E09-ASSIGNMENT 9

NAME: HRISHIKESH S

ROLL NO: 22B4217

Q1)

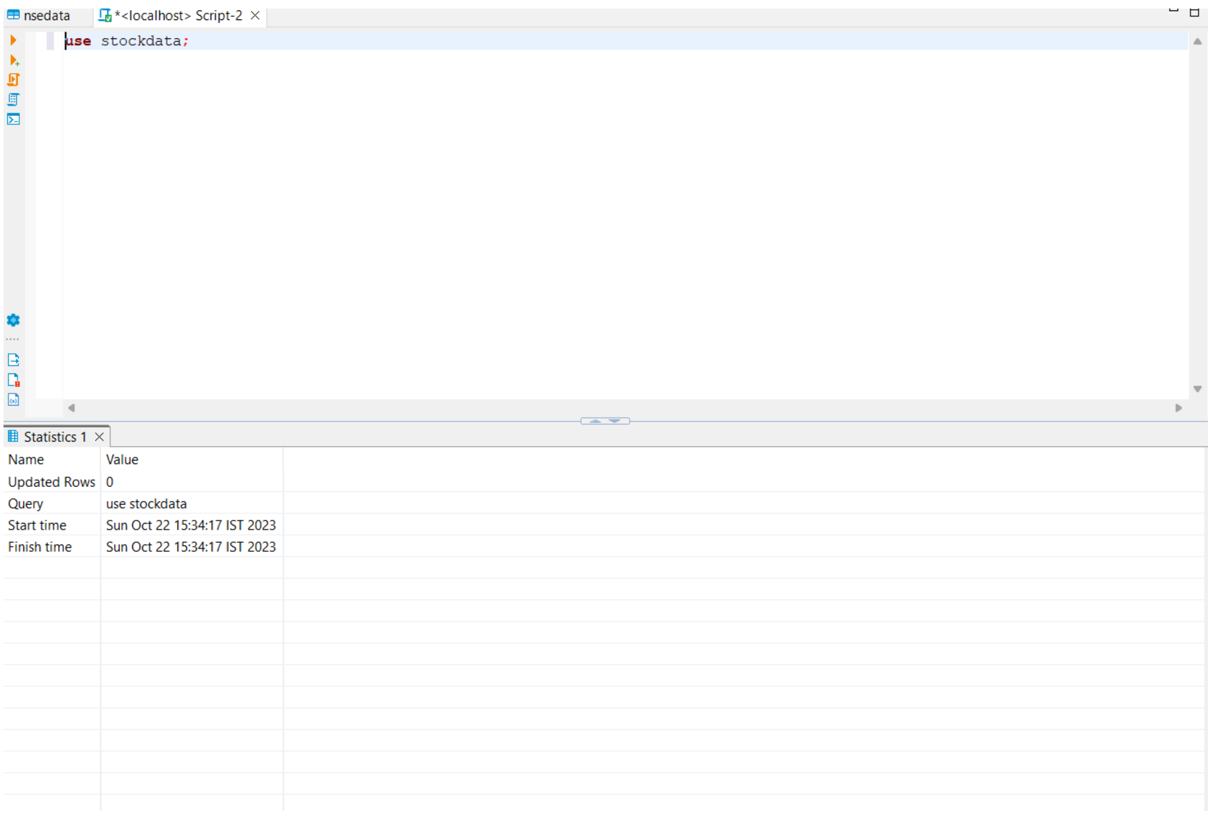
Do a manual review of the table nsedata and describe its contents (no SQL to be executed
for this task):

The given CSV file contains a huge amount of data organised into 13 columns and 1893059 rows!

The given data consists of the Stock price details like the opening Stock price,closing stock price,highest stock price,lowest stock price,etc which are all self explanatory.

Q2)

Select the database stockdata using SQL:

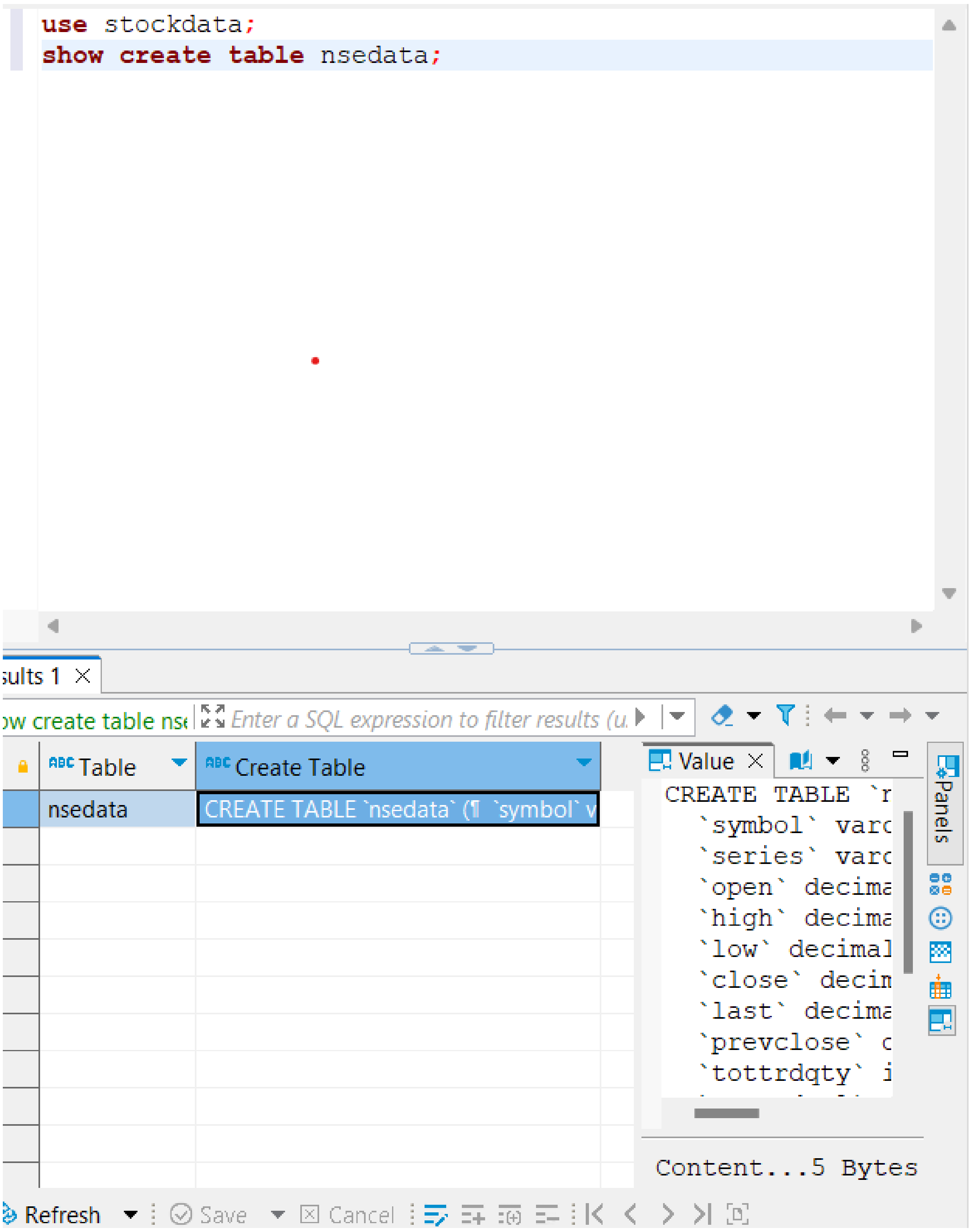


```
use stockdata;
```

Name	Value
Updated Rows	0
Query	use stockdata
Start time	Sun Oct 22 15:34:17 IST 2023
Finish time	Sun Oct 22 15:34:17 IST 2023

Q3)

Get a schema dump of the table nsedata:

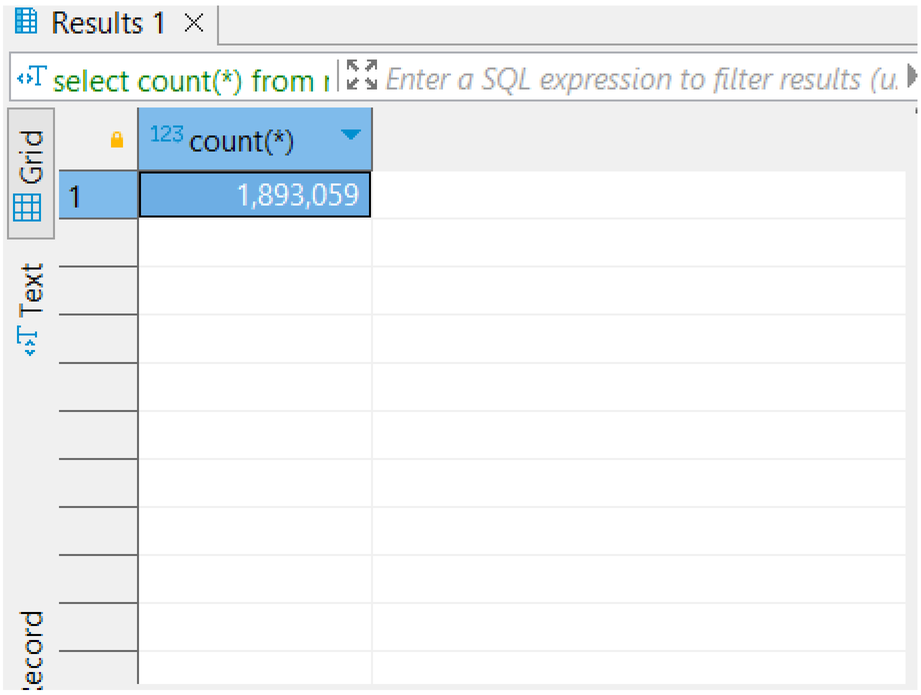


```
use stockdata;
show create table nsedata;
```

```
CREATE TABLE `r
  `symbol` varc
  `series` varc
  `open` decima
  `high` decima
  `low` decimal
  `close` decim
  `last` decima
  `prevclose` c
  `tottrdqty` i
```

Q4)

