1. Do a manual review of the table nsedata and describe its contents (no SQL to be executed

for this task.)

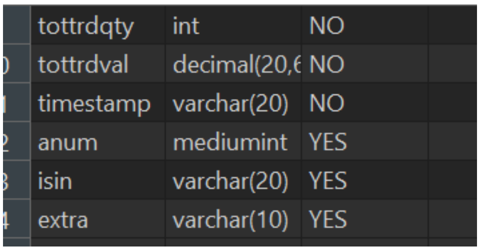
It's the stock price data of companies from 1st April, 2011 to 31st October, 2014, including the opening and closing prices, highs and lows, the previous day's closing, and the total quantity and total value traded for each stock.

2. Select the database stockdata using SQL.

```
use stockdata;
```

3. Get a schema dump of the table nsedata.

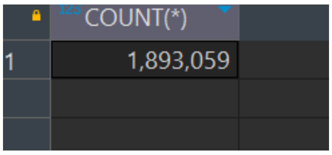
```
DESCRIBE nsedata;
```



tottrdqty	int	NO
tottrdval	decimal(20,6	NO
timestamp	varchar(20)	NO
anum	mediumint	YES
isin	varchar(20)	YES
extra	varchar(10)	YES

4. Get a count of the total number of records in nsedata.

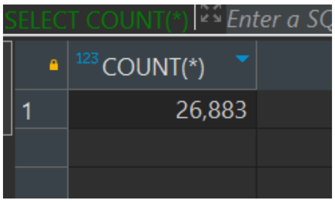
```
SELECT COUNT(*) FROM nsedata;
```



	COUNT(*)
1	1,893,059

5. Get the total count of the records for the month "October 2012".

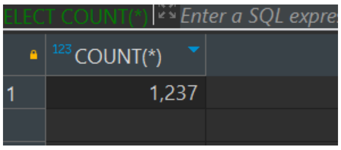
```
SELECT COUNT(*)
FROM nsedata
WHERE DATE_FORMAT(STR_TO_DATE(timestamp , '%d-%b-%Y'), '%Y-%m') = '2011-
04';
```



	COUNT(*)
1	26,883

6. Repeat '4', but only for the stock with symbol "GEOMETRIC".

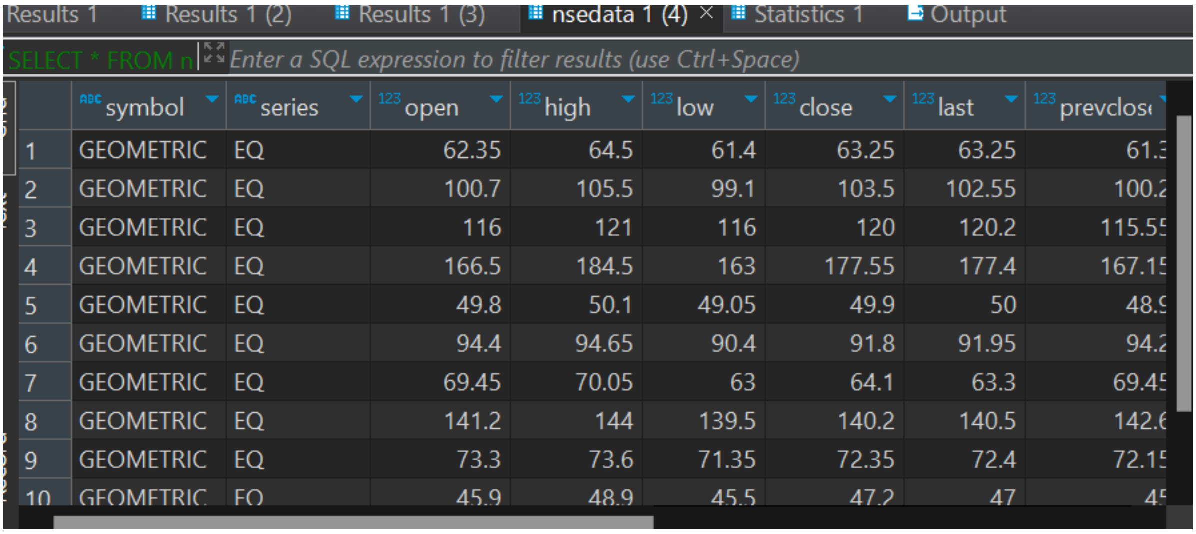
```
SELECT COUNT(*)
FROM nsedata
WHERE symbol = 'GEOMETRIC';
```



