

# **STAR TALENT**

## **Milestone: NoSQL Implementation (MongoDB)**

### **Group 14**

Student 1: Pooja Laxmi Sankarakameswaran

Student 2: Sabarish Subramaniam Anandhi Vijayaragavan

(857)-423-0754

(617)-992-5508

[sankarakameswaran.p@northeastern.edu](mailto:sankarakameswaran.p@northeastern.edu)

[anandhivijayaragav.s@northeastern.edu](mailto:anandhivijayaragav.s@northeastern.edu)

**Percentage of Effort Contributed by Student1: 50**

**Percentage of Effort Contributed by Student2: 50**

**Signature of Student 1:** 

**Signature of Student 2:** 

**Submission Date: 26th November 2023**

The below images show the Agent, Client, Job, Company, and Courses after we populate the required data in all the collections in MongoDB

## Agent

The screenshot shows the MongoDB Compass interface for the `Star_talent.Agent` collection. The left sidebar lists various collections, with `Agent` selected. The main panel displays the `Star_talent.Agent` collection with 40 documents and 1 index. The `Documents` tab is active, showing a list of documents. The first three documents are visible:

```
{ "_id": ObjectId("656665f2b1d68ef7888c52ad"), "AgentID": 1, "Name": "John Doe", "Address": "123 Main St, Cityville", "Phone": "555-1234", "Email": "john.doe@email.com", "Gender": "Male", "Age": 30 }
```

```
{ "_id": ObjectId("656665f2b1d68ef7888c52ae"), "AgentID": 2, "Name": "Jane Smith", "Address": "456 Oak St, Townsville", "Phone": "555-5678", "Email": "jane.smith@email.com", "Gender": "Female", "Age": 28 }
```

```
{ "_id": ObjectId("656665f2b1d68ef7888c52af"), "AgentID": 3, "Name": "Robert Johnson", "Address": "789 Pine St, Villagetown", "Phone": "555-9012", "Email": "robert.johnson@email.com", "Gender": "Male", "Age": 35 }
```

## Client

The screenshot shows the MongoDB Compass interface for the `Star_talent.Client` collection. The left sidebar lists various collections, with `Client` selected. The main panel displays the `Star_talent.Client` collection with 50 documents and 1 index. The `Documents` tab is active, showing a list of documents. The first three documents are visible:

```
{ "_id": ObjectId("65666612b1d68ef7888c5304"), "ClientID": 1, "Name": "Alice Johnson", "Age": 28, "Gender": "Female", "Email": "alice.johnson@email.com", "Phone": "555-1234", "Work_Exp": 5, "Curr_Comp_Name": "Infinity Films", "JobHistID": 1, "AgentID": 1 }
```

```
{ "_id": ObjectId("65666612b1d68ef7888c5305"), "ClientID": 2, "Name": "Bob Smith", "Age": 35, "Gender": "Male", "Email": "bob.smith@email.com", "Phone": "555-5678", "Work_Exp": 8, "Curr_Comp_Name": "Epic Productions", "JobHistID": 2, "AgentID": 2 }
```

```
{ "_id": ObjectId("65666612b1d68ef7888c5306"), "ClientID": 3 }
```

Job

localhost:27017

Documents  
Star\_talent.Job

My Queries

Databases

Search

Star\_talent

- Agent
- Assists
- Associates
- CareerSwitch
- Client
- ClientSkills
- Company
- Contract
- Courses
- DesiredSkills
- Feedback
- Fulltime
- Industry
- Job
- JobHistory
- PartTime
- Reviews
- Skills

Star\_talent.Job

201DOCUMENTSINDEXES

DocumentsAggregationsSchemaIndexesValidation

FilterType a query: { field: 'value' } or Generate query

ADD DATAEXPORT DATA

1 - 20 of 20

\_id: ObjectId('656664eb1d68ef7888c5459')  
JobID: 1  
JobDesc: "Film Editor"  
DSkillGroupID: 1  
CompanyID: 2

