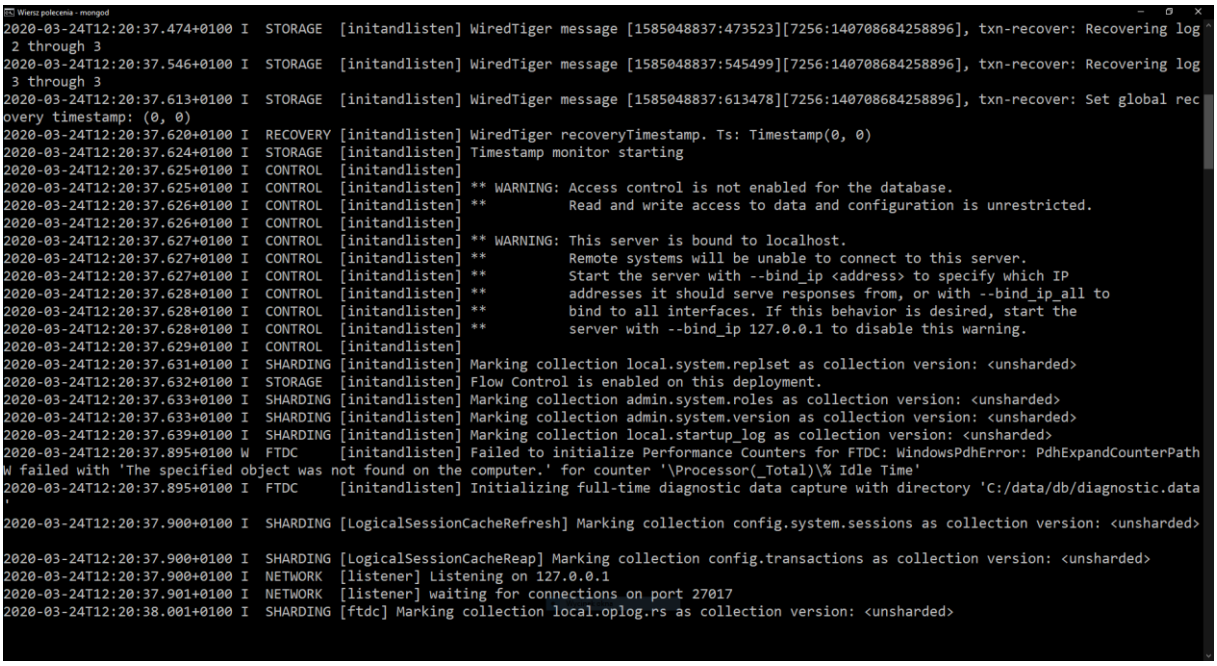
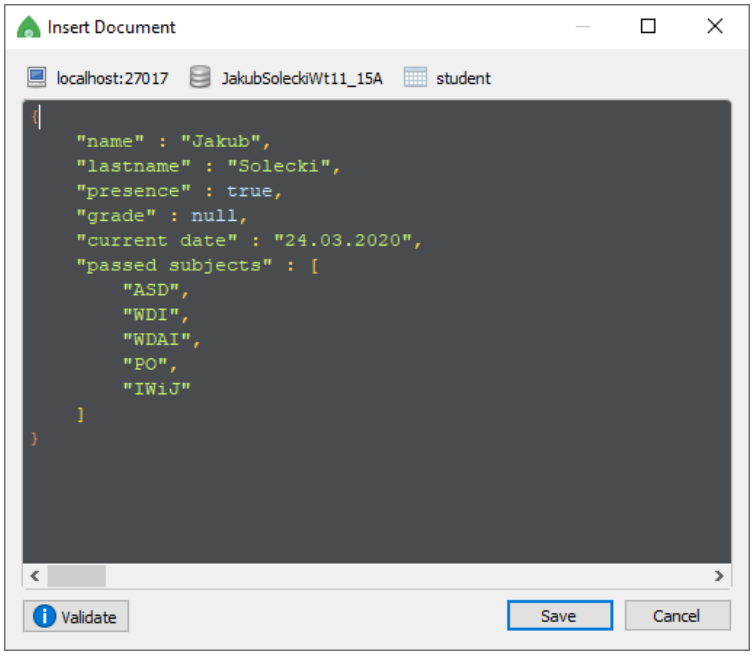
MongoDB – Raport z zajęć

