Spring Data MangoDB

==========

(spring data jpa is given to interact with SQL Dbs/w.. and

and spring data provides seperate sub module to interact with each NO SQL Dbs/w)

=> Spring Data provides unified model to work with both SQL and NoSQL DB s/ws...

=> SQL DB s/ws are oracle, postgreSQL, mysql and etc.. (Db tables based)

=> No SQL DB s/ws are MongoDB, cassendra, couchbase, neo4j and etc.. (key-value based, is

=> if data having fixed format and schema the go for SQL DB s/ws

document based,

graph based and etc..)

=> if data the is dynamically growing and no fixed format then go for NoSQL DB s/w Difference between SQL and NoSQL

Parameter

Definition

Design for

/Query Language

/Type

Schema

Ability to scale

Examples

Best suited for

Hierarchical data storage

Variations

Development Year

Open-source Consistency

Best Used for

Importance

✔ Best option

/ Hardware

/Network

-Storage Type

Best features

Top Companies Using

ACID vs. BASE Model

SQL

SQL databases are primarily called RDBMS or Relational Databases

Traditional RDBMS uses SQL syntax and queries to analyze and get the data for further insights. They are used for OLAP systems.

Structured query language (SQL)

SQL databases are table based databases

SQL databases have a predefined schema

SQL databases are vertically scalable

Oracle, Postgres, and MS-SQL.

An ideal choice for the complex query intensive environment.

SQL databases are not suitable for hierarchical data storage.

One type with minor variations.

It was developed in the 1970s to deal with issues with flat file storage

A mix of open-source like Postgres & MySQL, and commercial

like Oracle Database.

It should be configured for strong consistency.

RDBMS database is the right option for solving ACID problems.

It should be used when data validity is super important When you need to support dynamic queries Specialized DB hardware (Oracle Exadata, etc.) Highly available network (Infiniband, Fabric Path, etc.) Highly Available Storage (SAN, RAID, etc.) Cross-platform support, Secure and free Hootsuite, CircleCI, Gauges,ICICI Bank

ACID(Atomicity, Consistency, Isolation, and Durability) is a standard for RDBMS

NOSQL

Spring data mongodb

spring data cassendra

spring data couchbase spring data neo4j

note::: if the attributes(like cols) of the information/records are fixed then prefer working with SQL DB s/ws. eg:: employee with max of 30 attributes

if the attributes (like cols) of the information/records are not fixed then prefer working with NO SQL DB s/ws /category

eg: products that are having different attributes in different situation

NoSQL databases are primarily called as Non-relational or distributed database

NoSQL database system consists of various

kind of database technologies. These databases were developed in response

to the demands presented for the development of the modern application.

No declarative query language

NoSQL databases can be document based,

key-value pairs, graph databases

NoSQL databases use dynamic schema for unstructured data.

NoSQL databases are horizontally scalable

MongoDB, Redis,, Neo4j, Cassandra, Hbase.

It is not good fit complex queries.

More suitable for the hierarchical data store as it supports key-value pair method.

Many different types which include key-value

stores, document databases, and graph databases.

Developed in the late 2000s to overcome issues and limitations of SQL databases.

Open-source

It depends on DBMS as some offers strong consistency like MongoDB, whereas others offer only offers eventual consistency, like Cassandra. NoSQL is a best used for solving data

availability problems

Use when it's more important to have fast data than correct data

Use when you need to scale based on changing requirements Commodity hardware

Commodity network (Ethernet, etc.)

Commodity drives storage (standard HDDs, JBOD)

Easy to use, High performance, and Flexible tool. Airbnb, Uber, Kickstarter

Base (Basically Available, Soft state, Eventually Consistent) is a model of many NoSQL systems

OLAP:: Online Analyticall Processing

MongoDB

==========

=> It is Docucment based NO SQL DB s/w.

=> It internally maintains the data in the form of JSON documents

JSON :: Java Script Object Notation..

(key: value pairs)

Examples of JSON Data...

Document open &

Document closing

{

"Name": "Kim",

"address": {

"emp_details": [

{

Sub doc1

"emp_name": "Shubham",

"email": "ksingh.shubh@gmail.com", "job_profile": "intern"

"streetAddress": "1st Street",

Sub Document

"city": "Seoul",

represents array of

"emp_name": "Gaurav",

"postalCode":

Sub doc2

Single doc

"email": "gaurav.singh@gmail.com", "job_profile": "developer"

sub docs

}.

