



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт кибербезопасности и цифровых технологий
Кафедра КБ-14 «Цифровые технологии обработки данных»

Администрирование баз данных

Практическая работа № 1

Задание.

Создать базу данных интернет магазина (тематика на ваш выбор) используя MongoDB.

БД должна хранить информацию о товарах (название, количество, цена, производитель), их категориях (название категории), работниках и клиентах (имена и т.д.) и заказах (статус, дата).

Необходимо заполнить коллекцию демонстрационными данными

Добавить схеме валидацию. Определить, какие поля должны содержаться в каждой коллекции и типы данных, которые они должны хранить.

Реализовать следующие запросы к созданной бд:

1. Получение списка всех категорий
2. Получение списка всех продуктов в категории
3. Поиск продукта по названию
4. Добавление продукта в корзину клиента
5. Получение списка всех заказов клиента
6. Обновление статуса заказа
7. Получение списка топ-продаж за последние месяцы с учетом цены и количества проданных товаров.

8. Получение списка клиентов, которые сделали более чем N покупок в последнее время.
9. Получите какие категории товаров пользовались спросом в заданный срок
10. Какие товары не были проданы в какую-то дату

Создать следующих пользователей базы данных:

Администратор: имеет полный доступ к базе данных, в том числе может добавлять, редактировать и удалять продукты и пользователей.

Менеджер: имеет доступ к управлению продуктами, но не может управлять пользователями.

Пользователь: может только просматривать продукты и оформлять заказы.

Гость: не имеет доступа к личному кабинету и может только просматривать продукты.

Используя любой язык программирования на ваш выбор, реализовать следующий функционал:

1. Подключение к базе данных
2. Реализовать функционал поиска товаров, включая выбор категории, поиск по ключевым словам и фильтры цены и других характеристик
3. Реализовать функционал добавления товаров в корзину и подсчета общей стоимости заказа
4. Реализовать тесты для проверки схемы базы данных. Эти тесты должны проверять, что схема базы данных корректно реализует требования, определенные в первом шаге

★ Тема должна коррелировать с разработанной на предыдущем курсе CRM-системой, чтобы создать единую платформу (+1 балл).

Если системы, в силу различных причин, у вас нет, можно использовать [эталонную работу](#)

Отчет оформить в формате doc (docx) или pdf и выслать на проверку

Пример установки MongoDB.

В качестве примера будет установлен MongoDB Community Server версии 5.0.6 на операционную систему Windows 10. Для установки MongoDB на другие операционные системы или более подробной инструкции обратитесь к [официальной документации](#):

Для установки MongoDB зайдём на [официальный сайт](https://www.mongodb.com). Далее выбираем во вкладке Products версию Community Server (Рис. 1).

