

# Lab #4

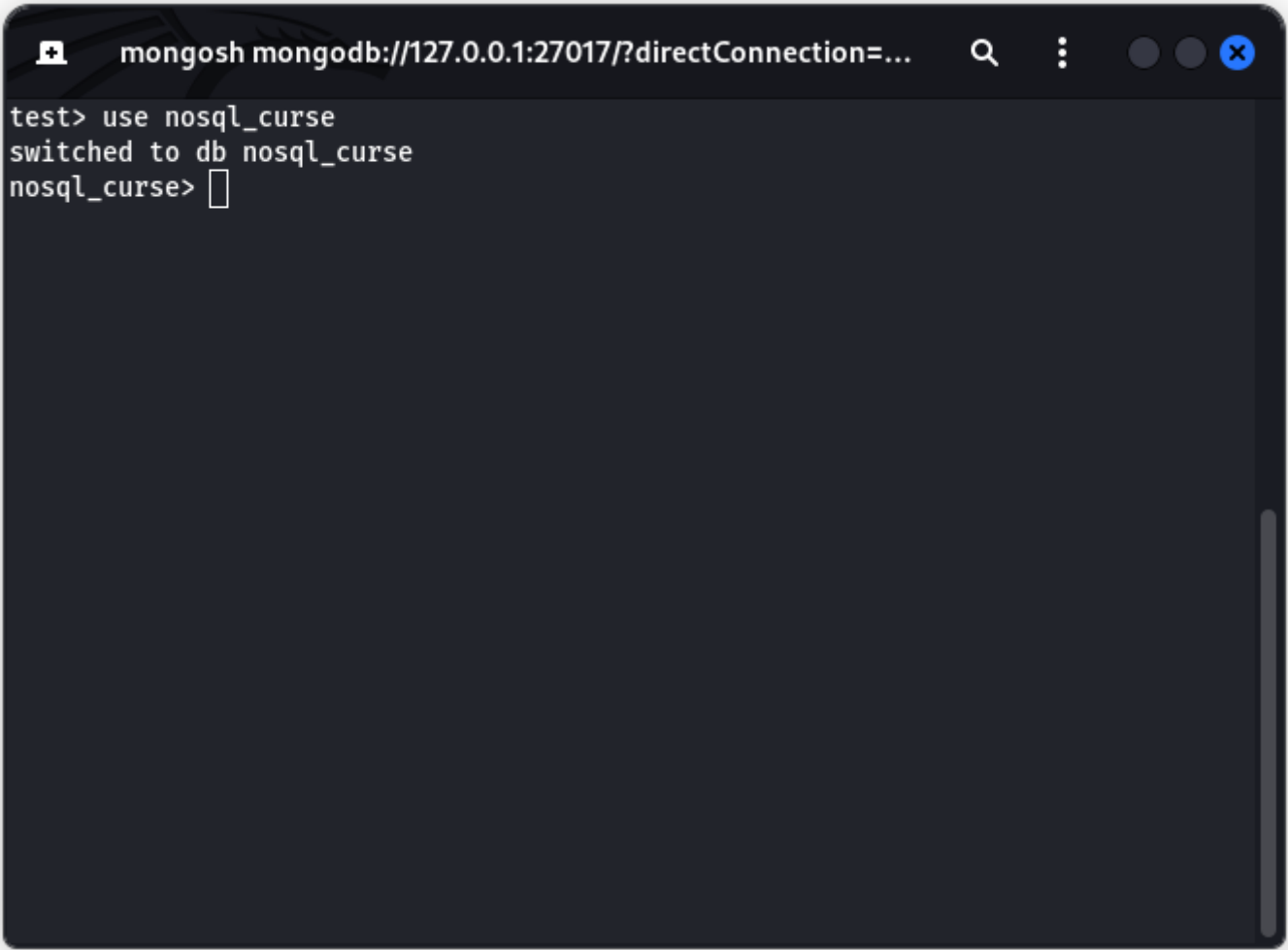
👤 Выполнил:	© Юрлов Павел ИАТ22-1м
@ FA E-mail:	<a href="mailto:221759@edu.fa.ru">221759@edu.fa.ru</a>
📎 Telegram:	<a href="https://t.me/paulyurlov">t.me/paulyurlov</a>

