

Отчёт

По выполнению практических заданий

Дисциплина: МДК 13.02

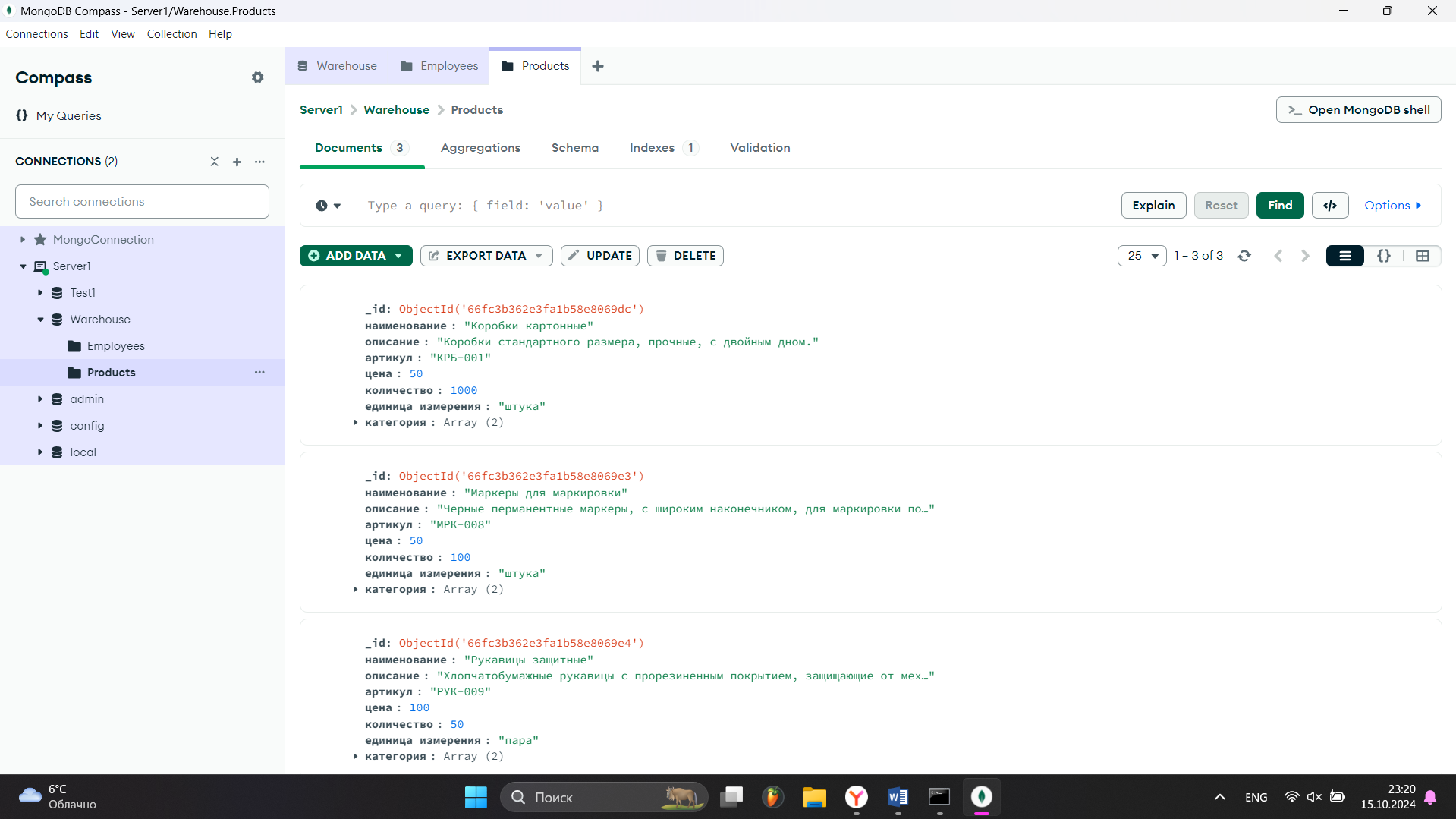
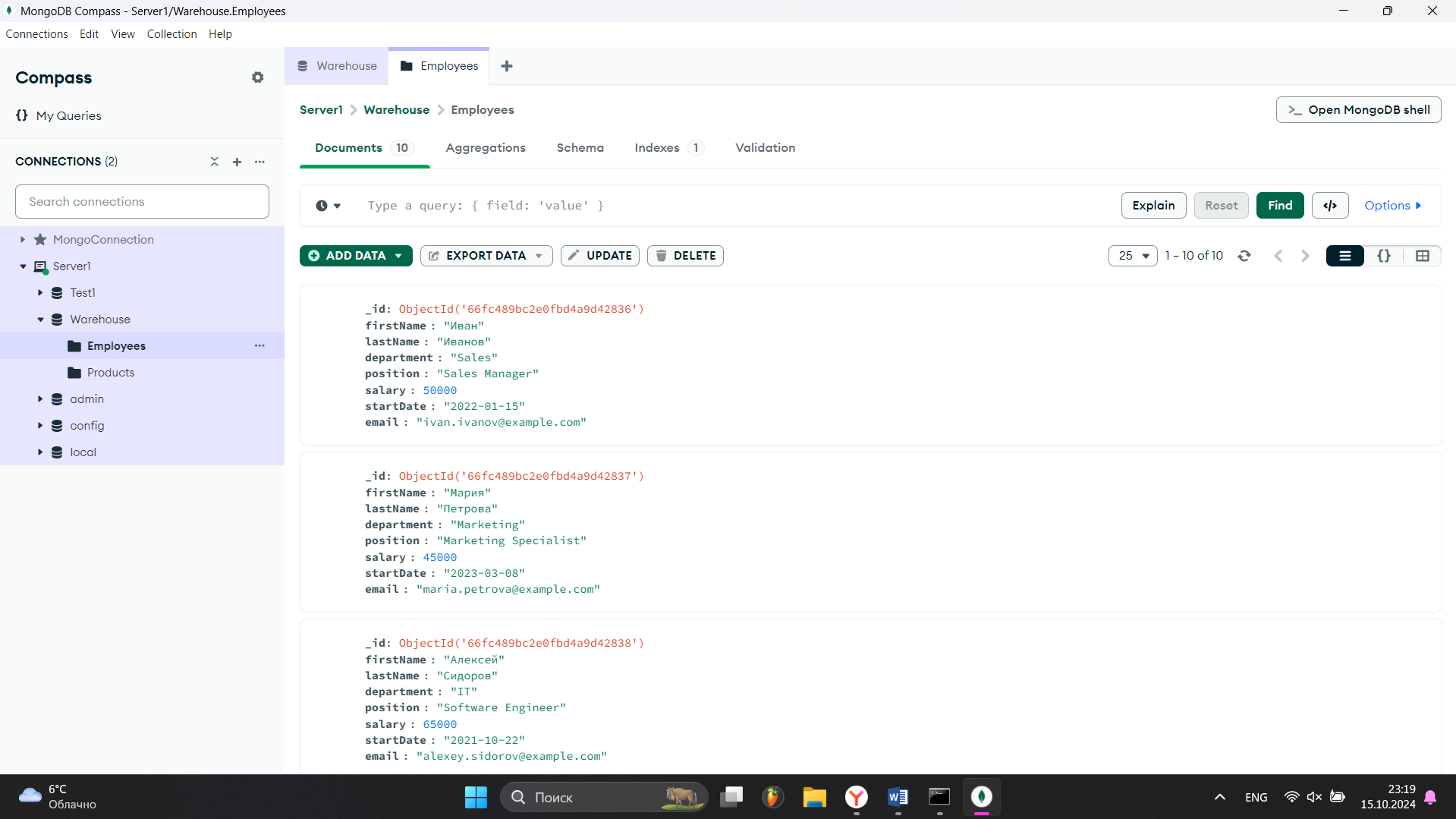
