

June 16, 2025

## SESSION 1

```

1  const { MongoClient } = require('mongodb');
2
3  const uri = 'mongodb+srv://VaishnaviUser:Vaish3402005@cluster0.mdyja8t.mongodb.net/?retryWrites=true&majority=';
4
5  const client = new MongoClient(uri);
6
7  async function run() {
8    try {
9      await client.connect();
10     console.log('Connected to MongoDB Atlas');
11     const database = client.db('myDB');
12     const collection = database.collection('myCollection');
13     const result = await collection.insertOne({ name: 'test' });
14     console.log('Insert result:', result);
15   } finally {
16     await client.close();
17   }
18 }
19
20 run().catch(console.dir);
21

```

```

Connected to MongoDB Atlas
Insert result: {
  acknowledged: true,
  insertedId: new ObjectId('684fbc219518d697dc3a2328')
}
PS E:\GNCIP\mongodb-practice>

```

```

mongosh mongod> use test; useAtlasDatabase('cluster0-mongodb-atlas-8fcd7-shard-0')
Atlas atlas-8fcd7-shard-0 [primary] test> db.createCollection('posts')
{ ok: 1 }
Atlas atlas-8fcd7-shard-0 [primary] test> db.posts.insertOne({'title': 'Post 1'})
TypeError: db.posts.insertOne is not a function
Atlas atlas-8fcd7-shard-0 [primary] test> db.posts.insertOne({'title': 'Post 1'})
TypeError: db.posts.insertOne is not a function
Atlas atlas-8fcd7-shard-0 [primary] test> db.posts.insertOne({
...  title: "Post Title 1"
...  title: "Post Title 1"
... })
SyntaxError: Unexpected token, expected "," (3:0)

   1 | db.posts.insertOne({
   2 |   title: "Post Title 1"
>  3 |   title: "Post Title 1"
     |   ^
   4 | }
Atlas atlas-8fcd7-shard-0 [primary] test> db.posts.insertOne({ title: "Post Title 1",
... body: "Body of post.",
... category: "News",
... likes: 1,
... tags: ["news", "events"],
... date: Date()
... })
{
  acknowledged: true,
  insertedId: ObjectId('604fc20ea71292893e50eb67')
}
Atlas atlas-8fcd7-shard-0 [primary] test>

```



```
File Edit Selection View Go Run Terminal Help ← → mongodb-practice
```

EXPLORER

- OPEN EDITORS
  - JS example.js
- MONGODB-PRACTICE
  - node\_modules
  - example.js
  - package-lock.json
  - package.json

```
1 // ...
2
3 async function updateOrInsertPost() {
4   const options = { upsert: true };
5   const result = await collection.updateOne(filter, updateDoc, options);
6
7   if (result.upsertedCount > 0) {
8     console.log("🎉 Inserted new post with ID:", result.upsertedId._id);
9   } else if (result.modifiedCount > 0) {
10    console.log("🔧 Updated existing post.");
11  } else {
12    console.log("🔕 No changes made.");
13  }
14
15  } catch (err) {
16    console.error("❌ Error:", err);
17  } finally {
18    await client.close();
19    console.log("🔌 Connection closed");
20  }
21
22  updateOrInsertPost();
23
24 // ...
```

PROBLEMS OUTPUT DEBUG CONSOLE PORTS AZURE CODE REFERENCE LOG TERMINAL

```
Multiple documents inserted: 3
Connection closed
PS E:\GMCIP\mongodb-practice> node example.js
Connected to MongoDB
Single document inserted: new ObjectId('684fc964a5b881189a36f413')
Multiple documents inserted: 3
Connection closed
PS E:\GMCIP\mongodb-practice> node example.js
Connected to MongoDB
Inserted new post with ID: undefined
Connection closed
PS E:\GMCIP\mongodb-practice>
```

