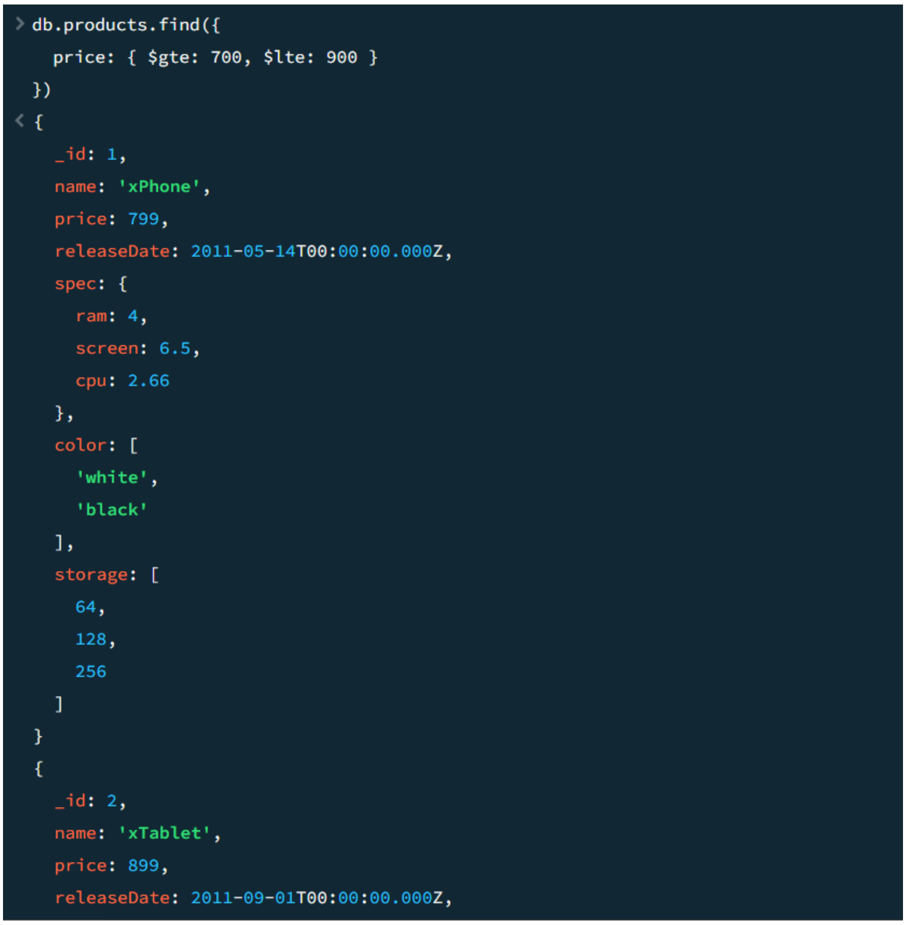
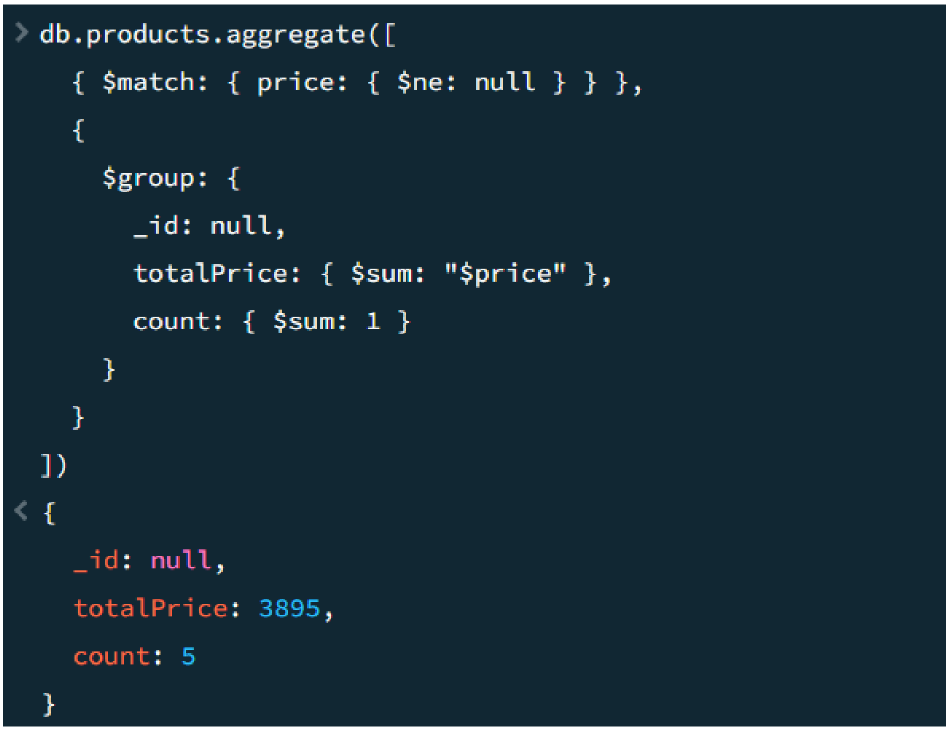
```
package connection;
import org.bson.Document;
import com.mongodb.client.FindIterable;
import com.mongodb.client.MongoClient;
import com.mongodb.client.MongoClients;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import com.mongodb.client.MongoIterable;

public class collection_two {
   public static void main(String[] args) {
      // Creating a Mongo client
      MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017");
      MongoDatabase database = mongoClient.getDatabase("myDb");
      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 product : products) {
          System.out.println(product.toJson());
       }


   }
}
```



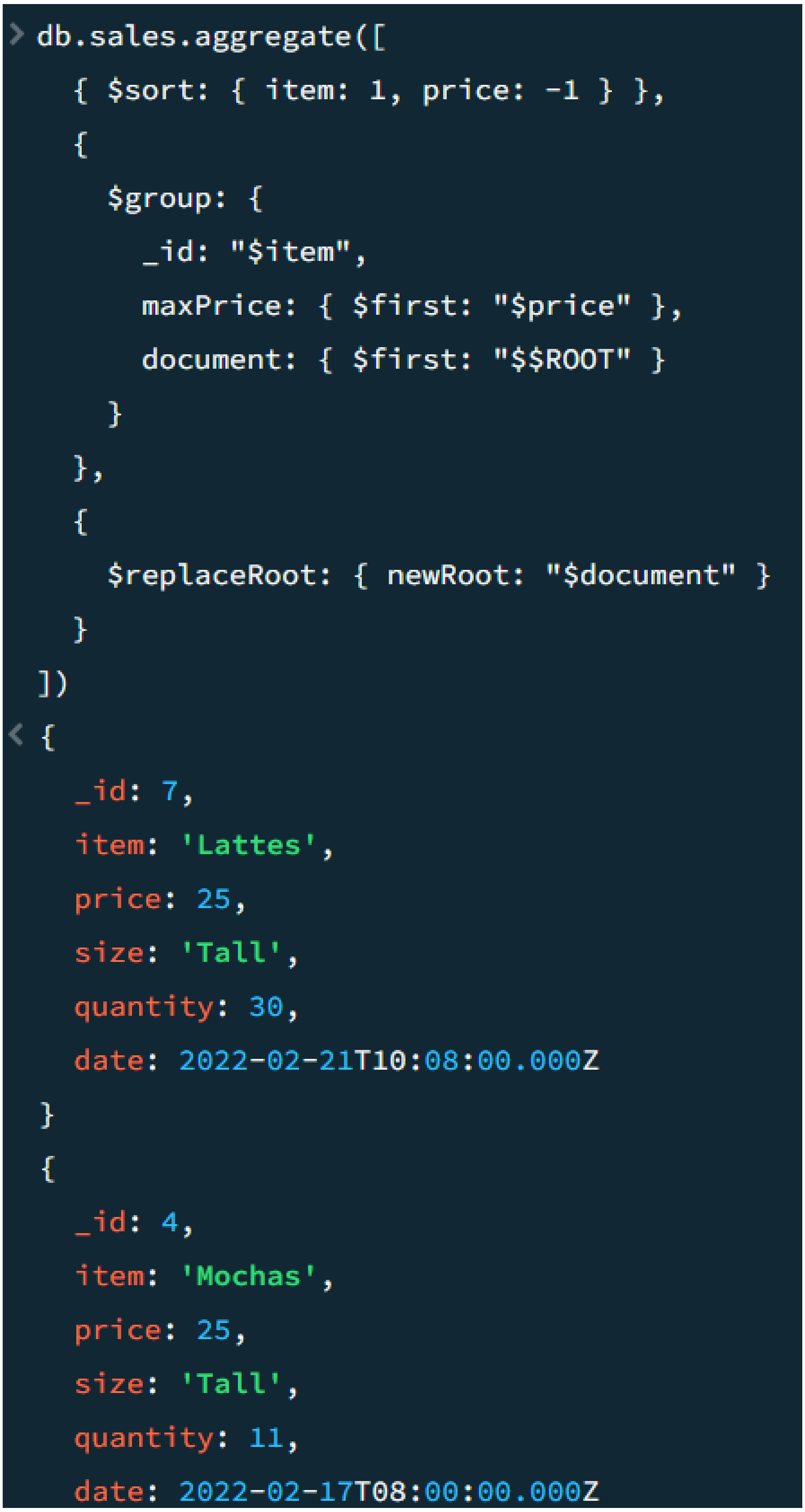
```
> db.products.find({
    price: { $gte: 700, $lte: 900 }
  })
< {
    _id: 1,
    name: 'xPhone',
    price: 799,
    releaseDate: 2011-05-14T00:00:00.000Z,
    spec: {
      ram: 4,
      screen: 6.5,
      cpu: 2.66
    },
    color: [
      'white',
      'black'
    ],
    storage: [
      64,
      128,
      256
    ]
  }
  {
    _id: 2,
    name: 'xTablet',
    price: 899,
    releaseDate: 2011-09-01T00:00:00.000Z,
```

```java
package connection;
import java.util.Arrays;

import org.bson.Document;

import com.mongodb.client.AggregateIterable;
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 com.mongodb.client.MongoIterable;

