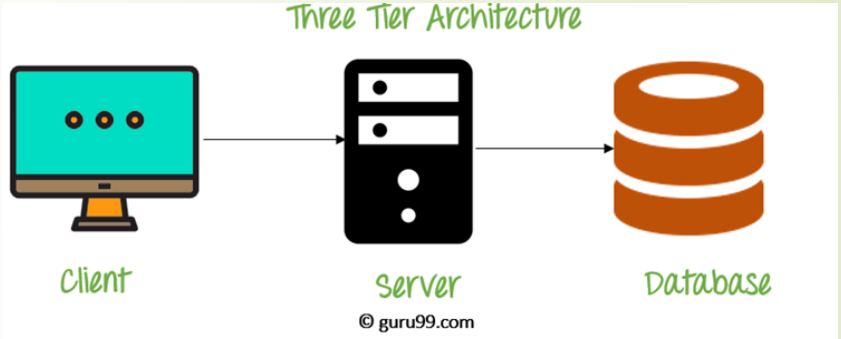
Mongodb

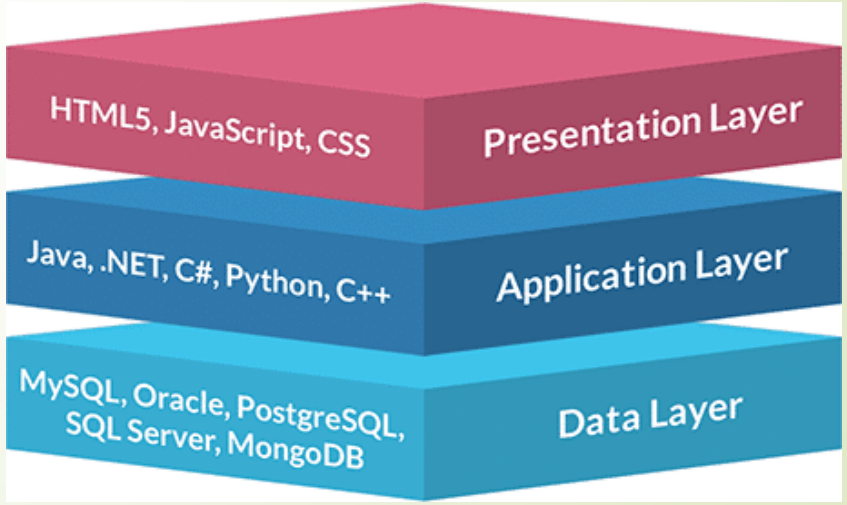
content

- Three Tier Architecture
- Data Layer
- SQL Vs NoSQL
- NoSQL Types
- mongodb

Three Tier Architecture

- Presentation Layer.
- Application Layer.
- Data Layer.





Data Layer

- Data storage system.

1. Files.
2. Relational Databases(ex: MySQL).
3. NoSQL Databases(ex: mongodb).

Relational Databases

- Strict schema.
- Relations.
- Tables, rows, and fields.
- Use query language(used to communicate with Relational Databases).

User					
UserID	User	Address	Phone	Email	Alternate
1	Alice	123 Foo St.	12345678	alice@example.org	alice@neo4j.org
2	Bob	456 Bar Ave.		bob@example.org	
...	...	...	...	...	...
99	Zach	99 South St.		zach@example.org	

Order	
OrderID	UserID
1234	1
5678	1
...	...
5588	99

LineItem		
OrderID	ProductID	Quantity
1234	765	2
1234	987	1
...	...	...
5588	765	1

Product		
ProductID	Description	Handling
321	strawberry ice cream	freezer
765	potatoes	
...	...	
987	dried spaghetti	

Big Data

Big data is the rapid expansion of structured, unstructured, and semi-structured data generated mostly from internet-connected devices.

- Volume
- Velocity
- Varity

NoSQL

- No Schema.
- No Relations.

NoSQL Types

- Key-Value Store.
- Document-based Store.
- Column-based Store.

