**BIG DATA ANALYTICS**

**PRACTICAL – IX**

**NAME: MAHEK BUKELIYA ROLL NO: 008**

**DATE: 02/01/2025**

**TOPIC: AGGREGATION PIPELINE IN MONDODB USING JAVA**

**Q1] Count the number of documents in the collection books using aggregation query.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.AggregateIterable;

import org.bson.Document;

import java.util.Arrays;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

AggregateIterable<Document> result = collection.aggregate(Arrays.asList(

new Document("$group", new Document("\_id", null).append("totalCount", new Document("$sum", 1)))

));

for (var document : result) {

System.out.println("Total documents: " + document.getInteger("totalCount"));

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

**OUTPUT:**



**Q2]** **Calculate the total price of all books.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.AggregateIterable;

import org.bson.Document;

import java.util.Arrays;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

AggregateIterable<Document> result = collection.aggregate(Arrays.asList(

new Document("$group", new Document("\_id", null).append("totalPrice", new Document("$sum", "$price\_in\_inr")))

));

for (var document : result) {

System.out.println("Total price of all books: " + document.getDouble("totalPrice"));

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

**OUTPUT:**



**Q3] Calculate the average price of all books.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.AggregateIterable;

import org.bson.Document;

import java.util.Arrays;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

AggregateIterable<Document> result = collection.aggregate(Arrays.asList(

new Document("$group", new Document("\_id", null).append("AveragePrice", new Document("$avg", "$price\_in\_inr")))

));

for (var document : result) {

System.out.println("Average price of all books: " + document.getDouble("AveragePrice"));

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

**OUTPUT:**



**Q4] Count the books with minimum pages in collection books.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.AggregateIterable;

import org.bson.Document;

import java.util.Arrays;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

AggregateIterable<Document> result = collection.aggregate(Arrays.asList(

new Document("$group", new Document("\_id", null)

.append("minPages", new Document("$min", "$no\_of\_pages")))

));

for (var document : result) {

System.out.println("Minimum number of pages: " + document.getInteger("minPages"));

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

**OUTPUT:**

****

**Q5] Count the books with maximum pages in collection books.**

import com.mongodb.client.MongoClients;

import com.mongodb.client.AggregateIterable;

import org.bson.Document;

import java.util.Arrays;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

AggregateIterable<Document> result = collection.aggregate(Arrays.asList(

new Document("$group", new Document("\_id", null)

.append("maxPages", new Document("$max", "$no\_of\_pages")))

));

for (var document : result) {

System.out.println("Maximum number of pages: " + document.getInteger("maxPages"));

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

**OUTPUT:**

****

**Q6] Display the documents of books with price greater than Rs 1000.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.FindIterable;

import org.bson.Document;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

FindIterable<Document> result = collection.find(new Document("price\_in\_inr", new Document("$gt", 1000)));

for (var document : result) {

System.out.println(document.toJson());

}

} catch (Exception e) {

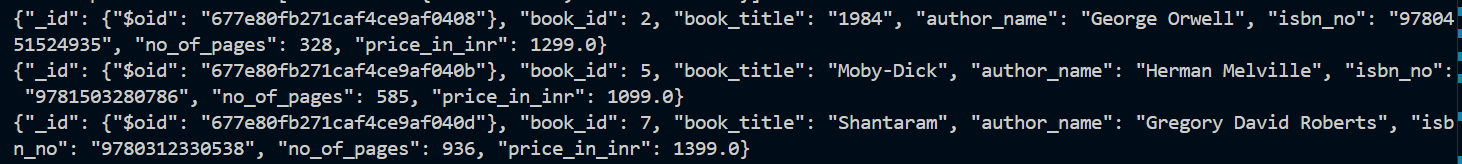
e.printStackTrace();

}

}

}

**OUTPUT:**

****

**Q7] Display the documents of books with price greater than Rs 600.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.FindIterable;

import org.bson.Document;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

FindIterable<Document> result = collection.find(new Document("price\_in\_inr", new Document("$lt", 600)));

for (var document : result) {

System.out.println(document.toJson());

