Questions and answers

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

db.sales.aggregate([{

$group:{

\_id: "$item",

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

}

},{

$sort:{

totalRevenue: -1

}

}])

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

totalQuantity: {$sum : '$quantity'}

}

},

{

$project:{

\_id: 0,

month: "$\_id",

quantity: "$totalQuantity"

}

},

])

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

db.sales.find({

"price": {$gt: 10},

"size": {$ne: "Short"}

},{

item:1, size: 1, quantity:1,

})

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

db.sales.find({

item: "Cappuccino",

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

});

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

db.sales.find({ "item": /^A/ },{

item: 1, price: 1, size: 1});

1. Find all records that do not have the field size.

db.sales.find({

size: {$exists: false}

})

1. List all items sold in February 2022

db.sales.find({

date: {$gte: new ISODate("2022-02-01"), $lt: new ISODate("2022-03-01")}

})

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

db.sales.find({

item: {$ne: "Americanos"},

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

})

1. Find sales where the quantity is more than twice the price.

db.sales.find({

$expr: {

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

}

});

1. Find all sales where the price is greater than the average price of their respective size.
2. Filter sales where the total revenue is even and exceeds 100.

db.sales.find({

$expr: {

$and: [ { $gt: [{ $multiply: ["$price", "$quantity"] }, 100] },

{ $eq: [{ $mod: [{ $multiply: ["$price", "$quantity"] }, 2] }, 0] }]}

});

1. 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.find({

$where: function(){

const day = this.date.getDay();

const lastDigit = this.quantity%10;

return day === 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]

db.sales.find({

$where : function(){

const month = this.date.getMonth();

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

const isQuantityOdd = this.quantity %2 !==0;

return isPrimeMonth && isQuantityOdd;

}

})

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

db.sales.find({

$expr: {

$or: [

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

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

]

}

})