mongoDB

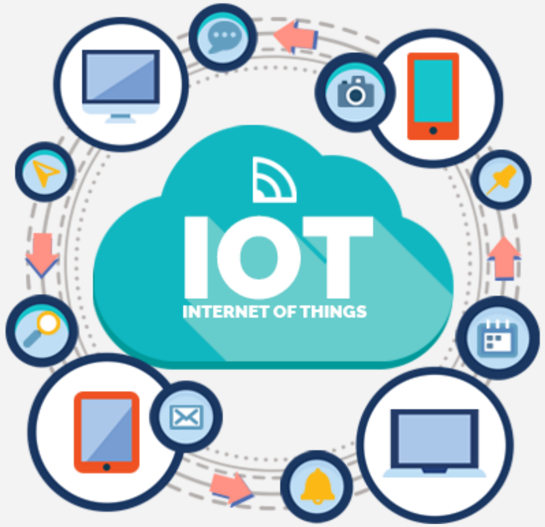
NoSQL

Introduction

By Eman Fathi

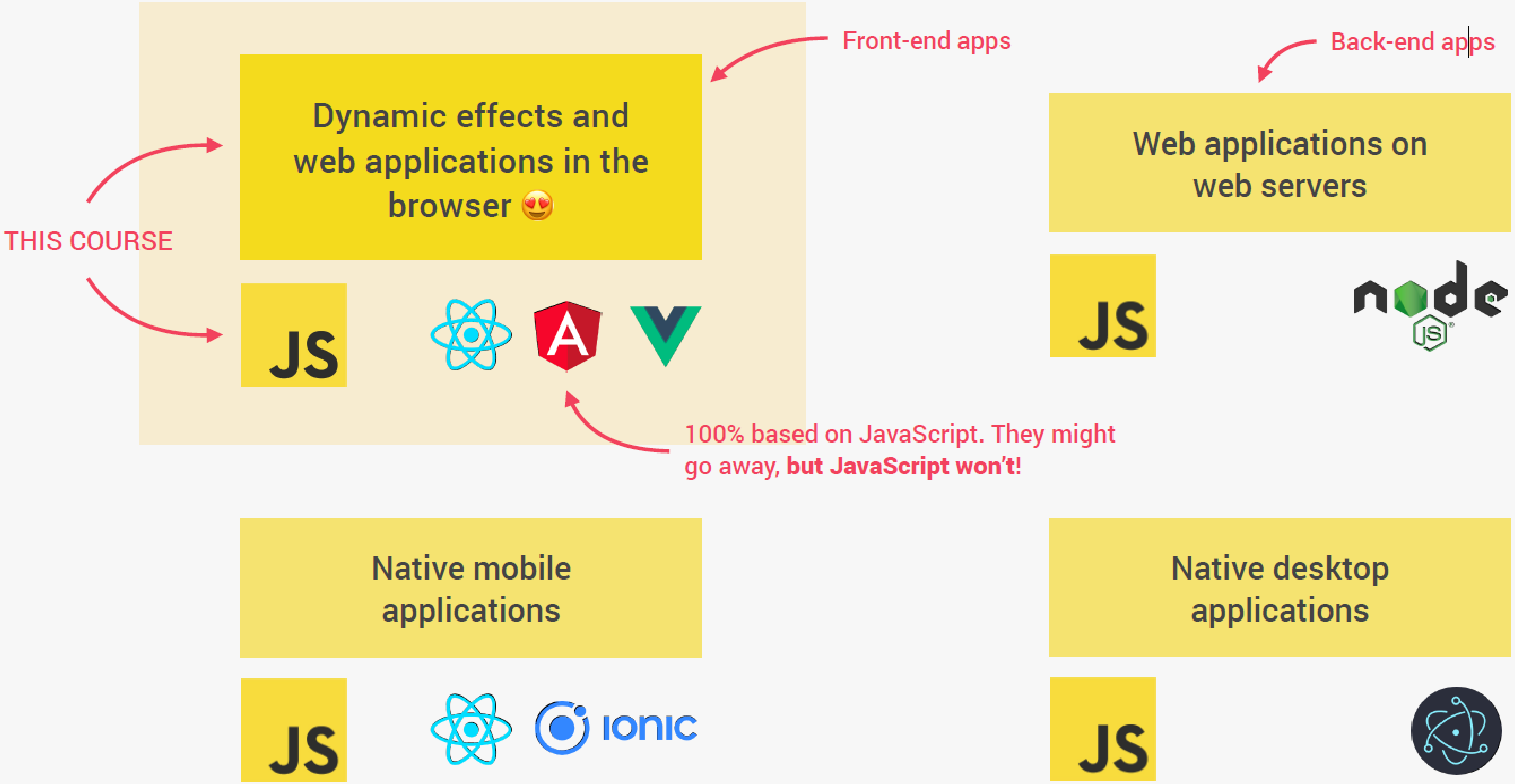
MACHINE
LEARNING

EVERYWHERE


IOT
INTERNET OF THINGS

JS
node
METEOR
ember
polymer
BACKBONE.JS
Knockout.
aurelia
ext js

THERE IS NOTHING YOU CAN'T DO WITH JAVASCRIPT (WELL, ALMOST...)



