

# Tutorato di Basi di dati e Web

Matthew Rossi

[matthew.rossi@unibg.it](mailto:matthew.rossi@unibg.it)



# NoSQL

Memorizza e recupera dati usando modelli diversi da quello relazionale

## Tipologie

- Colonnare
- Chiave-valore
- Documenti
- Grafi

Implementazioni reali spesso offrono rappresentazione dei dati appartenenti a più categoria

# Database colonnare

Memorizza ogni colonna separatamente permettendo scan dei dati più rapidi quando sono coinvolte un numero limitato di colonne

- Mappa i dati usando un identificativo artificiale (il numero di riga)
- Riduce lo storage necessario alla memorizzazione usando tecniche di compressione ottimizzate in base alla tipologia del dato

Product	
ID	Value
1	Beer
2	Beer
3	Vodka
4	Whiskey
5	Whiskey
6	Vodka
7	Vodka

Customer	
ID	Customer
1	Thomas
2	Thomas
3	Thomas
4	Christian
5	Christian
6	Alexei
7	Alexei

Date	
ID	Date
1	2011-11-25
2	2011-11-25
3	2011-11-25
4	2011-11-25
5	2011-11-25
6	2011-11-25
7	2011-11-25

Sale	
ID	Sale
1	2 GBP
2	2 GBP
3	10 GBP
4	5 GBP
5	5 GBP
6	10 GBP
7	10 GBP

# Quando usare un database colonnare?

In un database colonnare è semplice aggiungere nuove colonne, ma aggiungere un intero record richiede di modificare tutte le tabelle

- I database tradizionali sono preferibili per online transaction processing
- I database colonnari danno il loro meglio quando eseguono analisi dati

Soluzioni open-source



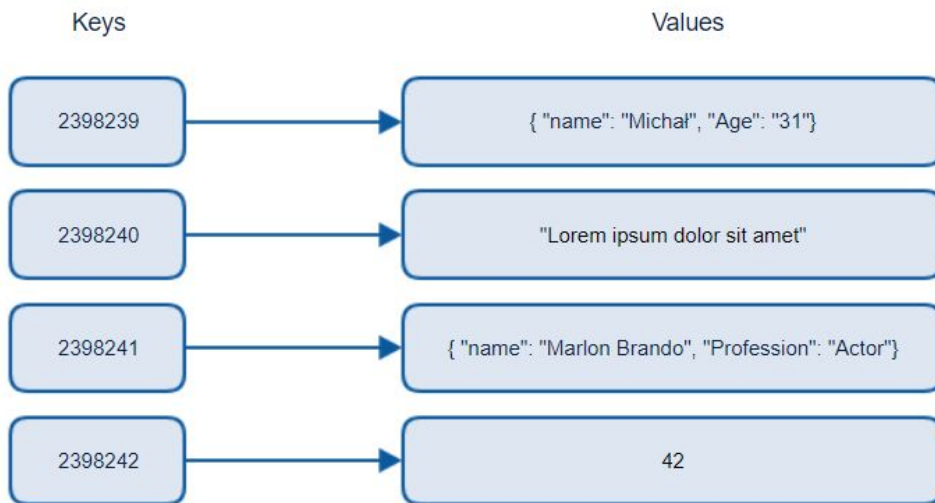
Soluzioni Platform-as-a-Service (PaaS)



# Database chiave-valore

Immagazzina i dati come un insieme di coppie chiave-valore

Le chiavi e i valori possono essere qualsiasi cosa, da un oggetto semplice ad articolati oggetti composti



# Proprietà dei database chiave-valore

- Molto veloci in lettura e scrittura
- Estremamente flessibili (privi di schema) e efficienti nello storage (privi di valori nulli)
- Facilmente partizionabili orizzontalmente (scaling lineare col numero di nodi)

# Quando usare un database chiave-valore?

- Non si ha uno schema di dati definito
- Lo scaling necessario non è soddisfabile da altre soluzioni
- Sono necessarie operazioni di lettura e scrittura molto performanti
- Le operazioni sono realizzate sulla base della chiave stabilita