\_id: ObjectId('656664eb1d68ef7888c545a')  
JobID: 2  
JobDesc: "Screenwriter"  
DSkillGroupID: 2  
CompanyID: 2

\_id: ObjectId('656664eb1d68ef7888c545b')  
JobID: 3  
JobDesc: "Cinematographer"  
DSkillGroupID: 3  
CompanyID: 5

\_id: ObjectId('656664eb1d68ef7888c545c')  
JobID: 4  
JobDesc: "Sound Designer"  
DSkillGroupID: 4  
CompanyID: 11

Company

localhost:27017

Documents  
Star\_talent.Comp...

My Queries

Databases

Search

Associates

- CareerSwitch
- Client
- ClientSkills
- Company
- Contract
- Courses
- DesiredSkills
- Feedback
- Fulltime
- Industry
- Job
- JobHistory
- PartTime
- Reviews
- Skills
- Training

adminconfiglocal

Star\_talent.Company

301DOCUMENTSINDEXES

DocumentsAggregationsSchemaIndexesValidation

FilterType a query: { field: 'value' } or Generate query

ADD DATAEXPORT DATA

1 - 20 of 30

\_id: ObjectId('65666620b1d68ef7888c538b')  
CompanyID: 1  
Name: "Star Media"

\_id: ObjectId('65666620b1d68ef7888c538c')  
CompanyID: 2  
Name: "DreamWorks"

\_id: ObjectId('65666620b1d68ef7888c538d')  
CompanyID: 3  
Name: "Infinity Films"

\_id: ObjectId('65666620b1d68ef7888c538e')  
CompanyID: 4  
Name: "Galaxy Entertainment"

\_id: ObjectId('65666620b1d68ef7888c538f')  
CompanyID: 5  
Name: "Epic Productions"

\_id: ObjectId('65666620b1d68ef7888c5390')  
CompanyID: 6

Courses

localhost:27017

Documents  
Star\_talent.Cours...

My Queries

Databases

Search

ASSUMPTIONS

CareerSwitch

Client

ClientSkills

Company

Contract

Courses

DesiredSkills

Feedback

Fulltime

Industry

Job

JobHistory

PartTime

Reviews

Skills

Training

admin

config

local

Star\_talent.Courses

29  
DOCUMENTS

1  
INDEXES

Documents

Aggregations

Schema

Indexes

Validation

Filter

Type a query: { field: 'value' } or [Generate query](#)

Explain

Reset

Find

Options

ADD DATA

EXPORT DATA

1 - 20 of 29

Navigation icons

\_id: ObjectId('6566662db1d68ef7888c53b5')

CourseGroupID: 1

CourseName: "Film Editing Workshop"

CourseID: 1

\_id: ObjectId('6566662db1d68ef7888c53b6')

CourseGroupID: 1

CourseName: "Advanced Video Editing Techniques"

CourseID: 2

\_id: ObjectId('6566662db1d68ef7888c53b7')

CourseGroupID: 1

CourseName: "Digital Editing Masterclass"

CourseID: 3

\_id: ObjectId('6566662db1d68ef7888c53b8')

CourseGroupID: 2

CourseName: "Screenwriting Basics"

CourseID: 1

\_id: ObjectId('6566662db1d68ef7888c53b9')

CourseGroupID: 2

CourseName: "Script Development Strategies"

CourseID: 1

Reviews:

localhost:27017

Documents  
Star\_talent.Revie...

My Queries

Databases

Search

ASSUMPTIONS

CareerSwitch

Client

ClientSkills

Company

Contract

Courses

DesiredSkills

Feedback

Fulltime

Industry

Job

JobHistory

PartTime

Reviews

Skills

Training

admin

config

local

Star\_talent.Reviews

40  
DOCUMENTS

1  
INDEXES

Documents

Aggregations

Schema

Indexes

Validation

Filter

Type a query: { field: 'value' } or [Generate query](#)

Explain

Reset

Find

Options

ADD DATA

EXPORT DATA

1 - 20 of 40

Navigation icons

\_id: ObjectId('6566668b1d68ef7888c547f')

ReviewID: 1

Comment: "Exceptional skills and dedication."

