Задание 1.1: Анализ исходных данных и проектирование схемы

Цель

Понять структуру исходных данных и спроектировать оптимальную схему для MongoDB
Исходные данные: parquet файл, где каждая строка содержит информацию об одном товаре с привязкой к категории через поле Category_FullPathName (формат: "Уровень1\Уровень2\Уровень3...")

Задачи

1. Определить, сколько уникальных категорий в датасете
2. Определить максимальную глубину вложенности категорий
3. Выявить категории с наибольшим количеством товаров
4. Понять, есть ли товары с одинаковыми Offer_ID у разных партнеров
5. Определить, сколько уникальных типов товаров (Offer_Type) представлено в данных

Ответ

- Короткая аналитическая справка (5-7 предложений) с описанием структуры данных:

Данные представлены в формате parquet со следующими столбцами:

- Partner_Name - имя маркетплейса. В этих данных везде равно _ozon - партнер один
- Category_ID - уникальный идентификатор категории (уникальный для каждого уникального Category_FullPathName) - всего 5261 уникальных
- Category_FullPathName - плоская иерархия категорий - 5261 уникальных - сходится с числом уникальных Category_ID, можно верить, что для двух одинаковых Category_FullPathName совпадут Category_ID. Максимальная глубина вложенности - 8.
- Offer_ID - идентификатор предложения, 1355049 уникальных при 1355049 записей всего - PK
- Offer_Name - название предложения, 983437 уникальных
- Offer_Type - тип предложения, 12379 уникальных

1) 5261 категория
2) Максимальная глубина - 8
3) Максимальное количество товаров в одной категории - 300, всего таких категорий 4246
4) Нет, все Offer_ID разные, а партнер один, значит не повторяются

5) 12379 уникальных Offer_Type

- Обоснование выбора создания двух коллекций (categories и products) вместо одной:

Разделение данных на две коллекции обусловлено различной природой и жизненным циклом сущностей «категория» и «товар». Категории представляют собой иерархическую справочную структуру с относительно редкими изменениями, но частыми иерархическими запросами (поиск подкатегорий, навигация по дереву, аналитика по уровням). Товары, напротив, являются высокообъёмными и часто читаемыми сущностями, для которых критично быстрое получение всей информации без дополнительных join-операций. Хранение категорий и товаров в одной коллекции привело бы к многократному дублированию иерархии категорий, усложнению обновлений и увеличению размера документов. Выделение отдельной коллекции categories с использованием паттерна Materialized Path позволяет эффективно выполнять иерархические запросы и аналитику по структуре каталога. Коллекция products, в свою очередь, использует денормализацию и встраивание информации о категории, что оптимизирует сценарии чтения и соответствует рекомендуемым практикам MongoDB для высоконагруженных read-oriented систем.

- Список основных преимуществ денормализации данных для данного кейса:

- **Отсутствие JOIN-операций при чтении данных**
  В MongoDB операции $lookup дороги и плохо масштабируются. Денормализация позволяет получать товар вместе с полной информацией о категории за один запрос.

- **Быстрое выполнение типовых сценариев чтения**
  Основные запросы ориентированы на витрину товаров (по категории, по типу, по навигации), а не на модификацию справочников. Встраивание категории в документ товара оптимизирует эти сценарии.

- **Упрощение индексирования и запросов**
  Денормализованные поля (category.id, category.breadcrumbs.name) можно напрямую индексировать, избегая сложных агрегаций и вложенных $lookup.

- **Предсказуемая производительность при росте объёма данных**
  Коллекция products масштабируется линейно по количеству документов, без зависимости от размера или сложности иерархии категорий.

- **Изоляция жизненных циклов данных**
  Категории изменяются редко, товары — часто. Денормализация позволяет обновлять товары независимо от справочной структуры категорий.

- **Проще аналитика и агрегации**
  Аналитические запросы (топ категорий, распределение по типам товаров)

выполняются напрямую по коллекции products без предварительных джойнов.

- **Соответствие рекомендованным практикам MongoDB**
  Модель данных ориентирована на чтение и соответствует принципу «store data the way you query it», что является базовой рекомендацией MongoDB.

Задание 1.2: Создание коллекции категорий (categories)

