МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Кемеровский государственный университет»

Институт цифры

**КУРСОВАЯ РАБОТА**

по дисциплине «Базы данных»

\_\_\_Информационная система автопредприятия города\_\_\_

(тема курсовой работы)

студента 2 курса

Земцева Данила Евгеньевича

направление подготовки 09.03.03 Прикладная информатика

направленность (профиль) подготовки «Прикладная информатика в экономике»

Научный руководитель:

кандидат технических наук, доцент

С.Ю. Завозкин

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Работа защищена с оценкой

«\_\_\_» (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

Кемерово 2024

**СОДЕРЖАНИЕ**

[1. Введение 3](#_Toc167718558)

[2. Программно-аппаратные средства, используемые при выполнении работы. 4](#_Toc167718568)

[3. Основная часть. 5](#_Toc167718574)

[3.1 Прототипирование базы данных 5](#_Toc167718575)

[3.2 Создание БД 6](#_Toc167718576)

[3.2.1 Создание коллекций 6](#_Toc167718577)

[3.2.2 Наполнение коллекций 7](#_Toc167718578)

[3.3 Запросы 12](#_Toc167718579)

[Заключение 22](#_Toc167718580)

[Список используемой литературы. 23](#_Toc167718584)

# Введение

# Современные городские автопредприятия играют ключевую роль в обеспечении мобильности и транспортной доступности для населения, обслуживая как пассажирские, так и грузоперевозочные потребности. Управление такими предприятиями требует комплексного подхода, который включает в себя эффективное управление автотранспортом, персоналом и техническим обслуживанием. В этом контексте разработка и внедрение информационных систем становится необходимостью для повышения операционной эффективности и качества предоставляемых услуг.

# Целью данной работы является создание информационной системы для автопредприятия города с целью оптимизации управления автотранспортом, персоналом и техническим обслуживанием.

# Для достижения этой цели поставлены следующие задачи:

# Разработка базы данных, включающей информацию о различных категориях автотранспорта (автобусы, такси, грузовые машины и др.), их характеристиках, а также о персонале и техническом оборудовании предприятия.

# Создание интерфейса для ввода и отображения данных, обеспечивающего удобное взаимодействие с системой для персонала предприятия.

# Реализация функционала для учета и планирования маршрутов для пассажирского транспорта с учетом интенсивности использования и потребностей пассажиров.

# Разработка инструментов аналитики для мониторинга и оптимизации процессов технического обслуживания автотранспорта, включая учет затрат на ремонт и замену деталей.

# Обеспечение безопасности и конфиденциальности данных, хранящихся в системе, с помощью соответствующих механизмов защиты информации.

# Решение поставленных задач позволит создать функциональную и эффективную информационную систему, способствующую оптимизации управления автопредприятием и повышению качества предоставляемых услуг городскому населению.

# Программно-аппаратные средства, используемые при выполнении работы.

# Для выполнения работы были использованы следующие программно-аппаратные средства:

# MongoSH: Интерактивная оболочка командной строки, предоставляющая возможность выполнения команд MongoDB напрямую из терминала. Этот инструмент упрощает взаимодействие с базой данных и выполнение различных операций.

# MongoDB Compass: Графический интерфейс пользователя для работы с MongoDB. MongoDB Compass предоставляет удобный способ создания, редактирования и просмотра документов в базе данных, а также выполнения запросов и анализа данных.

# NoSQLBooster: Еще один инструмент для администрирования MongoDB, предоставляющий широкий набор функций для работы с базой данных. NoSQLBooster включает в себя редактор запросов, инструменты анализа данных и возможности импорта и экспорта данных.

# Для редактирования коллекций при их добавлении был использован Visual Studio Code (VSCode) - мощный редактор кода с широким набором функций и расширений. VSCode предоставляет удобный интерфейс для работы с файлами JSON, что позволяет легко создавать и редактировать документы базы данных в формате JSON.

# Основная часть.

# 3.1 Прототипирование базы данных

На основе полученного задания, была проанализирована предметная область и построена концептуальная схема коллекций для создания базы данных:

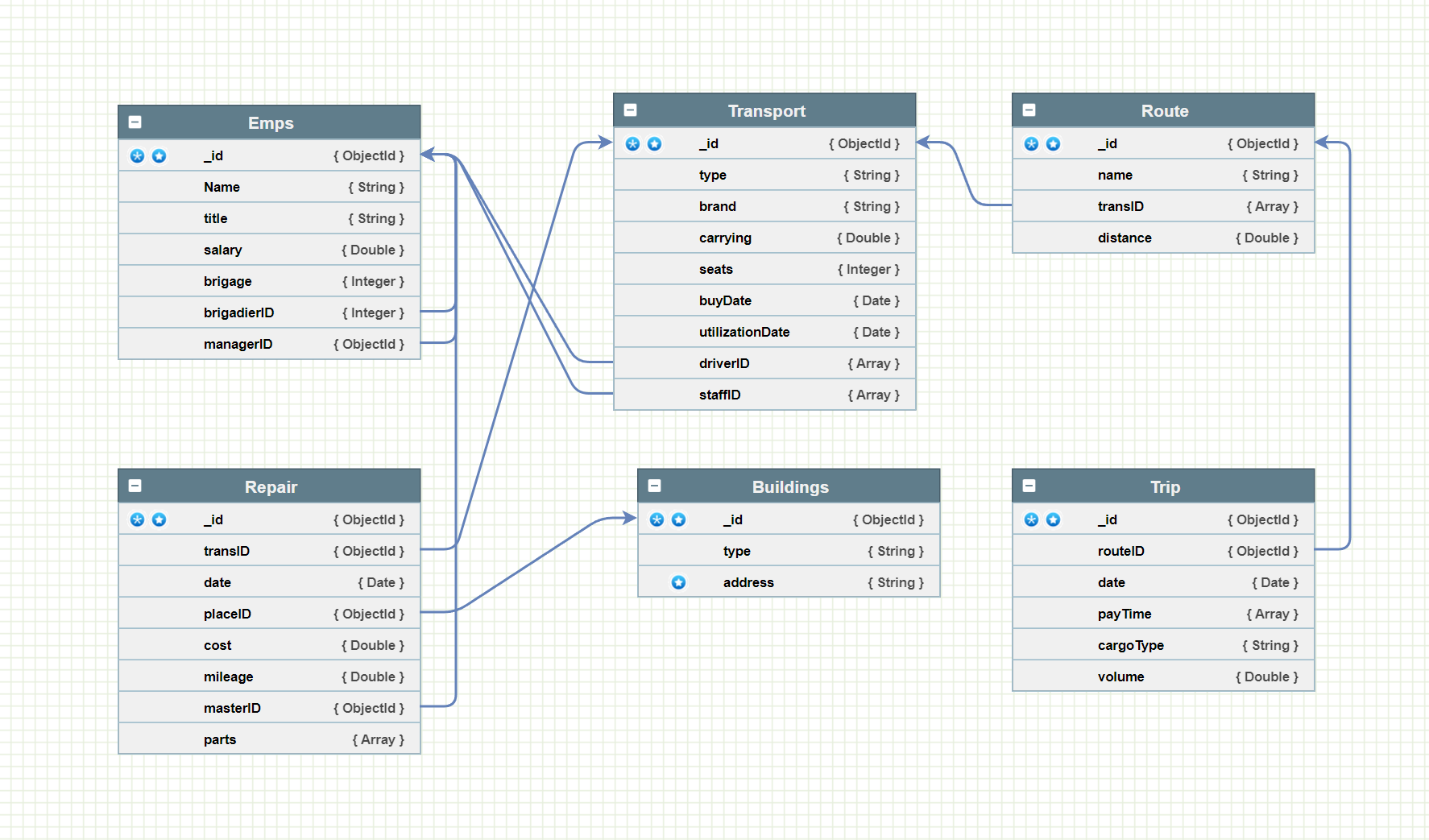


Рис. 1 – Схема коллекций

# 3.2 Создание БД

# 3.2.1 Создание коллекций

Для создания базы данных мы подключились к локальному серверу MongoDB в ПО MongoDBCompass (рис. 2) и создали базу данных под названием SemesterWork (рис.3).

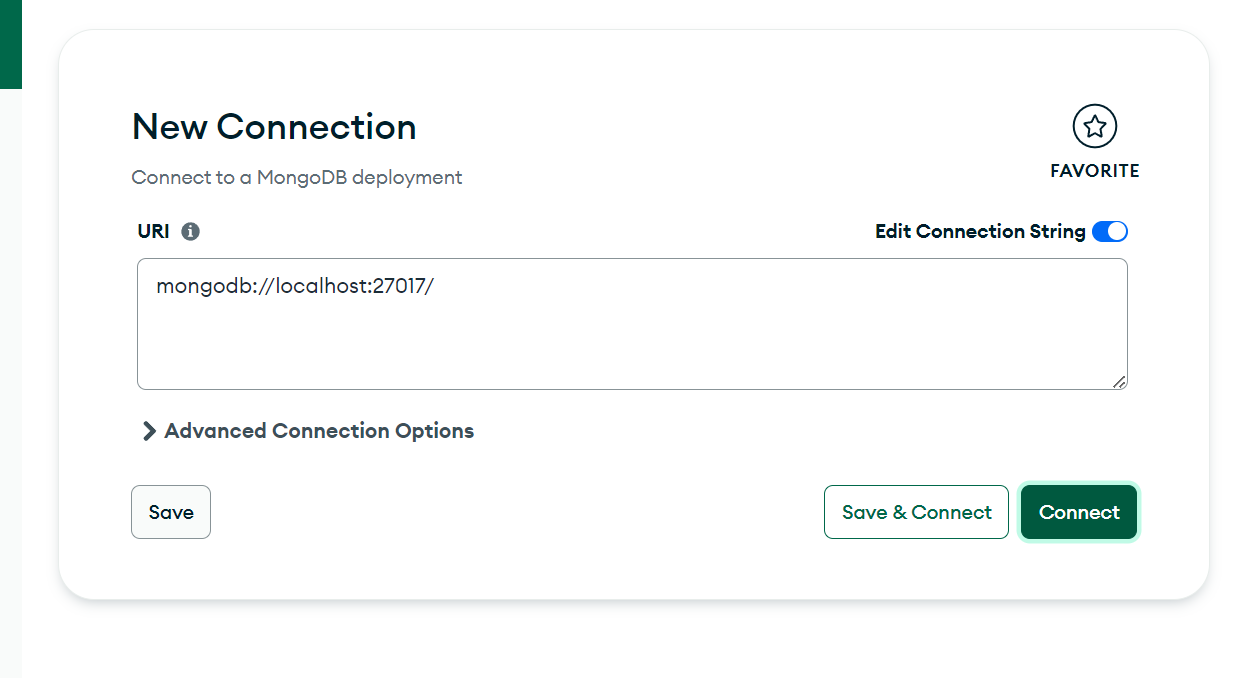


Рис. 2 – Подключение к локальному серверу БД

Далее мы создали необходимые коллекции (рис. 3), согласно ранее созданной схеме (рис. 1).

