**Raj Agrawal**

**22BCE2184**

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: { $dateToString: { format: "%Y-%m", date: "$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 }

})

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

Sol:

db.sales.find({

$and: [

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

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

]

})

1. List all items sold in February 2022.

Sol:

db.sales.aggregate([

{

$addFields: {

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

saleYear: { $year: "$date" }

}

},

{

$match: {

saleMonth: 2,

saleYear: 2022

}

},

{

$project: {

saleMonth: 0,

saleYear: 0

}

}

])

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

Sol:

db.sales.aggregate([

{

$match: {

$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([

// Lookup average price per size

{

$group: {

\_id: "$size",

avgPrice: { $avg: "$price" }

}

},

{

$merge: {

into: "avgPricesPerSize",

whenMatched: "replace",

whenNotMatched: "insert"

}

}

])

// Main query

db.sales.aggregate([

{

$lookup: {

from: "avgPricesPerSize",

localField: "size",

foreignField: "\_id",

as: "avgData"

}

},

{ $unwind: "$avgData" },

{

$match: {

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

}

},

{

$project: {

\_id: 1,

item: 1,

size: 1,

price: 1,

avgPriceForSize: "$avgData.avgPrice"

}

}

])

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 day = this.date.getUTCDay(); // 0 = Sunday

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]

Sol:

db.sales.aggregate([

{

$addFields: {

month: { $month: "$date" }

}

},

{

$match: {

$expr: {

$and: [

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

{ $eq: [{ $mod: ["$quantity", 2] }, 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({

$expr: {

$or: [

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

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

]

}

})