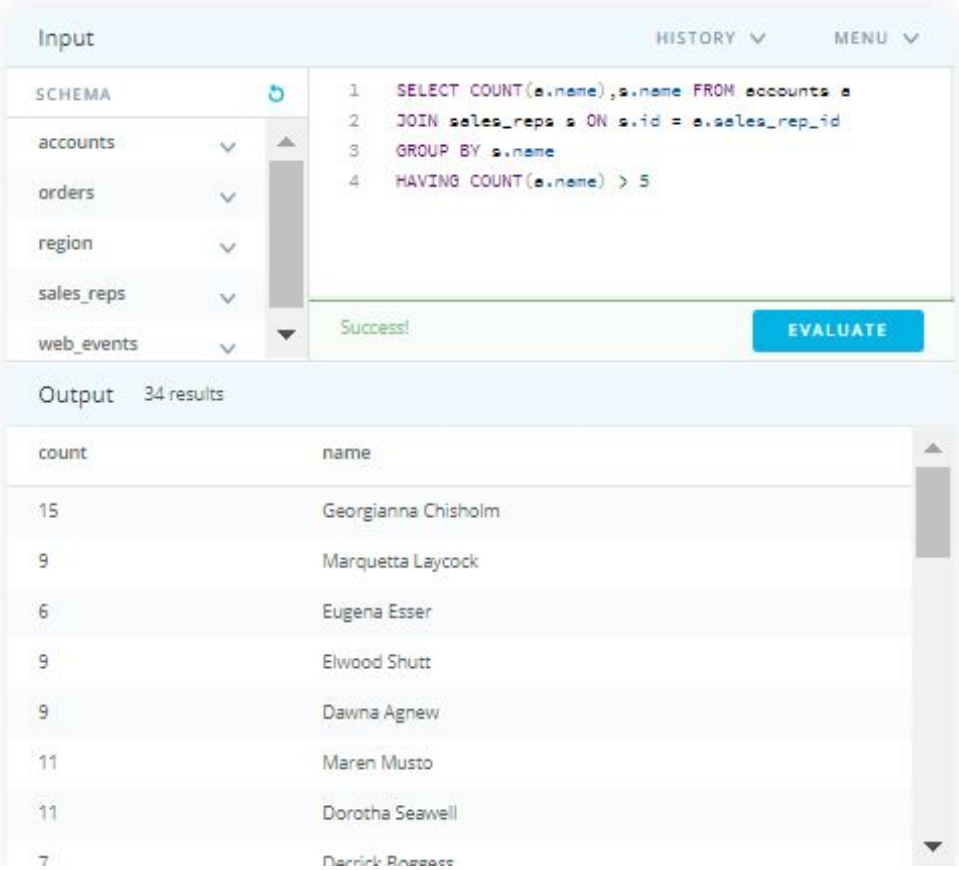
6. Which account has spent the most with us?

7. Which account has spent the least with us?

8. Which accounts used `facebook` as a **channel** to contact customers more than 6 times?

9. Which account used `facebook` most as a **channel**?

10. Which channel was most frequently used by most accounts?



1. How many of the **sales reps** have more than 5 accounts that they manage?

2. How many **accounts** have more than 20 orders?

3. Which account has the most orders?

4. Which accounts spent more than 30,000 usd total across all orders?

5. Which accounts spent less than 1,000 usd total across all orders?

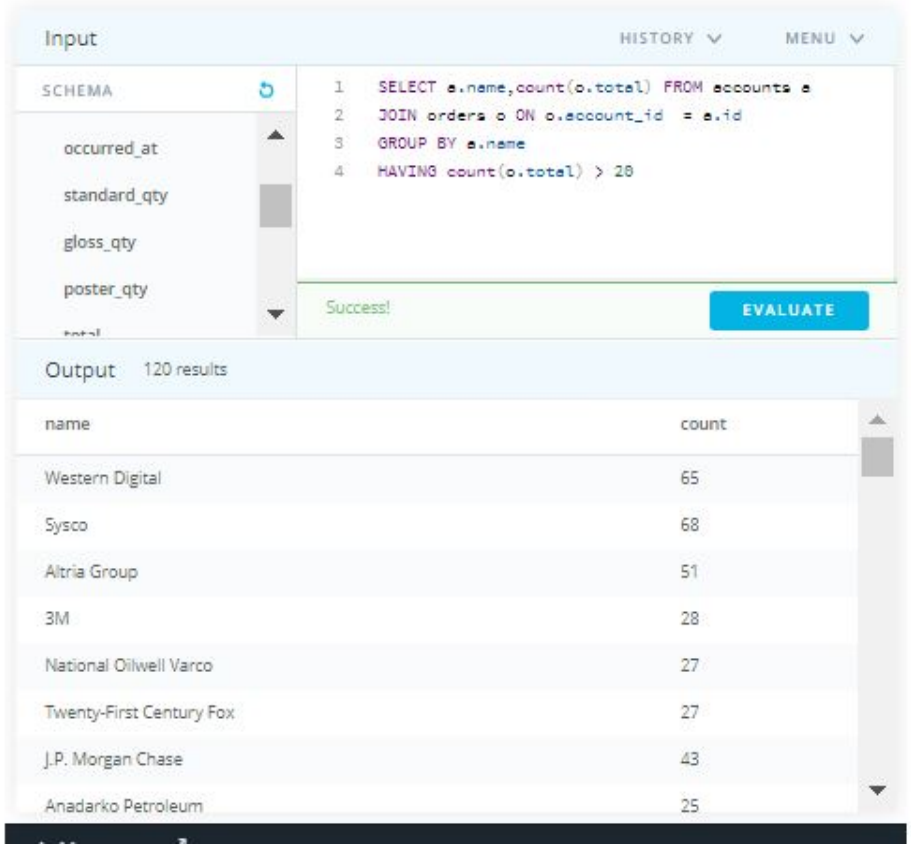
6. Which account has spent the most with us?

