

Fondamenti di basi di dati
Docente: Alessandro Fiori
Titolo argomento:
Esercizi su MongoDB

NoSQL in MongoDB Compass

Lo scopo dell'esercitazione è quello di acquisire familiarità con lo strumento **MongoDB Compass** e con il linguaggio di interrogazione di **MongoDB**. In questa esercitazione si richiede di esplorare i dati e di scrivere alcune query per recuperare i dati da database NoSQL basati su MongoDB.

1. Specifiche del problema

Il database contiene informazioni sul Car Sharing suddivise in due collezioni principali: Bookings e Parkings. Le informazioni più rilevanti per ciascuna collezione sono riportate nelle tabelle 1 (Parkings) e 2 (Bookings).

Nome	Tipo	Descrizione
_id	objectid	Identificatore del documento
address	string	Indirizzo del parcheggio
city	string	Città
engineType	string	Tipo di motore
exterior	string	Stringa che descrive le condizioni esterne del veicolo durante il parcheggio
final_date	date	Data e ora della fine del periodo di parcheggio
fuel	int32	Livello del carburante (0-100) durante il periodo di sosta
init_date	date	Data e ora di inizio del periodo di parcheggio
interior	string	Stringa che descrive la condizione interna del veicolo durante il parcheggio
loc	coordinates	Coordinate del parcheggio
plate	int32	Targa del veicolo
smartphoneRequired	Boolean	Valore booleano che indica se lo smartphone deve avviare o terminare il parcheggio
vendor	string	Proprietario del veicolo
vin	string	Identificatore del telaio del veicolo

Tabella 1: **Parkings** database

Nome	Tipo	Descrizione						
_id	objectid	Identificatore del documento						
car_name	string	Modello del veicolo						
city	string	Città in cui è stato prenotato il veicolo						
distance	int32	Distanza percorsa durante il noleggio del veicolo						
driving	object	<table> <tr> <td>distance</td><td>int32</td><td>Distanza percorsa durante il noleggio del veicolo (in metri)</td></tr> <tr> <td>duration</td><td>int32</td><td>Durata del noleggio (in secondi)</td></tr> </table>	distance	int32	Distanza percorsa durante il noleggio del veicolo (in metri)	duration	int32	Durata del noleggio (in secondi)
distance	int32	Distanza percorsa durante il noleggio del veicolo (in metri)						
duration	int32	Durata del noleggio (in secondi)						
engineType	string	Tipo di motore del veicolo						
exterior	string	Stringa che descrive le condizioni esterne del veicolo durante il noleggio						
final_address	string	Indirizzo della posizione finale del periodo di noleggio						
final_date	date	Data e ora della fine del periodo di noleggio						
final_fuel	int32	Livello del carburante (0-100) alla fine del periodo di noleggio						
init_address	int32	Indirizzo della posizione iniziale del periodo di noleggio						
init_date	date	Data e ora di inizio del periodo di noleggio						
init_fuel	int32	Livello del carburante (0-100) all'inizio del periodo di noleggio						
interior	string	Stringa che descrive la condizione interna del veicolo durante il noleggio						
plate	int32	Targa del veicolo						
smartphoneRequired	Boolean	Valore booleano che indica se lo smartphone deve avviare o terminare il parcheggio						
vendor	string	Proprietario del veicolo						
walking	object	<table> <tr> <td>distance</td><td>int32</td><td>Distanza a piedi per raggiungere il veicolo (in metri)</td></tr> </table>	distance	int32	Distanza a piedi per raggiungere il veicolo (in metri)			
distance	int32	Distanza a piedi per raggiungere il veicolo (in metri)						

		<table> <tr> <td>duration</td><td>int32</td><td>Durata del tragitto a piedi per raggiungere il veicolo (in secondi)</td></tr> </table>	duration	int32	Durata del tragitto a piedi per raggiungere il veicolo (in secondi)
duration	int32	Durata del tragitto a piedi per raggiungere il veicolo (in secondi)			

Tabella 2: **Bookings** database.