Front-end apps

Dynamic effects and web applications in the browser 😍

THIS COURSE

JS

100% based on JavaScript. They might go away, **but JavaScript won't!**

Back-end apps

Web applications on web servers

JS

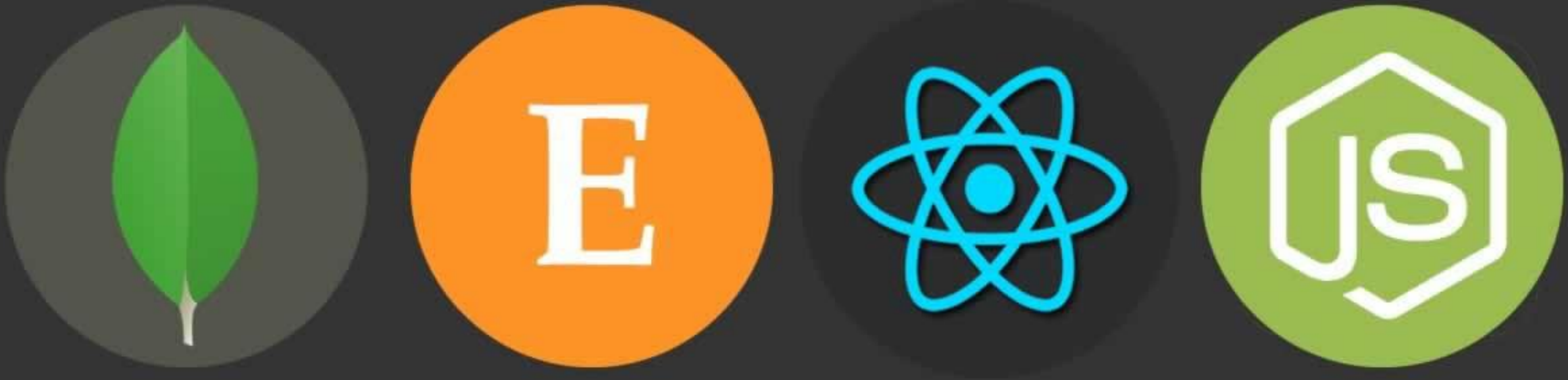
Native mobile applications

JS

Native desktop applications

JS

We need to turn them into developers who think of applications like this.



M E R N

What is Mongo DB ?