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

**OUTPUT:**

****

**Q8] Sort the books in ascending order of their name.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.FindIterable;

import com.mongodb.client.MongoCollection;

import org.bson.Document;

import com.mongodb.client.model.Sorts;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

FindIterable<Document> result = collection.find()

.sort(Sorts.ascending("book\_title"));

for (var document : result) {

System.out.println(document.toJson());

}

} catch (Exception e) {

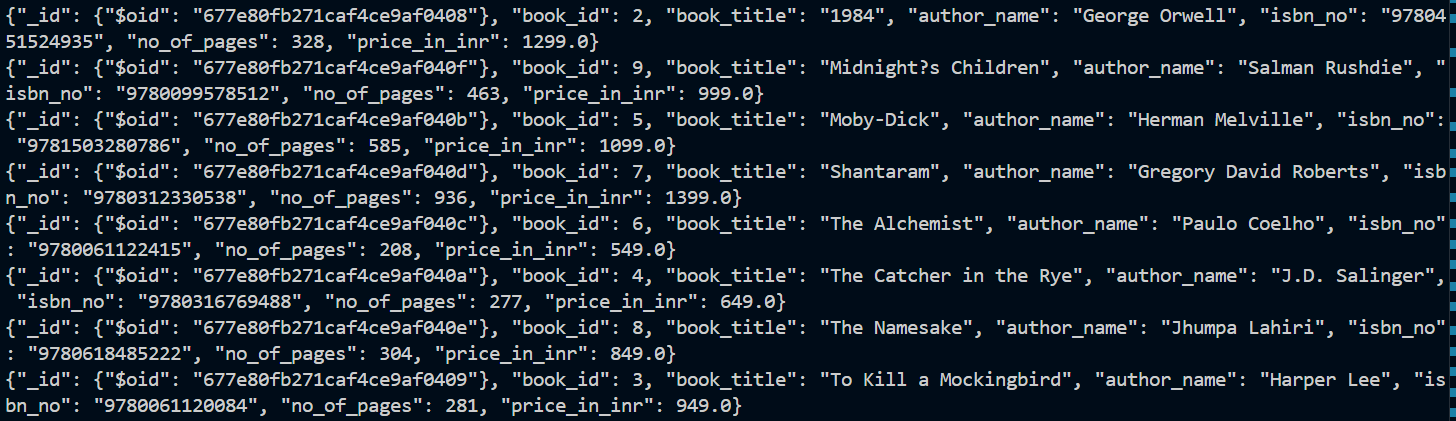
e.printStackTrace();

}

}

}

**OUTPUT:**

****

**Q9] Project only book\_title, author name and price of the books.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.FindIterable;

import com.mongodb.client.MongoCollection;

import org.bson.Document;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

Document projection = new Document("book\_title", 1)

.append("author\_name", 1)

.append("price", 1);

FindIterable<Document> result = collection.find()

.projection(projection)

.sort(new Document("book\_title", 1));

for (var document : result) {

System.out.println(document.toJson());

}

} catch (Exception e) {

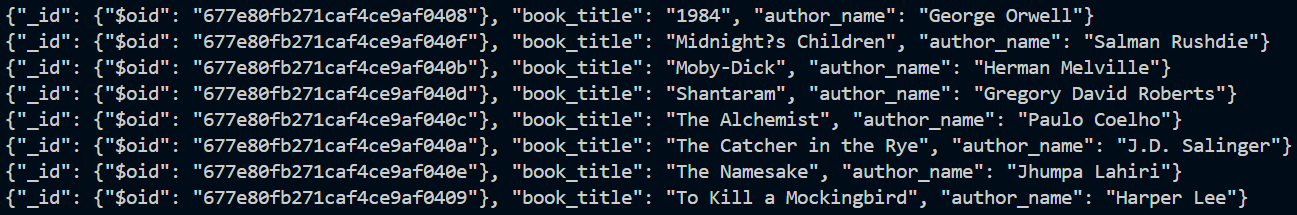
e.printStackTrace();