7. Which account has spent the least with us?

8. Which accounts used `facebook` as a **channel** to contact customers more than 6 times?

9. Which account used `facebook` most as a **channel**?

10. Which channel was most frequently used by most accounts?



3. Which account has the most orders?

4. Which accounts spent more than 30,000 usd total across all orders?

5. Which accounts spent less than 1,000 usd total across all orders?

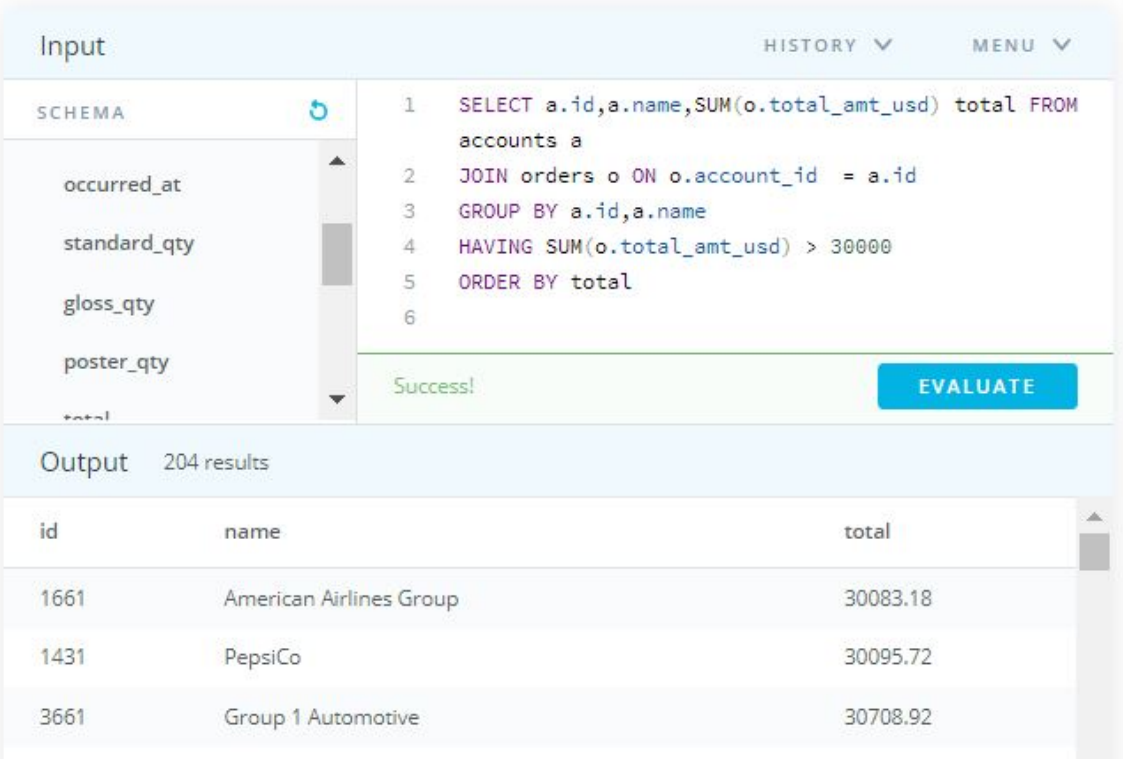
6. Which account has spent the most with us?

7. Which account has spent the least with us?

8. Which accounts used `facebook` as a **channel** to contact customers more than 6 times?

9. Which account used `facebook` most as a **channel**?

10. Which channel was most frequently used by most accounts?



5. Which accounts spent less than 1,000 usd total across all orders?

6. Which account has spent the most with us?

7. Which account has spent the least with us?

8. Which accounts used `facebook` as a **channel** to contact customers more than 6 times?

9. Which account used `facebook` most as a **channel**?

10. Which channel was most frequently used by most accounts?

Input HISTORY MENU

SCHEMA

accounts

orders

id

account_id

occurred_at

```
1  SELECT a.id,a.name,SUM(o.total_amt_usd) total FROM
   accounts a
2  JOIN orders o ON o.account_id  = a.id
3  GROUP BY a.id,a.name
4  HAVING SUM(o.total_amt_usd) < 1000
5  ORDER BY total
6
```

Success! EVALUATE

Output 3 results

id	name	total
1901	Nike	390.25
1671	Delta Air Lines	859.64
4321	Level 3 Communications	881.73