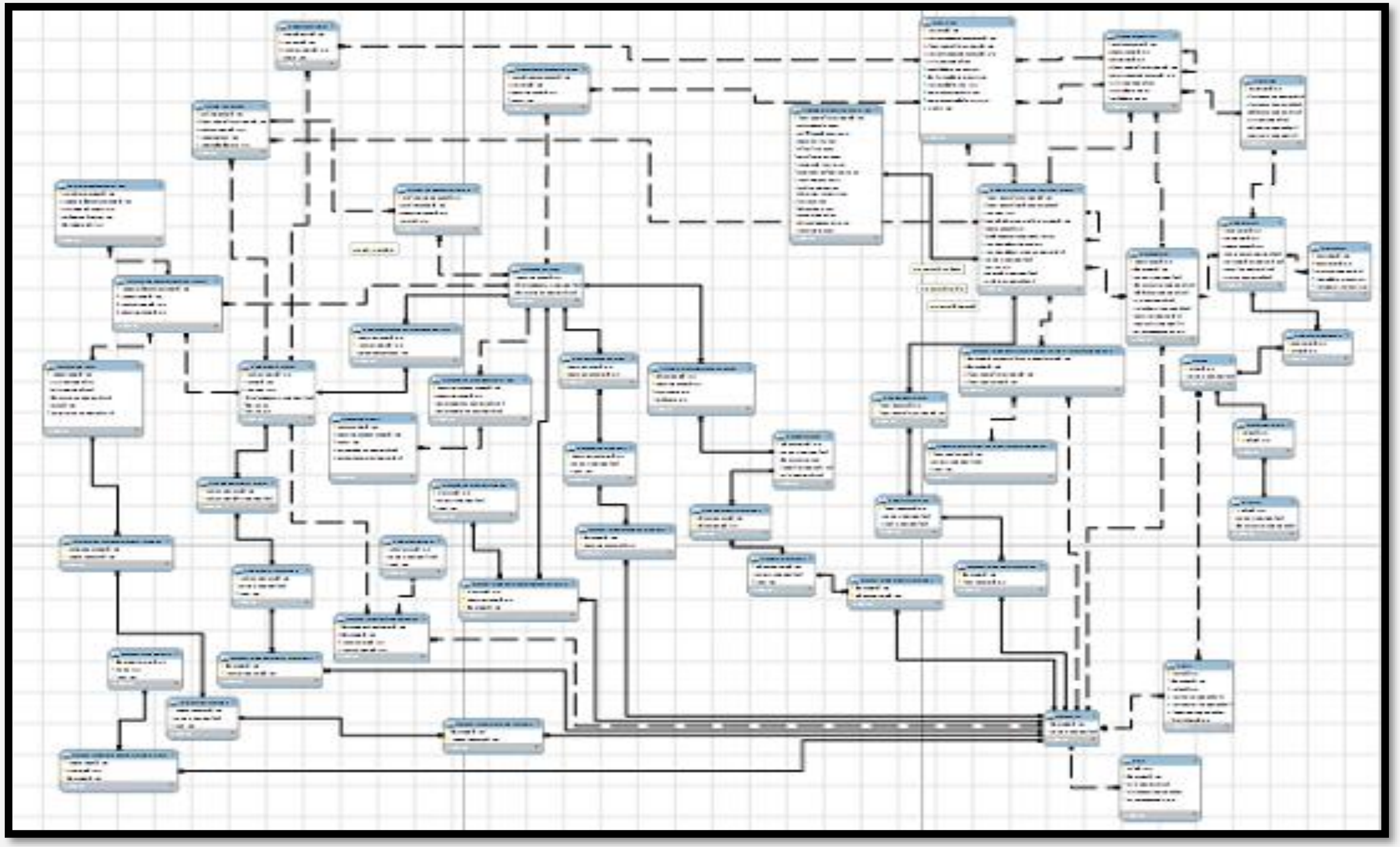
Hu*mongo*us

It can stores lots and lots of Data

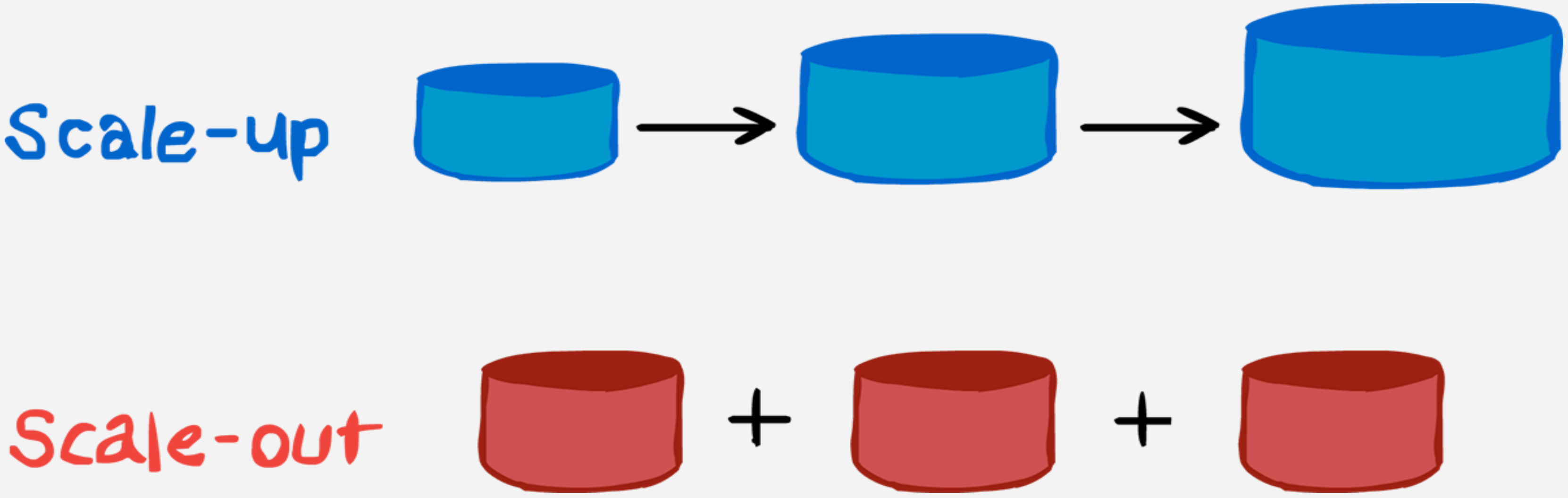
CONCEPT OF NOSQL DATABASES GREW WITH INTERNET GIANTS

Gigantic volume of data

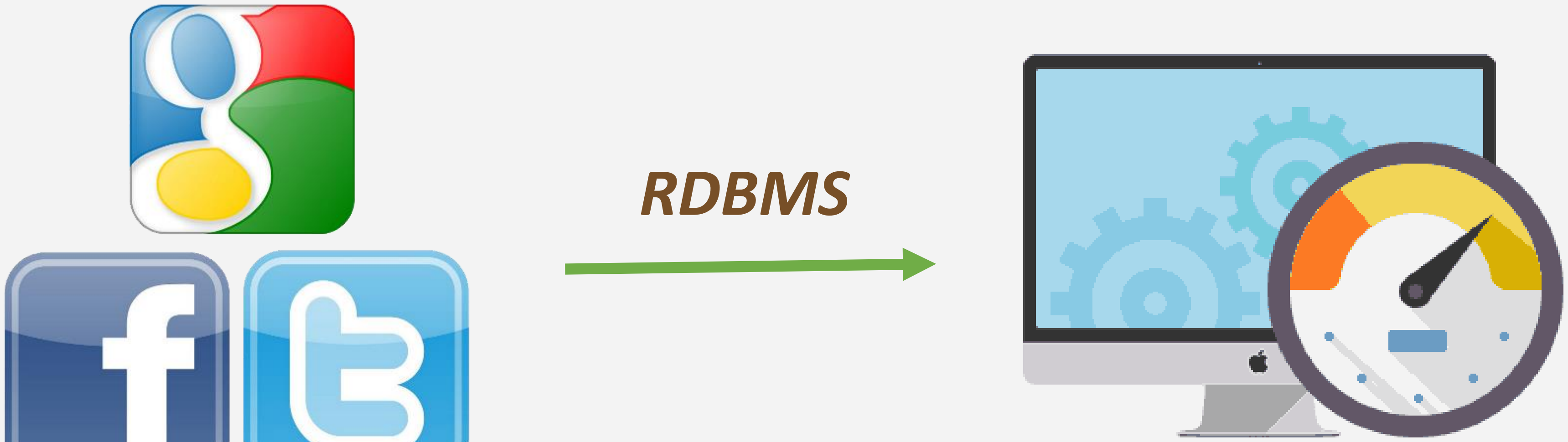
CONCEPT OF NOSQL DATABASES GREW WITH INTERNET GIANTS



