1. Calculate the average air temperature for all records.

2. Calculate the maximum wind speed for each type of wind direction.

3. Calculate the total estimated water depth for all records.

4. Find the most common sea state code and its occurrence count.

5. Calculate the average temperature and dew point for each month.

6. Calculate the average elevation and maximum wind speed for each type of sky condition.

7. Count the total number of documents in the collection.

8. Match documents with a temperature greater than 25 degrees Celsius and visibility greater than 1000 meters.

9. Count the number of documents with a visibility distance greater than 1000 meters.

10. Count the number of documents with a precipitation discrepancy of "2".

11. Match documents with a specific sky condition code (ceilingHeight.quality).

12. Match documents with a specific quality code for air temperature and dew point.

13. Match documents with a specific sea state code or visibility greater than 5000 meters.

14. Find distinct values of the "sections" array.

15. Retrieve unique "st" values along with their corresponding "dataSource" values.

16. Get unique "ts" values for documents with a "quality" of "1" in the "air Temperature" field.

17. Retrieve unique "value" and "quality" combinations from the "airTemperature" field.

18. Calculate the average air temperature and dew point temperature and store the results in a new collection called "average\_temperatures". (using mapReduce).

19. Calculate the average wind speed for each station and store the results in a new collection called "average\_wind\_speed\_by\_station". (using mapReduce).

20. Calculate the average sea surface temperature for each station and store the results in a new collection called "average\_sea\_surface\_temperature\_by\_station". (using mapReduce).

ANSWERS

1. db.data.aggregate({$group: {\_id: 0,avgtemp: {$avg: "$airTemperature.value"}}})
2. db.data.aggregate({$group: {\_id: "$wind.type", maxWindspeed: {$max: "$wind.speed.rate"}}})
3. db.data.aggregate({$group: {\_id: 0, totalWaterDepth: {$sum: "$precipitationEstimatedObservation.estimatedWaterDepth"}}})
4. not found
5. not found
6. db.data.aggregate({$group: {\_id:"$skyCondition",average\_elevation:{$avg:"$elevation"},maximum\_wind\_speed:{$max:"$wind.speed.rate"}}})
7. db.data.aggregate( { $group: { \_id: 0, count: { $sum: 1 }} } )
8. db.data.find({"airTemperature.value": {$gt: 25}, "visibility.distance.value": {$gt: 1000}})
9. db.data.find({"visibility.distance.value": {$gt: 1000}}).count()
10. db.data.find({"precipitationEstimatedObservation.discrepancy": {$eq: "2"}}).count()
11. db.data.find({"skyCondition.ceilingHeight.value": 99999})
12. easy
13. easy
14. db.data.aggregate({$unwind: "$sections"},{$group: {\_id: "$sections"}})