

УРОК 16.1. СОЗДАНИЕ И НАПОЛНЕНИЕ КОЛЛЕКЦИЙ. АГРЕГАЦИЯ

СОЗДАНИЕ И НАПОЛНЕНИЕ КОЛЛЕКЦИЙ	2
ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ	8
УДАЛЕНИЕ ДОКУМЕНТОВ И ВНЕСЕНИЕ ИЗМЕНЕНИЙ	10
ЭКСПОРТ И ИМПОРТ ДАННЫХ	12



СОЗДАНИЕ И НАПОЛНЕНИЕ КОЛЛЕКЦИЙ

Подключимся к базе данных ich_edit


У пользователя ich есть доступ ко всем коллекциям, но права на редактирование только в базе данных ich_edit у пользователя ich_edit

Вот строка подключения с правами на редактирование в базе ich_edit. Используем ее в mongo Compass:

```
mongodb://ich_editor:verystrongpassword@mongo.itcareerhub.de/?readPreference=primary&ssl=false&authMechanism=DEFAULT&authSource=ich_edit
```

New Connection


Connect to a MongoDB deployment

 FAVORITE

URI ⓘ Edit Connection String ☒

```
mongodb://ich_editor:verystrongpassword@3.67.41.21:27017/?
readPreference=primary&ssl=false&authMechanism=DEFAULT&authSource=ich_edit
```

> Advanced Connection Options

 TLS/SSL is disabled. If possible, enable TLS/SSL to avoid security vulnerabilities.

Save

Save & Connect

Connect



Попробуем создать коллекцию users и заполнить документами (4 штуки) со следующими свойствами (id, name, age, gender). Используйте следующие данные:

- 1 Анатолий 28 m
- 2 Светлана 25 f
- 3 Никита 33 m
- 4 Ольга 22 f

Можем выполнить при помощи mongosh и в самом compass
Вот какие команды нам надо выполнить в mongosh:

```
>_MONGOSH  
  
> use ich_edit;  
< switched to db ich_edit  
> db.createCollection("users_Andrew")  
< { ok: 1 }  
ich_edit> db.users_NAME.insertMany([  
  {  
    id: 1,  
    name: "Анатолий",  
    age: 28,  
    gender: "m"  
  },  
  {  
    id: 2,  
    name: "Светлана",  
    age: 25,  
    gender: "f"  
  },  
  {  
    id: 3,  
    name: "Никита",  
    age: 33,  
    gender: "m"
```

Не забываем заменить NAME на Ваше имя. Переключаемся на базу при помощи команды use .



Обратите внимание, что все команды мы начинаем с db и затем указываем коллекцию, над которой проводим изменения.

```
Unset
use ich_edit;

db.createCollection("users_NAME")

db.users_NAME.insertMany([
  {
    id: 1,
    name: "Анатолий",
    age: 28,
    gender: "m"
  },
  {
    id: 2,
    name: "Светлана",
    age: 25,
    gender: "f"
  },
  {
    id: 3,
    name: "Никита",
    age: 33,
    gender: "m"
  },
  {
    id: 4,
    name: "Ольга",
    age: 22,
    gender: "f"
  }
])
```



```
gender: "f"
},
{
  id: 3,
  name: "Никита",
  age: 33,
  gender: "m"
},
{
  id: 4,
  name: "Ольга",
  age: 22,
  gender: "f"
}
})
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("6581f78f6da63fd174ccce97"),
    '1': ObjectId("6581f78f6da63fd174ccce98"),
    '2': ObjectId("6581f78f6da63fd174ccce99"),
    '3': ObjectId("6581f78f6da63fd174ccce9a")
  }
}
```

Либо используя compass, нажав на ADD DATA и затем Insert document :

Обратите внимание на [] - квадратные скобки, так как мы вставляем несколько документов. Почти все запросы, которые вы будете искать в документации и онлайн будут рассчитаны на mongosh, поэтому ориентируемся на то, что идет внутри вызываемого метода.

