

Over the past 18 months, an emerging technology has caught the attention of the world; the NFT. What is an NFT? They are digital assets stored on the blockchain. And over \$22 billion was spent last year on purchasing NFTs. Why? People enjoyed the art, the speculated on what they might be worth in the future, and people didn't want to miss out.

The future of NFT's is unclear as much of the NFT's turned out to be scams of sorts since the field is wildly unregulated. They're also contested heavily for their impact on the environment.

Regardless of these controversies, it is clear that there is money to be made in NFT's. And one cool part about NFT's is that all of the data is recorded on the blockchain, meaning anytime something happens to an NFT, it is logged in this database.

In this project, you'll be tasked to analyze real-world [NFT data](#).

That data set is a sales data set of one of the most famous NFT projects, Cryptopunks. Meaning each row of the data set represents a sale of an NFT. The data includes sales from January 1st, 2018 to December 31st, 2021. The table has several columns including the buyer address, the ETH price, the price in U.S. dollars, the seller's address, the date, the time, the NFT ID, the transaction hash, and the NFT name.

You might not understand all the jargon around the NFT space, but you should be able to infer enough to answer the following prompts.

1. How many sales occurred during this time period?
2. Return the top 5 most expensive transactions (by USD price) for this data set. Return the name, ETH price, and USD price, as well as the date.
3. Return a table with a row for each transaction with an event column, a USD price column, and a moving average of USD price that averages the last 50 transactions.
4. Return all the NFT names and their average sale price in USD. Sort descending. Name the average column as `average_price`.
5. Return each day of the week and the number of sales that occurred on that day of the week, as well as the average price in ETH. Order by the count of transactions in ascending order.
6. Construct a column that describes each sale and is called `summary`. The sentence should include who sold the NFT name, who bought the NFT, who sold the NFT, the date, and what price it was sold for in USD rounded to the nearest thousandth.
Here's an example summary:
"CryptoPunk #1139 was sold for \$194000 to
0x91338ccfb8c0adb7756034a82008531d7713009d from
0x1593110441ab4c5f2c133f21b0743b2b43e297cb on 2022-01-14"

7. Create a view called "1919_purchases" and contains any sales where "0x1919db36ca2fa2e15f9000fd9cdc2edcf863e685" was the buyer.
8. Create a histogram of ETH price ranges. Round to the nearest hundred value.
9. Return a unioned query that contains the highest price each NFT was bought for and a new column called status saying "highest" with a query that has the lowest price each NFT was bought for and the status column saying "lowest". The table should have a name column, a price column called price, and a status column. Order the result set by the name of the NFT, and the status, in ascending order.
10. What NFT sold the most each month / year combination? Also, what was the name and the price in USD? Order in chronological format.
11. Return the total volume (sum of all sales), round to the nearest hundred on a monthly basis (month/year).
12. Count how many transactions the wallet "0x1919db36ca2fa2e15f9000fd9cdc2edcf863e685" had over this time period.
13. Create an "estimated average value calculator" that has a representative price of the collection every day based off of these criteria:
 - Exclude all daily outlier sales where the purchase price is below 10% of the daily average price
 - Take the daily average of remaining transactions
 - a) First create a query that will be used as a subquery. Select the event date, the USD price, and the average USD price for each day using a window function. Save it as a temporary table.
 - b) Use the table you created in Part A to filter out rows where the USD prices is below 10% of the daily average and return a new estimated value which is just the daily average of the filtered data.