Introduction

General Knowledge in MongoDB

Database Terminology

- Database
- RDBMS
- SQL

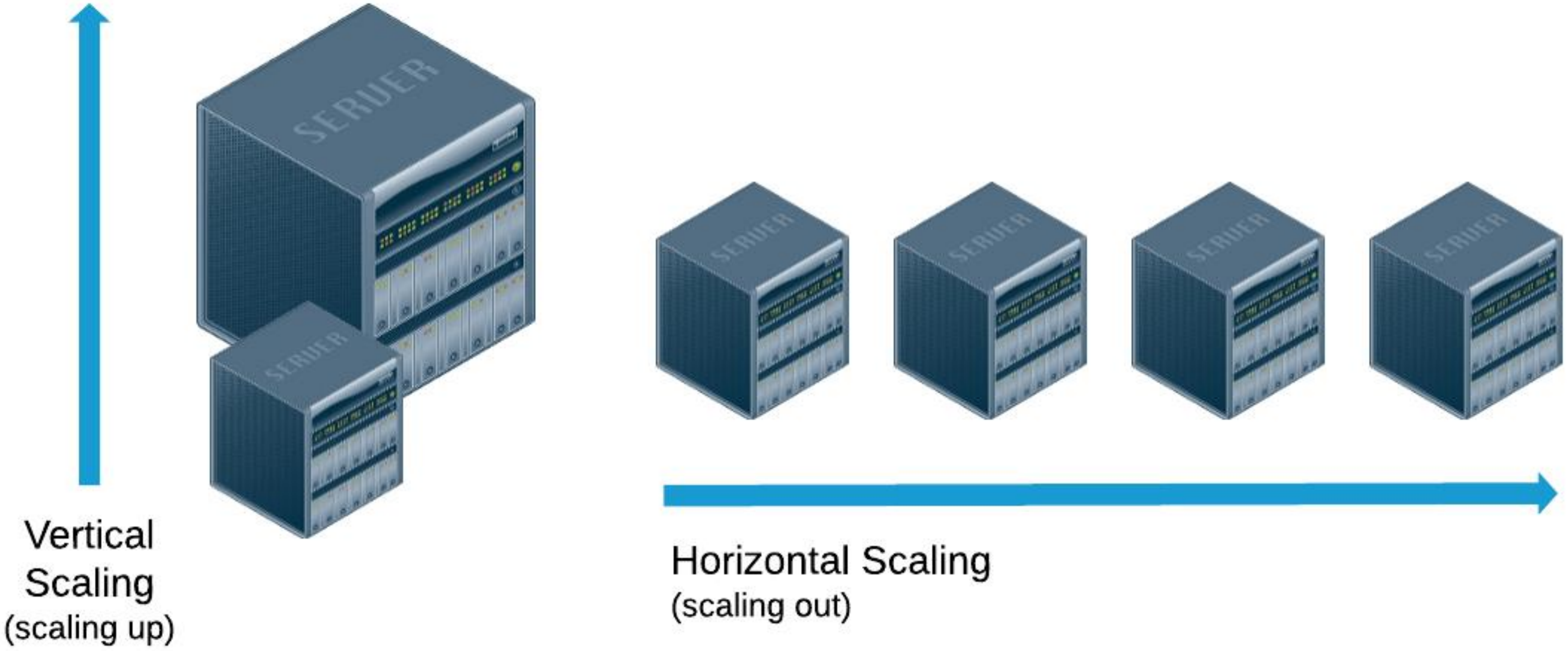
Introduction to NoSQL

- The term **NoSQL** came from the word **non SQL** or **nonrelational**
- Flexible Schema
- Employed for managing the massive collection of unstructured data and when your data is not piled up in a tabular format or relations like that of relational databases.
- unstructured data are haphazard data formats (such as document files, image files, video files, icons, etc)

SQL vs. NoSQL

	SQL	NoSQL
Scalability	Scaling up requires bigger load , bigger servers – Vertically Scalable (Scale up)	Scaling is elastic, and done effortlessly – Mainly horizontally scalable (Scale out)
Cost	More expensive – relies on expensive Servers	Less expensive – relies on a cluster of cheap commodity servers
Relations	RDBMS	Non-relational DB
Structure	Data Structured in a proper format. Models are inflexible, have to be careful when changing structure.	Data is non-structured. Structure is more relaxed and flexible.
DBA	Requires high trained experists	Requires less management, automatic repairs
Schema	Schema based DB	Schema-less or flexible schema. Documents do not have to be similar.