Цель

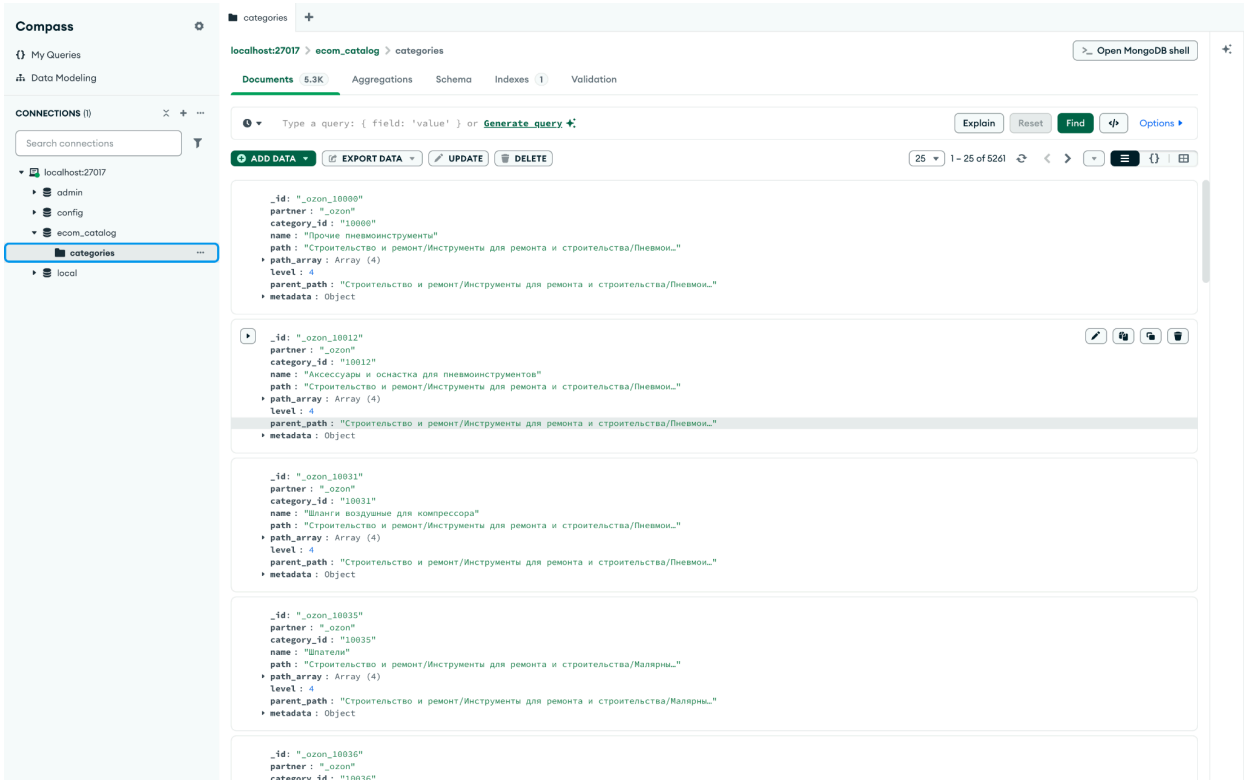
Преобразовать плоскую таблицу с иерархическими путями в структурированные документы MongoDB с применением паттерна "Materialized Path"
Требования к структуре документа:
- _id: строка вида "partner_categoryid" (например, "_ozon_10000")
- partner: название партнера (например, "_ozon")
- category_id: идентификатор категории (строка)
- name: название категории — последний элемент иерархии (например, "Прочие пневмоинструменты")
- path: полный путь категории с разделителем "/" (например, "Строительство и ремонт/Инструменты для ремонта и строительства/Пневмоинструменты/Прочие пневмоинструменты")
- path_array: массив строк с элементами пути для иерархических запросов (["Строительство и ремонт", "Инструменты для ремонта и строительства", "Пневмоинструменты", "Прочие пневмоинструменты"])
- level: числовое значение уровня вложенности (1, 2, 3, 4...)
- parent_path: путь к родительской категории (строка или null для категорий 1-го уровня)
- metadata: вложенный объект:
- total_products: количество товаров в категории (целое число)
- last_updated: дата и время обновления (ISODate)

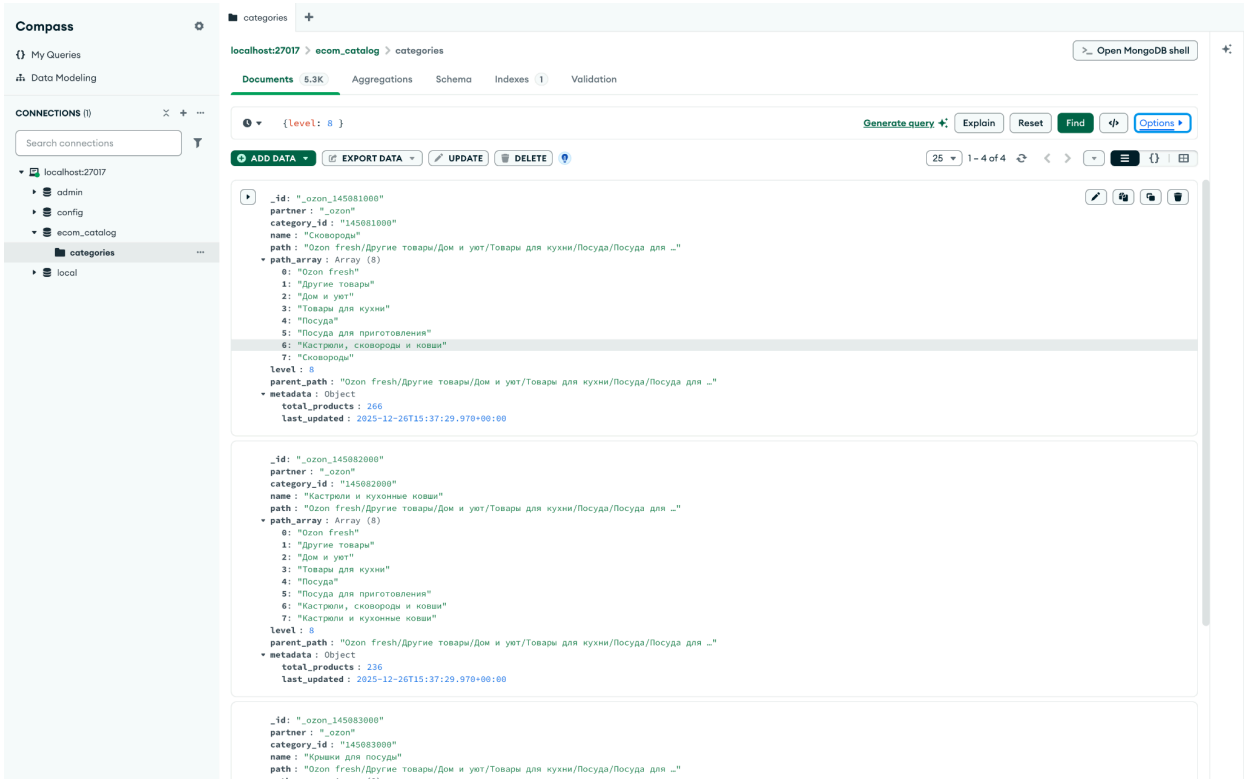
Алгоритм формирования:
- Пройти по всем строкам pq
- Для каждой уникальной комбинации Partner_Name + Category_ID создать один документ категории
- Подсчитать количество товаров (total_products) по всем строкам с этой категорией

