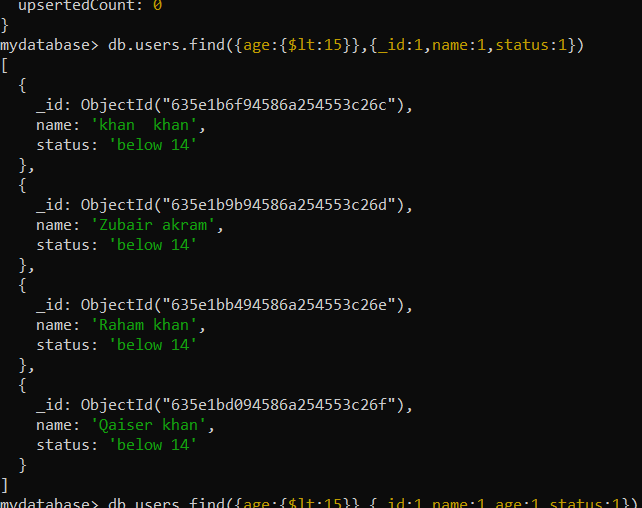
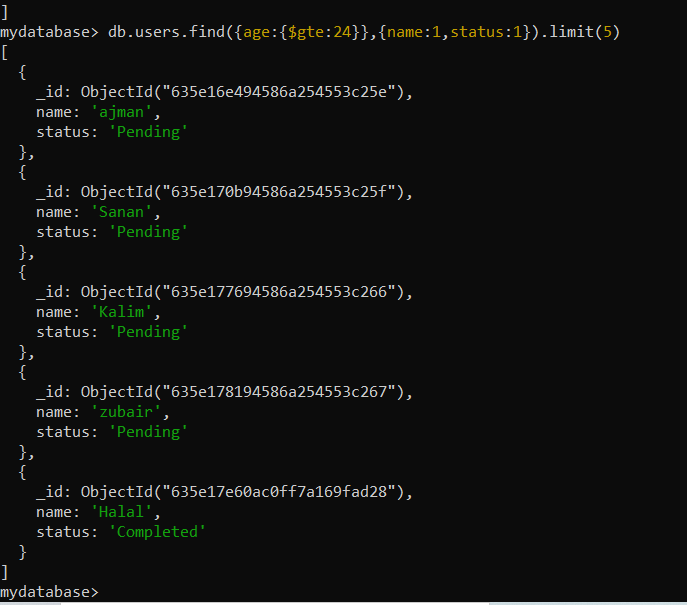
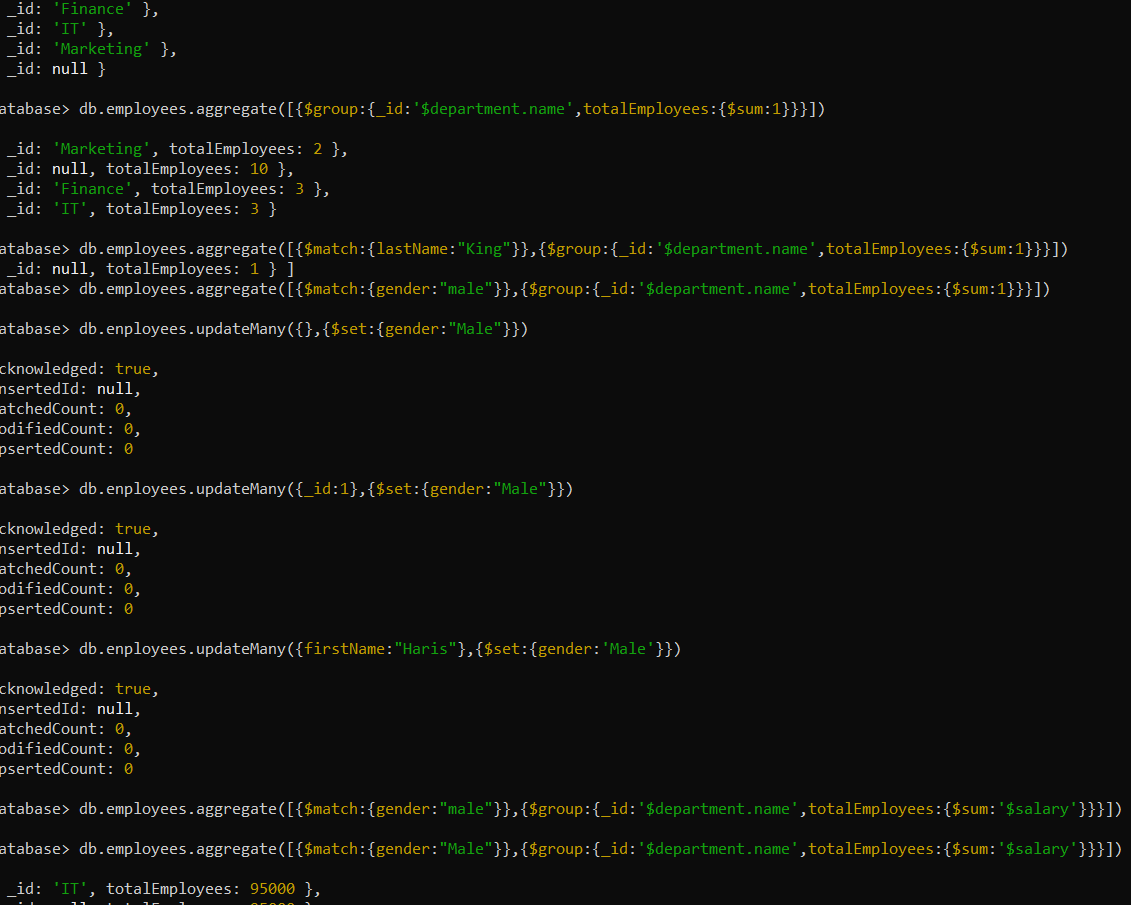
MongoDB Server Advanced 11



