K SAI RAXITA

22BCE8965

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

CODE:

db.sales.aggregate([

{

$group: {

\_id: "$item",

totalRevenue: {

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

}

}

},

{

$sort: { totalRevenue: -1 }

}

])

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

CODE:

db.sales.aggregate([

{

$match: {

date: {

$gte: ISODate("2022-01-01T00:00:00Z"),

$lt: ISODate("2023-01-01T00:00:00Z")

}

}

},

{

$group: {

\_id: {

year: { $year: "$date" },

month: { $month: "$date" }

},

totalQuantity: { $sum: "$quantity" }

}

},

{

$sort: { "\_id.year": 1, "\_id.month": 1 }

}

]);

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

CODE:

db.sales.find({

price:{$gt:10},

size:{$ne:'short'}})

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

CODE:

db.sales.find({

item: "Cappuccino",

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

})

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

CODE:

db.sales.find({

item: { $regex: /^A/ }

})

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

CODE:

db.sales.find({

size: { $exists: false }

})

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

CODE:

db.sales.find({

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

item: { $ne: "Americanos" }

})

1. List all items sold in February 2022.

CODE:

db.sales.find(

{ date: { $gte: ISODate("2022-02-01T00:00:00Z"), $lt: ISODate("2022-03-01T00:00:00Z") } },

{ item: 1, \_id: 0 }

)

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

CODE:

db.sales.find({

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

})

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

CODE:

db.sales.aggregate([

{

$setWindowFields: {

partitionBy: "$size",

output: {

avgPrice: { $avg: "$price" }

}

}

},

{

$match: {

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

}

}

])

1. Filter sales where the total revenue is even and exceeds 100.

CODE:

db.sales.find({ $where: function() { const total = this.price \* this.quantity; return total > 100 && total % 2 === 0; } })

1. Find Sales Where the Month is Prime and Quantity is Odd

CODE:

db.sales.find({

$expr: {

$eq: [

{ $dayOfWeek: "$date" },

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

]}})

13: 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]

CODE:

db.sales.aggregate([

{

$addFields: {

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

{$match: {$expr: {$and: [

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

{ $eq: [ { $mod: ["$quantity", 2] }, 1 ] }]}}}]);

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

CODE:

db.sales.find({

$expr: {

$or: [

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

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

]

}

})