SCALING UP VS. SCALING OUT



Type of NoSQL DB

- While SQL DB are table-based, NoSQL DBs have different type based on the stored data.
- Types of NoSQL DBS
  - **Key-value stores -** Here, each unstructured data is stored with a key for recognizing it. Ex: Memcache DB
  - **Graph stores:** data is stored mostly for networked data, It helps to relate data based on some existing data.
  - **Document-oriented stores** - Here, the key gets paired with a compound data structure. Ex:MongoDB, CouchDB
  - **Column family stores (wide-column stores)** – This type of data stores large data sets. Ex: Cassandra, BigTable

MongoDB

- It is also an open-source, a **document-oriented**, cross-platform database system that is written using C++.
- MongoDB is based on a NoSQL database that is used for storing data in a key-value pair.
- Its working is based on the concept of **document and collection.**
- it’s a **server-client** DB, where server runs with binary file **mongod**, and the client runs with **mongo**
- data is stored in a Binary **JSON-like** format called **BSON**.
- MongoDB does not provide SQL support.

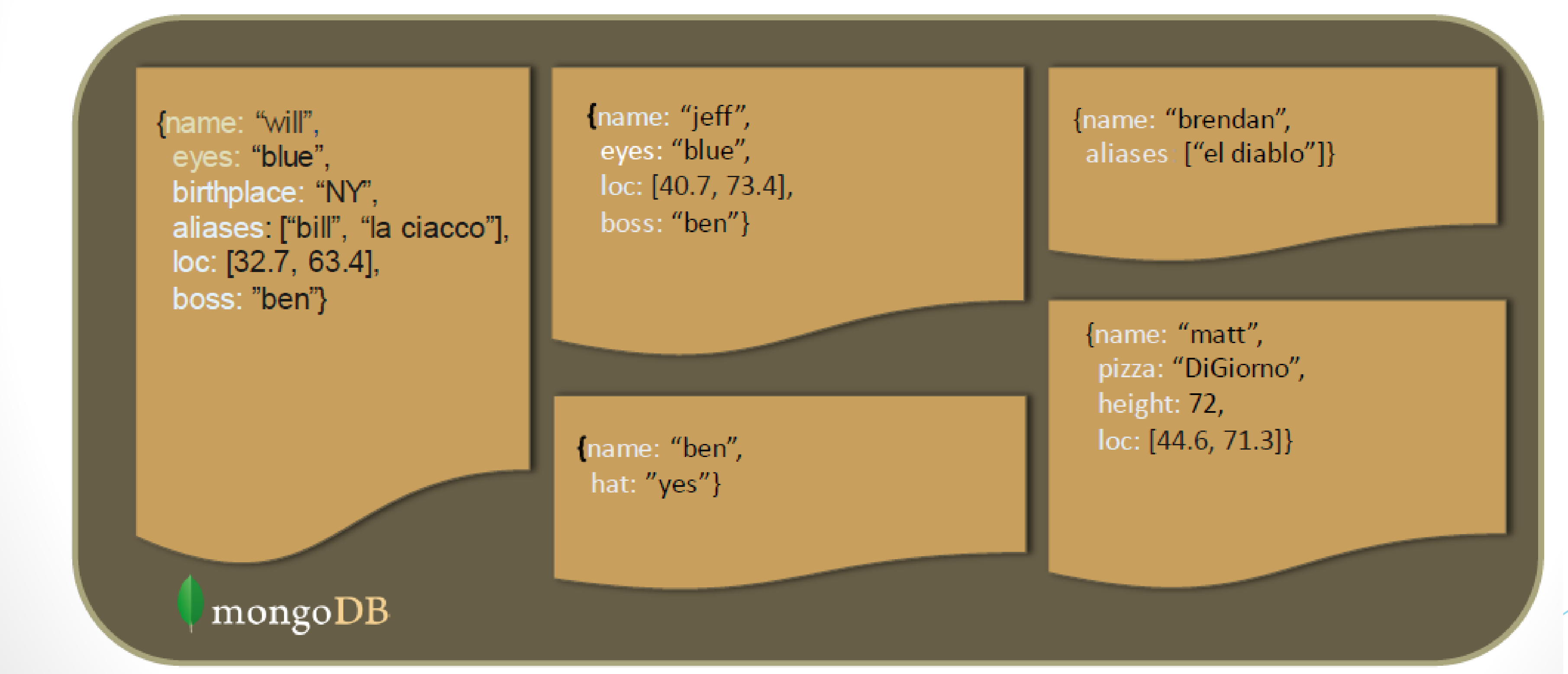
MongoDB Collections

- **Collections** here replace **tables** in the RDBMS.
