

MBA em IA e Big Data



Curso 01 - Linguagens e Ferramentas para Inteligência Artificial e Big Data (Python e SQL)

Introdução ao banco de dados MongoDB

Jose Fernando Rodrigues Junior
ICMC-USP São Carlos

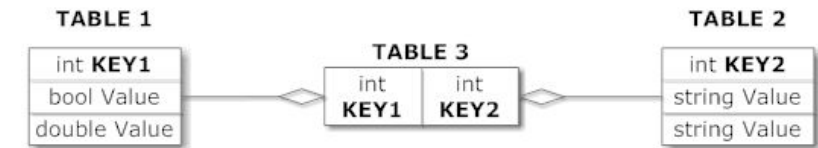
Objetivo: apresentar conceitos sobre sistemas não relacionais contrastando o modelo relacional com o sistema MongoDB

MongoDB

- mongoDB = “Humongous DB”
 - Open-source
 - Document-based
 - “High performance, high availability”
 - Escalabilidade facilitada

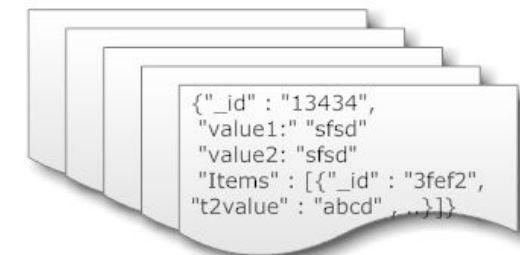


Relational Model



Document Model

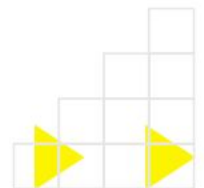
Collection ("Things")





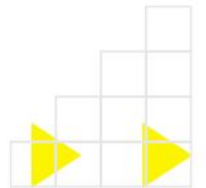
Modelo de Dados

- Document-Based (max 16 MB por entrada)
- Formato BSON, consistindo de pares field-value
- Cada documento faz parte de uma coleção
- Coleções
 - Semelhante a relações de um BD relacional (semelhante, não igual)
 - Documentos não possuem estrutura uniforme



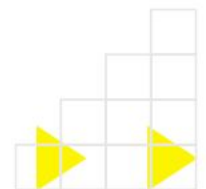
JSON

- ❑ “JavaScript Object Notation”
- ❑ Humanos leem/escrevem com facilidade,
- ❑ Computadores fazem o parsing com facilidade
- ❑ Objetos podem ser aninhados
- ❑ Baseados em:
 - ❑ pares nome-valor
 - ❑ listas de valores



BSON

- “Binary JSON”
- Objetos JSON serializados em formato binário: “Binary-encoded Serialization of JSON-like docs”
- Objetivos
 - Leve
 - Fluído
 - Codificação e decodificação eficientes





BSON Example

id	user_name	email	age	city
1	Mark Hanks	mark@abc.com	25	Los Angeles
2	Richard Peter	richard@abc.com	31	Dallas



```
{
  "_id": ObjectId("5146bb52d8524270060001f3"),
  "age": 25,
  "city": "Los Angeles",
  "email": "mark@abc.com",
  "user_name": "Mark Hanks"
}
{
  "_id": ObjectId("5146bb52d8524270060001f2"),
  "age": 31,
  "city": "Dallas",
  "email": "richard@abc.com",
  "user_name": "Richard Peter"
}
```





BSON Example

```
{
  "_id" : "37010"
  "city" : "ADAMS",
  "pop" : 2660,
  "state" : "TN",
  "congressmen": ["John", "Willian", "Adolf"]
  "mayor" : {
    name: "John Smith"
    address: "13 Scenic Way"
  }
}
```

← multi-valorado

← composto, um doc dentro de outro

□ Dados compostos e arranjos (arrays) em formato mais similar a linguagens de programação

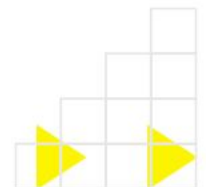


BSON Types

Type	Number
Double	1
String	2
Object	3
Array	4
Binary data	5
Object id	7
Boolean	8
Date	9
Null	10
Regular Expression	11
JavaScript	13
Symbol	14
JavaScript (with scope)	15
32-bit integer	16
Timestamp	17
64-bit integer	18
Min key	255
Max key	127

<http://docs.mongodb.org/manual/reference/bson-types/>

<https://docs.mongodb.com/manual/reference/operator/query/type/>

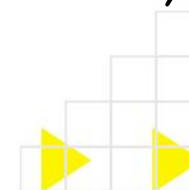


Usando o MongoDB



⇒ Instalação do MongoDB Community para Windows, Mac, e Linux:
<https://docs.mongodb.com/manual/administration/install-community/>

