

Date:

Day:

END BACKEN

Start
Mongo

DB

12/4/2014

Lecture 94 Day:

Key Point

- ① what is Database
- ② what is MongoDB
- ③ what is Different between MongoDB and MySQL
- ④ what relational Database and non relational Database
- ⑤ Install MongoDB and also install extension in VS code.

Q) What is Database:

A Database is a structured collection of Data that is organized and stored in a computer system. It is designed to efficiently manage, store, retrieve and manipulate data according to specific requirement. Database are essential component of modern Computer system and are used in various application and industries, including web Development, business operation Scientific research and more.

In a database, data is

Date: _____

Day: _____

typically organized into tables which consist of rows and columns. Each row represents a single record or entity, while each column represents a different attribute or characteristic of the data. The relationships between tables are defined using keys such as primary key and foreign key to establish connections between related pieces of information.

Databases provide several key functionalities, including:

① Data Storage: Databases store data in a structured format making it easy to organize and manage large volumes of information.

② Data Retrieval = User can query the database to retrieve specific pieces of data based on predefined criteria or conditions.

Date: Manipulation \Rightarrow Database allows user to insert, update, and delete data, enabling them to modify the information stored in the database as needed.

Date Security: Database often include features for controlling access to data ensuring that only authorized user can view or modify sensitive information.

Date Integrity \Rightarrow Database enforces rule and constraint to maintain the integrity and consistency of the data preventing error and inconsistencies.

Overall database play a crucial role in organizing and managing data effectively, enabling businesses and to make informed decision, streamline operation and deliver valuable services to this user.

Date: _____

Day: _____

What is MongoDB

MongoDB is a popular, open source NoSQL database program. It falls under the category of document-oriented database which means it stores data in flexible, JSON-like documents rather than traditional table-based structures used in relational databases. MongoDB is designed to provide high performance, scalability, and flexibility for managing structured and unstructured data.

Here are some key features and characteristics of MongoDB:

① Document-oriented → MongoDB stores data in collections where each collection contains documents. These documents are JSON-like objects that can have varying structures, allowing for flexibility in data representation.

② Schemaless → MongoDB is schemaless, meaning that documents

Date: _____ Day: _____
meaning within a collection do not need to have the same structure. The flexibility makes it easy to adopt to evolving data requirements without requiring a predefined schema.

High Performance Scalability:

Rich Query Language.

Replication and High Availability

overall MongoDB is widely used in various industries and application including web development, mobile app development, analytics.

Different Between MongoDB and MySQL.

Data Model.

Mongo DB: Mongo DB is a document-oriented NoSQL database. It stores data in flexible JSON-like document allowing for nested structure and dynamic schemas.

Date: _____

Day: _____

MySQL: MySQL is a relational database management system (RDBMS) based on the SQL (Structured Query language) model. It stores data in structured tables with predefined schemas and enforce relationships between tables using key.

② Query Language

- MongoDB: MongoDB uses a query language called MongoDB Query Language (MQL) which is similar to JSON syntax. It supports a wide range of query operations, including including CRUD (create, Read, update, Delete), aggregation and geospatial queries.
- MySQL: MySQL uses SQL (Structured Query Language) which is a standardized language for managing relational database. SQL provides a powerful and expressive way to query and manipulate data in MySQL database.

Schema:

Day:

Mongo DB: Mongo DB is schemaless or schema-flexible, meaning that documents within a collection do not need to have the same structure. This flexibility allows for easy adaptation to change data requirement without requiring a predefined schema.

My SQL: My SQL is schema-based meaning that tables must have a predefined schema that defines the structure of the data including data types, constraints and relationships between tables.

Use Cases =>

Mongo DB: Mongo DB is well-suited for use cases that require flexibility, scalability and performance for managing unstructured or semi-structured data, such as content management system, real-time analytics and IoT application.

Date:

Day:

- MySQL → MySQL is commonly used for traditional relational database applications such as e-commerce website, financial system and enterprise application that require strong data consistency and relational integrity.

Overall the choice between MongoDB and MySQL depends on factors such as data structure scalability requirement, transactional need and development preference while MongoDB offer flexibility and scalability for handling diverse data types and high traffic volumes, MySQL provide strong data consistency and relation features for traditional relational database application.

What is Relational DataBase and non Relational Database.

Relational DataBase: A relational database is a type of Database is a type of

Database management system (DBMS) that organizes data into tables which consist of rows and columns. Each row represents a record or entity, while each column represents a specific attribute or field of the data. The relationships between tables are defined by keys such as primary key and foreign key.

Non-Relational Database:-

also known as No SQL database store and retrieve data in a format other than tabular relation used in relational databases. They are designed to handle unstructured, semi structured or rapidly changing data.

install MongoDB and Mongo compass
Mongo compass used for
to Run mongoDB.