Например:

```
Unset
db.users_NAME.insertMany([
  { ... .. } ] )
```

А в работу в compass мы берем только то, что в [] - квадратных скобках



ich_edit.users_Andrew

Documents

Aggregations

Schema

Indexes

Validation

Filter



Type a query: { field: 'value' } or [Generate query](#)

+ ADD DATA

EXPORT DATA

UPDATE

DELETE

Import JSON or CSV file

Insert document



This collection has no data

It only takes a few seconds to import data from a JSON or CSV file.

Import Data

Unset

Либо используя compass, нажав на ADD DATA:

```
[{
  "id": 1,
  "name": "Анатолий",
  "age": 28,
  "gender": "m"
},
{
  "id": 2,
  "name": "Светлана",
  "age": 25,
```



```
"gender": "f"
},
{
  "id": 3,
  "name": "Никита",
  "age": 33,
  "gender": "m"
},
{
  "id": 4,
  "name": "Ольга",
  "age": 22,
  "gender": "f"
}]
```

Sort { field: -1 } or [['field', 'asc']]

Collation { locale: 'simple' }

ADD DATA

EXPORT DATA

_id: ObjectId('6581f78f6da63fd174cc...')

id: 1

name: "Анатолий"

age: 28

gender: "m"

_id: ObjectId('6581f78f6da63fd174cc...')

id: 2

name: "Светлана"

age: 25

gender: "f"

_id: ObjectId('6581f78f6da63fd174cc...')

id: 3

name: "Никита"

age: 33

gender: "m"

_id: ObjectId('6581f78f6da63fd174cc...')

id: 4

name: "Ольга"

age: 22

gender: "f"

Insert Document

To collection ich_edit.users_NAME

VIEW

JSON

YAML

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

{

"id": 1,

"name": "Анатолий",

"age": 28,

"gender": "m"

}

,

{

"id": 2,

"name": "Светлана",

"age": 25,

"gender": "f"

}

,

{

"id": 3,

"name": "Никита",

"age": 33,

"gender": "m"

}

,

{

"id": 4,

"name": "Ольга",

"age": 22,

"gender": "f"

}

}

Cancel

Insert



ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

1. Вывести данные о всех представительницах женского пола

ich_edit.users_Andrew

Documents Aggregations Schema Indexes Validation

Filter {gender: 'f'}

Generate query Explain Reset Find

ADD DATA EXPORT DATA UPDATE DELETE 1 - 2 of 2

```
{
  "_id": ObjectId("65bbd2fa9e77200d17c64942"),
  "id": 2,
  "name": "Светлана",
  "age": 25,
  "gender": "f"
}
```

```
{
  "_id": ObjectId("65bbd2fa9e77200d17c64944"),
  "id": 4,
  "name": "Ольга",
  "age": 22,
  "gender": "f"
}
```

Либо в mongosh :

Unset

```
db.users_NAME.find({gender: 'f'})
```

> MONGOSH

```
> db.users_Andrew.find({gender: 'f'})
< {
  _id: ObjectId('65bbd2fa9e77200d17c64942'),
  id: 2,
  name: 'Светлана',
  age: 25,
  gender: 'f'
}
{
  _id: ObjectId('65bbd2fa9e77200d17c64944'),
```

2. Вывести данные о всех представительницах женского пола и людях старше 30 лет



```
Unset
db.users_NAME.find({
  $or: [
    {gender: 'f'},
    {age: {$gt: 30}},
  ]
})
```

3. Вывести всех пользователей, у которых имя входит в список (Анатолий, Дмитрий, Ольга, Семен)

```
Unset
db.users_NAME.find({
  name: { $in: [ 'Анатолий', 'Дмитрий', 'Ольга', 'Семен' ] }
})
```

4. Вывести пользователя с идентификатором 3

```
Unset
db.users_NAME.find({ id: 3})
```

5. Вывести пользователей в возрасте от 30 до 40 (включая концы)

```
Unset
db.users_NAME.find({
  age: {$gte: 30, $lte: 40}
})
```



УДАЛЕНИЕ ДОКУМЕНТОВ И ВНЕСЕНИЕ ИЗМЕНЕНИЙ

В compass можно быстро удалить или отредактировать один документ , просто наведя на него курсор.

