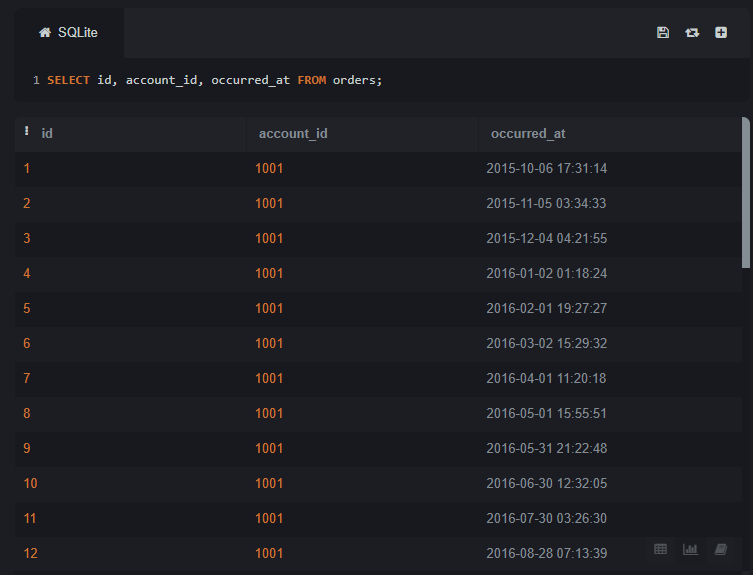
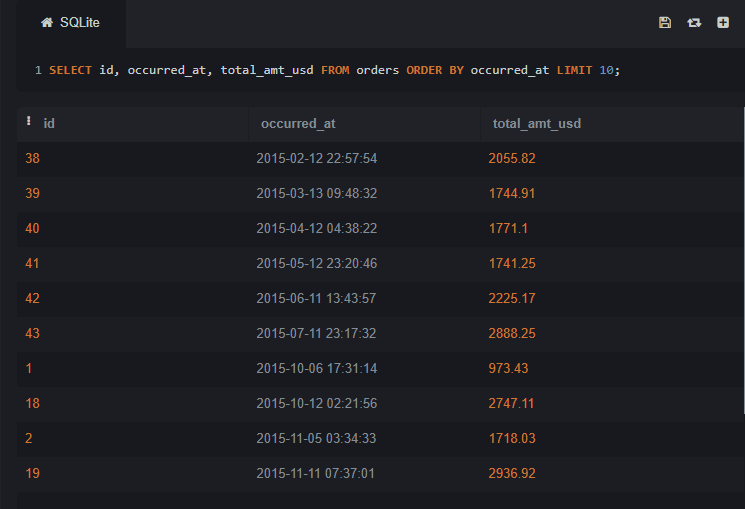
**SQL Queries**

**Lab 06**

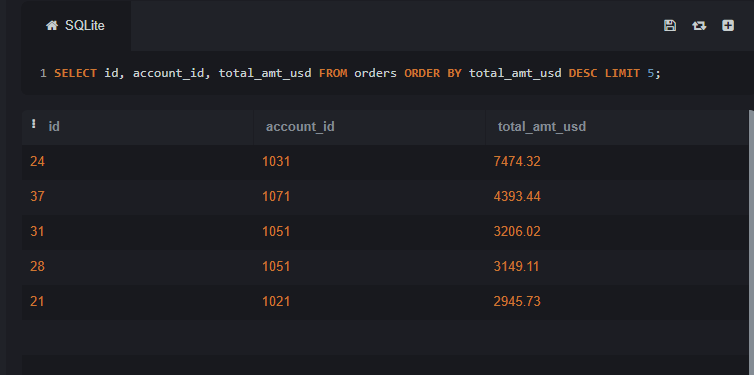
SELECT id, account\_id, occurred\_at FROM orders;



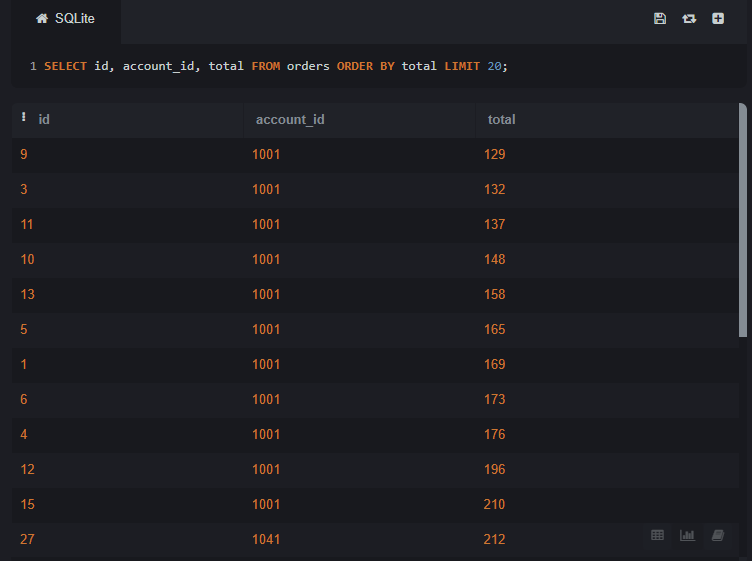
SELECT id, occurred\_at, total\_amt\_usd FROM orders ORDER BY occurred\_at LIMIT 10;



SELECT id, account\_id, total\_amt\_usd FROM orders ORDER BY total\_amt\_usd DESC LIMIT 5;



SELECT id, account\_id, total FROM orders ORDER BY total LIMIT 20;



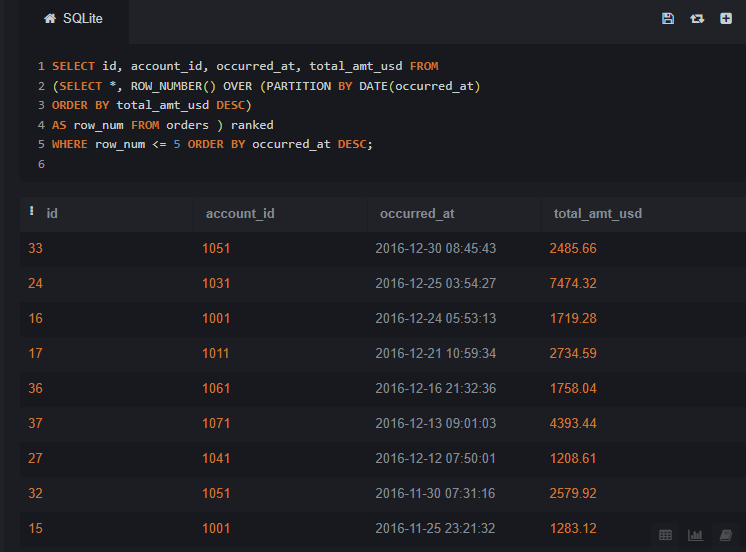
SELECT id, account\_id, occurred\_at, total\_amt\_usd FROM

