Agenda

1) NoSQL
2) MongoDB
3) CRUD, Index creation
4) CAP Theorem

Data – Information (Useful data)

1) Structured Data
2) Un-Structured Data

Database - A database is an organized collection of structured information, or data, typically stored electronically in a computer system

DBMS – Database Management System (It's a Computer Software to store and manage the data effectively)

RDBMS – Relational Database Management System –

Lotus

Foxpro – Lang with DB Support

MS-Access

Popular RDBMS

1) Oracle XE (Express Edition)
2) MySQL (Open Source – Community Edition)
3) Postgres (Open Source – SQL/NoSQL)
4) DB2 (IBM Product)

No SQL – Not Only SQL / Non-SQL - NoSQL is a database design approach that stores and retrieves data in ways other than the tabular relations used in relational databases. NoSQL stands for "non-SQL" or "non-relational

SQL – Structured Query Language (Query based Language for managing Structured Data)

In RDBMS, everything is Entity (Table, Query, Trigger, Procedures, Functions, Sequences)

In Java, everything is Object

Popular NoSQL DB

1) MongoDB
2) Cassandra
3) DynamoDB
4) Redis
5) GraphQL

MongoDB – Document Based DB

Download URL - https://www.mongodb.com/try/download/community

All DB – Client Server Technology

ATLAS – Cloud Version of MongoDB

Local Installation – Community Server of MongoDB installed in the System

MongoDB Compass – It's a GUI/CLI client to interact with MongoDB server

Different ways of Accessing Database

1) Using CLI client (Using Commands) knowledge of commands related to the RDBMS is important
2) Using GUI client (Usually provided by the DB Developers/ Generic One) – [Knowledge of SQL commands is not necessary]
3) Using any Programming Lang [C,C++,Java, Python, etc.,]

IP Address of Localhost 127.0.0.1

Default port number for MongoDB is 27017

Sl No	Database Name	URL	Default Port
1	Oracle XE	oracle:thin://localhost:8080/apex	8080
2	MySQL	mysql://localhost:3306/{db_name}	3306
3	Postgres	postgres://localhost:5432/{db_name}	5432
4	MongoDB	Mongodb://localhost:27017	27017

