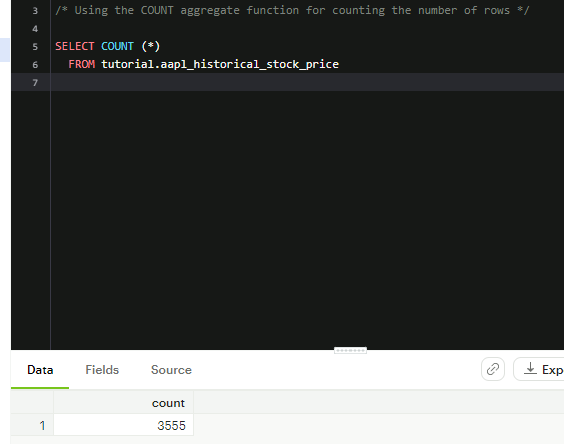
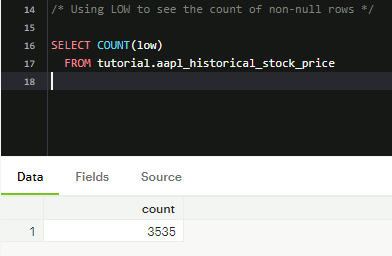


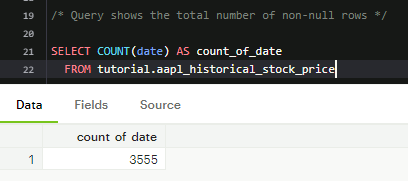
- Selecting all to view what the data has



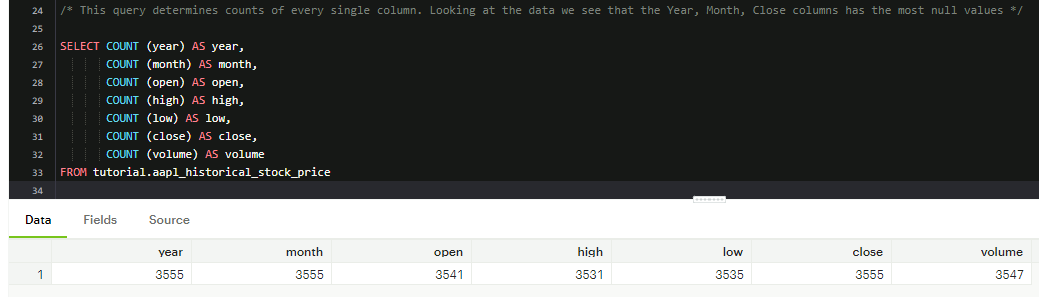
- Using the COUNT aggregate function



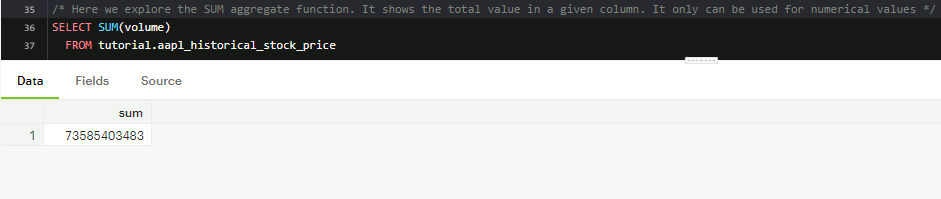
- Using low to count the number of non-null rows



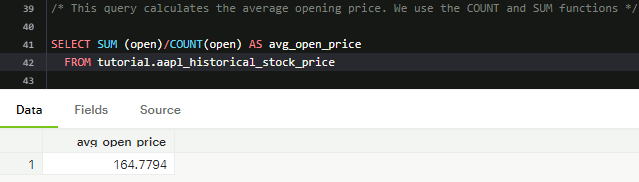
- Showing the total number of non-null rows, using AS to name the column itself for the count data.



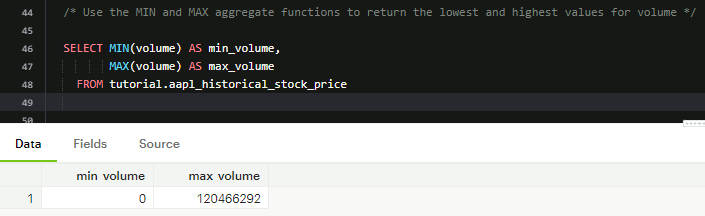
- Query to determine which columns have the most null values



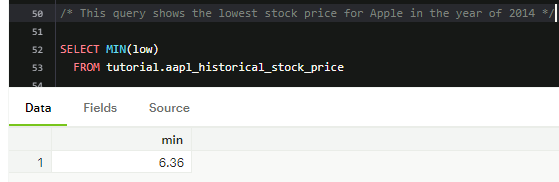
- Exploring the SUM function for volume



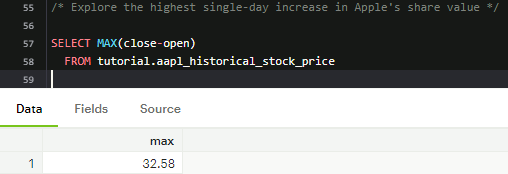
- Using SUM and diving by COUNT, we get the average price of stock open. We use AS to direct to a new column name. We observe 164.78 as the avg open price



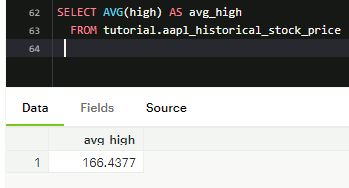
- Using the MIN and MAX aggregate functions show us the MIN and MAX volume size of Apple stocks. MIN and MAX can be used on non-numerical values as well



