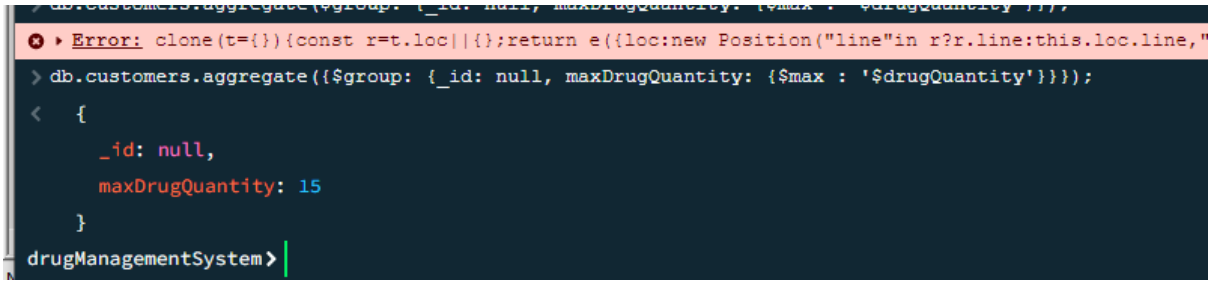
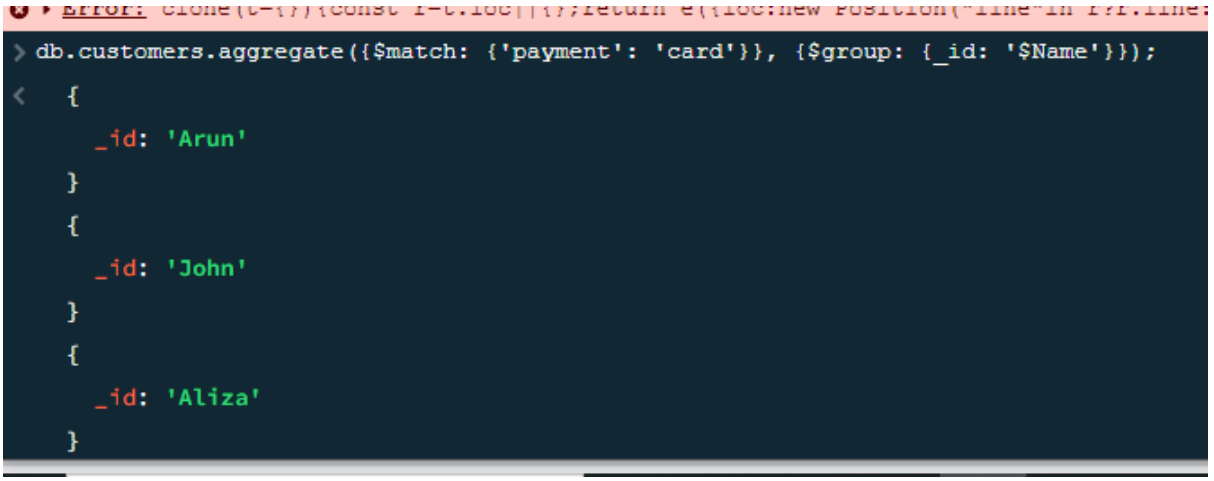
1. db.customers.aggregate({$group: {_id: null, maxDrugQuantity: {$max : '$drugQuantity'}}});

Find the maximum quantity of drugs:



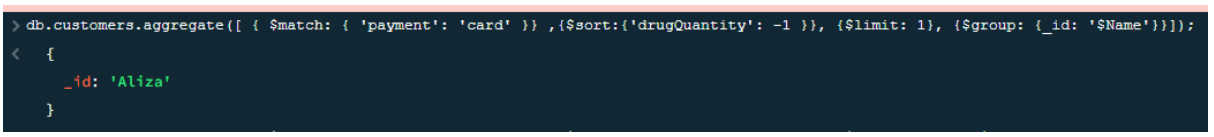
2. db.customers.aggregate({$match: {'payment': 'card'}}, {$group: {_id: '$Name'}});

Find all names which are having the payment method card.



