//RegId:23BCE9489  
//Name:Shaik Arshad  
db.sales.insertMany([

{ "\_id" : 1, "item" : "Americanos", "price" : 5, "size": "Short", "quantity" : 22, "date" : ISODate("2022-01-15T08:00:00Z") },

{ "\_id" : 2, "item" : "Cappuccino", "price" : 6, "size": "Short","quantity" : 12, "date" : ISODate("2022-01-16T09:00:00Z") },

{ "\_id" : 3, "item" : "Lattes", "price" : 15, "size": "Grande","quantity" : 25, "date" : ISODate("2022-01-16T09:05:00Z") },

{ "\_id" : 4, "item" : "Mochas", "price" : 25,"size": "Tall", "quantity" : 11, "date" : ISODate("2022-02-17T08:00:00Z") },

{ "\_id" : 5, "item" : "Americanos", "price" : 10, "size": "Grande","quantity" : 12, "date" : ISODate("2022-02-18T21:06:00Z") },

{ "\_id" : 6, "item" : "Cappuccino", "price" : 7, "size": "Tall","quantity" : 20, "date" : ISODate("2022-02-20T10:07:00Z") },

{ "\_id" : 7, "item" : "Lattes", "price" : 25,"size": "Tall", "quantity" : 30, "date" : ISODate("2022-02-21T10:08:00Z") },

{ "\_id" : 8, "item" : "Americanos", "price" : 10, "size": "Grande","quantity" : 21, "date" : ISODate("2022-02-22T14:09:00Z") },

{ "\_id" : 9, "item" : "Cappuccino", "price" : 10, "size": "Grande","quantity" : 17, "date" : ISODate("2022-02-23T14:09:00Z") },

{ "\_id" : 10, "item" : "Americanos", "price" : 8, "size": "Tall","quantity" : 15, "date" : ISODate("2022-02-25T14:09:00Z")}

]);

/\*  
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}}])

/\*  
2.Calculate the total quantity sold per month in 2022.

\*/  
db.sales.aggregate([$match: {

date: {

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

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

}

},])

/\*

3.Find all items where price is greater than 10 and size is not 'Short'.  
\*/  
db.sales.find({

price: { $gt: 10 },

size: { $ne: "Short" }

});

/\*

4.Get all Cappuccino sales with quantity between 10 and 20.  
\*/  
db.sales.find({item:{$eq:"Cappuccino"},quantity:{$gt:10,$lte:20}})

/\*

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

\*/

db.sales.find({item:{$regex:"^A"}})

/\*

6.Find all records that do not have the field size.  
\*/  
db.sales.find({"size":{exists:false}})

//there is no document that doesn’t consists of size field

/\*

7.List all items sold in February 2022.  
\*/  
db.sales.find(

{

date: {

$gte: ISODate("2022-02-01T00:00:00Z"),

$lt: ISODate("2022-03-01T00:00:00Z")

}

}

)

/\*  
8.Find all sales that are either "Grande" or "Tall" but not "Americanos".  
\*/

db.sales.find({size:{$in:["Grande","Tall"]},item:{$ne:"Americanos"}})  
/\*

9.Find sales where the quantity is more than twice the price.  
\*/

db.sales.find({price:{gt:{$mutliply:[2,"price"]}}})  
/\*

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

/\*

db.sales.aggregate([

{

$setWindowFields: {

partitionBy: "$size",

output: {

avgPrice: { $avg: "$price" }

}

}

},

{

$match: {

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

}

},

{

$project: {

\_id: 0,

item: 1,

size: 1,

price: 1,

avgPrice: 1

}

}

]);

11.Filter sales where the total revenue is even and exceeds 100.  
\*/

\*/  
db.sales.aggregate([

{

$addFields: {

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

}

},

{

$match: {

$expr: {

$and: [

{ $gt: ["$totalRevenue", 100] },

{ $eq: [{ $mod: ["$totalRevenue", 2] }, 0] }

]

}

}

},

{

$project: {

\_id: 0,

item: 1,

price: 1,

quantity: 1,

totalRevenue: 1

}

}

]);

/\*

12.Find Sales Where the Day of Week Matches Quantity's Last Digit

(Day of week: 0 = Sunday, 1 = Monday, ..., 6 = Saturday)

\*/  
db.sales.aggregate([

{

$addFields: {

dayoftheWeek: { $dayOfWeek: "$date" }, // 1 (Sunday) to 7 (Saturday)

lastdigit: { $mod: ["$quantity", 10] } // Gets last digit of quantity

}

},

{

$match: {

$expr: { $eq: ["$dayoftheWeek", "$lastdigit"] } // Match when equal

}

}

])//how ever we are doing pipelining where condition not used

/\*

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

(Prime months: 2, 3, 5, 7, 11)  
\*/  
db.sales.aggregate([

{

$addFields: {

month: { $month: "$date" }

}

},

{

$addFields: {

isPrimeMonth: {

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

},

isQuantityOdd: {

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

}

}

},

{

$match: {

isPrimeMonth: true,

isQuantityOdd: true

}

}

])

/\*

14.Find Sales with "Suspicious Quantities" (Divisible by 5 or 7).  
\*/

db.sales.aggregate([

{

$match: {

$or: [

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

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

]

}

}

])