

MONGO DB

Assignment-2

Name: P CHARAN RAGAVENDRA

Reg No: 21BCE1932

1. Calculate the average and total price of products that are either a "SmartTablet" or have a price less than \$700. Sort the results by average price in descending order.

```
eclipse-workspace - javamongo/src/main/java/javamongo/javamongoclient.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help

javamongoclient.java x
10 import java.text.ParseException;
11 import java.text.SimpleDateFormat;
12 import java.util.Arrays;
13 import java.util.List;
14
15 public class javamongoclient {
16
17     public static void main(String[] args) {
18
19         // Establish connection to MongoDB server
20         MongoClient mongoClient = new MongoClient("localhost", 27017);
21         System.out.println("Connected to MongoDB server successfully");
22
23         // Accessing the database
24         MongoDB database = mongoClient.getDatabase("mongodbjava");
25         System.out.println("Database 'mongodbjava' accessed successfully");
26
27         // Accessing the collection
28         MongoCollection<Document> collection = database.getCollection("products");
29
30         // Format for date parsing
31         SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
32
33         try {
34             // Task 1: Calculate the average and total price of products that are either a "SmartTablet" or have a price less than $700
35             List<Document> pipeline1 = Arrays.asList(
36                 new Document("$match", new Document("$or", Arrays.asList(
37                     new Document("name", "SmartTablet"),
38                     new Document("price", new Document("$lt", 700))
39                 ))),
40                 new Document("$group", new Document("_id", "$name")
41                     .append("totalPrice", new Document("$sum", "$price"))
42                     .append("averagePrice", new Document("$avg", "$price"))),
43                 new Document("$sort", new Document("averagePrice", -1))
44             );
45
46             AggregateIterable<Document> result1 = collection.aggregate(pipeline1);
47             System.out.println("Task 1 Results:");
48             for (Document doc : result1) {
49                 System.out.println(doc.toJson());
50             }
51
52         } catch (Exception e) {
53             e.printStackTrace();
54         }
55     }
56 }
```

```
Connected to MongoDB server successfully
Database 'mongodbjava' accessed successfully
Task 1 Results:
Jun 28, 2024 5:20:20 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:22}] to localhost:27017
Jun 28, 2024 5:20:20 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
Jun 28, 2024 5:20:20 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[7, 0, 11]}, m
Jun 28, 2024 5:20:20 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:23}] to localhost:27017
{"_id": "SmartTablet", "totalPrice": 899, "averagePrice": 899.0}
{"_id": "SmartPad", "totalPrice": 699, "averagePrice": 699.0}
{"_id": "SmartPhone", "totalPrice": 599, "averagePrice": 599.0}
```

2. Find the top 3 products released after January 1, 2015, with more than 8GB of RAM. Show only the name and price fields.

```
52 // Task 2: Find the top 3 products released after January 1, 2015, with more than 8GB of RAM
53 List<Document> pipeline2 = Arrays.asList(
54     new Document("$match", new Document("releaseDate", new Document("$gt", dateFormat.parse("2015-01-01"))
55         .append("spec.ram", new Document("$gt", 8))),
56     new Document("$sort", new Document("releaseDate", -1)),
57     new Document("$limit", 3),
58     new Document("$project", new Document("name", 1).append("price", 1).append("_id", 0))
59 );
60
61 AggregateIterable<Document> result2 = collection.aggregate(pipeline2);
62 System.out.println("Task 2 Results:");
63 for (Document doc : result2) {
64     System.out.println(doc.toJson());
65 }
66
```

(no output shown)

3. Calculate the average screen size of all products that have storage options of either 128GB or 256GB.

```
66 }
67
68 // Task 3: Calculate the average screen size of all products that have storage options of either 128GB or 256GB
69 List<Document> pipeline3 = Arrays.asList(
70     new Document("$match", new Document("$in", Arrays.asList(128, 256))),
71     new Document("$group", new Document("_id", null).append("averageScreenSize", new Document("$avg", "$spec.screen")))
72 );
73
74 AggregateIterable<Document> result3 = collection.aggregate(pipeline3);
75 System.out.println("Task 3 Results:");
76 for (Document doc : result3) {
77     System.out.println(doc.toJson());
78 }
79
80 } catch (ParseException e) {
81     e.printStackTrace();
82 }
83
84 // Close the connection
85 mongoClient.close();
86 }
87
```

