

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Лабораторная работа № 4
по дисциплине «Постреляционные базы данных»

Тема: «Создание документной базы данных и работа с ней на примере СУБД
MongoDb»

ИСПОЛНИТЕЛЬ:
группа ИУ5-24М

Журавлев Н.В.
ФИО

подпись

"20" марта 2024 г.

ПРЕПОДАВАТЕЛЬ:

Виноградова М.В.
ФИО

подпись

" " _____ 202_ г.

Москва - 2024

Цель работы

- Изучить модель данных и способы работы с документными БД NoSql.
- Освоить методы создания документной БД и языки запросов к ним.
- Получить навыки работы с документной БД MongoDB.

Задание

1. Создать в среде MongoDB свою БД по теме, выданной преподавателем, содержащую не менее трех коллекций. Добавить в коллекции БД объекты сложной структуры, содержащие вложенные структуры и коллекции. Продемонстрировать (вывести на экран) содержимое коллекций.
2. Продемонстрировать изменение объектов БД:
 - добавление элементов объекта
 - изменение элементов объекта
 - удаление элементов объекта
 - замена всего объекта
 - удаление объекта.
3. Выполнить запросы к базе данных:
 - вывод всех элементов коллекции
 - вывод с фильтрацией (условия И, ИЛИ)
 - проекция вывода (вывод части полей)
 - сортировка
 - удаление дубликатов
 - условия на поля вложенных структур
 - поиск по вложенным коллекциям объекта.
4. Выполнить запросы к базе данных:
 - проверка на наличие и отсутствие полей в документе
 - ограничение на количество документов в результате

- с операторами сравнения
- работа с массивами
- с группировкой и агрегированием

Ход работы

Создать в среде MongoDB свою БД и 3 коллекции:

```
mydb> db.createCollection("account")
{ ok: 1 }
mydb> db.createCollection("accountHistory")
{ ok: 1 }
mydb> db.createCollection("diet")
{ ok: 1 }
```

Рисунок 1

Добавить в коллекции БД объекты сложной структуры, содержащие вложенные структуры и коллекции:

```
db.account.updateMany({}, {$set: {"history_diet": [{name: "Название",
description: "desx"},{name: "name", description: "desc"}]}})
```

Вывод на экран содержимое коллекций:

```
db.account.find({"age": {$eq:18}}).pretty()
```

```
test> db.account.find({"age": {$eq:18}}).pretty()
[
  {
    _id: ObjectId('65fd8998a25a3d9487c00e88'),
    login: 'login',
    password: 'password',
    weight: 80,
    height: 180,
    age: 18,
    special: 'здоровый',
    history_diet: [
      { name: 'Название', description: 'desx' },
      { name: 'name', description: 'desc' }
    ],
    history_weight: []
  },
  {
    _id: ObjectId('65fd8998a25a3d9487c00e89'),
    login: 'l',
    password: 'password',
    weight: 80,
    height: 180,
    age: 18,
    special: 'здоровый',
    history_diet: [
      { name: 'Название', description: 'desx' },
      { name: 'name', description: 'desc' }
    ],
    history_weight: []
  },
  {
    _id: ObjectId('65fd8998a25a3d9487c00e8a'),
    login: 'login',
    password: 'password',
    weight: 80,
```

Рисунок 2

Добавление элементов объекта:

```
db.account.update({"age": {$eq:17}}, {$set: {"new_field": "value"}})
```

Для изменения элементов объекта используется команда:

```
db.account.update({"age": {$eq:18}}, {$set: {"password": "pass"}})
```

Для удаления элементов объекта используется команда:

```
db.account.update({"age": {$eq:17}}, {$unset: {"new_field": ""}})
```

Для замены всего объекта используется команда:

```
db.account.update({"age": {$eq:18}}, {$set: {"login": "logi", "password":  
"passwod", "weight":70, "height":170, "age":17, "special":"здоровый"}})
```

Для удаления объекта используется команда:

```
db.account.remove({"age": {$eq:18}}, true)
```

Вывод всех элементов коллекции используется команда:

```
db.account.find().pretty()
```

Вывод с фильтрацией (условия И, ИЛИ) используются команды:

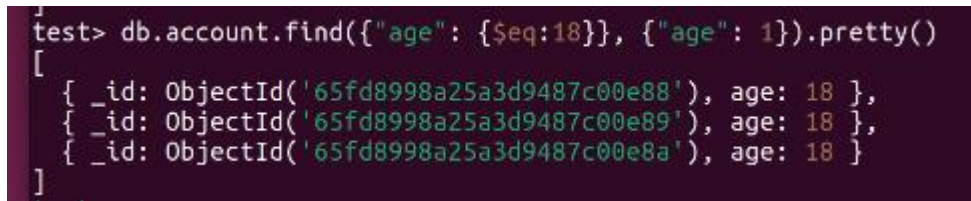
```
db.account.find({$and: [{"age": 18}, {"login": "l"}]}).pretty()
```

```
db.account.find({$or: [{"age": 17}, {"login": "l"}]}).pretty()
```

```
test> db.account.find({$or: [{"age": 17}, {"login": "l"}]}).pretty()  
[  
  {  
    _id: ObjectId('65fd88f5a25a3d9487c00e86'),  
    login: 'log',  
    password: 'passwod',  
    weight: 70,  
    height: 170,  
    age: 17,  
    special: 'здоровый',  
    history_diet: [  
      { name: 'Единственное название', description: 'desx' },  
      { name: 'name', description: 'desc' }  
    ],  
    history_weight: [ 180 ]  
  },  
  {  
    _id: ObjectId('65fd8998a25a3d9487c00e89'),  
    login: 'l',  
    password: 'password',  
    weight: 80,  
    height: 180,  
    age: 18,  
    special: 'здоровый',  
    history_diet: [  
      { name: 'Название', description: 'desx' },  
      { name: 'name', description: 'desc' }  
    ],  
    history_weight: []  
  }  
]
```

Вывод проекции вывода (вывод части полей) для этого используется команда:

```
db.account.find({"age": {$eq:18}}, {"age": 1}).pretty()
```



```
test> db.account.find({"age": {$eq:18}}, {"age": 1}).pretty()
[
  { _id: ObjectId('65fd8998a25a3d9487c00e88'), age: 18 },
  { _id: ObjectId('65fd8998a25a3d9487c00e89'), age: 18 },
  { _id: ObjectId('65fd8998a25a3d9487c00e8a'), age: 18 }
]
```

Рисунок 4

Для вывода с сортировкой используется команда:

```
db.account.find().sort({age: -1}).pretty()
```



```
test> db.account.find().sort({age: -1}).pretty()
[
  {
    _id: ObjectId('65fd8a38a25a3d9487c00e8b'),
    login: 'log2',
    password: 'password',
    weight: 90,
    height: 190,
    age: 19,
    special: 'здоровый',
    history_diet: [
      { name: 'Название', description: 'desx' },
      { name: 'name', description: 'desc' }
    ],
    history_weight: [ 170, 181 ]
  },
  {
    _id: ObjectId('65fd8998a25a3d9487c00e88'),
    login: 'login',
    password: 'password',
    weight: 80,
    height: 180,
    age: 18,
    special: 'здоровый',
    history_diet: [
      { name: 'Название', description: 'desx' },
      { name: 'name', description: 'desc' }
    ],
    history_weight: []
  },
  {
    _id: ObjectId('65fd8998a25a3d9487c00e89'),
    login: 'l',
    password: 'password',
    weight: 70,
    height: 170,
    age: 17,
    special: 'здоровый',
    history_diet: [
      { name: 'Название', description: 'desx' },
      { name: 'name', description: 'desc' }
    ],
    history_weight: []
  }
]
```

Рисунок 5

Для удаления дубликатов используется команда:

```
db.account.distinct("age")
```

Условия на поля вложенных структур и поиск по вложенным коллекциям объекта:

```
db.account.find({"history_diet.name": "Единственное название")).pretty()
```

Проверка на наличие и отсутствие полей в документе:

```
db.account.find({"age": {$exists:true}}).pretty()
```

```
db.account.find({"not_exists": {$exists:true}}).pretty()
```

```
test> db.account.find({"not_exists": {$exists:true}}).pretty()
test> db.account.find({"age": {$exists:true}}).pretty()
[
  {
    _id: ObjectId('65fd88f5a25a3d9487c00e86'),
    login: 'log',
    password: 'passwod',
    weight: 70,
    height: 170,
    age: 17,
    special: 'здоровый',
    history_diet: [
      { name: 'Единственное название', description: 'desx' },
      { name: 'name', description: 'desc' }
    ],
    history_weight: [ 180 ]
  },
  {
    _id: ObjectId('65fd8998a25a3d9487c00e88'),
    login: 'login',
    password: 'password',
    weight: 80,
    height: 180,
    age: 18,
    special: 'здоровый',
    history_diet: [
      { name: 'Название', description: 'desx' },
      { name: 'name', description: 'desc' }
    ],
    history_weight: [ ]
  }
]
```

Рисунок 6

Для ограничения на количество документов в результате используется команда:

```
db.account.find().limit(2).pretty()
```

```
test> db.account.find().limit(2).pretty()
[
  {
    _id: ObjectId('65fd88f5a25a3d9487c00e86'),
    login: 'log',
    password: 'passwod',
    weight: 70,
    height: 170,
    age: 17,
    special: 'здоровый',
    history_diet: [
      { name: 'Единственное название', description: 'desx' },
      { name: 'name', description: 'desc' }
    ],
    history_weight: [ 180 ]
  },
  {
    _id: ObjectId('65fd8998a25a3d9487c00e88'),
    login: 'login',
    password: 'password',
    weight: 80,
    height: 180,
    age: 18,
    special: 'здоровый',
    history_diet: [
      { name: 'Название', description: 'desx' },
      { name: 'name', description: 'desc' }
    ],
    history_weight: [ ]
  }
]
```

