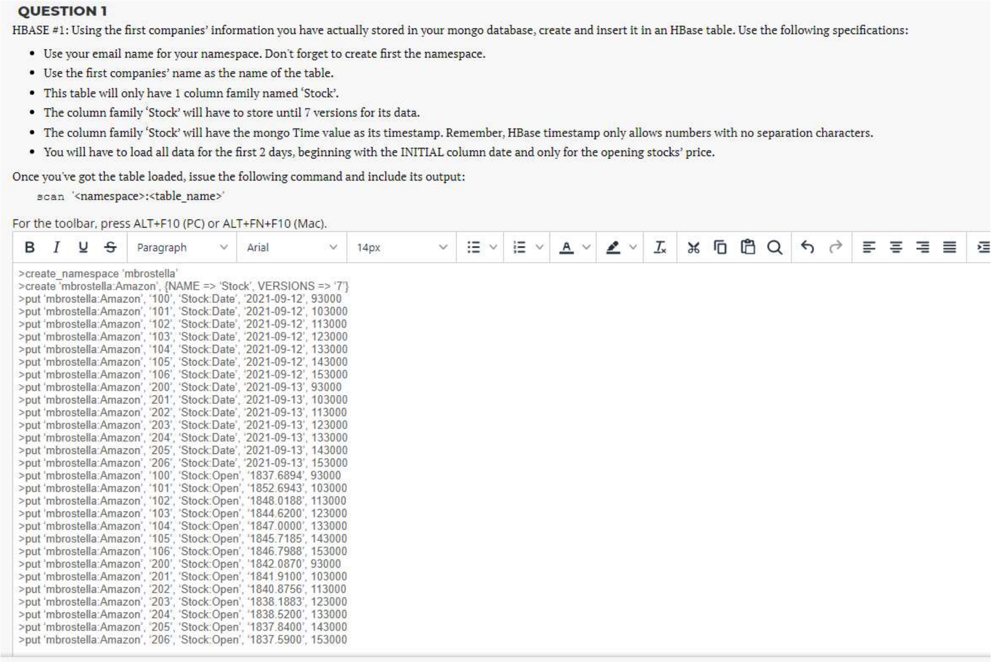
Group Assignment

Monday, October 25, 2021 9:34 PM



QUESTION 1

HBASE #1: Using the first companies' information you have actually stored in your mongo database, create and insert it in an HBase table. Use the following specifications:

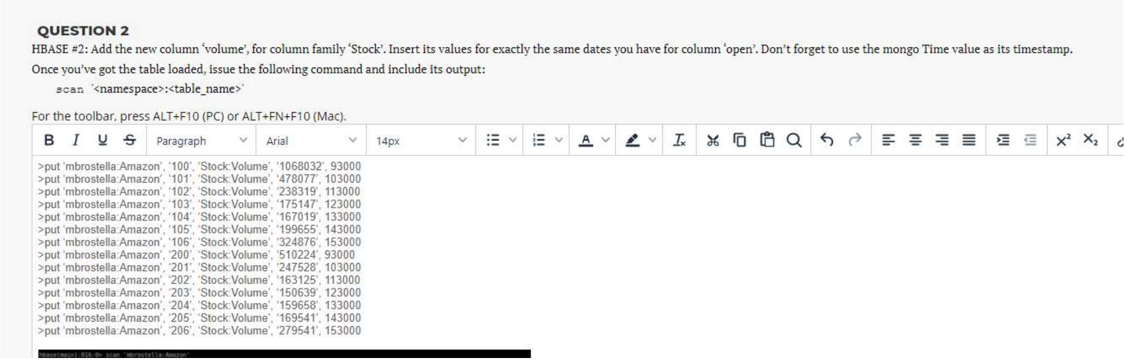
- Use your email name for your namespace. Don't forget to create first the namespace.
- Use the first companies' name as the name of the table.
- This table will only have 1 column family named 'Stock'.
- The column family 'Stock' will have to store until 7 versions for its data.
- The column family 'Stock' will have the mongo Time value as its timestamp. Remember, HBase timestamp only allows numbers with no separation characters.
- You will have to load all data for the first 2 days, beginning with the INITIAL column date and only for the opening stocks' price.

