**Crypto Data Analysis**

**Relevant tables:** members, prices , transactions   
**Tool used:** PgAdmin4

1. Sort all the rows in members table by first name in alphabetical order and show top 3 rows

select \* from members order by first\_name limit 3;

1. Which records from trading members are from region ‘United States’?

select \* from members where region = 'United States';

1. Select only member\_id and first\_name columns for members who are not from Australia.

select member\_id, first\_name from members where region!='Australia';

1. Select unique region values from members and sort the output in reverse alphabetical order.

select distinct region from members order by region desc;

1. How many members are there per region? Sort the output by regions with the most members to the least.

select region,count(\*) as no\_of\_members from members

group by region

order by no\_of\_members desc;

1. How many US members and how many non-US members are there?

select

(case when region!='United States' then 'Not United States'

else region

end

) region,count(\*) as member\_count

from members

group by (case when region!='United States' then 'Not United States'

else region

end

)

order by member\_count desc;

1. How many total records are there in prices table?

select count(\*) as total\_records from prices;

1. How many records are there in prices table per ticker value?

select ticker,count(\*) as no\_of\_records from prices

group by ticker;

1. What is the minimum and maximum market\_date values?

select ticker, min(market\_date),max(market\_date) from prices

group by ticker;

1. What’s the monthly average of the price column for Ethereum in 2020? Sort the output in chronological order and also round the average price to 2 decimals

select extract(month from market\_date) as market\_month,round(avg(price),2) as avg\_price

from prices

where ticker = 'ETH'

group by market\_month

order by avg\_price;

1. How many duplicate market\_date values for any ticker value in our table?

select market\_date,ticker, count(\*) as count\_extra

from prices

group by market\_date,ticker

having count(\*)>1;

1. How many days from prices table exists where high price of bitcoin is over $20,000?

select count(distinct market\_date) as row\_count from prices where high>'20000';

select \* from prices order by id desc limit 10;

1. How many “breakout” days were there in 2020 where the price column is creater than the open column for each ticker?

select ticker, count(distinct market\_date) as breakout\_days\_count from prices where price>open

group by ticker;

1. How many “non-breakout” days were there in 2020 where the price column is less than the open column for each ticker?

select ticker, count(distinct market\_date) as non\_breakout\_days\_count from prices where price<open

group by ticker;

1. What percentage of days in 2020 were breakout days and non-breakout days? Round the percentages to 2 decimal places.

SELECT

ticker,

ROUND(AVG(CASE WHEN price > open THEN 1 ELSE 0 END), 2)\*100 AS breakout\_day\_percent,

ROUND(AVG(CASE WHEN price < open THEN 1 ELSE 0 END), 2)\*100 AS non\_breakout\_day\_percent

FROM

prices

GROUP BY

ticker;

1. How many buy and sell transactions are there for bitcoin?

select txn\_type, count(\*) as transaction\_count from transactions

where ticker = 'BTC'

group by txn\_type;

1. Which members have sold less than 500 bitcoin? Sort the output from most BTC sold to the least.

with cte as (

select member\_id,sum(quantity) as btc\_sold\_quantity

from transactions

where ticker = 'BTC' and txn\_type = 'SELL'

group by member\_id

)

select members.\*, btc\_sold\_quantity

from members

inner join cte

on cte.member\_id = members.member\_id

where btc\_sold\_quantity<500

order by btc\_sold\_quantity desc;

1. Which member has the highest buy-to-sell ratio by quantity?

select members.\*,

sum(case when txn\_type = 'BUY' then quantity else 0 end)/sum(case when txn\_type = 'SELL' then quantity else 0 end) as buy\_to\_sell\_ratio

from transactions

inner join members

on members.member\_id = transactions.member\_id

group by members.member\_id

order by buy\_to\_sell\_ratio desc;

1. Which members have most bitcoin quantity at the end of the period?

select members.first\_name,

sum(case when txn\_type = 'BUY' then transactions.quantity

when txn\_type = 'SELL' then transactions.quantity end) as total\_quantity

from transactions

inner join members

on members.member\_id = transactions.member\_id

where ticker = 'BTC'

group by members.first\_name

order by total\_quantity desc;