



## NOME CORSO

### Fintech Software Developer

Unità Formativa (UF): Basi di dati SQL

Docente: Durando Giulio

Titolo argomento: MongoDB



# MongoDB

**MongoDB** è un database NoSQL orientato ai documenti, che nasce nel 2007 in California come servizio da utilizzare nell'ambito di un progetto più ampio, ma che presto è diventato un prodotto indipendente ed open-source. **MongoDB** archivia i dati in formato **BSON** ("**B**inary **J**SON") sia internamente che sulla rete. Tutto ciò che è possibile rappresentare in JSON può essere archiviato in modo nativo in MongoDB e recuperato altrettanto facilmente in JSON.

Gli oggetti BSON possono contenere oggetti **Date** o **Binary** che non sono rappresentabili in modo nativo in JSON puro.

Prima di iniziare è bene capire cosa significa avere a che fare con un database NoSQL, e qual'è la differenza tra questa categoria di basi di dati, ed i più tradizionali database SQL.

## MongoDB: caratteristiche di base

Come già detto, MongoDB è un database orientato ai documenti, ognuno dei quali è memorizzato nel formato JSON. Il documento è fondamentalmente un albero che può contenere molti dati, anche annidati. Un primo esempio è il seguente:

```
{
  nome: "Dante", cognome: "Alighieri",
  nato: 1265, morto: 1321,
  lingue: ["italiano", "latino"],
  opere: [
    {
      titolo: "Divina Commedia",
      iniziata: 1300,
      lingua: "italiano",
      tipo: "poesia",
      versi: "endecasillabi",
      libri: ["Inferno", "Purgatorio", "Paradiso"]
    }
  ]
}
```

I documenti sono raggruppati in collezioni che possono essere anche eterogenee. Ciò significa che non c'è uno schema fisso per i documenti. Tra le collezioni non ci sono relazioni o legami garantiti da MongoDB: in altre parole, non esiste un concetto analogo al vincolo di integrità referenziale.

Modello logico a parte, le caratteristiche chiave di MongoDB sono:

- consente servizi di alta disponibilità, dal momento che la replicazione di un database (Replica Set) può avvenire in modo molto semplice;
- garantisce la scalabilità automatica, ossia la possibilità di distribuire (Sharding) le collezioni in cluster di nodi, in modo da supportare grandi quantità di dati senza influire pesantemente sulle performance.

MongoDB si adatta a molti contesti, in generale quando si manipolano grandi quantità di dati eterogenei e senza uno schema. Non è invece opportuno quando si devono gestire molte relazioni tra oggetti, e si vuole garantire l'integrità referenziale tra essi (ad esempio in un ERP).

## Installazione



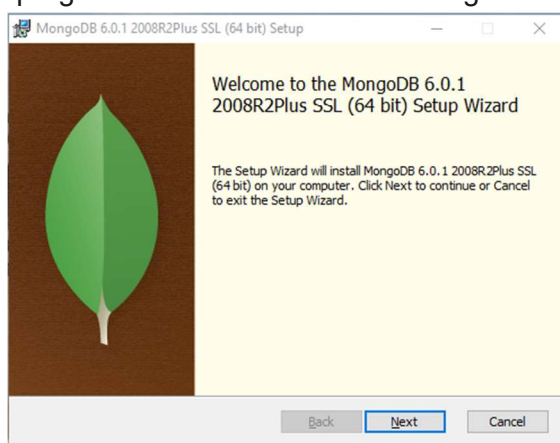
MongoDB è un prodotto **open source** e in quanto tale viene fornito sia sotto forma di file binari sia con i sorgenti compilabili. L'installazione con i binari è la via più breve per installare MongoDB sulla propria macchina di sviluppo.

Per prima cosa bisogna scaricare il pacchetto d'installazione dal sito ufficiale. Troviamo la disponibilità dei pacchetti per i principali sistema operativi: Windows, Linux, Solaris e Mac OS X.

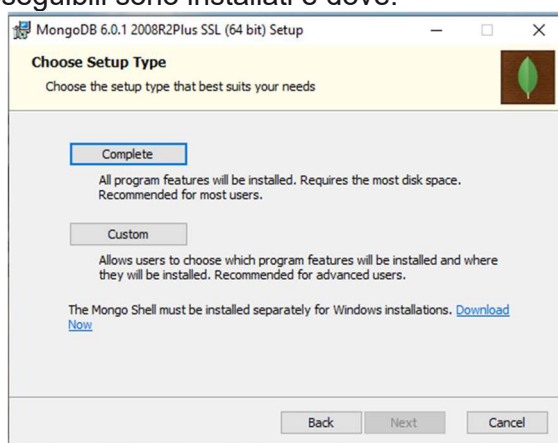
### Installare MongoDB su Windows

Per installare MongoDB 6.0 Community Edition su Windows seguiamo il seguente procedimento

- a) Dal Download center di MongoDB si deve scaricare il programma di installazione selezionando
  - a) Nel menu a discesa **Versione**, selezionare la versione di MongoDB da scaricare.
  - b) Nel menu a discesa **Piattaforma**, selezionare **Windows**.
  - c) Nell'elenco a discesa **Pacchetto**, selezionare **.msi**.
  - d) Fare clic su **Scarica**.
- b) Eseguire il programma di installazione di MongoDB



- c) Seguire la procedura guidata di installazione di MongoDB Community Edition.
  - Scegliere il tipo di installazione è possibile scegliere il tipo di configurazione Completa (consigliato per la maggior parte degli utenti) o personalizzata. L'opzione Configurazione completa installa MongoDB e gli strumenti MongoDB nella posizione predefinita. L'opzione Configurazione personalizzata consente di specificare quali eseguibili sono installati e dove.





