

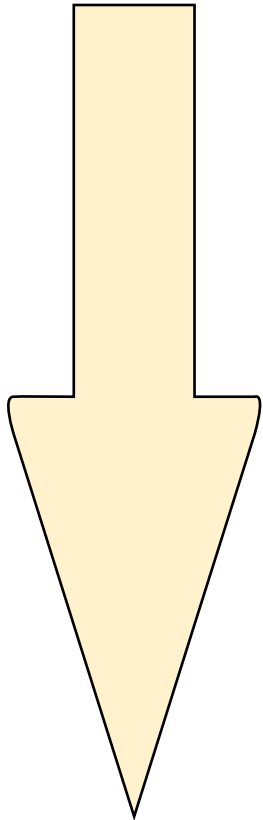
Torino Technologies Group
21 Ottobre 2010

NoSQL & MongoDB

Fabrizio Accatino
fhtino@gmail.com

Polpettone di necessità

Approccio classico per storage dati: **DB Relazionale**



"Nuove" necessità:

- Letture/scritture su pochi dati (web-application)
- Scalabilità orizzontale molto spinta
- Alte performance (soprattutto in lettura)
- Memorizzare dati non relazionali
- Modello relazionale troppo complesso per certi tipi di dati
- Dati non strutturati
- Struttura variabile, anche nel tempo

i DB relazionali classici non vanno sempre bene

NoSQL Movement

- NoSQL --> Not Relational / Not Fixed Schema
- Superare i limiti del modello relazionale
- Molti prodotti, molto diversi fra di loro
- Funzionalità non omogenee fra prodotti
- Caratteristiche generali:
 - no relazioni (--> no join)
 - no transazioni
 - denormalizzazione
 - scalano orizzontalmente
 - table schema non fisso
 - ID autogenerati
 - inclusione di sotto-oggetti
 - ricerche molto veloci per ID

MongoDB



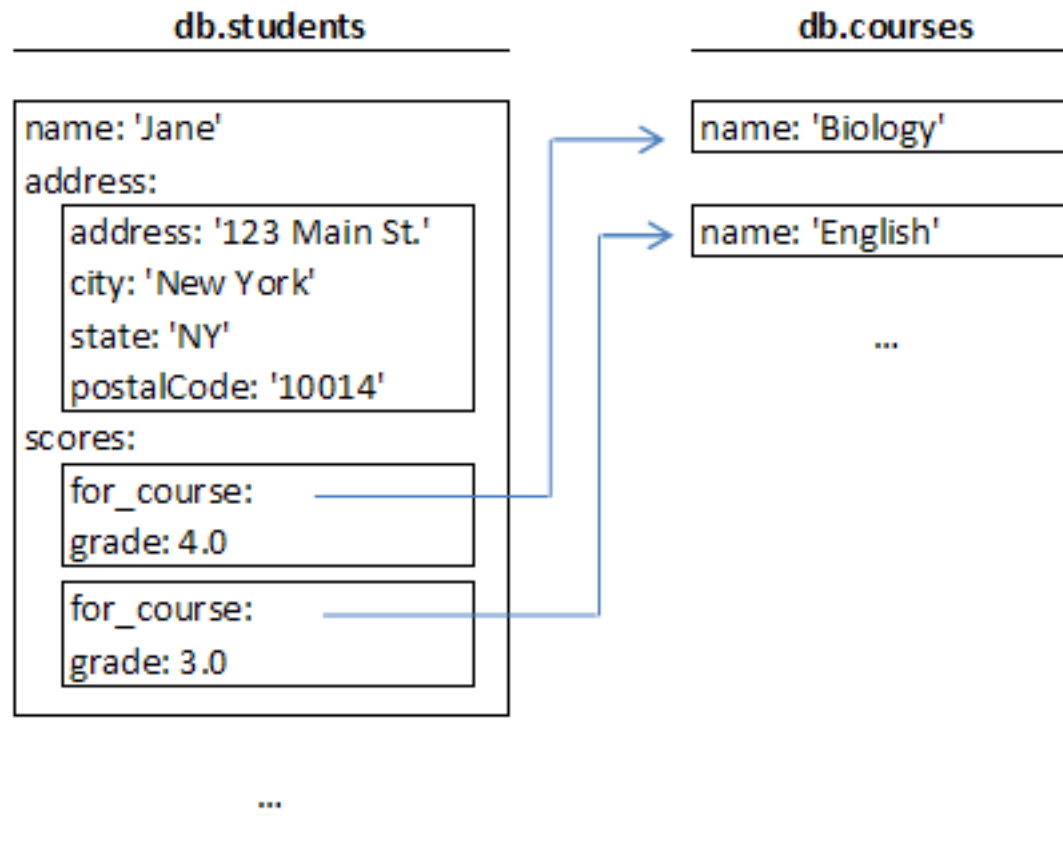
mongoDB

- Storage di documenti Json
- Scalabilità orizzontale
- Linux, Windows, Mac, Solaris
- Replica & Sharding
- Driver per molti framework/linguaggi
 - ... troppi driver :(

