*** Time Series Collections ***

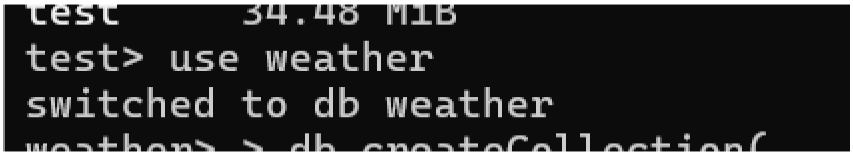
1. Make sure you have MongoDB Database Tools installed (https://docs.mongodb.com/database-tools/installation/installation/).

2. Make sure you have a local MongoDB instance running (see lab #2) or use the remote MongoDB Atlas instance (see lab #1). You'll need at least MongoDB v5 for this lab.

3. Download the Timisoara 2020 weather data

4. Connect to MongoDB using shell
mongo mongodb://mongoadmin:secret@localhost:27888/?authSource=admin

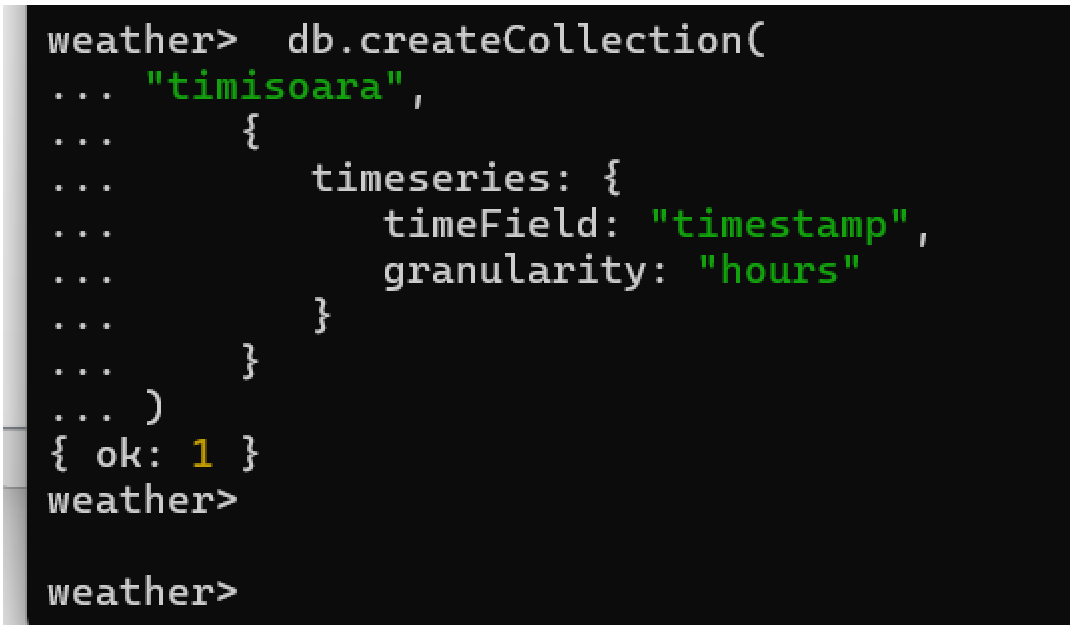
5. Create the database
> use weather



```
test      34.48 MiB
test> use weather
switched to db weather
weather> > db.createCollection(
```

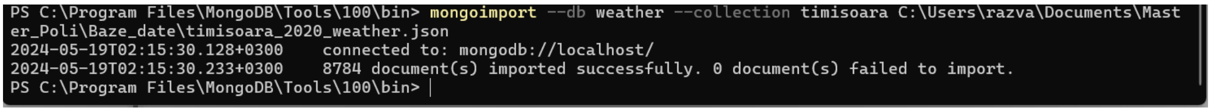
6. Create the time series collection to hold the historic weather data

```
> db.createCollection(
   "timisoara",
   {
     timeseries: {
       timeField: "timestamp",
       granularity: "hours"
     }
   }
)
```



