Querying Mongo DB

Query Exercises
Data Model

Given the following collection of books

```json
{ _id: ObjectId("5fb29ae15b99900c3fa24292"),
  title: "MongoDb Guide",
  tag: ["mongodb", "guide", "database"],
  n: 100,
  review_score: 4.3,
  price: [ { v: 19.99, c: "€", country: "IT" },
           { v: 18, c: "£", country: "UK" } ],
  author: { _id: 1,
            name: "Mario",
            surname: "Rossi" }
},
{ _id: ObjectId("5fb29b175b99900c3fa24293"),
  title: "Developing with Python",
  tag: ["python", "guide", "programming"],
  n: 352,
  review_score: 4.6,
  price: [ { v: 24.99, c: "€", country: "IT" },
           { v: 19.49, c: "£", country: "UK" } ],
  author: { _id: 2,
            name: "John",
            surname: "Black" }
}, ...
```
Exercises

1. Find all the books with a **number of pages** greater than 250

2. Find all the books **authored** by Mario Rossi

3. Find all the books with a **price** less than 20 € for **Italy** (IT)
Solutions

• Find all the books with a number of pages greater than 250

  \[ \text{db.book.find\{n: \{\$gt: 250 \}\}} \]

• Find all the books authored by Mario Rossi

  \[ \text{db.book.find\{\"author.name\": \"Mario\", \"author.surname\": \"Rossi\" \}} \]

• Find all the books with a price less than 20 € for the country Italy (IT)

  \[ \text{db.book.find\{\"price\": \{\$elemMatch: \{\"v\": \{\$lt: 20\}, \"country\": \"IT\" \}\}} \]
Data Model

Given the following collection of books

```
{ _id: ObjectId("5fb29ae15b99900c3fa24292"),
  title: "MongoDb Guide",
  tag: ["mongodb", "guide", "database"],
  n: 100,
  review_score: 4.3,
  price: [ { v: 19.99, c: "€", country: "IT" },
            { v: 18, c: "£", country: "UK" } ],
  author: { _id: 1,
            name: "Mario",
            surname: "Rossi" },
},
{ _id: ObjectId("5fb29b175b99900c3fa24293"),
  title: "Developing with Python",
  tag: ["python", "guide", "programming"],
  n: 352,
  review_score: 4.6,
  price: [ { v: 24.99, c: "€", country: "IT" },
            { v: 19.49, c: "£", country: "UK" } ],
  author: { _id: 2,
            name: "John",
            surname: "Black" },
}, ...
```
Exercises

1. Increase the **review score** of 0.2 points for all the books with the **tag** “database”

2. Insert the **tag** “NoSQL” for all the books with **tag** “mongodb”

3. Insert the **publisher** for all the documents **authored** by Mario Rossi with the default value `{‘name’: ‘Polito’, city:‘Turin’}`
Solutions

• Increase the review score of 0.1 for all the books with the tag database

```javascript
db.book.updateMany({tag: "database"}, { $inc: {review_score: 0.2} })
```

• Insert the tag “NoSQL” for all the books with tag “mongodb”

```javascript
db.book.updateMany({tag: "mongodb"}, { $addToSet: {tag: "NoSQL"} })
```

• Insert the publisher for all the documents authored by Mario Rossi with the default value {'name': 'Polito', 'city':'Turin'}

```javascript
db.book.updateMany(
    "author.name": "Mario", "author.surname": "Rossi",
    { $set: {publisher: {name:"Polito", city:"Turin"}}} )
```
Data Model

Given the following collection of books

```json
{
  _id: ObjectId("5fb29ae15b99900c3fa24292"),
  title: "MongoDb Guide",
  tag: ["mongodb","guide","database"],
  n: 100,
  review_score: 4.3,
  author: {_id: 1, name: "Mario", surname: "Rossi"}
},
{
  _id: ObjectId("5fb29b175b99900c3fa24293"),
  title: "Developing with Python",
  tag: ["python","guide","programming"],
  n: 352,
  review_score: 4.6,
  author: {_id: 2, name: "John", surname: "Black"}
}, ...
```
Exercises

1. Find the maximum, the minimum and the average price of all the books with tag “database”

2. Compute the number of books authored by Mario Rossi
Solutions

• Find the maximum, the minimum and the average price of all the books with tag “database”

```javascript
db.book.aggregate([
  { $match: { tag: "database" } },
  { $unwind: "$price" },
  { $group: { _id: null, avg: { $avg: "$price.v" }, min: { $min: "$price.v" }, max: { $max: "$price.v" } } }
])
```

• Compute the number of books authored by Mario Rossi

```javascript
db.book.count({ "author.name": "Mario", "author.surname": "Rossi" })
```

```javascript
db.book.find({ "author.name": "Mario", "author.surname": "Rossi" }).count()
```