Introduction to MongoDB

Agenda

- Database
- What is SQL?
- What is NoSQL?
- Types of NoSQL
- SQL Vs NoSQL
- When to use NoSQL
- What is MongoDB
- Features of MongoDB
- MongoDB Data types
- Installation
- Commands
- MongoDB compass & MongoDB shell(Mongosh)

Database:

A database is an organized collection of structured or unstructured information stored electronically on a machine locally or in a cloud. These are managed using a Database Management System(DBMS).

Database are used for various data processing operations the most basic being the Create ,Read, Update and Delete(CURD Operations)

What is SQL Database?

SQL stands for Structured Query Language.

SQL lets you access and manipulate databases . It can execute queries against a database.

It can retrive data ,insert , update , delete records in a database.

What is NoSQL Database?

The term "NoSQL database" refer to any "Non-relational" or "Not only SQL" databases provides a mechanism for storage and retrieve data in format other than tabular relations model used in relational databases.

NoSQL database doesn't use tables for storing data.

It generally used to store big data and real-time web applications.

It avoids joins and is easy to scale.

Types of NoSQL

- Key-value pair
- Column-oriented
- Graph-base
- Document-oriented

SQL Vs NoSQL

SQL	NoSQL
Stands for Structured Query Language	Stands for Not only SQL
Relational database management system (RDBMS)	Non-relational database management system
Data is stored in tables with columns and rows	Data is stored in collection or documents
Supports JOIN and complex queries	Does not supports JOIN and complex queries
Vertically scalable	Horizontally scalable
Ex : MySQL, PostgreSQL, Oracle, etc	Ex: MongoDB, HBase, Neo4j, etc

When to use NoSQL

- When a huge amount of data needs to be stored and retrieved.
- The relationship btw the data you store is not that important.
- The data changes over time and is not structured.
- Constraints and joins support is not required at database level.

The data is growing continuously and you need to scale the database regularly to handle the data.

What is MongoDB

MongoDB is a document database designed for ease of application development and scaling**.**

You can run MongoDB in the following environments:

- MongoDB Community : The source-available, free-to-use, and self-managed version of MongoDB.

- MongoDB Shell: The MongoDB shell,mongosh is a JavaScript and Node.js REPL environment for interacting with MongoDB developments in Atlas.

Features of MongoDB

- Support ad hoc queries
- Indexing
- Replication
- Duplication of data
- Load balancing
-

MongoDB Data types

- **String** − This is the most commonly used datatype to store the data. String in MongoDB must be UTF-8 valid.
- **Integer** − This type is used to store a numerical value. Integer can be 32 bit or 64 bit depending upon your server.
- **Boolean** − This type is used to store a boolean (true/ false) value.
- **Double** − This type is used to store floating point values.
- **Min/ Max keys** − This type is used to compare a value against the lowest and highest BSON elements.
- **Arrays** − This type is used to store arrays or list or multiple values into one key.
- **Object** − This datatype is used for embedded documents.
- **Null** − This type is used to store a Null value.
- **Symbol** − This datatype is used identically to a string; however, it's generally reserved for languages that use a specific symbol type.
- **Object ID** − This datatype is used to store the document’s ID.
- **Binary data** − This datatype is used to store binary data.
- **Regular expression** − This datatype is used to store regular expression.

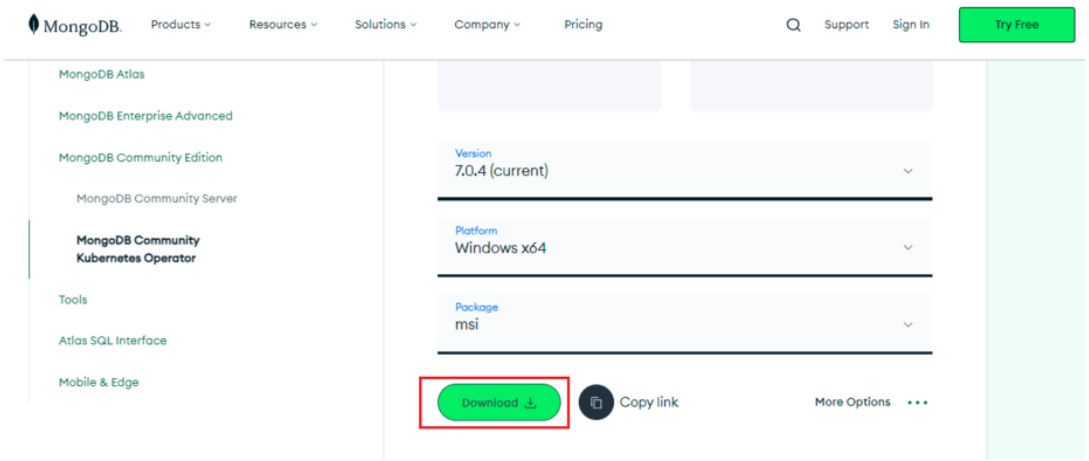
Installation

Steps to install MongoDB on Windows using MSI

To install MongoDB on Windows first download the MongoDB server and then install the MongoDB shell.

