Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет инфокоммуникационных технологий

**ОТЧЕТ**

**О ЛАБОРАТОРНОЙ РАБОТЕ № 8**

по теме: Работа с БД в СУБД MongoDB

по дисциплине: Проектирование и реализация баз данных

**Специальность**:

09.03.03 Мобильные и сетевые технологии

|  |  |
| --- | --- |
| **Проверил**:  Говорова М.М. \_\_\_\_\_\_\_\_\_\_  **Дата**: «\_\_» \_\_\_\_ 2021 г.  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Выполнил**:  студент группы K3240  Рейнгеверц В.А. |

Санкт-Петербург 2021 г.

**ЦЕЛЬ РАБОТЫ**

Овладеть практическими навыками работы с CRUD-операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, со ссылками и индексами в базе данных MongoDB.

**ПРАКТИЧЕСКОЕ ЗАДАНИЕ**

**Практическое задание 8.1.1:**

1. *Создайте базу данных learn.*
2. *Заполните коллекцию единорогов unicorns:*

db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});

db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});

db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});

db.unicorns.insert({name: 'Roooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99});

db.unicorns.insert({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});

db.unicorns.insert({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});

db.unicorns.insert({name:'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});

db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});

db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});

db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});

db.unicorns.insert({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});

1. *Используя второй способ, вставьте в коллекцию единорогов документ:*

{name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165}

1. *Проверьте содержимое коллекции с помощью метода find.*

**Практическое задание 8.1.2:**

1. *Сформируйте запросы для вывода списков самцов и самок единорогов. Ограничьте список самок первыми тремя особями. Отсортируйте списки по имени.*
2. *Найдите всех самок, которые любят carrot. Ограничьте этот список первой особью с помощью функций findOne и limit.*

**Практическое задание 8.1.3:**

*Модифицируйте запрос для вывода списков самцов единорогов, исключив из результата информацию о предпочтениях и поле.*

**Практическое задание 8.1.4:**

*Вывести список единорогов в обратном порядке добавления.*

**Практическое задание 8.1.5:**

*Вывести список единорогов с названием первого любимого предпочтения, исключив идентификатор.*

**Практическое задание 8.1.6:**

*Вывести список самок единорогов весом от полутонны до 700 кг, исключив вывод идентификатора.*

**Практическое задание 8.1.7:**

*Вывести список самцов единорогов весом от полутонны и предпочитающих grape и lemon, исключив вывод идентификатора.*

**Практическое задание 8.1.8:**

*Найти всех единорогов, не имеющих ключ vampires.*

**Практическое задание 8.1.9:**

*Вывести список упорядоченный список имен самцов единорогов с информацией об их первом предпочтении.*

**Практическое задание 8.2.1:**

1. *Создайте коллекцию towns, включающую следующие документы:*

{name: "Punxsutawney ",

populatiuon: 6200,

last\_sensus: ISODate("2008-01-31"),

famous\_for: [""],

mayor: {

name: "Jim Wehrle"

}}

{name: "New York",

populatiuon: 22200000,

last\_sensus: ISODate("2009-07-31"),

famous\_for: ["status of liberty", "food"],

mayor: {

name: "Michael Bloomberg",

party: "I"}}

{name: "Portland",

populatiuon: 528000,

last\_sensus: ISODate("2009-07-20"),

famous\_for: ["beer", "food"],

mayor: {

name: "Sam Adams",

party: "D"}}

1. *Сформировать запрос, который возвращает список городов с независимыми мэрами (*party="I")*. Вывести только название города и информацию о мэре.*
2. *Сформировать запрос, который возвращает список беспартийных мэров (*party *отсутствует*)*. Вывести только название города и информацию о мэре.*

**Практическое задание 8.2.2:**

1. *Сформировать функцию для вывода списка самцов единорогов.*
2. *Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.*
3. *Вывести результат, используя forEach.*
4. *Содержание коллекции единорогов unicorns:*

db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});

db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});

db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});

db.unicorns.insert({name: 'Roooooodles', 44), loves: ['apple'], weight: 575, gender: 'm', vampires: 99});

db.unicorns.insert({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});

db.unicorns.insert({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});

db.unicorns.insert({name:'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});

db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});

db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});

db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});

db.unicorns.insert ({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});

db.unicorns.insert ({name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165})

**Практическое задание 8.2.3:**

*Вывести количество самок единорогов весом от полутонны до 600 кг.*

**Практическое задание 8.2.4:**

*Вывести список предпочтений.*

**Практическое задание 8.2.5:**

*Посчитать количество особей единорогов обоих полов*

**Практическое задание 8.2.6:**

1. *Выполнить команду:*

> db.unicorns.save({name: 'Barny', loves: ['grape'],

weight: 340, gender: 'm'})

1. *Проверить содержимое коллекции unicorns.*

**Практическое задание 8.2.7:**

1. *Для самки единорога* Ayna *внести изменения в БД: теперь ее вес 800, она убила 51 вапмира.*
2. *Проверить содержимое коллекции unicorns.*

**Практическое задание 8.2.8:**

1. *Для самца единорога* Raleigh *внести изменения в БД: теперь он любит рэдбул.*
2. *Проверить содержимое коллекции unicorns.*

**Практическое задание 8.2.9:**

1. *Всем самцам единорогов увеличить количество убитых вапмиров на 5.*
2. *Проверить содержимое коллекции unicorns.*

**Практическое задание 8.2.10:**

1. *Изменить информацию о городе Портланд: мэр этого города теперь беспартийный.*
2. *Проверить содержимое коллекции towns.*

**Практическое задание 8.2.11:**

1. *Изменить информацию о самце единорога Pilot: теперь он любит и шоколад.*
2. *Проверить содержимое коллекции unicorns.*

**Практическое задание 8.2.12:**

1. *Изменить информацию о самке единорога* Aurora: *теперь она любит еще и сахар, и лимоны.*
2. *Проверить содержимое коллекции unicorns.*

**Практическое задание 8.2.13:**

1. *Создайте коллекцию towns, включающую следующие документы:*

{name: "Punxsutawney ",

popujatiuon: 6200,

last\_sensus: ISODate("2008-01-31"),

famous\_for: ["phil the groundhog"],

mayor: {

name: "Jim Wehrle"

}}

{name: "New York",

popujatiuon: 22200000,

last\_sensus: ISODate("2009-07-31"),

famous\_for: ["status of liberty", "food"],

mayor: {

name: "Michael Bloomberg",

party: "I"}}

{name: "Portland",

popujatiuon: 528000,

last\_sensus: ISODate("2009-07-20"),

famous\_for: ["beer", "food"],

mayor: {

name: "Sam Adams",

party: "D"}}

1. *Удалите документы с беспартийными мэрами.*
2. *Проверьте содержание коллекции.*
3. *Очистите коллекцию.*
4. *Просмотрите список доступных коллекций.*

**Практическое задание 8.3.1:**

1. *Создайте коллекцию зон обитания единорогов, указав в качестве идентификатора кратко название зоны, далее включив полное название и описание.*
2. *Включите для нескольких единорогов в документы ссылку на зону обитания, использую второй способ автоматического связывания.*
3. *Проверьте содержание коллекции едиорогов.*
4. *Содержание коллекции единорогов unicorns:*

db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});

db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});

db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});

db.unicorns.insert({name: 'Roooooodles', 44), loves: ['apple'], weight: 575, gender: 'm', vampires: 99});

db.unicorns.insert({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});

db.unicorns.insert({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});

db.unicorns.insert({name:'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});

db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});

db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});

db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});

db.unicorns.insert ({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});

db.unicorns.insert {name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165}