}

}

}

**OUTPUT:**

****

**Q10] Retrieve only the documents of the top 5 books by price.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.FindIterable;

import com.mongodb.client.MongoCollection;

import org.bson.Document;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

Document projection = new Document("book\_title", 1)

.append("author\_name", 1)

.append("price\_in\_inr", 1);

FindIterable<Document> result = collection.find()

.projection(projection)

.sort(new Document("price\_in\_inr", -1))

.limit(5);

for (var document : result) {

System.out.println(document.toJson());

}

} catch (Exception e) {

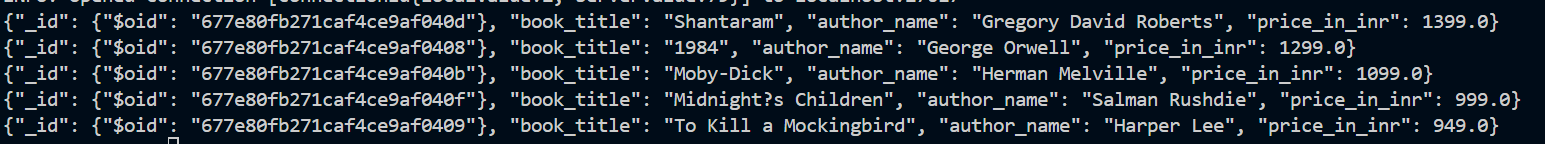
e.printStackTrace();

}

}

}

**OUTPUT:**

****

**Q11] Display the books starting from letter ‘T’ and project their book\_title and price.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.FindIterable;

import com.mongodb.client.MongoCollection;

import org.bson.Document;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

Document projection = new Document("book\_title", 1)

.append("price\_in\_inr", 1);

Document filter = new Document("book\_title", new Document("$regex", "^T"));

FindIterable<Document> result = collection.find(filter)

.projection(projection);

for (var document : result) {

System.out.println(document.toJson());

}

} catch (Exception e) {

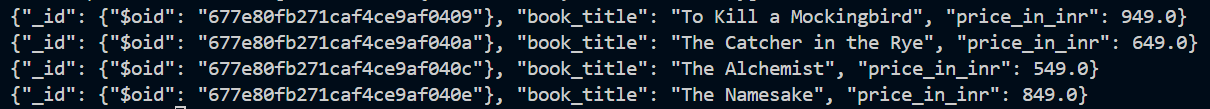
e.printStackTrace();

}

}

}

**OUTPUT:**

****

**Q12] Project only book\_title and no of pages, then sort by no. of pages in ascending order.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.FindIterable;

import com.mongodb.client.MongoCollection;

import org.bson.Document;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

Document projection = new Document("book\_title", 1)

.append("no\_of\_pages", 1);

FindIterable<Document> result = collection.find()

.projection(projection)

.sort(new Document("no\_of\_pages", 1));

for (var document : result) {

System.out.println(document.toJson());

}

} catch (Exception e) {

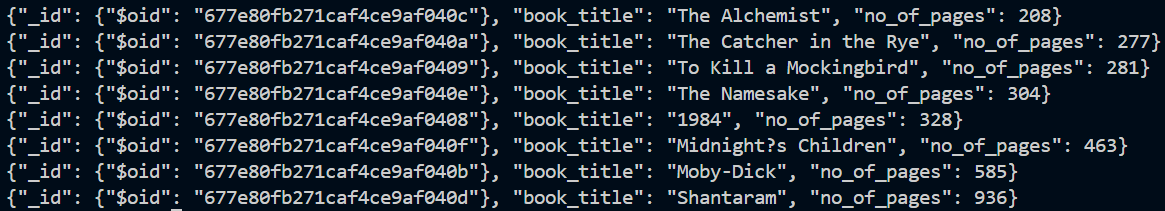
e.printStackTrace();

}

}

}

**OUTPUT:**

****

**Q13] Match books with price above Rs 1000 , project their book\_title and price and sort by book title.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.FindIterable;

import com.mongodb.client.MongoCollection;

import org.bson.Document;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

Document projection = new Document("book\_title", 1)

.append("price\_in\_inr", 1);