https://gmail.com

mysql://localhost:3306/{db_name}

oracle:thin://localhost:8080/apex

postgres://localhost:5432/{db_name}

CRUD – Create/Insert Read Update Delete

```
use sapient
```

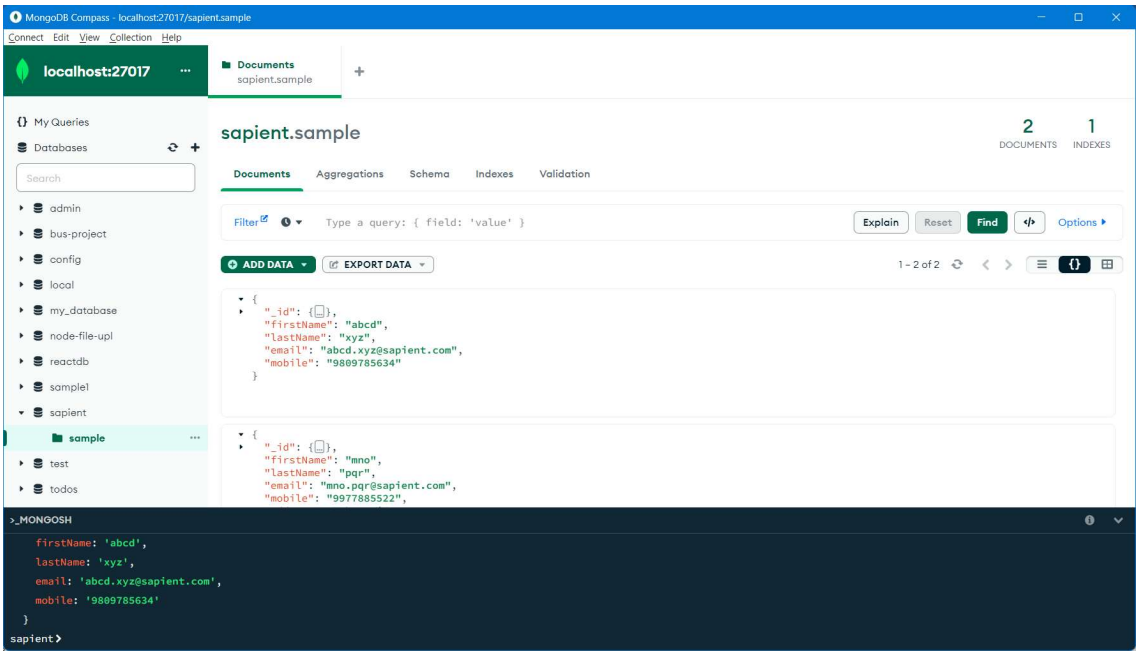
```
db.sample.find()
```

```
{
  _id: ObjectId("65277a3e64ba513dcf4927df"),
  firstName: 'abcd',
  lastName: 'xyz',
  email: 'abcd.xyz@sapient.com',
  mobile: '9809785634'
}
{
  _id: ObjectId("65277b4c64ba513dcf4927e0"),
  firstName: 'mno',
  lastName: 'pqr',
  email: 'mno.pqr@sapient.com',
  mobile: '9977885522',
  address: 'chennai'
}
```

```
db.sample.find({firstName : "abcd"})
```

```
{
  _id: ObjectId("65277a3e64ba513dcf4927df"),
  firstName: 'abcd',
  lastName: 'xyz',
  email: 'abcd.xyz@sapient.com',
  mobile: '9809785634'
}
```

```
sapient
```



Accessing the DB using programming Lang.

JDBC – Java DataBase Connectivity API

API – Application Programming Interface

Connecting Laptop with Printer

1) Install Driver
2) Establish Connection
3) Test Print
4) Actual Print
5) Power Off

Interacting to a DB using Java JDBC Api

1) Load & Register the Driver
2) Establish the Connection
3) Create Queries to perform some operation
4) Execute & Process the Result
5) Disconnect

Download and add Mongodb driver to the Java Project

https://mvnrepository.com/artifact/org.mongodb/mongodb-driver-sync/4.11.0

Index in MongoDB

- Single Index
- Single Index on an Object
- Compound Index

Index increase the performance

Use Case : use index when there are more read operations than write operations

Costly Process/Operation ---- If a process/task/operation consumes more time/resource/memory

```
collection.createIndex( <key and index type specification>,
<options> )

db.collection.getIndexes()
```

BSON – Binary Javascript Object Notation

Employees

Id	Name	Pwd	Email
100	Abc	Abc123	abc@xyz.com
101	Def	Def123	def@xyz.com
102	Ghi	Ghi123	ghi@xyz.com

XML – eXtensible Markup Lang (Tag based) – XML is case and space sensitive. [<html> </HTML> is wrong in xml]

- XML tags are user defined

Employees.xml

<employees>

<employee>

<id> 100 </id>

<name> abc </name>

<pwd> abc123 </pwd>

```
        <email> abc@xyz.com </email>
    </employee>
    <employee>
        <id> 101</id>
        <name> def </name>
        <pwd> def123 </pwd>
        <email> def@xyz.com </email>
    </employee>
    <employee>
        <id> 102 </id>
        <name> ghi </name>
        <pwd> ghi123 </pwd>
        <email> ghi@xyz.com </email>
    </employee>

</employees>
```

Employees.json

```
Employees = [
 {"id":100,"name":"abc","pwd":"abc123","email":"abc@xyz.com"},
{"id":101,"name":"def","pwd":"def123","email":"def@xyz.com"},
{"id":102,"name":"ghi","pwd":"ghi123","email":"ghi@xyz.com"}
];
```

Parser – Parsing

CAP Theorem

ACID – Atomicity Consistency Isolation Durability

Assignment – Simple Java Program to read name from user and display the following message.

“Welcome {name}, to ASDE 2023!!!”