**Практическое задание 8.3.2:**

1. *Проверьте, можно ли задать для коллекции unicorns индекс для ключа name с флагом* unique*.*
2. *Содержание коллекции единорогов unicorns:*

db.unicorns.insert({name: 'Horny', dob: new Date(1992,2,13,7,47), loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});

db.unicorns.insert({name: 'Aurora', dob: new Date(1991, 0, 24, 13, 0), loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});

db.unicorns.insert({name: 'Unicrom', dob: new Date(1973, 1, 9, 22, 10), loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});

db.unicorns.insert({name: 'Roooooodles', dob: new Date(1979, 7, 18, 18, 44), loves: ['apple'], weight: 575, gender: 'm', vampires: 99});

db.unicorns.insert({name: 'Solnara', dob: new Date(1985, 6, 4, 2, 1), loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});

db.unicorns.insert({name:'Ayna', dob: new Date(1998, 2, 7, 8, 30), loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});

db.unicorns.insert({name:'Kenny', dob: new Date(1997, 6, 1, 10, 42), loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});

db.unicorns.insert({name: 'Raleigh', dob: new Date(2005, 4, 3, 0, 57), loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});

db.unicorns.insert({name: 'Leia', dob: new Date(2001, 9, 8, 14, 53), loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});

db.unicorns.insert({name: 'Pilot', dob: new Date(1997, 2, 1, 5, 3), loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});

db.unicorns.insert ({name: 'Nimue', dob: new Date(1999, 11, 20, 16, 15), loves: ['grape', 'carrot'], weight: 540, gender: 'f'});

db.unicorns.insert {name: 'Dunx', dob: new Date(1976, 6, 18, 18, 18), loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 16

**Практическое задание 8.3.3:**

1. *Получите информацию о всех индексах коллекции unicorns .*
2. *Удалите все индексы, кроме индекса для идентификатора.*
3. *Попытайтесь удалить индекс для идентификатора*

**Практическое задание 8.3.4:**

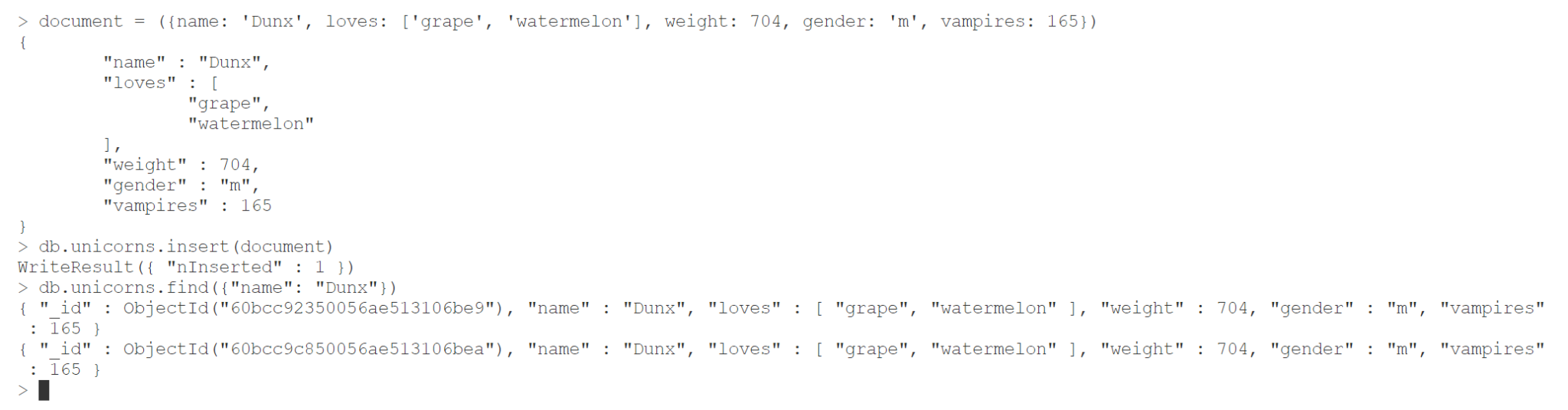
1. *Создайте объемную коллекцию numbers, задействовав курсор:*

for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}

1. *Выберите последних четыре документа.*
2. *Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра executionTimeMillis)*
3. *Создайте индекс для ключа value.*
4. *Получите информацию о всех индексах коллекции numbres.*
5. *Выполните запрос 2.*
6. *Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?*
7. *Сравните время выполнения запросов с индексом и без. Дайте ответ на вопрос: какой запрос более эффективен?*

**ВЫПОЛНЕНИЕ**

**Практическое задание 8.1.1:**

****

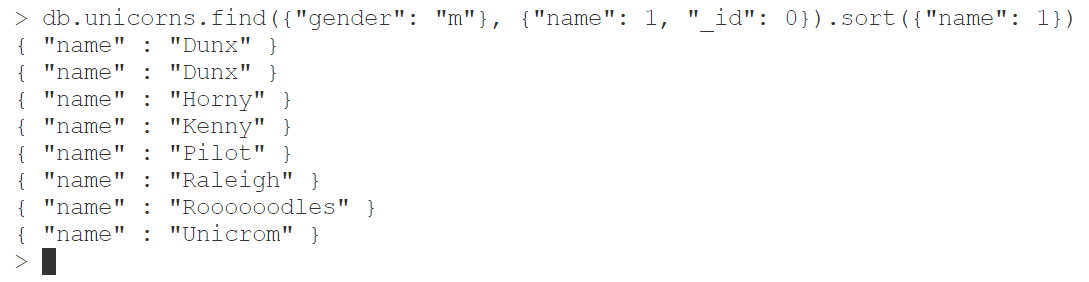
(рис. 1 - Поиск)



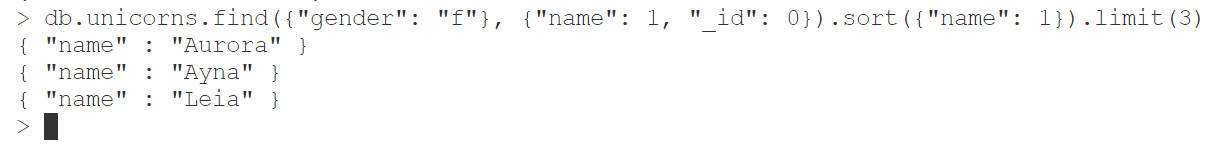
(рис. 2 - Содержимое коллекции)

**Практическое задание 8.1.2:**

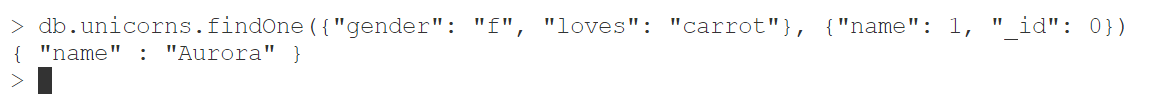
1. *Сформируйте запросы для вывода списков самцов и самок единорогов. Ограничьте список самок первыми тремя особями. Отсортируйте списки по имени.*
2. *Найдите всех самок, которые любят carrot. Ограничьте этот список первой особью с помощью функций findOne и limit.*

****

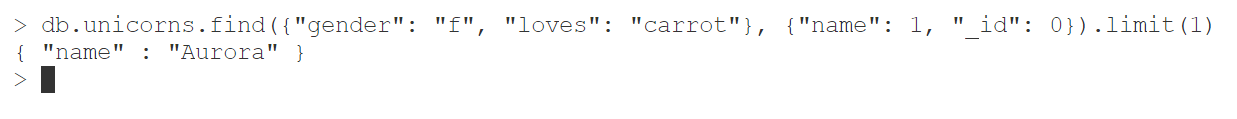
(рис. 3 - Поиск самцов)



(рис. 4 - Поиск самок)



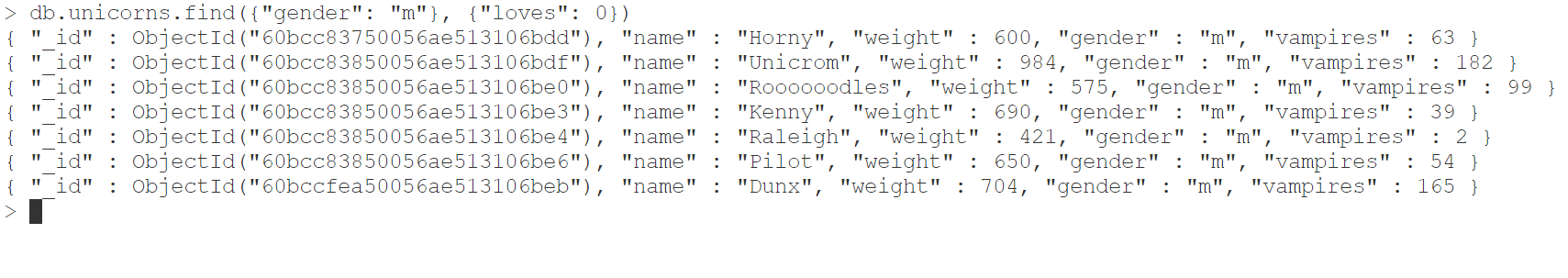
(рис. 5 - Использована функция *findOne*)



