

Basi di Dati

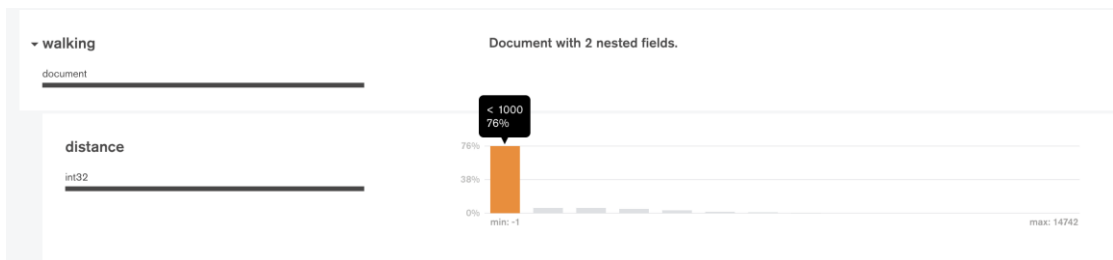
MongoDB Compass - Esercitazione n. 5 - Soluzioni

1. Analizzare la base dati con lo *schema analyzer*

1. (Parkings) Identificare l'intervallo/gli intervalli orari nei quali i veicoli **vengono noleggiati (fine stazionamento)** più di frequente.



2. (Bookings) Identificare il range di **distanza percorsa a piedi** più frequente per raggiungere il veicolo.



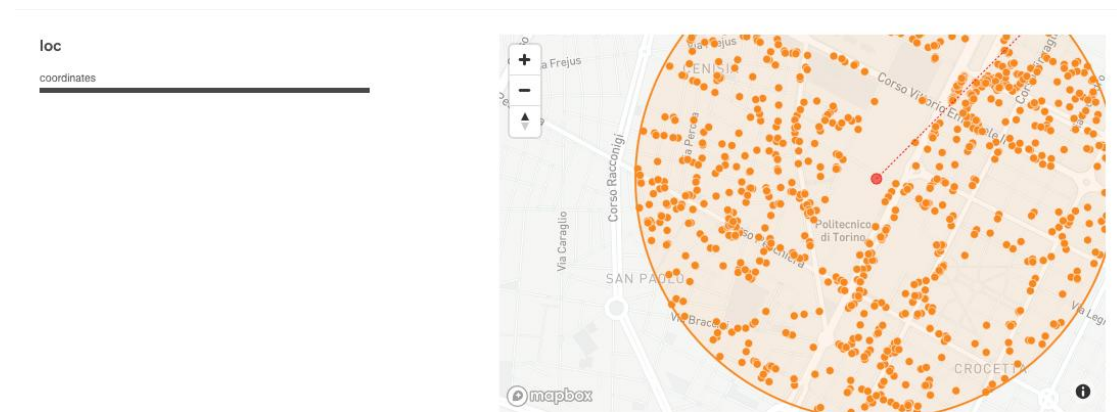
3. (Bookings) Identificare la/le percentuali più frequenti di **livello di carburante a inizio noleggio (fine stazionamento)**.



4. (Bookings) Identificare la/le percentuali più frequenti di **livello di carburante a fine noleggio (inizio stazionamento)**.



- (Parkings) Filtrare sulla mappa una zona di interesse e analizzare l'intervallo/gli intervalli orari di **inizio noleggio (fine stazionamento)** più frequenti.



Analizzare l'intervallo/gli intervalli orari di inizio noleggio (fine stazionamento) più frequenti come fatto nell'esercizio 1.

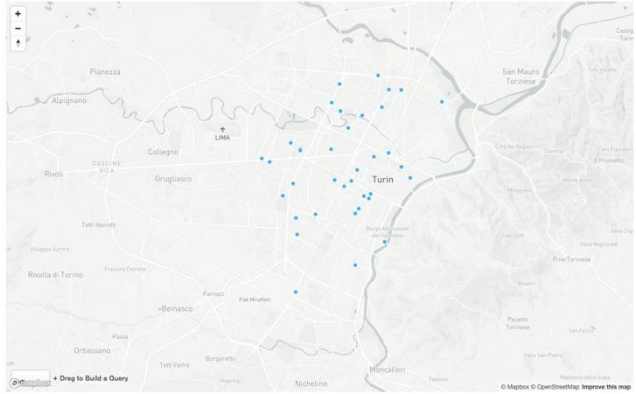
- (Parkings) Per i veicoli filtrati al passo precedente, visualizzare solo quelli che hanno un livello di carburante residuo maggiore del 90%.



