test> use collections

switched to db collections

collections> db.collections.insertMany([

... {name: "biryani" , category: "non-veg", price: 300 },

... {name: "Dal" , category: "veg", price: 250 },

... {name: "Rice" , category: "veg", price: 200 },

... {name: "Chicken" , category: "non-veg", price: 200 }

... ]);

{

acknowledged: true,

insertedIds: {

'0': ObjectId('67330f557450300bb40d8190'),

'1': ObjectId('67330f557450300bb40d8191'),

'2': ObjectId('67330f557450300bb40d8192'),

'3': ObjectId('67330f557450300bb40d8193')

}

}

collections> db.collections.aggregrate([

... { $group:{ ... { $group:{

... \_id:"$category" , totalcount: {$sum:1}}}]);

TypeError: db.collections.aggregrate is not a function

collections> db.collections.aggregate([ { $group: { \_id: "$category", totalcount: { $sum: 1 } } }]);

[ { \_id: 'non-veg', totalcount: 2 }, { \_id: 'veg', totalcount: 2 } ]

collections> db.collections.aggregate([ { $group: { \_id: "$category", totalcount: { $sum: "$price" } } }]);

[

{ \_id: 'veg', totalcount: 450 },

{ \_id: 'non-veg', totalcount: 500 }

]

collections> db.collections.aggregate([ { $group: { \_id: "$category", totalcount: { $avg: "$price" } } }]);

[

{ \_id: 'veg', totalcount: 225 },

{ \_id: 'non-veg', totalcount: 250 }

]

collections> .

Invalid REPL keyword

collections> use orders

switched to db orders

orders> db.orders.insertMany([

023 },

... { orderId: 1, userId: 101, orderTotal: 150, orderStatus: "shipped", year: 2 { orderId: 2, userId: 102, orderTotal: 75, orde

orderStatus: "processing", year: 2022 },

Uncaught:

SyntaxError: Unexpected token, expected "," (2:79)

1 | db.orders.insertMany([

> 2 | { orderId: 1, userId: 101, orderTotal: 150, orderStatus: "shipped", year: 2 { orderId: 2, userId: 102, orderTotal: 75, orderStatus: "processing", year: 2022 },

|

^

3 |

orders> { orderId: 3, userId: 101, orderTotal: 200, orderStatusorderStatus: "shipped", year: 2023 },

... { orderId: 4, userId: 103, orderTotal: 50, orderStatus: "shipped", year: 2023 },

... { orderId: 5, userId: 104, orderTotal: 300, orderStatus: "canceled", year: 2021 },

... { orderId: 6, userId: 101, orderTotal: 500, orderStatus: "shipped", year: 2023 }

... ]);

Uncaught:

SyntaxError: Missing semicolon. (1:22)

> 1 | { orderId: 3, userId: 101, orderTotal: 200, orderStatus: "shipped", year: 2023 },

| ^

2 | { orderId: 4, userId: 103, orderTotal: 50, orderStatus: "shipped", year: 2023 },

3 | { orderId: 5, userId: 104, orderTotal: 300, orderStatus: "canceled", year: 2021 },

4 | { orderId: 6, userId: 101, orderTotal: 500, orderStatus: "shipped", year: 2023 }

orders> db.orders.insertMany([

, orderTotal: 150, orderStatus: "shipped", year: 2023 },

... { orderId: 1, userId: 101 { orderId: 2, userId: 102, orderorderTotal: 75, orderStatus: "processing", year: 2022 },

103, orderTotal: 50, orderStatus: "shipped", year: 2023 },

Uncaught:

SyntaxError: Unexpected token, expected "," (2:29)

1 | db.orders.insertMany([

> 2 | { orderId: 1, userId: 101 { orderId: 2, userId: 102, orderTotal: 75, orderStatus: "processing", year: 2022 },

| ^

3 |

orders> { ord

... erId: 3, userId: 101, orderTotal: 200, orderStatus: "shipped", year: 2023 },

Uncaught:

SyntaxError: Missing semicolon. (2:15)

1 | { ord

> 2 | erId: 3, userId: 101, orderTotal: 200, orderStatus: "shipped", year: 2023 },

| ^

3 |

orders> { orderId: 4, userId: { orderId: 5, userId: 104, orderorderTotal: 300, orderStatus: "canceled", year: 2021 },

... { orderId: 6, userId: 101, orderTotal: 500, orderStatus: "shipped", year: 2023 }

Uncaught:

SyntaxError: Missing semicolon. (1:22)

> 1 | { orderId: 4, userId: { orderId: 5, userId: 104, orderTotal: 300, orderStatus: "canceled", year: 2021 },

| ^

2 | { orderId: 6, userId: 101, orderTotal: 500, orderStatus: "shipped", year: 2023 }