Jakub Solecki

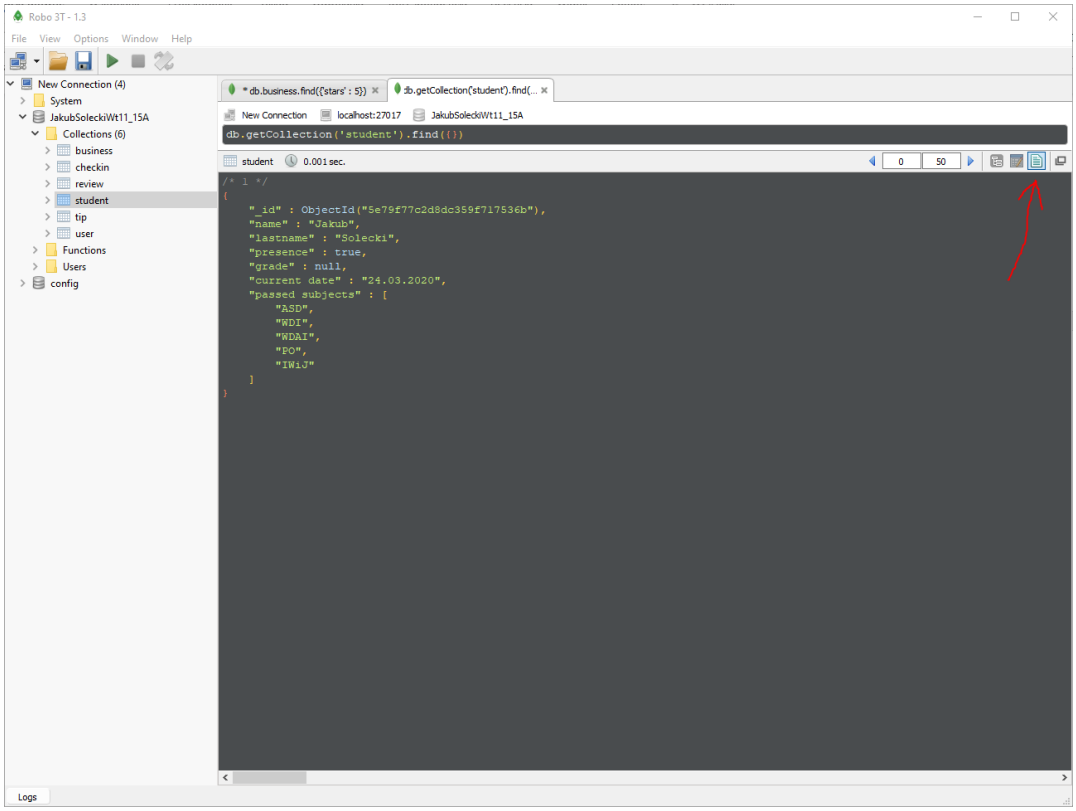
Zadanie 1, 2 oraz 3 zostały przeze mnie pomyślnie wykonane, jednak ze względu na konfiguracyjny chaos, który przy nich przeżyłem, nie posiadam z nich żadnych dowodów w postaci zrzutów ekranu, oprócz wciąż idącego procesu:



