ASSIGNMENT-1

1)

db.TEACHER.insertOne({"ID":10101,"Name":"Srinivasan","Dept\_name":"Comp. Sci.","Salary":65000,"Teaches":[{"Course\_id":"CS-101","Title":"Intro. to Computer Science","Department\_name":"Comp. Sci.","Credits":4,"Prereq\_id":"","Sec\_id":1,"Semester":"Fall","Year":2009,"Teaches\_Dept":{"Building":"Taylor","Budget":100000}},{"Course\_id":"CS-315","Title":"Robotics","Department\_name":"Comp. Sci.","Credits":3,"Prereq\_id":"CS-101","Sec\_id":1,"Semester":"Spring","Year":2010,"Teaches\_Dept":{"Building":"Taylor","Budget":100000}},{"Course\_id":"CS-347","Title":"Database System Concepts","Department\_name":"Comp. Sci.","Credits":3,"Prereq\_id":"CS-101","Sec\_id":1,"Semester":"Fall","Year":2009,"Teaches\_Dept":{"Building":"Taylor","Budget":100000}}]})

db.TEACHER.insertOne({"ID":12121,"Name":"Wu","Dept\_name":"Finance","Salary":90000,"Teaches":[{"Course\_id":"FIN-201","Title":"Investment Banking","Department\_name":"Finance","Credits":3,"Prereq\_id":"","Sec\_id":1,"Semester":"Spring","Year":2010,"Teaches\_Dept":{"Building":"Painter","Budget":120000}}]})

db.TEACHER.insertOne({"ID":15151,"Name":"Mozart","Dept\_name":"Music","Salary":40000,"Teaches":[{"Course\_id":"MU-199","Title":"Music Video Production","Department\_name":"Music","Credits":3,"Prereq\_id":"","Sec\_id":1,"Semester":"Spring","Year":2010,"Teaches\_Dept":{"Building":"Packard","Budget":80000}}]})

db.TEACHER.insertOne({"ID":22222,"Name":"Einstein","Dept\_name":"Physics","Salary":95000,"Teaches":[{"Course\_id":"PHY-101","Title":"Physical Principles","Department\_name":"Physics","Credits":4,"Prereq\_id":"","Sec\_id":1,"Semester":"Fall","Year":2009,"Teaches\_Dept":{"Building":"Watson","Budget":70000}}]})

db.TEACHER.insertOne({"ID":32343,"Name":"El Said","Dept\_name":"History","Salary":60000,"Teaches":[{"Course\_id":"HIS-351","Title":"World History","Department\_name":"History","Credits":3,"Prereq\_id":"","Sec\_id":1,"Semester":"Spring","Year":2010,"Teaches\_Dept":{"Building":"Painter","Budget":50000}}]})

db.TEACHER.insertOne({"ID":33456,"Name":"Gold","Dept\_name":"Physics","Salary":87000,"Teaches":[{"Course\_id":"","Title":"","Department\_name":"","Credits":null,"Prereq\_id":"","Sec\_id":null,"Semester":"","Year":null,"Teaches\_Dept":{"Building":"","Budget":null}}]})

db.TEACHER.insertOne({"ID":45565,"Name":"Katz","Dept\_name":"Comp. Sci.","Salary":75000,"Teaches":[{"Course\_id":"CS-101","Title":"Intro. to Computer Science","Department\_name":"Comp. Sci.","Credits":4,"Prereq\_id":"","Sec\_id":1,"Semester":"Spring","Year":2010,"Teaches\_Dept":{"Building":"Taylor","Budget":100000}},{"Course\_id":"CS-319","Title":"Image Processing","Department\_name":"Comp. Sci.","Credits":3,"Prereq\_id":"CS-101","Sec\_id":1,"Semester":"Spring","Year":2010,"Teaches\_Dept":{"Building":"Taylor","Budget":100000}}]})

db.TEACHER.insertOne({"ID":58583,"Name":"Califieri","Dept\_name":"History","Salary":62000,"Teaches":[{"Course\_id":"","Title":"","Department\_name":"","Credits":null,"Prereq\_id":"","Sec\_id":null,"Semester":"","Year":null,"Teaches\_Dept":{"Building":"","Budget":null}}]})

db.TEACHER.insertOne({"ID":76543,"Name":"Singh","Dept\_name":"Finance","Salary":80000,"Teaches":[{"Course\_id":"","Title":"","Department\_name":"","Credits":null,"Prereq\_id":"","Sec\_id":null,"Semester":"","Year":null,"Teaches\_Dept":{"Building":"","Budget":null}}]})

db.TEACHER.insertOne({"ID":76766,"Name":"Crick","Dept\_name":"Biology","Salary":72000,"Teaches":[{"Course\_id":"BIO-101","Title":"Intro. to Biology","Department\_name":"Biology","Credits":4,"Prereq\_id":"","Sec\_id":1,"Semester":"Summer","Year":2009,"Teaches\_Dept":{"Building":"Watson","Budget":90000}},{"Course\_id":"BIO-301","Title":"Genetics","Department\_name":"Biology","Credits":4,"Prereq\_id":"BIO-101","Sec\_id":1,"Semester":"Summer","Year":2010,"Teaches\_Dept":{"Building":"Watson","Budget":90000}}]})

