

УРОК 15.2. MONGODB COMPASS AND MONGOSH. ФОРМИРОВАНИЕ ВЫБОРКИ: ЗНАКОМСТВО, ПРОЕКТИРОВАНИЕ БД

РАБОТА С MONGOSH (MONGO SHELL)	2
ОПЕРАТОРЫ СРАВНЕНИЯ	3
ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ	4
ЛОГИЧЕСКИЕ ОПЕРАТОРЫ	5
ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ	6
ПАГИНАЦИЯ И СОРТИРОВКА	8



РАБОТА С MONGOSH (MONGO SHELL)

Команда Find - Чтобы извлечь все документы из коллекции, мы можем использовать следующую команду:

Unset

```
db.customers.find()
```

Если нам надо получить не все документы, а только те, которые удовлетворяют определенному требованию, необходимо в find() указать условие. Например, выведем все документы, у которых city="Berlin":

Unset

```
db.customers.find({City: "Berlin"})
```



ОПЕРАТОРЫ СРАВНЕНИЯ

- $\$eq$: значения равны
- $\$ne$: значения не равны
- $\$gt$: значение больше другого значения
- $\$gte$: значение больше или равно другому значению
- $\$lt$: значение меньше другого значения
- $\$lte$: значение меньше или равно другому значению
- $\$in$: значение соответствует одному из значений в массиве



ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

1. Вернуть пользователей, чей возраст больше или равен 30

Unset

```
db.US_Adult_Income.find({age: {$gte: 30}})
```

2. Вернуть пользователей, чей возраст меньше 30 лет

Unset

```
db.US_Adult_Income.find({age: {$lt: 30}})
```

3. Вернуть пользователей, чей возраст равен 22 или 32

Unset

```
db.US_Adult_Income.find({age: {$in : [22, 32]}})
```

4. Вернуть всех пользователей, ID которых не ANTON

Unset

```
db.customers.find({ CustomerID: {$not: {$eq: "ANTON"}}})
```

5. Подсчитать количество пользователей, ID которых не ANTON

Unset

```
db.customers.countDocuments({ CustomerID: {$not: {$eq: "ANTON"}}})
```



ЛОГИЧЕСКИЕ ОПЕРАТОРЫ

- `$and` : возвращает документы, в которых выполняются оба условия
- `$or` : возвращает документы, в которых выполняется хотя бы одно условие
- `$nor` : возвращает документы, в которых оба условия не выполняются
- `$not` : возвращает документы, в которых условие не выполняется



ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

1. Вернуть пользователей, чей возраст находится в диапазоне от 22 до 30 лет

```
Unset
db.US_Adult_Income.find(
  {$and: [
    {age: {$gte: 22}},
    {age: {$lte: 30}}
  ]
}
```

2. Вернуть то же, что и прошлый запрос, используя короткий аналог указания двух условий

```
Unset
db.US_Adult_Income.find({
  age: {$gte: 22, $lte: 30}
})
```

3. Вернуть тех, кто отвечает хотя бы одному из условий

```
Unset
db.US_Adult_Income.find(
  {$or: [
```



```
{education: " Bachelors"},  
  
{age: {$lte: 30}}  
  
]  
  
}  
  
)
```



ПАГИНАЦИЯ И СОРТИРОВКА



Функция `limit()` задает максимально допустимое количество получаемых документов.

Количество передается в виде числового параметра.

Вывести три первых документа из коллекции:

```
Unset
db.US_Adult_Income.find().limit(3)
```



Если мы хотим произвести выборку не сначала, а пропустив какое-то количество документов, то воспользуемся функцией `skip()`.

Вывести все документы коллекции, кроме первых трех:

```
Unset
db.US_Adult_Income.find().skip(3)
```

Функции `limit()` и `skip()` можно комбинировать:

```
Unset
db.US_Adult_Income.find().skip(3).limit(3)
```



Оператор `$slice` является в некотором роде комбинацией функций `limit` и `skip`.

Но в отличие от них `$slice` может работать с массивами.



В выборку попадет только один язык из массива languages, а не весь массив:

Unset

```
db.US_Adult_Income.find ({CustomerID: "Anton"}, {languages: {$slice : 1}})
```



Чтобы отсортировать полученную выборку, необходимо воспользоваться функцией `sort()`.

Передавая в функцию значения 1 или -1, мы можем указать в каком порядке сортировать: по возрастанию (1) или по убыванию (-1).

Unset

```
db.US_Adult_Income.find().sort({age: 1})
```