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

Ans:

db.sales.aggregate([

{

$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.

db.sales.aggregate([

{

$match: {

date: {

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

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

}

}

},

{

$group: {

\_id: { $month: "$date" },

totalQuantity: { $sum: "$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({

item: "Cappuccino",

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) 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")

}

})

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

db.sales.find({

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

item: { $ne: "Americanos" }

})

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

db.sales.find({

$expr: { $gt: ["$quantity", { $multiply: [2, "$price"] }] }

})

10) Find sales where the quantity is more than twice the price.

db.sales.aggregate([

{

$group: {

\_id: "$size",

avgPrice: { $avg: "$price" }

}

},

{

$lookup: {

from: "sales",

localField: "\_id",

foreignField: "size",

as: "sales\_docs"

}

},

{ $unwind: "$sales\_docs" },

{

$match: {

$expr: { $gt: ["$sales\_docs.price", "$avgPrice"] }

}

},

{

$replaceRoot: { newRoot: "$sales\_docs" }

}

])

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

db.sales.find({

$where: function() {

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

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

}

})

12) 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: {

dayOfWeek: { $dayOfWeek: "$date" },

lastDigitQuantity: { $mod: ["$quantity", 10] }

}

},

{

$match: {

$expr: {

$eq: [{ $subtract: ["$dayOfWeek", 1] }, "$lastDigitQuantity"]

}

}

}

])

13) . 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.aggregate([

{

$addFields: {

month: { $month: "$date" }

}

},

{

$match: {

month: { $in: [2, 3, 5, 7, 11] },

$expr: { $eq: [{ $mod: ["$quantity", 2] }, 1] }

}

}

])

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

db.sales.find({

$or: [

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

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

]

})