

Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра экономической информатики

ОТЧЕТ
Лабораторная работа №5

Выполнил: студент гр.

914302 Борисенко А.И.

Проверил: Лукашевич А.Э.

Минск 2022

1. Установка MongoDB для обеих типов систем (32/64 бита)
2. Проверка работоспособность системы запуском клиента mongo
3. Выполнение методов:
 - a) db.help()

```
> _MONGOSH
> db.help()
< Database Class

getMongo           Returns the current database connection
getName            Returns the name of the DB
getCollectionNames Returns an array containing the names of all collections in the current database.
getCollectionInfos Returns an array of documents with collection information, i.e. collection name and options,
                  for the current database.

runCommand          Runs an arbitrary command on the database.
adminCommand        Runs an arbitrary command against the admin database.
aggregate           Runs a specified admin/diagnostic pipeline which does not require an underlying collection.
getSiblingDB        Returns another database without modifying the db variable in the shell environment.
getCollection        Returns a collection or a view object that is functionally equivalent to using the db.
                  <collectionName>.

dropDatabase        Removes the current database, deleting the associated data files.
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.
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.
changeUserPassword   Updates a user's password. Run the method in the database where the user is defined, i.e.
```

- b) db.help
- c) db.stats()

```
> db.stats
< [Function: stats] AsyncFunction {
  apiVersions: [ 0, 0 ],
  returnsPromise: true,
  serverVersions: [ '0.0.0', '999.999.999' ],
  topologies: [ 'ReplSet', 'Sharded', 'LoadBalanced', 'Standalone' ],
  returnType: { type: 'unknown', attributes: {} },
  deprecated: false,
  platforms: [ 0, 1, 2 ],
  isDirectShellCommand: false,
  acceptsRawInput: false,
  shellCommandCompleter: undefined,
  help: [Function (anonymous)] Help
}
test>
```

4. Создание БД learn
5. Получение списка доступных БД
6. Создание коллекции unicorns, вставив в нее документ {name: 'Aurora', gender: 'f', weight: 450}
7. Просмотр списка текущих коллекций

8. Переименование коллекции unicorns

```
> db.createCollection("unicorns")
< { ok: 1 }
> show collections
< unicorns
> db.unicorns.insertOne({name: 'Aurora', gender: 'f', weight: 450})
< { acknowledged: true,
    insertedId: ObjectId("638141f8ccd148e86cc6ab99") }
> db.unicorns.renameCollection("beautiful_unicorns")
< { ok: 1 }
> show collections
< beautiful_unicorns
```

9. Просмотр статистики коллекции

10. Удаление коллекции

11. Удаление БД learn

```
> db.unicorns.stats
< [Function: stats] AsyncFunction {
  apiVersions: [ 0, 0 ],
  returnsPromise: true,
  serverVersions: [ '0.0.0', '999.999.999' ],
  topologies: [ 'ReplSet', 'Sharded', 'LoadBalanced', 'Standalone' ],
  returnType: { type: 'unknown', attributes: {} },
  deprecated: false,
  platforms: [ 0, 1, 2 ],
  isDirectShellCommand: false,
  acceptsRawInput: false,
  shellCommandCompleter: undefined,
  help: [Function (anonymous)] Help
}
> db.unicorns.drop
< [Function: drop] AsyncFunction {
  apiVersions: [ 1, Infinity ],
  returnsPromise: true,
  serverVersions: [ '0.0.0', '999.999.999' ],
  topologies: [ 'ReplSet', 'Sharded', 'LoadBalanced', 'Standalone' ],
  returnType: { type: 'unknown', attributes: {} },
  deprecated: false,
  platforms: [ 0, 1, 2 ],
  isDirectShellCommand: false,
  acceptsRawInput: false,
  shellCommandCompleter: undefined,
  help: [Function (anonymous)] Help
}
```

1) Создание базы данных learn

2) Заполнение коллекции единорогов unicorns

```
db.unicorns.insert({name: 'Horny', dob: new Date(1992,2,13,7,47), loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});
```

```

db.unicorns.insert({name: 'Aurora', dob: new Date(1991, 0, 24, 13, 0), loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});
db.unicorns.insert({name: 'Unicrom', dob: new Date(1973, 1, 9, 22, 10), loves: ['energo n', 'redbull'], weight: 984, gender: 'm', vampires: 182});
db.unicorns.insert({name: 'Rooodooles', dob: new Date(1979, 7, 18, 18, 44), loves: ['apple'], weight: 575, gender: 'm', vampires: 99});
db.unicorns.insert({name: 'Solnara', dob: new Date(1985, 6, 4, 2, 1), loves: ['apple', 'carrot', 'chocolate'], weight: 550, gender: 'f', vampires: 80});
db.unicorns.insert({name: 'Ayna', dob: new Date(1998, 2, 7, 8, 30), loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});
db.unicorns.insert({name: 'Kenny', dob: new Date(1997, 6, 1, 10, 42), loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});
db.unicorns.insert({name: 'Raleigh', dob: new Date(2005, 4, 3, 0, 57), loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});
db.unicorns.insert({name: 'Leia', dob: new Date(2001, 9, 8, 14, 53), loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});
db.unicorns.insert({name: 'Pilot', dob: new Date(1997, 2, 1, 5, 3), loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});
db.unicorns.insert({name: 'Nimue', dob: new Date(1999, 11, 20, 16, 15), loves: ['grape', 'carrot'], weight: 540, gender: 'f'});