↑ Menu ⤢ Expand

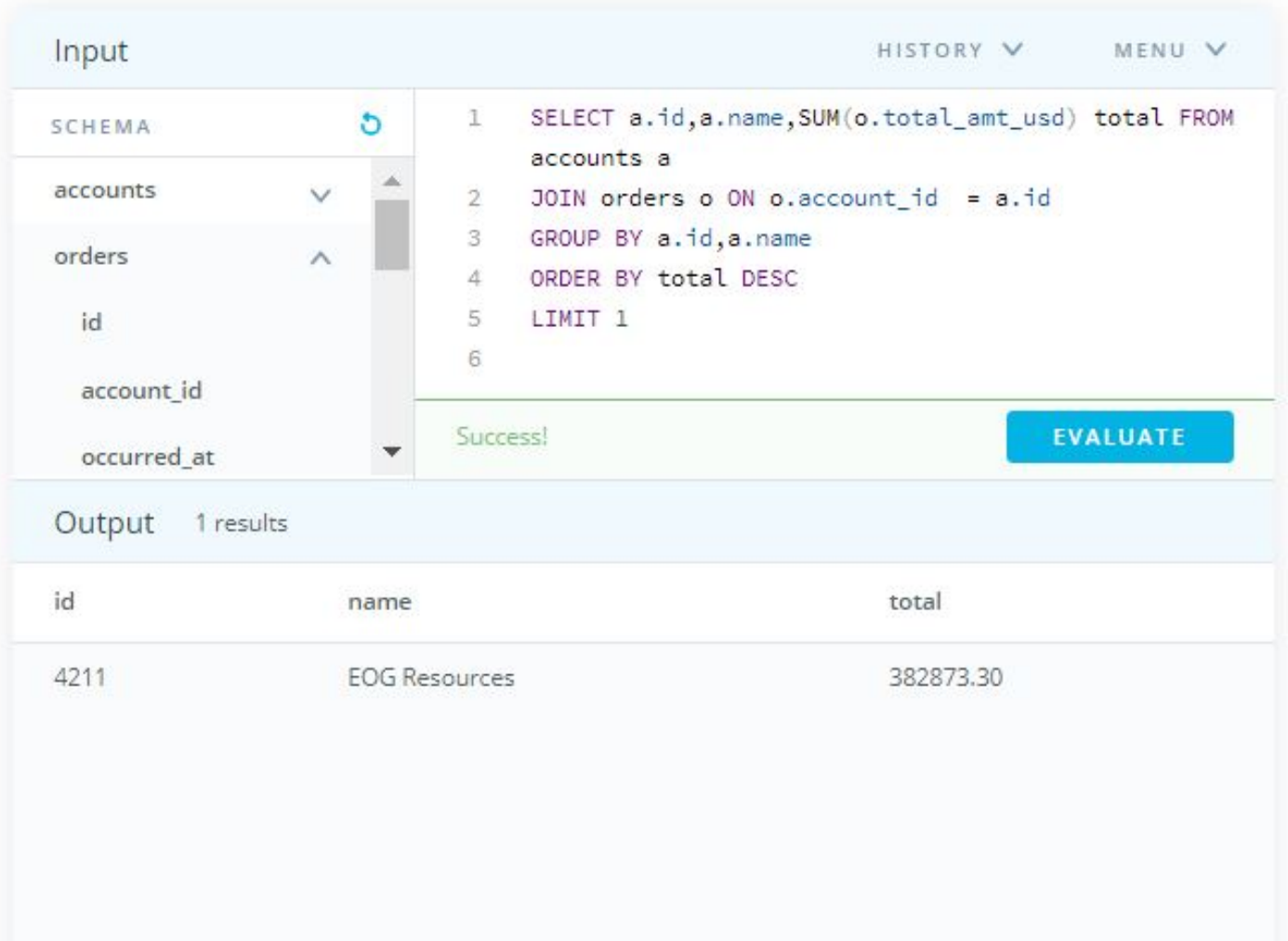
6. Which account has spent the most with us?

7. Which account has spent the least with us?

8. Which accounts used `facebook` as a **channel** to contact customers more than 6 times?

9. Which account used `facebook` most as a **channel**?

10. Which channel was most frequently used by most accounts?



7. Which account has spent the least with us?

8. Which accounts used `facebook` as a **channel** to contact customers more than 6 times?

9. Which account used `facebook` most as a **channel**?

10. Which channel was most frequently used by most accounts?

Input HISTORY MENU

SCHEMA

accounts

orders

id

account_id

occurred_at

```
1  SELECT a.id,a.name,SUM(o.total_amt_usd) total FROM
   accounts a
2  JOIN orders o ON o.account_id  = a.id
3  GROUP BY a.id,a.name
4  ORDER BY total
5  LIMIT 1
6
```

Success! EVALUATE

Output 1 results

id	name	total
1901	Nike	390.25

8. Which accounts used `facebook` as a **channel** to contact customers more than 6 times?

9. Which account used `facebook` most as a **channel**?

10. Which channel was most frequently used by most accounts?

Input HISTORY MENU

SCHEMA

web_events

- id
- account_id
- occurred_at
- channel

```
1  SELECT a.name, COUNT(*) eve ,w.channel FROM
   accounts a
2  JOIN web_events w ON w.account_id  = a.id
3  WHERE channel  = 'facebook'
4  GROUP BY a.name,w.channel
5  HAVING  COUNT(a.name) >6
6  ORDER BY eve
```