Key-Value Store

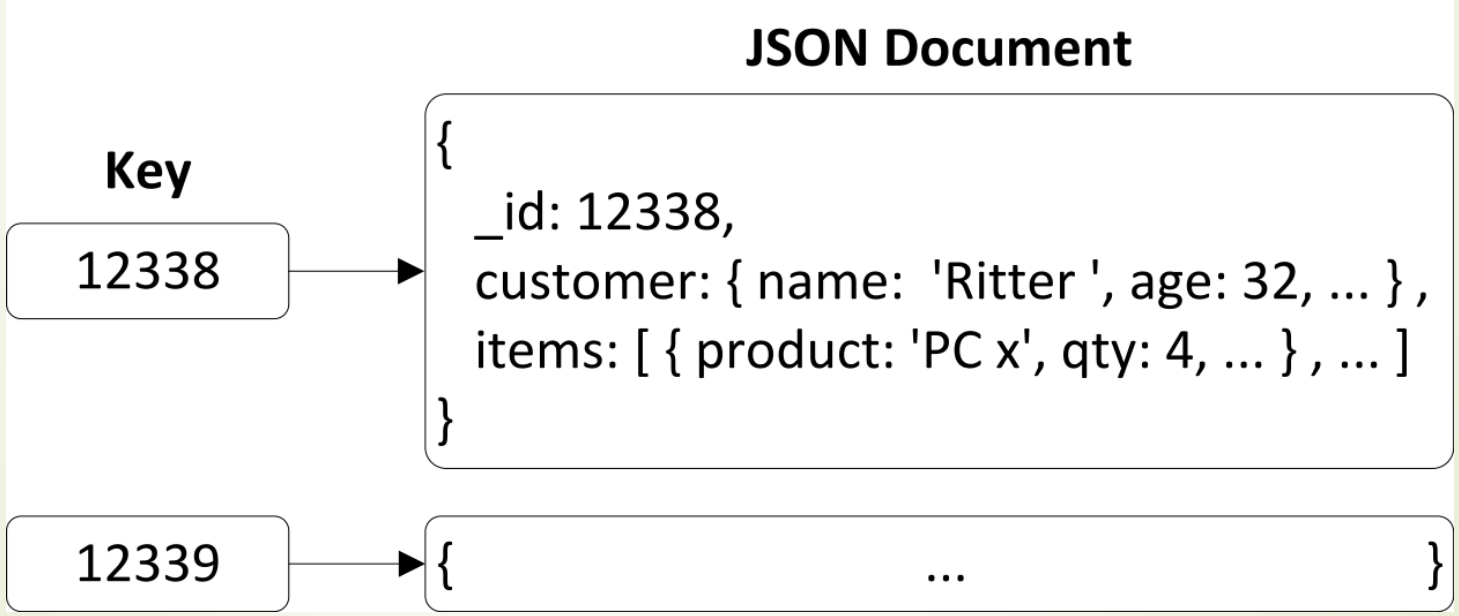
- Simple queries (ex: Redis).

KEY	VALUE
KEY1	STR1,STR2
KEY2	STR1
KEYN	2,AA:A,01/02

Key	Value (Opaque)
User:2:friends	{23, 76, 233, 11}
User:2:settings	Theme: dark, cookies: false
User:3:friends	[234, 3466, 86, 55]