1893059



select count(*) from r

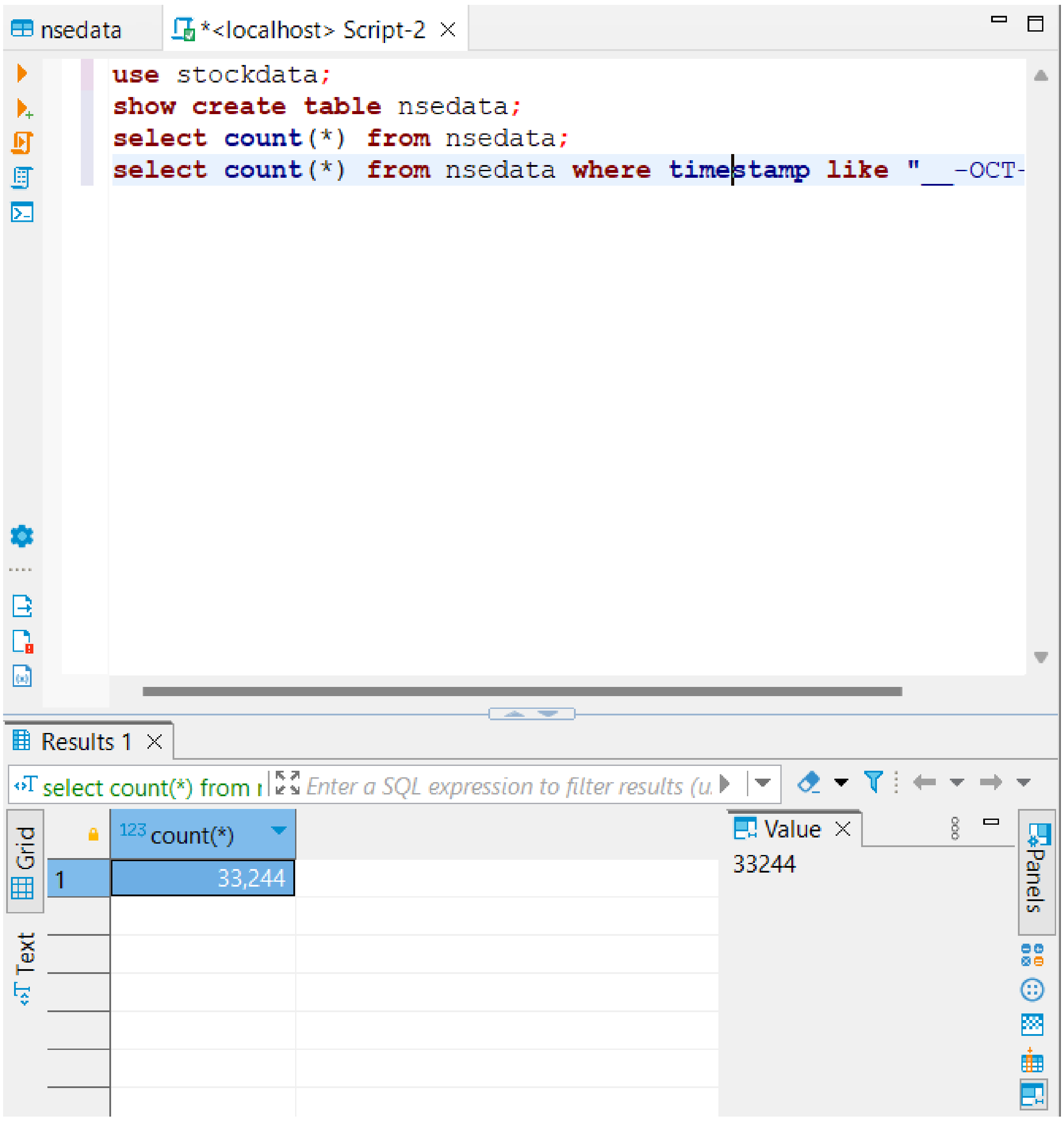
Enter a SQL expression to filter results (u

	count(*)
1	1,893,059

Q5)

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

select count(*) **from** nsedata **where timestamp like** "__-OCT-2012";

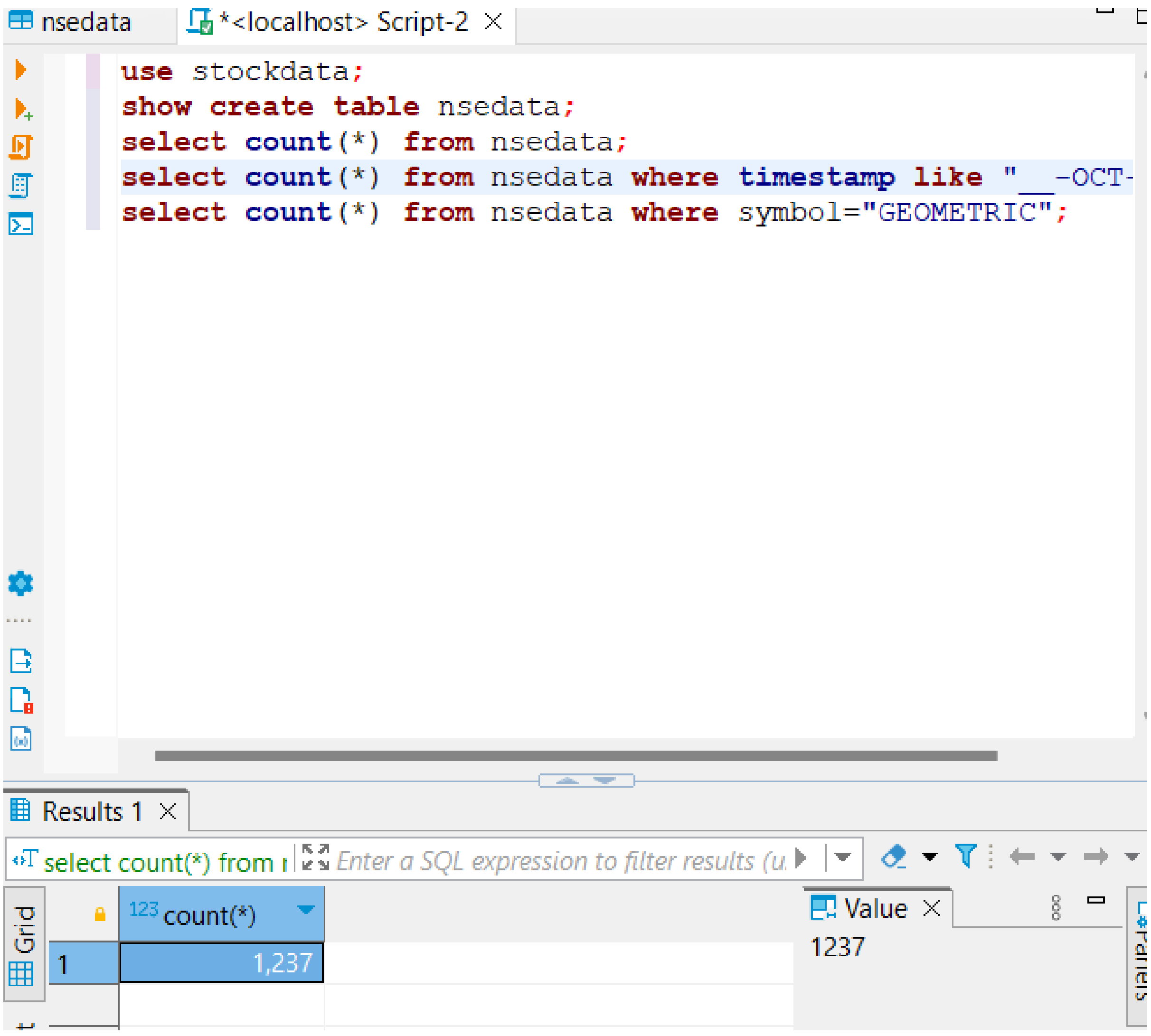


```
use stockdata;
show create table nsedata;
select count(*) from nsedata;
select count(*) from nsedata where timestamp like "__-OCT-
```

```
select count(*) from n
Enter a SQL expression to filter results (u
   123 count(*)
1          33,244
Value
33244
```

Q6)

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



nsedata *<localhost> Script-2

```
use stockdata;
show create table nsedata;
select count(*) from nsedata;
select count(*) from nsedata where timestamp like "__-OCT-
select count(*) from nsedata where symbol="GEOMETRIC";
```

Results 1

select count(*) from r Enter a SQL expression to filter results (u

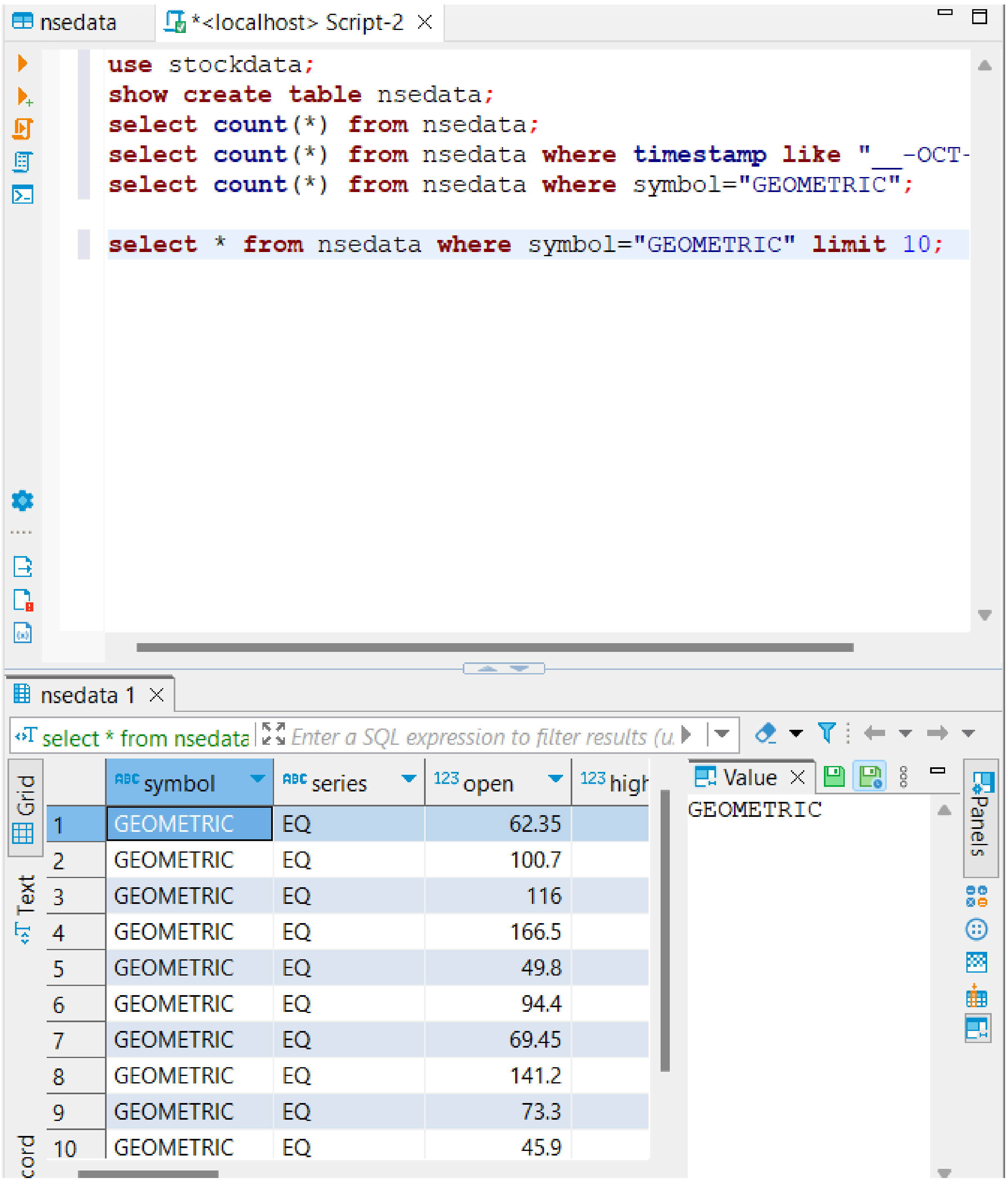
```
  count(*)
1    1,237
```

Value

1237

Q7)