## Задачи лабораторной:

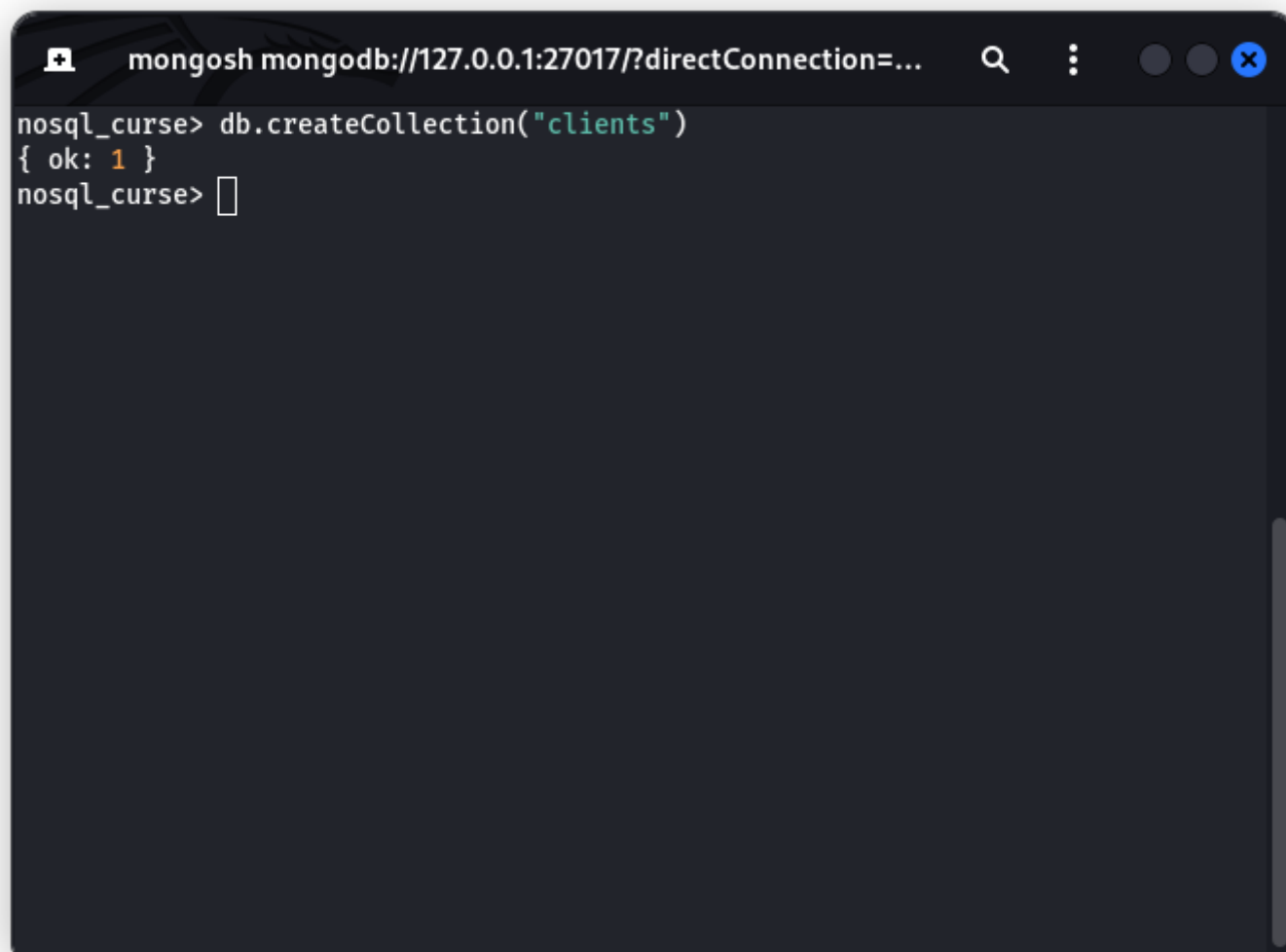
1. Создайте базу данных, с которой вы в дальнейшем будете работать
2. Создайте коллекцию, в которой у вас будут храниться документы.
3. Наполните коллекцию документами в соответствии с вашим индивидуальным вариантом. Создайте не менее 10 документов. Документы коллекции должны содержать, по крайней мере, один массив, и один вложенный документ.
4. Создайте составной и полнотекстовый индексы для документов в вашей коллекции.
5. Создайте запрос, производящий замену документа.
6. Создайте запрос, производящий обновление любого поля документа и запрос, добавляющий в него новые поля.
7. Создайте запрос для получения всех документов коллекции и определите, используется ли для него индекс.

