Data Base and Data Mining Group of Politecnico di Torino

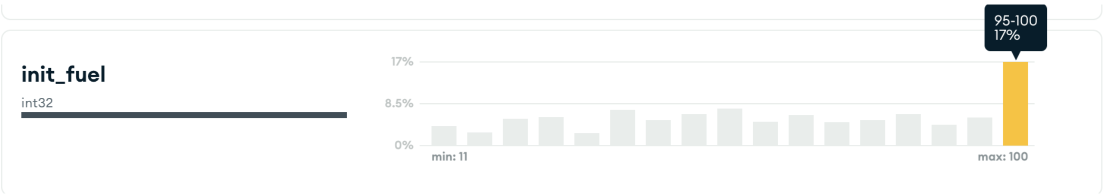
Data Management and Visualization

Politecnico di Torino

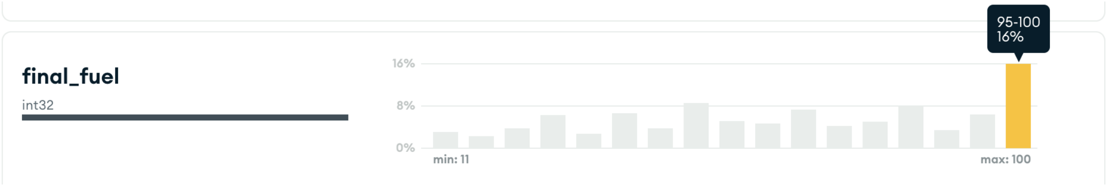
NoSQL in MongoDB Compass – Practice 4

1. Analyze the database using the Schema Analyzer

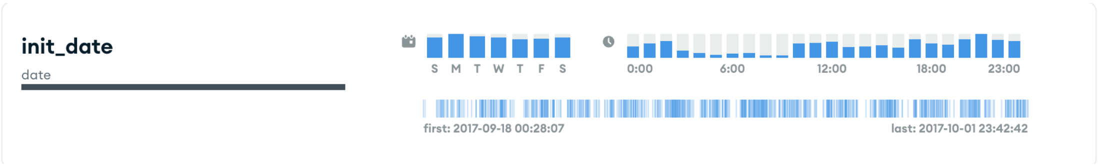
1. (Bookings) Identify the most common percentage(s) of fuel level at the beginning of the renting period.



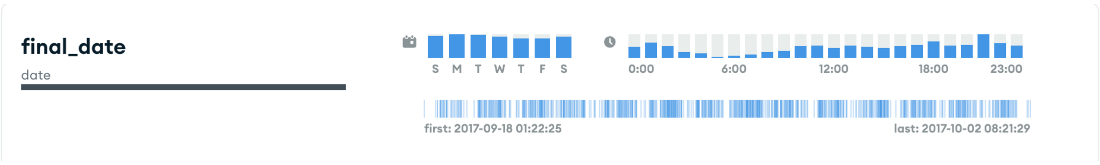
2. (Bookings) Identify the most common percentage(s) of fuel level at the end of the renting period.



3. (Parkings) Identify the time range(s) with most parking requests (start parking).

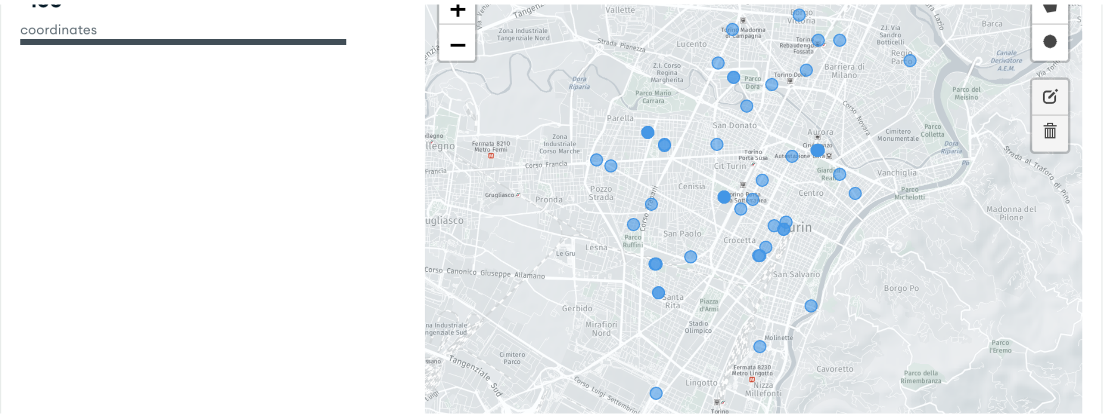


4. (Parkings) Identify the time range(s) with most booking requests (end parking).