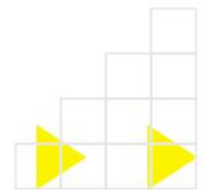
- Após a instalação, na linha de comando (terminal), digite: `mongo`
- Nome do database em uso: `db`
- Mostrar todos os databases: `show dbs`
- Criar/trocar de database: `use <name>`
- Mostrar as coleções existentes: `show collections`
- Criar coleção: `db.createCollection("<name>")`





O campo _id

- Cada documento possui um _id para tornar cada um deles único:
 - chave primária da coleção
 - imutável e de qualquer tipo, com exceção a arrays
 - tipo default ObjectId, o qual objetiva desempenho, e pouco armazenamento
- ⇒ A teoria de conjuntos também se aplica a NoSQL





O campo _id

Time = {nome, estado, tipo, saldo_gols}

- Valor default _id:

```
db.Time.insert({nome:"Aparecido", estado: "RN", tipo: "profissional", saldo_gols: 10})
```

- _id com valor próprio:

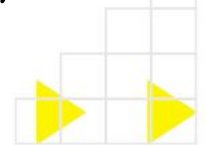
```
db.Time.insert({_id: 2, nome: "Marcos", estado: "PR", tipo: "amador", saldo_gols: 5})
```

Aqui, o _id é um documento

- _id com valor composto:

```
db.Time.insert({_id:{nome:"Leo",sobrenome:"Oliveira"}, estado: "RN", tipo: "profissional", saldo_gols: 10})
```

⇒ Listar os documentos: `db.Time.find()`





NoSQL MongoDB vs. Relacional

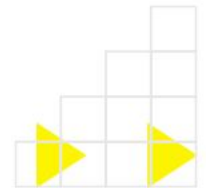
Documento/chave-valor	Relacional
Document	Tuple
Collection	Table/View
PK: _id Field	PK: Any Attribute(s)
Uniformity not Required	Uniform Relation Schema
Index	Index
Embedded Structure	Joins
Shard	Partition
CRUD	DML