(SELECT \*, ROW\_NUMBER() OVER (PARTITION BY DATE(occurred\_at)

ORDER BY total\_amt\_usd DESC)

AS row\_num FROM orders ) ranked

WHERE row\_num <= 5 ORDER BY occurred\_at DESC;



SELECT id, account\_id, occurred\_at, total\_amt\_usd

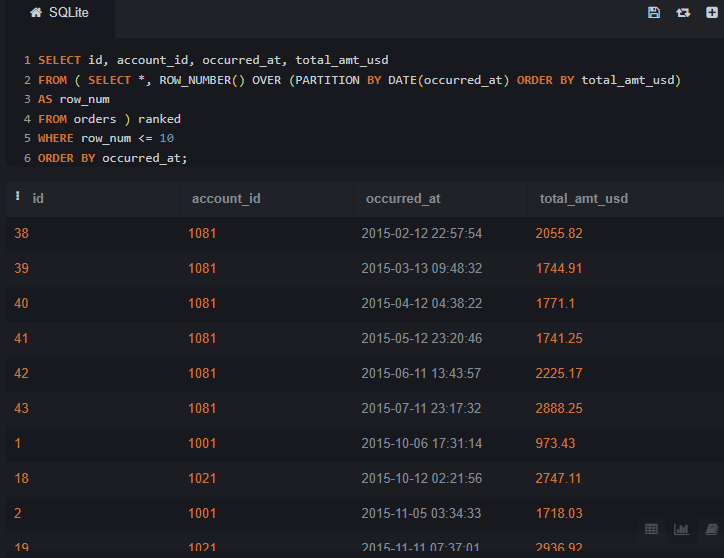
FROM ( SELECT \*, ROW\_NUMBER() OVER (PARTITION BY DATE(occurred\_at) ORDER BY total\_amt\_usd)

AS row\_num

FROM orders ) ranked

WHERE row\_num <= 10

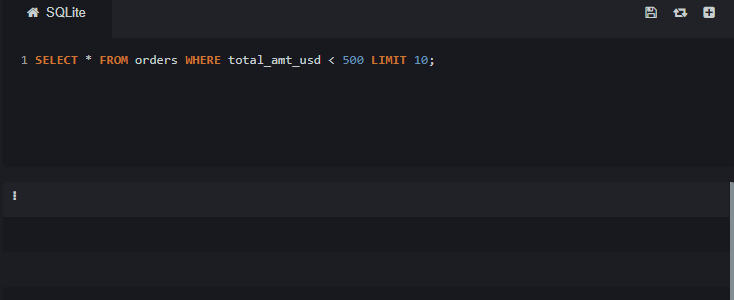
ORDER BY occurred\_at;



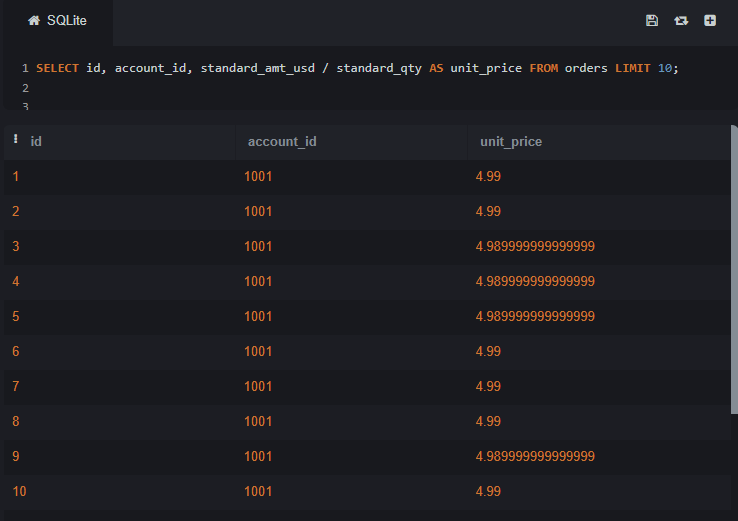
SELECT \* FROM orders WHERE gloss\_amt\_usd >= 1000 LIMIT 5;



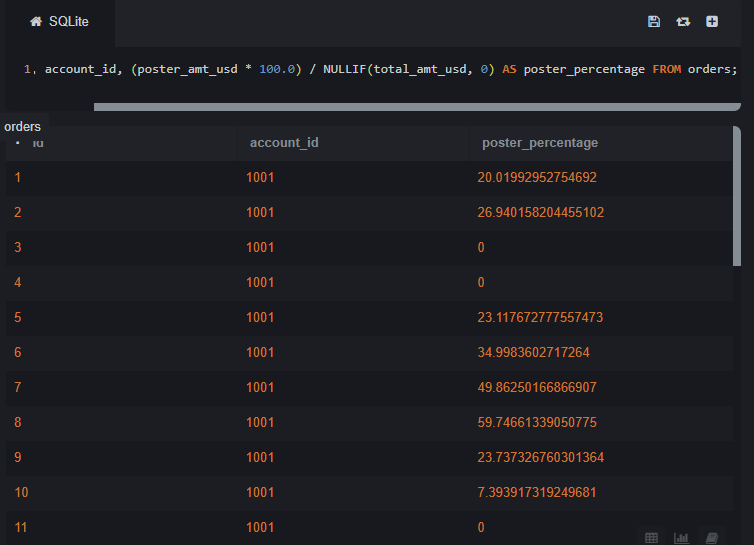
SELECT \* FROM orders WHERE total\_amt\_usd < 500 LIMIT 10;

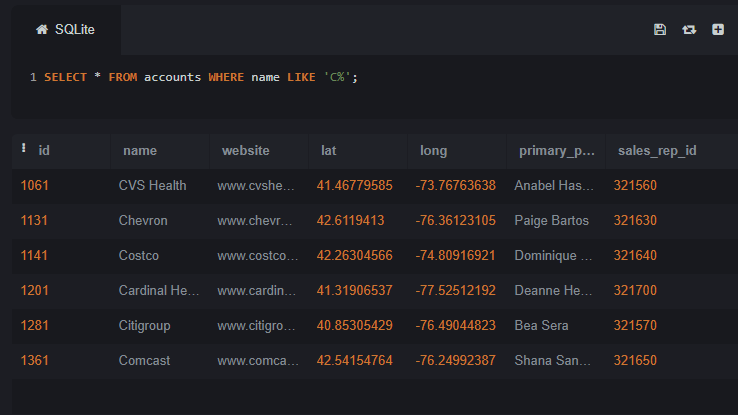


SELECT id, account\_id, standard\_amt\_usd / standard\_qty AS unit\_price FROM orders LIMIT 10;