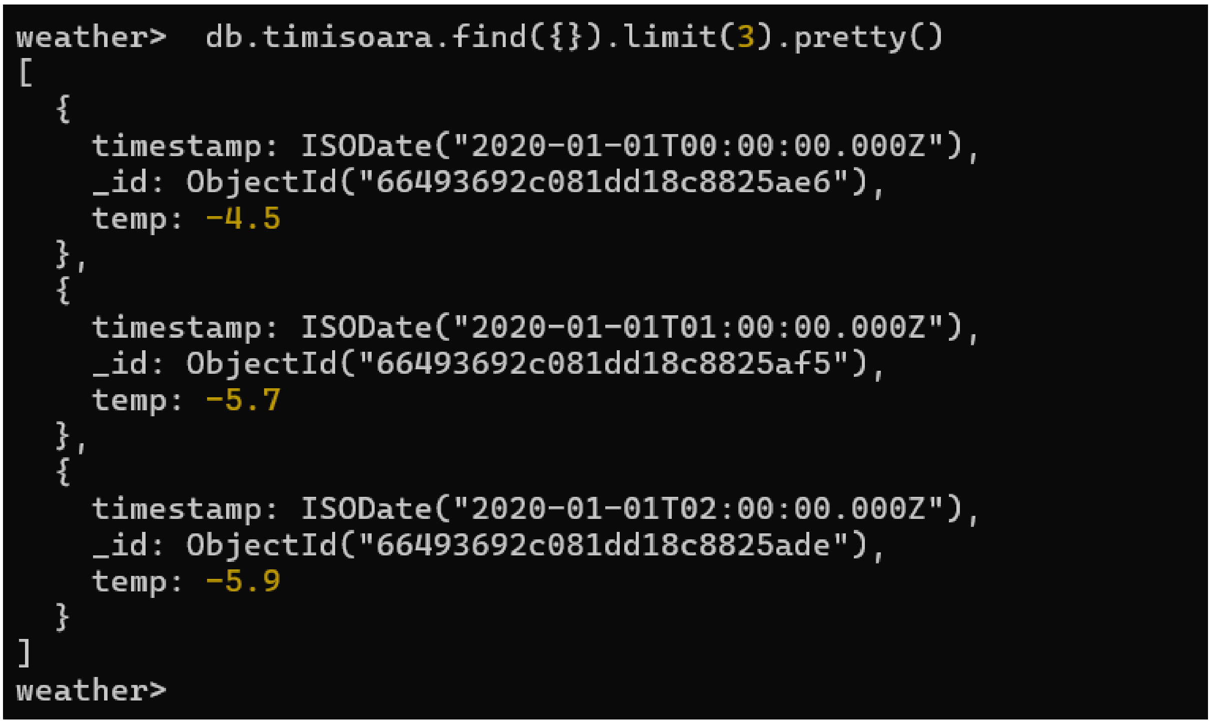
```
weather>  db.createCollection(
... "timisoara",
...     {
...         timeseries: {
...            timeField: "timestamp",
...            granularity: "hours"
...         }
...     }
... )
{ ok: 1 }
weather>

weather>
```

6. Import the weather sample data (make sure you run this at the command prompt, not inside the Mongo Shell)
mongoimport --db=weather --collection=timisoara
mongodb://mongoadmin:secret@localhost:27888/?authSource=admin
timisoara_2020_weather.json



```
PS C:\Program Files\MongoDB\Tools\100\bin> mongoimport --db weather --collection timisoara C:\Users\razva\Documents\Mast
er_Poli\Baze_date\timisoara_2020_weather.json
2024-05-19T02:15:30.128+0300    connected to: mongodb://localhost/
2024-05-19T02:15:30.233+0300    8784 document(s) imported successfully. 0 document(s) failed to import.
PS C:\Program Files\MongoDB\Tools\100\bin>
```

