Exercise 8

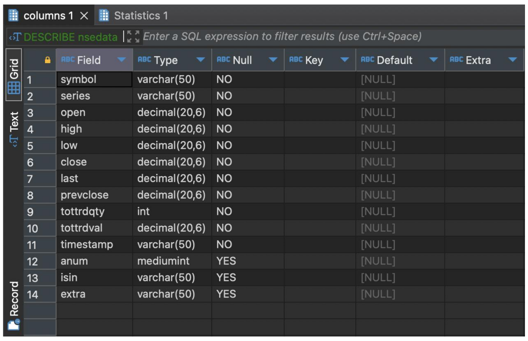
1. Do a manual review of the table nsedata and describe its contents (no SQL to be executed for this task) .

The table nsedata contains the following columns:

- symbol: Represents the symbol or abbreviation of the company whose stock data is recorded.
- series: Denotes the series of the stock.
- open: Indicates the opening price of the stock for a particular timestamp.
- high: Represents the highest price of the stock reached during the trading period.
- low: Represents the lowest price of the stock reached during the trading period.
- close: Denotes the closing price of the stock for a particular timestamp.
- last: Indicates the last traded price of the stock.
- prevclose: Represents the previous day's closing price of the stock.
- tottrdqty: Represents the total quantity of stocks traded.
- tottrdval: Represents the total value of stocks traded.
- timestamp: Represents the timestamp indicating the date and time of the recorded data.
- anum: Initially set to 0, possibly to represent a numerical identifier.
- isin: Initially set to 0, possibly to represent the International Securities Identification Number (ISIN) of the stock.
- extra: An empty column, possibly for future additional data or purposes.

2. Select the database stockdata using SQL
3. Get a schema dump of the table nsedata using SQL

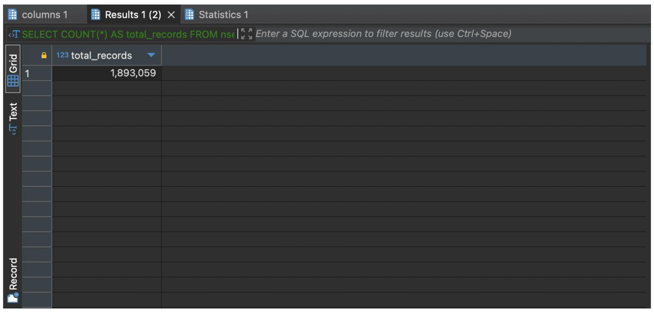
```
use stockdata;
DESCRIBE nsedata;
```



	Field	Type	Null	Key	Default	Extra
1	symbol	varchar(50)	NO		[NULL]	
2	series	varchar(50)	NO		[NULL]	
3	open	decimal(20,6)	NO		[NULL]	
4	high	decimal(20,6)	NO		[NULL]	
5	low	decimal(20,6)	NO		[NULL]	
6	close	decimal(20,6)	NO		[NULL]	
7	last	decimal(20,6)	NO		[NULL]	
8	prevclose	decimal(20,6)	NO		[NULL]	
9	tottrdqty	int	NO		[NULL]	
10	tottrdval	decimal(20,6)	NO		[NULL]	
11	timestamp	varchar(50)	NO		[NULL]	
12	anum	mediumint	YES		[NULL]	
13	isin	varchar(50)	YES		[NULL]	
14	extra	varchar(50)	YES		[NULL]	

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

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

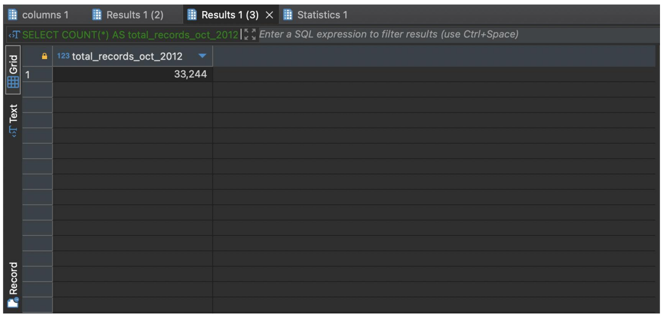


```
SELECT COUNT(*) AS total_records FROM nse
Enter a SQL expression to filter results (use Ctrl+Space)
```

	total_records
1	1,893,059

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

```sql
SELECT COUNT(*) AS total_records_oct_2012
FROM nsedata
WHERE YEAR(STR_TO_DATE(timestamp, '%d-%b-%Y')) = 2012
AND MONTH(STR_TO_DATE(timestamp, '%d-%b-%Y')) = 10;
```