Once you open Mongo compass

Date: 4/13/24

Day: Saturday

② There is two way to create database One use local host of mongo DB Two you can create Cloud database.

③ First we try localhost we can create New Connection past the URL of localhost
url = mongodb://localhost:27017/ click on the Connect button.

④ There is already predefined database you can create your own database.

⑤ If you want to add data in Database there is also Id

For example:

Write Data is like JSON

```
"_id": {  
  "SocialID": "6654321F423"  
},  
"Name": "C++",  
"Price": "20000",  
}
```

Day: There is a terminal in mongo shell
open.

1. show Database => To view all Database.

2. use Database => To specific which Database you want to use.

③ DB.Course.find() = use for Inside the Course Data.

④ insert() There is two types of Add Data in Data base.

① insertOne({Name: "Javascript"})

② insertMany([{Name: "Khan"}, {Name: "Furqan"}, {Name: "Abid"}])

CRUD operation in MongoDB

Let's break down each CRUD operation in MongoDB.

① Create (Insert):

Purpose => The create operation is used to add new document to a collection.

Method => In MongoDB you can use the "insertOne()" method to insert a single document or the "insertMany()" method to insert multiple document into a collection.

Example =

```
DB.users.insertOne({ name: "John",  
                    age: 30 })
```

```
the insertMany ([ { }, { } ])
```

② Read (Retrieve):

Purpose => The read operation is used to retrieve document from a collection.

Method \Rightarrow In MongoDB, you can use the `find()` method to retrieve document that match specified query criteria. You can also use methods like `findOne()` to retrieve a single document.

Example \Rightarrow

```
db.users.find({age: {$gte: 25}});
```

Update \Rightarrow

Purpose \Rightarrow The update operation is used to modify document in a collection.

Method \Rightarrow In MongoDB you can use the "`updateOne()`" method to update a single document or the "`updateMany()`" method to update multiple document that match a specified query criteria.

Example \Rightarrow

```
DB.user.updateOne({name: "John"},  
{$set: {age: 35}});
```

Date: _____

Day: _____

(4) Delete :

Purpose → The delete operation is used to remove document from a collection.

Method → In MongoDB, you can use the "deleteOne()" method a single document or the "deleteMany" method to delete multiple documents that match a specified query criteria.

Example →

```
db.users.deleteOne({ name: "John" });
```

These are the basic CRUD operations in MongoDB. Each operation allows you to perform specific tasks, creating new document, retrieving existing document, updating document field, and delete document retrieved from a collection. By combining these operations you can manipulate MongoDB collection according to your application requirements.

Date: 4/11/2024

Day:

Query and Projection Operators.

1) Query Operators = Query operators allows you to filter document based on certain criteria with in the find() method to specify that document must meet to be included in the result set.

Here are some common query operators:

Syntax = db.collection.find({age:{\$gt:25}})

Comparison Operators =

- \$eq : Matches value that are equal to a specified value.
- \$gt : Matches value that are greater than a specified value.
- \$lt : Matches value that are less than a specified value.
- \$gte : Matches value that are greater than or equal to a specified value.
- \$lte : Match value that are less than or equal to a specified value.
- \$ne : Match all values that are not equal to a specified value.

Date: _____

Day: _____

- Logical Operators ⇒

- \$and ⇒ Join query clauses with a logical AND.
- \$or ⇒ Join query clauses with a logical OR.
- \$not ⇒ Inverts the effect of a query expression.
- \$nor ⇒ Join query clauses with a logical NOT.

- Element Operators ⇒

used for querying document based on the existence of Fields.

- \$exists ⇒ Match document that contain the Specified Field.

- Array Operators: used for querying document based on Array.

• \$in : Match any of the value specified in an array.

• \$nin : Match none of the value specified in an array.

• \$all : Match array that contain all element Specified in an Array

- Regular Expression Operator used for pattern matching.

• \$regex : Match document that contain a field matching a regular expression.

Date:

Day:

Projection Operators:- projection operators are used to include or exclude fields from the result set. They are used within the `find()` method to specify which field should be included or excluded from the document returned. Some

Include Field:- You can specify that you want to include in the output document. This is achieved by setting the field name to '1'

// Include only the name and age fields

```
{ name: 1, age: 1 }
```

For example:-
`db.collection.find({ age: { $gt: 25 } }, { name: 1, age: 1, _id: 0 })`

Exclude Fields:- Conversely you can specify that you want to exclude from the output document. This is achieved by setting the field name by '0'.

```
{ age: 0 }
```

Date: _____

Day: _____

- ③ control - id - field \Rightarrow By default MongoDB includes the '-id' field in the result set you can explicitly include or exclude it using 0 and 1.
- -id: 0 Exclude the -id field
 - -id: 1 Includes the -Id field.

{ -id: 1 } // includes
{ -id: 0 } // exclude

These projection operator can be used in conjunction with query operators to find - tune the output of your MongoDB query. It's important to note that projection operation operator do not affect the matching of document they only shape the output.