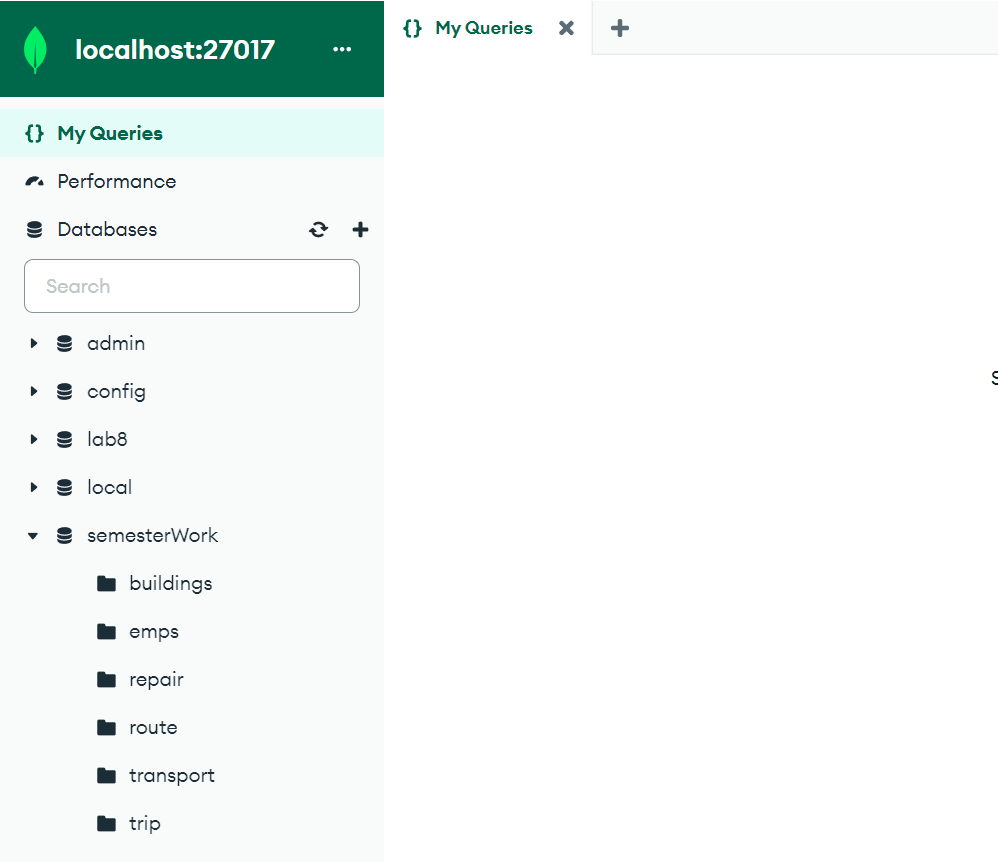


Рис. 3 – Созданные коллекции

# 3.2.2 Наполнение коллекций

В качестве метода наполнения коллекций мы выбрали функцию “load(“путь к файлу”)”.

Следовательно, в программной среде разработки VSCode были написаны скрипты JavaScript для наполнения каждой коллекции, используя функцию: “db.”название коллекции”.insertMany([”наполнение для каждого документа”])” (рис.4 – 9)