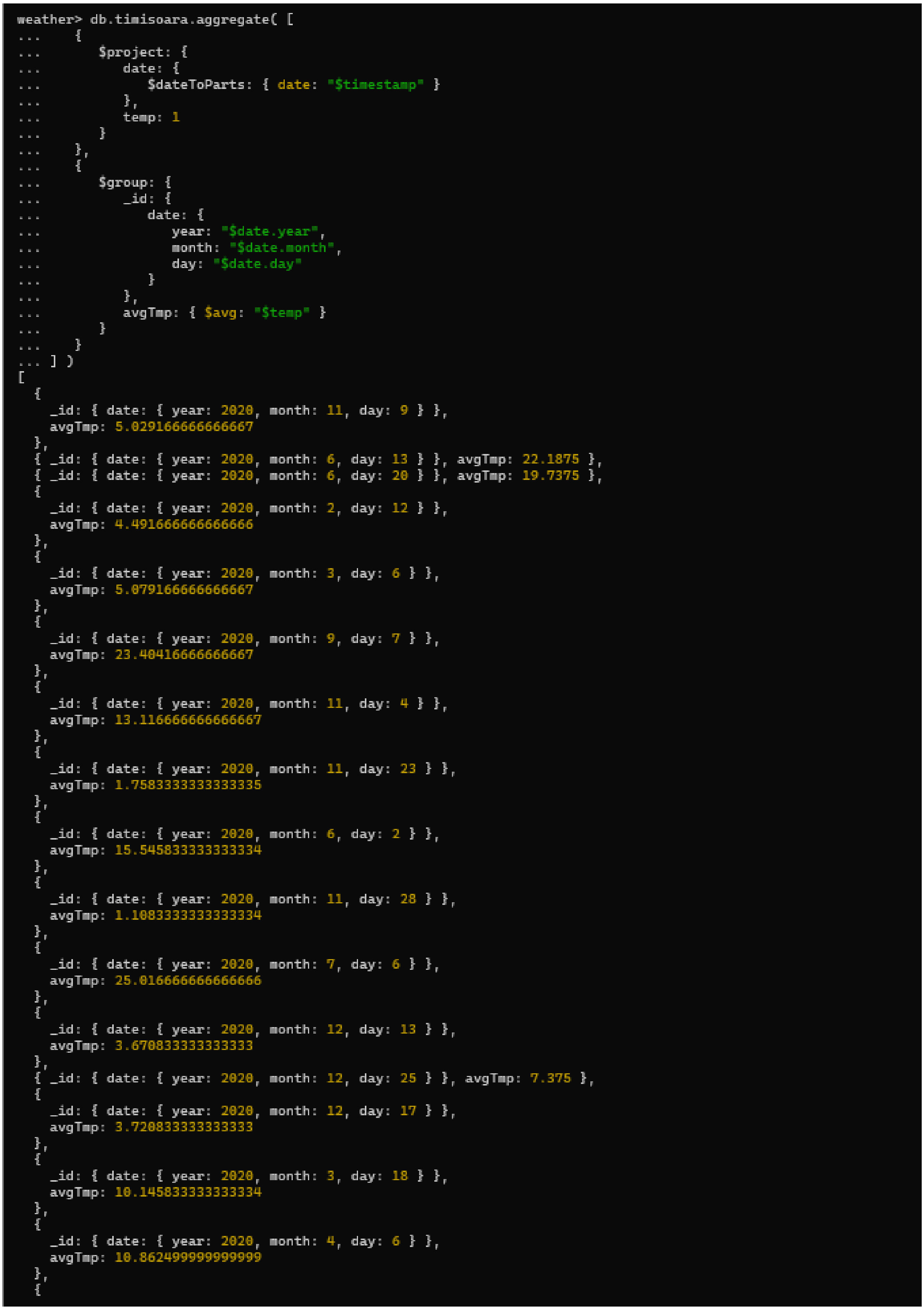
7. See some sample documents, to get a taste of the sample data
> use weather
> db.timisoara.find({}).limit(3).pretty()



```
weather>  db.timisoara.find({}).limit(3).pretty()
[
  {
    timestamp: ISODate("2020-01-01T00:00:00.000Z"),
    _id: ObjectId("66493692c081dd18c8825ae6"),
    temp: -4.5
  },
  {
    timestamp: ISODate("2020-01-01T01:00:00.000Z"),
    _id: ObjectId("66493692c081dd18c8825af5"),
    temp: -5.7
  },
  {
    timestamp: ISODate("2020-01-01T02:00:00.000Z"),
    _id: ObjectId("66493692c081dd18c8825ade"),
    temp: -5.9
  }
]
weather>
```