5. (Parkings) Visualize on the map the vehicles having the fuel level lower than 5%.

{fuel: {$lt: 5}}

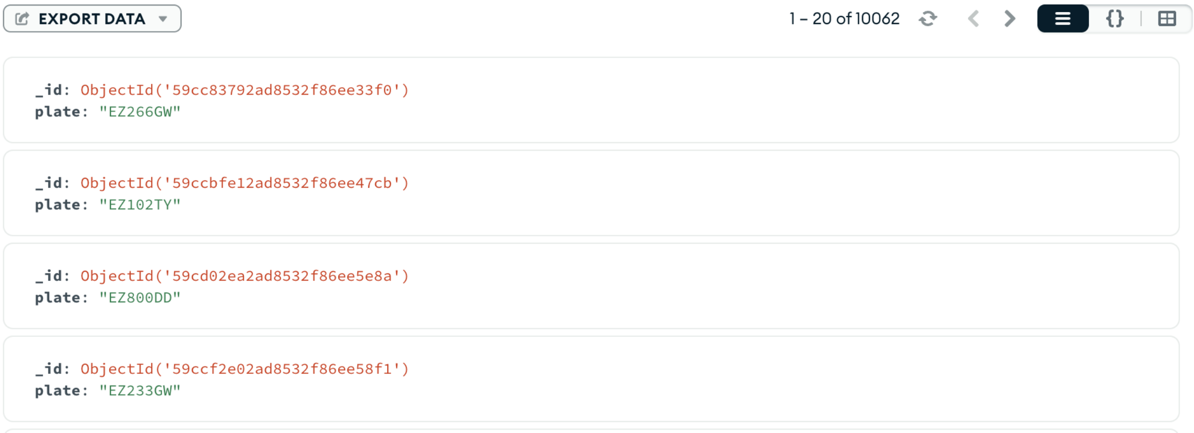


2. Querying the database

1. (Parkings) Find the plates and the parking addresses of the vehicles that begin the booking (end parking) after 2017-09-30 at 6AM

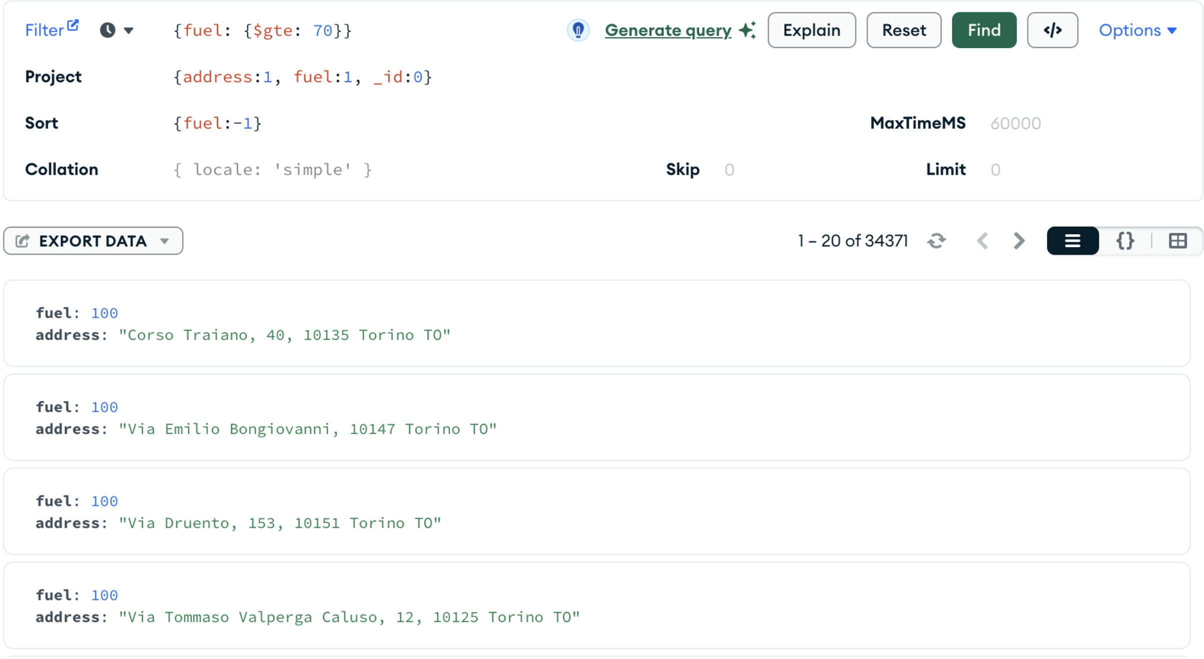
(Hint: it is possible to use the function: new Date("<YYYY-mm-ddTHH:MM:ss>"))

