



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 region.name as regionname, sales_reps.name as salesrepname, accounts.name as accountsname
from region join sales_reps on
sales_reps.region_id=region.id
join accounts on accounts.sales_rep_id=sales_reps.id and region.name='Midwest' order by
accounts.name ASC;
  
```

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 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 region.name as regionname,sales_reps.name as salesrepname,accounts.name as accountsname
from region join sales_reps on
sales_reps.region_id=region.id
join accounts on accounts.sales_rep_id=sales_reps.id and region.name='Midwest' and sales_reps.name
like 'S%' order by accounts.name ASC;
```

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 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 region.name as regionname,sales_reps.name as salesrepname,accounts.name as accountsname
from region join sales_reps on
sales_reps.region_id=region.id
join accounts on accounts.sales_rep_id=sales_reps.id and region.name='Midwest' and sales_reps.name
like '% K%' order by accounts.name ASC;
```

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.

```
SELECT region.name AS regionname, accounts.name AS accountsname, (orders.total_amt_usd /
(orders.total + 0.01)) AS orderpercent
FROM region
JOIN sales_reps ON sales_reps.region_id = region.id
JOIN accounts ON accounts.sales_rep_id = sales_reps.id
JOIN orders ON orders.account_id = accounts.id
WHERE orders.standard_qty > 100;
```

Provide the name for each region for every order, as well as the account name and the unit price they paid ($\text{total_amt_usd}/\text{total}$) 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 ($\text{total_amt_usd}/(\text{total}+0.01)$).

```
SELECT region.name AS regionname, accounts.name AS accountsname, (orders.total_amt_usd /  
(orders.total + 0.01)) AS orderpercent
```

```
FROM region
```

```
JOIN sales_reps ON sales_reps.region_id = region.id
```

```
JOIN accounts ON accounts.sales_rep_id = sales_reps.id
```

```
JOIN orders ON orders.account_id = accounts.id
```

```
WHERE orders.standard_qty > 100 and poster_qty>50 order By orderpercent;
```

Provide the name for each region for every order, as well as the account name and the unit price they paid ($\text{total_amt_usd}/\text{total}$) 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 ($\text{total_amt_usd}/(\text{total}+0.01)$).

```
SELECT region.name AS regionname, accounts.name AS accountsname, (orders.total_amt_usd /  
(orders.total + 0.01)) AS orderpercent
```

```
FROM region
```

```
JOIN sales_reps ON sales_reps.region_id = region.id
```

```
JOIN accounts ON accounts.sales_rep_id = sales_reps.id
```

```
JOIN orders ON orders.account_id = accounts.id
```

```
WHERE orders.standard_qty > 100 and poster_qty>50 order By orderpercent DESC;
```

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,w.channel
```

```
FROM accounts as a
```

```
Join web_events as w
```

on w.account_id=a.id where a.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,o.total,o.total_amt_usd FROM accounts as a join  
orders as o on a.id=o.account_id Where extract(year from o.occurred_at)='2015';
```