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

Ans:  
[

{

"$project": {

"item": 1,

"revenue": { "$multiply": ["$price", "$quantity"] }

}

},

{

"$group": {

"\_id": "$item",

"totalRevenue": { "$sum": "$revenue" }

}

},

{

"$sort": { "totalRevenue": -1 }

}

]

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

Ans:

[

{

"$match": {

"date": {

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

"$lt": ISODate("2023-01-01T00:00:00Z")

}

}

},

{

"$group": {

"\_id": {

"month": { "$month": "$date" }

},

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

}

},

{

"$sort": { "\_id.month": 1 }

}

]

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

Ans:

{

"price": { "$gt": 10 },

"size": { "$ne": "Short" }

}

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

Ans:

{

"item": "Cappuccino",

"quantity": {

"$gte": 10,

"$lte": 20

}

}

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

Ans:

{

"item": { "$regex": "^A" }

}

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

Ans:  
{

"size": { "$exists": false }

}

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

Ans:

db.sales.find({

$and: [

{ size: { $in: ["Grande", "Tall"] } },

{ item: { $ne: "Americanos" } }

]

})

8.List all items sold in February 2022.

Ans:

db.sales.find({

date: {

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

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

}

}, {

item: 1,

\_id: 0

})

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

Ans:

db.sales.find({

$where: function() {

return 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",

sortBy: { \_id: 1 },

output: {

avgPriceForSize: { $avg: "$price", window: { documents: ["unbounded", "unbounded"] } }

}

}

},

{

$match: {

$expr: { $gt: ["$price", "$avgPriceForSize"] }

}

}

])

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]

Ans:

})db.sales.find({

$where: function () {

const dayOfWeek = this.date.getDay();

const lastDigit = this.quantity % 10;

return dayOfWeek === 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]

Ans:

db.sales.find({

$where: function () {

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

const primeMonths = [2, 3, 5, 7, 11];

return primeMonths.includes(month) && this.quantity % 2 === 1;

}

});

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({

$expr: {

$or: [

{ $eq: [{ $mod: ["$quantity", 5] }, 0] },

{ $eq: [{ $mod: ["$quantity", 7] }, 0] }

]

}

});