## Soluzioni open-source



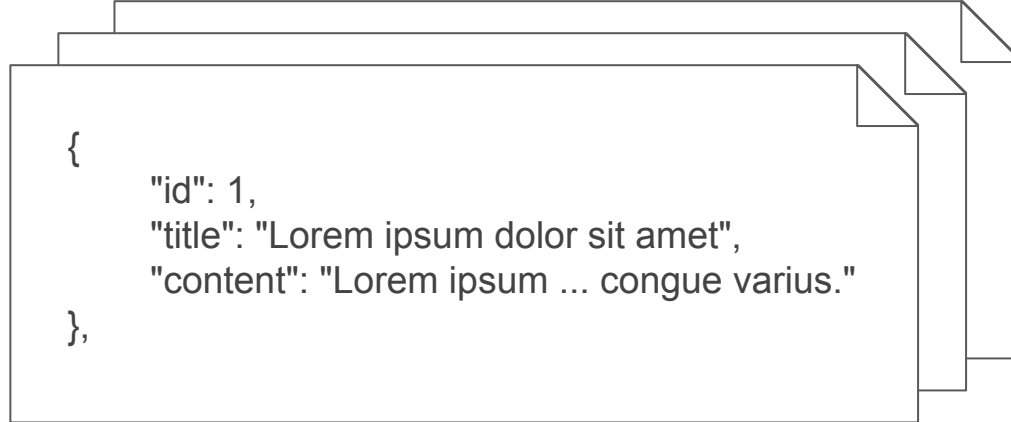
## Soluzioni Platform-as-a-Service



# Database di documenti

Immagazzina dati semi strutturati organizzati in documenti

I documenti sono identificati mediante l'uso di una chiave univoca e sono codificati usando un solo formato (XML, YAML, JSON e BSON)





# Proprietà dei database di documenti

- Veloci in lettura e scrittura
- Flessibili, ma devono aderire ad un formato di codifica prestabilito
- Partizionabili orizzontalmente
- Permettono query complesse sui dati

# Quando usare un database di documenti?

- Lo schema è sottoposto a cambiamenti frequenti
- Il sistema richiede frequentemente i dati in forma aggregata
- Lo scaling necessario non è soddisfabile dalle soluzioni tradizionali

## Soluzioni open-source



## Soluzioni Platform-as-a-Service (PaaS)

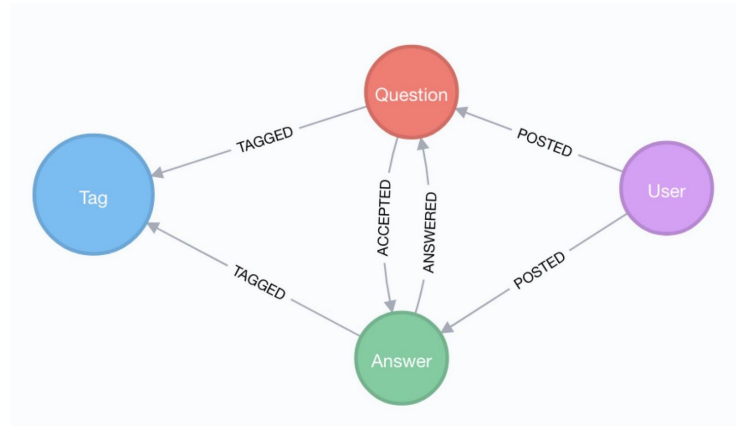


# Database a grafo

Immagazzina i dati sotto forma di grafo

Due componenti:

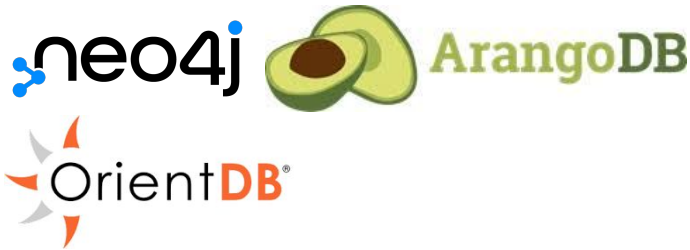
- Nodi: le entità che si vuole rappresentare
- Archi: le relazioni tra due entità con le sue proprietà



# Quando usare un database a grafo?

- Necessità di gestire in modo efficiente le relazioni tra entità
- I dati sono estremamente interconnessi

Soluzioni open-source



Soluzioni Platform-as-a-Service (PaaS)



# MongoDB

Un database di documenti progettato per facilitare lo sviluppo software e lo scaling

Un record in MongoDB è un documento, ossia una struttura dati composta da coppie campo-valore

Il valore dei campi può includere documenti, liste e liste di documenti

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```



← field: value

← field: value

← field: value

← field: value

# MongoDB - Collezioni

Le collezioni memorizzano insiemi di documenti e sono del tutto analoghe alle tabelle nei database relazionali

Oltre alle collezioni, MongoDB supporta la definizione di

- Viste di sola lettura
- Viste materializzate on-demand

# MongoDB - Caratteristiche

- Elevate performance → Indicizzazione
- Linguaggio di query ricco → CRUD
- Elevata disponibilità → Replicazione dei dati
- Scalabilità orizzontale → Sharding e Zone