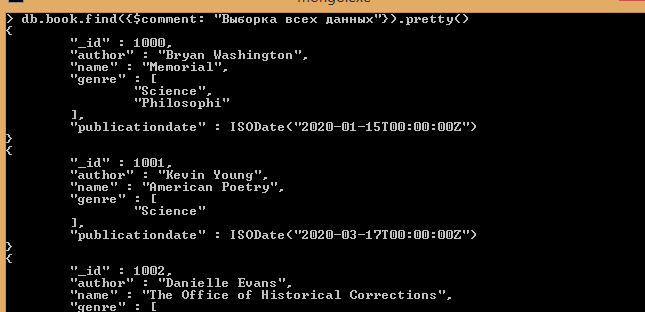
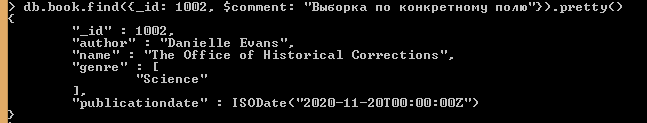
**Поляков Илья, 31 группа**

**1. Выборка всех данных, фильтрация по конкретному полю, массиву, встроенному объекту.**

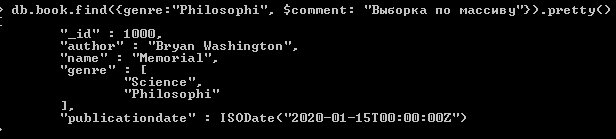
db.book.find({$comment: "Выборка всех данных"}).pretty()



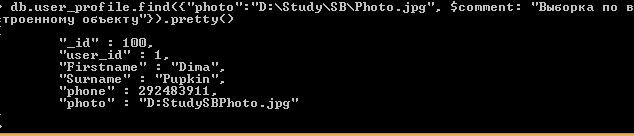
db.book.find({\_id: 1002, $comment: "Выборка по конкретному полю"}).pretty()



db.book.find({genre:"Philosophi", $comment: "Выборка по массиву"}).pretty()

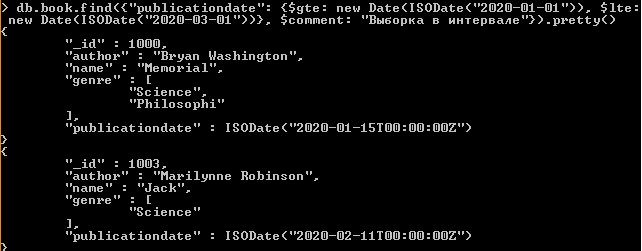


db.user\_profile.find({"photo":"D:\Study\SB\Photo.jpg", $comment: "Выборка по встроенному объекту"}).pretty()

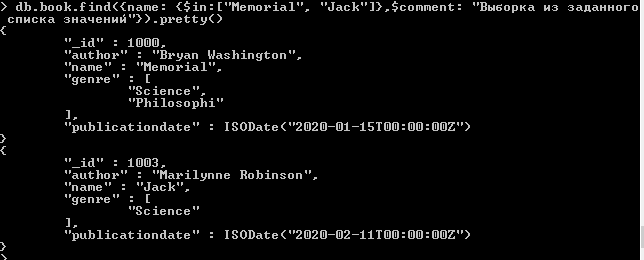


**2. Выборка в интервале, выборка из заданного списка значений.**

db.book.find({"publicationdate": {$gte: new Date(ISODate("2020-01-01")), $lte: new Date(ISODate("2020-03-01"))}, $comment: "Выборка в интервале"}).pretty()

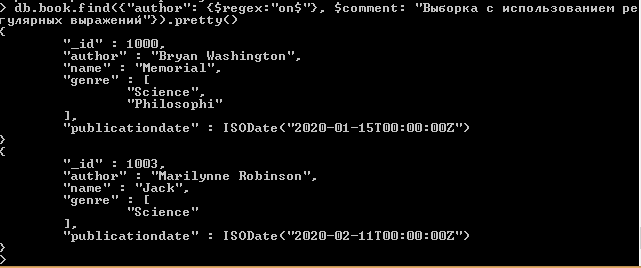


db.book.find({name: {$in:["Memorial", "Jack"]},$comment: "Выборка из заданного списка значений"}).pretty()



