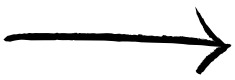
DataScience and Database Technology

Homework #4 - Revalor Riccardo - s339423

Query 1: “How many stations have (extra.status) “online”status. How many stations have “offline” status?”

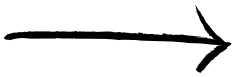
```
db["BIKES"].aggregate([
  {$match: {
$or: [{"extra.status": "online"},
      {"extra.status": "offline"}]}
  },
  {$group: {
_id: "extra.status",
count: {$sum: 1}}}]);
```



```
{
  _id: 'offline',
  count: 28
}
{
  _id: 'online',
  count: 33
}
```

Query 2: “How many stations have a status different than “online” e “offline”?”

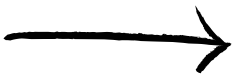
```
db["BIKES"].find(
  {"extra.status": {$nin: ["offline", "online"]}}).count();
```



```
4
```

Query 3: “For stations that have a status different than “offline” and “online” status, visualize only the value of the status field.”

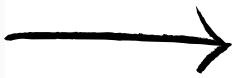
```
db["BIKES"].find(
  {"extra.status": {$nin: ["offline", "online"]}},
  {_id: 0,
        "extra.status": 1});
```



```
{
  extra: {
    status: 'maintenance'
  }
}
{
  extra: {
    status: 'maintenance'
  }
}
{
  extra: {
    status: 'maintenance'
  }
}
{
  extra: {
    status: 'maintenance'
  }
}
```

Query 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”

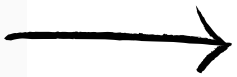
```
db["BIKES"].find(
 {
 "extra.status": "online",
 "extra.score": {
 $gte: 4
 }
 },
 { _id:0,
 "name": 1 /
 }
).sort(
 {
 "name": 1
 }
).toArray();
```



```
[
  { 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' }
]
```

Query 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)? How many free slots and how many bikes are available?”

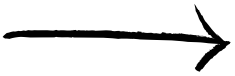
```
db["BIKES"].aggregate([
  { //Matching step
$match: {"extra.status": "offline",
$or: [
  {"empty_slots": {$gt: 0}},
  {"free_bikes": {$gt: 0}}]}
},
       //Grouping step
  {$group: {
_id: null, //group into one big group
names_list: {$push: "$name"}, //put all the names in an array
free_slots_counter: {$sum:  "$empty_slots" }, //sum all the free slots
bikes_counter: {$sum: "$free_bikes"}}
}
]);
```



```
  _id: null,
  names_list: [
    '06. Le Serre',
    '05. Corso Garibaldi'
  ],
  free_slots_counter: 1,
  bikes_counter: 5
}
```

Query 6: “What is the total number of reviews (extra.reviews) for all stations?”

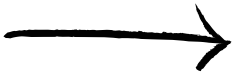
```
db["BIKES"].aggregate([
  {$group:
  { //group all stations and sum all the reviews
_id: null,
reviews_all_stations: {$sum: "$extra.reviews" }
}}]);
```



```
{
  _id: null,
  reviews_all_stations: 15311
}
```

Query 7: “For each value of average ratings (score), how many stations have that rating? Sort the result by descending rating.”

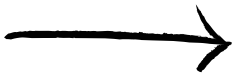
```
db["BIKES"].aggregate([
  {$group: {
      //Group by score
     _id: "$extra.score",
     rating_count: {$sum: 1}}
},
  {
       //sort in descending order
       $sort: {_id: -1}
}]).toArray();
```



```
[
  { _id: 4.7, rating_count: 1 },
  { _id: 4.5, rating_count: 2 },
  { _id: 4.4, rating_count: 2 },
  { _id: 4.3, rating_count: 2 },
  { _id: 4.2, rating_count: 7 },
  { _id: 4.1, rating_count: 5 },
  { _id: 4, rating_count: 9 },
  { _id: 3.9, rating_count: 9 },
  { _id: 3.8, rating_count: 1 },
  { _id: 3.7, rating_count: 2 },
  { _id: 3.6, rating_count: 1 },
  { _id: 3.5, rating_count: 4 },
  { _id: 3.4, rating_count: 3 },
  { _id: 3.2, rating_count: 1 },
  { _id: 3, rating_count: 4 },
  { _id: 2.8, rating_count: 2 },
  { _id: 2.7, rating_count: 1 },
  { _id: 2.5, rating_count: 1 },
  { _id: 2.4, rating_count: 1 },
  { _id: 2.1, rating_count: 1 },
  { _id: 1.5, rating_count: 1 },
  { _id: 1.4, rating_count: 1 },
  { _id: 1.2, rating_count: 1 },
  { _id: 1, rating_count: 3 }
]
```

Query 8: “What is the average rating for active (status = online) and inactive (status = offline) stations?”

```
db["BIKES"].aggregate([
  {
       //filter stations
       $match: {
              $or: [{ "extra.status": "online" },
        {"extra.status": "offline"}]
       }},
  {
       //Group by extra.status and use $avg
       $group: {
              _id: "$extra.status",
              avg_score: {$avg: "$extra.score"}
}}
]).toArray();
```



```
[
  { _id: 'offline', avg_score: 3.0285714285714285 },
  { _id: 'online', avg_score: 3.8454545454545452 }
]
```