- Configurazione del servizio

A partire da MongoDB 4.0, è possibile configurare MongoDB come servizio Windows durante l'installazione o semplicemente installare i binari.

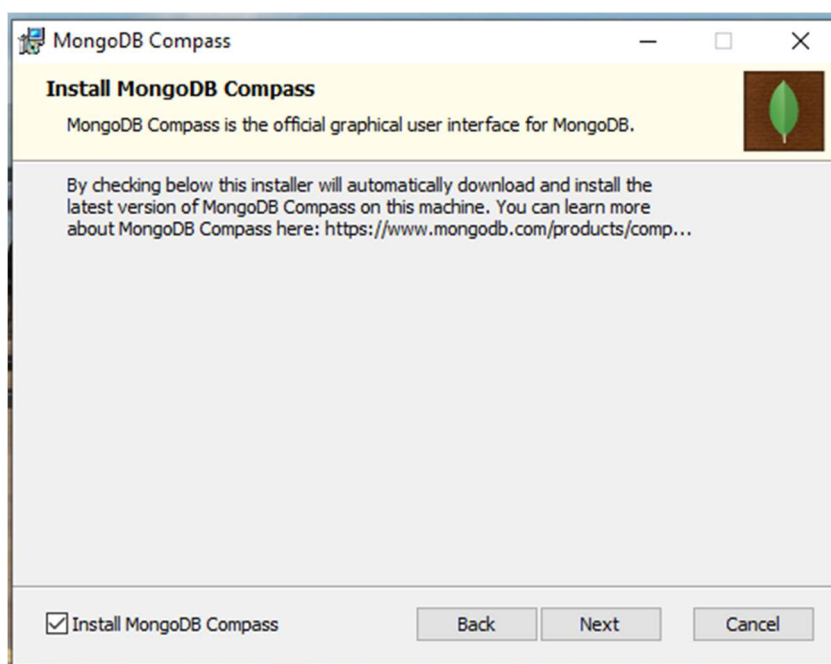
- Selezionare **Installa MongoDB come servizio** MongoDB come servizio.
- Selezionare una delle seguenti opzioni:
  - **Eseguire il servizio come utente del servizio di rete** (impostazione predefinita)  
Questo è un account utente di Windows integrato in Windows
  - **Eseguire il servizio come utente locale o di dominio**
    - Per un account utente locale esistente, specificare un punto (ad es. .) per il **dominio dell'account** e specificare il **nome dell'account** e la **password dell'account** per l'utente.
    - Per un utente di dominio esistente, specificare il **dominio dell'account**, il **nome dell'account** e la **password dell'account** per quell'utente.
- **Nome del servizio.** Specificare il nome del servizio. Il nome predefinito è MongoDB. Se si ha già un servizio con il nome specificato, si deve scegliere un altro nome.
- **Directory dati.** Specificare la directory dei dati, che corrisponde a `--dbpath`. Se la directory non esiste, il programma di installazione creerà la directory e imposterà l'accesso alla directory per l'utente del servizio.



- **Directory di registro.** Specificare la directory Log, che corrisponde a `--logpath`. Se la directory non esiste, il programma di installazione creerà la directory e imposterà l'accesso alla directory per l'utente del servizio.

d) Installare MongoDB Compass

Se si vuole installare anche MongoDB Compass selezionare Installa MongoDB Compass (predefinito).



e) Fare clic su **Installa**

L'**installazione standard** prevede: il server (*mongod*), il client, strumenti di monitoraggio e strumenti per importare ed esportare dati. Il router (*mongos*) è lo strumento che permette di effettuare lo **sharding**.

Scegliamo di installare tutto e andiamo avanti. Dopo qualche minuto l'installazione è terminata, quindi possiamo lanciare MongoDB server. L'eseguibile da lanciare è *mongod.exe* che si trova nella sottocartella *bin* nel percorso di installazione di MongoDB, ma prima dobbiamo creare una cartella per memorizzare i file del database. Apriamo dunque un prompt dei comandi in modalità amministratore e creiamo una cartella:

```
C:\> mkdir mongodata
```

Quindi possiamo **avviare il server**:

```
C:\> "C:\Programmi\MongoDB 2.6 Standard\bin\mongod.exe" --dbpath C:\mongodata
```

Ovviamente il percorso dell'eseguibile *mongod.exe* dipende dal percorso in cui abbiamo scelto di installare MongoDB.

