

Parte 1 - Compass

2) Analizzare il database con il tool "Schema analyzer"

- 1. (Bookings) Identificare i valori più comuni relativi al livello di carburante all'inizio del noleggio.



- 2. (Bookings) Identificare i valori più comuni del livello di carburante alla fine del noleggio.



- 3. (Parkings) Identificare gli intervalli di tempo con più richieste di parcheggio (init_date di Parkings).

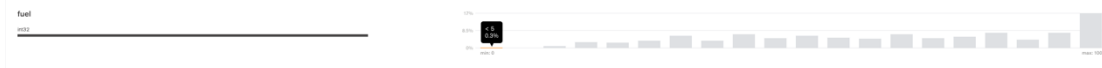


Maggior numero di richieste il mercoledì e nelle fasce orarie indicate in figura.

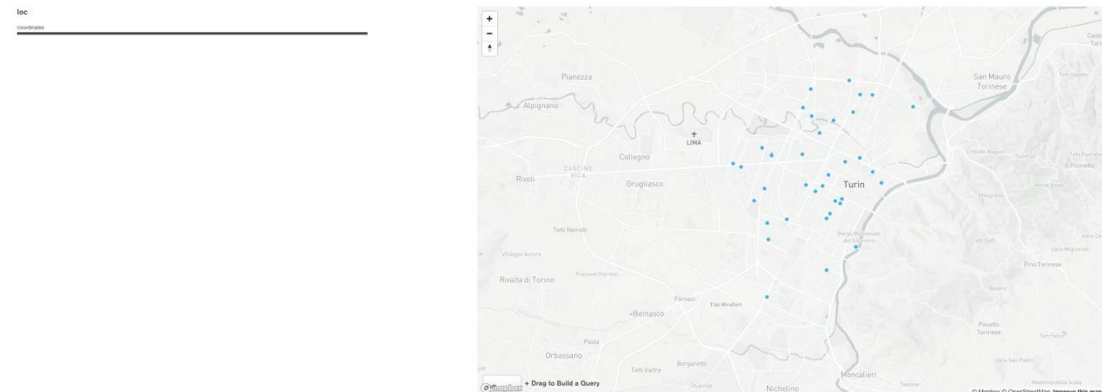
- 4. (Parkings) Identificare gli intervalli di tempo con più richieste di noleggio (final_date di Parkings).



- 5. (Parkings) Visualizzare sulla mappa i veicoli con livello di carburante minore del 5%. Filtrare i record selezionando sull'istogramma (o utilizzando la funzione filter in cima alla finestra di Compass):



Analizzare i risultati sull'attributo loc (location).



Eseguire query sul database

1. (Parkings) Trovare le targhe (plate) e gli indirizzi di parcheggio dei veicoli che cominciano il noleggio (final date) dopo il 30-09-2017 alle 6 di mattina. (Consiglio: è possibile usare la funzione `Date(" <YYYY-mm-ddTHH:MM:ssZ > ")`, dove Z indica il fuso orario UTC)

The screenshot shows a MongoDB query interface with the following configuration:

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

The results are displayed in a table view, showing 4 documents:

plate	address
"429/FK630LX"	"Via Amalia Guglielminetti, 24, 10136 Torino T0"
"500/FK903LX"	"Via Candiolo, 54, 10127 Torino T0"
"321/FF2965J"	"Via Zino Zini, 139, 10134 Torino T0"
"340/FF144VY"	"Piazzetta Operaie della Manifattura Tabacchi, 10154 Torino T0"

2. (Parkings) Trovare gli indirizzi ed il livello di carburante dei veicoli che durante il periodo di parcheggio hanno avuto almeno il 70% di livello di carburante. Ordinare in modo decrescente i risultati per livello di carburante.

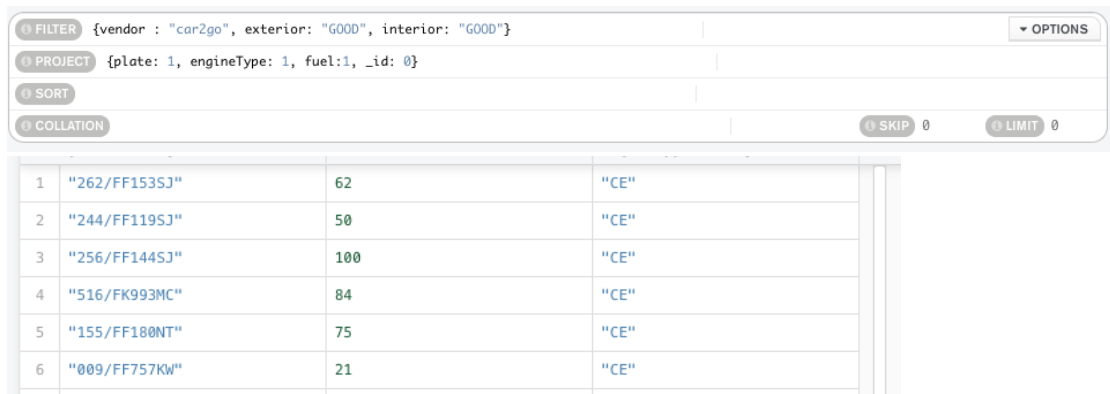
The screenshot shows a MongoDB query interface with the following configuration:

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

The results are displayed in a table view, showing 4 documents:

fuel	address
100	"Via Don Bartolomeo Grazioli, 26, 10137 Torino T0"
100	"Via Antonio Rosmini, 6, 10126 Torino T0"
100	"Via Bistagno, 5, 10136 Torino T0"
100	"Piazza Gian Lorenzo Bernini, 20, 10143 Torino T0"

3. (Parkings) Trovare la targa, il tipo di motore ed il livello di carburante per i veicoli 'car2go' (vendor) con buone condizioni interne ed esterne.