"phoneNumbers": [

"111-222-333",

key with array of values

Sub doc3

"555-666-Z

"emp_name": "Nikhil",

"email": "nikhil@geeksforgeeks.org'

"profession": "Software Emloyee", "salary":99999

]

]

}

JSON (Java Script Object Notation) is format defining text data

as key=value pairs alternate to the tag based xml format

=>XML, JSON are the two global formats to pass data between two incompitable Apps with out worrying about their hardware setup, software setup, Operating System, network setup and etc..

Json Document

=>Xml docs are tag based docs and they are cross platform docs holding the text data .. but in xml docs compare to data...the tags decoration more..

xml document

==========

<students>

(Best)

<student>

<sno>101</sno>

<sname>raja</sname>

{

{

"sno": 101,

<avg> 45.66 </avg>

"sname": "raja",

</student>

<student>

"avg" : 45.66

}

<sno>102</sno>

<sname>rajesh</sname>

{

Here data and docoration

<avg> 46.66 </avg>

"sno" : 102,

are equally managed.

"sname": "rakesh",

</student>

</students>

"addrs": "hyd",

(Data is less, but tags decoration

}

is more)

}

In MongoDB (physical NoSQL Db s/w) |---> Logical DB

|---> Collection (like db table)

|--->doc1

|-->doc2

db table records

These are like

|--->doc 3 (json docs)

=>Each MongogDB physical Db s/w can have 0 or more Logical DBs => Each Logical DB can have 0 or more Collections (like db tables) => Each Collection can have 0 or more documents (like records)

=> Each Doc can have 0 or more "key": "value" pairs (like col values)

Logical DB is a Logical Partition of physical DB s/w.. Every Logical DB will be taken on 1 per each Project basis.. Every Logical DB is identified with name

MangoDB s/w installation

MongoDB download::

https://www.mongodb.com/try/download/community --> select current version (6.0.3) ---> select widows --> select msi ---> download.

(7.0.8)

↓

Install like any other

l/shell

windows software..

note:: we have multiple MongoDB gui Tools or clients s/ws like compass, Robot3T and etc..

To start MongoDB software to open mongoDB shell

Studio 3T

Go to C:\Program Files\MongoDB\Server\4.4\bin a and use mango.exe file...

SQL terminlogies

Physical DB s/w (like oracle)

|---->Logical DB1 (SID)

I

|---->db table1 (cols and rows)

|---->db table2 (cols and rows)

|---->db table3 (cols and rows)

|---> Logical DB2 (SID)

7.0.8

Db table and cols maintain records/rows

NoSQL MongoDB terminologies Physical Db s/w (like MongoDB) |-----> Logical DB1 (db name)

Generally the MongoDB s/w

will be started automatically

along with OS

(check using services exe file)

|---->Collection1 (document1, document2,.. (json docs) |---->Collection2 (document1, document2,.. (json docs) |----->Logical DB2 (db name)

=> From MongoDB 6.x, the MongoShell does not come along with the installation i.e it must be installation seperately

collections maintain documents (equals to recrods)

https://www.mongodb.com/try/download

=>When we insert document to collection the mongoDb

dynamically generates unique ID as Hexadecimal number as the document id value

Binary --> (Base2) ::01

Decimal --> (Base10) :: 0-9

Octal --> (Base8) :: 0-7

Hexa ---> (Base16) --> 0-9, a-f

Doc N Doc 2 Doc 1

Collection

_id: (ObjectID: 60078de7de61029897aadf studentName: "Viraj",

mongosh-2.1.1-win32-x64

X

+

reg No: "7007",

(dynamically generated unique id value)

↑ C

A collection can have

course: "BTECH",

› Downloads > mongosh-2.1.1-win32-x64 mongosh-2.1.1-win32-x64 >

Mongodb Shell commands

multiple documents

Mongoshell is like SQL Prompt /SQL Plus

1) add <MongoDB_home>\bin folder to PATH env... variable

This PC --->properties ---> advanced system setttings --> env.. variables -->system variable ---> name: PATH

value::

C:\Program Files\Java\jdk1.8.0_31\bin

C:\Program Files\Java\jdk-13.0.2\bin

C:\Program Files\MongoDB\Server\4.4\bin

(old versions)

address: "Delhi"

--> ok --> ok -->ok

2) Open MongoDB Shell using mongo.exe file from any command prompt..

cmd> mongo

(from any location)

note: Logical DBs are logical partiations of the

physical Db s/w that are created on 1 per Project basis

New v

Sort✓ View

Extract all

Name

Type

Compressed size

Password pr

Home

bin

File folder

Gallery

new version

OneDrive

LICENSE-crypt-library

File

6 KB No

LICENSE-mongosh

File

4 KB No

=>extract the MongoDB MongoShell software zip file and use Mongos.exe file to launch the mongoShell

Desktop

mongosh.1

Compressed Archive Folder

7 KB No

(Do this in the new versions

README

File

1 KB No

Downloads

of MongoDB (from 6.x version)

