1. Provide a table for all web\_events associated with account name of Walmart. There should be three columns. Be sure to include the primary\_poc, time of the event, and the channel for each event. Additionally, you might choose to add a fourth column to assure only Walmart events were chosen.

SELECT acounts.primary\_poc,

web\_events.occurred\_at,

web\_events.channel,accounts.name

FROM web\_events

JOIN accounts

ON web\_events.account\_id = accounts.id

WHERE accounts.name = 'Walmart';

2. Provide a table that provides the region for each sales\_rep along with their associated accounts. 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.

SELECT r.name region,

s.name sales\_rep,

a.name account

FROM region r

JOIN sales\_reps s

ON r.id =s.region\_id

JOIN accounts a

ON s.id = a.sales\_rep\_id

ORDER BY a.name;

3.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 I divided by (total + 0.01) to assure not dividing by zero.

SELECT r.name region, a.name, (o.total\_amt\_usd/(o.total+0.01)) unit\_price

FROM orders o

JOIN accounts a

ON o.account\_id = a.id

JOIN sales\_reps s

ON a.sales\_rep\_id = s.id

JOIN region r

ON s.region\_id = r.id;

4. Write a query to list all calls with their start time and end time, for each call, we want to display what was the outcome as well as the first and last name of the employee who made the call. We’ll sort out calls by start time ascending

SELECT c.start\_time, c.end\_time, o.outcome\_text, e.first\_name, e.last\_name

From call c

Join call\_outcome o

ON c.call\_outcome\_id = o.id

Join employee e

ON c.employee\_id = e.id

ODER By c.start\_time;

5. 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.

Select r.name region\_name, s.name sales\_rep, a.name account

From sales\_reps s

Join region r

On s.region\_id =r.id

Join accounts a

On a.sales\_rep\_id= s.id

Where r.name = 'Midwest'

Order by a.name

;

6. 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.

Select r.name region\_name, s.name sales\_rep, a.name account

From sales\_reps s

Join region r

On s.region\_id =r.id

Join accounts a

On a.sales\_rep\_id= s.id

Where r.name = 'Midwest' and s.name like 'S%'

Order by a.name

;

7. 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.

Select r.name region\_name, s.name sales\_rep, a.name account

From sales\_reps s

Join region r

On s.region\_id =r.id

Join accounts a

On a.sales\_rep\_id= s.id

Where r.name = 'Midwest' and s.name like ' % K%'

Order by a.name

;

8. 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).

Select r.name region, a.name account, o.total\_amt\_usd/(o.total+0.01) unit\_price

From orders o

Join accounts a

On o.account\_id = a.id

Join sales\_reps s

On a.sales\_rep\_id = s.id

Join region r

On s.region\_id = r.id

Where o.standard\_qty > 100;

9. 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. In order to avoid a division by zero error, adding .01 to the denominator here is helpful (total\_amt\_usd/(total+0.01).

Select r.name region, a.name account, o.total\_amt\_usd/(o.total+0.01) unit\_price

From orders o

Join accounts a

On o.account\_id = a.id

Join sales\_reps s

On a.sales\_rep\_id = s.id

Join region r

On s.region\_id = r.id

Where o.standard\_qty > 100 and o.poster\_qty > 50

Order by unit\_price ;

10. 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 largest unit price first. In order to avoid a division by zero error, adding .01 to the denominator here is helpful (total\_amt\_usd/(total+0.01).

Select r.name region, a.name account, o.total\_amt\_usd/(o.total+0.01) unit\_price

From orders o

Join accounts a

On o.account\_id = a.id

Join sales\_reps s

On a.sales\_rep\_id = s.id

Join region r

On s.region\_id = r.id

Where o.standard\_qty > 100 and o.poster\_qty > 50

Order by unit\_price Desc ;

11. What are the different channels used by account id 1001? Your final table should have only 2 columns: account name and the different channels. You can try SELECT DISTINCT to narrow down the results to only the unique values.

Select distinct a.name Account, w.channel Channel

From accounts a

Join web\_events w

On w.account\_id =a.id

Where a.id = 1001;

12. Find all the orders that occurred in 2015. Your final table should have 4 columns: occurred\_at, account name, order total, and order total\_amt\_usd.

SELECT o.occurred\_at, a.name, o.total, o.total\_amt\_usd

FROM accounts a

JOIN orders o

ON o.account\_id = a.id

WHERE o.occurred\_at BETWEEN '01-01-2015' AND '01-01-2016'

ORDER BY o.occurred\_at DESC;