

  
  
// Insert data into MongoDB collections

db.bank.insertMany([

{ code: 'SBI', name: 'State Bank of India', address: 'Bangalore', phone: 1234567890 },

{ code: 'HDFC', name: 'HDFC Bank', address: 'Mumbai', phone: 9876543210 }

]);

db.branch.insertMany([

{ \_id: 1, name: 'SBI Bangalore Main', address: 'Bangalore Main Street', phone: 1111111111, b\_code: 'SBI' },

{ \_id: 2, name: 'SBI Bangalore Branch 2', address: 'Bangalore Street 2', phone: 2222222222, b\_code: 'SBI' },

{ \_id: 3, name: 'HDFC Mumbai Main', address: 'Mumbai Main Street', phone: 3333333333, b\_code: 'HDFC' }

]);

db.account.insertMany([

{ ac\_no: 1, ac\_type: 'joint', balance: 1000, bid: 1 },

{ ac\_no: 2, ac\_type: 'sb', balance: 500, bid: 1 },

{ ac\_no: 3, ac\_type: 'joint', balance: 1500, bid: 2 },

{ ac\_no: 4, ac\_type: 'sb', balance: 200, bid: 3 }

]);

db.customers.insertMany([

{ ssn: 123456789, name: 'Alice', address: 'Address 1', phone: 1111111111, age: 25 },

{ ssn: 234567890, name: 'Bob', address: 'Address 2', phone: 2222222222, age: 30 },

{ ssn: 345678901, name: 'Charlie', address: 'Address 3', phone: 3333333333, age: 35 },

{ ssn: 456789012, name: 'David', address: 'Address 4', phone: 4444444444, age: 40 }

]);

db.cust\_ac.insertMany([

{ acc\_no: 1, ssn: 123456789 },

{ acc\_no: 2, ssn: 123456789 },

{ acc\_no: 3, ssn: 234567890 },

{ acc\_no: 4, ssn: 345678901 }

]);

db.loan.insertMany([

{ lno: 1, duration: 12, interest: 5.5, amount: 2000, bid: 1 },

{ lno: 2, duration: 24, interest: 6.2, amount: 3000, bid: 2 },

{ lno: 3, duration: 18, interest: 5.8, amount: 2500, bid: 3 },

{ lno: 4, duration: 36, interest: 7.0, amount: 4000, bid: 1 }

]);

db.cust\_loan.insertMany([

{ lno: 1, ssn: 123456789 },

{ lno: 2, ssn: 234567890 },

{ lno: 3, ssn: 345678901 },

{ lno: 4, ssn: 456789012 }

]);

// Queries

// a) List the details of customers who have a joint account and also have at least one loan.

db.customers.find({

ssn: {

$in: db.cust\_ac.distinct("ssn", {

acc\_no: {

$in: db.account.distinct("ac\_no", { ac\_type: "joint" })

}

}),

$in: db.cust\_loan.distinct("ssn")

}

});

// b) List the details of the branch which has given the maximum loan.

db.branch.findOne({

\_id: db.loan.find().sort({ amount: -1 }).limit(1).next().bid

});

// c) List the details of saving accounts opened in the SBI branches located at Bangalore.

db.account.find({

ac\_type: 'sb',

bid: {

$in: db.branch.find({

b\_code: 'SBI',

address: /Bangalore/

}).toArray().map(b => b.\_id)

}

});

// d) List the name of the branch along with its b\nk name and total amount of loan given by it.

db.branch.aggregate([

{

$lookup: {

from: "loan",

localField: "\_id",

foreignField: "bid",

as: "loans"

}

},

{

$group: {

\_id: "$\_id",

branch\_name: { $first: "$name" },

bank\_name: { $first: "$b\_code" },

total\_loan\_amount: { $sum: "$loans.amount" }

}

}

]);

// e) Retrieve the names of customers who have accounts in all the branches located in a specific city.

db.customers.find({

ssn: {

$in: db.cust\_ac.aggregate([

{

$lookup: {

from: "account",

localField: "acc\_no",

foreignField: "ac\_no",

as: "accounts"

}

},

{

$lookup: {

from: "branch",

localField: "accounts.bid",

foreignField: "\_id",

as: "branches"

}

},

{

$match: {

"branches.address": "Bangalore"

}

},

{

$group: {

\_id: "$ssn",

branch\_count: { $sum: 1 }

}

},

{

$match: {

branch\_count: db.branch.count({ address: "Bangalore" })

}

}

]).map(ca => ca.\_id)

}

});