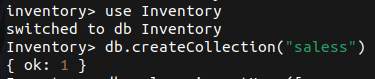
**Lab cycle 6**

>Use Inventory

>db.createCollection("saless")



>db.saless.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":

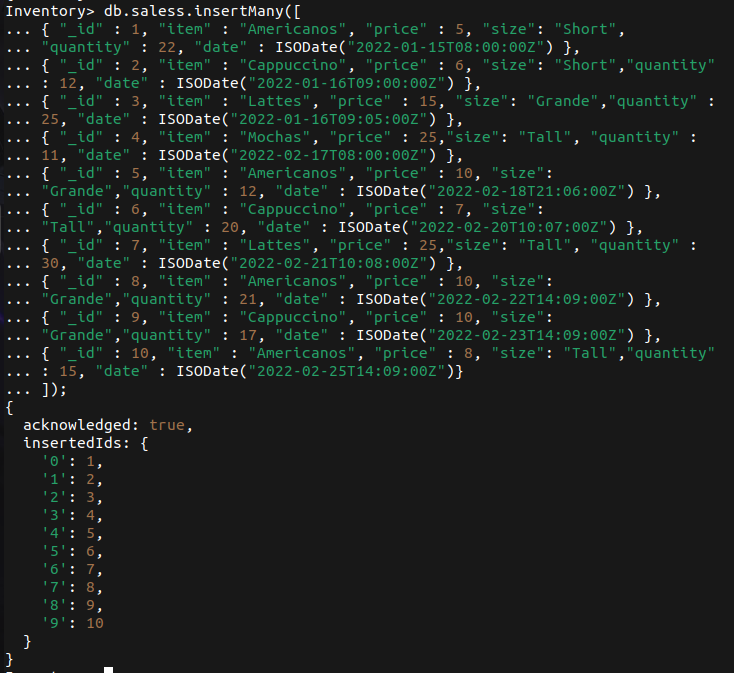
"Grande","quantity" : 21, "date" : ISODate("2022-02-22T14:09:00Z") },

{ "\_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. Groups the documents by the item field and use the $avg to calculate the average amount for

each group.

db.saless.aggregate([

{

$group: {

\_id: '$item',

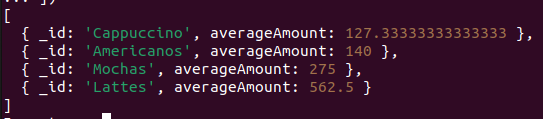
averageAmount: { $avg: { $multiply: ['$quantity', '$price'] } },

},

},

{ $sort: { averageAmount: 1 } },

])



2.Calculate the average amount per group and returns the group with the average

amount greater than 150.

db.saless.aggregate([

{

$group: {

\_id: '$item',

averageAmount: { $avg: { $multiply: ['$quantity', '$price'] } },

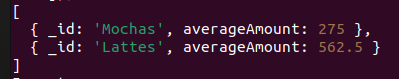
},

},

{ $match: { averageAmount: { $gt: 150 } } },

{ $sort: { averageAmount: 1 } },

]);



3. Return the number of items in the sales collection.

db.saless.aggregate([

{

$group: {

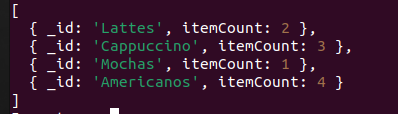
\_id: '$item',

itemCount: { $count: {} },

},

},

])



4. Calculate the number of documents per item and returns the item with a count greater than

two.

db.saless.aggregate([

{

$group: {

\_id: '$item',

itemCount: { $count: {} },

},

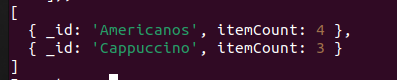
},

{

$match: { itemCount: { $gt: 2 } },

},

]);



5. Calculates the total quantity of coffee sales in the sales collection.

db.saless.aggregate([

{

$group: {

\_id: null,

totalQty: { $sum: '$quantity' },

},

},

{ $project: { \_id: 0 } },

]);



6. Calculate the sum of quantity grouped by items

db.saless.aggregate([

{

$group: {

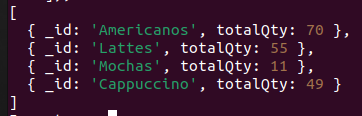
\_id: '$item',

totalQty: { $sum: '$quantity' },

},

},

]);



7.Returns the total quantity of each item and sorts the result documents by the

totalQty in descending order.

db.saless.aggregate([

{

$group: {

\_id: '$item',

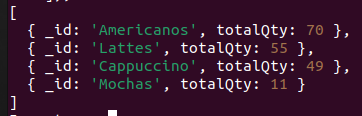
totalQty: { $sum: '$quantity' },

},

},

{ $sort: { totalQty: -1 } },

]);



8. Find the maximum quantity from all the sales documents.

db.saless.aggregate([

{

$group: {

\_id: null,

maxQty: { $max: '$quantity' },

},

},

{

$project: {

\_id: 0,

},

},

]);



9. Group documents in the item field and returns the maximum quantity per group of

documents.

db.saless.aggregate([

{

$group: {

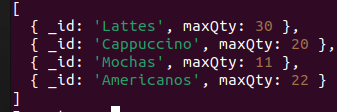
\_id: '$item',

maxQty: { $max: '$quantity' },

},

},

]);



10. Groups the documents by the item field and returns the maximum amount for each group of

sales.

db.saless.aggregate([

{

$group: {

\_id: '$item',

maxQty: { $max: { $multiply: ['$quantity', '$price'] } },

},

},

]);

