

Задание

1. Установить redis (сервер и клиент)
2. Продемонстрировать простые запросы с string, list, set
3. Установить MongoDB (сервер mongod и клиент mongo)
4. Создать базу данных для своего варианта
Добавить тестовые данные
5. Выполнить основные запросы (минимум 2)

1 Установить redis (сервер и клиент)

```
name_regex@-PC:~$ sudo apt-get update name_regex@-PC:~$ sudo apt-get install redis
```

Проверка работы

```
name_regex@-PC:~$ sudo service redis-server start
Starting redis-server: redis-server.
name_regex@-PC:~$ redis-cli
127.0.0.1:6379> ping
PONG
127.0.0.1:6379>
```

Сервер redis запущен

2 Продемонстрировать простые запросы с string, list, set

2.1 string

```
127.0.0.1:6379> set mykey "Hello, world!"
OK
```

Получим по ключу значение и длину строки

```
127.0.0.1:6379> get mykey
"Hello, world!"
127.0.0.1:6379> strlen mykey
(integer) 13
127.0.0.1:6379>
```

2.2 list

```
127.0.0.1:6379> rpush mylist i1 i2 i3  
(integer) 3
```

Получим значение по индексу, а также выведем все элементы списка

```
127.0.0.1:6379> lindex mylist 0  
"i1"  
127.0.0.1:6379> lindex mylist -1  
"i3"  
127.0.0.1:6379> lrange mylist 0 -1  
1) "i1"  
2) "i2"  
3) "i3"
```

2.3 set

Создадим множество

```
127.0.0.1:6379> sadd myset s1 s2 s3 s4 s5  
(integer) 5
```

Все элементы

```
127.0.0.1:6379> smembers myset  
1) "s4"  
2) "s3"  
3) "s1"  
4) "s2"  
5) "s5"
```

Проверка наличия элементов

```
127.0.0.1:6379> sismember myset s1  
(integer) 1  
127.0.0.1:6379> sismember myset i1  
(integer) 0
```

Удаление элементов

```
127.0.0.1:6379> srem myset s2 s3
(integer) 2
127.0.0.1:6379> smembers myset
1) "s1"
2) "s5"
3) "s4"
```

3 Установить MongoDB (сервер mongod и клиент mongo)

```
C:\Users\Пользователь>mongosh
Current Mongosh Log ID: 6640f8a32caa263df046b798
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.5
Using MongoDB:      7.0.9
Using Mongosh:       2.2.5

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
The server generated these startup warnings when booting
  2024-05-12T20:09:02.201+03:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----
```

4 Создать базу данных для своего варианта. Добавить тестовые данные

```
test> use mydb1
switched to db mydb1
```

Создам коллекцию players и заполню её

```
mydb1> db.createCollection("players")
{ ok: 1 }
```

```

mydb1> db.players.insertMany([{"id_player":1,"nickname":"osha"}, {"id_player":2,"nickname":"yellowFlash"}, {"id_player":3,"nickname":"Sasha"}, {"id_player":4, "nickname":"hunter"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6640fca92caa263df046b799'),
    '1': ObjectId('6640fca92caa263df046b79a'),
    '2': ObjectId('6640fca92caa263df046b79b'),
    '3': ObjectId('6640fca92caa263df046b79c')
  }
}

```

Создам коллекцию tanks и заполню её

```

mydb1> db.createCollection("tanks")
{ ok: 1 }
mydb1> db.tanks.insertMany([{"id_tank":1,"name":"T-34","type":"AT"}, {"id_tank":2,"name":"BT-2","type":"AT"}, {"id_tank":3,"name":"MATILDA","type":"HT"}, {"id_tank":4, "name":"LT-42","type":"LT"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6640ff512caa263df046b79d'),
    '1': ObjectId('6640ff512caa263df046b79e'),
    '2': ObjectId('6640ff512caa263df046b79f'),
    '3': ObjectId('6640ff512caa263df046b7a0')
  }
}

