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

Query:

db.sales.aggregate([

{

$group: {

\_id: "$item",

totalRevenue: { $sum: { $multiply: ["$price", "$quantity"] } }

}

},

{ $sort: { totalRevenue: -1 } }

])

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

Query:

db.sales.aggregate([ {

$match: {

date: {

$gte: ISODate("2022-01-01T00:00:00Z"),$lt: ISODate("2023-01-01T00:00:00Z") }}},

{

$group: {

\_id: {

year: { $year: "$date" },

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

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

{

$sort: { "\_id.year": 1, "\_id.month": 1 } }])

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

Query:

db.sales.find({

price: { $gt: 10 },

size: { $ne: "Short" }

})

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

Query:

db.sales.find({

item: "Cappuccino",

quantity: { $gte: 10, $lte: 20 }

})

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

Query:

db.sales.find({

item: { $regex: /^A/, $options: 'i' }

})

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

Query:

db.sales.find({

size: { $exists: false }

})

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

Query:

db.sales.find({

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

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

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

Query:

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.**

Query:

db.sales.find({

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

})

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

Query:

db.sales.aggregate([

{

$group: {

\_id: "$size",

avgPrice: { $avg: "$price" }} },

{

$lookup: {

from: "sales",

localField: "\_id",

foreignField: "size",

as: "salesBySize"

} },

{ $unwind: "$salesBySize" },

{

$match: {

$expr: {

$gt: ["$salesBySize.price", "$avgPrice"] } } },

{

$replaceRoot: { newRoot: "$salesBySize" }

  }

]); {

$setWindowFields: {

partitionBy: "$size",

sortBy: { \_id: 1 },

output: {

avgPricePerSize: { $avg: "$price" } } } },

{

$match: {

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

])

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

Query:

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]

Query:

db.sales.aggregate([

{

$addFields: {

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

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

{

$match: {

$expr: {

$eq: ["$dayOfWeek", { $add: ["$lastDigit", 1] }] }} }

]);

**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]

Query:

db.sales.aggregate([

{

$addFields: {

saleMonth: { $month: "$date" }

}

},

{

$match: {

$expr: {

$and: [

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

{ $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]

Query:

db.sales.find({

$expr: {

$or: [

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

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

]

}

})