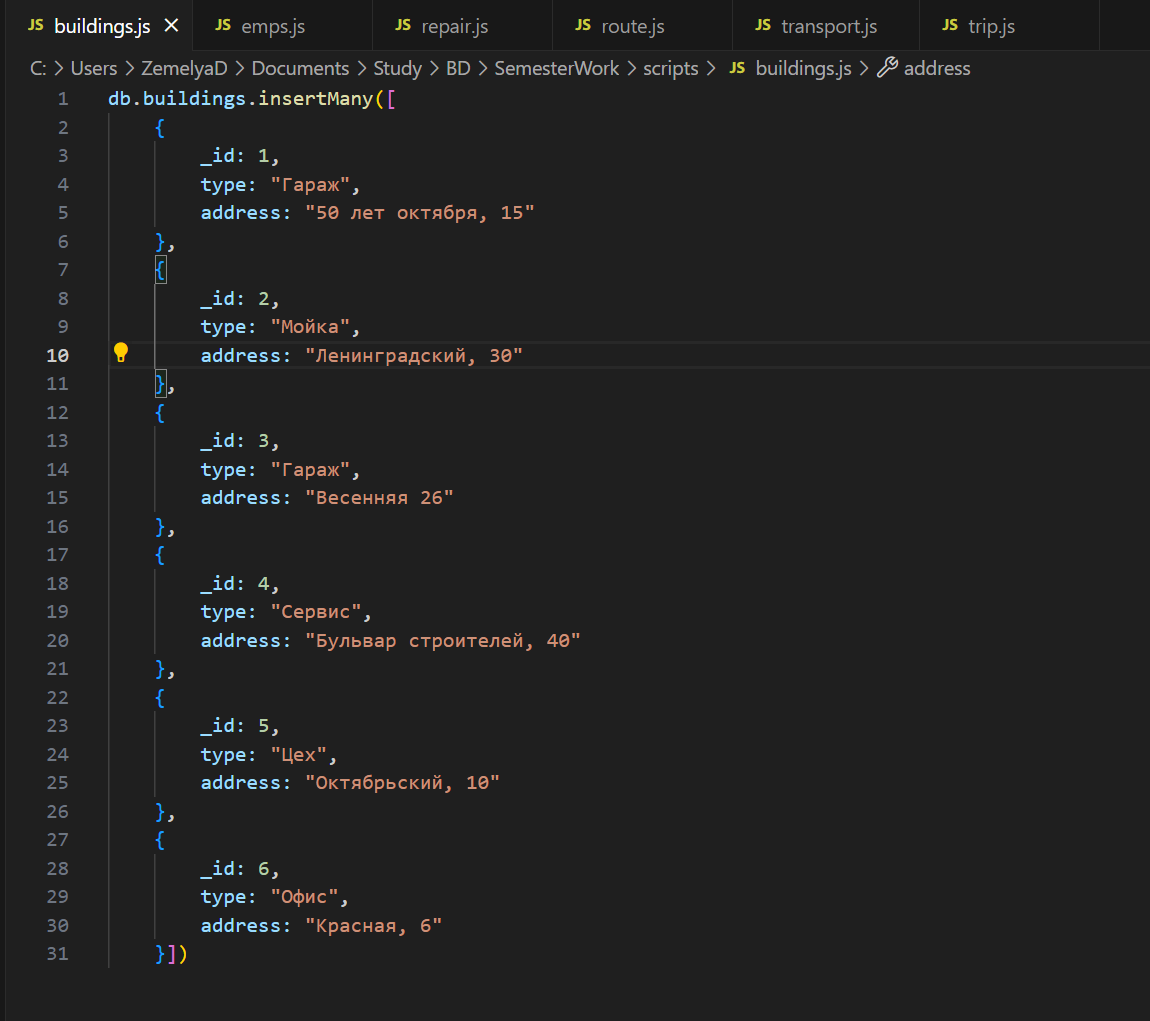


Рис. 4 – Наполнение коллекции buildings

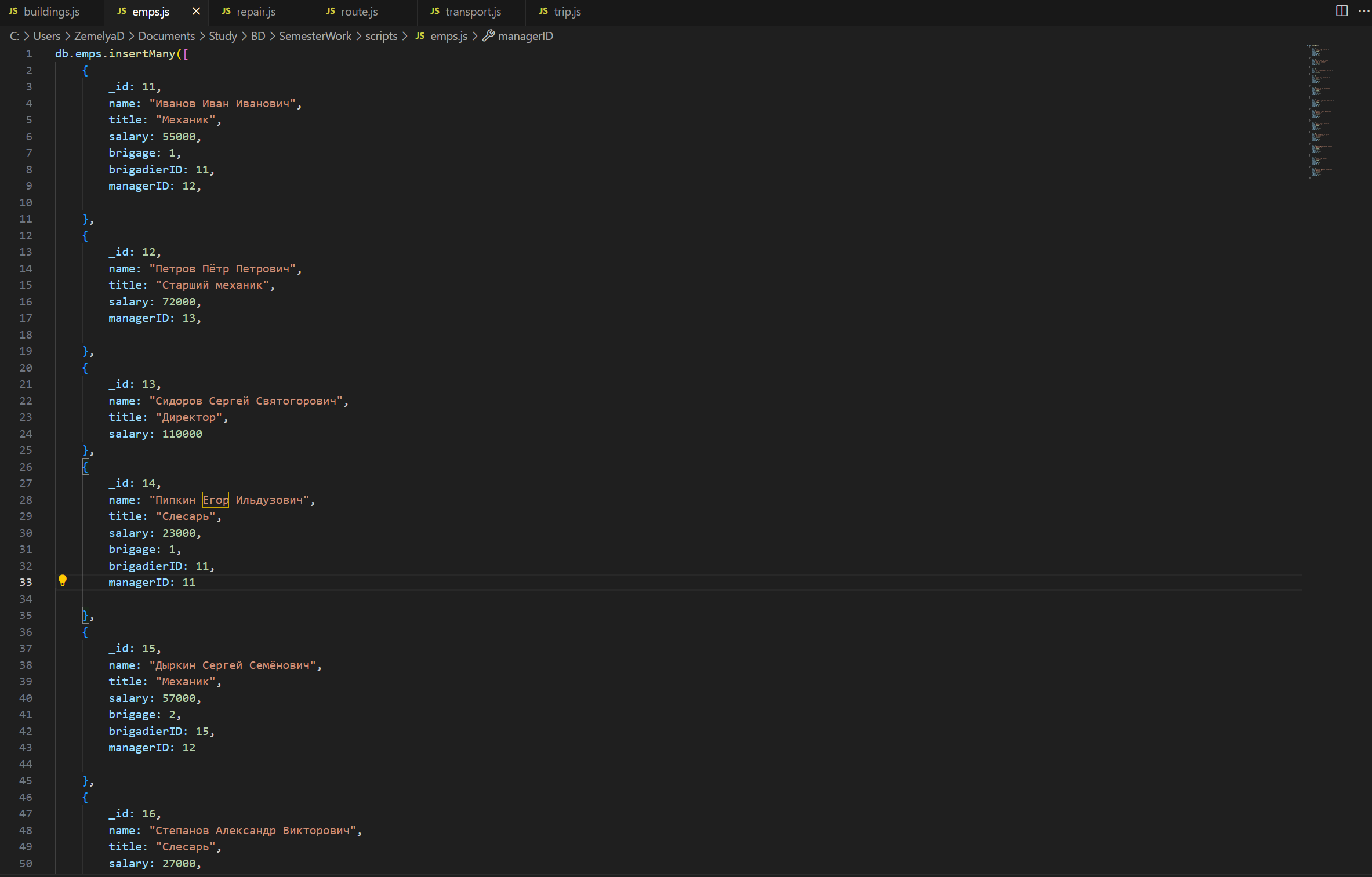


Рис. 5 – Наполнение коллекции emps

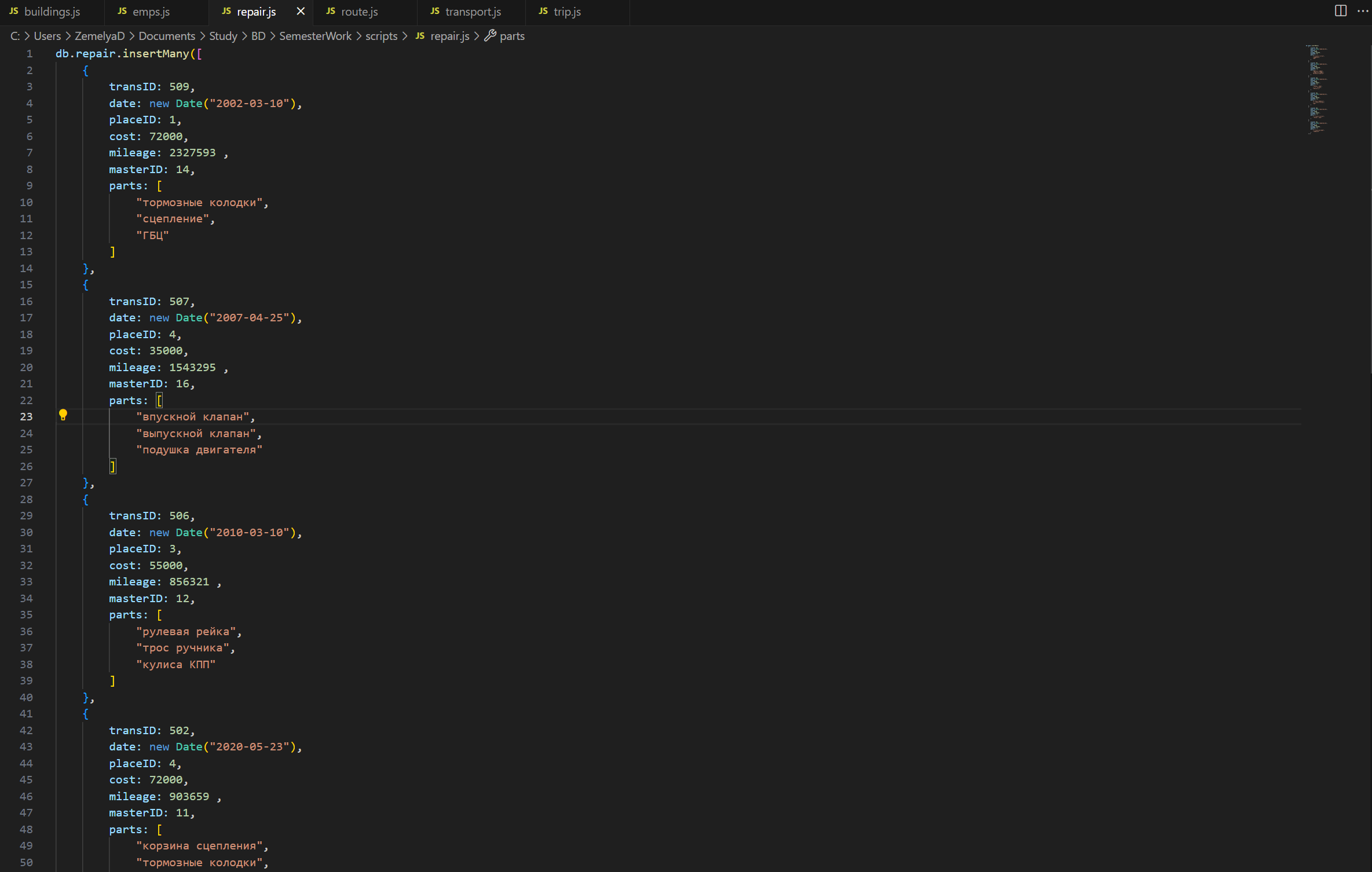


Рис. 6 – Наполнение коллекции repair

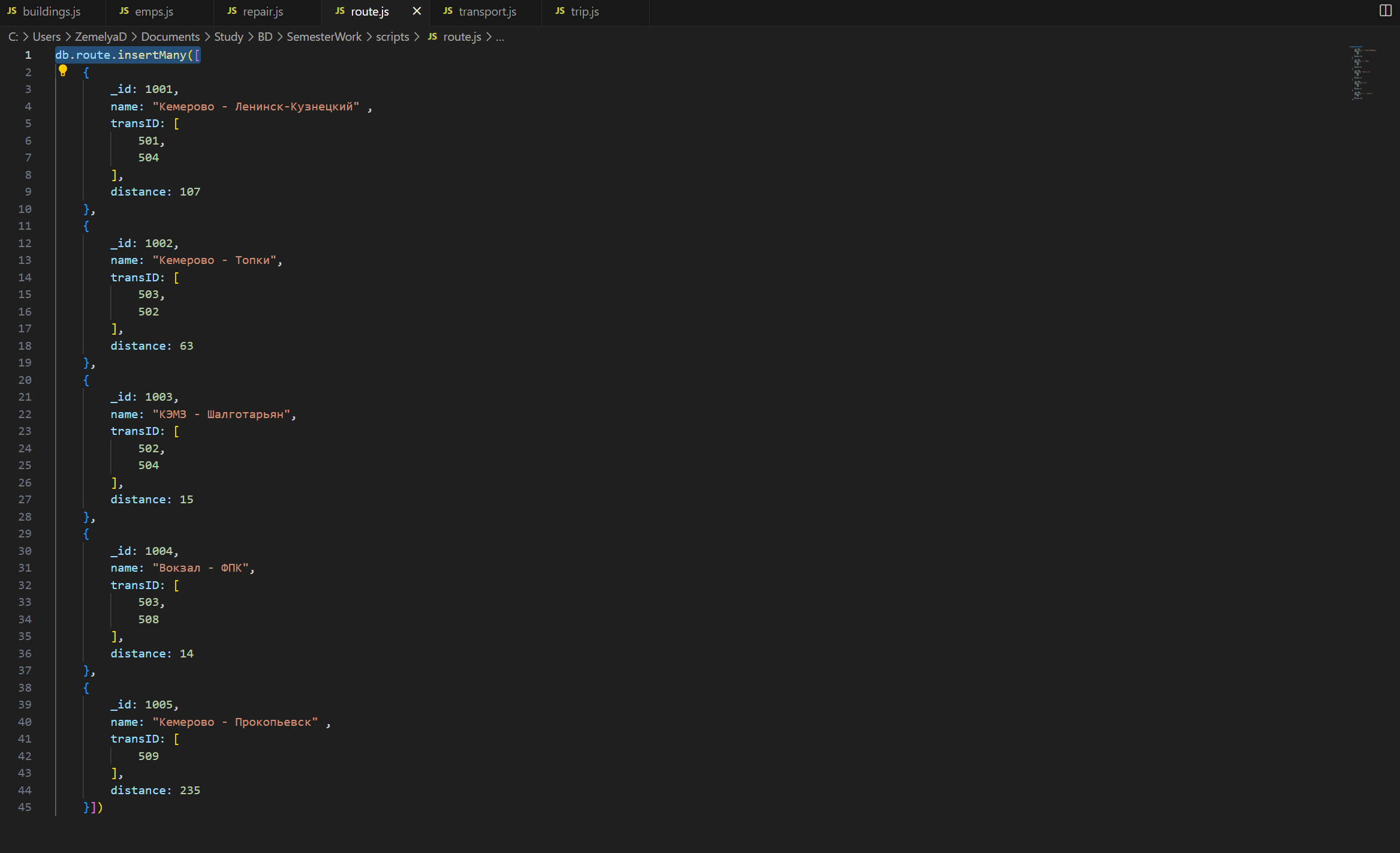


Рис. 7 – Наполнение коллекции route

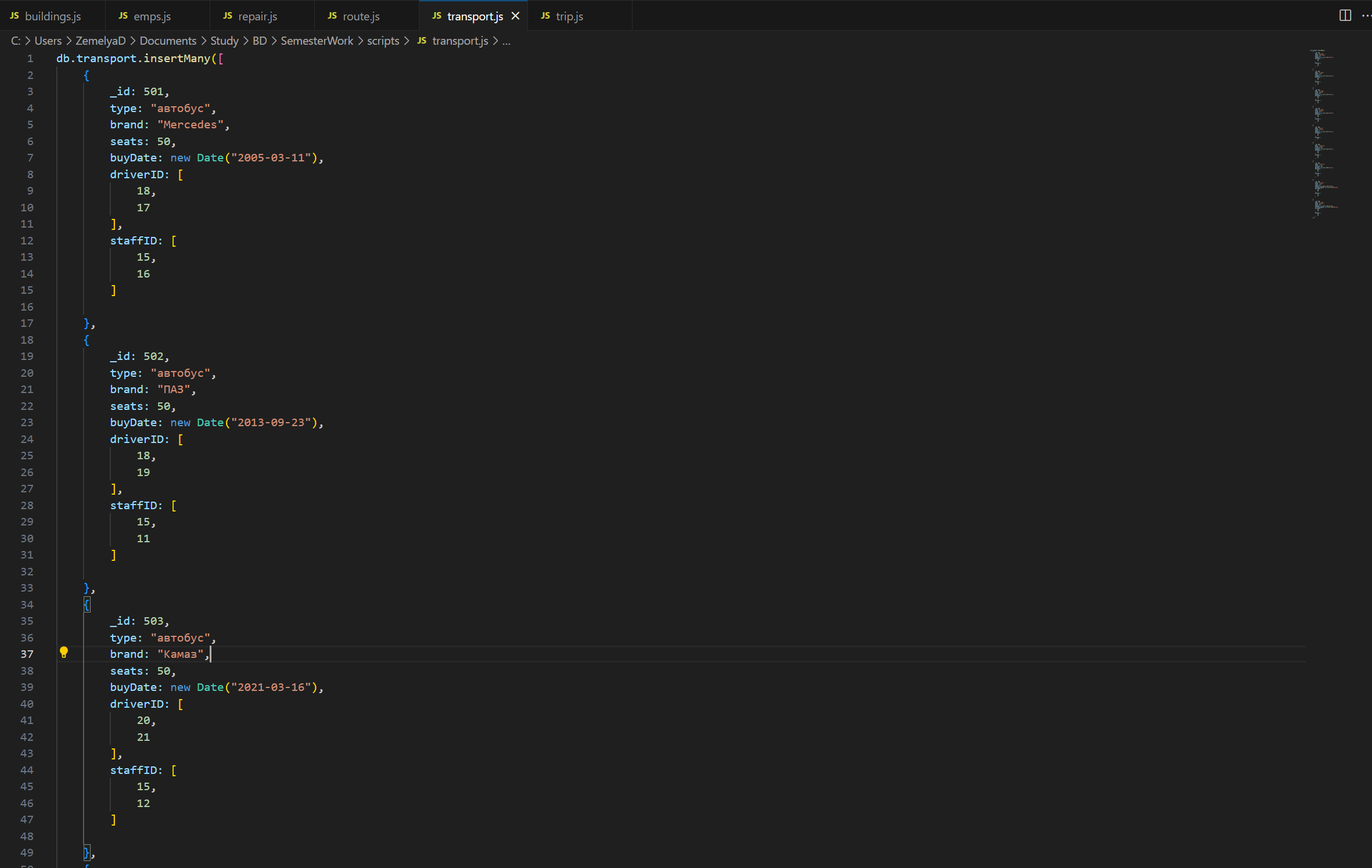
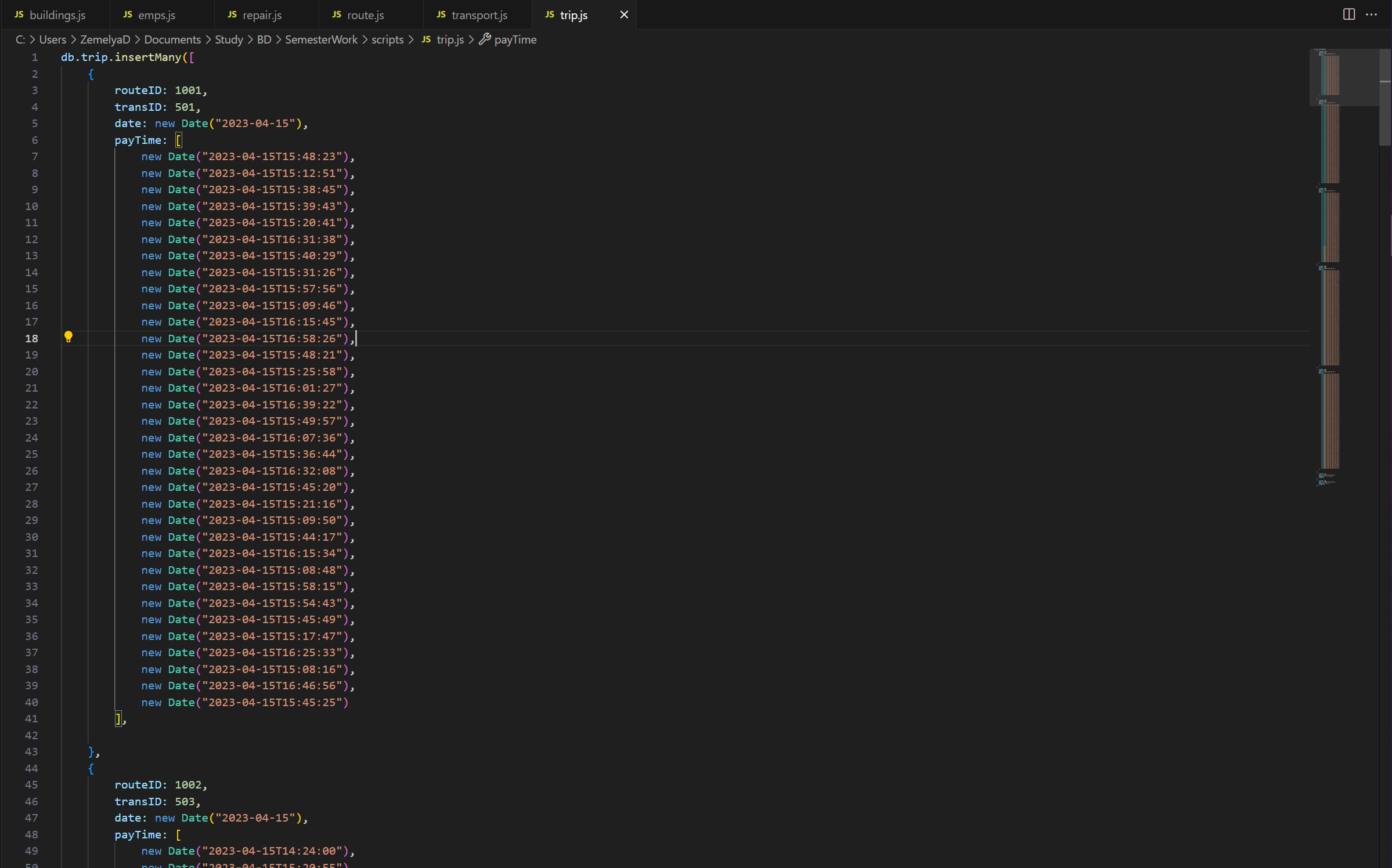


Рис. 8 – Наполнение коллекции transport

 Рис. 9 – Наполнение коллекции trip