Query 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)? ”

```
db["BIKES"].aggregate([
  {
$project: {
        //assign 'no_bikes' if no bikes, otherwise assign 'yes_bikes' (like the map phase in mapred)
        new_id: {$cond: { if: { $eq: ["$free_bikes", 0] }, then: "no_bikes", else: "yes_bikes" } },
        "extra.score": 1
}
},
  {
   $group: {
                //group by new_id and use $avg
                _id: "$new_id",
                avg_score: {$avg: "$extra.score"}
}
}]).toArray();
```

```
[
  { _id: 'yes_bikes', avg_score: 3.8758620689655174 },
  { _id: 'no_bikes', avg_score: 3.2305555555555556 }
]
```

Query 9 - using mapReduce and JS

```
var mapper = function() {
var new_id = (this.free_bikes === 0) ? "no_bikes" : "yes_bikes";
//emit (new_id, (1, extra.score))
//count of 1 will be used to manually compute avg
emit(new_id, {count: 1, score: this.extra.score})
};

var reducer = function(key, values) {
//compute avg by doing sum of score / sum of n
var reduced_value = {count: 0, total_score: 0};
values.forEach(function (value) {
    reduced_value.count += value.count; // 1+1+1+1+1+1.....
    reduced_value.totalScore += value.totalScore; // Sum the score
});

if (reduced_value.count > 0) {
        //perform avg
        return reduced_value.total_score / reduced_value.count;
}

return 0;

};

//Apply mapreduce job
db["BIKES"].mapReduce(
mapper,
reducer,
  {
        out: {inline: 1}
}
).toArray();
```

Query 10: “Answer question 9, referring only to active stations (status = online). ”
I just report the aggregate version of the solution, as it’s more concise and more optimized.

```
db["BIKES"].aggregate([
  {
    $match: {
      "extra.status": "online" // Filter online stations
    }
  },
  {
    $project: {
      new_id: { $cond: { if: { $eq: ["$free_bikes", 0] }, then: "no_bikes", else: "yes_bikes" } },
      "extra.score": 1,
      "extra.status": 1
    }
  },
  {
    $group: {
      _id: "$new_id",
      avg_score: { $avg: "$extra.score" },
      statuses: { $addToSet: "$extra.status" } // Just for debugging
    }
  }
]);
```

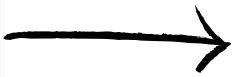
```
{
  _id: 'yes_bikes',
  avg_score: 3.8642857142857143,
  statuses: [
    'online'
  ]
}
{
  _id: 'no_bikes',
  avg_score: 3.7399999999999998,
  statuses: [
    'online'
  ]
}
```

Query 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?”

2dspere index creation:

```
db["BIKES"].createIndex( {location: "2dsphere"});
```

```
db["BIKES"].find(
  {
    free_bikes: { $gt: 0 }, // Filter for stations with more than 0 free bikes
    location: {
      $near: { // Find stations near the specified point
        $geometry: {
          type: "Point",
          coordinates: [45.054267, 7.69463] // Specify the point: [longitude, latitude]
        }
      }
    }
  },
  {
    _id: 0, // Exclude the _id field from the result
    name: 1, // Include the 'name' field
    free_bikes: 1, // Include the 'free_bikes' field
    "location.coordinates": 1 // Include the 'location.coordinates' field
  }
).limit(3); // Limit the result to the first 3 nearest stations
```



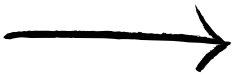
```
{
  free_bikes: 4,
  name: 'San Francesco da Paola',
  location: {
    coordinates: [
      45.068617,
      7.689097
    ]
  }
}
{
  free_bikes: 9,
  name: 'Carlo Alberto',
  location: {
    coordinates: [
      45.06806,
      7.686688
    ]
  }
}
{
  free_bikes: 3,
  name: 'Belfiore',
  location: {
    coordinates: [
      45.054267,
      7.678409
    ]
  }
}
```

Query 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?”

2dspere index creation:

```
db["BIKES"].createIndex( {location: "2dsphere"});
```

```
db["BIKES"].find(
  {
    free_bikes: { $gt: 0 }, // Filter for stations with more than 0 free bikes
    location: {
      $near: { // Find stations near the specified point
        $geometry: {
          type: "Point",
        //Nested Query for finding Politecnico 4 coordinaates
          coordinates: db["BIKES"].findOne(
            {
                "name": "Politecnico 4"
            },
            {
                _id:0,
                "location.coordinates": 1 //extract just the coordinates field
        }).location.coordinates
        }
      }
    }
  },
  {
    _id: 0, // Exclude the _id field from the result
    name: 1, // Include the 'name' field
    free_bikes: 1, // Include the 'free_bikes' field
    "location.coordinates": 1 // Include the 'location.coordinates' field
  }
).limit(3); // Limit the result to the first 3 nearest stations
```



```
{
  free_bikes: 9,
  name: 'Politecnico 1',
  location: {
    coordinates: [
      45.062387,
      7.662494
    ]
  }
}
{
  free_bikes: 5,
  name: 'Politecnico 3',
  location: {
    coordinates: [
      45.060759,
      7.661325
    ]
  }
}
{
  free_bikes: 3,
  name: 'Tribunale',
  location: {
    coordinates: [
      45.07084597193492,
      7.6612794399261475
    ]
  }
}
```