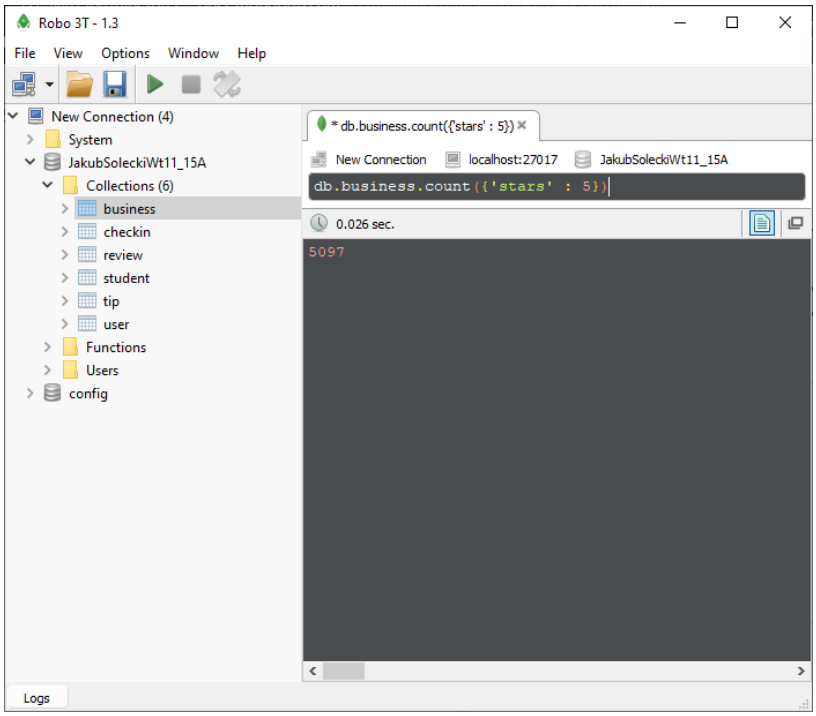
Zad 4a



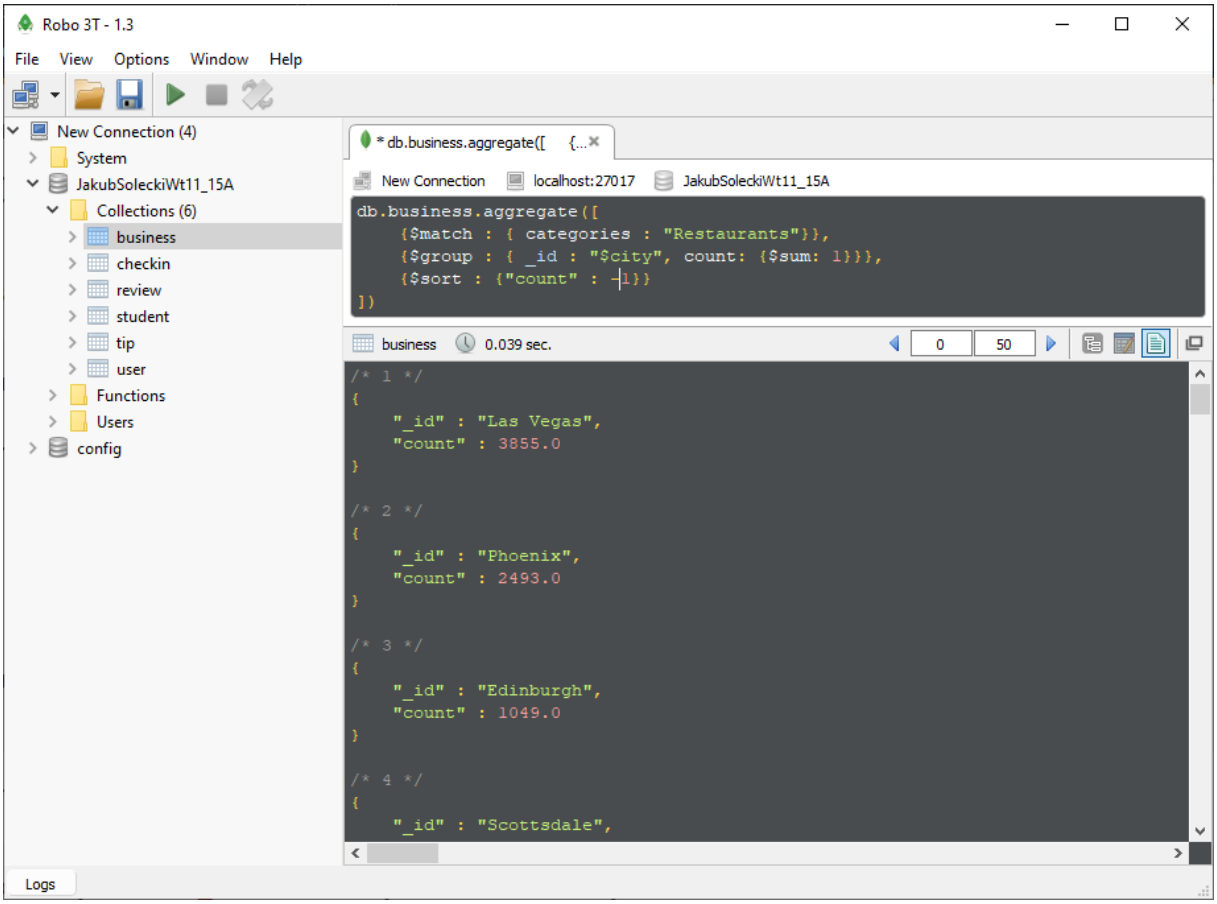
Zad 4b


Robo 3T - 1.3
File View Options Window Help
New Connection (4)
System
JakubSoleckiWt11_15A
Collections (6)
business
checkin
review
student
tip
user
Functions
Users
config
* db.business.find({'stars' : 5}) ×
db.getCollection('student').find(... ×
New Connection
localhost:27017
JakubSoleckiWt11_15A
db.getCollection('student').find({})
student
0.001 sec.
0
50
/* 1 */
{
"_id" : ObjectId("5e79f77c2d8dc359f717536b"),
"name" : "Jakub",
"lastname" : "Solecki",
"presence" : true,
"grade" : null,
"current date" : "24.03.2020",
"passed subjects" : [
"ASD",
"WDI",
"WDAI",
"PO",
"IWiJ"
]
}
Logs

