db.products.insertMany([

{ "\_id" : 1, "name" : "xPhone", "price" : 799, "releaseDate": ISODate("2011-05-14"), "spec" : { "ram" : 4, "screen" : 6.5, "cpu" : 2.66 },"color":["white","black"],"storage":[64,128,256]},

{ "\_id" : 2, "name" : "xTablet", "price" : 899, "releaseDate": ISODate("2011-09-01") , "spec" : { "ram" : 16, "screen" : 9.5, "cpu" : 3.66 },"color":["white","black","purple"],"storage":[128,256,512]},

{ "\_id" : 3, "name" : "SmartTablet", "price" : 899, "releaseDate": ISODate("2015-01-14"), "spec" : { "ram" : 12, "screen" : 9.7, "cpu" : 3.66 },"color":["blue"],"storage":[16,64,128]},

{ "\_id" : 4, "name" : "SmartPad", "price" : 699, "releaseDate": ISODate("2020-05-14"),"spec" : { "ram" : 8, "screen" : 9.7, "cpu" : 1.66 },"color":["white","orange","gold","gray"],"storage":[128,256,1024]},

{ "\_id" : 5, "name" : "SmartPhone", "price" : 599,"releaseDate": ISODate("2022-09-14"), "spec" : { "ram" : 4, "screen" : 5.7, "cpu" : 1.66 },"color":["white","orange","gold","gray"],"storage":[128,256]}

])

package MONGODBCONNECTION;

import org.bson.Document;

import com.mongodb.client.MongoClient;

import com.mongodb.client.MongoClients;

import com.mongodb.client.MongoCollection;

import com.mongodb.client.MongoDatabase;

import java.util.Arrays;

public class MONGODBCONNECTION {

public static void main(String[] args) {

MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017");

MongoDatabase database = mongoClient.getDatabase("test");

MongoCollection<Document> collection = database.getCollection("products");

Document avgResult = collection.aggregate(Arrays.asList(

new Document("$group",

new Document("\_id", null)

.append("averagePrice", new Document("$avg", "$price"))

))

).first();

if (avgResult != null) {

System.out.println(avgResult.getDouble("averagePrice"));

} else {

System.out.println("No products found");

}

mongoClient.close();

}

}

MongoDatabase database = mongoClient.getDatabase("vit");

MongoCollection<Document> collection = database.getCollection("products");

Document priceRangeQuery = new Document("price", new Document("$gte", 700).append("$lte", 900));

FindIterable<Document> products = collection.find(priceRangeQuery);

package MONGODBCONNECTION;

import org.bson.Document;

import com.mongodb.client.FindIterable;

import com.mongodb.client.MongoClient;

import com.mongodb.client.MongoClients;

import com.mongodb.client.MongoCollection;

import com.mongodb.client.MongoDatabase;

import static com.mongodb.client.model.Sorts.descending;

public class MONGODBCONNECTION {

public static void main(String[] args) {

MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017");

MongoDatabase database = mongoClient.getDatabase("test");

MongoCollection<Document> collection = database.getCollection("products");

Document priceRangeQuery = new Document("price", new Document("$gte", 700).append("$lte", 900));

FindIterable<Document> products = collection.find(priceRangeQuery);

for(Document document : products){

System.out.println(document);

}

mongoClient.close();

}

}

package MONGODBCONNECTION;

import org.bson.Document;

import com.mongodb.client.MongoClient;

import com.mongodb.client.MongoClients;

import com.mongodb.client.MongoCollection;

import com.mongodb.client.MongoDatabase;

import com.mongodb.client.MongoIterable;

import java.util.Arrays;

public class MONGODBCONNECTION {

public static void main(String[] args) {

MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017");

MongoDatabase database = mongoClient.getDatabase("test");

MongoCollection<Document> collection = database.getCollection("sales");

try {

MongoIterable<Document> sales = collection.aggregate(Arrays.asList(

new Document("$group",

new Document("\_id", "$item")

.append("maxPrice", new Document("$max", "$price"))),

new Document("$sort", new Document("\_id", 1))

));

System.out.println("Item\t\tMax Price");

System.out.println("------------------------");

for (Document doc : sales) {

System.out.printf("%-12s\t$%d%n",

doc.getString("\_id"),

doc.getInteger("maxPrice"));

}

} finally {

mongoClient.close();

}

}

}

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.