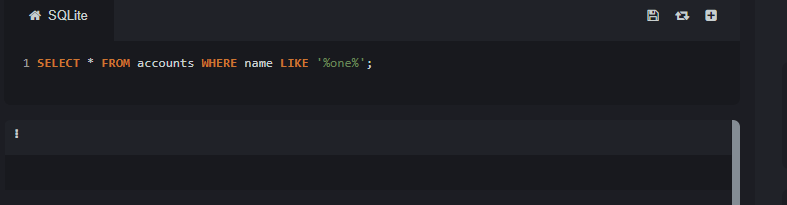


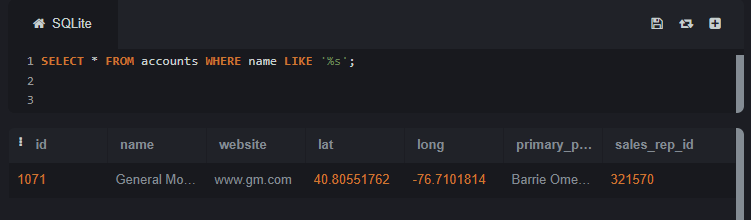
SELECT id, account\_id, (poster\_amt\_usd \* 100.0) / NULLIF(total\_amt\_usd, 0) AS poster\_percentage FROM orders;



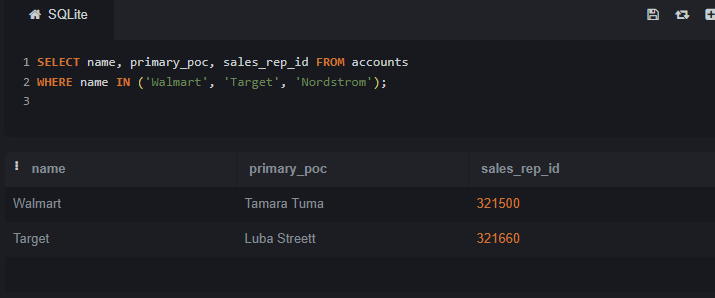
SELECT \* FROM accounts WHERE name LIKE 'C%';

SELECT \* FROM accounts WHERE name LIKE '%one%';



SELECT \* FROM accounts WHERE name LIKE '%s';

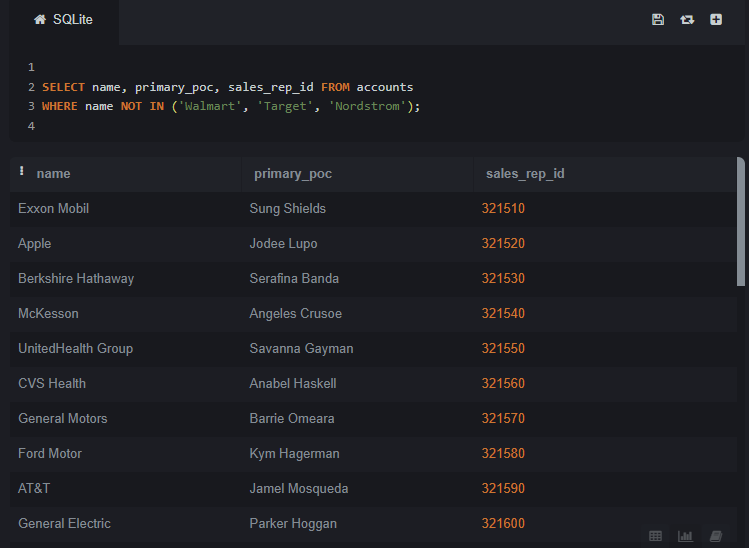
SELECT name, primary\_poc, sales\_rep\_id FROM accounts WHERE name IN ('Walmart', 'Target', 'Nordstrom');



SELECT \* FROM web\_events WHERE channel IN ('organic', 'adwords');

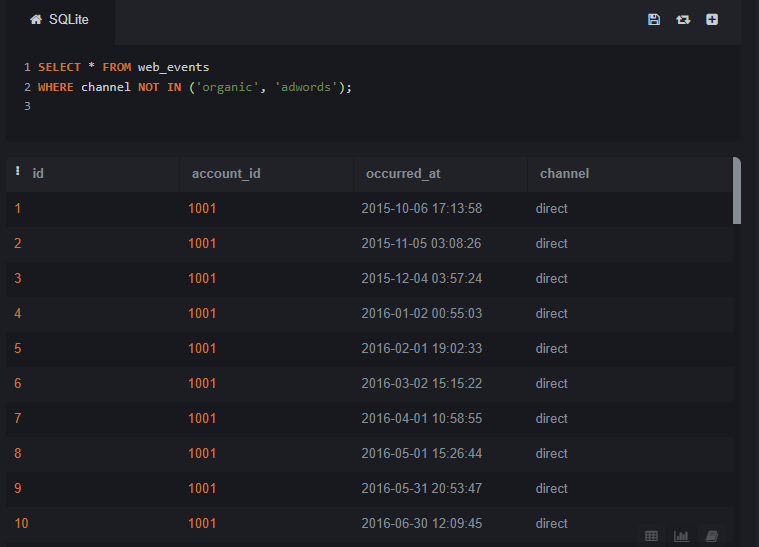
SELECT name, primary\_poc, sales\_rep\_id FROM accounts

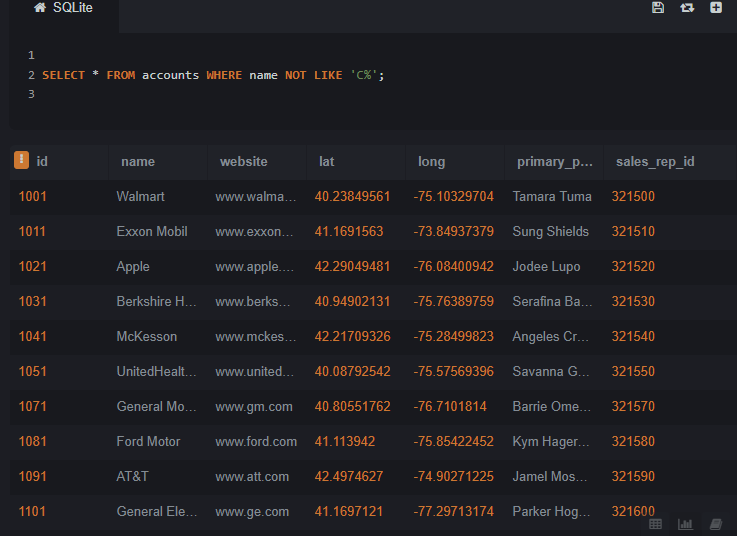
WHERE name NOT IN ('Walmart', 'Target', 'Nordstrom');

