## Лабораторная работа № 9. MongoDB. Основы работы с MongoDB Compass

**MongoDB** − это кроссплатформенная документно-ориентированная технология организации данных. Работа с базой данных MongoDB основана на таких понятиях как коллекция и документ. **Документом** является набор пар ключ-значение. Формат хранения данных − BSON (binary JSON − JavaScript Object Notation).

**Коллекция** − это группа документов, которые вместе формируют набор данных. В рамках одной коллекции допускается наличие различных по структуре документов, однако обычно различия стараются свести к минимуму.

|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения на примере БД «Товары»** |
| 1. Создать с помощью команд языка JavaScriptбазу данных с именем **BaseMongo\_Yan** в соответствии с вариантом, приведенным в таблице п. 6.  Изучить создание базы данных с помощью графического интерфейса **MongoDB Compass**. | Для создания базы данных с помощью консольного клиента **MongoDB Compass** надо нажать кнопку **Connect** на первой странице приложения. Откроется страница с информацией об имеющихся базах данных. Можно создавать базы данных с помощью как графического интерфейса, так и с помощью команд JavaScript.  Команды вводятся во встроенной оболочке, которая открывается щелчком по ссылке **\_MONGOSH** в нижнем левом углу окна.  Чтобы посмотреть, какая база данных используется в текущий момент времени, надо выполнить команду **db**. Для просмотра списка всех созданных баз служит команда **show dbs**.  Создать базу данных **Sales** и наполнить коллекцию **tovary** можно с помощью команд, которые вводятся по одной:  **use Sales**  **db.createCollection("tovary")**  **db.tovary.insert([ {"name": "Стол", price: 19, quantity: 50},**  **{"name": "Диван", price: 450, quantity: 10},**  **{"name": "Шкаф", price: 500, quantity: 20},**  **{"name": "Стол письменный", price: 49, quantity: 50},**  **{"name": "Стул", price: 20, quantity: 90}**  **])**  Получен ответ от сервера об успешном создании коллекции в виде объекта с ключом «ok», установленным в 1.    Чтобы просмотреть информацию о всех коллекциях базы данных используется команда **show collections**.  Функция для создания коллекций в MongoDB **db.createCollection(name, options)** принимает два аргумента. Первый − имя коллекции, второй, если необходимо, используется для уточнения параметров коллекции (имеет ли коллекция фиксированный размер, будет ли производиться автоматическая индексация коллекции, определяется размер коллекции и максимальное количество документов, если коллекция фиксированного размера).  Для удаления коллекций предусмотрена функция **db.collection\_name.drop()**, где вместо **collection\_name** указывается имя коллекции, которую необходимо удалить.  Используя *графический интерфейс* **MongoDB Compass** можно просмотреть все коллекции и их содержимое. Можно удалить коллекцию и вновь добавить другую.  После внесения изменений надо **обновлять** базу данных с помощью команды **Reload Data** (в пункте меню **View**).  Для просмотра статистики по определенной базе данных существует функция **db.stats()**, которая выводит информацию об имени базы, количестве коллекций, объектов, и т. д. |
| 2. Изучить способы добавления информации в базу данных **BaseMongo\_Yan** с помощью команд.  Добавить информацию с использованием графического интерфейса. | Существует несколько команд, с помощью которых добавляются данные:  − **insertOne()** в качестве аргумента принимает только один документ;  − **insertMany()** добавляет несколько документов в коллекцию;  − **insert()** − универсальная функция.  Добавление в коллекцию **tovary** одного документа:  **use Sales**  **db.tovary.insertOne({"name": "Полка", price: 21, quantity: 20})**  Проверить, добавлен ли документ, можно с помощью оператора:  **db.tovary.find()**  В случае, когда нужно за один запрос вставить несколько документов в коллекцию, целесообразно использовать команду **insertMany()**, проверить работу которой можно также с использованием **find()**.  Функция **insert()** является универсальной, она объединяет возможности **insertOne()** и **insertMany()**, а также предоставляет другие возможности добавления документов. Можно заранее определить документ и только затем выполнить его вставку.  