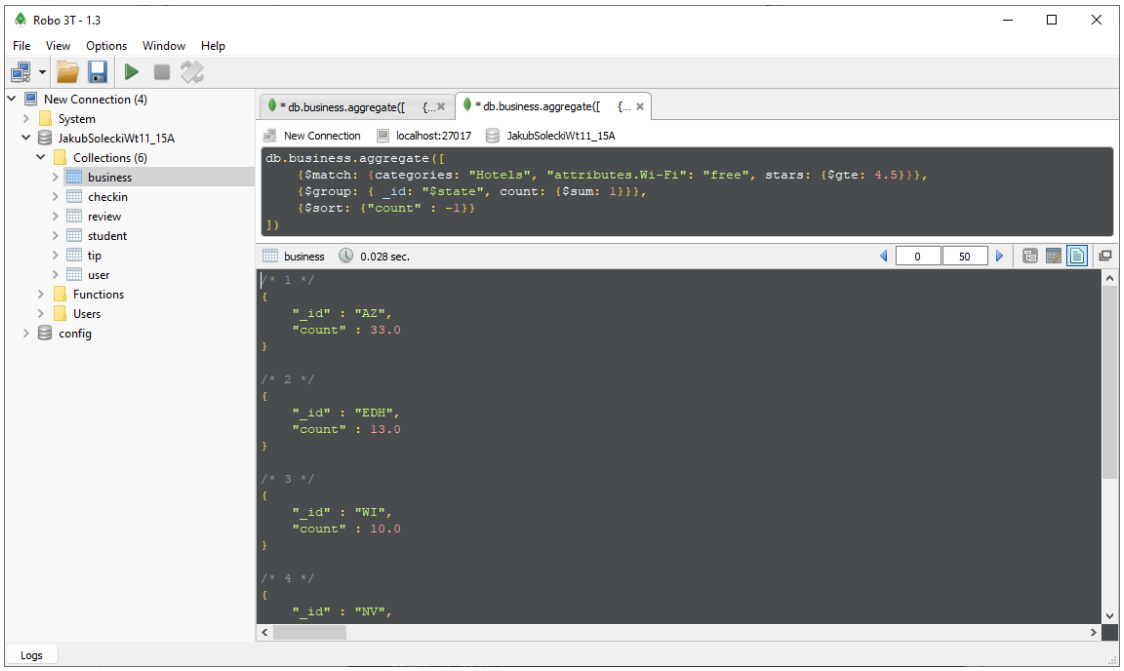
Zad 5a


Robo 3T - 1.3
File View Options Window Help
New Connection (4)
System
JakubSoleckiWt11_15A
Collections (6)
business
checkin
review
student
tip
user
Functions
Users
config
* db.business.count({'stars' : 5}) ×
New Connection
localhost:27017
JakubSoleckiWt11_15A
db.business.count({'stars' : 5})
0.026 sec.
5097
Logs

Zad 5b


Robo 3T - 1.3
File View Options Window Help
New Connection (4)
System
JakubSoleckiWt11_15A
Collections (6)
business
checkin
review
student
tip
user
Functions
Users
config
* db.business.aggregate([{...
New Connection localhost:27017 JakubSoleckiWt11_15A
db.business.aggregate([
{$match : { categories : "Restaurants"}},
{$group : { _id : "$city", count: {$sum: 1}}},
{$sort : {"count" : -1}}
])
business 0.039 sec.
0 50
/* 1 */
{
"_id" : "Las Vegas",
"count" : 3855.0
}
/* 2 */
{
"_id" : "Phoenix",
"count" : 2493.0
}
/* 3 */
{
"_id" : "Edinburgh",
"count" : 1049.0
}
/* 4 */
{
"_id" : "Scottsdale",
Logs

Zad 5c


Robo 3T - 1.3
File View Options Window Help
New Connection (4)
System
JakubSoleckiWt11_15A
Collections (6)
business
checkin
review
student
tip
user
Functions
Users
config
* db.business.aggregate([{... * db.business.aggregate([{...
New Connection localhost:27017 JakubSoleckiWt11_15A
db.business.aggregate([
{$match: {categories: "Hotels", "attributes.Wi-Fi": "free", stars: {$gte: 4.5}}},
{$group: { _id: "$state", count: {$sum: 1}}},
{$sort: {"count" : -1}}
])
business 0.028 sec.
0 50
/* 1 */
{
"_id" : "AZ",
"count" : 33.0
}
/* 2 */
{
"_id" : "EDH",
"count" : 13.0
}
/* 3 */
{
"_id" : "WI",
"count" : 10.0
}
/* 4 */
{
"_id" : "NV",
Logs