	COUNT(*)
1	1,237

7. Repeat ‘6’, but only display the first 10 records.

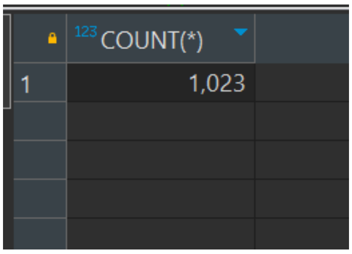
```
SELECT *
FROM nsedata
WHERE symbol = 'GEOMETRIC'
LIMIT 10;
```



	symbol	series	open	high	low	close	last	prevclos
1	GEOMETRIC	EQ	62.35	64.5	61.4	63.25	63.25	61.3
2	GEOMETRIC	EQ	100.7	105.5	99.1	103.5	102.55	100.2
3	GEOMETRIC	EQ	116	121	116	120	120.2	115.55
4	GEOMETRIC	EQ	166.5	184.5	163	177.55	177.4	167.15
5	GEOMETRIC	EQ	49.8	50.1	49.05	49.9	50	48.9
6	GEOMETRIC	EQ	94.4	94.65	90.4	91.8	91.95	94.2
7	GEOMETRIC	EQ	69.45	70.05	63	64.1	63.3	69.45
8	GEOMETRIC	EQ	141.2	144	139.5	140.2	140.5	142.6
9	GEOMETRIC	EQ	73.3	73.6	71.35	72.35	72.4	72.15
10	GEOMETRIC	EQ	45.9	48.9	45.5	47.2	47	45

8. Totally, how many records of “INFY” does the table contain?

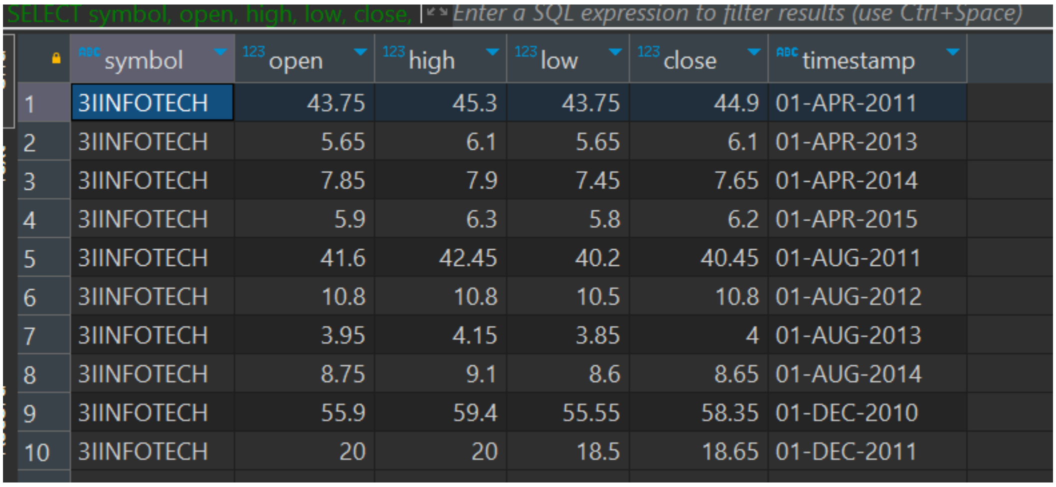
```
SELECT COUNT(*)
FROM nsedata
WHERE symbol = 'INFY';
```



	COUNT(*)
1	1,023

9. Get a listing of the first 10 records of “3IINFOTECH”, but the listing should contain only the following columns: symbol, open, high, low, close, and timestamp.

```
SELECT symbol, open, high, low, close, timestamp
FROM nsedata
WHERE symbol = '3IINFOTECH'
LIMIT 10;
```

