

# Отчет Лаб 5

Ковалев В.Д. к32421

1. Создайте базу данных *learn*.
2. Заполните коллекцию единорогов *unicorns*:

My Queries

```
> _MONGOSH
> use learn
< switched to db learn
> db.unicorns.insertMany([
  {name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63},
  {name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43},
  {name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182},
  {name: 'Rooodoodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99},
  {name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80},
  {name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40},
  {name:'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39},
  {name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2},
  {name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33},
  {name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54},
  {name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'}
]);
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("649bfb51a743bf6bbbf217e1"),
    '1': ObjectId("649bfb51a743bf6bbbf217e2"),
    '2': ObjectId("649bfb51a743bf6bbbf217e3"),
    '3': ObjectId("649bfb51a743bf6bbbf217e4"),
    '4': ObjectId("649bfb51a743bf6bbbf217e5"),
    '5': ObjectId("649bfb51a743bf6bbbf217e6"),
    '6': ObjectId("649bfb51a743bf6bbbf217e7"),
    '7': ObjectId("649bfb51a743bf6bbbf217e8"),
    '8': ObjectId("649bfb51a743bf6bbbf217e9"),
    '9': ObjectId("649bfb51a743bf6bbbf217ea"),
    '10': ObjectId("649bfb51a743bf6bbbf217eb")
  }
}
```

Используя второй способ, вставьте в коллекцию единорогов документ:

```

> document=({name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165})
< {
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
> db.unicorns.insert(document)
< DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("649bfbfea743bf6bbbf217ec")
  }
}

```

1. Сформируйте запросы для вывода списков самцов и самок единорогов. Ограничьте список самок первыми тремя особями. Отсортируйте списки по имени.
2. Найдите всех самок, которые любят carrot. Ограничьте этот список первой особью с помощью функций *findOne* и *limit*.

```
> db. unicorns . find ( {gender:"m"}).sort({name:1})
< {
  _id: ObjectId("649bfbf6ea743bf6bbbf217ec"),
  name: 'Dunx',
  loves: [
    'grape',
    'watermelon'
  ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
{
  _id: ObjectId("649bf51a743bf6bbbf217e1"),
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
{
  _id: ObjectId("649bf51a743bf6bbbf217e7"),
  name: 'Kenny',
  loves: [
    'grape',
```

> MONGODB

```
> db.unicorns.find({gender:"f"}).limit(3).sort({name:1})
```

```
< {
```

```
  _id: ObjectId("649bfb51a743bf6bbbf217e2"),
```

```
  name: 'Aurora',
```

```
  loves: [
```

```
    'carrot',
```

```
    'grape'
```

```
  ],
```

```
  weight: 450,
```

```
  gender: 'f',
```

```
  vampires: 43
```

```
}
```

```
{
```

```
  _id: ObjectId("649bfb51a743bf6bbbf217e6"),
```

```
  name: 'Ayna',
```

```
  loves: [
```

```
    'strawberry',
```

```
    'lemon'
```

```
  ],
```

```
  weight: 733,
```

```
  gender: 'f',
```

```
  vampires: 40
```

```
}
```

```
{
```

```
  _id: ObjectId("649bfb51a743bf6bbbf217e9"),
```

```
  name: 'Leia',
```

```
  loves: [
```

```
    'apple',
```

```
    'watermelon'
```

```
  ],
```

```
  weight: 601,
```

```

> db.unicorns.find({gender: 'f', loves:'carrot'}).limit(1)
< {
  _id: ObjectId("649bfb51a743bf6bbbf217e2"),
  name: 'Aurora',
  loves: [
    'carrot',
    'grape'
  ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
learn>|

```

```

> db.unicorns.find({gender: 'm'}, {loves: 0, gender: 0})
< {
  _id: ObjectId("649bfb51a743bf6bbbf217e1"),
  name: 'Horny',
  weight: 600,
  vampires: 63
}
{
  _id: ObjectId("649bfb51a743bf6bbbf217e3"),
  name: 'Unicrom',
  weight: 984,
  vampires: 182
}
{
  _id: ObjectId("649bfb51a743bf6bbbf217e4"),
  name: 'Roooooodles',
  weight: 575,
  vampires: 99
}
{
  _id: ObjectId("649bfb51a743bf6bbbf217e7"),
  name: 'Kenny',
  weight: 690,
  vampires: 39
}
{
  _id: ObjectId("649bfb51a743bf6bbbf217e8"),