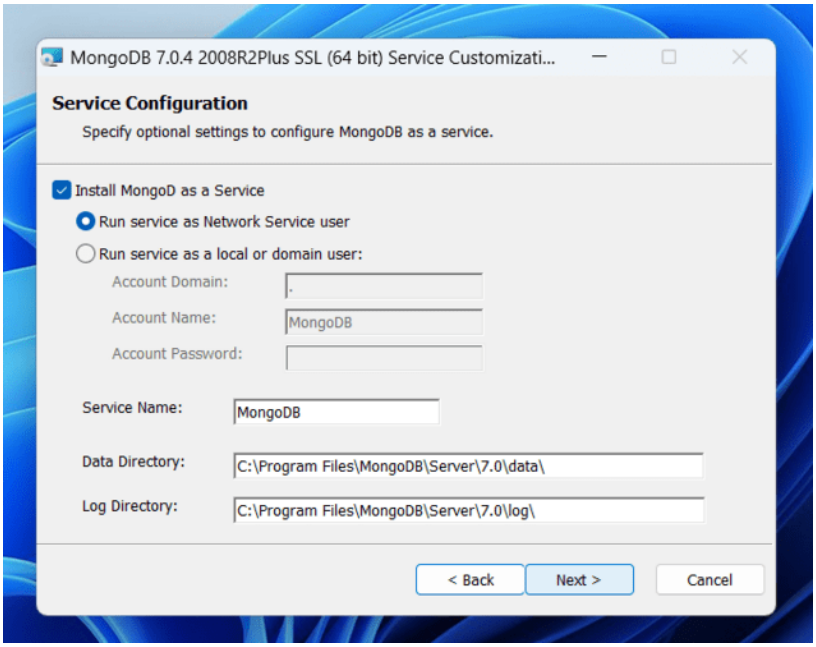
The steps below explain the installation process in detail and provide the required resources for the smooth download and install MongoDB.

Step 1: Go to the [MongoDB Download Center] to download the MongoDB Community Server.

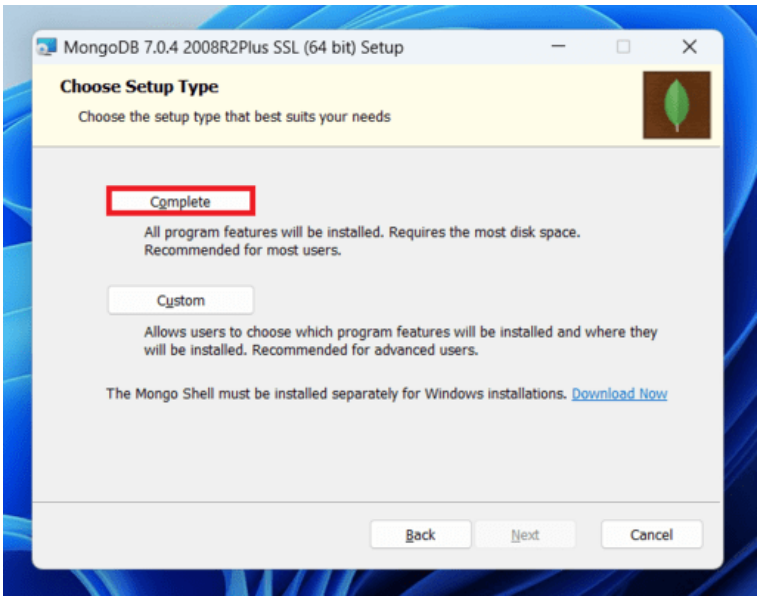


Step 2: When the download is complete open the msi file and click the *next button* in the startup screen