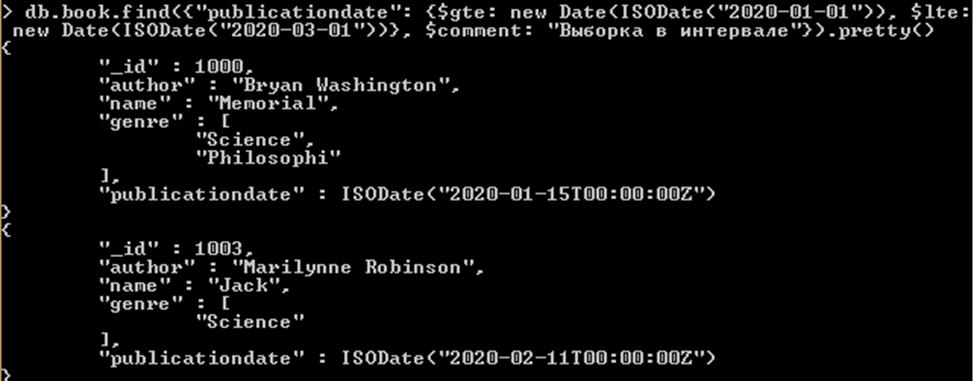
**3. Выборка с использованием регулярных выражений.**

db.book.find({"author": {$regex:"on$"}, $comment: "Выборка с использованием регулярных выражений"}).pretty()



**4. Выборка данных за конкретный период времени, на сегодняшний день, за последний месяц.**

db.book.find({"publicationdate": {$gte: new Date(ISODate("2020-01-01")), $lte: new Date(ISODate("2020-03-01"))}, $comment: "Выборка в интервале"}).pretty()



db.book.find({"publicationdate": {$gte: new Date(new Date().getFullYear(), new Date().getMonth(), new Date().getDate(), 0, 0, 0), $lte: new Date()},$comment: "Выборка на сегодняшний день"}).pretty()



db.book.find({"publicationdate ": {$gte: new Date(new Date().getFullYear(),new Date().getMonth(), 1, 0, 0, 0), $lt: new Date(new Date().getFullYear(),new Date().getMonth()+1, 1, 0, 0, 0)}, $comment: "Выборка за последний месяц"}).pretty()



**55.Выборка данных из нескольких таблиц.**

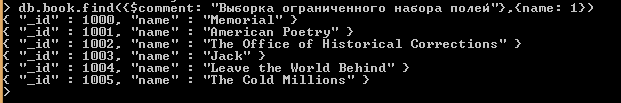
db.read\_card.find().map(({user\_profile\_id }) => {{Surname} = db.user\_profile.findOne({\_id: user\_profile\_id }); return { Surname };})

**6. Выборка из нескольких таблиц с фильтрацией.**

db.read\_card.find({user\_profile\_id: 100, $comment: "Выборка из нескольких таблиц с фильтрацией"}).map(({user\_profile\_id}) => {const {Surname} = db.user\_profile.findOne({\_id: user\_profile\_id}); return {Surname};})

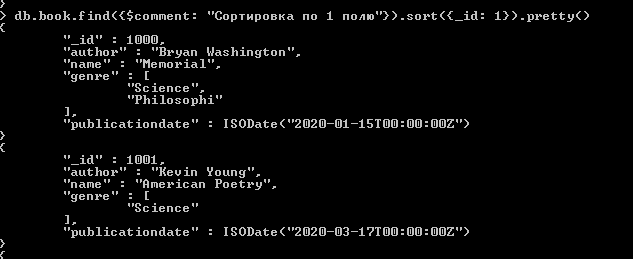
**7. Выборка ограниченного набора полей.**

db.book.find({$comment: "Выборка ограниченного набора полей"},{name: 1})

