Revisit

MySQL

1. RDBMS – Relational Database Management System
2. MySQL, Oracle, Cybase, MS-SQL
3. MySQL is a Open Source RDBMS Community & Enterprise Edition
4. This is a good DB for development & Testing purpose
5. Client – Server, MySQL Server, MySQL Command Line Client, Workbench, MySQL Shell, Connector
6. Accessing MySQL using command line, gui(workbench) & programming Lang [c,c++,java,python,php,js]
7. SQL – Structured Query Lang
8. Relation – In mathematics, Union, intersect, venn diagram
9. .txt, .rtf – Flat files, data is organized and stored in the form of rows & columns
10. The intersection of a row & column is called as cell.
11. Row is also known as record or tuple
12. Column will have same data for different person/object/entity
13. In RDBMS, everything is entity
14. Entity – Table, View, Query, Functions, Stored Procedures, Trigger, Sequnces, Materialized views, cursor, pl/sql
15. DDL, DML, DCL, TCL, DQL
16. ER- Diagram (Entity Relationship Diagram) – Using MySQL Workbench
17. Accessing the db entities using command client (mysql -d root -p, show databases, conn <db\_name>, show tables, select \* from <table\_name>,
18. CRUD – Create/Insert, Read, Update, Delete.
19. Normalization – It’s a process of reducing redundancy in the data. (duplicate\_values, empty cells, null values) FirstNormalForm(1NF), SecondNormalForm(2NF)….BCNF(Boyce-Codd NormalForm)…
20. Writing Queries, Sub Queries, Joins (inner join, outer join)
21. Constraints (Primary Key, Foreign Key Constraint, Check, Not Null, Unique)
22. Relationship between tables (One to One, One-to-Many, many-to-many)
23. Clause – Where, like, group by, order by, in, any, between
24. Built-in Functions – Mathematical, String, Group functions etc.,
25. Sub-Queries – Query Within a Query (The inner most query will get executed first)

Java

1. Java is a High Level, OOP based, Multi Threaded, highly secured, auto memory managed general purpose programming lang.
2. Keywords- Reserved Words (if, else, for, while, class, enum, interface, finally, throw, throws, extends, implements….)
3. Data types ( i) primitive (boolean, byte, char, short, int, long, float, double) ii) Derived Data types [arrays, objects, collections])
4. Rich support of operators (+,-,\*,/,%, ~!#@, ::, ||, (), [], {}, <>)
5. Statements Normal, Conditional Control Statements, Looping / Repetitive Statement
6. Installing JDK
7. Adding the Java to the path environment table. (JAVA\_HOME)
8. JDK , JRE & JVM (These are all not platform in-dependant)
9. Jdk – javac, java
10. JRE – java (Using this exe file we can invoke JVM in any system
11. Installed Eclipse EE version
12. Created a new Java project
13. Executed from the command prompt
14. Classes & Objects

Mongo DB – No-SQL Database ( Client – Server Concept === Request & Response object)

No-SQL DB – It won’t store the data in Table format.

It’s open source, cross-platform, No SQL DB.

It’s written in C++

Adv

1. High Performance
2. High Availability
3. Automatic Scaling

Mongo DB – Document based Database

Documents -- table

Set – rows,

JSON – BSON

DB Types

1. RDBMS – Relational Database Management System)
2. OLAP – Online Analytical Processing
3. NoSQL – Document/Graph based Database (MongoDB/Neo4J) [Non-SQL/Non-Relational]

In MongoDB, javascript is used to perform any operation.

MongoDB is a Schema Less (db less) database.

It’s a document based in which one collection can hold different documents.

There may be difference between no of fields, content, size of the document

No complex joins

Deep query

1. Complex data – social media , mobile
2. Content mgmt. & delivery

|  |  |  |
| --- | --- | --- |
| Sl No | RDBMS | No-SQL |
| 1 | Table | Collection |
| 2 | Schema/Database | Document |
| 3 | Data Types support | BSON format – similar to JSON |
|  |  |  |