Например, пусть создан *массив* документов с именем **tovaryArray**, содержащий информацию о двух товарах:  **tovaryArray = [**  **{"name": "Ковер", price: 201, quantity: 10},**  **{"name": "Пуфик", price: 32, quantity: 20}**  **]**  Теперь можно использовать функцию **insert()**, где в качестве параметра следует передать ранее созданный массив:  **db.tovary.insert(tovaryArray)**  Документы могут содержать вложенные объекты. Добавим в коллекцию **tovary** документ:  **db.tovary.insert({"name": "Стол",**  **characteristics: { material: "сосна", length: "150см"}, price: 19, quantity: 50})**  Здесь определяется вложенный объект с ключом **characteristics**. |
| 3. Изучить способы модификации и удаления информации из базы данных с **BaseMongo\_Yan** с помощью команд.  Опробовать способы изменения и удаления данных с использованием графического интерфейса **MongoDB Compass**. | Изменить документ можно с помощью команд:   * **updateOne** обновляет один документ в коллекции, соответствующий фильтру. Если совпадают несколько документов, то будет обновлен только первый совпадающий документ; * **updateMany** обновляет несколько документов в коллекции, соответствующих фильтру.   **db.tovary.updateOne({"name": "Стул"}, {$set: {price: 25}})**  Для удаления данных имеются команды:   * **deleteOne** удаляет один документ из коллекции, соответствующий фильтру. Если совпадают несколько документов, удаляется только первый совпадающий документ; * **deleteMany** удаляет несколько документов из коллекции, соответствующих фильтру.   **db.tovary.deleteOne({"name": "Стул"})**  Если надо полностью заменить один документ другим может использоваться функция **replaceOne**:  **db.tovary.replaceOne({"name": "Стол"}, {"name": "Стол",**  **price: 22, quantity: 60})** |
| 4. Опробовать операции удаления документов, коллекций и баз данных с помощью команд и с использованием графического интерфейса. | Для удаления документа из коллекции существует функция **remove()**, которая принимает два аргумента. Первый аргумент − это критерий, по которому документы будут удаляться, второй аргумент (не обязательный) позволяет допустить удаление только одного документа.  Чтобы удалить товар с названием **Стол** из коллекции **tovary** базы данных **Sales** можно использовать запрос:  **db.tovary.remove({name: "Стол"})**  После удаления документа будет выведен обновленный список предметов.  Чтобы удалить все документы в коллекции, но при этом коллекция должна сохраниться, нужно выполнить функцию **remove()** и в качестве критерия передать пустой объект:  **db.tovary.remove( {} )**  Если нужно удалить все документы и содержащую их коллекцию используется функция **drop()**.  **db.tovary.drop()**  **show collections**  Чтобы удалить всю базу данных нужно использовать команду **dropDatabase()**.  Например, чтобы удалить базу данных **Sales** и затем проверить ее отсутствие среди баз данных на сервере, надо использовать команду:  **db.dropDatabase({"dropped": "Sales", "ok": 1})**  После обновления базы с помощью **Reload Data** можно убедиться, что база данных была успешно удалена, как и все её коллекции и документы.  **show dbs** |
| 5.Добавить в базу данных **BaseMongo\_Yan** новые элементы. Произвести выборку данных, используя функции **find()** и **pretty()**. | Для получения документов из коллекции используются два метода:  − **find()** находит документы и отображает результат в неструктурированном формате;  − **pretty()** находит документы и отображает результат в структурированном формате.  Вывод записей, в которых содержится информация о товаре:  **db.tovary.find({"name": "Диван"})**  Чтобы вывести записи с несколькими критериями, необходимо перечислить эти критерии через запятую. Также несложно отыскать документ по элементу в массиве.  Помимо **find()** существует еще метод **findOne()**, который возвращает только один документ.  Метод **pretty()** используется совместно с **find():**  **db.tovary.find().pretty()**  В результате выполнения запроса будет выведено содержимое всей коллекции **probaIndex** в отформатированном виде. |

