

# Structured and Non-Structured data

15 July 2025 21:48

## >STRUCTURED:

- ▶ in structured data the inputs or the data is arranged systematically...specifically in the form of tables
- ▶ in structured data the data type of the data in the table is fixed
- ▶ it is like data in an excel sheet in the form of rows and columns in fixed places
- ▶ The examples :
  - 1.bank records
  - 2.hospital records
  - 3.SQL databases
- ▶ Databases used: MySQL , PostgreSQL

## >NON-STRUCTURED:

- ▶ the non-structured data is the one which cannot be put into a table due to its varied schema , i.e. , the data type of this data varies.....
- ▶ hence cannot be put in specific columns and rows , the unstructured data has no fixed format or flow
- ▶ And since it is a bit messy .....it is hard to find and sort the non-structured data.
- ▶ example: text files
- ▶ Databases used: MongoDB, Cassandra , Redis , etc.

## ● MongoDB :

- MongoDB document stores data in the form of key-value pairs
- It is a NoSQL database , popularly used
- It makes it easy to handle large unstructured data
- It consists of three parts:  
Databases----->Collections----->Documents

# CRUD in MongoDB

19 July 2025 10:43

1. Create : to create a field in MongoDB we use
  - >insert One() command
  - >insert Many() command
2. Read : to read the inserted fields we use the find() command
3. Update : to update an already existing field we use
  - > update One()
  - >update Many()
4. delete : to delete a field we use
  - >delete One()
  - >delete Many()

# Aggregate

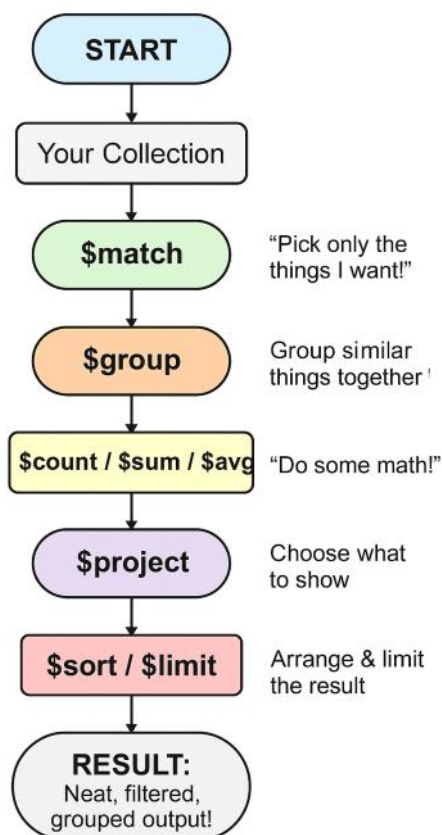
30 April 2025 23:36

Aggregate is to sort or group a certain collection of data...in your desired categories.

We can think of it as the root of a plant which passes the data through different stages to reach the top , i.e., to get the result.

Example:

Assuming our data is just a list of fruits....and that we want to pick a certain fruit



----->the match command picks our desired fruit from the list using aggregate

----->if we want to make a juice of three fruits...we will use the \$match command to pick out those...and \$group command to make a group of them

----->the mathematical functions help us in various calculations

----->the \$project command will help us choose what to show...if we make two diff groups using the \$group command for 2 diff juices...then this will help om what to show

-----> \*sort: this may help us sort the fruit price wise...in ascending or descending order...so that we can estimate the price of our juices.

\*limit: this will help us decide the definite amount of fruits to be used at a time.

The dollar sign(\$) in MongoDB is used to represent the operators or the field references...so that we make out the difference .

Operator	Purpose
\$match	Filters documents
\$group	Groups documents by a certain field
\$sum	Adds values together
\$ avg	Averages values
\$ gt	Greater than
\$ lte	Less than or equal

NOTE: \$match operator is used for filtering or comparing data....when to calculate the sum or average we use the \$group operator.

operators	purpose
<b>\$set</b>	To add or update a field
\$project	To choose what fields to show
\$lookup	To look into other collections or to use the fields from there
\$unwind	It takes a field and separates it into separate documents

# Indexing

19 July 2025 20:33

## Indexing:

- Indexing in mongoDB is just like the index of a book.....it is used to find the data faster
- It sorts data by the fields we have chosen for indexing .....
- Ex: if we are handling student data then we can do indexing by roll nos. ...so each time I want a particular data.... i will just search by roll nos.
- this saves our time of going through each student's records just to find the one I need.
- So , since mongoDB uses unstructured data .....
- indexing makes it easier to keep a large pile of data sorted and easy to find.

## JSON

- Javascript Object Notations
- Easy to read and write
- Works for javascript , python , java , etc.
- Unsrtructred data can be easily stored