Ln 39, Col 18 Spaces: 4 UTF-8 CRLF JavaScript Signed out Go Live Quokka

## SESSION 2

```
File Edit Selection View Go Run Terminal Help ← → mongodb-practice
```

EXPLORER

- OPEN EDITORS
  - JS example.js
  - JS pipeline.js
- MONGODB-PRACTICE
  - node\_modules
  - example.js
  - package-lock.json
  - package.json
  - pipeline.js

```
1 const uri = "mongodb://localhost:27017"; // or your Atlas URI
2 const client = new MongoClient(uri);
3
4 async function main() {
5   try {
6     await client.connect();
7     console.log("🔌 Connected to MongoDB");
8
9     const db = client.db("myDatabase");
10    const collection = db.collection("posts");
11
12    // 📄 INSERT
13    const insertResult = await collection.insertOne({
14      title: "Post Title 1",
15      body: "This is the first post",
16      likes: 5,
17      tags: ["intro", "mongodb"],
18      date: new Date()
19    });
20    console.log("🎉 Inserted:", insertResult.insertedId);
21
22    // 🔍 FIND
23    const found = await collection.find({ title: "Post Title 1" }).toArray();
24    console.log("🔍 Found:", found);
25
26    // 📝 UPDATE
27    const updateResult = await collection.updateOne(
28      { title: "Post Title 1" },
29      { $set: { likes: 10 } }
30    );
31    console.log("🔧 Updated count:", updateResult.modifiedCount);
32
33    // ❌ DELETE
34    const deleteResult = await collection.deleteOne({ title: "Post Title 1" });
35    console.log("🗑 Deleted count:", deleteResult.deletedCount);
36  } catch (err) {
37    console.error("Error:", err);
38  } finally {
39    await client.close();
40  }
41
42 // ...
```

Ln 1, Col 1 (146 selected) Spaces: 2 UTF-8 CRLF JavaScript Signed out Go Live Quokka

The screenshot shows a VS Code editor with a file named `example.js` open. The code defines an asynchronous function `main()` that connects to a MongoDB instance, inserts a document, and then finds it. The terminal output shows the successful execution of the script, including the connection status, the inserted document, and the found document.

```
6 async function main() {
15   const insertResult = await collection.insertOne({
20     date: new Date()
21   });
22   console.log("Inserted:", insertResult.insertedId);
23
24   // FIND
25   const found = await collection.find({ title: "Post Title 1" }).toArray();

[monodemon] watching path(s): *.*
[monodemon] watching extensions: js,mjs,cjs,json
[monodemon] starting 'node example.js'
[monodemon] Connected to MongoDB
[monodemon] Inserted: new ObjectId('684ffec998b4870eed00db5')
[monodemon] Found: [
  {
    _id: new ObjectId('684ffec998b4870eed00db5'),
    title: 'Post Title 1',
    body: 'This is the first post',
    likes: 5,
    tags: [ 'intro', 'mongodb' ],
    date: 2025-06-16T11:23:50.773Z
  }
]
[monodemon] Updated count: 1
[monodemon] Deleted count: 1
[monodemon] Connection closed
[monodemon] Updated count: 1
[monodemon] Deleted count: 1
[monodemon] Connection closed
[monodemon] Deleted count: 1
[monodemon] Connection closed
[monodemon] clean exit - waiting for changes before restart
```

The screenshot shows the same VS Code editor with the `example.js` file. The code has been updated to include a `deleteMany` operation. The terminal output shows the successful execution of the script, including the connection status, the inserted document, the deleted count, and the connection closure.

```
6 async function main() {
47   // INSERT: updateResult.insertedId - sample: 684ffec998b4870eed00db5
38   // { title: "Post Title 1" },
39   // { $set: { likes: 10 } }
40   // });
41   // console.log("Updated count:", updateResult.modifiedCount);
42
43   // DELETE
44   const deleteResult = await collection.deleteMany({ category: "technology" });
45   console.log("Deleted count:", deleteResult.deletedCount);
46 }
47 catch (err) {
48   console.error("Error:", err);
49 }
50 finally {
51   await client.close();
52   console.log("Connection closed");
53 }
54
55 main();
56
```

```

    },
    {
      _id: ObjectId('684fc9075d2f8095d5f515e7'),
      title: 'Post Title 2',
      body: 'Body of post.',
      category: 'Event',
      likes: 2,
      tags: [ 'news', 'events' ],
      date: ISODate('2025-06-16T07:34:31.899Z')
    },
    {
      _id: ObjectId('684fc9075d2f8095d5f515e9'),
      title: 'Post Title 4',
      body: 'Body of post.',
      category: 'Event',
      likes: 4,
      tags: [ 'news', 'events' ],
      date: ISODate('2025-06-16T07:34:31.899Z')
    },
    {
      _id: ObjectId('684fc964a5b081180a36f413'),
      name: 'Vaishnavi',
      age: 19
    },
    {
      _id: ObjectId('684fc964a5b081180a36f414'),
      title: 'Post Title 2',
      body: 'Body of post.',
      likes: 2,
      tags: [ 'news', 'events' ],
      date: ISODate('2025-06-16T07:36:04.718Z')
    },
    {
      _id: ObjectId('684fc964a5b081180a36f41b'),
      title: 'Post Title 4',
      body: 'Body of post.',
      category: 'Event',
      likes: 4,
      tags: [ 'news', 'events' ],
      date: ISODate('2025-06-16T07:36:04.719Z')
    },
    {
      _id: ObjectId('684fc9cd16113df675d7481d'),
      title: 'Post Title 5',
      body: 'Body of post.',
      category: 'Event',
      date: ISODate('2025-06-16T07:50:41.669Z'),
      likes: 5,
      tags: [ 'news', 'events' ]
    },
  ],
}

