

MongoDB

Import bazy danych

```
mongoimport --db ziarko --collection jeopardy --type json  
--file JEOPARDY_QUESTIONS1.json --jsonArray
```

Rezultat:

```
kuba@jakub-ux410:~/Workspace/Bazy Danych/MongoDB$ mongoimport --db ziarko --coll  
ection jeopardy --type json --file JEOPARDY_QUESTIONS1.json --jsonArray  
2018-12-11T09:21:20.237+0100    connected to: localhost  
2018-12-11T09:21:22.328+0100    imported 216930 documents
```

```
kuba@jakub-ux410:~/Workspace/Bazy Danych/MongoDB$ mongo  
MongoDB shell version v3.6.3  
connecting to: mongodb://127.0.0.1:27017  
MongoDB server version: 3.6.3  
Server has startup warnings:  
2018-12-11T09:05:38.177+0100 I STORAGE  [initandlisten]  
2018-12-11T09:05:38.177+0100 I STORAGE  [initandlisten] ** WARNING: Using the XFS  
filesystem is strongly recommended with the WiredTiger storage engine  
2018-12-11T09:05:38.177+0100 I STORAGE  [initandlisten] ** See http://d  
ochub.mongodb.org/core/prodnotes-filesystem  
2018-12-11T09:05:38.721+0100 I CONTROL  [initandlisten]  
2018-12-11T09:05:38.721+0100 I CONTROL  [initandlisten] ** WARNING: Access contr  
ol is not enabled for the database.  
2018-12-11T09:05:38.721+0100 I CONTROL  [initandlisten] ** Read and wri  
te access to data and configuration is unrestricted.  
2018-12-11T09:05:38.721+0100 I CONTROL  [initandlisten]  
> use ziarko  
switched to db ziarko  
> db.jeopardy.count()  
216930  
>
```

Rezultat pozytywny. Baza zaimportowana poprawnie.

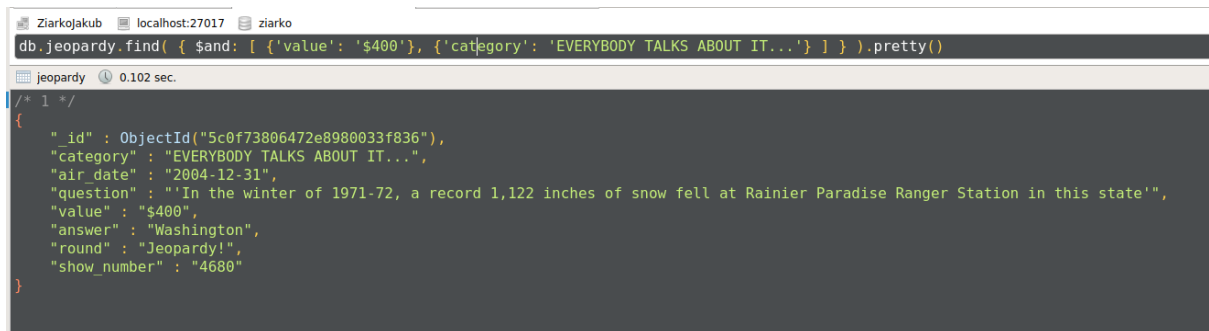
Zadanie 1

jedno proste (ale np. z warunkami, sortowaniem itp)

Komenda:

```
db.jeopardy.find( { $and: [ {'value': '$400'}, {'category': 'EVERYBODY TALKS ABOUT IT...'} ] } ).pretty()
```

