Name : Rajkumar B L
Reg.No : 2047120
Course : MCS 273 NoSQL (Lab 02)

Aggregation:

A. Creation of database of my domain – Car Rental System

DB name – binodcarshare

Table/Collection – members

```
> show collections
> db.createCollection('members')
{ "ok" : 1 }
> show collections
members
> db.members.insertMany([
... {MemId:01, Name:"Raj", Age:22, title:"Student", Address:{city:"Tup", State:
"TN"}, Booking:{car:"Alto", body:"hb", price:3000}, Balance:3000},
... {MemId:02, Name:"Rohan", Age:24, title:"Doctor", Address:{city:"Cbe", State
:"TN"}, Booking:{car:"Mazda", body:"sedan", price:5000}, Balance:5570},
... {MemId:03, Name:"Hareesma", Age:23, title:"Phsycatrist", Address:{city:"Ldn
", State:"ON"}, Booking:{car:"Kia", body:"Sedan", price:7000}, Balance:7570},
... {MemId:04, Name:"Prem", Age:30, title:"Engineer", Address:{city:"Blr", Stat
e:"KN"}, Booking:{car:"Kia", body:"Full", price:10000}, Balance:2000},
... {MemId:05, Name:"Kumar", Age:22, title:"Student", Address:{city:"Tup", Stat
e:"TN"}, Booking:{car:"None", body:"None", price:0}, Balance:0},
... {MemId:06, Name:"Kevin", Age:32, title:"Lawyer", Address:{city:"Cbe", State
:"TN"}, Booking:{car:"Inova", body:"Full", price:12000}, Balance:12970}
... ]);
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("601995668ed7c3052bbedee3"),
                ObjectId("601995668ed7c3052bbedee4"),
                ObjectId("601995668ed7c3052bbedee5"),
                ObjectId("601995668ed7c3052bbedee6"),
                ObjectId("601995668ed7c3052bbedee7"),
                ObjectId("601995668ed7c3052bbedee8")
        ]
}
>
```

B. Five Aggregate Operations.

I. Finding the count of numbers who are yet to pay the balance.

```
> db.members.aggregate(
... {$match:{Balance:{$gt:0}}},
... {$group:{_id:null,"Count_members_with_debt" : {"$sum":1}}}
... );
{ "_id" : null, "Count_members_with_debt" : 5 }
>
```

II. Finding the minimum balance payable.

```
> db.members.aggregate( {$match:{Balance:{$gt:0}}}, {$group:{_id:null,"Min Balance" : {"$min":"$Balance"}}} );
{ "_id" : null, "Min Balance" : 2000 }
```

III. Finding the maximum balance payable.

```
> db.members.aggregate( {$match:{Balance:{$gt:0}}}, {$group:{_id:null,"Max Balance" : {"$max":"$Balance"}}} );
{ "_id" : null, "Max Balance" : 12970 }
>
```

IV. Sum the company's total recivable amount from the members.

```
> db.members.aggregate([{ $group: { _id: null, Balance_Receivable: { $sum: "$Balance" } } } ]);
{ "_id" : null, "Balance_Receivable" : 31110 }
```

V. Getting the names of all payees from respective states

```
> db.members.aggregate( [{$match:{Balance:{$gt:0}}}, {$group: { _id:"$Address.State" , Payee_Names:{$addToSet:"$Name"}}}]);
{ "_id" : "TN", "Payee_Names" : [ "Rohan", "Raj", "Kevin" ] }
{ "_id" : "ON", "Payee_Names" : [ "Hareesma" ] }
{ "_id" : "KN", "Payee_Names" : [ "Prem" ] }
>
```

VI. Finiding the name, title and balance from respective states

```
> db.members.aggregate( [ {$group: { _id:"$Address.State" , BalanceByState:{$push:
 {Name:"$Name", Title:"$title", Balance:"$Balance"}}}} ] ).pretty();
{
        "_id" : "TN",
        "BalanceByState" : [
                {
                        "Name" : "Raj",
                        "Title" : "Student",
                        "Balance" : 3000
                },
                {
                        "Name" : "Rohan",
                        "Title" : "Doctor",
                        "Balance" : 5570
                },
                {
                        "Name" : "Kumar",
                        "Title" : "Student",
                        "Balance" : 0
                },
                {
                        "Name" : "Kevin",
                        "Title" : "Lawyer",
                        "Balance" : 12970
                }
        ]
}
```

```
{
        "_id" : "KN",
        "BalanceByState" : [
                {
                        "Name" : "Prem",
                        "Title" : "Engineer",
                        "Balance" : 2000
                }
        ]
}
{
        "_id" : "ON",
        "BalanceByState" : [
                {
                        "Name" : "Hareesma",
                        "Title" : "Phsycatrist",
                        "Balance" : 7570
                }
        ]
}
```