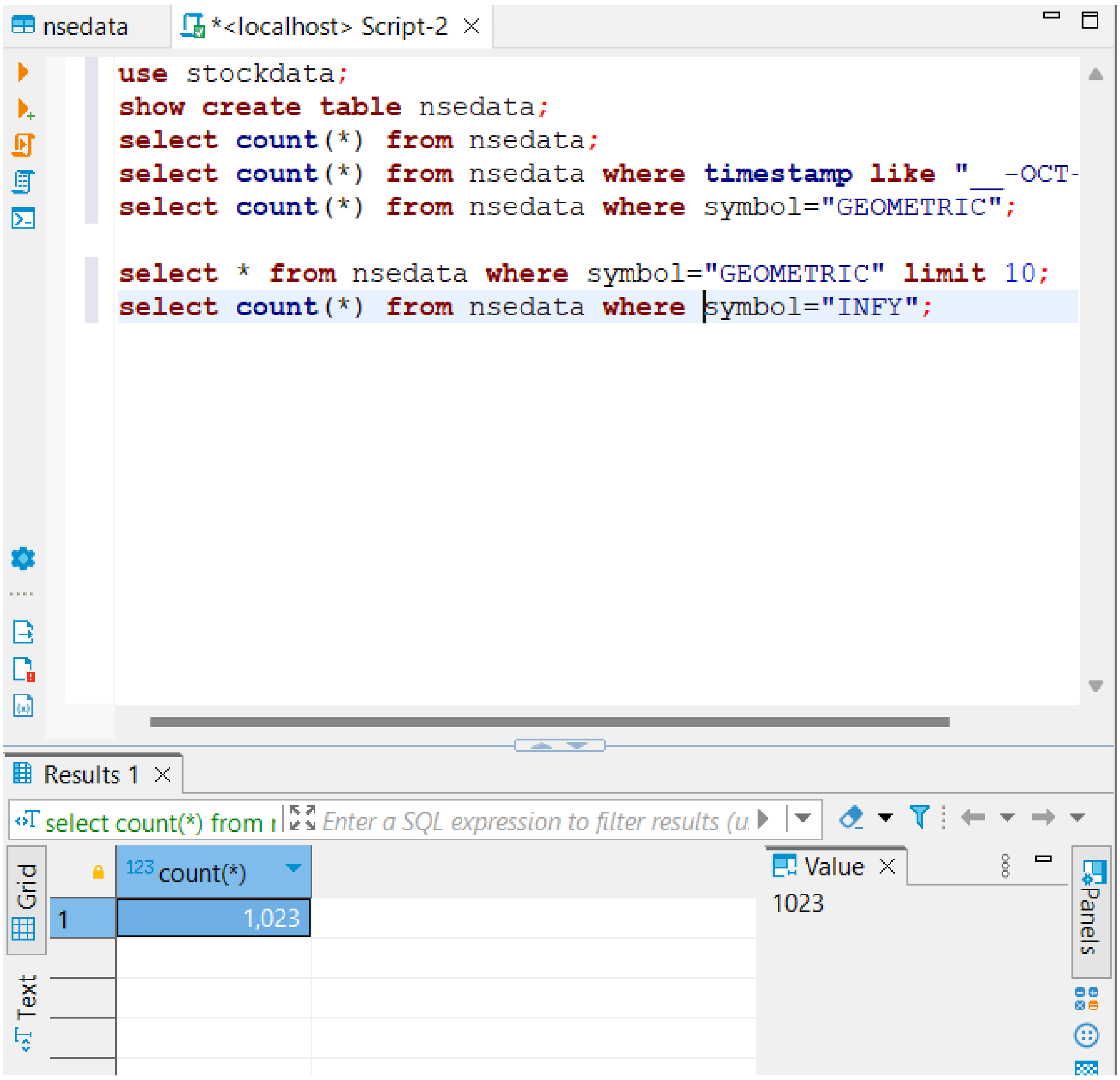
Repeat ‘6’, but only display the first 10 records:



```
use stockdata;
show create table nsedata;
select count(*) from nsedata;
select count(*) from nsedata where timestamp like "__-OCT-
select count(*) from nsedata where symbol="GEOMETRIC";

select * from nsedata where symbol="GEOMETRIC" limit 10;
```

	symbol	series	open	high
1	GEOMETRIC	EQ	62.35	
2	GEOMETRIC	EQ	100.7	
3	GEOMETRIC	EQ	116	
4	GEOMETRIC	EQ	166.5	
5	GEOMETRIC	EQ	49.8	
6	GEOMETRIC	EQ	94.4	
7	GEOMETRIC	EQ	69.45	
8	GEOMETRIC	EQ	141.2	
9	GEOMETRIC	EQ	73.3	
10	GEOMETRIC	EQ	45.9	

Q8)Totally, how many records of “INFY” does the table contain?:



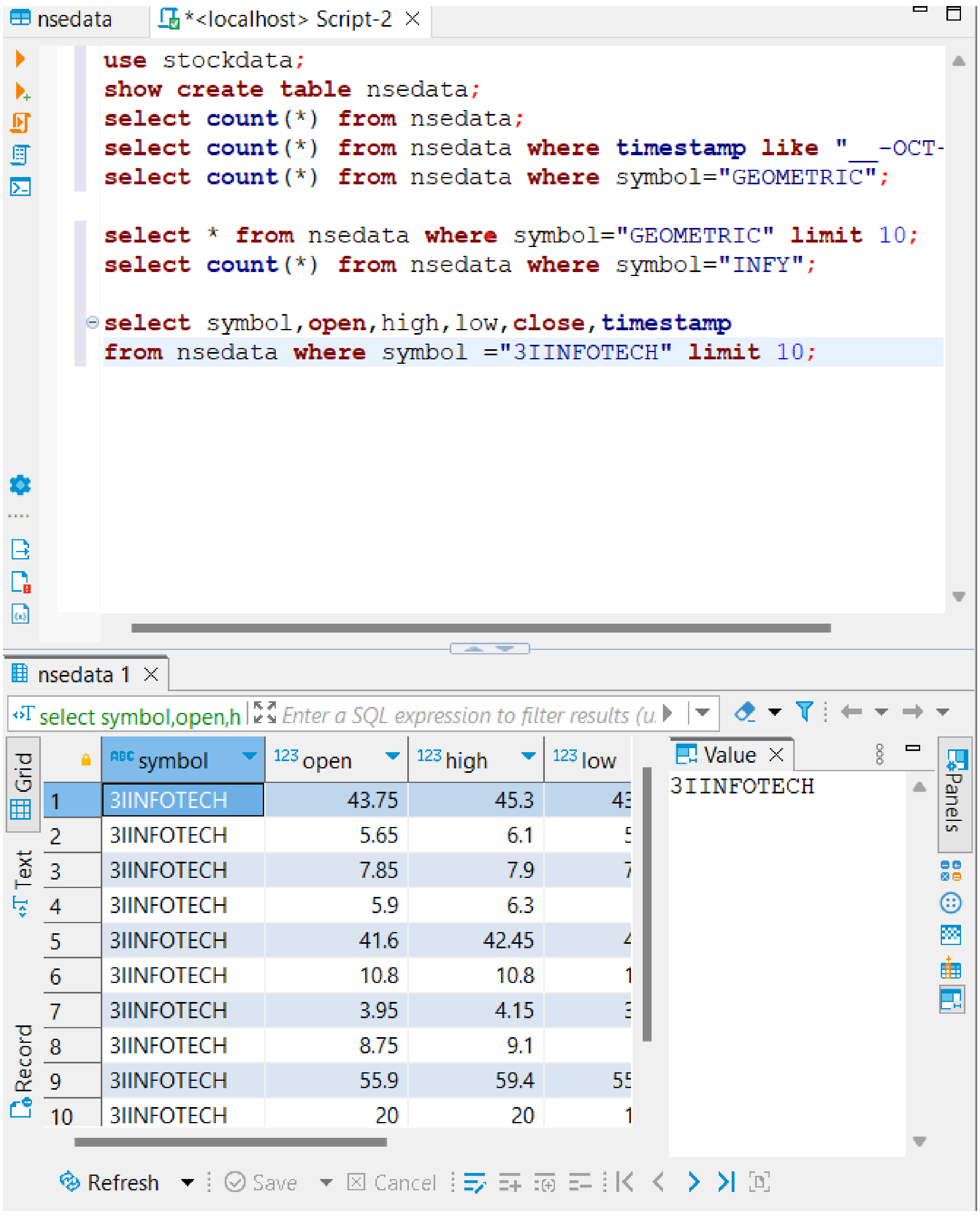
```
use stockdata;
show create table nsedata;
select count(*) from nsedata;
select count(*) from nsedata where timestamp like "__-OCT-
select count(*) from nsedata where symbol="GEOMETRIC";

select * from nsedata where symbol="GEOMETRIC" limit 10;
select count(*) from nsedata where symbol="INFY";
```

```
   count(*)
1     1,023
```

```
Value
1023
```

Q9)

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:



nsedata *<localhost> Script-2

```
use stockdata;
show create table nsedata;
select count(*) from nsedata;
select count(*) from nsedata where timestamp like "__-OCT-
select count(*) from nsedata where symbol="GEOMETRIC";

select * from nsedata where symbol="GEOMETRIC" limit 10;
select count(*) from nsedata where symbol="INFY";

select symbol,open,high,low,close,timestamp
from nsedata where symbol ="3IINFOTECH" limit 10;
```

nsedata 1

select symbol,open,h | Enter a SQL expression to filter results (u.

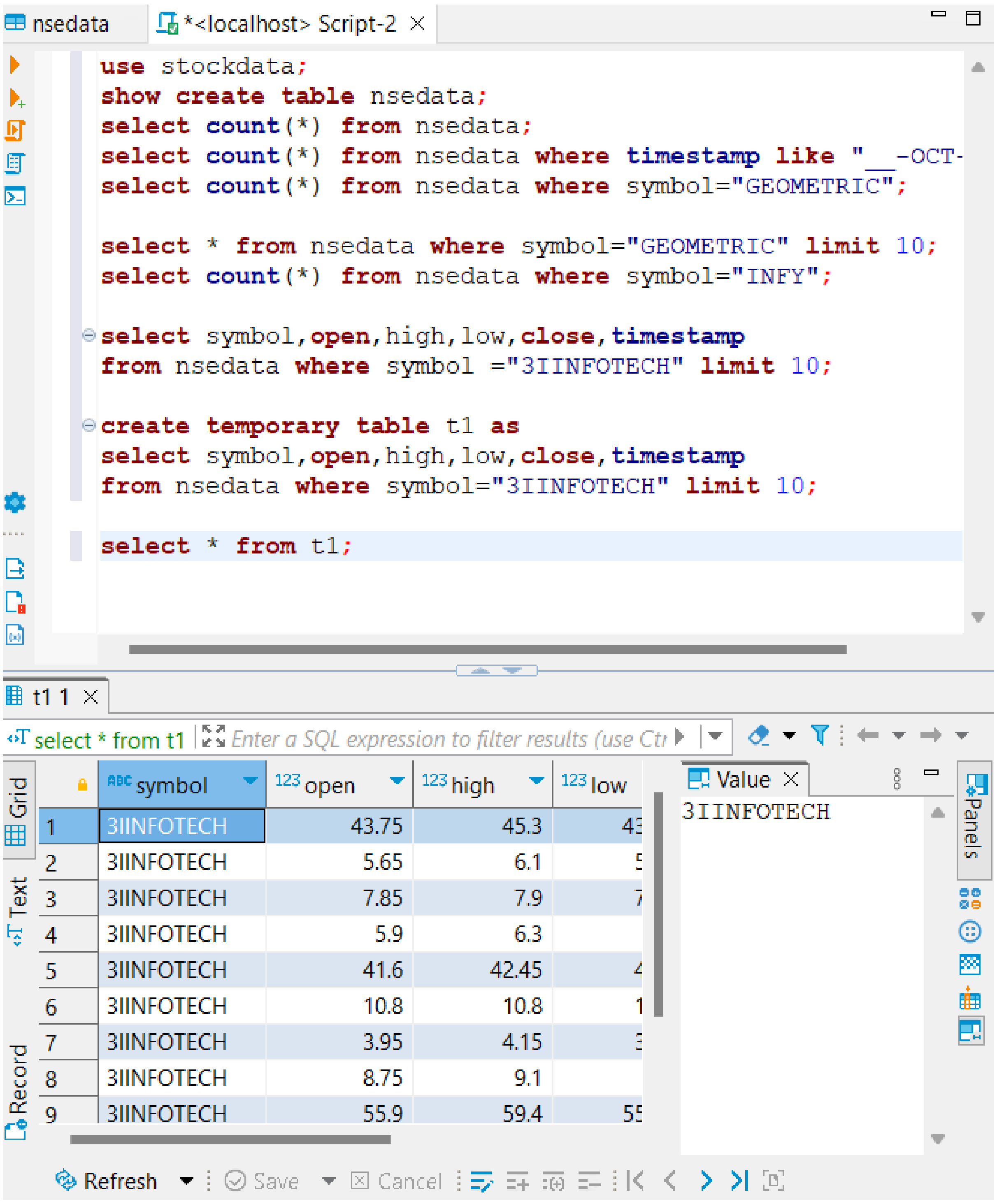
	symbol	open	high	low
1	3IINFOTECH	43.75	45.3	43
2	3IINFOTECH	5.65	6.1	5
3	3IINFOTECH	7.85	7.9	7
4	3IINFOTECH	5.9	6.3	
5	3IINFOTECH	41.6	42.45	4
6	3IINFOTECH	10.8	10.8	1
7	3IINFOTECH	3.95	4.15	3
8	3IINFOTECH	8.75	9.1	
9	3IINFOTECH	55.9	59.4	55
10	3IINFOTECH	20	20	1

Value

```
3IINFOTECH
```

Q10)

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



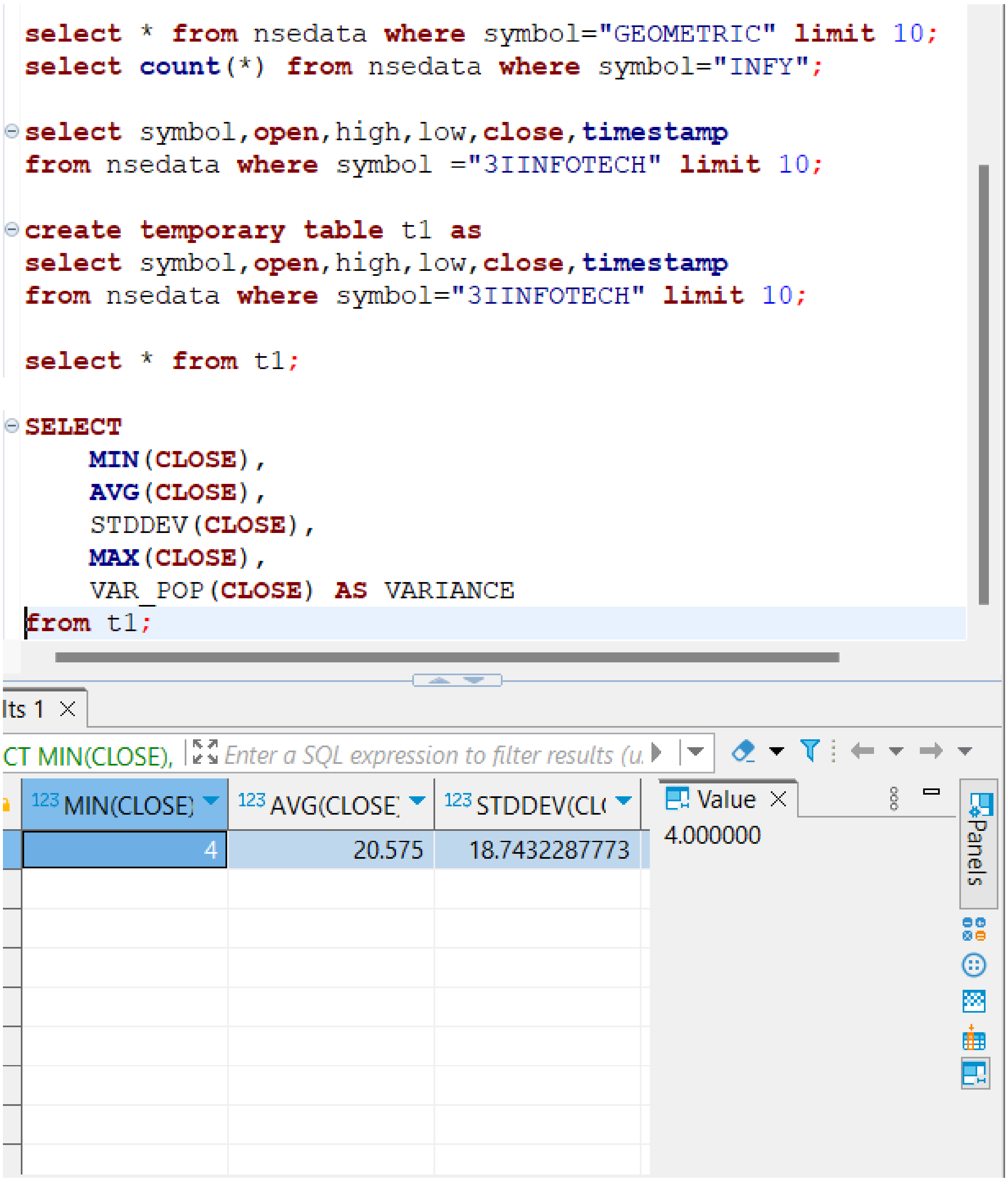
```
use stockdata;
show create table nsedata;
select count(*) from nsedata;
select count(*) from nsedata where timestamp like "__-OCT-
select count(*) from nsedata where symbol="GEOMETRIC";

select * from nsedata where symbol="GEOMETRIC" limit 10;
select count(*) from nsedata where symbol="INFY";

select symbol,open,high,low,close,timestamp
from nsedata where symbol ="3IINFOTECH" limit 10;

create temporary table t1 as
select symbol,open,high,low,close,timestamp
from nsedata where symbol="3IINFOTECH" limit 10;

select * from t1;
```

	symbol	open	high	low
1	3IINFOTECH	43.75	45.3	43
2	3IINFOTECH	5.65	6.1	5
3	3IINFOTECH	7.85	7.9	7
4	3IINFOTECH	5.9	6.3	
5	3IINFOTECH	41.6	42.45	4
6	3IINFOTECH	10.8	10.8	1
7	3IINFOTECH	3.95	4.15	3
8	3IINFOTECH	8.75	9.1	
9	3IINFOTECH	55.9	59.4	55

```
3IINFOTECH
```

Q11)

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



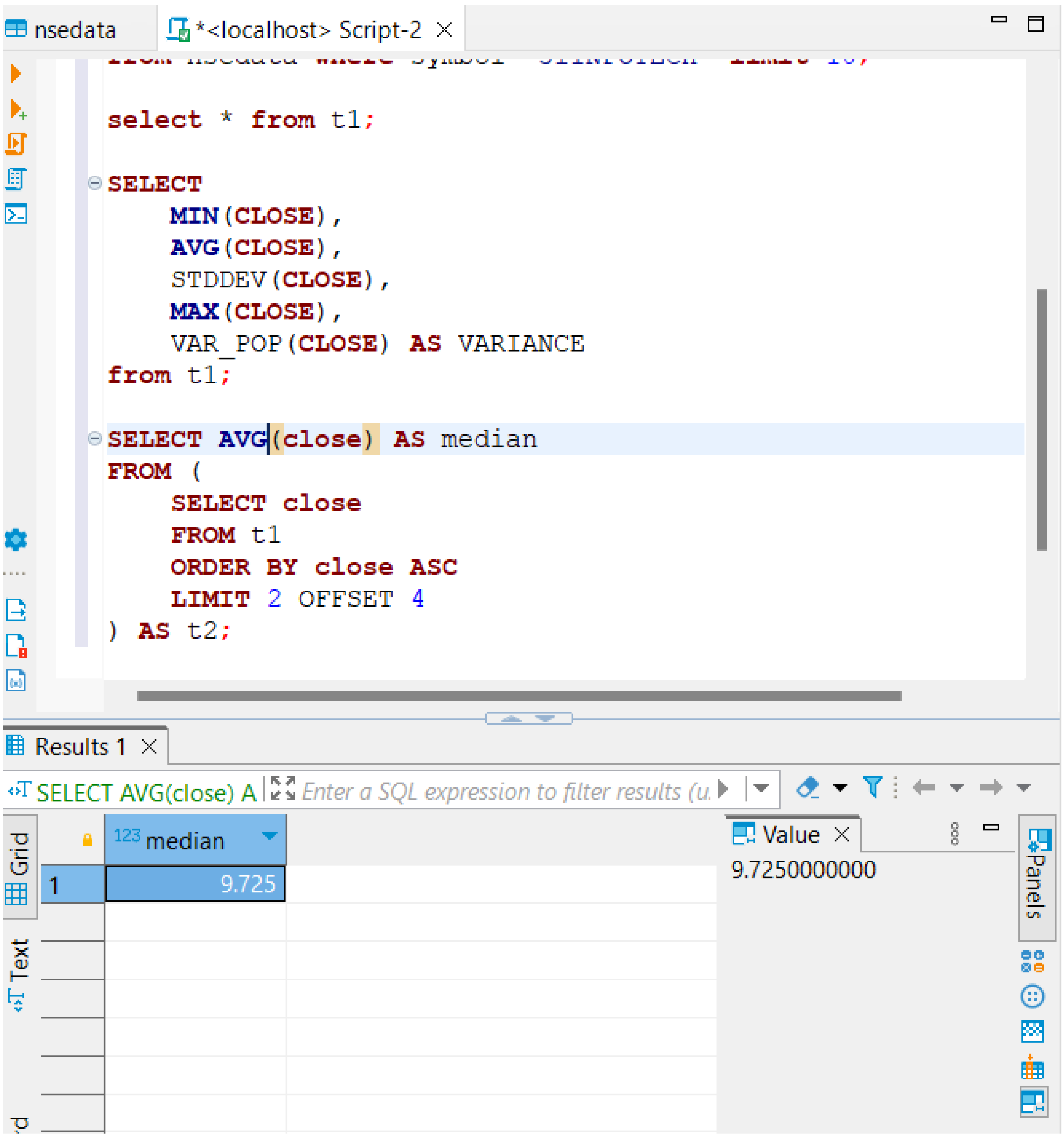
```
select * from nsedata where symbol="GEOMETRIC" limit 10;
select count(*) from nsedata where symbol="INFY";

select symbol,open,high,low,close,timestamp
from nsedata where symbol ="3IINFOTECH" limit 10;

create temporary table t1 as
select symbol,open,high,low,close,timestamp
from nsedata where symbol="3IINFOTECH" limit 10;

select * from t1;

SELECT
      MIN(CLOSE),
      AVG(CLOSE),
      STDDEV(CLOSE),
      MAX(CLOSE),
      VAR_POP(CLOSE) AS VARIANCE
from t1;
```