THIRD PARTY NOTICES

File

122 KB No

Documents

3) To list all logical Dbs.. cmd> show dbs

4) To create new Logical DB

cmd> use NTSP715DB switched to db NTSP715DB

5) To know current logical DB cmd> db

NTSP715DB

> show dbs;

admin config local

0.000GB

0.000GB 0.000GB

These are default Logical DBs

> use ntsp715db

> show dbs

admin

switched to db ntsp715db

0.000GB

config

local

DB will not appear in the list of Logical Dbs

0.000GB

(new Logical DB

0.000GB

is not appearing here)

> -

Unless and untill we add collections

to Logical Db, the newly created Logical

6)To create collection with one document..

cmd> db.customer.insertOne({cno:1001,cname:"raja",cadd:"hyd",billAmt:89000.0})

cmd> db.customer.insertOne({cno:1002,cname:"rajesh"})

{

"acknowledged": true,

"insertedId": ObjectId("60078eb8de61029897aadf29")

{

}

"acknowledged": true,

cmd> db.customer.insertOne({cno:1003,cname:"rajesh",mobileNo:987677445})

"insertedId": ObjectId("60078de7de61029897aadf28")

dynamically generated

{

}

"customer" collection is created

PK column kind of property

"acknowledged": true,

7) To list out documments of the collection cmd> db.customer.find()

with unique value.

"insertedid" : ObjectId("60078edfde61029897aadf2a")

}

27017

{"_id" : ObjectId("60078de7de61029897aadf28"), "cno" : 1001, "cname" : "raja", "cadd" : "hyd", "billAmt" : 89
}

cmd> db.customer.find().pretty()

{

"_id" : ObjectId("60078de7de61029897aadf28"),

"cno" : 1001,

"cname": "raja",

note:: In recent versions pretty() is optional to use

"cadd" : "hyd",

"billAmt": 89000

}

8) To insert many documents at once to a Collection

> db.customer.insertMany([{cno:567,cname:"ramesh",cadd:"hyd"},{cno:789,cname:"rakesh"}])

{

doc1

"acknowledged": true "insertedIds":[

ObjectId("60078fb1de61029897aadf2b"),

ObjectId("60078fb1de61029897aadf2c")

1

}

doc2

I

↑

D

Q

Π

>

Downloads >

mongosh-2.1.1-win32-x64 >

mong

↑ Sort

View

E

Name

Type

ت

mongosh

Application

mongosh_crypt_v1.dll

Application extension

Please enter a MongoDB connection string (Default: mongodb://localhost/): 27017

Current Mongosh Log ID: 6614008aa51ad763d816c9b4

Connecting to:

osh+2.2.3

Using MongoDB:

Using Mongosh:

mongodb://127.0.0.1:27017/27017?directConnection=true&serverSelectio

7.0.8

2.2.3

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

The server generated these startup warnings when booting

2024-04-08T19:54:48.289+05:30: Access control is not enabled for the database. Read and w

iguration is unrestricted

27017>

for reference ::

https://docs.mongodb.com/manual/tutorial/insert-documents/

9) To insert document to a collection with array values..

> db.customer.insertOne({cno:1003,cname:"rajesh",mobileNo:[987671245,6788886777]})

{

"acknowledged": true,

"insertedId" : ObjectId("6007911ade61029897aadf2d")

}

To list all the collections of

a Logical DB

> show collcetions

> db.customer.find({cadd:"hyd"}).pretty()

{

10) To find docs of a collection with condition

> db.customer.find({cadd:"hyd",cname:"raja"}).pretty()

NITDB> db.customer.find({$or:[{"billamt": {$gte:4000.0,$lte:10000.0}},{"addrs": "hyd"}]})

[

{

"_id" : ObjectId("60078de7de61029897aadf28"),

{

{

"cno" : 1001,

> db.customer.find({ bil1Amt: {$gte:2000, $1te:40000 }}).pretty() "_id" : ObjectId("630c23482b5ad64dd17cd16c"),

_id: ObjectId('68312e9858abc8227ecdcdf6'), cno: 1001,

"_id" : ObjectId("60078de7de61029897aadf28"),

"cno"

: 1001,

cname: 'raja'

"cname": "raja",

"cno" : 1001,

"cname"

: "raja",

billamt: 9000

"cadd"

"hyd"

"cadd" : "hyd",

"cname": "raja",

"billAmt": 9000

}

"billAmt": 89000

"cadd" : "hyd",

{

}

"billAmt": 89000

"_id" : ObjectId("630c251a2b5ad64dd17cd16f"),

"cno" : 4567,

_id: ObjectId('68312ef758abc8227ecdcdf7'),

{

}

"_id" : ObjectId("60078fb1de61029897aadf2b"),

"cname"