Zad 6

```
private long zad6_5a(){
    MongoCollection<Document> businessColl = db.getCollection( s: "business");
    BasicDBObject whereQuery = new BasicDBObject();
    whereQuery.put("stars", 5);
    return businessColl.countDocuments(whereQuery);

}
```

```
5097
```

```
private void zad6_5b() {
    MongoCollection<Document> businessColl = db.getCollection( s: "business");
    AggregateIterable agg = businessColl.aggregate(Arrays.asList(
                    Aggregates.match(Filters.eq( fieldName: "categories",  value: "Restaurants")),
                    Aggregates.group( id: "$city", Accumulators.sum( fieldName: "count",  expression: 1)),
                    Aggregates.sort((Sorts.descending( ...fieldNames: "count")))
            )
    );

    for(Object res : agg) {
        System.out.println((res));
    }
}
```

```
Document{{_id=Las Vegas, count=3855}}
Document{{_id=Phoenix, count=2493}}
Document{{_id=Edinburgh, count=1049}}
Document{{_id=Scottsdale, count=1023}}
Document{{_id=Mesa, count=693}}
Document{{_id=Madison, count=679}}
Document{{_id=Tempe, count=672}}
Document{{_id=Henderson, count=564}}
Document{{_id=Chandler, count=548}}
Document{{_id=Glendale, count=422}}
Document{{_id=Gilbert, count=317}}
Document{{_id=Peoria, count=221}}
Document{{_id=North Las Vegas, count=198}}
Document{{_id=Surprise, count=144}}
Document{{_id=Goodyear, count=119}}
Document{{_id=Waterloo, count=117}}
Document{{_id=Avondale, count=100}}
Document{{_id=Kitchener, count=96}}
Document{{_id=Queen Creek, count=82}}
Document{{_id=Middleton, count=66}}
Document{{_id=Cave Creek, count=63}}
Document{{_id=Casa Grande, count=61}}
Document{{_id=Fountain Hills, count=47}}
Document{{_id=Apache Junction, count=44}}
Document{{_id=Buckeye, count=42}}
Document{{_id=Sun Prairie, count=39}}
Document{{_id=Fitchburg, count=38}}
Document{{_id=Maricopa, count=37}}
Document{{_id=Monona, count=32}}
Document{{_id=Wickenburg, count=31}}
Document{{_id=Sun City, count=31}}
Document{{_id=Litchfield Park, count=27}}
Document{{_id=Paradise Valley, count=26}}
Document{{_id=Laveen, count=25}}
```

(i dużo więcej)

```java
private void zad6_5c() {
    MongoCollection<Document> businessColl = db.getCollection( s: "business");
    AggregateIterable agg = businessColl.aggregate(Arrays.asList(
            Aggregates.match(Filters.and(
                    Filters.eq( fieldName: "categories", value: "Hotels"),
                    Filters.eq( fieldName: "attributes.Wi-Fi", value: "free"),
                    Filters.gte( fieldName: "stars", value: 4.5)
                    )),
            Aggregates.group( id: "$state", Accumulators.sum( fieldName: "count", expression: 1)),
            Aggregates.sort(Sorts.descending( ...fieldNames: "count"))
            )
    );

    for(Object res : agg) {
        System.out.println((res));
    }

}
```

```
Document{{_id=AZ, count=33}}
Document{{_id=EDH, count=13}}
Document{{_id=WI, count=10}}
Document{{_id=NV, count=10}}
Document{{_id=ON, count=2}}
Document{{_id=MLN, count=1}}
```

Zad 8

```java
private List<String> zad8() {
    MongoCollection<Document> revs = db.getCollection( s: "review");
    List<String> count = new ArrayList<>();

    count.add("funny: " + revs.countDocuments(Filters.gt( fieldName: "votes.funny", value: 0)));
    count.add("cool: " + revs.countDocuments(Filters.gt( fieldName: "votes.cool", value: 0)));
    count.add("useful: " + revs.countDocuments(Filters.gt( fieldName: "votes.useful", value: 0)));

    return count;
}
```

```
[funny: 269256, cool: 346519, useful: 549519]
```