(рис. 6 - Использована функция *limit*)

**Практическое задание 8.1.3:**

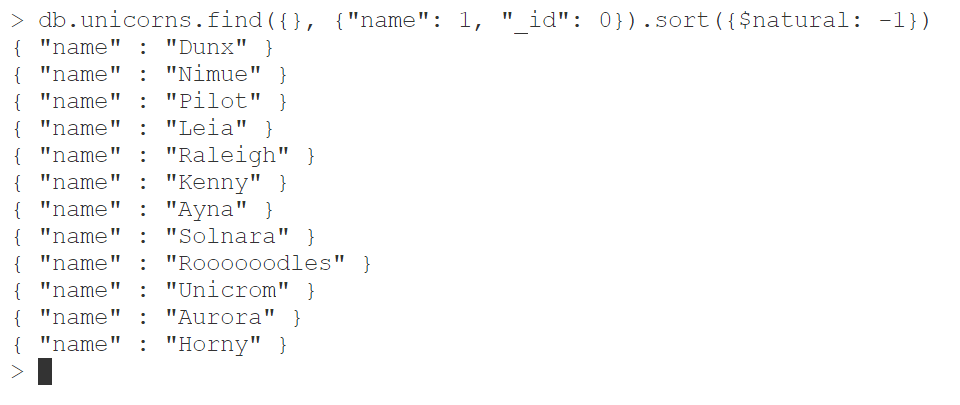
*Модифицируйте запрос для вывода списков самцов единорогов, исключив из результата информацию о предпочтениях и поле.*



(рис. 7 - Самцы без предпочтений)

**Практическое задание 8.1.4:**

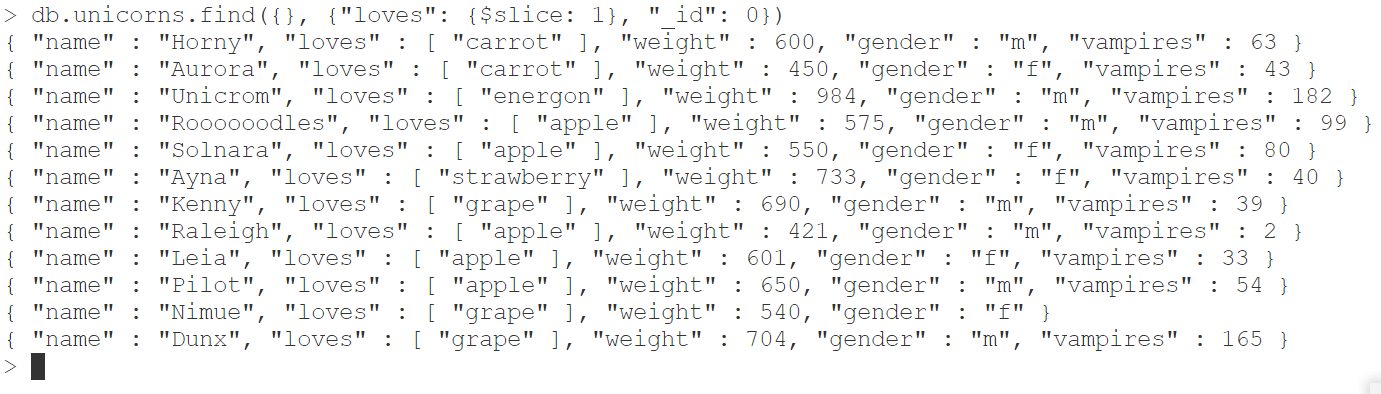
*Вывести список единорогов в обратном порядке добавления.*

****

(рис. 8 - Обратный порядок добавления)

**Практическое задание 8.1.5:**

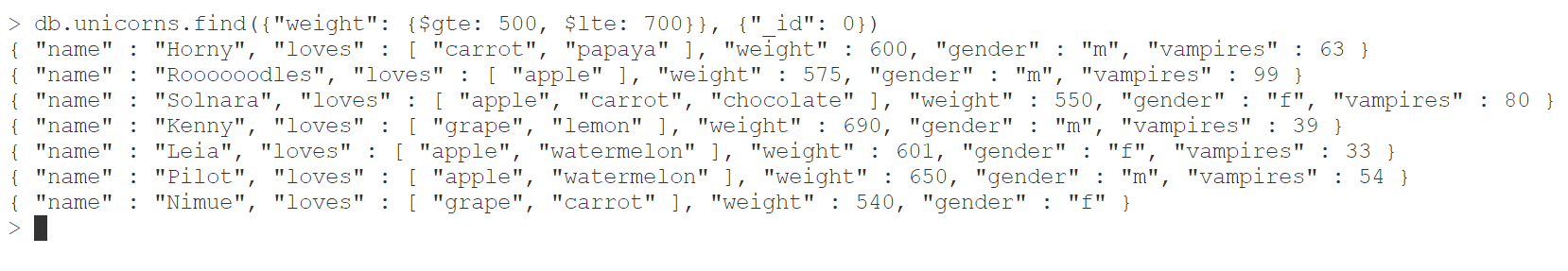
*Вывести список единорогов с названием первого любимого предпочтения, исключив идентификатор.*



(рис. 9 - Первое предпочтение)

**Практическое задание 8.1.6:**

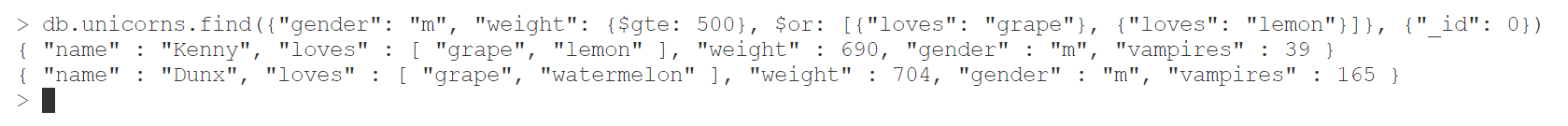
*Вывести список самок единорогов весом от полутонны до 700 кг, исключив вывод идентификатора.*



(рис. 10 - Вес 500-700 кг)

**Практическое задание 8.1.7:**

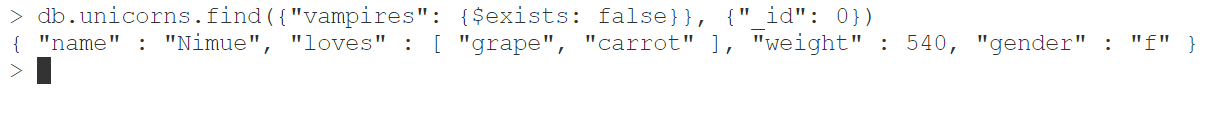
*Вывести список самцов единорогов весом от полутонны и предпочитающих grape и lemon, исключив вывод идентификатора.*

**

(рис. 11 - Самцы 500+, grape, lemon)

**Практическое задание 8.1.8:**

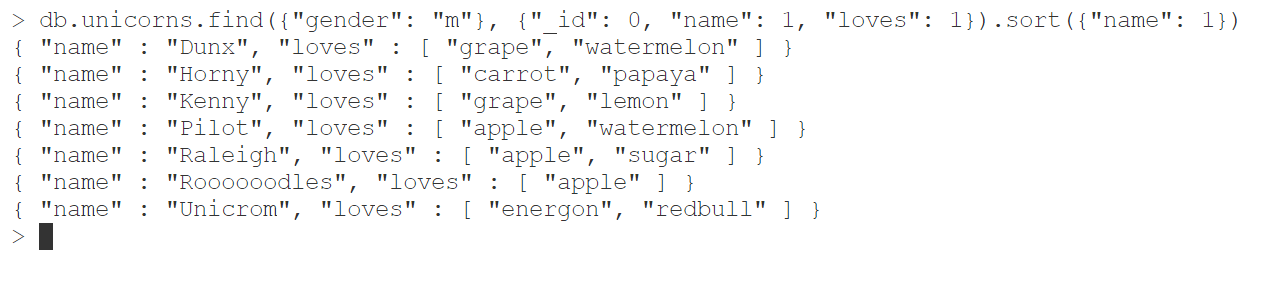
*Найти всех единорогов, не имеющих ключ vampires.*