- **Collections** can be defined as a cluster of MongoDB documents that exist within a single database
- Each collection consists of a group of documents, that can have different structure.
- It has no concept of joins, joins are achieved functionally through Aggregation.

MongoDB Documents

- **Documents** here replace rows in a RDBMS.
- A document can be defined as a collection of key-value pairs that contain **dynamic schema**.
- Each document has a unique value key "_id"
- Documents in MongoDB can hold any data type that is valid in MongoDB
- **Dynamic schema** is something that documents of the equal collection do not require for having the same collection of fields or construction, and a common field is capable of holding various types of data.

Schema free



```
{name: “will”,
  eyes: “blue”,
  birthplace: “NY”,
  aliases: [“bill”, “la ciacco”],
  loc: [32.7, 63.4],
  boss: ”ben”}
```

```
{name: “jeff”,
  eyes: “blue”,
  loc: [40.7, 73.4],
  boss: “ben”}
```

```
{name: “brendan”,
  aliases: [“el diablo”]}
```

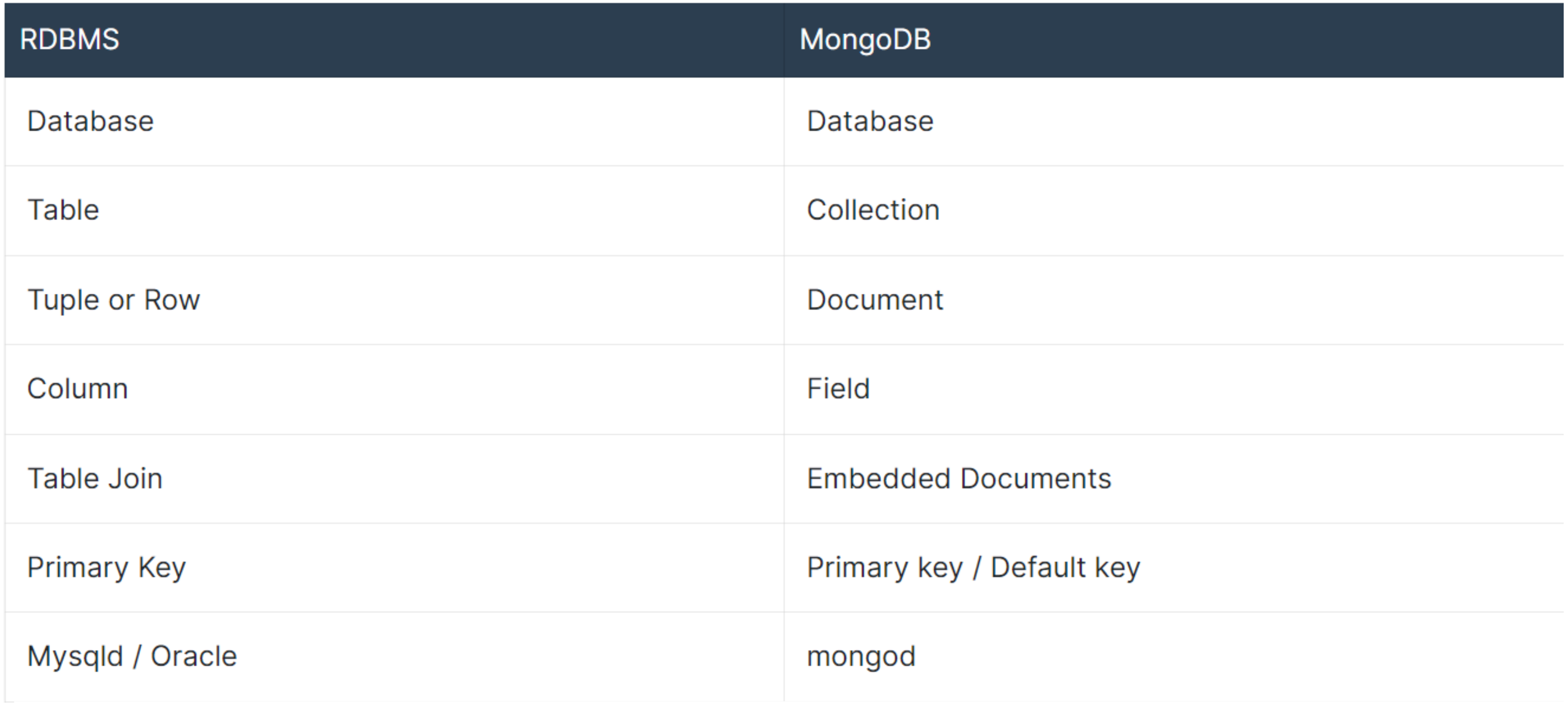
```
{name: “ben”,
  hat: ”yes”}
```

```
{name: “matt”,
  pizza: “DiGiorno”,
  height: 72,
  loc: [44.6, 71.3]}
```

