

NOSQL & MongoDB ASSIGNMENT

Name: PRIYESHWAR M D

Mail: priyesh2664@gmail.com

1. What is NoSQL?

NoSQL refers to a non-relational database management system that is schema-less, avoids complex joins, and supports easy horizontal scaling. It is particularly well-suited for handling large volumes of distributed, semi-structured, and unstructured data, making it ideal for modern, high-performance, and agile applications.

2. NoSQL vs SQL

Feature	SQL	NoSQL
Type	Relational	Document, Key-Value, Graph, etc.
Schema	Fixed	Dynamic
Scaling	Vertical	Horizontal
Transaction Support	Full	Limited (depends on DB)
Examples	MySQL, Oracle	MongoDB, Cassandra, Neo4j

3. What is MongoDB?

MongoDB is an open-source, document-oriented NoSQL database that stores data in BSON (Binary JSON) format.

It delivers high performance, availability, and scalability, which are essential for modern web and cloud applications.

Its key features include:

- High Performance: thanks to embedded documents and indexing
- Rich Query Language: supports aggregation, text search, and geospatial queries
- High Availability: uses replica sets for redundancy and failover
- Horizontal Scalability: via sharding across servers
- Multiple Storage Engines: allowing flexible data management

4. Comparison of RDMS & MongoDB

MongoDB uses terminology and structure that differs from relational databases:

- A Table becomes a Collection

- A Row becomes a Document
- A Column is a Field
- A Join is handled by Embedded Documents
- The Primary Key is replaced by an `_id` Field, which MongoDB creates automatically and ensures uniqueness

5. Schema Design

- Combine related data into a document.
- Duplicate data when read efficiency matters.
- Do joins on write, not read.
- Model according to application queries.

Installing MongoDB on Windows

1. Download MongoDB

- Go to the official MongoDB download center: <https://www.mongodb.com/try/download/community>
- Select:
 - Edition: Community Server
 - Version: Choose the latest stable version
 - Platform: Windows
 - Package: MSI (Windows Installer)

Click Download and wait for the .msi file to complete.

2. Install MongoDB Using MSI Installer

1. Run the downloaded .msi installer.
2. Choose "Complete" setup (recommended).
3. On the Service Configuration screen:
 - Install MongoDB as a Service: Checked (recommended)
 - Service Name: Leave as default
 - Choose Run service as Network Service user
4. Optionally, install MongoDB Compass, a GUI for MongoDB.
5. Click Install and wait for the installation to complete.

3. Set Up the MongoDB Environment

Create Data Folders:

MongoDB needs a folder to store data. By default, it uses:

- C:\data\db

4. Add MongoDB to the System Path

To use mongo and mongod commands from any directory:

1. Open **Control Panel** → **System and Security** → **System**.
2. Click on **Advanced system settings** → Environment Variables.
3. Under **System variables**, find the variable Path, and click **Edit**.
4. Add the path to MongoDB's bin directory.
Example:
5. C:\Program Files\MongoDB\Server\<version>\bin.

5. Start MongoDB Server

Open Command Prompt and run:

mongod

If MongoDB was installed as a service, it will start automatically after reboot. You can also manually start it via:

net start MongoDB

To stop:

net stop MongoDB

6. Start Mongo Shell

In another command prompt window, run:

mongo

This connects to the MongoDB server and opens the Mongo shell, where you can begin interacting with the database.

7. Verify Installation

Once mongo shell opens, try the following commands to verify:

show dbs

use test

db.test.insert({ name: "MongoDB", type: "NoSQL" })

db.test.find()

Hands On

1. Select Database

```
use mongodb
```

```
test> use mongodb
switched to db mongodb
```

2. Create a Collection

```
db.createCollection("user")
```

```
mongdb> db.createCollection("user")
{ ok: 1 }
```

3. Show All Collections

```
show collections
```

```
mongdb> show collections
user
```

4. Insert One Document into user Collection

```
db.user.insertOne({name:"Arun", age:"24", status:"pending"})
```

```
mongdb> db.user.insertOne({name:"Arun", age:"24", status:"pending"})
{
  acknowledged: true,
  insertedId: ObjectId('687e0630581509c6d1eec4a9')
}
```

5. Retrieve All Documents from user Collection

```
db.user.find()
```

```
mongdb> db.user.find()
[
  {
    _id: ObjectId('687e0630581509c6d1eec4a9'),
    name: 'Arun',
    age: '24',
    status: 'pending'
  }
]
```