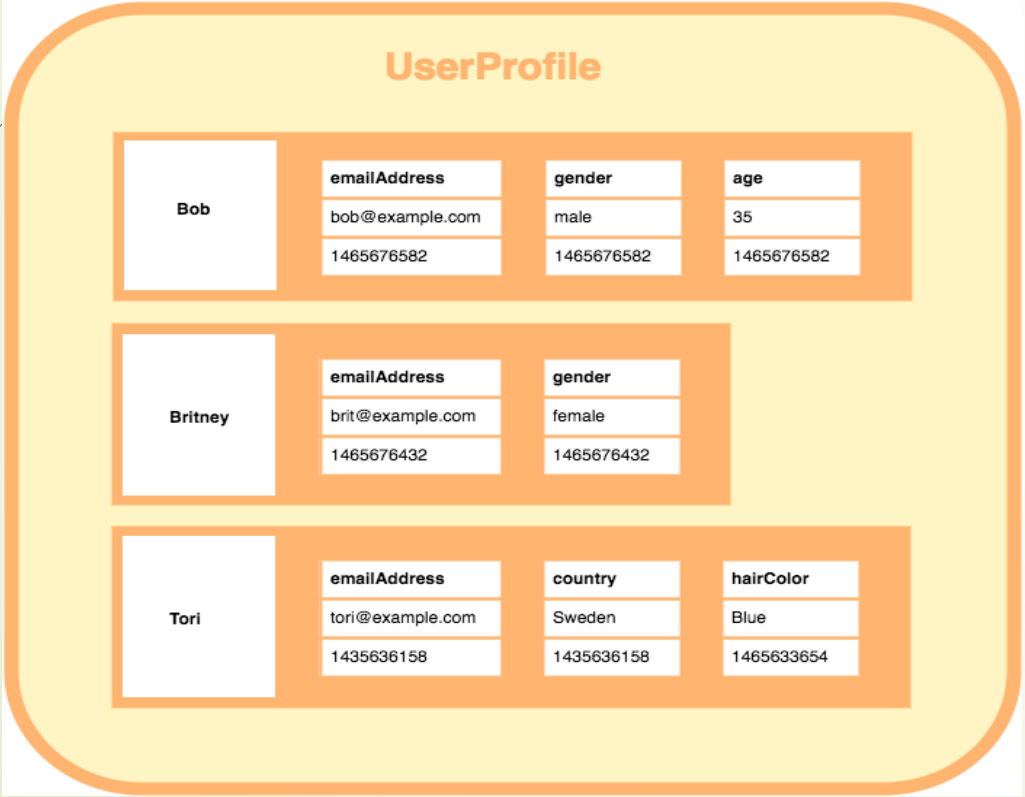
Document-based Store

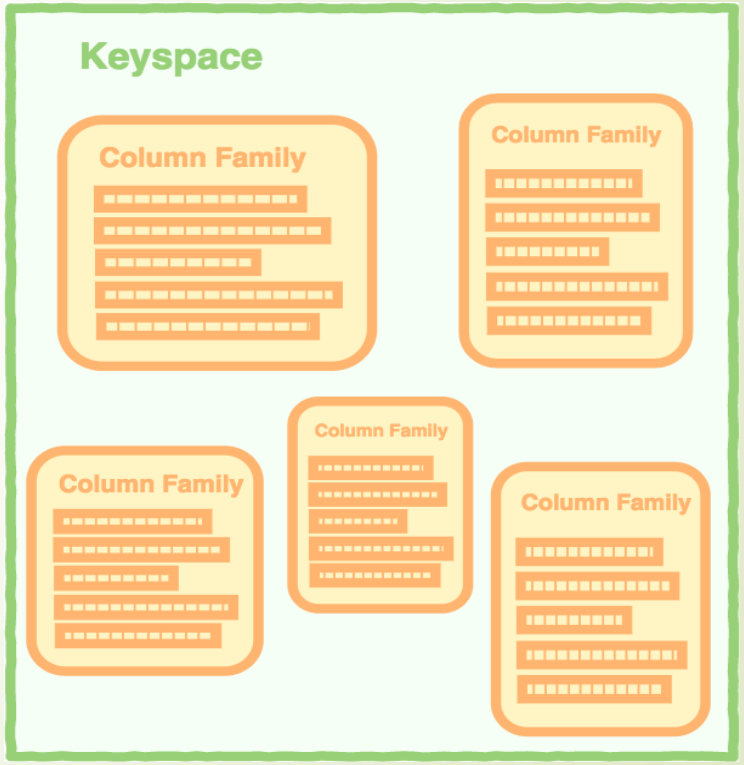
- Document contain a group of key value fields(ex: mongodb).



