**PRACTICAL NO.10**

* **Aggregation Function :**

Microsoft Windows [Version 10.0.19045.6332]

(c) Microsoft Corporation. All rights reserved.

C:\Program Files\MongoDB\Server\8.2\bin>Mongosh

Current Mongosh Log ID: 68e4b8b640bf74e606cebea3

Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.8

Using MongoDB: 8.2.0

Using Mongosh: 2.5.8

For mongosh info see: <https://www.mongodb.com/docs/mongodb-shell/>

------

The server generated these startup warnings when booting

2025-10-03T16:02:33.628+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

------

test> show dbs;

Bank 8.00 KiB

admin 40.00 KiB

college 8.00 KiB

company 68.00 KiB

config 108.00 KiB

Bank>

demobhakti 80.00 KiB

demoexmple 68.00 KiB

local 64.00 KiB

test 56.00 KiB

bank> use Bank;

switched to db Bank

Bank> db.createCollection('Customer');

{ ok: 1 }

Bank> db.customer.insertOne({\_id:101,name:'Sai',bal:20000,bank:'SBI',location:'Pune'});

{ acknowledged: true, insertedId: 101 }

Bank> db.customer.insertOne({\_id:102,name:'Ram',bal:3000,bank:'SBI',location:'Shirdi'});

{ acknowledged: true, insertedId: 102 }

Bank> db.customer.insertOne({\_id:103,name:'Sita',bal:4000,bank:'IBM',location:'Satara'});

{ acknowledged: true, insertedId: 103 }

Bank> db.customer.insertOne({\_id:104,name:'Geeta',bal:40000,bank:'Baroda',location:'Maval'});

{ acknowledged: true, insertedId: 104 }

Bank> db.customer.count();

DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.

4

Bank> db.customer.aggregate([{$group:{\_id:null,total\_bal:{$sum:'$bal'}}}]);

[ { \_id: null, total\_bal: 67000 } ]

Bank> db.customer.aggregate([{$group:{\_id:null,total\_bal:{$min:'$bal'}}}]);

[ { \_id: null, total\_bal: 3000 } ]

Bank> db.customer.aggregate([{$group:{\_id:null,total\_bal:{$max:'$bal'}}}]);

[ { \_id: null, total\_bal: 40000 } ]

Bank> db.customer.aggregate([{$group:{\_id:null,total\_bal:{$avg:'$bal'}}}]);

[ { \_id: null, total\_bal: 16750 } ]

Bank> db.customer.aggregate([{$group:{\_id:null,total\_bal:{$push:'$bal'}}}]);

[ { \_id: null, total\_bal: [ 20000, 3000, 4000, 40000 ] } ]

Bank> db.customer.insertOne({\_id:105,name:'Seema',bal:40000,bank:'Baroda',location:'Nanaj'});

{ acknowledged: true, insertedId: 105 }

Bank> db.customer.aggregate([{$group:{\_id:null,total\_bal:{$push:'$bal'}}}]);

[ { \_id: null, total\_bal: [ 20000, 3000, 4000, 40000, 40000 ] } ]

Bank> db.customer.aggregate([{$group:{\_id:null,total\_bal:{$addToSet:'$bal'}}}]);

[ { \_id: null, total\_bal: [ 3000, 20000, 4000, 40000 ] } ]

Bank> db.customer.aggregate([{$group:{\_id:null,total\_bal:{$first:'$bal'}}}]);

[ { \_id: null, total\_bal: 20000 } ]

Bank> db.customer.aggregate([{$group:{\_id:null,total\_bal:{$last:'$bal'}}}]);

[ { \_id: null, total\_bal: 40000 } ]

Bank> db.customer.distinct("bal");

[ 3000, 4000, 20000, 40000 ]

* **ASCENDING /DESCENDING :**

Bank> db.customer.find().sort({name:1});

[emoexmple 68.00 KiB

{al 64.00 KiB

\_id: 104,56.00 KiB

name: 'Geeta',

bal: 40000,bank

bank: 'Baroda',ection('Customerr');

location: 'Maval'baseDifferCase]: db already exists with different case already have: [Bank

