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 representative, 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**

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 representative, 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 s.name like 'S%' and r.name = 'Midwest'**

**order by a.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.  
**select r.name region, s.name reps, 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 s.name like '% K%' and r.name = 'Midwest'**

**order by a.name**

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) as 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**

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) as 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**

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) as 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**

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.

Adwords, direct, twitter, banner, organic, and facebook.

**select distinct a.name account, w.channel**

**from orders o**

**join accounts a**

**on o.account\_id = a.id**

**join web\_events w**

**on w.account\_id = a.id**

**where o.account\_id = 1001**

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 account, o.total, o.total\_amt\_usd**

**from orders o**

**join accounts a**

**on o.account\_id = a.id**

**where o.occurred\_at::text like '2015%'**

**//this is the correct was so you can keep the same data type for occurred\_at in**

**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**;