```

3) Используя второй способ, вставка в коллекцию единорогов документа

```

{name: 'Dunx', dob: new Date(1976, 6, 18, 18, 18), loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165}

```

```

> use learn
< switched to db learn
> db.unicorns.insert({name: 'Horny', dob: new Date(1992,2,13,7,47), loves: ['carrot','papaya'], weight: 600, gender: 'm', vamp
db.unicorns.insert({name: 'Aurora', dob: new Date(1991, 0, 24, 13, 0), loves: ['carrot', 'grape'], weight: 450, gender: 'f',
db.unicorns.insert({name: 'Unicrom', dob: new Date(1973, 1, 9, 22, 10), loves: ['energon', 'redbull'], weight: 984, gender:
db.unicorns.insert({name: 'Rooodooles', dob: new Date(1979, 7, 18, 18, 44), loves: ['apple'], weight: 575, gender: 'm', vam
db.unicorns.insert({name: 'Solnara', dob: new Date(1985, 6, 4, 2, 1), loves: ['apple', 'carrot', 'chocolate'], weight: 550, ge
db.unicorns.insert({name: 'Ayna', dob: new Date(1998, 2, 7, 8, 30), loves: ['strawberry', 'lemon'], weight: 733, gender: 'f',
db.unicorns.insert({name: 'Kenny', dob: new Date(1997, 6, 1, 10, 42), loves: ['grape', 'lemon'], weight: 690, gender: 'm', v
db.unicorns.insert({name: 'Raleigh', dob: new Date(2005, 4, 3, 0, 57), loves: ['apple', 'sugar'], weight: 421, gender: 'm',
db.unicorns.insert({name: 'Leia', dob: new Date(2001, 9, 8, 14, 53), loves: ['apple', 'watermelon'], weight: 601, gender: 'f
db.unicorns.insert({name: 'Pilot', dob: new Date(1997, 2, 1, 5, 3), loves: ['apple', 'watermelon'], weight: 650, gender: 'm'
db.unicorns.insert({name: 'Nimue', dob: new Date(1999, 11, 20, 16, 15), loves: ['grape', 'carrot'], weight: 540, gender: 'f'

< DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{ acknowledged: true,
  insertedIds: { '0': ObjectId("63814402ced148e86cc6aba4") } }

```

4) Проверка содержимого коллекции с помощью метода find

```
> db.unicorns.insertOne({name: 'Dunx', dob: new Date(1976, 6, 18, 18, 18), loves: ['grape', 'watermelon'], weight: 704, gender: 'm'})
< { acknowledged: true,
  insertedId: ObjectId("6381444eccd148e86cc6aba5") }
> db.unicorns.find()
< { _id: ObjectId("63814402ccd148e86cc6ab9a"),
  name: 'Horny',
  dob: 1992-03-13T04:47:00.000Z,
  loves: [ 'carrot', 'papaya' ],
  weight: 600,
  gender: 'm',
  vampires: 63 }
{ _id: ObjectId("63814402ccd148e86cc6ab9b"),
  name: 'Aurora',
  dob: 1991-01-24T10:00:00.000Z,
  loves: [ 'carrot', 'grape' ],
  weight: 450,
  gender: 'f',
  vampires: 43 }
{ _id: ObjectId("63814402ccd148e86cc6ab9c"),
  name: 'Unicrom',
  dob: 1973-02-09T19:10:00.000Z,
  loves: [ 'energon', 'redbull' ],
  weight: 984,
  gender: 'm',
  vampires: 182 }
{ _id: ObjectId("63814402ccd148e86cc6ab9d"),
  name: 'Boooooodles',
  dob: 1979-08-18T15:44:00.000Z,
  loves: [ 'apple' ],
  weight: 575,
  gender: 'm',
  vampires: 99 }
{ _id: ObjectId("63814402ccd148e86cc6ab9e"),
  name: 'Solnara',
```