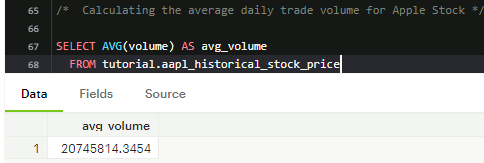
- This query explores the lowest price of the Apple stock in 2014



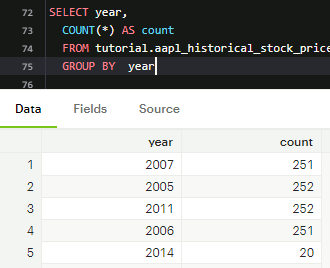
- This query we are exploring the highest single-day increase in Apple’s share value. We use the MAX function to get the “highest” value and have to subtract close by open to get the value of increase/decrease.



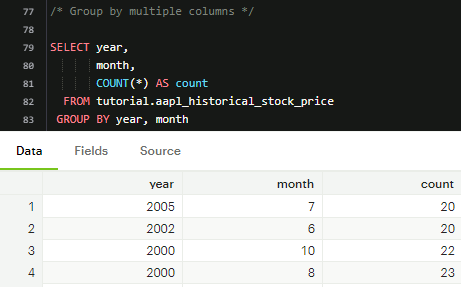
- Here we use the AVG function for the highest value of stock share in 2014



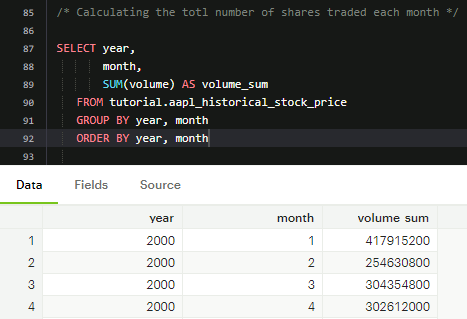
- We query the average daily trade volume for Apple stock in 2014, using the AS clause to specify the table name as avg volume.



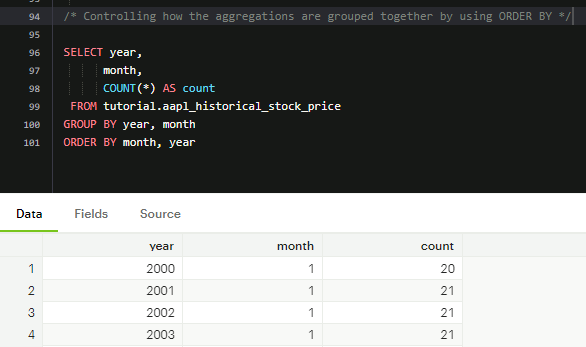
- When using aggregate functions, we can use GROUP BY clause to separate data into groups, which can be aggregated independently of one another



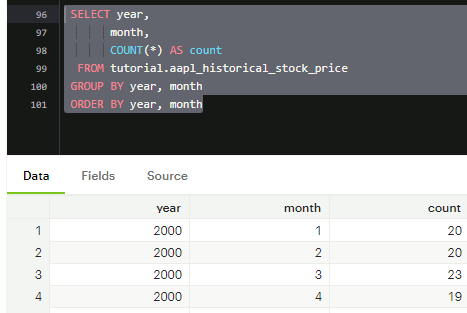
* Group by multiple columns to see the count of month and year



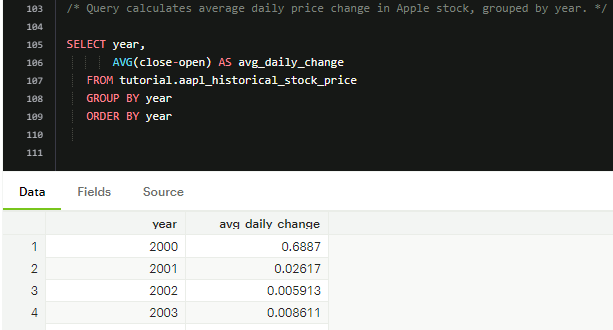
- Here we selected year, month and did SUM of volume to look at total number of shares traded each month



- Controlling how the aggregations are grouped together by the use of ORDER BY. Reversing month and year for ORDER BY, shows us the results in ASC order for the months starting from 1 then the year, if we reverse it to year then month, it groups the year and month in a different order.



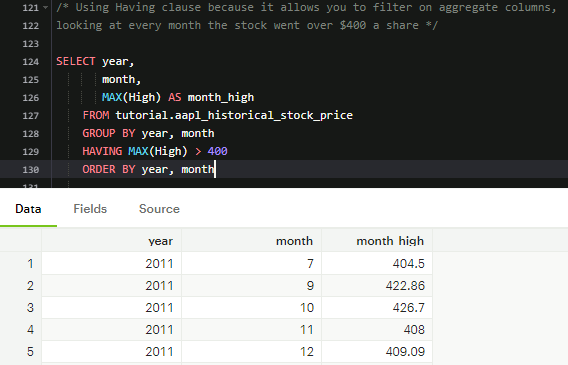
As seen, priority goes to the year.



- Observations show the avg price per daily change



- This query calculates the lowest and highest prices that the Apple Stock achieved each month. We have to use MIN and MAX with low and high prices to show the prices. We need year and month to show correlation.



- In this dataset we observe many different months, we are looking for every month that the Apple stock price has worked its way over $400. With the WHERE clause this can’t be done due to it now allowing you to filter on the aggregate columns itself. By using the HAVING clause we can. We observe the data starting by the year and which months it showed above $400