Coding part:

try {

db.pizzas.bulkWrite( [

{ insertOne: { document: { \_id: 3, type: "beef", size: "medium", price: 6 } } },

{ insertOne: { document: { \_id: 4, type: "sausage", size: "large", price: 10 } } },

{ updateOne: {

filter: { type: "cheese" },

update: { $set: { price: 8 } }

} },

{ deleteOne: { filter: { type: "pepperoni"} } },

{ replaceOne: {

filter: { type: "vegan" },

replacement: { type: "tofu", size: "small", price: 4 }

} }

] )

} catch( error ) {

print( error )

}



try{db.pizzas.bulkWrite([{insertOne:{document:{\_id:3,type:"beef",size:"medium",price:6}}},{insertOne:{document:{\_id:4:type:"sausage",size:"large",price:10}}},{updateOne:{filter:{type:"cheese"},update:{$set:{price:8}}}},{deleteOne:{filter:{type:"pepperoni"}}},{replaceOne:{filter:{type:"vegan"},replacement:{type:"tofu",size:"small",price:4}}}])} catch(error){print(error)}

Uncaught:

SyntaxError: Unexpected token, expected "," (1:117)

> 1 | try{db.pizzas.bulkWrite([{insertOne:{document:{\_id:3,type:"beef",size:"medium",price:6}}},{insertOne:{document:{\_id:4:type:"sausage",size:"large",price:10}}},{updateOne:{filter:{type:"cheese"},update:{$set:{price:8}}}},{deleteOne:{filter:{type:"pepperoni"}}},{replaceOne:{filter:{type:"vegan"},replacement:{type:"tofu",size:"small",price:4}}}])} catch(error){print(error)}

| ^

2 |

mydatabase> try{db.pizzas.bulkWrite([{insertOne:{document:{\_id:3,type:"beef",size:"medium",price:6}}},{insertOne:{document:{\_id:4:type:"sausage",size:"large",price:10}}},{updateOne:{filter:{type:"cheese"},update:{$set:{price:8}}},{deleteOne:{filter:{type:"pepperoni"}}},{replaceOne:{filter:{type:"vegan"},replacement:{type:"tofu",size:"small",price:4}}}])} catch(error){print(error)}

Uncaught:

SyntaxError: Unexpected token, expected "," (1:117)

> 1 | try{db.pizzas.bulkWrite([{insertOne:{document:{\_id:3,type:"beef",size:"medium",price:6}}},{insertOne:{document:{\_id:4:type:"sausage",size:"large",price:10}}},{updateOne:{filter:{type:"cheese"},update:{$set:{price:8}}},{deleteOne:{filter:{type:"pepperoni"}}},{replaceOne:{filter:{type:"vegan"},replacement:{type:"tofu",size:"small",price:4}}}])} catch(error){print(error)}

| ^

2 |

mydatabase> try{db.pizzas.bulkWrite([{insertOne:{document:{\_id:3,type:"beef",size:"medium",price:6}}},{insertOne:{document:{\_id:4:type:"sausage",size:"large",price:10}}},{updateOne:{filter:{type:"cheese"},update:{$set:{price:8}}}},{deleteOne:{filter:{type:"pepperoni"}}},{replaceOne:{filter:{type:"vegan"},replacement:{type:"tofu",size: "small", price:4}}}]) catch(error){print(error)}

Uncaught:

SyntaxError: Unexpected token, expected "," (1:117)