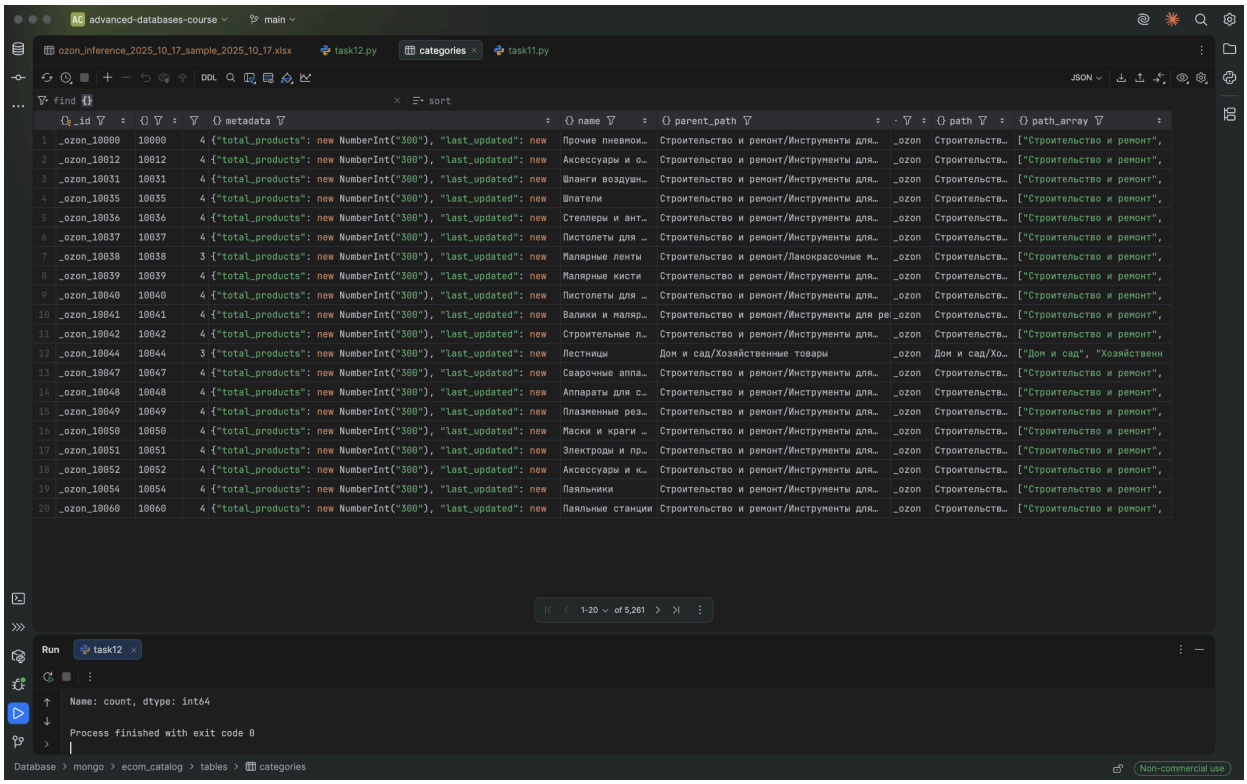
Ответ

Результат:

- Коллекция categories в базе ecom_catalog
- Документы должны быть уникальными (одна категория = один документ, даже если в неё входит много товаров)
- Скриншот примера 2-3 документов в MongoDB Compass с разным уровнем вложенности
- Статистика: общее количество категорий, распределение по уровням (сколько категорий 1-го уровня, 2-го, 3-го, 4-го)




Compass
My Queries
Data Modeling
CONNECTIONS (1)
Search connections
localhost:27017
admin
config
ecom_catalog
categories
local
categories
localhost:27017 > ecom_catalog > categories
Open MongoDB shell
Documents 5.3K
Aggregations
Schema
Indexes 1
Validation
{level: 8 }
Generate query
Explain
Reset
Find
Options
ADD DATA
EXPORT DATA
UPDATE
DELETE
25
1 - 4 of 4
_id: "_ozon_145081000"
partner : "_ozon"
category_id : "145081000"
name : "Сковороды"
path : "Ozon fresh/Другие товары/Дом и уют/Товары для кухни/Посуда/Посуда для …"
path_array : Array (8)
0: "Ozon fresh"
1: "Другие товары"
2: "Дом и уют"
3: "Товары для кухни"
4: "Посуда"
5: "Посуда для приготовления"
6: "Кастрюли, сковороды и ковши"
7: "Сковороды"
level : 8
parent_path : "Ozon fresh/Другие товары/Дом и уют/Товары для кухни/Посуда/Посуда для …"
metadata : Object
total_products : 266
last_updated : 2025-12-26T15:37:29.970+00:00
_id: "_ozon_145082000"
partner : "_ozon"
category_id : "145082000"
name : "Кастрюли и кухонные ковши"
path : "Ozon fresh/Другие товары/Дом и уют/Товары для кухни/Посуда/Посуда для …"
path_array : Array (8)
0: "Ozon fresh"
1: "Другие товары"
2: "Дом и уют"
3: "Товары для кухни"
4: "Посуда"
5: "Посуда для приготовления"
6: "Кастрюли, сковороды и ковши"
7: "Кастрюли и кухонные ковши"
level : 8
parent_path : "Ozon fresh/Другие товары/Дом и уют/Товары для кухни/Посуда/Посуда для …"
metadata : Object
total_products : 236
last_updated : 2025-12-26T15:37:29.970+00:00
_id: "_ozon_145083000"
partner : "_ozon"
category_id : "145083000"
name : "Крышки для посуды"
path : "Ozon fresh/Другие товары/Дом и уют/Товары для кухни/Посуда/Посуда для …"




```
Общее количество категорий: 5261
Распределение по уровням:
level
1       2
2      85
3    1491
4    3232
5     304
6     131
7      12
8       4
```

Задание 1.3: Создание коллекции товаров (products)

Цель

Создать коллекцию товаров с денормализованной информацией о категориях (паттерн "Embedded Documents")
Требования к структуре документа:
- _id: строка вида "partner_offerid" (например, "_ozon_1606856085")

- partner: название партнера (строка)
- offer_id: идентификатор товара (строка)
- name: название товара (строка)
- type: тип товара (строка, например, "Степлер строительный")
- category: вложенный объект с информацией о категории:
- id: идентификатор категории (строка)
- name: название категории (строка)
- full_path: полный путь категории (строка с "/")
- breadcrumbs: массив объектов для навигации, каждый элемент содержит:
- level: уровень в иерархии (1, 2, 3...)
- name: название категории на этом уровне
- created_at: дата создания записи (ISODate)
- updated_at: дата последнего обновления (ISODate)

Важно: В отличие от коллекции categories, здесь каждая строка pq превращается в один документ товара. Информация о категории дублируется (денормализация) для быстрого доступа без JOIN операций.