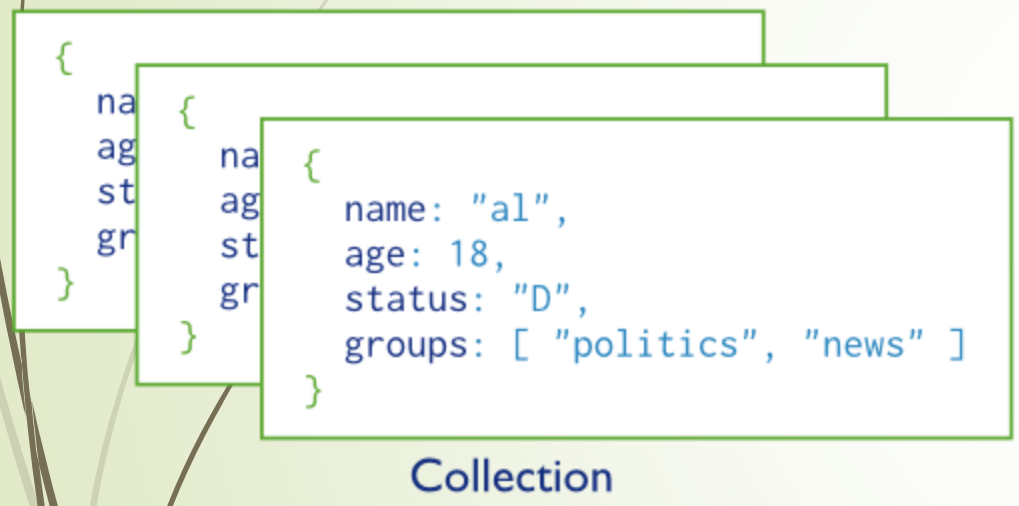
Column-based Store

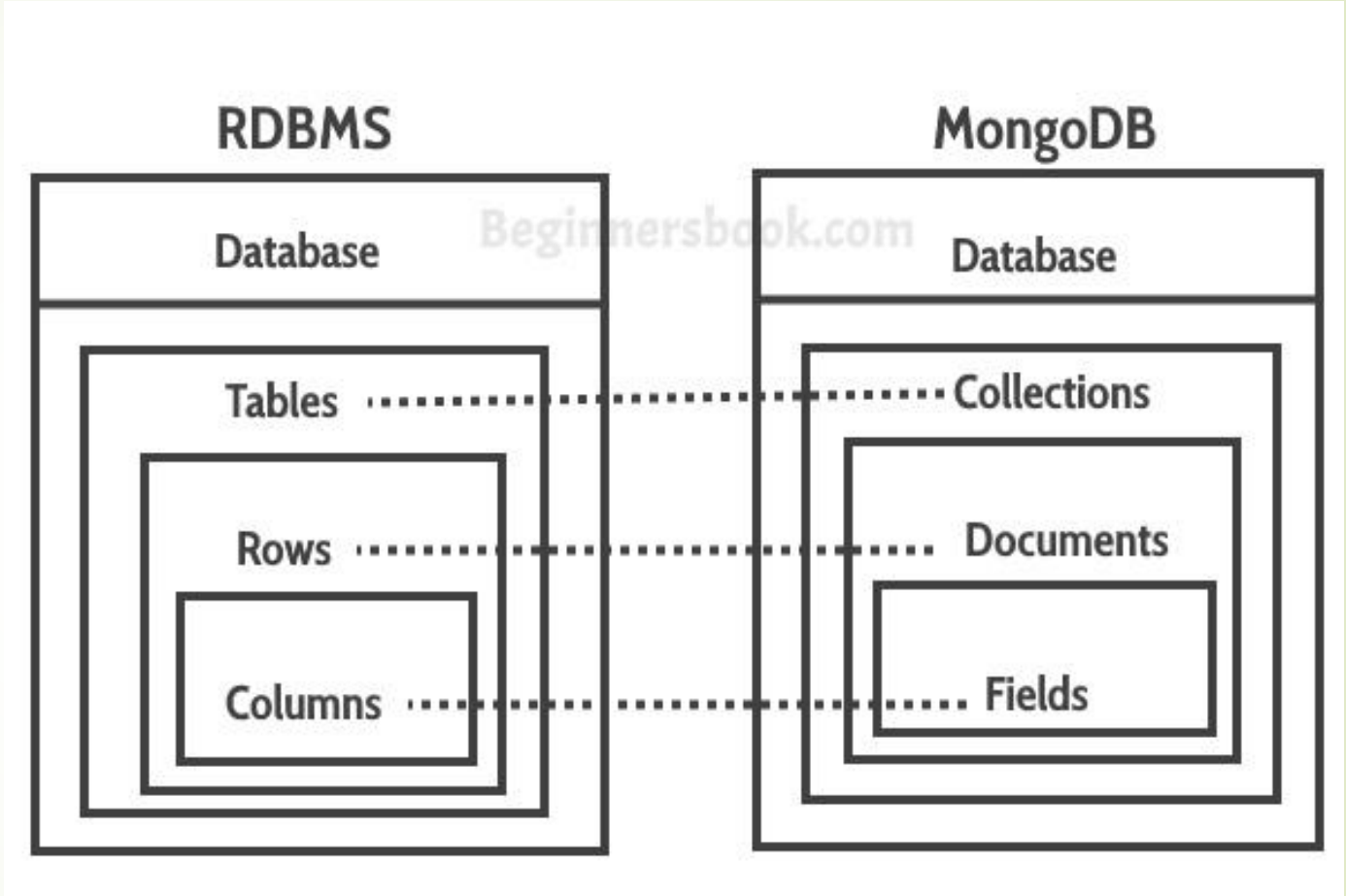
It uses tables, rows, and columns, but unlike a relational database, the names and format of the columns can vary from row to row in the same table (ex: cassandra).