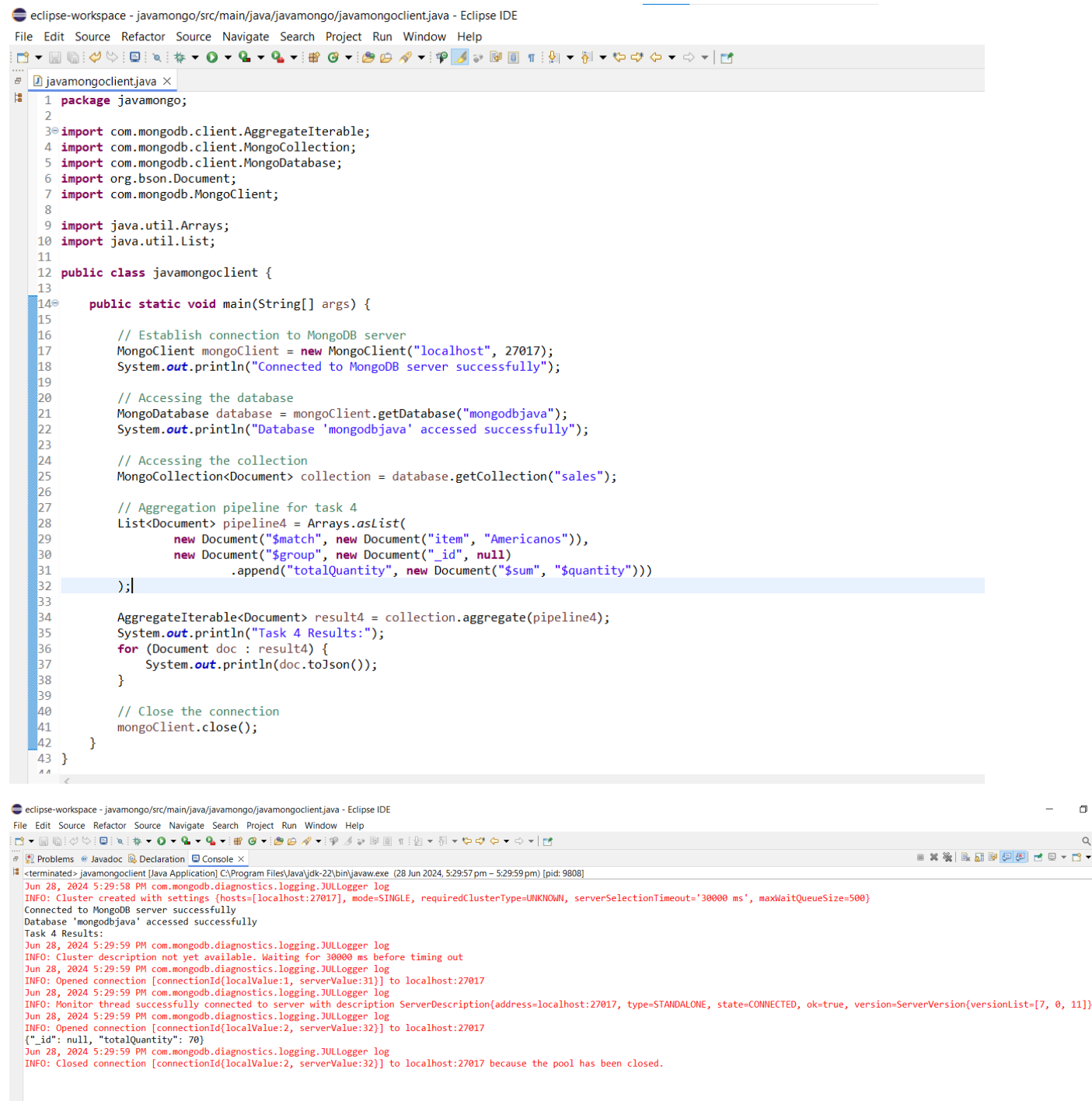
Task 3 Results:

```
{"_id": null, "averageScreenSize": 9.02}
```

Jun 28, 2024 5:20:20 PM com.mongodb.diagnostics.logging.JULLogger log

INFO: Closed connection [connectionId{localValue:2, serverValue:23}] to localhost:27017 because the pool has been closed.

4.How many “Americanos” were sold in total?