MIN(CLOSE)	AVG(CLOSE	STDDEV(CL
4	20.575	18.7432287773

Value: 4.000000

Q12)

How will you find out the value of the median?:

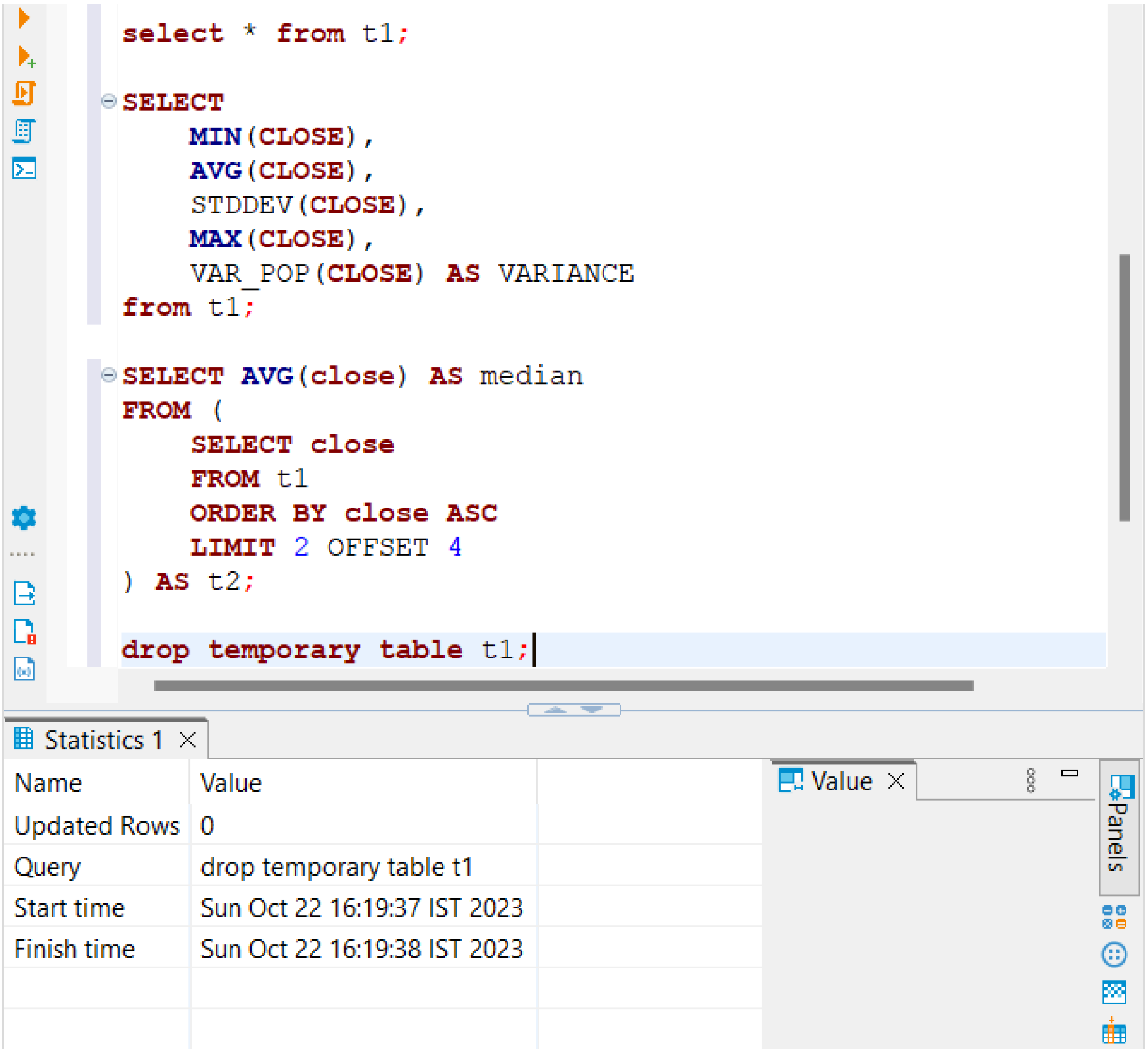


```
select * from t1;

SELECT
     MIN(CLOSE),
     AVG(CLOSE),
     STDDEV(CLOSE),
     MAX(CLOSE),
     VAR_POP(CLOSE) AS VARIANCE
from t1;

SELECT AVG(close) AS median
FROM (
     SELECT close
     FROM t1
     ORDER BY close ASC
     LIMIT 2 OFFSET 4
) AS t2;
```

```
   median
1  9.725

Value
9.7250000000
```

Q13)

Delete table t1:



```
select * from t1;

SELECT
     MIN(CLOSE),
     AVG(CLOSE),
     STDDEV(CLOSE),
     MAX(CLOSE),
     VAR_POP(CLOSE) AS VARIANCE
from t1;

SELECT AVG(close) AS median
FROM (
     SELECT close
     FROM t1
     ORDER BY close ASC
     LIMIT 2 OFFSET 4
) AS t2;

drop temporary table t1;
```

Statistics 1

Name	Value
Updated Rows	0
Query	drop temporary table t1
Start time	Sun Oct 22 16:19:37 IST 2023
Finish time	Sun Oct 22 16:19:38 IST 2023

Q14)

**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:**

SELECT AVG(close) AS median

FROM (

SELECT close

FROM t1

ORDER BY close ASC

LIMIT 2 OFFSET 4

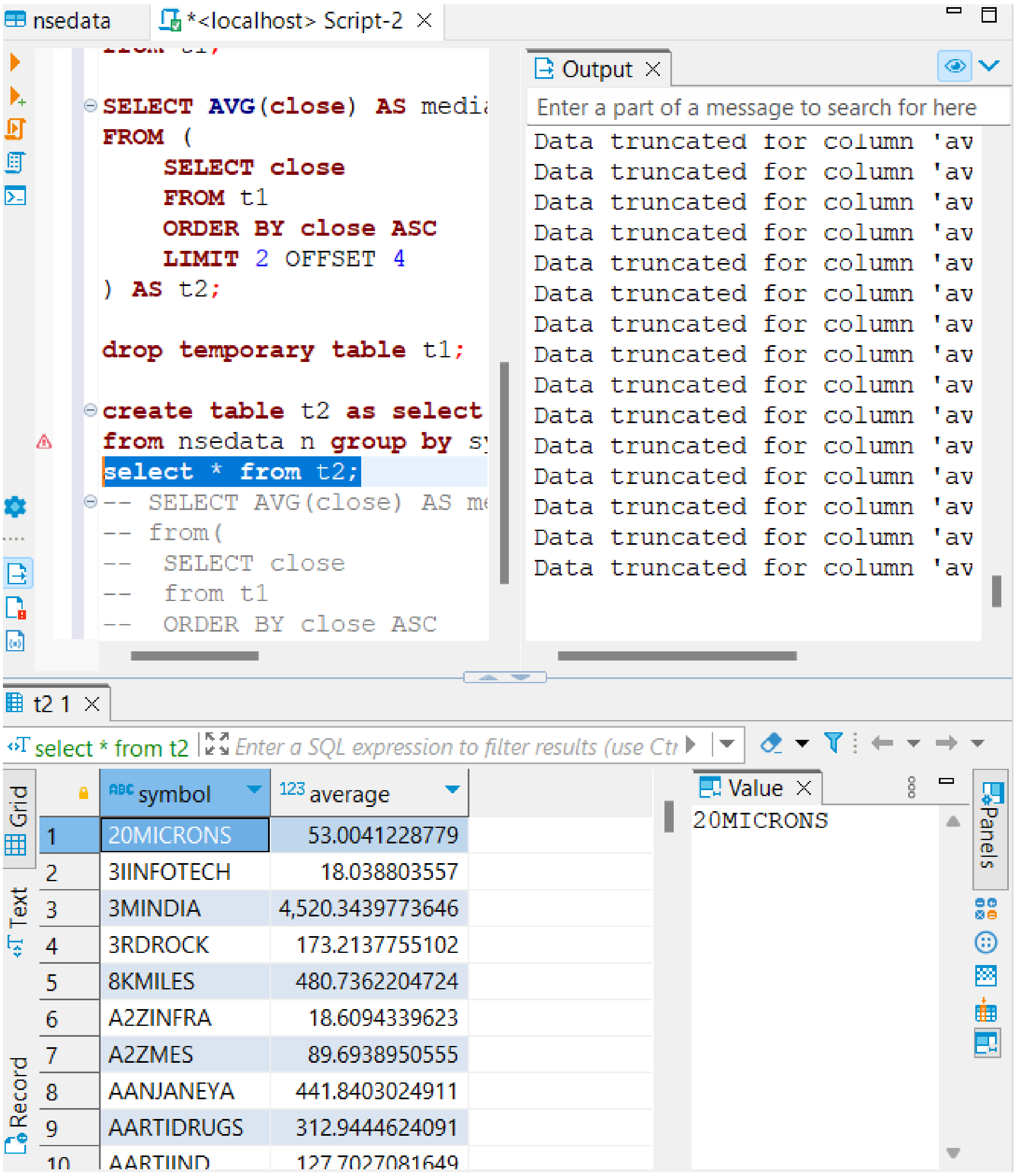
) **AS** t2;

drop temporary table t1;

create table t2 **as select** symbol,**avg**(**close**) **as** average

from nsedata n **group by** symbol;

select * **from** t2;



```
SELECT AVG(close) AS medi
FROM (
     SELECT close
     FROM t1
     ORDER BY close ASC
     LIMIT 2 OFFSET 4
) AS t2;

drop temporary table t1;

create table t2 as select
from nsedata n group by sy
select * from t2;
-- SELECT AVG(close) AS me
-- from(
--   SELECT close
--   from t1
--   ORDER BY close ASC
```

```
Data truncated for column 'av
Data truncated for column 'av
Data truncated for column 'av
Data truncated for column 'av
Data truncated for column 'av
Data truncated for column 'av
Data truncated for column 'av
Data truncated for column 'av
Data truncated for column 'av
Data truncated for column 'av
Data truncated for column 'av
Data truncated for column 'av
Data truncated for column 'av
Data truncated for column 'av
Data truncated for column 'av
```