2. Connessione al database

Connessione al database remoto

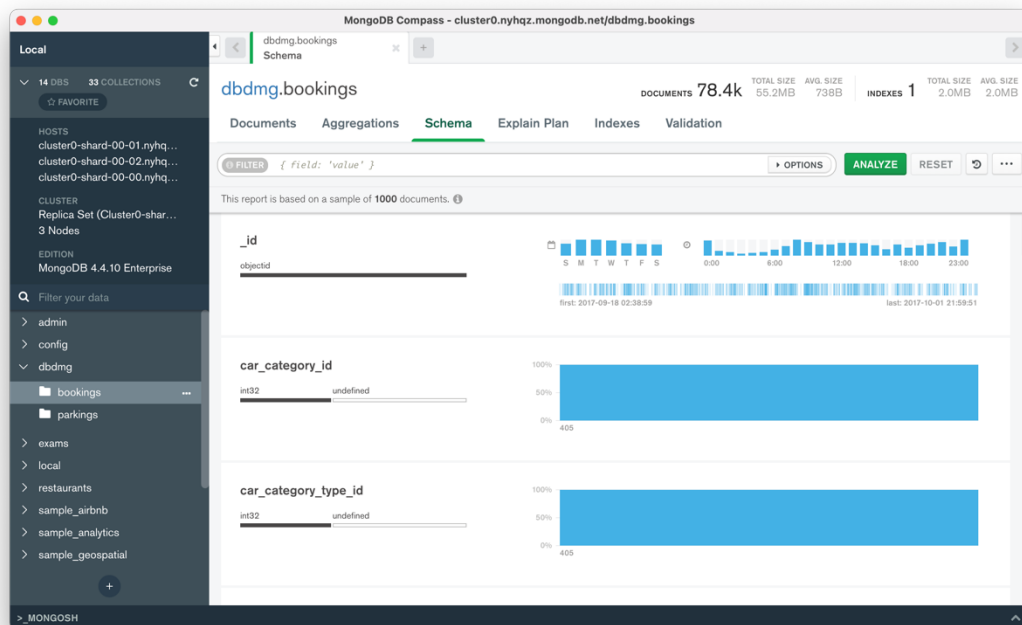
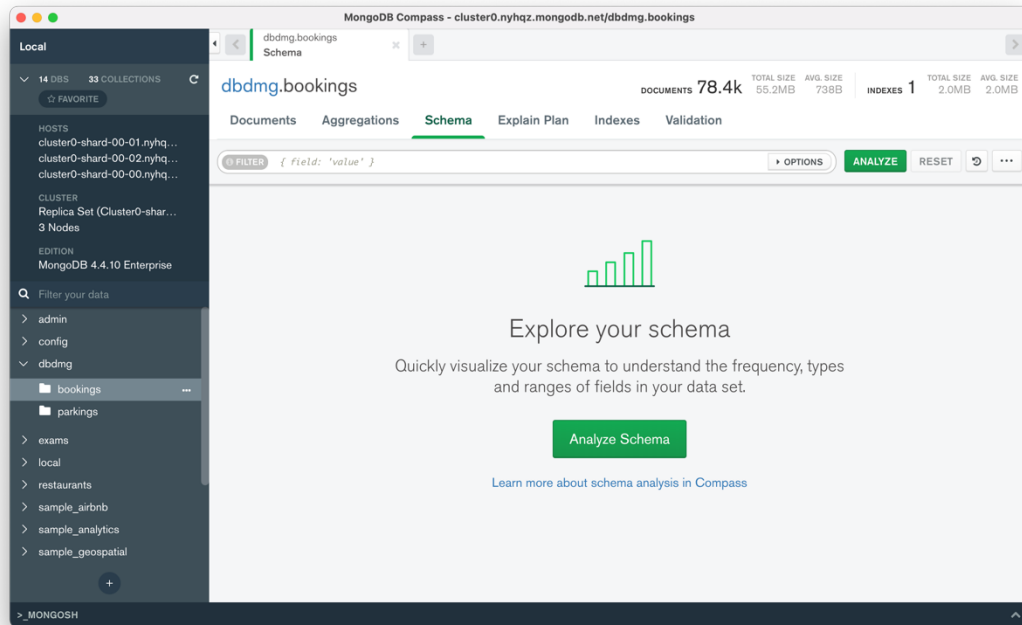
1. Scaricate MongoDB Compass all'indirizzo <https://www.mongodb.com/try/download/compass>
2. Installare e aprire l'applicazione.
3. Creare un cluster gratuito (se non ne avete già uno)
 - a. creare un account MongoDB (<https://www.mongodb.com/cloud/atlas/register>)
 - b. Selezionare l'opzione "Shared cluster" (cluster gratuito). Le impostazioni predefinite sono quelle impostate per ottenere l'account completamente gratuito.
 - c. dare un nome al cluster
 - d. accedere a Database Access
 - i. click su *"Add New Database User"*
 - ii. selezionare il metodo di autenticazione con password
 - iii. compilare il modulo
 - iv. lasciare tutte le opzioni predefinite
 - e. configurare l'accesso remoto
 - i. click su *"Network Access"*
 - ii. click su *"Add IP Address"*
 - iii. inserire in *Access List Entry* il valore 0.0.0.0/0
 - f. aggiungere i database di esempio
 - i. click sul bottone "..."
 - ii. click su Load Sample Dataset
4. Ottenere la stringa di connessione dal server MongoDB Atlas
 - a. click sul bottone *CONNECTIONS*
 - b. selezionare *"Connecting using MongoDB Compass"*
 - c. copiare la stringa di connessione
5. Incollare la stringa di connessione in MongoDB Compass
6. Click su **Connect**