```

```
> db.unicorns.find().sort({$natural: -1})
< {
  _id: ObjectId("649bfbf6ea743bf6bbbf217ec"),
  name: 'Dunx',
  loves: [
    'grape',
    'watermelon'
  ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
{
  _id: ObjectId("649bf51a743bf6bbbf217eb"),
  name: 'Nimue',
  loves: [
    'grape',
    'carrot'
  ],
  weight: 540,
  gender: 'f'
}
{
  _id: ObjectId("649bf51a743bf6bbbf217ea"),
  name: 'Pilot',
  loves: [
    'apple',
    'watermelon'
  ],

```

```
> db.unicorns.find({}, {loves: {$slice : 1}, _id: 0});
< {
  name: 'Horny',
  loves: [
    'carrot'
  ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
{
  name: 'Aurora',
  loves: [
    'carrot'
  ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
{
  name: 'Unicrom',
  loves: [
    'energon'
  ],
  weight: 984,
  gender: 'm',
  vampires: 182
}
{
  name: 'Roooooodles',
```

*Вывести список самцов единорогов весом от полутонны и предпочитающих грапе и летоп, исключив вывод идентификатора.*

```

> db.unicorns.find({gender: 'f', weight: {$gte: 500, $lte: 700}}, {_id: 0})
< {
  name: 'Solnara',
  loves: [
    'apple',
    'carrot',
    'chocolate'
  ],
  weight: 550,
  gender: 'f',
  vampires: 80
}
{
  name: 'Leia',
  loves: [
    'apple',
    'watermelon'
  ],
  weight: 601,
  gender: 'f',
  vampires: 33
}
{
  name: 'Nimue',
  loves: [
    'grape',
    'carrot'
  ],

```

```

> db.unicorns.find({gender: 'm', weight: {$gte: 500}, loves: {$all: ['grape', 'lemon']}}, {_id: 0})
< {
  name: 'Kenny',
  loves: [
    'grape',
    'lemon'
  ],
  weight: 690,
  gender: 'm',
  vampires: 39
}

```



```

> db.unicorns.find({vampires: {$exists: false}})
< {
  _id: ObjectId("649bfb51a743bf6bbbf217eb"),
  name: 'Nimue',
  loves: [
    'grape',
    'carrot'
  ],
  weight: 540,
  gender: 'f'
}

```

```

> db.unicorns.find({gender: 'm'}, {name: 1, loves: {$slice : 1}, _id: 0}).sort({name: 1});
< {
  name: 'Dunx',
  loves: [
    'grape'
  ]
}
{
  name: 'Horny',
  loves: [
    'carrot'
  ]
}
{
  name: 'Kenny',
  loves: [
    'grape'
  ]
}
{
  name: 'Pilot',
  loves: [
    'apple'
  ]
}
{
  name: 'Raleigh',

```

1. Создайте коллекцию *towns*, включающую следующие документы:

```

> db.town.insert({name: "Punxsutawney ",
  populatiuon: 6200,
  last_sensus: ISODate("2008-01-31"),
  famous_for: [""],
  mayor: {
    name: "Jim Wehrle"
  }}
)
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("649c029aa743bf6bbbf217ee")
  }
}
learn>

```

```

> db.town.insert({name: "New York",
  populatiuon: 22200000,
  last_sensus: ISODate("2009-07-31"),
  famous_for: ["status of liberty", "food"],
  mayor: {
    name: "Michael Bloomberg",
    party: "I"}}
)
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("649c02e8a743bf6bbbf217ef")
  }
}
learn>

```

```

> db.town.insert({name: "Portland",
  populatiuon: 528000,
  last_sensus: ISODate("2009-07-20"),
  famous_for: ["beer", "food"],
  mayor: {
    name: "Sam Adams",
    party: "D"}})
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("649c030fa743bf6bbbf217f0")
  }
}
learn>

