1. Find the total revenue (price × quantity) for each item, sorted from highest to lowest.

db.sales.aggregate({$group:{\_id:'$item',total:{$sum:{$multiply:['$price','$quantity']}}}},{$sort:{total:-1}})

2.//2. Calculate the total quantity sold per month in 2022.

db.sales.aggregate([{$match:{date:{$gte:ISODate("2022-01-01"),$lt:ISODate("2023-01-01")}}},{$group:{\_id:{$month:'$date'},total:{$sum:'$quantity'}}},{$sort:{\_id:1}}])

//3. Find all items where price is greater than 10 and size is not 'Short'.

db.sales.find({price:{$gt:10},size:{$ne:'Short'}})

//4. Get all Cappuccino sales with quantity between 10 and 20.

db.sales.find({quantity:{$gte:10,$lte:20}})

//5. Query to find items where the item name starts with "A".

db.sales.find({item:{$regex:/^A/}})

//6. Find all records that do not have the field size.

db.sales.find({size:{$exists:false}})

//7. Find all sales that are either "Grande" or "Tall" but not "Americanos".

db.sales.find({item:{$ne:'Americanos'},size:{$in:['Grande','Tall']}})

//8. List all items sold in February 2022.

db.sales.find({date: {

$gte: ISODate("2022-02-01T00:00:00Z"),

$lt: ISODate("2022-03-01T00:00:00Z")

}})

//9. Find sales where the quantity is more than twice the price.

db.sales.aggregate([{$match:{$expr: { $gt: ["$quantity", { $multiply: [2, "$price"] }] }

}

}

])

//11. Filter sales where the total revenue is even and exceeds 100.

db.sales.find({

$where: function() {

const total = this.price \* this.quantity;

return total > 100 && total % 2 === 0;

}

})

//11. Find Sales Where the Day of Week Matches Quantity's Last Digit

//[Filter sales where the day of the week (0=Sunday, 1=Monday, etc.) matches the last digit of quantity]

db.sales.aggregate([{$addFields:{day:{$mod:[{$subtract:[{$dayOfWeek:"$date"},1]},10]},lastDigit:{$mod:["$quantity",10]}}},{$match:{$expr:{$eq:["$day","$lastDigit"]}}}])

//12. Find Sales Where the Month is Prime and Quantity is Odd

//[Filter sales where the month (1-12) is a prime number (2,3,5,7,11) AND quantity is odd]

db.sales.find({$where: function() {const month = this.date.getMonth() + 1; const quantity = this.quantity; const isMonthPrime = (month === 2 || month === 3 || month === 5 || month === 7 || month === 11); const isQuantityOdd = (quantity % 2 !== 0); return isMonthPrime && isQuantityOdd;}});

//13. Find Sales with "Suspicious Quantities" (Divisible by 5 or 7)

//[Filter sales where quantity is divisible by 5 or 7]

db.sales.aggregate([{$match:{$expr:{$or:[{$eq:[{$mod:["$quantity",5]},0]},{$eq:[{$mod:["$quantity",7]},0]}]}}}])