mongodb


{
name: "al",
age: 18,
status: "D",
groups: ["politics", "news"]
}
Collection

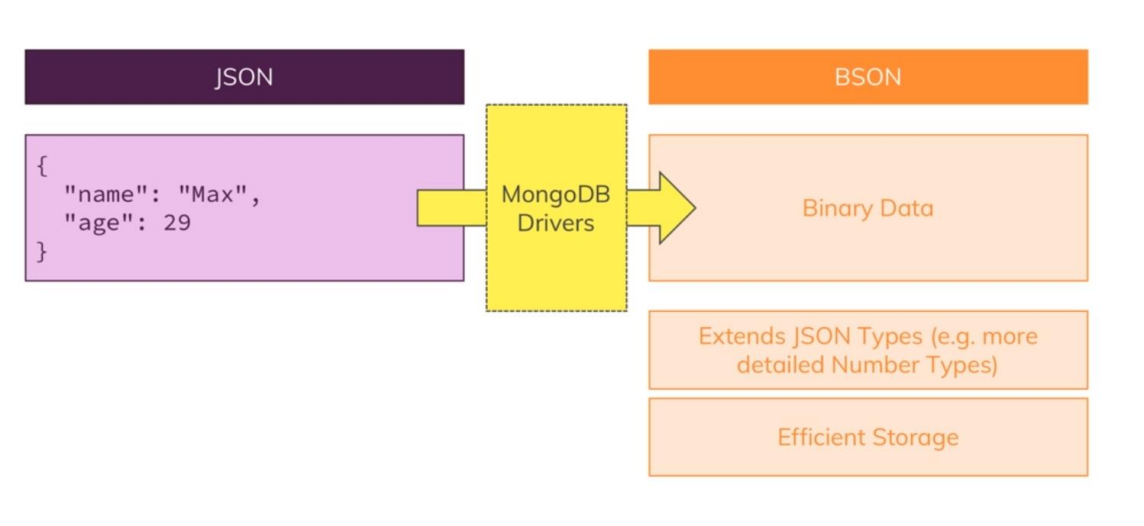

RDBMS
MongoDB
Database
Tables
Rows
Columns
Database
Collections
Documents
Fields
Beginnersbook.com

JSON

- JavaScript Object Notation.
- Commonly used for transmitting data in web applications.
- JSON is purely a data format — it contains only properties, no methods.
- Only quoted strings may be used as properties.
- Single quotes are not valid.

Binary JSON (BSON)

- BSON is the binary encoding of JSON-like documents that MongoDB uses when storing documents in collections.
- adds support for data types.



ObjectId

- MongoDB database drivers by default generate an ObjectID identifier that is assigned to the _id field of each document.
- ObjectID is a 96-bit number which is composed as follows:

1. a 4-byte value representing the seconds since the Unix epoch.
2. a 3-byte machine identifier(derived from mac).
3. a 2-byte process id.
4. a 3-byte counter, starting with a random value.

- ObjectID can be considered globally unique for all practical purposes.

Why mongodb?

- Performance (embedded documents).
- High availability(through replica).
- Horizontal Scaling(through sharding).

mongod

- The primary daemon process.
- Handle data requests.
- Listen on port 27017.