Once you've got the table loaded, issue the following command and include its output:

```
scan '<namespace>:<table_name>'
```

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).

```
>create_namespace 'mbrostella'
>create 'mbrostella:Amazon', {NAME => 'Stock', VERSIONS => '7'}
>put 'mbrostella:Amazon', '100', 'Stock:Date', '2021-09-12', 93000
>put 'mbrostella:Amazon', '101', 'Stock:Date', '2021-09-12', 103000
>put 'mbrostella:Amazon', '102', 'Stock:Date', '2021-09-12', 113000
>put 'mbrostella:Amazon', '103', 'Stock:Date', '2021-09-12', 123000
>put 'mbrostella:Amazon', '104', 'Stock:Date', '2021-09-12', 133000
>put 'mbrostella:Amazon', '105', 'Stock:Date', '2021-09-12', 143000
>put 'mbrostella:Amazon', '106', 'Stock:Date', '2021-09-12', 153000
>put 'mbrostella:Amazon', '200', 'Stock:Date', '2021-09-13', 93000
>put 'mbrostella:Amazon', '201', 'Stock:Date', '2021-09-13', 103000
>put 'mbrostella:Amazon', '202', 'Stock:Date', '2021-09-13', 113000
>put 'mbrostella:Amazon', '203', 'Stock:Date', '2021-09-13', 123000
>put 'mbrostella:Amazon', '204', 'Stock:Date', '2021-09-13', 133000
>put 'mbrostella:Amazon', '205', 'Stock:Date', '2021-09-13', 143000
>put 'mbrostella:Amazon', '206', 'Stock:Date', '2021-09-13', 153000
>put 'mbrostella:Amazon', '100', 'Stock:Open', '1837.6894', 93000
>put 'mbrostella:Amazon', '101', 'Stock:Open', '1852.6943', 103000
>put 'mbrostella:Amazon', '102', 'Stock:Open', '1848.0188', 113000
>put 'mbrostella:Amazon', '103', 'Stock:Open', '1844.6200', 123000
>put 'mbrostella:Amazon', '104', 'Stock:Open', '1847.0000', 133000
>put 'mbrostella:Amazon', '105', 'Stock:Open', '1845.7185', 143000
>put 'mbrostella:Amazon', '106', 'Stock:Open', '1846.7988', 153000
>put 'mbrostella:Amazon', '200', 'Stock:Open', '1842.0870', 93000
>put 'mbrostella:Amazon', '201', 'Stock:Open', '1841.9100', 103000
>put 'mbrostella:Amazon', '202', 'Stock:Open', '1840.8756', 113000
>put 'mbrostella:Amazon', '203', 'Stock:Open', '1838.1883', 123000
>put 'mbrostella:Amazon', '204', 'Stock:Open', '1838.5200', 133000
>put 'mbrostella:Amazon', '205', 'Stock:Open', '1837.8400', 143000
>put 'mbrostella:Amazon', '206', 'Stock:Open', '1837.5900', 153000
```



QUESTION 2

HBASE #2: Add the new column 'volume', for column family 'Stock'. Insert its values for exactly the same dates you have for column 'open'. Don't forget to use the mongo Time value as its timestamp.

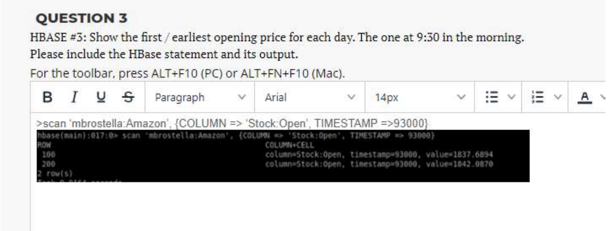
Once you've got the table loaded, issue the following command and include its output:

```
scan '<namespace>:<table_name>'
```

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).

```
>put 'mbrostella:Amazon', '100', 'Stock:Volume', '1068032', 93000
>put 'mbrostella:Amazon', '101', 'Stock:Volume', '478077', 103000
>put 'mbrostella:Amazon', '102', 'Stock:Volume', '238319', 113000
>put 'mbrostella:Amazon', '103', 'Stock:Volume', '175147', 123000
>put 'mbrostella:Amazon', '104', 'Stock:Volume', '167019', 133000
>put 'mbrostella:Amazon', '105', 'Stock:Volume', '199655', 143000
>put 'mbrostella:Amazon', '106', 'Stock:Volume', '324876', 153000
>put 'mbrostella:Amazon', '200', 'Stock:Volume', '510224', 93000
>put 'mbrostella:Amazon', '201', 'Stock:Volume', '247528', 103000
>put 'mbrostella:Amazon', '202', 'Stock:Volume', '163125', 113000
>put 'mbrostella:Amazon', '203', 'Stock:Volume', '150639', 123000
>put 'mbrostella:Amazon', '204', 'Stock:Volume', '159658', 133000
>put 'mbrostella:Amazon', '205', 'Stock:Volume', '169541', 143000
>put 'mbrostella:Amazon', '206', 'Stock:Volume', '279541', 153000
```

```
hbase(main):016:0> scan 'mbrostella:Amazon'
```