3 |

orders> ]);

Uncaught:

SyntaxError: Unexpected token (1:0)

> 1 | ]);

| ^

2 |

orders> db.orders.insertMany([ { orderId: 1, userId: 101 { orderId: 2, userId: 102, orderTotal: 75, orderStatus: "processing", year: 2022 },

Uncaught:

SyntaxError: Unexpected token, expected "," (1:50)

> 1 | db.orders.insertMany([ { orderId: 1, userId: 101 { orderId: 2, userId: 102, orderTotal: 75, orderStatus: "processing", year: 2022 },

| ^

2 |

orders> db.orders.insertMany([ { orderId: 1, userId: 101 { orderId: 2, userId: 102, orderTotal: 75, orderStatus: "processing", year: 2022 },

Uncaught:

SyntaxError: Unexpected token, expected "," (1:50)

> 1 | db.orders.insertMany([ { orderId: 1, userId: 101 { orderId: 2, userId: 102, orderTotal: 75, orderStatus: "processing", year: 2022 },

| ^

2 |

orders> db.orders.insertMany([ { orderId: 1, userId: 101 { orderId: 2, userId: 102, orderTotal: 75, orderStatus: "processing", year: 2022 },{

orders> db.orders.insertMany([ { orderId: 1, userId: 101, orderTorders> db.orders.insertMany([ { orderId: 1, userId: 101, orderTorders> db.orders.insertMany([ { orderId: 1, userId: 101, orderTotal: 150, orderStatus: "shipped", year: 2023 }, { orderId: 2, userId: 102, orderTotal: 75, orderStatus: "processing", year: 2022 }, { orderId: 3, userId: 101, orderTotal: 200, orderStatus: "shipped", year: 2023 }, { orderId: 4, userId: 103, orderTotal: 50, orderStatus: "shipped", year: 2023 }, { orderId: 5, userId: 104, orderTotal: 300, orderStatus: "canceled", year: 2021 }, { orderId: 6, userId: 101, orderTotal: 500, orderStatus: "shipped", year: 2023 }]);

{

acknowledged: true,

insertedIds: {

'0': ObjectId('673313187450300bb40d8194'),

'1': ObjectId('673313187450300bb40d8195'),

'2': ObjectId('673313187450300bb40d8196'),

'3': ObjectId('673313187450300bb40d8197'),

'4': ObjectId('673313187450300bb40d8198'),

'5': ObjectId('673313187450300bb40d8199')

}

}

orders> use sales

switched to db sales

sales> db.sales.insertMany([ { productId: 1, category: "Electronics", quantity: 10 }, { productId: 2, category: "Clothing", quantity: 20 }, { productId: 3, category: "Electronics", quantity: 5 }, { productId: 4, category: "Home Appliances", quantity: 8 }, { productId: 5, category: "Clothing", quantity: 15 }]);

{

acknowledged: true,

insertedIds: {

'0': ObjectId('673313287450300bb40d819a'),

'1': ObjectId('673313287450300bb40d819b'),

'2': ObjectId('673313287450300bb40d819c'),

'3': ObjectId('673313287450300bb40d819d'),

'4': ObjectId('673313287450300bb40d819e')

}

}

sales> use users

switched to db users

users> db.users.insertMany([ { userId: 101, name: "Alice" }, { userId: 102, name: "Bob" }, { userId: 103, name: "Charlie" }, {

userId: 104, name: "Diana" }]);

{

acknowledged: true,

insertedIds: {

'0': ObjectId('673313367450300bb40d819f'),

'1': ObjectId('673313367450300bb40d81a0'),

'2': ObjectId('673313367450300bb40d81a1'),

'3': ObjectId('673313367450300bb40d81a2')

}

}

users> use transactions

switched to db transactions

transactions> db.transactions.insertMany([ { transactionId: 1, customerId: 101, amount: 150 }, { transactionId: 2, customerId: 102, amount: 200 }, { transactionId: 3, customerId: 101, amount: 100 }, { transactionId: 4, customerId: 103, amount: 75 }, { tratransactionId: 5, customerId: 102, amount: 50 }]);

{

acknowledged: true,

insertedIds: {

'0': ObjectId('673313477450300bb40d81a3'),

'1': ObjectId('673313477450300bb40d81a4'),

'2': ObjectId('673313477450300bb40d81a5'),

'3': ObjectId('673313477450300bb40d81a6'),

'4': ObjectId('673313477450300bb40d81a7')

}

}

transactions> use employees

switched to db employees

employees> db.employees.insertMany([ { employeeId: 1, departmentemployees> db.employees.insertMany([ { employeeId: 1, departmentId: 1, name: "Eve" }, { employeeId: 2, departmentId: 2, name: "Frank" }, { employeeId: 3, departmentId: 1, name: "Grace" }]);