StarRating: 4.8

Reviewer: "Company"

JobID: 1

\_id: ObjectId('6566668b1d68ef7888c5480')

ReviewID: 2

Comment: "Consistently meets project deadlines."

StarRating: 4.7

Reviewer: "Company"

JobID: 2

\_id: ObjectId('6566668b1d68ef7888c5481')

ReviewID: 3

Comment: "Adaptable and quick learner."

StarRating: 4.9

Reviewer: "Company"

JobID: 3

\_id: ObjectId('6566668b1d68ef7888c5482')

ReviewID: 4

Comment: "Demonstrates strong problem-solving abilities."

StarRating: 4.6

## NoSQL Queries:

1. Write a NoSQL query to select all the clients who are assisted by agents with agent ID "2" and "4"

Query:

```
[Star_talent> db.Agent. find( {AgentID : { $in: [2,4]}} )
```

Output:

```
[
  {
    _id: ObjectId('656665f2b1d68ef7888c52ae'),
    AgentID: 2,
    Name: 'Jane Smith',
    Address: '456 Oak St, Townsville',
    Phone: '555-5678',
    Email: 'jane.smith@email.com',
    Gender: 'Female',
    Age: 28
  },
  {
    _id: ObjectId('656665f2b1d68ef7888c52b0'),
    AgentID: 4,
    Name: 'Emily Davis',
    Address: '101 Cedar St, Hamletville',
    Phone: '555-3456',
    Email: 'emily.davis@email.com',
    Gender: 'Female',
    Age: 32
  }
]
```

2. Write a NoSQL query to select all clients whose contracts started before January 2022 (using aggregate function).

Query:

```
startalent> db.Client.aggregate([
...   {
...     $lookup: {
...       from: "Contract",
...       localField: "ClientID",
...       foreignField: "ClientID",
...       as: "contracts"
...     }
...   },
...   {
...     $unwind: "$contracts"
...   },
...   {
...     $match: {
...       "contracts.startDate": { $lt: ISODate("2022-01-01") }
...     }
...   },
...   {
...     $group: {
...       _id: "$ClientID",
...       ClientName: { $first: "$Name" }
...     }
...   }
... ])
```

Output:

```
[
  { _id: 33, ClientName: 'Giselle Fisher' },
  { _id: 28, ClientName: 'Benjamin Hayes' },
  { _id: 38, ClientName: 'Liam Murphy' },
  { _id: 3, ClientName: 'Charlie Davis' },
  { _id: 13, ClientName: 'Mia Turner' },
  { _id: 48, ClientName: 'Vincent Murphy' },
  { _id: 8, ClientName: 'Henry Anderson' },
  { _id: 18, ClientName: 'Robert Turner' }
]
```

3. Write a NoSQL query to select clients with a rating greater than 4.3 stars (using aggregate function).

```
startalent> db.Client.aggregate([
...   {
...     $lookup: {
...       from: "JobHistory",
...       localField: "JobHistID",
...       foreignField: "JobHistID",
...       as: "jobHistory"
...     }
...   },
...   {
...     $unwind: "$jobHistory"
...   },
...   {
...     $lookup: {
...       from: "Job",
...       localField: "jobHistory.JobID",
...       foreignField: "JobID",
...       as: "job"
...     }
...   },
...   {
...     $unwind: "$job"
...   },
...   {
...     $lookup: {
...       from: "Reviews",
...       localField: "job.JobID",
...       foreignField: "JobID",
...       as: "reviews"
...     }
...   },
...   {
...     $unwind: "$reviews"
...   },
...   {
...     $match: {
...       "reviews.StarRating": { $gt: 4.3 }
...     }
...   },
...   {
...     $group: {
...       _id: "$ClientID",
...       ClientName: { $first: "$Name" }
...     }
...   }
... ])
```