Ответ

Результат:

- Коллекция products в базе ecom_catalog
- Скриншот примера документа с полной развернутой структурой вложенных объектов
- Статистика: общее количество товаров, топ-5 типов товаров по частоте встречаемости, распределение товаров по партнерам

```
_id: "_ozon_1606856085"
partner : "_ozon"
offer_id : "1606856085"
name : "Пневматический скобозабивной каркасный пистолет MEITE MT-N851-H"
type : "Степлер строительный"
▾ category : Object
    id : "10000"
    name : "Прочие пневмоинструменты"
    full_path : "Строительство и ремонт/Инструменты для ремонта и строительства/Пневмои…"
  ▾ breadcrumbs : Array (4)
    ▾ 0: Object
        level : 1
        name : "Строительство и ремонт"
        created_at : 2025-12-26T15:49:53.057+00:00
        updated_at : 2025-12-26T15:49:53.057+00:00
    ▾ 1: Object
        level : 2
        name : "Инструменты для ремонта и строительства"
        created_at : 2025-12-26T15:49:53.057+00:00
        updated_at : 2025-12-26T15:49:53.057+00:00
    ▾ 2: Object
        level : 3
        name : "Пневмоинструменты"
        created_at : 2025-12-26T15:49:53.057+00:00
        updated_at : 2025-12-26T15:49:53.057+00:00
    ▾ 3: Object
        level : 4
        name : "Прочие пневмоинструменты"
        created_at : 2025-12-26T15:49:53.057+00:00
        updated_at : 2025-12-26T15:49:53.057+00:00
  created_at : 2025-12-26T15:50:58.849+00:00
  updated_at : 2025-12-26T15:50:58.849+00:00
```


```
Общее количество товаров: 1355049
Топ-5 типов товаров:
Offer_Type
Лекарственное средство безрецептурное    10167
Печатная книга                            9657
Лекарственное средство рецептурное        8528
Электронная книга                         3382
Сумка                                     3106
Name: count, dtype: int64
Распределение товаров по партнерам:
Partner_Name
_ozon    1355049
```

Задание 1.4: Создание индексов для оптимизации запросов

Цель

Создать индексы для ускорения типовых запросов к иерархическим данным
Требования для коллекции categories:
1. Текстовый индекс на поле path — для полнотекстового поиска по категориям
2. Индекс на поле path_array — для запросов "найти все подкатегории"
3. Составной индекс на поля partner и level — для выборок категорий определенного уровня у партнера
4. Индекс на поле metadata.total_products — для сортировки по популярности категорий
Требования для коллекции products:
1. Составной индекс на поля partner и category.id — для выборки товаров категории
2. Индекс на поле category.breadcrumbs.name — для поиска по любому уровню категории
3. Составной индекс на поля type и partner — для аналитики по типам товаров
4. Индекс на поле offer_id — для быстрого поиска товара по идентификатору

Ответ

Результат:
- Скриншот вывода команды db.categories.getIndexes() и db.products.getIndexes()
- Краткий вывод: какой процент от данных занимают индексы и как это соотносится с рекомендациями по MongoDB

```
> db.categories.getIndexes()
< [
    { v: 2, key: { _id: 1 }, name: '_id_' },
    {
      v: 2,
      key: { _fts: 'text', _ftsx: 1 },
      name: 'path_text',
      weights: { path: 1 },
      default_language: 'english',
      language_override: 'language',
      textIndexVersion: 3
    },
    { v: 2, key: { path_array: 1 }, name: 'path_array_1' },
    { v: 2, key: { partner: 1, level: 1 }, name: 'partner_1_level_1' },
    {
      v: 2,
      key: { 'metadata.total_products': -1 },
      name: 'metadata.total_products_-1'
    }
  ]
> db.products.getIndexes()
< [
    { v: 2, key: { _id: 1 }, name: '_id_' },
    {
      v: 2,
      key: { partner: 1, 'category.id': 1 },
      name: 'partner_1_category.id_1'
    },
    {
      v: 2,
      key: { 'category.breadcrumbs.name': 1 },
      name: 'category.breadcrumbs.name_1'
    },
    { v: 2, key: { type: 1, partner: 1 }, name: 'type_1_partner_1' },
    { v: 2, key: { offer_id: 1 }, name: 'offer_id_1' }
  ]
```

Индексы и их влияние на производительность

categories

Поле / состав	Тип индекса	Размер (B)
_id_	PK	98 304
path	text	663 552
path_array	1 (ascending)	319 488
partner, level	1,1 (составной)	49 152
metadata.total_products	-1 (descending)	45 056

Общие цифры:

- Количество документов: 5 261
- Размер данных: 3 327 310 B (~3,3 MB)
- Размер индексов: 1 175 552 B (~1,12 MB)
- Процент индексов от данных: ≈35%

Вывод:
Индексы занимают чуть больше трети объёма данных из-за текстового поиска, но для коллекции с 5 261 документом это полностью оправдано. Все индексы покрывают типовые запросы: поиск по категории, фильтрация по уровню и популярности. Структура соответствует рекомендациям MongoDB для коллекций с Materialized Path и текстовым поиском.