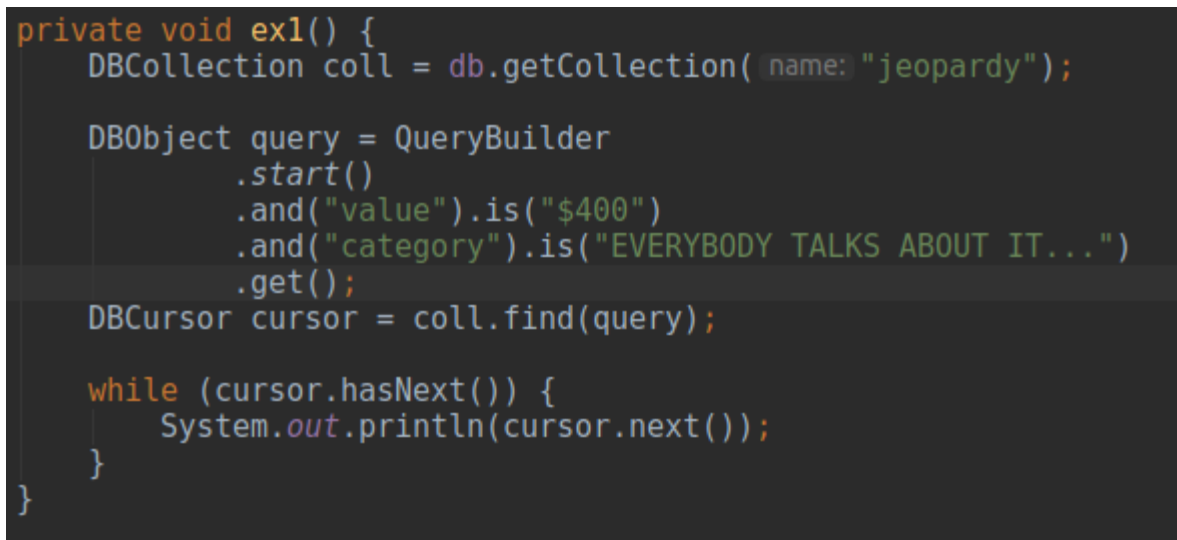
W Robomongo:



The screenshot shows the Robomongo web interface. At the top, the browser address bar shows 'localhost:27017' and the database 'ziarko'. The main area displays a MongoDB query: `db.jeopardy.find({ $and: [{'value': '$400'}, {'category': 'EVERYBODY TALKS ABOUT IT...'}] }).pretty()`. Below the query, the execution time is shown as '0.102 sec.'. The result is a single JSON document:

```
{
  "_id" : ObjectId("5c0f73806472e8980033f836"),
  "category" : "EVERYBODY TALKS ABOUT IT...",
  "air_date" : "2004-12-31",
  "question" : "'In the winter of 1971-72, a record 1,122 inches of snow fell at Rainier Paradise Ranger Station in this state'",
  "value" : "$400",
  "answer" : "Washington",
  "round" : "Jeopardy!",
  "show_number" : "4680"
}
```

W Javie:



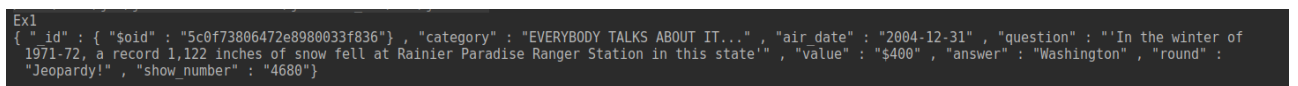
```
private void ex1() {
    DBCollection coll = db.getCollection( name: "jeopardy");

    DBObject query = QueryBuilder
        .start()
        .and("value").is("$400")
        .and("category").is("EVERYBODY TALKS ABOUT IT...")
        .get();

    DBCursor cursor = coll.find(query);

    while (cursor.hasNext()) {
        System.out.println(cursor.next());
    }
}
```

Rezultat z wywołania:



```
Ex1
{ "id" : { "$oid" : "5c0f73806472e8980033f836" }, "category" : "EVERYBODY TALKS ABOUT IT...", "air_date" : "2004-12-31", "question" : "'In the winter of 1971-72, a record 1,122 inches of snow fell at Rainier Paradise Ranger Station in this state'", "value" : "$400", "answer" : "Washington", "round" : "Jeopardy!", "show_number" : "4680" }
```

Zadanie 2

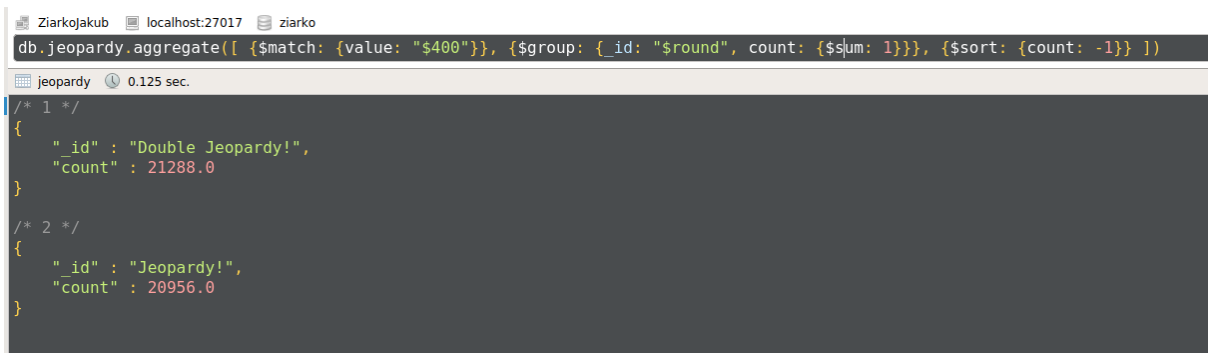
jedno z wykorzystaniem agregacji (dowolny sposób)

Tworzę zapytanie o produkty, które mają wartość '\$400', grupuję po round, zliczam ilość wystąpień oraz sortuję od największej ilości.