Рисунок 7

Команды для работы с массивами:

```
db.account.updateMany({}, {$set: {"history_weight": []}})
```

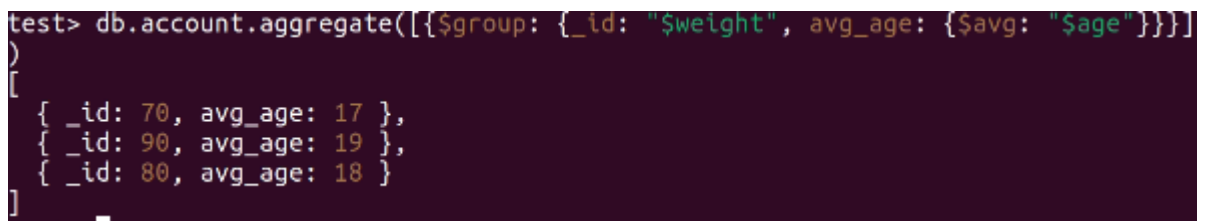
```

db.account.update({"login": "log"}, {$push: {"history_weight":180}})
db.account.find({"age": {$eq:17}}).pretty()
db.account.find({"history_weight": {$in: [180]}}).pretty()
db.account.find({"history_weight": {$all: [180, 170, 160]}}).pretty()
db.account.updateOne({"login": "log2"}, {$pull: {"history_weight": 180}})
db.account.update({"login":"log2"}, {$set: {"history_weight.1": 181}})

Команды выходы с группировкой и агрегированием:
db.account.aggregate([{$group: {_id: "$weight", avg_age: {$avg:

```

```
"$age"}}]])
```



```

test> db.account.aggregate([{$group: {_id: "$weight", avg_age: {$avg: "$age"}}}])
[
  { _id: 70, avg_age: 17 },
  { _id: 90, avg_age: 19 },
  { _id: 80, avg_age: 18 }
]

```

Рисунок 8

Вывод

В результате выполнения работы были изучены модель данных и способы работы с документными БД NoSql. Так же освоить методы создания документной БД и языки запросов к ним. По итогу получить навыки работы с документной БД MongoDB.

Список используемой литературы

1. Фаулер, Мартин, Садаладж, Прамодкумар Дж. NoSQL: новая методология разработки нереляционных баз данных. : Пер. с англ. - М.: ООО "И.Д. Вильямс", 2013г.
2. Что такое документная база данных? – Текст. Изображение : электронные //Сервисы облачных вычислений – Amazon Web Services (AWS): [сайт]. – URL:<https://aws.amazon.com/ru/nosql/document/> (дата обращения: 11.05.2022)
3. 11 типов современных баз данных: краткие описания, схемы и примеры БД. – Текст. Изображение : электронные // Библиотека

- программиста: [сайт]. – URL: <https://proglib.io/index.php/p/11-tipov-sovremennyh-baz-dannyh-kratkie-opisaniyashemy-i-primery-bd-2020-01-07> (дата обращения: 11.05.2022)
4. MongoDB | Устройство базы данных. Документы. – Текст. Изображение : электронные // METANIT.COM Сайт о программировании: [сайт]. – URL: <https://metanit.com/nosql/mongodb/2.1.php> (дата обращения: 25.04.2022).
 5. MongoDB | Введение. – Текст. Изображение : электронные // METANIT.COM Сайт о программировании: [сайт]. – URL: <https://metanit.com/nosql/mongodb/1.1.php> (дата обращения: 11.05.2022)
 6. Ответы на вопросы на собеседование MongoDB. | Вопросы на собеседование Junior Java Developer. – Текст. Изображение : электронные // Вопросы на собеседование Junior Java Developer: [сайт]. – URL: <https://jsehelper.blogspot.ru/2016/05/mongodb.html> (дата обращения: 11.05.2022)
 7. MongoDB – Краткое руководство. – Текст. Изображение : электронные // Уроки и статьи по программированию и IT: [сайт]. – URL: <https://coderlessons.com/tutorials/bazy-dannykh/uchitsia-mongodb/mongodb-kratkoerukovodstvo> (дата обращения: 11.05.2022)
 8. Introduction to MongoDB – MongoDB Manual. – Текст. Изображение : электронные // MongoDB: The Application Data Platform | MongoDB: [сайт]. – URL: <https://docs.mongodb.com/manual/introduction/> (дата обращения: 11.05.2022)
 9. Data Modeling Introduction – MongoDB Manual. – Текст. Изображение : электронные // MongoDB: The Application Data Platform | MongoDB: [сайт]. – URL: <https://docs.mongodb.com/manual/core/data-modeling-introduction/> (дата обращения: 11.05.2022)
 10. Официальный сайт MongoDB. – Текст. Изображение : электронные // MongoDB: The Application Data Platform | MongoDB: [сайт]. – URL: <https://www.mongodb.com/> (дата обращения: 25.04.2022)