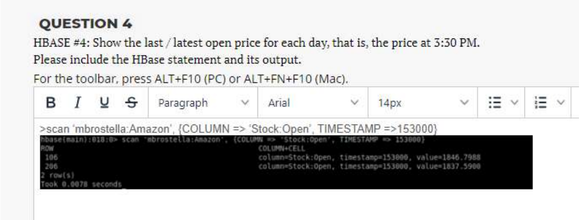
QUESTION 3

HBASE #3: Show the first / earliest opening price for each day. The one at 9:30 in the morning.
Please include the HBase statement and its output.

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).

```
>scan 'mbrostella:Amazon', {COLUMN => 'Stock:Open', TIMESTAMP =>93000}
```

```
hbase(main):017:0> scan 'mbrostella:Amazon', {COLUMN => 'Stock:Open', TIMESTAMP => 93000}
ROW                                         COLUMN+CELL
 100                                        column=Stock:Open, timestamp=93000, value=1837.6894
 200                                        column=Stock:Open, timestamp=93000, value=1842.0870
2 row(s)
```



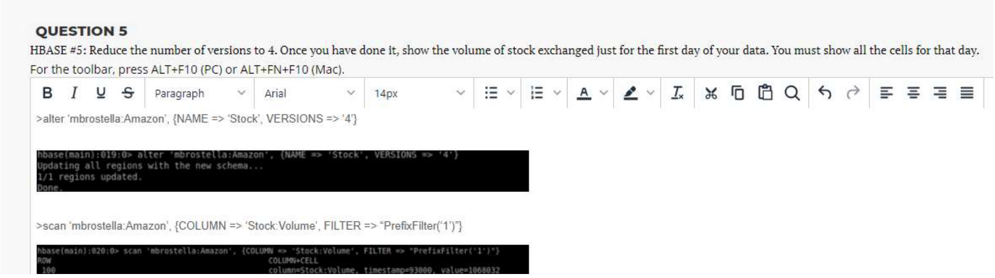
QUESTION 4

HBASE #4: Show the last / latest open price for each day, that is, the price at 3:30 PM.
Please include the HBase statement and its output.

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).

```
>scan 'mbrostella:Amazon', {COLUMN => 'Stock:Open', TIMESTAMP =>153000}
```

```
hbase(main):018:0> scan 'mbrostella:Amazon', {COLUMN => 'Stock:Open', TIMESTAMP => 153000}
ROW                                         COLUMN+CELL
 106                                        column=Stock:Open, timestamp=153000, value=1846.7988
 206                                        column=Stock:Open, timestamp=153000, value=1837.5900
2 row(s)
Took 0.0078 seconds
```



QUESTION 5

HBASE #5: Reduce the number of versions to 4. Once you have done it, show the volume of stock exchanged just for the first day of your data. You must show all the cells for that day.

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).

```
>alter 'mbrostella:Amazon', {NAME => 'Stock', VERSIONS => '4'}
```

```
hbase(main):019:0> alter 'mbrostella:Amazon', {NAME => 'Stock', VERSIONS => '4'}
Updating all regions with the new schema...
1/1 regions updated.
Done.
```

```
>scan 'mbrostella:Amazon', {COLUMN => 'Stock:Volume', FILTER => "PrefixFilter('1')"}
```

```
hbase(main):020:0> scan 'mbrostella:Amazon', {COLUMN => 'Stock:Volume', FILTER => "PrefixFilter('1')"}
ROW                                         COLUMN+CELL
 100                                        column=Stock:Volume, timestamp=93000, value=1068032
```

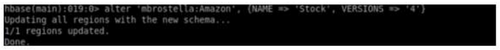
QUESTION 5

HBASE #5: Reduce the number of versions to 4. Once you have done it, show the volume of stock exchanged just for the first day of your data. You must show all the cells for that day.