Коллекция products

Поле / состав	Тип индекса	Размер (MB)
_id_	PK	25,1
partner, category.id	1,1 (составной)	13,6

category.breadcrumbs.name	1 (ascending)	25,5
type, partner	1,1 (составной)	7,0
offer_id	1 (ascending)	21,4

Общие цифры:

- Количество документов: 1 355 049
- Размер данных: 1 237 MB (~1,24 GB)
- Размер индексов: 92,5 MB
- Процент индексов от данных: ≈7,5%

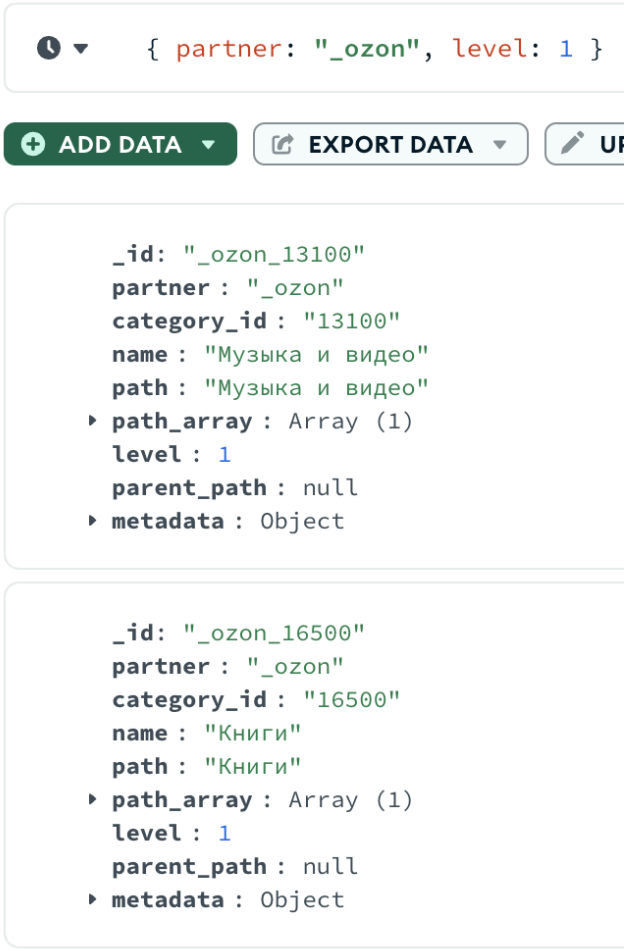
Вывод:
Индексы покрывают все типовые сценарии чтения: выборка по категории, поиск по уровню категорий, аналитика по типу товаров и быстрый поиск по offer_id. Размер индексов относительно объёма данных небольшой (~7-8%), что соответствует best practices MongoDB для read-heavy коллекций с денормализованной структурой. Денормализация и правильно настроенные индексы позволяют выполнять запросы без $lookup, обеспечивая высокую производительность.

Задание 2.1: Навигация по иерархии категорий

Выполнить следующие запросы к коллекции categories и зафиксировать результаты:
Запрос 1: Найти все категории первого уровня (корневые) для партнера "_ozon"
- Условие: level = 1 и partner = "_ozon"
Запрос 2: Найти все подкатегории (любого уровня), которые входят в "Строительство и ремонт"
- Использовать поле path_array и оператор поиска в массиве
Запрос 3: Топ-10 самых "населенных" категорий по количеству товаров
- Сортировка по metadata.total_products (по убыванию)

Результат для каждого запроса:
- Количество найденных документов
- Первые 3 результата (названия категорий и количество товаров)
- Использование индекса: выполнить explain() и указать, какой индекс использовался (или COLLSCAN, если индекс не применился)