db.TEACHER.insertOne({"ID":83821,"Name":"Brandt","Dept\_name":"Comp. Sci.","Salary":92000,"Teaches":[{"Course\_id":"CS-190","Title":"Game Design","Department\_name":"Comp. Sci.","Credits":4,"Prereq\_id":"CS-101","Sec\_id":1,"Semester":"Spring","Year":2009,"Teaches\_Dept":{"Building":"Taylor","Budget":100000}},{"Course\_id":"CS-190","Title":"Game Design","Department\_name":"Comp. Sci.","Credits":4,"Prereq\_id":"CS-101","Sec\_id":2,"Semester":"Spring","Year":2009,"Teaches\_Dept":{"Building":"Taylor","Budget":100000}},{"Course\_id":"CS-319","Title":"Image Processing","Department\_name":"Comp. Sci.","Credits":3,"Prereq\_id":"CS-101","Sec\_id":2,"Semester":"Spring","Year":2010,"Teaches\_Dept":{"Building":"Taylor","Budget":100000}}]})

db.TEACHER.insertOne({"ID":null,"Name":"","Dept\_name":"","Salary":null,"Teaches":[{"Course\_id":"BIO-399","Title":"Computational Biology","Department\_name":"Biology","Credits":3,"Prereq\_id":"BIO-101","Sec\_id":null,"Semester":"","Year":null,"Teaches\_Dept":{"Building":"Watson","Budget":90000}}]})

2)

a) db.TEACHER.find({})

b) db.TEACHER.find({"Name":{"$ne":""}},{"Name":1,"Dept\_name":1,"\_id":0})

c) db.TEACHER.find({"Dept\_name":{"$eq":"Comp. Sci."}},{"Name":1,"Salary":1,"\_id":0})

d) db.TEACHER.find({"Dept\_name":{"$eq":"Physics"},"Salary":{"$lt":90000}},{"\_id":0})

e) db.TEACHER.find({"Dept\_name":{"$nin":["Comp. Sci.",""]}},{"Name":1,"Dept\_name":1,"\_id":0})

f) db.TEACHER.distinct("Dept\_name",{"Dept\_name":{"$ne":""}})

g) db.TEACHER.aggregate([{"$project":{"Teaches":{"$filter":{"input":"$Teaches","as":"teach"," cond":{"$and":[{"$eq":["$$teach.Semester","Spring"]},{"$eq":["$$teach.Year",2009]}]}}}}},{"$match":{"Teaches":{"$ne":[]}}},{"$project":{"Teaches.Course\_id":1,"\_id":0}}])

h) db.TEACHER.aggregate([{"$match":{"Teaches.Department\_name":"Comp. Sci."}},{"$project":{"Teaches":{"$filter":{"input":"$Teaches","as":"teach","cond":{"$ne":["$$teach.Credits",3]}}}}},{"$match":{"Teaches":{"$ne":[]}}},{"$project":{"Teaches.Title":1,"\_id":0}}])

i) db.TEACHER.find({},{"Teaches.Course\_id":1,"Teaches.Title":1,"Teaches.Department\_name":1, "Teaches.Credits":1,"\_id":0}).sort({"Teahes.Department\_name":-1})

j) db.TEACHER.updateMany({},{$set:{"Date\_of\_Join":null}})

k) db.TEACHER.updateOne({"ID":10101},{"$set":{"Date\_of\_Join":new Date("1980-10-12")}})

l) db.TEACHER.updateMany({},{"$rename":{"Dept\_name":"Department"}},false,true)

n) db.TEACHER.updateMany({},[{"$set":{"Teaches":{"$map":{"input":"$Teaches", "in":{"Sub\_code":"$$this.Course\_id","Title":"$$this.Title","Department\_name":"$$this.Department\_name","Credits":"$$this.Credits","Prereq\_id":"$$this.Prereq\_id","Sec\_id":"$$this.Sec\_id","Semester":"$$this.Semester","Year":"$$this.Year","Teaches\_Dept":{"Building":"$$this.Teaches\_Dept.Building","Budget":"$$this.Teaches\_Dept.Budget"}}}}}}])

p) db.TEACHER.renameCollection("FACULTY\_INFO")

r) db.FACULTY\_INFO.updateMany({},{"$unset":{"Teaches.$[].Prereq\_id": 1}},false,true)

s) db.rough.updateMany({},[{"$set":{"Teaches":{"$map":{"input":"$Teaches", "in":{"Sub\_code":"$$this.Sub\_code","Title":"$$this.Title","Department\_name":"$$this.Department\_name","Credits":"$$this.Credits","Prereq\_id":"$$this.Prereq\_id","Sec\_id":"$$this.Sec\_id","Semester":"$$this.Semester","Year":"$$this.Year","Teaches\_Dept":{"Builder":"$$this.Teaches\_Dept.Building","Budget":"$$this.Teaches\_Dept.Budget"}}}}}}])

ASSIGNMENT-2

1)

a) db.FACULTY\_INFO.find({"Teaches.Department\_name":{"$in":['Physics','Music','Finance', 'Biology']}}, {"Name":1,"Teaches.Sub\_code":1,"Teaches.Title":1,"Teaches.Credits":1,"\_id":0})

b) db.FACULTY\_INFO.find({"Name":{$regex:/K/},"Salary":{"$gt":65000}})

c) db.FACULTY\_INFO.updateMany({},[{"$set":{"Gross\_Salary":{"$add":[{"$multiply": [{"$add":[1.05,0.2]},"$Salary"]},"$Salary"]}}}])

db.FACULTY\_INFO.updateMany({},[{"$set":{"Net\_Salary":{"$subtract":["$Gross\_Salary",{"$multiply":["$Salary",0.3]}]}}}])

d) db.FACULTY\_INFO.find({"Salary":{"$gte":60000,"$lte":80000}})