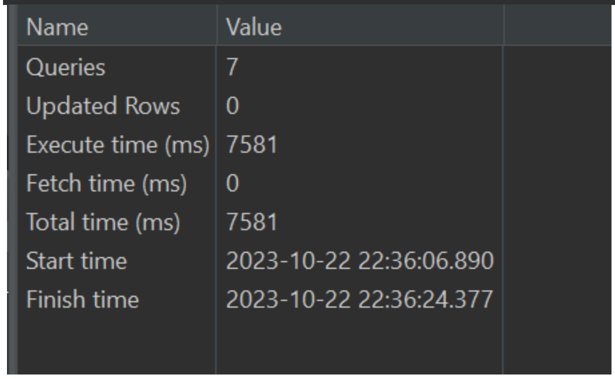


SELECT symbol, open, high, low, close, | Enter a SQL expression to filter results (use Ctrl+Space)

	symbol	open	high	low	close	timestamp
1	3IINFOTECH	43.75	45.3	43.75	44.9	01-APR-2011
2	3IINFOTECH	5.65	6.1	5.65	6.1	01-APR-2013
3	3IINFOTECH	7.85	7.9	7.45	7.65	01-APR-2014
4	3IINFOTECH	5.9	6.3	5.8	6.2	01-APR-2015
5	3IINFOTECH	41.6	42.45	40.2	40.45	01-AUG-2011
6	3IINFOTECH	10.8	10.8	10.5	10.8	01-AUG-2012
7	3IINFOTECH	3.95	4.15	3.85	4	01-AUG-2013
8	3IINFOTECH	8.75	9.1	8.6	8.65	01-AUG-2014
9	3IINFOTECH	55.9	59.4	55.55	58.35	01-DEC-2010
10	3IINFOTECH	20	20	18.5	18.65	01-DEC-2011

10. Repeat '9', but this time use the results to create a temporary table t1.

```
CREATE TABLE t1 (
    symbol VARCHAR(255),
    open DECIMAL(10, 2),
    high DECIMAL(10, 2),
    low DECIMAL(10, 2),
    close DECIMAL(10, 2),
    timestamp TIMESTAMP
);
INSERT INTO t1 (symbol, open, high, low, close, timestamp)
SELECT symbol, open, high, low, close, STR_TO_DATE(timestamp, '%d-%b-%Y')
FROM nsedata
WHERE symbol = '3IINFOTECH'
LIMIT 10;
```



Name	Value
Queries	7
Updated Rows	0
Execute time (ms)	7581
Fetch time (ms)	0
Total time (ms)	7581
Start time	2023-10-22 22:36:06.890
Finish time	2023-10-22 22:36:24.377

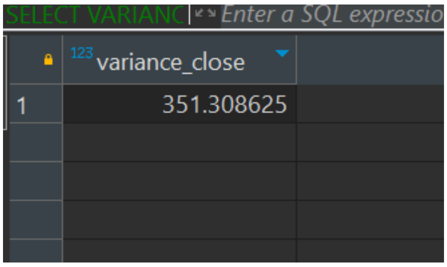
11. Using t1 find out the following for the column close: max, min, mean. standard deviation and variance.

```
-- Maximum
SELECT MAX(close) AS max_close FROM t1;

-- Minimum
SELECT MIN(close) AS min_close FROM t1;

-- Mean (Average)
SELECT AVG(close) AS mean_close FROM t1;

-- Standard Deviation
SELECT STD(close) AS std_dev_close FROM t1;
-- Variance
SELECT VARIANCE(close) AS variance_close FROM t1;
```



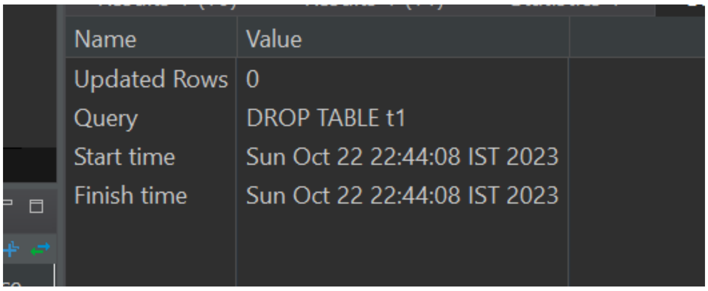
	variance_close
1	351.308625

12. How will you find out the value of the median?

For odd rows, we will select the middle value. For even rows, we will select the average of the two middle values.