3. db.customers.aggregate([{ $match: { 'payment': 'card' } } ,{$sort:{'drugQuantity': -1 }}, {$limit: 1}, {$group: {_id: '$Name'}}]);

Display the name which is having the payment method is card and having the maximum drugQuantity.

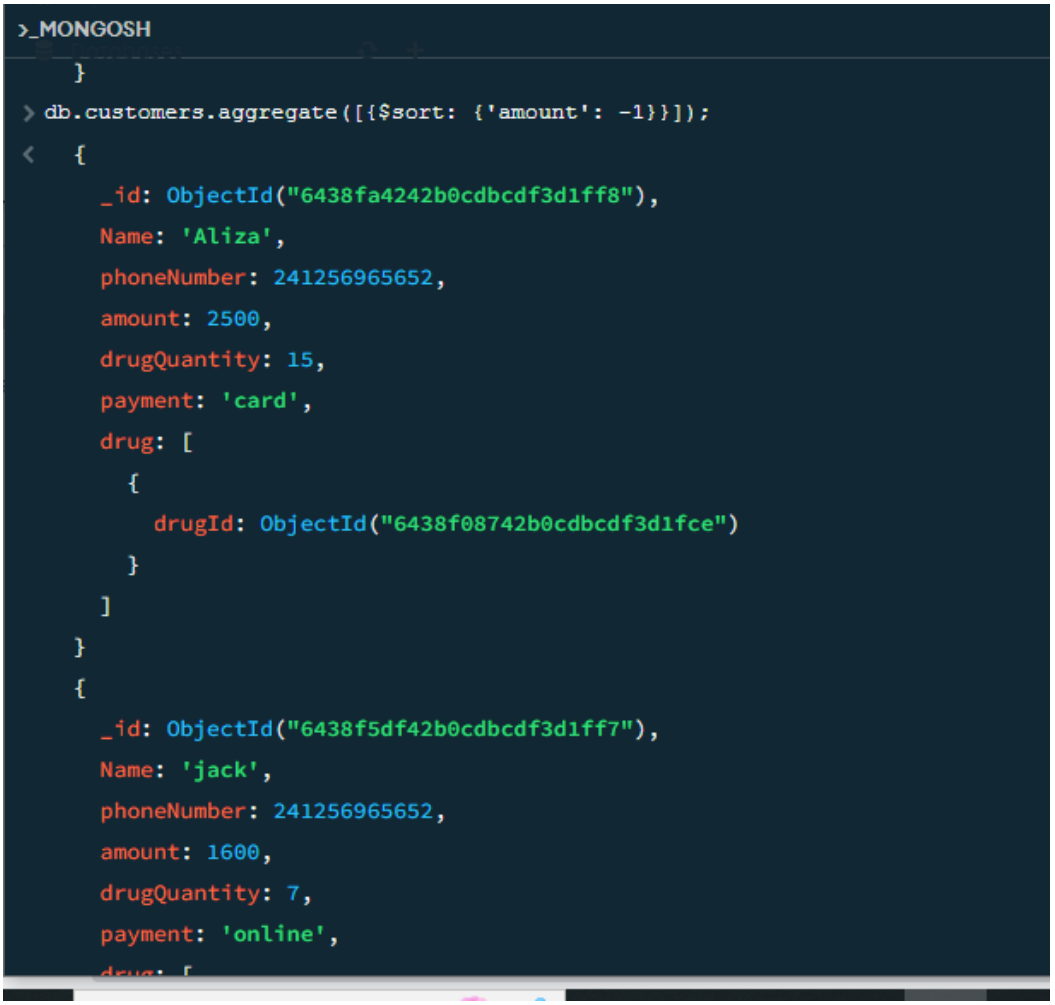


4. db.customers.aggregate([{ $match: { 'payment': 'card' } } ,{$sort:{'Name': -1 }}, {$group: { _id: '$Name'}}]);

   Here, names are sorted in descending order which are having payment method card.