Далее, все полученные файлы \*.js были загружены в соответствующие коллекции (рис. 10)

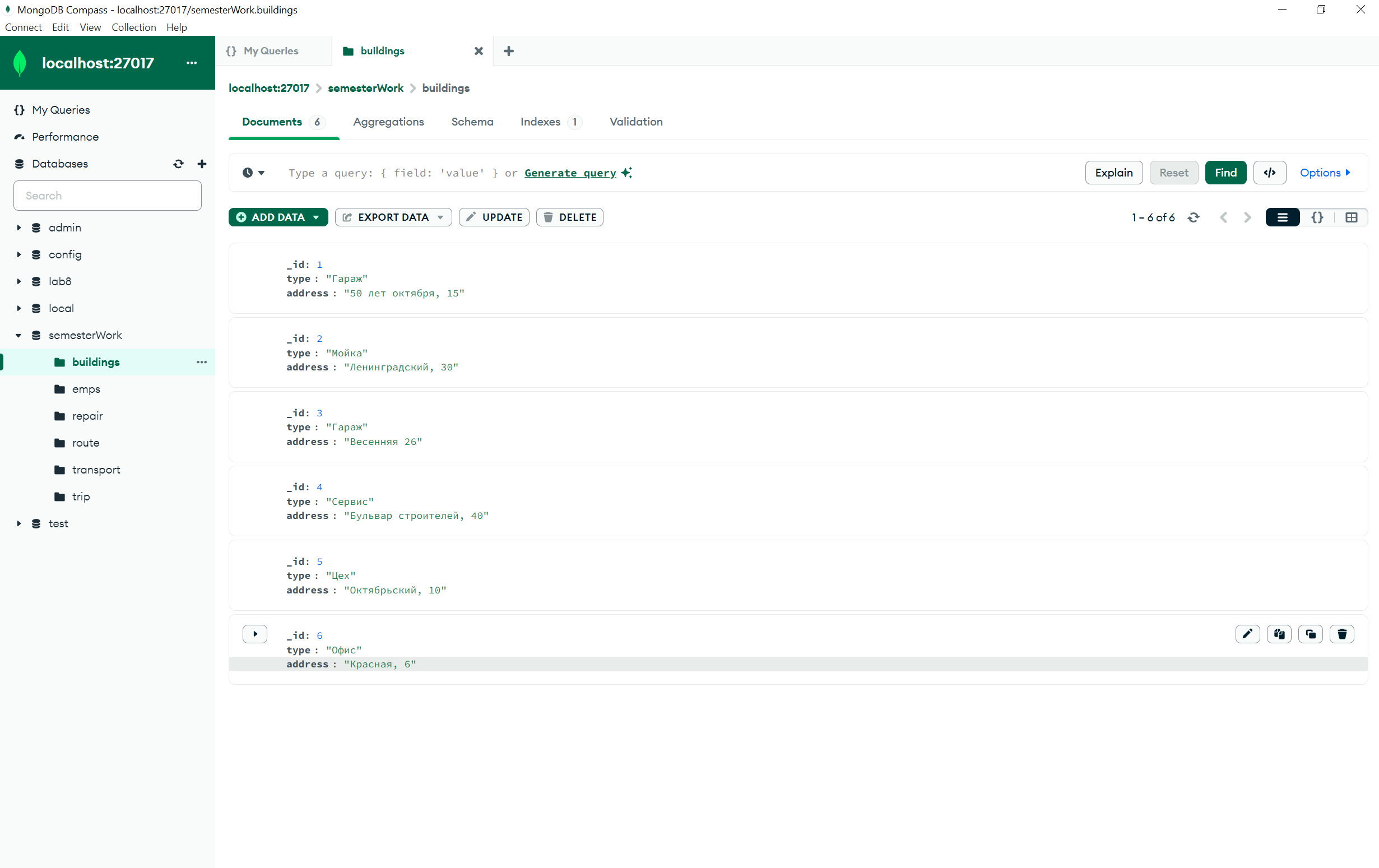


Рис. 10 – Пример заполненной коллекции buildings

# 3.3 Запросы

Согласно полученному заданию нам необходимо выполнить запросы к созданной базе данных:

3.3.1 . Получили перечень и общее число водителей по предприятию, по указанной автомашине.

В качестве автомобиля, для выполнения запроса был выбран автобус под номером 502.