{

acknowledged: true,

insertedIds: {

'0': ObjectId('6733135a7450300bb40d81a8'),

'1': ObjectId('6733135a7450300bb40d81a9'),

'2': ObjectId('6733135a7450300bb40d81aa')

}

}

employees> use departments

switched to db departments

departments> db.departments.insertMany([ { departmentId: 1, depadepartments> db.departments.insertMany([ { departmentId: 1, departmentName: "Sales" }, { departmentId: 2, departmentName: "Engineering" }]);

{

acknowledged: true,

insertedIds: {

'0': ObjectId('6733136a7450300bb40d81ab'),

'1': ObjectId('6733136a7450300bb40d81ac')

}

}

departments> use products

switched to db products

products> db.products.insertMany([ { productId: 1, category: "Elproducts> db.products.insertMany([ { productId: 1, category: "Elproducts> db.products.insertMany([ { productId: 1, category: "Electronics", brand: "BrandA", sales: 1000 }, { productId: 2, category: "Electronics", brand: "BrandB", sales: 1500 }, { productId: 3, category: "Clothing", brand: "BrandA", sales: 500 }, { prodproductId: 4, category: "Clothing", brand: "BrandB", sales: 800 }]);

{

acknowledged: true,

insertedIds: {

'0': ObjectId('673313787450300bb40d81ad'),

'1': ObjectId('673313787450300bb40d81ae'),

'2': ObjectId('673313787450300bb40d81af'),

'3': ObjectId('673313787450300bb40d81b0')

}

}

products> use books

switched to db books

books> db.books.insertMany([ { bookId: 1, title: "Book One" }, { bookId: 2, title: "Book Two" }, { bookId: 3, title: "Book Thre

e" }]);

{

acknowledged: true,

insertedIds: {

'0': ObjectId('673313857450300bb40d81b1'),

'1': ObjectId('673313857450300bb40d81b2'),

'2': ObjectId('673313857450300bb40d81b3')

}

}

books> use borrowRecords

switched to db borrowRecords

borrowRecords> db.borrowRecords.insertMany([ { recordId: 1, book

borrowRecords> db.borrowRecords.insertMany([ { recordId: 1, bookId: 1 }, { recordId: 2, bookId: 1 }, { recordId: 3, bookId: 2 }, { recordId: 4, bookId: 1 }, { recordId: 5, bookId: 3 }, { rerecordId: 6, bookId: 2 }]);

{

acknowledged: true,

insertedIds: {

'0': ObjectId('673313be7450300bb40d81b4'),

'1': ObjectId('673313be7450300bb40d81b5'),

'2': ObjectId('673313be7450300bb40d81b6'),

'3': ObjectId('673313be7450300bb40d81b7'),

'4': ObjectId('673313be7450300bb40d81b8'),

'5': ObjectId('673313be7450300bb40d81b9')

}

}

borrowRecords> use reviews

switched to db reviews

reviews> db.reviews.insertMany([ { reviewId: 1, productId: 1, rareviews> db.reviews.insertMany([ { reviewId: 1, productId: 1, rareviews> db.reviews.insertMany([ { reviewId: 1, productId: 1, rating: 4 }, { reviewId: 2, productId: 1, rating: 5 }, { reviewId: 3, productId: 2, rating: 3 }, { reviewId: 4, productId: 2, ratrating: 4 }, { reviewId: 5, productId: 3, rating: 5 }]);

{

acknowledged: true,

insertedIds: {

'0': ObjectId('673313cc7450300bb40d81ba'),

'1': ObjectId('673313cc7450300bb40d81bb'),

'2': ObjectId('673313cc7450300bb40d81bc'),

'3': ObjectId('673313cc7450300bb40d81bd'),

'4': ObjectId('673313cc7450300bb40d81be')

}

}

reviews> use orders

switched to db orders

orders> db.orders.find({

... oderTotal: { $gt: 100 },

... orderStatus: "shipped"

... })

orders> db.orders.find({ oderTotal: { $gt: 100 }, orderStatus: "shipped" });

orders> db.orders.find({ orderTotal: { $gt: 100 }, orderStatus: "shipped" });

[

{

\_id: ObjectId('673313187450300bb40d8194'),

orderId: 1,

userId: 101,

orderTotal: 150,

orderStatus: 'shipped',

year: 2023

},

{

\_id: ObjectId('673313187450300bb40d8196'),

orderId: 3,

userId: 101,

orderTotal: 200,

orderStatus: 'shipped',

year: 2023

},

{

\_id: ObjectId('673313187450300bb40d8199'),

orderId: 6,

userId: 101,

orderTotal: 500,

orderStatus: 'shipped',

year: 2023

}

]

orders> db.orders.aggregate([

... { $match : { orderTotal : { $gt : 100 }, orderStatus: "shipped"}}]);