****

(рис. 12 - Не vampires)

**Практическое задание 8.1.9:**

*Вывести список упорядоченный список имен самцов единорогов с информацией об их первом предпочтении.*

****

(рис. 13 - Предпочтения самцов)

**Практическое задание 8.2.1:**

1. *Создайте коллекцию towns, включающую следующие документы:*

{name: "Punxsutawney ",

populatiuon: 6200,

last\_sensus: ISODate("2008-01-31"),

famous\_for: [""],

mayor: {

name: "Jim Wehrle"

}}

{name: "New York",

populatiuon: 22200000,

last\_sensus: ISODate("2009-07-31"),

famous\_for: ["status of liberty", "food"],

mayor: {

name: "Michael Bloomberg",

party: "I"}}

{name: "Portland",

populatiuon: 528000,

last\_sensus: ISODate("2009-07-20"),

famous\_for: ["beer", "food"],

mayor: {

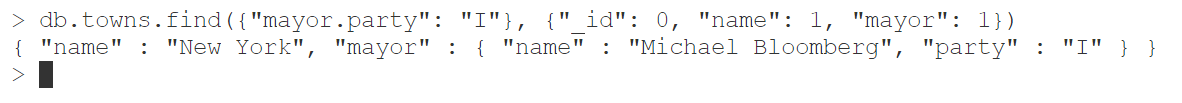
name: "Sam Adams",

party: "D"}}

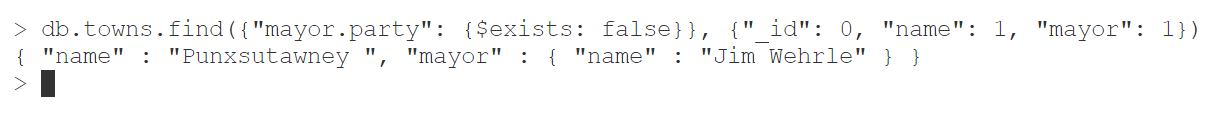
1. *Сформировать запрос, который возвращает список городов с независимыми мэрами (*party="I")*. Вывести только название города и информацию о мэре.*
2. *Сформировать запрос, который возвращает список беспартийных мэров (*party *отсутствует*)*. Вывести только название города и информацию о мэре.*

**

(рис. 14 - Создание коллекции towns)



(рис. 15 - Independent мэр)



(рис. 16 - Нет партий)

**Практическое задание 8.2.2:**

1. *Сформировать функцию для вывода списка самцов единорогов.*
2. *Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.*
3. *Вывести результат, используя forEach.*
4. *Содержание коллекции единорогов unicorns:*

db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});

db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});

db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});

db.unicorns.insert({name: 'Roooooodles', 44), loves: ['apple'], weight: 575, gender: 'm', vampires: 99});

db.unicorns.insert({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});

db.unicorns.insert({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});

db.unicorns.insert({name:'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});

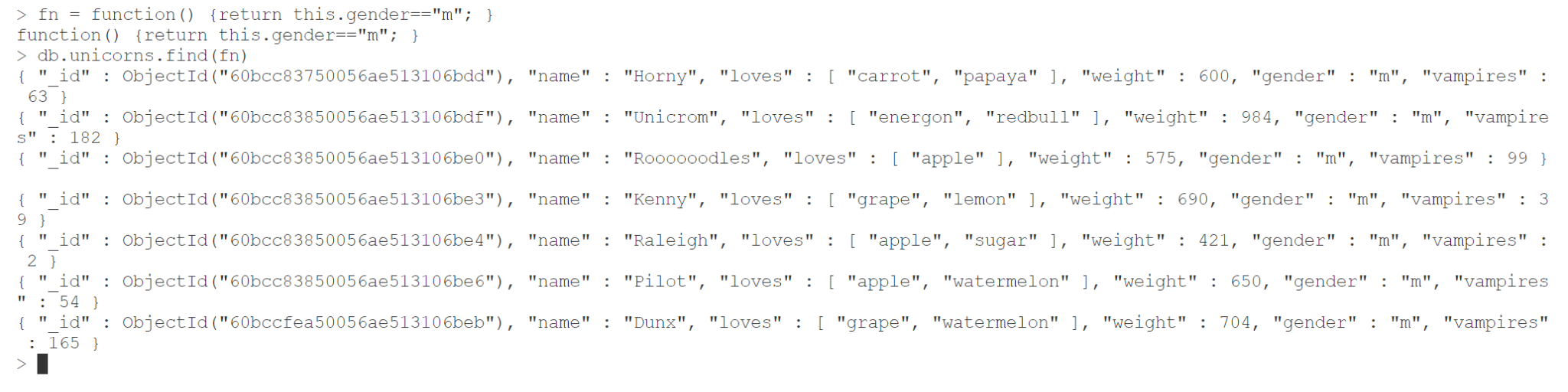
db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});

db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});

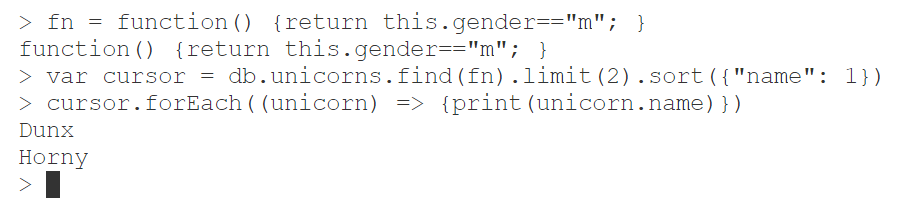
db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});

db.unicorns.insert ({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});

db.unicorns.insert ({name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165})



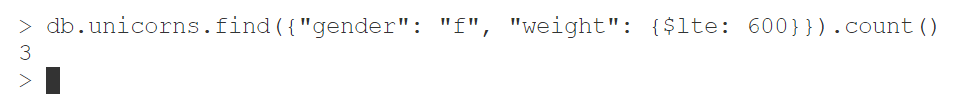
(рис. 17 - Функция)



(рис. 18 - Результат forEach)

**Практическое задание 8.2.3:**

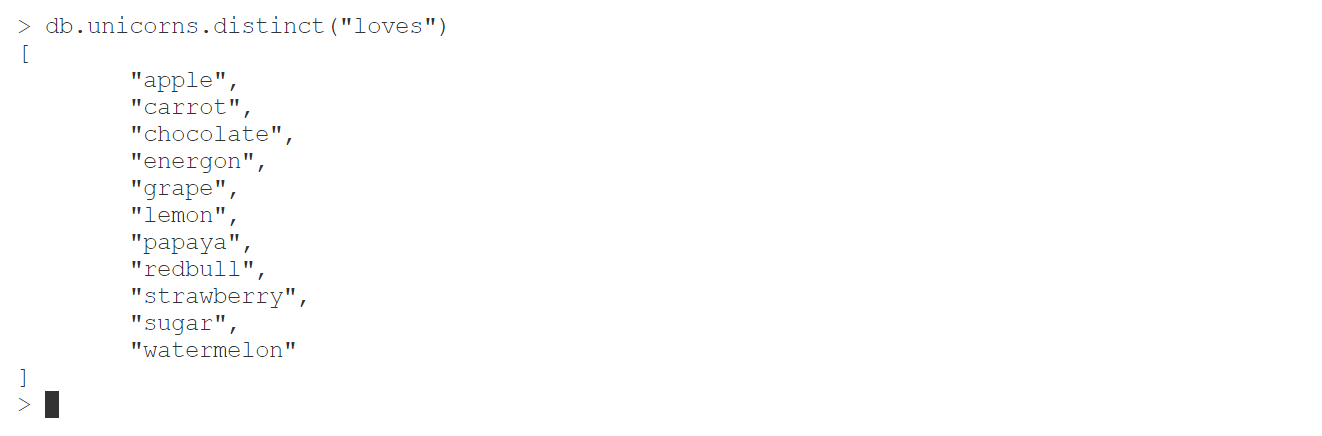
*Вывести количество самок единорогов весом от полутонны до 600 кг.*



