MONGODB EXAM

1. Find the total revenue (price × quantity) for each item, sorted from highest to lowest.
2. db.sales.aggregate([{$match:{$expr:{$eq:[{$year:"$date"},2022]}}},{$group:{\_id:{month:{$month:"$date"}},totalQty:{$sum:"$quantity"}}},{$sort:{"\_id.month":1}}]);
3. Calculate the total quantity sold per month in 2022.
4. db.sales.aggregate([{$match:{$expr:{$eq:[{$year:"$date"},2022]}}},{$group:{\_id:{month:{$month:"$date"}},totalQty:{$sum:"$quantity"}}},{$sort:{"\_id.month":1}}]);
5. Find all items where price is greater than 10 and size is not 'Short'.
6. db.sales.find({price:{$gt:10}, size:{$ne:"Short"}});
7. 4. Get all Cappuccino sales with quantity between 10 and 20.
8. db.sales.find({item:"Cappuccino", quantity:{$gte:10, $lte:20}});
9. Query to find items where the item name starts with "A".
10. db.sales.find({item: /^A/});

1. Find all records that do not have the field size.
2. db.sales.find({size: { $exists: false }});
3. List all items sold in February 2022.
4. db.sales.distinct("item",{date:{$gte:ISODate("2022-02-01T00:00:00Z"),$lt:ISODate("2022-03-01T00:00:00Z")}});
5. Find all sales that are either "Grande" or "Tall" but not "Americanos".
6. db.sales.find({size: {$in: ["Grande", "Tall"]}, item: {$ne: "Americanos"}});
7. Find sales where the quantity is more than twice the price.
8. db.sales.find({ $expr: { $gt: ["$quantity", { $multiply: [2, "$price"] }] } });
9. Find all sales where the price is greater than the average price of their respective size.
10. db.sales.aggregate([{$group:{\_id:"$size",avgPrice:{$avg:"$price"}}},{$lookup:{from:"sales",localField:"\_id",foreignField:"size",as:"salesWithSize"}},{$unwind:"$salesWithSize"},{$replaceRoot:{newRoot:{$mergeObjects:["$salesWithSize",{avgPrice:"$avgPrice"}]}}},{$match:{$expr:{$gt:["$price","$avgPrice"]}}}]);
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]
12. db.sales.find({

$where: function () {

const dayOfWeek = this.date.getDay();

const lastDigit = this.quantity % 10;

return dayOfWeek === lastDigit;

}

});

1. 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]
2. db.sales.find({

$where: function () {

const month = this.date.getMonth() + 1;

const isPrimeMonth = [2, 3, 5, 7, 11].includes(month);

return isPrimeMonth && this.quantity % 2 === 1;

}

});

1. Find Sales with "Suspicious Quantities" (Divisible by 5 or 7) [Filter sales where quantity is divisible by 5 or 7]
2. db.sales.find({

$where: function () {

return this.quantity % 5 === 0 || this.quantity % 7 === 0;

}

});