## Output:

```
[
  { _id: 28, ClientName: 'Benjamin Hayes' },
  { _id: 33, ClientName: 'Giselle Fisher' },
  { _id: 26, ClientName: 'Zane Powell' },
  { _id: 44, ClientName: 'Ryan Turner' },
  { _id: 9, ClientName: 'Ivy Brown' },
  { _id: 16, ClientName: 'Peter Hayes' },
  { _id: 45, ClientName: 'Sophie Powell' },
  { _id: 31, ClientName: 'Eva Turner' },
  { _id: 29, ClientName: 'Catherine Brown' },
  { _id: 36, ClientName: 'James Powell' },
  { _id: 6, ClientName: 'Frank Turner' },
  { _id: 22, ClientName: 'Victor Turner' },
  { _id: 43, ClientName: 'Quinn Hayes' },
  { _id: 15, ClientName: 'Olivia Ward' },
  { _id: 38, ClientName: 'Liam Murphy' },
  { _id: 40, ClientName: 'Noah Fisher' },
  { _id: 20, ClientName: 'Tom Murphy' },
  { _id: 19, ClientName: 'Samantha Brooks' },
  { _id: 7, ClientName: 'Grace Taylor' },
  { _id: 17, ClientName: 'Quinn Powell' }
]
```

Type "it" for more  
startalent> it

```
[
  { _id: 8, ClientName: 'Henry Anderson' },
  { _id: 42, ClientName: 'Preston Murphy' },
  { _id: 30, ClientName: 'Dylan Murphy' },
  { _id: 13, ClientName: 'Mia Turner' },
  { _id: 48, ClientName: 'Vincent Murphy' },
  { _id: 50, ClientName: 'Xavier Turner' },
  { _id: 49, ClientName: 'Willa Fisher' },
  { _id: 47, ClientName: 'Uma Turner' },
  { _id: 11, ClientName: 'Karen Green' },
  { _id: 25, ClientName: 'Yara Fisher' },
  { _id: 12, ClientName: 'Leo Reed' },
  { _id: 2, ClientName: 'Bob Smith' },
  { _id: 21, ClientName: 'Ursula Ross' },
  { _id: 35, ClientName: 'Isabel Hayes' },
  { _id: 39, ClientName: 'Megan Hayes' },
  { _id: 34, ClientName: 'Harrison Turner' },
  { _id: 4, ClientName: 'David Wilson' },
  { _id: 1, ClientName: 'Alice Johnson' },
  { _id: 24, ClientName: 'Xander Hayes' }
]
```



4. Write a NoSQL query to find the average age of clients per company (using MapReduce).

```
Star_talent> db.runCommand({
...   mapReduce: "Client",
...   map: function() {
...     emit(this.CompanyID, { age: this.Age, count: 1 });
...   },
...   reduce: function(key, values) {
...     var result = { age: 0, count: 0 };
...     values.forEach(function(value) {
...       result.age += value.age;
...       result.count += value.count;
...     });
...     return result;
...   },
...   finalize: function(key, reducedValue) {
...     reducedValue.average_age = reducedValue.age / reducedValue.count;
...     return { average_age: reducedValue.average_age };
...   },
...   out: { inline: 1 }
... });
```

Output:

```
{
  results: [ { _id: null, value: { average_age: 33.5 } } ],
  ok: 1
}
```

5. Write a NoSQL query to group all the clients by gender and count (using MapReduce).

```
Star_talent> db.runCommand({
...   mapReduce: "Client",
...   map: function() {
...     emit(this.Gender, 1);
...   },
...   reduce: function(key, values) {
...     return Array.sum(values);
...   },
...   out: { inline: 1 }
... });
```

Output

```
{
  results: [ { _id: 'Male', value: 26 }, { _id: 'Female', value: 24 } ],
  ok: 1
}
```