(рис. 19 - Самки весом до 600 кг)

**Практическое задание 8.2.4:**

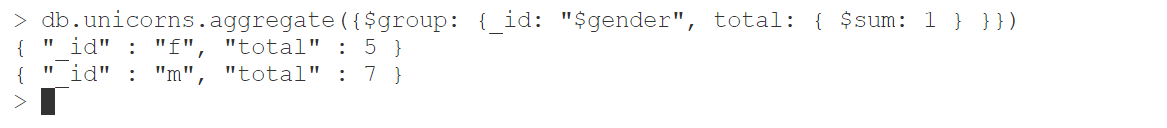
*Вывести список предпочтений.*

**

(рис. 20 - Список предпочтений)

**Практическое задание 8.2.5:**

*Посчитать количество особей единорогов обоих полов.*



(рис. 21 - Количество самцов и самок)

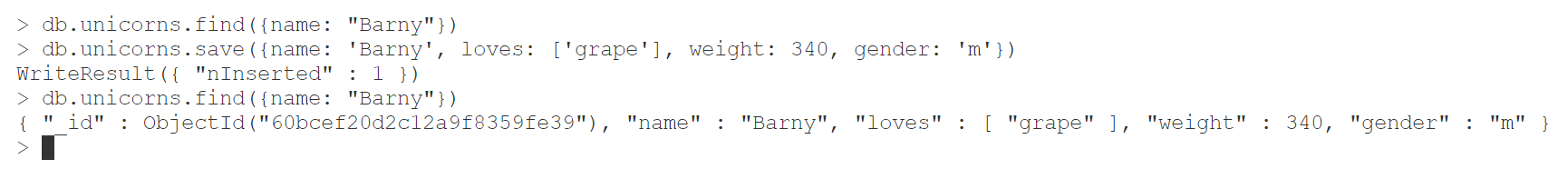
**Практическое задание 8.2.6:**

1. *Выполнить команду:*

> db.unicorns.save({name: 'Barny', loves: ['grape'],

weight: 340, gender: 'm'})

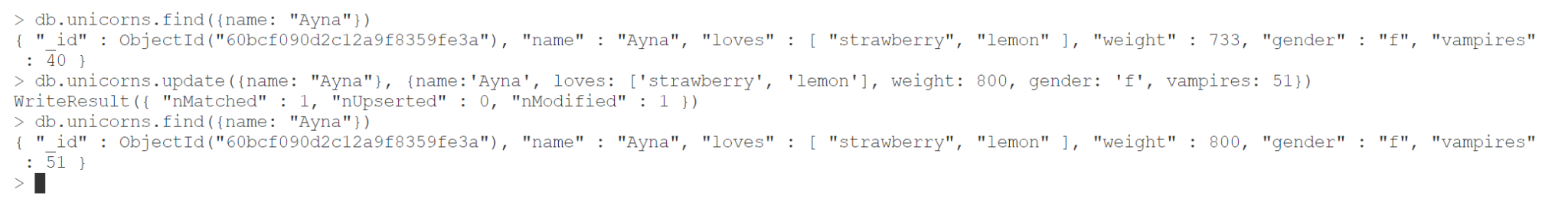
1. *Проверить содержимое коллекции unicorns.*



(рис. 22 - Добавление *Barny*)

**Практическое задание 8.2.7:**

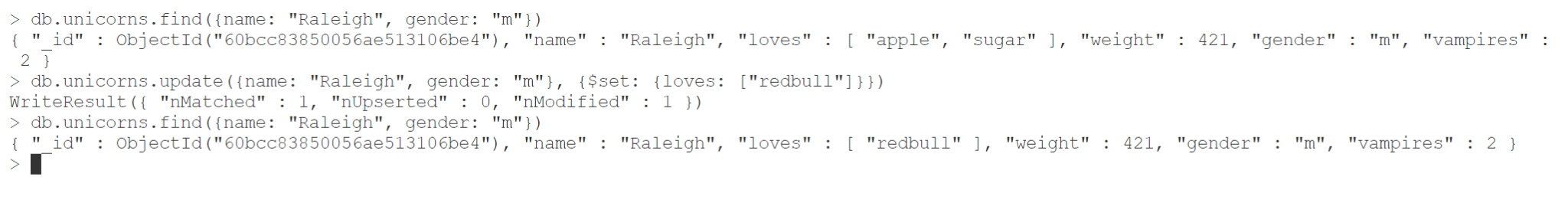
1. *Для самки единорога* Ayna *внести изменения в БД: теперь ее вес 800, она убила 51 вампира.*
2. *Проверить содержимое коллекции unicorns.*

**

(рис. 23 - Обновление данных об *Ayna*)

**Практическое задание 8.2.8:**

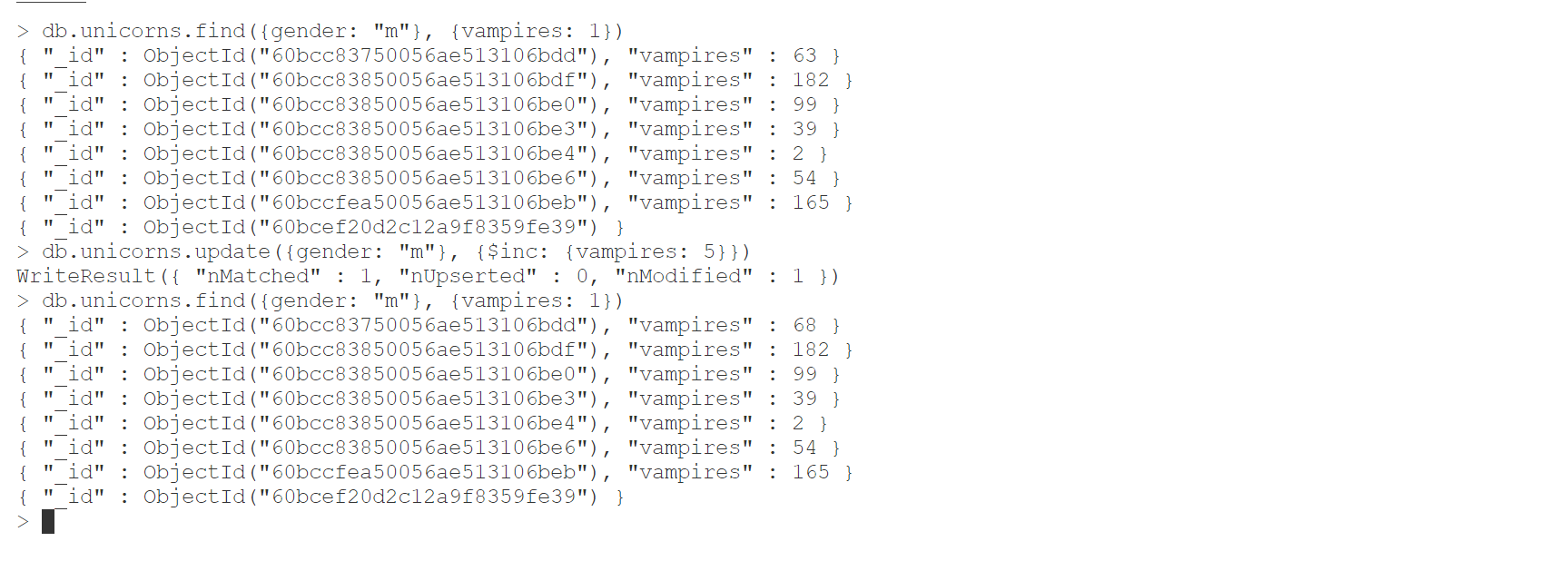
1. *Для самца единорога* Raleigh *внести изменения в БД: теперь он любит рэдбул.*
2. *Проверить содержимое коллекции unicorns.*

**

(рис. 24 - Обновление данных об *Raleigh*)