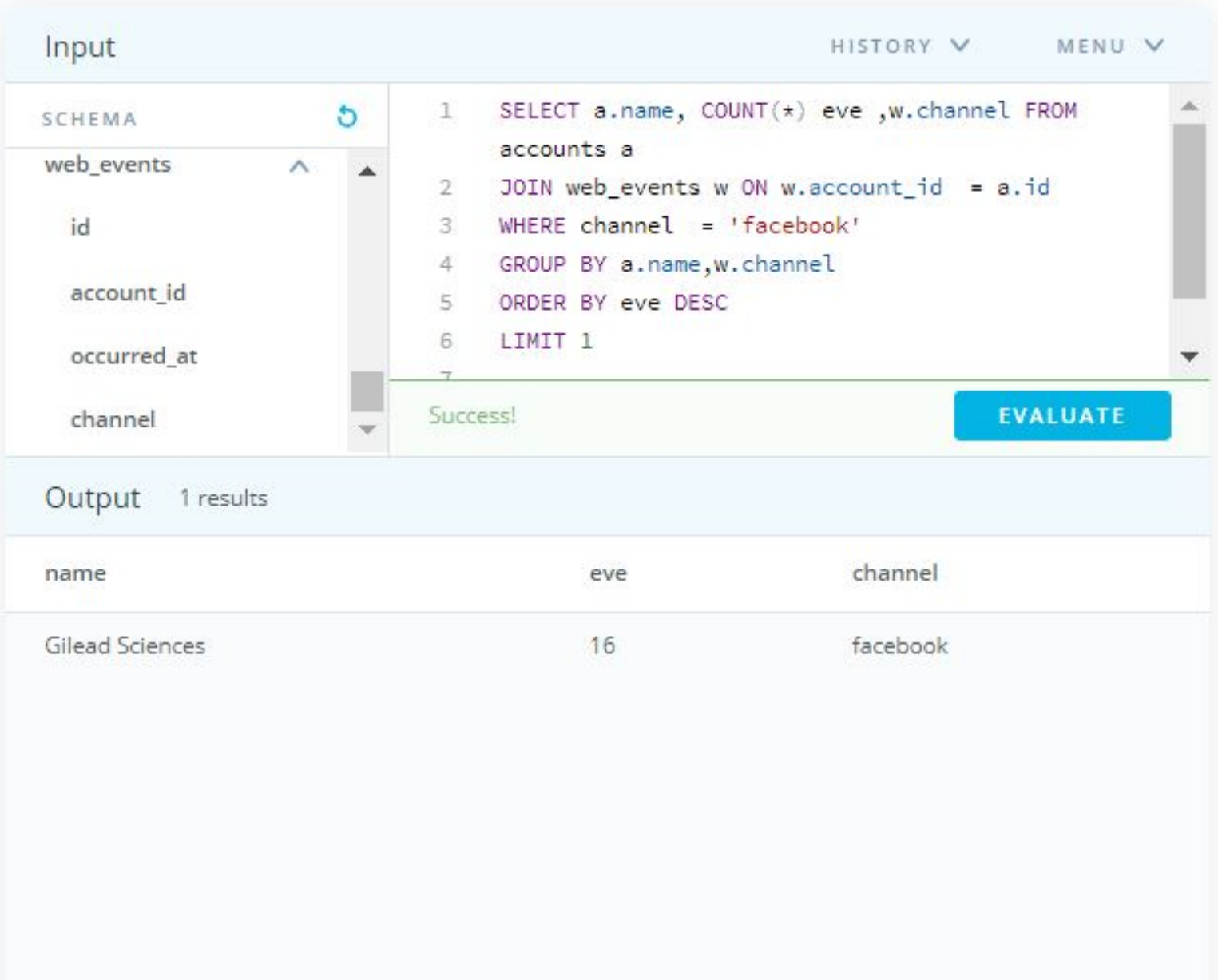
Success! EVALUATE

Output 46 results

name	eve	channel
Avis Budget Group	7	facebook
J.P. Morgan Chase	7	facebook
Best Buy	7	facebook
Honeywell International	7	facebook

9. Which account used `facebook` most as a **channel**?

10. Which channel was most frequently used by most accounts?



10. Which channel was most frequently used by most accounts?

Input HISTORY MENU

SCHEMA

web_events
- id
- account_id
- occurred_at
- channel

```
1  SELECT a.id, COUNT(*) eve,w.channel FROM accounts
   a
2  JOIN web_events w ON w.account_id  = a.id
3  GROUP BY a.id,w.channel
4  ORDER BY eve DESC
5
6
```

EVALUATE

Output 1509 results

id	eve	channel
3411	52	direct
1601	51	direct
2731	51	direct
2051	49	direct
2351	48	direct
3471	48	direct

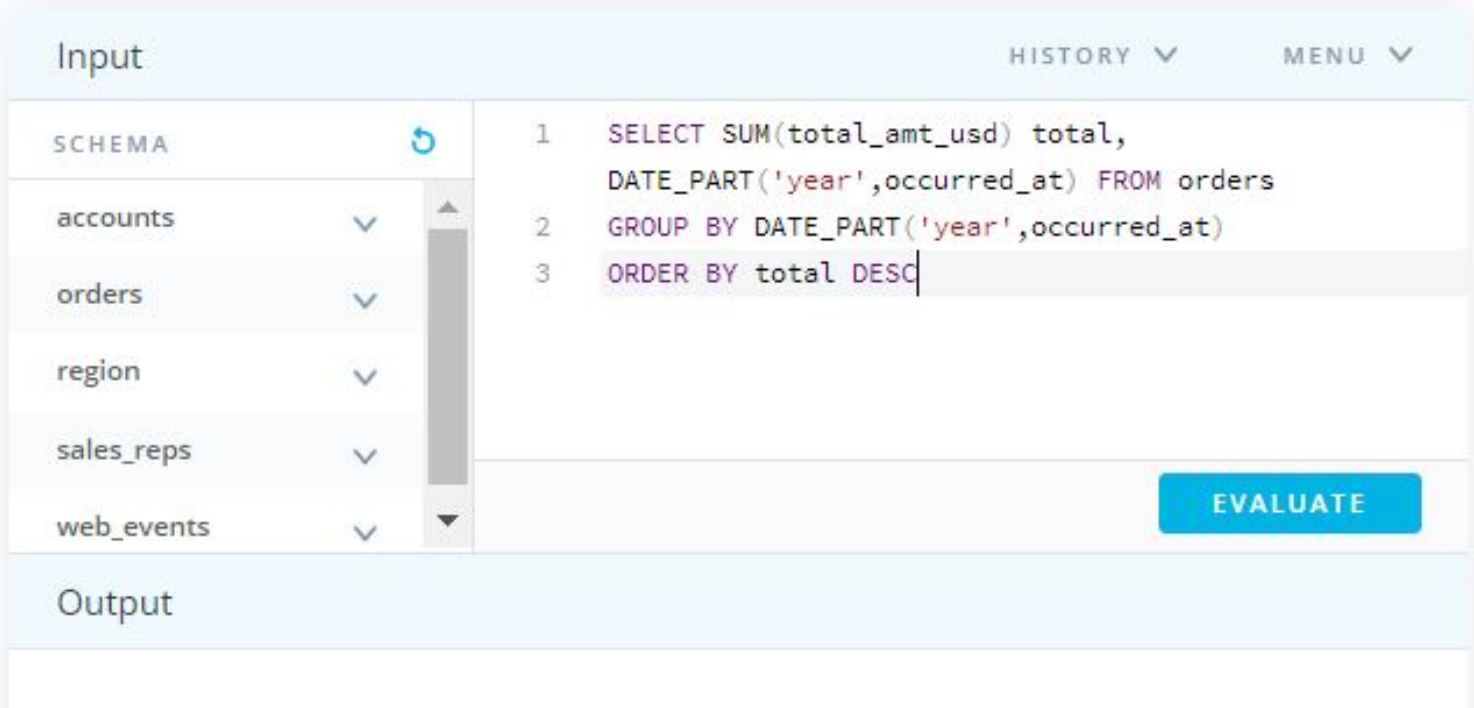
1. Find the sales in terms of total dollars for all orders in each `year`, ordered from greatest to least. Do you notice any trends in the yearly sales totals?