CONCEPT OF NOSQL DATABASES GREW WITH INTERNET GIANTS



CONCEPT OF NOSQL DATABASES GREW WITH INTERNET GIANTS



Gigantic volume of data

RDBMS

Slow Response Time

APACHE HBASE

cassandra

Neo4j

mongoDB

CouchDB

redis

NoSQL

Not Only SQL

Key Value

redis

Usage:
Briskly changing data and high availability

Popular DBs:
Riak, Redis

1

Column Based

cassandra

Usage:
Read/write extensions

Popular DBs:
HBase,
Cassandra

2

Document Database

mongoDB

Usage:
Working with occasionally changing consistent data

Popular DBs:
Couchbase,
MongoDB

3

Graph Database

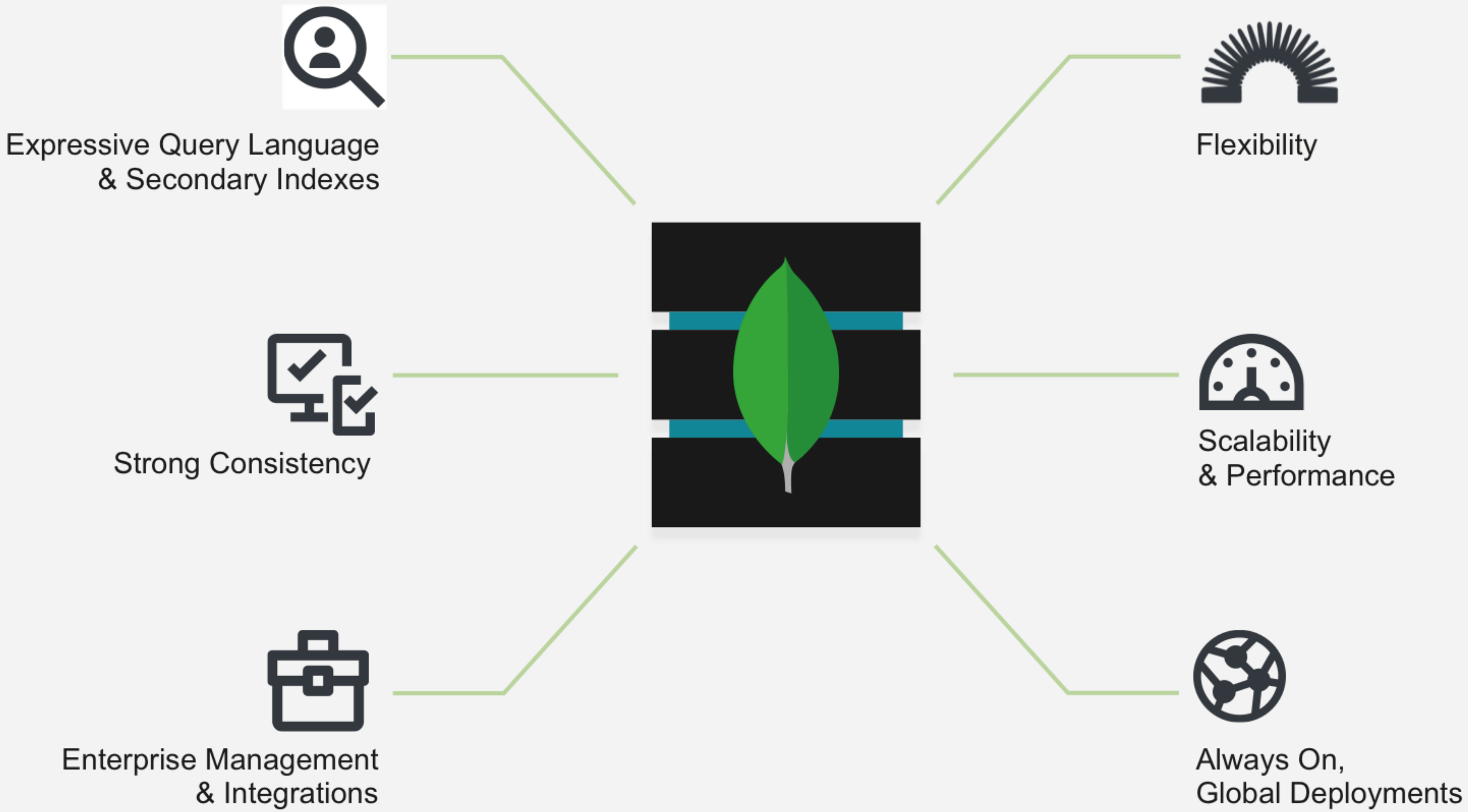
neo4j

Usage:
Spatial data storage

Popular DBs:
Neo4J,
Big data

4

NoSQL Features



Expressive Query Language
& Secondary Indexes

Strong Consistency

Enterprise Management
& Integrations

Flexibility

Scalability
& Performance

Always On,
Global Deployments

mongoDB

Humongouse

Produced by 10gen company In 2007. In 2013 10gen renamed itself to MongoDB.

10gen

MongoDB is a **cross-platform**, document oriented database that provides, **high performance**, **high availability**, and **easy scalability**. MongoDB works on concept of collection and document.

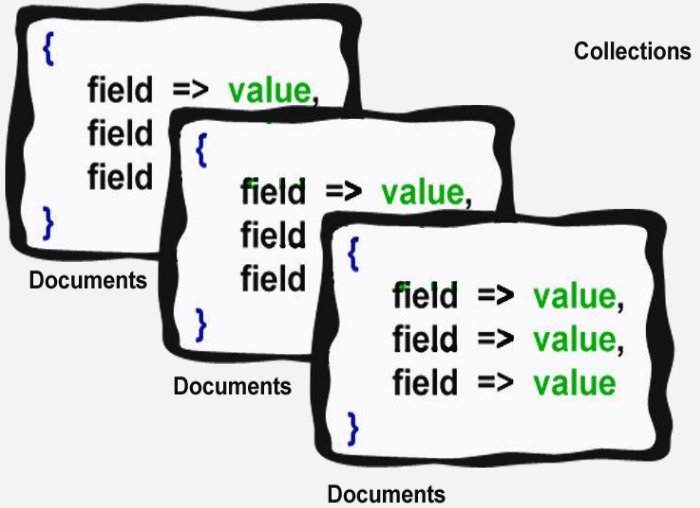
MongoDB is a database management system designed to rapidly develop web applications and internet infrastructure. The data model and persistence strategies are built for high read-and-write throughput and the ability to scale easily with automatic failover. Whether an application requires just one database node or dozens of them, MongoDB can provide surprisingly good performance.

mongoDB

Humongouse

Collections

Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table .Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.