6. Insert One Document into inventory Collection

```
db.inventory.insertOne({  
    item: "canvas",  
    qty: 100,  
    tags: ["cotton"],  
    size: { h: 28, w: 35.5, uom: "cm" }  
})  
  
mongdb> db.inventory.insertOne({ item: "canvas", qty: 100, tags: ["cotton"],  
size: { h: 28, w: 35.5, uom: "cm" } })  
{  
    acknowledged: true,  
    insertedId: ObjectId('687e06cd581509c6d1eec4aa')  
}
```

7. Insert Multiple Documents into inventory Collection

```
db.inventory.insertMany([  
    { item: "journal", qty: 25, tags: ["blank", "red"], size: { h: 14, w: 21, uom: "cm" } },  
    { item: "mat", qty: 85, tags: ["gray"], size: { h: 27.9, w: 35.5, uom: "cm" } },  
    { item: "mousepad", qty: 25, tags: ["gel", "blue"], size: { h: 19, w: 22.85, uom: "cm" } }  
])
```

```
mongdb> db.inventory.insertMany([{ item: "journal", qty: 25, tags: ["blank",  
"red"], size: { h: 14, w: 21, uom: "cm" } }, { item: "mat", qty: 85, tags: [  
"gray"], size: { h: 27.9, w: 35.5, uom: "cm" } }, { item: "mousepad", qty: 2  
5, tags: ["gel", "blue"], size: { h: 19, w: 22.85, uom: "cm" } }])  
{  
    acknowledged: true,  
    insertedIds: {  
        '0': ObjectId('687e06ed581509c6d1eec4ab'),  
        '1': ObjectId('687e06ed581509c6d1eec4ac'),  
        '2': ObjectId('687e06ed581509c6d1eec4ad')  
    }  
}
```

8. Retrieve All Documents from inventory Collection

```
db.inventory.find()
```

```
mongodb> db.inventory.find()
[  
  {  
    _id: ObjectId('687e06cd581509c6d1eec4aa'),  
    item: 'canvas',  
    qty: 100,  
    tags: [ 'cotton' ],  
    size: { h: 28, w: 35.5, uom: 'cm' }  
  },  
  {  
    _id: ObjectId('687e06ed581509c6d1eec4ab'),  
    item: 'journal',  
    qty: 25,  
    tags: [ 'blank', 'red' ],  
    size: { h: 14, w: 21, uom: 'cm' }  
  },  
  {  
    _id: ObjectId('687e06ed581509c6d1eec4ac'),  
    item: 'mat',  
    qty: 85,  
    tags: [ 'gray' ],  
    size: { h: 27.9, w: 35.5, uom: 'cm' }  
  },  
  {  
    _id: ObjectId('687e06ed581509c6d1eec4ad'),  
    item: 'mousepad',  
    qty: 25,  
    tags: [ 'gel', 'blue' ],  
    size: { h: 19, w: 22.85, uom: 'cm' }  
  }  
]
```

10. Project Specific Fields (Include item, qty)

```
db.inventory.find({}, { item: 1, qty: 1 })
```

```
mongodb> db.inventory.find({}, {item:1,qty:1})
[  
  {  
    _id: ObjectId('687e06cd581509c6d1eec4aa'),  
    item: 'canvas',  
    qty: 100  
  },  
  {  
    _id: ObjectId('687e06ed581509c6d1eec4ab'),  
    item: 'journal',  
    qty: 25  
  },  
  { _id: ObjectId('687e06ed581509c6d1eec4ac'), item: 'mat', qty: 85 },  
  {  
    _id: ObjectId('687e06ed581509c6d1eec4ad'),  
    item: 'mousepad',  
    qty: 25  
  }  
]
```

11. Exclude _id Field in Projection

```
db.inventory.find({}, { item: 1, qty: 1, _id: 0 })  
mongdb> db.inventory.find({}, {item:1,qty:1,_id:0})  
[  
  { item: 'canvas', qty: 100 },  
  { item: 'journal', qty: 25 },  
  { item: 'mat', qty: 85 },  
  { item: 'mousepad', qty: 25 }  
]
```