{final_date: {$gte: new Date('2017-09-30T06:00:00.000Z')}}



```
_id: ObjectId('59cc83792ad8532f86ee33f0')
plate: "EZ266GW"

_id: ObjectId('59ccbfe12ad8532f86ee47cb')
plate: "EZ102TY"

_id: ObjectId('59cd02ea2ad8532f86ee5e8a')
plate: "EZ800DD"

_id: ObjectId('59ccf2e02ad8532f86ee58f1')
plate: "EZ233GW"
```

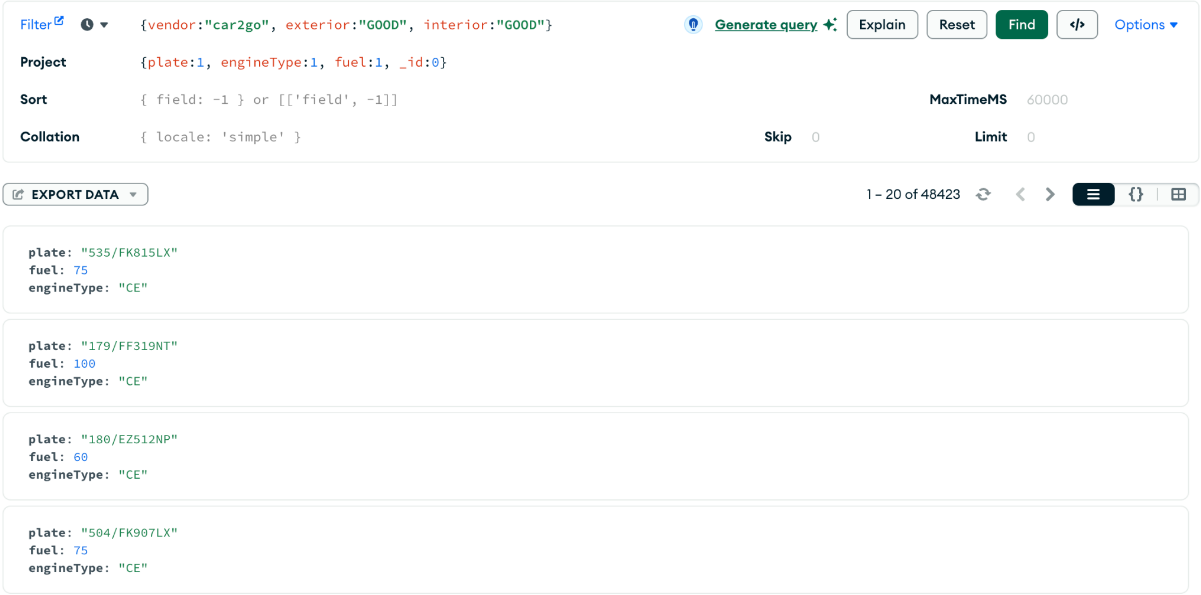
2. (Parkings) Find the addresses and the level of fuel of the vehicles that during the parking period had at least 70% of fuel level. Order the results according to descending value of fuel level.



```
Filter      {fuel: {$gte: 70}}
Project     {address:1, fuel:1, _id:0}
Sort        {fuel:-1}                                   MaxTimeMS  60000
Collation   { locale: 'simple' }          Skip  0       Limit  0
```

```
fuel: 100
address: "Corso Traiano, 40, 10135 Torino TO"

fuel: 100
address: "Via Emilio Bongiovanni, 10147 Torino TO"

fuel: 100
address: "Via Druento, 153, 10151 Torino TO"

fuel: 100
address: "Via Tommaso Valperga Caluso, 12, 10125 Torino TO"
```