3. Creare un database in MongoDB Compass

1. Click **Create Database**
 - a. Assegnare un nome al database
 - b. inserire il nome "Bookings" nel campo "Collection Name"
 - c. lasciare tutti i valori predefiniti per le altre opzioni
2. Aggiungere una nuova collezione
 - a. click sul bottone *"Create Collection"*
 - b. inserire il nome "Parkings"
 - c. lasciare tutti i valori predefiniti per le altre opzioni
3. Scaricare i dati di collezione dal cloud
4. Aggiungere dati a ciascuna collezione
 - a. selezionare una collezione
 - b. click sul bottone *"Add Data"*, poi Import file
 - c. selezionare il file della collezione corrispondente
 - d. selezionare Json type

- e. click *Import*
5. Selezionare la collezione Parkings
6. Andare nel campo "Indexes"
7. Creare un indice 2dsphere sul campo loc

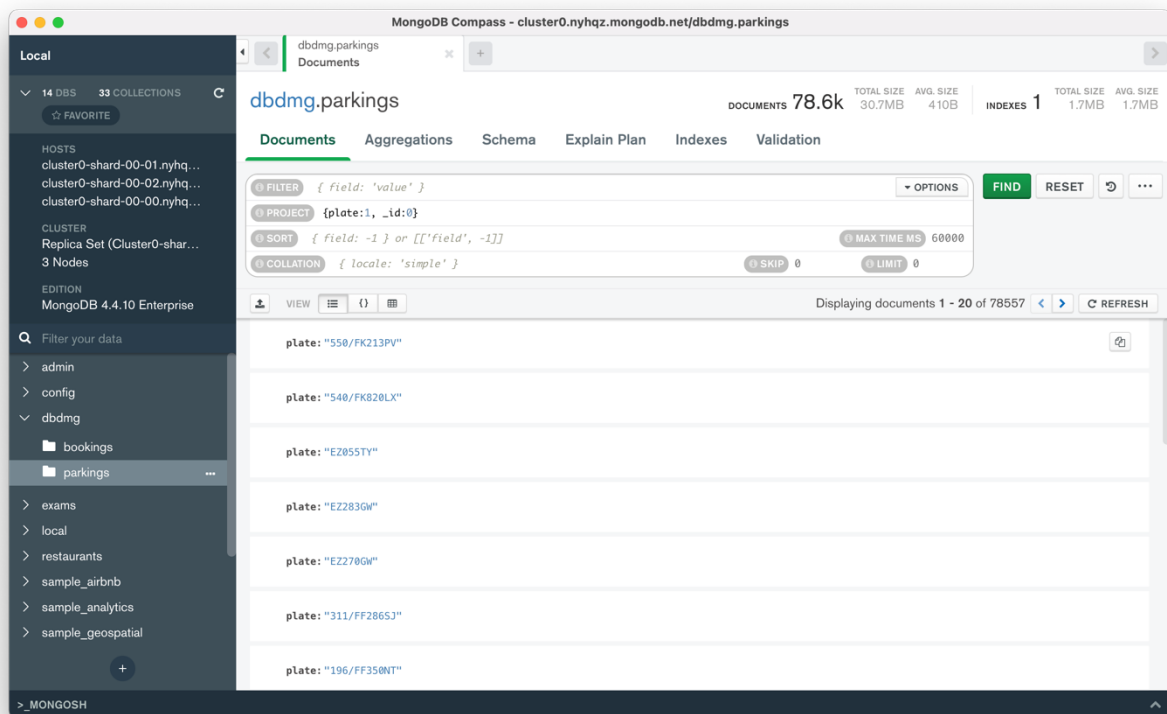
4. Analizzare il database utilizzando l'analizzatore di schemi