**Практическое задание 8.2.9:**

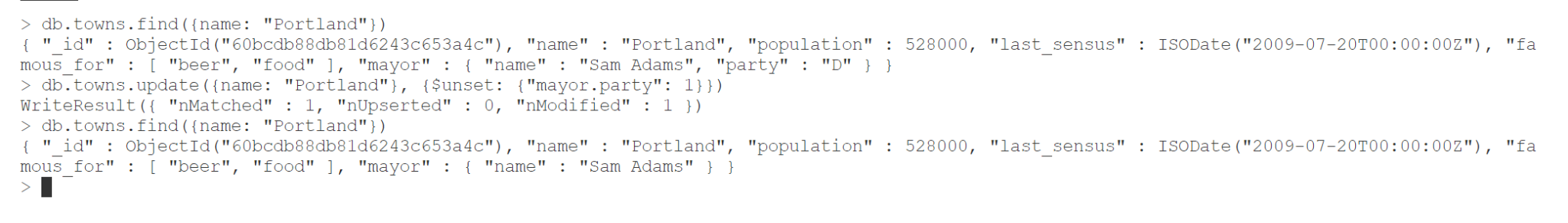
1. *Всем самцам единорогов увеличить количество убитых вампиров на 5.*
2. *Проверить содержимое коллекции unicorns.*

**

(рис. 25 - Увеличение *vampires*)

**Практическое задание 8.2.10:**

1. *Изменить информацию о городе Портланд: мэр этого города теперь беспартийный.*
2. *Проверить содержимое коллекции towns.*

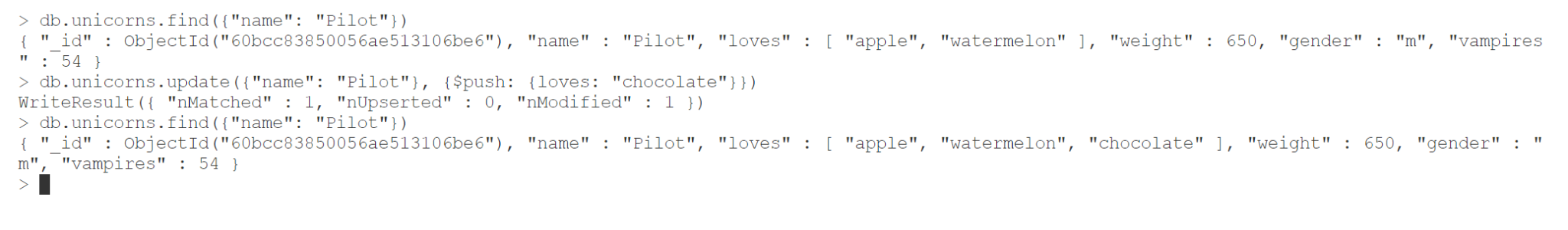
****

(рис. 26 - Удаление *party*)

**Практическое задание 8.2.11:**

1. *Изменить информацию о самце единорога Pilot: теперь он любит и шоколад.*

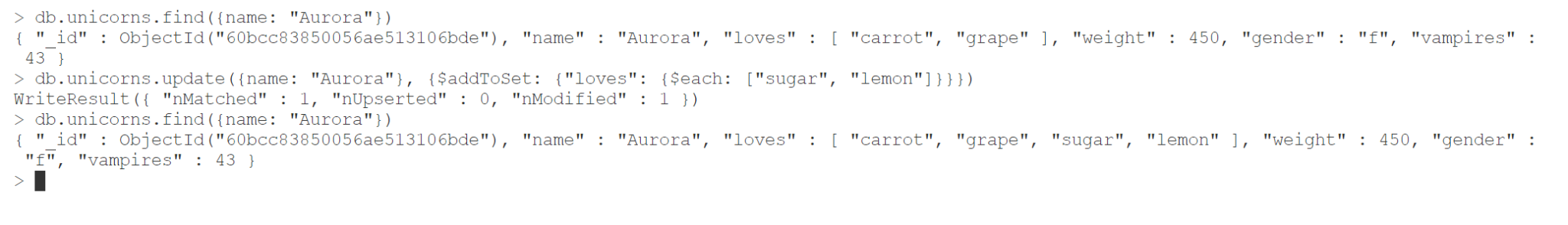
*Проверить содержимое коллекции unicorns.*



(рис. 27 - Добавление в *loves*)

**Практическое задание 8.2.12:**

1. *Изменить информацию о самке единорога* Aurora: *теперь она любит еще и сахар, и лимоны.*
2. *Проверить содержимое коллекции unicorns.*

**

(рис. 28 - Обновление данных об *Aurora*)

**Практическое задание 8.2.13:**

1. *Создайте коллекцию towns, включающую следующие документы:*

{name: "Punxsutawney",

population: 6200,

last\_sensus: ISODate("2008-01-31"),

famous\_for: ["phil the groundhog"],

mayor: {

name: "Jim Wehrle"

}}

{name: "New York",

population: 22200000,

last\_sensus: ISODate("2009-07-31"),

famous\_for: ["statue of liberty", "food"],

mayor: {

name: "Michael Bloomberg",

party: "I"}}

{name: "Portland",

population: 528000,

last\_sensus: ISODate("2009-07-20"),

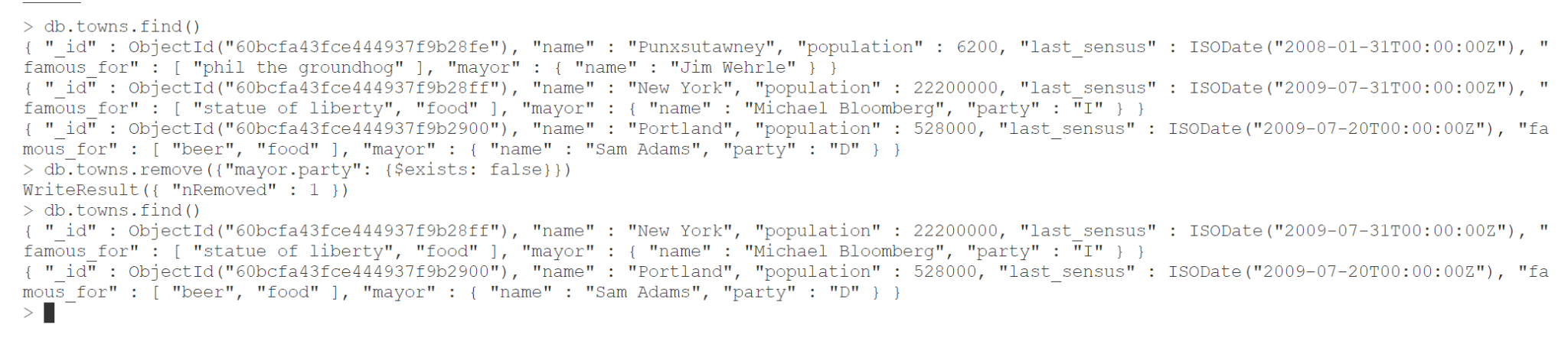
famous\_for: ["beer", "food"],

mayor: {

name: "Sam Adams",

party: "D"}}

1. *Удалите документы с беспартийными мэрами.*
2. *Проверьте содержание коллекции.*
3. *Очистите коллекцию.*
4. *Просмотрите список доступных коллекций.*

**

(рис. 29 - Удаление беспартийных меров)



(рис. 30 - Удаление коллекции *towns*)

**Практическое задание 8.3.1:**

1. *Создайте коллекцию зон обитания единорогов, указав в качестве идентификатора кратко название зоны, далее включив полное название и описание.*
2. *Включите для нескольких единорогов в документы ссылку на зону обитания, использую второй способ автоматического связывания.*
3. *Проверьте содержание коллекции единорогов.*
4. *Содержание коллекции единорогов unicorns:*

db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});

db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});

db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});

db.unicorns.insert({name: 'Roooooodles', 44), loves: ['apple'], weight: 575, gender: 'm', vampires: 99});

db.unicorns.insert({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});

db.unicorns.insert({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});

db.unicorns.insert({name:'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});

