Lesson 2 SQL Joins

19. Quiz: Last Check

Diagram

Description automatically generated

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.

Select

r.name region,

s.name sales\_rep,

a.name account\_name

from region r

join sales\_reps s

on r.id=s.region\_id

join accounts a

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

where r.name ='Midwest'

order by account\_name;

1. 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,

s.name sales\_rep,

a.name account\_name

from region r

join sales\_reps s

on r.id=s.region\_id

join accounts a

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

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

order by account\_name;

1. 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,

s.name sales\_rep,

a.name account\_name

from region r

join sales\_reps s

on r.id=s.region\_id

join accounts a

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

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

order by account\_name;

1. 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\_name,

o.total\_amt\_usd/(o.total+1) unit\_price

from region r

join sales\_reps s

on r.id=s.region\_id

join accounts a

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

join orders o

on a.id=o.account\_id

where o.standard\_qty>100;

1. 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\_name,

o.total\_amt\_usd/(o.total+1) unit\_price

from region r

join sales\_reps s

on r.id=s.region\_id

join accounts a

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

join orders o

on a.id=o.account\_id

where o.standard\_qty>100

and o.poster\_qty>50

order by unit\_price;

1. 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\_name,

o.total\_amt\_usd/(o.total+1) unit\_price

FROM region r

JOIN sales\_reps s

ON r.id=s.region\_id

JOIN accounts a

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

JOIN orders o

ON a.id=o.account\_id

WHERE o.standard\_qty>100

AND o.poster\_qty>50

ORDER BY unit\_price desc;

1. 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.

SELECT

DISTINCT

a.name account\_name,

w.channel channel

FROM accounts a

JOIN web\_events w

ON a.id=w.account\_id

AND a.id='1001';

1. 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 orders o

JOIN accounts a

ON o.account\_id=a.id

WHERE DATE\_PART('year',o.occurred\_at)='2015';