12. Conditional Queries

- Greater than (qty > 25):

```
db.inventory.find({ qty: { $gt: 25 } })
```

```
mongdb> db.inventory.find({qty:{$gt:25}})  
[  
  {  
    _id: ObjectId('687e06cd581509c6d1eec4aa'),  
    item: 'canvas',  
    qty: 100,  
    tags: [ 'cotton' ],  
    size: { h: 28, w: 35.5, uom: 'cm' }  
  },  
  {  
    _id: ObjectId('687e06ed581509c6d1eec4ac'),  
    item: 'mat',  
    qty: 85,  
    tags: [ 'gray' ],  
    size: { h: 27.9, w: 35.5, uom: 'cm' }  
}
```

- Less than (qty < 85):

```
db.inventory.find({ qty: { $lt: 85 } })
```

```
mongodb> db.inventory.find({qty:{$lt:85}})  
[  
  {  
    _id: ObjectId('687e06ed581509c6d1eec4ab'),  
    item: 'journal',  
    qty: 25,  
    tags: [ 'blank', 'red' ],  
    size: { h: 14, w: 21, uom: 'cm' }  
  },  
  {  
    _id: ObjectId('687e06ed581509c6d1eec4ad'),  
    item: 'mousepad',  
    qty: 25,  
    tags: [ 'gel', 'blue' ],  
    size: { h: 19, w: 22.85, uom: 'cm' }  
  }  
]
```

- Range (qty between 24 and 100):

```
db.inventory.find({ qty: { $lte: 100, $gte: 24 } })
```

```
mongdb> db.inventory.find({qty:{$lte:100,$gte:24}})  
[  
  {  
    _id: ObjectId('687e06cd581509c6d1eec4aa'),  
    item: 'canvas',  
    qty: 100,  
    tags: [ 'cotton' ],  
    size: { h: 28, w: 35.5, uom: 'cm' }  
  },  
  {  
    _id: ObjectId('687e06ed581509c6d1eec4ab'),  
    item: 'journal',  
    qty: 25,  
    tags: [ 'blank', 'red' ],  
    size: { h: 14, w: 21, uom: 'cm' }  
  },  
  {  
    _id: ObjectId('687e06ed581509c6d1eec4ac'),  
    item: 'mat',  
    qty: 85,  
    tags: [ 'gray' ],  
    size: { h: 27.9, w: 35.5, uom: 'cm' }  
  },  
  {  
    _id: ObjectId('687e06ed581509c6d1eec4ad'),  
    item: 'mousepad',  
    qty: 25,  
    tags: [ 'gel', 'blue' ],  
    size: { h: 19, w: 22.85, uom: 'cm' }  
  }  
]
```

13. Regex Queries on item Field

- Starts with 'ma':

```
db.inventory.find({ item: /^ma/ })
```

```
mongdb> db.inventory.find({item:/^ma/})  
[  
  {  
    _id: ObjectId('687e06ed581509c6d1eec4ac'),  
    item: 'mat',  
    qty: 85,  
    tags: [ 'gray' ],  
    size: { h: 27.9, w: 35.5, uom: 'cm' }  
  }  
]
```

- Contains 'ad':

```
db.inventory.find({ item: /ad/ })
```

- Starts with 'm':

```
db.inventory.find({ item: /^m/ })
```

```
mongdb> db.inventory.find({item:/ad/})
[
  {
    _id: ObjectId('687e06ed581509c6d1eec4ad'),
    item: 'mousepad',
    qty: 25,
    tags: [ 'gel', 'blue' ],
    size: { h: 19, w: 22.85, uom: 'cm' }
  }
]
mongdb> db.inventory.find({item:/^m/})
[
  {
    _id: ObjectId('687e06ed581509c6d1eec4ac'),
    item: 'mat',
    qty: 85,
    tags: [ 'gray' ],
    size: { h: 27.9, w: 35.5, uom: 'cm' }
  },
  {
    _id: ObjectId('687e06ed581509c6d1eec4ad'),
    item: 'mousepad',
    qty: 25,
    tags: [ 'gel', 'blue' ],
    size: { h: 19, w: 22.85, uom: 'cm' }
  }
]
```

14. OR Query

```
db.inventory.find({
  $or: [{ qty: 25 }, { item: /^m/ }]
})
```

```
mongodb> db.inventory.find({$or:[{qty:25},{item:/^m/}]})
[  
  {  
    _id: ObjectId('687e06ed581509c6d1eec4ab'),  
    item: 'journal',  
    qty: 25,  
    tags: [ 'blank', 'red' ],  
    size: { h: 14, w: 21, uom: 'cm' }  
  },  
  {  
    _id: ObjectId('687e06ed581509c6d1eec4ac'),  
    item: 'mat',  
    qty: 85,  
    tags: [ 'gray' ],  
    size: { h: 27.9, w: 35.5, uom: 'cm' }  
  },  
  {  
    _id: ObjectId('687e06ed581509c6d1eec4ad'),  
    item: 'mousepad',  
    qty: 25,  
    tags: [ 'gel', 'blue' ],  
    size: { h: 19, w: 22.85, uom: 'cm' }  
}
```