The screenshot displays the Eclipse IDE environment with a Java project named 'javamongo'. The main editor shows the file 'javamongoclient.java' with the following code:

```
1 package javamongo;
2
3 import com.mongodb.client.AggregateIterable;
4 import com.mongodb.client.MongoCollection;
5 import com.mongodb.client.MongoDatabase;
6 import org.bson.Document;
7 import com.mongodb.MongoClient;
8
9 import java.util.Arrays;
10 import java.util.List;
11
12 public class javamongoclient {
13
14     public static void main(String[] args) {
15
16         // Establish connection to MongoDB server
17         MongoClient mongoClient = new MongoClient("localhost", 27017);
18         System.out.println("Connected to MongoDB server successfully");
19
20         // Accessing the database
21         MongoDatabase database = mongoClient.getDatabase("mongodbjava");
22         System.out.println("Database 'mongodbjava' accessed successfully");
23
24         // Accessing the collection
25         MongoCollection<Document> collection = database.getCollection("sales");
26
27         // Aggregation pipeline for task 4
28         List<Document> pipeline4 = Arrays.asList(
29             new Document("$match", new Document("item", "Americanos")),
30             new Document("$group", new Document("_id", null)
31                 .append("totalQuantity", new Document("$sum", "$quantity")))
32         );
33
34         AggregateIterable<Document> result4 = collection.aggregate(pipeline4);
35         System.out.println("Task 4 Results:");
36         for (Document doc : result4) {
37             System.out.println(doc.toJson());
38         }
39
40         // Close the connection
41         mongoClient.close();
42     }
43 }
```

The bottom of the screenshot shows the 'Console' view with the following output:

```
<terminated> javamongoclient [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (28 Jun 2024, 5:29:57 pm - 5:29:59 pm) [pid: 9808]
Jun 28, 2024 5:29:58 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500}
Connected to MongoDB server successfully
Database 'mongodbjava' accessed successfully
Task 4 Results:
Jun 28, 2024 5:29:59 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
Jun 28, 2024 5:29:59 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:31}] to localhost:27017
Jun 28, 2024 5:29:59 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[7, 0, 11]}
Jun 28, 2024 5:29:59 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:32}] to localhost:27017
{"_id": null, "totalQuantity": 70}
Jun 28, 2024 5:29:59 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Closed connection [connectionId{localValue:2, serverValue:32}] to localhost:27017 because the pool has been closed.
```