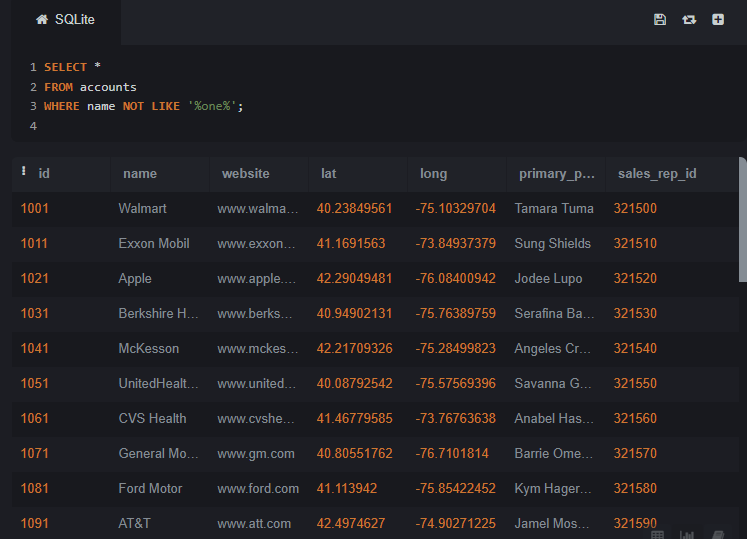


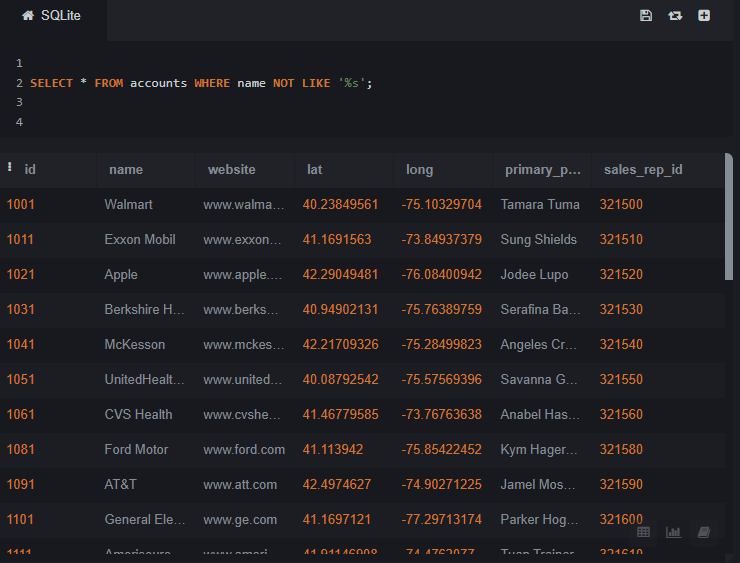
SELECT \* FROM web\_events

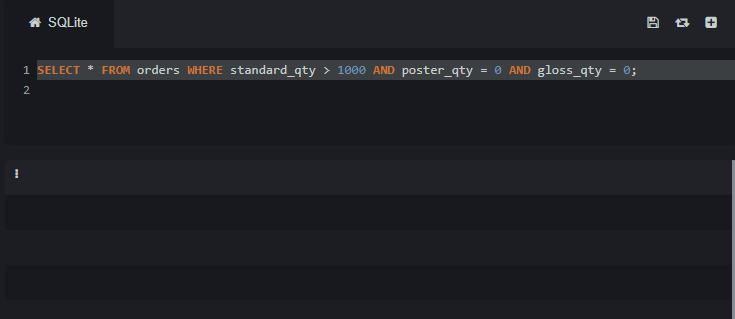
WHERE channel NOT IN ('organic', 'adwords');



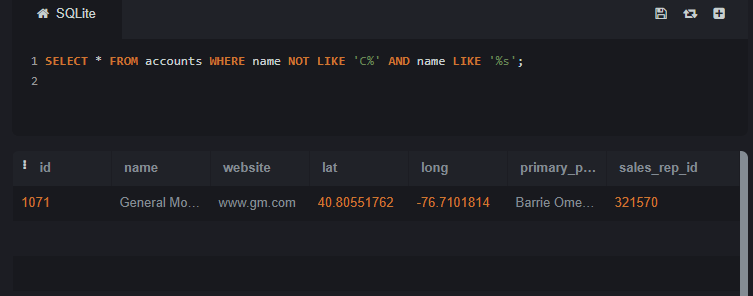
SELECT \* FROM accounts WHERE name NOT LIKE 'C%';

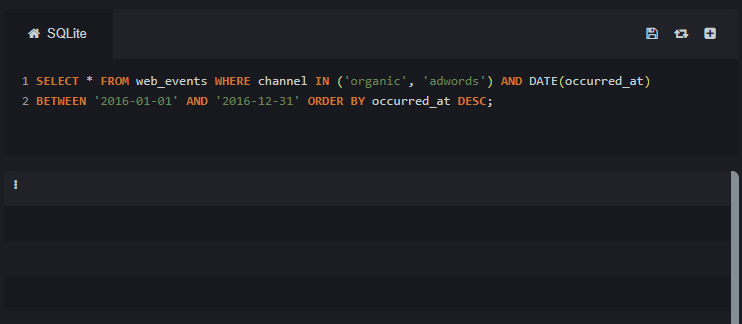
SELECT \* FROM acounts WHERE name NOT LIKE '%one%';

SELECT \* FROM accounts WHERE name NOT LIKE '%s';

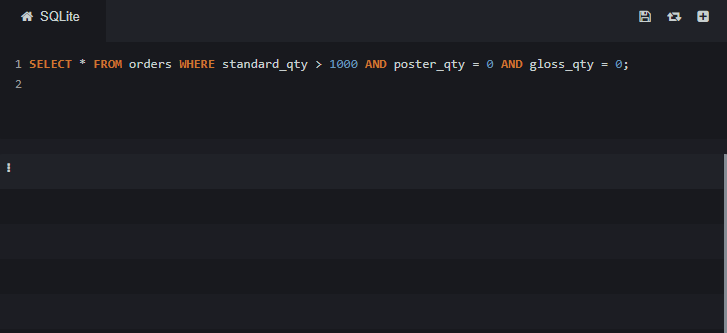
SELECT \* FROM orders WHERE standard\_qty > 1000 AND poster\_qty = 0 AND gloss\_qty = 0;

SELECT \* FROM accounts WHERE name NOT LIKE 'C%' AND name LIKE '%s';