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).

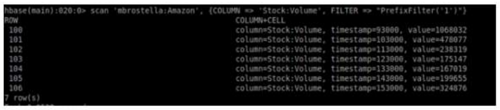
B I U S Paragraph Arial 14px

>alter 'mbrostella:Amazon', {NAME => 'Stock', VERSIONS => '4'}



```
hbase(main):019:0> alter 'mbrostella:Amazon', {NAME => 'Stock', VERSIONS => '4'}
Updating all regions with the new schema...
1/1 regions updated.
Done.
```

>scan 'mbrostella:Amazon', {COLUMN => 'Stock:Volume', FILTER => "PrefixFilter('1')"}



```
hbase(main):020:0> scan 'mbrostella:Amazon', {COLUMN => 'Stock:Volume', FILTER => "PrefixFilter('1')"}
ROW                                          COLUMN+CELL
 100                                         column=Stock:Volume, timestamp=93000, value=1068032
 101                                         column=Stock:Volume, timestamp=103000, value=478077
 102                                         column=Stock:Volume, timestamp=113000, value=238319
 103                                         column=Stock:Volume, timestamp=123000, value=175147
 104                                         column=Stock:Volume, timestamp=133000, value=167019
 105                                         column=Stock:Volume, timestamp=143000, value=199655
 106                                         column=Stock:Volume, timestamp=153000, value=324876
7 row(s)
```

QUESTION 6

MONGO #1: Load your first companies' JSON documents into your default database and the stock collection. Each group should only load the companies' files specified under the column FIRST. You should use the mongoimport command. The aim is to have your first companies' data loaded in the stock collection.

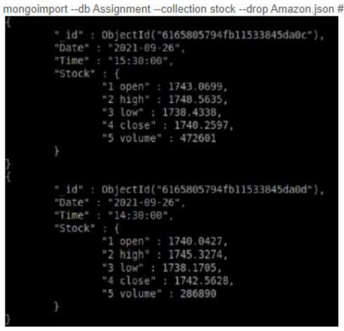
Once you've got all documents in the collection, please do the following transformation in the collection:

1. Insert the companies' stock name into a new field named Company. Show the mongo statement and its output.
2. Update the name of the embedded fields in the Stock document. You should remove the number and the space that follows the number from all the embedded fields. For example, field "1 open" should be converted to "open". The same should be done for the rest of fields in the embedded stock document. As usual, show the mongo statement and its output.

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).

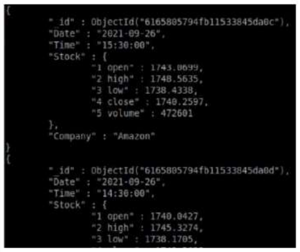
B I U S Paragraph Arial 14px

mongoimport --db Assignment --collection stock --drop Amazon.json #part 6.1



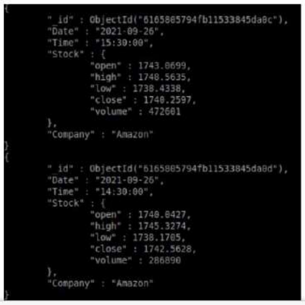
```
{
        "_id" : ObjectId("6165805794fb11533845da0c"),
        "Date" : "2021-09-26",
        "Time" : "15:30:00",
        "Stock" : {
                "1 open" : 1743.0699,
                "2 high" : 1748.5635,
                "3 low" : 1738.4338,
                "4 close" : 1740.2597,
                "5 volume" : 472601
        }
}
{
        "_id" : ObjectId("6165805794fb11533845da0d"),
        "Date" : "2021-09-26",
        "Time" : "14:30:00",
        "Stock" : {
                "1 open" : 1740.0427,
                "2 high" : 1745.3274,
                "3 low" : 1738.1705,
                "4 close" : 1742.5628,
                "5 volume" : 286890
        }
}
```