Basically

- Group of documents → Collections
- Group of collections → Database

Related terminology



RDBMS	MongoDB
Database	Database
Table	Collection
Tuple or Row	Document
Column	Field
Table Join	Embedded Documents
Primary Key	Primary key / Default key
Mysqld / Oracle	mongod
Foreign key	**Reference**

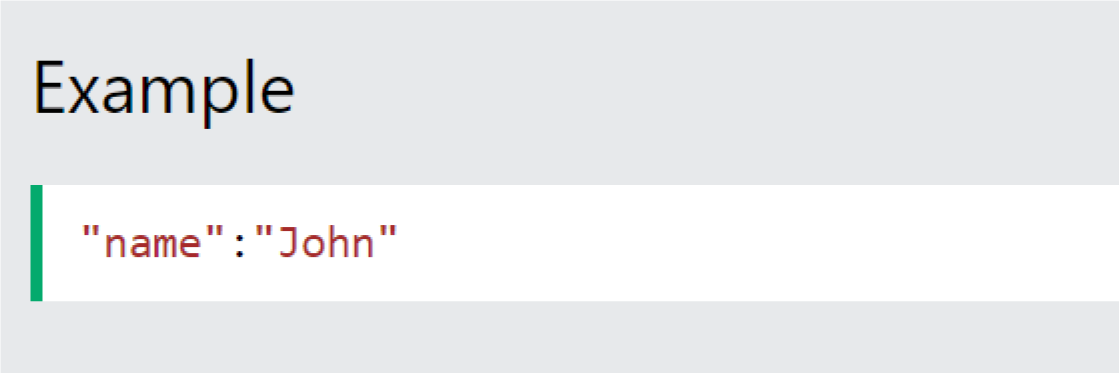
Advantages of using MongoDB

- Easy to install
- Since MongoDB is a schema-less database, so there is no hassle of schema migration.
- Easily scalable.
- MongoDB also supports the searching using the concept of regex (regular expression) as well as fields.
- It does not require any VM to run on different platforms.

JSON format

- JavaScript Object Notation is **a standard text-based format for representing structured data based on JavaScript object syntax.**
- It is commonly used for transmitting data in web applications.
- JSON is "self-describing" and easy to understand (human readable)

JSON Syntax Rules

- Data is in name/value pairs (key/value)
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays
- **JSON names require double quotes.**
- 

Example

```
"name":"John"
```

JSON values

- In **JSON**, *values* must be one of the following data types:
- a string
- a number
- an object
- an array
- a boolean
- null

BSON

- Mongo DB data type used to
  - Process and store data
- Binary-encoded JSON value
- It has some extended data type that are not supported by JSON
  - Date
  - Timestamp
  - Object ID
- Faster performance and data retrieval and insert