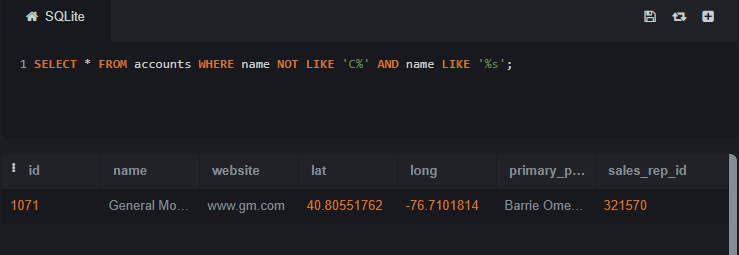


SELECT \* FROM web\_events WHERE channel IN ('organic', 'adwords') AND DATE(occurred\_at) BETWEEN '2016-01-01' AND '2016-12-31' ORDER BY occurred\_at DESC;

SELECT \* FROM orders WHERE standard\_qty > 1000 AND poster\_qty = 0 AND gloss\_qty = 0;



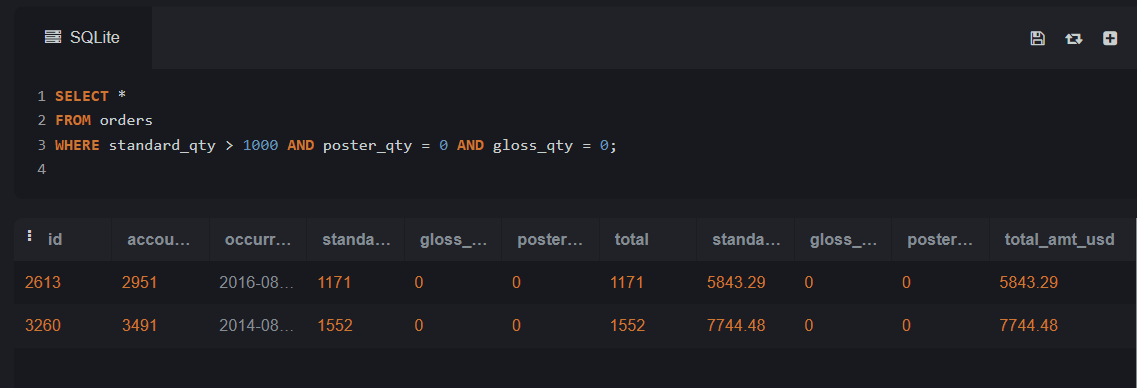
SELECT \* FROM accounts WHERE name NOT LIKE 'C%' AND name LIKE '%s';

