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

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

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

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

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

Ans)db.sales.find({$and:[{price:{$gt:10}},{size:{$ne:"Short"}}]})

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

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

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

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

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

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

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

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

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

Ans)db.sales.find({date: {$gte: ISODate("2022-02-01"), $lt: ISODate("2022-03-01")}},{name:1,\_id:1});

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

Ans)db.sales.find({$where: "this.quantity > 2 \* this.price"})

**10. Find all sales where the price is greater than the average price of their respective size.**

Ans)db.sales.aggregate([{$setWindowFields: {partitionBy: "$size",output: {avgPrice: { $avg: "$price" }}}},{$match: {$expr: { $gt: ["$price", "$avgPrice"] }}}]);

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

Ans)db.sales.find({$where: function() {const dayOfWeek = this.date.getDay();const lastDigit = this.quantity % 10;return dayOfWeek === lastDigit;}});

1. **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]**

Ans)db.sales.find({$where: function () {

const month = this.date.getMonth() + 1;

const isPrimeMonth = [2, 3, 5, 7, 11].includes(month);

const isOddQuantity = this.quantity % 2 === 1;

return isPrimeMonth && isOddQuantity;

  }

});

**13. Find Sales with "Suspicious Quantities" (Divisible by 5 or 7) [Filter sales where quantity is divisible by 5 or 7]**

Ans)db.sales.find({

$where: "this.quantity % 5 === 0 || this.quantity % 7 === 0"

});