8. Get average temperature for each day in the data sample

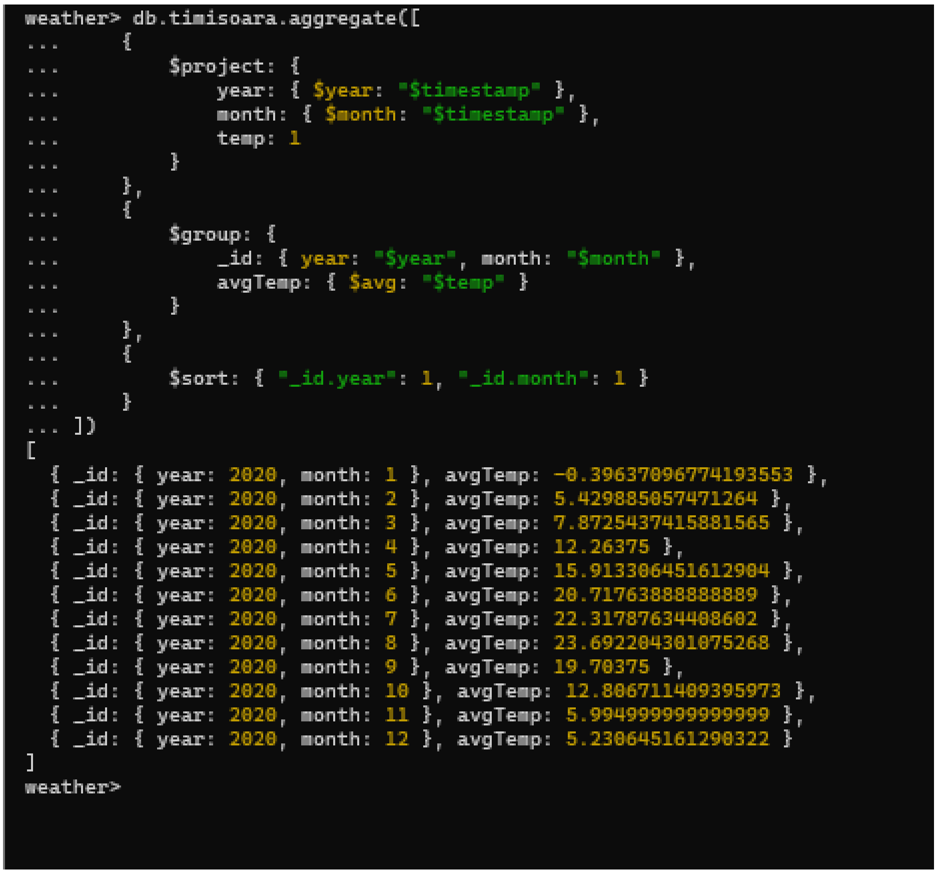
```
> db.timisoara.aggregate( [
  {
    $project: {
      date: {
        $dateToParts: { date: "$timestamp" }
      },
      temp: 1
    }
  },
  {
    $group: {
      _id: {
        date: {
          year: "$date.year",
          month: "$date.month",
          day: "$date.day"
        }
      },
      avgTmp: { $avg: "$temp" }
    }
  }
] )
```



```
weather> db.timisoara.aggregate( [
...    {
...       $project: {
...          date: {
...             $dateToParts: { date: "$timestamp" }
...          },
...          temp: 1
...       }
...    },
...    {
...       $group: {
...          _id: {
...             date: {
...                year: "$date.year",
...                month: "$date.month",
...                day: "$date.day"
...             }
...          },
...          avgTmp: { $avg: "$temp" }
...       }
...    }
... ] )
[
  {
    _id: { date: { year: 2020, month: 11, day: 9 } },
    avgTmp: 5.029166666666667
  },
  { _id: { date: { year: 2020, month: 6, day: 13 } }, avgTmp: 22.1875 },
  { _id: { date: { year: 2020, month: 6, day: 20 } }, avgTmp: 19.7375 },
  {
    _id: { date: { year: 2020, month: 2, day: 12 } },
    avgTmp: 4.491666666666666
  },
  {
    _id: { date: { year: 2020, month: 3, day: 6 } },
    avgTmp: 5.079166666666667
  },
  {
    _id: { date: { year: 2020, month: 9, day: 7 } },
    avgTmp: 23.404166666666667
  },
  {
    _id: { date: { year: 2020, month: 11, day: 4 } },
    avgTmp: 13.116666666666667
  },
  {
    _id: { date: { year: 2020, month: 11, day: 23 } },
    avgTmp: 1.7583333333333335
  },
  {
    _id: { date: { year: 2020, month: 6, day: 2 } },
    avgTmp: 15.545833333333334
  },
  {
    _id: { date: { year: 2020, month: 11, day: 28 } },
    avgTmp: 1.1083333333333334
  },
  {
    _id: { date: { year: 2020, month: 7, day: 6 } },
    avgTmp: 25.016666666666666
  },
  {
    _id: { date: { year: 2020, month: 12, day: 13 } },
    avgTmp: 3.670833333333333
  },
  { _id: { date: { year: 2020, month: 12, day: 25 } }, avgTmp: 7.375 },
  {
    _id: { date: { year: 2020, month: 12, day: 17 } },
    avgTmp: 3.720833333333333
  },
  {
    _id: { date: { year: 2020, month: 3, day: 18 } },
    avgTmp: 10.145833333333334
  },
  {
    _id: { date: { year: 2020, month: 4, day: 6 } },
    avgTmp: 10.862499999999999
  },
  {
```