e) db.FACULTY\_INFO.find({"Name":{"$regex":/^[A-Z][r][a-z]\*/}})

f) db.FACULTY\_INFO.find({"Department":"Comp. Sci."},{"Name":1,"\_id":0}).sort({"Salary":-1})

g) db.FACULTY\_INFO2.updateMany({},[{"$set":{"Salary":{"$add":[{"$multiply":["$Salary",0.15]}, "$Salary"]}}}])

h) db.FACULTY\_INFO3.updateMany({"Department":"Comp. Sci.","Salary":{"$lt":70000}},[{"$set":{"Salary":{"$add":[{"$multiply":["$Salary",0.03]},"$Salary"]}}}])

i) db.FACULTY\_INFO.updateMany({},[{"$set":{"Annual\_Salary":{"$multiply":["$Salary",12]}}}])

j) db.FACULTY\_INFO.updateMany({"Teaches.Title":"Game Design"},[{"$set":{"Teaches.Title":"Game Theory"}}])

k) db.Faculty\_Info3.deleteMany({"Department":"History"})

l) db.FACUTY\_INFO3.updateMany({"Teaches.Sub\_code":{"$regex":/^BIO/}}, {"$unset":{"Teaches.$[].Sub\_code":"","Teaches.$[].Title":"","Teaches.$[].Department\_name":"", "Teaches.$[].Credits":""}})

2)

a) db.FACULTY\_INFO.aggregate([{"$match":{"Department":"Physics"}},{"$group":{"\_id":1, "Avg\_Salary":{"$avg":"$Salary"}}},{"$project":{"Avg\_Salary":1,"\_id":0,"Department":1}}])

b) db.FACULTY\_INFO.aggregate([{"$group":{"\_id":"$Department","Avg\_Salary":{"$avg":"$Salary"}}}, {"$match":{"Avg\_Salary":{"$ne":null}}},{"$project":{"\_id":0,"Department":"$\_id","Avg\_Salary":1}}])

c) db.FACULTY\_INFO.find({},{"ID":1,"Name":1,"Department":1,"Salary":1}).sort({"Salary":-1}).limit(1)

d) db.FACULTY\_INFO.find({"Department":"Comp. Sci."}).count()

e) db.FACULTY\_INFO.aggregate([{"$match":{"Teaches.Department\_name":"Comp. Sci."}},{"$unwind":"$Teaches"},{"$group":{"\_id":"$Teaches.Department\_name","Total\_credits":{"$sum":"$Teaches.Credits"}}},{"$project":{"\_id":0,"Teaches.Department\_name":"$\_id","Total\_credits":1}}])

f) db.FACULTY\_INFO.aggregate([{"$match":{"Department":{"$in":["Comp. Sci.","Physics"]}}},{"$group":{"\_id":"$Department","Total\_Salary":{"$sum":"$Salary"},"No\_of\_Instructors":{"$count":{}}}},{"$match":{"Total\_Salary":{"$ne":0}}},{"$project":{"\_id":0,"Department":"$\_id","Total\_Salary":1,"No\_of\_Instructors":1}}])

g) db.FACULTY\_INFO.aggregate([{"$match":{"Teaches.Department\_name":{"$in":["Comp. Sci.", "Biology"]}}}, {"$unwind":"$Teaches"}, {"$group":{"\_id":"$Teaches.Department\_name","Total\_credits":{"$sum":"$Teaches.Credits"}}}, {"$project":{"\_id":0,"Teaches.Department\_name":"$\_id","Total\_credits":1}}])

h) db.FACULTY\_INFO.aggregate([{"$unwind":"$Teaches"},{"$group":{ "\_id":"$Teaches.Teaches\_Dept.Builder","Total\_budget":{"$sum":"$Teaches.Teaches\_Dept.Budget"}}},{"$match":{"Total\_budget":{"$ne":0}}},{"$project":{"\_id":0,"Teaches.Teaches\_Dept.Builder":"$\_id", "Total\_budget":1}}])

i) db.FACULTY\_INFO.aggregate([{"$match":{"Department":{"$ne":""}}}, {"$group":{"\_id":"$Department","No\_of\_Instructors":{"$count":{}}}},{"$project":{"\_id":0, "Department":"$\_id","No\_of\_Instructors":1}}])

j) db.FACULTY\_INFO.aggregate([{"$match":{"Department":{"$ne":""}}}, {"$group":{"\_id":"$Department","No\_of\_Instructors":{"$count":{}}}},{"$project":{"\_id":0, "Department":"$\_id","Total\_Salary":1,"No\_of\_Instructors":1}}]).sort({"No\_of\_Instructors":-1})

k) db.FACULTY\_INFO.aggregate([{"$group":{"\_id":"$Teaches.Semester", "No\_of\_Courses":{"$count":{}}}},{"$project":{"\_id":0,"Teaches.Semester":"$\_id", "No\_of\_Courses":1}}])

l) db.FACULTY\_INFO.aggregate([{"$group":{"\_id":"$Department","Instructor\_Count":{"$count":{}}}}, {"$match":{"Instructor\_Count":{"$lt",2}}},

