

LABORATORIUM NIERELACYJNE BAZY DANYCH

**Data wykonania
ćwiczenia:**

22.04.2023

Rok studiów:

3

Semestr:

6

Grupa studencka:

2

Grupa laboratoryjna:

2B

Ćwiczenie nr.

6

Temat: Administracja Systemy zarządzania bazą danych (SZBD)

Osoby wykonujące ćwiczenia:

1. Igor Gawłowicz

Katedra Informatyki i Automatyki

Administracja Systemy zarządzania bazą danych (SZBD)

Uzyskaj informacje diagnostyczne dotyczące bazy danych i zawartych w niej kolekcji.

Aby uzyskać podstawowe dane diagnostyczne dla bazy danych możemy użyć poleceń `show dbs` oraz `show collections`, zobaczymy w ten sposób istniejące na tym hoście bazy danych oraz kolekcje, ciężko jednak nazwać to rzeczywistymi danymi diagnostycznymi

```
my_app> show dbs
admin    180.00 KiB
config   60.00 KiB
local    72.00 KiB
my_app    5.18 MiB
test     88.00 KiB

my_app> show collections
articles
courses
numbers
users
my_app>
```

Jeśli chcemy otrzymać kompletne dane na temat interesującej nas bazy możemy użyć poleceni `db.serverStatus()`, który zwróci nam gigantyczny blok danych, gdzie będziemy mogli dowiedzieć się między innymi takich rzeczy jak:

1. host
2. version
3. process
4. pid
5. uptime
6. uptimeMillis
7. uptimeEstimate
8. localTime
9. asserts
10. connections
11. replication
12. metrics
13. network
14. opcounters
15. opcountersRepl
16. oplog
17. backgroundFlushing
18. storageEngine
19. rocksdb
20. wiredTiger

21. logicalSessionRecordCache
22. transportSecurity
23. extra_info

Są to kompletne dane servera i analiza ich może być dość trudna, jednak zawierają praktycznie wszystkie informacje dotyczące danego servera na którym mieści się nasza baza danych.

Jeśli potrzebujemy jednak informacji o naszych kolekcjach możemy użyć polecenia

`db.getCollectionInfos({ type: "collection" })` na poziomie naszej bazy. W ten sposób dowiemy się wszystkiego co ważne o wszystkich kolekcjach w tym zbiorze.

```
my_app> db.getCollectionInfos({ type: "collection" })
[
  {
    name: 'articles',
    type: 'collection',
    options: {},
    info: {
      readOnly: false,
      uuid: UUID('c00aabbf1-1de7-4320-b9f7-0a63d295c76f')
    },
    idIndex: { v: 2, key: { _id: 1 }, name: '_id_' }
  },
  {
    name: 'numbers',
    type: 'collection',
    options: {},
    info: {
      readOnly: false,
      uuid: UUID('ce96ccd0-8b60-4dd2-8a36-39084ba0abf8')
    },
    idIndex: { v: 2, key: { _id: 1 }, name: '_id_' }
  },
  {
    name: 'users',
    type: 'collection',
    options: {},
    info: {
      readOnly: false,
      uuid: UUID('fcaa2bbf-1b00-4dd4-b251-4746ce07be8c')
    },
    idIndex: { v: 2, key: { _id: 1 }, name: '_id_' }
  },
  {
    name: 'courses',
    type: 'collection',
    options: {},
    info: {
      readOnly: false,
      uuid: UUID('feb39d54-393f-4fda-9e17-2949cab4c58')
    },
    idIndex: { v: 2, key: { _id: 1 }, name: '_id_' }
  }
]
```

```
}  
]
```

Utwórz kopię zapasową danych bazy danych.

Aby utworzyć kopię zapasową wykorzystamy narzędzie **mongodump**, należy je wcześniej zainstalować przez mongo toolsy po czym znaleźć jego lokalizację i w terminalu napisać

```
>mongodump --uri="connection string/db name" --out /path/to/backup/directory
```

W moim przypadku wyglądało to następująco

```
C:\Program Files\MongoDB\Tools\100\bin>mongodump --  
uri="mongodb://localhost:27017/my_app" --out C:\szkola\Szkola\SEMESTR6\NBD\lab6  
2024-04-22T13:55:22.732+0200    writing my_app.articles to  
C:\szkola\Szkola\SEMESTR6\NBD\lab6\my_app\articles.bson  
2024-04-22T13:55:22.734+0200    writing my_app.numbers to  
C:\szkola\Szkola\SEMESTR6\NBD\lab6\my_app\numbers.bson  
2024-04-22T13:55:22.734+0200    writing my_app.courses to  
C:\szkola\Szkola\SEMESTR6\NBD\lab6\my_app\courses.bson  
2024-04-22T13:55:22.761+0200    writing my_app.users to  
C:\szkola\Szkola\SEMESTR6\NBD\lab6\my_app\users.bson  
2024-04-22T13:55:22.772+0200    done dumping my_app.articles (0 documents)  
2024-04-22T13:55:22.779+0200    done dumping my_app.courses (100 documents)  
2024-04-22T13:55:22.780+0200    done dumping my_app.users (4 documents)  
2024-04-22T13:55:23.392+0200    done dumping my_app.numbers (200000 documents)
```

W wybranym folderze pojawił się teraz folder zawierający wszystkie dane mojej bazy.