15
Mon 4/11/2024

Lecture 96

Day: Sunday

Mongoose and use it with Express.

What is Mongoose?

Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It provides a higher-level abstraction over the MongoDB Node.js driver, making it easier to work with MongoDB databases by providing a schema-based solution.

Install Mongoose:

NPM: `npm i mongoose`

- 1 Import mongoose from "mongoose";
- 2 Import express from "Express"
- 1① Import the Mongoose library which provides an interface for interacting with MongoDB in Node.js.
- 1② Imports the Express, which is a web application framework for Node.js that simplifies the process of creating web applications and APIs.

③ Import { Script } from "./models/schema"
110 First we create a file
'script.mjs'
| script.mjs

```
import mongoose from "mongoose"  
import { boolean } from "webidl-conver
```

```
const Script Schema = new mongoose.Schema  
(
```

11 we can provide as object type
Data

```
title: { type: String, required: True,  
default: "Hey" },
```

```
Name: String,
```

```
age: Number,
```

```
job: String },
```

```
export const Script = mongoose.  
model ('script', Script Schema)
```

Defining your schema
Every thing in Mongoose starts
with a schema. Each schema
maps to a mongoDB collection
and defines the shape of
the document within the
collection.

Mon 4/16/2024

Day:

- The permitted Schema Type
- (1) String
 - (2) Number
 - (3) Date
 - (4) Buffer
 - (5) Boolean
 - (6) Mixed
 - (7) ObjectID
 - (8) Array
 - (9) Map

Every schema have these type
11 age : Number;

creating a model.

To use our schema definition we need to convert our blogSchema into a model we can work with. To do so, we pass into mongoose.model()

```
const Script = mongoose.model('script',  
    ScriptSchema);
```

Now back to file main.js

```
① let conn = await mongoose.connect  
(`mongodb://localhost:27017/emi`)  
② connect to a MongoDB  
database running locally on  
port 27017 and uses the database  
named "script". This line establishes  
a connection to the MongoDB  
server and return a connection  
object ('conn')
```

Date: _____

Day: _____

```
⑤ app.get('/get', (req, res) =>
  ⑥ const script = new Script({name:
    ⑦ "Furqan", age: 43, job: false})
  ⑧ script.save()
    , res.send('Hello world')
  })
```

⑨ // Const script = new Script({name:
"Furqan", age: 43, job: false})
create a new instance of the
'Script' model with the specified
data (name, age, job)

⑩ // Script.save(); Save the
newly created document to
the MongoDB database.

Show Data in Html file in web

```
⑪ app.get('/q', async (res, req)
  Let scripts = await Script.find({})
  → // use for the (find) method
  to retrieve all document from
  the Script collection.
```

// Return the document as a json
res.json(scripts)

Ques: Why we use Mongoose?
Ans: Mongoose is a popular Node.js library used with MongoDB. There are several reasons why developer choose to use Mongoose.

Simplified Schema-Based Data Modeling:

Mongoose allows developer to define schemas for their data. Providing a structured way to model the application data. Schemas define the shape of document within a collection including the properties, types and validation.

2 Build-In Data Validation:

Mongoose provides built-in data validation capabilities, allowing developer to enforce constraint on the data being stored in the database. This helps ensure data integrity and consistency.

Date: _____

Day: _____

③ Easy CRUD Operation.

Mongoose simplifies common database operation such as creating, reading, updating and deleting (CRUD) document. It provides intuitive API for performing these operation, reducing boilerplate code and making database interaction more straightforward.

④ middleware Support

Mongoose support middleware function which are function that execute before or after certain operation, such as saving a document or removing a document. middleware can be used to implement custom logic validation or data transformation.

⑤ Schema Relationships

Mongoose support defining relationship between schema, such as one-to-one, one-to-many, many-to-many relationship. This enable developer to model complex data structure and perform efficient across related data.

Integration with Express and ecosystem Day:
Mongoose integrates seamlessly with Express.js, a popular Node.js framework for building web applications and APIs that interact with MongoDB. Additionally, Mongoose works well with other Node.js libraries and frameworks, enhancing the development experience.

Community and Ecosystem: Mongoose has a large and active community of developers who contribute to its development and provide support through forums, documentation, and tutorials. There are also many third-party plugins and extensions available for extending Mongoose functions.

Overall, developers use Mongoose to streamline the development of Node.js applications that interact with MongoDB, providing features for data modeling, validation, querying, and more.

01/12/16 Dummy Data Generation

1) Generate a dummy data
in this format in a collection
called employee in a db called
company.

```
name: "Furqan"  
salary: 20000  
language: Python  
city: "Ambiri"  
isManager: true
```

- ① Generate 10 such record
when a button called
generate data is clicked
- ② Create an Express app with
mongoose to achieve it,
- ③ Everytime the button is
clicked you should clear
the collection.