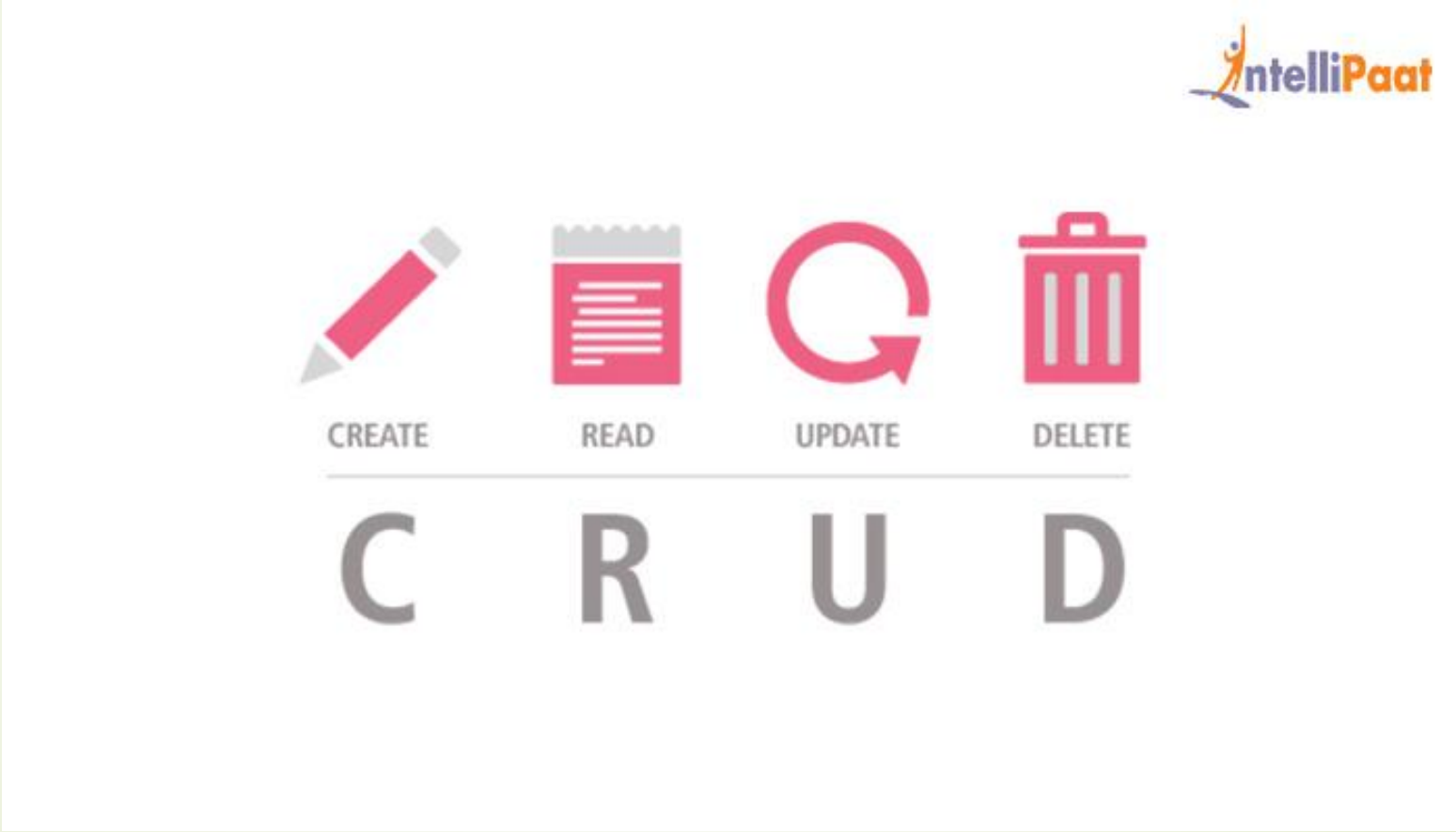
mongod --port --dbpath --maxConns

mongo