15. Sort Documents

- **Ascending by size:**

```
db.inventory.find({ qty: 25 }).sort({ size: 1 })
```

- **Descending by size:**

```
db.inventory.find({ qty: 25 }).sort({ size: -1 })
```

```

mongdb> db.inventory.find({qty:25}).sort({size:1})
[
  {
    _id: ObjectId('687e06ed581509c6d1eec4ab'),
    item: 'journal',
    qty: 25,
    tags: [ 'blank', 'red' ],
    size: { h: 14, w: 21, uom: 'cm' }
  },
  {
    _id: ObjectId('687e06ed581509c6d1eec4ad'),
    item: 'mousepad',
    qty: 25,
    tags: [ 'gel', 'blue' ],
    size: { h: 19, w: 22.85, uom: 'cm' }
  }
]
mongdb> db.inventory.find({qty:25}).sort({size:-1})
[
  {
    _id: ObjectId('687e06ed581509c6d1eec4ad'),
    item: 'mousepad',
    qty: 25,
    tags: [ 'gel', 'blue' ],
    size: { h: 19, w: 22.85, uom: 'cm' }
  },
  {
    _id: ObjectId('687e06ed581509c6d1eec4ab'),
    item: 'journal',
    qty: 25,
    tags: [ 'blank', 'red' ],
    size: { h: 14, w: 21, uom: 'cm' }
  }
]

```

16. Count Documents

- **Total:**

```
db.inventory.count()
```

- **Filtered:**

```
db.inventory.find({ qty: 25 }).count()
```

```

mongdb> db.inventory.count()
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or
estimatedDocumentCount.
4
mongdb> db.inventory.find().count()
4
mongdb> db.inventory.find({qty:25}).count()
2

```

17. Get Distinct Values

```
db.inventory.distinct("qty")
```

```
mongodb> db.inventory.distinct("qty")
[ 25, 85, 100 ]
```

18. Explain Query Plan

```
db.inventory.find({ qty: 25 }).explain()
```

```
mongodb> db.inventory.find({qty:25}).explain()
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'mongodb.inventory',
    parsedQuery: { qty: { '$eq': 25 } },
    indexFilterSet: false,
    queryHash: 'FD30B09C',
    planCacheShapeHash: 'FD30B09C',
    planCacheKey: 'A5384E34',
    optimizationTimeMillis: 0,
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    prunedSimilarIndexes: false,
    winningPlan: {
      isCached: false,
      stage: 'COLLSCAN',
      filter: { qty: { '$eq': 25 } },
      direction: 'forward'
    },
    rejectedPlans: []
  },
  queryShapeHash: '6D75FFFF7665A0CBE3A8D2769369213BFCA96611FC83430C962AE874907D96D6',
  command: { find: 'inventory', filter: { qty: 25 }, '$db': 'mongodb' },
  serverInfo: {
    host: 'Priyeshwar',
    port: 27017,
    version: '8.0.11',
    gitVersion: 'bed99f699da6cb2b74262aa6d473446c41476643'
  },
  serverParameters: {
    internalQueryFacetBufferSizeBytes: 104857600,
    internalQueryFacetMaxOutputDocSizeBytes: 104857600,
    internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
    internalDocumentSourceGroupMaxMemoryBytes: 104857600,
    internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
    internalQueryProhibitBlockingMergeOnMongoS: 0,
    internalQueryMaxAddToSetBytes: 104857600,
    internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,
    internalQueryFrameworkControl: 'trySbeRestricted',
    internalQueryPlannerIgnoreIndexWithCollationForRegex: 1
  }
}
```

19. Update a Field

```
db.inventory.updateOne(
```

```
  { item: "mat" },
```

```
{ $set: { "size.uom": "mm" } }

}

mongdb> db.inventory.updateOne({item:"mat"}, {$set:{size.uom:'mm'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

20. Delete All Documents from user Collection

```
db.user.deleteMany({})
```

```
mongdb> db.user.deleteMany({})
{ acknowledged: true, deletedCount: 1 }
mongdb> db.user.find()
```