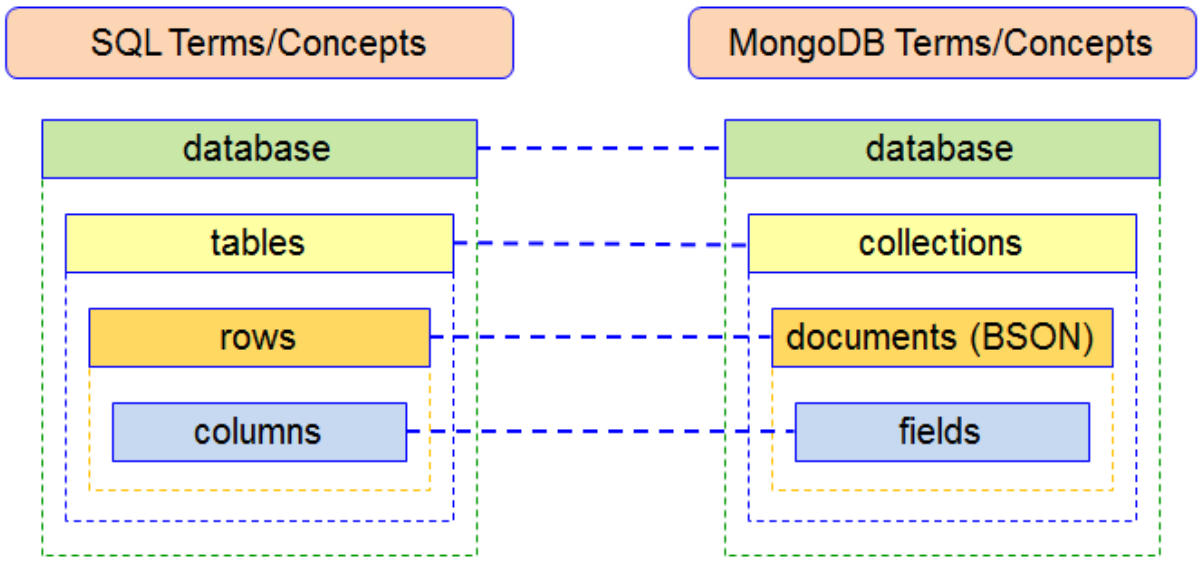
MONGODB

mongod --version / mongo --version :
This will give the version of MongoDB on your system

mongod :
This command will start the mongoDB database process. When you run mongod this starts the mongodb server and makes it available to clients like mongo, mongosh or any application.

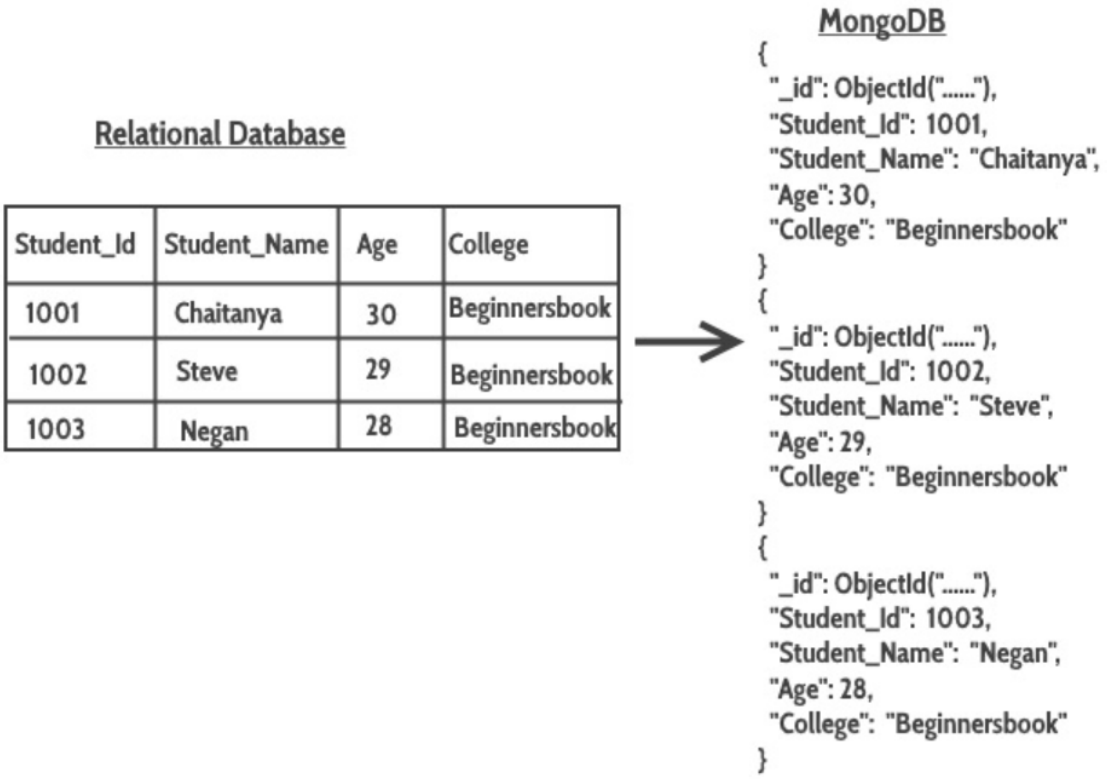
When you run this command, it tries to find configuration files in C:/data/db.
If it doesn't find the folder, **it will give you an error and the server will not be started**. So we need to create this folder before running mongod.