1. (Bookings) Identificare le percentuali più comuni del livello di carburante all'inizio del periodo di noleggio.
2. (Bookings) Identificare le percentuali più comuni del livello di carburante alla fine del periodo di

- noleggio.
3. (Parkings) Identificare la fascia oraria con il maggior numero di richieste di parcheggio (inizio del parcheggio).
4. (Parkings) Identificare la fascia oraria con il maggior numero di richieste di prenotazione (fine parcheggio).
5. (Parkings) Visualizzare sulla mappa i veicoli con un livello di carburante inferiore al 5%.

5. Interrogazione del database



1. (Parkings) Trovare le targhe e gli indirizzi di parcheggio dei veicoli che iniziano la prenotazione (fine parcheggio) dopo il 2017-09-30 alle 6AM.
Suggerimento: è possibile utilizzare la funzione `Date("<YYYY-mm-ddTHH:MM:ss")`
2. (Parkings) Trovare gli indirizzi e il livello di carburante dei veicoli che durante il periodo di sosta avevano almeno il 70% del livello di carburante. Ordinare i risultati in base al valore decrescente del livello di carburante.
3. (Parkings) Trova la targa, il tipo di motore e il livello di carburante per i veicoli "car2go" (venditore) con buone condizioni interne ed esterne.
4. (Bookings) Per i noleggi che richiedono un percorso a piedi superiore a 15 km (per raggiungere il veicolo), trovare l'ora e il livello di carburante all'inizio del periodo di noleggio. Ordinare i risultati in base al livello di carburante iniziale decrescente.

6. Data Aggregation

5. (Bookings) Raggruppare i documenti in base al loro livello di carburante alla fine del noleggio. Per ogni gruppo, selezionare il livello medio di carburante all'inizio del periodo di noleggio.
6. (Bookings) Selezionare la distanza media di guida per ciascun fornitore. In media, per quale fornitore gli utenti coprono distanze maggiori?

7. Query geografiche

7. (Parkings) Trovate i veicoli parcheggiati a meno di un chilometro da Piazza San Carlo (coordinate: 7.683016, 45.067764).
Suggerimento: utilizzare gli operatori [\\$geoWithin](#) e [\\$centerSphere](#).
8. (Parkings) Ripetere l'interrogazione del passo precedente utilizzando le coordinate di un luogo di interesse personale a Torino (ad esempio Piazza dei Mestieri) utilizzando Open Street Maps per trovare le coordinate esatte (www.openstreetmap.org, invertire l'ordine delle coordinate).

8. Query aggiuntive

9. (sample_training.grades) Per ogni tipologia di voto (scores.type) calcolare il voto medio.
10. (sample_training.inspections) Trovare i risultati che sono stati assegnati più di 1000 volte.
11. (sample_training.posts) Trovare i 3 tags associati al maggior numero di commenti
12. (sample_weatherdata) Per ogni giorno del mese (usare l'operatore \$dayOfMonth), calcolare la temperatura media, la velocità media del vento e la pressione media.
13. (sample_weatherdata) Per ogni tipo di misurazione e giorno del mese (usare l'operatore \$dayOfMonth), calcolare la temperatura massima, la velocità minima del vento e la distanza minima della visibilità.
14. (sample_supplies). Per ogni tag associato agli oggetti, visualizzare il prezzo medio. Filtrare il risultato per i tag associati a più di 5000 oggetti.
15. (sample_supplies). Considerando solo i clienti con età maggiore di 40, trovare i negozi che hanno venduto a più di 200 clienti.
16. (sample_supplies). Considerando solo i clienti con età inferiore a 50 che hanno usato il coupon, separatamente per ogni negozio e genere del cliente, visualizzare il valore medio di soddisfazione.
17. (sample_supplies). Considerando solo i prodotti con un prezzo di vendita superiore a 100, visualizzare per ogni mese e anno il numero complessivo di prodotti venduti e l'incasso complessivo.
18. (sample_resaturants.restaurants) Considerando solo i ristoranti di Manhattan, visualizzare il numero complessivo di recensioni e il numero totali di ristoranti.
19. (sample_resaturants.restaurants) Per ogni distretto e tipologia di cucina, visualizzare i primi 5 risultati che hanno un valore medio delle recensioni più alto.