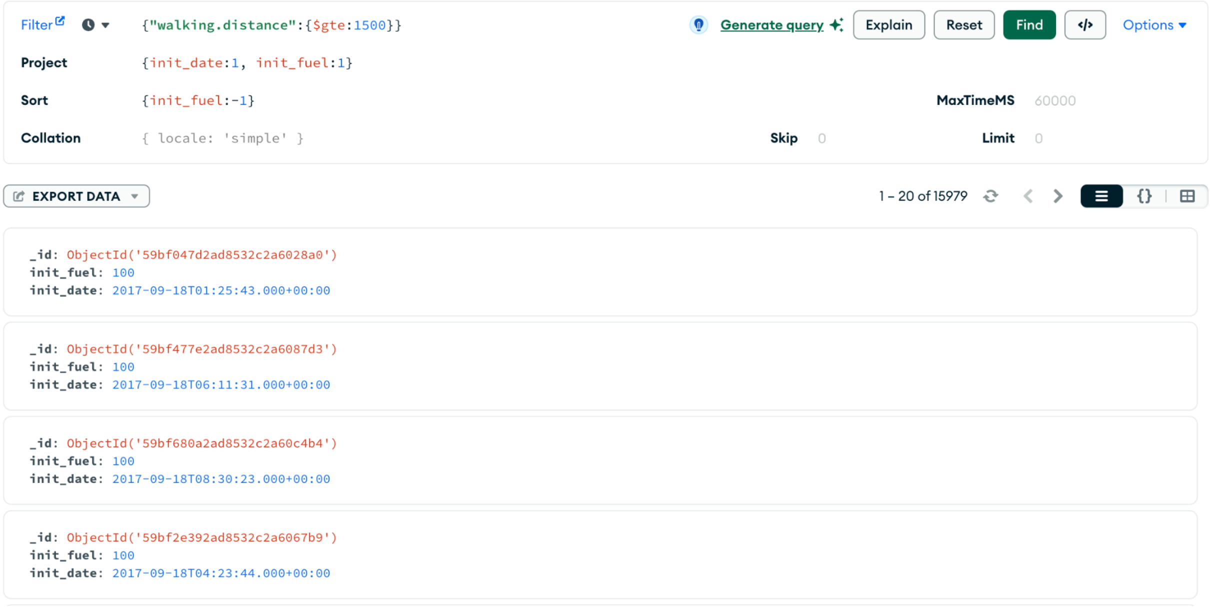
3. (Parkings) Find the plate, the engine type and fuel level for 'car2go' vehicles (vendor) with good internal and external conditions.



```
Filter      {vendor:"car2go", exterior:"GOOD", interior:"GOOD"}
Project     {plate:1, engineType:1, fuel:1, _id:0}
Sort        { field: -1 } or [['field', -1]]            MaxTimeMS  60000
Collation   { locale: 'simple' }          Skip  0       Limit  0
```

```
plate: "535/FK815LX"
fuel: 75
engineType: "CE"

plate: "179/FF319NT"
fuel: 100
engineType: "CE"

plate: "180/EZ512NP"
fuel: 60
engineType: "CE"

plate: "504/FK907LX"
fuel: 75
engineType: "CE"
```

4. (Bookings) For the renting that required a walking distance greater than 15 Km (to reach the vehicle), find the hour and the fuel level at the beginning of the renting period. Order results according to decreasing initial fuel level.