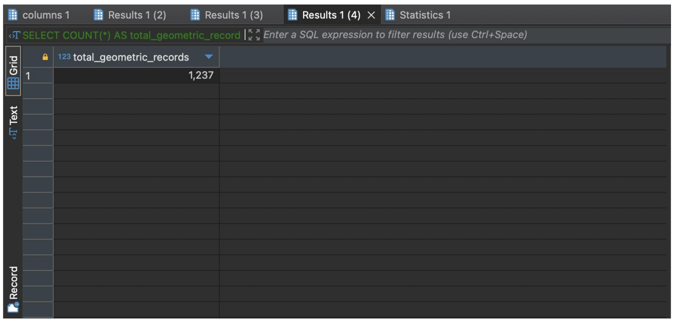


```
SELECT COUNT(*) AS total_records_oct_2012
Enter a SQL expression to filter results (use Ctrl+Space)
```

	total_records_oct_2012
1	33,244

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

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

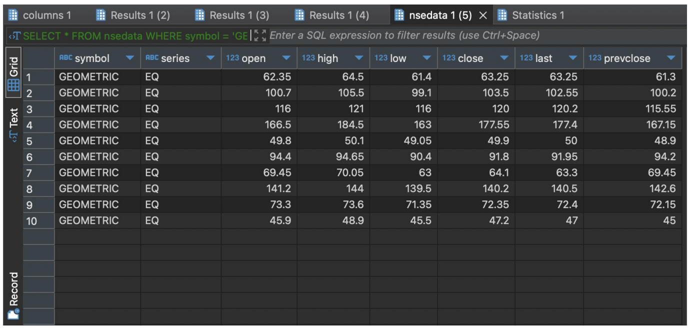


```
SELECT COUNT(*) AS total_geometric_record
Enter a SQL expression to filter results (use Ctrl+Space)
```

	total_geometric_records
1	1,237

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

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

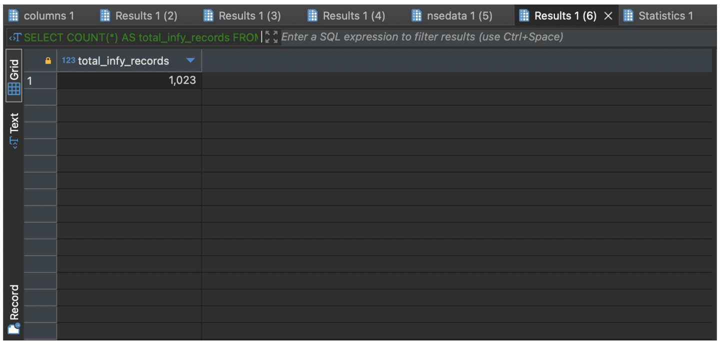


```
SELECT * FROM nsedata WHERE symbol = 'GE
```

	symbol	series	open	high	low	close	last	prevclose
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(*) AS total_infy_records
FROM nsedata
WHERE symbol = 'INFY';
```

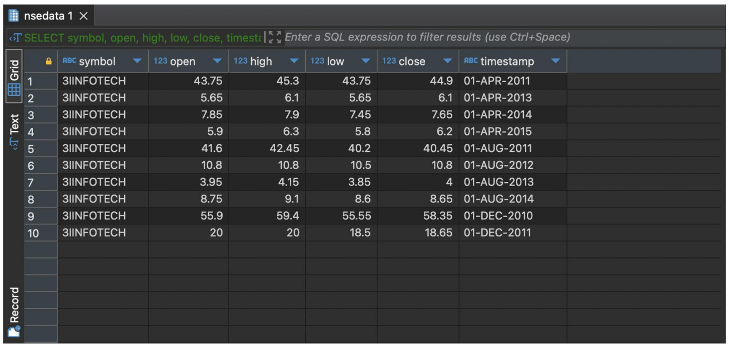


