

LABORATORIUM NIERELACYJNE BAZY DANYCH

**Data wykonania
ćwiczenia:**

08.04.2023

Rok studiów:

3

Semestr:

6

Grupa studencka:

2

Grupa laboratoryjna:

2B

Ćwiczenie nr.

4

Temat: Indeksowanie w MongoDB

Osoby wykonujące ćwiczenia:

1. Igor Gawłowicz

Katedra Informatyki i Automatyki

Tworzenie bazy danych

Do zadania wykorzystamy bazę danych utworzoną na poprzednim zajęciach, obejmuje ona informację o ocenach studentów.

Struktura każdego rekordu wygląda w sposób następujący

```
[
  {
    "student_id": 1,
    "student_name": "Anna",
    "student_surname": "Nowak",
    "subjects": [
      {
        "subject": "math",
        "grade": 4,
        "final": true
      },
      {
        "subject": "history",
        "grade": 5,
        "final": true
      }
    ]
  }
]
```

Na potrzeby zadania wygenerowałem 100 w pełni losowych rekordów.

Teraz zbadamy wyniki przed i po stworzeniu indeksów, napisałem do tego program w Pythonie który wykona prostą kwerendę i zmierzy jej czas

```
from pymongo import MongoClient
import time

client = MongoClient('mongodb://localhost:27017')

db = client['my_app']

collection = db['courses']

query = {"student_surname": "Smith"}

start_time = time.time()
result = collection.find(query)
end_time = time.time()

execution_time = end_time - start_time
```

```
for doc in result:
    print(doc)

print("Query execution time: {:.6f} seconds".format(execution_time))
```

Po uruchomieniu widzimy, że pomimo braku indeksów baza działa na tyle szybko że przy tak małej liczbie rekordów wyniki są zbyt bliskie zeru

```
{'_id': ObjectId('65fa0fa4a88d6ebfd7d14a5a'), 'student_id': 76, 'student_name':
'Sherri', 'student_surname': 'Smith', 'subjects': [{'subject': 'Mathematics',
'grade': 2, 'final': False}, {'subject': 'English Language', 'grade': 2, 'final':
True}, {'subject': 'History', 'grade': 4, 'final': False}, {'subject': 'Science',
'grade': 5, 'final': False}, {'subject': 'Computer Science', 'grade': 5, 'final':
False}, {'subject': 'Physical Education', 'grade': 2, 'final': True}]}
```

Query execution time: 0.000000 seconds

Utworzymy teraz indeksy według polecenia

```
my_app> db.courses.createIndex({student_surname: 1})
student_surname_1
my_app> db.courses.createIndex({student_surname: 1, student_name: 1})
student_surname_1_student_name_1
my_app>
```

Z wcześniej wymienionych powodów wyniki są identyczne

```
{'_id': ObjectId('65fa0fa4a88d6ebfd7d14a5a'), 'student_id': 76, 'student_name':
'Sherri', 'student_surname': 'Smith', 'subjects': [{'subject': 'Mathematics',
'grade': 2, 'final': False}, {'subject': 'English Language', 'grade': 2, 'final':
True}, {'subject': 'History', 'grade': 4, 'final': False}, {'subject': 'Science',
'grade': 5, 'final': False}, {'subject': 'Computer Science', 'grade': 5, 'final':
False}, {'subject': 'Physical Education', 'grade': 2, 'final': True}]}
```

Query execution time: 0.000000 seconds

Następnie spróbujemy uruchomić program z nieco bardziej zaawansowaną kwerendą z użyciem sortowania

```
query = {"student_surname": "Smith"}
sort_criteria = [("subject", 1), ("grade", -1)]
result = collection.find(query).sort(sort_criteria)
```

Jednak wciąż, 100 rekordów to zbyt mała ilość żeby sprawdzić efektywność indeksów dlatego wynik w obu przypadkach jest zbyt zbliżony do zera

```
{'_id': ObjectId('65fa0fa4a88d6ebfd7d14a5a'), 'student_id': 76, 'student_name':  
'Sherri', 'student_surname': 'Smith', 'subjects': [{ 'subject': 'Mathematics',  
'grade': 2, 'final': False}, { 'subject': 'English Language', 'grade': 2, 'final':  
True}, { 'subject': 'History', 'grade': 4, 'final': False}, { 'subject': 'Science',  
'grade': 5, 'final': False}, { 'subject': 'Computer Science', 'grade': 5, 'final':  
False}, { 'subject': 'Physical Education', 'grade': 2, 'final': True}]}
```

Query execution time: 0.000000 seconds