L2. NoSQL[1]

S2. Document stores: MongoDB

Dr A Boronat

[1] Materials based on [1, ch 5,9]

Table of Contents

MongoDB: features

- Document stores can be seen as key-value stores
  - each document has a key
  - the document is the value and it can be inspected
- MongoDB stores and retrieves documents
  - a document is a hierarchical data structure which may consist of maps, collections, and scalar values
  - Ex: XML, JSON, BSON, ...
- A collection can contain documents with different fields
  - if a document has no data, the corresponding field is marked to null (for univalued properties) and to empty (for multivalued properties)
  - new fields can be added without having to change the existing documents

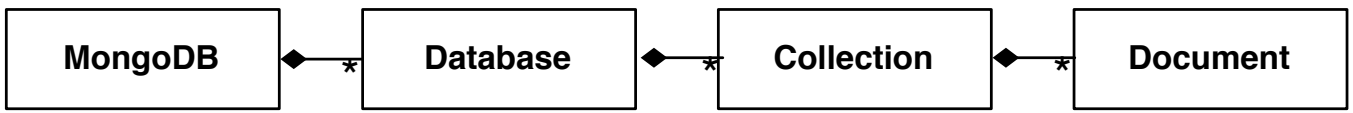
MongoDB: features

Comparison of relational databases and document store terminology:

RDBMS	MongoDB
database instance	MongoDB instance
schema	database
table	collection
row	document
join	DBRef

MongoDB: usage

- MongoDB structure



- to insert a document `database.collection.insert(document)`

Use Cases

- Event logging
  - data captured by events keeps changing
  - events can be sharded by the name of the application where the event originated or by the type of event, e.g. orderProcessed, customerLogged
- CMSs, blogging platforms
- web analytics or real-time analytics
  - to store page views or unique visitors
  - new metrics can be added without schema changes
- e-commerce applications
  - flexible schema for products and orders

Limitations

- Complex transactions spanning different operations
  - cross-document operations
  - although some document stores support such operations, e.g. RavenDB
- Queries against varying aggregate structure
  - your queries will change
  - data normalization may help to find a static structure

References

Pramod J. Sadalage and Martin Fowler.
NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence.
Addison-Wesley Professional, 1st edition, 2012.