Przywróć bazę danych z kopii zapasowej.

Aby przywrócić wcześniej utworzoną bazę danych zaczniemy od użycia polecenia **drop** aby pozbyć się bazy z systemu.

Następnie użyjemy narzędzia **mongorestore** aby przywrócić naszą bazę

```
mongorestore --uri="conneciton string" /path/to/backup/directory
```

W moim przypadku wygląda to następująco

```
>mongorestore --uri="mongodb://localhost:27017" C:\szkola\Szkola\SEMESTR6\NBD\lab6  
2024-04-22T14:02:58.339+0200    preparing collections to restore from  
2024-04-22T14:02:58.340+0200    don't know what to do with file  
"C:\szkola\Szkola\SEMESTR6\NBD\lab6\SprawozdanieNBD6_IgorGawlowicz.md",  
skipping...  
2024-04-22T14:02:58.343+0200    reading metadata for my_app.users from  
C:\szkola\Szkola\SEMESTR6\NBD\lab6\my_app\users.metadata.json  
2024-04-22T14:02:58.345+0200    reading metadata for my_app.articles from  
C:\szkola\Szkola\SEMESTR6\NBD\lab6\my_app\articles.metadata.json
```

```

2024-04-22T14:02:58.346+0200    reading metadata for my_app.courses from
C:\szkola\Szkola\SEMESTR6\NBD\lab6\my_app\courses.metadata.json
2024-04-22T14:02:58.346+0200    reading metadata for my_app.numbers from
C:\szkola\Szkola\SEMESTR6\NBD\lab6\my_app\numbers.metadata.json
2024-04-22T14:02:58.484+0200    restoring my_app.numbers from
C:\szkola\Szkola\SEMESTR6\NBD\lab6\my_app\numbers.bson
2024-04-22T14:02:58.484+0200    restoring my_app.courses from
C:\szkola\Szkola\SEMESTR6\NBD\lab6\my_app\courses.bson
2024-04-22T14:02:58.514+0200    restoring my_app.users from
C:\szkola\Szkola\SEMESTR6\NBD\lab6\my_app\users.bson
2024-04-22T14:02:58.535+0200    restoring my_app.articles from
C:\szkola\Szkola\SEMESTR6\NBD\lab6\my_app\articles.bson
2024-04-22T14:02:58.547+0200    finished restoring my_app.courses (100 documents,
0 failures)
2024-04-22T14:02:58.548+0200    finished restoring my_app.articles (0 documents, 0
failures)
2024-04-22T14:02:58.548+0200    finished restoring my_app.users (4 documents, 0
failures)
2024-04-22T14:03:01.344+0200    [#####.....] my_app.numbers
2.54MB/5.91MB  (43.0%)
2024-04-22T14:03:04.339+0200    [#####.....] my_app.numbers
5.03MB/5.91MB  (85.0%)
2024-04-22T14:03:05.292+0200    [#####] my_app.numbers
5.91MB/5.91MB  (100.0%)
2024-04-22T14:03:05.292+0200    finished restoring my_app.numbers (200000
documents, 0 failures)
2024-04-22T14:03:05.293+0200    no indexes to restore for collection
my_app.numbers
2024-04-22T14:03:05.293+0200    no indexes to restore for collection my_app.users
2024-04-22T14:03:05.293+0200    no indexes to restore for collection
my_app.articles
2024-04-22T14:03:05.293+0200    no indexes to restore for collection
my_app.courses
2024-04-22T14:03:05.293+0200    200104 document(s) restored successfully. 0
document(s) failed to restore.

```

Możemy się upewnić że baza danych jest spowrotem w systemie i wszystko przywróciło się poprawnie.

Utwórz wielu użytkowników dla bazy danych z różnymi rolami.

Zacniemy od utworzenia najprostszego użytkownika z dostępem tylko do odczytu danych na poziomie naszej bazy

```

use my_app

db.createUser({
  user: "reader",
  pwd: "123",
  roles: [{ role: "read", db: "my_app" }]
})

```

Następnie utworzymy użytkownika z dostępem do odczytu oraz zapisu

```
use my_app

db.createUser({
  user: "reader_write",
  pwd: "123",
  roles: [{ role: "readWrite", db: "my_app" }]
})
```

Następnie na poziomie systemu utworzymy administratora dla bazy danych

```
use admin

db.createUser({
  user: "db_admin",
  pwd: "123",
  roles: [{ role: "dbAdmin", db: "my_app" }]
})
```

Następnie zwiększymy zakres i stworzymy użytkownika z dostępem do całej klastry baz danych

```
use admin

db.createUser({
  user: "cluster_admin",
  pwd: "123",
  roles: [{ role: "clusterAdmin", db: "admin" }]
})
```

Finalnie utworzymy administratora całego systemu na rootcie

```
use admin

db.createUser({
  user: "root_admin",
  pwd: "123",
  roles: ["root"]
})
```