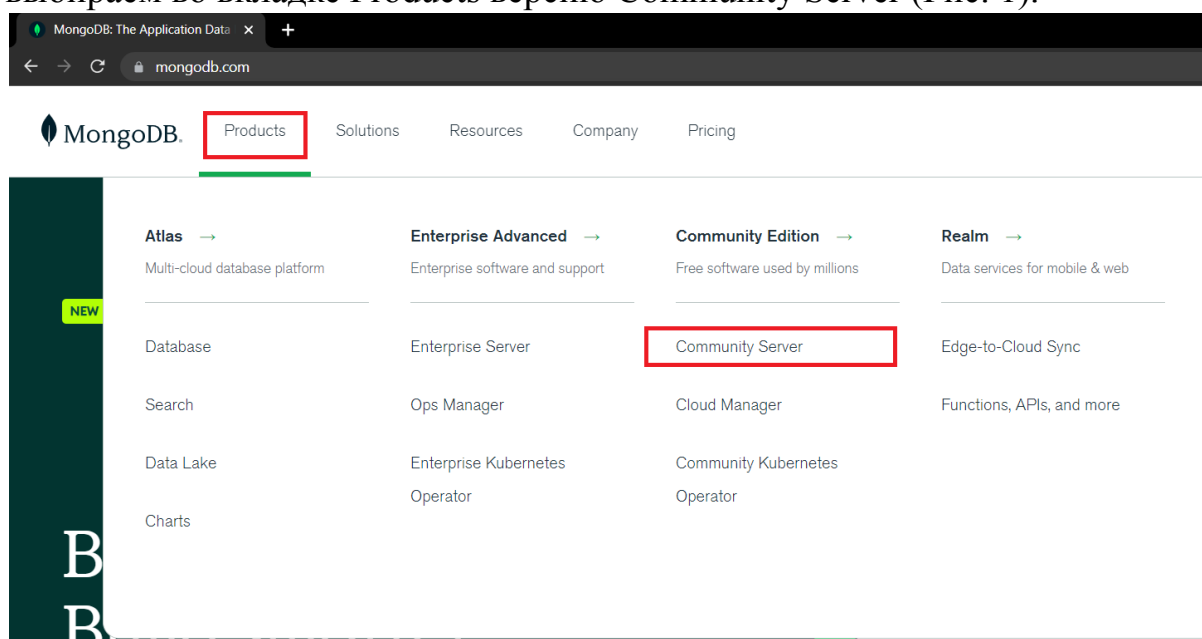


Рисунок 1. Выбор БД

Если не выбрано по умолчанию выбираем нужную нам версию и платформу, и нажимаем кнопку скачать (Рис. 2).

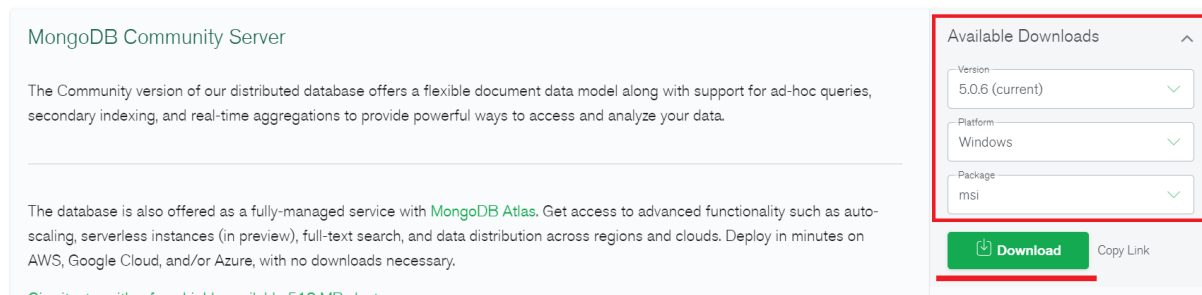


Рисунок 2. Выбор версии и платформы

Запускаем скачанный файл следуем инструкции установки.

После установки в директории установленной MongoDB по пути C:\Program Files\MongoDB\Server\5.0\bin (если выбрана установка по умолчанию) будут располагаться следующие файлы (Рис.3):

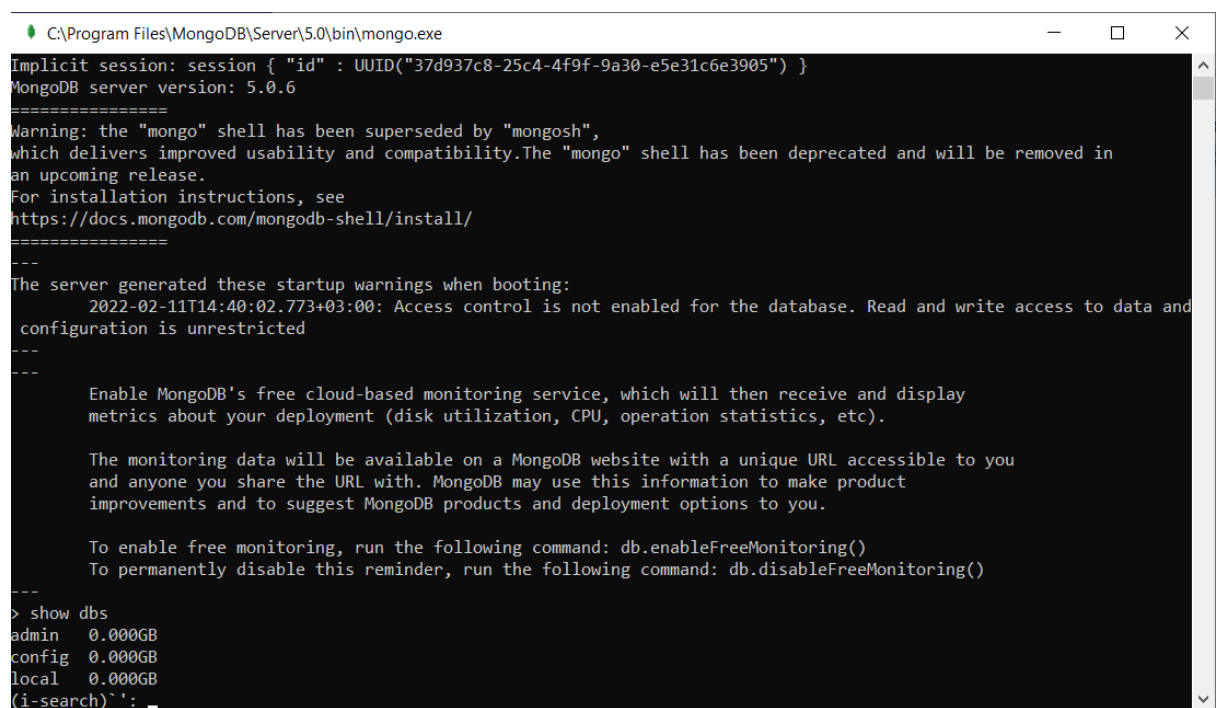
- **mongo**: представляет консольный интерфейс для взаимодействия с базами данных, своего рода консольный клиент
- **mongod**: сервер баз данных MongoDB. Он обрабатывает запросы, управляет форматом данных и выполняет различные операции в фоновом режиме по управлению базами данных
- **mongos**: служба маршрутизации MongoDB, которая помогает обрабатывать запросы и определять местоположение данных в кластере MongoDB