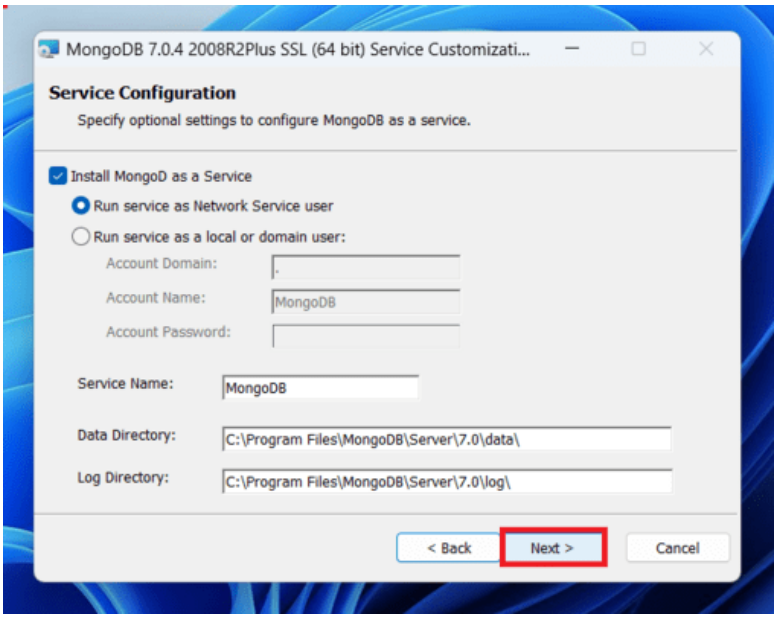
Step 3: Now accept the End-User License Agreement and click the next button:



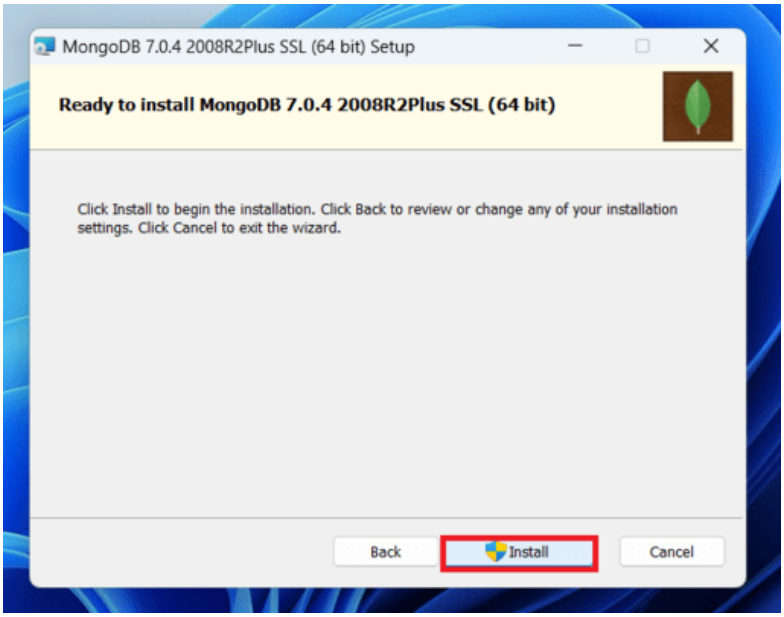
Step 4: Now select the *complete option* to install all the program features. Here, if you can want to install only selected program features and want to select the location of the installation, then use the *Custom option:*



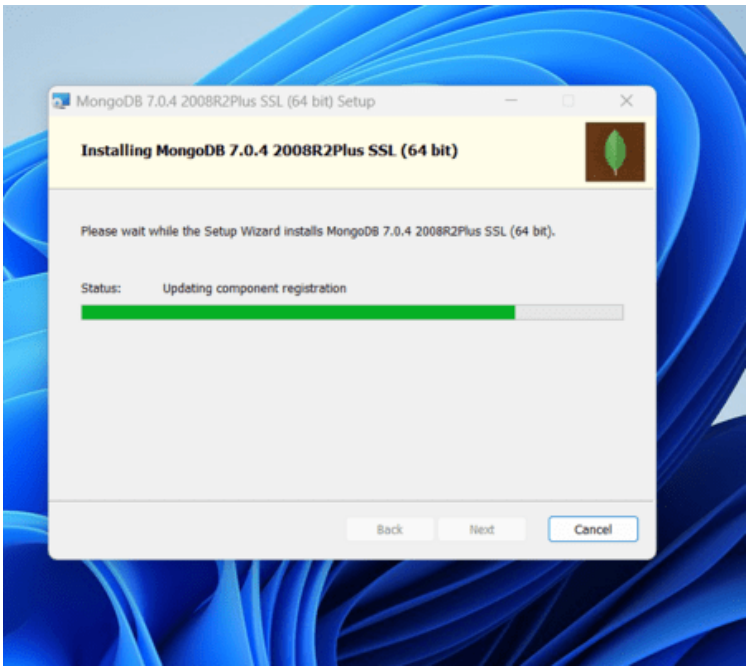
Step 5: Select “Run service as Network Service user” and copy the path of the data directory. Click Next:



Step 6: Click the *Install button* to start the MongoDB installation process:

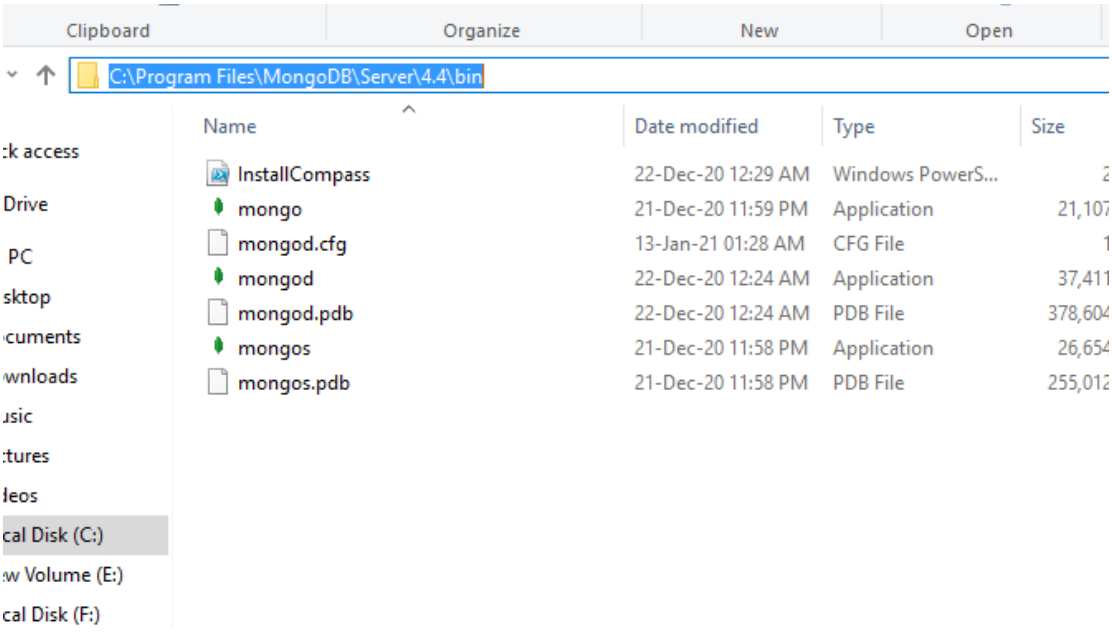


Step 7: After clicking on the install button installation of MongoDB begins:



Step 8: Now click the ***Finish button*** to complete the MongoDB installation process:

Step 9: Now we go to the location where MongoDB installed in step 5 in your system and copy the bin path:



MongoDB Commands

View all databases

```
show dbs
```

Create a new or switch databases

```
use dbName
```

View current Database

```
db
```

Delete Database

```
db.dropDatabase()
```

2. Collection Commands

Show Collections

```
show collections
```

Create a collection named 'comments'

```
db.createCollection('comments')
```

Drop a collection named 'comments'

```
db.comments.drop()
```

3. Row(Document) Commands

Show all Rows in a Collection

Insert One Row

```
db.comments.insert({
    'name': 'Harry',
```

```
    'lang': 'JavaScript',
    'member_since': 5
})
```

Insert many Rows

```
db.comments.insertMany([{
    'name': 'Harry',
    'lang': 'JavaScript',
    'member_since': 5
    },
    {'name': 'Rohan',
    'lang': 'Python',
    'member_since': 3
    },
    {'name': 'Lovish',
    'lang': 'Java',
    'member_since': 4
}])
```

Search in a MongoDb Database

```
db.comments.find({lang:'Python'})
```

Limit the number of rows in output

```
db.comments.find().limit(2)
```

Count the number of rows in the output

```
db.comments.find().count()
```

Update a row

```
db.comments.updateOne({name: 'Shubham'},
```

```
{$set: {'name': 'Harry',
    'lang': 'JavaScript',
    'member_since': 51
}}, {upsert: true})
```

Mongodb Increment Operator

```
db.comments.update({name: 'Rohan'},
{$inc:{
    member_since: 2
}})
```

Mongodb Rename Operator

```
db.comments.update({name: 'Rohan'},
{$rename:{
    member_since: 'member'
}})
```

Delete Row

```
db.comments.remove({name: 'Harry'})
```

Less than/Greater than/ Less than or Eq/Greater than or Eq

```
db.comments.find({member_since: {$lt: 90}})
db.comments.find({member_since: {$lte: 90}})
db.comments.find({member_since: {$gt: 90}})
db.comments.find({member_since: {$gte: 90}})
```