5. Increase the price of “Lattes” by 5 units

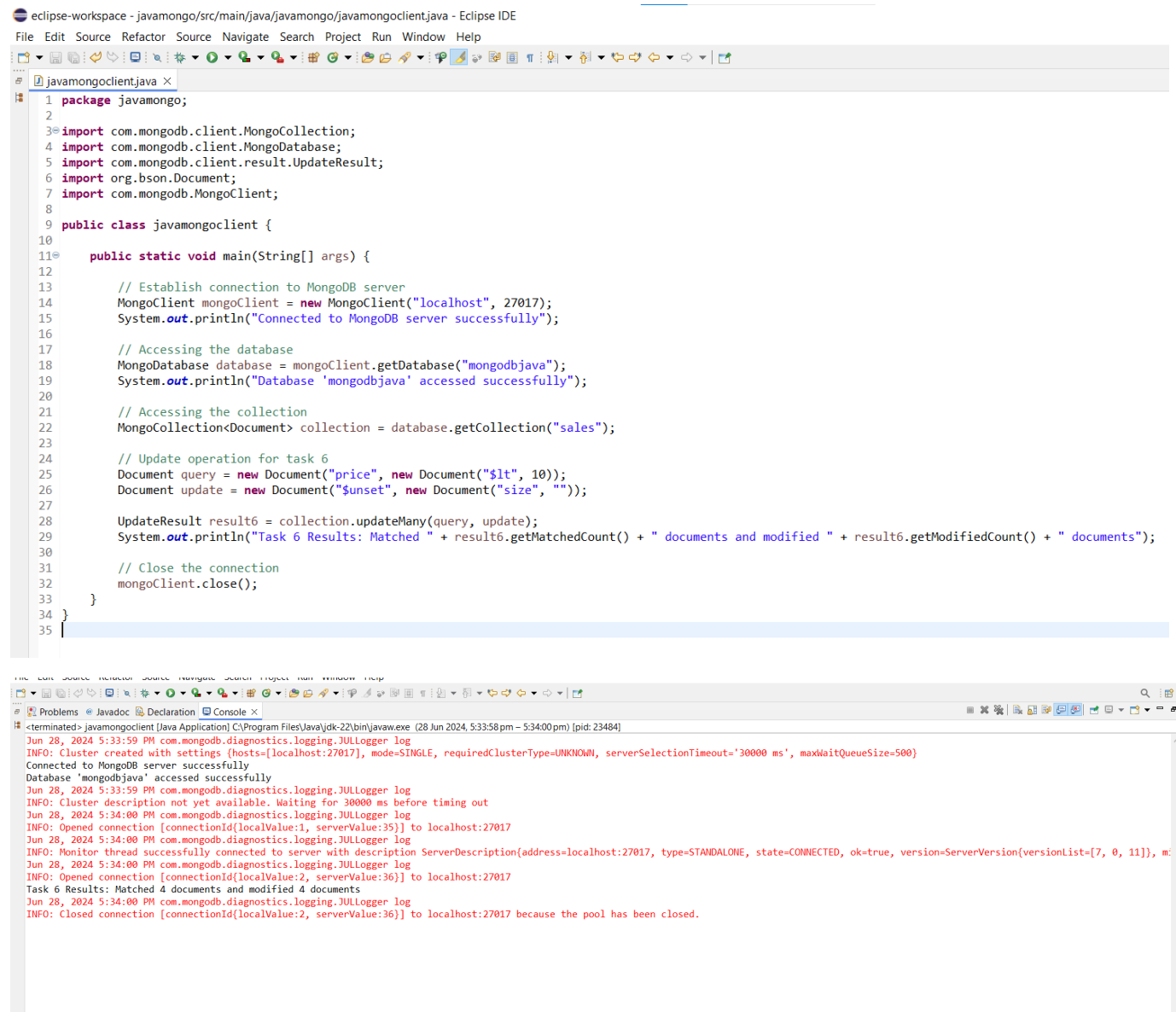
```
javamongoclient.java x
1 package javamongo;
2
3 import com.mongodb.client.MongoCollection;
4 import com.mongodb.client.MongoDatabase;
5 import com.mongodb.client.result.UpdateResult;
6 import org.bson.Document;
7 import com.mongodb.MongoClient;
8
9 public class javamongoclient {
10
11     public static void main(String[] args) {
12
13         // Establish connection to MongoDB server
14         MongoClient mongoClient = new MongoClient("localhost", 27017);
15         System.out.println("Connected to MongoDB server successfully");
16
17         // Accessing the database
18         MongoDatabase database = mongoClient.getDatabase("mongodbjava");
19         System.out.println("Database 'mongodbjava' accessed successfully");
20
21         // Accessing the collection
22         MongoCollection<Document> collection = database.getCollection("sales");
23
24         // Update operation for task 5
25         Document query = new Document("item", "Lattes");
26         Document update = new Document("$inc", new Document("price", 5));
27
28         UpdateResult result5 = collection.updateMany(query, update);
29         System.out.println("Task 5 Results: Matched " + result5.getMatchedCount() + " documents and modified " + result5.getModifiedCount() + " documents");
30
31         // Close the connection
32         mongoClient.close();
33     }
34 }
35
```

Problems Javadoc Declaration Console x

<terminated> javamongoclient [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (28 Jun 2024, 5:32:44 pm – 5:32:45 pm) [pid: 22288]

```
Jun 28, 2024 5:32:45 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings (hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500)
Connected to MongoDB server successfully
Database 'mongodbjava' accessed successfully
Jun 28, 2024 5:32:45 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
Jun 28, 2024 5:32:45 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:33}] to localhost:27017
Jun 28, 2024 5:32:45 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[7, 0, 11]}, m:
Jun 28, 2024 5:32:45 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:34}] to localhost:27017
Task 5 Results: Matched 2 documents and modified 2 documents
Jun 28, 2024 5:32:45 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Closed connection [connectionId{localValue:2, serverValue:34}] to localhost:27017 because the pool has been closed.
```

