**MongoDB Homework**

cd mongodb-getting-started

./run-mongo.sh

**Solution for Q1 and Q2 of Assignment**

>db.createCollection("customers"))

>db.customers.insertMany([ { Id: 100, userName : "John", mail: "John@gmail.com", mobile: 123456789, Transaction: [ { ItemId:"a100", price: 200 }, {"ItemId":"a110", "price": 200} ], Payment: { Type: "Credit-Card", Total: 400, Success: true }, Remarks: "1st Complete Record, payment successful" }, { Id: 102, userName : "Alice", mail: "Alice@gmail.com" , mobile: 987654321, Transaction: [ { "ItemId":"a100", "price": 200 }, { "ItemId":"a110", price: 200 } ], Payment: { Type: "Credit-Card", Total: 400, Success: true }, Remarks: "null" } ])

>db.customers.insertMany([ { Id: 103, userName : "Lim", mail: "LimKL@gmail.com", mobile: 98761234, Transaction: [ { ItemId:"b100", price: 701 }, {"ItemId":"b110", "price": 719} ], Payment: { Type: "Credit-Card", Total: 1420, Success: true }, Remarks: "1st Complete Record, payment successful" }, { Id: 104, userName : "Tan", mail: "TanST@gmail.com" , mobile: 87651234, Transaction: [ { "ItemId":"b100", "price": 700 }, { "ItemId":"b110", price: 800 } ], Payment: { Type: "Credit-Card", Total: 1500, Success: true }, Remarks: "null" }, { Id: 105, userName : "Mohd Ali", mail: "MAli@gmail.com" , mobile: 91123456, Transaction: [ { "ItemId":"b100", "price": 700 }, { "ItemId":"a110", price: 730 } ], Payment: { Type: "Credit-Card", Total: 1430, Success: true }, Remarks: "null" }, { Id: 106, userName : "Krish Mohan", mail: "KMohan@gmail.com" , mobile: 82234567, Transaction: [ { "ItemId":"c100", "price": 720 }, { "ItemId":"c110", price: 730 } ], Payment: { Type: "Credit-Card", Total: 1450, Success: true }, Remarks: "null" } ])

**Solution for Q3 of Assignment: payment amounts exceeding $700**

>db.customers.find({"Payment.Total":{$gt:700}})

>db.customers.aggregate( {$group:{\_id:"$Payment.Total", cnt:{$sum:1}}})

Not the right answer but learn from this

**Solution for Q4 of Assignment: grand total of all payments**

>db.customers.aggregate({$group:{\_id:0, total:{$sum:"$Payment.Total"}}})

{ "\_id" : 0, "total" : 6600 }

Note curly brackets { } around the $group will be needed

>db.customers.aggregate({$group:{\_id:"$Id", total:{$sum:"$Transaction.price"}}})

Not the right answer but learn from this

**Solution for Q5 of Assignment: total transaction price**

>db.customers.aggregate({$project:{ total:{$sum:"$Transaction.price"}}})

**Solution for Q6 of Assignment: max transaction price**

>db.customers.aggregate({$project:{maxpr:{$max:"$Transaction.price"}}})

**Solution for Q7 of Assignment: max and min transaction prices**

db.customers.aggregate({$group:{\_id:0, maxpay:{$max:"$Payment.Total"}}})

{ "\_id" : 0, "maxpay" : 1500 }

db.customers.aggregate({$group:{\_id:0, minpay:{$min:"$Payment.Total"}}})

{ "\_id" : 0, "minpay" : 400 }

**Solution for Q8 of Assignment: find records with either John or Alice**

>db.customers.find({$or:[{userName:"John"},{userName:"Alice"}]})

{ "\_id" : ObjectId("5f6fef4656630ffc6e93cb8b"), "Id" : 100, "userName" : "John", "mail" : "John@gmail.com", "mobile" : 123456789, "Transaction" : [ { "ItemId" : "a100", "price" : 200 }, { "ItemId" : "a110", "price" : 200 } ], "Payment" : { "Type" : "Credit-Card", "Total" : 400, "Success" : true }, "Remarks" : "1st Complete Record, payment successful" }

{ "\_id" : ObjectId("5f6fef4656630ffc6e93cb8c"), "Id" : 102, "userName" : "Alice", "mail" : "Alice@gmail.com", "mobile" : 987654321, "Transaction" : [ { "ItemId" : "a100", "price" : 200 }, { "ItemId" : "a110", "price" : 200 } ], "Payment" : { "Type" : "Credit-Card", "Total" : 400, "Success" : true }, "Remarks" : "null" }

**Solution for Q9 of Assignment: Delete a record**

>db.customers.remove({userName:"Krish Mohan"})

WriteResult({ "nRemoved" : 1 })

To reinstate:

 >db.customers.insert({ Id: 106, userName : "Krish Mohan", mail: "KMohan@gmail.com" , mobile: 82234567, Transaction: [ { "ItemId":"c100", "price": 720 }, { "ItemId":"c110", price: 730 } ], Payment: { Type: "Credit-Card", Total: 1450, Success: true }, Remarks: "null" })WriteResult({ "nInserted" : 1 })

**Solution for Q10 of Assignment: Arrange name of users in ascending order**

>db.customers.find().sort({"userName":1})

{ "\_id" : ObjectId("5f6fef4656630ffc6e93cb8c"), "Id" : 102, "userName" : "Alice", "mail" : "Alice@gmail.com", "mobile" : 987654321, "Transaction" : [ { "ItemId" : "a100", "price" : 200 }, { "ItemId" : "a110", "price" : 200 } ], "Payment" : { "Type" : "Credit-Card", "Total" : 400, "Success" : true }, "Remarks" : "null" }

{ "\_id" : ObjectId("5f6fef4656630ffc6e93cb8b"), "Id" : 100, "userName" : "John", "mail" : "John@gmail.com", "mobile" : 123456789, "Transaction" : [ { "ItemId" : "a100", "price" : 200 }, { "ItemId" : "a110", "price" : 200 } ], "Payment" : { "Type" : "Credit-Card", "Total" : 400, "Success" : true }, "Remarks" : "1st Complete Record, payment successful" }

{ "\_id" : ObjectId("5f7002c356630ffc6e93cb91"), "Id" : 106, "userName" : "Krish Mohan", "mail" : "KMohan@gmail.com", "mobile" : 82234567, "Transaction" : [ { "ItemId" : "c100", "price" : 720 }, { "ItemId" : "c110", "price" : 730 } ], "Payment" : { "Type" : "Credit-Card", "Total" : 1450, "Success" : true }, "Remarks" : "null" }

{ "\_id" : ObjectId("5f6fef5c56630ffc6e93cb8d"), "Id" : 103, "userName" : "Lim", "mail" : "LimKL@gmail.com", "mobile" : 98761234, "Transaction" : [ { "ItemId" : "b100", "price" : 701 }, { "ItemId" : "b110", "price" : 719 } ], "Payment" : { "Type" : "Credit-Card", "Total" : 1420, "Success" : true }, "Remarks" : "1st Complete Record, payment successful" }

{ "\_id" : ObjectId("5f6fef5c56630ffc6e93cb8f"), "Id" : 105, "userName" : "Mohd Ali", "mail" : "MAli@gmail.com", "mobile" : 91123456, "Transaction" : [ { "ItemId" : "b100", "price" : 700 }, { "ItemId" : "a110", "price" : 730 } ], "Payment" : { "Type" : "Credit-Card", "Total" : 1430, "Success" : true }, "Remarks" : "null" }

{ "\_id" : ObjectId("5f6fef5c56630ffc6e93cb8e"), "Id" : 104, "userName" : "Tan", "mail" : "TanST@gmail.com", "mobile" : 87651234, "Transaction" : [ { "ItemId" : "b100", "price" : 700 }, { "ItemId" : "b110", "price" : 800 } ], "Payment" : { "Type" : "Credit-Card", "Total" : 1500, "Success" : true }, "Remarks" : "null" }

For descending order, use -1 as in:

>db.customers.find().sort({"userName":-1})