Json document

```
{  
  "Name": "Fabrizio",  
  "Email": "fabrizio@fhtino.it",  
  "Age": 39,  
  "Address": {  
    "Street": "Via Roma 1",  
    "City": "Moncalieri"  
  }  
  "Friends": ["Marco","Stefano","Ugo"],  
  "Contacts": [  
    {"Type": "email", "address": "me@fhtino.local"},  
    {"Type": "cell", "number": "000111222333"},  
    {"Type": "chat", "address": "skype:myid"}  
  ],  
  "Notes": "Lorem ipsum..."  
}
```

MongoDB - Linking vs Embedding



MongoDB

NO foreign key

NO transazioni

NO join

NO lock

NO group by & c.

SI: operazioni atomiche su singolo documento

- insert new
- modifier su operazione
- "update if current" (Optimistic concurrency)

SI: indici su "campi"

SI: indici univoci su "campi"

+ Dati geografici (Lat/Long)

MongoDB

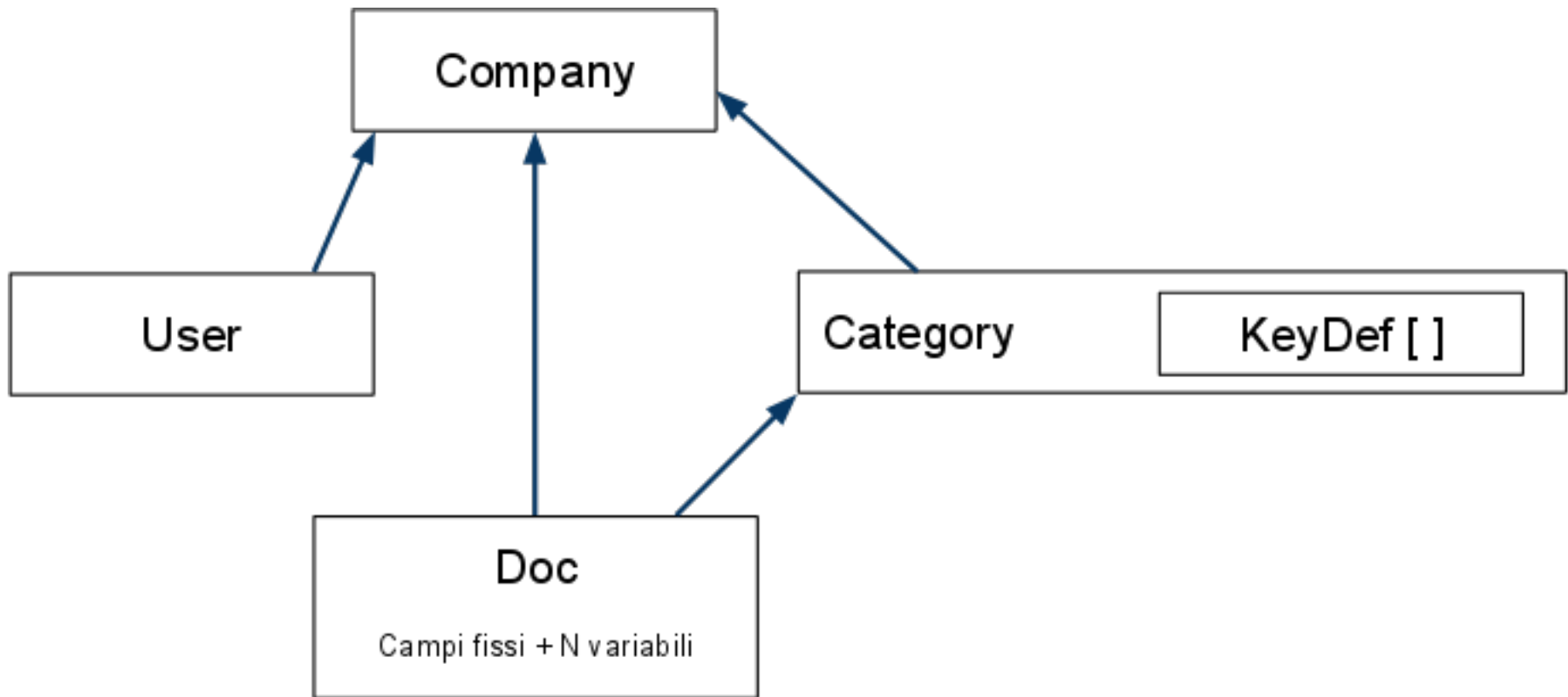
- Similitudini con il mondo relazionale
 - Record --> Json document
 - Table --> Collection
 - Index --> Index
 - Join --> Embed and/or link

MongoDB-csharp

- Molte funzionalità
- Ottime performance
- Supporto a LINQ
 - attenzione a cosa viene fatto sul server e cosa sul client

<http://github.com/mongodb-csharp/mongodb-csharp>

DEMO - Schema



MongoDB + VS2010

DEMO

... pessimo codice, abbiate pietà! :)