**Name: Rohan Chimaji Dhadke**

**Roll No: 13136**

**Group B: Practical No. 2** Design & Develop MongoDB Queries using Aggregation.

Please enter a MongoDB connection string (Default: mongodb://localhost/): db

db

Current Mongosh Log ID: 652cd4b02d29764c3d8888e5

Connecting to: mongodb://127.0.0.1:27017/db?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.0.1

Using MongoDB: 7.0.2

Using Mongosh: 2.0.1

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

------

The server generated these startup warnings when booting

2023-10-15T23:12:59.167-07:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

------

db> show dbs;

admin 40.00 KiB

config 84.00 KiB

local 72.00 KiB

stud 72.00 KiB

db> use stud;

switched to db stud

stud> use employee;

switched to db employee

employee> db.createCollection("employee\_data");

{ ok: 1 }

employee> db.employee\_data.insert([{empId: 101, empName: 'Rohan', salary: 29000}]);

DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.

{

acknowledged: true,

insertedIds: { '0': ObjectId("652cd68d2d29764c3d8888e6") }

}

employee> db.employee\_data.find();

[

{

\_id: ObjectId("652cd68d2d29764c3d8888e6"),

empId: 101,

empName: 'Rohan',

salary: 29000

}

]

employee> db.employee\_data.updateOne([{emp:101, empName:'Rohan', deptName:'comp', salary: 30000}]);

MongoInvalidArgumentError: Update document requires atomic operators

employee> db.employee\_data.updateOne([{emp:101, empName:'Rohan', deptName:'comp', salary: 30000}]);

MongoInvalidArgumentError: Update document requires atomic operators

employee> db.employee\_data.remove({empId: 101});

DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.

{ acknowledged: true, deletedCount: 1 }

employee> db.employee\_data.insert([{empId: 101, empName: 'Rohan',deptName: 'comp', salary: 29000}]);

{

acknowledged: true,

insertedIds: { '0': ObjectId("652cd7702d29764c3d8888e7") }

}

employee> db.employee\_data.insert([{empId: 102, empName: 'Anurag',deptName: 'IT', salary: 59000}, {empId: 103, empName: 'Ishvar', deptName: 'AI', salary: 50000} ]);

{

acknowledged: true,

insertedIds: {

'0': ObjectId("652cd8192d29764c3d8888e8"),

'1': ObjectId("652cd8192d29764c3d8888e9")

}

}

employee> db.employee\_data.insert([{empId: 104, empName: 'Vijay',deptName: 'comp', salary: 79000}, {empId: 105, empName: 'Nikhil', deptName: 'AI', salary: 50000} ]

... );

{

acknowledged: true,

insertedIds: {

'0': ObjectId("652cd8542d29764c3d8888ea"),

'1': ObjectId("652cd8542d29764c3d8888eb")

}

}

employee> db.employee\_data.find();

[

{

\_id: ObjectId("652cd7702d29764c3d8888e7"),

empId: 101,

empName: 'Rohan',

deptName: 'comp',

salary: 29000

},

{

\_id: ObjectId("652cd8192d29764c3d8888e8"),

empId: 102,

empName: 'Anurag',

deptName: 'IT',

salary: 59000

},

{

\_id: ObjectId("652cd8192d29764c3d8888e9"),

empId: 103,

empName: 'Ishvar',

deptName: 'AI',

salary: 50000

},

{

\_id: ObjectId("652cd8542d29764c3d8888ea"),

empId: 104,

empName: 'Vijay',

deptName: 'comp',

salary: 79000

},

{

\_id: ObjectId("652cd8542d29764c3d8888eb"),

empId: 105,

empName: 'Nikhil',

deptName: 'AI',

salary: 50000

}

]

employee> db.employee\_data.insert([{ empId: 106, empName: 'Suresh', deptName: 'comp', salary: 59000 }, { empId: 107, empName: 'Vaibhav', deptName: 'IT', salary: 50000 }] );

{

acknowledged: true,

insertedIds: {

'0': ObjectId("652cd8a12d29764c3d8888ec"),

'1': ObjectId("652cd8a12d29764c3d8888ed")

}

}

employee> db.employee\_data.find();

[

{

\_id: ObjectId("652cd7702d29764c3d8888e7"),

empId: 101,

empName: 'Rohan',

deptName: 'comp',

salary: 29000

},

{

\_id: ObjectId("652cd8192d29764c3d8888e8"),

empId: 102,

empName: 'Anurag',

deptName: 'IT',

salary: 59000

},

{

\_id: ObjectId("652cd8192d29764c3d8888e9"),

empId: 103,

empName: 'Ishvar',

deptName: 'AI',

salary: 50000

},

{

\_id: ObjectId("652cd8542d29764c3d8888ea"),

empId: 104,

empName: 'Vijay',

deptName: 'comp',

salary: 79000

},

{

\_id: ObjectId("652cd8542d29764c3d8888eb"),

empId: 105,

empName: 'Nikhil',

deptName: 'AI',

salary: 50000

},

{

\_id: ObjectId("652cd8a12d29764c3d8888ec"),

empId: 106,

empName: 'Suresh',

deptName: 'comp',

salary: 59000

},

{

\_id: ObjectId("652cd8a12d29764c3d8888ed"),

empId: 107,

empName: 'Vaibhav',

deptName: 'IT',

salary: 50000

}

]

employee> db.employee\_data.aggregate([{$group:{\_Id:"$deptName", Total.sum:{$sum:{$salary}}}}]);

Uncaught:

SyntaxError: Unexpected token, expected "," (1:59)

employee> db.employee\_data.aggregate([{$group:{\_Id:"$deptName", total.sum:{$sum:{$salary}}}}]);

Uncaught:

SyntaxError: Unexpected token, expected "," (1:59)

employee> db.employee\_data.aggregate([{$group:{\_Id:"$deptName", total\_sum:{$sum:{$salary}}}}]);

ReferenceError: $salary is not defined

employee> db.employee\_data.aggregate([{$group:{\_Id:"$deptName", total\_sum:{$sum:"$salary"}}}]);

MongoServerError: The field '\_Id' must be an accumulator object

employee> db.employee\_data.aggregate([{$group:{\_id:"$deptName", total\_sum:{$sum:{$salary}}}}]);

ReferenceError: $salary is not defined

employee> db.employee\_data.aggregate([{$group:{\_id:"$deptName", total\_sum:{$sum:{"$salary"}}}}]);

Uncaught:

SyntaxError: Unexpected token (1:80)

employee> db.employee\_data.aggregate([{$group:{\_id:"$deptName", total\_sum:{$sum:"$salary"}}}]);

[

{ \_id: 'AI', total\_sum: 100000 },

{ \_id: 'comp', total\_sum: 167000 },

{ \_id: 'IT', total\_sum: 109000 }

]

employee> db.employee\_data.aggregate([{$group:{\_id:" ", total\_sum:{$sum:"$salary"}}}]);

[ { \_id: ' ', total\_sum: 376000 } ]

employee> db.employee\_data.aggregate([{$group:{\_id:"$deptName", total\_sum:{$min:"$salary"}}}]);

[

{ \_id: 'IT', total\_sum: 50000 },

{ \_id: 'AI', total\_sum: 50000 },

{ \_id: 'comp', total\_sum: 29000 }

]

employee> db.employee\_data.fing();

TypeError: db.employee\_data.fing is not a function

employee> db.employee\_data.find();

[

{

\_id: ObjectId("652cd7702d29764c3d8888e7"),

empId: 101,

empName: 'Rohan',

deptName: 'comp',

salary: 29000

},

{

\_id: ObjectId("652cd8192d29764c3d8888e8"),

empId: 102,

empName: 'Anurag',

deptName: 'IT',

salary: 59000

},

{

\_id: ObjectId("652cd8192d29764c3d8888e9"),

empId: 103,

empName: 'Ishvar',

deptName: 'AI',

salary: 50000

},

{

\_id: ObjectId("652cd8542d29764c3d8888ea"),

empId: 104,

empName: 'Vijay',

deptName: 'comp',

salary: 79000

},

{

\_id: ObjectId("652cd8542d29764c3d8888eb"),

empId: 105,

empName: 'Nikhil',

deptName: 'AI',

salary: 50000

},

{

\_id: ObjectId("652cd8a12d29764c3d8888ec"),

empId: 106,

empName: 'Suresh',

deptName: 'comp',

salary: 59000

},

{

\_id: ObjectId("652cd8a12d29764c3d8888ed"),

empId: 107,

empName: 'Vaibhav',

deptName: 'IT',

salary: 50000

}

]

employee> db.employee\_data.aggregate([{$group:{\_id:"$deptName", total\_sum:{$max:"$salary"}}}]);

[

{ \_id: 'IT', total\_sum: 59000 },

{ \_id: 'AI', total\_sum: 50000 },

{ \_id: 'comp', total\_sum: 79000 }

]

employee> db.employee\_data.aggregate([{$group:{\_id:"$deptName", total\_sum:{$max:"$salary"}}}]);

[

{ \_id: 'IT', total\_sum: 59000 },

{ \_id: 'AI', total\_sum: 50000 },

{ \_id: 'comp', total\_sum: 79000 }

]

employee> db.employee\_data.aggregate([{$group:{\_id:"$deptName", total\_sum:{$mix:"$salary"}}}]);

MongoServerError: unknown group operator '$mix'

employee> db.employee\_data.aggregate([{$group:{\_id:"$deptName", total\_sum:{$min:"$salary"}}}]);

[

{ \_id: 'IT', total\_sum: 50000 },

{ \_id: 'AI', total\_sum: 50000 },

{ \_id: 'comp', total\_sum: 29000 }

]

employee> db.employee\_data.aggregate([{$group:{\_id:"$deptName", total\_sum:{$avg:"$salary"}}}]);

[

{ \_id: 'AI', total\_sum: 50000 },

{ \_id: 'comp', total\_sum: 55666.666666666664 },

{ \_id: 'IT', total\_sum: 54500 }

]

employee> db.employee\_data.aggregate([{$group:{\_id:"$deptName", total\_sum:{$first:"$salary"}}}]);

[

{ \_id: 'comp', total\_sum: 29000 },

{ \_id: 'IT', total\_sum: 59000 },

{ \_id: 'AI', total\_sum: 50000 }

]

employee> db.employee\_data.aggregate([{$group:{\_id:"$deptName", total\_sum:{$last:"$salary"}}}]);

[

{ \_id: 'IT', total\_sum: 50000 },

{ \_id: 'AI', total\_sum: 50000 },

{ \_id: 'comp', total\_sum: 59000 }

]

employee> db.employee\_data.aggregate([{$group:{\_id:"$deptName", total\_sum:{$push:"$salary"}}}]);

[

{ \_id: 'IT', total\_sum: [ 59000, 50000 ] },

{ \_id: 'AI', total\_sum: [ 50000, 50000 ] },

{ \_id: 'comp', total\_sum: [ 29000, 79000, 59000 ] }

] 