```
SELECT COUNT(*) AS total_infy_records FROI
```

	total_infy_records
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;
```

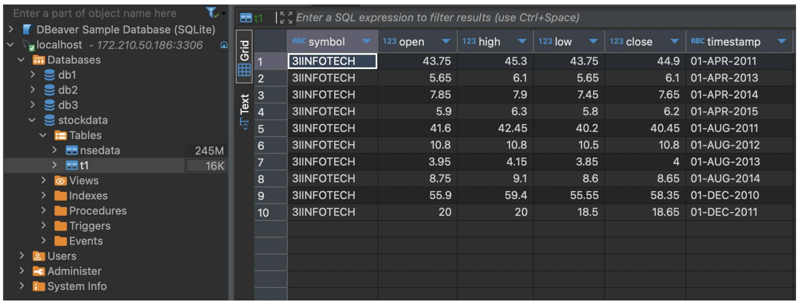


```
SELECT symbol, open, high, low, close, timest:
```

	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 table t1 in the stockdata database

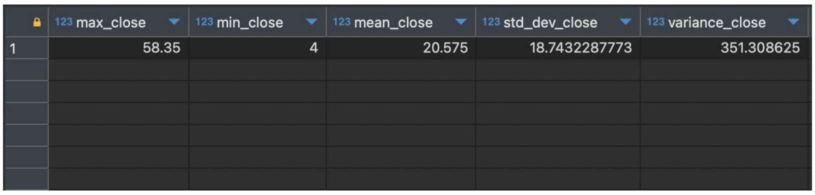
```
CREATE TABLE stockdata.t1 AS
SELECT symbol, open, high, low, close, timestamp
FROM nsedata
WHERE symbol = '3IINFOTECH'
LIMIT 10;
```


Enter a part of object name here
DBeaver Sample Database (SQLite)
localhost - 172.210.50.186:3306
Databases
db1
db2
db3
stockdata
Tables
nsedata 245M
t1 16K
Views
Indexes
Procedures
Triggers
Events
Users
Administer
System Info
t1
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text

	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

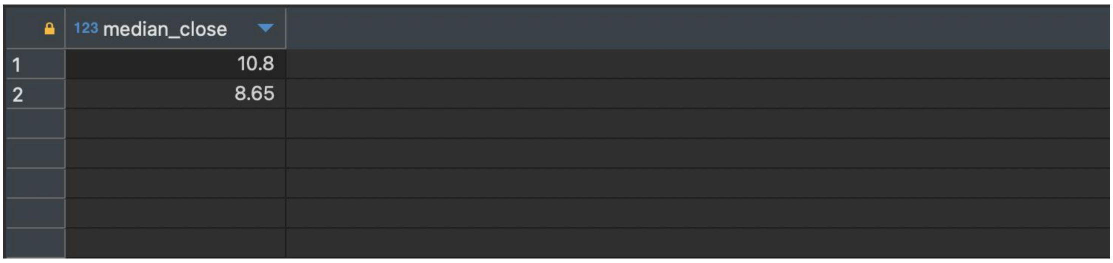
```
SELECT
MAX(close) AS max_close,
MIN(close) AS min_close,
AVG(close) AS mean_close,
STDDEV(close) AS std_dev_close,
VARIANCE(close) AS variance_close
FROM t1;
```



	max_close	min_close	mean_close	std_dev_close	variance_close
1	58.35	4	20.575	18.7432287773	351.308625

12. How will you find out the value of the median, if that is also required?

```
SELECT close AS median_close
FROM t1
ORDER BY close DESC
LIMIT 4, 2;
```