```

The image shows a VS Code editor window with a file named `pipeline.js` open. The file contains a JavaScript script that connects to a MongoDB database, runs an aggregation pipeline, and logs the results. The pipeline is designed to find the category with the highest total likes.

```
1 const uri = 'mongodb://localhost:27026';
2 const client = new MongoClient(uri);
3
4 pipeline.js > runAggregation > pipeline > $group
5   { _id: 'Intro', totalLikes: 10 }
6   { _id: 'Event', totalLikes: 32 }
7   { _id: null, totalLikes: 5 }
8
9 async function runAggregation() {
10   const client = new MongoClient(uri);
11   try {
12     await client.connect();
13     const database = client.db("MyDatabase");
14     const posts = database.collection("posts");
15
16     const pipeline = [
17       {
18         $match: { likes: { $gt: 1 } }
19       },
20       {
21         $group: {
22           _id: "$category",
23           totalLikes: { $sum: "$likes" }
24         }
25       }
26     ];
27
28     const results = await posts.aggregate(pipeline).toArray();
29     console.log("Aggregation Results:", results);
30   } catch (err) {
31     console.error("Error connecting to MongoDB:", err);
32   }
33 }
```

The terminal output shows the command to run the aggregation and the resulting output:

```
PS E:\GITHUB\mongodb-practice> node pipeline.js
(node) clean exit - waiting for changes before restart
(node) 3.1.10
(node) to restart at any time, enter `rs`
(node) watching path(s): *.*
(node) watching extensions: js,mjs,cjs,json
(node) starting 'node pipeline.js'
Aggregation Results: [
  { _id: 'Intro', totalLikes: 10 },
  { _id: null, totalLikes: 5 },
  { _id: 'Event', totalLikes: 32 }
]
(node) clean exit - waiting for changes before restart
```

```
File Edit Selection View Go Run Terminal Help ← → mongodb-practice
EXPLORER
OPEN EDITORS
MONGODB-PRACTICE
node_modules
example.js
package-lock.json
package.json
pipeline.js
pipeline.js X
6 async function runAggregation() {
25 const sample = await posts.findOne();
26 console.log("Sample Document:", sample);
27
28 // Step 3: Run aggregation
29 const pipeline = [
30   {
31     $group: {
32       _id: "$customerId",
33       totalAmount: { $sum: "$amount" },
34       orderCount: { $sum: 1 }
35     }
36   }
37 ];
38
39 const results = await posts.aggregate(pipeline).toArray();
40 console.log("Aggregation Results:", results);
41 } catch (err) {
42   console.error("Error:", err);
43 } finally {
44   await client.close();
45 }
46
[Symbol(shapeMode)]: false,
[Symbol(kcapture)]: false
}
[nodemon] clean exit - waiting for changes before restart
[nodemon] restarting due to changes...
[nodemon] starting 'node pipeline.js'
Sample data inserted.
Sample Document: { _id: 1, customerId: 'C001', amount: 250 }
Aggregation Results: [
  { _id: 'C001', totalAmount: 350, orderCount: 2 },
  { _id: 'C003', totalAmount: 300, orderCount: 1 },
  { _id: 'C002', totalAmount: 400, orderCount: 1 }
]
[nodemon] clean exit - waiting for changes before restart
PROBLEMS OUTPUT DEBUG CONSOLE PORTS AZURE CODE REFERENCE LOG TERMINAL
node + v ... ^ x
Launchpad 0 0 0 Live Share RHDA analysis has failed Ln 49, Col 1 Spaces: 4 UTF-8 CRLF JavaScript Signed out Go Live Quokka
```

```
File Edit Selection View Go Run Terminal Help ← → mongodb-practice
EXPLORER
OPEN EDITORS
MONGODB-PRACTICE
node_modules
example.js
package-lock.json
package.json
pipeline.js
pipeline.js X
6 async function runAggregation() {