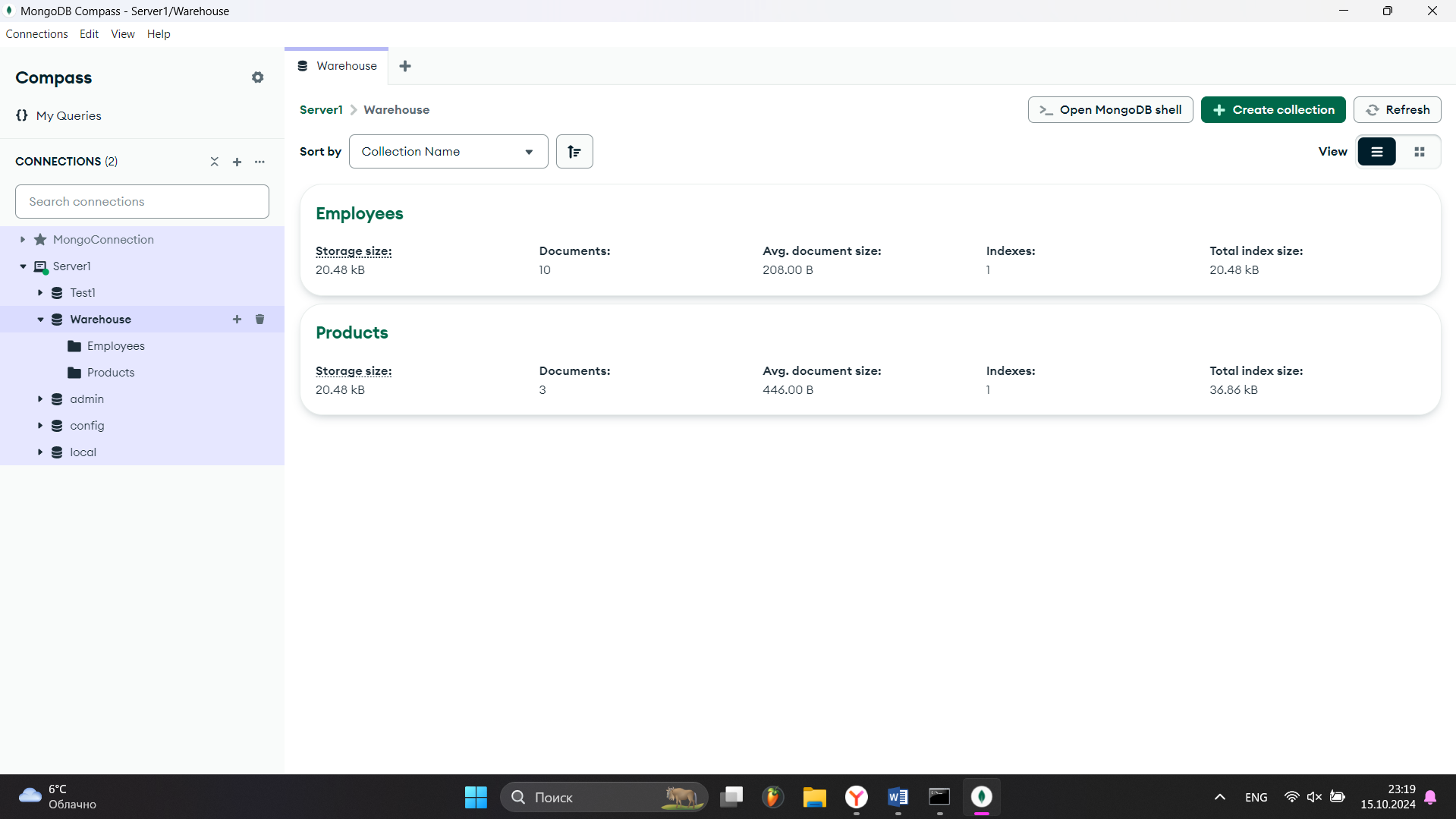
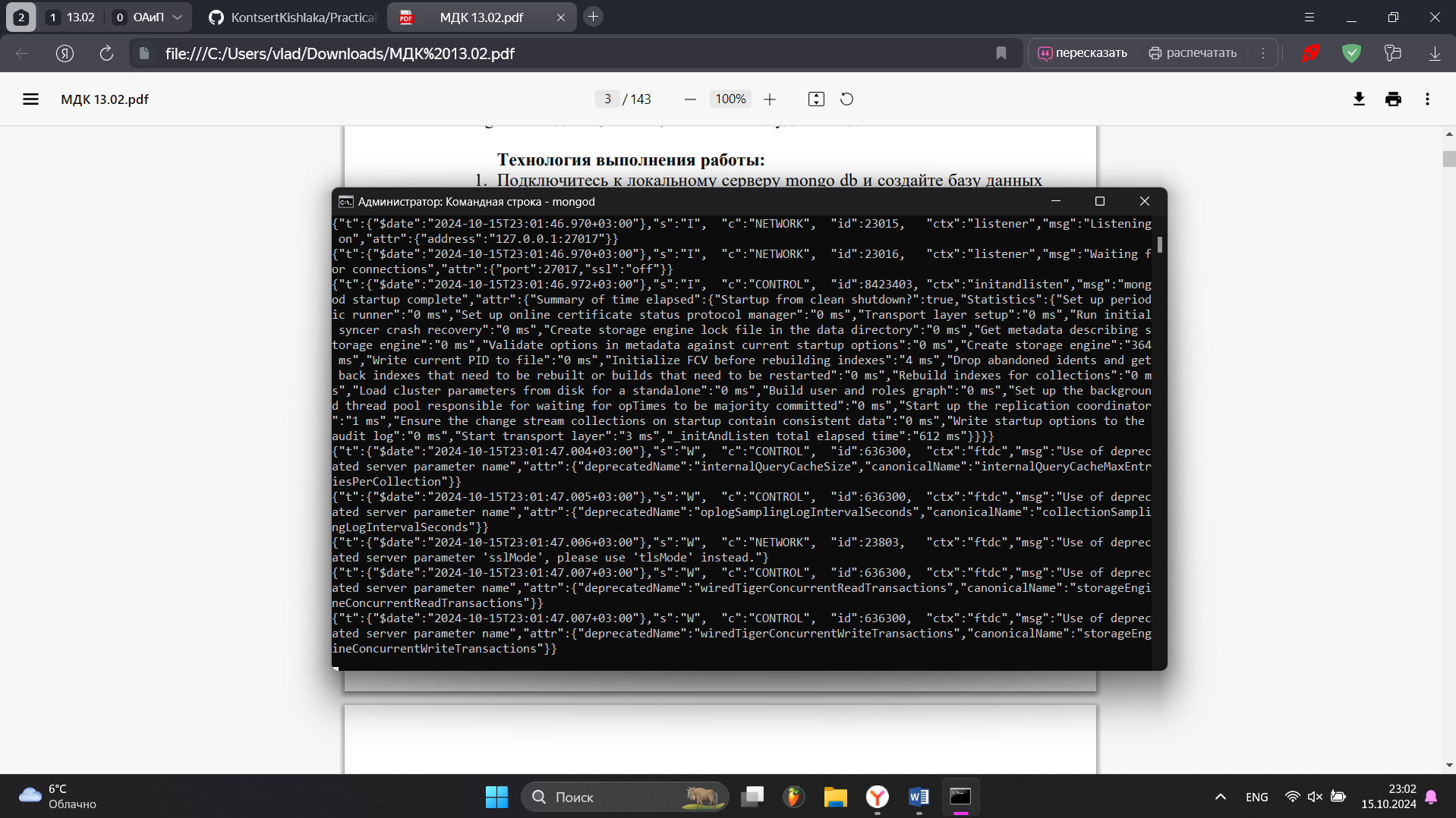
Исследуемая область: База данных MongoDB