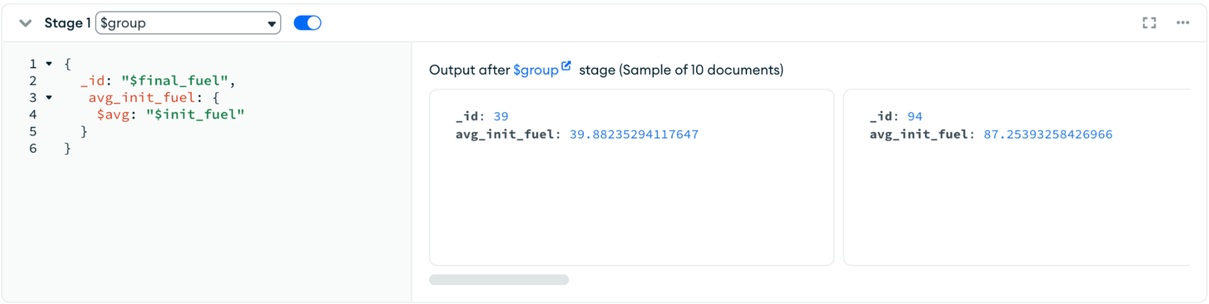
```
Filter      {"walking.distance":{$gte:1500}}
Project     {init_date:1, init_fuel:1}
Sort        {init_fuel:-1}                                MaxTimeMS  60000
Collation   { locale: 'simple' }          Skip  0         Limit      0
```

1 – 20 of 15979

```
_id: ObjectId('59bf047d2ad8532c2a6028a0')
init_fuel: 100
init_date: 2017-09-18T01:25:43.000+00:00

_id: ObjectId('59bf477e2ad8532c2a6087d3')
init_fuel: 100
init_date: 2017-09-18T06:11:31.000+00:00

_id: ObjectId('59bf680a2ad8532c2a60c4b4')
init_fuel: 100
init_date: 2017-09-18T08:30:23.000+00:00

_id: ObjectId('59bf2e392ad8532c2a6067b9')
init_fuel: 100
init_date: 2017-09-18T04:23:44.000+00:00
```

5. (Bookings) Group documents according to their fuel level at the end of the renting. For each group, select the average fuel level at the beginning of the renting period.

```
$group = {
  _id: "$final_fuel",
   avg_init_fuel: {
    $avg: "$init_fuel"
  }
}
```


Stage 1 $group

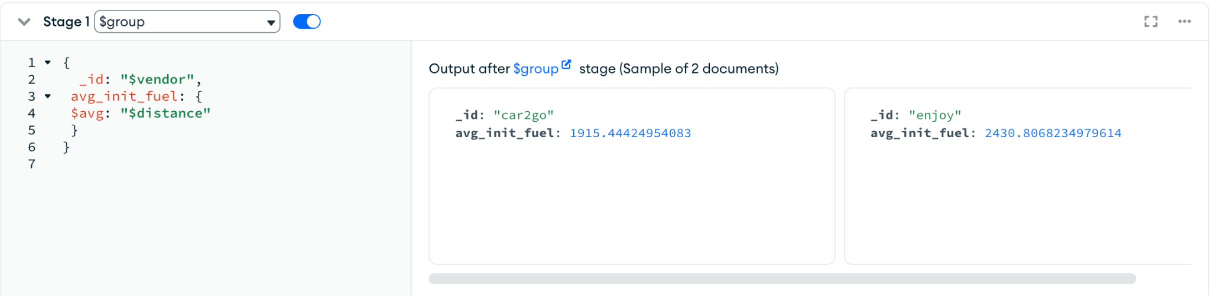
```
{
   _id: "$final_fuel",
    avg_init_fuel: {
     $avg: "$init_fuel"
   }
 }
```

Output after $group stage (Sample of 10 documents)

```
_id: 39
avg_init_fuel: 39.88235294117647

_id: 94
avg_init_fuel: 87.25393258426966
```

6. (Bookings) Select the average driving distance for each vendor. On average, for which vendor the users cover longer distances?

```
$group = {
  _id: "$vendor",
 avg_init_fuel: {
 $avg: "$distance"
 }
}
```



```
{
  _id: "$vendor",
  avg_init_fuel: {
  $avg: "$distance"
  }
}
```

Output after $group stage (Sample of 2 documents)