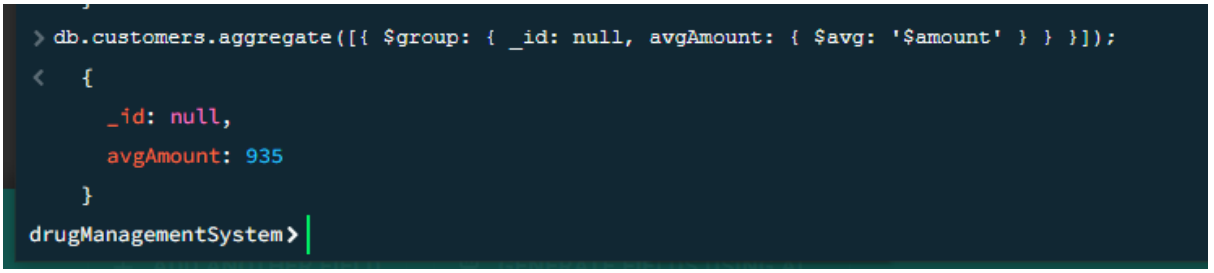
5. db.customers.aggregate([{$sort: {'amount': -1}}]);

   Sort all the data in descending order according to amount.



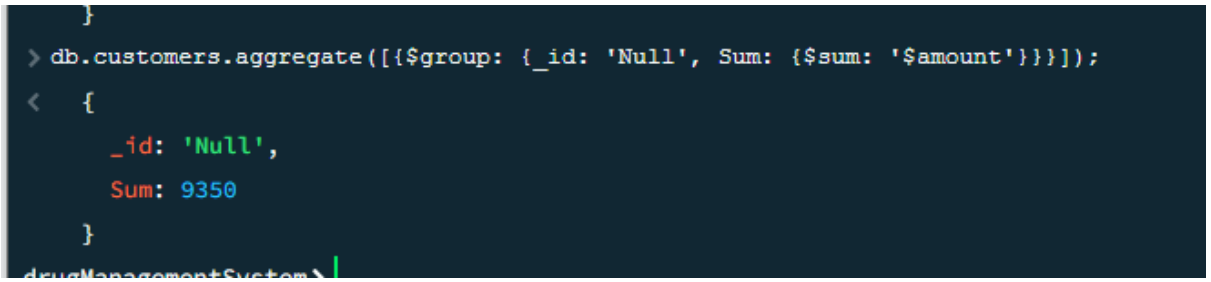
6. db.customers.aggregate([{ $group: { _id: null, avgAmount: { $avg: '$amount' } } }]);

   This is calculating the average of the amount.



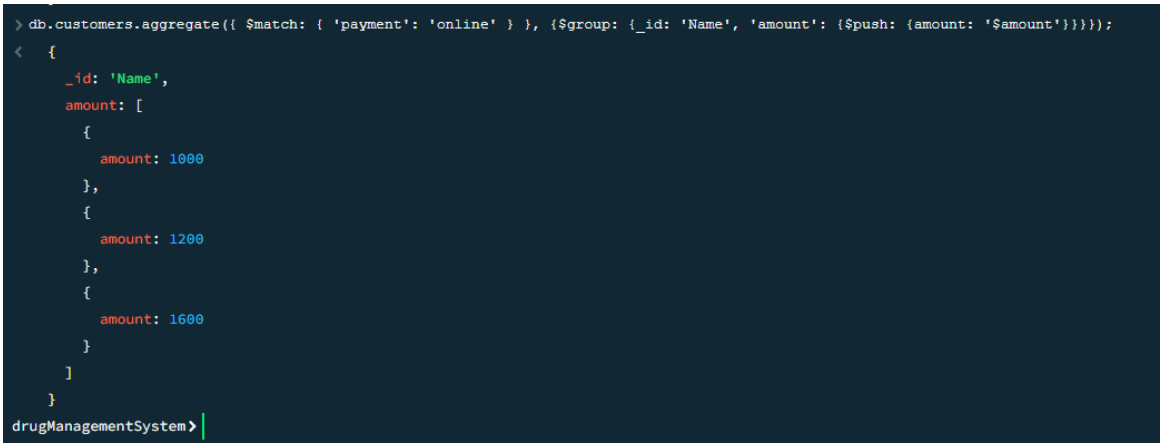
7. db.customers.aggregate([{$group: {_id: 'Null', Sum: {$sum: '$amount'}}}]);

   Here, the total sum of the amount.