Document filter = new Document("price\_in\_inr", new Document("$gt", 1000));

FindIterable<Document> result = collection.find(filter)

.projection(projection)

.sort(new Document("book\_title", 1));

for (var document : result) {

System.out.println(document.toJson());

}

} catch (Exception e) {

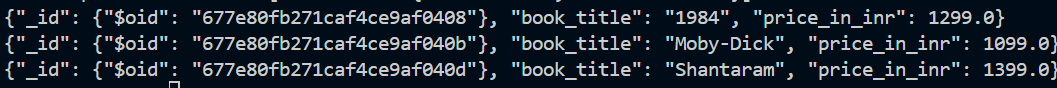
e.printStackTrace();

}

}

}

**OUTPUT:**

****

**Q14] Unwind field book title and group them to get the sum of total pages.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.MongoCollection;

import com.mongodb.client.AggregateIterable;

import org.bson.Document;

import java.util.Arrays;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

AggregateIterable<Document> result = collection.aggregate(Arrays.asList(

new Document("$unwind", "$book\_title"),

new Document("$group", new Document("\_id", null)

.append("total\_pages", new Document("$sum", "$no\_of\_pages")))

));

for (var document : result) {

System.out.println(document.toJson());

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

**OUTPUT:**

****

**Q15] Store the books with title starting from letter ‘T’ , author name and price in new collection books2.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.MongoCollection;

import com.mongodb.client.FindIterable;

import org.bson.Document;

import java.util.ArrayList;

import java.util.List;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

var collection2 = mongoClient.getDatabase("newdatabase").getCollection("books2");

Document filter = new Document("book\_title", new Document("$regex", "^T").append("$options", "i"));

FindIterable<Document> result = collection.find(filter);

List<Document> booksToInsert = new ArrayList<>();

for (var document : result) {

Document newDocument = new Document("book\_title", document.get("book\_title"))

.append("author\_name", document.get("author\_name"))

.append("price", document.get("price"));

booksToInsert.add(newDocument);

}

if (!booksToInsert.isEmpty()) {

collection2.insertMany(booksToInsert);

System.out.println("Books inserted into 'books2' collection.");

} else {

System.out.println("No books found with title starting with 'T'.");

}

} catch (Exception e) {

e.printStackTrace();

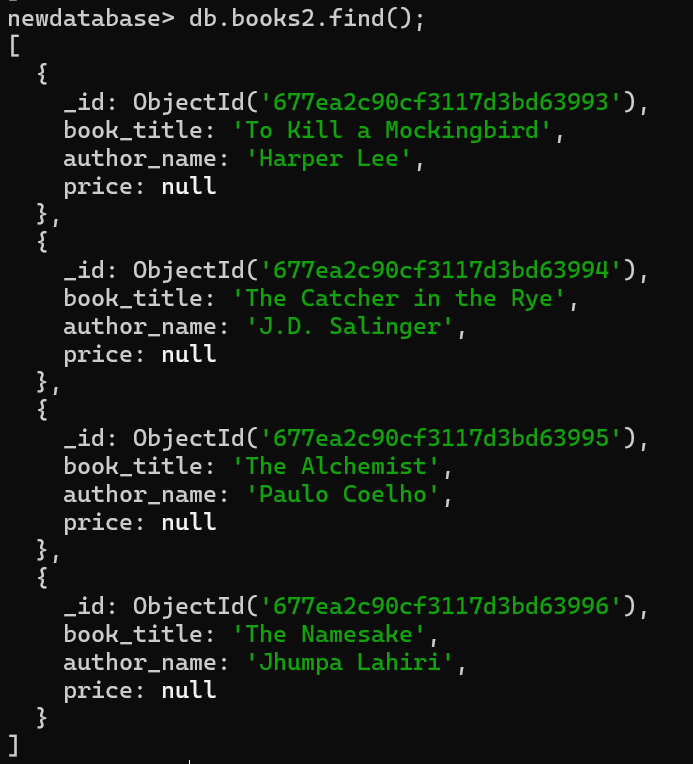
}

}

}

**OUTPUT:**

****

****