db.stock.update({},{$set: {"Company": "Amazon"}}, {multi: true}) #part 6.1



```
{
        "_id" : ObjectId("6165805794fb11533845da0c"),
        "Date" : "2021-09-26",
        "Time" : "15:30:00",
        "Stock" : {
                "1 open" : 1743.0699,
                "2 high" : 1748.5635,
                "3 low" : 1738.4338,
                "4 close" : 1740.2597,
                "5 volume" : 472601
        },
        "Company" : "Amazon"
}
{
        "_id" : ObjectId("6165805794fb11533845da0d"),
        "Date" : "2021-09-26",
        "Time" : "14:30:00",
        "Stock" : {
                "1 open" : 1740.0427,
                "2 high" : 1745.3274,
                "3 low" : 1738.1705,
```

```
db.stock.update({}, {$rename: {"Stock.1 open": "Stock.open"}}, {multi:true})
db.stock.update({}, {$rename: {"Stock.2 high": "Stock.high"}}, {multi:true})
db.stock.update({}, {$rename: {"Stock.3 low": "Stock.low"}}, {multi:true})
db.stock.update({}, {$rename: {"Stock.4 close": "Stock.close"}}, {multi:true})
db.stock.update({}, {$rename: {"Stock.5 volume": "Stock.volume"}}, {multi:true})
```



```
{
        "_id" : ObjectId("6165805794fb11533845da0c"),
        "Date" : "2021-09-26",
        "Time" : "15:30:00",
        "Stock" : {
                "open" : 1743.0699,
                "high" : 1748.5635,
                "low" : 1738.4338,
                "close" : 1740.2597,
                "volume" : 472601
        },
        "Company" : "Amazon"
}
{
        "_id" : ObjectId("6165805794fb11533845da0d"),
        "Date" : "2021-09-26",
        "Time" : "14:30:00",
        "Stock" : {
                "open" : 1740.0427,
                "high" : 1745.3274,
                "low" : 1738.1705,
                "close" : 1742.5628,
                "volume" : 286890
        },
        "Company" : "Amazon"
}
```

10 points Save Answer

QUESTION 7

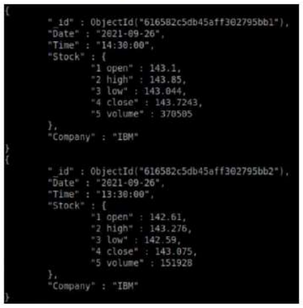
MONGO #2: Next, load your second companies' JSON documents into the same database and collection as before. Once you're done with the load or insert, do exactly the same transformations you did for your first company. Be careful with the companies' names. Each company has to identify its own data. So you can't change your first companies' content.

On both cases, please show the mongo statements you used to accomplish those transformations. Include also the output of both statements.

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).

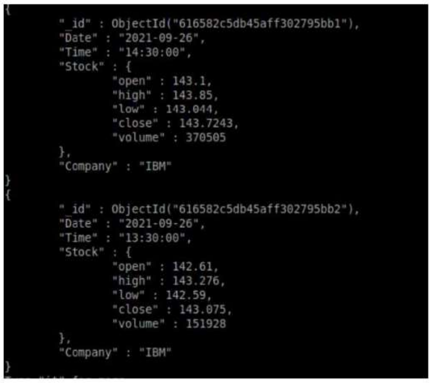
```
mongoimport --db Assignment --collection stock IBM.json

db.stock.update({Company: {$ne: "Amazon"}},{$set: {"Company": "IBM"}}, {multi: true})
```



```
{
        "_id" : ObjectId("616582c5db45aff302795bb1"),
        "Date" : "2021-09-26",
        "Time" : "14:30:00",
        "Stock" : {
                "1 open" : 143.1,
                "2 high" : 143.85,
                "3 low" : 143.044,
                "4 close" : 143.7243,
                "5 volume" : 370505
        },
        "Company" : "IBM"
}
{
        "_id" : ObjectId("616582c5db45aff302795bb2"),
        "Date" : "2021-09-26",
        "Time" : "13:30:00",
        "Stock" : {
                "1 open" : 142.61,
                "2 high" : 143.276,
                "3 low" : 142.59,
                "4 close" : 143.075,
                "5 volume" : 151928
        },
        "Company" : "IBM"
}
```

```
db.stock.update({Company: {$ne: "Amazon"}}, {$rename: {"Stock.1 open": "Stock.open"}}, {multi:true})
db.stock.update({Company: {$ne: "Amazon"}}, {$rename: {"Stock.2 high": "Stock.high"}}, {multi:true})
db.stock.update({Company: {$ne: "Amazon"}}, {$rename: {"Stock.3 low": "Stock.low"}}, {multi:true})
db.stock.update({Company: {$ne: "Amazon"}}, {$rename: {"Stock.4 close": "Stock.close"}}, {multi:true})
db.stock.update({Company: {$ne: "Amazon"}}, {$rename: {"Stock.5 volume": "Stock.volume"}}, {multi:true})
```