{"$project":{"\_id":0,"Teaches.Semester":"$\_id","No\_of\_Courses":1}}])

m) db.FACULTY\_INFO.aggregate([{"$match":{"Department":{"$ne":"Finance"}}}, {"$group":{"\_id":"$Department","Instructor\_Count":{"$count":{}}}},{"$match":{"Instructor\_Count":{"$gte",2}}},{"$project":{"\_id":0,"Department":"$\_id","Instructor\_Count":1}}])

n) db.FACULTY\_INFO.aggregate([{"$group":{"\_id":"$Department", "Total\_Salary":{"$sum":"$Salary"}}}, {"$match":{"Total\_Salary":{"$ne":0,"$gt":50000}}}, {"$project":{"\_id":0,"Department":"$\_id", "Total\_Salary":1}}])

o) db.FACULTY\_INFO.aggregate([{"$match" :{"Teaches.Teaches\_Dept.Builder": { "$eq": "Watson" } } }, { "$unwind": "$Teaches" }, { "$group": { "\_id": "$Teaches.Teaches\_Dept.Builder", "Total\_budget": { "$sum": "$Teaches.Teaches\_Dept.Budget" } } }, { "$project": { "\_id": 0, "Teaches.Teaches\_Dept.Builder": "$\_id", "Total\_budget": 1 } }])

p) db.FACULTY\_INFO.find({"Department":"Comp. Sci."},{"Name":1,"Salary":1,"Department":1,"\_id":0}).sort({"Salary":-1}).limit(1)

ASSIGNMENT-3

1)

db.CUSTOMERS.insertOne({"Cust\_no":"C0001","Name":"Raj Anand Singh","Phone\_no":9861258466,"City":"Delhi","Account":[{"Account\_no":"A0005","Type":"FD","Balance":28500,"Branch\_code":"B005","Branch\_name":"SaltLake Branch","Branch\_city":"Kolkata"}],"Loan":{"Loan\_no":"L0002","Amount":50000,"Branch\_code":"B005","Branch\_name":"SaltLake Branch","Branch\_city":"Kolkata","Installment":[{"Inst\_no":1,"Inst\_Amount":10000},{"Inst\_no":2,"Inst\_Amount":10000},{"Inst\_no":3,"Inst\_Amount":10000}]}})

db.CUSTOMERS.insertOne({"Cust\_no":"C0002","Name":"Ankita Singh","Phone\_no":9879958651,"City":"Bangalore","Account":[{"Account\_no":"A0005","Type":"FD","Balance":28500,"Branch\_code":"B005","Branch\_name":"SaltLake Branch","Branch\_city":"Kolkata"}],"Loan":{"Loan\_no":"L0003","Amount":8000000,"Branch\_code":"B004","Branch\_name":"Andheri Branch","Branch\_city":"Mumbai","Installment":[{"Inst\_no":1,"Inst\_Amount":50000},{"Inst\_no":2,"Inst\_Amount":50000}]}})

db.CUSTOMERS.insertOne({"Cust\_no":"C0003","Name":"Soumya Jha","Phone\_no":9885623344,"City":"Mumbai","Account":[{"Account\_no":"A0001","Type":"SB","Balance":20000,"Branch\_code":"B003","Branch\_name":"Juhu Branch","Branch\_city":"Mumbai"}],"Loan":{"Loan\_no":"","Amount":null,"Branch\_code":"","Branch\_name":"","Branch\_city":"","Installment":[{"Inst\_no":null,"Inst\_Amount":null}]}})

db.CUSTOMERS.insertOne({"Cust\_no":"C0004","Name":"Abhijit Mishra","Phone\_no":9455845425,"City":"Mumbai","Account":[{"Account\_no":"A0001","Type":"SB","Balance":20000,"Branch\_code":"B003","Branch\_name":"Juhu Branch","Branch\_city":"Mumbai"},{"Account\_no":"A0002","Type":"SB","Balance":15000,"Branch\_code":"B002","Branch\_name":"Chandnichowk Branch","Branch\_city":"Delhi"}],"Loan":{"Loan\_no":"","Amount":null,"Branch\_code":"","Branch\_name":"","Branch\_city":"","Installment":[{"Inst\_no":null,"Inst\_Amount":null}]}})

db.CUSTOMERS.insertOne({"Cust\_no":"C0005","Name":"Yash Saraf","Phone\_no":9665854585,"City":"Kolkata","Account":[{"Account\_no":"","Type":"","Balance":null,"Branch\_code":"","Branch\_name":"","Branch\_city":""}],"Loan":{"Loan\_no":"L0001","Amount":3000000,"Branch\_code":"B006","Branch\_name":"Srirampuram Branch","Branch\_city":"Chennai","Installment":[{"Inst\_no":null,"Inst\_Amount":null}]}})

db.CUSTOMERS.insertOne({"Cust\_no":"C0006","Name":"Swaroop Ray","Phone\_no":9437855466,"City":"Chennai","Account":[{"Account\_no":"A0003","Type":"CA","Balance":850000,"Branch\_code":"B004","Branch\_name":"Andheri Branch","Branch\_city":"Mumbai"},{"Account\_no":"A0004","Type":"CA","Balance":35000,"Branch\_code":"B004","Branch\_name":"Andheri Branch","Branch\_city":"Mumbai"},{"Account\_no":"A0010","Type":"CA","Balance":99000,"Branch\_code":"B004","Branch\_name":"Andheri Branch","Branch\_city":"Mumbai"}],"Loan":{"Loan\_no":"","Amount":null,"Branch\_code":"","Branch\_name":"","Branch\_city":"","Installment":[{"Inst\_no":null,"Inst\_Amount":null}]}})

db.CUSTOMERS.insertOne({"Cust\_no":"C0007","Name":"Surya Narayan","Phone\_no":9937955215,"City":"Gurgaon","Account":[{"Account\_no":"A0009","Type":"SB","Balance":18750,"Branch\_code":"B003","Branch\_name":"Juhu Branch","Branch\_city":"Mumbai"}],"Loan":{"Loan\_no":"","Amount":null,"Branch\_code":"","Branch\_name":"","Branch\_city":"","Installment":[{"Inst\_no":null,"Inst\_Amount":null}]}})