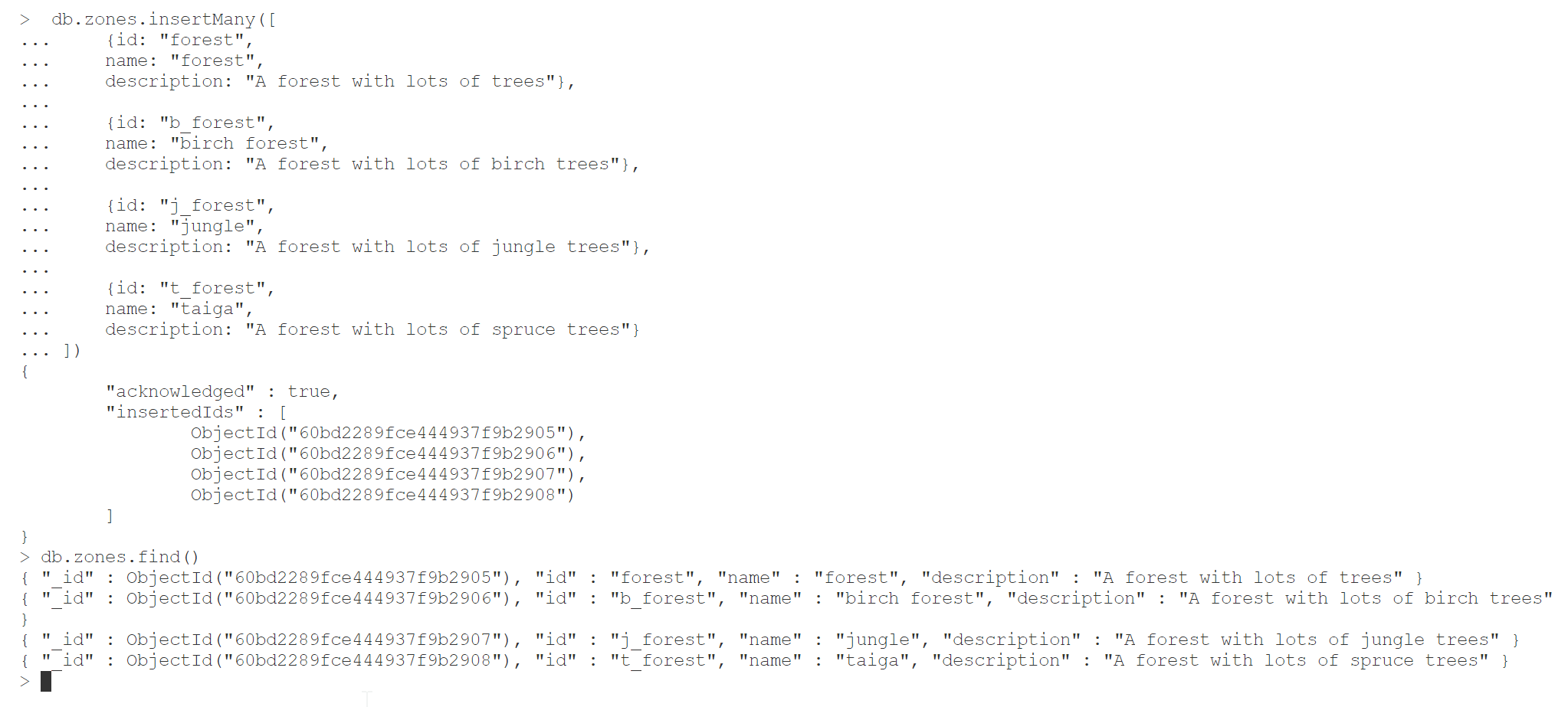
db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});

db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});

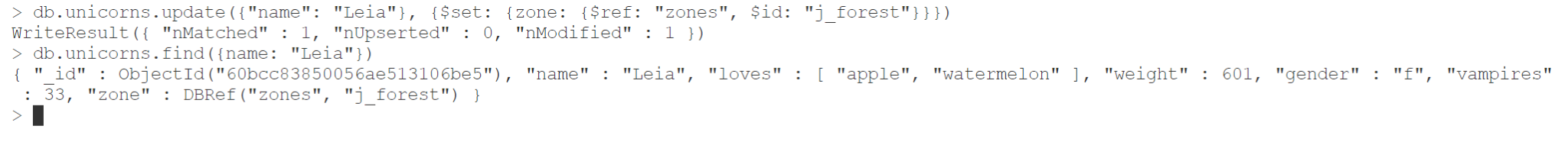
db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});

db.unicorns.insert ({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});

db.unicorns.insert {name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165}



(рис. 31 - Создание коллекции *zones*)



(рис. 32 - Вставка)



(рис. 33 - Результат)

**Практическое задание 8.3.2:**

1. *Проверьте, можно ли задать для коллекции unicorns индекс для ключа name с флагом* unique*.*
2. *Содержание коллекции единорогов unicorns:*

db.unicorns.insert({name: 'Horny', dob: new Date(1992,2,13,7,47), loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});

db.unicorns.insert({name: 'Aurora', dob: new Date(1991, 0, 24, 13, 0), loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});

db.unicorns.insert({name: 'Unicrom', dob: new Date(1973, 1, 9, 22, 10), loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});

db.unicorns.insert({name: 'Roooooodles', dob: new Date(1979, 7, 18, 18, 44), loves: ['apple'], weight: 575, gender: 'm', vampires: 99});

db.unicorns.insert({name: 'Solnara', dob: new Date(1985, 6, 4, 2, 1), loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});

db.unicorns.insert({name:'Ayna', dob: new Date(1998, 2, 7, 8, 30), loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});

db.unicorns.insert({name:'Kenny', dob: new Date(1997, 6, 1, 10, 42), loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});

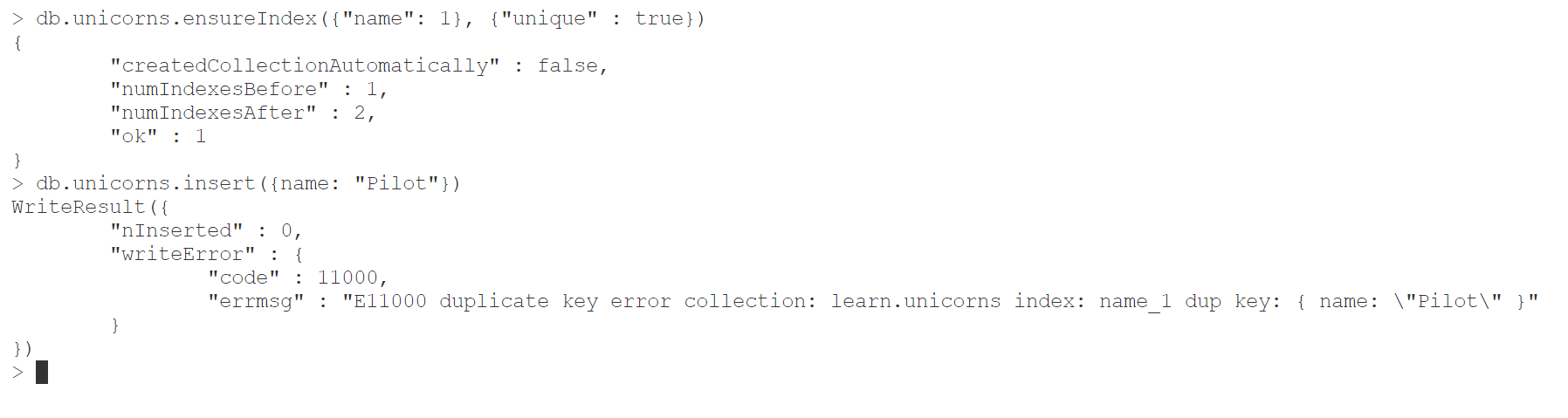
db.unicorns.insert({name: 'Raleigh', dob: new Date(2005, 4, 3, 0, 57), loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});

db.unicorns.insert({name: 'Leia', dob: new Date(2001, 9, 8, 14, 53), loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});

db.unicorns.insert({name: 'Pilot', dob: new Date(1997, 2, 1, 5, 3), loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});

db.unicorns.insert ({name: 'Nimue', dob: new Date(1999, 11, 20, 16, 15), loves: ['grape', 'carrot'], weight: 540, gender: 'f'});