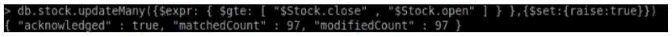
```
{
        "_id" : ObjectId("616582c5db45aff302795bb1"),
        "Date" : "2021-09-26",
        "Time" : "14:30:00",
        "Stock" : {
                "open" : 143.1,
                "high" : 143.85,
                "low" : 143.044,
                "close" : 143.7243,
                "volume" : 370505
        },
        "Company" : "IBM"
}
{
        "_id" : ObjectId("616582c5db45aff302795bb2"),
        "Date" : "2021-09-26",
        "Time" : "13:30:00",
        "Stock" : {
                "open" : 142.61,
                "high" : 143.276,
                "low" : 142.59,
                "close" : 143.075,
                "volume" : 151928
        },
        "Company" : "IBM"
}
```

QUESTION 8

MONGO #3: Add a new Boolean field named "raise" in all embedded stock documents. If the stocks' closing price is higher or equal than its open price, we'll store a true value in the raise field. Else, if the closing price is smaller than the open price, we'll store a false value. Show the mongo statement you used, and its output, of course.

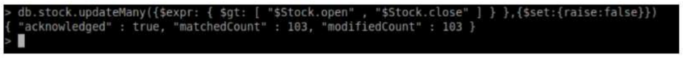
For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).

```
db.stock.updateMany({$expr: { $gte: [ "$Stock.close" , "$Stock.open" ] } },{$set:{raise:true}})
```

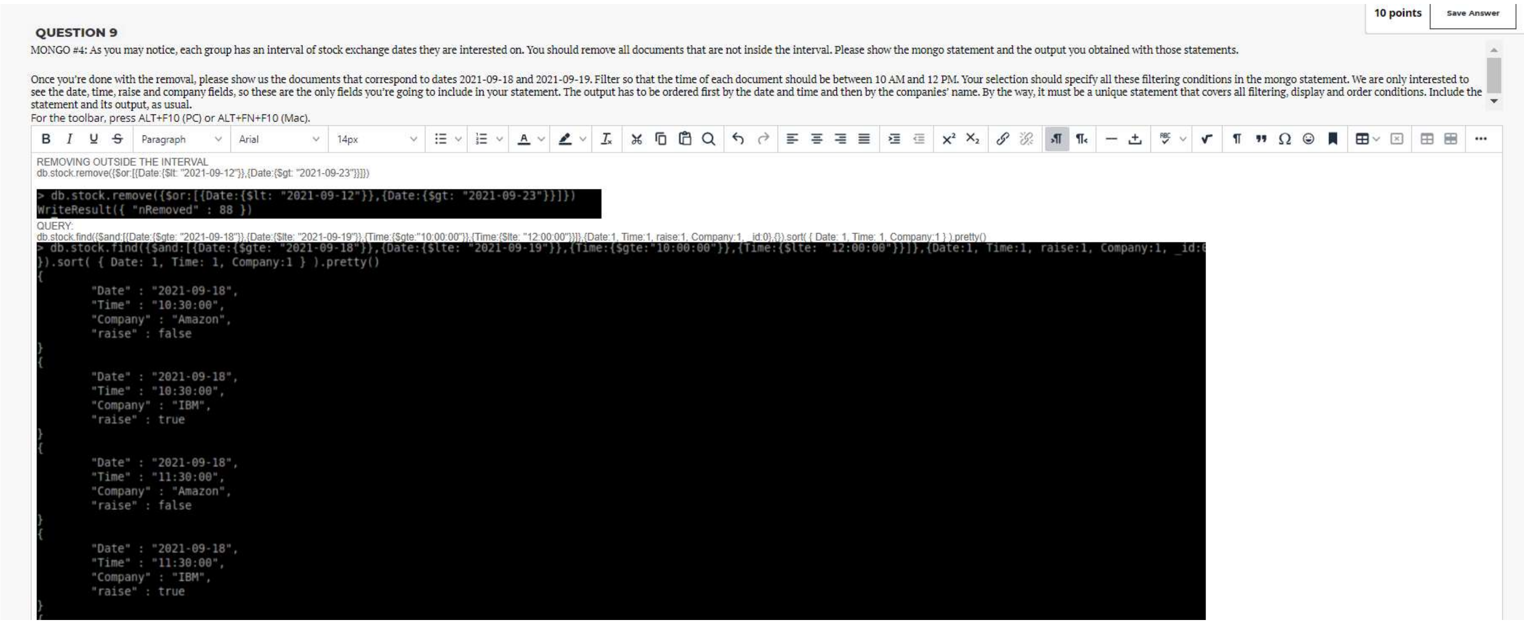


```
> db.stock.updateMany({$expr: { $gte: [ "$Stock.close" , "$Stock.open" ] } },{$set:{raise:true}})
{ "acknowledged" : true, "matchedCount" : 97, "modifiedCount" : 97 }
```

```
db.stock.updateMany({$expr: { $gt: [ "$Stock.open" , "$Stock.close" ] } },{$set:{raise:false}})
```



```
> db.stock.updateMany({$expr: { $gt: [ "$Stock.open" , "$Stock.close" ] } },{$set:{raise:false}})
{ "acknowledged" : true, "matchedCount" : 103, "modifiedCount" : 103 }
>
```



