**MongoDB Test:**

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

**Code:**

db.sales.aggregate([{$group:{\_id:"$item",total\_revenue:{$sum:{$multiply:['$price','$quantity']}}}},{$sort:{total\_revenue:-1}}]);

**Output:**

{

\_id: 'Lattes',

total\_revenue: 1125

} {

\_id: 'Americanos',

total\_revenue: 560

} {

\_id: 'Cappuccino',

total\_revenue: 382

} {

\_id: 'Mochas',

total\_revenue: 275}

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

**Code:**

db.sales.aggregate([{$group:{\_id:{$month:"$date"},total\_qty:{$sum:"$quantity"}}},{$sort:{\_id:1}}])

**Output:**

{

\_id: 1,

total\_qty: 59

}

{

\_id: 2,

total\_qty: 126

}

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

**Code:**

db.sales.find({price:{$gt:10},size:{$ne:"Short"}})

**Output:**

{

\_id: 3,

item: 'Lattes',

price: 15,

size: 'Grande',

quantity: 25,

date: 2022-01-16T09:05:00.000Z

}{

\_id: 4,

item: 'Mochas',

price: 25,

size: 'Tall',

quantity: 11,

date: 2022-02-17T08:00:00.000Z

}{

\_id: 7,

item: 'Lattes',

price: 25,

size: 'Tall',

quantity: 30,

date: 2022-02-21T10:08:00.000Z

}

**4.** Get all Cappuccino sales with quantity between 10 and 20.

**Code:**

**For >10 and <20:**

db.sales.find({item:"Cappuccino",quantity:{$gt:10,$lt:20}})

**Output:**

{

\_id: 2,

item: 'Cappuccino',

price: 6,

size: 'Short',

quantity: 12,

date: 2022-01-16T09:00:00.000Z

}{

\_id: 9,

item: 'Cappuccino',

price: 10,

size: 'Grande',

quantity: 17,

date: 2022-02-23T14:09:00.000Z

}

**Code:**

**For >=10 and <=20:**

db.sales.find({item:"Cappuccino",quantity:{$gte:10,$lte:20}})

**Output:**

{

\_id: 2,

item: 'Cappuccino',

price: 6,

size: 'Short',

quantity: 12,

date: 2022-01-16T09:00:00.000Z

}{

\_id: 6,

item: 'Cappuccino',

price: 7,

size: 'Tall',

quantity: 20,

date: 2022-02-20T10:07:00.000Z

}{

\_id: 9,

item: 'Cappuccino',

price: 10,

size: 'Grande',

quantity: 17,

date: 2022-02-23T14:09:00.000Z

}

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

**Code:**

db.sales.find({item: { $gte: "A", $lt: "B" }});

**Output:**

{

\_id: 1,

item: 'Americanos',

price: 5,

size: 'Short',

quantity: 22,

date: 2022-01-15T08:00:00.000Z

}

\_id: 5,

item: 'Americanos',

price: 10,

size: 'Grande',

quantity: 12,

date: 2022-02-18T21:06:00.000Z

}{

\_id: 8,

item: 'Americanos',

price: 10,

size: 'Grande',

quantity: 21,

date: 2022-02-22T14:09:00.000Z

}{

\_id: 8,

item: 'Americanos',

price: 10,

size: 'Grande',

quantity: 21,

date: 2022-02-22T14:09:00.000Z

}

**6**. Find all records that do not have the field size.

**Code:**

db.sales.find({size:{$exists: false }});

**Output:**

----- no output -----

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

**Code:**

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

**Output:**

{

\_id: 3,

item: 'Lattes',

price: 15,

size: 'Grande',

quantity: 25,

date: 2022-01-16T09:05:00.000Z

}{

\_id: 4,

item: 'Mochas',

price: 25,

size: 'Tall',

quantity: 11,

date: 2022-02-17T08:00:00.000Z

}{

\_id: 6,

item: 'Cappuccino',

price: 7,

size: 'Tall',

quantity: 20,

date: 2022-02-20T10:07:00.000Z

}{

\_id: 7,

item: 'Lattes',

price: 25,

size: 'Tall',

quantity: 30,

date: 2022-02-21T10:08:00.000Z

}{

\_id: 7,

item: 'Lattes',

price: 25,

size: 'Tall',

quantity: 30,

date: 2022-02-21T10:08:00.000Z

}

**8**. List all items sold in February 2022.

**Code:**

db.sales.find({date: { $gte: new Date("2022-02-01"), $lt: new Date("2022-03-01") }}, { item: 1, \_id: 0 });