[

{

\_id: ObjectId('673313187450300bb40d8194'),

orderId: 1,

userId: 101,

orderTotal: 150,

orderStatus: 'shipped',

year: 2023

},

{

\_id: ObjectId('673313187450300bb40d8196'),

orderId: 3,

userId: 101,

orderTotal: 200,

orderStatus: 'shipped',

year: 2023

},

{

\_id: ObjectId('673313187450300bb40d8199'),

orderId: 6,

userId: 101,

orderTotal: 500,

orderStatus: 'shipped',

year: 2023

}

]

orders> use sales

switched to db sales

sales> db.sales.aggregate([

... { $group: {

... \_id: "$category",

... totalQuantity: { $sum: "$quantity" }

... }

... }

... ])

[

{ \_id: 'Home Appliances', totalQuantity: 8 },

{ \_id: 'Clothing', totalQuantity: 35 },

{ \_id: null, totalQuantity: 25 },

{ \_id: 'Electronics', totalQuantity: 15 }

]

sales> db.transactions.aggregate([

... { $group : { \_id: " $customerId" , totalspent: { $sum: "$amou

nt" }

... }

... }

... ])

sales> db.transactions.aggregate([

... {

... $group: {

... \_id : "$customerId" ,

... totalSpent: { $sum: "$amount" }

... }

... }

... ]);

sales> use transactions

switched to db transactions

transactions> db.transactions.aggregate([ { $group: { \_id: "$customerId", totalSpent: { $sum: "$amount" } } }] );

[

{ \_id: 103, totalSpent: 75 },

{ \_id: 101, totalSpent: 250 },

{ \_id: 102, totalSpent: 250 }

]

transactions> use orders

switched to db orders

orders> db.orders.aggregate([

... {

... $match : { year : 2023 }

... },

... { $group: {

... \_id: null,

... totalRevenue: { $sum: "orderTotal" }

... }

... }

... ])

[ { \_id: null, totalRevenue: 0 } ]

orders> use products

switched to db products

products> db.products.aggregate([

... {

... $group : {

... \_id: { category : "$category" , brand: "$brand" },

... totalsales: { $sum: "$sales" }

... }

... }

... ])

[

{

\_id: { category: 'Electronics', brand: 'BrandB' },

totalsales: 1500

},

{ \_id: {}, totalsales: 0 },

{ \_id: { category: 'Clothing', brand: 'BrandA' }, totalsales: 500 },

{ \_id: { category: 'Clothing', brand: 'BrandB' }, totalsales: 800 },

{

\_id: { category: 'Electronics', brand: 'BrandA' },

totalsales: 1000

}

]

products> use orders

switched to db orders

orders> db.orders.aggregate([

... {

... $match : { customerMembershipTier: "VIP" }

... },

... { $group:{

... \_id: null,

... totalRevenue: { $sum : "$orderTotal" }

... }

... }

... ])

orders> db.orders.insertMany([ { orderId: 1, customerId: 101, ororders> db.orders.insertMany([ { orderId: 1, customerId: 101, or

orders> db.orders.insertMany([ { orderId: 1, customerId: 101, ororders> db.orders.insertMany([ { orderId: 1, customerId: 101, orderTotal: 150, customerMembershipTier: "VIP" }, { orderId: 2, customerId: 102, orderTotal: 75, customerMembershipTier: "Regular" }, { orderId: 3, customerId: 101, orderTotal: 200, customerMembershipTier: "VIP" }, { orderId: 4, customerId: 103, orderTotal: 50, customerMembershipTier: "VIP" }, { orderId: 5, customerId: 10

4, orderTotal: 300, customerMembershipTier: "Regular" }]);

{

acknowledged: true,

insertedIds: {

'0': ObjectId('67331db17450300bb40d81bf'),

'1': ObjectId('67331db17450300bb40d81c0'),

'2': ObjectId('67331db17450300bb40d81c1'),

'3': ObjectId('67331db17450300bb40d81c2'),

'4': ObjectId('67331db17450300bb40d81c3')

}

otal" } } }] )

[ { \_id: null, totalRevenue: 400 } ]

orders> use reviews

switched to db reviews

reviews> db.reviews.aaggregate([

...

... {

... $group : {

... \_id: "$productId" ,

... averageRating: {$avg: "$rating" }

... }

... }

... ])

TypeError: db.reviews.aaggregate is not a function

reviews> db.reviews.aggregate([ { $group: { \_id: "$productId", averageRating: { $avg: "$rating" } } }] )

[

{ \_id: 1, averageRating: 4.5 },

{ \_id: 2, averageRating: 3.5 },

{ \_id: 3, averageRating: 5 }

]

reviews>