Shards

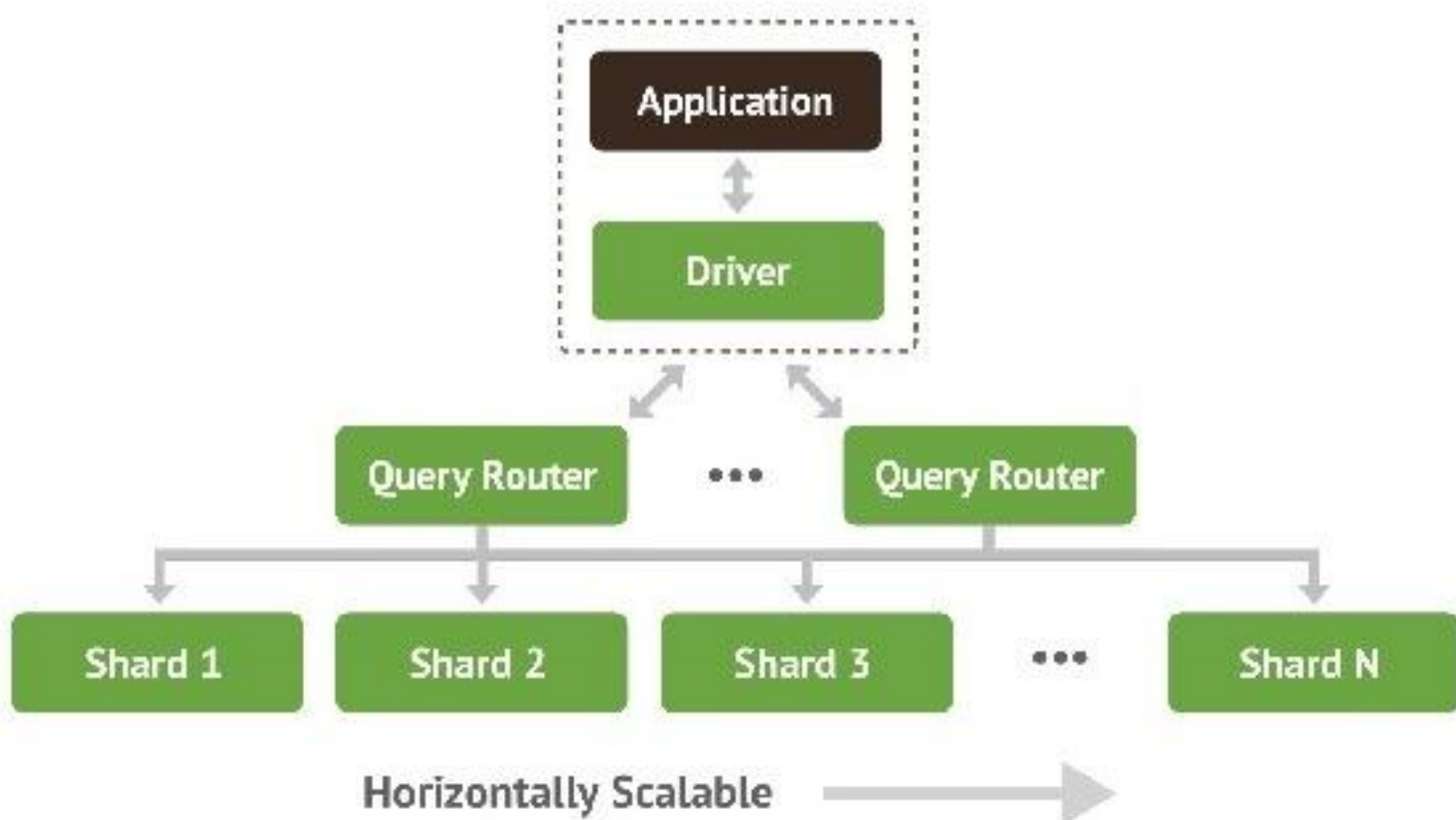


- Conceito lógico de armazenamento;
- Um shard (ou fragmento) equivale a um pedaço dos dados (de uma coleção, por exemplo) que é replicado em vários servidores;
- Os vários servidores passam a ser nós de replicação e processamento;
- Os shards seguem à “shard key”; por exemplo, se há muitas consultas baseadas no atributo nome, usar este atributo como shard key pode melhorar o desempenho;
- Permite:
 - escalabilidade horizontal facilitada - mais servidores;
 - replicação;
 - processamento distribuído.

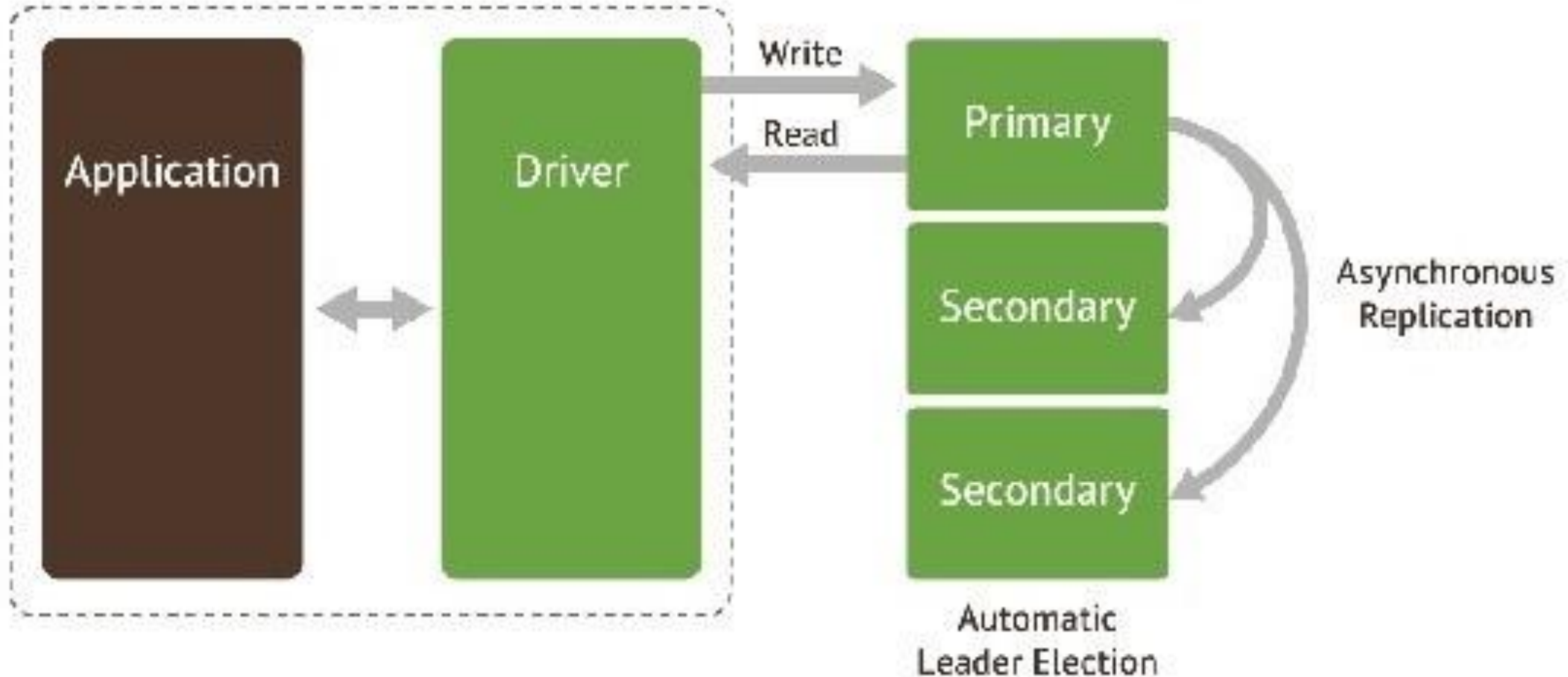
⇒ [MongoDB sharding](#)