1) Формирование запросов для вывода списков самцов и самок единорогов. Ограничение списка самок первыми тремя особями. Сортировка списки по имени.

```
> db.unicorns.find({gender: "m"}).sort({name: 1}).limit(3)
< { _id: ObjectId("6381444eccd148e86cc6aba5"),
  name: 'Dunx',
  dob: 1976-07-18T15:18:00.000Z,
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165 }
{ _id: ObjectId("63814402ccd148e86cc6ab9a"),
  name: 'Horny',
  dob: 1992-03-13T04:47:00.000Z,
  loves: [ 'carrot', 'papaya' ],
  weight: 600,
  gender: 'm',
  vampires: 63 }
{ _id: ObjectId("63814402ccd148e86cc6aba0"),
  name: 'Kenny',
  dob: 1997-07-01T06:42:00.000Z,
  loves: [ 'grape', 'lemon' ],
  weight: 690,
  gender: 'm',
  vampires: 39 }
learn>
```

```
> db.unicorns.find({gender: "f"}).sort({name: 1}).limit(3)
< { _id: ObjectId("63814402ccd148e86cc6ab9b"),
  name: 'Aurora',
  dob: 1991-01-24T10:00:00.000Z,
  loves: [ 'carrot', 'grape' ],
  weight: 450,
  gender: 'f',
  vampires: 43 }
{ _id: ObjectId("63814402ccd148e86cc6ab9f"),
  name: 'Ayna',
  dob: 1998-03-07T05:30:00.000Z,
  loves: [ 'strawberry', 'lemon' ],
  weight: 733,
  gender: 'f',
  vampires: 40 }
{ _id: ObjectId("63814402ccd148e86cc6aba2"),
  name: 'Leia',
  dob: 2001-10-08T10:53:00.000Z,
  loves: [ 'apple', 'watermelon' ],
  weight: 601,
  gender: 'f',
  vampires: 33 }
learn>
```

2) Поиск всех самок, которые любят carrot. Ограничение этого списка первой особью с помощью функций `findOne` и `limit`.

```
> db.unicorns.find({gender: "m"}, {loves: 'carrot'}).limit(1)
< { _id: ObjectId("63814402ccd148e86cc6ab9a"), loves: 'carrot' }
```

Модификация запроса для вывода списков самцов единорогов, исключив из результата информацию о дате рождения и поле.

```
> db.unicorns.find({gender: "m"}, {gender: 0}, {dob: 0}).limit(1)
< { _id: ObjectId("63814402ccd148e86cc6ab9a"),
  name: 'Horny',
  dob: 1992-03-13T04:47:00.000Z,
  loves: [ 'carrot', 'papaya' ],
  weight: 600,
  vampires: 63 }
learn>
```

Вывод списка единорогов в обратном порядке добавления.

```
> db.unicorns.find().sort({$natural: -1})
< { _id: ObjectId("6381444eccd148e86cc6aba5"),
  name: 'Dunx',
  dob: 1976-07-18T15:18:00.000Z,
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165 }
{ _id: ObjectId("63814402ccd148e86cc6aba4"),
  name: 'Nimne',
  dob: 1999-12-20T13:15:00.000Z,
  loves: [ 'grape', 'carrot' ],
  weight: 540,
  gender: 'f' }
{ _id: ObjectId("63814402ccd148e86cc6aba3"),
  name: 'Pilot',
```

Вывод списка самок единорогов весом от полутонны до 700 кг, исключив вывод идентификатора.

```
> db.unicorns.find({weight: {$gt: 500, $lt: 700}, gender: "f"}, {_id:false})
< { name: 'Solnara',
  dob: 1985-07-03T22:01:00.000Z,
  loves: [ 'apple', 'carrot', 'chocolate' ],
  weight: 550,
  gender: 'f',
  vampires: 80 }
{ name: 'Leia',
  dob: 2001-10-08T10:53:00.000Z,
  loves: [ 'apple', 'watermelon' ],
  weight: 601,
  gender: 'f',
  vampires: 33 }
{ name: 'Nimue',
  dob: 1999-12-20T13:15:00.000Z,
  loves: [ 'grape', 'carrot' ],
  weight: 540,
  gender: 'f' }
learn>
```

Вывести список самцов единорогов весом от полутонны и предпочитающих grape и lemon, исключив вывод идентификатора.

```
> db.unicorns.find({weight: {$gt: 500}, gender: "m", $or: [{loves: "lemon"}, {loves: "grape"}]}, {_id:false})
< { name: 'Kenny',
  dob: 1997-07-01T06:42:00.000Z,
  loves: [ 'grape', 'lemon' ],
  weight: 690,
  gender: 'm',
  vampires: 39 }
{ name: 'Dunx',
  dob: 1976-07-18T15:18:00.000Z,
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165 }
learn>
```

Найти всех единорогов, не имеющих ключ vampires

```
> db.unicorns.find({vampires: {$exists: false}})
< { _id: ObjectId("63814402ccd148e86cc6aba4"),
  name: 'Nimue',
  dob: 1999-12-20T13:15:00.000Z,
  loves: [ 'grape', 'carrot' ],
  weight: 540,
  gender: 'f' }
learn>
```