2. Which **month** did Parch & Posey have the greatest sales in terms of total dollars? Are all months evenly represented by the dataset?

3. Which **year** did Parch & Posey have the greatest sales in terms of total number of orders? Are all years evenly represented by the dataset?

4. Which **month** did Parch & Posey have the greatest sales in terms of total number of orders? Are all months evenly represented by the dataset?

5. In which **month** of which **year** did `Walmart` spend the most on gloss paper in terms of dollars?

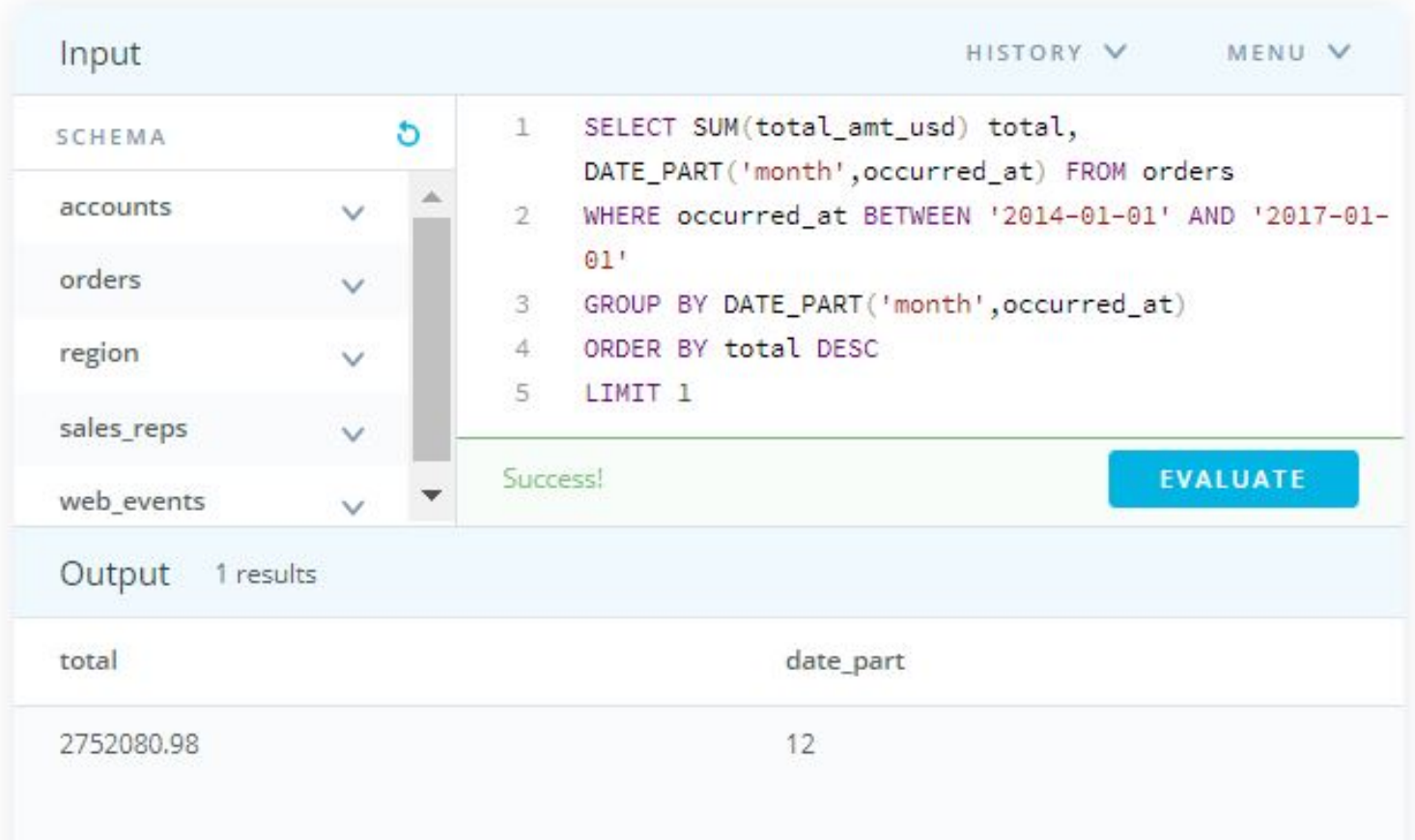


2. Which **month** did Parch & Posey have the greatest sales in terms of total dollars? Are all months evenly represented by the dataset?

3. Which **year** did Parch & Posey have the greatest sales in terms of total number of orders? Are all years evenly represented by the dataset?

4. Which **month** did Parch & Posey have the greatest sales in terms of total number of orders? Are all months evenly represented by the dataset?

5. In which **month** of which **year** did `Walmart` spend the most on gloss paper in terms of dollars?



3. Which **year** did Parch & Posey have the greatest sales in terms of total number of orders? Are all years evenly represented by the dataset?

4. Which **month** did Parch & Posey have the greatest sales in terms of total number of orders? Are all months evenly represented by the dataset?

5. In which **month** of which **year** did `Walmart` spend the most on gloss paper in terms of dollars?

Input HISTORY MENU

SCHEMA

- accounts
- orders
- region
- sales_reps
- web_events

```
1  SELECT count(total) total,
   DATE_PART('year',occurred_at) FROM orders
2  GROUP BY DATE_PART('year',occurred_at)
3  ORDER BY total DESC
```

Success! EVALUATE

Output 5 results

total	date_part
3757	2016
1725	2015
1306	2014
99	2013
25	2017

4. Which **month** did Parch & Posey have the greatest sales in terms of total number of orders? Are all months evenly represented by the dataset?

