

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ФАКУЛЬТЕТ ПРИКЛАДНОЙ ИНФОРМАТИКИ

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

курса «Проектирование и реализация баз данных»

по теме:

«MongoDB»

Выполнил студент:
Лаптев Егор Игоревич
группа: К3239

Санкт-Петербург, 2025 г.

1 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

2 Практическое задание

1. Научиться работе с MongoDB через консольный интерфейс
2. Понять базовые принципы работы NoSQL БД
3. Уметь писать сложные запросы

3 Выполнение работы

3.1 CRUD-операции в СУБД MongoDB. Вставка данных. Выборка данных

3.1.1 Вставка документов в коллекцию

Практическое задание 2.1.1:

- Создайте базу данных `learn`.
- Заполните коллекцию единорогов `unicorns`.
- Используя второй способ, вставьте в коллекцию документ.
- Проверьте содержимое коллекции с помощью метода `find()`.

3.1.2 Выборка данных из БД

Практическое задание 2.2.1:

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

Практическое задание 2.2.2:

- Модифицируйте запрос для вывода списка самцов единорогов, исключив из результата информацию о предпочтениях и поле `gender`.

Практическое задание 2.2.3:

- Выведите список единорогов в обратном порядке добавления.

Практическое задание 2.2.4:

- Выведите список единорогов с названием первого любимого предпочтения, исключив поле `_id`.

3.2 Логические операторы

Практическое задание 2.3.1

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

Практическое задание 2.3.2

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

Практическое задание 2.3.3

Найти всех единорогов, не имеющих ключ `vampires`.

Практическое задание 2.3.4

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

3.3 Выборка данных. Вложенные объекты. Использование курсоров. Агрегированные запросы. Изменение данных

3.3.1 Запрос к вложенным объектам

Практическое задание 3.1.1

- Создайте коллекцию `towns`, включающую следующие документы.
- Сформировать запрос, который возвращает список городов с независимыми мэрами (`party = "I"`). Вывести только название города и информацию о мэре.
- Сформировать запрос, который возвращает список беспартийных мэров (`party` отсутствует). Вывести только название города и информацию о мэре.

3.3.2 Использование JavaScript и курсоры

Практическое задание 3.1.2

- Сформировать функцию для вывода списка самцов единорогов.
- Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.
- Вывести результат, используя `forEach`.

3.3.3 Агрегированные запросы

Практическое задание 3.2.1

Вывести количество самок единорогов весом от полутонны до 600 кг.

Практическое задание 3.2.2

Вывести список предпочтений.

Практическое задание 3.2.3

Посчитать количество особей единорогов обоих полов.

3.3.4 Редактирование данных

Практическое задание 3.3.1

Выполнить команду:

```
db.unicorns.save({name: 'Barney', loves: ['grape'], weight: 340, gender: 'm'})
```

Проверить содержимое коллекции `unicorns`.

Практическое задание 3.3.2

Для самки единорога Аупа внести изменения в БД: теперь её вес — 800, она убила 51 вампира.

Практическое задание 3.3.3

Для самца единорога Raleigh внести изменения в БД: теперь он любит редбул.

Практическое задание 3.3.4

Всем самцам единорогов увеличить количество убитых вампиров на 5.

Практическое задание 3.3.5

Изменить информацию о городе Портланд: мэр этого города теперь беспартийный.

Практическое задание 3.3.6

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

Практическое задание 3.3.7

Изменить информацию о самке единорога Auroga: теперь она любит ещё и сахар, и лимоны.

3.3.5 Удаление данных из коллекции

Практическое задание 3.4.1

- Создайте коллекцию `towns`.
- Удалите документы с беспартийными мэрами.
- Проверьте содержимое коллекции.
- Очистите коллекцию.
- Просмотрите список доступных коллекций.

3.4 Ссылки и работа с индексами в базе данных MongoDB

3.4.1 Ссылки в БД

Практическое задание 4.1.1

- Создайте коллекцию зон обитания единорогов, указав в качестве идентификатора краткое название зоны, полное название и описание.
- Включите для нескольких единорогов в документы ссылку на зону обитания, используя второй способ автоматического связывания.
- Проверьте содержание коллекции `unicorns`.

3.4.2 Настройка индексов

Практическое задание 4.2.1

Проверьте, можно ли задать для коллекции `unicorns` индекс для ключа `name` с флагом `unique`.

3.4.3 Управление индексами

Практическое задание 4.3.1

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

3.4.4 План запроса

Практическое задание 4.4.1

- Создайте объёмную коллекцию **numbers**, задействовав курсор:

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

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

4 Выводы

В ходе лабораторной работы были изучены и практически освоены базовые и продвинутые возможности MongoDB: Создание коллекций и вставка документов, Организация связей между коллекциями, Работа с индексами, а также Анализ производительности запросов

Работа позволила понять структуру документов в MongoDB, принципы связи коллекций, важность индексов и методы анализа производительности. Индексы значительно повышают эффективность выборки, особенно при сортировке и фильтрации по большим коллекциям.