13. Delete table t1.

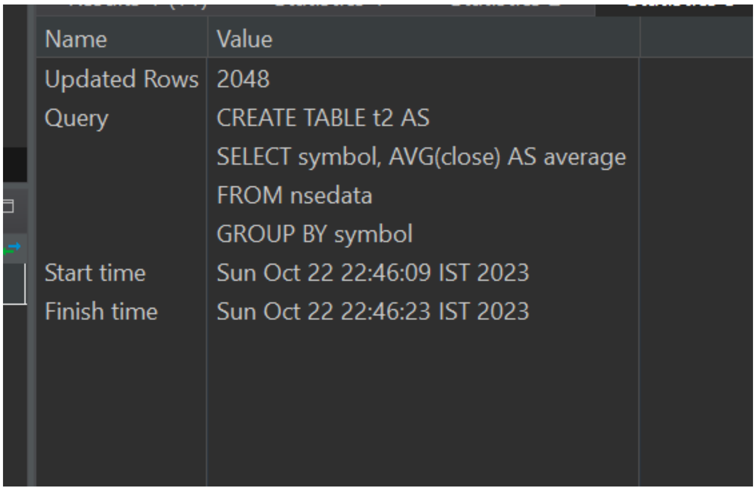
```
DROP TABLE t1;
```



Name	Value
Updated Rows	0
Query	DROP TABLE t1
Start time	Sun Oct 22 22:44:08 IST 2023
Finish time	Sun Oct 22 22:44:08 IST 2023

14. Use nsedata. Using the GROUP BY functionality of SQL create a table t2 containing the average value of close for each and every symbol in the table. Hint: the table will have the columns: symbol, average.

```
CREATE TABLE t2 AS
SELECT symbol, AVG(close) AS average
FROM nsedata
GROUP BY symbol;
```



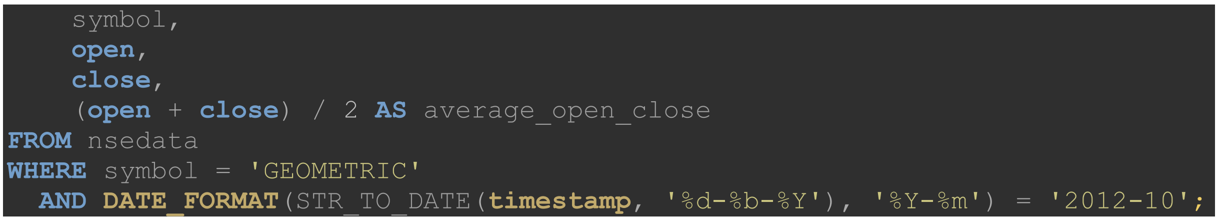
Name	Value
Updated Rows	2048

Query	CREATE TABLE t2 AS
	SELECT symbol, AVG(close) AS average
	FROM nsedata
	GROUP BY symbol

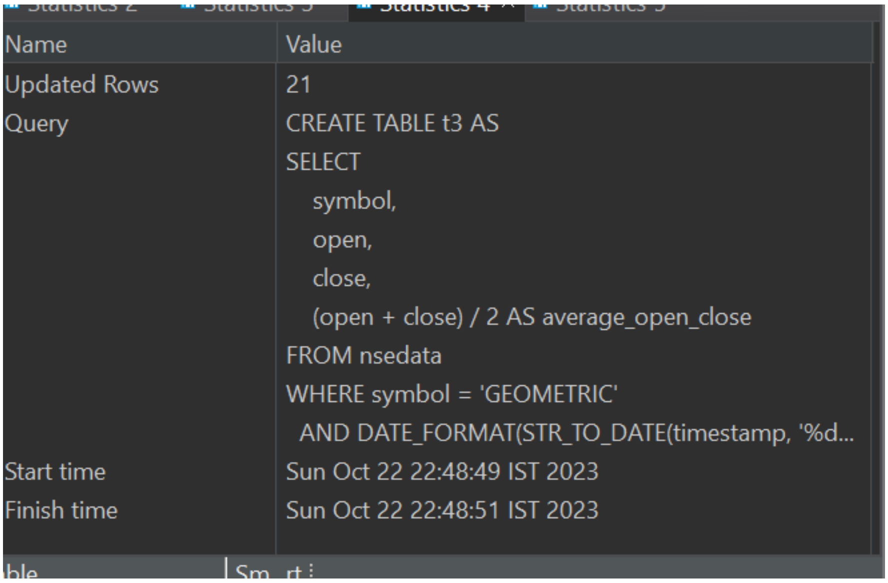
Start time	Sun Oct 22 22:46:09 IST 2023
Finish time	Sun Oct 22 22:46:23 IST 2023