5. In which **month** of which **year** did `Walmart` spend the most on gloss paper in terms of dollars?

Input HISTORY MENU

SCHEMA

- accounts
- orders
- region
- sales_reps
- web_events

```
1  SELECT count(total) total,
   DATE_PART('month',occurred_at) FROM orders
2  WHERE occurred_at BETWEEN '2014-01-01' AND '2017-01-
   01'
3  GROUP BY DATE_PART('month',occurred_at)
4  ORDER BY total DESC
```

Success! EVALUATE

Output 12 results

total	date_part
783	12
713	11
675	10
603	8
602	9
571	7
527	6
518	5

5. In which **month** of which **year** did `Walmart` spend the most on gloss paper in terms of dollars?

Input HISTORY MENU

SCHEMA

- accounts
- orders
- region
- sales_reps
- web_events

```
1  SELECT SUM(o.gloss_amt_usd) total,
   DATE_TRUNC('month',o.occurred_at),a.name FROM
   accounts a
2  JOIN orders o ON o.account_id = a.id
3  WHERE a.name = 'Walmart'
4  GROUP BY DATE_TRUNC('month',o.occurred_at),a.name
5  ORDER BY total DESC
6  LIMIT 1
```

Success! EVALUATE

Output 1 results

total	date_trunc	name
9257.64	2016-05-01T00:00:00.000Z	Walmart

1. 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 $3000 or more, or smaller than $3000.

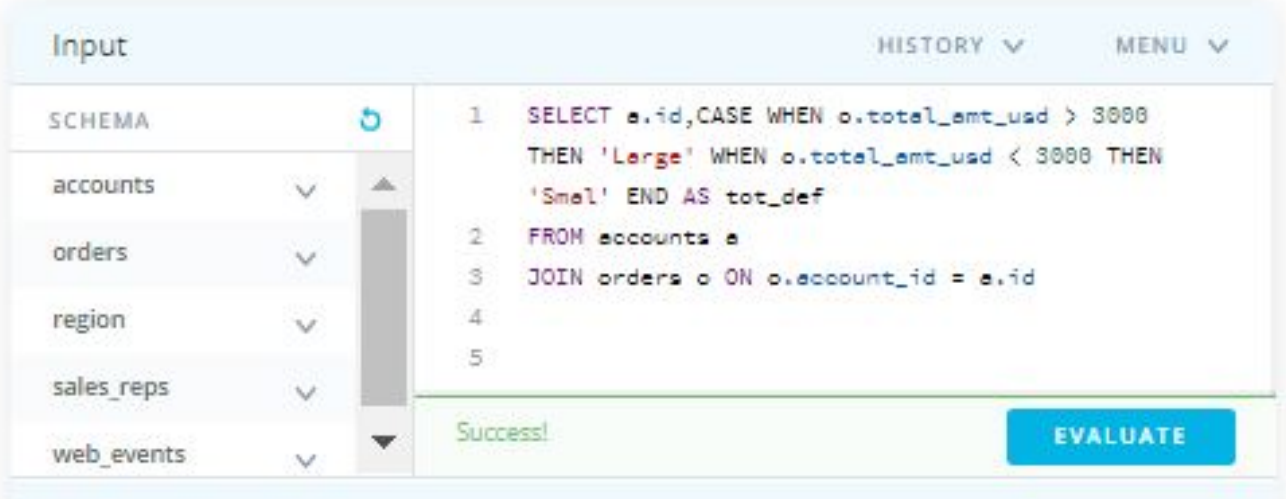
2. Write a query to display the number of orders in each of three categories, based on the `total` number of items in each order. The three categories are: 'At Least 2000', 'Between 1000 and 2000' and 'Less than 1000'.

3. We would like to understand 3 different levels of customers based on the amount associated with their purchases. The top level includes anyone with a Lifetime Value (total sales of all orders) `greater than 200,000` usd. The second level is between `200,000 and 100,000` usd. The lowest level is anyone `under 100,000` usd. Provide a table that includes the **level** associated with each **account**. You should provide the **account name**, the **total sales of all orders** for the customer, and the **level**. Order with the top spending customers listed first.