> 1 | try{db.pizzas.bulkWrite([{insertOne:{document:{\_id:3,type:"beef",size:"medium",price:6}}},{insertOne:{document:{\_id:4:type:"sausage",size:"large",price:10}}},{updateOne:{filter:{type:"cheese"},update:{$set:{price:8}}}},{deleteOne:{filter:{type:"pepperoni"}}},{replaceOne:{filter:{type:"vegan"},replacement:{type:"tofu",size: "small", price:4}}}]) catch(error){print(error)}

| ^

2 |

mydatabase> try{db.pizzas.bulkWrite([{insertOne:{document:{\_id:3,type:"beef",size:"medium",price:6}}},{insertOne:{document:{\_id:4:type:"sausage",size:"large",price:10}}},{updateOne:{filter:{type:"cheese"},update:{$set:{price:8}}}},{deleteOne:{filter:{type:"pepperoni"}}},{replaceOne:{filter:{type:"vegan"},replacement:{type:"tofu",size:"small", price:4}}}])} catch (error) {print'error'}

Uncaught:

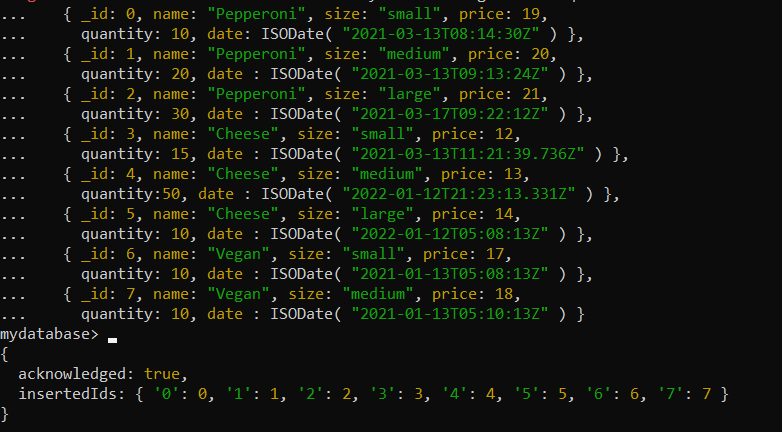
SyntaxError: Unexpected token, expected "," (1:117)

> 1 | try{db.pizzas.bulkWrite([{insertOne:{document:{\_id:3,type:"beef",size:"medium",price:6}}},{insertOne:{document:{\_id:4:type:"sausage",size:"large",price:10}}},{updateOne:{filter:{type:"cheese"},update:{$set:{price:8}}}},{deleteOne:{filter:{type:"pepperoni"}}},{replaceOne:{filter:{type:"vegan"},replacement:{type:"tofu",size:"small", price:4}}}])} catch (error) {print'error'}

| ^

2 |

mydatabase> try{db.pizzas.bulkWrite([{insertOne:{document:{\_id:3,type:"beef",size:"medium",price:6}}},{insertOne:{document:{\_id:4:type:"sausage",size:"large",price:10}}},{updateOne:{filter:{type:"cheese"},update:{$set:{price:8}}}},{deleteOne:{filter:{type:"pepperoni"}}},{replaceOne:{filter:{type:"vegan"},replacement:{type:"tofu",size:"small", price:4}}}])} catch (error) {print(error)}



db.orders.insertMany( [

{ \_id: 0, name: "Pepperoni", size: "small", price: 19,

quantity: 10, date: ISODate( "2021-03-13T08:14:30Z" ) },

{ \_id: 1, name: "Pepperoni", size: "medium", price: 20,

quantity: 20, date : ISODate( "2021-03-13T09:13:24Z" ) },

{ \_id: 2, name: "Pepperoni", size: "large", price: 21,

quantity: 30, date : ISODate( "2021-03-17T09:22:12Z" ) },

{ \_id: 3, name: "Cheese", size: "small", price: 12,

quantity: 15, date : ISODate( "2021-03-13T11:21:39.736Z" ) },

{ \_id: 4, name: "Cheese", size: "medium", price: 13,

quantity:50, date : ISODate( "2022-01-12T21:23:13.331Z" ) },

{ \_id: 5, name: "Cheese", size: "large", price: 14,

quantity: 10, date : ISODate( "2022-01-12T05:08:13Z" ) },

{ \_id: 6, name: "Vegan", size: "small", price: 17,

quantity: 10, date : ISODate( "2021-01-13T05:08:13Z" ) },

{ \_id: 7, name: "Vegan", size: "medium", price: 18,

quantity: 10, date : ISODate( "2021-01-13T05:10:13Z" ) }

] )



db.orders.aggregate([ /\* Stage 1: Filter pizza order documents by date range\*/ { $match: { "date": { $gte: new ISODate("2020-01-30"), $lt: new ISODate("2022-01-30") } } }, /\* Stage 2: Group remaining documents by date and calculate results\*/ { $group: { \_id: { $dateToString: { format: "%Y-%m-%d", date: "$date" } }, totalOrderValue: { $sum: { $multiply: ["$price", "$quantity"] } }, averageOrderQuantity: { $avg: "$quantity" } } }, /\* Stage 3: Sort documents by totalOrderValue in descending order\*/ { $sort: { totalOrderValue: -1 } }])