10 points

Save Answer

QUESTION 9

MONGO #4: As you may notice, each group has an interval of stock exchange dates they are interested on. You should remove all documents that are not inside the interval. Please show the mongo statement and the output you obtained with those statements.

Once you're done with the removal, please show us the documents that correspond to dates 2021-09-18 and 2021-09-19. Filter so that the time of each document should be between 10 AM and 12 PM. Your selection should specify all these filtering conditions in the mongo statement. We are only interested to see the date, time, raise and company fields, so these are the only fields you're going to include in your statement. The output has to be ordered first by the date and time and then by the companies' name. By the way, it must be a unique statement that covers all filtering, display and order conditions. Include the statement and its output, as usual.

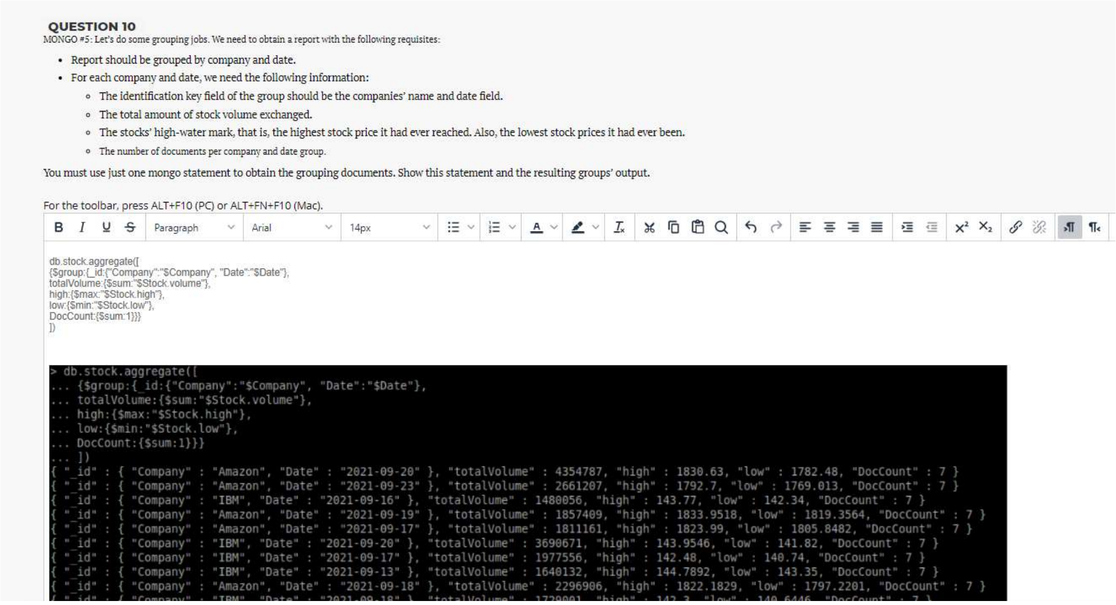
For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).