Учебная группа: 1ИСП-11-41

Студенты: Бедин Владислав, Вельдяева Александра

Практическое задание №1

Скриншоты выполненного задания:



Команды, использованные в задании:

use Warehouse; // Создание БД “Warehouse”

show databases; // Показать все БД

db.createCollection(“Products”); // Создать коллекцию “Products” в текущей БД

db.createCollection(“Employees”); // Создать коллекцию “Employees” в текущей БД

show collections; // Показать все коллекции в текущей БД

db.Products.insertMany([{1-ый документ}, {2-ой документ}, {3-й документ}]); // Добавить документы в указанную коллекцию

load(“employees.json”); // Загрузить данные в коллекцию из указанного файла (не работал из-за специфики моей ОС)

db.Products.find(); // Вывести все документы коллекции “Products”

db.Employees.find(); // Вывести все документы коллекции “Employees”

db.Products.deleteMany({цена: {$gt: 100} }); // Удалить все документы из коллекции “Products”, чья цена БОЛЬШЕ чем 100

db.Employees.replaceOne({firstName: “Елена”}, lastName: “Романова”}, { ПОЛНОСТЬЮ ИЗМЕНЁННЫЙ СООТВЕТСТВУЮЩИЙ ДОКУМЕНТ }); // Заменить первый документ в коллекции “Employees”, соответствующий фильтру на новый указанный документ

db.Products.drop(); // Удалить коллекцию “Products”

db.Employees.drop(); // Удалить коллекцию “Employees”

db.dropDatabase(); // Удалить текущую БД

Контрольные вопросы:

1. Что такое нереляционные БД?

2. Какие плюсы имеют нереляционные БД?

3. Что такое MongoDB?

4. Дайте определение понятиям документ, коллекция.

5. В каком формате данные хранятся в MongoDB?

6. Расскажите об основных командах, используемых при работе с MongoDB

Ответы на контрольные вопросы:

1. Нереляционные БД – это базы данных, не использующие таблицы и отношения между ними. Они используют другие модели хранения данных (например: ключ-значение, документы, графы).
2. Плюсы нереляционных БД:

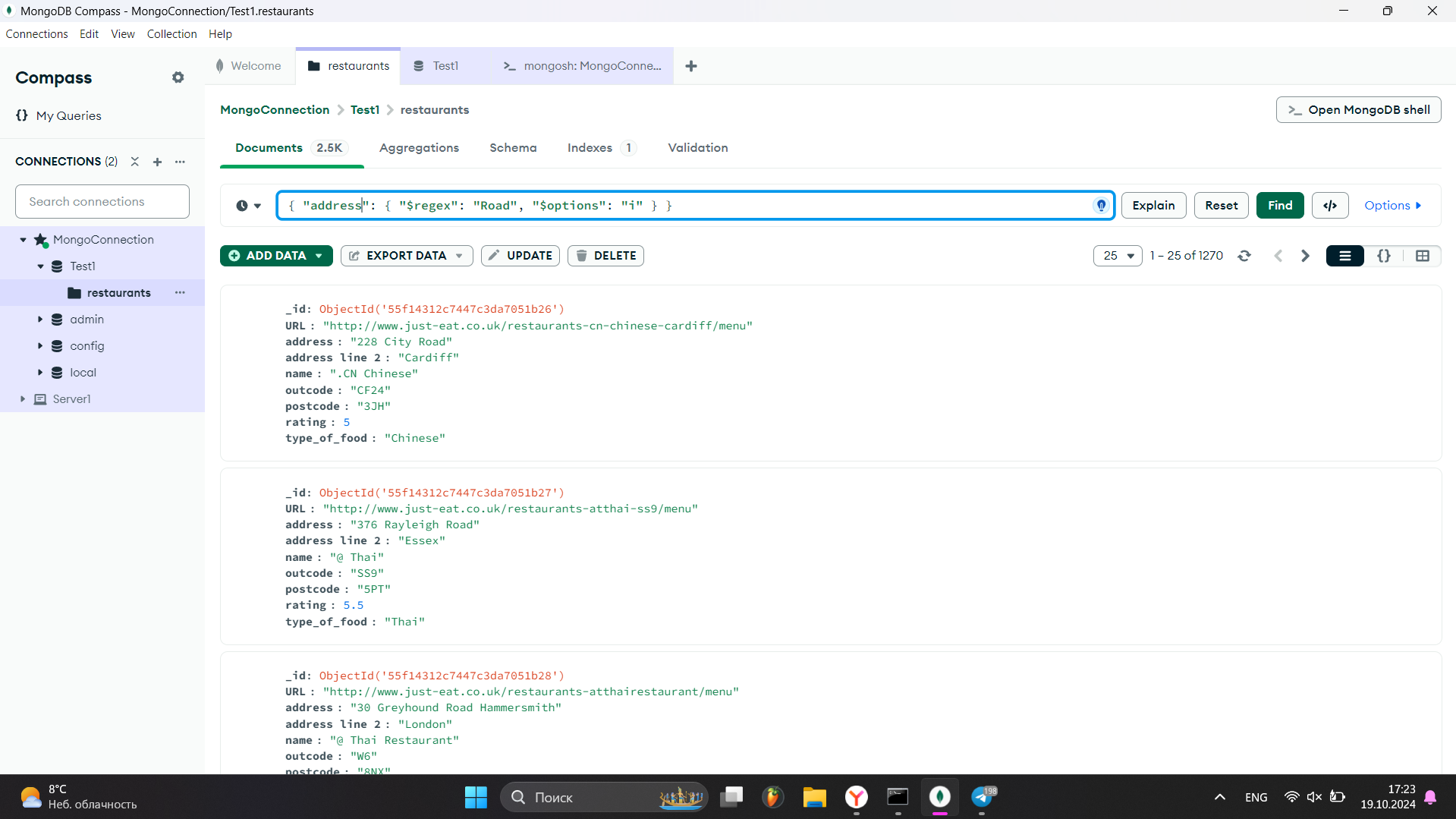
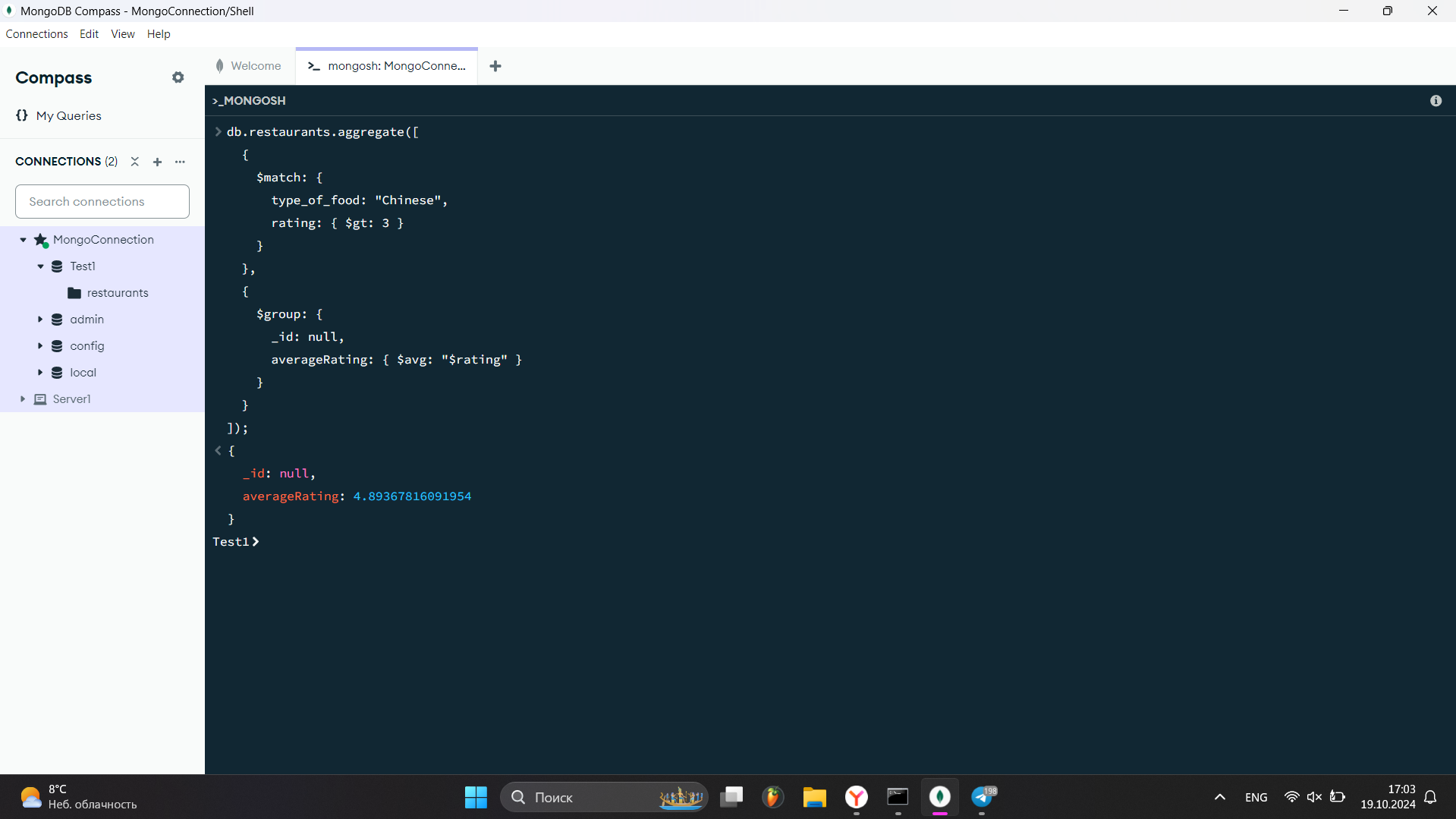
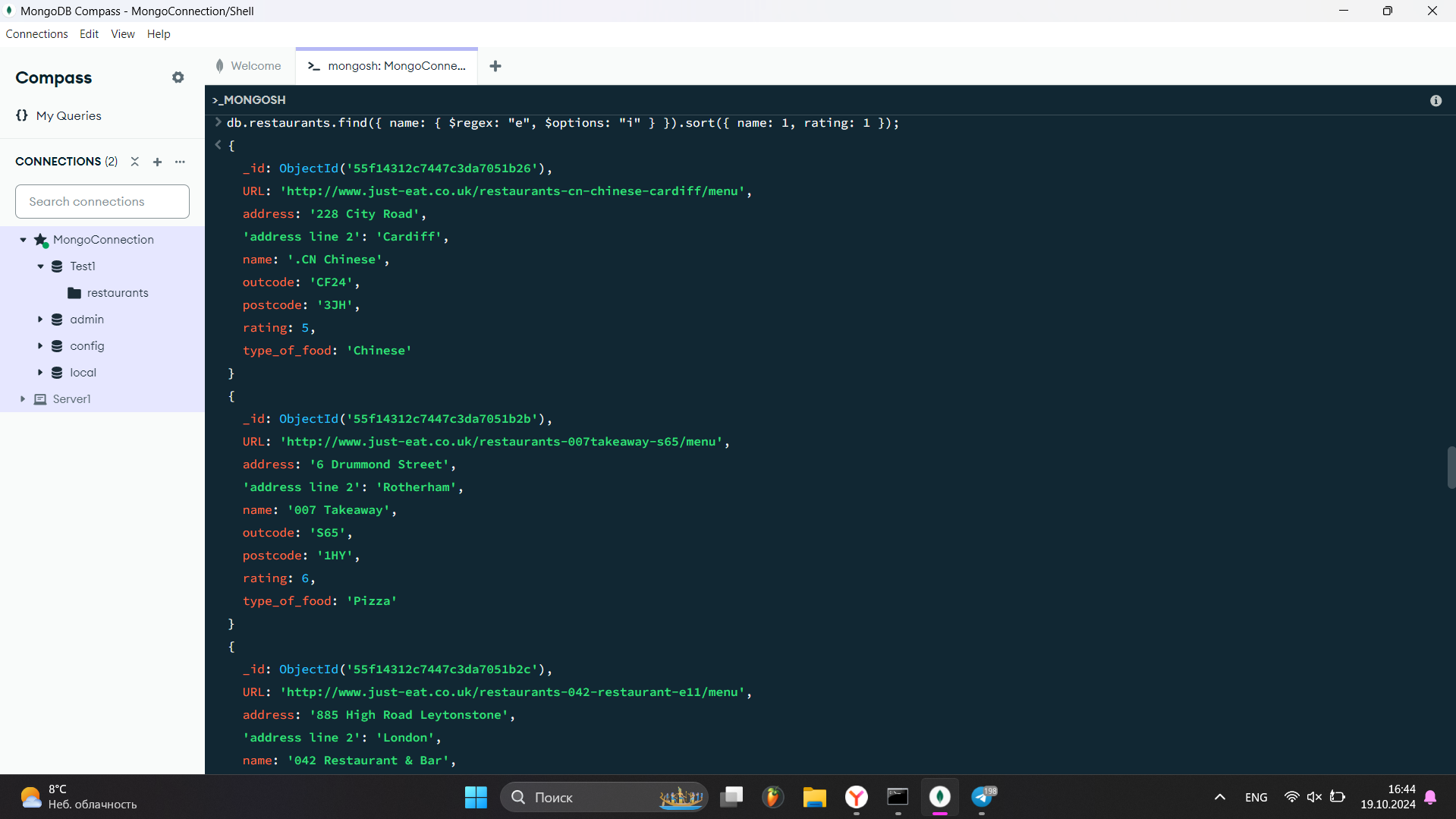
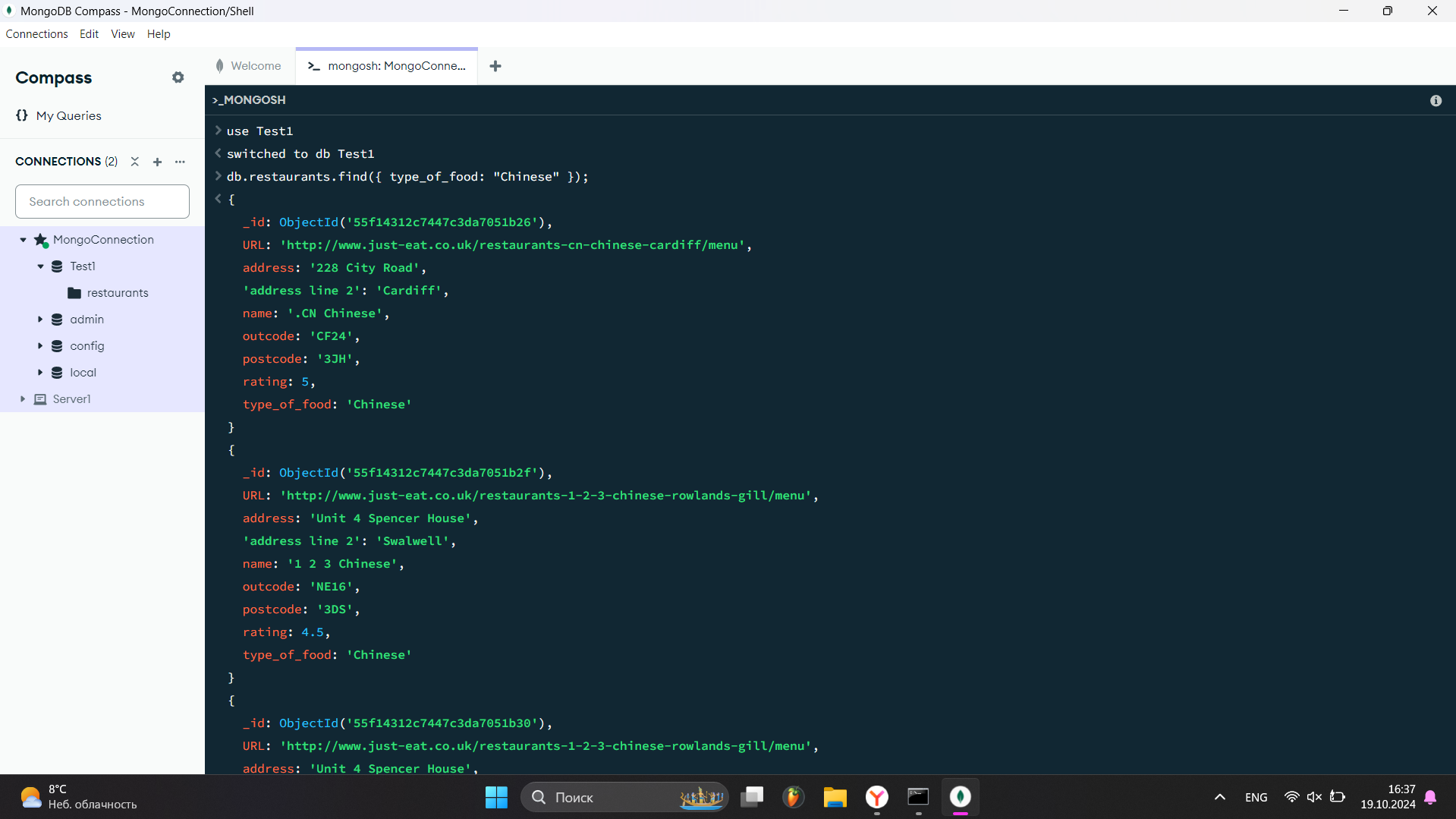
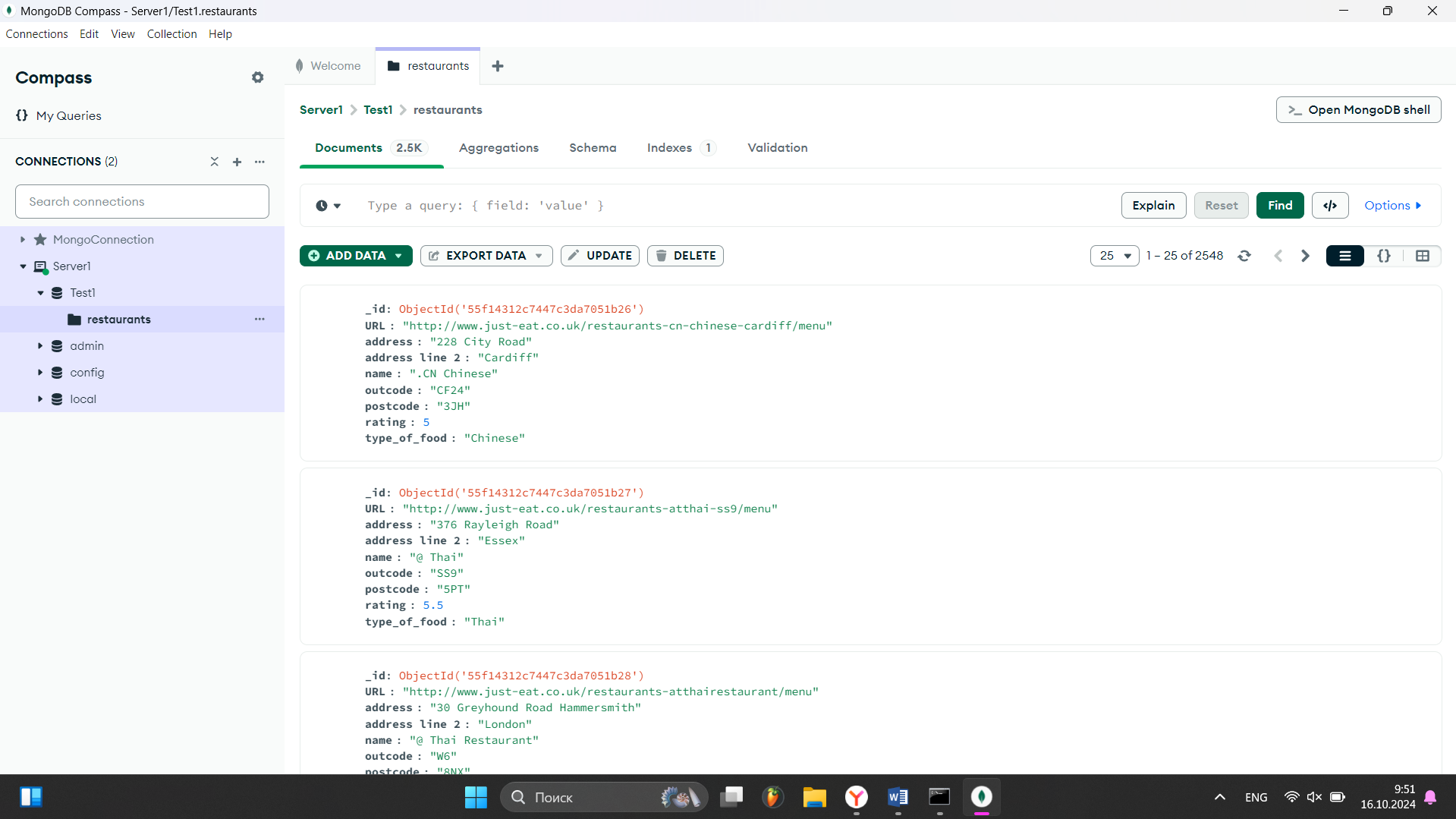
* Высокая производительность: Быстро обрабатывают большие объемы данных.
* Гибкость: Легко адаптируются к изменяющимся потребностям данных.
* Масштабируемость: Легко масштабируются как вертикально (добавление ресурсов), так и горизонтально (добавление узлов).