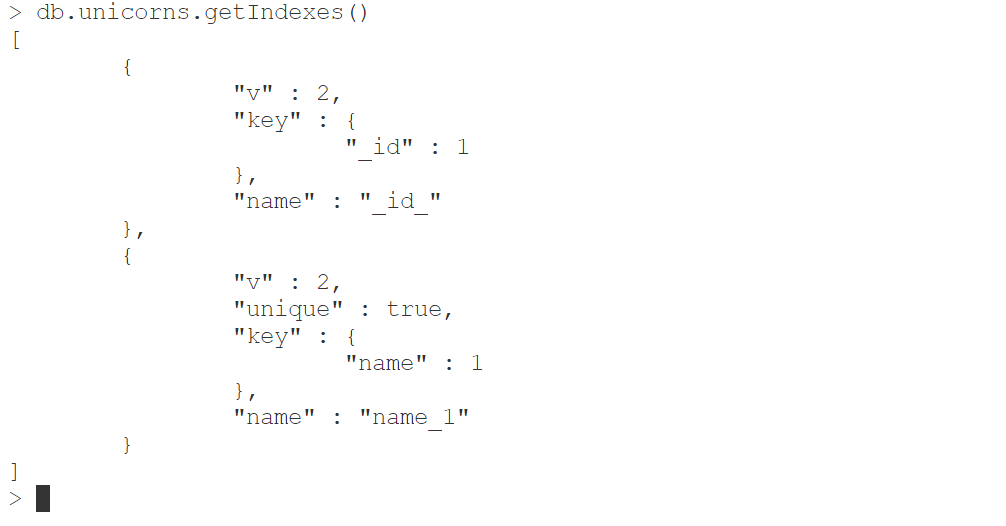
db.unicorns.insert {name: 'Dunx', dob: new Date(1976, 6, 18, 18, 18), loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165



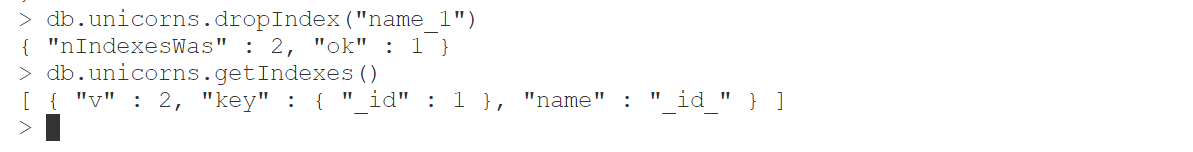
(рис. 34 - Создание ‘уникального’ индекса)

**Практическое задание 8.3.3:**

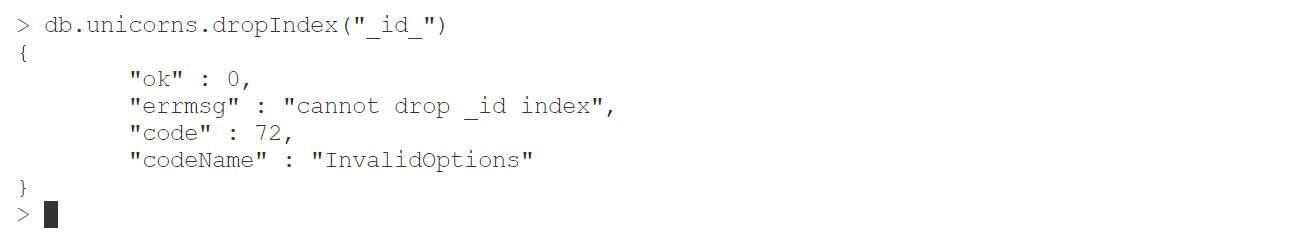
1. *Получите информацию о всех индексах коллекции unicorns .*
2. *Удалите все индексы, кроме индекса для идентификатора.*
3. *Попытайтесь удалить индекс для идентификатора.*

**

(рис. 35 - Индексы)



(рис. 36 - Удаление индекса)



(рис. 37 - Попытка удаления индекса *\_id\_*)

**Практическое задание 8.3.4:**

1. *Создайте объемную коллекцию numbers, задействовав курсор:*

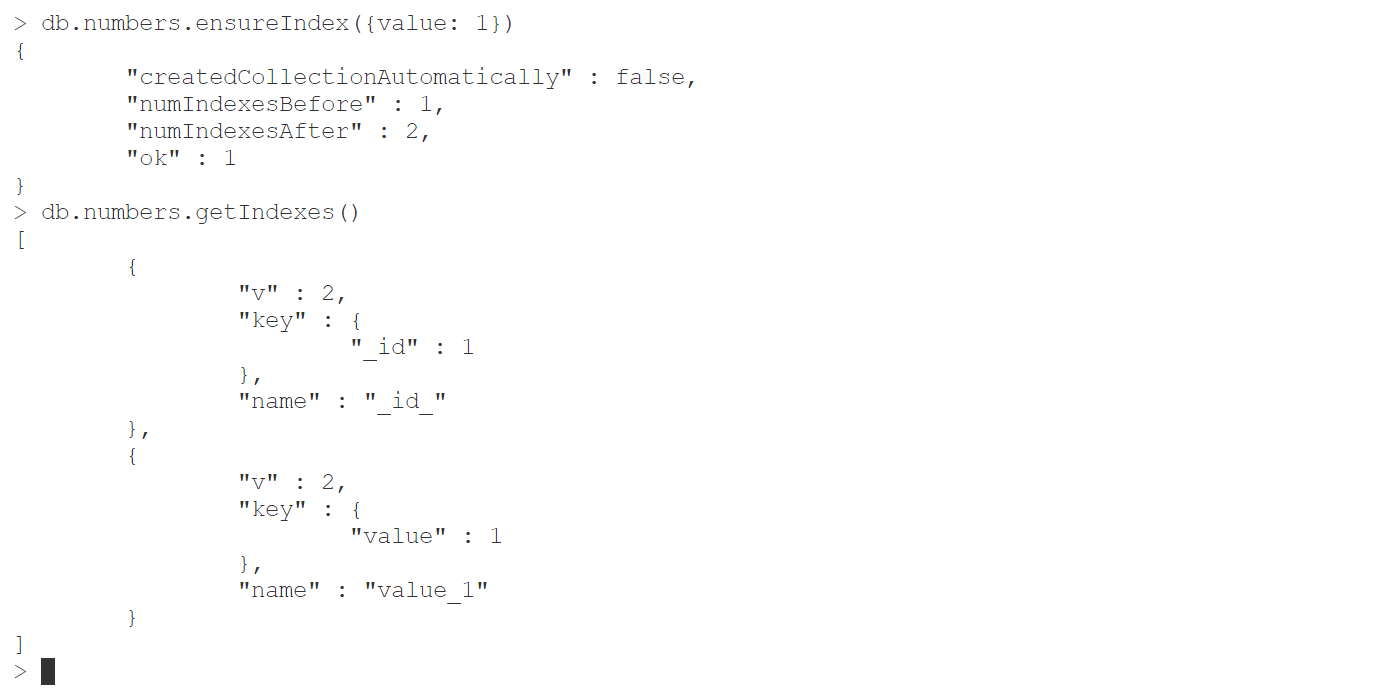
for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}

1. *Выберите последних четыре документа.*
2. *Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра executionTimeMillis)*
3. *Создайте индекс для ключа value.*
4. *Получите информацию о всех индексах коллекции numbers.*
5. *Выполните запрос 2.*
6. *Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?*
7. *Сравните время выполнения запросов с индексом и без. Дайте ответ на вопрос: какой запрос более эффективен?*

При выборе последних 4-х документов время исполнения команды колеблется оказалось 76 мс

**

(рис. 38 - Выбор последних 4-х документов)



(рис. 39 - Создание индекса для *numbers*)

После создания индекса время исполнения команды стало 0

(рис. 40 - Выбор последних 4-х документов после создания индекса)

Отвечая на вопрос “Какой из запросов эффективней?” можно сказать бессомненно, что индексирование сократило время выполнения с 76 мс до 0 мс, из этого следует, что запрос с индексом более эффективен.

**ВЫВОДЫ**

MongoDB предоставляет мощный CLI интерфейс для выполнения CRUD операций, отличительной особенностью является интеграция полноценного языка программирования: Javascript.