

Министерство науки и высшего образования Российской  
Федерации  
Российский химико-технологический университет имени Д.  
И. Менделеева

---

Факультет цифровых технологий и химического  
инжиниринга  
Кафедра информационных компьютерных технологий

Мохов Марк Геннадьевич КС-34

**Лабораторная работа №7 по предмету управление  
данными:  
«ВЕДЕНИЕ В СУБД MONGODB. УСТАНОВКА MONGODB.  
НАЧАЛО РАБОТЫ С БД» .**

**ВАРИАНТ №11**

Научный руководитель:

Семёнов Геннадий Николаевич

Москва – 2024

# СОДЕРЖАНИЕ

1. Цель работы и практическое задание .....	3
1.1. Цель .....	3
1.2. Задание .....	3
2. Выполнение практического задания. ....	5
2.1. Установка MongoDB. ....	5
2.2. Проверьте работоспособность системы запуском клиента <code>mongo</code> или <code>mongosh</code> . ....	5
2.3. Выполнение методов .....	5
2.4. Получите список доступных БД .....	6
2.5. Создание Базы Данных. ....	7
2.6. Создание коллекции и добавление документов. ....	7
2.7. Список коллекций и Содержимого коллекции. ....	8
2.8. Создание экспорта .....	10
3. Вывод .....	11
4. Приложения .....	12

# 1. ЦЕЛЬ РАБОТЫ И ПРАКТИЧЕСКОЕ ЗАДАНИЕ

## 1.1. Цель

Овладеть практическими навыками установки СУБД MongoDB.

## 1.2. Задание

1. Установите MongoDB в зависимости от типов систем (32/64 бита).
2. Проверьте работоспособность системы запуском клиента `mongo` или `mongosh`.
3. Выполните методы:
  1. `db.help()`
  2. `db.help`
  3. `db.stats()`
4. Получите список доступных БД.
5. Создайте БД в соответствии с вариантом задания Лабораторной работы №1, имя БД вводится латиницей в формате «Фамилия\_группа», например, `db> use Ivanov_ks30`
6. Создайте коллекцию, вставив в нее документы в соответствии со строками таблицы вашего варианта задания Лабораторной работы №1 Например: Результаты ЕГЭ:

Дата экзамена	Школа №	Балл	Ученик	Возраст	Дисциплина	Экзаменатор
3.06.06	114	85	Иванов	18	Математика	Кузнецов
15.06.06	295	79	Петрова	17	Русский Язык	Филатова
2.06.06	1197	91	Сидорова	19	Математика	Кузнецов
2.06.06	114	62	Иванов	18	Русский Язык	Филатова
14.06.06	1197	77	Сидорова	19	Физика	Гришин

4.06.06	295	81	Петрова	17	Математика	Кузнецов
---------	-----	----	---------	----	------------	----------

Рисунок 1.1 – В СУБД MongoDB создайте базу данных

7. Выведете список текущих коллекций вашей БД.
8. Выведете содержимое созданной коллекции
9. В приложении MongoDB Compass сделайте экспорт коллекции вашей БД в файл \*.json.
10. Файл отчета в формате WORD и файл коллекции в формате \*.json загрузите на учебный портал в курсе «Управление данными» в раздел «Сдать Лаб.7»

## 2. ВЫПОЛНЕНИЕ ПРАКТИЧЕСКОГО ЗАДАНИЯ.

### 2.1. Установка MongoDB.

Из-за особенностей операционной системы, я устанавливал MongoDB Через docker контейнер.

В начале установил образ MongoDB:

```
1 doas docker pull mongo:latest
```

bash

Затем создал контейнер с названием :

```
1 doas docker run -d -p 27017:27017 --name mongo-lab7-8 mongo:latest
```

bash

### 2.2. Проверьте работоспособность системы запуском клиента mongo или mongosh.

В моём случае не было клиента mongo – я использовал mongosh, чтобы запустить mongosh для docker контейнера, выполнил:

```
1 doas docker exec -it mongo-lab7-8 mongosh
```

bash

После запуска открылся shell для управления базой данных. База данных по умолчанию была test.