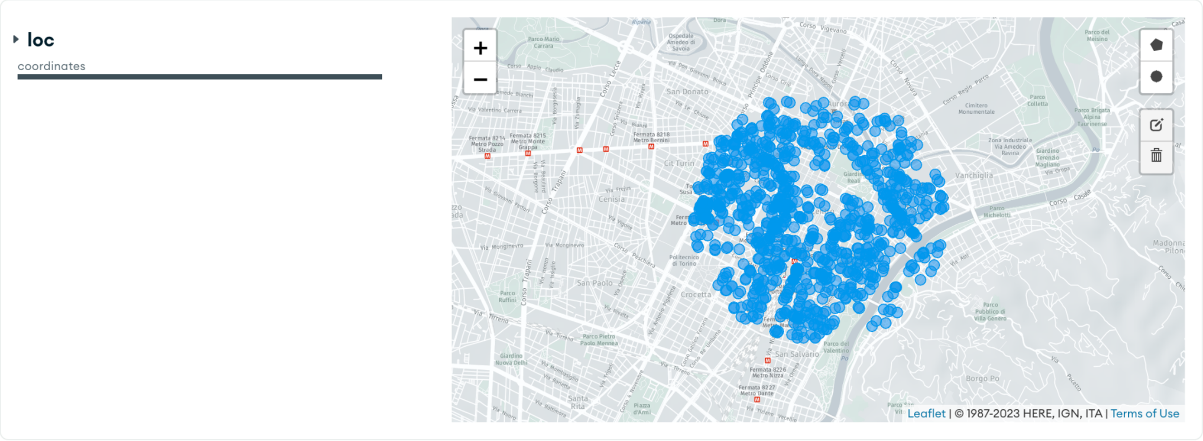
```
_id: "car2go"
avg_init_fuel: 1915.44424954083
```

```
_id: "enjoy"
avg_init_fuel: 2430.8068234979614
```

7. (Parkings) Find the vehicles parked less than a kilometre far from Piazza San Carlo (coordinates: 7.683016, 45.067764).

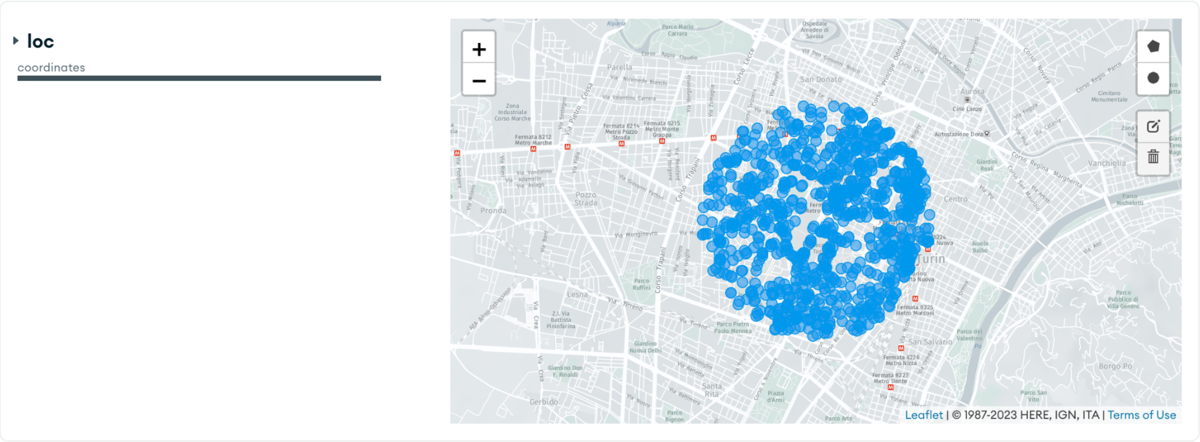
Hint: use the operator $geoWithin in conjunction with $centerSphere.

Filter: {loc : { $geoWithin : { $centerSphere : [[7.683016, 45.067764], 1/3963.2] }}}



8. (Parkings) Repeat the query at the previous step using the coordinates of a place of personal interest in Turin (e.g. Politecnico di Torino) using Open Street Maps to find the exact coordinates (www.openstreetmap.org, inverse the coordinates order As in the previous step changing the coordinates (check inversion from OpenStreetMap).

{loc: {$geoWithin: { $centerSphere: [[7.661035358905793, 45.065990536481806], 0.00023435588986155505]}}}


loc
coordinates
+
−
Leaflet | © 1987-2023 HERE, IGN, ITA | Terms of Use