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

**A)** 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.**

A) db.sales.aggregate([

{

$group: {

\_id: { $dateToString: { format: "%Y-%m", date: "$date" } },

totalQuantity: { $sum: "$quantity" }

}

},

{

$sort: { \_id: 1 }

}

]);

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

A) db.sales.find({

price: { $gt: 10 },

size: { $ne: "Short" }

});

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

**A)**

db.sales.find({

item: "Cappuccino",

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

});

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

**A)**

db.sales.find({

item: { $regex: /^A/ }

});

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

A)

db.sales.find({

size: { $exists: false }

});

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

A)

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".**

**A)**

db.sales.find({

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

item: { $ne: "Americanos" }

});

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

**A)**

db.sales.aggregate([

{

$addFields: {

doublePrice: { $multiply: [2, "$price"] }

}

},

{

$match: {

$expr: { $gt: ["$quantity", "$doublePrice"] }

}

}

]);

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

**A)**

db.sales.aggregate([

{ $group: { \_id: "$size", avgPrice: { $avg: "$price" } } },

{ $lookup: { from: "sales", localField: "\_id", foreignField: "size", as: "sales" } },

{ $unwind: "$sales" },

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

{ $project: { \_id: "$sales.\_id", item: "$sales.item", price: "$sales.price", size: "$sales.size", quantity: "$sales.quantity", date: "$sales.date" } }

]);

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

A)

db.sales.aggregate([{

$addFields: {

dayOfWeek: { $subtract: [{ $dayOfWeek: "$date" }, 1] },

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

}

},

{

$match: {

$expr: {

$eq: ["$dayOfWeek", "$lastDigitOfQuantity"]

}

}

},

{

$project: {

dayOfWeek: 0,

lastDigitOfQuantity: 0

}

}

])

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

**A)**

db.sales.aggregate([

{

$addFields: {

saleMonth: { $month: "$date" }

}

},

{

$match: {

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

quantity: { $mod: [ "$quantity", 2 ] },

quantity: { $ne: 0 }

}

},

{

$project: {

saleMonth: 0

}

}

])

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

**A)**

db.sales.find({

$or: [

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

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

]

});