

1. 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.
2. 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 Midwestregion. 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.
3. 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 Midwestregion. 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.
4. 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).
5. 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).
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 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).
7. What are the different **channel**s used by **account id** 1001? Your final table should have only 2 columns: **account name** and the different **channel**s. You can try **SELECT DISTINCT** to narrow down the results to only the unique values.
8. 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**.

/\* First Request

\*/

SELECT

r.name region\_name

,sr.name sales\_rep\_name

,a.name account\_name

FROM

region r

JOIN

sales\_reps sr

ON

sr.region\_id = r.id

JOIN

accounts a

ON

a.sales\_rep\_id = sr.id

WHERE

r.name = 'Midwest'

ORDER BY

a.name;

/\* Second Request

\*/

SELECT

r.name region\_name

,sr.name sales\_rep\_name

,a.name account\_name

FROM

region r

JOIN

sales\_reps sr

ON

sr.region\_id = r.id

JOIN

accounts a

ON

a.sales\_rep\_id = sr.id

WHERE

r.name = 'Midwest'

AND sr.name LIKE 'S%'

ORDER BY

a.name;

/\* Third Request

\*/

SELECT

r.name region\_name

,sr.name sales\_rep\_name

,a.name account\_name

FROM

region r

JOIN

sales\_reps sr

ON

sr.region\_id = r.id

JOIN

accounts a

ON

a.sales\_rep\_id = sr.id

WHERE

r.name = 'Midwest'

AND name LIKE '% K%'

ORDER BY

a.name;

/\* Fourth Request

\*/

SELECT

r.name region\_name

,a.name account\_name

,total/(total\_amt\_usd+.01) unit\_price

FROM

region r

JOIN

sales\_reps sr

ON

sr.region\_id = r.id

JOIN

accounts a

ON

a.sales\_rep\_id = sr.id

JOIN

orders o

ON

o.account\_id = a.id

WHERE

standard\_qty > 100;

/\* Fifth Request

\*/

SELECT

r.name region\_name

,a.name account\_name

,total/(total\_amt\_usd+.01) unit\_price

FROM

region r

JOIN

sales\_reps sr

ON

sr.region\_id = r.id

JOIN

accounts a

ON

a.sales\_rep\_id = sr.id

JOIN

orders o

ON

o.account\_id = a.id

WHERE

standard\_qty > 100

AND poster\_qty > 50

ORDER BY

total/(total\_amt\_usd+.01) asc;

/\* Sixth Request

\*/

SELECT

r.name region\_name

,a.name account\_name

,total/(total\_amt\_usd+.01) unit\_price

FROM

region r

JOIN

sales\_reps sr

ON

sr.region\_id = r.id

JOIN

accounts a

ON

a.sales\_rep\_id = sr.id

JOIN

orders o

ON

o.account\_id = a.id

WHERE

standard\_qty > 100

AND poster\_qty > 50

ORDER BY

total/(total\_amt\_usd+.01) desc;

/\* Seventh Request\*/

SELECT DISTINCT

a.name account\_name

,we.channel

FROM

accounts a

JOIN

web\_events we

ON

we.account\_id = a.id

WHERE

a.id = 1001;

/\* Eight Request \*/

SELECT

a.name account\_name

,o.occurred\_at

,o.total as "order total"

,o.total\_amt\_usd

FROM

accounts a

JOIN

orders o

ON

o.account\_id = a.id

WHERE

extract(year from o.occurred\_at) = 2015;