**MongoDB Query:**

db.products.find({price: { $gte: 700, $lte: 900 }})

**Java Code:**

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;

public class MongoDB {

public static void main(String[] args) {

MongoClient mongoClient = MongoClients.create("mongodb://localhost:20472");

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);

System.out.println("Products with price between $700 and $900:");

for (Document product : products) {

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

}

mongoClient.close();

}

}

**MongoDB Query:**

db.products.aggregate([

{

$group: {

\_id: null,

avgPrice: { $avg: "$price" }

}

}

])

**Java Code:**

**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.AggregateIterable;

**import** **static** com.mongodb.client.model.Aggregates.\*;

**import** **static** com.mongodb.client.model.Accumulators.\*;

**import** java.util.Arrays;

**public** **class** MongoDB {

**public** **static** **void** main(String[] args) {

MongoClient mongoClient = MongoClients.*create*("mongodb://localhost:20472");

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

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

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

*group*(**null**, *avg*("avgPrice", "$price"))

));

**for** (Document doc : result) {

System.***out***.println("Average Price: " + doc.getDouble("avgPrice"));

}

mongoClient.close();

}

}

**MongoDB Query:**

db.sales.aggregate([{$group: {

\_id: "$item",

highestPrice: { $max: "$price" },

sizesAtHighestPrice: { $addToSet: "$size" }}},

{

$sort: { highestPrice: -1 }

}])

**Java Code:**

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 static com.mongodb.client.model.Aggregates.\*;

import static com.mongodb.client.model.Accumulators.\*;

import static com.mongodb.client.model.Sorts.\*;

import java.util.Arrays;

public class MongoDB {

public static void main(String[] args) {

try (MongoClient mongo = MongoClients.*create*("mongodb://localhost:20472")) {

MongoDatabase db = mongo.getDatabase("vit");

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

sales.aggregate(Arrays.*asList*(

*group*("$item",

*max*("maxPrice", "$price"),

*addToSet*("sizes", "$size")

),

*sort*(*descending*("maxPrice"))

)).forEach((Document doc) -> {

System.*out*.printf("%-10s: $%2d (Sizes: %s)%n",

doc.getString("\_id"),

doc.getInteger("maxPrice"),

doc.getList("sizes", String.class));

});

} } }}