```

```

> db.town.find({"mayor.party": 'I'}, {name: 1, mayor: 1, _id: 0});
< {
  name: 'New York',
  mayor: {
    name: 'Michael Bloomberg',
    party: 'I'
  }
}
learn>

```

```

<
> db.town.find({"mayor.party": {$exists: false}}, {name: 1, mayor: 1, _id: 0});
< {
  name: 'Punxsutawney ',
  mayor: {
    name: 'Jim Wehrle'
  }
}
learn>

```

```

> fn = function() { return this.gender=='m'; }
< [Function: fn]
learn>

```

```

> var cursor = db.unicorns.find({gender: 'm'});null;
< null
> cursor.limit(2).sort({name: 1});null;
< null
> cursor.forEach(function(obj){ print(obj); })
< {
  _id: ObjectId("649bfbfea743bf6bbbf217ec"),
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
< {
  _id: ObjectId("649bfb51a743bf6bbbf217e1"),
  name: 'Horny',
  loves: [ 'carrot', 'papaya' ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
learn>|

```

```

> db.unicorns.find({gender: 'f', weight: {$gte: 500, $lte: 600}}).count()
< 2
> db.unicorns.distinct("loves")
< [
  'apple',      'carrot',
  'chocolate', 'energon',
  'grape',      'lemon',
  'papaya',     'redbull',
  'strawberry', 'sugar',
  'watermelon'
]
> db.unicorns.aggregate({"$group": {_id: "Gender", count: {$sum: 1}}})
< {
  _id: 'Gender',
  count: 12
}
learn>|

```

```

> db.unicorns.find()
< {
  _id: ObjectId("649bf51a743bf6bbbf217e1"),
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
{
  _id: ObjectId("649bf51a743bf6bbbf217e2"),
  name: 'Aurora',
  loves: [
    'carrot',
    'grape'
  ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
{
  _id: ObjectId("649bf51a743bf6bbbf217e3"),
  name: 'Unicrom',
  loves: [
    'energon',
    'redbull'
  ],
  weight: 700,
  gender: 'm',
  vampires: 50
}

```

```

> db.unicorns.updateOne({name: "Anna", gender: 'f'}, {$set: {weight: 800, vampires: 51}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
learn> |

```

```

> db.unicorns.updateOne({name: "Anna", gender: 'f'}, {$set: {weight: 800, vampires: 51}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
> db.unicorns.find({name: "Ayna"})
< {
  _id: ObjectId("649bfb51a743bf6bbbf217e6"),
  name: 'Ayna',
  loves: [
    'strawberry',
    'lemon'
  ],
  weight: 733,
  gender: 'f',
  vampires: 40
}
learn>|

```

```

> db.unicorns.find({name: "Raleigh"})
< {
  _id: ObjectId("649bfb51a743bf6bbbf217e8"),
  name: 'Raleigh',
  loves: [
    'apple',
    'sugar'
  ],
  weight: 421,
  gender: 'm',
  vampires: 2
}
learn>

```

```

> db.unicorns.updateOne({name: "Raleigh", gender: 'm'}, {$push: {loves: "redbull"}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn>

```

```
}  
> db.unicorns.find({name: "Raleigh"})  
< {  
  _id: ObjectId("649bfb51a743bf6bbbf217e8"),  
  name: 'Raleigh',  
  loves: [  
    'apple',  
    'sugar',  
    'redbull'  
  ],  
  weight: 421,  
  gender: 'm',  
  vampires: 2  
}  
learn>
```

```
> db.unicorns.updateMany({gender: 'm'}, {$inc: {vampires: 5}})  
< {  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 7,  
  modifiedCount: 7,  
  upsertedCount: 0  
}  
learn>
```

```

> db.unicorns.find({gender: 'm'})
< {
  _id: ObjectId("649bfb51a743bf6bbbf217e1"),
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ],
  weight: 600,
  gender: 'm',
  vampires: 68
}
{
  _id: ObjectId("649bfb51a743bf6bbbf217e3"),
  name: 'Unicrom',
  loves: [
    'energon',
    'redbull'
  ],
  weight: 984,
  gender: 'm',
  vampires: 187
}
{
  _id: ObjectId("649bfb51a743bf6bbbf217e4"),
  name: 'Rooooooodles',
  loves: [
    'apple'
  ],

