

MongoDB

Lezione 19/09/2023

JSON

è un formato per lo scambio di dati, facile da leggere e da scrivere per le persone e facile da capire per le macchine.

Formato di testo indipendente dal linguaggio di programmazione.

Il json accetta numeri (anche con virgole), stringhe, NULL (valore) e boolean.

Due strutture --> insieme di coppie chiave/valore

--> un elenco ordinato di valori (array, vettori, elenchi, ecc...)

Tutti i linguaggi di programmazione moderni supportano le due strutture dei json.

insieme di coppie --> {coppie chiave/valore}

array --> []

Come si dichiara un json vuoto? --> { }

NO SQL

è un approccio che consente lo storage e l'esecuzione di query al di fuori delle classiche strutture dei database relazionali.

Si posso archiviare dati di database relazionali in database non relazionali, semplicemente i dati vengono memorizzati in maniera diversa.

DB relazionali --> dato più sicuro, più lenti

DB non relazionali --> dato meno sicuro, più performanti

Invece di una struttura tabulare, presentano una struttura di dati, come appunto i JSON. Siccome questi database non hanno dati strutturati, sono facilmente scalabili e vengono utilizzati per gestire grandi quantità di dati.

NOSQL sono formati da documenti e collezioni

Una collezione è un gruppo di documenti associato ad un database . Sono più flessibili rispetto alle tabelle e non dipendono da uno schema.

I documenti corrispondono agli oggetti del codice dell'applicazione, e vengono memorizzati da mongoDB come JSON.

MONGOSH

```
show dbs
```

mostra la lista dei DB esistenti

```
use *nomeDB*
```

se usi un database che esiste già, entra nel suo contesto

se ne crei uno nuovo, entra nel suo contesto ma finchè non lo popoli non lo crea

```
db.*nomeCollection*.insertOne({})
```

crea una collection

```
{
  acknowledged: true,
  insertedId: ObjectId("65096a3ac1f20e29b96c5e74")
}
```

collection vuota creata con valore NULL, creato ID univoco per l'oggetto

```
db.*nomeCollection*.find()
```

visualizza la collection e gli eventuali valori contenuti

ObjectId

valore esadecimale composto da 12 byte:

- 4 byte: numero di secondi dall'inizio dell'epoca Unix (1 Gennaio 1970), in questo modo è possibile risalire alla data di creazione di un document
- 3 byte per identificare la macchina
- 2 byte per l'id del processo
- 3 byte sono contatori che partono da un numero casuale
- .

```
db.*nomeCollection*.updateOne(*valore da cambiare*, {$set:{*campi da modificare o da aggiungere*}})
```

Consente di modificare il documento, se il valore esiste, lo modifica, se non esiste, lo crea

Questa funzione lavora su un documento, seguendo l'ordine degli ID, anche se abbiamo più documenti che soddisfano le opzioni del filtro

```
db.*nomeCollection*.deleteOne(*valore da eliminare*)
```

Rimuovo uno o più documenti da una collection.