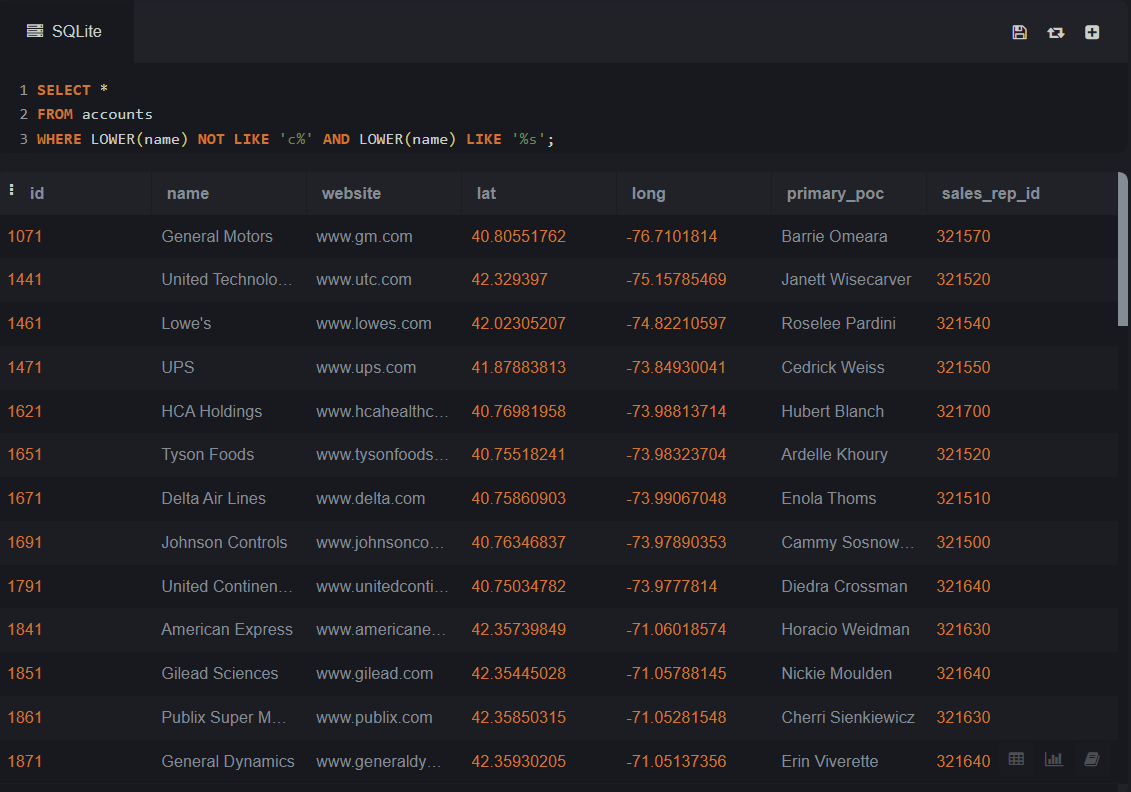


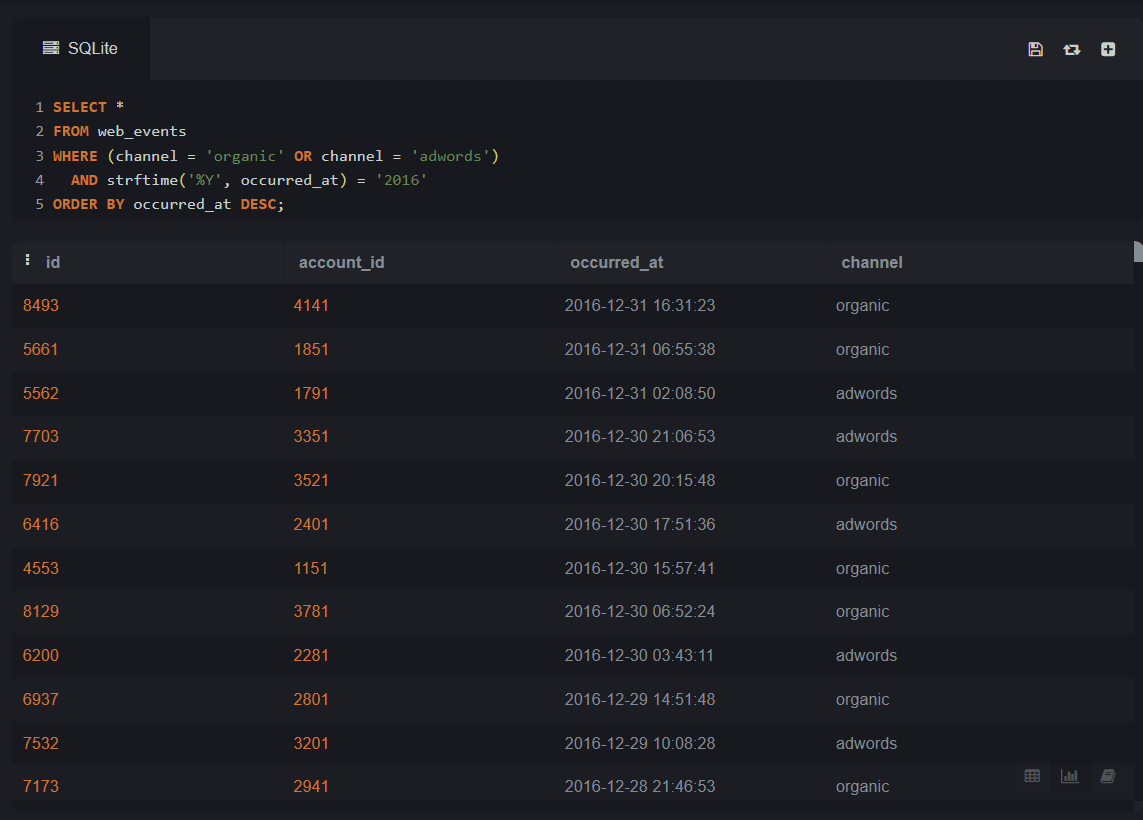
Updated

Task#01 – Basic SQL -Run the following queries:

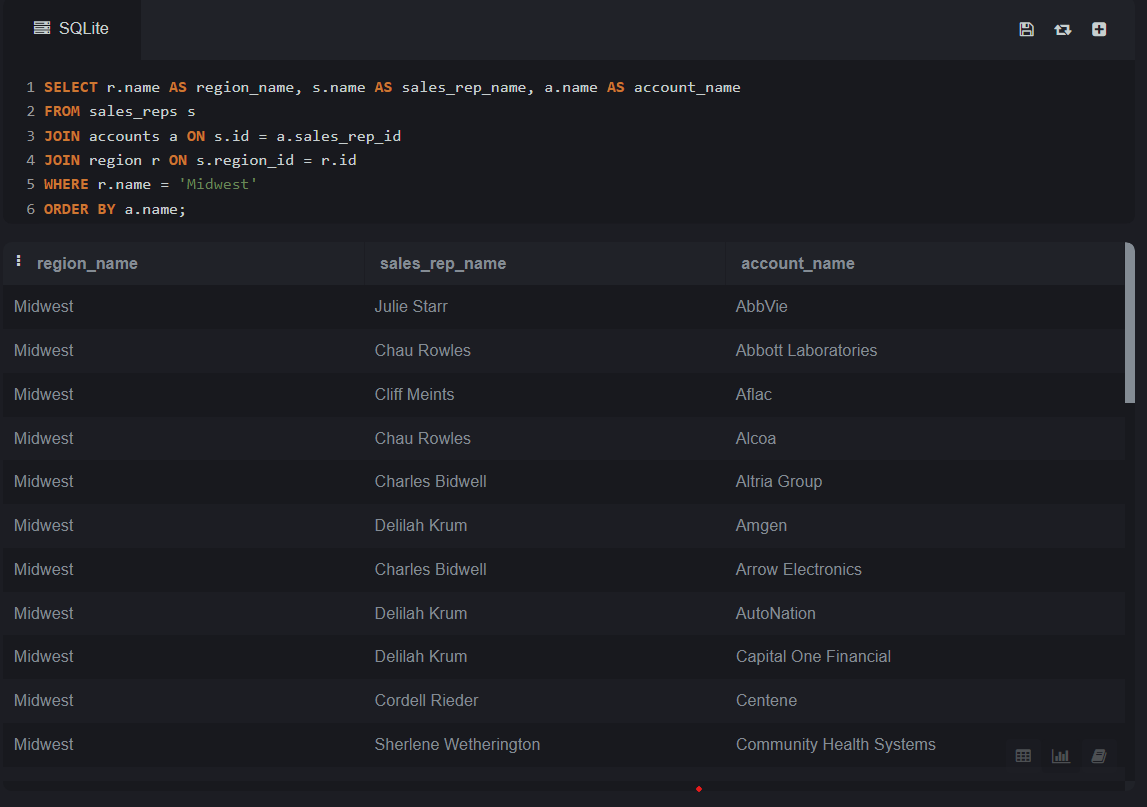
Write a query that returns all the orders where the standard\_qty is over 1000, the poster\_qty is 0, and the gloss\_qty is 0.



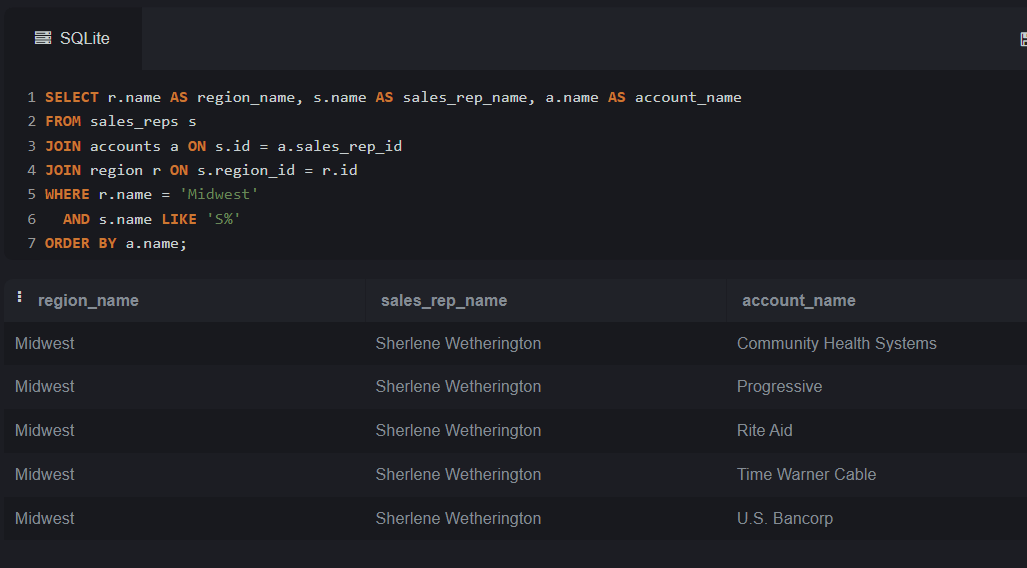
Using the accounts table find all the companies whose names do not start with 'C' and end with 's'