This PC > Local Disk (C:) > Program Files > MongoDB > Server > 5.0 > bin

Name	Date modified	Type	Size
InstallCompass	27.01.2022 16:20	Windows PowerSh...	2 KB
mongo	27.01.2022 18:03	Application	21 780 KB
mongod	11.02.2022 14:40	Configuration Sou...	1 KB
mongod	27.01.2022 18:02	Application	46 547 KB
mongod.pdb	27.01.2022 18:02	Program Debug D...	519 684 KB
mongos	27.01.2022 17:17	Application	29 367 KB
mongos.pdb	27.01.2022 17:17	Program Debug D...	304 908 KB

Рисунок 3. Директория с MongoDB

При запуске файла *mongo.exe* будет выведено окно консоли для взаимодействия с БД. Можно использовать команду *show dbs* показывающую имена всех существующих на данный момент баз данных (Рис. 4).



```
C:\Program Files\MongoDB\Server\5.0\bin\mongo.exe
Implicit session: session { "id" : UUID("37d937c8-25c4-4f9f-9a30-e5e31c6e3905") }
MongoDB server version: 5.0.6
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
---
The server generated these startup warnings when booting:
  2022-02-11T14:40:02.773+03:00: Access control is not enabled for the database. Read and write access to data and
configuration is unrestricted
---
---
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

  To enable free monitoring, run the following command: db.enableFreeMonitoring()
  To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> show dbs
admin  0.000GB
config 0.000GB
local  0.000GB
(i-search)`':
```

Рисунок 4. Демонстрация консольной команды

Пример создания коллекций и документов.

Для создания коллекции воспользуемся консольной оболочкой. Для создания коллекций нужно выбрать базу данных в качестве текущей, чтобы затем ее использовать её. Для этого используется команда *use 'имя базы данных'*. В качестве примера создадим базу данных магазина (*shop*). Для проверки выбранной БД используется команда *db* (Рис.5). Команда *db.stats()* выводит статистику по текущей базе данных.

```
C:\Program Files\MongoDB\Server\5.0\bin\mongo.exe
> use shop
switched to db shop
> db
shop
> db.stats()
{
  "db" : "shop",
  "collections" : 0,
  "views" : 0,
  "objects" : 0,
  "avgObjSize" : 0,
  "dataSize" : 0,
  "storageSize" : 0,
  "totalSize" : 0,
  "indexes" : 0,
  "indexSize" : 0,
  "scaleFactor" : 1,
  "fileSize" : 0,
  "fsUsedSize" : 0,
  "fsTotalSize" : 0,
  "ok" : 1
}
> _
```

Рисунок 5. Пример команды db

Коллекции создаются при помощи команды *db.createCollection('имя_коллекции', 'свойства_коллекции')*. Имя коллекции является обязательным, а свойства – нет. В свойствах можно указывать приведённые в таблице 1 параметры коллекции.

Таблица 1. Параметры коллекции

Поле	Тип	Описание
capped	Boolean	Создаёт ограниченную по размеру коллекцию. Если количество элементов в коллекции достигло максимума, то каждая следующая запись будет перезаписывать более старые данные.

autoIndexID	Boolean	Автоматически создаёт индекс для поля <code>_id</code> field.s
size	number	Определяет максимальный размер (в байтах) ограниченной (capped) коллекции. Если коллекция ограничена, то необходимо обязательно указать максимальный размер.
max	number	Определяет максимальное количество документов в ограниченной коллекции.