```

Создам коллекцию presence и заполню её

```

mydb1> db.createCollection("presence")
{ ok: 1 }
mydb1> db.presence.insertMany([{"id_presence":1,"id_player":1,"id_tank":1},{ "id_presence":2,"id_player":1,"id_tank":2},{ "id_presence":3,"id_player":1,"id_tank":3},{ "id_presence":4,"id_player":1,"id_tank":4},{ "id_presence":5,"id_player":2, "id_tank":3},{ "id_presence":6, "id_player":3, "id_tank":4}, {"id_presence":7,"id_player":4,"id_tank":2}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6641025f2caa263df046b7a1'),
    '1': ObjectId('6641025f2caa263df046b7a2'),
    '2': ObjectId('6641025f2caa263df046b7a3'),
    '3': ObjectId('6641025f2caa263df046b7a4'),
    '4': ObjectId('6641025f2caa263df046b7a5'),
    '5': ObjectId('6641025f2caa263df046b7a6'),
    '6': ObjectId('6641025f2caa263df046b7a7')
  }
}

```

5 Выполнить основные запросы (минимум 2)

Выведем всех игроков

```
mydb1> db.players.find({})
[
  {
    _id: ObjectId('6640fca92caa263df046b799'),
    id_player: 1,
    nickname: 'osha'
  },
  {
    _id: ObjectId('6640fca92caa263df046b79a'),
    id_player: 2,
    nickname: 'yellowFlash'
  },
  {
    _id: ObjectId('6640fca92caa263df046b79b'),
    id_player: 3,
    nickname: 'Sasha'
  },
  {
    _id: ObjectId('6640fca92caa263df046b79c'),
    id_player: 4,
    nickname: 'hunter'
  }
]
```

Выведу конкретного игрока

```
mydb1> db.players.find({"nickname":"osha"})
[
  {
    _id: ObjectId('6640fca92caa263df046b799'),
    id_player: 1,
    nickname: 'osha'
  }
]
```

Вывести все танки, которые есть у игрока (например, id_player = 1)

```
mydb1> db.players.aggregate([ { $match: { id_player: 1 } }, { $lookup: { from: "presence", localField: "id_player", foreignField: "id_player", as: "player_tanks" } }, { $unwind: "$player_tanks" }, { $lookup: { from: "tanks", localField: "player_tanks.id_tank", foreignField: "id_tank", as: "tank_info" } }, { $project: { _id: 0, player_id: 1, tank_names: "$tank_info.name" } } ] )
[
  { tank_names: [ 'T-34' ] },
  { tank_names: [ 'BT-2' ] },
  { tank_names: [ 'MATILDA' ] },
  { tank_names: [ 'LT-42' ] }
]
```

```
db.players.aggregate([ { $match: { id_player: 1 } }, { $lookup: { from: "presence", localField: "id_player", foreignField: "id_player", as: "player_tanks" } }, { $unwind: "$player_tanks" }, { $lookup: { from: "tanks", localField: "player_tanks.id_tank", foreignField: "id_tank", as: "tank_info" } }, { $project: { _id: 0, player_id: 1, tank_names: "$tank_info.name" } } ] )
```

Танков каких типов больше всего у игроков

```
db.players.aggregate([
  { $lookup: {
    from: "presence",
    localField: "id_player",
    foreignField: "id_player",
    as: "player_tanks"
  } },
  { $unwind: "$player_tanks" },
  { $lookup: {
    from: "tanks",
    localField: "player_tanks.id_tank",
    foreignField: "id_tank",
    as: "tank_info"
  } },
  { $unwind: "$tank_info" },
  { $group: {
    _id: "$tank_info.type",
    count: { $sum: 1 }
  } },
  { $sort: { count: -1 } },
  { $limit: 1 }
])
```

```
mydb1> db.players.aggregate([{$lookup: {from: "presence",localField: "id_player",foreignField: "id_player",as: "player_tanks"},{ $unwind: "$player_tanks" },{$lookup: {from: "tanks",localField: "player_tanks.id_tank",foreignField: "id_tank",as: "tank_info" },{$unwind: "$tank_info" },{$group: {_id: "$tank_info.type",count: { $sum: 1 }} },{$sort: { count: -1 } },{$limit: 1 }])
[ { _id: 'AT', count: 3 } ]
mydb1> █
```