```

```

> db.town.updateOne({name: "Portland"}, {$unset: {"mayor.party": 1}});
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn>

```



```
> db.unicorns.updateOne({name: "Pilot", gender: 'm'}, {$push: {loves: "chocolate"}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn>|
```

```
> db.unicorns.find({name: "Pilot"})
< {
  _id: ObjectId("649bfb51a743bf6bbbf217ea"),
  name: 'Pilot',
  loves: [
    'apple',
    'watermelon',
    'chocolate'
  ],
  weight: 650,
  gender: 'm',
  vampires: 59
}
learn>
```

```
> db.unicorns.find({name: "Aurora"})
< {
  _id: ObjectId("649bfb51a743bf6bbbf217e2"),
  name: 'Aurora',
  loves: [
    'carrot',
    'grape'
  ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
learn>
```

```

> db.unicorns.updateOne({name: "Aurora", gender: 'f'}, {$addToSet: {loves: {$each: ['sugar', 'lemon']}}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn>

```

```

> db.town.find()
< {
  _id: ObjectId("649c029aa743bf6bbbf217ee"),
  name: 'Punxsutawney ',
  populatiuon: 6200,
  last_sensus: 2008-01-31T00:00:00.000Z,
  famous_for: [
    ''
  ],
  mayor: {
    name: 'Jim Wehrle'
  }
}
{
  _id: ObjectId("649c02e8a743bf6bbbf217ef"),
  name: 'New York',
  populatiuon: 22200000,
  last_sensus: 2009-07-31T00:00:00.000Z,
  famous_for: [
    'status of liberty',
    'food'
  ],
  mayor: {
    name: 'Michael Bloomberg',
    party: 'I'
  }
}
{

```

```

> db.towns.remove({"mayor.party": {$exists: false}})
< DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
< {
  acknowledged: true,
  deletedCount: 0
}
learn>

```

```
> db.town.remove({})
< {
  acknowledged: true,
  deletedCount: 3
}
learn>|
```

```
>
> show collections
< town
  unicorns
learn>|
```

```
> db.unicorns.update({name: 'Dunx'}, {$set: {location: {$ref:"locations", $id: "fst"}}})
< DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn>|
```

```
>
> db.unicorns.update({name: 'Ayna'}, {$set: {location: {$ref:"locations", $id: "mtn"}}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn>|
```

```
>
> db.unicorns.update({name: 'Horny'}, {$set: {location: {$ref:"locations", $id: "bch"}}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn>
```

```

> db.unicorns.find({location: {$exists: true}})
< {
  _id: ObjectId("649bfb51a743bf6bbbf217e1"),
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ],
  weight: 600,
  gender: 'm',
  vampires: 68,
  location: DBRef("locations", 'bch')
}
{
  _id: ObjectId("649bfb51a743bf6bbbf217e6"),
  name: 'Ayna',
  loves: [
    'strawberry',
    'lemon'
  ],
  weight: 733,
  gender: 'f',
  vampires: 40,
  location: DBRef("locations", 'mtn')
}
{
  _id: ObjectId("649bfbfea743bf6bbbf217ec"),
  name: 'Dunx',
  loves: [
    'grape',

```

```

> db.unicorns.ensureIndex({"name" : 1}, {"unique" : true})
< [ 'name_1' ]
learn>

```

```

<
> db.numbers.explain("executionStats").find().sort({$natural: -1}).limit(4)
< {
  explainVersion: '1',
  queryPlanner: {
    namespace: 'learn.numbers',
    indexFilterSet: false,
    parsedQuery: {},
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'EOF'
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 0,
    executionTimeMillis: 0,
    totalKeysExamined: 0,
    totalDocsExamined: 0,
    executionStages: {
      stage: 'EOF',
      nReturned: 0,
      executionTimeMillisEstimate: 0,
      works: 1,
      advanced: 0,
      needTime: 0,
      needYield: 0,

```

```

    }
  }
}
> db.numbers.ensureIndex({"value" : -1})
< [ 'value_-1' ]
learn>

```