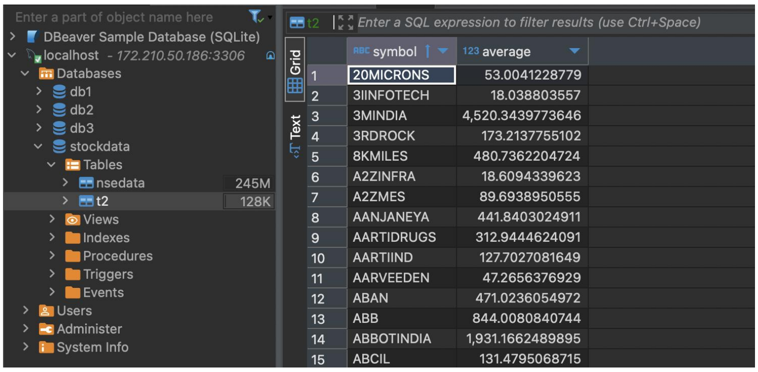
	median_close
1	10.8
2	8.65

13. Delete table t1

```
DROP TABLE t1;
```

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

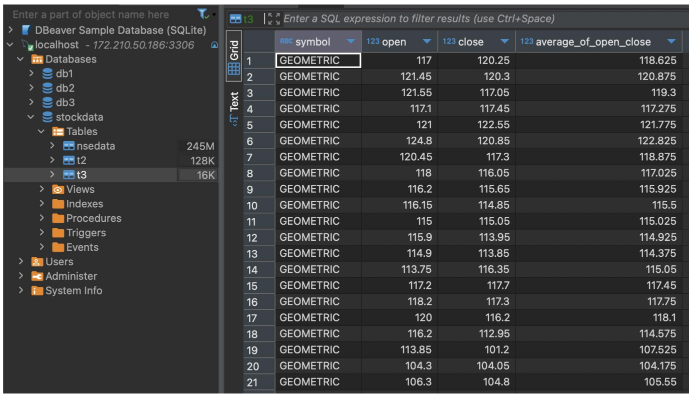
```sql
CREATE TABLE t2 AS
SELECT symbol, AVG(close) AS average
FROM nsedata
GROUP BY symbol;
```


Enter a part of object name here
DBeaver Sample Database (SQLite)
localhost - 172.210.50.186:3306
Databases
db1
db2
db3
stockdata
Tables
nsedata 245M
t2 128K
Views
Indexes
Procedures
Triggers
Events
Users
Administer
System Info
t2
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text

	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

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.

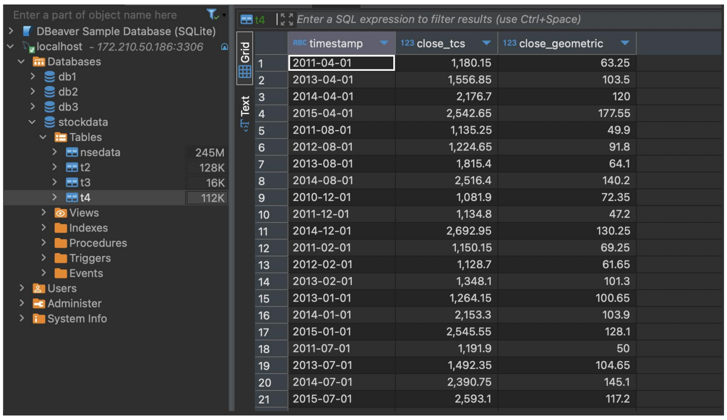
```sql
SELECT COUNT(*) AS total_records_oct_2012
FROM nsedata
WHERE YEAR(timestamp) = 2012
AND MONTH(timestamp) = 10;
CREATE TABLE t3 AS
SELECT
    symbol,
    open,
    close,
    (open + close) / 2 AS average_of_open_close
FROM
    nsedata
WHERE
    symbol = 'GEOMETRIC'
    AND YEAR(timestamp) = 2012
    AND MONTH(timestamp) = 10;
```