**Output:**

{

item: 'Mochas'

}{

item: 'Americanos'

}{

item: ‘Cappuccino'

}{

item: 'Lattes'

}{

item: 'Americanos'

}{

item: 'Cappuccino'

}{

item: 'Americanos'

}

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

**Code:**

db.sales.find({$expr: { $gt: ["$quantity", { $multiply: [2, "$price"] }] }});

**Output:**

{

\_id: 1,

item: 'Americanos',

price: 5,

size: 'Short',

quantity: 22,

date: 2022-01-15T08:00:00.000Z

} {

\_id: 6,

item: 'Cappuccino',

price: 7,

size: 'Tall',

quantity: 20,

date: 2022-02-20T10:07:00.000Z

} {

\_id: 6,

item: 'Cappuccino',

price: 7,

size: 'Tall',

quantity: 20,

date: 2022-02-20T10:07:00.000Z

}

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

**Code:**

db.sales.find({

$where: function() {

const avgPrices = {

"Short": 5.5,

"Grande": 12.5,

"Tall": 13.33

};

return this.price > avgPrices[this.size];

}});

**Output:**

{

\_id: 2,

item: 'Cappuccino',

price: 6,

size: 'Short',

quantity: 12,

date: 2022-01-16T09:00:00.000Z

}{

\_id: 3,

item: 'Lattes',

price: 15,

size: 'Grande',

quantity: 25,

date: 2022-01-16T09:05:00.000Z{

\_id: 4,

item: 'Mochas',

price: 25,

size: 'Tall',

quantity: 11,

date: 2022-02-17T08:00:00.000Z

}{

\_id: 4,

item: 'Mochas',

price: 25,

size: 'Tall',

quantity: 11,

date: 2022-02-17T08:00:00.000Z

}

**11**. 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]

**Code:**

db.sales.find({

$where: function () {

const day = this.date.getDay();

const lastDigit = this.quantity % 10;

return day === lastDigit;

}});

**Output:**

{

\_id: 6,

item: 'Cappuccino',

price: 7,

size: 'Tall',

quantity: 20,

date: 2022-02-20T10:07:00.000Z

} {

\_id: 10,

item: 'Americanos',

price: 8,

size: 'Tall',

quantity: 15,

date: 2022-02-25T14:09:00.000Z

}

**12.** 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: {month: { $in: [2, 3, 5, 7, 11] },quantity: { $mod: [2, 1] }}}]);

**Output:**

{

\_id: 4,

item: **'Mochas'**,

price: 25,

size: **'Tall'**,

quantity: 11,

date: 2022-02-17T08:00:00.000Z,

month: 2

}

{

\_id: 8,

item: **'Americanos'**,

price: 10,

size: **'Grande'**,

quantity: 21,

date: 2022-02-22T14:09:00.000Z,

month: 2

}

{

\_id: 9,

item: **'Cappuccino'**,

price: 10,

size: **'Grande'**,

quantity: 17,

date: 2022-02-23T14:09:00.000Z,

month: 2

}

{

\_id: 10,

item: **'Americanos'**,

price: 8,

size: **'Tall'**,

quantity: 15,

date: 2022-02-25T14:09:00.000Z,

month: 2

}

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

**Code:**

db.sales.aggregate([{$match: {$or: [{ quantity: { $mod: [5, 0] } },{ quantity: { $mod: [7, 0] } }]}}]);

**Output:**

{

\_id: 3,

item: **'Lattes'**,

price: 15,

size: **'Grande'**,

quantity: 25,

date: 2022-01-16T09:05:00.000Z

}

{

\_id: 6,

item: **'Cappuccino'**,

price: 7,

size: **'Tall'**,

quantity: 20,

date: 2022-02-20T10:07:00.000Z

}

{

\_id: 7,

item: **'Lattes'**,

price: 25,

size: **'Tall'**,

quantity: 30,

date: 2022-02-21T10:08:00.000Z

}

{

\_id: 8,

item: **'Americanos'**,

price: 10,

size: **'Grande'**,

quantity: 21,

date: 2022-02-22T14:09:00.000Z

}

{

\_id: 10,

item: **'Americanos'**,

price: 8,

size: **'Tall'**,

quantity: 15,

date: 2022-02-25T14:09:00.000Z

}