MongoDB Assignment – done by Yeo Theng Hee

1 & 2. We add a series of collections into users, by executing these **insert** queries; result is shown below.

db.users.insert({id: 100, userName: "John", mail: "John@gmail.com", mobile: 96969696, transaction: [{ItemId: "a100", price: 200},{ItemId: "a110", price: 200}], payment: {Type: "Credit-Card", Total: 400, Success: true}, Remarks: "1st Complete Record, payment successful"})

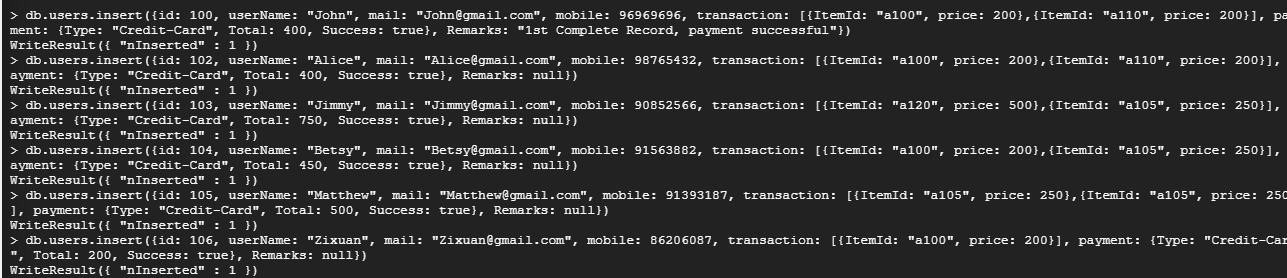
db.users.insert({id: 102, userName: "Alice", mail: "Alice@gmail.com", mobile: 98765432, transaction: [{ItemId: "a100", price: 200},{ItemId: "a110", price: 200}], payment: {Type: "Credit-Card", Total: 400, Success: true}, Remarks: null})

db.users.insert({id: 103, userName: "Jimmy", mail: "Jimmy@gmail.com", mobile: 90852566, transaction: [{ItemId: "a120", price: 500},{ItemId: "a105", price: 250}], payment: {Type: "Credit-Card", Total: 750, Success: true}, Remarks: null})

db.users.insert({id: 104, userName: "Betsy", mail: "Betsy@gmail.com", mobile: 91563882, transaction: [{ItemId: "a100", price: 200},{ItemId: "a105", price: 250}], payment: {Type: "Credit-Card", Total: 450, Success: true}, Remarks: null})

db.users.insert({id: 105, userName: "Matthew", mail: "Matthew@gmail.com", mobile: 91393187, transaction: [{ItemId: "a105", price: 250},{ItemId: "a105", price: 250}], payment: {Type: "Credit-Card", Total: 500, Success: true}, Remarks: null})

db.users.insert({id: 106, userName: "Zixuan", mail: "Zixuan@gmail.com", mobile: 86206087, transaction: [{ItemId: "a100", price: 200}], payment: {Type: "Credit-Card", Total: 200, Success: true}, Remarks: null})



3. Find any record where the total payment amount is greater than 700.

db.users.find({"payment.Total": {$gte: 700}})



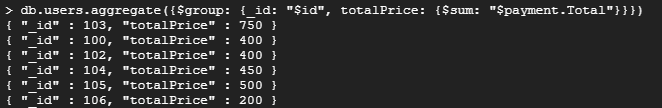
4. Find the total payment of all the records.

db.users.aggregate({$group : {\_id: null, totalPayment: {$sum: "$payment.Total"}}})



5. Find the total transaction price per record.

db.users.aggregate({$group: {\_id: "$id", totalPrice: {$sum: "$payment.Total"}}})



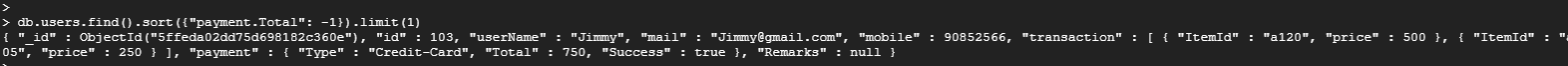
6. Find the max Transaction price.

db.users.aggregate({$group : {\_id: null, maxPrice: {$max: "$transaction.price"}}})



7. Find the record having the highest payment and the record having the lowest payment.

db.users.find().sort({"payment.Total": -1}).limit(1)

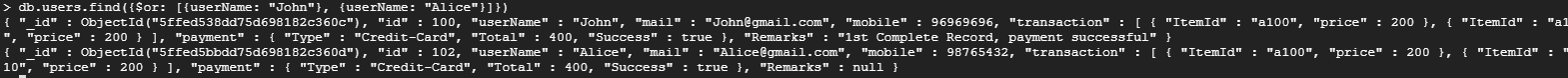


db.users.find().sort({"payment.Total": 1}).limit(1)



8. Find the records that have the username either “John” or “Alice”.

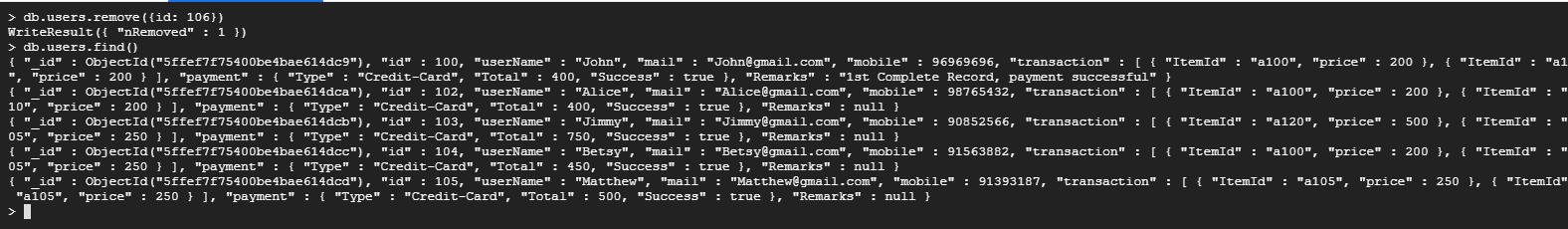
db.users.find({$or: [{name: "John"}, {name: "Alice"}]})



9. Delete a record by using Id.

After we remove the last record with the following query, we can only see the remaining (five) left.

db.users.remove({id: 106})



10. Arrange the name of the users in ascending order.

db.users.find({},{"userName": 1, "\_id": 0}).sort({userName: 1})