db.CUSTOMERS.insertOne({"Cust\_no":"C0008","Name":"Pranav Praveen","Phone\_no":9336652441,"City":"Pune","Account":[{"Account\_no":"A0008","Type":"SB","Balance":7200,"Branch\_code":"B002","Branch\_name":"Chandnichowk Branch","Branch\_city":"Delhi"}],"Loan":{"Loan\_no":"L0006","Amount":25000,"Branch\_code":"B006","Branch\_name":"Srirampuram Branch","Branch\_city":"Chennai","Installment":[{"Inst\_no":1,"Inst\_Amount":3000}]}})

db.CUSTOMERS.insertOne({"Cust\_no":"C0009","Name":"Stuti Mishra","Phone\_no":7870266534,"City":"Delhi","Account":[{"Account\_no":"A0007","Type":"SB","Balance":48000,"Branch\_code":"B001","Branch\_name":"Janakpuri Branch","Branch\_city":"Delhi"}],"Loan":{"Loan\_no":"L0005","Amount":9500000,"Branch\_code":"B005","Branch\_name":"SaltLake Branch","Branch\_city":"Kolkata","Installment":[{"Inst\_no":1,"Inst\_Amount":500000},{"Inst\_no":2,"Inst\_Amount":500000}]}})

db.CUSTOMERS.insertOne({"Cust\_no":"C0010","Name":"Aslesha Tiwari","Phone\_no":null,"City":"Mumbai","Account":[{"Account\_no":"A0006","Type":"FD","Balance":550000,"Branch\_code":"B005","Branch\_name":"SaltLake Branch","Branch\_city":"Kolkata"}],"Loan":{"Loan\_no":"L0004","Amount":100000,"Branch\_code":"B004","Branch\_name":"Andheri Branch","Branch\_city":"Mumbai","Installment":[{"Inst\_no":1,"Inst\_Amount":20000},{"Inst\_no":2,"Inst\_Amount":20000}]}})

2)a) db.CUSTOMERS.find({"Account.Account\_no":"A0004"},{"Name":1,"Phone\_no":1,"Cust\_no":1, "\_id":0})

b) db.CUSTOMERS.find({"Name":"Yash Saraf"},{"Loan.Amount":1,"Loan.Branch\_code":1,"\_id":0})

c) db.CUSTOMERS.find({"Loan.Loan\_no":""},{"Name":1,"\_id":0})

d) db.CUSTOMERS.find({"Cust\_no":"C0010"},{"Account.Account\_no":1,"Account.Balance":1,"\_id":0})

e) db.CUSTOMERS.find({"Name":"Aslesha Tiwari"},{"Loan.Branch\_city":1,"\_id":0})

f) db.CUSTOMERS.find({"Name":"Ankita Singh"}, {"Loan.Loan\_no":1, "Loan":{"Installment":{"Inst\_no":1, "Inst\_Amount":1}},"\_id":0})

g) db.CUSTOMERS.find({"Name":"Abhijit Mishra"}, {"Account.Branch\_name":1, "Account.Branch\_city":1, "\_id":0})

h) db.CUSTOMERS.aggregate([{"$unwind":"$Account"}, {"$match":{"Account.Balance":{"$lt":50000}}}, {"$group":{"\_id":{"Account\_no":"$Account.Account\_no", "Type":"$Account.Type"}}}, {"$project":{"\_id":0,"Account\_no":"$\_id.Account\_no","Type":"$\_id.Type"}},{"$out":"ACCOUNT\_TYPE"}])

i) const accountNos = db.CUSTOMERS.findOne({ Cust\_no: 'C0007' }, { 'Account.Account\_no': 1, \_id: 0 }).Account.map(doc => doc.Account\_no);

db.ACCOUNT\_TYPE.updateMany({ Account\_no:{$in:accountNos}},{$set:{Type:'FD'}})

j) const n = db.CUSTOMERS.aggregate([{"$project":{"Account":{"$filter":{"input":"$Account","as":"acc","cond":{"$lt":["$$acc.Balance",20000]}}}}},{"$match":{"Account":{"$ne":[]},"Account.Account\_no":{"$ne":""}}},{"$project":{"Account.Account\_no":1,"\_id":0}}]).toArray();

let arr=[];n.forEach(function (arrayItem){var x=arrayItem.Account;x.forEach(function (p){var z=p.Account\_no;arr.push(z);})});

db.ACCOUNT\_TYPE.deleteMany({"Account\_no":{"$in":arr}})

k) db.CUSTOMERS.find({"Account.Account\_no":{"$ne":""},"Loan.Loan\_no":{"$ne":""}}, {"Name":1,"\_id":0})

l) db.CUSTOMERS.find({"Account.Account\_no":{"$eq":""},"Loan.Loan\_no":{"$ne":""}}, {"Name":1,"\_id":0})

m)db.CUSTOMERS.aggregate([{"$unwind":"$Account"},{"$group":{"\_id":"$Name","Account\_count":{"$count":{}}}},{"$match":{"Account\_count":{"$gt":1}}},{"$project":{"Name":"$\_id","\_id":0,"Account\_count":1}}])

o) const x= db.CUSTOMERS.aggregate([{"$unwind":"$Account"}, {"$match":{"Account.Account\_no":"A0001"}},{"$group":{"\_id":{"Type":"$Account.Type"}}},{"$project":{"Type":"$\_id.Type","\_id":0}}]).toArray();

const x1=[];x.forEach(function (item){x1.push(item.Type);});

const y = db.CUSTOMERS.aggregate([{"$unwind":"$Account"}, {"$match":{"Account.Account\_no":"A0001"}},{"$group":{"\_id":{"Branch\_code":"$Account.Branch\_code"}}},{"$project":{"Branch\_code":"$\_id.Branch\_code","\_id":0}}]).toArray()

const y1=[]; y.forEach(function (item){y1.push(item.Branch\_code)});

db.CUSTOMERS.find({"Account.Type":{"$in":x1},"Account.Branch\_code":{"$in":y1},"Account.Account\_no":{"$ne":"A0001"}},{"Account.Account\_no":1,"Account.Type":1,"Account.Balance":1,"Account.Branch\_code":1,"\_id":0})

p) db.CUSTOMERS.aggregate([{"$unwind":"$Account"},{"$group":{"\_id":"$Name", "Account\_count":{"$count":{}}}},{"$match":{"Account\_count":{"$gt":1}}},{"$project":{"Name":"$\_id","\_id":0,"Account\_count":1}}])

q) db.CUSTOMERS.aggregate([{"$unwind":"$Account"}, {"$group":{"\_id":"$Account.Branch\_code","Average\_balance":{"$avg":"$Account.Balance"}}}, {"$match":{"Average\_balance":{"$gt":60000}}},{"$project":{"Branch\_code":"$\_id","Average\_balance":1,"\_id":0}}])

r) const t = db.CUSTOMERS.aggregate([{"$unwind":"$Account"}, {"$match":{"Account.Type":{"$eq":"FD"}}},{"$group":{"\_id":"$Account.Type","Maximum\_balance":{"$max":"$Account.Balance"}}},{"$project":{"\_id":0,"Maximum\_balance":1}}]).toArray()

var max=0;t.forEach(function (item){max=item.Maximum\_balance;});

db.CUSTOMERS.aggregate([{"$unwind":"$Account"},{"$match":{"Account.Balance":{"$gt":max}}},{"$project":{"\_id":0,"Account.Account\_no":1}}])

t) db.CUSTOMERS.aggregate([{"$match":{"Loan.Branch\_code":{"$ne":""}}},{"$group":{"\_id": {"Branch\_code":"$Loan.Branch\_code","Branch\_name":"$Loan.Branch\_name","Branch\_city":"$Loan.Branch\_city"}}},{"$project":{"\_id":0,"Branch\_code":"$\_id.Branch\_code","Branch\_name":"$\_id.Branch\_name","Branch\_city":"$\_id.Branch\_city"}}])

u) db.CUSTOMERS.aggregate([{"$match":{"Loan.Loan\_no":{"$ne":""},"Loan.Installment.Inst\_no": {"$eq":null}}},{"$group":{"\_id":{"Branch\_code":"$Loan.Branch\_code","Amount":"$Loan.Amount","Loan\_no":"$Loan.Loan\_no","Cust\_no":"$Cust\_no"}}},{"$project":{"\_id":0,"Cust\_no":"$\_id.Cust\_no","Branch\_code":"$\_id.Branch\_code","Amount":"$\_id.Amount","Loan\_no":"$\_id.Loan\_no"}}])

v) db.CUSTOMERS.aggregate([{"$project":{"Cust\_no":1,"Account":{"$map":{"input":"$Account", "as":"acc","in":{"$mergeObjects":["$$acc",{"Balance":{"$cond":{"if":{"$gt":["$$acc.Balance",80000]},"then":{"$multiply":["$$acc.Balance",1.06]},"else":{"$multiply":["$$acc.Balance",1.05]}}}}]}}}}}])

3)a) db.CUSTOMERS.find({"Loan.Branch\_city":"Mumbai"},{"Loan.Loan\_no":1,"\_id":0})

b)db.CUSTOMERS.aggregate([{"$unwind":"$Account"},{"$match":{"Account.Branch\_city":"Delhi"}}, {"$group":{"\_id":{"Type":"$Account.Type","Branch\_city":"$Account.Branch\_city"}}},{"$project":{"Account.Type":"$\_id.Type","\_id":0}}])

c)db.CUSTOMERS.find({"Account.Balance":{"$gt":100000}},{"Name":1,"Phone\_no":1,"\_id":0})

d) db.CUSTOMERS.find({"Name":"Raj Anand Singh"}, {"\_id":0,"Loan.Installment.Inst\_no":1, "Loan.Installment.Inst\_Amount":1})

e)db.CUSTOMERS.aggregate([{"$unwind":"$Account"},{"$match":{"Account.Type":{"$nin": ["SB",""]}}}, {"$group":{"\_id":{"Name":"$Name","Type":"$Account.Type"}}},{"$project": {"Name":"$\_id.Name","Type":"$\_id.Type","\_id":0}}])

f)db.CUSTOMERS.find({"Loan.Installment.Inst\_Amount":50000},{"Name":1,"\_id":0})

g)db.CUSTOMERS.find({"Account.Branch\_name":{"$regex":/^SaltLake/}},{"Name":1,"Phone\_no":1, "\_id":0})

h) db.CUSTOMERS.find({"Name":"Abhijit Mishra"}, {"\_id":0, "Account.Branch\_name":1, "Account.Branch\_city":1})

i) db.CUSTOMERS.find({"Name":"Swaroop Ray"},{"\_id":0,"Account.Type":1,"Account.Balance":1})