6. Unset the “size” field for all records where the price is less than 10.



```
eclipse-workspace - javamongo/src/main/java/javamongo/javamongoclient.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help

javamongoclient.java
1 package javamongo;
2
3 import com.mongodb.client.MongoCollection;
4 import com.mongodb.client.MongoDatabase;
5 import com.mongodb.client.result.UpdateResult;
6 import org.bson.Document;
7 import com.mongodb.MongoClient;
8
9 public class javamongoclient {
10
11     public static void main(String[] args) {
12
13         // Establish connection to MongoDB server
14         MongoClient mongoClient = new MongoClient("localhost", 27017);
15         System.out.println("Connected to MongoDB server successfully");
16
17         // Accessing the database
18         MongoDatabase database = mongoClient.getDatabase("mongodbjava");
19         System.out.println("Database 'mongodbjava' accessed successfully");
20
21         // Accessing the collection
22         MongoCollection<Document> collection = database.getCollection("sales");
23
24         // Update operation for task 6
25         Document query = new Document("price", new Document("$lt", 10));
26         Document update = new Document("$unset", new Document("size", ""));
27
28         UpdateResult result6 = collection.updateMany(query, update);
29         System.out.println("Task 6 Results: Matched " + result6.getMatchedCount() + " documents and modified " + result6.getModifiedCount() + " documents");
30
31         // Close the connection
32         mongoClient.close();
33     }
34 }
35 }
```

```
Problems Javadoc Declaration Console
<terminated> javamongoclient [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (28 Jun 2024, 5:33:58 pm - 5:34:00 pm) [pid: 23484]
Jun 28, 2024 5:33:59 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500}
Connected to MongoDB server successfully
Database 'mongodbjava' accessed successfully
Jun 28, 2024 5:33:59 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
Jun 28, 2024 5:34:00 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:35}] to localhost:27017
Jun 28, 2024 5:34:00 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[7, 0, 11]}, m
Jun 28, 2024 5:34:00 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:36}] to localhost:27017
Task 6 Results: Matched 4 documents and modified 4 documents
Jun 28, 2024 5:34:00 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Closed connection [connectionId{localValue:2, serverValue:36}] to localhost:27017 because the pool has been closed.
```

7.the total revenue generated from all sales

eclipse-workspace - javamongo/src/main/java/javamongo/javamongoclient.java - Eclipse IDE

File Edit Source Refactor Search Navigate Search Project Run Window Help

```
1 package javamongo;
2
3 import com.mongodb.client.AggregateIterable;
4 import com.mongodb.client.MongoCollection;
5 import com.mongodb.client.MongoDatabase;
6 import org.bson.Document;
7 import com.mongodb.MongoClient;
8
9 import java.util.Arrays;
10 import java.util.List;
11
12 public class javamongoclient {
13
14     public static void main(String[] args) {
15
16         // Establish connection to MongoDB server
17         MongoClient mongoClient = new MongoClient("localhost", 27017);
18         System.out.println("Connected to MongoDB server successfully");
19
20         // Accessing the database
21         MongoDatabase database = mongoClient.getDatabase("mongodbjava");
22         System.out.println("Database 'mongodbjava' accessed successfully");
23
24         // Accessing the collection
25         MongoCollection<Document> collection = database.getCollection("sales");
26
27         // Aggregation pipeline for task 7
28         List<Document> pipeline7 = Arrays.asList(
29             new Document("$group", new Document("_id", null)
30                 .append("totalRevenue", new Document("$sum", new Document("$multiply", Arrays.asList("$price", "$quantity")))))
31         );
32
33         AggregateIterable<Document> result7 = collection.aggregate(pipeline7);
34         System.out.println("Task 7 Results:");
35         for (Document doc : result7) {
36             System.out.println(doc.toJson());
37         }
38
39         // Close the connection
40         mongoClient.close();
41     }
42 }
43
```

<terminated> javamongoclient [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (28 Jun 2024, 5:35:10 pm – 5:35:11 pm) [pid: 11904]

```
Jun 28, 2024 5:35:11 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000
Connected to MongoDB server successfully
Database 'mongodbjava' accessed successfully
Task 7 Results:
Jun 28, 2024 5:35:11 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
Jun 28, 2024 5:35:11 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:37}] to localhost:27017
Jun 28, 2024 5:35:11 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, st
Jun 28, 2024 5:35:11 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:38}] to localhost:27017
{"_id": null, "totalRevenue": 2617}
Jun 28, 2024 5:35:11 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Closed connection [connectionId{localValue:2, serverValue:38}] to localhost:27017 because the pool has been closed.
```