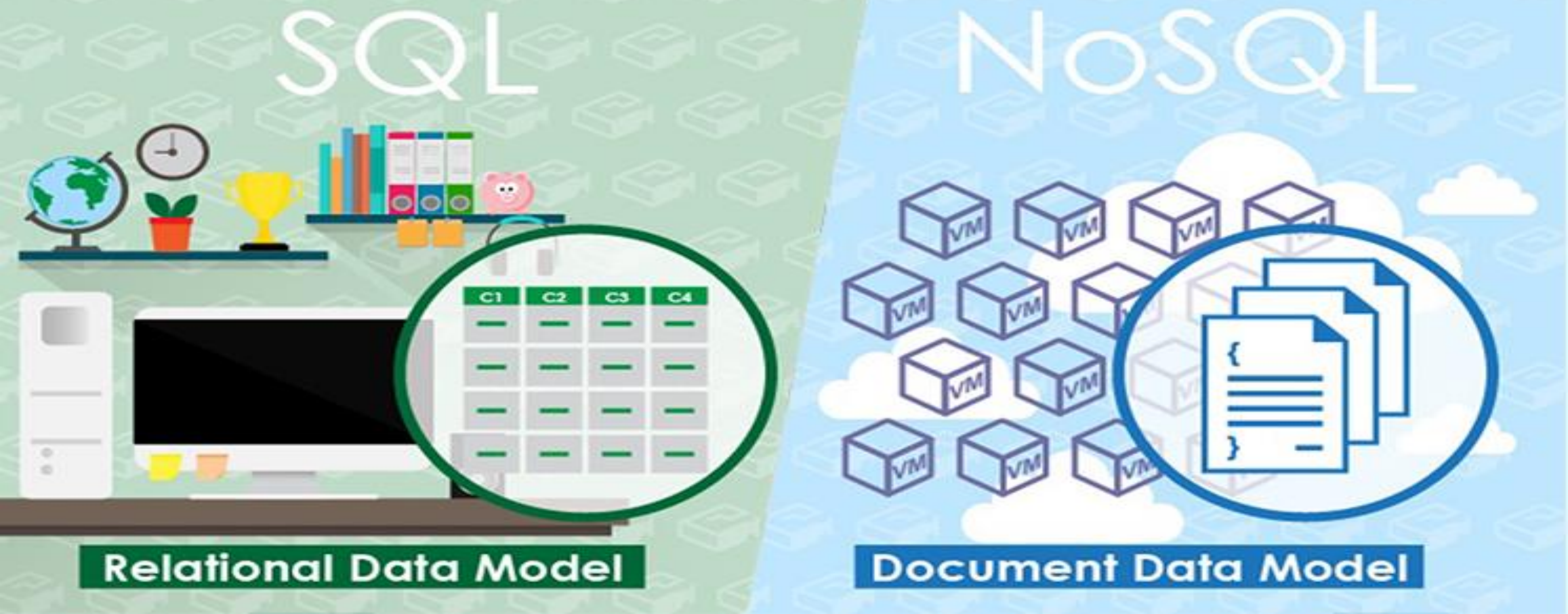


Documents

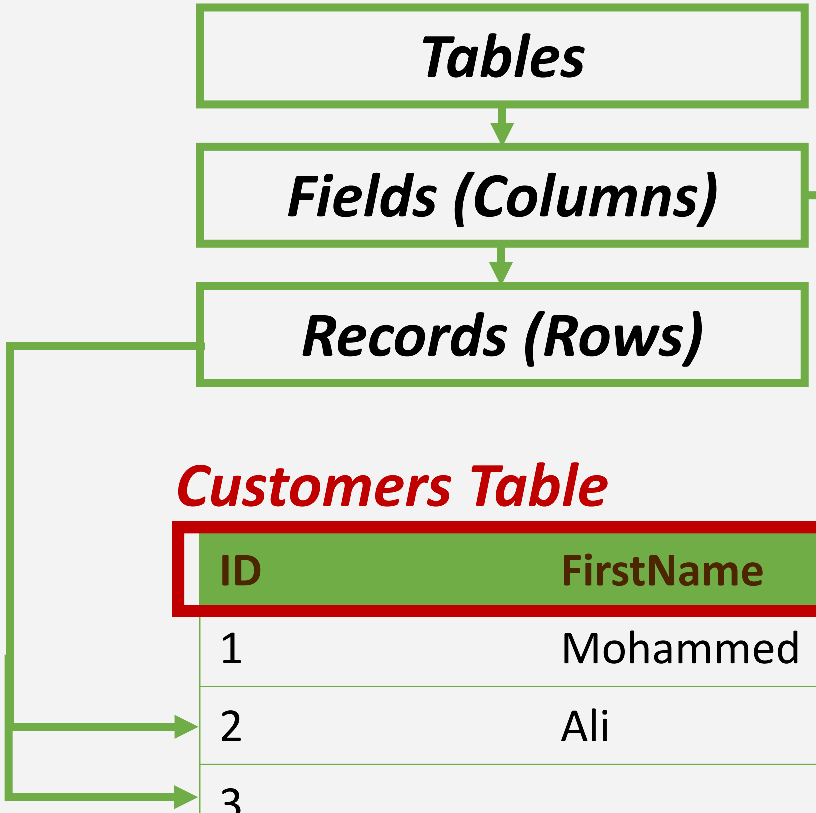
A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

NoSQL

Not only SQL



SQL (Structure Query Language)



Tables

Fields (Columns)

Records (Rows)

Microsoft SQL Server

Shopping DB

Customers Table

ID	FirstName	LastName	Address	Gender
1	Mohammed	Khaled	…..	M
2	Ali	Ahmed	………	M
3				M
4	….	….	….	…

Schema

SQL (Structure Query Language)

Relations

Customers

ID	*FirstName*	*LastName*	*Address*	*Gender*
1	Ali	Ahmed	……..	M
2	Hana	Mohamed	……	F
3				M