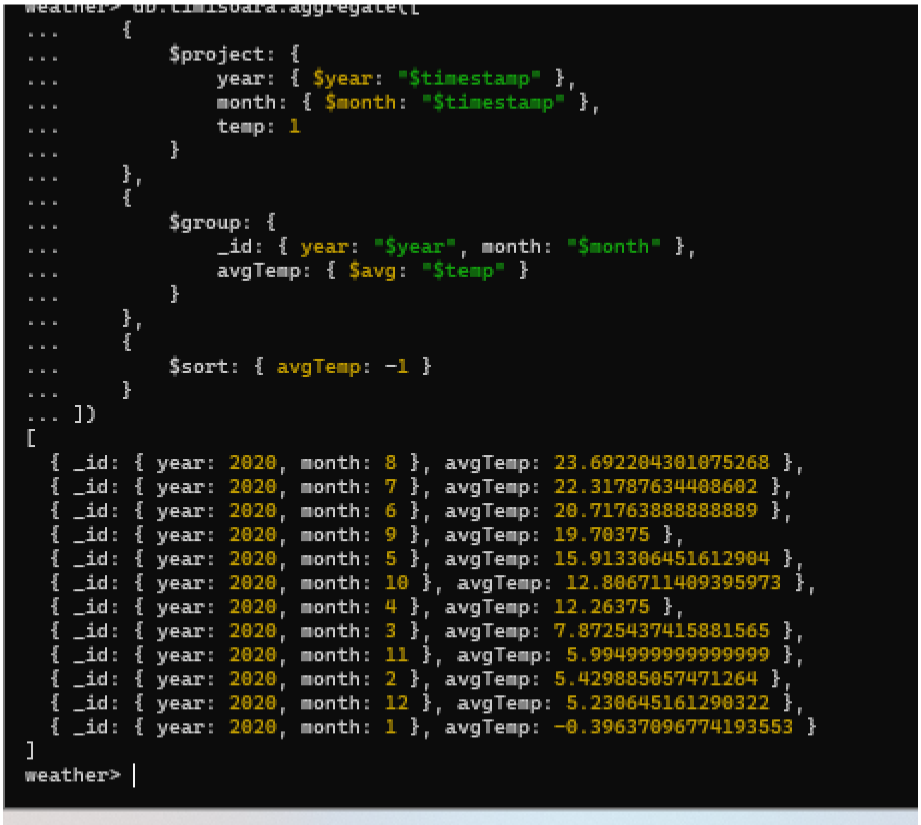
DO IT YOURSELF:

A. Get average temperature for each month in the data sample.



```
weather> db.timisoara.aggregate([
...     {
...         $project: {
...             year: { $year: "$timestamp" },
...             month: { $month: "$timestamp" },
...             temp: 1
...         }
...     },
...     {
...         $group: {
...             _id: { year: "$year", month: "$month" },
...             avgTemp: { $avg: "$temp" }
...         }
...     },
...     {
...         $sort: { "_id.year": 1, "_id.month": 1 }
...     }
... ])
[
  { _id: { year: 2020, month: 1 }, avgTemp: -0.39637096774193553 },
  { _id: { year: 2020, month: 2 }, avgTemp: 5.429885057471264 },
  { _id: { year: 2020, month: 3 }, avgTemp: 7.8725437415881565 },
  { _id: { year: 2020, month: 4 }, avgTemp: 12.26375 },
  { _id: { year: 2020, month: 5 }, avgTemp: 15.913306451612904 },
  { _id: { year: 2020, month: 6 }, avgTemp: 20.71763888888889 },
  { _id: { year: 2020, month: 7 }, avgTemp: 22.31787634408602 },
  { _id: { year: 2020, month: 8 }, avgTemp: 23.692204301075268 },
  { _id: { year: 2020, month: 9 }, avgTemp: 19.70375 },
  { _id: { year: 2020, month: 10 }, avgTemp: 12.806711409395973 },
  { _id: { year: 2020, month: 11 }, avgTemp: 5.9949999999999999 },
  { _id: { year: 2020, month: 12 }, avgTemp: 5.230645161290322 }
]
weather>
```

B. Display the previous list in descending order of the average temperatures.



```
weather> db.timisoara.aggregate([
...     {
...         $project: {
...             year: { $year: "$timestamp" },
...             month: { $month: "$timestamp" },
...             temp: 1
...         }
...     },
...     {
...         $group: {
...             _id: { year: "$year", month: "$month" },
...             avgTemp: { $avg: "$temp" }
...         }
...     },
...     {
...         $sort: { avgTemp: -1 }
...     }
... ])
[
  { _id: { year: 2020, month: 8 }, avgTemp: 23.692204301075268 },
  { _id: { year: 2020, month: 7 }, avgTemp: 22.31787634408602 },
  { _id: { year: 2020, month: 6 }, avgTemp: 20.71763888888889 },
  { _id: { year: 2020, month: 9 }, avgTemp: 19.70375 },
  { _id: { year: 2020, month: 5 }, avgTemp: 15.913306451612904 },
  { _id: { year: 2020, month: 10 }, avgTemp: 12.806711409395973 },
  { _id: { year: 2020, month: 4 }, avgTemp: 12.26375 },
  { _id: { year: 2020, month: 3 }, avgTemp: 7.8725437415881565 },
  { _id: { year: 2020, month: 11 }, avgTemp: 5.9949999999999999 },
  { _id: { year: 2020, month: 2 }, avgTemp: 5.429885057471264 },
  { _id: { year: 2020, month: 12 }, avgTemp: 5.230645161290322 },
  { _id: { year: 2020, month: 1 }, avgTemp: -0.39637096774193553 }
]
weather> |
```

NEXT:
It's about time to check whether you master MongoDB.

Next time we'll see the project that will get you the grade for this semester. For now just relax.