use test

already on db test

show collections

Vit

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")}

]);

```
{
  acknowledged: true,
  insertedIds: {
    '0': 1,
    '1': 2,
    '2': 3,
    '3': 4,
    '4': 5,
    '5': 6,
    '6': 7,
    '7': 8,
    '8': 9,
    '9': 10
  }
}
```

// 1. How do you find all documents where the field "item" is "Mochas"?

```
> db.sales.find({ "item": "Mochas" }).pretty()
{
    _id: 4,
    item: 'Mochas',
    price: 25,
    size: 'Tall',
    quantity: 11,
    date: 2022-02-17T08:00:00.000Z
}
```

// 2. How do you find all items who are less than 15 of price?

```
db.sales.find({ "price": { $lt: 15 } }).pretty()
{
    _id: 1,
    item: 'Americanos',
    price: 5,
    size: 'Short',
    quantity: 22,
    date: 2022-01-15T08:00:00.000Z
}
{
    _id: 2,
```

// 3. How do you find items who are either less than 10 or greater than 20?

```
db.sales.find({
    $or: [
        { "price": { $lt: 10 } },
        { "price": { $gt: 20 } }
    ]
}).pretty()
{
    _id: 1,
    item: 'Americanos',
    price: 5,
    size: 'Short',
    quantity: 22,
    date: 2022-01-15T08:00:00.000Z
}
{
    _id: 2,
```

// 4. How do you find all documents where the quantity field exists?

```
db.sales.find({ "quantity": { $exists: true } }).pretty()
{
  _id: 1,
  item: 'Americanos',
  price: 5,
  size: 'Short',
  quantity: 22,
  date: 2022-01-15T08:00:00.000Z
}
{
  _id: 2,
```

db.products.insertMany([

{ "_id" : 1, "name" : "xPhone", "price" : 799, "releaseDate": ISODate("2011-05-14"), "spec" : { "ram" : 4, "screen" : 6.5, "cpu" : 2.66 },"color":["white","black"],"storage":[64,128,256]},

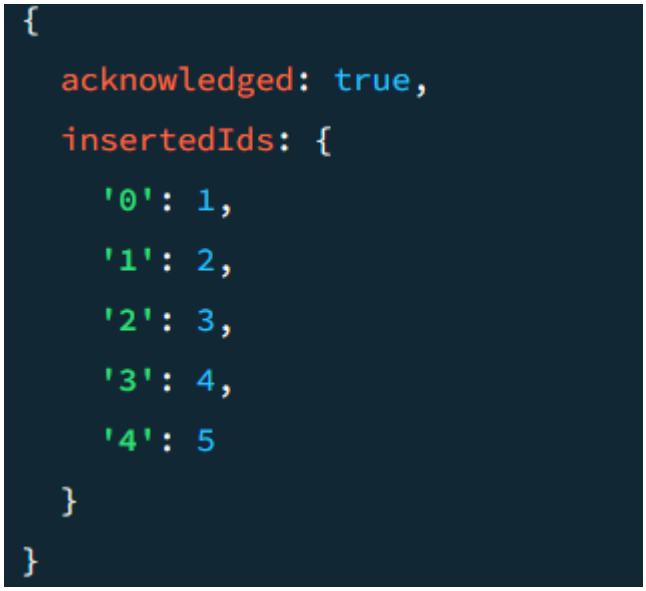
{ "_id" : 2, "name" : "xTablet", "price" : 899, "releaseDate": ISODate("2011-09-01") , "spec" : { "ram" : 16, "screen" : 9.5, "cpu" : 3.66 },"color":["white","black","purple"],"storage":[128,256,512]},

{ "_id" : 3, "name" : "SmartTablet", "price" : 899, "releaseDate": ISODate("2015-01-14"), "spec" : { "ram" : 12, "screen" : 9.7, "cpu" : 3.66 },"color":["blue"],"storage":[16,64,128]},

{ "_id" : 4, "name" : "SmartPad", "price" : 699, "releaseDate": ISODate("2020-05-14"),"spec" : { "ram" : 8, "screen" : 9.7, "cpu" : 1.66 },"color":["white","orange","gold","gray"],"storage":[128,256,1024]},

{ "_id" : 5, "name" : "SmartPhone", "price" : 599,"releaseDate": ISODate("2022-09-14"), "spec" : { "ram" : 4, "screen" : 9.7, "cpu" : 1.66 },"color":["white","orange","gold","gray"],"storage":[128,256]}

])



// 5. How do you find all documents where the color array contains the value "white"?

// 6. How do you update the ram of the user named "xTablet" to 24?

```
db.products.updateOne(
  { "name": "xTablet" },
  { $set: { "spec.ram": 24 } }
)
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

// 7. How do you find all products and only return their 'spec' field?

```
db.products.find({}, { "spec": 1, "_id": 0 }).pretty()
{
  spec: {
    ram: 4,
    screen: 6.5,
    cpu: 2.66
  }
}
{
  spec: {
```

// 8. How do you find all products and sort them by their price in descending order?

// 9. How do you find the first 2 products, skipping the first one?

10. How do you find all users whose name starts with the letter "S" and price should be greater of 700?

```
db.products.find(
  {
    "name": { $regex: "^S", $options: "i" },
    "price": { $gt: 700 }
  }
).pretty()
{
  _id: 3,
  name: 'SmartTablet',
  price: 899,
  releaseDate: 2015-01-14T00:00:00.000Z,
  spec: {
    ram: 12,
    screen: 9.7,
    cpu: 3.66
  },
  color: [
    'blue'
  ],
  storage: [
    16,
    64,
    128
  ]
}
```

// 11. How do you calculate the total price of all items?

```
db.sales.aggregate([
  {
    $group: {
      _id: null,
      totalPrice: { $sum: "$price" }
    }
  }
])
{
  _id: null,
  totalPrice: 121
}
```

// 12.How do you project only the size field for each items?

```
db.sales.find({}, { "size": 1, "_id": 0 }).pretty()
{
  size: 'Short'
}
{
  size: 'Short'
}
{
  size: 'Grande'
}
{
```

// 13. How do you find items who are Tall size & group them by item wise?

```
db.sales.aggregate([
  { $match: { "size": "Tall" } },
  {
    $group: {
      _id: "$item",
      totalQuantity: { $sum: "$quantity" }
    }
  }
])
{
  _id: 'Mochas',
  totalQuantity: 11
}
{
  _id: 'Lattes',
  totalQuantity: 30
}
{
```

// 14. How do you find the second item when sorted by price in ascending order?

```
db.sales.find().sort({ "price": 1 }).skip(1).limit(1).pretty()
{
  _id: 2,
  item: 'Cappuccino',
  price: 6,
  size: 'Short',
  quantity: 12,
  date: 2022-01-16T09:00:00.000Z
}
```