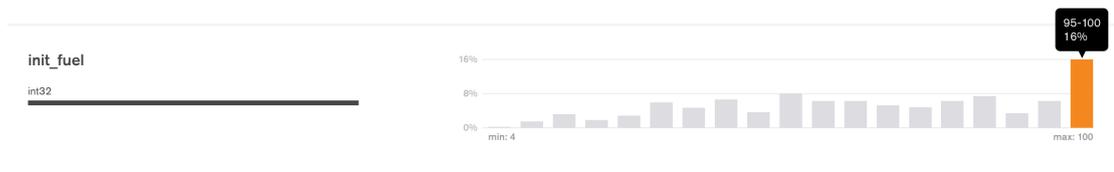


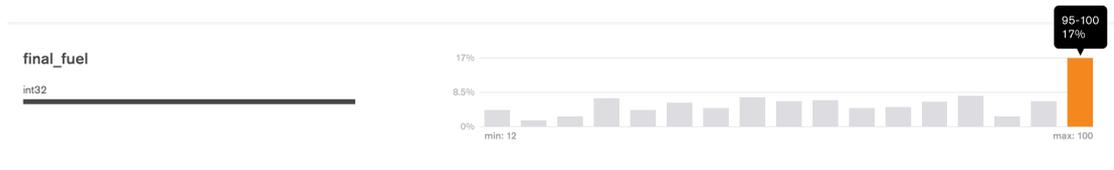
NoSQL in MongoDB Compass – Practice 4

1. Analyze the database using the *Schema Analyzer*

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



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



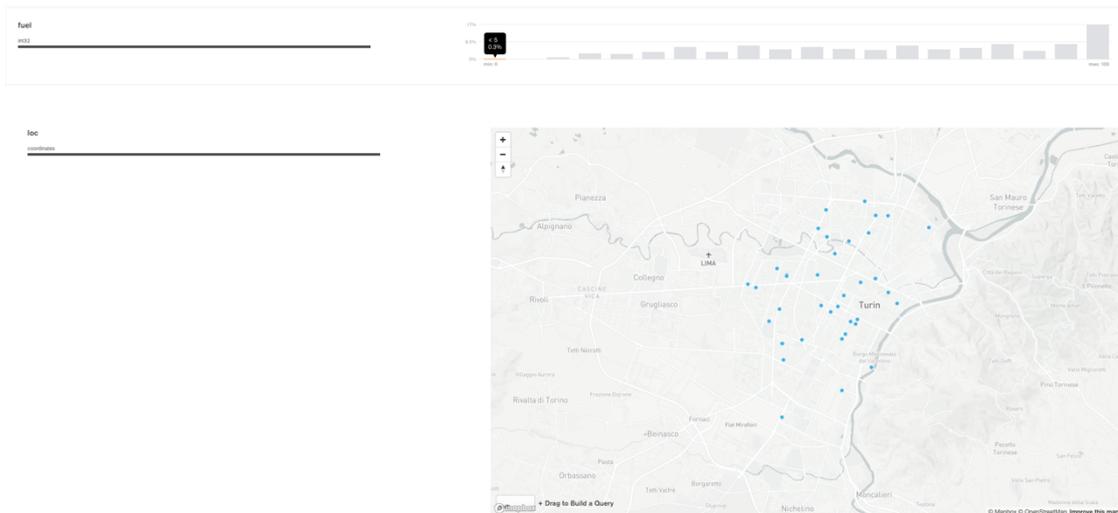
- (Parkings) Identify the time range(s) with most parking requests (start parking).



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



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



2. Querying the database

- (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 Date("<YYYY-mm-ddTHH:MM:ss"))

Query interface showing the following configuration:

- FILTER:** `{final_date: {$gte : Date("2017-09-30T06:00:00Z")}}`
- PROJECT:** `{address: 1, plate: 1, _id: 0}`
- SKIP:** 0
- LIMIT:** 0

View: LIST | TABLE | Displaying documents 1 - 20 of 10062

plate: "429/FK630LX"	address: "Via Amalia Guglielminetti, 24, 10136 Torino TO"
plate: "500/FK903LX"	address: "Via Candiolo, 54, 10127 Torino TO"
plate: "321/FF296SJ"	address: "Via Zino Zini, 139, 10134 Torino TO"
plate: "340/FF144VY"	address: "Piazzetta Operaie della Manifattura Tabacchi, 10154 Torino TO"

- (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.

Query interface showing the following configuration:

- FILTER:** `{fuel: {$gte : 70}}`
- PROJECT:** `{address: 1, fuel: 1, _id: 0}`
- SORT:** `{fuel: -1}`
- SKIP:** 0
- LIMIT:** 0

fuel: 100 address: "Via Don Bartolomeo Grazioli, 26, 10137 Torino TO"
fuel: 100 address: "Via Antonio Rosmini, 6, 10126 Torino TO"
fuel: 100 address: "Via Bistagno, 5, 10136 Torino TO"
fuel: 100 address: "Piazza Gian Lorenzo Bernini, 20, 10143 Torino TO"

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

{vendor : "car2go", exterior: "GOOD", interior: "GOOD"} ▼ OPTIONS
 {plate: 1, engineType: 1, fuel:1, _id: 0}

 0 0

1	"262/FF153SJ"	62	"CE"
2	"244/FF119SJ"	50	"CE"
3	"256/FF144SJ"	100	"CE"
4	"516/FK993MC"	84	"CE"
5	"155/FF180NT"	75	"CE"
6	"009/FF757KW"	21	"CE"

- (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.

{"walking.distance" : {\$gte : 15000} }
 {init_date: 1, init_fuel: 1}
 {init_fuel: -1}

- (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"
  }
}

```

```

1 //**
2 * _id - The id of the group.
3 * field1 - The first field name.
4 */
5 {
6   _id: "$final_fuel",
7   avg_init_fuel: {
8     $avg: "$init_fuel"
9   }
10 }

```

Output after \$group stage (Sample of 20 documents)

```

_id: 39
avg_init_fuel: 39.88235294117647

_id: 59
avg_init_fuel: 60.04111405835544

```

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"
  }
}

```

```

1 //**
2 * _id - The id of the group.
3 * field1 - The first field name.
4 */
5 {
6   _id: "$vendor",
7   avg_distance: {
8     $avg: "$distance"
9   }
10 }

```

Output after \$group stage (Sample of 2 documents)

```

_id: "car2go"
avg_distance: 1915.44424954083

_id: "enjoy"
avg_distance: 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 ] }}}

```

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

Query returned 21834 documents. This report is based on a sample of 1000 documents (4.58%).

loc

coordinates

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).