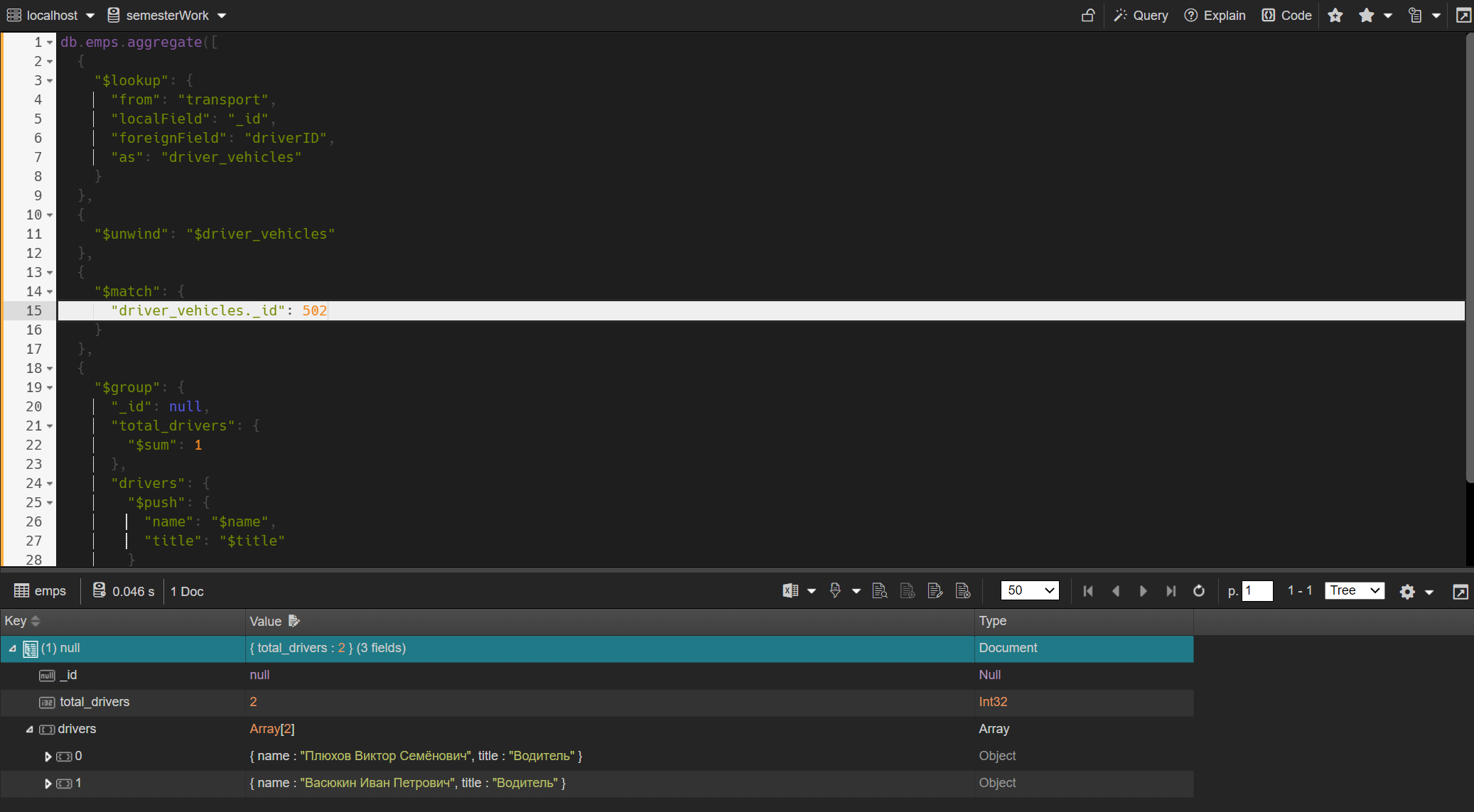


Рис. 11 – Результаты выполнения запроса

Код:

db.emps.aggregate([{

"$lookup": {

"from": "transport",

"localField": "\_id",

"foreignField": "driverID",

"as": "driver\_vehicles"

}},

{"$unwind": "$driver\_vehicles"},

{"$match": {

"driver\_vehicles.\_id": 502}},

{"$group": {

"\_id": null,

"total\_drivers": {

"$sum": 1},

"drivers": {

"$push": {

"name": "$name",

"title": "$title"}}}}])

3.3.2 Получили распределение водителей по автомобилям.

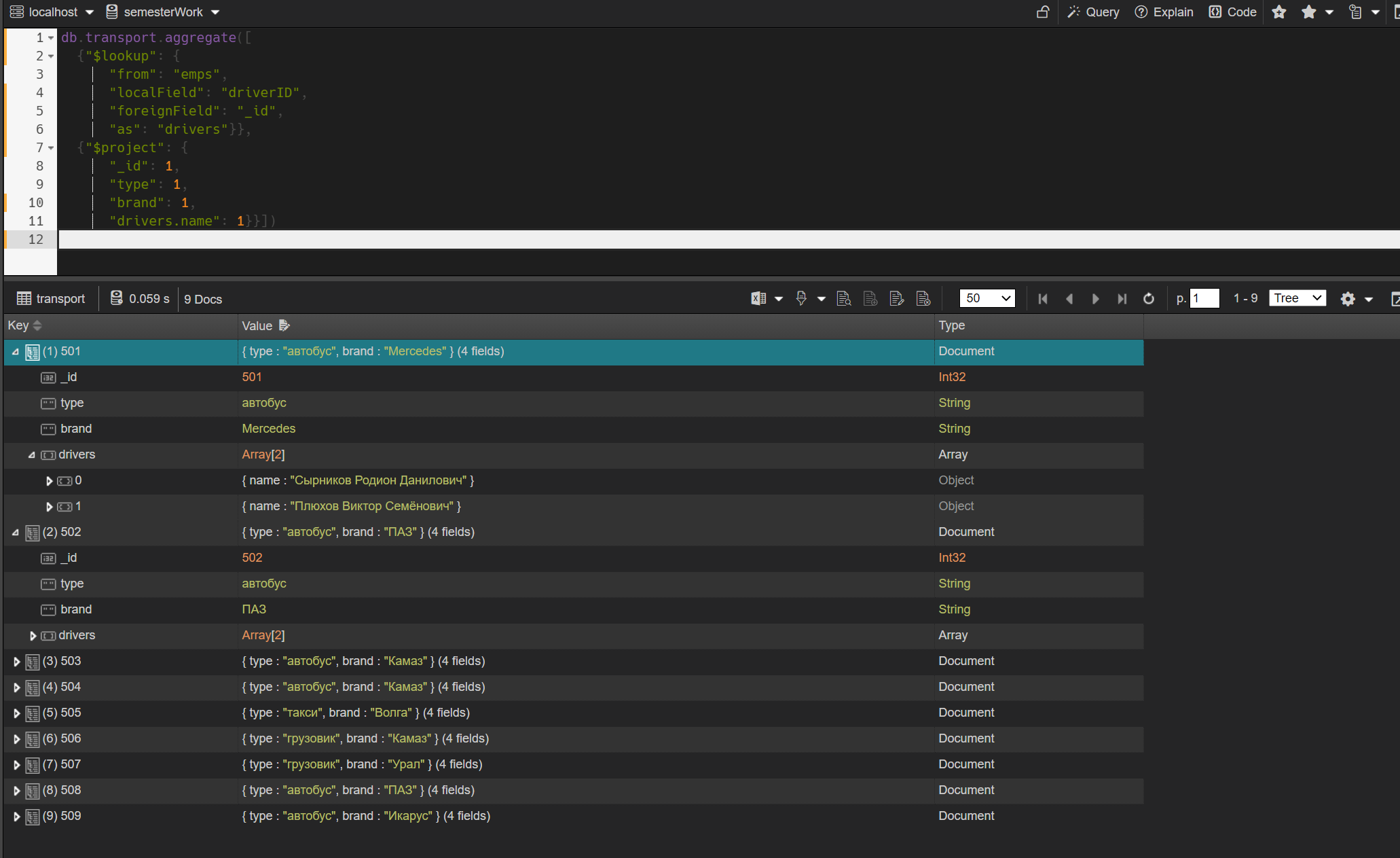


Рис. 12 – Результаты выполнения запроса

Код:

db.transport.aggregate([

{"$lookup": {

"from": "emps",

"localField": "driverID",

"foreignField": "\_id",

"as": "drivers"}},

{"$project": {

"\_id": 1,

"type": 1,

"brand": 1,

"drivers.name": 1}}])

3.3.3 Получили сведения о пробеге автотранспорта определенной категории или конкретной автомашины за указанный день, месяц, год.

Был выбран автомобиль №505 В качестве даты была выбрана дата обслуживания автотранспорта, т.е 10.03.2017

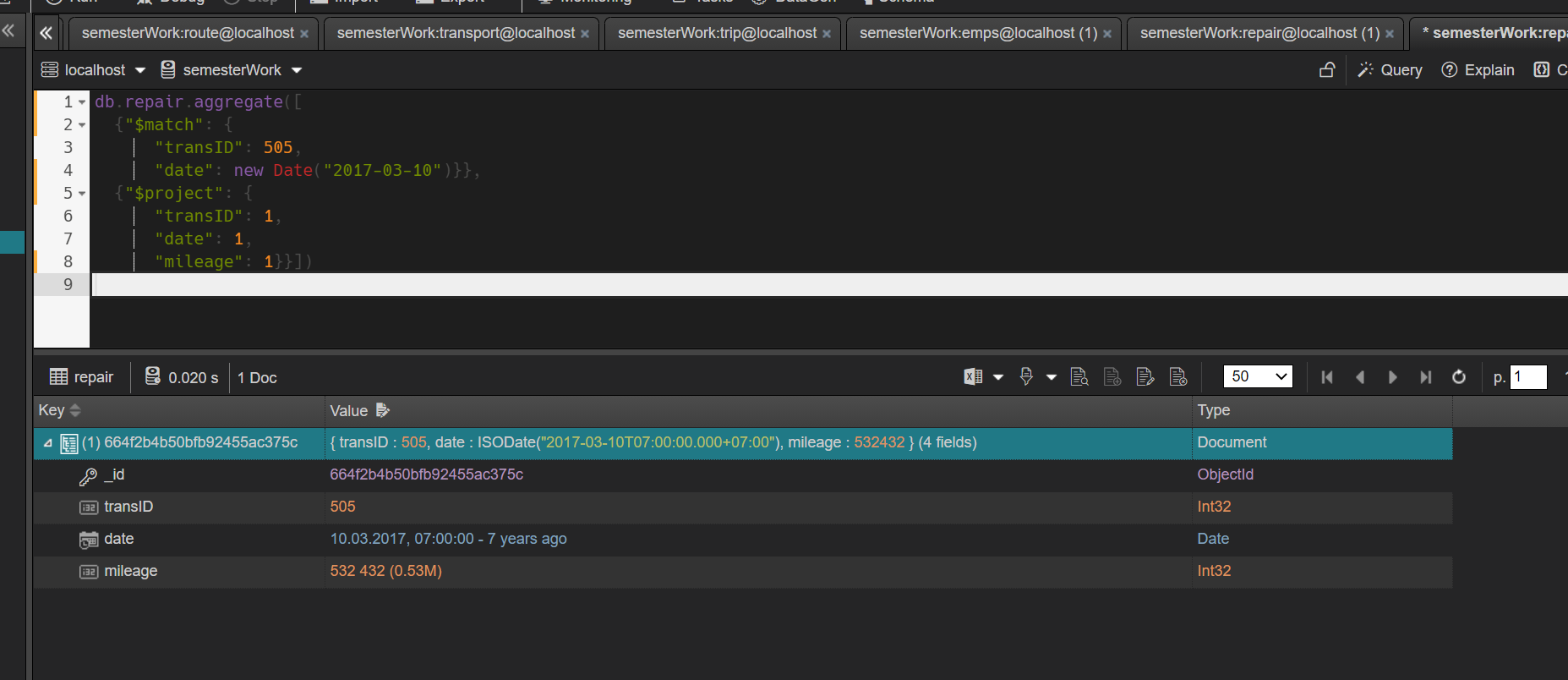


Рис. 13 – Результаты выполнения запроса

Код:

db.repair.aggregate([

{"$match": {

"transID": 505,

"date": new Date("2017-03-10")}},

{"$project": {

"transID": 1,

"date": 1,

"mileage": 1}}])

3.3.4 Получили данные о числе ремонтов и их стоимости для автотранспорта определенной категории, отдельной марки автотранспорта или указанной автомашины за указанный период.