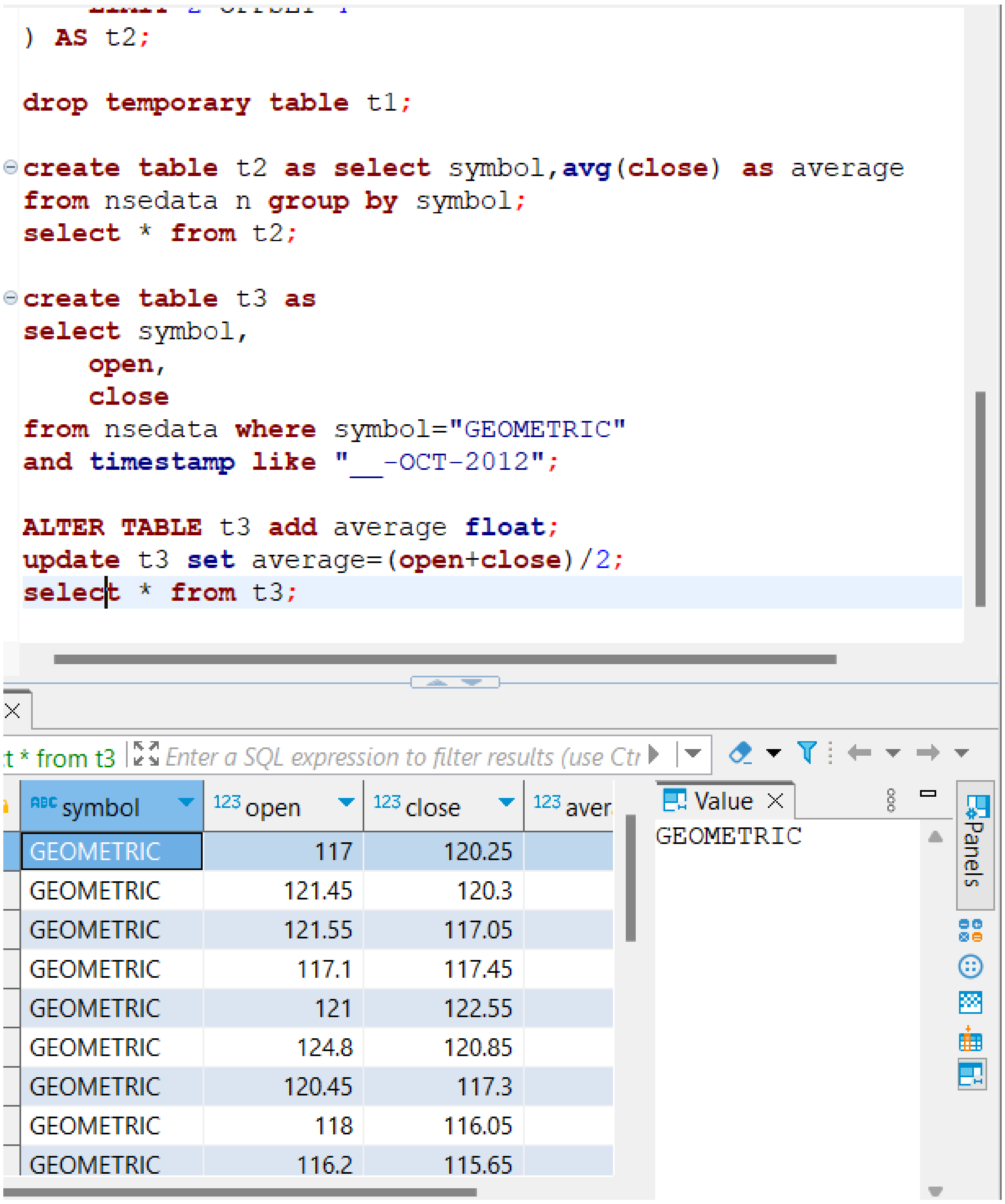
select * from t2

	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

Value: 20MICRONS

Q15)

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:

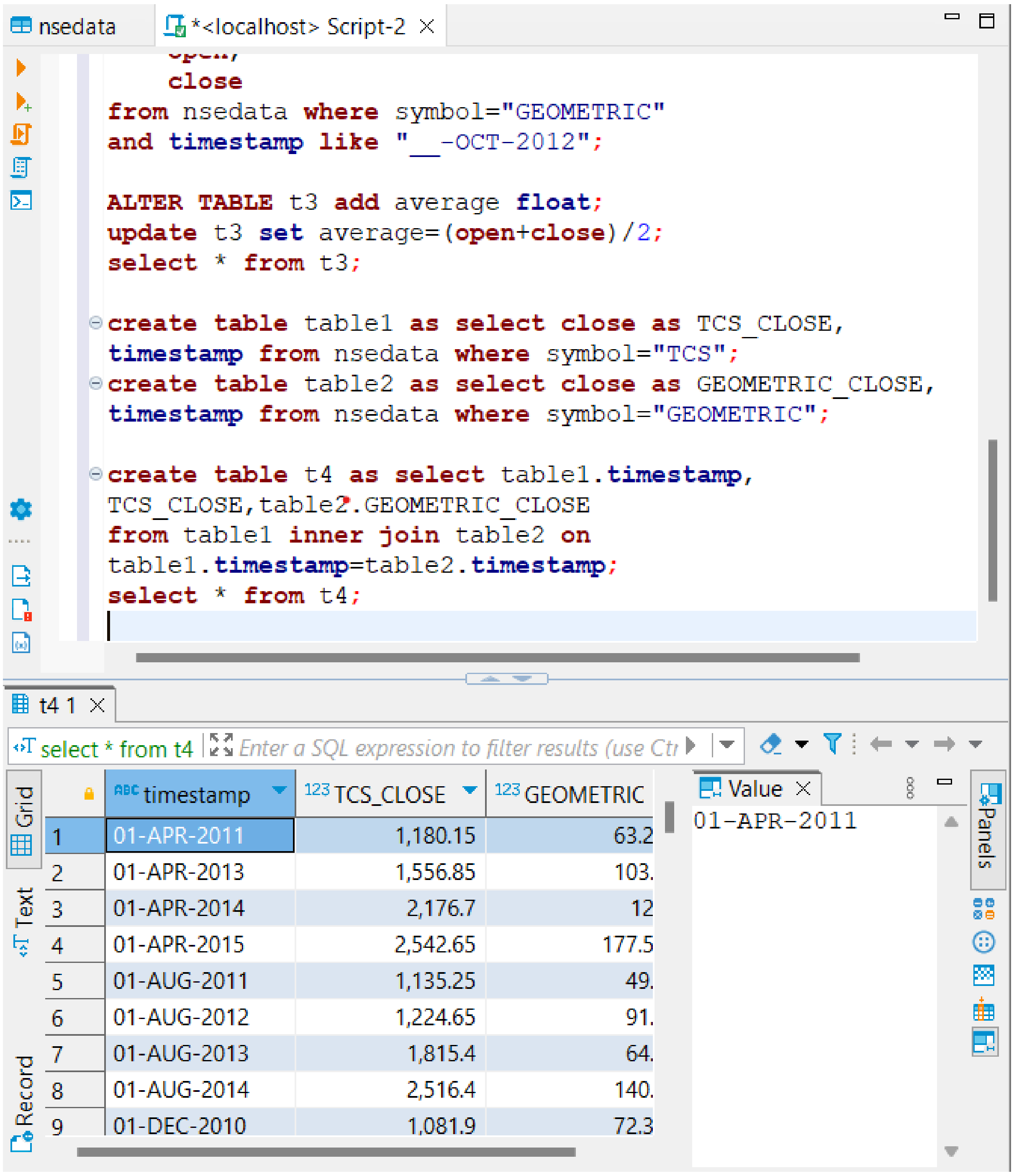


```
) AS t2;

drop temporary table t1;

create table t2 as select symbol,avg(close) as average
from nsedata n group by symbol;
select * from t2;

create table t3 as
select symbol,
    open,
    close
from nsedata where symbol="GEOMETRIC"
and timestamp like "__-OCT-2012";

ALTER TABLE t3 add average float;
update t3 set average=(open+close)/2;
select * from t3;
```

symbol	open	close	aver
GEOMETRIC	117	120.25	
GEOMETRIC	121.45	120.3	
GEOMETRIC	121.55	117.05	
GEOMETRIC	117.1	117.45	
GEOMETRIC	121	122.55	
GEOMETRIC	124.8	120.85	
GEOMETRIC	120.45	117.3	
GEOMETRIC	118	116.05	
GEOMETRIC	116.2	115.65	

Q16)

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:



```
    close
from nsedata where symbol="GEOMETRIC"
and timestamp like "__-OCT-2012";

ALTER TABLE t3 add average float;
update t3 set average=(open+close)/2;
select * from t3;

create table table1 as select close as TCS_CLOSE,
timestamp from nsedata where symbol="TCS";
create table table2 as select close as GEOMETRIC_CLOSE,
timestamp from nsedata where symbol="GEOMETRIC";

create table t4 as select table1.timestamp,
TCS_CLOSE,table2.GEOMETRIC_CLOSE
from table1 inner join table2 on
table1.timestamp=table2.timestamp;
select * from t4;
```