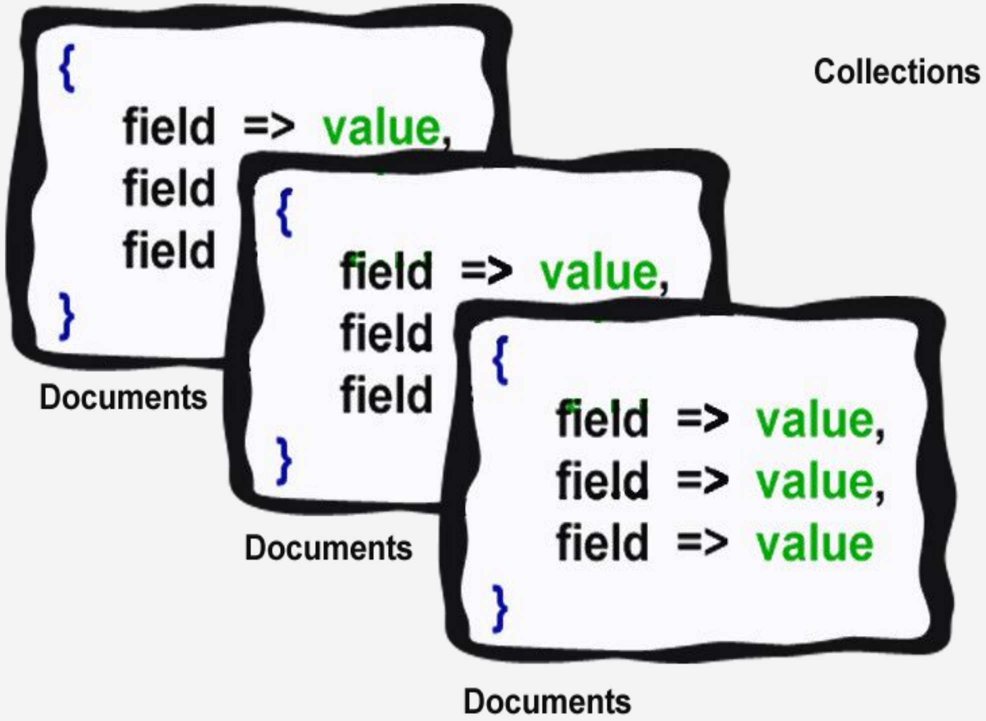
Orders

ID	*CustomerId*	*ProductId*
1	1	1
2	1	2
3	2	1

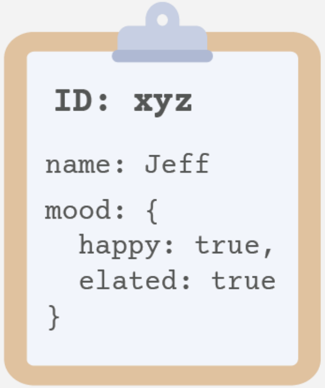
Products

ID	*Price*	*Description*	*Quantity*
1	20.05	……	…….
2	597	…….	……
3	342.8	…….	……..

NoSQL (Not Only SQL)



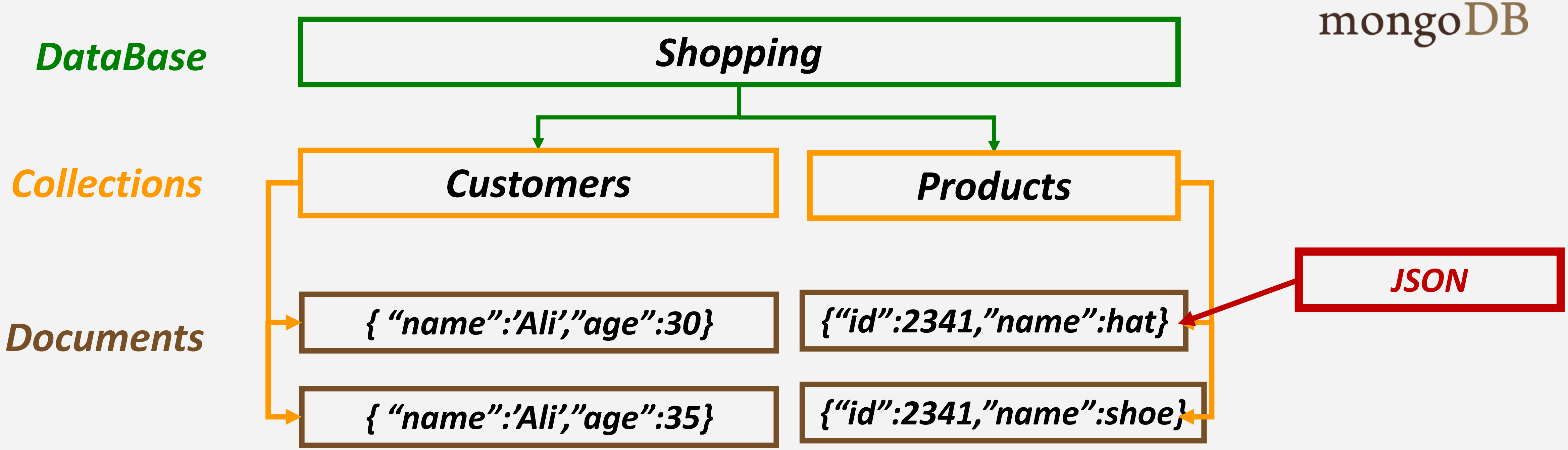
mongoDB



```
ID: xyz
name: Jeff
mood: {
  happy: true,
  elated: true
}
```

NoSQL (Not Only SQL)

mongoDB



DataBase

Shopping

Collections

Customers

Products

Documents

{ “name”:’Ali’,”age”:30}

{ “name”:’Ali’,”age”:35}

{“id”:2341,”name”:hat}

{“id”:2341,”name”:shoe}

JSON

NoSQL (Not Only SQL)

No Schema

Documents

{id:1 name:’Ali’,age:30}

{id:2 name:’Ahmed’,age:35}

{id:4 name:’Khaled’,email:Khaled@gmail.com}

NoSQL (Not Only SQL)