1. MongoDB – это популярная нереляционная база данных, основанная на модели документов.
2. Документ – это структурированный набор данных в MongoDB (наподобие строки в реляционных БД). Он содержит поля с различными типами данных. Коллекция – аналогия таблицы в реляционных БД. Коллекция содержит набор документов.
3. В MongoDB хранятся данные в формате JSON (JavaScript Object Notation).
4. Некоторые основные команды MongoDB:

* db.collection.insertOne(); // Вставляет один документ в коллекцию.
* db.collection.find(); // Выбирает документы, соответствующие заданному условию.
* db.collection.updateMany(); // Обновляет документы, соответствующие заданному условию.
* db.collection.deleteMany(); // Удаляет документы, соответствующие заданному условию.

Практическое задание №2

Скриншоты выполненного задания:



Команды, использованные в задании:

db.restaurants.find(); // Вывести все документы коллекции “restaurants”

db.restaurants.countDocuments(); // Вывести количество документов в коллекции “restaurants”

db.restaurants.find().limit(10); // Вывести первые 10 документов коллекции “restaurants”

db.restaurants.find().limit(10).skip(7); // Вывести первые 10 документов коллекции “restaurants”, начиная с седьмого

db.restaurants.find().limit(10).skip(7); // Вывести первые 10 документов коллекции “restaurants”, начиная с седьмого

db.restaurants.find({ rating: {$gte: 4} }).itcount(); // Вывести количество ресторанов с рейтингом 4 и выше коллекции “restaurants”

db.restaurants.find( { $and: [{ rating: {$gte: 2} }, { rating: { $lt: 5 } }] }); // Вывести все рестораны с рейтингом от 2 до 5 коллекции “restaurants”

db.restaurants.find( { $and: [{ rating: {$gte: 2} }, { rating: { $lt: 5 } }] }).itcount(); // Вывести количество ресторанов с рейтингом от 2 до 5 коллекции “restaurants”

db.restaurants.find({ type\_of\_food: "Chinese" }); // Вывести все рестораны с китайской кухней из коллекции “restaurants”

db.restaurants.find({ name: { $regex: "e", $options: "i" } }).sort({ name: 1, rating: 1 }); // Вывести рестораны из коллекции “restaurants”, в которых в названии есть буква «e» и отсортировать их в алфавитном порядке, а потом в порядке возрастания рейтинга

db.restaurants.find({ URL: { $exists: true, $ne: null } }).sort({ postcode: -1 }).limit(5); // Вывести первые 5 ресторанов из коллекции “restaurants”, в которых есть ссылка на меню и отсортировать их в порядке убывания postcode

db.restaurants.aggregate([ { $match: { type\_of\_food: "Chinese", rating: { $gt: 3 } } }, { $group: { \_id: null, averageRating: { $avg: "$rating" } } } ]);

// Вывести все рестораны китайской кухни с рейтингом выше 3 из коллекции “restaurants”, а затем посчитать среднее арифметическое их рейтинга.

{ "address": { "$regex": "Road", "$options": "i" } } // Интерфейс MongoDB Compass: вывести те рестораны из коллекции “restaurants”, в которых в «address» есть слово «Road» (не зависит от регистра)

Контрольные вопросы:

1. Дайте определение следующих понятий – выборка, фильтрация, аггрегация, лимитизация, ранжирование (сортировка).
2. Перечислите основные команды, используемые при поиске документов
3. Перечислите методы оптимизации запросов
4. Что такое MongoDB Compass?
5. Как производить импорт и экспорт файлов, делать запросы в Mongo DB Compass?

Ответы на контрольные вопросы:

1. • Выбор (Projection) - это определение полей, которые нужно включить в результат запроса.

• Фильтрация (Filtering) – это отбор документов, соответствующих заданным условиям.

• Агрегация (Aggregation) – это группировка документов и выполнение операций над ними (например, подсчет, среднее значение).

• Лимитизация (Limit) – это ограничение количества документов в результате запроса.

• Ранжирование (Сортировка) – это сортировка документов по одному или нескольким полям в определенном порядке.

2. Основные команды поиска документов:

• find(): Основная команда для поиска документов.

• findOne(): Выводит только один документ, соответствующий условиям.

• count(): Подсчитывает количество документов в коллекции.

• distinct(): Возвращает уникальные значения заданного поля.

3. Методы оптимизации запросов:

• Использование индексов для ускорения поиска.

• Оптимизация запросов с помощью операторов (например, $gt, $lt, $in).

• Минимизация количества полей, возвращаемых запросом.

• Использование агрегации для выполнения сложных операций над данными.

4. Что такое MongoDB Compass?

MongoDB Compass – это графический интерфейс для работы с MongoDB, позволяющий визуализировать данные, выполнять запросы, создавать индексы, управлять базами данных и коллекциями.

5. Импорт/экспорт и запросы в MongoDB Compass:

• Импорт:

\* Выбираем коллекцию, куда будем импортировать данные.

\* Нажимаем кнопку "ADD DATA" (зелёная кнопка с плюсиком).

\* Выбираем формат файла (например, .json или .csv) и указываем путь к файлу.

• Экспорт:

\* Выбираем коллекцию, которую нужно экспортировать.

\* Нажимаем кнопку "EXPORT DATA" (белая кнопка, справа от кнопки «ADD DATA»).

\* Выбираем необходимые для экспорта столбцы, а затем формат файла и его путь экспорта.

• Запросы:

\* Выбираем коллекцию для запроса.

\* Вводим запрос в текстовом поле поиска по фильтру.

\* Нажмаем кнопку "Find".

Практическое задание №3

Тема: Управление индексами. Ускорение запросов при помощи индексов.

Цель: Освоить методы ускорения запросов и создания индексов в Mongo DB

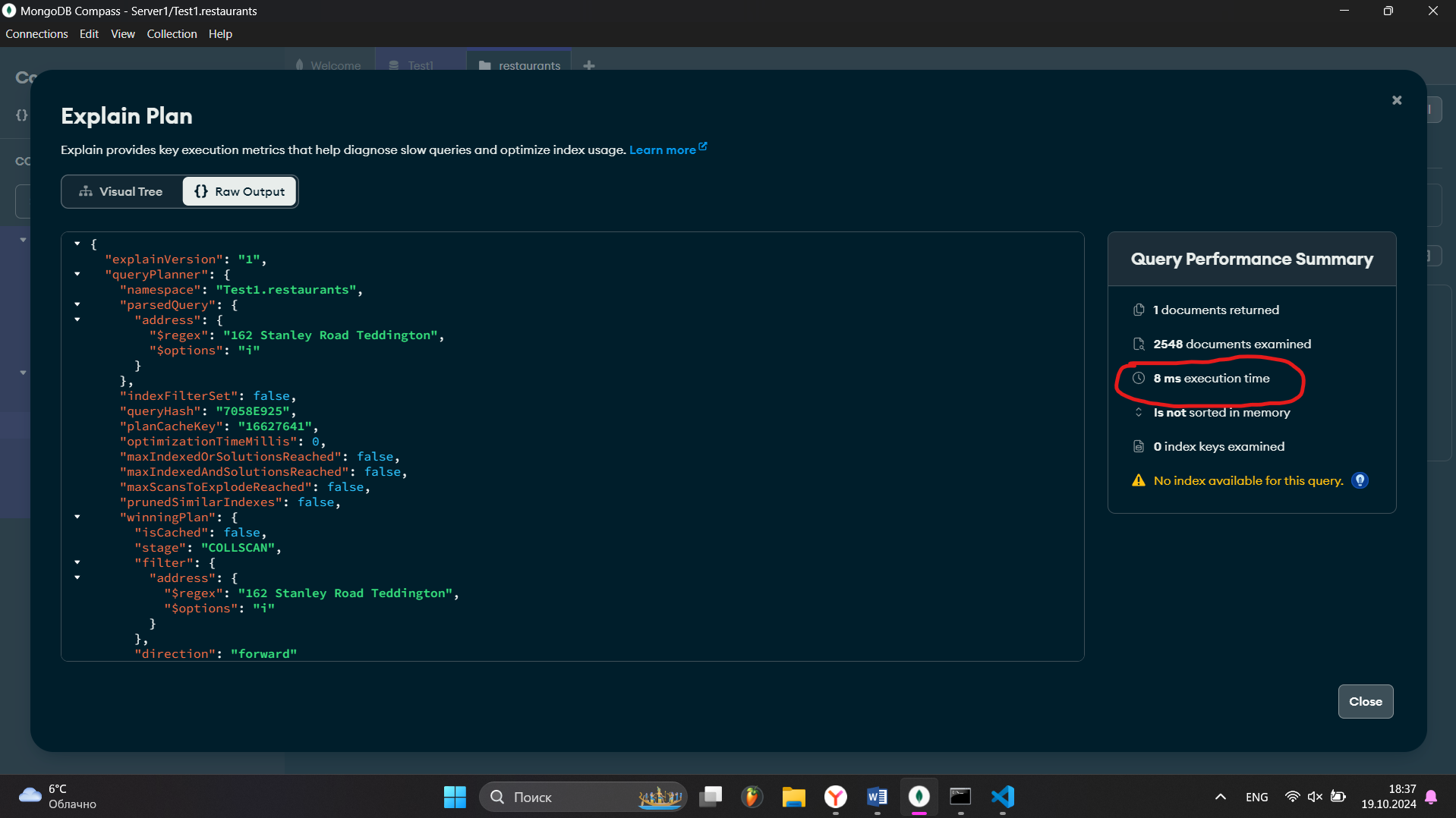
Скриншоты выполненного задания:

Запрос по индексу: нет

Фильтр поиска: { address: RegExp("162 Stanley Road Teddington", "i") }

Время выполнения запроса: 8 мс

Скриншот:



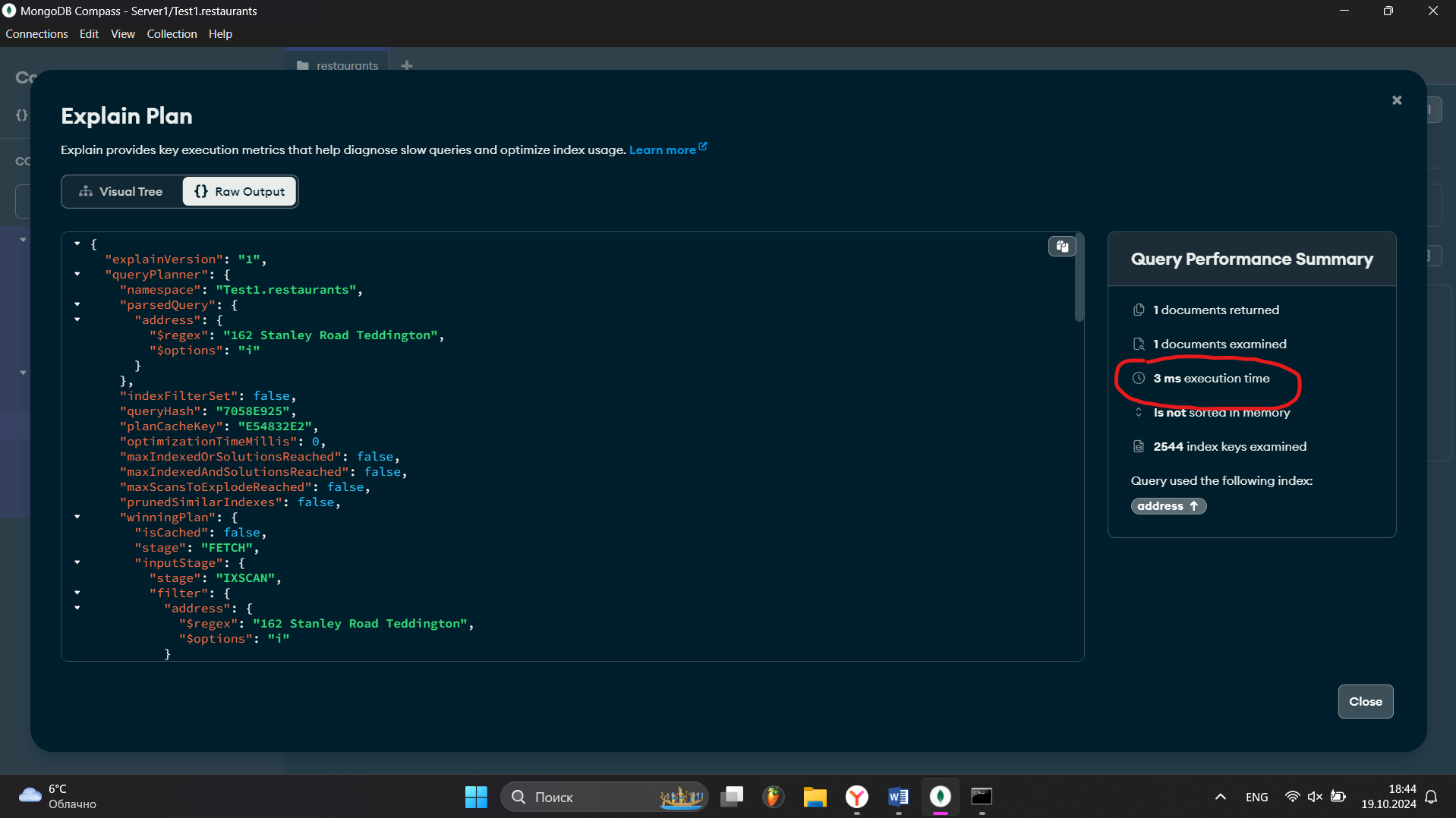
Запрос по индексу: да

Название индекса: address\_1

Фильтр поиска: { address: RegExp("162 Stanley Road Teddington", "i") }

Время выполнения запроса: 3 мс

Скриншот:



Контрольные вопросы:

1. Что представляют собой индексы в базах данных?
2. Какие преимущества предоставляют индексы в управлении базами данных?
3. Какие типы индексов существуют?
4. Какие оптимизации происходят при использовании индексов в запросах?

Ответы на контрольные вопросы:

* + - 1. Индексы – это специальные структуры данных, которые хранят упорядоченные значения одного или нескольких полей из коллекции. Они служат для быстрого поиска документов по указанным полям, подобно указателю в книге.
      2. Ускоренный поиск (индексы позволяют быстро находить документы по заданным критериям, не просматривая всю коллекцию); повышение производительности (индексы значительно ускоряют выполнение запросов, особенно при работе с большими объемами данных); оптимизация операций сортировки и агрегации (индексы могут использоваться для оптимизации операций сортировки и агрегации, что также повышает производительность).
      3. Типы индексов:

• "single": Индекс по одному полю.

• "compound": Индекс по нескольким полям.

• "hashed": Индекс с использованием хэширования.

• "geoSpatial": Индекс для работы с географическими данными.

• "text": Индекс для полнотекстового поиска.

4. Оптимизации при использовании индексов:

• Ускорение поиска: Индексы позволяют MongoDB сканировать только часть коллекции, содержащую нужные значения.

• Сортировка: Индексы могут использоваться для ускорения операций сортировки, так как данные в индексе уже упорядочены.

• Агрегация: Индексы могут применяться для оптимизации агрегации, сокращая количество операций сканирования.