25 const sample = await posts.findOne();
26 console.log("Sample Document:", sample);
27
28 // Step 3: Run aggregation
29 const pipeline = [
30   // {
31   //   $group: {
32   //     _id: "$customerId",
33   //     totalAmount: { $sum: "$amount" },
34   //     orderCount: { $sum: 1 }
35   //   }
36   // },
37   { $limit: 3 }
38 ];
39
40 const results = await posts.aggregate(pipeline).toArray();
41 console.log("Aggregation Results:", results);
42 } catch (err) {
43   console.error("Error:", err);
44 } finally {
45   await client.close();
46 }
47
Sample data inserted.
Sample Document: { _id: 1, customerId: 'C001', amount: 250 }
Aggregation Results: [ { _id: 1, customerId: 'C001', amount: 250 } ]
[nodemon] clean exit - waiting for changes before restart
[nodemon] restarting due to changes...
[nodemon] starting 'node pipeline.js'
Sample data inserted.
Sample Document: { _id: 1, customerId: 'C001', amount: 250 }
Aggregation Results: [
  { _id: 1, customerId: 'C001', amount: 250 },
  { _id: 2, customerId: 'C002', amount: 400 },
  { _id: 3, customerId: 'C001', amount: 100 }
]
[nodemon] clean exit - waiting for changes before restart
PROBLEMS OUTPUT DEBUG CONSOLE PORTS AZURE CODE REFERENCE LOG TERMINAL
node + v ... ^ x
Launchpad 0 0 0 Live Share RHDA analysis has failed Ln 37, Col 16 Spaces: 4 UTF-8 CRLF JavaScript Signed out Go Live Quokka
```

```
File Edit Selection View Go Run Terminal Help ← → mongodb-practice

EXPLORER
  OPEN EDITORS
  MONGODB-PRACTICE
    node_modules
    example.js
    package-lock.json
    package.json
    pipeline.js

pipeline.js
  6 async function runAggregation() {
  29   const pipeline = [
  36     // {
  37     //   $limit: 3
  38     // },
  39     // {
  40     //   $project: {
  41     //     "customerId": 'C001'
  42     //     "cuisine": 1,
  43     //     "address": 1
  44     //   },
  45     // {
  46     //   $limit: 2
  47     // },
  48     {
  49       $sort: { "amount": -1 }
  50     }
  51   ];
  52
  53   const results = await posts.aggregate(pipeline).toArray();
  54   console.log("Aggregation Results:", results);
  55   // catch (err) {
  56   //   console.error("Error:", err);
  57   // }
  58   finally {
  59     await client.close();
  60   }
  61 }
  62
  63 runAggregation();
  64
  65 }

[monodemon] clean exit - waiting for changes before restart
[monodemon] restarting due to changes...
[monodemon] starting 'node pipeline.js'
Sample data inserted.
Sample Document: { _id: 1, customerId: 'C001', amount: 250 }
Aggregation Results: [
  { _id: 2, customerId: 'C002', amount: 400 },
  { _id: 4, customerId: 'C003', amount: 300 },
  { _id: 1, customerId: 'C001', amount: 250 },
  { _id: 3, customerId: 'C001', amount: 100 }
]
[monodemon] clean exit - waiting for changes before restart
```

```
File Edit Selection View Go Run Terminal Help ← → mongodb-practice

EXPLORER
  OPEN EDITORS
  MONGODB-PRACTICE
    node_modules
    example.js
    package-lock.json
    package.json
    pipeline.js

pipeline.js
  6 async function runAggregation() {
  29   const pipeline = [
  48     // {
  49     //   $sort: { "amount": 1 }
  50     // },
  51     { $match: { customerId: "C001" } }
  52   ];
  53
  54   const results = await posts.aggregate(pipeline).toArray();
  55   console.log("Aggregation Results:", results);
  56   // catch (err) {
  57   //   console.error("Error:", err);
  58   // }
  59   finally {
  60     await client.close();
  61   }
  62 }
  63
  64 runAggregation();
  65
  66 }

