First address the issue with data:

The day column is not correct formats are different so it cannot be used without setting them in YYYY-MM-DD, for changing them following code is used:

```
UPDATE cryptopunkdata

SET day =

   CASE

      WHEN `day` LIKE '%/%' THEN DATE_FORMAT(STR_TO_DATE(`day`, '%m/%d/%Y'), '%Y-%m-%d')

      WHEN `day` LIKE '%-%' THEN DATE_FORMAT(STR_TO_DATE(`day`, '%m-%d-%Y'), '%Y-%m-%d')

      ELSE `day`

   END;
```

1. How many sales occurred during this time period?

A. SELECT COUNT(*) FROM cryptopunkdata;

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.

A. CREATE TEMPORARY TABLE usd_top_5 SELECT * FROM cryptopunkdata ORDER BY `usd_price` DESC LIMIT 5;

SELECT `name`, `eth_price`, `usd_price`, `day` FROM usd_top_5;

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.

A. SELECT `day`, `usd_price`, AVG(usd_price) OVER (ORDER BY `day` ROWS BETWEEN 49 PRECEDING AND CURRENT ROW) FROM cryptopunkdata;

4. Return all the NFT names and their average sale price in USD. Sort descending. Name the average column as average_price.

A. SELECT `name`, AVG(usd_price) AS 'average_price' FROM cryptopunkdata GROUP BY `name` ORDER BY average_price DESC;

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.

A. SELECT DAYNAME(day) AS day_name, AVG(eth_price), COUNT(*) AS sales FROM cryptopunkdata GROUP BY day_name ORDER BY sales ASC;

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"

A. SELECT CONCAT(`NAME`," was sold for ",ROUND(usd_price,-3)," to ",`ï»¿buyer_address`," from ",`seller_address`," on ",`day`) AS summary FROM cryptopunkdata;

7. Create a view called "1919_purchases" and contains any sales where "0x1919db36ca2fa2e15f9000fd9cdc2edcf863e685" was the buyer.

A. CREATE VIEW `1919_purchases` AS

SELECT * FROM cryptopunkdata WHERE `buyer_address`="0x1919db36ca2fa2e15f9000fd9cdc2edcf863e685";

SELECT * FROM 1919_purchases;

8. Create a histogram of ETH price ranges. Round to the nearest hundred value.

A. SELECT ROUND(eth_price,-2) `eth`, COUNT(*) AS count, RPAD('',COUNT(*),'*') AS bar FROM cryptopunkdata GROUP BY `eth` ORDER BY`eth`;

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.

A. SELECT name, price, 'highest' AS status

```
FROM (SELECT name, MAX(usd_price) AS price

   FROM cryptopunkdata

   GROUP BY name) AS highest_prices

UNION ALL

SELECT name, price, 'lowest' AS status

FROM (SELECT name, MIN(usd_price) AS price

   FROM cryptopunkdata

   GROUP BY name) AS lowest_prices

ORDER BY name ASC, status ASC;
```

10. What NFT sold the most each month / year combination? Also, what was the name and the price in USD? Order in chronological format.

A. SELECT name AS NFT_Name, CONCAT(MONTH(day), '-', YEAR(day)) AS Month_Year, usd_price AS Price_USD FROM cryptopunkdata cpd

JOIN (SELECT MONTH(day) AS month, YEAR(day) AS year, MAX(usd_price) AS max_price

  FROM cryptopunkdata

  GROUP BY MONTH(day), YEAR(day)) AS max_prices

ON MONTH(cpd.day) = max_prices.month AND YEAR(cpd.day) = max_prices.year AND cpd.usd_price = max_prices.max_price

ORDER BY YEAR(cpd.day), MONTH(cpd.day);

11. Return the total volume (sum of all sales), round to the nearest hundred on a monthly basis (month/year).

A. SELECT CONCAT(MONTH(day), '-', YEAR(day)) AS Month_Year, SUM(usd_price) FROM cryptopunkdata GROUP BY CONCAT(MONTH(day), '-', YEAR(day)) ORDER BY MIN(day);

12. Count how many transactions the wallet "0x1919db36ca2fa2e15f9000fd9cdc2edcf863e685"had over this time period.

A. SELECT COUNT(*) FROM cryptopunkdata WHERE `buyer_address` = "0x1919db36ca2fa2e15f9000fd9cdc2edcf863e685" OR `seller_address` = "0x1919db36ca2fa2e15f9000fd9cdc2edcf863e685";