В качестве автомобиля были выбраны автомобилей категории «автобус» и интервал ремонтов по ней с 02.04.2016 по 10.08.2020

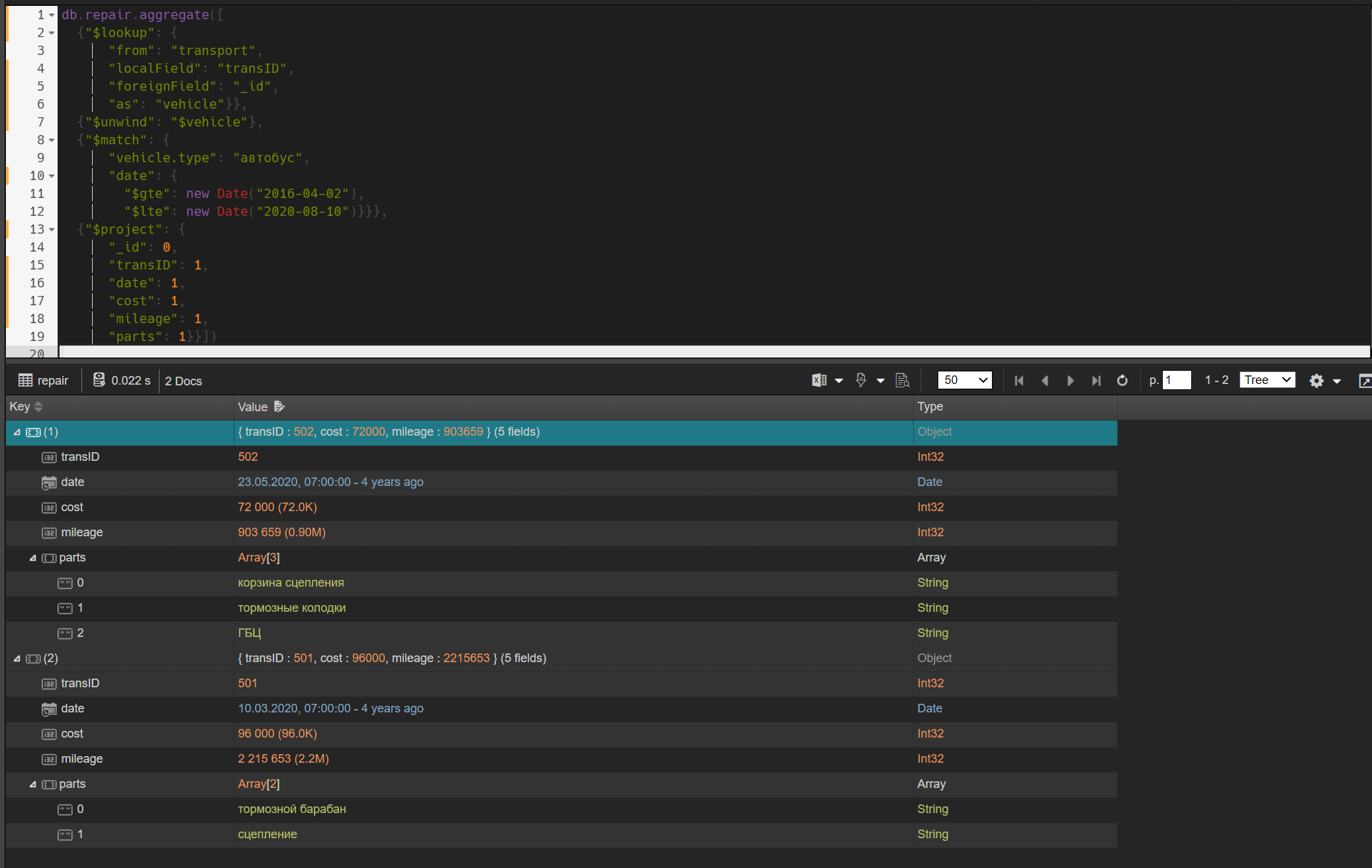


Рис. 14 – Результаты выполнения запроса

Код:

db.repair.aggregate([

{"$lookup": {

"from": "transport",

"localField": "transID",

"foreignField": "\_id",

"as": "vehicle"}},

{"$unwind": "$vehicle"},

{"$match": {

"vehicle.type": "автобус",

"date": {

"$gte": new Date("2016-04-02"),

"$lte": new Date("2020-08-10")}}},

{"$project": {

"\_id": 0,

"transID": 1,

"date": 1,

"cost": 1,

"mileage": 1,

"parts": 1}}])

3.3.5 Получить данные о подчиненности персонала.

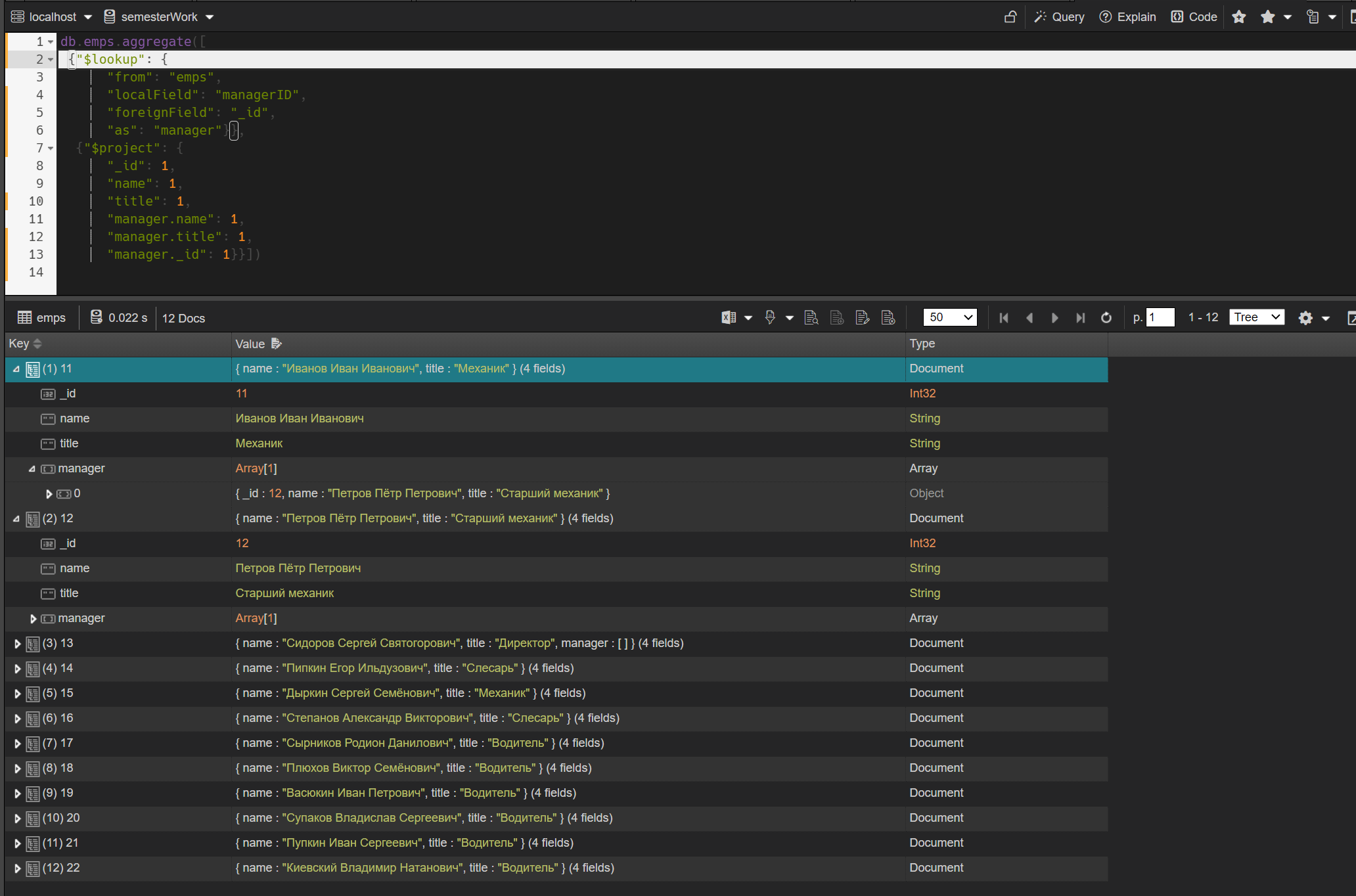


Рис. 15 – Результаты выполнения запроса

Код:

db.emps.aggregate([

{"$lookup": {

"from": "emps",

"localField": "managerID",

"foreignField": "\_id",

"as": "manager"}},

{"$project": {

"\_id": 1,

"name": 1,

"title": 1,

"manager.name": 1,

"manager.title": 1,

"manager.\_id": 1}}])

3.3.6 Получили данные о распределении автотранспорта на предприятии.