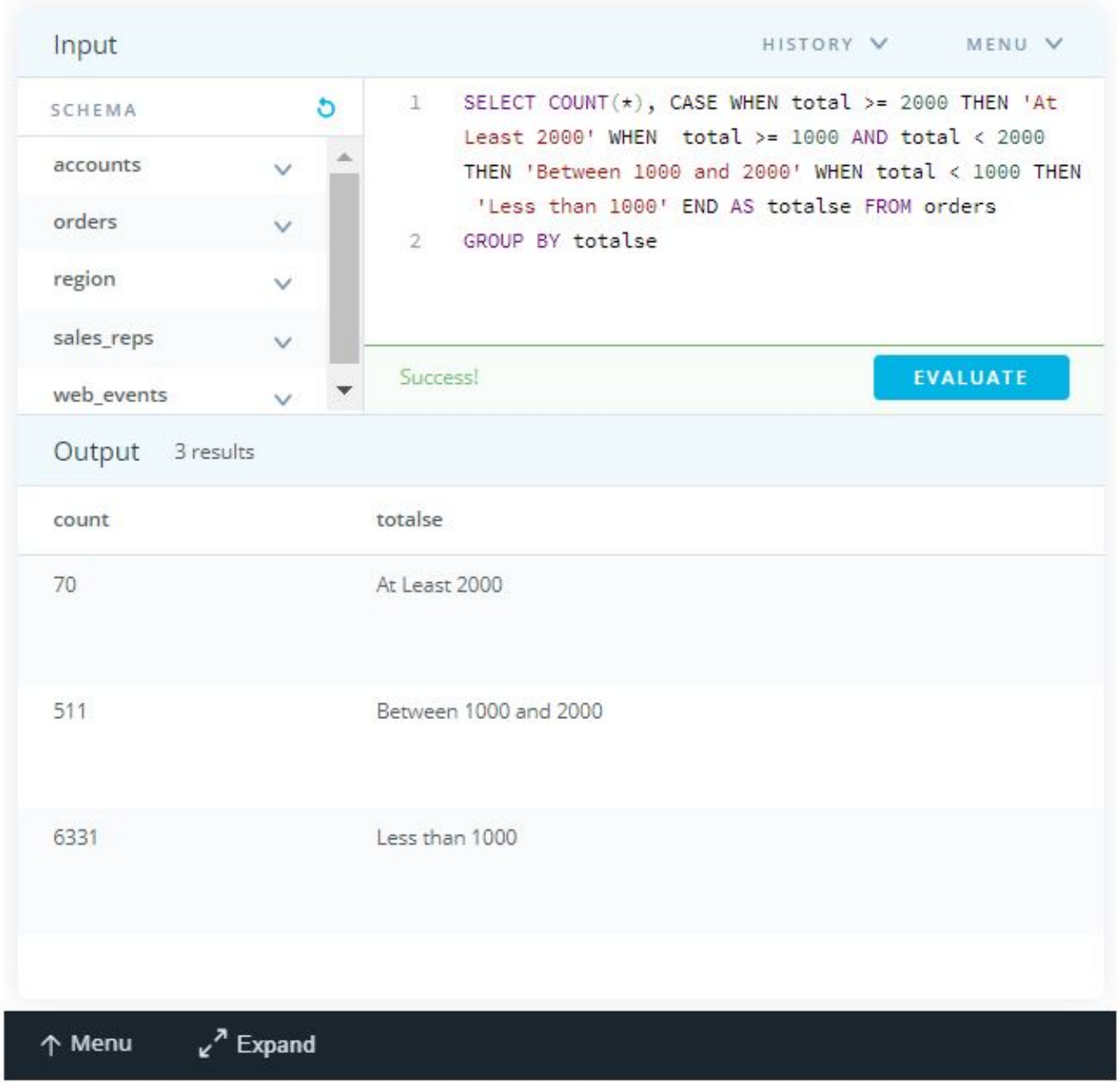
4. We would now like to perform a similar calculation to the first, but we want to obtain the total amount spent by customers only in `2016` and `2017`. Keep the same **levels** as in the previous question. Order with the top spending customers listed first.

5. 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.

6. The previous didn't account for the middle, nor the dollar amount associated with the sales. Management decides they want to see these characteristics represented as well. We would like to identify top performing **sales reps**, which are sales reps associated with more than `200` orders or more than `750000` in total sales. The `middle` group has any **rep** with more than 150 orders or `500000` in sales. Create a table with the **sales rep name**, the total number of orders, total sales across all orders, and a column with `top`, `middle`, or `low` depending on this criteria. Place the top sales people based on dollar amount of sales first in your final table. You might see a few upset sales people by this criteria!



2. 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'.


Input
HISTORY
MENU
SCHEMA
accounts
orders
region
sales_reps
web_events
1 SELECT COUNT(*), CASE WHEN total >= 2000 THEN 'At Least 2000' WHEN total >= 1000 AND total < 2000 THEN 'Between 1000 and 2000' WHEN total < 1000 THEN 'Less than 1000' END AS totalse FROM orders
2 GROUP BY totalse
Success!
EVALUATE
Output 3 results
count
totalse
70
At Least 2000
511
Between 1000 and 2000
6331
Less than 1000
↑ Menu
Expand

3. We would like to understand 3 different levels of customers based on the amount associated with their purchases. The top level includes anyone with a Lifetime Value (total sales of all orders) `greater than 200,000` usd. The second level is between `200,000 and 100,000` usd. The lowest level is anyone `under 100,000` usd. Provide a table that includes the **level** associated with each **account**. You should provide the **account name**, the **total sales of all orders** for the customer, and the **level**. Order with the top spending customers listed first.

Input

HISTORY MENU

SCHEMA

occurred_at

standard_qty

gloss_qty

poster_qty

total

```
1  SELECT a.name,CASE WHEN SUM(o.total_amt_usd) >
   200000 THEN ' top level' WHEN
   SUM(o.total_amt_usd) > 100000 THEN 'second level'
   WHEN SUM(o.total_amt_usd) < 100000 THEN 'lowest
   level' END AS classification
2  FROM accounts a
3  JOIN orders o ON o.account_id = a.id
4  GROUP BY a.name
```

Success!

EVALUATE

Output 350 results

name	classification
EOG Resources	top level
Archer Daniels Midland	top level
Fluor	top level
Freddie Mac	top level
Sysco	top level
Pacific Life	top level
Western Digital	top level
Wells Fargo	top level

4. 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 **level**s as in the previous question. Order with the top spending customers listed first.

5. 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.

6. The previous didn't account for the middle, nor the dollar amount associated with the sales. Management decides they want to see these characteristics represented as well. We would like to identify top performing **sales reps**, which are sales reps associated with more than `200` orders or more than `750000` in total sales. The `middle` group has any **rep** with more than 150 orders or `500000` in sales. Create a table with the **sales rep name**, the total number of orders, total sales across all orders, and a column with `top`, `middle`, or `low` depending on this criteria. Place the top sales people based on dollar amount of sales first in your final table. You might see a few upset sales people by this criteria!