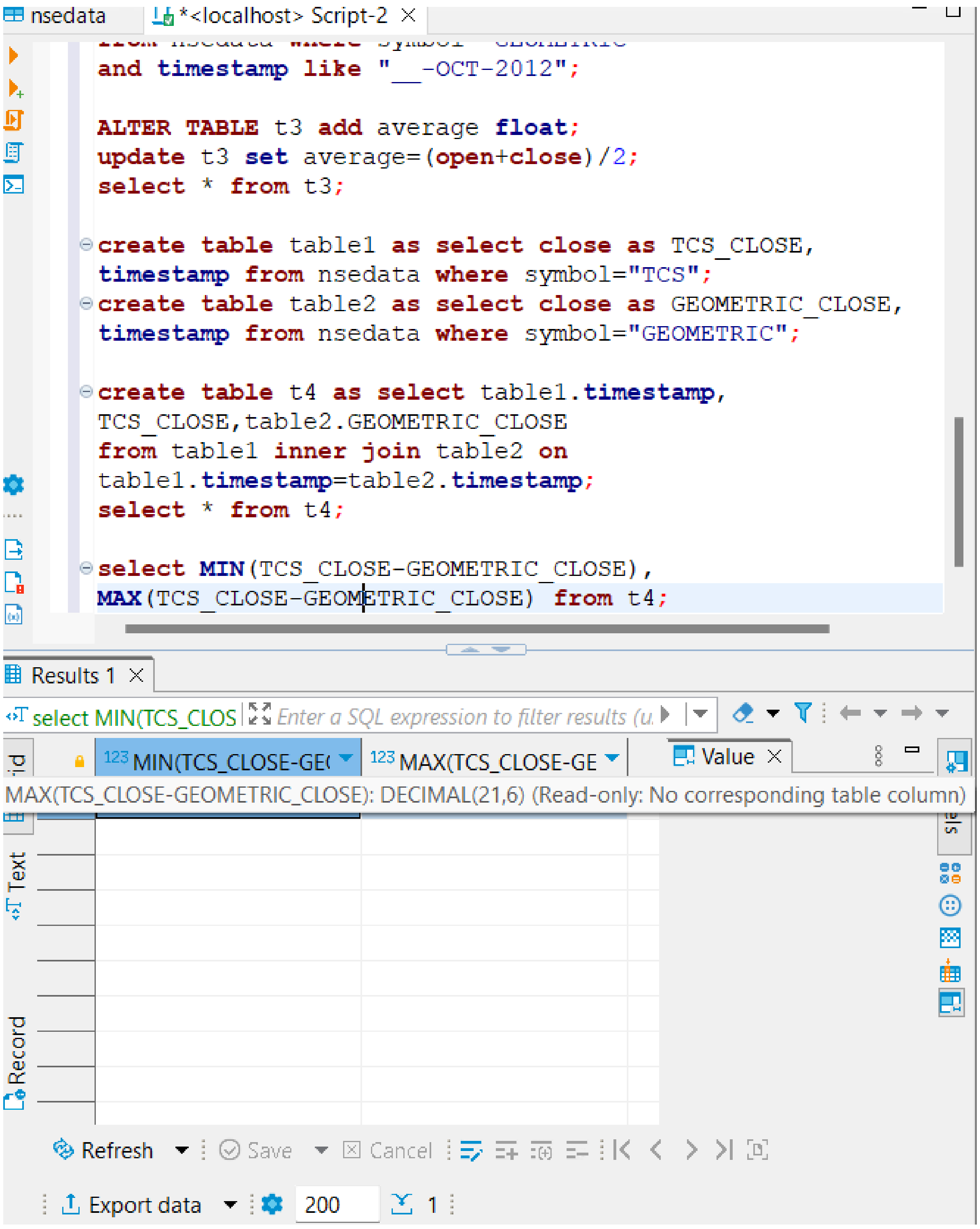
select * from t4

	timestamp	TCS_CLOSE	GEOMETRIC
1	01-APR-2011	1,180.15	63.2
2	01-APR-2013	1,556.85	103.
3	01-APR-2014	2,176.7	12
4	01-APR-2015	2,542.65	177.5
5	01-AUG-2011	1,135.25	49.
6	01-AUG-2012	1,224.65	91.
7	01-AUG-2013	1,815.4	64.
8	01-AUG-2014	2,516.4	140.
9	01-DEC-2010	1,081.9	72.3

Value: 01-APR-2011

Q17)

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



nsedata   *<localhost> Script-2

```
and timestamp like "__-OCT-2012";

ALTER TABLE t3 add average float;
update t3 set average=(open+close)/2;
select * from t3;

create table table1 as select close as TCS_CLOSE,
timestamp from nsedata where symbol="TCS";
create table table2 as select close as GEOMETRIC_CLOSE,
timestamp from nsedata where symbol="GEOMETRIC";

create table t4 as select table1.timestamp,
TCS_CLOSE,table2.GEOMETRIC_CLOSE
from table1 inner join table2 on
table1.timestamp=table2.timestamp;
select * from t4;

select MIN(TCS_CLOSE-GEOMETRIC_CLOSE),
MAX(TCS_CLOSE-GEOMETRIC_CLOSE) from t4;
```

Results 1

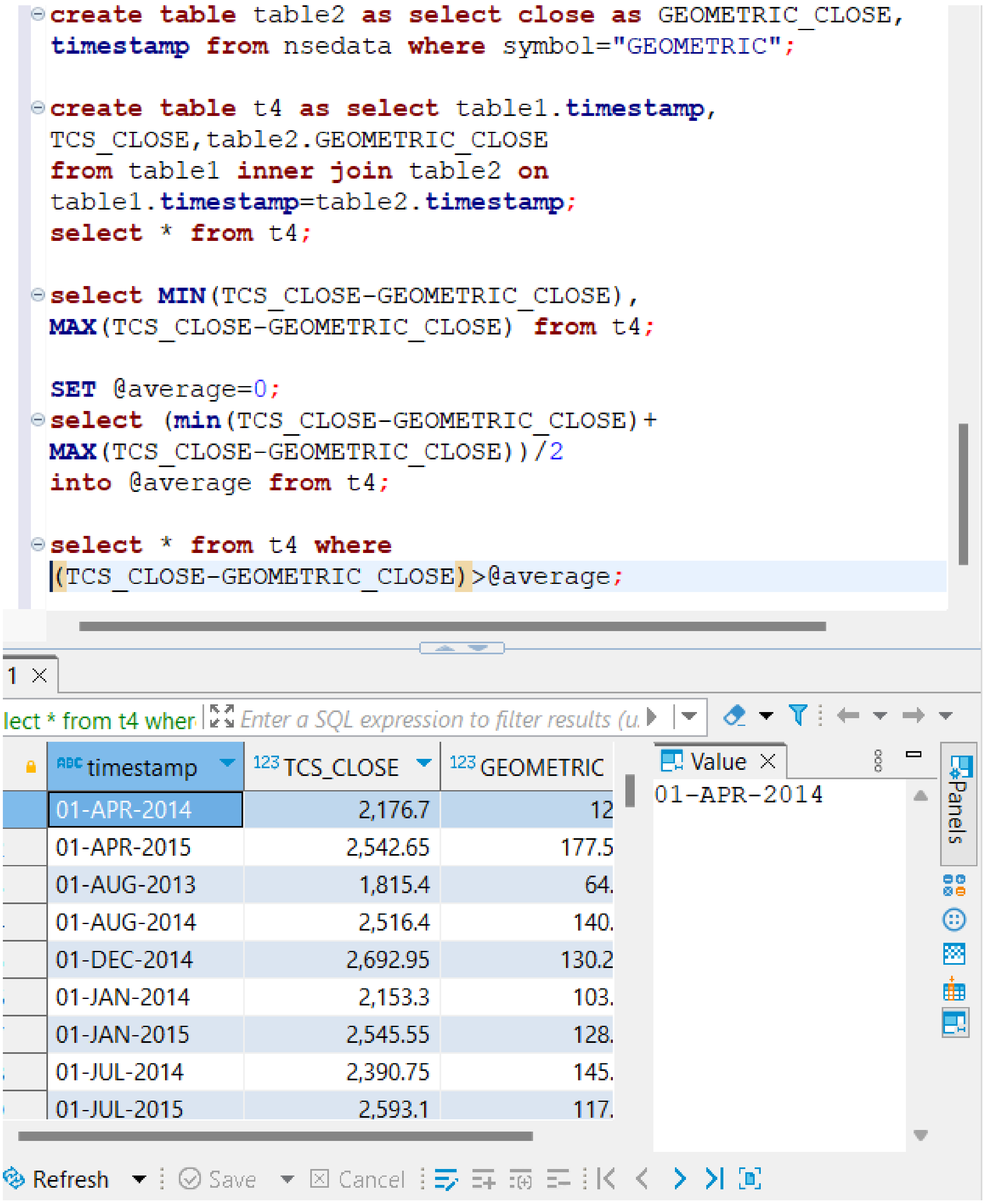
select MIN(TCS_CLOS   Enter a SQL expression to filter results

MIN(TCS_CLOSE-GE   MAX(TCS_CLOSE-GE   Value

MAX(TCS_CLOSE-GEOMETRIC_CLOSE): DECIMAL(21,6) (Read-only: No corresponding table column)

Q18)

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.:



```
create table table2 as select close as GEOMETRIC_CLOSE,
timestamp from nsedata where symbol="GEOMETRIC";

create table t4 as select table1.timestamp,
TCS_CLOSE,table2.GEOMETRIC_CLOSE
from table1 inner join table2 on
table1.timestamp=table2.timestamp;
select * from t4;

select MIN(TCS_CLOSE-GEOMETRIC_CLOSE),
MAX(TCS_CLOSE-GEOMETRIC_CLOSE) from t4;

SET @average=0;
select (min(TCS_CLOSE-GEOMETRIC_CLOSE)+
MAX(TCS_CLOSE-GEOMETRIC_CLOSE))/2
into @average from t4;

select * from t4 where
(TCS_CLOSE-GEOMETRIC_CLOSE)>@average;
```

lect * from t4 wher

timestamp	TCS_CLOSE	GEOMETRIC
01-APR-2014	2,176.7	12
01-APR-2015	2,542.65	177.5
01-AUG-2013	1,815.4	64.
01-AUG-2014	2,516.4	140.
01-DEC-2014	2,692.95	130.2
01-JAN-2014	2,153.3	103.
01-JAN-2015	2,545.55	128.
01-JUL-2014	2,390.75	145.
01-JUL-2015	2,593.1	117.