C. Use Group By.

Grouping member details by their city

```
> db.members.aggregate( [ {$group: { _id:"$Address.city" , Meber_details_by_City:
{$push:{Name:"$Name", Balance:"$Balance"}}}} ] ).pretty();
{
        "_id" : "Blr",
        "Meber_details_by_City" : [
                {
                        "Name" : "Prem",
                        "Balance" : 2000
                }
        ]
}
{
        "_id" : "Tup",
        "Meber_details_by_City" : [
                {
                        "Name" : "Raj",
                        "Balance" : 3000
                },
                {
                        "Name" : "Kumar",
                        "Balance" : 0
                }
        ]
}
{
        "_id" : "Cbe",
        "Meber_details_by_City" : [
                {
                        "Name" : "Rohan",
                        "Balance" : 5570
                },
                {
                        "Name" : "Kevin",
                        "Balance" : 12970
                }
        ]
```

```
{
        "_id" : "Ldn",
        "Meber_details_by_City" : [
                {
                        "Name" : "Hareesma",
                        "Balance" : 7570
                }
        ]
}
>
```

D. Sorting data in ascending and descending order

Find -> Sort using name in ascending order

```
> db.members.find({},{MemId:1,Name:1,Age:1}).sort({Name:1});
{ "_id" : ObjectId("601995668ed7c3052bbedee5"), "MemId" : 3, "Name" : "Hareesma", "Age" : 23 }
{ "_id" : ObjectId("601995668ed7c3052bbedee8"), "MemId" : 6, "Name" : "Kevin", "Age" : 32 }
{ "_id" : ObjectId("601995668ed7c3052bbedee7"), "MemId" : 5, "Name" : "Kumar", "Age" : 22 }
{ "_id" : ObjectId("601995668ed7c3052bbedee6"), "MemId" : 4, "Name" : "Prem", "Age" : 30 }
{ "_id" : ObjectId("601995668ed7c3052bbedee3"), "MemId" : 1, "Name" : "Raj", "Age" : 22 }
{ "_id" : ObjectId("601995668ed7c3052bbedee4"), "MemId" : 2, "Name" : "Rohan", "Age" : 24 }
>
```

Find -> Sort using name in descending order

```
> db.members.find({},{MemId:1,Name:1,Age:1}).sort({Name:-1});
{ "_id" : ObjectId("601995668ed7c3052bbedee4"), "MemId" : 2, "Name" : "Rohan", "Age" : 24 }
{ "_id" : ObjectId("601995668ed7c3052bbedee3"), "MemId" : 1, "Name" : "Raj", "Age" : 22 }
{ "_id" : ObjectId("601995668ed7c3052bbedee6"), "MemId" : 4, "Name" : "Prem", "Age" : 30 }
{ "_id" : ObjectId("601995668ed7c3052bbedee7"), "MemId" : 5, "Name" : "Kumar", "Age" : 22 }
{ "_id" : ObjectId("601995668ed7c3052bbedee8"), "MemId" : 6, "Name" : "Kevin", "Age" : 32 }
{ "_id" : ObjectId("601995668ed7c3052bbedee5"), "MemId" : 3, "Name" : "Hareesma", "Age" : 23 }
>
```

Aggregate -> Sort using id - State in ascending order

```
> db.members.aggregate( [ {$group: { _id:"$Address.State" , Payee_Names:
{$addToSet:"$Name"}}},{$sort:{_id:1}}]);
{ "_id" : "KN", "Payee_Names" : [ "Prem" ] }
{ "_id" : "ON", "Payee_Names" : [ "Hareesma" ] }
{ "_id" : "TN", "Payee_Names" : [ "Raj", "Kumar", "Kevin", "Rohan" ] }
>
```

Aggregate -> Sort using id - State in descending order

```
> db.members.aggregate( [ {$group: { _id:"$Address.State" , Payee_Names:
{$addToSet:"$Name"}}},{$sort:{_id:-1}}]);
{ "_id" : "TN", "Payee_Names" : [ "Kevin", "Kumar", "Raj", "Rohan" ] }
{ "_id" : "ON", "Payee_Names" : [ "Hareesma" ] }
{ "_id" : "KN", "Payee_Names" : [ "Prem" ] }
>
```