Datatypes supported in MongoDB

1. String
2. Integer
3. Boolean (YES/NO)
4. Double
5. Min/Max
6. Arrays
7. Object
8. Null
9. Symbol
10. Date

Accessing MongoDB

1. Using GUI – MongoDBCompass
2. Using CLI – open command prompt & type “mongo” – Mongo cli

Social Media App – Post, Comments, tags\_list

Post – id, title, description, url, likes, posted\_by

Comments – zero or more comments, contains username, message, date-time, likes

Tags – one or more tags

{

\_id:POST\_ID,

Title: TITLE\_OF\_POST,

Description : POST\_DESCRIPTION,

By: POSTED\_BY,

url : URL\_OF\_POST,

tags : [TAG1, TAG2, TAG3 ….],

likes: TOTAL\_LIKES,

comments : [ {

user:Comment\_by,

message : text,

dateCreated :DATE\_TIME,

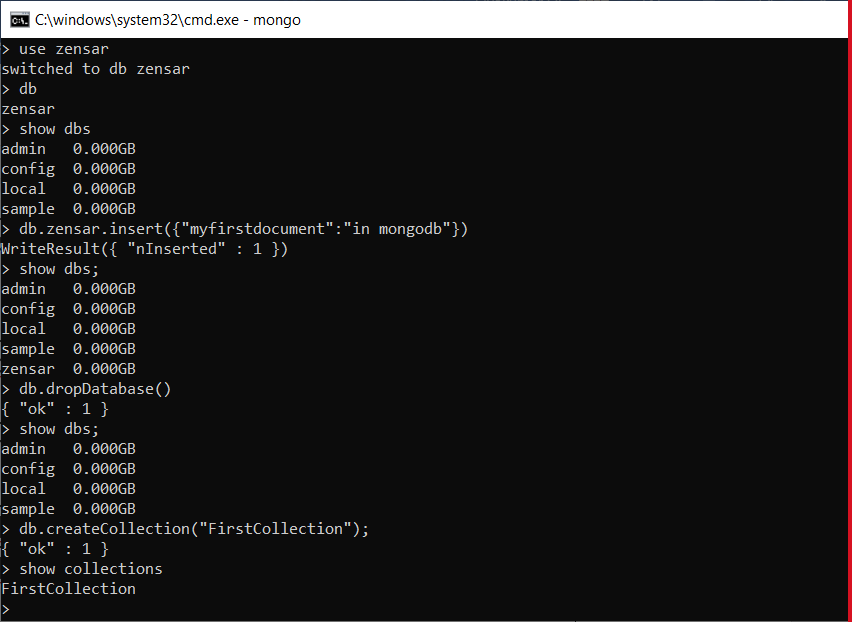
like : LIKES

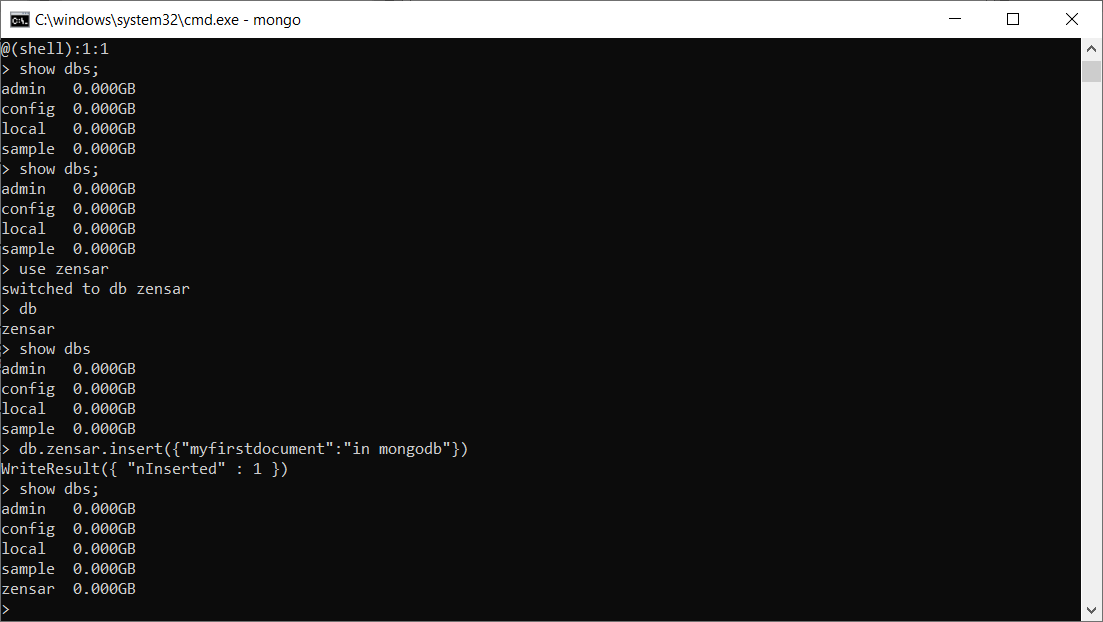
}

]

}

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No | Commands | Usage | Example |
| 1 | Use <db\_name> | To create or connect to a database in mongo | Use Zensar |
| 2 | Db | To display currently selected database | db |
| 3 | Show Dbs | To display all the databases | Show Dbs |
| 4 | Db.db\_name.insert | To add/insert a data i | Db.zensar.insert({“id”:200,”name”:”Test”}); |
| 5 | dropDatabase() | To drop/delete a database | Db.dropDatabase() |
| 6 | Db.createCollection (name,options) | To create a collection | Db.createCollection(“myCollection”) |
| 7 | showCollections |  |  |
| 8 | Db.collection.find() |  |  |





{} – Represents object

[] – array

Key:value

|  |  |  |  |
| --- | --- | --- | --- |
| Create /Insert | Db.collection.insert() /upsert()/save() | Db.mycollection.insert({  Course : “FSD”, modules: { “generic” : “2 months”, topics:”html,css,js”},  Batch :[ {size:”small”,count:15},  {size:”medium”,count:20}]  }) |  |
| Read/Retrieve |  | Db.mycollection.find() |  |
| Update/Edit | Update(selection\_criteria, updated\_data) | Db.mycollection.update({`course`:’FSD’},{$set:{‘course’:’MEAN’}}); |  |
| Delete/Remove | Remove(DELETION\_CRITERIA) |  |  |

MongoDB is case sensitive No-SQL DB

Var myObject = { ‘firstName’ :’abc’,’lastName’:’xyz’});

db.myCollection.insert(myObject);

db.myCollection.find();

db.mycollection.remove({batch:’small’},1);

db.mycollection.insert({

course: "FSD",

modules:{"generic" :"2 months", topics:"html,css,js"},

Batch :[

{size:"small",count:15},

{size:"medium",count:20}

]

});

var myObject = {firstName :"abc",lastName:"xyz"};

db.myCollection.insert(myObject);

db.myCollection.find();

db.mycollection.update({'course':'FSD'},{$set:{'course':'MEAN'}});

db.mycollection.remove({'firstName':'abc'});

MongoDB Types

1. Local Instance (Installed in the System)
2. Cloud Instance (No need to Install Mongo Server – Available in cloud) – Atlas

mongodb://127.0.0.1:27017/?readPreference=primary&appname=MongoDB%20Compass&directConnection=true&ssl=false

mongodb+srv://<username>:<password>@cluster0.hfeko.mongodb.net/test

mongodb+srv://root:root@cluster0.hfeko.mongodb.net/test