Input HISTORY MENU

```
1  SELECT a.name, CASE WHEN SUM(total_amt_usd) > 200000 THEN 'Top Level' WHEN
   SUM(total_amt_usd) > 100000 AND SUM(total_amt_usd)< 200000 THEN 'second level'
   WHEN SUM(total_amt_usd) < 100000 THEN 'Lowest level' END AS Levels FROM accounts
   a
2  JOIN orders o ON o.account_id = a.id
3  WHERE o.occurred_at > '2015-12-31'
4  GROUP BY a.name
```

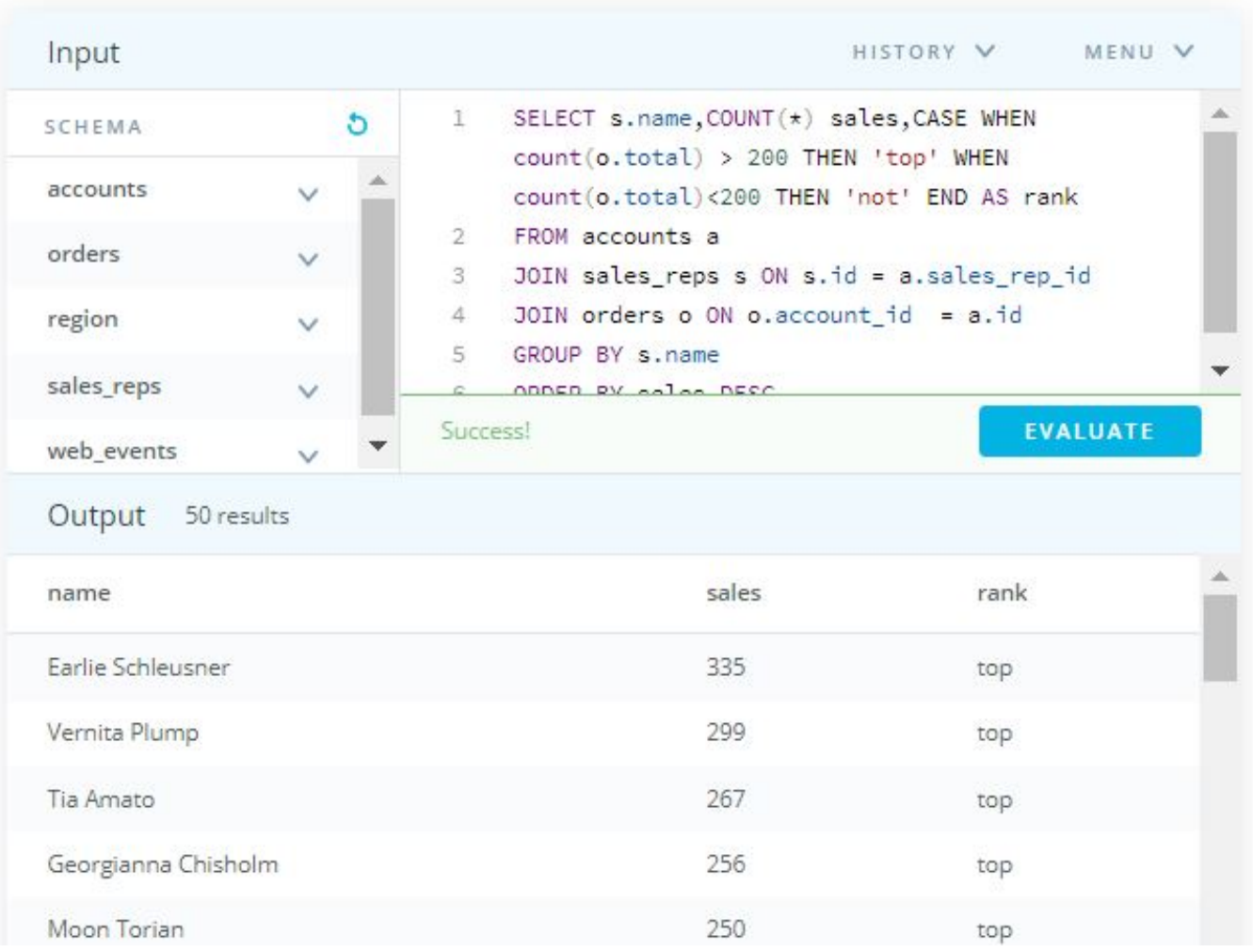
Success! EVALUATE

Output 322 results

name	levels
Boeing	Lowest level
Western Digital	Lowest level

5. 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.

6. The previous didn't account for the middle, nor the dollar amount associated with the sales. Management decides they want to see these characteristics represented as well. We would like to identify top performing **sales reps**, which are sales reps associated with more than `200` orders or more than `750000` in total sales. The `middle` group has any **rep** with more than 150 orders or `500000` in sales. Create a table with the **sales rep name**, the total number of orders, total sales across all orders, and a column with `top`, `middle`, or `low` depending on this criteria. Place the top sales people based on dollar amount of sales first in your final table. You might see a few upset sales people by this criteria!



6. The previous didn't account for the middle, nor the dollar amount associated with the sales. Management decides they want to see these characteristics represented as well. We would like to identify top performing **sales reps**, which are sales reps associated with more than `200` orders or more than `750000` in total sales. The `middle` group has any **rep** with more than 150 orders or `500000` in sales. Create a table with the **sales rep name**, the total number of orders, total sales across all orders, and a column with `top`, `middle`, or `low` depending on this criteria. Place the top sales people based on dollar amount of sales first in your final table. You might see a few upset sales people by this criteria!

Input HISTORY MENU

```
1  SELECT s.name,COUNT(*) sales, SUM(total_amt_usd) summ,
2  CASE
3  WHEN count(o.total) > 200 AND SUM(total_amt_usd) > 750000 THEN 'top'
4  WHEN count(o.total)> 150 AND SUM(total_amt_usd)>500000  THEN 'middle'
5  ELSE 'not' END AS rank
6  FROM accounts a
7  JOIN sales_reps s ON s.id = a.sales_rep_id
8  JOIN orders o ON o.account_id = a.id
```

Success! EVALUATE

Output 50 results

name	sales	summ	rank
Earlie Schleusner	335	1098137.72	top
Tia Amato	267	1010690.60	top
Vernita Plump	299	934212.93	top
Georgianna Chisholm	256	886244.12	top
Dorotha Seawell	208	766935.04	top