Запрос 1

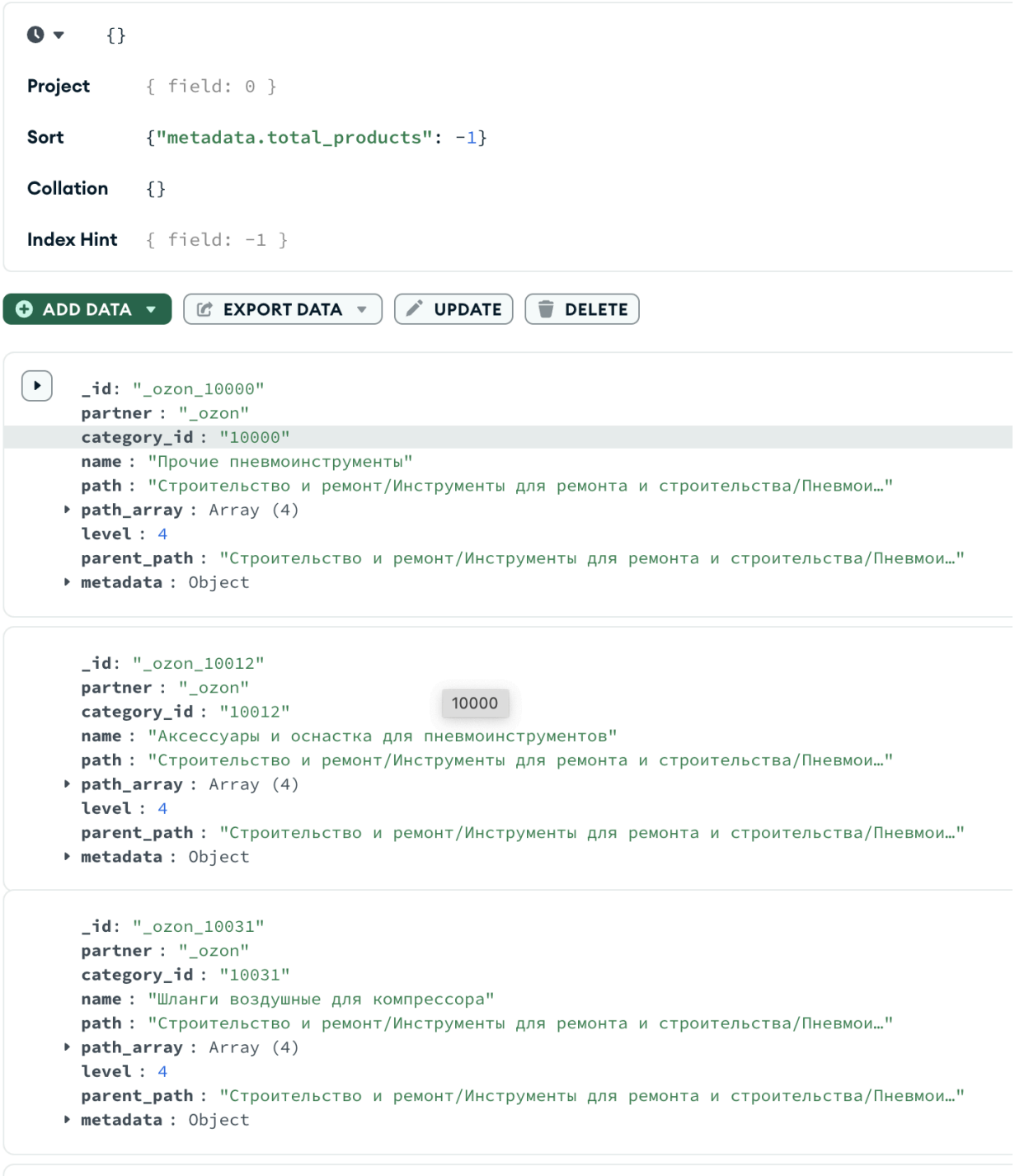


Использовался индекс partner_1_level_1, найдено 2 документа

Запрос 2

Использовался индекс path_array_1, результатов 540

Запрос 3

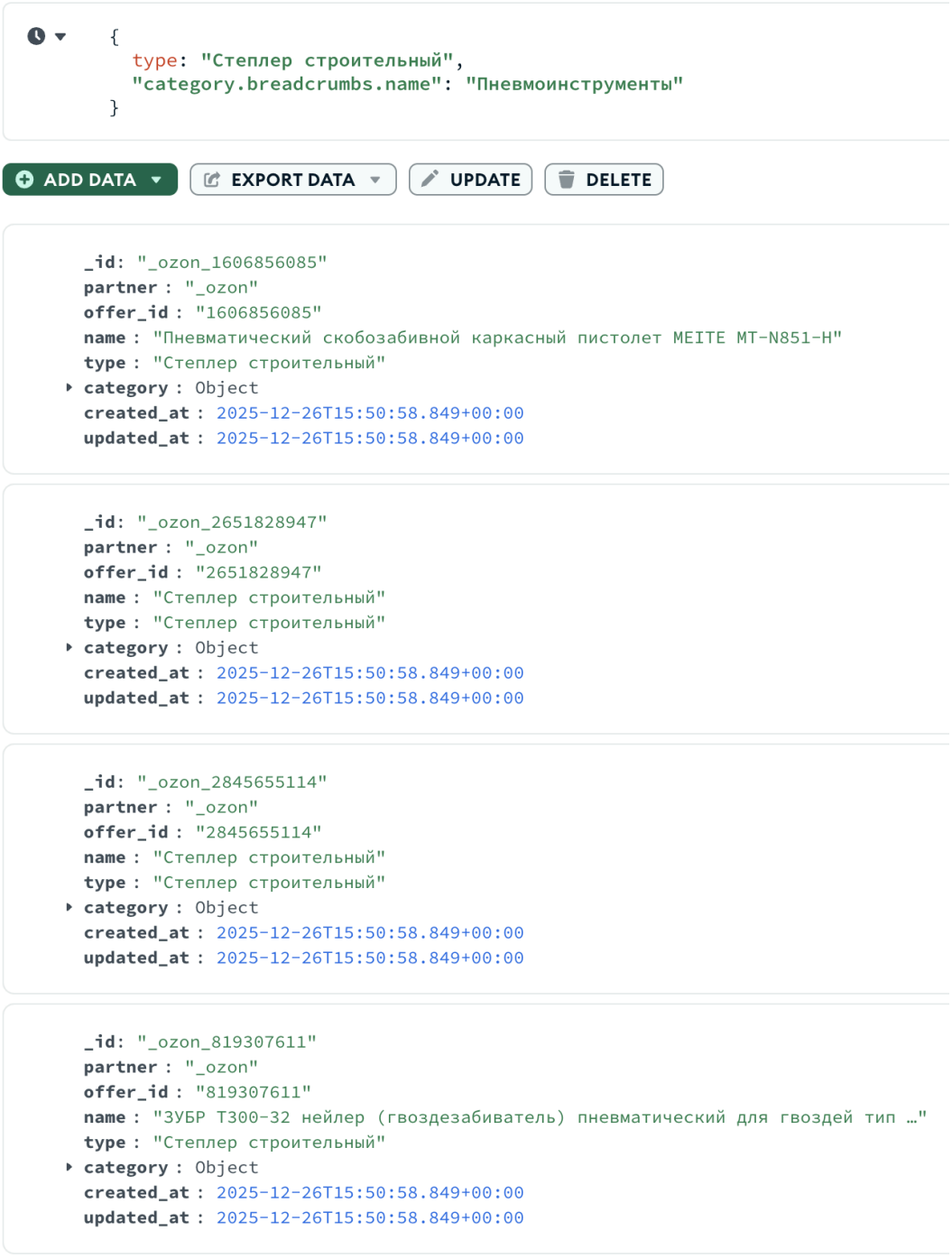


Использовался индекс metadata.total_products_-1, 5261 результатов

Задание 2.2: Работа с товарами и вложенными документами

Выполнить следующие запросы к коллекции products:
Запрос 1: Найти все товары типа "Степлер строительный", у которых в иерархии категорий (breadcrumbs) есть уровень "Пневмоинструменты"
- Условие: type = "Степлер строительный" И category.breadcrumbs.name содержит "Пневмоинструменты"
Запрос 2: Найти товары, которые находятся на 4-м уровне иерархии категорий
- Использовать размер массива breadcrumbs или проверку существования 4-го элемента
Запрос 3: Найти количество товаров в каждой категории 1-го уровня
- Группировка: взять первый элемент из category.breadcrumbs (level = 1)
- Подсчитать количество товаров в каждой группе
- Использовать базовую агрегацию с $group
Результат:
- Для запросов:количество найденных документов и примеры (2-3 документа или агрегированный результат)
- Скриншот результата запроса 3 (распределение товаров по категориям 1-го уровня) в виде таблицы