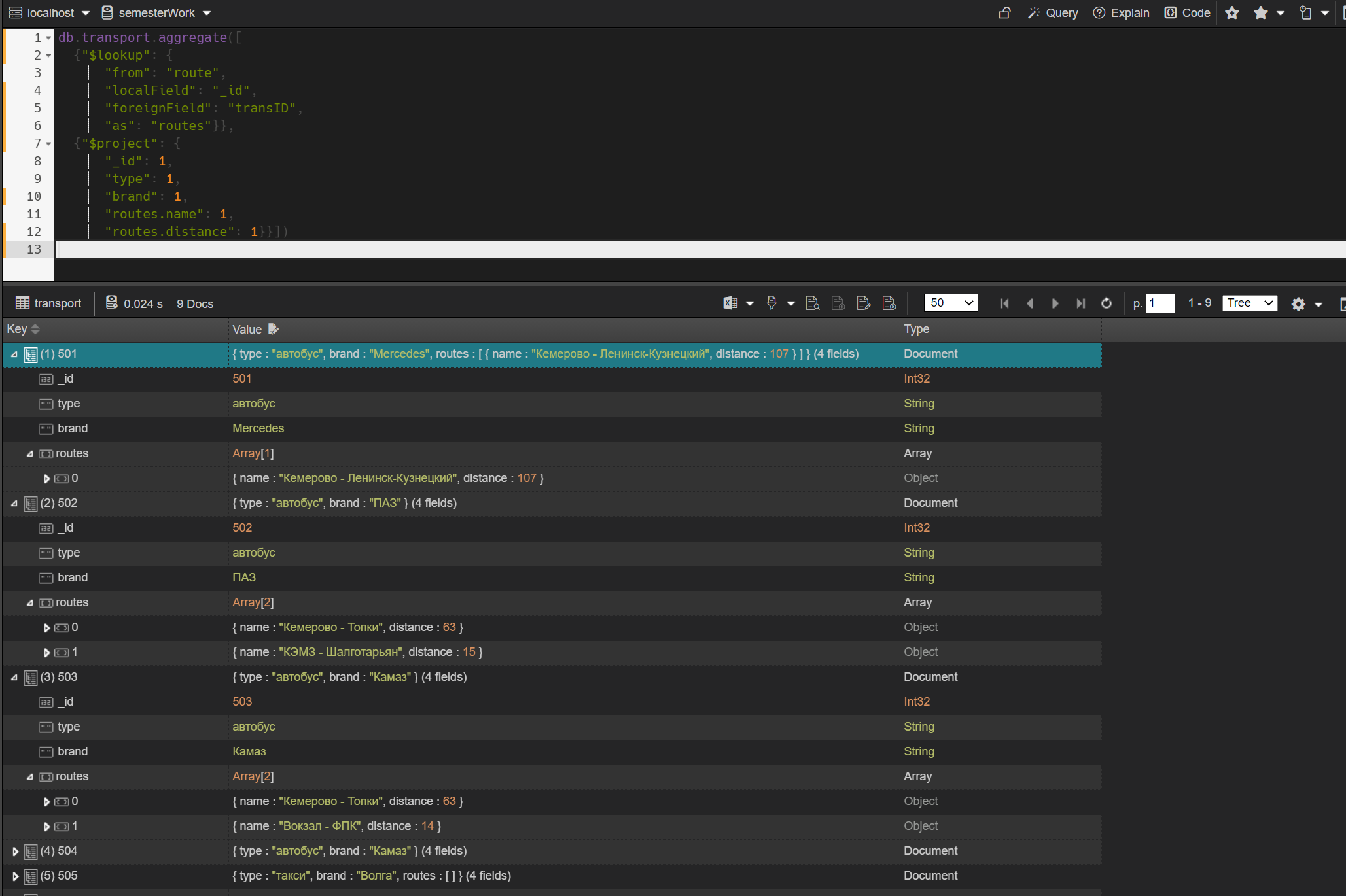


Рис. 16 – Результаты выполнения запроса

Код:

db.transport.aggregate([

{"$lookup": {

"from": "route",

"localField": "\_id",

"foreignField": "transID",

"as": "routes"}},

{"$project": {

"\_id": 1,

"type": 1,

"brand": 1,

"routes.name": 1,

"routes.distance": 1}}])

3.3.7 Получили сведения о грузоперевозках, выполненных указанной автомашиной за обозначенный период. Был выбран автомобиль под номером 507 и интервал с 01.01.2023 по 15.10.2023

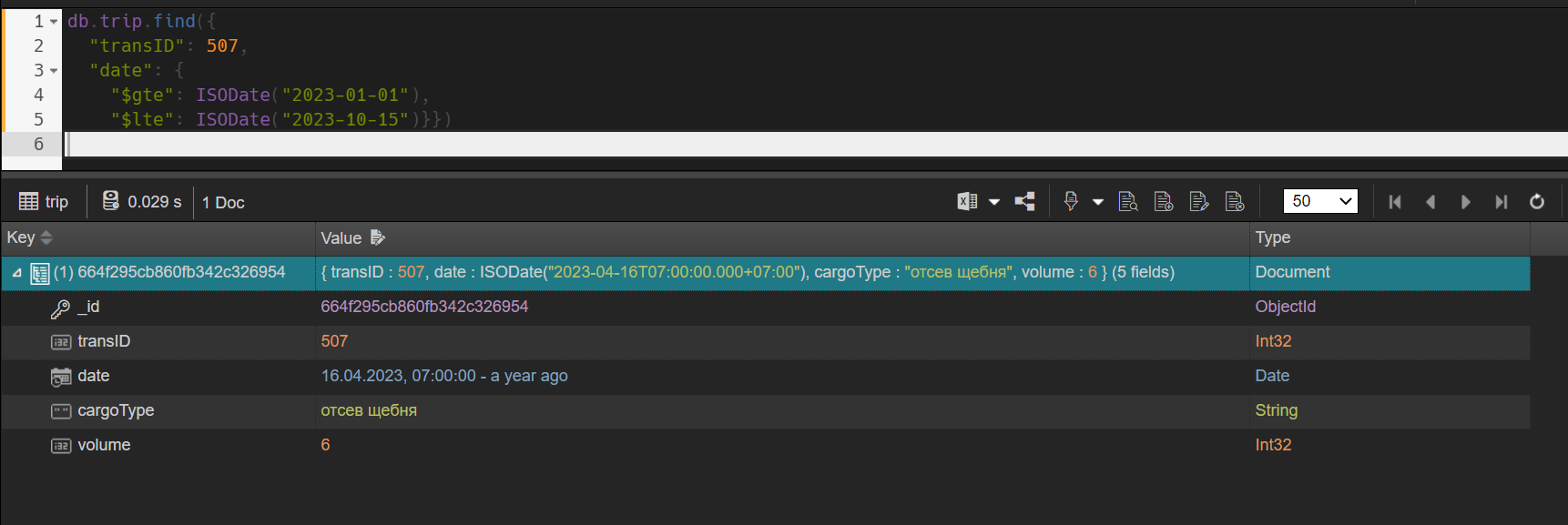


Рис. 17 – Результаты выполнения запроса

Код:

db.trip.find({

"transID": 507,

"date": {

"$gte": ISODate("2023-01-01"),

"$lte": ISODate("2023-10-15")}})

3.3.8 Получили данные о числе использованных для ремонта указанных узлов и агрегатов для транспорта определенной категории, отдельной марки автотранспорта или конкретной автомашины за указанный период.

В качестве ремонтируемого узла была выбрана запасная часть «тормозные колодки», для автомобилей категории «автобус» , в период с 01.01.2001 по 20.06.2010

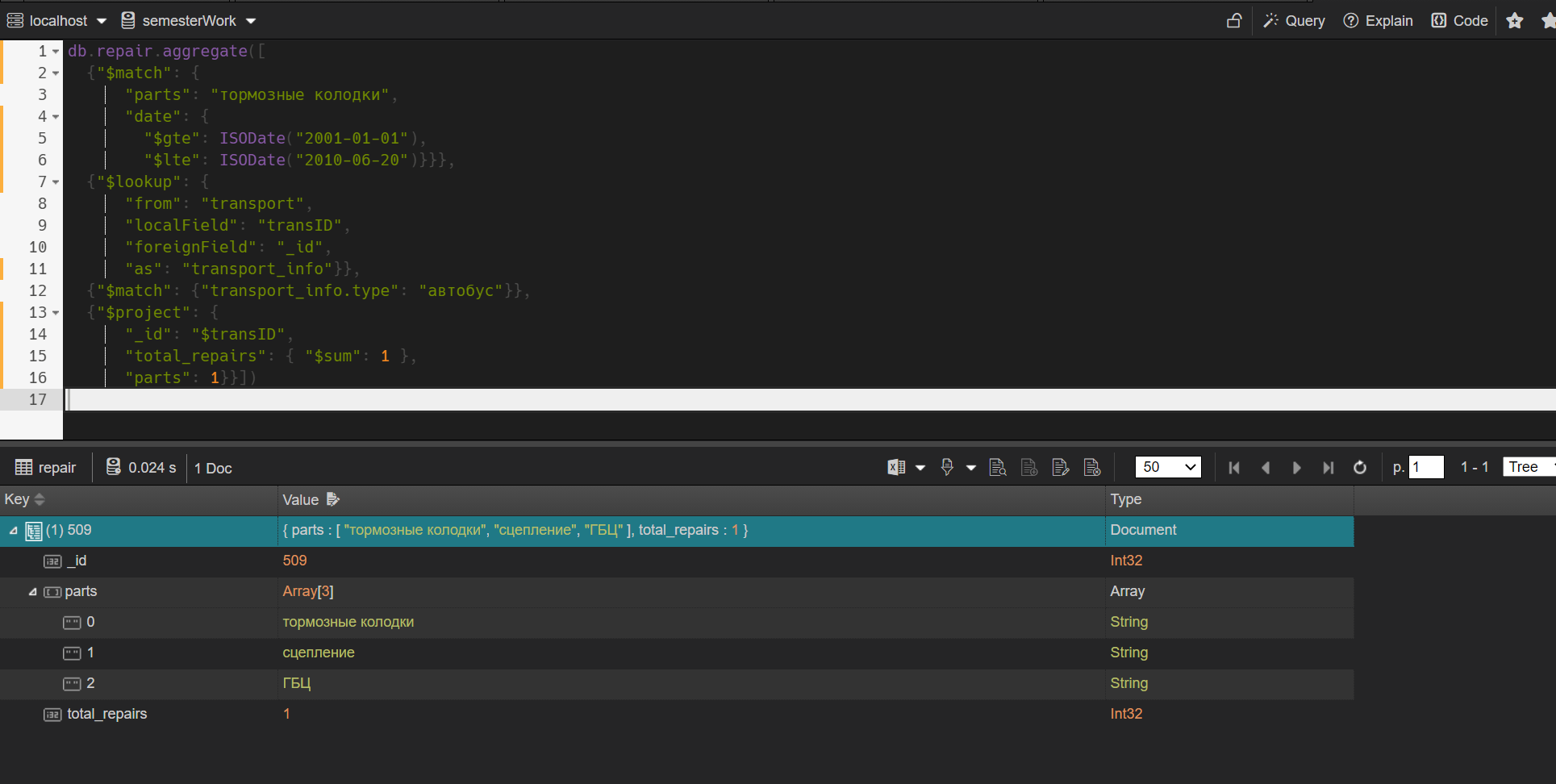


Рис. 18 – Результаты выполнения запроса

Код:

db.repair.aggregate([

{"$match": {

"parts": "тормозные колодки",

"date": {

"$gte": ISODate("2001-01-01"),

"$lte": ISODate("2010-06-20")}}},

{"$lookup": {

"from": "transport",

"localField": "transID",

"foreignField": "\_id",

"as": "transport\_info"}},

{"$match": {"transport\_info.type": "автобус"}},

{"$project": {

"\_id": "$transID",

"total\_repairs": { "$sum": 1 },

"parts": 1}}])

3.3.9 Получили сведения о полученной и списанной автотехники за указанный период.