## Ход выполнения работы:

1. **Создайте базу данных, с которой вы в дальнейшем будете работать**



2. Создайте коллекцию, в которой у вас будут храниться документы.



```
mongosh mongodb://127.0.0.1:27017/?directConnection=...
nosql_curse> db.createCollection("clients")
{ ok: 1 }
nosql_curse> 
```

3. Наполните коллекцию документами в соответствии с вашим индивидуальным вариантом. Создайте не менее 10 документов. Документы коллекции должны содержать, по крайней мере, один массив, и один вложенный документ.

```
db.clients.insertMany([
  {
    "name": "Charles",
    "surname": "Gagnon",
    "phone_number": "744-939-5373",
    "email": "charles.gagnon@example.com",
    "discount": 7,
    "document": {
      "type": "passport",
      "doc_id": "41f29f9b"
    },
    "deposit": [
      "document",
      "300$"
    ]
  },
  .....(тут еще 7 документов).....
  {
    "name": "Amy",
    "surname": "Smith",
    "phone_number": "431-616-6933",
    "email": "amy.smith@example.com",
    "discount": 9,
    "document": {
      "type": "id",
      "doc_id": "3248461e"
    },
    "deposit": [
      "document",
      "smartphone"
    ]
  },
  {
    "name": "Dylan",
    "surname": "Gauthier",
    "phone_number": "557-199-9993",
    "email": "dylan.gauthier@example.com",
    "discount": 12,
    "document": {
```

```

    "type": "id",
    "doc_id": "4dac684f"
  },
  "deposit": [
    "document",
    "300$"
  ]
}
])

```

```

...     "document",
...     "300$"
...   ]
... }
... ])
... {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("63bfbe3eabce91445b61cf6c"),
    '1': ObjectId("63bfbe3eabce91445b61cf6d"),
    '2': ObjectId("63bfbe3eabce91445b61cf6e"),
    '3': ObjectId("63bfbe3eabce91445b61cf6f"),
    '4': ObjectId("63bfbe3eabce91445b61cf70"),
    '5': ObjectId("63bfbe3eabce91445b61cf71"),
    '6': ObjectId("63bfbe3eabce91445b61cf72"),
    '7': ObjectId("63bfbe3eabce91445b61cf73"),
    '8': ObjectId("63bfbe3eabce91445b61cf74"),
    '9': ObjectId("63bfbe3eabce91445b61cf75")
  }
}
nosql_curse>

```

4. Создайте составной и полнотекстовый индексы для документов в вашей коллекции.