No/Few Relations

Customers

{id:1 name:’Ali’,age:30}

{id:2 name:’Khaled’,age:35}

{id:3 name:’Lara’,age:30,email:lara@gmail.com}

{……………}

Products

{id:1,title:’hat’,price:120}

{id:2,title:’shoe’,price:320}

{…………………}

Orders

{id:14 , customer : {id:3,email:lara@gmail.com} , product : 3}

{id:15 , customer : {id:2,name:”Khaled”} , product : 3}

{………………………}

SQl Server	MongoDB
Tables	Collections
Rows	Documents
coulmns	Fields
Joins	Embeded Documents
Primary Keys	PrimaryKey providd by mongodb itself

SQl

Data Schemas

Data distributed across multiple tablesd

Relations

Horzontal scalling is difficult, vertical is possible

Limitations for huge numbers of read/write queries /second

NoSQL

Schemas less

Data merged in few collections

No/few Relations

Both horizontal & vertical scalling are possible

Great performance for huge read/write requests https://www.mongodb.com/

Community Server

```
> show dbs;
> Show collections;
> db.getUsers;
```

Installation

Running Through Mongo Shell
All instances of MongoDB come with a command line program we can use to interact with our database using Javascript

MongoDB server (mongod)

After Installation

mongod is the basic process for mongoDB system. It handles data requests ,manages data access and performs background management operations.

By default it listens to port 27017.

mongod runs with some options as:
--dbpath
--port
--maxconns

Documents are JSON-Like Objects

{JSON}
BSON
01010100
11101011
10101110
01010101

MongoDB save Data in documents in a format called BSON

Binary JSON

MongoDB BSON

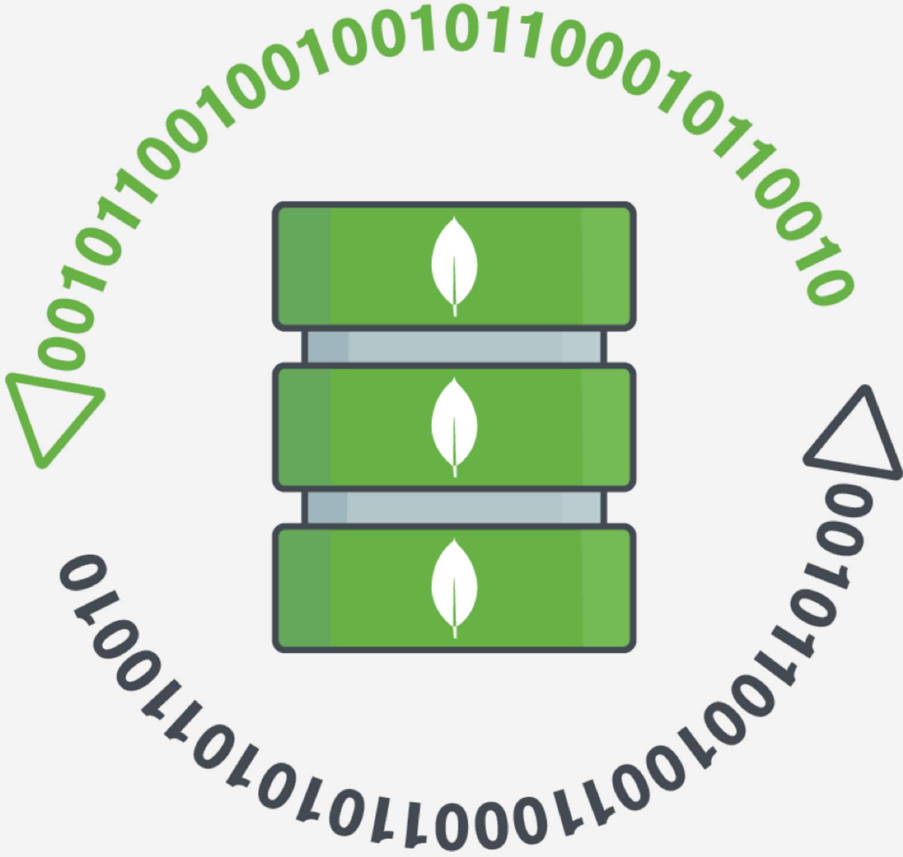
How mongodb respresents data and how mongo shell interpret that data comming from database?

Mongogdb does not use stringly format for storing or retrieving data.

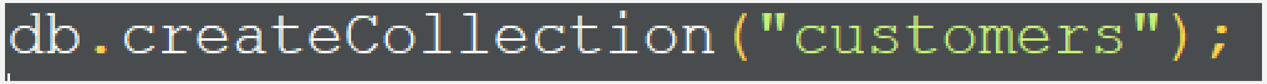
Instead it uses binary representation to store data inside documents BSON

BSON is a binary representation for json and support Data types not in JSON like BinData , ObjectID and Timestamp.

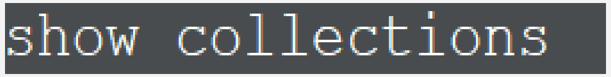
bsonspec.org



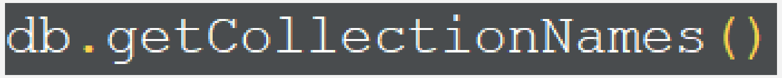
Collections



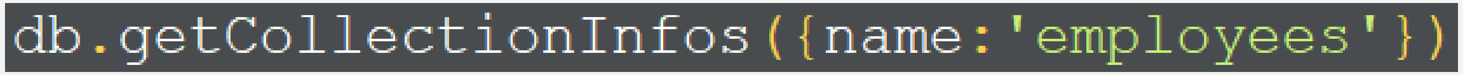
```
db.createCollection("customers");
```