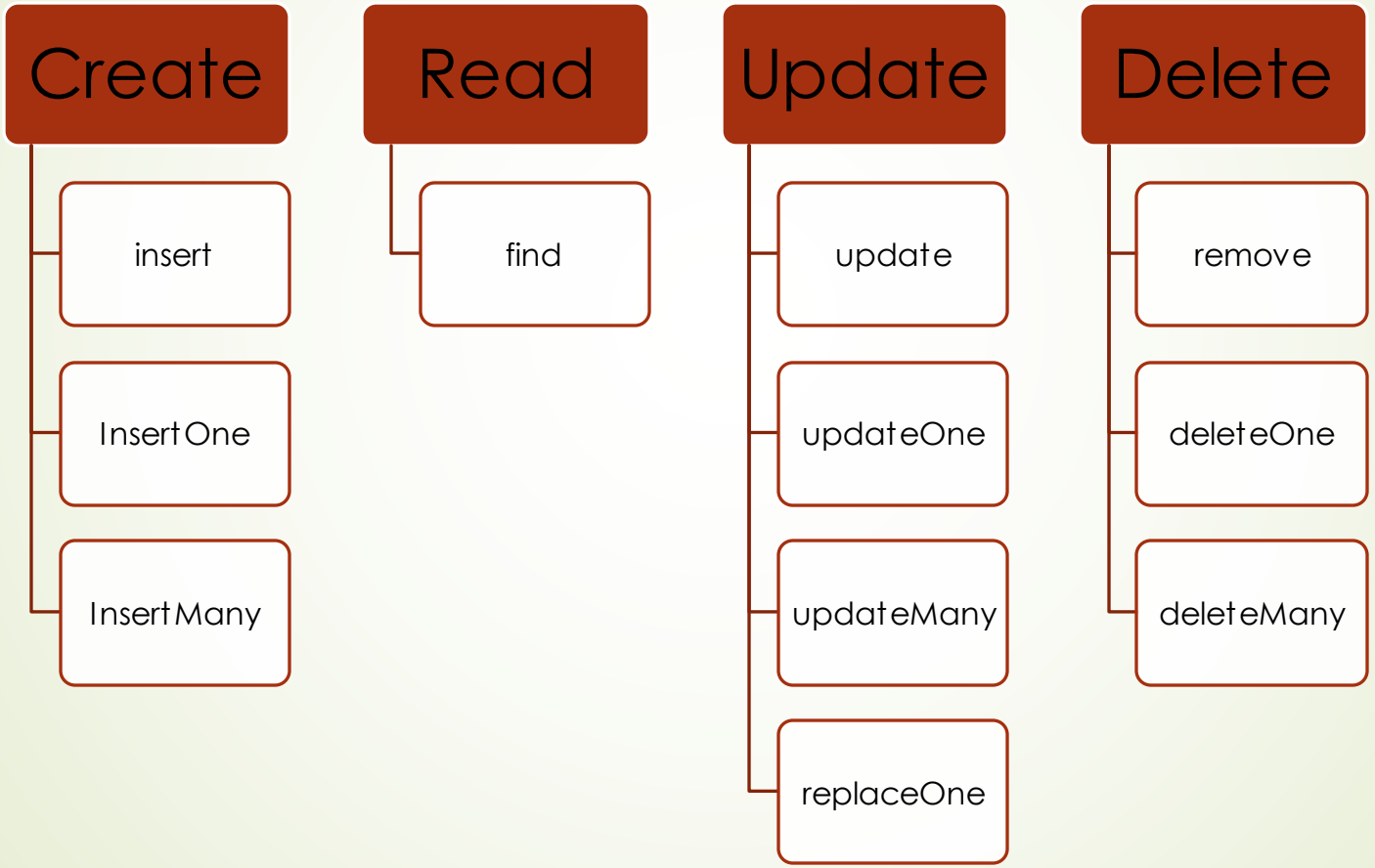
- JavaScript shell interface.
- Test queries.
- Administration.