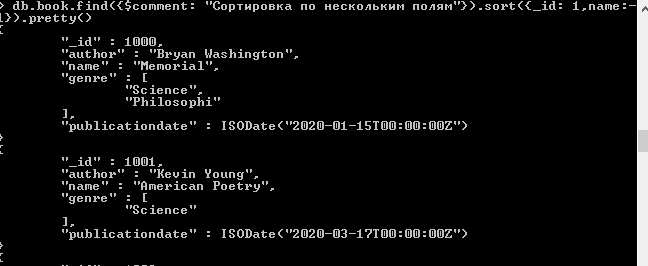


**8. Сортировка по 1 полю, по нескольким полям.**

db.book.find({$comment: "Сортировка по 1 полю"}).sort({\_id: 1}).pretty()

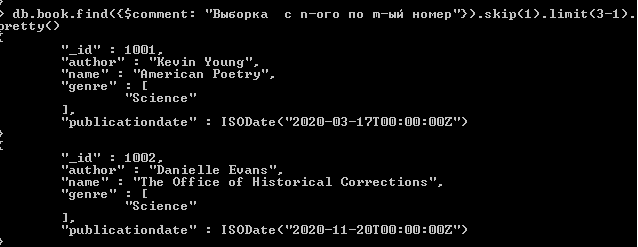


db.book.find({$comment: "Сортировка по нескольким полям"}).sort({\_id: 1,name:-1}).pretty()



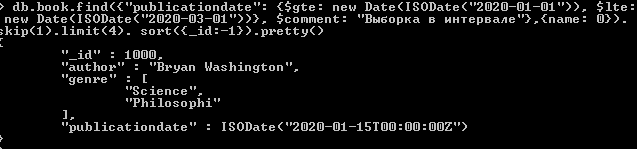
**9.Вывод данных с n-ого по m-ый номер**.

db.book.find({$comment: "Выборка с n-ого по m-ый номер"}).skip(1).limit(3-1).pretty()



**10.Написать запрос, который объединяет 1-9 пункты.**

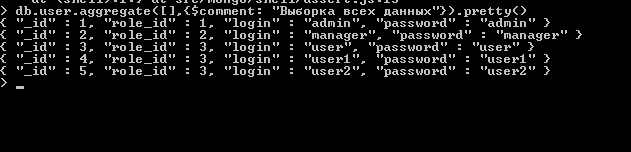
db.book.find({"publicationdate": {$gte: new Date(ISODate("2020-01-01")), $lte: new Date(ISODate("2020-03-01"))}, $comment: "Выборка в интервале"},{name: 0}). skip(1).limit(4). sort({\_id:-1}).pretty()



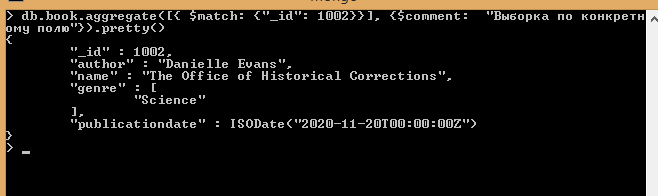
**С помощью Аggregation framework**

**1. Выборка всех данных, фильтрация по конкретному полю, массиву, встроенному объекту.**

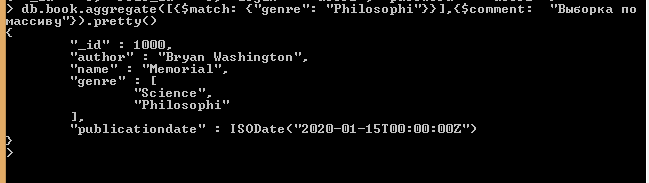
db.user.aggregate([],{$comment: "Выборка всех данных"}).pretty()



