**11-06-2025**

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]}

])

db.products.find(

{ price: { $gte: 700, $lte: 900 } },

{ \_id: 0, name: 1, price: 1 }

)

package connection;

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.FindIterable;

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

public class Sorting {

public static void main(String[] args) {

// Connect to MongoDB server

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

// Get the database

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

// Get the collection

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

// Query with price filter and apply descending sort on "First\_Name"

FindIterable<Document> products = collection

.find(new Document("price", new Document("$gte", 700).append("$lte", 900)))

.sort(descending("First\_Name"));

// Print the sorted documents

for (Document document : products) {

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

}

// Close the connection

mongoClient.close();

}

}

db.sales.insertMany([

{ "\_id" : 1, "item" : "Americanos", "price" : 5, "size": "Short", "quantity" : 22, "date" : ISODate("2022-01-15T08:00:00Z") },

{ "\_id" : 2, "item" : "Cappuccino", "price" : 6, "size": "Short","quantity" : 12, "date" : ISODate("2022-01-16T09:00:00Z") },

{ "\_id" : 3, "item" : "Lattes", "price" : 15, "size": "Grande","quantity" : 25, "date" : ISODate("2022-01-16T09:05:00Z") },

{ "\_id" : 4, "item" : "Mochas", "price" : 25,"size": "Tall", "quantity" : 11, "date" : ISODate("2022-02-17T08:00:00Z") },

{ "\_id" : 5, "item" : "Americanos", "price" : 10, "size": "Grande","quantity" : 12, "date" : ISODate("2022-02-18T21:06:00Z") },

{ "\_id" : 6, "item" : "Cappuccino", "price" : 7, "size": "Tall","quantity" : 20, "date" : ISODate("2022-02-20T10:07:00Z") },

{ "\_id" : 7, "item" : "Lattes", "price" : 25,"size": "Tall", "quantity" : 30, "date" : ISODate("2022-02-21T10:08:00Z") },

{ "\_id" : 8, "item" : "Americanos", "price" : 10, "size": "Grande","quantity" : 21, "date" : ISODate("2022-02-22T14:09:00Z") },

{ "\_id" : 9, "item" : "Cappuccino", "price" : 10, "size": "Grande","quantity" : 17, "date" : ISODate("2022-02-23T14:09:00Z") },

{ "\_id" : 10, "item" : "Americanos", "price" : 8, "size": "Tall","quantity" : 15, "date" : ISODate("2022-02-25T14:09:00Z")}

]);

db.sales.aggregate([

{

$group: {

\_id: { item: "$item", size: "$size" },

maxPrice: { $max: "$price" }

}

},

{

$sort: { "\_id.item": 1, "maxPrice": -1 }

},

{

$group: {

\_id: "$\_id.item",

sizeWithMaxPrice: { $first: "$\_id.size" },

maxPrice: { $first: "$maxPrice" }

}

}

])

package connection;

import org.bson.Document;

import com.mongodb.client.\*;

import java.util.Arrays;

import java.util.List;

public class MaxPricePerItem {

public static void main(String[] args) {

// Connect to MongoDB server

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

// Get the database and collection

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

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

// Define the aggregation pipeline

List<Document> pipeline = Arrays.asList(

// Stage 1: Group by item and size, get max price for each group

new Document("$group", new Document("\_id", new Document("item", "$item").append("size", "$size"))

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

// Stage 2: Sort by item and maxPrice descending

new Document("$sort", new Document("\_id.item", 1).append("maxPrice", -1)),

// Stage 3: Group by item, take the first (highest) maxPrice

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

.append("sizeWithMaxPrice", new Document("$first", "$\_id.size"))

.append("maxPrice", new Document("$first", "$maxPrice")))

);

// Run the aggregation

AggregateIterable<Document> result = collection.aggregate(pipeline);

// Print the results

for (Document doc : result) {

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

}

// Close the connection

mongoClient.close();

}

}