Se tutto è andato a buon fine vedremo una schermata come la seguente, che ci indica che il server è avviato e su quale porta è in ascolto.



```
2014-10-23T14:20:57.468+0200 [initandlisten] db version v2.6.5
2014-10-23T14:20:57.468+0200 [initandlisten] git version: e99d4fcb4279c0279796f2
37aa92fe3b64560bf6
2014-10-23T14:20:57.469+0200 [initandlisten] build info: windows sys.getwindowsv
ersion(major=6, minor=1, build=7601, platform=2, service_pack='Service Pack 1')
BOOST_LIB_VERSION=1_49
2014-10-23T14:20:57.470+0200 [initandlisten] allocator: system
2014-10-23T14:20:57.471+0200 [initandlisten] options: { storage: { dbPath: "c:\m
ongodata" } }
2014-10-23T14:20:57.474+0200 [initandlisten] journal dir=c:\mongodata\journal
2014-10-23T14:20:57.475+0200 [initandlisten] recover : no journal files present,
no recovery needed
2014-10-23T14:20:57.599+0200 [initandlisten] allocating new ns file c:\mongodata
\local.ns, filling with zeroes...
2014-10-23T14:20:57.854+0200 [FileAllocator] allocating new datafile c:\mongodat
a\local.0, filling with zeroes...
2014-10-23T14:20:57.855+0200 [FileAllocator] creating directory c:\mongodata\tm
p
2014-10-23T14:20:57.991+0200 [FileAllocator] done allocating datafile c:\mongoda
ta\local.0, size: 64MB, took 0.134 secs
2014-10-23T14:20:58.021+0200 [initandlisten] build index on: local.startup_log p
roperties: { v: 1, key: { _id: 1 }, name: "_id_", ns: "local.startup_log" }
2014-10-23T14:20:58.022+0200 [initandlisten] added index to empty collection
2014-10-23T14:20:58.022+0200 [initandlisten] command local.$cmd command: create
{ create: "startup_log", size: 10485760, capped: true } noreturn:1 keyUpdates:0
numYields:0 reslen:37 423ms
2014-10-23T14:20:58.027+0200 [initandlisten] waiting for connections on port 270
```

Per interagire con il nostro server abbiamo bisogno ovviamente di un **client**. La console client si chiama *mongo.exe* e si trova nello stesso percorso del server. Avviamo dunque un altro prompt dei comandi e lanciamo:

```
C:\> "C:\Programmi\MongoDB 2.6 Standard\bin\mongo.exe"
```

Se la connessione va a buon fine, verrà visualizzato un messaggio di benvenuto come il seguente:

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\System32>"c:\Program Files\MongoDB 2.6 Standard\bin\mongo.exe"
MongoDB shell version: 2.6.5
connecting to: test
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
    http://docs.mongodb.org/
Questions? Try the support group
    http://groups.google.com/group/mongodb-user
>
```

## La shell e le operazioni CRUD

La shell di MongoDB permette di effettuare le principali operazioni di gestione del database (Create, Read, Update e Delete) su MongoDB.

Per installare la Shell si devono seguire i seguenti passi

- Aprire la pagina di download di MongoDB Shell al seguente indirizzo:  
<https://www.mongodb.com/try/download/tools>
- Selezionare Mongo Shell
- Selezionare il sistema operativo su cui va installata la Shell
- Selezionare il tipo di installazione che si desidera fare (con file .Zip o .Msi)
- Eseguire il file di installazione
- Seguire le istruzioni per l'installazione di mongosh.

Per utilizzare la Shell di MongoDB, è necessario disporre di una distribuzione MongoDB a cui connettersi





Dopo aver avviato il nostro server MongoDB (***mongod***) e la shell (***mongosh***), possiamo utilizzare quest'ultima per interagire attivamente con il nostro database.

MongoDB memorizza i documenti in collezioni (**collection**). Non dobbiamo pensare alle collezioni come se fossero tabelle di un database relazionale: infatti, i documenti di una stessa collezione potrebbero avere strutture diverse, nonostante essi condividano gli stessi indici.

Un **database**, invece, è un insieme di collezioni che corrisponde ad un insieme di file nel disco fisso. Un'istanza di MongoDB può gestire più database. Di seguito mostreremo brevemente come utilizzare la shell per effettuare le **operazioni CRUD (Create, Read, Update, Delete)**.

### **Inserimento di documenti nel database**

Dopo aver aperto la shell, selezioniamo il database da usare con il comando:

#### **use calciatori**

MongoDB risponderà con il messaggio **switched to db calciatori**. In realtà non abbiamo ancora creato alcun database con nome html.it. Esso **verrà generato al momento del suo primo effettivo utilizzo**.

Su MongoDB, infatti, non esiste alcun comando esplicito per creare un database. Possiamo però sapere quali database sono stati effettivamente creati, utilizzando il comando **show databases**.

Anche le collezioni vengono create implicitamente al primo utilizzo, quindi non è necessario farlo con un comando specifico. È sufficiente passare direttamente alla creazione di un documento: