

# Syllabus

- ▶ **Introduction to MongoDB:** definition of MongoDB: what is a document database, what is JSON, How does it look, cross-platform, scalable, flexible.
- ▶ An overview of MongoDB architecture, introduction to mongo dB basics, core concepts and vocabulary, mongo dB databases
- ▶ Mongo dB collections, mongo dB documents, introduction to mongo dB shell, types of storage engines, locks

# Mongo DB Terminology

- ▶ **MongoDB Document:** A document can be defined as an instance of a MongoDB collection. It includes a set of key-value pairs.
- ▶ All the documents include a dynamic schema which means the documents that comprise of the same collection do not need to have the same set of fields and structure.

# Mongo DB Verses NOSQL

SQL Terms/Concepts	MongoDB Terms/Concepts
database	database
table	collection
row	document or BSON (Binary JSON) document
column	field
index	index
table joins	embedded documents
primary key	primary key
Specify any unique column or column combination as the primary key.	The primary key is automatically set to the <code>_id</code> field.

# What is Database?

- ▶ A database is a collection of data stored in the Database Management Systems (DBMS).
- ▶ Data is a small unit which, after processing, is converted into information. So, this data stays in the database in a structured manner and is then processed by the help of a DBMS to be converted into some information which is helpful for analyzing and decision making process.
- ▶ The main function of any DBMS is to:
  - ▶ Store the data
  - ▶ Retrieve the data
  - ▶ Manipulate the data
  - ▶ Process the data

# What is Relational Database?

- ▶ A relational database contains data having some sort of relationships among two or more data records.
- ▶ These kinds of databases usually have tables interlinked with key-value or field-value pairs.
- ▶ For example, if there is a table which has a unique primary key column in table A then it is linked with table B which has a column that stores the primary key of table A, (also known as foreign key) in table B in a manner such that these two are interlinked.

# What is NOSQL Database?

- ▶ A NoSQL database is a kind of database that provides the mechanism of storage, retrieval and manipulations of data in a different approach than what the **Relational Database Management System** (an SQL based database) provides.
- ▶ It is sometimes called as a **NoSQL** database and sometimes "Not Only SQL"

# NOSQL Database

- ▶ It is a myth that NoSQL data cannot store relationship data while in reality NoSQL databases like MongoDB, stores relationship data in a different manner than SQL based databases.
- ▶ In case of NoSQL database like MongoDB, the relationship data is not stored and linked in a simpler manner because here, the relationship data doesn't split between the tables as it does in the SQL based databases.

# MongoDB Database

- ▶ MongoDB database is a document oriented database which stores data in JSON like documents with the dynamic schema.
- ▶ The concept of dynamic schema is to store the records without worrying too much on the structural part of the database. MongoDB database is a collection of data, or documents that have JSON like structure and are not very rigid in terms of schema.



# MongoDB Collections

- ▶ A collection in MongoDB is a grouping of MongoDB documents that hold data, usually dynamic in nature, because collection does not enforce a schema.
- ▶ A collection is equivalent to a table in RDBMS.
- ▶ Here, the important aspect of collection is that it allows dynamic schema, which means that a collection can hold documents that could be different in terms of their structure

# EXAMPLE OF MONGODB COLLECTIONS

A Collection of Documents in MongoDB

```
{  
  "student_id":1,  
  "student_name":"Siya Sharma",  
  "student_class":"4A",  
  "student_hobbies":["Singing", "Painting", "Dance"]  
}
```

```
{  
  "student_id":2,  
  "student_name":"Harry Dsouza",  
  "student_class":"7A",  
  "student_hobbies":["Cricket", "Football"],  
  "student_bus_route":"E9"  
}
```

```
{  
  "student_id":4,  
  "student_name":"Md. Hussain",  
  "student_class":"10B",  
  "student_hobbies":["Poetry"]  
}
```

# ROW AND COLUMN IN RDBMS

Row



Document

Column



Field

Relationships



Linking and  
Embedding

# Document in MongoDB

- ▶ A document based database and each record in MongoDB is termed as document.
- ▶ These documents are made up of key-value or field-value pairs which are just like JSON

# EXAMPLE

```
{  
  key1: value1,  
  key2: value2,  
  key3: value3,  
  key4: value4,  
  ...  
  keyN: valueN  
}
```

OR

```
{  
  field1: value1,  
  field2: value2,  
  field3: value3,  
  field4: value4,  
  ...  
  fieldN: valueN  
}
```

# EXAMPLE

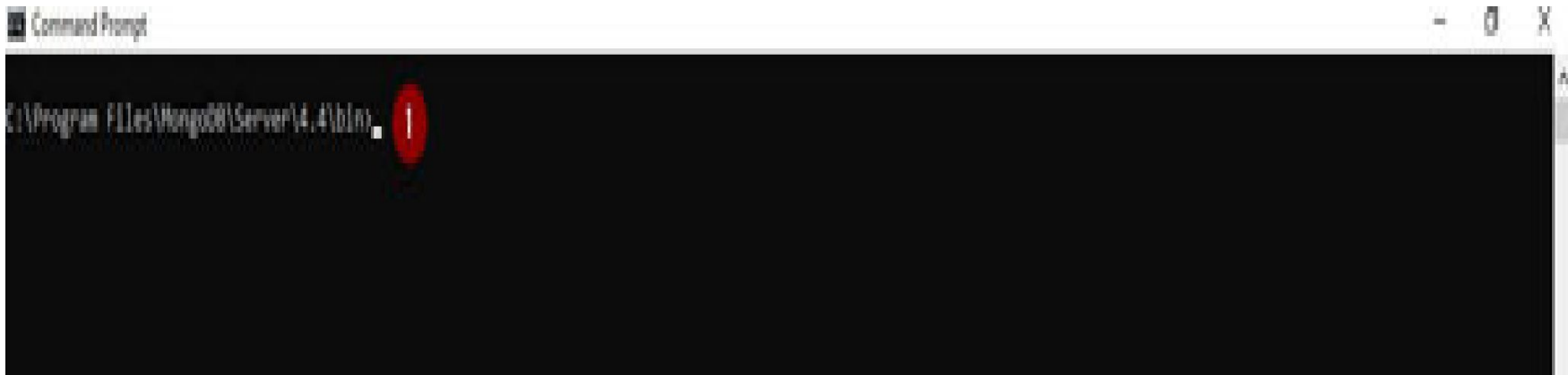
```
[
  {
    "employee_id": "001",
    "fname": "Manish",
    "lname": "Sharma",
    "department": "IT"
  },
  {
    "employee_id": "002",
    "fname": "Pooja",
    "lname": "Kaushik",
    "department": "HR"
  },
  {
    "employee_id": "003",
    "fname": "Shahid",
    "lname": "Reza",
    "department": "IT"
  }
]
```

# MongoDB Shell

- ▶ MongoDB Shell is an interface used in MongoDB which allows users to interact with MongoDB database for database-related queries to perform various **CRUD (Create, Read, Update, and Delete)** operations as well as administration of MongoDB.
- ▶ MongoDB Shell is installed automatically by MongoDB installer program.
- ▶ It is also available as a standalone program and you can install it separately from MongoDB Inc. official website.

# Connecting to MongoDB Shell

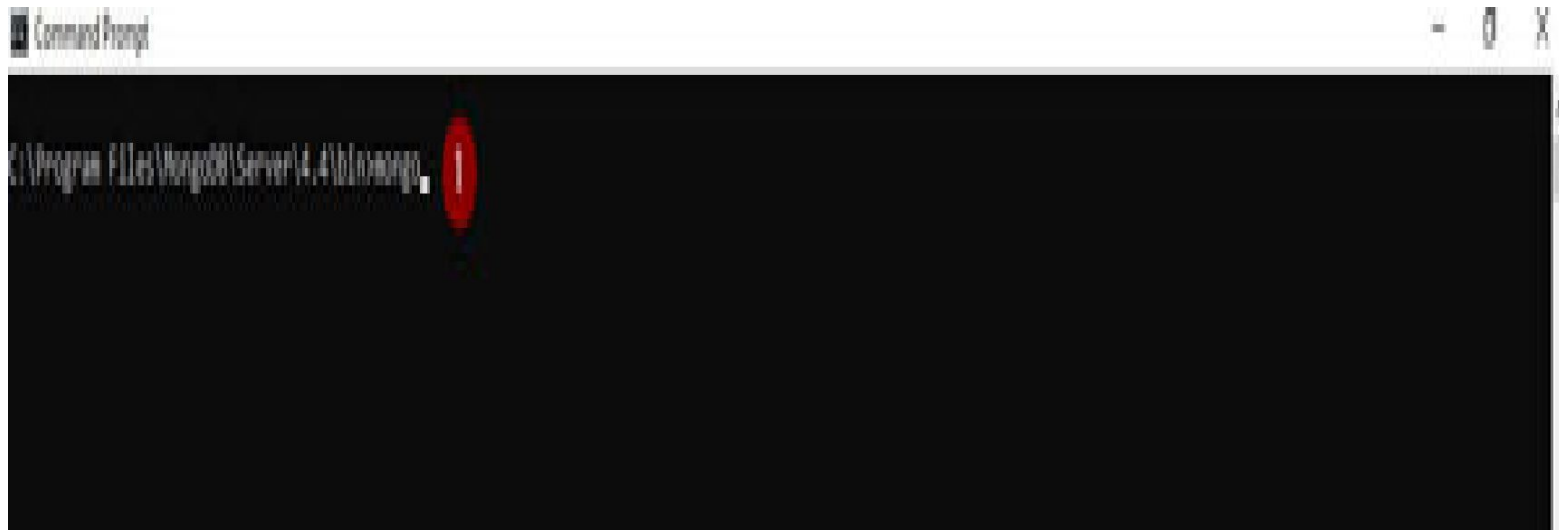
- ▶ To connect to MongoDB from your Windows machine, follow these steps:
- ▶ 1. Open a separate Shell command window and navigate to the `bin` directory of MongoDB.
- ▶ The path could be as follows:
- ▶ **`C:\Program Files\MongoDB\Server\4.4\bin`**





# Connecting to MongoDB Shell

- 2. Now, give the following command and press *Enter*, as shown in the following screenshot:



# Connecting to MongoDB Shell

- ▶ 3. Once you give this command and press enter, it will open MongoDB Shell and you can type the MongoDB related commands, as shown in the following screenshot:

```
Command Prompt - mongo
C:\Program Files\MongoDB\Server\4.4\bin>mongo
MongoDB shell version v4.4.1
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("d739a88c-5443-4232-b535-358eb49da7f5") }
MongoDB server version: 4.4.1
***
The server generated these startup warnings when booting:
  2020-09-11T13:58:48.644+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
***
***
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
  metrics about your deployment (disk utilization, CPU, operation statistics, etc).

  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
  and anyone you share the URL with. MongoDB may use this information to make product
  improvements and to suggest MongoDB products and deployment options to you.

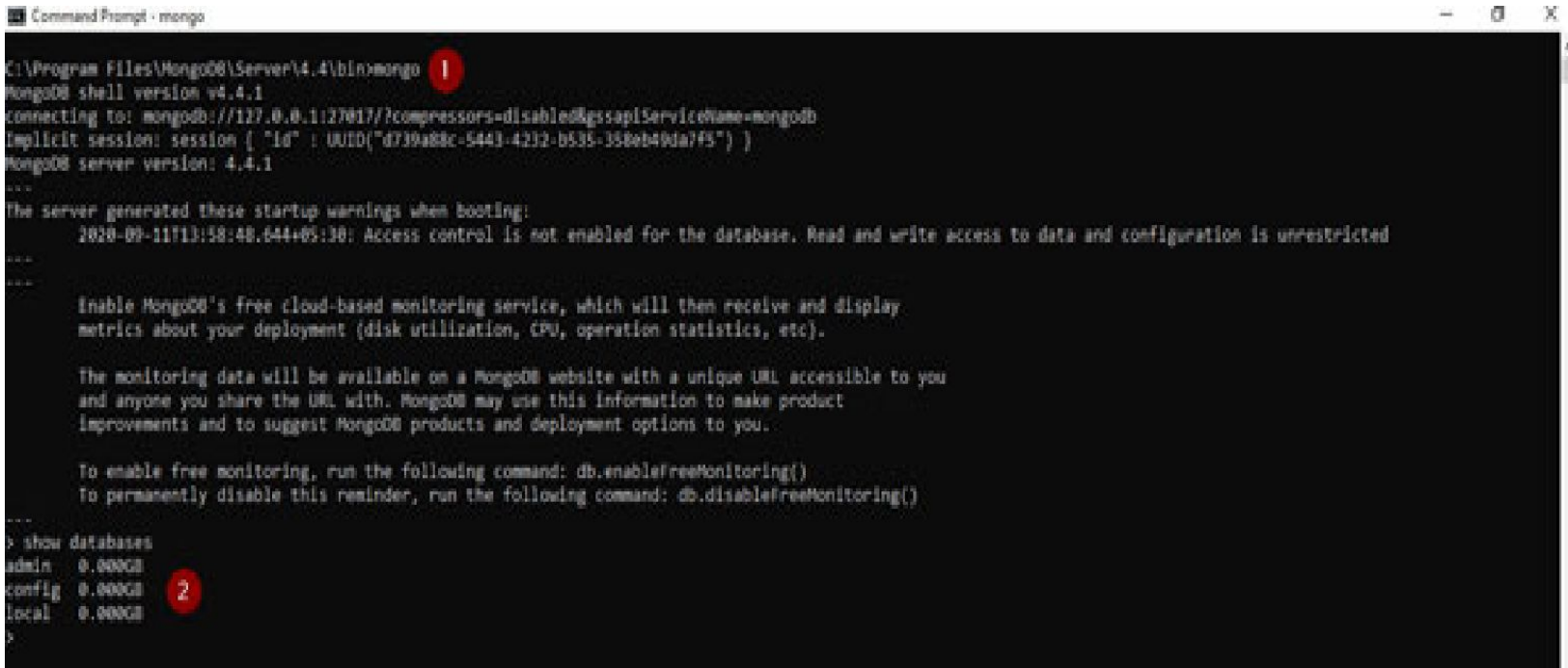
  To enable free monitoring, run the following command: db.enableFreeMonitoring()
  To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
***
>
```

2

MongoDB Shell

# Connecting to MongoDB Shell

- ▶ 4. To test further, issue the following MongoDB command and press *Enter*, as shown in the following screenshot:



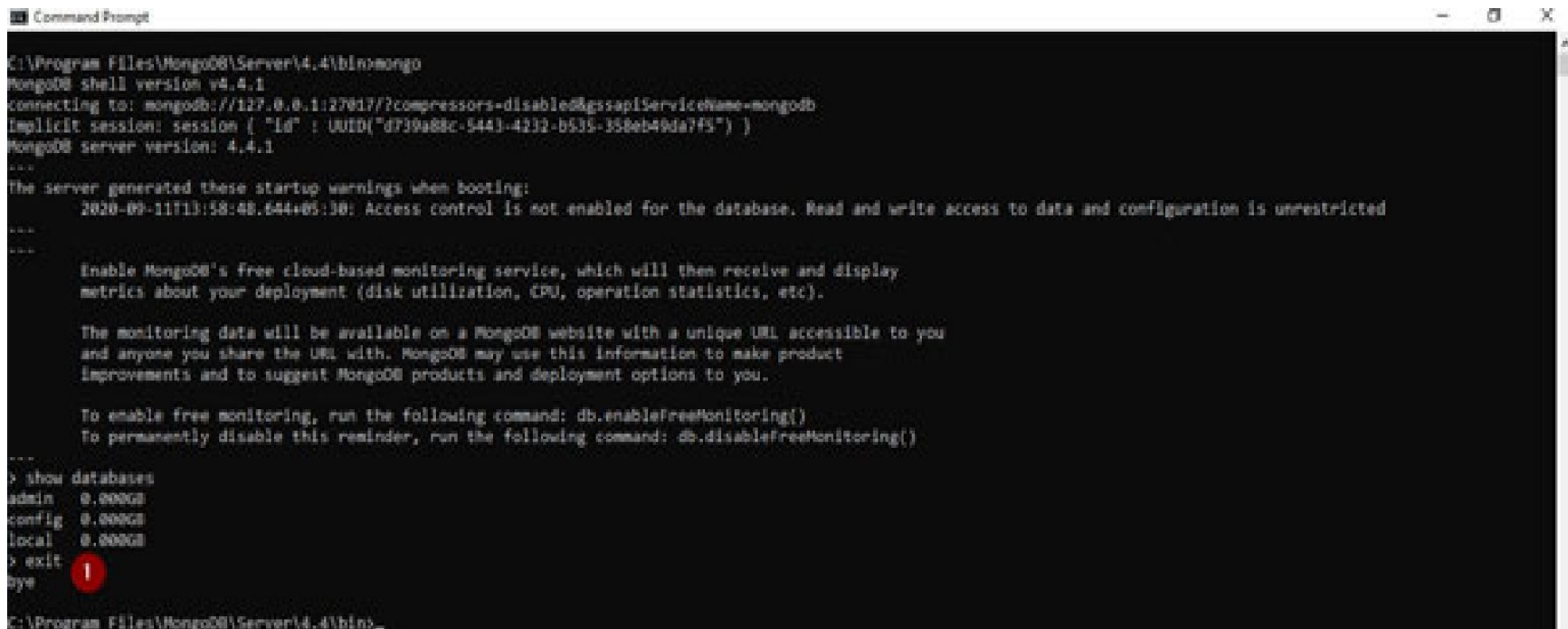
```
Command Prompt - mongo
C:\Program Files\MongoDB\Server\4.4\bin>mongo 1
MongoDB shell version v4.4.1
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("d739a88c-5443-4232-b535-358eb49da7f5") }
MongoDB server version: 4.4.1
***
The server generated these startup warnings when booting:
  2020-09-11T13:58:48.644+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
***
***
  Inable MongoDB's free cloud-based monitoring service, which will then receive and display
  metrics about your deployment (disk utilization, CPU, operation statistics, etc).

  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
  and anyone you share the URL with. MongoDB may use this information to make product
  improvements and to suggest MongoDB products and deployment options to you.

  To enable free monitoring, run the following command: db.enableFreeMonitoring()
  To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
***
> show databases
admin   0.000GB
config  0.000GB 2
local   0.000GB
>
```

# Connecting to MongoDB Shell

- 5. To exit from MongoDB Shell, type the following command and press *Enter*:  
**"exit"**
- This will take you out from MongoDB Shell, as shown in the following screenshot



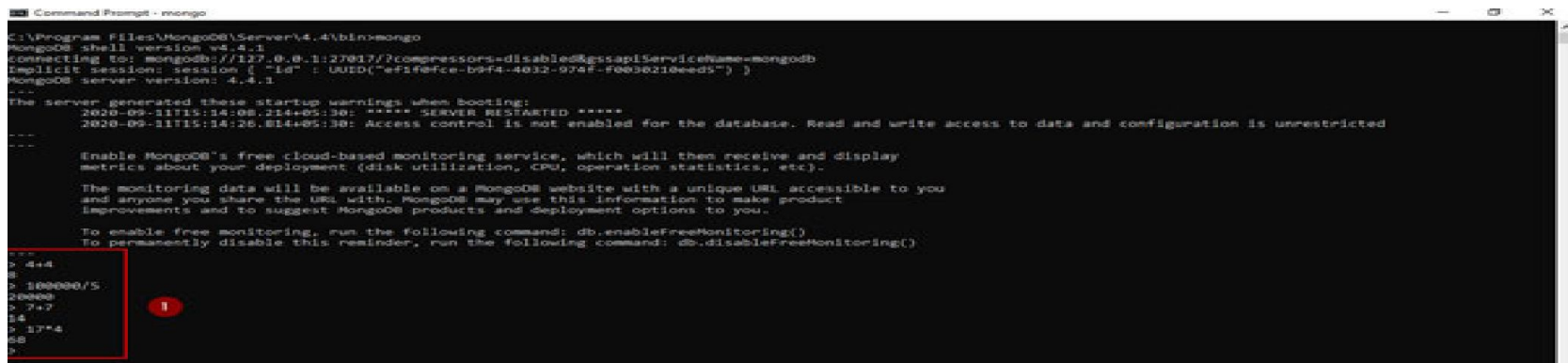
```
Command Prompt
C:\Program Files\MongoDB\Server\4.4\bin>mongo
MongoDB shell version v4.4.1
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("d7f9a88c-5443-4232-b535-358eb49da7f5") }
MongoDB server version: 4.4.1
....
The server generated these startup warnings when booting:
  2020-09-11T13:58:48.644+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
....
....
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
  metrics about your deployment (disk utilization, CPU, operation statistics, etc).

  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
  and anyone you share the URL with. MongoDB may use this information to make product
  improvements and to suggest MongoDB products and deployment options to you.

  To enable free monitoring, run the following command: db.enableFreeMonitoring()
  To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
....
> show databases
admin   0.000GB
config  0.000GB
local   0.000GB
> exit
bye
C:\Program Files\MongoDB\Server\4.4\bin>
```

# Basic Shell Commands

- ▶ MongoDB Shell is a JavaScript based interface which allows you to run various CRUD (Create, Read, Update, and Delete) and administrative operations.
- ▶ As it is a JavaScript based interface, it has the ability to interpret JavaScript commands as well, other than MongoDB specific operations. A simple example is arithmetic operations like addition or multiplication, as shown in the following screenshot



```
Command Prompt - mongo
C:\Program Files\MongoDB\Server\4.4\bin>mongo
MongoDB shell version v4.4.1
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("ef1fe1ce-b9f4-4032-974f-f0030210eed5") }
MongoDB server version: 4.4.1
---
The server generated these startup warnings when booting:
2020-09-11T15:14:08.214+05:30: ***** SERVER RESTARTED *****
2020-09-11T15:14:26.814+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

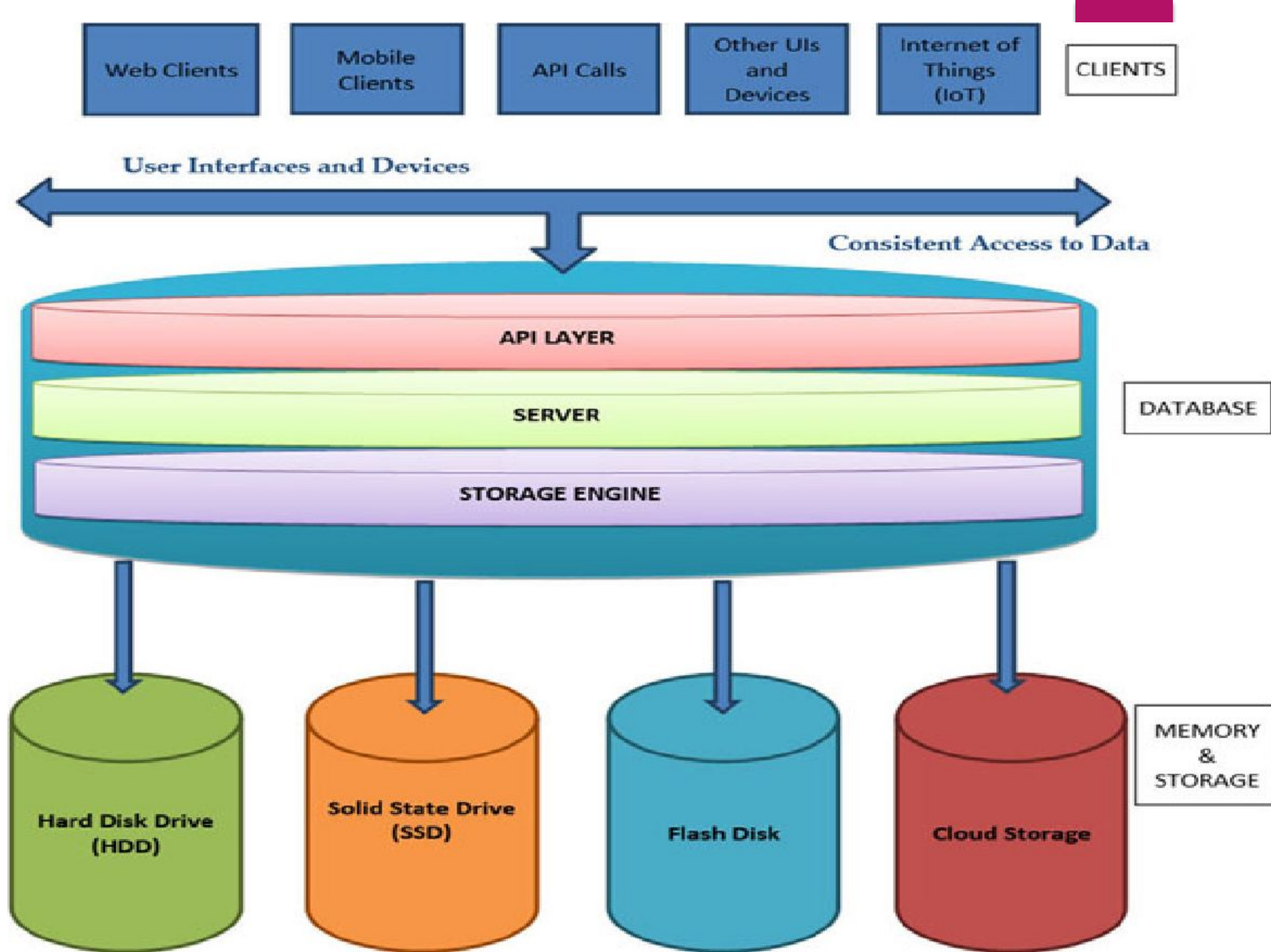
The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> 4+4
8
> 1000000/5
200000
> 7*7
49
> 17*4
68
>
```

# Storage Engines

- ▶ Storage engine is a software component that works in the **Database Management System (DBMS)** to provide the user **CRUD (Create, Read, Update, and Delete)** functionality.
- ▶ The database engine defines a way to store the data into the database and also the manner in which the data is stored in both the memory and the disk.





# TYPES OF STORAGE ENGINES IN MONGODB

The following storage engines are supported by MongoDB:

- ▶ WiredTiger
- ▶ In-memory
- ▶ Encrypted



# WIRED TIGER STORAGE ENGINE

- ▶ The WiredTiger storage engine uses document-level concurrency control for write operations, which means that it can handle multiple requests without conflicting with each other.
- ▶ In case, there is a conflict between two operations, then the WiredTiger storage engine will retry that operation with ease.
- ▶ While using WiredTiger as a storage engine, MongoDB utilizes the WiredTiger internal cache as well as the file system cache. WiredTiger is helpful in the efficient use of the CPU and RAM and is also helpful when it comes to tuning the database storage engine.
- ▶ It can be tuned more than the MMAP storage engine.

# IN MEMORY STORAGE ENGINE

- ▶ The in-memory storage engine is available in the MongoDB Enterprise Edition.
- ▶ These provide high output with low latency and high availability.
- ▶ The in-memory storage engines support high-level infrastructure based on zonal sharding.
- ▶ These also come with MongoDB rich query capability and indexing support

# ENCRYPTED STORAGE ENGINE

- ▶ This feature allows MongoDB to encrypt data and decrypt whenever required.
- ▶ Many a times, data encryption is forced by the government bodies or some industry standards like HIPAA, PCI-DSS, and FERPA.
- ▶ These are some security standards and guidelines which help in ensuring compliance with security and privacy policies of the industry or organization.

# THIRD PARTY STORAGE PLUGGED STORAGE ENGINES

- ▶ MongoDB also supports the third-party storage engines. These storage engines can be plugged in like modules and can also be independently updated.
- ▶ An example of the third-party storage engine is RocksDB developed by Facebook Inc. and designed to handle write-intensive workloads.
- ▶ The RocksDB storage engine is the first one to use the module system for the MongoDB storage integration layer.

# Database LOCK

- ▶ A database lock is a mechanism used by the database to prevent a scenario where two users or two sessions modify the same data at the same time.
- ▶ What this means is that there could be a scenario where there are two or more database users who are working on the same set of data and want to update the same data or record.
- ▶ In this case, there would be lot of issues that would arise. The mostly aroused situation could be related to which data is latest and up-to-date.

# Database Lock Operations

- ▶ There are mainly three different types of locking operation that are mostly done in databases:
- ▶ Read lock operation
- ▶ Write lock operation
- ▶ Unlock operation

# Database Locks in MongoDB

- ▶ MongoDB uses multiple granularity locking. This type of locking ensures that the database locking can be done at the child level (record level).
- ▶ MongoDB allows multiple clients to read and write data at the same time and thus, it uses locking at different levels and other concurrency control methods to achieve this.

# Database Locks in MongoDB

- ▶ MongoDB's WiredTiger engine uses intent locks as well as optimistic concurrency control
    - 1) Global
    - 2) Database and
    - 3) Collection Level
- so it always prevents the conflicts



# Database Locks in MongoDB

- ▶ MongoDB also allows the database engines to implement their own concurrency and locking mechanisms at the document level.
- ▶ In case WiredTiger detects any write conflict with any client, then that client will transparently retry its operation