REMOVING OUTSIDE THE INTERVAL
db.stock.remove({$or:[{Date:{$lt: "2021-09-12"}},{Date:{$gt: "2021-09-23"}}]})

```
> db.stock.remove({$or:[{Date:{$lt: "2021-09-12"}},{Date:{$gt: "2021-09-23"}}]})
WriteResult({ "nRemoved" : 88 })
```

QUERY:
db.stock.find({$and:[{Date:{$gte: "2021-09-18"}},{Date:{$lte: "2021-09-19"}},{Time:{$gte:"10:00:00"}},{Time:{$lte: "12:00:00"}}]},{Date:1, Time:1, raise:1, Company:1, _id:0},{}).sort({ Date: 1, Time: 1, Company:1 }).pretty()

```
> db.stock.find({$and:[{Date:{$gte: "2021-09-18"}},{Date:{$lte: "2021-09-19"}},{Time:{$gte:"10:00:00"}},{Time:{$lte: "12:00:00"}}]},{Date:1, Time:1, raise:1, Company:1, _id:(
}).sort( { Date: 1, Time: 1, Company:1 } ).pretty()
{
        "Date" : "2021-09-18",
        "Time" : "10:30:00",
        "Company" : "Amazon",
        "raise" : false
}
{
        "Date" : "2021-09-18",
        "Time" : "10:30:00",
        "Company" : "IBM",
        "raise" : true
}
{
        "Date" : "2021-09-18",
        "Time" : "11:30:00",
        "Company" : "Amazon",
        "raise" : false
}
{
        "Date" : "2021-09-18",
        "Time" : "11:30:00",
        "Company" : "IBM",
        "raise" : true
}
{
```



QUESTION 10

MONGO #5: Let's do some grouping jobs. We need to obtain a report with the following requisites:

- Report should be grouped by company and date.
- For each company and date, we need the following information:
  - The identification key field of the group should be the companies' name and date field.
  - The total amount of stock volume exchanged.
  - The stocks' high-water mark, that is, the highest stock price it had ever reached. Also, the lowest stock prices it had ever been.
  - The number of documents per company and date group.

You must use just one mongo statement to obtain the grouping documents. Show this statement and the resulting groups' output.

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).

```
db.stock.aggregate([
{$group:{_id:{"Company":"$Company", "Date":"$Date"},
totalVolume:{$sum:"$Stock.volume"},
high:{$max:"$Stock.high"},
low:{$min:"$Stock.low"},
DocCount:{$sum:1}}}
])
```

```
> db.stock.aggregate([
... {$group:{_id:{"Company":"$Company", "Date":"$Date"},
... totalVolume:{$sum:"$Stock.volume"},
... high:{$max:"$Stock.high"},
... low:{$min:"$Stock.low"},
... DocCount:{$sum:1}}}
... ])
{ "_id" : { "Company" : "Amazon", "Date" : "2021-09-20" }, "totalVolume" : 4354787, "high" : 1830.63, "low" : 1782.48, "DocCount" : 7 }
{ "_id" : { "Company" : "Amazon", "Date" : "2021-09-23" }, "totalVolume" : 2661207, "high" : 1792.7, "low" : 1769.013, "DocCount" : 7 }
{ "_id" : { "Company" : "IBM", "Date" : "2021-09-16" }, "totalVolume" : 1480056, "high" : 143.77, "low" : 142.34, "DocCount" : 7 }
{ "_id" : { "Company" : "Amazon", "Date" : "2021-09-19" }, "totalVolume" : 1857409, "high" : 1833.9518, "low" : 1819.3564, "DocCount" : 7 }
{ "_id" : { "Company" : "Amazon", "Date" : "2021-09-17" }, "totalVolume" : 1811161, "high" : 1823.99, "low" : 1805.8482, "DocCount" : 7 }
{ "_id" : { "Company" : "IBM", "Date" : "2021-09-20" }, "totalVolume" : 3690671, "high" : 143.9546, "low" : 141.82, "DocCount" : 7 }
{ "_id" : { "Company" : "IBM", "Date" : "2021-09-17" }, "totalVolume" : 1977556, "high" : 142.48, "low" : 140.74, "DocCount" : 7 }
{ "_id" : { "Company" : "IBM", "Date" : "2021-09-13" }, "totalVolume" : 1640132, "high" : 144.7892, "low" : 143.35, "DocCount" : 7 }
{ "_id" : { "Company" : "Amazon", "Date" : "2021-09-18" }, "totalVolume" : 2296906, "high" : 1822.1829, "low" : 1797.2201, "DocCount" : 7 }
{ "_id" : { "Company" : "IBM", "Date" : "2021-09-18" }, "totalVolume" : 1729001, "high" : 142.3, "low" : 140.6446, "DocCount" : 7 }
```