j)db.CUSTOMERS.aggregate([{"$unwind":"$Account"},{"$group":{"\_id":"$Name", "Account\_count":{"$count":{}}}},{"$match":{"Account\_count":{"$gt":1}}},{"$project":{"Name":"$\_id","\_id":0,"Account\_count":1}}])

k) const t = db.CUSTOMERS.aggregate([{"$unwind":"$Account"},{"$group":{ "\_id":"$Account.Balance", "Max\_balance":{"$max":"$Account.Balance"}}},{"$project":{"\_id":0, "Max\_balance":1}},{"$sort":{"Max\_balance":-1}},{"$limit":1}]).toArray()

var max1=0;t.forEach(function (item){max1=item.Max\_balance;});

db.CUSTOMERS.aggregate([{"$unwind":"$Account"},{"$match":{"Account.Balance":max1}},{"$project":{"\_id":0,"Name":1,"Account.Account\_no":1,"Account.Balance":1}}])

l) var result= db.CUSTOMERS.aggregate([{"$unwind":"$Account"},{"$match":{ "Account.Branch\_code": {"$ne":""}}},{"$group":{"\_id":"$Account.Branch\_code","Total\_Balance": {"$sum":"$Account.Balance"}}},{"$project":{"\_id":0,"Total\_Balance":1}}]).toArray()

var new\_totalBalance=0,count=0;result.forEach(function (item){new\_totalBalance+=item.Total\_Balance;count++;})

var Avg\_Balance=new\_totalBalance/count;

db.CUSTOMERS.aggregate([{"$unwind":"$Account"},{"$match":{"Account.Branch\_code":{"$ne":""}}},{"$group":{"\_id":"$Account.Branch\_code","Total\_balance":{"$sum":"$Account.Balance"}}},{"$match":{"Total\_balance":{"$gt":Avg\_Balance}}},{"$project":{"\_id":0,"Account.Branch\_code":"$\_id","Total\_balance":1}}]

ASSIGNMENT-4

**Customer.js**

import React, { useEffect, useState } from 'react';

import { Link } from 'react-router-dom';

import axios from 'axios';

import 'bootstrap/dist/css/bootstrap.min.css';

function Customer() {

const [inputs, setInputs] = useState([]);

useEffect(() => {

axios.get('http://localhost:7010/customer')

.then(res => {

setInputs(res.data);

})

.catch(errors => console.log(errors))

}, []);

const deleteCustomer = (e,id) => {

e.preventDefault();

const thisClicked = e.currentTarget;

const confirmation = window.confirm("Are you sure you want to delete?")

if(confirmation)

{

thisClicked.innerText = "Deleting...";

axios.delete(`http://localhost:7010/customer/delete/${id}`)

.then((res)=>{

alert("Customer deleted successfully")

thisClicked.closest("tr").remove();

})

.catch(function (err) {

console.log(err)

})

}

}

var customerDetails = "";

customerDetails = inputs.map((input) => {

return (

<tr key={input.\_id}>

<td>{input.Cust\_no}</td>

<td>{input.Name}</td>

<td>{input.Phone\_no}</td>

<td>{input.City}</td>

<td>

<Link to={`/customer/${input.\_id}/update`} className="btn btn-success">Update</Link>

{/\* <Link to={`/customer/edit`}>

<button onClick={() => handleEdit(input.\_id,input.Cust\_no,input.Name,input.Phone\_no,input.City)}>Update</button>

</Link> \*/}

</td>

<td>

<button type='button' onClick={(e) => deleteCustomer(e,input.\_id)} className='btn btn-danger'>Delete</button>

</td>

</tr>

)

});

return (

<div className='container mt-5'>

<div className='row'>

<div className='col-md-12'>

<div className='card'>

<div className='card-header'>

<h4>Customers List

<Link to='/customer/insert' className='btn btn-primary float-end'>Insert</Link>

</h4>

</div>

<div className="card-body">

<table className='table table-striped'>

<thead>

<tr>

<th>CUST\_NO</th>

<th>NAME</th>

<th>PHONE\_NO</th>

<th>CITY</th>

<th>Update</th>

<th>Delete</th>

</tr>

</thead>

<tbody>

{customerDetails}

</tbody>

</table>

</div>

</div>

</div>

</div>

</div>

)

}

export default Customer;

**CustomerInsert.js**

import React, { useState } from "react";

import { Link, useNavigate } from 'react-router-dom';

import axios from "axios";

function CustomerInsert() {

const [inputs, setInputs] = useState({

Cust\_no: '',

Name: '',

Phone\_no: '',

City: ''

});

const history = useNavigate();

const handleInput = (event) => {

event.persist();

setInputs({...inputs, [event.target.name]:event.target.value});

}

const saveCustomer = (event) => {

event.preventDefault();

const data = {

Cust\_no: inputs.Cust\_no,

Name: inputs.Name,

Phone\_no: inputs.Phone\_no,

City: inputs.City

}

axios.post('http://localhost:7010/customer',data)

.then(res => {

// alert(res.data)

alert("New customer inserted")

history("/customer");

})

.catch(function (error) {

console.error(error)

})

}

return(

<div>

<div className='container mt-5'>

<div className='row'>

<div className='col-md-12'>

<div className='card'>

<div className='card-header'>

<h4>Insert Customers

<Link to='/customer' className='btn btn-danger float-end'>Back</Link>

</h4>

</div>

<div className="card-body">

<form onSubmit={saveCustomer}>

<div className="mb-3">

<label>CUST\_NO</label>

<input type="text" name="Cust\_no" value={inputs.Cust\_no} onChange={handleInput} className="form-control"></input>