Запрос 1



```
{
  type: "Степлер строительный",
  "category.breadcrumbs.name": "Пневмоинструменты"
}
```

ADD DATA | EXPORT DATA | UPDATE | DELETE

```
_id: "_ozon_1606856085"
partner : "_ozon"
offer_id : "1606856085"
name : "Пневматический скобозабивной каркасный пистолет MEITE MT-N851-H"
type : "Степлер строительный"
▸ category : Object
created_at : 2025-12-26T15:50:58.849+00:00
updated_at : 2025-12-26T15:50:58.849+00:00
```

```
_id: "_ozon_2651828947"
partner : "_ozon"
offer_id : "2651828947"
name : "Степлер строительный"
type : "Степлер строительный"
▸ category : Object
created_at : 2025-12-26T15:50:58.849+00:00
updated_at : 2025-12-26T15:50:58.849+00:00
```

```
_id: "_ozon_2845655114"
partner : "_ozon"
offer_id : "2845655114"
name : "Степлер строительный"
type : "Степлер строительный"
▸ category : Object
created_at : 2025-12-26T15:50:58.849+00:00
updated_at : 2025-12-26T15:50:58.849+00:00
```

```
_id: "_ozon_819307611"
partner : "_ozon"
offer_id : "819307611"
name : "ЗУБР Т300-32 нейлер (гвоздезабиватель) пневматический для гвоздей тип …"
type : "Степлер строительный"
▸ category : Object
created_at : 2025-12-26T15:50:58.849+00:00
updated_at : 2025-12-26T15:50:58.849+00:00
```

47 результатов

Запрос 2

915661 результат

Запрос 3

```
[
  {
    $project: {
      root_category: { $arrayElemAt: ["$category.breadcrumbs", 0] }
    }
  },
  {
    $group: {
      _id: "$root_category.name",
      total_products: { $sum: 1 }
    }
  },
  {
    $sort: { total_products: -1 }
  }
]
```

2.2

	_id String	total_products Int32
1	"Строительство и ремонт"	158566
2	"Дом и сад"	144499
3	"Автотовары"	119483
4	"Ozon fresh"	78327
5	"Спорт и отдых"	72323
6	"Хобби и творчество"	66380
7	"Электроника"	65984
8	"Продукты питания"	63366
9	"Красота и здоровье"	62400
10	"Аптека"	60031
11	"Детские товары"	56793
12	"Бытовая техника"	54351
13	"Товары для животных"	52777
14	"Одежда"	51538
15	"Туризм, рыбалка, охота"	44811
16	"Канцелярские товары"	43606
17	"Аксессуары"	38442
18	"Мебель"	18900
19	"Обувь"	18600
20	"Антиквариат и коллекцион…	16675
21	"Книги"	15600
22	"Бытовая химия и гигиена"	12411
23	"Товары для взрослых"	9000
24	"Игры и консоли"	8864
25	"Ювелирные украшения"	8586

Задание 3.1: Аналитика по категориям

Агрегация 1: Топ-10 категорий по количеству товаров с детальной статистикой
Создать агрегационный pipeline:
- Группировка по category.id

- Вычислить: общее количество товаров (count), название категории (взять из первого документа), полный путь категории
- Отсортировать по количеству товаров (убывание)
- Ограничить результат первыми 10 записям

Результат:
- Таблица с результатами (10 строк): category_id, название категории, полный путь, количество товаров
- Время выполнения агрегации (из explain)
- Интерпретация: какая категория самая большая по количеству товаров

Агрегация 2: Иерархическая статистика по уровням
Создать агрегационный pipeline:
- Развернуть массив category.breadcrumbs с помощью $unwind
- Группировка по двум полям: уровню (breadcrumbs.level) и названию категории (breadcrumbs.name)
- Подсчитать количество товаров на каждом уровне для каждой категории
- Отсортировать: сначала по уровню (возрастание), затем по количеству товаров (убывание)
- Ограничить первыми 30 записям

Результат:
- Таблица результатов (30 строк): уровень, название категории, количество товаров
- Интерпретация: на каком уровне иерархии больше всего товаров? Какие категории доминируют на каждом уровне (топ-3 на 1-м, 2-м, 3-м уровнях)?

Агрегация 1

	_id String	count Int32	category_name String	full_path String
1	"9496"	300	"Кукурузные палочки"	"Продукты питания/Орехи и…
2	"39183"	300	"Рейлинги, держатели и по…	"Дом и сад/Посуда и кухон…
3	"9810"	300	"Морилки"	"Строительство и ремонт/Л…
4	"7140"	300	"Треки, авторалли и парко…	"Детские товары/Игрушки и…
5	"11028"	300	"Звонки и сирены велосипе…	"Спорт и отдых/Товары для…
6	"11526"	300	"Походная посуда"	"Туризм, рыбалка, охота/Т…
7	"30034"	300	"Препараты при диспепсии,…	"Аптека/Лекарственные сре…
8	"11293"	300	"Бадминтон"	"Спорт и отдых/Теннис и б…
9	"34597"	300	"Наковальни"	"Строительство и ремонт/И…
10	"32445"	300	"Снаряжение для рыбалки"	"Туризм, рыбалка, охота/Р…

3971 мс
Как я уже упоминал, максимум в одной категории - 300, и таких категорий 4246 разных. Так что одного максимума нет