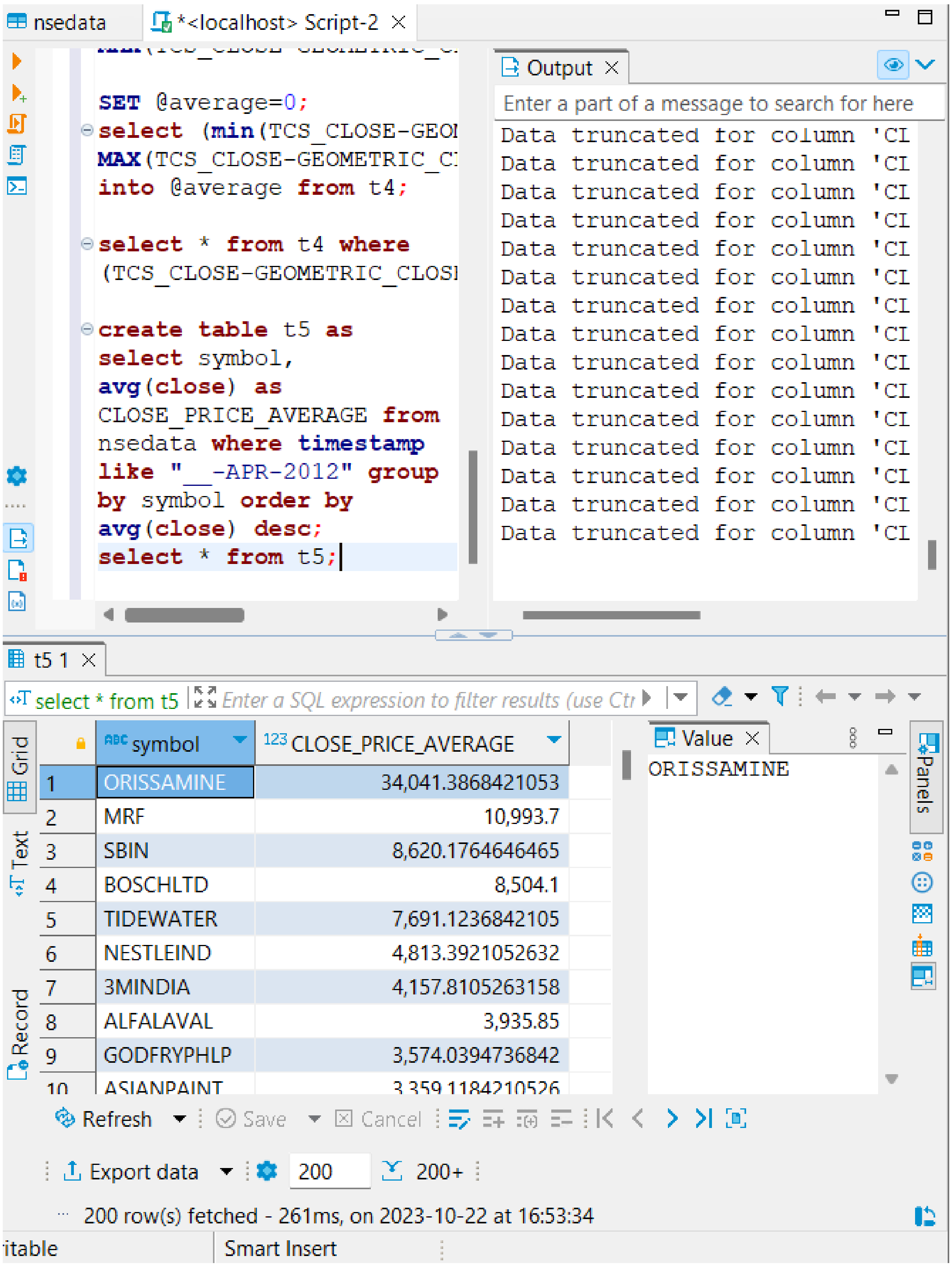
Value

01-APR-2014

Q19)

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.:

```
SET @average=0;

select (min(TCS_CLOSE-GEOMETRIC_CLOSE)+

MAX(TCS_CLOSE-GEOMETRIC_CLOSE))/2

into @average from t4;

select * from t4 where

(TCS_CLOSE-GEOMETRIC_CLOSE)>@average;

create table t5 as

select symbol,

avg(close) as

CLOSE_PRICE_AVERAGE from

nsedata where timestamp

like "__-APR-2012" group

by symbol order by

avg(close) desc;

select * from t5;
```



```
SET @average=0;
select (min(TCS_CLOSE-GEO
MAX(TCS_CLOSE-GEOMETRIC_C
into @average from t4;

select * from t4 where
(TCS_CLOSE-GEOMETRIC_CLOS

create table t5 as
select symbol,
avg(close) as
CLOSE_PRICE_AVERAGE from
nsedata where timestamp
like "__-APR-2012" group
by symbol order by
avg(close) desc;
select * from t5;
```

```
Data truncated for column 'CI
Data truncated for column 'CI
Data truncated for column 'CI
Data truncated for column 'CI
Data truncated for column 'CI
Data truncated for column 'CI
Data truncated for column 'CI
Data truncated for column 'CI
Data truncated for column 'CI
Data truncated for column 'CI
Data truncated for column 'CI
Data truncated for column 'CI
Data truncated for column 'CI
Data truncated for column 'CI
Data truncated for column 'CI
```

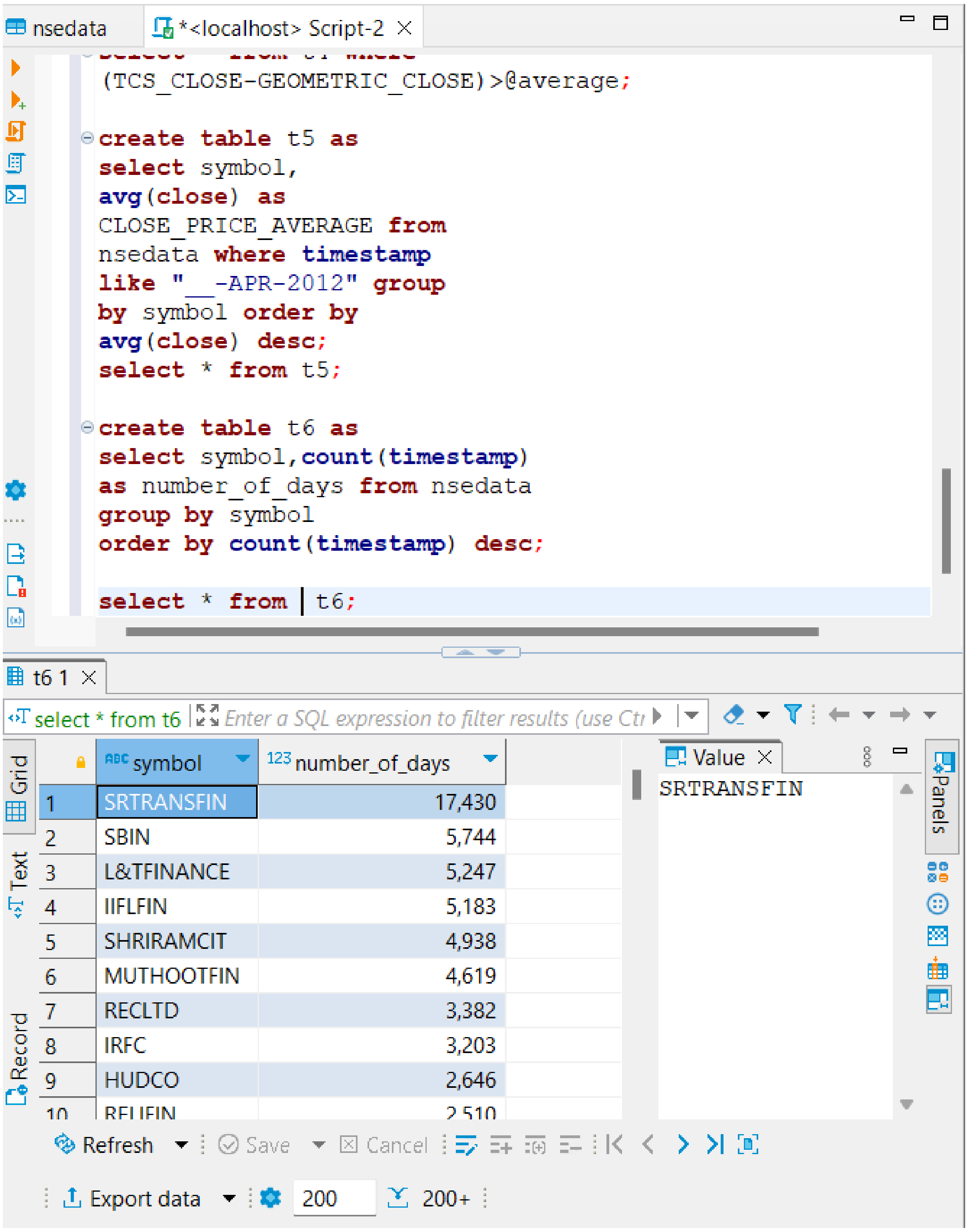
	symbol	CLOSE_PRICE_AVERAGE
1	ORISSAMINE	34,041.3868421053
2	MRF	10,993.7
3	SBIN	8,620.1764646465
4	BOSCHLTD	8,504.1
5	TIDEWATER	7,691.1236842105
6	NESTLEIND	4,813.3921052632
7	3MINDIA	4,157.8105263158
8	ALFALAVAL	3,935.85
9	GODFRYPHLP	3,574.0394736842
10	ASIANPAINT	3,359.1184210526

```
200 row(s) fetched - 261ms, on 2023-10-22 at 16:53:34
```

Q20)

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.:



nsedata

*<localhost> Script-2

```
(TCS_CLOSE-GEOMETRIC_CLOSE)>@average;

create table t5 as
select symbol,
avg(close) as
CLOSE_PRICE_AVERAGE from
nsedata where timestamp
like "__-APR-2012" group
by symbol order by
avg(close) desc;
select * from t5;

create table t6 as
select symbol,count(timestamp)
as number_of_days from nsedata
group by symbol
order by count(timestamp) desc;

select * from t6;
```

t6 1

select * from t6

	symbol	number_of_days
1	SRTRANSFIN	17,430
2	SBIN	5,744
3	L&TFINANCE	5,247
4	IIFLFIN	5,183
5	SHRIRAMCIT	4,938
6	MUTHOOTFIN	4,619
7	RECLTD	3,382
8	IRFC	3,203
9	HUDCO	2,646
10	RELIFIN	2,510

Value: SRTRANSFIN

Code used(for reference):

```
use stockdata;
show create table nsedata;
select count(*) from nsedata;
select count(*) from nsedata where timestamp like "__-OCT-2012";
select count(*) from nsedata where symbol="GEOMETRIC";
select * from nsedata where symbol="GEOMETRIC" limit 10;
select count(*) from nsedata where symbol="INFY";
select symbol,open,high,low,close,timestamp
from nsedata where symbol ="3IINFOTECH" limit 10;
create temporary table t1 as
select symbol,open,high,low,close,timestamp
from nsedata where symbol="3IINFOTECH" limit 10;
select * from t1;
SELECT
MIN(CLOSE),
AVG(CLOSE),
STDDEV(CLOSE),
MAX(CLOSE),
VAR_POP(CLOSE) AS VARIANCE
from t1;
SELECT AVG(close) AS median
FROM (
SELECT close
FROM t1
ORDER BY close ASC
LIMIT 2 OFFSET 4
) AS t2;
drop temporary table t1;
create table t2 as select symbol,avg(close) as average
```

```
from nsedata n group by symbol;
select * from t2;
create table t3 as
select symbol,
open,
close
from nsedata where symbol="GEOMETRIC"
and timestamp like "__-OCT-2012";
ALTER TABLE t3 add average float;
update t3 set average=(open+close)/2;
select * from t3;
create table table1 as select close as TCS_CLOSE,
timestamp from nsedata where symbol="TCS";
create table table2 as select close as GEOMETRIC_CLOSE,
timestamp from nsedata where symbol="GEOMETRIC";
create table t4 as select table1.timestamp,
TCS_CLOSE,table2.GEOMETRIC_CLOSE
from table1 inner join table2 on
table1.timestamp=table2.timestamp;
select * from t4;
select MIN(TCS_CLOSE-GEOMETRIC_CLOSE),
MAX(TCS_CLOSE-GEOMETRIC_CLOSE) from t4;
SET @average=0;
select (min(TCS_CLOSE-GEOMETRIC_CLOSE)+
MAX(TCS_CLOSE-GEOMETRIC_CLOSE))/2
into @average from t4;
select * from t4 where
(TCS_CLOSE-GEOMETRIC_CLOSE)>@average;
create table t5 as
```

```
select symbol,
avg(close) as
CLOSE_PRICE_AVERAGE from
nsedata where timestamp
like "__-APR-2012" group
by symbol order by
avg(close) desc;
select * from t5;
create table t6 as
select symbol,count(timestamp)
as number_of_days from nsedata
group by symbol
order by count(timestamp) desc;
select * from  t6;
```

The above code has been executed line by line,one line at a time.