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

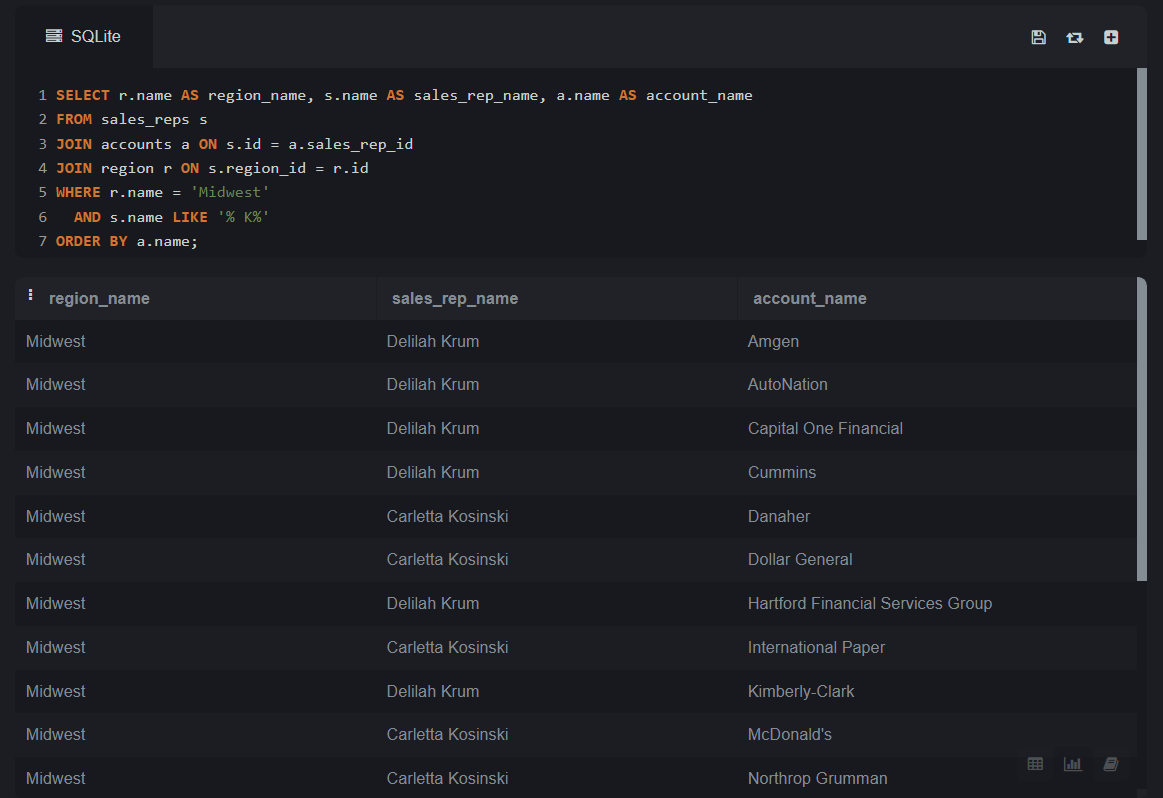
Task#02 – Joins -Run the following queries:

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

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.