- (Parkings) Visualizzare sulla mappa i veicoli che hanno un livello di carburante residuo inferiore al 5%.



loc



2. Interrogare la base dati – *Documents e Aggregation Tab*

1. (Parkings) Trovare le targhe e gli indirizzi di parcheggio dei veicoli che hanno iniziato il noleggio (finito stazionamento) dopo le 06:00 del 30-09-2017.
(Hint: usare la funzione [ISODate\("YYYY-mm-ddTHH:MM:ss"\)](#))

FILTER `{final_date: {$gte : Date("2017-09-30T06:00:00Z")}}` **OPTIONS**

PROJECT `{address: 1, plate: 1, _id: 0}`

SORT

COLLATION **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/FF2965J" address: "Via Zino Zini, 139, 10134 Torino TO"

2. (Parkings) Trovare gli indirizzi e il livello di carburante residuo per le auto che hanno avuto durante lo stazionamento almeno il 70% di carburante residuo e ordinare i risultati in base al loro livello di carburante decrescente.

FILTER `{fuel: {$gte : 70}}` **OPTIONS**

PROJECT `{address: 1, fuel: 1, _id: 0}`

SORT `{fuel: -1}`

COLLATION **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"

- (Parkings) Trovare la targa, tipo di motore e livello di carburante dei veicoli di 'car2go' che hanno buone condizioni interne ed esterne.

id	plate	engineType	fuel
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) Per i noleggi che hanno richiesto un percorso a piedi maggiore di 15 km per raggiungere il veicolo, visualizzare la data e l'orario di inizio noleggio e il livello di carburante a inizio noleggio. Visualizzare i risultati ordinati in base al livello di carburante iniziale decrescente.

```

FILTER: {'walking.distance': {'$gte': 15000}}
PROJECT: {init_date: 1, init_fuel: 1, _id: 0}
SORT: {init_fuel: -1}
COLLATION:
  
```

Query returned 131 documents. This report is based on a sample of 131 documents (100.00%).

Utilizzare il tab “Aggregations” per scrivere le operazioni di aggregazione (e.g., \$group, \$count, \$sort, etc.). Se necessario, aggiungere stages multipli per ottenere il risultato finale.

- (Bookings) Raggruppare i documenti in base al loro **livello di carburante a fine noleggio**. Per ogni gruppo visualizzare il **livello di carburante medio a inizio noleggio**.

```

$group = {_id: "$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	avg_init_fuel
39	39.88235294117647
59	60.04111405835544

- (Bookings) Visualizzare la **distanza media** percorsa nei noleggi per ciascun **fornitore del servizio**. In media con quale fornitore del servizio gli utenti percorrono una distanza maggiore?

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

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

_id	avg_distance
"car2go"	1915.44424954083
"enjoy"	2430.8068234979614

Interrogazioni Bonus

- (Parkings) Trovare le macchine parcheggiate a meno di 1 km da Piazza San Carlo (coordinate 7.683016, 45.067764).

(Hint: usare l'operatore [\\$geoWithin](#) insieme a [\\$centerSphere](#)

Il raggio della Terra espresso in km è pari a 6378.1 km)

```
{loc: {$geoWithin: {$centerSphere: [[7.683016, 45.067764], 1/6378.1]}}}
```

Query returned **9749** documents. This report is based on a sample of **1000** documents (10.26%).

loc

coordinates

- (Parkings) Ripetere l'interrogazione al passo precedente con un punto di interesse personale nell'area metropolitana di Torino (e.g., indirizzo del Politecnico) usando Open Street Map per trovare le coordinate esatte (www.openstreetmap.org, invertire l'ordine delle coordinate).

Come la query precedente cambiando le coordinate con quelle di interesse.