},ying to create [bank]

{ \_id: 102, name: 'Ram', bal: 3000, bank: 'SBI', location: 'Shirdi' },

{ \_id: 101, name: 'Sai', bal: 20000, bank: 'SBI', location: 'Pune' },

{k> db.dropDatabase;

\_id: 105,opDatabase] AsyncFunction {

name: 'Seema',, Infinity ],

bal: 40000,e: true,

bank: 'Baroda', '0.0.0', '999.999.999' ],

location: 'Nanaj'Set', 'Sharded', 'LoadBalanced', 'Standalone' ],

},turnType: { type: 'unknown', attributes: {} },

{eprecated: false,

\_id: 103,[ 'Compass', 'Browser', 'CLI' ],

name: 'Sita',mmand: false,

bal: 4000,put: false,

bank: 'IBM',mpleter: undefined,

location: 'Satara'eter: undefined,

}elp: [Function (anonymous)] Help

]

Bank> db.customer.find().sort({name:-1});

[lready on db bank

{k> db.createCollection('Customer');

\_id: 103,ror[DatabaseDifferCase]: db already exists with different case already have: [Bank

name: 'Sita',e [bank]

bal: 4000,;

bank: 'IBM',ank

location: 'Satara'ion('Customer');

},: 1 }

{k> db.customer.insertOne({\_id:101,name:'Sai',bal:20000,bank:'SBI',location:'Pune'});

\_id: 105,d: true, insertedId: 101 }

name: 'Seema',insertOne({\_id:102,name:'Ram',bal:3000,bank:'SBI',location:'Shirdi'});

bal: 40000, true, insertedId: 102 }

bank: 'Baroda',nsertOne({\_id:103,name:'Sita',bal:4000,bank:'IBM',location:'Satara'});

location: 'Nanaj' insertedId: 103 }

},> db.customer.insertOne({\_id:104,name:'Geeta',bal:40000,bank:'Baroda',location:'Maval'});

{ \_id: 101, name: 'Sai', bal: 20000, bank: 'SBI', location: 'Pune' },

{ \_id: 102, name: 'Ram', bal: 3000, bank: 'SBI', location: 'Shirdi' },

{recationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCo

\_id: 104,

name: 'Geeta',

bal: 40000,er.aggregate([{$group:{\_id:null,total\_bal:{$sum:'$bal'}}}]);

bank: 'Baroda',l\_bal: 67000 } ]

location: 'Maval'regate([{$group:{\_id:null,total\_bal:{$min:'$bal'}}}]);

} \_id: null, total\_bal: 3000 } ]

]ank> db.customer.aggregate([{$group:{\_id:null,total\_bal:{$max:'$bal'}}}]);

* **INDEXES:**

Bank> db.customer.createIndex({\_id:101});

\_id\_101b.customer.aggregate([{$group:{\_id:null,total\_bal:{$avg:'$bal'}}}]);

Bank> db.customer.createIndex({\_id:1});

\_id\_1

Bank> db.customer.createIndex({name:1});

name\_1

Bank> db.customer.createIndex({bal:1});

bal\_1

Bank> db.customer.createIndex({bank:1});

bank\_1

Bank> db.customer.createIndex({location:1});

location\_1

Bank> db.customer.find({\_id:101});

[

{ \_id: 101, name: 'Sai', bal: 20000, bank: 'SBI', location: 'Pune' }

]

Bank> db.customer.getIndexes();

[

{ v: 2, key: { \_id: 1 }, name: '\_id\_' },

{ v: 2, key: { \_id: 101 }, name: '\_id\_101' },

{ v: 2, key: { name: 1 }, name: 'name\_1' },

{ v: 2, key: { bal: 1 }, name: 'bal\_1' },

{ v: 2, key: { bank: 1 }, name: 'bank\_1' },

{ v: 2, key: { location: 1 }, name: 'location\_1' }