**NOTE**

**Removed in 5.0**

`db.collection.ensureIndex()` has been replaced by `db.collection.createIndex()`.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serv...
nosql_curse> db.clients.createIndex({"_id": 1, "email": 1})
_id_1_email_1
nosql_curse> db.clients.indexes.find()

nosql_curse> db.clients.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { _id: 1, email: 1 }, name: '_id_1_email_1' }
]
nosql_curse> db.clients.createIndex({"phone_number" : "text"})
phone_number_text
nosql_curse> db.clients.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { _id: 1, email: 1 }, name: '_id_1_email_1' },
  {
    v: 2,
    key: { _fts: 'text', _ftsx: 1 },
    name: 'phone_number_text',
    weights: { phone_number: 1 },
    default_language: 'english',
    language_override: 'language',
    textIndexVersion: 3
  }
]
nosql_curse> 
```

5. Создайте запрос, производящий замену документа.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serv...
  phone_number: '431-616-6933',
  email: 'amy.smith@example.com',
  discount: 9,
  document: { type: 'id', doc_id: '3248461e' },
  deposit: [ 'document', 'smartphone' ]
},
{
  _id: ObjectId("63bfb3eabce91445b61cf75"),
  name: 'Dylan',
  surname: 'Gauthier',
  phone_number: '557-199-9993',
  email: 'dylan.gauthier@example.com',
  discount: 12,
  document: { type: 'id', doc_id: '4dac684f' },
  deposit: [ 'document', '300$' ]
}
]
nosql_curse> db.clients.replaceOne({"name": "Dylan"}, {"name": "wow, so empty"})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
nosql_curse> 
```

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serv...
document: { type: 'driving_license', doc_id: 'c216f4ef' },
deposit: [ 'document', '300$' ]
},
{
  _id: ObjectId("63bfbe3eabce91445b61cf73"),
  name: 'Benjamin',
  surname: 'Scott',
  phone_number: '860-722-3816',
  email: 'benjamin.scott@example.com',
  discount: 4,
  document: { type: 'driving_license', doc_id: '5e0ab275' },
  deposit: [ 'document', '300$' ]
},
{
  _id: ObjectId("63bfbe3eabce91445b61cf74"),
  name: 'Amy',
  surname: 'Smith',
  phone_number: '431-616-6933',
  email: 'amy.smith@example.com',
  discount: 9,
  document: { type: 'id', doc_id: '3248461e' },
  deposit: [ 'document', 'smartphone' ]
},
{ _id: ObjectId("63bfbe3eabce91445b61cf75"), name: 'wow, so empty' }
]
nosql_curse> 
```

6. Создайте запрос, производящий обновление любого поля документа и запрос, добавляющий в него новые поля.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelecti...
document: { type: 'id', doc_id: '3248461e' },
deposit: [ 'document', 'smartphone' ]
},
{ _id: ObjectId("63bfbe3eabce91445b61cf75"), name: 'wow, so empty' }
]
nosql_curse> db.clients.updateOne({"name": "wow, so empty"}, {$set: {"name": "lol kek!"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
nosql_curse> db.clients.find({"name": "lol kek!"})
[ { _id: ObjectId("63bfbe3eabce91445b61cf75"), name: 'lol kek!' } ]
nosql_curse> db.clients.updateOne({"name": "lol kek!"}, {$set: {"hmm": "i love bananas!"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
nosql_curse> db.clients.find({"name": "lol kek!"})
[
  {
    _id: ObjectId("63bfbe3eabce91445b61cf75"),
    name: 'lol kek!',
    hmm: 'i love bananas!'
  }
]
nosql_curse> 
```

7. Создайте запрос для получения всех документов коллекции и определите, используется ли для него индекс.

Запустим команду:

```
db.clients.find().explain("executionStats")
```



```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
discount: 4,
document: { type: 'driving_license', doc_id: '5e0ab275' },
deposit: [ 'document', '300$' ]
},
{
  _id: ObjectId("63bfbe3eabce91445b61cf74"),
  name: 'Amy',
  surname: 'Smith',
  phone_number: '431-616-6933',
  email: 'amy.smith@example.com',
  discount: 9,
nosql_curse> db.clients.find().explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'nosql_curse.clients',
    indexFilterSet: false,
    parsedQuery: {},
    queryHash: '17830885',
    planCacheKey: '17830885',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: { stage: 'COLLSCAN', direction: 'forward' },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
```

## Теоретические вопросы

1. Какие способы создания БД присутствуют в MongoDB?