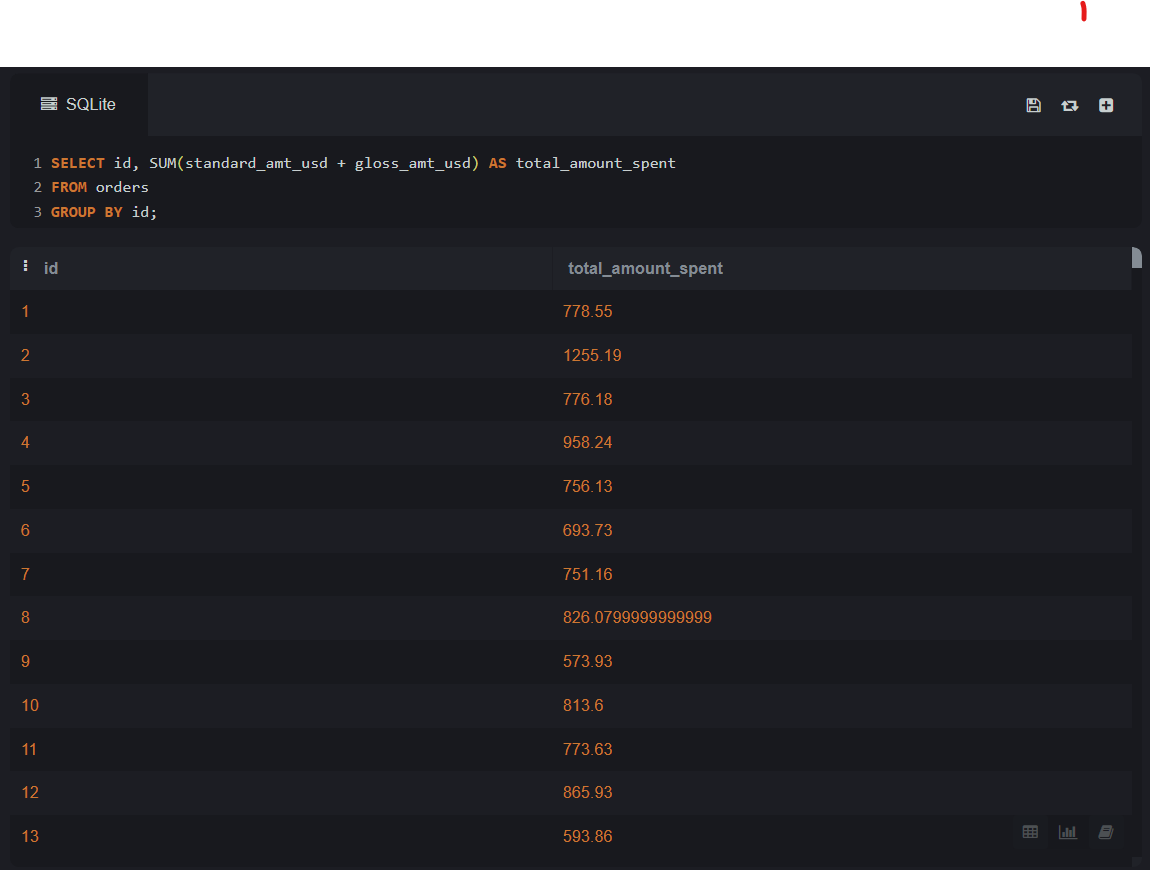


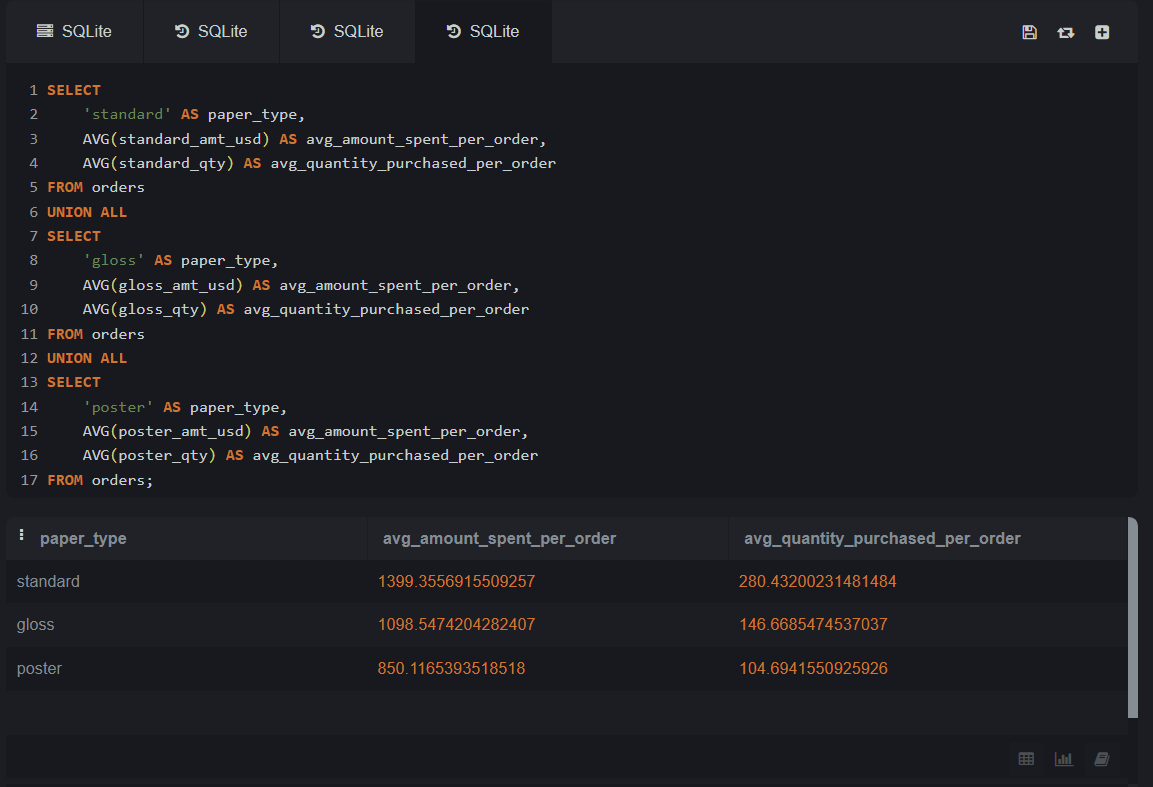
Provide a table that provides the region for each sales\_rep along with their associated accounts. This time only for accounts where the sales rep has a last name starting with K and in the Midwest region. Your final table should include three columns: the region name, the sales rep name, and the account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



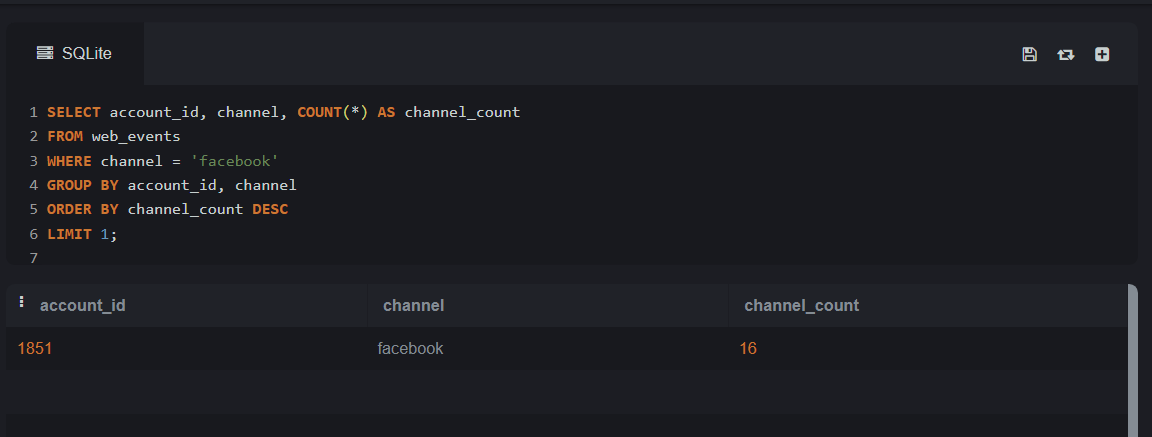
Task#03 – SQL Aggregations -Run the following queries:

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.

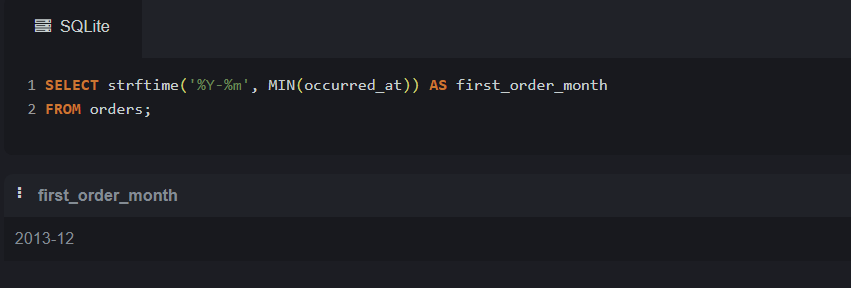


Find the mean (AVERAGE) amount spent per order on each paper type, as well as the mean amount of each paper type purchased per order. Your final answer should have 6 values - one for each paper type for the average number of sales, as well as the average amount

Which account used facebook most as a channel?



Task#04 –SQL Subqueries-Run the following queries:

Use DATE\_TRUNC or EXTRACT to pull month level information about the first order ever placed in the orders table.

Task#05 –SQL Data Cleaning -Run the following queries:

In the accounts table, there is a column holding the website for each company. The last three digits specify what type of web address they are using. A list of extensions (and pricing) is provided here. Pull these extensions and provide how many of each website type exist in the accounts table.