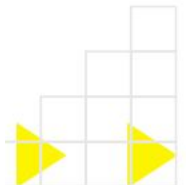
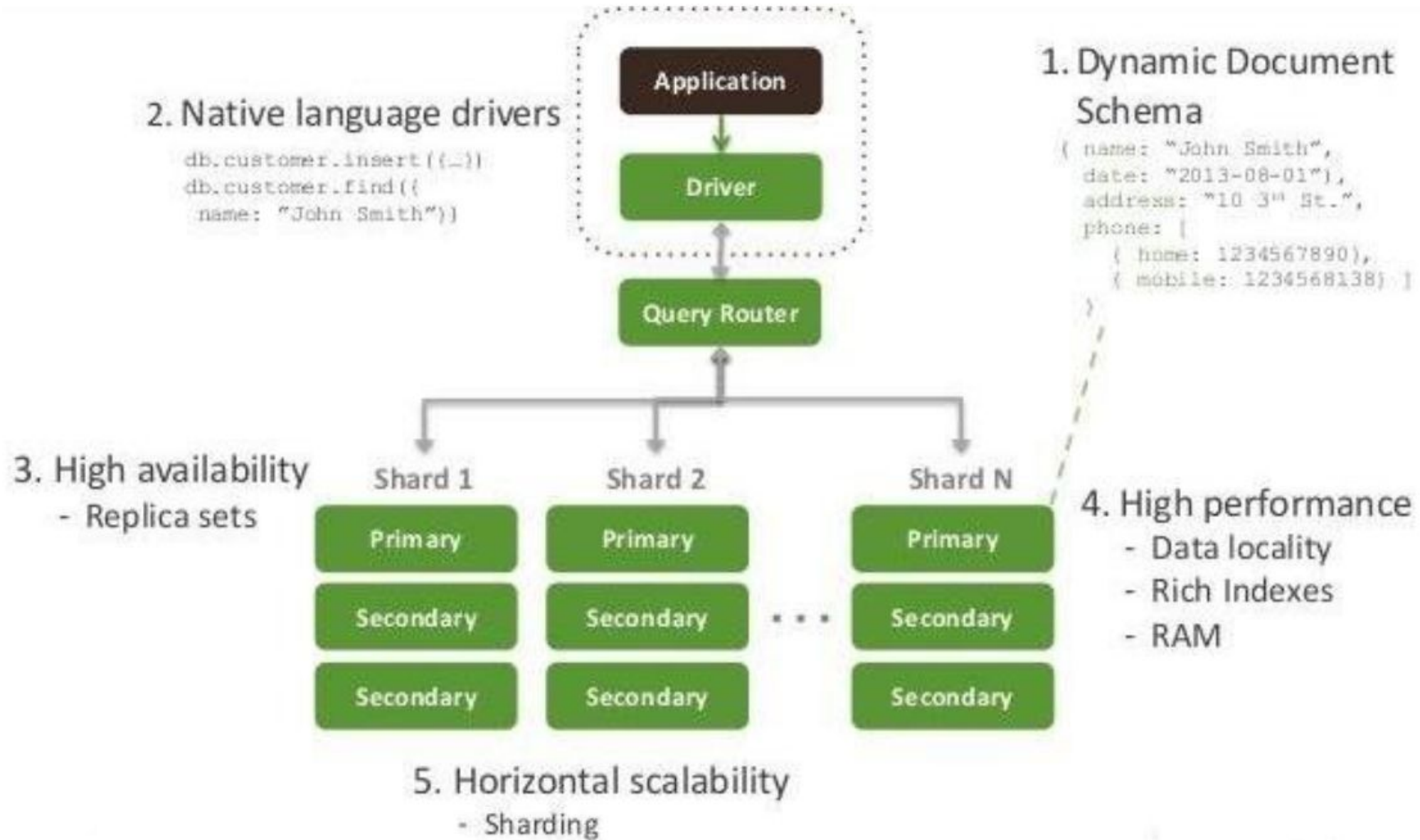
Horizontal scaling



Easy replication



Overall architecture



Mongodb: The Definitive Guide: Powerful and Scalable Data Storag,

By Kristina Chodorow and Mike Dirolf

Published: 9/24/2019

Pages: 514

Language: English

Publisher: O'Reilly Media, CA