В качестве периода был выбран временной интервал с 01.01.2005 по 10.12.2015

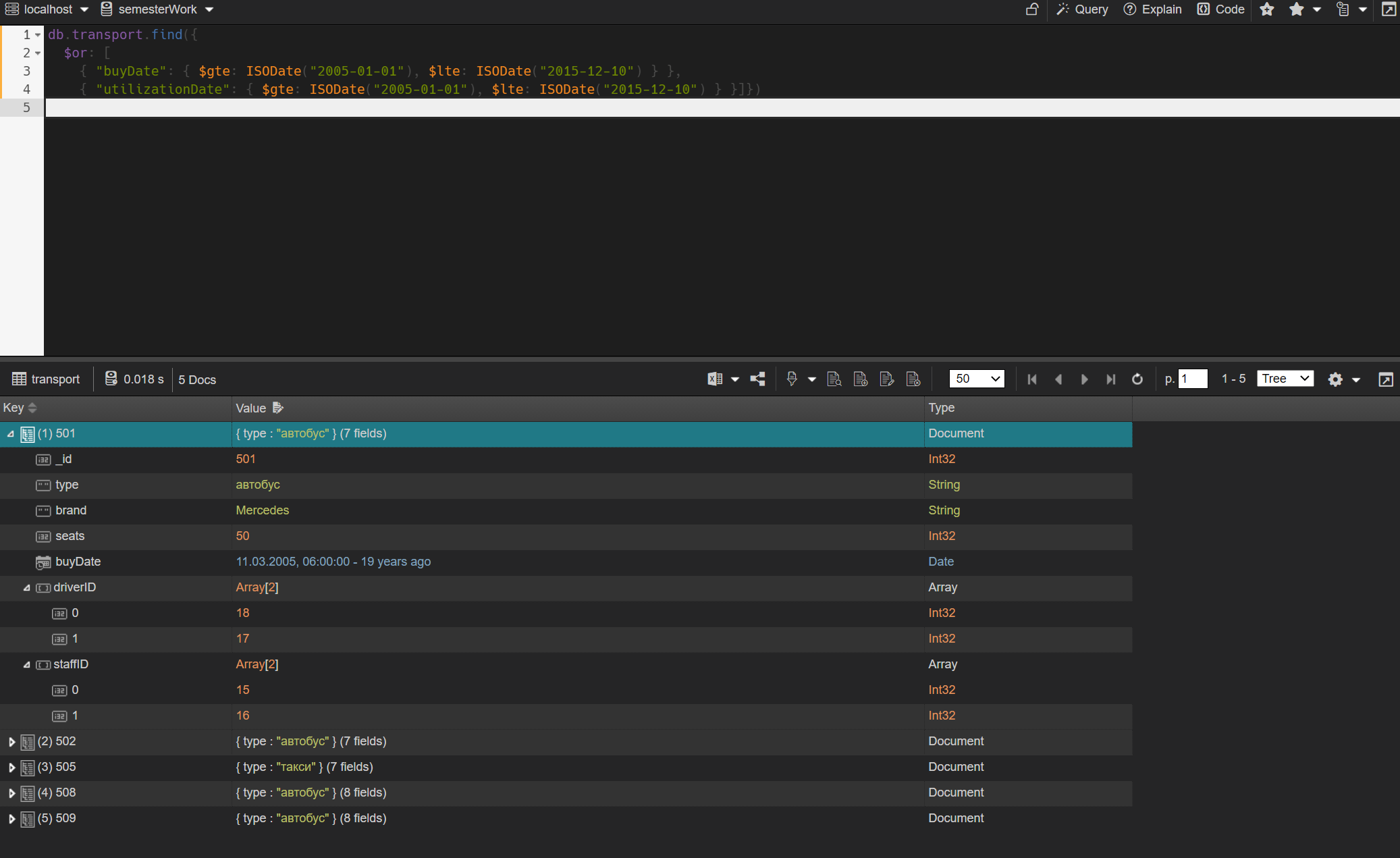


Рис. 19 – Результаты выполнения запроса

Код:

db.transport.find({

$or: [

{ "buyDate": { $gte: ISODate("2005-01-01"), $lte: ISODate("2015-12-10") } },

{ "utilizationDate": { $gte: ISODate("2005-01-01"), $lte: ISODate("2015-12-10") } }]})

3.3.10 Получили данные о работах, выполненных указанным специалистом (сварщиком, слесарем и т.д.) за обозначенный период в целом и по конкретной автомашине.

Был выбрана должность «Механик», автомобиль с номером 502 и временной интервал с 01.01.2015 по текущую дату

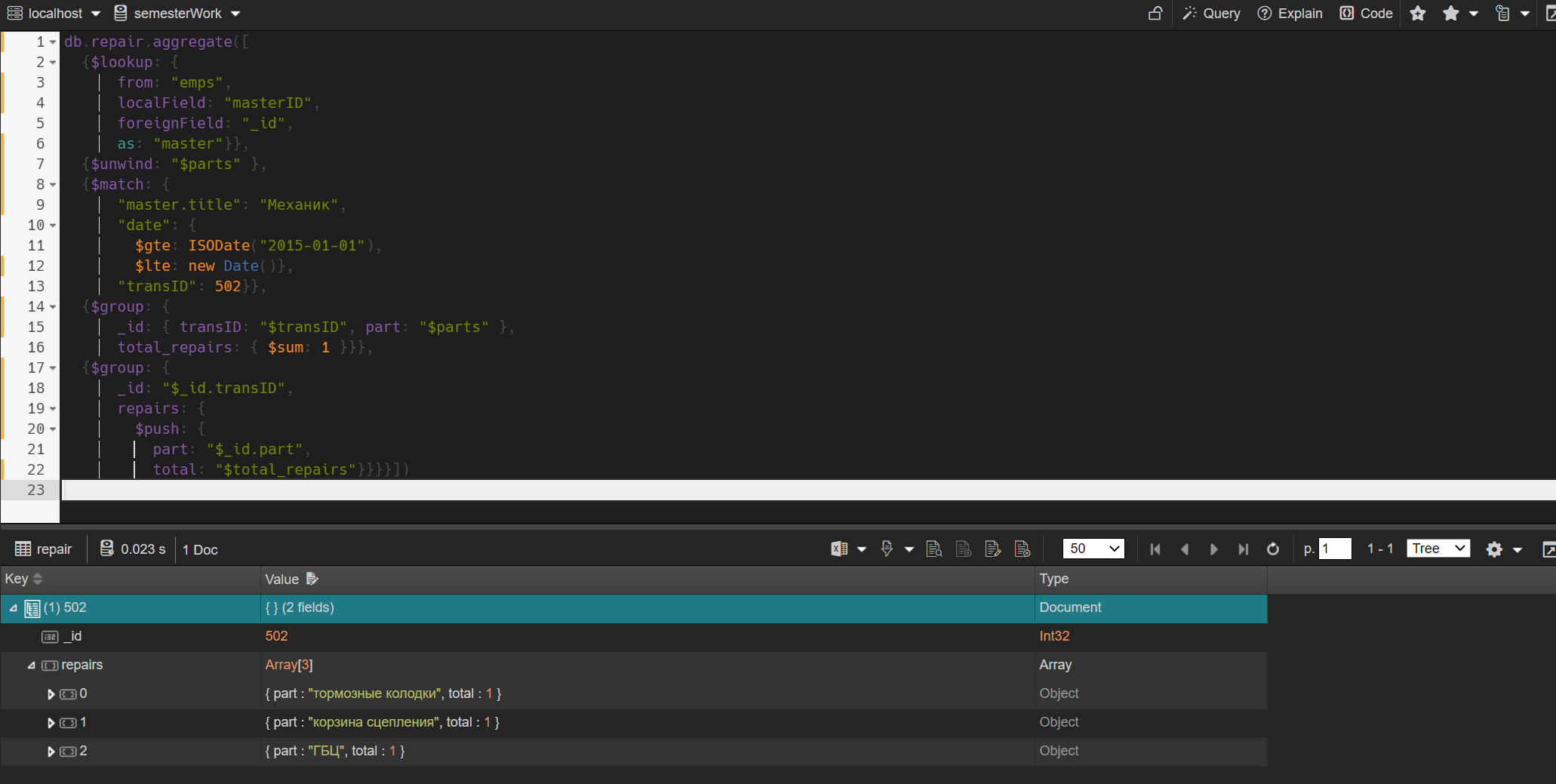


Рис. 20 – Результаты выполнения запроса

Код:

db.repair.aggregate([

{$lookup: {

from: "emps",

localField: "masterID",

foreignField: "\_id",

as: "master"}},

{$unwind: "$parts" },

{$match: {

"master.title": "Механик",

"date": {

$gte: ISODate("2015-01-01"),

$lte: new Date()},

"transID": 502}},

{$group: {

\_id: { transID: "$transID", part: "$parts" },

total\_repairs: { $sum: 1 }}},

{$group: {

\_id: "$\_id.transID",

repairs: {

$push: {

part: "$\_id.part",

total: "$total\_repairs"}}}}])

# Заключение

# В ходе данной работы мы разработали базу данных для информационной системы автопредприятия города, целью которой является эффективное управление процессами перевозок внутри города. Задачи нашего исследования включали в себя определение структур данных для хранения информации о транспортных средствах, персонале, техническом оборудовании и других аспектах деятельности предприятия, а также разработку запросов для получения аналитических данных.

# Благодаря разработанной базе данных и запросам, мы смогли получить информацию о состоянии автотранспорта, его техническом обслуживании, а также о работе персонала. Эти данные позволяют оптимизировать процессы управления, планирования и анализа деятельности предприятия, что способствует повышению эффективности его работы.

# Однако данная работа является лишь первым шагом в развитии информационной системы автопредприятия. Дальнейшее развитие базы данных и расширение ее функциональности позволят более полно удовлетворить потребности управления автопредприятием и обеспечить более качественное обслуживание городского транспорта.

# Список используемой литературы.

1. Руководство по JavaScript / <https://metanit.com/web/javascript>
2. MongoDB: Пагинация и сортировка / <https://metanit.com/nosql/mongodb/2.14.php>
3. MongoDB: Установка и начало работы с MongoDB на Windows / <https://metanit.com/nosql/mongodb/1.2.php>
4. Шеннон Брэдшоу, Йон Брэзил, Кристина Ходоров MongoDB: полное руководство. Мощная и масштабируемая система управления базами данных / пер. с англ. Д. А. Беликова – М.: ДМК Пресс, 2020. – 540 с.: ил.
5. Кайл Бэнкер MongoDB в действии / Пер. с англ. Слинкина А. А. - М.: ДМК Пресс, 2012. - 394с.: ил.
6. Онлайн-руководство по MongoDB / <https://metanit.com/nosql/mongodb/>
7. $group (aggregation) / <https://www.mongodb.com/docs/manual/reference/operator/aggregation/group/#mongodb-pipeline-pipe.-group>
8. Как Создать Базу Данных и Коллекцию в MongoDB? / https://serverspace.ru/support/help/kak-sozdat-bazu-dannyh-i-kollekcziyu-v-mongodb/