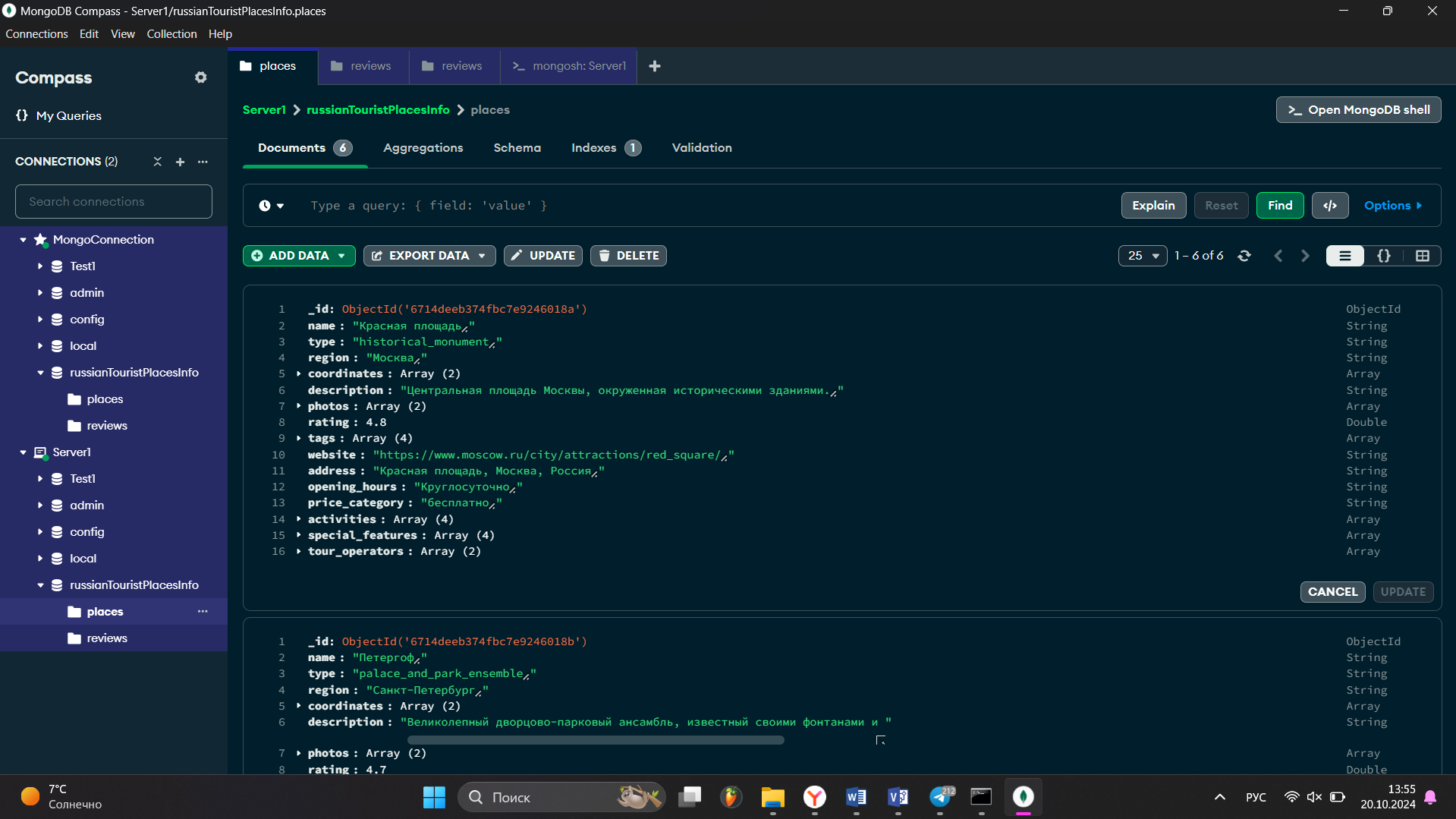
Практическое задание №4

Тема: Планирование БД и коллекций, моделирование данных.

Цель: Освоить методы планирования БД и коллекций

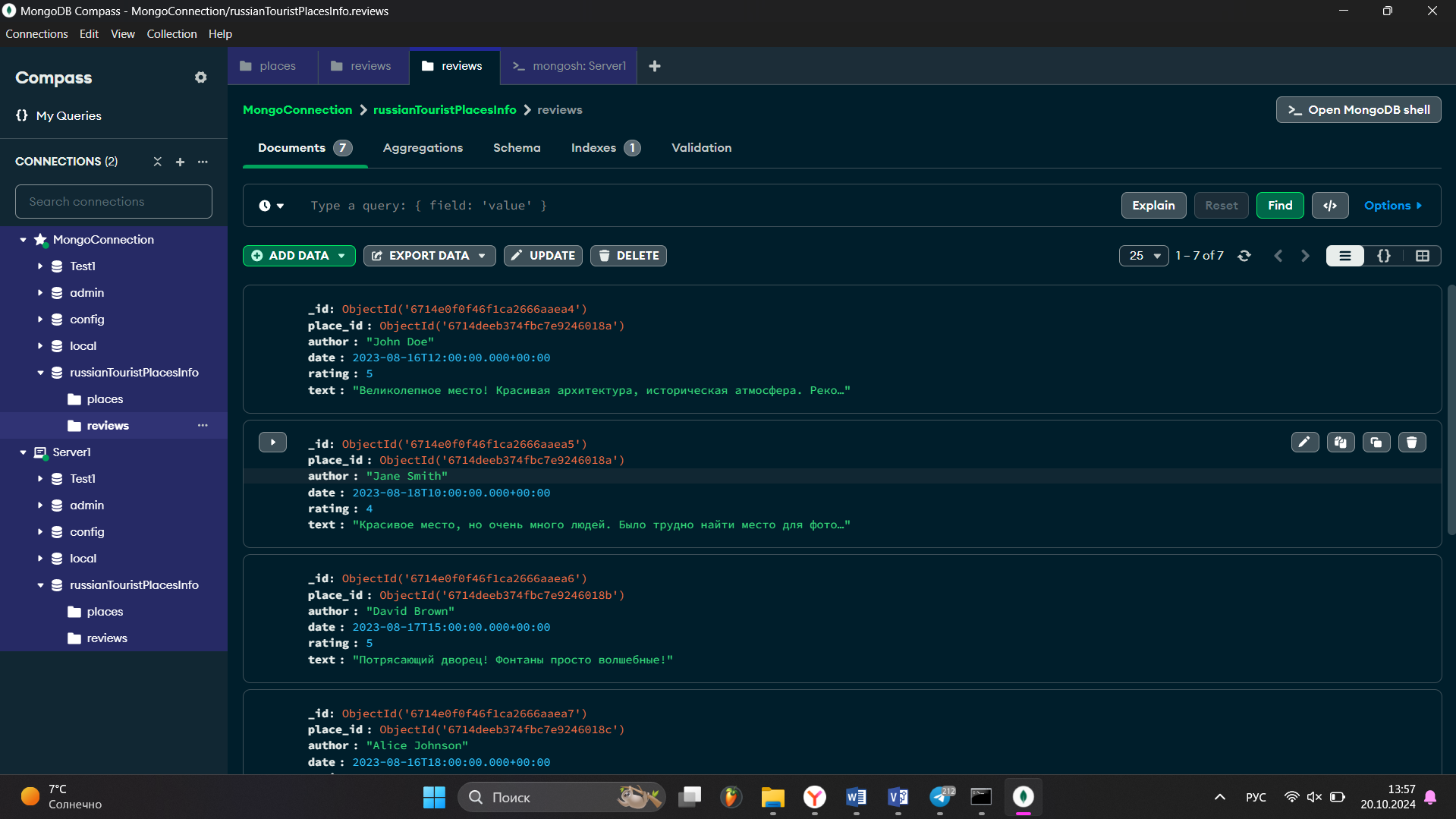
Скриншоты выполненного задания:

 Uml-диаграмма БД “russianTouristPlacesInfo”





Коллекция “places” (+документы в формате .json)



Коллекция “reviews” (+документы в формате .json)

Контрольные вопросы:

1. Виды диаграмм, показывающих схему БД?
2. Какие есть модели данных?
3. Какая модель данных используется в Mongo DB?

Ответы на контрольные вопросы:

Виды диаграмм, показывающих схему БД:

Entity Relationship Diagram (ERD): Показывает сущности (коллекции) и связи между ними;

Schema Diagram: Показывает структуру каждой коллекции, включая поля и типы данных.

Data Flow Diagram (DFD): Показывает поток данных между различными компонентами системы, включая БД.

Модели данных:

Реляционная модель: Данные хранятся в таблицах с определенными столбцами и связями между ними.

Документная модель: Данные хранятся в документах, которые представляют собой JSON-подобные структуры.

Графовая модель: Данные хранятся в виде узлов и ребер, представляющих связи между ними. Ключевая модель: Данные хранятся в виде ключей и значений.

Модель данных в MongoDB:

* + - * Документная модель. MongoDB использует JSON-подобные документы для хранения данных, и работает с коллекциями и документами внутри них.