15. Create a table t3 such that it contains the following columns: symbol, open, close, "average of open and close". Fill up this table for the company GEOMETRIC, for the month of October 2012.

```
CREATE TABLE t3 AS
SELECT
```

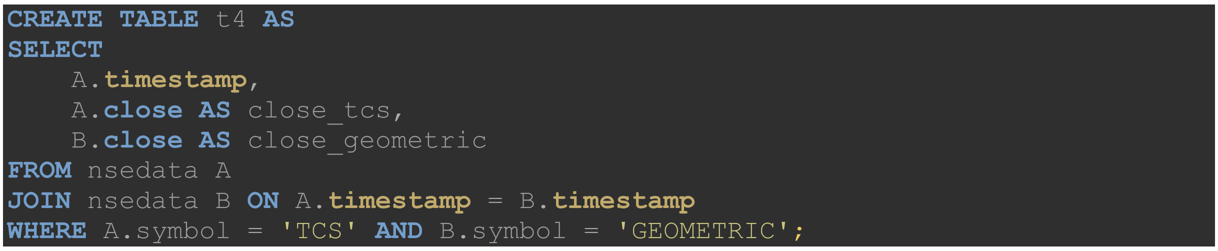


```
    symbol,
    open,
    close,
    (open + close) / 2 AS average_open_close
FROM nsedata
WHERE symbol = 'GEOMETRIC'
  AND DATE_FORMAT(STR_TO_DATE(timestamp, '%d-%b-%Y'), '%Y-%m') = '2012-10';
```

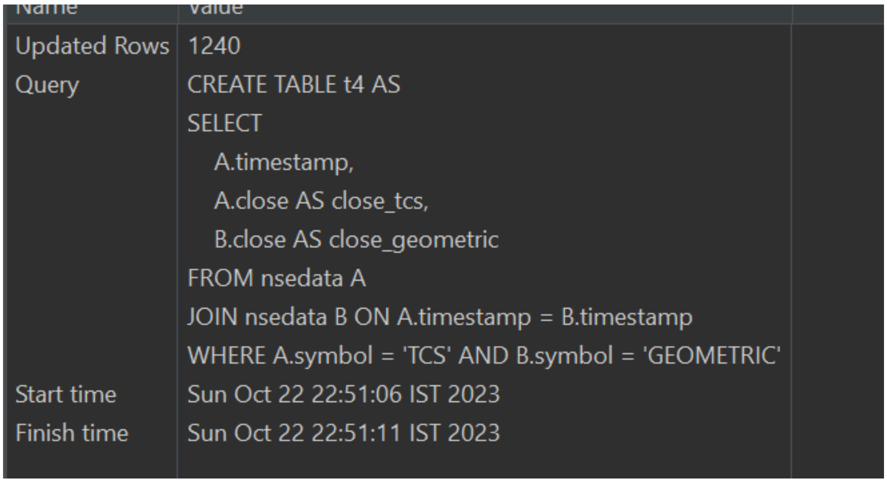


Name	Value
Updated Rows	21
Query	CREATE TABLE t3 AS SELECT symbol, open, close, (open + close) / 2 AS average_open_close FROM nsedata WHERE symbol = 'GEOMETRIC' AND DATE_FORMAT(STR_TO_DATE(timestamp, '%d...
Start time	Sun Oct 22 22:48:49 IST 2023
Finish time	Sun Oct 22 22:48:51 IST 2023

16. It is required to create a table t4 such that it contains the data for two companies GEOMETRIC and TCS. The columns of this table should be as follows: timestamp, close_tcs, close_geometric. Hint: use JOIN.



```
CREATE TABLE t4 AS
SELECT
    A.timestamp,
    A.close AS close_tcs,
    B.close AS close_geometric
FROM nsedata A
JOIN nsedata B ON A.timestamp = B.timestamp
WHERE A.symbol = 'TCS' AND B.symbol = 'GEOMETRIC';
```