Через shell оболочку используется только команда use, в GUI интерфейсах есть свой функционал

2. Возможно ли неявное создание коллекции?

Да

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

[
  {
    name: 'Lab #4',
    type: 'collection',
    options: {},
    info: {
      readOnly: false,
      uuid: new UUID("55519e9b-44a5-4a0f-bef9-4e72f3022ba3")
    },
    idIndex: { v: 2, key: { _id: 1 }, name: '_id_' }
  },
  {
    name: 'clients',
    type: 'collection',
    options: {},
    info: {
      readOnly: false,
      uuid: new UUID("81d6ee51-8f0c-4ec9-87f0-3f8af6bc38bd")
    },
    idIndex: { v: 2, key: { _id: 1 }, name: '_id_' }
  },
  {
    name: 'test',
    type: 'collection',
    options: {},
    info: {
      readOnly: false,
      uuid: new UUID("9882e0d9-79a2-41e1-b2cc-22c04f2890cb")
    },
    idIndex: { v: 2, key: { _id: 1 }, name: '_id_' }
  }
]
nosql_curse> db.test_2.insert({"kek": "lol"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63bfe64b1b01dd81dccd03ed") }
}
nosql_curse> 
```

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

type: 'collection',
options: {},
info: {
  readOnly: false,
  uuid: new UUID("9882e0d9-79a2-41e1-b2cc-22c04f2890cb")
},
idIndex: { v: 2, key: { _id: 1 }, name: '_id_' }
}
]
nosql_curse> db.test_2.insert({"kek": "lol"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63bfe64b1b01dd81dccd03ed") }
}
nosql_curse> db.getCollectionInfos()
[
  {
    name: 'test_2',
    type: 'collection',
    options: {},
    info: {
      readOnly: false,
      uuid: new UUID("42cb2300-9ecb-4b79-b226-fc764afb4605")
    },
    idIndex: { v: 2, key: { _id: 1 }, name: '_id_' }
  },
  {
    name: 'Lab #4',
    type: 'collection',
    options: {},
    info: {
      readOnly: false,
      uuid: new UUID("55519e9b-44a5-4a0f-bef9-4e72f3022ba3")
    },
    idIndex: { v: 2, key: { _id: 1 }, name: '_id_' }
  },
  {
    name: 'clients',

```

3. Каков синтаксис функции обновления документа?



```

db.collection.updateOne(
  <filter>,
  <update>,
  {
    upsert: <boolean>,
    writeConcern: <document>,
    collation: <document>,
    arrayFilters: [ <filterdocument1>, ... ],
    hint: <document|string>          // Available starting in MongoDB 4.2.1
  }
)

```

4. Приведите пример запроса для обновления поля документа?

```

db.clients.updateOne({"name": "kekerman"}, {$set: {"name": "lolman"}})

```

5. Какие модификаторы функции обновления вы знаете? Перечислите их назначение.

```

upsert: <boolean>,
writeConcern: <document>,
collation: <document>,
arrayFilters: [ <filterdocument1>, ... ],
hint: <document|string>

```

6. С помощью какой функции вы можете удалить документ?  
deleteOne

7. Для чего используется индекс в БД попро?

Для ускорения работы запросов

8. Как создается и удаляется индекс в MongoDB?

createIndex - создать индекс

dropIndex - удалить индекс

9. Перечислите опции создания индекса?

unique, name, partialFilterExpression, sparse, expireAfterSeconds, hidden, storageEngine, collation

10. Какие типы индексов можно создать в Мопро)В?

Single Field, Compound Index, Multikey Index, Geospatial Index, Text Index, Hashed Index, Clustered Index