**MONGO-DB - TEST**

NAME - AKSHIV AMRESH SAXENA

REGISTRATION NUMBER - 23BCE10785

horizontal line

# **Q1)Find the total revenue (price × quantity) for each item, sorted from highest to lowes**t.

# db.sales.aggregate([{$project: {item: 1,totalRevenue: { $multiply: ["$price", "$quantity"] }} },{$sort: { totalRevenue: -1 }}])

# **Q2)Calculate the total quantity sold per month in 2022.**

# db.sales.aggregate([{$match: {date: {$gte: new ISODate("2022-01-01"),$lt: new ISODate("2023-01-01")} } },{$group: {\_id: { month: { $month: "$date" } },totalQuantity: { $sum: "$quantity" }} },{ $sort: { "\_id.month": 1 } }])

# **Q3)Find all items where price is greater than 10 and size is not 'Short'**.

db.sales.find({

price: { $gt: 10 },

size: { $ne: "Short" }

})

**Q4). Get all Cappuccino sales with quantity between 10 and 20.**

db.sales.find({

item: "Cappuccino",

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

})

**Q5)Query to find items where the item name starts with "A"**

db.sales.find({item: { $regex: /^A/, $options: "i" }})

# **Q6)Find all records that do not have the field size.**

db.sales.find({

size: { $exists: false }

})

# **Q7)Find all sales that are either "Grande" or "Tall" but not "Americanos".**

db.sales.find({

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

item: { $ne: "Americanos" }

})

# **Q8)List all items sold in February 2022.**

db.sales.find({date: {

$gte: new ISODate("2022-02-01"),

$lt: new ISODate("2022-03-01")}})

# **Q9)Find sales where the quantity is more than twice the price**

db.sales.aggregate([{$match: {$expr: { $gt: ["$quantity", { $multiply: ["$price", 2] }] }}}])

# **Q10) Find all sales where the price is greater than the average price of their respective size.**

db.sales.aggregate([{ $group: { \_id: "$size", avgPrice: { $avg: "$price" } } }, { $lookup: { from: "sales", localField: "\_id", foreignField: "size", as: "salesWithSize" } }, { $unwind: "$salesWithSize" }, { $match: { $expr: { $gt: ["$salesWithSize.price", "$avgPrice"] } } }, { $project: { \_id: "$salesWithSize.\_id", item: "$salesWithSize.item", price: "$salesWithSize.price", size: "$salesWithSize.size", quantity: "$salesWithSize.quantity", date: "$salesWithSize.date", avgPriceForSize: { $round: ["$avgPrice", 2] } } }])

# **Q11) 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: { $mod: [{ $subtract: [{ $dayOfWeek: "$saleDate" }, 1] }, 7]},lastDigit: { $mod: ["$quantity", 10] } // Last digit of quantity}},{$match: {$expr: { $eq: ["$dayOfWeek", "$lastDigit"] }}},{$project: {\_id: 0,saleDate: 1,quantity: 1,dayOfWeek: 1,lastDigit: 1}}])

# **Q12) 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: { saleMonth: { $month: "$saleDate" }, isQuantityOdd: { $eq: [{ $mod: ["$quantity", 2] }, 1] } } }, { $match: { saleMonth: { $in: [2, 3, 5, 7, 11] }, isQuantityOdd: true } }, { $project: { \_id: 0, saleDate: 1, quantity: 1, saleMonth: 1 } }])

# **Q13) Find Sales with "Suspicious Quantities" (Divisible by 5 or 7)[Filter sales where quantity is divisible by 5 or 7]**

db.sales.aggregate([{ $match: { $or: [ { $expr: { $eq: [ { $mod: ["$quantity", 5] }, 0 ] } }, { $expr: { $eq: [ { $mod: ["$quantity", 7] }, 0 ] } } ] } }, { $project: { \_id: 0, saleDate: 1, quantity: 1 } }])