</div>

<div className="mb-3">

<label>NAME</label>

<input type="text" name="Name" value={inputs.Name} onChange={handleInput} className="form-control"></input>

</div>

<div className="mb-3">

<label>PHONE\_NO</label>

<input type="tel" name="Phone\_no" value={inputs.Phone\_no} onChange={handleInput} className="form-control"></input>

</div>

<div className="mb-3">

<label>CITY</label>

<input type="text" name="City" value={inputs.City} onChange={handleInput} className="form-control"></input>

</div>

<div className="mb-3">

<button type="submit" className="btn btn-primary">Save Customer</button>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

</div>

)

}

export default CustomerInsert;

**CustomerUpdate.js**

import React, { useEffect, useState } from "react";

import { Link, useParams, useNavigate } from 'react-router-dom';

import axios from "axios";

function CustomerUpdate() {

const { id } = useParams();

const history = useNavigate();

const [inputs, setInputs] = useState({

Cust\_no: '',

Name: '',

Phone\_no: '',

City: '',

});

useEffect(() => {

axios.get(`http://localhost:7010/customer/update/${id}`)

.then(res => {

console.log(res.data)

setInputs(res.data)

})

.catch(function (error) {

console.error(error)

});

}, [id]);

const handleInput = (event) => {

event.persist();

setInputs({...inputs, [event.target.name]:event.target.value});

}

const updateCustomer = (event) => {

event.preventDefault();

const data = {

Cust\_no: inputs.Cust\_no,

Name: inputs.Name,

Phone\_no: inputs.Phone\_no,

City: inputs.City

}

console.log(data)

axios.put(`http://localhost:7010/customer/update/${id}`,data)

.then(res => {

// alert(res.data)

alert("Updated Customer details")

history("/customer");

})

.catch(function (error) {

console.error(error)

})

}

return(

<div>

<div className='container mt-5'>

<div className='row'>

<div className='col-md-12'>

<div className='card'>

<div className='card-header'>

<h4>Update Customers

<Link to='/customer' className='btn btn-danger float-end'>Back</Link>

</h4>

</div>

<div className="card-body">

<form onSubmit={updateCustomer}>

<div className="mb-3">

<label>CUST\_NO</label>

<input type="text" name="Cust\_no" title="Cust\_no" placeholder="Enter the customer no." value={inputs.Cust\_no} onChange={handleInput} className="form-control"></input>

</div>

<div className="mb-3">

<label>NAME</label>

<input type="text" name="Name" title="Name" placeholder="Enter the customer name" value={inputs.Name} onChange={handleInput} className="form-control"></input>

</div>

<div className="mb-3">

<label>PHONE\_NO</label>

<input type="tel" name="Phone\_no" title="Phone\_no" placeholder="Enter the customer phone no." value={inputs.Phone\_no} onChange={handleInput} className="form-control"></input>

</div>

<div className="mb-3">

<label>CITY</label>

<input type="text" name="City" title="City" placeholder="Enter the city" value={inputs.City} onChange={handleInput} className="form-control"></input>

</div>

<div className="mb-3">

<button type="submit" className="btn btn-primary">Update Customer</button>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

</div>

)

}

export default CustomerUpdate;

**customerRouter.js**

const express = require("express")

const mongoose = require('mongoose')

const Schema = mongoose.Schema;

const customerRouter = express.Router();

const customerSchema = new Schema({

Cust\_no: String,

Name: String,

Phone\_no: Number,

City: String

});

const Customer = mongoose.model("customer",customerSchema);

customerRouter.route('/')

.get((req,res,next) => {

Customer.find({})

.then((mod)=>{

res.statusCode = 200;

res.setHeader('Content-Type','application/json');

res.json(mod)

},(err) => next(err));

})

.post((req,res,next) => {

console.log('In nodejs')

console.log(req.body)

const cont ={

Cust\_no : req.body.Cust\_no,

Name : req.body.Name,

Phone\_no : req.body.Phone\_no,

City : req.body.City

}

Customer.create(cont)

.then((con) => {

console.log("New contact created",con)

res.statusCode=200;

res.setHeader('Content-Type','application/json');

res.json(con)

},(err)=>next(err))

.catch((err)=>next(err))

})

customerRouter.route('/update/:id')

.get((req,res,next) => {

console.log(req.params.id)

Customer.findById({"\_id":req.params.id})

.then((mod)=>{

res.statusCode = 200;

res.setHeader('Content-Type','application/json');

res.json(mod)

},(err) => next(err));

})

.put((req,res,next) => {

console.log('In nodejs')

console.log(req.body)

const id = req.params.id;

const update = req.body;

Customer.findByIdAndUpdate(id, {$set: update}, {new:true})

.then((cus) => {

console.log("Customer details updated",cus)

res.statusCode=200;

res.setHeader('Content-Type','application/json');

res.json(cus)

},(err)=>next(err))

.catch((err)=>next(err))

})

customerRouter.route('/delete/:id')

.delete((req, res, next) => {

const id = req.params.id; // Capture the customer ID from the URL parameter

Customer.findByIdAndDelete(id) // Use findByIdAndDelete to delete the customer document

.then((cus) => {

if (cus) {

console.log("Customer deleted", cus);

res.statusCode = 200;

res.setHeader('Content-Type', 'application/json');

res.json(cus);

} else {

const err = new Error(`Customer with ID ${id} not found`);

err.status = 404;

return next(err);

}

})

.catch((err) => next(err));

});

module.exports = customerRouter







