

Database NoSQL

Gianluca Pironato

Database relazionali

- ▶ I DBMS più diffusi seguono il modello relazionale, ovvero organizzano i dati in tabelle, ciascuna corrispondente a un'entità, e collegano le entità mediante chiavi (la chiave primaria di un record di una tabella può fungere da chiave esterna per più record di un'altra tabella, indicando così una relazione tra le due tabelle).
- ▶ I punti di forza sono noti e tra questi troviamo il rispetto delle proprietà ACID:
 - ▶ **Atomicità:** le transazioni devono essere completate interamente o fallire.
 - ▶ **Consistenza:** il database rimane in uno stato valido dopo ogni transazione.
 - ▶ **Isolamento:** le transazioni concorrenti non interferiscono tra loro (prima si esegue una, poi l'altra).
 - ▶ **Durabilità:** i dati confermati rimangono memorizzati anche in caso di guasti.

...hanno anche punti deboli?

- ▶ L'aumento dei contenuti generati dagli utenti (ormai Big Data) e l'onnipresente paradigma distribuito hanno evidenziato alcuni limiti del modello relazionale.
- ▶ In particolare, i DBMS relazionali scalano bene in verticale (un grande server database), ma non altrettanto in orizzontale (un database distribuito su più server).
- ▶ Inoltre, i DBMS relazionali necessitano di dati strutturati secondo uno schema rigido, ma ciò non sempre è facilmente ottenibile nella realtà.
 - ▶ Esempio: ho alcuni utenti che non hanno fornito il numero di telefono, ma devo comunque prevedere una colonna apposita nella mia tabella per anche solo un utente!
 - ▶ Altro esempio: devo per forza operare una ALTER TABLE su tutta la tabella se voglio aggiungere una colonna per un attributo precedentemente non previsto...

L'emergere dei database NoSQL

- ▶ I Big Data e le applicazioni distribuite richiedono soluzioni più flessibili rispetto ai database relazionali.
- ▶ Nasce il paradigma NoSQL (Not only SQL), che combina tecniche dei database relazionali (le query ancora si fanno...) con nuovi modelli di archiviazione.
- ▶ NoSQL supporta il cloud computing e l'alta scalabilità.

Il modello BASE nei database distribuiti

- ▶ Contrariamente a ACID, NoSQL segue il modello BASE:
 - ▶ **Basically Available**: il sistema garantisce disponibilità anche in caso di guasti parziali.
 - ▶ **Soft State**: la consistenza dei dati non è garantita come da ACID (ma nella pratica non avremo problemi di sorta).
 - ▶ **Eventual Consistency**: il sistema converge gradualmente a uno stato consistente.
- ▶ Questo modello consente alte prestazioni e scalabilità, rinunciando a una rigida consistenza immediata.

Caratteristiche principali dei database NoSQL

- ▶ **Non relazionali:** non si basano su tabelle e schemi fissi (schemaless addirittura! Se un utente non fornisce il numero di telefono non mantengo l'attributo "phone" impostato a NULL...).
- ▶ **Flessibili:** permettono modifiche dinamiche della struttura dei dati (se l'utente fornisce il numero di telefono, ovviamente lo posso memorizzare senza problemi).
- ▶ **Scalabilità orizzontale:** possono crescere distribuendo il carico su più server.
- ▶ **Prestazioni elevate:** ottimizzati per operazioni di lettura/scrittura veloci.
- ▶ **Alta funzionalità:** supportano accessi ad alte prestazioni per applicazioni web, social media e big data (il prezzo pagato è quello del mancato rispetto delle proprietà ACID).

Tipologie di database NoSQL

- ▶ **Key-Value Store:** coppie chiave-valore, ideali per cache e sessioni.
- ▶ **Column-Family Store:** dati organizzati in colonne, usati per analisi distribuite.
- ▶ **Document Store:** memorizzano dati in formato JSON o BSON, ottimi per applicazioni web.
- ▶ **Graph Database:** modellano dati come nodi e relazioni, perfetti per social network.

Per completezza: questi modelli di organizzazione sono più facili da distribuire. Contrariamente, una singola tabella andrebbe spaccata in sotto-tabelle... raggruppo le righe (istanze di entità)? raggruppo le colonne (raccolte di attributi)? È un problema più complicato.

MongoDB

MongoDB è il DBMS che utilizzeremo noi:

- ▶ Database NoSQL orientato ai documenti.
- ▶ Ogni record è un documento BSON (JSON binario) con una chiave univoca ('_id').
- ▶ I documenti sono organizzati in collezioni dello stesso tipo (es. per ogni utente ho un documento, tutti gli utenti stanno nella collezione di documenti "utenti").
- ▶ Il recupero dei dati segue un modello di tabella di hash, garantendo efficienza.



Conclusione

- ▶ I database NoSQL rappresentano una soluzione flessibile e scalabile per i Big Data.
- ▶ Sono utilizzati in contesti moderni come social network, IoT e analisi di grandi dati.
- ▶ La scelta tra SQL e NoSQL dipende dalle necessità specifiche dell'applicazione.