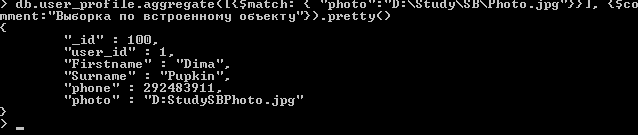
db.book.aggregate([{ $match: {"\_id": 1002}}], {$comment: "Выборка по конкретному полю"}).pretty()



db.book.aggregate([{$match: {"genre": "Philosophi"}}],{$comment: "Выборка по массиву"}).pretty()

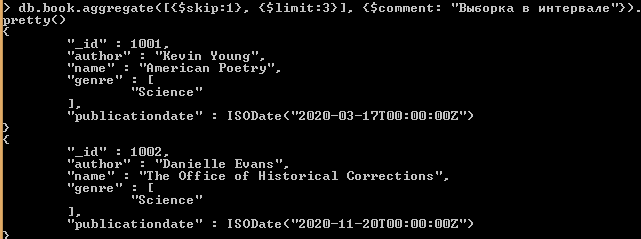


db.user\_profile.aggregate([{$match: { "photo":"D:\Study\SB\Photo.jpg"}}], {$comment:"Выборка по встроенному объекту"}).pretty()

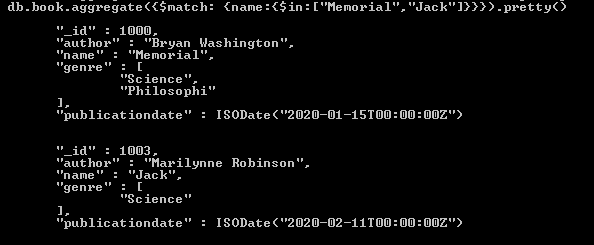


**2. Выборка в интервале, выборка из заданного списка значений.**

db.book.aggregate([{$skip:1}, {$limit:3}], {$comment: "Выборка в интервале"}).pretty()

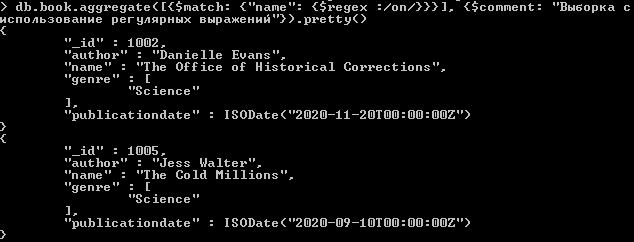


db.book.aggregate({$match:{name:{$in:["Memorial","Jack"]}}},{$comment: "Выборка в интервале"}).pretty()



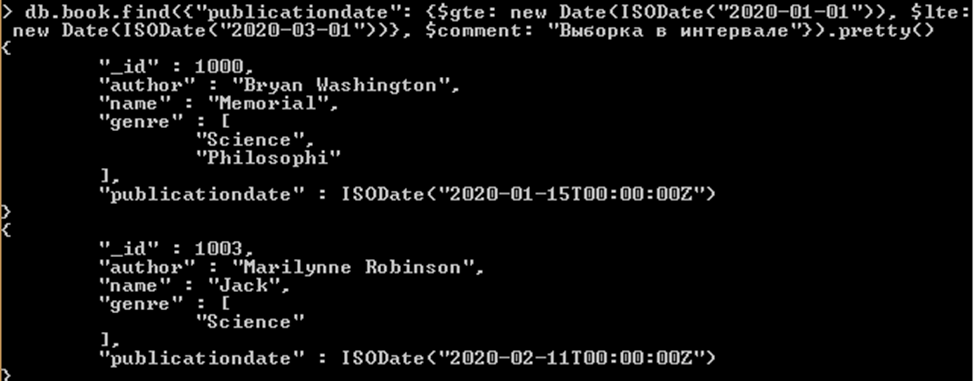
**3. Выборка с использованием регулярных выражений.**

db.book.aggregate([{$match: {"name": {$regex :/on/}}}], {$comment: "Выборка с использование регулярных выражений"}).pretty()