[monodemon] clean exit - waiting for changes before restart
[monodemon] restarting due to changes...
[monodemon] starting 'node pipeline.js'
Sample data inserted.
Aggregation Results: [ { _id: 2, customerId: 'C002', amount: 400 } ]
[monodemon] clean exit - waiting for changes before restart
[monodemon] restarting due to changes...
[monodemon] starting 'node pipeline.js'
Sample data inserted.
Aggregation Results: [
  { _id: 1, customerId: 'C001', amount: 250 },
  { _id: 3, customerId: 'C001', amount: 100 }
]
[monodemon] clean exit - waiting for changes before restart
```



```
File Edit Selection View Go Run Terminal Help ← → mongodb-practice
EXPLORER
OPEN EDITORS
MONGODB-PRACTICE
node_modules
package-lock.json
package.json
pipeline.js
example.js
pipeline.js X
pipeline.js
6 async function runAggregation() {
29   const pipeline = [
48     // {
49     //   $sort: { "amount": 1 }
50     // }
51     // { $match: { customerId: "C001" } }
52     {
53       $addFields: {
54         avgGrade: { $avg: "$grades.score" }
55       }
56     }
57   ];
58
59   const results = await posts.aggregate(pipeline).toArray();
60   console.log("Aggregation Results:", results);
61 } catch (err) {
62   console.error("Error:", err);
63 } finally {
64   await client.close();
65 }
66 }
67 }
68

PROBLEMS OUTPUT DEBUG CONSOLE PORTS AZURE CODE REFERENCE LOG TERMINAL
node + v ...
{ _id: 3, customerId: 'C001', amount: 100 },
{ _id: 4, customerId: 'C003', amount: 300 }
}
[nodemon] clean exit - waiting for changes before restart
[nodemon] restarting due to changes...
[nodemon] starting 'node pipeline.js'
Sample data inserted.
Aggregation Results: [
  { _id: 1, customerId: 'C001', amount: 250, avgGrade: null },
  { _id: 2, customerId: 'C002', amount: 400, avgGrade: null },
  { _id: 3, customerId: 'C001', amount: 100, avgGrade: null },
  { _id: 4, customerId: 'C003', amount: 300, avgGrade: null }
]
[nodemon] clean exit - waiting for changes before restart
```

```
File Edit Selection View Go Run Terminal Help ← → mongodb-practice
EXPLORER
OPEN EDITORS
MONGODB-PRACTICE
node_modules
package-lock.json
package.json
pipeline.js
example.js
pipeline.js X
pipeline.js
6 async function runAggregation() {
29   const pipeline = [
48     // {
49     //   $sort: { "amount": 1 }
50     // }
51     // { $match: { customerId: "C001" } }
52     // {
53     //   $addFields: {
54     //     avgGrade: { $avg: "$grades.score" }
55     //   }
56     // }
57     {
58       $count: "amount"
59     }
60   ];
61
62   const results = await posts.aggregate(pipeline).toArray();
63   console.log("Aggregation Results:", results);
64 } catch (err) {
65   console.error("Error:", err);
66 } finally {
67   await client.close();
68 }
69 }
70 }
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }

PROBLEMS OUTPUT DEBUG CONSOLE PORTS AZURE CODE REFERENCE LOG TERMINAL
node + v ...
{ _id: 3, customerId: 'C001', amount: 100 },
{ _id: 4, customerId: 'C003', amount: 300 }
}
[nodemon] clean exit - waiting for changes before restart
[nodemon] restarting due to changes...
[nodemon] starting 'node pipeline.js'
Sample data inserted.
Aggregation Results: [ { totalChinese: 4 } ]
[nodemon] clean exit - waiting for changes before restart
[nodemon] restarting due to changes...
[nodemon] starting 'node pipeline.js'
Sample data inserted.
Aggregation Results: [ { amount: 4 } ]
[nodemon] clean exit - waiting for changes before restart
```



```
File Edit Selection View Go Run Terminal Help ← → mongodb-practice