Агрегация 2

	total_products Int32	level Int32	category_name String
1	158566	1	"Строительство и ремонт"
2	144499	1	"Дом и сад"
3	119483	1	"Автотовары"
4	78327	1	"Ozon fresh"
5	72323	1	"Спорт и отдых"
6	66380	1	"Хобби и творчество"
7	65984	1	"Электроника"
8	63366	1	"Продукты питания"
9	62400	1	"Красота и здоровье"
10	60031	1	"Аптека"
11	56793	1	"Детские товары"
12	54351	1	"Бытовая техника"
13	52777	1	"Товары для животных"
14	51538	1	"Одежда"
15	44811	1	"Туризм, рыбалка, охота"
16	43606	1	"Канцелярские товары"
17	38442	1	"Аксессуары"
18	18900	1	"Мебель"
19	18600	1	"Обувь"
20	16675	1	"Антиквариат и коллекцион…
21	15600	1	"Книги"
22	12411	1	"Бытовая химия и гигиена"
23	9000	1	"Товары для взрослых"
24	8864	1	"Игры и консоли"
25	8586	1	"Ювелирные украшения"
26	6600	1	"Музыка и видео"
27	3600	1	"Товары для курения и акс…
28	2236	1	"Цифровые товары"
29	300	1	"Автомобили"
30	41419	2	"Запчасти для легковых ав…

На 1 уровне, ожидаемо, больше всего товаров. Топ-3: Строительство и ремонт, Дом и сад, Автотовары
На 2 уровне Запчасти для легковых автомобилей, Инструменты для ремонта и строительства, Дача и сад
На 3 уровне доминирует Бытовая техника, Аксессуары и материалы для рукоделия, Садовый инструмент

Задание 3.3: Анализ структуры категорий

Цель: Понять глубину и разветвленность дерева категорий
Выполнить следующие агрегации на коллекции categories:
Агрегация А: Распределение категорий по уровням и партнерам
- Группировка по partner и level
- Подсчитать количество категорий и сумму товаров для каждой комбинации
- Отсортировать по партнеру и уровню

Агрегация Б: Категории-"листья" (конечные категории без подкатегорий)
- Найти категории, для которых не существует других категорий с parent_path равным их path
- Использовать $lookup для самосоединения коллекции categories
- Отфильтровать те, у которых массив дочерних категорий пустой
- Показать топ-10 листьев по количеству товаров

Результат:
- Текст обеих агрегаций
- Таблица результатов для агрегации А: партнер, уровень, количество категорий, сумма товаров
- Список топ-10 категорий-листьев с количеством товаров
- Вывод: какова средняя глубина категорий? Есть ли категории-листья с очень большим количеством товаров (потенциальные кандидаты на разбиение)?

Агрегация А

```
[
  {
    $group: {
      _id: {
        partner: "$partner",
        level: "$level"
      },
      categories_count: { $sum: 1 },
      total_products: {
        $sum: "$metadata.total_products"
      }
    }
  },
  {
    $project: {
      _id: 0,
      partner: "$_id.partner",
      level: "$_id.level",
```

```
    categories_count: 1,
    total_products: 1
   }
  },
  {
   $sort: {
    partner: 1,
    level: 1
   }
  }
]
```

```
categories_count : 2
total_products : 600
partner : "_ozon"
level : 1
```

```
categories_count : 85
total_products : 21753
partner : "_ozon"
level : 2
```

```
categories_count : 1491
total_products : 417035
partner : "_ozon"
level : 3
```

1491

```
categories_count : 3232
total_products : 861977
partner : "_ozon"
level : 4
```

```
categories_count : 304
total_products : 35774
partner : "_ozon"
level : 5
```

```
categories_count : 131
total_products : 15762
partner : "_ozon"
level : 6
```

```
categories_count : 12
total_products : 1605
partner : "_ozon"
level : 7
```

```
categories_count : 4
total_products : 543
partner : "_ozon"
level : 8
```

Агрегация Б

```
[
```

```
  {
    $lookup: {
      from: "categories",
      localField: "path",
      foreignField: "parent_path",
      as: "children"
    }
  },
  {
    $match: {
      children: { $eq: [] }
    }
  },
  {
    $project: {
      _id: 0,
      partner: 1,
      category_id: 1,
      name: 1,
      path: 1,
      level: 1,
      total_products: "$metadata.total_products"
    }
  },
  {
    $sort: {
      total_products: -1
    }
  },
  {
    $limit: 10
  }
]
```

```
partner : "_ozon"
category_id : "10040"
name : "Пистолеты для герметиков"
path : "Строительство и ремонт/Инструменты для ремонта и строительства/Монтажн…"
level : 4
total_products : 300
```

```
partner : "_ozon"
category_id : "10041"
name : "Валики и малярные ванночки"
path : "Строительство и ремонт/Инструменты для ремонта и строительства/Малярны…"
level : 4
total_products : 300
```

```
partner : "_ozon"
category_id : "10036"
name : "Степлеры и антистеплеры строительные"
path : "Строительство и ремонт/Инструменты для ремонта и строительства/Монтажн…"
level : 4
total_products : 300
```

```
partner : "_ozon"
category_id : "10039"
name : "Малярные кисти"
path : "Строительство и ремонт/Инструменты для ремонта и строительства/Малярны…"
level : 4
total_products : 300
```

```
partner : "_ozon"
category_id : "10037"
name : "Пистолеты для пены"
path : "Строительство и ремонт/Инструменты для ремонта и строительства/Монтажн…"
level : 4
total_products : 300
```

```
partner : "_ozon"
category_id : "10035"
name : "Шпатели"
path : "Строительство и ремонт/Инструменты для ремонта и строительства/Малярны…"
level : 4
total_products : 300
```

```
partner : "_ozon"
category_id : "10012"
name : "Аксессуары и оснастка для пневмоинструментов"
path : "Строительство и ремонт/Инструменты для ремонта и строительства/Пневмои…"
level : 4
total_products : 300
```

```
partner : "_ozon"
category_id : "10000"
name : "Прочие пневмоинструменты"
path : "Строительство и ремонт/Инструменты для ремонта и строительства/Пневмои…"
level : 4
total_products : 300
```

```
partner : "_ozon"
category_id : "10031"
name : "Шланги воздушные для компрессора"
path : "Строительство и ремонт/Инструменты для ремонта и строительства/Пневмои…"
level : 4
total_products : 300
```

```
partner : "_ozon"
category_id : "10038"
name : "Малярные ленты"
path : "Строительство и ремонт/Лакокрасочные материалы/Малярные ленты"
level : 3
total_products : 300
```