In mongo, **Collections** is equivalent to **tables** in rdbms
Documents in mongo are equivalent to **rows** in rdbms

Fields in mongo are equivalent to **columns** in rdbms

Table vs Collection



In MongoDB data is stored in a collection of documents which can have dynamic schema.

A collection is a group of documents, and documents within the collection can have different fields.

use sample : creating database named **sample**
db.createCollection("employee") : to create collection/table name **employee** in the database
show collections : to show existing collections/tables
show dbs : to show databases

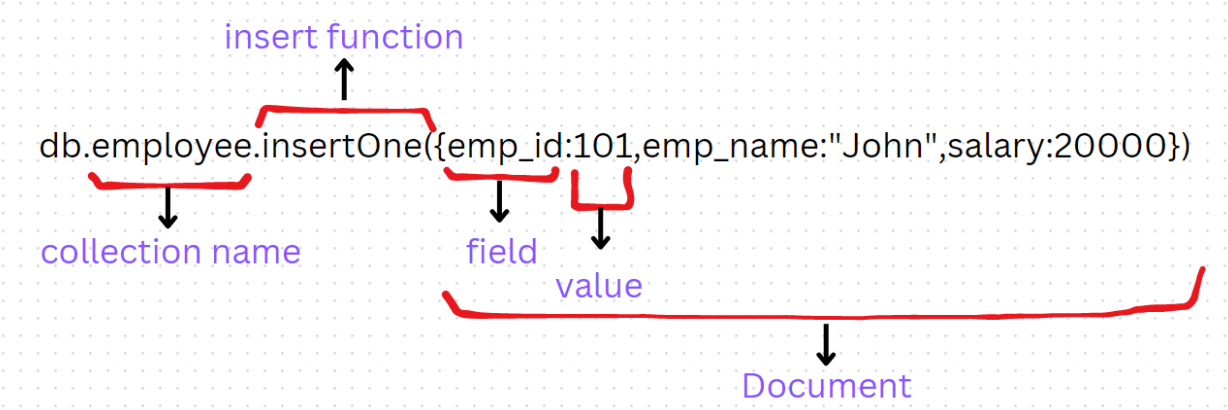
Field

- You cannot have **NULL** characters in your field name.
- The server permits storage of the field name that contains **. (dot)** or **$** sign.
- The field name **_id** is reserved to use as the primary key. It has a value that must be unique in the collection. If a user doesn't define **_id** mongo will create an object id that can be uniquely identified throughout the collection.

Insert Command:

Insert command is used to insert a document in a collection

db.employee.insertOne({emp_id:101,emp_name:"John",salary:20000})




```
sample> db.employee.find()
[
  {
    _id: ObjectId("6443c5a6f8715e437564a1f3"),
    emp_id: 101,
    emp_name: 'John',
    salary: 20000
  }
```

```
]
```

```
db.employee.insertMany([{_id:"123",emp_id:102,emp_name:"Smith",salary:
2000},{_id:"122",emp_id:103,emp_name:"dave",salary:4500}])
{ acknowledged: true, insertedIds: { '0': '123', '1': '122' } }
sample> db.employee.find()
[
  {
    _id: ObjectId("6443c5a6f8715e437564a1f3"),
    emp_id: 101,
    emp_name: 'John',
    salary: 20000
  },
  { _id: '123', emp_id: 102, emp_name: 'Smith', salary: 2000 },
  { _id: '122', emp_id: 103, emp_name: 'dave', salary: 4500 }
]
```

```
const
documents=[{name:"johny",age:30},{name:"ram",age:23,name:"rocky",age
:450}];

db.employee.insertMany(documents,function(err,result){if (err){
console.log(err);}else{console.log(result.insertedCount+ "document
inserted");} });
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("6443c9def8715e437564a1f4"),
    '1': ObjectId("6443c9def8715e437564a1f5")
  }
}
```

```
db.employee.insertMany([{
    "_id": ObjectId("6090196dbb1cfc7842916d61"),
    "name": "John Doe",
    "position": "Software Engineer",
    "department": "Engineering",
    "salary": 90000
},
{
    "_id": ObjectId("6090196dbb1cfc7842916d62"),
    "name": "Jane Smith",
    "position": "Project Manager",
    "department": "Management",
    "salary": 110000
},
{
    "_id": ObjectId("6090196dbb1cfc7842916d63"),
    "name": "Bob Johnson",
    "position": "Marketing Manager",
    "department": "Marketing",
    "salary": 105000
},
{
    "_id": ObjectId("6090196dbb1cfc7842916d64"),
    "name": "Sara Lee",
    "position": "Human Resources",
    "department": "Human Resources",
    "salary": 95000
},
{
    "_id": ObjectId("6090196dbb1cfc7842916d65"),
    "name": "Tom Williams",
    "position": "Sales Representative",
    "department": "Sales",
    "salary": 85000
},
{
    "_id": ObjectId("6090196dbb1cfc7842916d66"),
```

```
    "name": "Emily Jones",
    "position": "Accountant",
    "department": "Accounting",
    "salary": 100000
},
{
    "_id": ObjectId("6090196dbb1cfc7842916d67"),
    "name": "Mike Davis",
    "position": "Operations Manager",
    "department": "Operations",
    "salary": 120000
},
{
    "_id": ObjectId("6090196dbb1cfc7842916d68"),
    "name": "Kelly Brown",
    "position": "IT Specialist",
    "department": "Information Technology",
    "salary": 95000
},
{
    "_id": ObjectId("6090196dbb1cfc7842916d69"),
    "name": "Samuel Kim",
    "position": "Customer Service Representative",
    "department": "Customer Service",
    "salary": 80000
},
{
    "_id": ObjectId("6090196dbb1cfc7842916d6a"),
    "name": "Anna Lee",
    "position": "Graphic Designer",
    "department": "Design",
    "salary": 85000
}])
```

```
db.employee.find({name:"John Doe"})
[
  {
    _id: ObjectId("6090196dbb1cfc7842916d61"),
    name: 'John Doe',
    position: 'Software Engineer',
    department: 'Engineering',
    salary: 90000
  }
]


db.employee.find({position:'IT Specialist',"name": "Mike Davis"})
[
  {
    _id: ObjectId("6090196dbb1cfc7842916d68"),
    name: 'Kelly Brown',
    position: 'IT Specialist',
    department: 'Information Technology',
    salary: 95000
  }
]

 db.employee.find({$or:[{name: 'Mike Davis'},{position: 'IT Specialist'}]})
[
  {
    _id: ObjectId("6090196dbb1cfc7842916d67"),
    name: 'Mike Davis',
    position: 'Operations Manager',
    department: 'Operations',
    salary: 120000
  },
  {
    _id: ObjectId("6090196dbb1cfc7842916d68"),
    name: 'Kelly Brown',
    position: 'IT Specialist',
```

```
    department: 'Information Technology',
    salary: 95000
  }
]
```

Sort:

When we pass 1, that means we are sorting the data in ascending order.
And when we pass -1 that means we are sorting in descending order.

```
db.employee.find().sort({salary:1}) #Sorts in ascending
db.employee.find().sort({salary:-1}) #Sorts in descending


db.emp.find({salary:{$gte:100000}}) #Greater than

db.emp.find({salary:{$lt:100000}}) #Less than

db.employee.find({salary:{$in:[80000,85000]}})
#In operator - Will return values specified in the array

db.employee.find({name:{$regex:/Kim/i}})
Here i, is case insensitive flag.
Ie. It will find for any of the occurrence of Kim in upper or lower case

db.emp.find({$and:[{salary:85000},{name:'Anna Lee'}]})

skipfil
db.emp.find({$where:"this.salary>100000"})
All operator
db.employee.find({name:{$all:["Anna Lee"]}})
```