### 2.3. Выполнение методов

1. `db.help()`: В результате выполнения метода, были выведены все возможные методы для объекта `db`. [Приложение 1]
2. `db.help`: Результат выполнения аналогичен предыдущему выводу.

3. db.stats():

```
1  {
2    "db": "test",
3    "collections": 0,
4    "views": 0,
5    "objects": 0,
6    "avgObjSize": 0,
7    "dataSize": 0,
8    "storageSize": 0,
9    "indexes": 0,
10   "indexSize": 0,
11   "totalSize": 0,
12   "scaleFactor": 1,
13   "fsUsedSize": 67391426560,
14   "fsTotalSize": 116441427968,
15   "ok": 1
16 }
17
```

## 2.4. Получите список доступных БД

Для получения списка доступных таблиц, я выполнил в mongosh следующую команду:

```
1 show databases
```

Результат выполнения:

```
1  {
2    "type": "ShowDatabasesResult",
3    "value": [
4      {
5        "name": "Mokhov_KS34",
6        "sizeOnDisk": 73728,
7        "empty": false
8      },
9      {
10       "name": "admin",
11       "sizeOnDisk": 40960,
12       "empty": false
13     },
14   ]
15 }
```

```

14    {
15        "name": "config",
16        "sizeOnDisk": 110592,
17        "empty": false
18    },
19    {
20        "name": "local",
21        "sizeOnDisk": 40960,
22        "empty": false
23    }
24 ]
25 }
26

```

## 2.5. Создание Базы Данных.

```
1 use Mokhov_KS34
```

js

## 2.6. Создание коллекции и добавление документов.

Для упрощения добавления документов, я решил создать функцию:

```

1  function insert_row(start_date, proj_name, family_name, age, proj_type,
duration, cost){
    db.projects.insertOne({ start_date: start_date, project_name: proj_name,
2  family_name: family_name, age: age, proj_type: proj_type, duration: duration, cost:
cost });
3  }

```

js

Затем вручную с помощью функции `insert_row` добавил все документы в соответствии с заданием первой лабораторной работы 11-го варианта.

Пример команды:

```
1 insert_row("3.02.04", "Проект1", "Иванов", 30, "Технология", 5, 1200)
```

js

## 2.7. Список коллекций и Содержимого коллекции.

Для вывода списка коллекций использовал команду:

```
1 show collections
```

js

результат:

```
1  {
2    "type": "ShowCollectionsResult",
3    "value": [
4      {
5        "name": "projects",
6        "badge": ""
7      }
8    ]
9  }
10
```

json

Для вывода содержимого коллекции использовал команду:

```
1 db.projects.find()
```

js

Результат вывода.

```
1  [
2    {
3      _id: ObjectId('6758258a4000176ef9e9496a'),
4      start_date: '3.02.04',
5      project_name: 'Проект1',
6      family_name: 'Иванов',
7      age: 30,
8      proj_type: 'Технология',
9      duration: 5,
10     cost: 1200
11   },
12   {
13     _id: ObjectId('675825e44000176ef9e9496b'),
14     start_date: '15.01.05',
15     project_name: 'Проект2',
16     family_name: 'Петрова',
17     age: 30,
18     proj_type: 'НИР',
19     duration: 4,
20     cost: 800
21   }
22 ]
```