Komenda:

```
db.jeopardy.aggregate([ {$match: {value: "$400"}},  
{$group: {_id: "$round", count: {$sum: 1}}}, {$sort:  
{count: -1}} ])
```

W Robomongo:



```
db.jeopardy.aggregate([ {$match: {value: "$400"}},  
{$group: {_id: "$round", count: {$sum: 1}}}, {$sort: {count: -1}} ])
```

jeopardy 0.125 sec.

```
/* 1 */  
{  
  "_id" : "Double Jeopardy!",  
  "count" : 21288.0  
}  
  
/* 2 */  
{  
  "_id" : "Jeopardy!",  
  "count" : 20956.0  
}
```

W Javie:

```
private void ex2() {  
    DBCollection coll = db.getCollection( name: "jeopardy");  
  
    List<DBObject> pipeline = new ArrayList<>(Arrays.asList(  
        new BasicDBObject("$match", new BasicDBObject("value", "$400")),  
        new BasicDBObject("$group",  
            new BasicDBObject("_id", "$round")  
                .append( key: "count", new BasicDBObject("$sum", 1))),  
        new BasicDBObject("$sort", new BasicDBObject("count", -1))));  
  
    Cursor cursor = coll.aggregate(  
        pipeline,  
        AggregationOptions.builder().outputMode(AggregationOptions.OutputMode.CURSOR).build());  
  
    while (cursor.hasNext()) {  
        System.out.println(cursor.next());  
    }  
}
```

Rezultat z wywołania:

```
Ex2  
{ "_id" : "Double Jeopardy!" , "count" : 21288}  
{ "_id" : "Jeopardy!" , "count" : 20956}
```

Zadanie 3

jedno z wykorzystaniem mechanizmu map reduce

Zapytanie analogiczne do poprzedniego.

Komenda:

```
db.jeopardy.mapReduce(  
  function() { emit(this.round,1); },  
  function(key, values) {return Array.sum(values)}, {  
    query:{value:"$400"},  
    out:"count",  
  }  
)
```

W Robomongo:

```
db.jeopardy.mapReduce(  
  function() { emit(this.round,1); },  
  function(key, values) {return Array.sum(values)}, {  
    query:{value:"$400"},  
    out:"count",  
  }  
)
```

count 0.318 sec.

```
/* 1 */  
{  
  "_id" : "Double Jeopardy!",  
  "value" : 21288.0  
}  
  
/* 2 */  
{  
  "_id" : "Jeopardy!",  
  "value" : 20956.0  
}
```

W Javie:

```
private void ex3() {
    DBCollection coll = db.getCollection( name: "jeopardy");

    DBObject query = QueryBuilder
        .start()
        .and("value").is("$400")
        .get();

    MapReduceCommand command = new MapReduceCommand(
        coll,
        map: "function() { emit(this.round,1); }",
        reduce: "function(key, values) {return Array.sum(values)}",
        outputCollection: null,
        MapReduceCommand.OutputType.INLINE,
        query);

    MapReduceOutput result = coll.mapReduce(command);

    for (DBObject o : result.results()) {
        System.out.println(o.toString());
    }
}
```

Rezultat z wywołania:

```
Ex3
{ "_id" : "Double Jeopardy!" , "value" : 21288.0}
{ "_id" : "Jeopardy!" , "value" : 20956.0}
```

Dodatkowo

Pozostały kod:

```
public MongoLab() throws UnknownHostException {  
    mongoClient = new MongoClient();  
    db = mongoClient.getDB( dbname: "ziarko");  
}
```

```
public static void main(String[] args) throws UnknownHostException {  
    MongoLab mongoLab = new MongoLab();  
    System.out.println("Ex1");  
    mongoLab.ex1();  
    System.out.println("Ex2");  
    mongoLab.ex2();  
    System.out.println("Ex3");  
    mongoLab.ex3();  
}
```

runProject.sh w głównym katalogu:

```
java -jar ./mongoDbTask/out/artifacts/mongoDbTask_jar/mongoDbTask.jar|
```