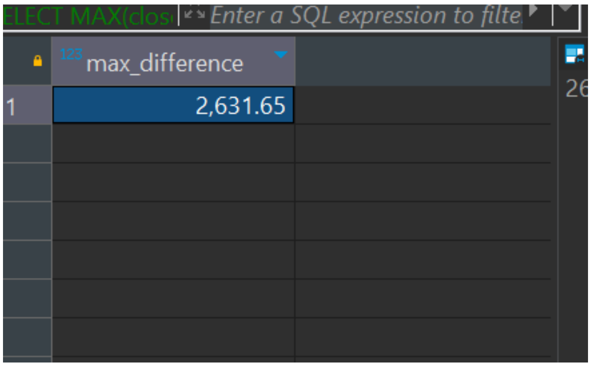
Name	Value
Updated Rows	1240

Query	CREATE TABLE t4 AS
	SELECT
	A.timestamp,
	A.close AS close_tcs,
	B.close AS close_geometric
	FROM nsedata A
	JOIN nsedata B ON A.timestamp = B.timestamp
	WHERE A.symbol = 'TCS' AND B.symbol = 'GEOMETRIC'

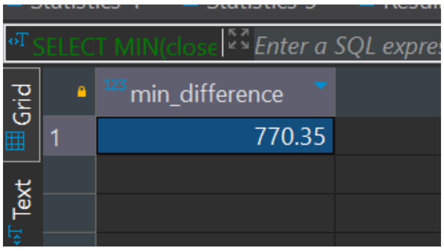
Start time	Sun Oct 22 22:51:06 IST 2023
Finish time	Sun Oct 22 22:51:11 IST 2023

17. Find out the maximum and minimum difference in the daily closing prices of these two companies.

```sql
SELECT MAX(close_tcs - close_geometric) AS max_difference
FROM t4;
SELECT MIN(close_tcs - close_geometric) AS min_difference
FROM t4;
```



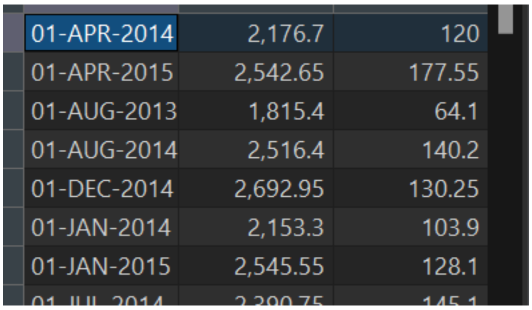
	max_difference
1	2,631.65



	min_difference
1	770.35

18. Based on t4 can you identify those days on which the difference in their closing price was more than the average of the minimum and maximum difference.

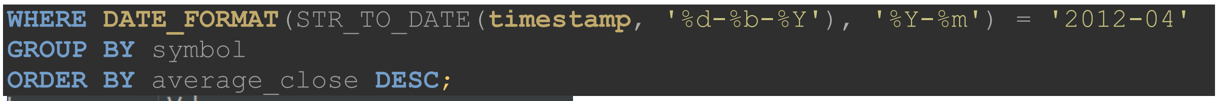
```sql
SELECT timestamp, close_tcs, close_geometric
FROM t4
WHERE close_tcs - close_geometric > (SELECT (MIN(close_tcs -
close_geometric) + MAX(close_tcs - close_geometric)) / 2
                                     FROM t4);
```



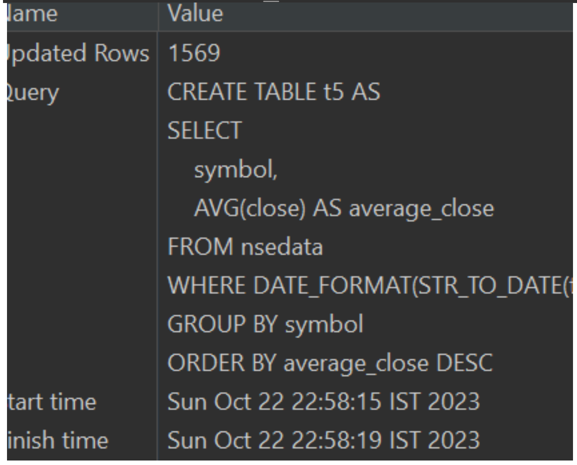
01-APR-2014	2,176.7	120
01-APR-2015	2,542.65	177.55
01-AUG-2013	1,815.4	64.1
01-AUG-2014	2,516.4	140.2
01-DEC-2014	2,692.95	130.25
01-JAN-2014	2,153.3	103.9
01-JAN-2015	2,545.55	128.1
01-JUL-2014	2,390.75	145.1