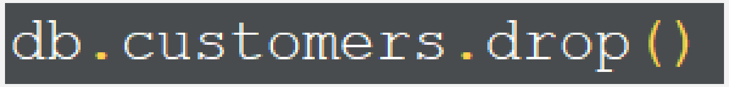
```
show collections
```



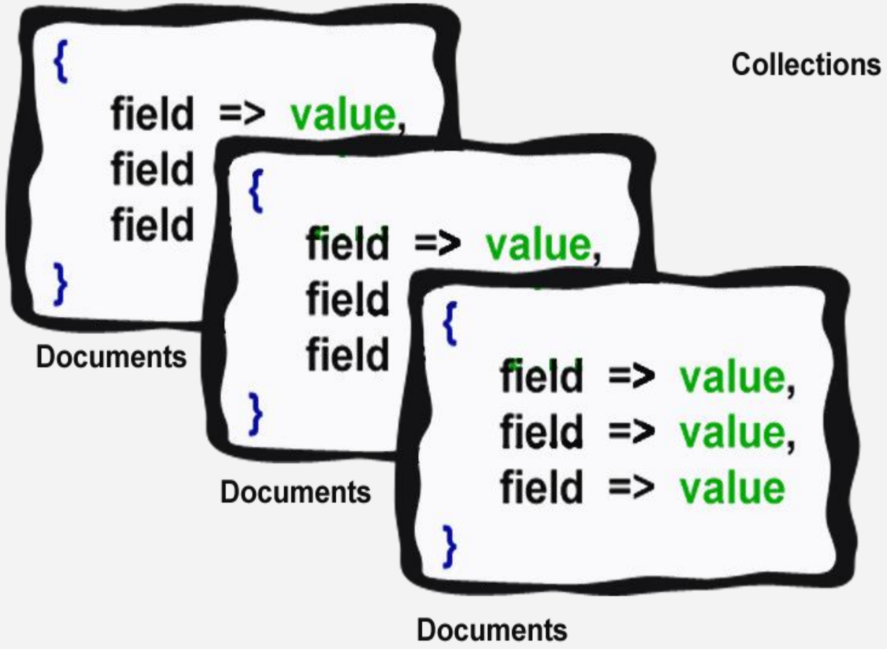
```
db.getCollectionNames()
```



```
db.getCollectionInfos({name:'employees'})
```



```
db.customers.drop()
```



CREATE	READ	UPDATE	DELETE
C	R	U	D

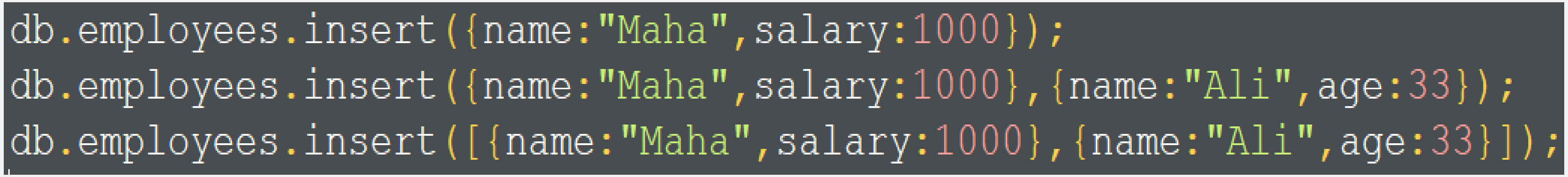
CREATE

db.collection.insert();

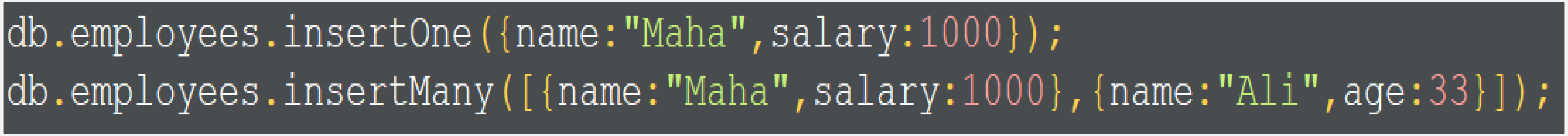
insert(): insert document or documents into a collection

insertOne(): insert only one document into a collection

insertMany(): insert multiple documents into a collection



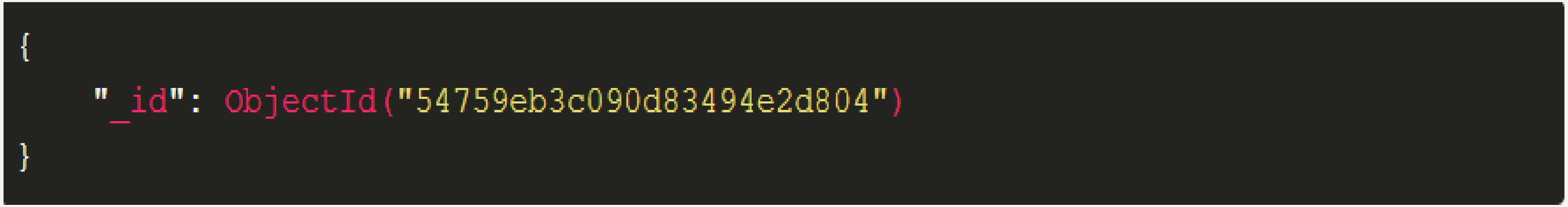
```
db.employees.insert({name:"Maha",salary:1000});
db.employees.insert({name:"Maha",salary:1000},{name:"Ali",age:33});
db.employees.insert([{name:"Maha",salary:1000},{name:"Ali",age:33}]);
```



```
db.employees.insertOne({name:"Maha",salary:1000});
db.employees.insertMany([{name:"Maha",salary:1000},{name:"Ali",age:33}]);
```

ObjectId

The ObjectId Class is the default primary key for a MongoDB document and is found in _id field in an inserted document.



```
{
    "_id": ObjectId("54759eb3c090d83494e2d804")
}
```

✓Immutable

✓Unique

✓BSON DataType

✓12 byte Value