MONGODB TEST

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

**Sol:-** db.sales.aggregate([

{

$project: {

item: 1,

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

}

},

{

$group: {

\_id: "$item",

totalRevenue: { $sum: "$revenue" }

}

},

{

$sort: {

totalRevenue: -1

}

}

])

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

**Sol:-** db.sales.aggregate([

{

$group: {

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

totalQuantity: { $sum: "$quantity" }

}

},

{

$sort: { \_id: 1 }

}])

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

**Sol:-** db.sales.find({

price: { $gt: 10 },

size: { $ne: "Short" }

})

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

**Sol:-** db.sales.find({

item: "Cappuccino",

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

})

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

**Sol:-** db.sales.find({

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

})

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

**Sol:-** db.sales.find({

size: { $exists: false }

})

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

**Sol:-** 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".**

**Sol:-** db.sales.find({

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

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

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

**Sol:-** db.sales.aggregate([

{

$match: {

$expr: {

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

}

}

}

])

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

**Sol:-** db.sales.aggregate([

{

$group: {

\_id: "$size",

avgPrice: { $avg: "$price" }

}

},

{

$lookup: {

from: "sales",

localField: "\_id",

foreignField: "size",

as: "salesList"

}

},

{ $unwind: "$salesList" },

{

$match: {

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

}

},

{

$project: {

\_id: 0,

item: "$salesList.item",

size: "$salesList.size",

price: "$salesList.price",

avgPrice: 1

}

}

])

**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]**

**Sol:-** db.sales.find({

$where: function() {

var dayOfWeek = this.date.getDay();

var quantityStr = this.quantity.toString();

var lastDigit = parseInt(quantityStr.charAt(quantityStr.length - 1));

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]**

Sol:- db.sales.find({

$expr: {

$and: [

{ $in: [{ $month: "$date" }, [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]**

**Sol:-** db.sales.find({

$expr: {

$or: [

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

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

]

}

})