public class collection_two {
   public static void main(String[] args) {
      // Creating a Mongo client
      MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017");
      MongoDatabase database = mongoClient.getDatabase("myDb");
      MongoCollection<Document> collection = database.getCollection("products");
      AggregateIterable<Document> result = collection.aggregate(Arrays.asList(
              new Document("$match", new Document("price", new Document("$ne", 
null))),  // filter documents where price is not null
              new Document("$group", new Document("_id", null)
                  .append("totalPrice", new Document("$sum", "$price"))
                  .append("count", new Document("$sum", 1)))
          ));

          for (Document doc : result) {
              System.out.println("Total Price: " + doc.getInteger("totalPrice"));
              System.out.println("Count: " + doc.getInteger("count"));
          }


   }
}
```



```
> db.products.aggregate([
    { $match: { price: { $ne: null } } },
    {
      $group: {
        _id: null,
        totalPrice: { $sum: "$price" },
        count: { $sum: 1 }
      }
    }
  ])
< {
    _id: null,
    totalPrice: 3895,
    count: 5
  }
```

```
package connection;
import java.util.Arrays;

import org.bson.Document;

import com.mongodb.client.AggregateIterable;
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 com.mongodb.client.MongoIterable;

public class collection_two {
   public static void main(String[] args) {
      // Creating a Mongo client
      MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017");
      MongoDatabase database = mongoClient.getDatabase("myDb");
      MongoCollection<Document> collection = database.getCollection("sales");
      AggregateIterable<Document> result = collection.aggregate(Arrays.asList(
              new Document("$sort", new Document("item", 1).append("price", -1)),
              new Document("$group", new Document("_id", "$item")
                  .append("maxPrice", new Document("$first", "$price"))
                  .append("document", new Document("$first", "$$ROOT"))),
              new Document("$replaceRoot", new Document("newRoot", "$document"))
          ));

          for (Document doc : result) {
              System.out.println(doc.toJson());
          }



   }
}
```



```
> db.sales.aggregate([
    { $sort: { item: 1, price: -1 } },
    {
      $group: {
        _id: "$item",
        maxPrice: { $first: "$price" },
        document: { $first: "$$ROOT" }
      }
    },
    {
      $replaceRoot: { newRoot: "$document" }
    }
  ])
< {
    _id: 7,
    item: 'Lattes',
    price: 25,
    size: 'Tall',
    quantity: 30,
    date: 2022-02-21T10:08:00.000Z
  }
  {
    _id: 4,
    item: 'Mochas',
    price: 25,
    size: 'Tall',
    quantity: 11,
    date: 2022-02-17T08:00:00.000Z
```