Enter a part of object name here
DBeaver Sample Database (SQLite)
localhost - 172.210.50.186:3306
Databases
db1
db2
db3
stockdata
Tables
nsedata 245M
t2 128K
t3 16K
Views
Indexes
Procedures
Triggers
Events
Users
Administer
System Info
t3
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text

	symbol	open	close	average_of_open_close
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
12	GEOMETRIC	115.9	113.95	114.925
13	GEOMETRIC	114.9	113.85	114.375
14	GEOMETRIC	113.75	116.35	115.05
15	GEOMETRIC	117.2	117.7	117.45
16	GEOMETRIC	118.2	117.3	117.75
17	GEOMETRIC	120	116.2	118.1
18	GEOMETRIC	116.2	112.95	114.575
19	GEOMETRIC	113.85	101.2	107.525
20	GEOMETRIC	104.3	104.05	104.175
21	GEOMETRIC	106.3	104.8	105.55

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

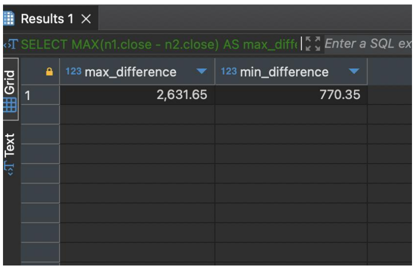
```
CREATE TABLE t4 AS
SELECT
     n1.timestamp,
     n1.close AS close_tcs,
     n2.close AS close_geometric
FROM
     nsedata n1
JOIN
     nsedata n2 ON n1.timestamp = n2.timestamp
WHERE
     n1.symbol = 'TCS'
     AND n2.symbol = 'GEOMETRIC';
```


Enter a part of object name here
DBeaver Sample Database (SQLite)
localhost - 172.210.50.186:3306
Databases
db1
db2
db3
stockdata
Tables
nsedata 245M
t2 128K
t3 16K
t4 112K
Views
Indexes
Procedures
Triggers
Events
Users
Administer
System Info
t4
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text

	timestamp	close_tcs	close_geometric
1	2011-04-01	1,180.15	63.25
2	2013-04-01	1,556.85	103.5
3	2014-04-01	2,176.7	120
4	2015-04-01	2,542.65	177.55
5	2011-08-01	1,135.25	49.9
6	2012-08-01	1,224.65	91.8
7	2013-08-01	1,815.4	64.1
8	2014-08-01	2,516.4	140.2
9	2010-12-01	1,081.9	72.35
10	2011-12-01	1,134.8	47.2
11	2014-12-01	2,692.95	130.25
12	2011-02-01	1,150.15	69.25
13	2012-02-01	1,128.7	61.65
14	2013-02-01	1,348.1	101.3
15	2013-01-01	1,264.15	100.65
16	2014-01-01	2,153.3	103.9
17	2015-01-01	2,545.55	128.1
18	2011-07-01	1,191.9	50
19	2013-07-01	1,492.35	104.65
20	2014-07-01	2,390.75	145.1
21	2015-07-01	2,593.1	117.2

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

```
SELECT
     MAX(n1.close - n2.close) AS max_difference,
     MIN(n1.close - n2.close) AS min_difference
FROM
     nsedata n1
JOIN
     nsedata n2 ON n1.timestamp = n2.timestamp
WHERE
     n1.symbol = 'TCS'
     AND n2.symbol = 'GEOMETRIC';
```

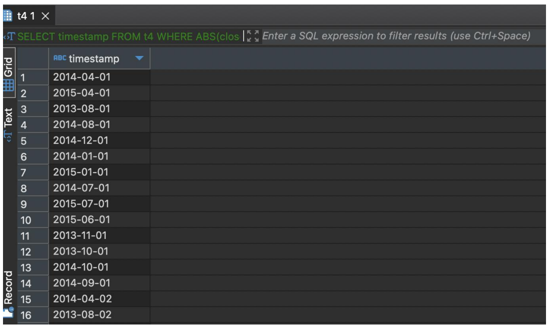


SELECT MAX(n1.close - n2.close) AS max_diff

	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 differences of their closing prices.

