Data Science and Data Base Technologies - Homework 4 - StudentID s332135

Answer the questions giving the query used to obtain what requested and show the result

Question 1

How many stations have (extra.status) “online” status?

```
db.bike_stations.find({"extra.status":"online"}).count()
```

Result

```
33
```

How many stations have “offline” status?

```
db.bike_stations.find({"extra.status":"offline"}).count()
```

Result

```
28
```

Question 2

How many stations have a status different than “online” e “offline”?

```
db.bike_stations.find({"extra.status": {$nin: ["online", "offline"]}}).count()
```

Result

```
4
```

Question 3

For stations that have a status different than “offline” and “online” status, visualize only the value of the status field.

```
db.bike_stations.find({"extra.status": {$nin: ["online", "offline"]}} , {"extra.status": 1})
```

Result

(Showing also the id to distinguish records, otherwise the result is the status attribute repeated 4 times)

```
[
  {
    _id: ObjectId('658573912cb0c0526c6b6a45'),
    extra: { status: 'maintenance' }
  },
  {
    _id: ObjectId('658573912cb0c0526c6b6a46'),
    extra: { status: 'maintenance' }
  },
  {
    _id: ObjectId('658573912cb0c0526c6b6a47'),
    extra: { status: 'maintenance' }
  },
  {
    _id: ObjectId('658573912cb0c0526c6b6a4a'),
    extra: { status: 'maintenance' }
  }
]
```

Question 4

What are the active stations (status = online) with an average rating (extra.score) greater than or equal to 4? Extract the list of the names of these stations, sorted in alphabetical order.

```
db.bike_stations.find(
  {"extra.status": "online",
   "extra.score": {$gte: 4}
   },
  {_id:0, name:1}).sort({name:1})
```

Result

```
[
  { name: '02. Pettiti' },
  { name: '04. Reggia' },
  { name: '06. Municipio' },
  { name: '08. San Marchese' },
  { name: '10. Gallo Praile' },
  { name: 'Belfiore' },
  { name: 'Borromini' },
  { name: 'Castello 1' },
  { name: 'Corte d`Appello' },
  { name: 'Giolitti 1' },
  { name: 'Politecnico 1' },
  { name: 'Politecnico 3' },
  { name: 'Porta Palatina' },
```

```
  { name: 'Principi d`Acaja 1' },
  { name: 'Principi d`Acaja 2' },
  { name: 'San Francesco da Paola' },
  { name: 'Sant´Anselmo' },
  { name: 'Tribunale' }
]
```

Question 5

What is the name of the inactive stations (status = offline) that have at least one free slot (empty_slots> 0) or have at least one bike available (free_bikes> 0)?

```
db.bike_stations.find({
  "extra.status": "offline",
  $or: [
    { empty_slots: { $gt: 0 } },
    { free_bikes: { $gt: 0 } }
  ]
}, { name: 1, _id:0 })
```

Result

```
[{ name: '06. Le Serre' }, { name: '05. Corso Garibaldi' }]
```

How many free slots and how many bikes are available?

```
db.bike_stations.aggregate(
{$match: {
 "extra.status": "offline",
 $or: [{empty_slots: {$gt: 0}},{free_bikes: {$gt: 0}}]}},
{$group: {
 _id: null,
 totalEmptySlots: {$sum: "$empty_slots"},
 totalFreeBikes: {$sum: "$free_bikes"}}},
{$project: {
 _id: 0,
 totalEmptySlots: 1,
 totalFreeBikes: 1
}})
```

Result

```
[{ totalEmptySlots: 1, totalFreeBikes: 5 }]
```

Question 6

What is the total number of reviews (extra.reviews) for all stations?

```
db.bike_stations.aggregate([
  {
    $group: {
      _id: null,
      AmountReviews: { $sum: "$extra.reviews" }
    }
  },
  {
    $project: {
      _id: 0,
      AmountReviews: 1
    }
  }
])
```

Result

```
[{ AmountReviews: 15311 }]
```

Question 7

For each value of average ratings (score), how many stations have that rating? Sort the result by descending rating.

```
db.bike_stations.aggregate([
  {
    $group: {
      _id: "$extra.score",
      numberOfStations: { $sum: 1 }
    }
  },
  {
    $project: {
      _id: 0,
      numberOfStations: 1,
      "extra.score": "$_id"
    }
  },
  {
    $sort: {
      "extra.score": -1
    }
  }
])
```

Result

```
[
  { numberOfStations: 1, extra: { score: 4.7 } },
```

```
  { numberOfStations: 2, extra: { score: 4.5 } },
  { numberOfStations: 2, extra: { score: 4.4 } },
  { numberOfStations: 2, extra: { score: 4.3 } },
  { numberOfStations: 7, extra: { score: 4.2 } },
  { numberOfStations: 5, extra: { score: 4.1 } },
  { numberOfStations: 9, extra: { score: 4 } },
  { numberOfStations: 9, extra: { score: 3.9 } },
  { numberOfStations: 1, extra: { score: 3.8 } },
  { numberOfStations: 2, extra: { score: 3.7 } },
  { numberOfStations: 1, extra: { score: 3.6 } },
  { numberOfStations: 4, extra: { score: 3.5 } },
  { numberOfStations: 3, extra: { score: 3.4 } },
  { numberOfStations: 1, extra: { score: 3.2 } },
  { numberOfStations: 4, extra: { score: 3 } },
  { numberOfStations: 2, extra: { score: 2.8 } },
  { numberOfStations: 1, extra: { score: 2.7 } },
  { numberOfStations: 1, extra: { score: 2.5 } },
  { numberOfStations: 1, extra: { score: 2.4 } },
  { numberOfStations: 1, extra: { score: 2.1 } },
  { numberOfStations: 1, extra: { score: 1.5 } },
  { numberOfStations: 1, extra: { score: 1.4 } },
  { numberOfStations: 1, extra: { score: 1.2 } },
  { numberOfStations: 3, extra: { score: 1 } }
]
```

