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%.

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 `Date("<YYYY-mm-ddTHH:MM:ss">`)
3. (Parkings) Find the plate, the engine type and fuel level for 'car2go' vehicles (vendor) with good internal and external conditions.

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.

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

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

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:**
```json
{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).