<pre> _id: ObjectId('6581fc95a66dd296b4ad46c3') id: 1 name: "Анатолий" age: 28 gender: "m" </pre>
<pre> _id: ObjectId('6581fc95a66dd296b4ad46c4') id: 2 name: "Светлана" age: 25 gender: "f" </pre>
<pre> _id: ObjectId('6581fc95a66dd296b4ad46c5') </pre>

Удаление множества документов в mongosh

Unset

```
db.users_NAME.deleteMany({gender: "m"})
```

```

> db.users_Ndrew.deleteMany({gender: "m"})
{
  acknowledged: true,
  deletedCount: 2
}

```

Редактирование одного документа выполняем в compass или множества документов в mongosh, при помощи updateMany

Пример - В коллекции users_NAME добавим к возрасту каждого 3 года, используя функцию \$inc



Unset

```
db.users_Ndrew.updateMany( {"age" : {$exists: true}}, { $inc: { age: 3 } })
```

```
> db.users_Andrew.updateMany( {"age" : {$exists: true}}, { $inc: { age: 3 } })
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}
ich_edit>
```

Проверяем : Обращаем внимание на изменившийся возраст:

Filter	Type a query: { field: 'value' } or Generate query	Explain	Reset	Find
ADD DATA	EXPORT DATA	UPDATE	DELETE	1 - 2 of 2
<pre>_id: ObjectId('65bbd2fa9e77200d17c64942') id: 2 name: "Светлана" age: 28 gender: "f"</pre>				
<pre>_id: ObjectId('65bbd2fa9e77200d17c64944') id: 4 name: "Ольга" age: 25 gender: "f"</pre>				



ЭКСПОРТ И ИМПОРТ ДАННЫХ

Из коллекции movies выберите названия и год выпуска фильмов. Отсортируйте по дате выхода (по возрастанию), не показывая идентификатор документа.

```
Unset
{
  sort: {
    year: 1
  },
  project: {
    _id: 0,
    title: 1,
    year: 1
  }
}
```

Filter ⓘ ⓘ {}

Project {}_id: 0,title: 1,year: 1}

Sort {year: 1}

Collation { locale: 'simple' }

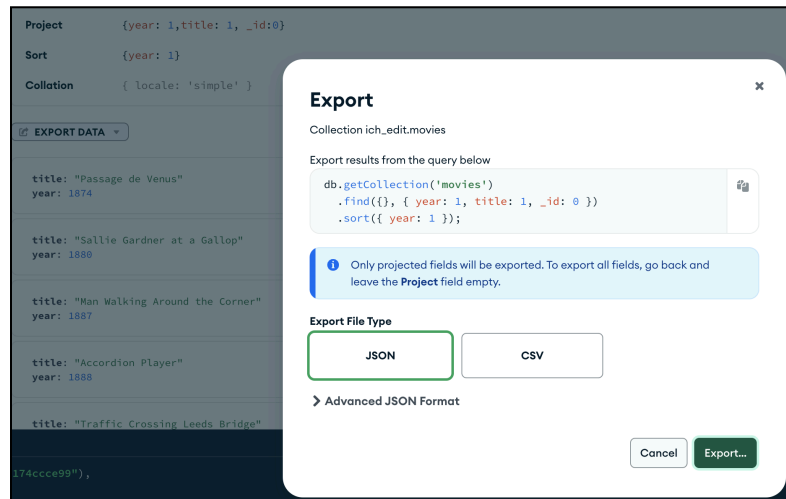
EXPORT DATA

title : "Passage de Venus"
year : 1874

title : "Sallie Gardner at a Gallop"
year : 1880

title : "Man Walking Around the Corner"
year : 1887

Экспортируем данные, полученные в этом запросе как json, нажав на кнопку EXPORT DATA



Сохраняем полученный файл у себя на жестком диске.

Импортируем полученные данные в новую коллекцию `movies_NAME`, где NAME - ваше имя. Создаем коллекцию, нажимаем ADD DATA и выбираем файл