**4. Выборка данных за конкретный период времени, на сегодняшний день, за последний месяц.**

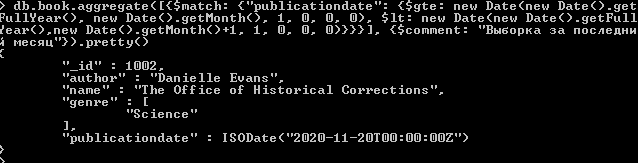
db.book.aggregate([{$match: {" publicationdate": {$gte: new Date(ISODate("2020-01-01")), $lte: new Date(ISODate("2020-03-01"))}}}], {$comment: "Выборка за конкретный период времени"}).pretty()



db.book.aggregate([{$match: {"publicationdate": {$gte: new Date(new Date().getFullYear(), new Date().getMonth(), new Date().getDate(), 0, 0, 0), $lte: new Date()}}}], {$comment: "Выборка за сегодняшний день"}).pretty()



db.book.aggregate([{$match: {"publicationdate": {$gte: new Date(new Date().getFullYear(), new Date().getMonth(), 1, 0, 0, 0), $lt: new Date(new Date().getFullYear(),new Date().getMonth()+1, 1, 0, 0, 0)}}}], {$comment: "Выборка за последний месяц"}).pretty()



**5.Выборка данных из нескольких таблиц**

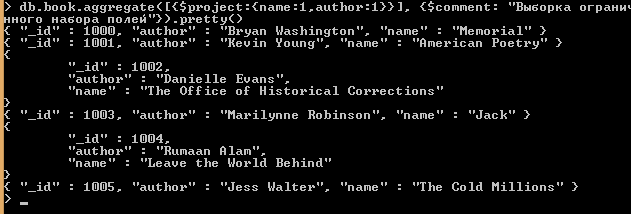
db.user.aggregate([{$lookup:{from : “user\_profile”,localField : “\_id”,foreignField : “user\_id”,as : “games”} } ] ).pretty()

**6.Выборка из нескольких таблиц с фильтрацией.**

db.user.aggregate([{$lookup: {from: "user\_profile", localField: "\_id", foreignField: "user\_id", as: "games"}}, {$match: {"\_id": 3}}], {$comment: "Выборка данных из нескольких таблиц"}).pretty()

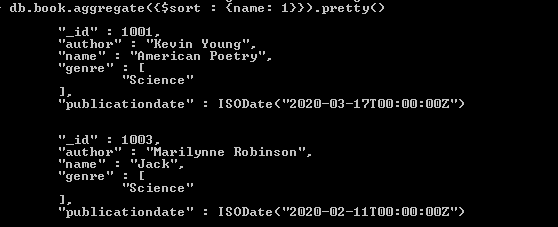
**7.Выборка ограниченного набора полей.**

db.book.aggregate([{$project:{name:1,author:1}}], {$comment: "Выборка ограниченного набора полей"}).pretty()

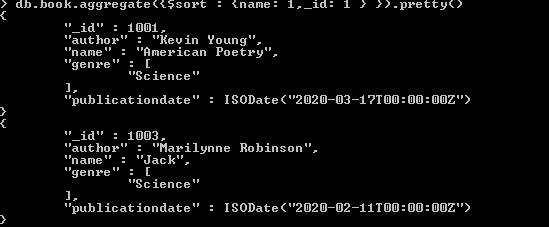


**8. Сортировка по 1 полю, по нескольким полям.**

db.book.aggregate({$sort : {name: 1}}).pretty()



db.book.aggregate({$sort : {name: 1,\_id: 1 } }).pretty()



**9.Вывод данных с n-ого по m-ый номер.**

db.user\_profile.aggregate([{$skip:(1)}, {$limit: (3)}], {$comment: "Выборка от n до m"}).pretty()



**10.Написать запрос, который объединяет 1-9 пункты.**

db.book.aggregate([{$match: {"publicationdate": {$gte: ISODate("2020-01-01"), $lte: ISODate("2020-11-01")}}}, {$skip:1}, {$limit:5}, {$project:{name: 1}}, {$sort: {"\_id": 1}}]). pretty()