mongo --port --host --username --password

CRUD



CRUD


Create
insert
InsertOne
InsertMany
Read
find
Update
update
updateOne
updateMany
replaceOne
Delete
remove
deleteOne
deleteMany

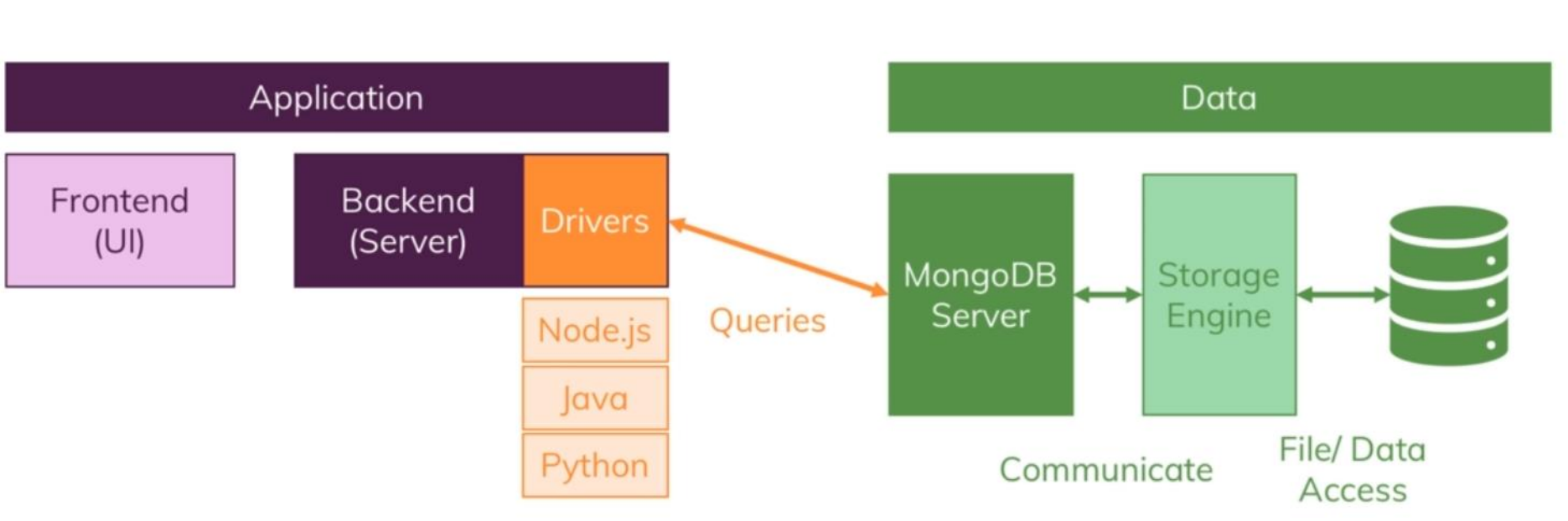
Create Collections

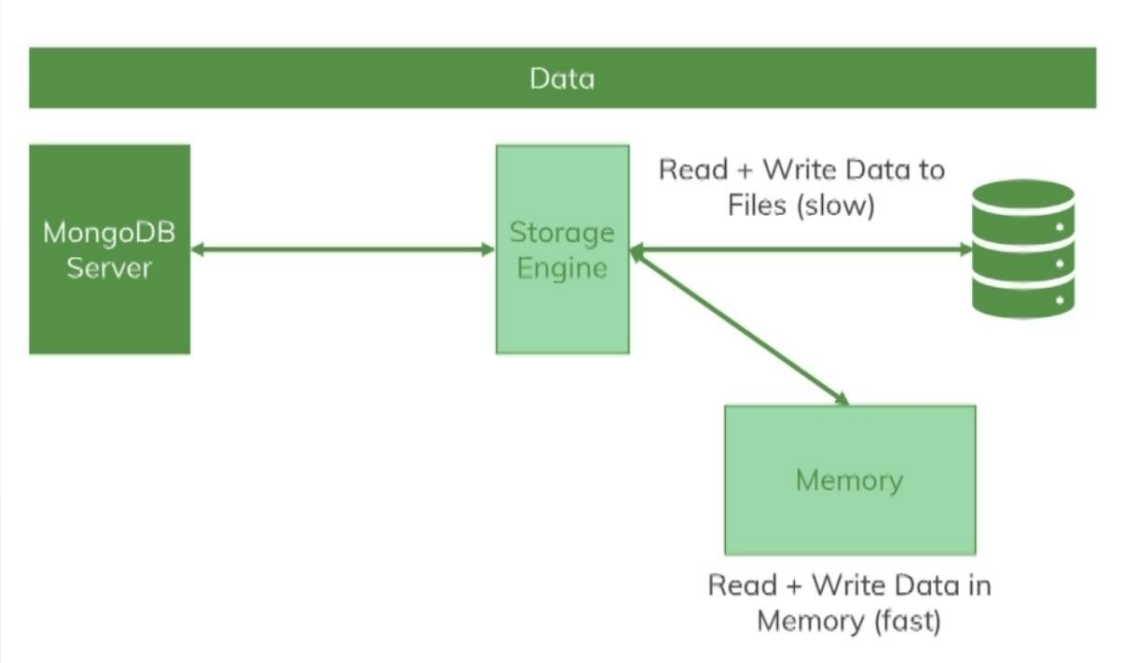
- Implicit Creation.
- Explicit Creation.

```
db.createCollection(<name>,{
                    capped: <boolean>,
                    autoIndexId: <boolean>,
                    size: <number>,
                    max: <number>,
                    validator: <document>,
                } )
```

Enable Auth

1. Run the mongod server in a non-auth mode.
2. Run mongo shell and use admin db.
3. Create a user with the admin db.
4. Re-start the mongod instance with the --auth
5. mongo --port 27017 -u "myUserAdmin" -p "abc123" --authenticationDatabase "admin"
6.
```
use admin
db.createUser(
  {
    user: "admin",
    pwd: "abc123",
    roles: [ { role: "userAdminAnyDatabase", db: "admin" }, "readWriteAnyDatabase"
]
  }
)
```


Application
Frontend (UI)
Backend (Server)
Drivers
Node.js
Java
Python
Queries
Data
MongoDB Server
Storage Engine
Communicate
File/ Data Access


Data
MongoDB Server
Storage Engine
Read + Write Data to Files (slow)
Memory
Read + Write Data in Memory (fast)

Lab 1

- Create database with name “Facebook” and use it.
- Create Collection with name “posts” (implicitly) with properties [“post_text”,”images”,”likes”,”comments”,”Datetime”,”owner”,”live”]

  (create one post owned by "ahmed")
- Create Capped Collection with name users with Size5 MB , 10 users Maximum and must has username field “String” and email end with @gmail.com (validation bouns)
- insert 20 post
- insert 10 users
- display all users
- display user “ahmed” posts
- update ahmed ‘s posts to have likes 10000
- delete ahmed ‘s posts
- Create database user with name “ali” who has only access to read only facebook database