19. Based on nsedata, create table t5 such that it contains the average close price of each company traded in the month of April 2012. The table should be sorted in descending order of the average close price.

```sql
CREATE TABLE t5 AS
SELECT
    symbol,
    AVG(close) AS average_close
FROM nsedata
```

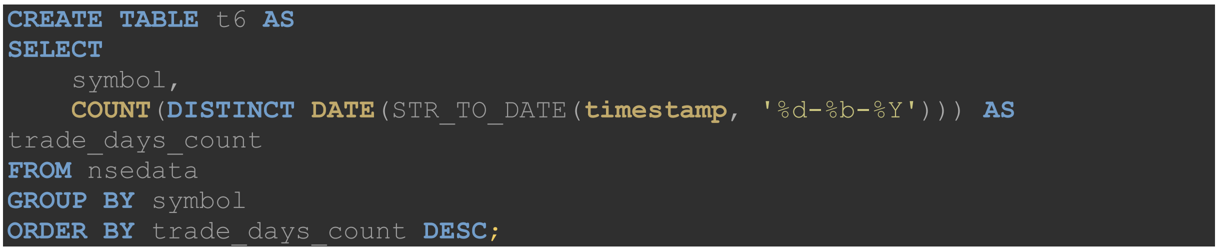


```
WHERE DATE_FORMAT(STR_TO_DATE(timestamp, '%d-%b-%Y'), '%Y-%m') = '2012-04'
GROUP BY symbol
ORDER BY average_close DESC;
```

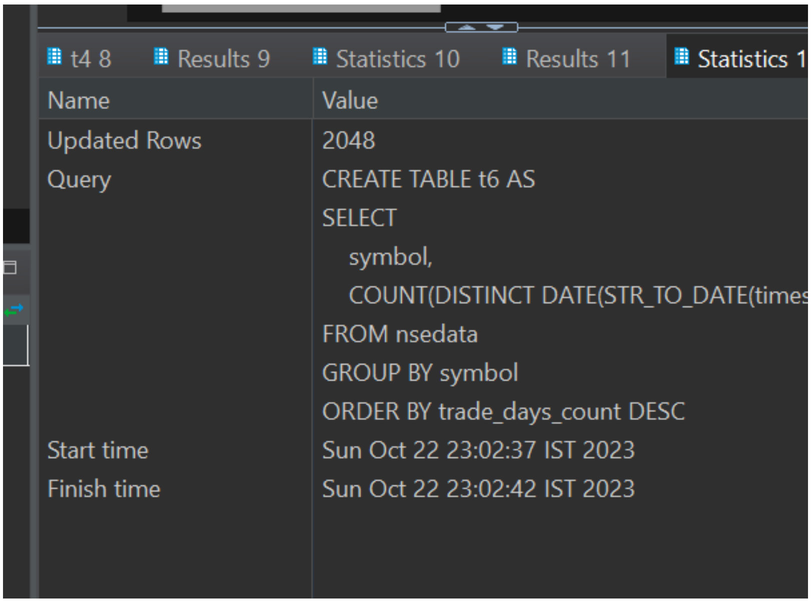


```
lame            Value
Jpdated Rows    1569
Query           CREATE TABLE t5 AS
                SELECT
                    symbol,
                    AVG(close) AS average_close
                FROM nsedata
                WHERE DATE_FORMAT(STR_TO_DATE(t
                GROUP BY symbol
                ORDER BY average_close DESC
tart time       Sun Oct 22 22:58:15 IST 2023
inish time      Sun Oct 22 22:58:19 IST 2023
```

20. Not all companies are traded every day. It is required to create a table that contains a count of the days each company has been traded. The table should be sorted in descending order of the count.



```
CREATE TABLE t6 AS
SELECT
    symbol,
    COUNT(DISTINCT DATE(STR_TO_DATE(timestamp, '%d-%b-%Y'))) AS
trade_days_count
FROM nsedata
GROUP BY symbol
ORDER BY trade_days_count DESC;
```



```
Name            Value
Updated Rows    2048
Query           CREATE TABLE t6 AS
                SELECT
                    symbol,
                    COUNT(DISTINCT DATE(STR_TO_DATE(times
                FROM nsedata
                GROUP BY symbol
                ORDER BY trade_days_count DESC
Start time      Sun Oct 22 23:02:37 IST 2023
Finish time     Sun Oct 22 23:02:42 IST 2023
```