6. В соответствии со своим вариантом, номер которого определяет преподаватель, создать базу данных **BaseMongo\_Yan**. При этом использовать вложенные документы.

|  |  |
| --- | --- |
| **№ варианта** | **Исходная информация** |
| 15 | **Мастерская**. Коллекция документов с информациях о заказах: дана начала и окончания, имя мастера, имя клиента, техника, ее вид, список работ. |

[В начало практикума](#Оглавление)

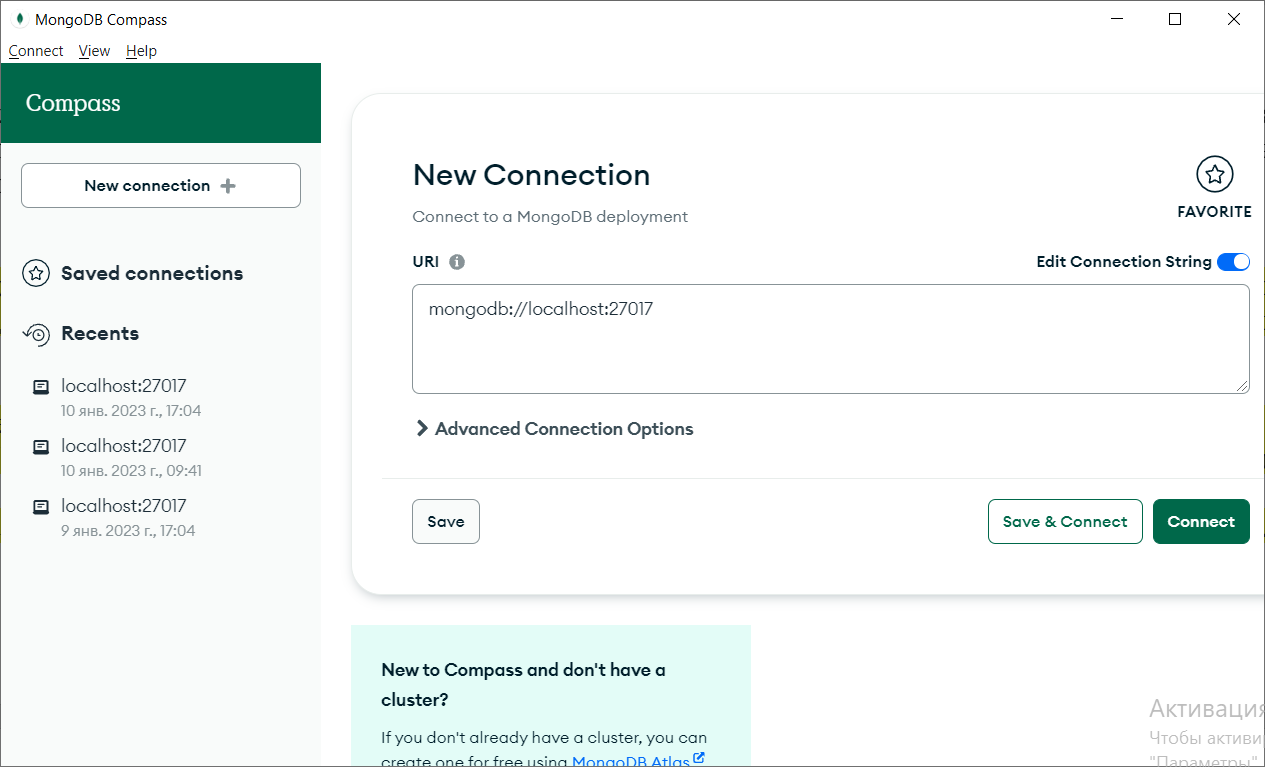
**Приложение 2**

**Установка базы данных MongoDB**

Для установки базы данных MongoDB нужно перейти на официальный сайт разработчиков: **https://www.mongodb.com/try/download/community-edition**.

На открывшейся странице следует выбрать сборку **MongoDB Community Server**, затем определить необходимые параметры, такие как версия, платформа, тип загружаемого архива или оставить все по умолчанию и нажать кнопку загрузки.

## Подключение к серверу и создание базы данных с помощью клиента MongoDB Compass

При установке сервера MongoDB также загружается консольный клиент MongoDB Compass для работы с базами данных. Используя графический интерфейс программы Compass, можно управлять данными, добавлять, изменять, удалять их.

На первой странице следует нажать на кнопку **Connect**. После успешного подключения откроется содержимое сервера.

На вкладке **DataBases** отображается список всех базы данных по текущему соединению, с размером хранилища каждого из них, количество коллекций и количество индексов.

Вкладка **Performance** предоставляет информацию о производительности сервера, о том, выполнение каких операций занимает больше всего времени, как используется память.

Для создания базы данных на вкладке **DataBases** служит кнопка **Create Databases**.

На странице **https://docs.mongodb.com/ecosystem/drivers/** находятся драйвера для PHP, C++, C# и др. Используя драйвера MongoDB может работать с приложениями на этих языках.

Для начала работы с коллекцией и документами нужно раскрыть панель, которая находится внизу программы. На окне подключения к БД данной строки не видно, она появляется только после подключения.

|  |  |  |
| --- | --- | --- |
|  |  |  |

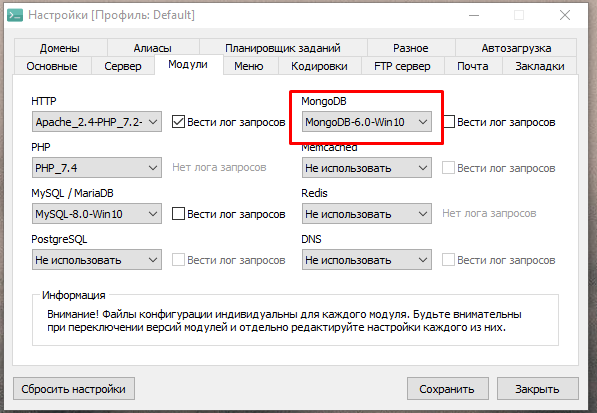
**Работа с MongoDB через командную строку**

Чтобы работать с MongoDB не с помощью клиента MongoDB Compass, а через командную строку требуется, чтобы сервер MongoDB был уже развернут.

При этом надо учесть, что при установке сервера Open Server, используемого веб-разработчиками, необходимо указать наличие MongoDB в OS.

После установки OS следует запустить приложение. В панели задач появится красный флажок, после нажатия на который надо выбрать команду **Настройки**.

В появившемся окне нужно перейти на вкладку **Модули** и в разделе MongoDB указать желаемую версию модуля. Нажать **Сохранить**.



В контекстном меню красного флажка выбрать команду **Запустить**. После того, как красный флажок станет зеленым, снова перейти в контекстное меню, перейти на вкладку **Дополнительно** и выбрать **Консоль**. В появившемся окне в командной строке ввести команду **mongo**.



Далее можно работать с базой данных MongoDB, для чего используются соответствующие команды.

[В начало практикума](#Оглавление)