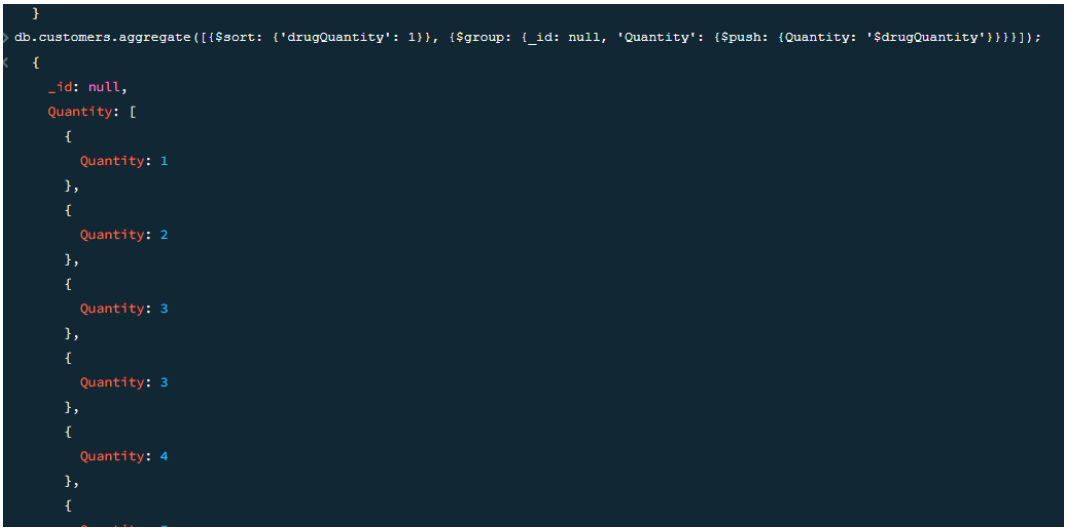
8. db.customers.aggregate({ $match: { 'payment': 'online' } }, {$group: {_id: 'Name', 'amount': {$push: {amount: '$amount'}}}});

Here, making the array of amounts which are having the payment method online.



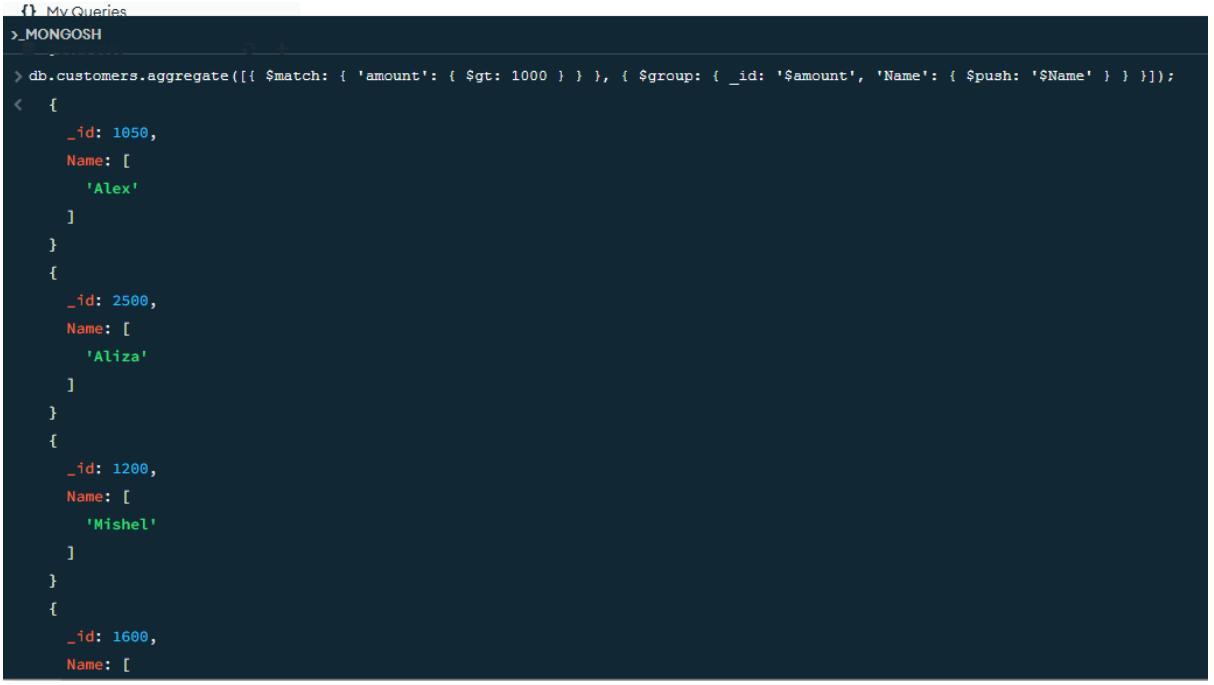
9. db.customers.aggregate([{$sort: {'drugQuantity': 1}}, {$group: {_id: null, 'Quantity': {$push: {Quantity: '$drugQuantity'}}}}]);

Here, we are sorting the drug quantity in increasing order and pushing that in array.



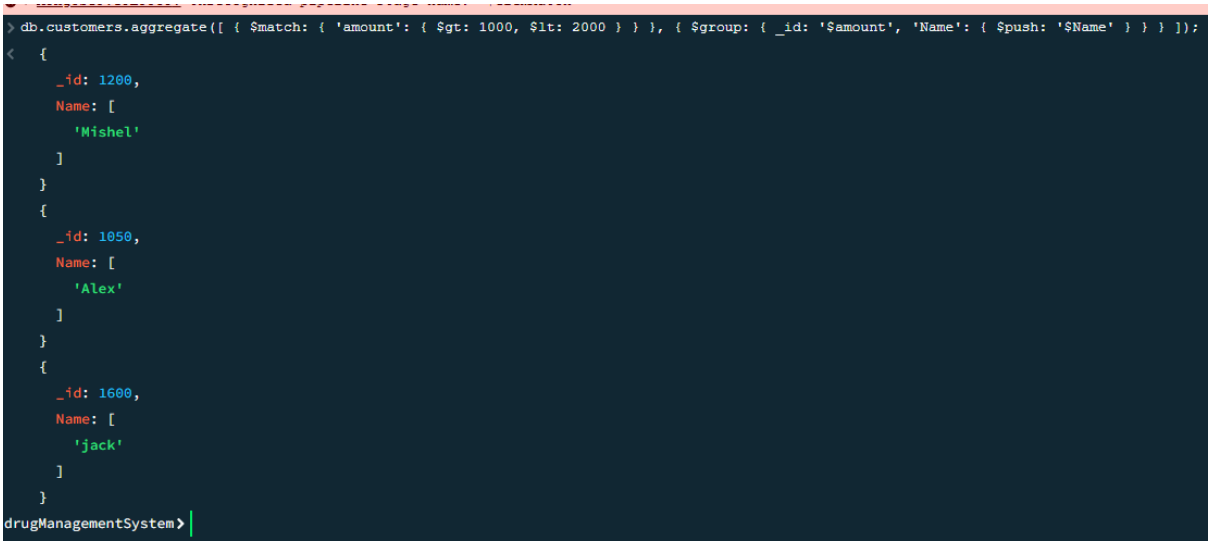
10. db.customers.aggregate([{ $match: { 'amount': { $gt: 1000 } } }, { $group: { _id: '$amount', 'Name': { $push: '$Name' } } }]);

    Here, we find the amount greater than 1000 and then push all the names in the array.



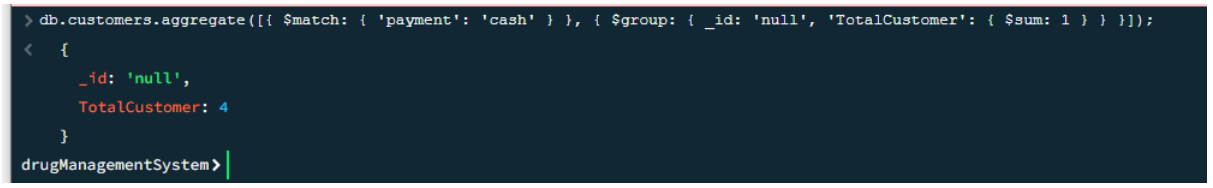
11. db.customers.aggregate([{ $match: { 'amount': { $gt: 1000, $lt: 2000 } } }, { $group: { _id: '$amount', 'Name': { $push: '$Name' } } }]);

    Here, we are filtering the data by the range that how much data is present between the amount of 1000 to 2000, and we push the data into array



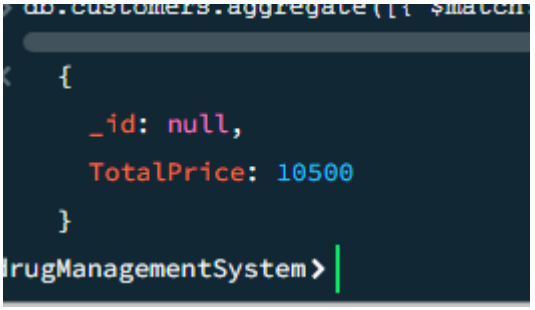
12. db.customers.aggregate([{ $match: { 'payment': 'cash' } }, { $group: { _id: 'null', 'TotalCustomer': { $sum: 1 } } }]);

    Here, we find the total customers that are having the payment method cash.



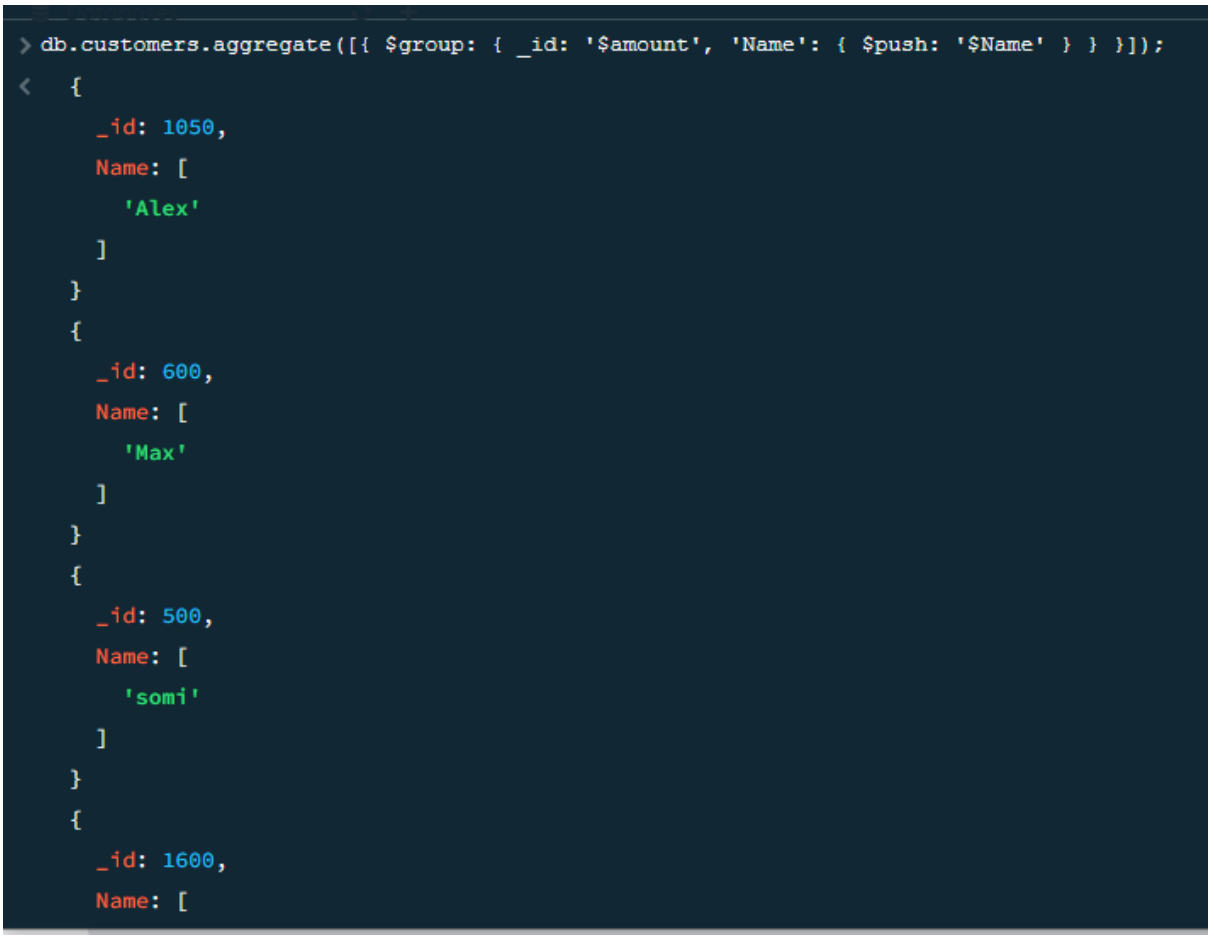
13. db.customers.aggregate([{ $match: { 'Name': 'Alex' } }, { $addFields: { 'Price': { $multiply: ['$amount', '$drugQuantity'] } } },{ $group: { _id: null, 'TotalPrice': { $sum: '$Price' } } }]);

    Here, we are multiply the amount and quantity of Alex(Customer)



14. db.customers.aggregate([{ $group: { _id: '$amount', 'Name': { $push: '$Name' } } }]);

    Here, we are pushing all the customer names in array that are present in collection.



15. db.customers.aggregate({$match:{"Name": {$in:['Alex', 'John']}}});

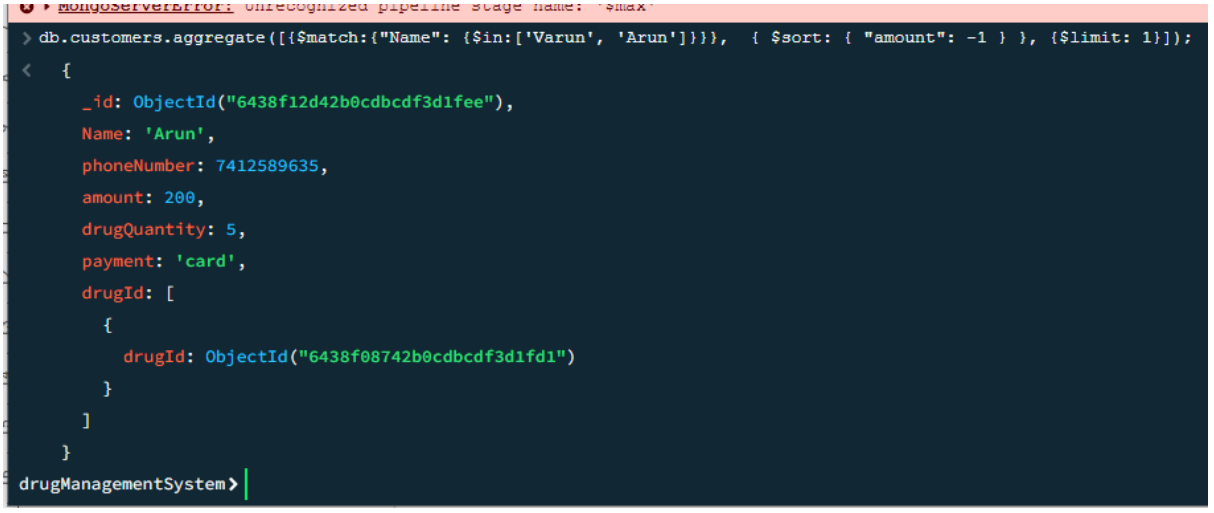
    Here, we are getting the object of customer names Alex and John.

16. db.customers.aggregate([{$match:{"Name": {$in:['Alex', 'John']}}}, {$match:{'payment': 'cash'}}]);

    Here, we are finding between Alex and John which one is having the payment method cash.

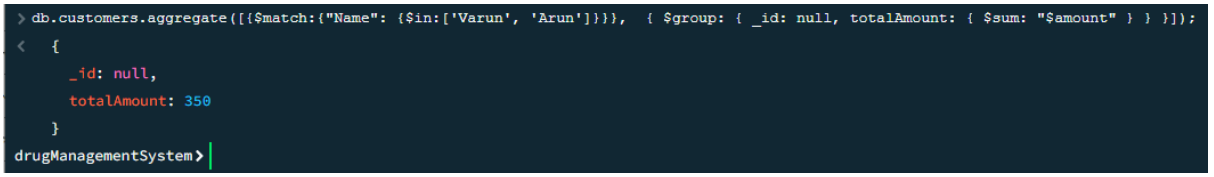
17. db.customers.aggregate([{$match:{"Name": {$in:['Varun', 'Arun']}}}, { $sort: { "amount": -1 } }, {$limit: 1}]);

    Here, we are finding between Varun and Arun which one is having the more amount.



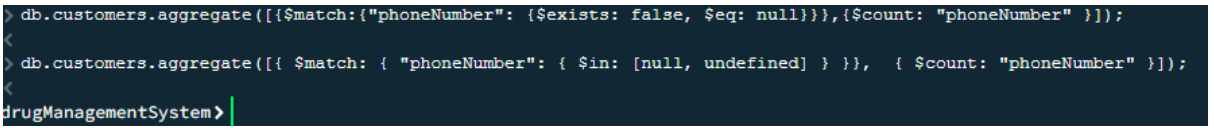
18. db.customers.aggregate([{$match:{"Name": {$in:['Varun', 'Arun']}}}, { $group: { _id: null, totalAmount: { $sum: "$amount" } } }]);

    Here, we are finding the sum of the amounts of Varun and Arun.



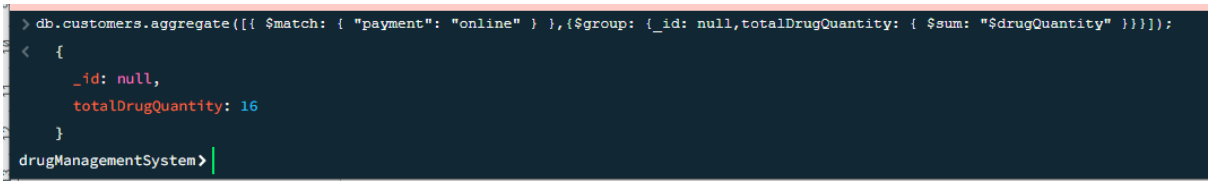
19. db.customers.aggregate([{$match:{"phoneNumber": {$exists: false, $eq: null}}},{$count: "phoneNumber" }]);

    Here, we are checking that whether is any object having empty phone Number or the PhoneNumber is exist or not.



20. db.customers.aggregate([{ $match: { "payment": "online" } },{$group: {_id: null,totalDrugQuantity: { $sum: "$drugQuantity" }}}]);

    Here, filtering the total drug Quantity of the payment method online.



21. db.customers.aggregate([{$match: { "_id": ObjectId("6438f2bf42b0cdbcdf3d1ff0") }},{ $lookup: {from: "drugs",localField: "drugId.drugId",foreignField: "_id", as: "drugDetails"}}]);

    Here, we are finding the details of the drug through the drugId which is present in the customer collection with the lookup method.