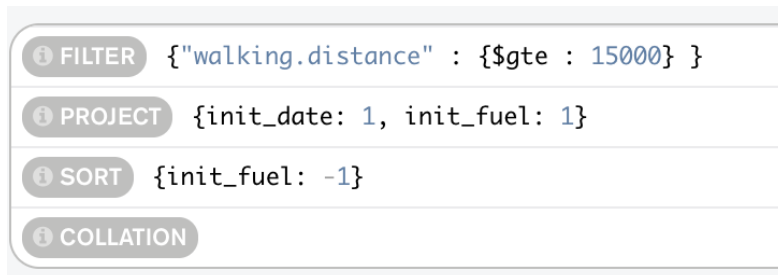
```

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

```

	plate	engineType	fuel	_id
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"

4. (Bookings) Per i noleggi che necessitano di una distanza a piedi maggiore di 15 Km (per raggiungere il veicolo), trovare l'ora ed il livello di carburante all'inizio del noleggio. Ordinare i risultati secondo livello di carburante iniziale decrescente.



```

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

```

3) Data Aggregation

5. (Bookings) Raggruppare i documenti in base al livello di carburante alla fine del noleggio. Per ogni gruppo selezionare il livello di carburante medio all'inizio del periodo di noleggio.

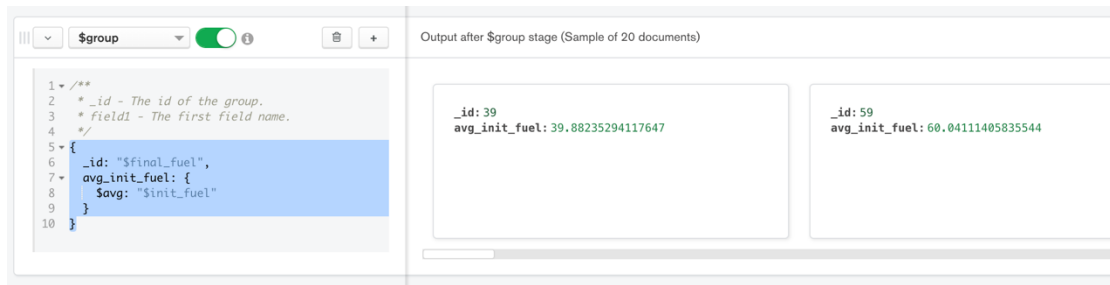
Query MongoDB:

```

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

```

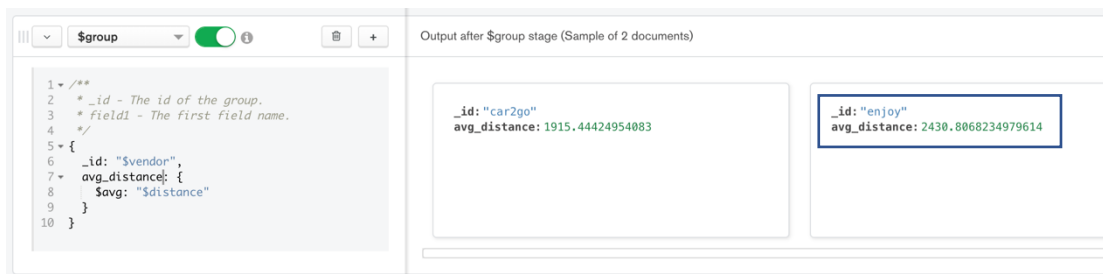
Esecuzione su Compass (selezionare "\$group" dal menu a tendina, poi completare i campi nel riquadro di testo)



6. (Bookings) Selezionare la distanza media coperta dai veicoli di ogni azienda (vendor). In media, per quali aziende gli utenti coprono le maggiori distanze?

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

Enjoy copre in media le distanze maggiori:




```

g. db.restaurants.mapReduce(function() {
    emit(null, 1);
  },
  function(key, values) {
    return Array.sum(values);
  },
  {query : { review:{$gt:4.5},
    maxPeople:{$gt:5}
  },
  out : "restaurants_reviewsGT5_maxPGT5"
})

```

```

h. db.restaurants.createIndex({location: "2dsphere"})
db.restaurants.findOne({location : {
  $near : {
    $geometry : {
      type: "Point",
      coordinates: [45.0644,7.6598]
    }
  }
}})

```

→ 006

```

i. db.restaurants.createIndex({location: "2dsphere"})
db.restaurants.find({location : {
  $near:{
    $geometry:{
      type: "Point",
      coordinates: [45.0623,7.6627]
    },
    $maxDistance: 500
  }
}}).count()

```

→ 3