: "rajesh",

"billAmt" : 6000

"cno" : 567,

"cname": "ramesh",

"cadd" : "hyd"

To update the document

==========

}

set clause modifications

11) To delete mongodb document

> db.customer.remove({cno:1001}) To Remove

WriteResult({ "nRemoved" : 1 })

where cluase cond db.customer.updateOne({cname:"raja"},{$set:{billAmt:20000}})

"matchedCount

{ "acknowledged" : true,

multiple docs

: 1, "modifiedCount": 1 }

billamt: 7891,

NITDB> db.customer.updateMany({"addrs":"hyd"}, {$set:{"billamt":10000.0}})

cno: 1002,

cname: 'rajesh',

addrs: 'hyd',

billamt: 7899

},

{

_id: ObjectId('68312f1458abc8227ecdcdf8'),

cname: 'suresh',

addrs: 'vizag',

removes

> db.customer.deleteOne({cno:1002})

only 1st matching

{ "acknowledged" : true, "deletedCount": 1 }

> db.customer.delete Many({cno:1003})

{ "acknowledged" : true, "deletedCount": 2 }

> db.customer.find().pretty()

_id key holds dynamically generated

{

"_id" : ObjectId("60078fb1de61029897aadf2b"),

"cno" : 567,

"cname": "ramesh",

"cadd" : "hyd"

}

{

unique value as pk VALUE.

_id is fixed property for documents

added to a collection... That propery

values (hexa decimal values) are

unique values and they will be used

to indentify the docs..

"_id" : ObjectId("60078fb1de61029897aadf2c"),

}

acknowledged: true,

insertedId: null,

matchedCount: 2,

modifiedCount: 2,

upsertedCount: 0

"cno": 789,

For MongoDB shell commands

"cname": "rakesh"

=====

=======

}

16) To drop logical DB

https://docs.mongodb.com/manual/reference/mongo-shell/

12) To remove collection

> db.dropDatabase();

{ "dropped":"ntspbms615db", "ok": 1 }

> db.customer.drop()

true

13) To list all collections

> db

ntspbms615db

> show dbs;

NTSPBMS714DB1 0.001GB

>> show collections

admin

0.000GB

config

0.000GB

14) To switch to other logical DB

local

0.000GB

>use CollegeDB

ntsp713DB

0.000GB

=>In spring Data JPA, we need to devleop @Entity class to map with db table and we have generators to generate id property value dynamically. if do not use Generators we can manually assign value to id proepery.

=> In spring data MogoDB,we ened to devleop @Document class to map with Collection of MongoDb Db s/w.. Here there are no value to Generators to generte the values for id property.. So either we need to assign unique or random value id property manually or we need to allow to store the MongoDB generated hexa decimal unique value to Id property.

Version1::: (Storing mongoDB generated _id property value (hexadecimal value)

(Best)

to @Id property of @Document class)

@Document /

@Data (lombok api)

public class Student{

@Id

property name must be "id" and type must be "String"

of

private String id; to hold value generated for the_id property of JSON doc to this "id" property @Document class

private Integer sho;

private String sname;

Student st=new Student();

private String sadd;

example inserted JSON document

st.setSno(678); st.setName("mahesh"); st.setSadd("vizag");

{

}

"_id" : ObjectId("62012bfb4a76b1a1ead13923"),

"sno" : 678,

"sname": "mahesh",

"sadd" : "vizag"

}

=>To work with MongoDB s/w generated id value which is unique hexadecimal number we need to take property name as the "id" and property type as the "java.lang.String"

=> The MongoDB s/w uses that following 4/5 details to generate the

hexa decimal as the unique value

Version2: (Make MondoDB to use the programmer assigned /generated value to @Id property of Document obj as

========= id value for each inserted json Document)

a) System IP address b) Process id c) thread id d) system date and time

(not recomanded)

@Document

@Data

public class Student{

@Id

S

Student st=new Student();

st.setSno(678);

private Integer sno;

private String sname;

private String sadd;

st.setName("mahesh");

st.setSadd("vizag");

example inserted JSON document

}

"sno" 678 ==> this becomes id value "sname": "mahesh",

"sadd" : "vizag"

}

Now duplicate "sno"

will give error.

=>we can use Custom Generator or Third party Generator to store the generated id

value to id property.

=> In spring data mongodB, we do not have readymade generators, So we can use

either custom generators or third party generators to generate the id value dynamically

=> Create @Document class by keeping multiple possible properties.. but be flexible to add more properties to the @Document class

as the MongoDB documents are dynamically growing docs