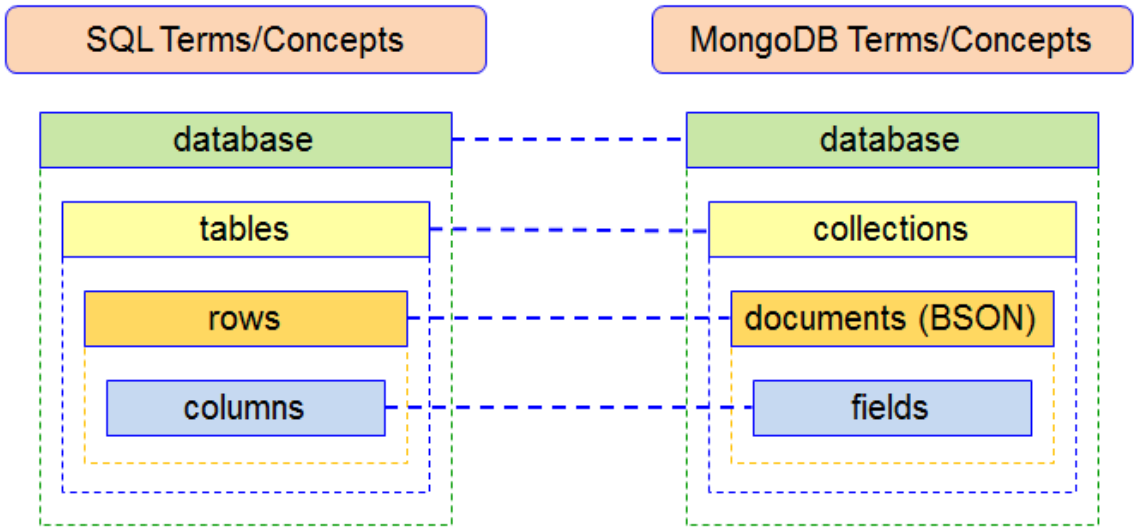
MongoDB

- A DBMS provides the mechanism to store and retrieve the data. There are different kinds of DBMS available in market
  - SQL (Relation)
  - NoSQL (Non-Relational)
- MongoDB is a popular NoSQL database
- It is used to deal with huge amount of data and unstructured data.
- It is open-source database and document-oriented database written in C++ & JS
- MongoDB provides High performances, high availability and automatic scaling



Important terms

Database:

- Database is a physical container for collections. Each database gets its own set of files on the file system. A single Mongodb server typically has multiple databases.

Collection

- Collection is a group of documents and is similar to an RDBMS table.
- A collection exists within a single database.
- Collections do not enforce a schema. Documents within a collection can have different fields.

Document:

- A document is a set of key-value pairs.
- Documents have dynamic schema.
- Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

Mongo DB Installation Setup:

- docker run -d -p 27019:27017 --name=mongo-example mongo:latest
- docker exec -it mongo-example mongo

GUI –Mongo Compass

MongoDB Compass Download | MongoDB

Mongo Commands

Show Default Databases in Mongo

```
> show dbs
MYDB    0.000GB
admin   0.000GB
config  0.000GB
local   0.000GB
>
```

View Current Database

```
> show dbs
MYDB    0.000GB
admin   0.000GB
config  0.000GB
local   0.000GB
> db
test
>
```

Create your own Database

```
> use springboot_db
switched to db springboot_db
>
```

Create New Collection

If we insert some data then automatically product collection is created in db

```
> db.product.insert({"id":501,"name":"Mi 4Phones"})
WriteResult({ "nInserted" : 1 })
>
```

```
> show collections
product
```

If we want to create a collection, use below command

cmd > db. createCollection(name)

Example : db.createCollection("employee")

Name: is a string type, specifies the name of the collection to be created.

View Existing Collections:

```
> show collections
Employee
product
>
```

Insert Data into New Collection

In MongoDB, the db. collection.insert() method is used to add or insert new documents into a collection in database

```
> db.Employee.insert({ "id":501,"name":"Ramesh","sal":45000});
WriteResult({ "nInserted" : 1 })
```

Drop Collection

```
> show collections
Employee
product
> db.product.drop()
true
> show collections
Employee
```

How to view Existing Records in Collection

```
> db.Employee.find().pretty()
{
        "_id" : ObjectId("621c536b360e34039b1f668d"),
        "id" : 501,
        "name" : "Ramesh",
        "sal" : 45000
}
>
```

insert_json.txt

Update Existing Document

```
> db.course.update({'category':'PL'},{$set:{'category':'python'}})
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
```

```
> db.emp.update( {"email_address":""}, {$set: {"email_address":"dummy@gmail.com"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.emp.updateMany( {"email_address":""}, {$set: {"email_address":"dummy@gmail.com"}})
{ "acknowledged" : true, "matchedCount" : 2591, "modifiedCount" : 2591 }
> db.emp.updateMany( {"founded_day":1}, {$unset: {"founded_day":""}})
{ "acknowledged" : true, "matchedCount" : 1251, "modifiedCount" : 1251 }
>
```

Delete Document in Collection

In Mongodb, db.collection.remove() is used to delete a documents from a collection.

Remove All Documents

```
> db.course.remove({})
WriteResult({ "nRemoved" : 2 })
```

Remove a Single Document that matches a condition:

```
> db.course.remove({ 'category' : 'python'},1)
WriteResult({ "nRemoved" : 1 })
```

```
> db.course.find({},{_id:1})
{ "_id" : ObjectId("621ccfc44f4eee77c806425c") }
{ "_id" : ObjectId("621ccfc44f4eee77c806425d") }
{ "_id" : ObjectId("621ccfc44f4eee77c806425e") }
> db.course.find({},{Course:1})
{ "_id" : ObjectId("621ccfc44f4eee77c806425c"), "Course" : "Java" }
{ "_id" : ObjectId("621ccfc44f4eee77c806425d"), "Course" : ".Net" }
{ "_id" : ObjectId("621ccfc44f4eee77c806425e"), "Course" : "Web Desinging" }
> db.course.find({},{Course:1,category:2})
{ "_id" : ObjectId("621ccfc44f4eee77c806425c"), "Course" : "Java", "category" : "Programming Lanague" }
{ "_id" : ObjectId("621ccfc44f4eee77c806425d"), "Course" : ".Net", "category" : "Programming Lanague" }
{ "_id" : ObjectId("621ccfc44f4eee77c806425e"), "Course" : "Web Desinging", "category" : "Programming Lanague" }
>
```