E9 - SQL Assignment

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

Solution:

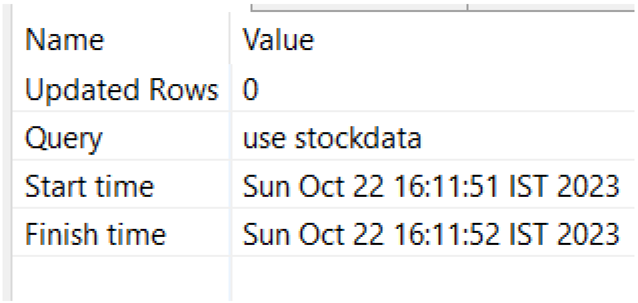
The nsedata table has the data related to many stocks in the National Stock Exchange. It provides data regarding 13 fields of each stock. These columns include:

- Series: The type of stock i.e whether it is an equity stock, derivatives, mutual funds, etc.
- Open: The price at which a security starts trading for the day
- High: The highest price reached during the day
- Low: The lowest price reached during the day
- Close: The final price of a security at the end of the training day
- Last: The most recent or the last traded price of a security
- PrevClose: The closing price from the previous trading session
- TotTrdQty: Total quantity of shares or contracts traded during a trading session
- TotTrdVal: Total value of shares or contracts traded during a trading session
- Timestamp: The time at which the data was recorded
- Anum:
- ISIN: Internation Securities Identification Number is a unique code used to identify securities
- Extra: Additional or supplementary information related to the data

2. Select the database stockdata using SQL

Solution:

```
use stockdata;
```

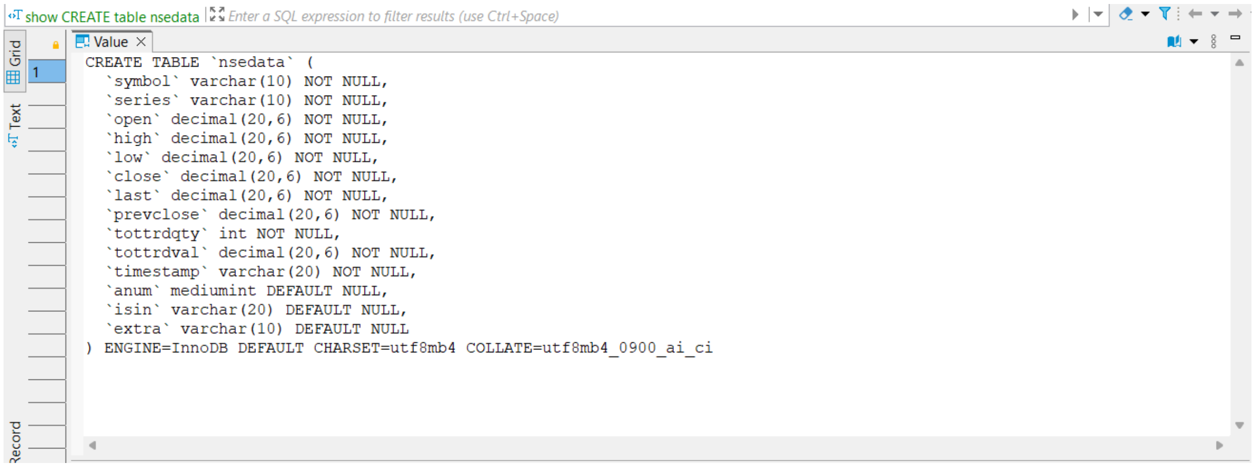


Name	Value
Updated Rows	0
Query	use stockdata
Start time	Sun Oct 22 16:11:51 IST 2023
Finish time	Sun Oct 22 16:11:52 IST 2023

3. Get a schema dump of the table nsedata

Solution:

```
show CREATE table nsedata;
```

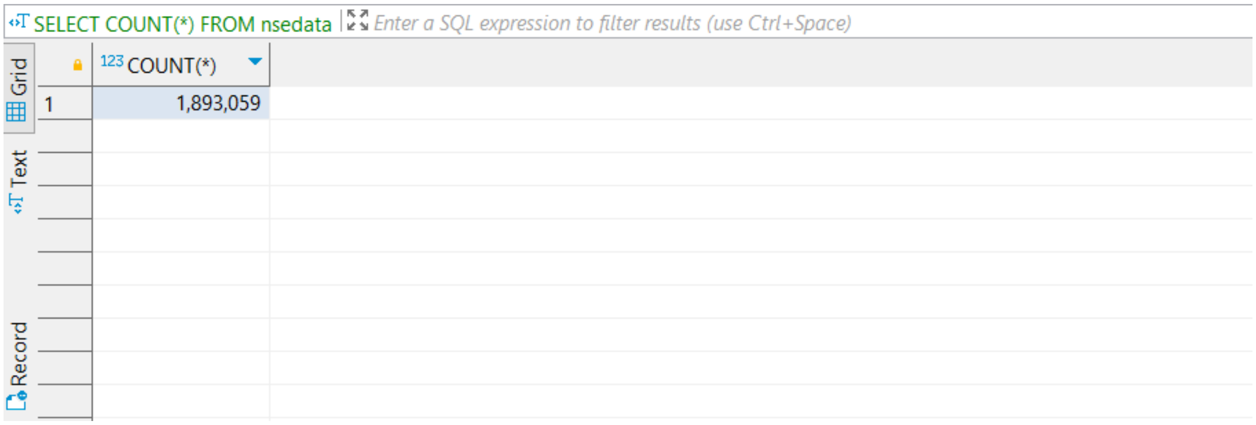

show CREATE table nsedata
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
Record
Value

```
CREATE TABLE `nsedata` (
  `symbol` varchar(10) NOT NULL,
  `series` varchar(10) NOT NULL,
  `open` decimal(20,6) NOT NULL,
  `high` decimal(20,6) NOT NULL,
  `low` decimal(20,6) NOT NULL,
  `close` decimal(20,6) NOT NULL,
  `last` decimal(20,6) NOT NULL,
  `prevclose` decimal(20,6) NOT NULL,
  `tottrdqty` int NOT NULL,
  `tottrdval` decimal(20,6) NOT NULL,
  `timestamp` varchar(20) NOT NULL,
  `anum` mediumint DEFAULT NULL,
  `isin` varchar(20) DEFAULT NULL,
  `extra` varchar(10) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

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

Solution:

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

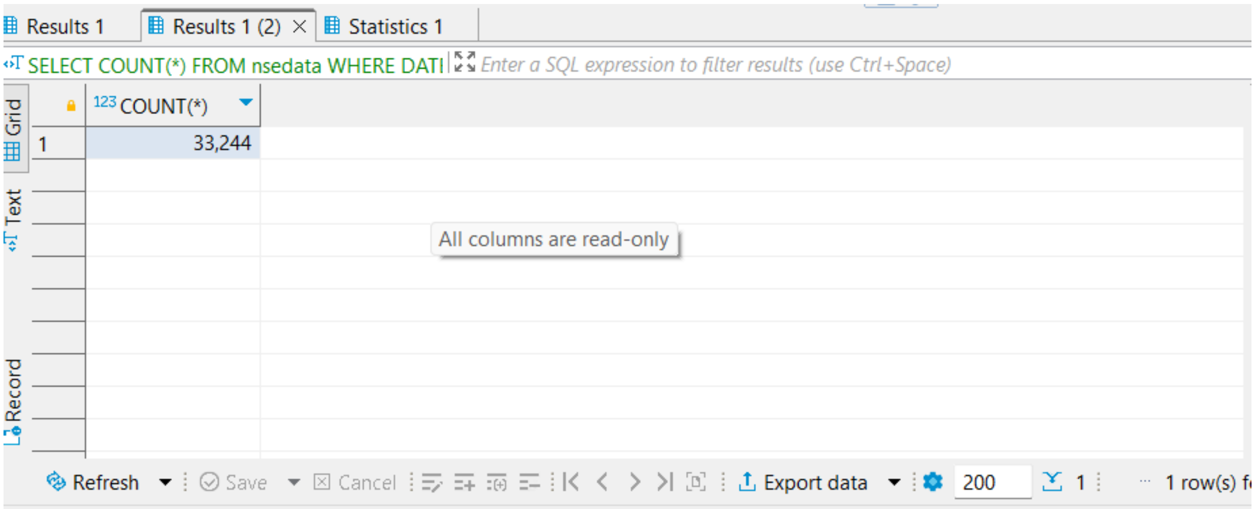

SELECT COUNT(*) FROM nsedata
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
Record

	123 COUNT(*)
1	1,893,059

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

Solution:

```
SELECT COUNT(*)
FROM nsedata
WHERE DATE_FORMAT(STR_TO_DATE(timestamp, '%d-%b-%Y'), '%m-%Y') = '10-2012';
```



SELECT COUNT(*) FROM nsedata WHERE DATI

	COUNT(*)
1	33,244

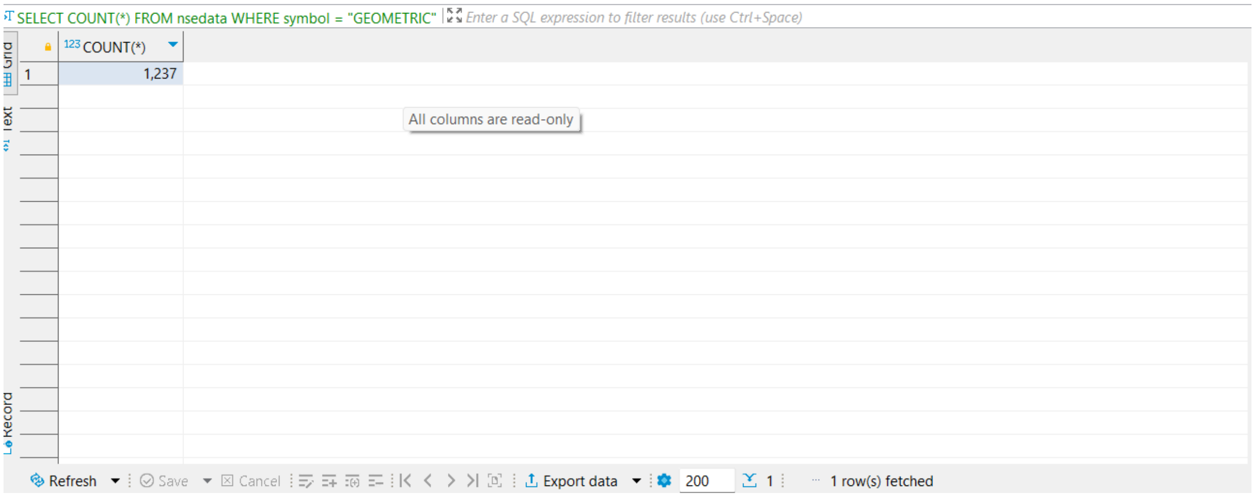
All columns are read-only

1 row(s) f

6. Repeat ‘4’, but only for the stock with symbol “GEOMETRIC”

Solution:

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



SELECT COUNT(*) FROM nsedata WHERE symbol = "GEOMETRIC"

	COUNT(*)
1	1,237

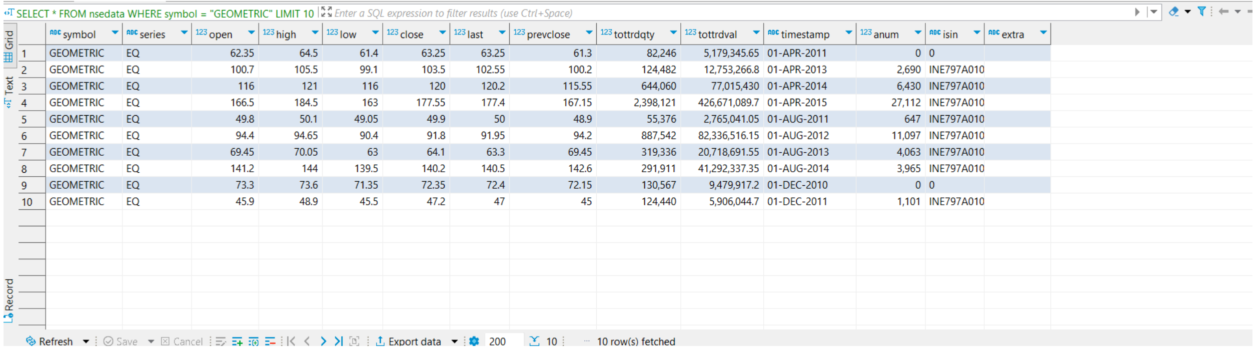
All columns are read-only

1 row(s) fetched

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

Solution:

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



SELECT * FROM nsedata WHERE symbol = "GEOMETRIC" LIMIT 10

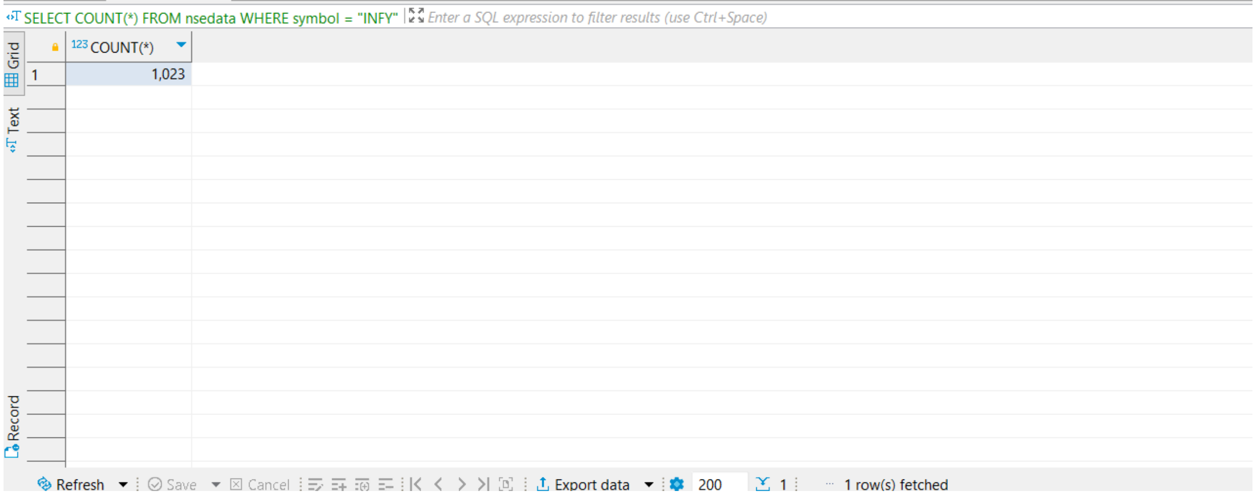
	symbol	series	open	high	low	close	last	prevclose	tottrdqty	tottrdval	timestamp	anum	isin	extra
1	GEOMETRIC	EQ	62.35	64.5	61.4	63.25	63.25	61.3	82,246	5,179,345.65	01-APR-2011	0	0	
2	GEOMETRIC	EQ	100.7	105.5	99.1	103.5	102.55	100.2	124,482	12,753,266.8	01-APR-2013	2,690	INE797A010	
3	GEOMETRIC	EQ	116	121	116	120	120.2	115.55	644,060	77,015,430	01-APR-2014	6,430	INE797A010	
4	GEOMETRIC	EQ	166.5	184.5	163	177.55	177.4	167.15	2,398,121	426,671,089.7	01-APR-2015	27,112	INE797A010	
5	GEOMETRIC	EQ	49.8	50.1	49.05	49.9	50	48.9	55,376	2,765,041.05	01-AUG-2011	647	INE797A010	
6	GEOMETRIC	EQ	94.4	94.65	90.4	91.8	91.95	94.2	887,542	82,336,516.15	01-AUG-2012	11,097	INE797A010	
7	GEOMETRIC	EQ	69.45	70.05	63	64.1	63.3	69.45	319,336	20,718,691.55	01-AUG-2013	4,063	INE797A010	
8	GEOMETRIC	EQ	141.2	144	139.5	140.2	140.5	142.6	291,911	41,292,337.35	01-AUG-2014	3,965	INE797A010	
9	GEOMETRIC	EQ	73.3	73.6	71.35	72.35	72.4	72.15	130,567	9,479,917.2	01-DEC-2010	0	0	
10	GEOMETRIC	EQ	45.9	48.9	45.5	47.2	47	45	124,440	5,906,044.7	01-DEC-2011	1,101	INE797A010	

10 row(s) fetched

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

Solution:

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



SELECT COUNT(*) FROM nsedata WHERE symbol = "INFY"

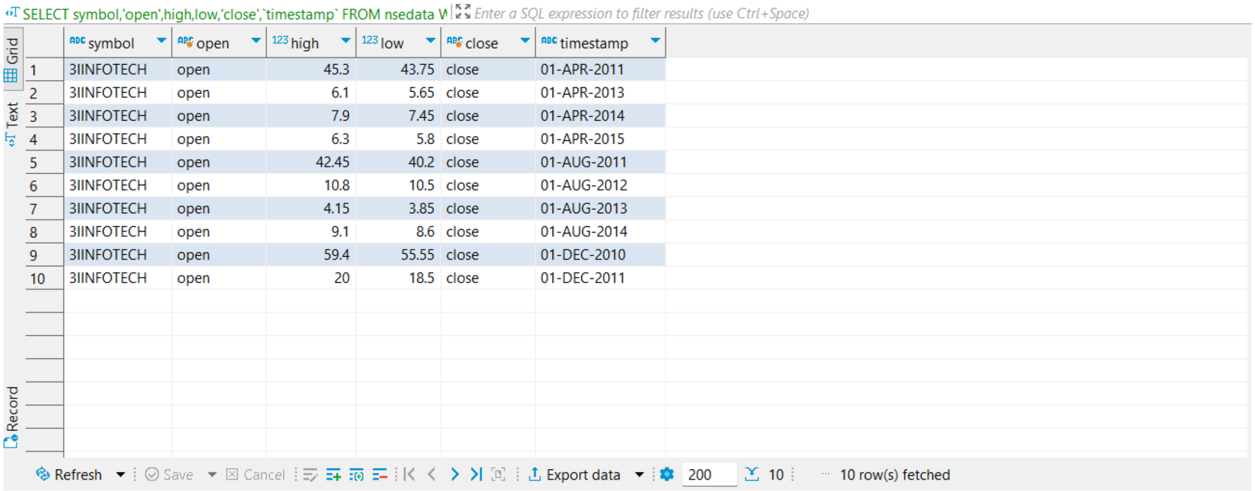
	COUNT(*)
1	1,023

1 row(s) fetched

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

Solution:

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



SELECT symbol,'open',high,low,'close',`timestamp` FROM nsedata W

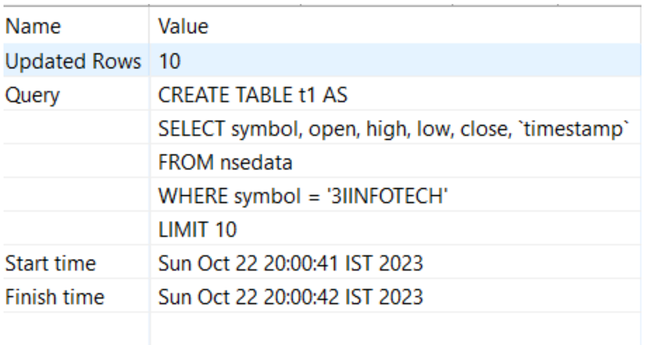
	symbol	open	high	low	close	timestamp
1	3IINFOTECH	open	45.3	43.75	close	01-APR-2011
2	3IINFOTECH	open	6.1	5.65	close	01-APR-2013
3	3IINFOTECH	open	7.9	7.45	close	01-APR-2014
4	3IINFOTECH	open	6.3	5.8	close	01-APR-2015
5	3IINFOTECH	open	42.45	40.2	close	01-AUG-2011
6	3IINFOTECH	open	10.8	10.5	close	01-AUG-2012
7	3IINFOTECH	open	4.15	3.85	close	01-AUG-2013
8	3IINFOTECH	open	9.1	8.6	close	01-AUG-2014
9	3IINFOTECH	open	59.4	55.55	close	01-DEC-2010
10	3IINFOTECH	open	20	18.5	close	01-DEC-2011

10 row(s) fetched

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

Solution:

```
CREATE TABLE t1 AS
SELECT symbol, open, high, low, close, `timestamp`
FROM nsedata
WHERE symbol = '3IINFOTECH'
LIMIT 10;
```

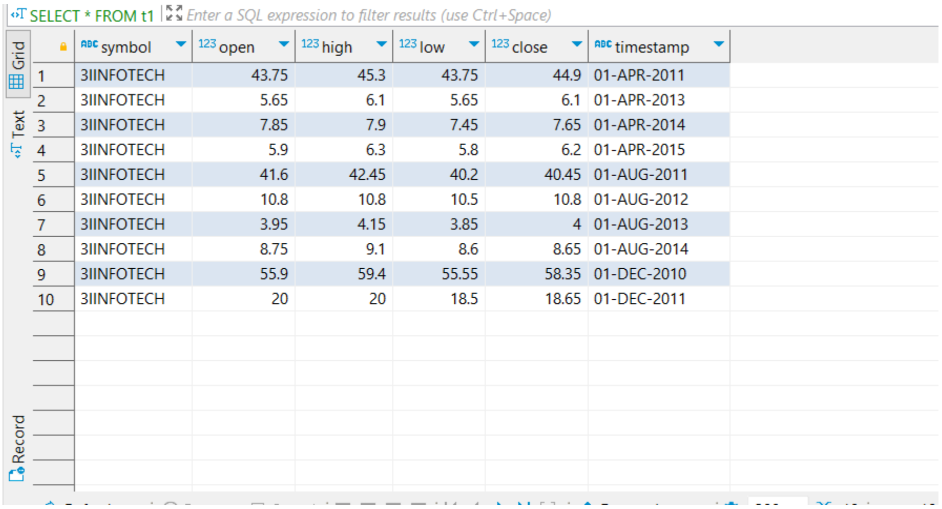


Name	Value
Updated Rows	10

Query	CREATE TABLE t1 AS
	SELECT symbol, open, high, low, close, `timestamp`
	FROM nsedata
	WHERE symbol = '3IINFOTECH'
	LIMIT 10

Start time	Sun Oct 22 20:00:41 IST 2023
Finish time	Sun Oct 22 20:00:42 IST 2023

This is how the table looks



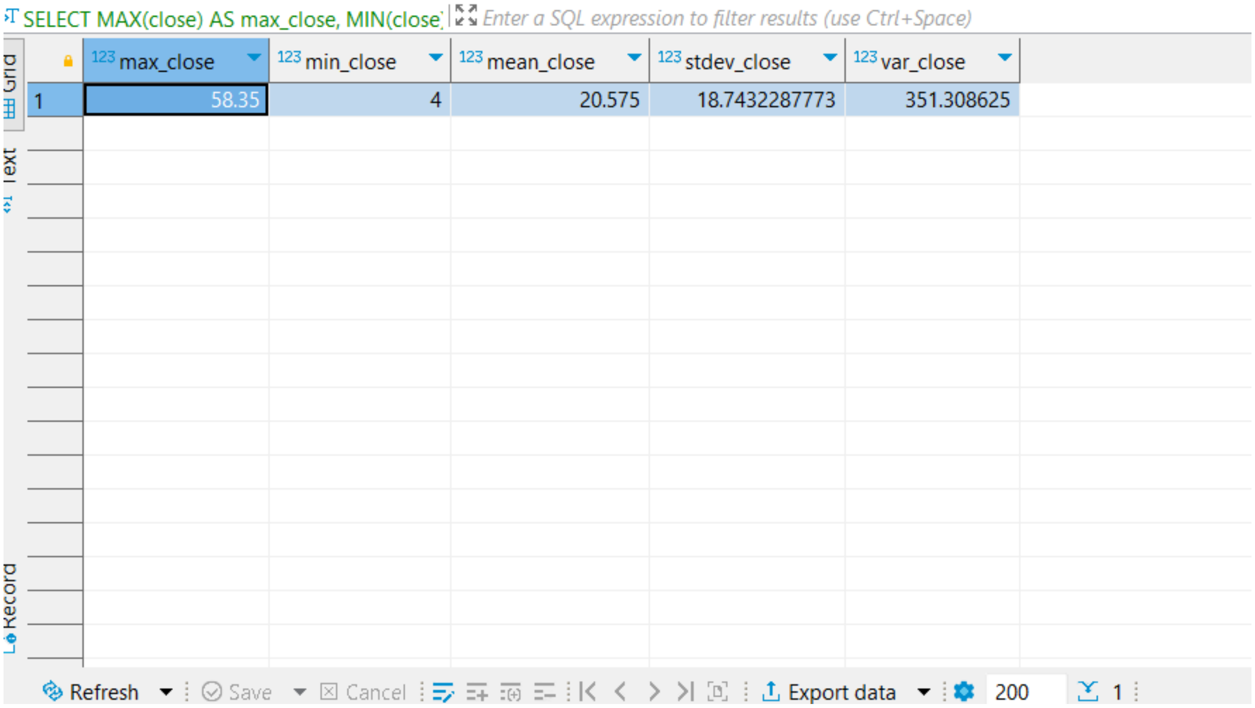
SELECT * FROM t1

	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

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

Solution:

```
SELECT
MAX(close) AS max_close,
MIN(close) AS min_close,
AVG(close) AS mean_close,
SQRT(AVG((close - mean_close) * (close - mean_close))) AS stdev_close,
AVG((close - mean_close) * (close - mean_close)) AS var_close
FROM (
SELECT close, (SELECT AVG(close) FROM t1) AS mean_close
FROM t1
) subquery;
```



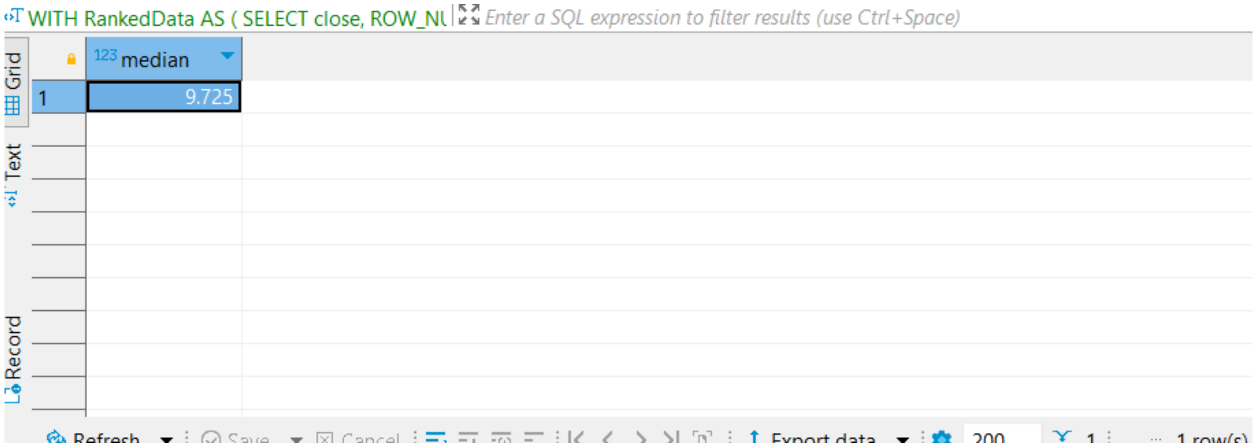
SELECT MAX(close) AS max_close, MIN(close

	max_close	min_close	mean_close	stdev_close	var_close
1	58.35	4	20.575	18.7432287773	351.308625

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

Solution:

```
WITH RankedData AS (
SELECT close, ROW_NUMBER() OVER (ORDER BY close) AS RowAsc,
ROW_NUMBER() OVER (ORDER BY close DESC) AS RowDesc
FROM t1
)
SELECT AVG(close) AS median
FROM RankedData
WHERE RowAsc = RowDesc
OR RowAsc + 1 = RowDesc
OR RowAsc = RowDesc + 1;
```



WITH RankedData AS (SELECT close, ROW_NU

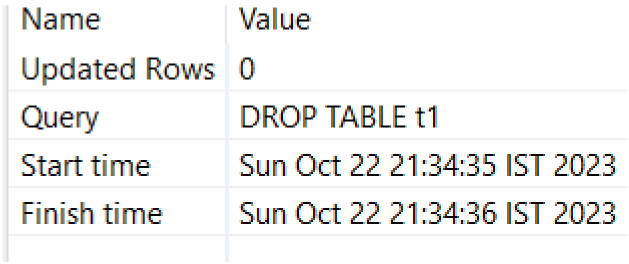
Enter a SQL expression to filter results (use Ctrl+Space)

	median
1	9.725

13. Delete table t1

Solution:

```
DROP TABLE t1;
```

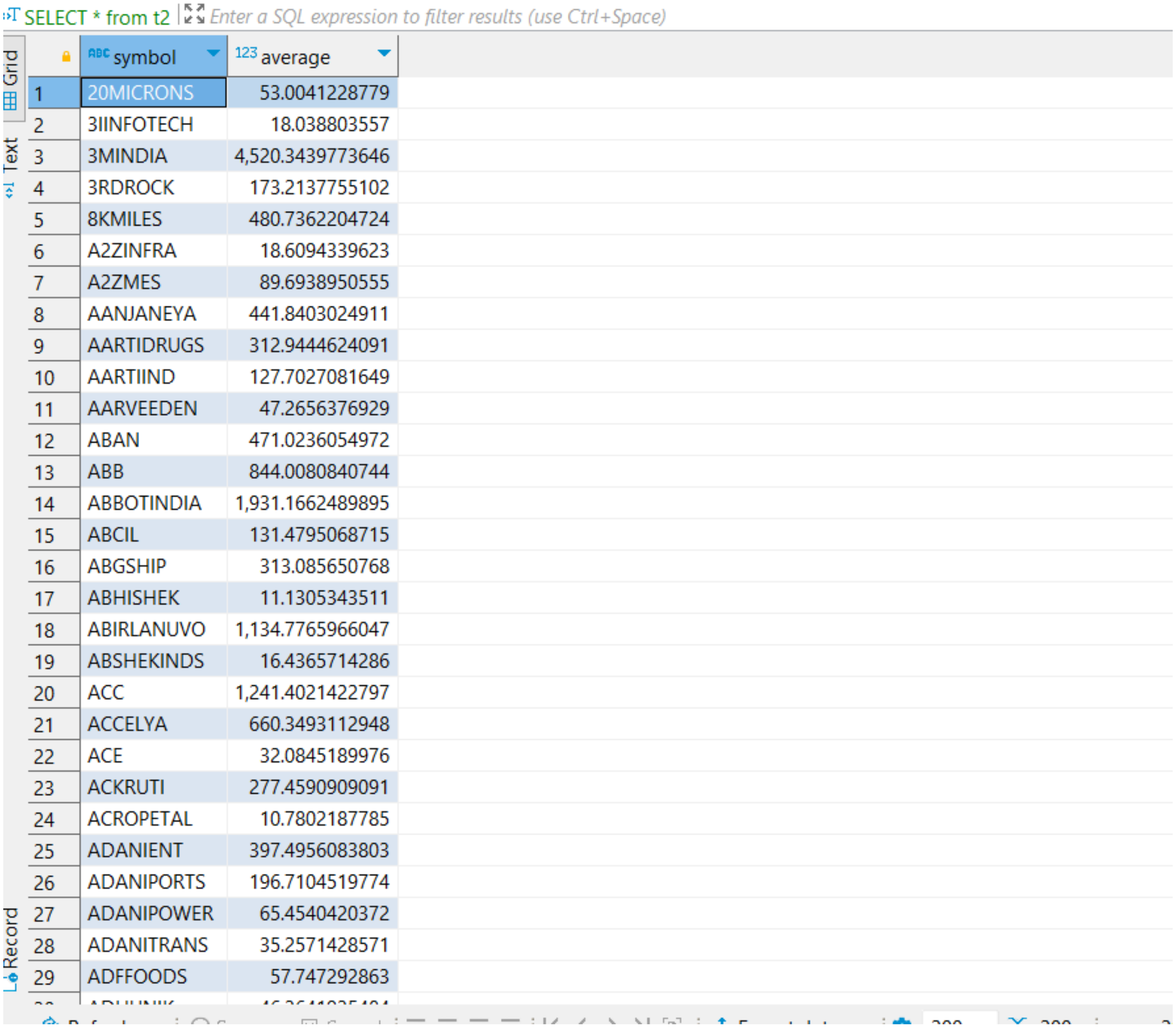


Name	Value
Updated Rows	0
Query	DROP TABLE t1
Start time	Sun Oct 22 21:34:35 IST 2023
Finish time	Sun Oct 22 21:34:36 IST 2023

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

Solution:

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



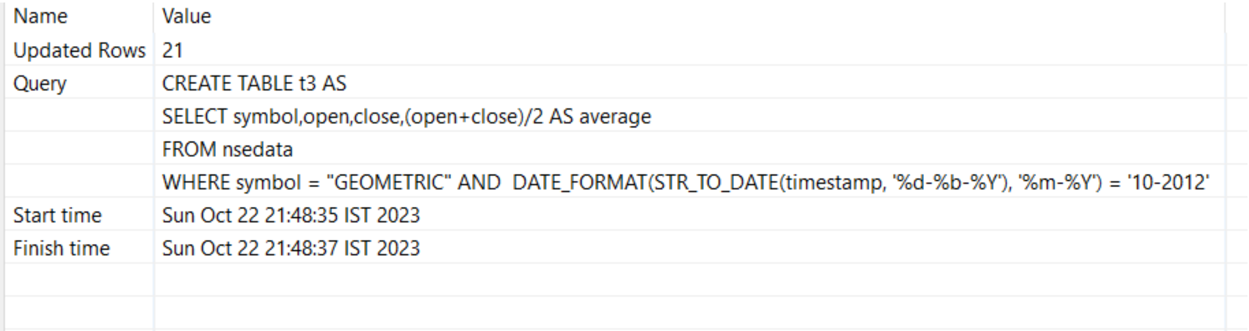
```
SELECT * from t2
```

	symbol	average
1	20MICRONS	53.0041228779
2	3IINFOTECH	18.038803557
3	3MINDIA	4,520.3439773646
4	3RDROCK	173.2137755102
5	8KMILES	480.7362204724
6	A2ZINFRA	18.6094339623
7	A2ZMES	89.6938950555
8	AANJANEYA	441.8403024911
9	AARTIDRUGS	312.9444624091
10	AARTIIND	127.7027081649
11	AARVEEDEN	47.2656376929
12	ABAN	471.0236054972
13	ABB	844.0080840744
14	ABBOTINDIA	1,931.1662489895
15	ABCIL	131.4795068715
16	ABGSHIP	313.085650768
17	ABHISHEK	11.1305343511
18	ABIRLANUVO	1,134.7765966047
19	ABSHEKINDS	16.4365714286
20	ACC	1,241.4021422797
21	ACCELYA	660.3493112948
22	ACE	32.0845189976
23	ACKRUTI	277.4590909091
24	ACROPETAL	10.7802187785
25	ADANIENT	397.4956083803
26	ADANIPORTS	196.7104519774
27	ADANIPOWER	65.4540420372
28	ADANITRANS	35.2571428571
29	ADFFOODS	57.747292863

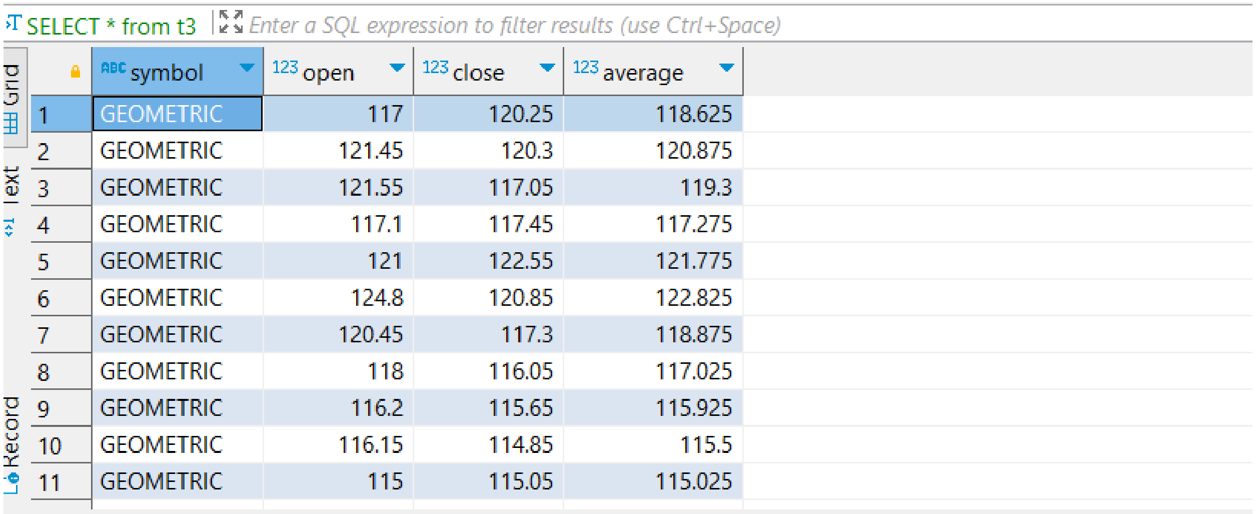
15. Create a table t3 such that it contains the following columns: symbol, open, close, "average of open and close". Fill up this table for the company GEOMETRIC, for the month of October 2012.

Solution:

```
CREATE TABLE t3 AS
SELECT symbol,open,close,(open+close)/2 AS average
FROM nsedata
WHERE symbol = "GEOMETRIC" AND DATE_FORMAT(STR_TO_DATE(timestamp, '%d-%b-%Y'),
'%m-%Y') = '10-2012';
```



Name	Value
Updated Rows	21
Query	CREATE TABLE t3 AS
	SELECT symbol,open,close,(open+close)/2 AS average
	FROM nsedata
	WHERE symbol = "GEOMETRIC" AND DATE_FORMAT(STR_TO_DATE(timestamp, '%d-%b-%Y'), '%m-%Y') = '10-2012'
Start time	Sun Oct 22 21:48:35 IST 2023
Finish time	Sun Oct 22 21:48:37 IST 2023



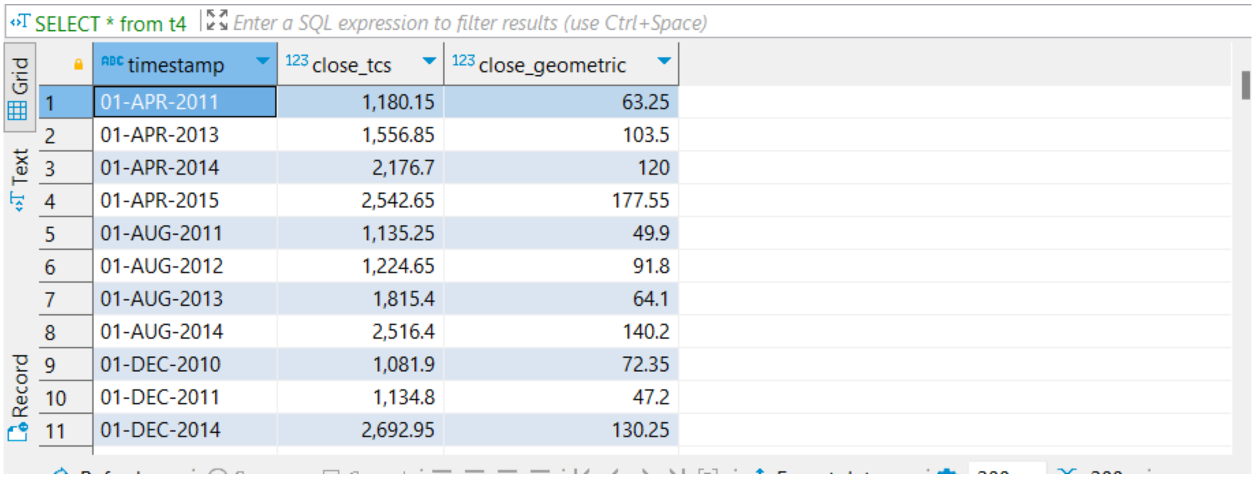
SELECT * from t3

	symbol	open	close	average
1	GEOMETRIC	117	120.25	118.625
2	GEOMETRIC	121.45	120.3	120.875
3	GEOMETRIC	121.55	117.05	119.3
4	GEOMETRIC	117.1	117.45	117.275
5	GEOMETRIC	121	122.55	121.775
6	GEOMETRIC	124.8	120.85	122.825
7	GEOMETRIC	120.45	117.3	118.875
8	GEOMETRIC	118	116.05	117.025
9	GEOMETRIC	116.2	115.65	115.925
10	GEOMETRIC	116.15	114.85	115.5
11	GEOMETRIC	115	115.05	115.025

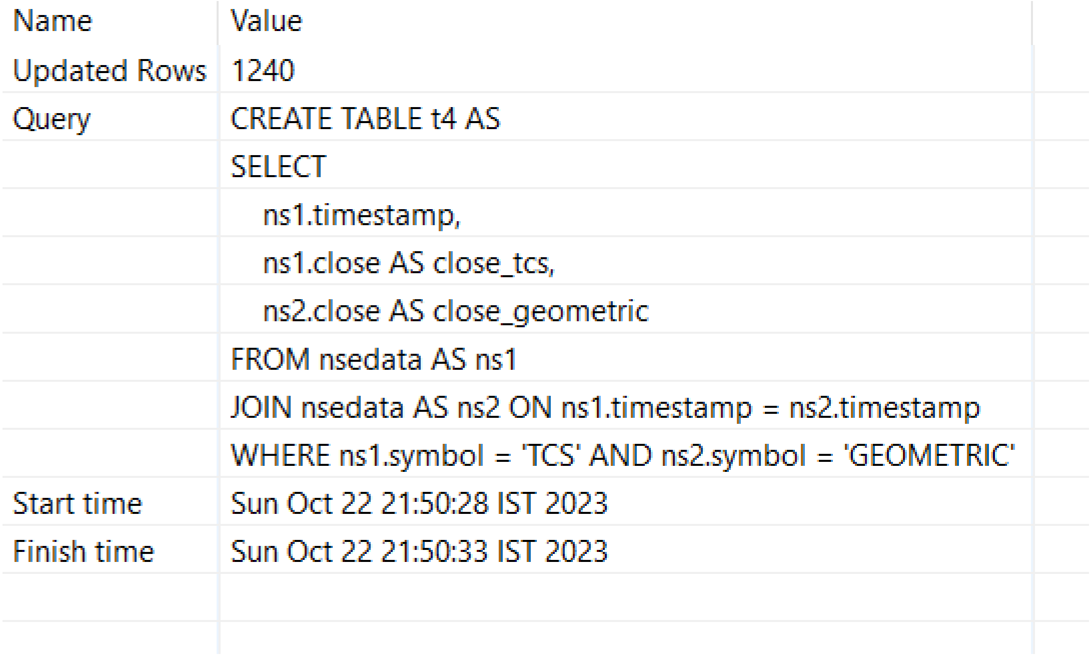
16. It is required to create a table t4 such that it contains the data for two companies GEOMETRIC and TCS. The columns of this table should be as follows: timestamp, close_tcs, close_geometric. Hint: use JOIN

Solution:

```
CREATE TABLE t4 AS
SELECT
ns1.timestamp,
ns1.close AS close_tcs,
ns2.close AS close_geometric
FROM nsedata AS ns1
JOIN nsedata AS ns2 ON ns1.timestamp = ns2.timestamp
WHERE ns1.symbol = 'TCS' AND ns2.symbol = 'GEOMETRIC';
```



	timestamp	close_tcs	close_geometric
1	01-APR-2011	1,180.15	63.25
2	01-APR-2013	1,556.85	103.5
3	01-APR-2014	2,176.7	120
4	01-APR-2015	2,542.65	177.55
5	01-AUG-2011	1,135.25	49.9
6	01-AUG-2012	1,224.65	91.8
7	01-AUG-2013	1,815.4	64.1
8	01-AUG-2014	2,516.4	140.2
9	01-DEC-2010	1,081.9	72.35
10	01-DEC-2011	1,134.8	47.2
11	01-DEC-2014	2,692.95	130.25

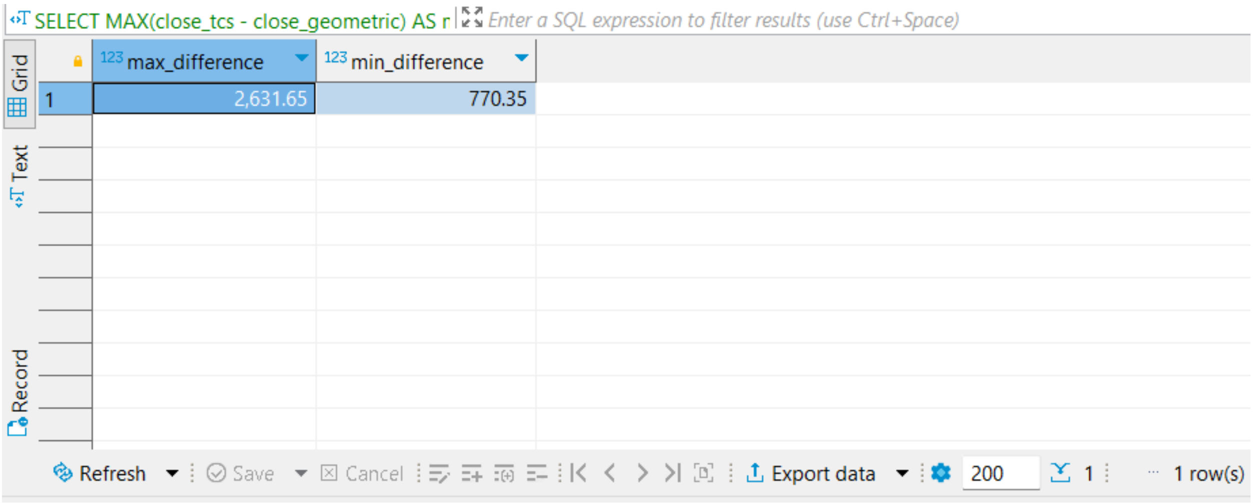


Name	Value
Updated Rows	1240
Query	CREATE TABLE t4 AS
	SELECT
	ns1.timestamp,
	ns1.close AS close_tcs,
	ns2.close AS close_geometric
	FROM nsedata AS ns1
	JOIN nsedata AS ns2 ON ns1.timestamp = ns2.timestamp
	WHERE ns1.symbol = 'TCS' AND ns2.symbol = 'GEOMETRIC'
Start time	Sun Oct 22 21:50:28 IST 2023
Finish time	Sun Oct 22 21:50:33 IST 2023

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

Solution:

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



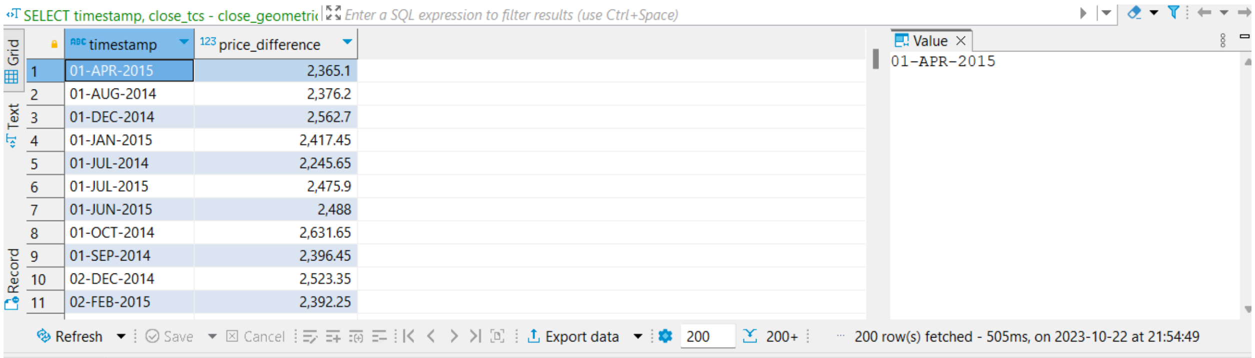
SELECT MAX(close_tcs - close_geometric) AS r

	max_difference	min_difference
1	2,631.65	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.

Solution:

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



SELECT timestamp, close_tcs - close_geometri

	timestamp	price_difference
1	01-APR-2015	2,365.1
2	01-AUG-2014	2,376.2
3	01-DEC-2014	2,562.7
4	01-JAN-2015	2,417.45
5	01-JUL-2014	2,245.65
6	01-JUL-2015	2,475.9
7	01-JUN-2015	2,488
8	01-OCT-2014	2,631.65
9	01-SEP-2014	2,396.45
10	02-DEC-2014	2,523.35
11	02-FEB-2015	2,392.25

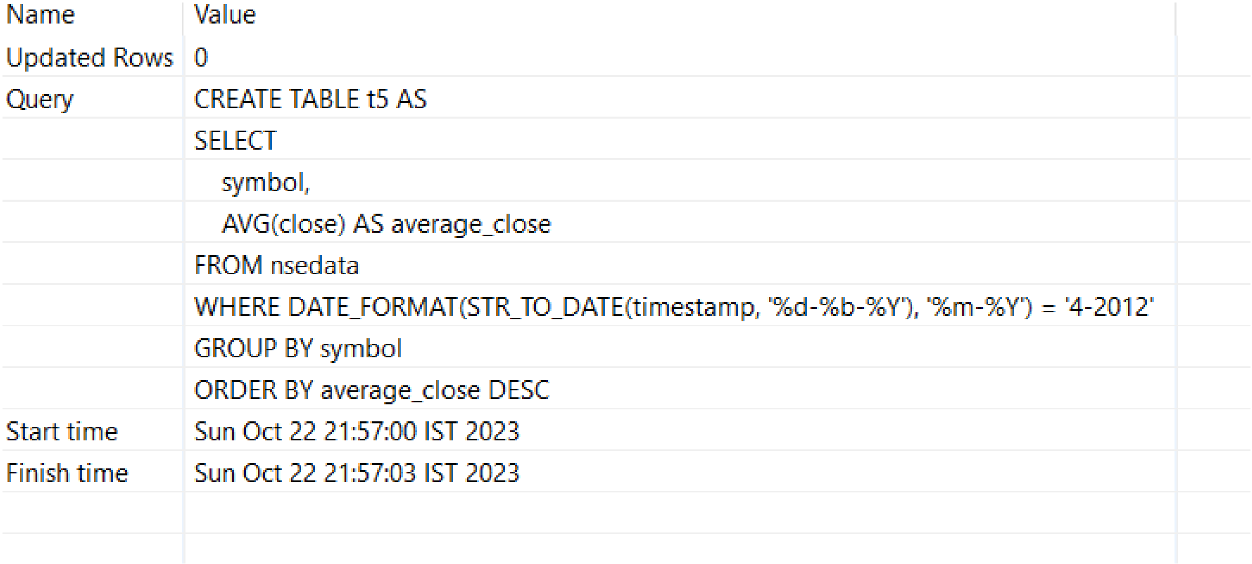
Value
01-APR-2015

200 row(s) fetched - 505ms, on 2023-10-22 at 21:54:49

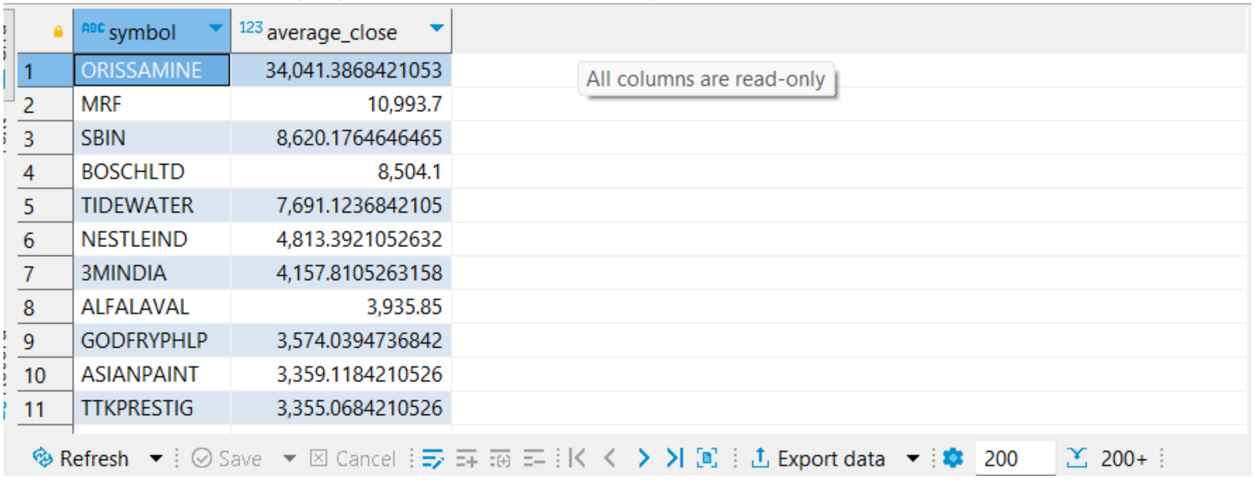
19. Based on nsedata, create table t5 such that it contains the average close price of each company traded in the month of April 2012. The table should be sorted in descending order of the average close price.

Solution:

```
CREATE TABLE t5 AS
SELECT
symbol,
AVG(close) AS average_close
FROM nsedata
WHERE DATE_FORMAT(STR_TO_DATE(timestamp, '%d-%b-%Y'), '%m-%Y') = '04-2012'
GROUP BY symbol
ORDER BY average_close DESC;
```



Name	Value
Updated Rows	0
Query	CREATE TABLE t5 AS
	SELECT
	symbol,
	AVG(close) AS average_close
	FROM nsedata
	WHERE DATE_FORMAT(STR_TO_DATE(timestamp, '%d-%b-%Y'), '%m-%Y') = '4-2012'
	GROUP BY symbol
	ORDER BY average_close DESC
Start time	Sun Oct 22 21:57:00 IST 2023
Finish time	Sun Oct 22 21:57:03 IST 2023

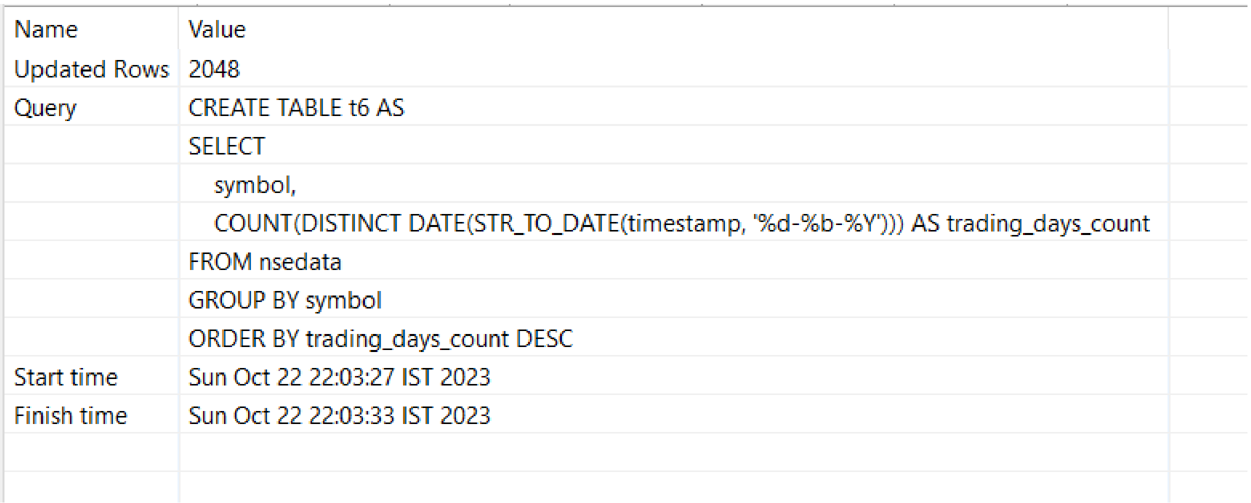


	symbol	average_close
1	ORISSAMINE	34,041.3868421053
2	MRF	10,993.7
3	SBIN	8,620.1764646465
4	BOSCHLTD	8,504.1
5	TIDEWATER	7,691.1236842105
6	NESTLEIND	4,813.3921052632
7	3MINDIA	4,157.8105263158
8	ALFALAVAL	3,935.85
9	GODFRYPHLP	3,574.0394736842
10	ASIANPAINT	3,359.1184210526
11	TTKPRESTIG	3,355.0684210526

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.

Solution:

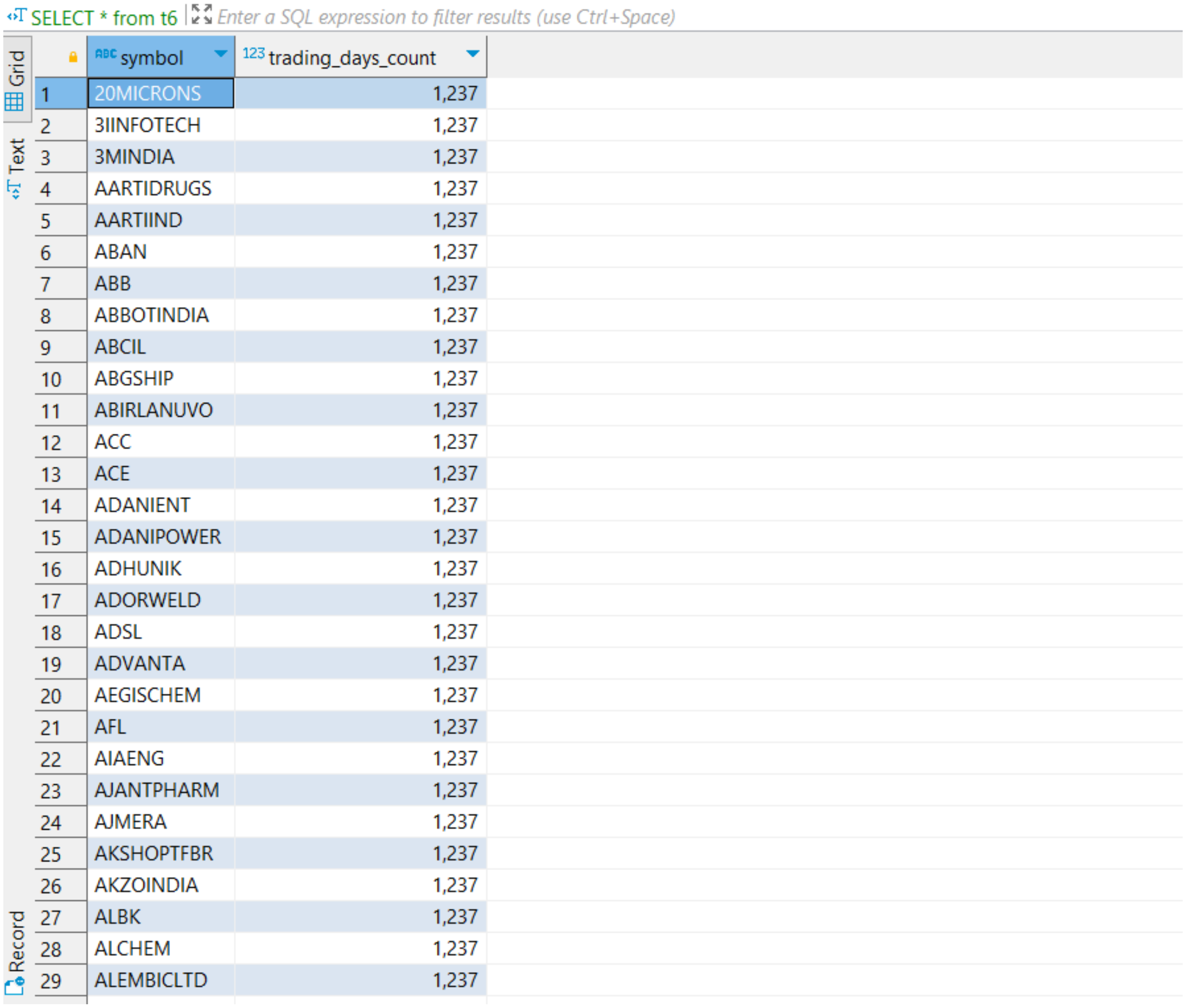
```
CREATE TABLE t6 AS
SELECT
symbol,
COUNT(DISTINCT DATE(STR_TO_DATE(timestamp, '%d-%b-%Y'))) AS trading_days_count
FROM nsedata
GROUP BY symbol
ORDER BY trading_days_count DESC;
```



Name	Value
Updated Rows	2048

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

Start time	Sun Oct 22 22:03:27 IST 2023
Finish time	Sun Oct 22 22:03:33 IST 2023



	symbol	trading_days_count
1	20MICRONS	1,237
2	3IINFOTECH	1,237
3	3MINDIA	1,237
4	AARTIDRUGS	1,237
5	AARTIIND	1,237
6	ABAN	1,237
7	ABB	1,237
8	ABBOTINDIA	1,237
9	ABCIL	1,237
10	ABGSHIP	1,237
11	ABIRLANUVO	1,237
12	ACC	1,237
13	ACE	1,237
14	ADANIENT	1,237
15	ADANIPOWER	1,237
16	ADHUNIK	1,237
17	ADORWELD	1,237
18	ADSL	1,237
19	ADVANTA	1,237
20	AEGISCHEM	1,237
21	AFL	1,237
22	AIAENG	1,237
23	AJANTPHARM	1,237
24	AJMERA	1,237
25	AKSHOPTFBR	1,237
26	AKZOINDIA	1,237
27	ALBK	1,237
28	ALCHEM	1,237
29	ALEMBICLTD	1,237