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



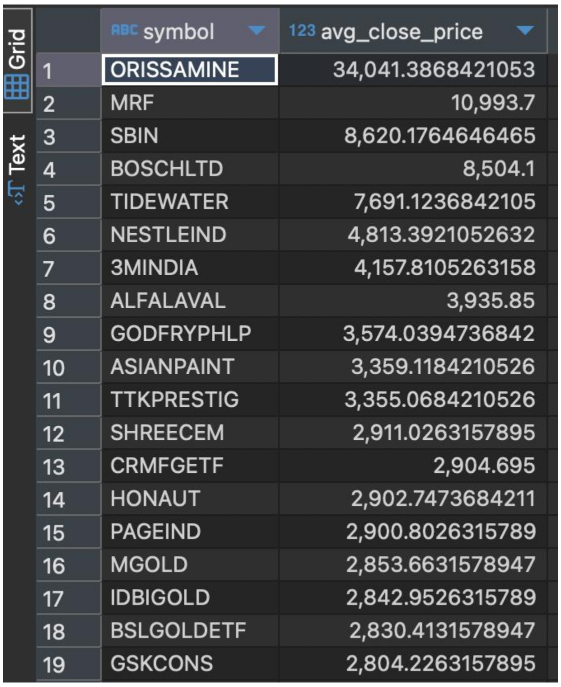
SELECT timestamp FROM t4 WHERE ABS(clos

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

	timestamp
1	2014-04-01
2	2015-04-01
3	2013-08-01
4	2014-08-01
5	2014-12-01
6	2014-01-01
7	2015-01-01
8	2014-07-01
9	2015-07-01
10	2015-06-01
11	2013-11-01
12	2013-10-01
13	2014-10-01
14	2014-09-01
15	2014-04-02
16	2013-08-02

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.

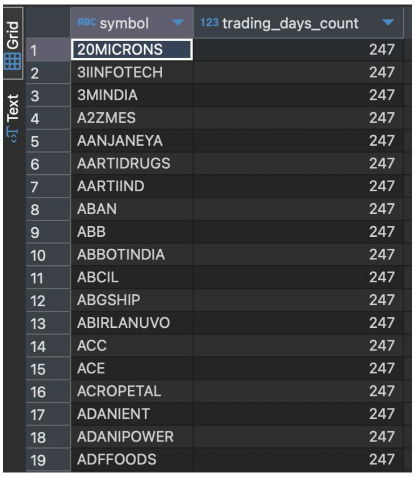
```
CREATE TABLE t5 AS
SELECT
     symbol,
     AVG(close) AS avg_close_price
FROM
     nsedata
WHERE
     YEAR(timestamp) = 2012
     AND MONTH(timestamp) = 4
GROUP BY
     symbol
ORDER BY
     avg_close_price DESC;
```



	symbol	avg_close_price
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
12	SHREECEM	2,911.0263157895
13	CRMFGETF	2,904.695
14	HONAUT	2,902.7473684211
15	PAGEIND	2,900.8026315789
16	MGOLD	2,853.6631578947
17	IDBIGOLD	2,842.9526315789
18	BSLGOLDETF	2,830.4131578947
19	GSKCONS	2,804.2263157895

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 in a selected year – say 2012. The table should be sorted in descending order of the count.

```
CREATE TABLE t6 AS
SELECT
     symbol,
     COUNT(DISTINCT DATE(timestamp)) AS
     trading_days_count
FROM
     nsedata
WHERE
     YEAR(timestamp) = 2012
GROUP BY
     symbol
ORDER BY
     trading_days_count DESC;
```



	symbol	trading_days_count
1	20MICRONS	247
2	3IINFOTECH	247
3	3MINDIA	247
4	A2ZMES	247
5	AANJANEYA	247
6	AARTIDRUGS	247
7	AARTIIND	247
8	ABAN	247
9	ABB	247
10	ABBOTINDIA	247
11	ABCIL	247
12	ABGSHIP	247
13	ABIRLANUVO	247
14	ACC	247
15	ACE	247
16	ACROPETAL	247
17	ADANIENT	247
18	ADANIPOWER	247
19	ADFFOODS	247