EXPLORER
  OPEN EDITORS
  MONGODB-PRACTICE
    node_modules
    package-lock.json
    package.json
    pipeline.js
  pipeline.js X
  pipeline.js
    6 async function runAggregation() {
    29   const pipeline = [
    52     //
    53     //   $addFields: {
    54     //     avgGrade: { $avg: "$grades.score" }
    55     //   }
    56     // }
    57     // {
    58     //   $count: "amount"
    59     // }
    60     {
    61       $lookup: {
    62         from: "movies",
    63         localField: "movie_id",
    64         foreignField: "_id",
    65         as: "movie_details",
    66       }
    67     }
    68   ];
    69 };
    70
    71 const results = await posts.aggregate(pipeline).toArray();
    72 console.log("Aggregation Results:", results);
    73
    74 at node:internal/main/run_main_module:36:49
    75
    76 Node.js v22.16.0
    77 [nodemon] app crashed - waiting for file changes before starting...
    78 [nodemon] restarting due to changes...
    79 [nodemon] starting "node pipeline.js"
    80 Sample data inserted.
    81 Aggregation Results: [
    82   { _id: 1, customerId: 'C001', amount: 250, movie_details: [] },
    83   { _id: 2, customerId: 'C002', amount: 400, movie_details: [] },
    84   { _id: 3, customerId: 'C001', amount: 100, movie_details: [] },
    85   { _id: 4, customerId: 'C001', amount: 300, movie_details: [] }
    86 ]
    87 [nodemon] clean exit - waiting for changes before restart
    88
    89 ...
    90
    91 > OUTLINE
    92 > TIMELINE
    93
    94 Launchpad 0 0 0 Live Share RHDA analysis has failed Ln 67, Col 2 Spaces: 4 UTF-8 CRLF JavaScript Signed out Go Live Quokka
```

```
File Edit Selection View Go Run Terminal Help ← → mongodb-practice
EXPLORER
  OPEN EDITORS
  MONGODB-PRACTICE
    node_modules
    package-lock.json
    package.json
    pipeline.js
  pipeline.js X
  pipeline.js
    6 async function runAggregation() {
    14
    15   // Step 1: Insert sample documents
    16   await posts.insertMany([
    17     { customerId: "C001", amount: 500 },
    18     { customerId: "C002", amount: 200 },
    19     { customerId: "C003", amount: 700 },
    20     { customerId: "C001", amount: 300 }
    21   ]);
    22   console.log("Sample data inserted.");
    23
    24   // Step 2: Aggregation pipeline to group totals and write to new collection
    25   const pipeline = [
    26     {
    27       $group: {
    28         _id: "$customerId",
    29         totalAmount: { $sum: "$amount" }
    30       }
    31     },
    32     {
    33       $out: "customer_totals" // Creates or overwrites this collection
    34     }
    35   ];
    36
    37   at process.processTicksAndRejections (node:internal/process/task_queues:105:5)
    38
    39 Node.js v22.16.0
    40 [nodemon] app crashed - waiting for file changes before starting...
    41 [nodemon] restarting due to changes...
    42 [nodemon] starting "node pipeline.js"
    43 Sample data inserted.
    44 Aggregation written to 'customer_totals' collection.
    45 customer_totals Results: [
    46   { _id: 'C002', totalAmount: 200 },
    47   { _id: 'C001', totalAmount: 800 },
    48   { _id: 'C003', totalAmount: 700 }
    49 ]
    50 [nodemon] clean exit - waiting for changes before restart
    51
    52 ...
    53
    54 > OUTLINE
    55 > TIMELINE
    56
    57 Launchpad 0 0 0 Live Share RHDA analysis has failed Ln 52, Col 1 Spaces: 4 UTF-8 CRLF JavaScript Signed out Go Live Quokka
```