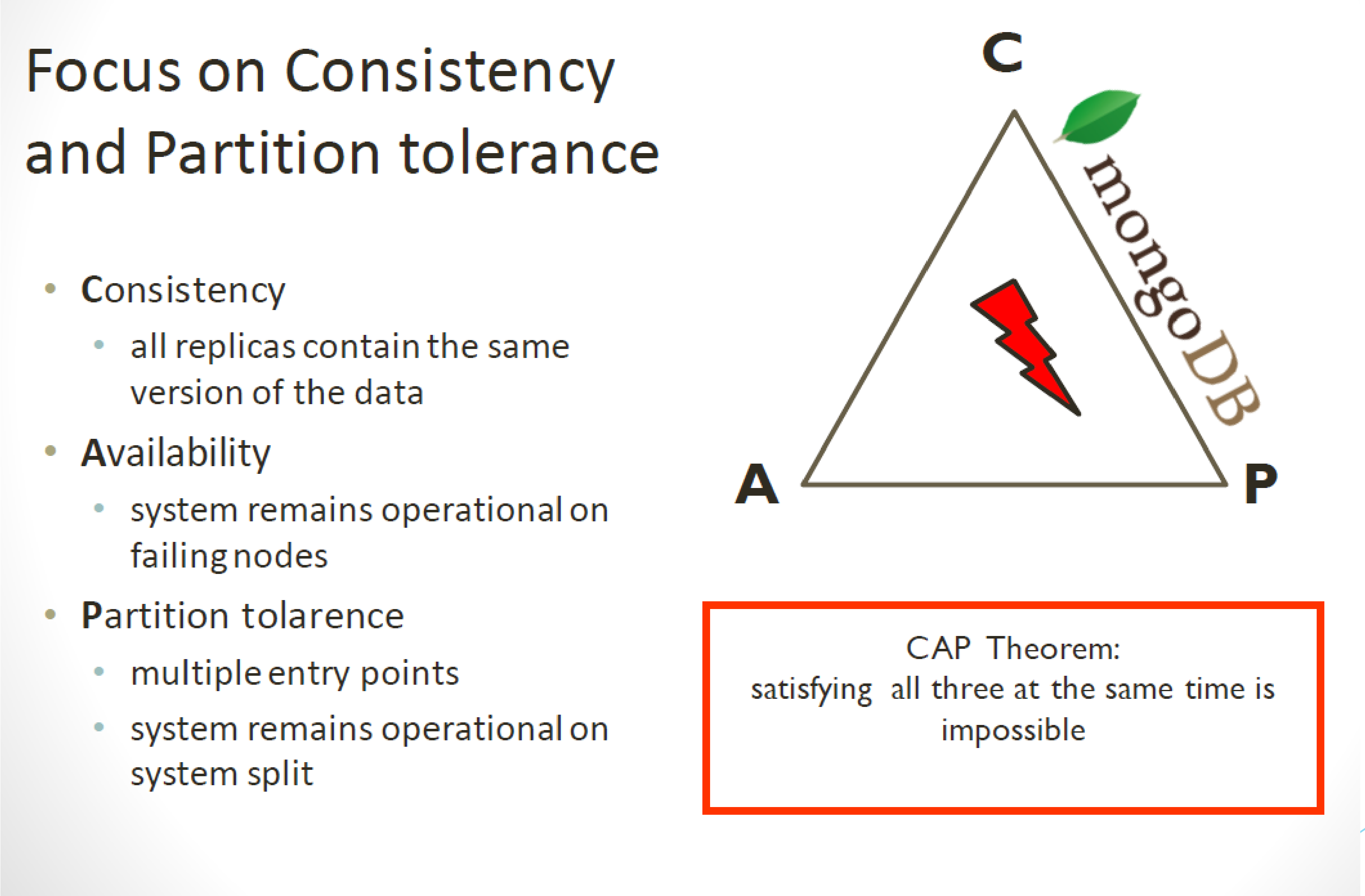
In application

- MongoDB can be used in
  - CMS
  - Gaming applications
  - E-commerce Systems
  - Mobile applications
  - IOT
  - Real-time analyzation
  - MEAN stack, MERN stack

Companies that implement MongoDB

- Cisco
- Adobe
- Astra Zeneca
- Toyota

CAP Approach



Focus on Consistency and Partition tolerance

- Consistency
  - all replicas contain the same version of the data
- Availability
  - system remains operational on failing nodes
- Partition tolarence
  - multiple entry points
  - system remains operational on system split

C
A
P
mongoDB

CAP Theorem:
satisfying all three at the same time is impossible

CAP theorem

- CA → systems have problems with partition, deals with it usually by replicas

Ex : RDBMS:- MySQL.

- CP → Systems has problem with availability, while keeping data consistent across all nodes, ex: MongoDB, BigTable
- AP→ achieves “eventual consistency”, ex: CouchDB, Cassandra.