Создадим коллекцию товаров. Для просмотра всех коллекций используется команда *show collections* (рис 6).

```

C:\Program Files\MongoDB\Server\5.0\bin\mongo.exe
> db.createCollection("products")
{ "ok" : 1 }
> show collections
products
>

```

Рисунок 6. Команда show collection

Для добавления в коллекцию могут использоваться три ее метода:

- *insertOne()*: добавляет один документ
- *insertMany()*: добавляет несколько документов
- *insert()*: может добавлять как один, так и несколько документов

Пример добавления документа в коллекцию представлен на рисунке 7. Для вывода всех документов находящихся в коллекции используется функция *find()*. При добавлении в коллекцию идентификатор документа генерируется автоматически, но мы можем добавлять его явно. Стоит отметить, что при добавлении документов в несуществующую коллекцию, она создаётся автоматически.

```
C:\Program Files\MongoDB\Server\5.0\bin\mongo.exe
> db.products.insert({name : "iPhone 13", prise : "1099$"})
WriteResult({ "nInserted" : 1 })
> db.products.find()
{ "_id" : ObjectId("620665367027dabb8006df00"), "name" : "iPhone 13", "prise" : "1099$" }
>
```

Рисунок 7. Добавление документа в коллекцию

Операции чтения

Для чтения данных из коллекции можно использовать два метода. У этих методов есть несколько операторов и опций, которые могут помочь нам фильтровать наши данные в соответствии с нашими потребностями.

- *find*: возвращает несколько документов
- *findOne*: возвращает только первый документ

В этих двух операциях мы можем передавать фильтры. Если не указан фильтр, он вернет все документы в случае *find* и первый документ в случае *findOne*.

Операции обновления

Для чтения данных из коллекции можно использовать два метода.

- *updateOne*: обновляет только один документ
- *updateMany*: позволяет обновить множество документов.

Для обновления отдельных полей в этих функциях применяется оператор *\$set*. Если документ не содержит обновляемое поле, то оно создается.

Если нам надо полностью заменить один документ другим, также может использоваться функция *replaceOne*.

Операции удаления

Для удаления документов в MongoDB предусмотрены функции:

- *deleteOne()* - удаляет один документ
- *deleteMany()* - позволяет удалить несколько документов

Оба похожи на методы обновления и поиска. Они так же могут использовать фильтр, чтобы определить документ, который необходимо удалить.

Для удаления коллекций используется функция *drop*.

Типы данных MongoDB

Null

Нулевой тип можно использовать для обозначения как нулевого значения, так и несуществующего поля:

```
{ "x" : null }
```

Логический тип

Существует логический тип данных, который можно использовать для значений true и false:

```
{"x" : true}
```

Число

По умолчанию оболочка использует 64-битные числа с плавающей точкой. Таким образом, в оболочке эти числа выглядят «нормально»:

```
{"x" : 3.14}
```

```
{"x" : 3}
```

В случае с целыми числами используйте классы NumberInt или NumberLong, которые обозначают 4-байтовые или 8-байтовые целые числа со знаком соответственно.

```
{"x" : NumberInt ("3")}
```

```
{"x" : NumberLong ("3")}
```

Строка

Любая строка символов в кодировке UTF-8 может быть представлена с использованием типа строки:

```
{"x" : "foobar"}
```

Дата

MongoDB хранит даты в виде 64-битных целых чисел, обозначающих миллисекунды с момента эпохи Unix (1 января 1970 г.). Часовой пояс не сохраняется:

```
{"x" : new Date()}
```

Регулярное выражение

Запросы могут использовать регулярные выражения, используя синтаксис регулярных выражений JavaScript:

```
{"x" : / foobar / i}
```

Массив

Наборы или списки значений могут быть представлены в виде массивов:

```
{"x" : ["a", "b", "c"]}
```

Встраиваемый документ

Документы могут содержать целые документы, встроенные в качестве значений в родительский документ:

```
{"x" : {"foo": "bar"}}
```


Идентификатор объекта

Идентификатор объекта – это 12-байтовый идентификатор для документов (если не указывается явно, то генерируется автоматически неявно):

```
{"x" : ObjectId ()}
```