Question 8

What is the average rating for active (status = online) and inactive (status = offline) stations?
Note: Stations that do not fit into either category (see question 3) will not be considered in the count query.

```
db.bike_stations.aggregate([
  {
    $match: {
      $or: [
        { "extra.status": "offline" },
        { "extra.status": "online" }
      ]
    }
  },
  {
    $group: {
      _id: 0,
      average: { $avg: "$extra.score" }
    }
  },
  {
    $project: {
          _id:0,
          average:1
    }
```

```
  }
])
```

Result

```
[ { average: 3.4704918032786884 } ]
```

Question 9

What are the average ratings for stations without bikes (free_bikes = 0) and for those with at least one bike available (free_bikes> 0)?
Hint: You can use the map-reduce to answer this question. The mapReduce () function was deprecated in MongoDB 5.0. However, the paradigm remains a viable alternative, used, for example, in Hadoop. For this reason, its use is recommended for the resolution of this exercise.

```
db.bike_stations.aggregate([
  {
    $match:
      {
        free_bikes: 0,
      },
  },
  {
    $group:
      {
        _id: null,
        AvgNoBikes: {
          $avg: "$extra.score",
        },
      },
  },
  {
    $unionWith:
      {
        coll: "bike_stations",
        pipeline: [
          {
            $match: {
              free_bikes: {
                $gt: 0,
              },
            },
          },
          {
            $group: {
              _id: null,
              AvgBikesAvailable: {
                $avg: "$extra.score",
              },
            },
```

```
          },
        ],
      },
  },
  {
    $project:
      {
        _id: 0,
      },
  },
])
```

Result

```
[
  { AvgNoBikes: 3.2305555555555556 },
  { AvgBikesAvailable: 3.8758620689655174 }
]
```

The query using the mapReduce function is the following:

```
db.bikes_stations.mapReduce(
        function(){
                emit(this.free_bikes==0? 'AvgNoBikes': 'AvgBikesAvailable',
this.extra.score)
        },
        function(key, values){
                var s = sum(values);
                var count = len(values);
                return s/count;
        }
)
```

Question 10

Answer question 9, referring only to active stations (status = online).
Hint: Also for this exercise, the use of the map-reduce paradigm is recommended.

```
db.bike_stations.aggregate([
  {
    $match:
      {
        free_bikes: 0,
        "extra.status": "online",
      },
  },
  {
    $group:
      {
        _id: null,
```

```
        AvgNoBikes: {
          $avg: "$extra.score",
        },
      },
  },
  {
    $unionWith:
      {
        coll: "bike_stations",
        pipeline: [
          {
            $match: {
              free_bikes: {
                $gt: 0,
              },
              "extra.status": "online",
            },
          },
          {
            $group: {
              _id: null,
              AvgBikesAvailable: {
                $avg: "$extra.score",
              },
            },
          },
        ],
      },
  },
  {
    $project:
      {
        _id: 0,
      },
  },
])
```

Result

```
[
  { AvgNoBikes: 3.7399999999999998 },
  { AvgBikesAvailable: 3.8642857142857143 }
]
```

The query using the mapReduce function is the following:

```
db.bike_stations.mapReduce(
        function(){
                if(this.extra.status == "online")
                  emit(this.free_bikes==0? 'AvgNoBikes': 'AvgBikesAvailable',
this.extra.score);
```

```
        },
        function(key, values){
                var s = sum(values);
                var count = len(values);
                return s/count;
        }
)
```

Question 11

What are the names of the 3 stations with available bikes (free_bikes> 0) closest to the point [45.07456, 7.69463]? How many bikes are available?
Note: You need to create a 2dsphere index on "location" to use the $near operator.
Note : You can use the limit(n) method to limit the number of results extracted.

```
db.bike_stations.find(
  {
    location: {
      $near: {
        $geometry: { type: "Point", coordinates: [45.07456, 7.69463] }
      }
    },
    free_bikes: { $gt: 0 }
  },
  { free_bikes:1, name: 1,  _id: 0 }
).limit(3)
```

Result

```
[
  { free_bikes: 5, name: 'Palermo 2' },
  { free_bikes: 5, name: 'Castello 1' },
  { free_bikes: 4, name: 'San Francesco da Paola' }
]
```

Question 12

What are the names of the 3 stations with available bikes (free_bikes> 0) closest to the "Politecnico 4" station? How many bikes are available?
Note: You need to create a 2dsphere index on "location" to use the $near operator.
Requirement: Solve the exercise using a nested query to extract the position of the "Politecnico 4" station.

```
db.bike_stations.find(
  {
    location: {
      $near: {
        $geometry: db.bike_stations.findOne(
          { name: "Politecnico 4" },
```

```
          { location: 1, _id: 0 }
        ).location
      }
    },
    free_bikes: { $gt: 0 }
  },
  { free_bikes: 1, name: 1, _id: 0 }
).limit(3);
```

Result

```
[
  { free_bikes: 9, name: 'Politecnico 1' },
  { free_bikes: 5, name: 'Politecnico 3' },
  { free_bikes: 3, name: 'Tribunale' }
]
```