**Name: Chinnari Krishna Madhav**

**Reg no: 22BCE2121**

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([

{

$match: {

date: {

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

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

}

}

},

{

$group: {

\_id: {

month: { $month: "$date" }

},

totalQuantity: { $sum: "$quantity" }

}

},

{

$sort: { "\_id.month": 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 }

});

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

Sol:

db.sales.find({

item: { $regex: /^A/ }

});

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

Sol:

db.sales.find({

size: { $exists: false }

});

1. List all items sold in February 2022.

Sol:

db.sales.find({

$expr: {

$and: [

{ $eq: [{ $month: "$date" }, 2] },

{ $eq: [{ $year: "$date" }, 2022] }

]

}

})

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

Sol:

db.sales.find({

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

item: { $ne: "Americanos" }

})

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

Sol:

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.

Sol:

db.sales.aggregate([

{

$setWindowFields: {

partitionBy: "$size",

sortBy: { \_id: 1 },

output: {

avgPriceForSize: {

$avg: "$price",

window: { documents: ["unbounded", "unbounded"] }

}

}

}

},

{

$match: {

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

}

},

{

$project: {

item: 1,

size: 1,

price: 1,

avgPriceForSize: 1

}

}

])

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]

Sol:

db.sales.find({

$where: function () {

const dayOfWeek = this.date.getUTCDay();

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]

Sol:

db.sales.aggregate([

{

$addFields: {

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

isOdd: { $mod: ["$quantity", 2] }

}

},

{

$match: {

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

isOdd: 1

}

}

]);

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

Sol:

db.sales.find({

$or: [

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

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

]

});