js



```

21  },
22  {
23      _id: ObjectId('675826254000176ef9e9496c'),
24      start_date: '2.02.05',
25      project_name: 'Проект3',
26      family_name: 'Сидорова',
27      age: 41,
28      proj_type: 'Технология',
29      duration: 3,
30      cost: 950
31  },
32  {
33      _id: ObjectId('6758265d4000176ef9e9496d'),
34      start_date: '2.03.05',
35      project_name: 'Проект4',
36      family_name: 'Иванов',
37      age: 30,
38      proj_type: 'НИР',
39      duration: 5,
40      cost: 1000
41  },
42  {
43      _id: ObjectId('6758272b4000176ef9e9496e'),
44      start_date: '14.02.05',
45      project_name: 'Проект5',
46      family_name: 'Сидорова',
47      age: 41,
48      proj_type: 'ОКР',
49      duration: 3,
50      cost: 900
51  },
52  {
53      _id: ObjectId('6758275c4000176ef9e9496f'),
54      start_date: '4.04.05',
55      project_name: 'Проект6',
56      family_name: 'Петрова',
57      age: 29,
58      proj_type: 'Технология',
59      duration: 4,
60      cost: 1100
61  }
62 ]
63

```

## 2.8. Создание экспорта

Экспорт был создан командой `mongoexport`

```
1 doas docker -it mongo-lab7-8 mongoexport -d Mokhov_KS34 -c projects \
2 --jsonArray --quiet > lab7_dump.json
```

Содержания файла:

```
1 [{"_id":{"$_oid":"6758258a4000176ef9e9496a"},"start_date":"3.02.04","project_name":"'
  {"_id":{"$_oid":"675825e44000176ef9e9496b"},"start_date":"15.01.05","project_name":"'
  {"_id":{"$_oid":"675826254000176ef9e9496c"},"start_date":"2.02.05","project_name":"'
  {"_id":{"$_oid":"6758265d4000176ef9e9496d"},"start_date":"2.03.05","project_name":"'
  {"_id":{"$_oid":"6758272b4000176ef9e9496e"},"start_date":"14.02.05","project_name":"'
  {"_id":{"$_oid":"6758275c4000176ef9e9496f"},"start_date":"4.04.05","project_name":"'
2
```

### 3. ВЫВОД

В результате выполнения лабораторной работы я освоился с управлением и основным интерфейсом базы данных MongoDB, с тем как запускать её через docker контейнер и применения инструмента `mongodump` для создания экспорта определённой коллекции.

## 4. ПРИЛОЖЕНИЯ

1	]0;mongosh mongodb://127.0.0.1:27017/? directConnection=true&serverSelectionTimeoutMS=2000		<a href="#">txt</a>
2	[1m [33mDatabase Class: [0m		
3			
4	getMongo	Returns the current database connection	
5	getName	Returns the name of the DB	
6	getCollectionNames	Returns an array containing the names of all collections in the current database.	
7	getCollectionInfos	Returns an array of documents with collection information, i.e. collection name and options, for the current database.	
8	runCommand	Runs an arbitrary command on the database.	
9	adminCommand	Runs an arbitrary command against the admin database.	
10	aggregate	Runs a specified admin/diagnostic pipeline which does not require an underlying collection.	
11	getSiblingDB	Returns another database without modifying the db variable in the shell environment.	
12	getCollection	Returns a collection or a view object that is functionally equivalent to using the db.<collectionName>.	
13	dropDatabase	Removes the current database, deleting the associated data files.	
14	createUser	Creates a new user for the database on which the method is run. db.createUser() returns a duplicate user error if the user already exists on the database.	
15	updateUser	Updates the user's profile on the database on which you run the method. An update to a field completely replaces the previous field's values. This includes updates to the user's roles array.	
16	changeUserPassword	Updates a user's password. Run the method in the database where the user is defined, i.e. the database you created the user.	
17	logout	Ends the current authentication session. This function has no effect if the current session is not authenticated.	
18	dropUser	Removes the user from the current database.	
19	dropAllUsers	Removes all users from the current database.	

20	auth	Allows a user to authenticate to the database from within the shell.
21	grantRolesToUser	Grants additional roles to a user.
22	revokeRolesFromUser	Removes a one or more roles from a user on the current database.
23	getUser	Returns user information for a specified user. Run this method on the user's database. The user must exist on the database on which the method runs.
24	getUsers	Returns information for all the users in the database.
25	createCollection	Create new collection
26	createEncryptedCollection	Creates a new collection with a list of encrypted fields each with unique and auto-created data encryption keys (DEKs). This is a utility function that internally utilises ClientEncryption.createEncryptedCollection.
27	createView	Create new view
28	createRole	Creates a new role.
29	updateRole	Updates the role's profile on the database on which you run the method. An update to a field completely replaces the previous field's values.
30	dropRole	Removes the role from the current database.
31	dropAllRoles	Removes all roles from the current database.
32	grantRolesToRole	Grants additional roles to a role.
33	revokeRolesFromRole	Removes a one or more roles from a role on the current database.
34	grantPrivilegesToRole	Grants additional privileges to a role.
35	revokePrivilegesFromRole	Removes a one or more privileges from a role on the current database.
36	getRole	Returns role information for a specified role. Run this method on the role's database. The role must exist on the database on which the method runs.
37	getRoles	Returns information for all the roles in the database.
38	currentOp	Runs an aggregation using \$currentOp operator. Returns a document that contains information on in-progress operations for the database instance. For further information, see \$currentOp.
39	killOp	Calls the killOp command. Terminates an operation as specified by the operation ID. To find operations and their corresponding IDs, see \$currentOp or db.currentOp().

40	shutdownServer	Calls the shutdown command. Shuts down the current mongod or mongos process cleanly and safely. You must issue the db.shutdownServer() operation against the admin database.
41	fsyncLock	Calls the fsync command. Forces the mongod to flush all pending write operations to disk and locks the entire mongod instance to prevent additional writes until the user releases the lock with a corresponding db.fsyncUnlock() command.
42	fsyncUnlock	Calls the fsyncUnlock command. Reduces the lock taken by db.fsyncLock() on a mongod instance by 1.
43	version buildinfo command	returns the db version. uses the
44	serverBits buildInfo command	returns the db serverBits. uses the
45	isMaster	Calls the isMaster command
46	hello	Calls the hello command
47	serverBuildInfo uses the buildInfo command	returns the db serverBuildInfo.
48	serverStatus serverStatus command	returns the server stats. uses the
49	stats dbStats command	returns the db stats. uses the
50	hostInfo	Calls the hostInfo command
51	serverCmdLineOpts uses the getCmdLineOpts command	returns the db serverCmdLineOpts.
52	rotateCertificates	Calls the rotateCertificates command
53	printCollectionStats each collection in the db.	Prints the collection.stats for
54	getProfilingStatus uses the profile command	returns the db getProfilingStatus.
55	setProfilingLevel uses the profile command	returns the db setProfilingLevel.
56	setLogLevel the setParameter command	returns the db setLogLevel. uses
57	getLogComponents uses the getParameter command	returns the db getLogComponents.
58	cloneDatabase	deprecated, non-functional
59	cloneCollection	deprecated, non-functional
60	copyDatabase	deprecated, non-functional
61	commandHelp the passed in command with help: true	returns the db commandHelp. uses
62	listCommands	Calls the listCommands command
63	getLastErrorObj	Calls the getLastError command
64	getLastError	Calls the getLastError command

65	<code>printShardingStatus</code>	Calls <code>sh.status(verbose)</code>
66	<code>printSecondaryReplicationInfo</code>	Prints secondary replicaset information
67	<code>getReplicationInfo</code>	Returns replication information
68	<code>printReplicationInfo</code>	Formats <code>sh.getReplicationInfo</code>
69	<code>printSlaveReplicationInfo</code>	DEPRECATED. Use <code>db.printSecondaryReplicationInfo</code>
70	<code>setSecondaryOk</code>	This method is deprecated. Use <code>db.getMongo().setReadPref()</code> instead
71	<code>watch</code> the database	Opens a change stream cursor on the database
72	<code>sql</code>	(Experimental) Runs a SQL query against Atlas Data Lake. Note: this is an experimental feature that may be subject to change in future releases.
73	<code>checkMetadataConsistency</code> about metadata inconsistencies	Returns a cursor with information about metadata inconsistencies
74		
75		

Приложение 1 – Результат выполнения команды `db.help()`.