DEVIKA B
Roll No:22

LAB CYCLE -6 :

MongoDB Aggregation Operations

Use the following sales collection

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") },

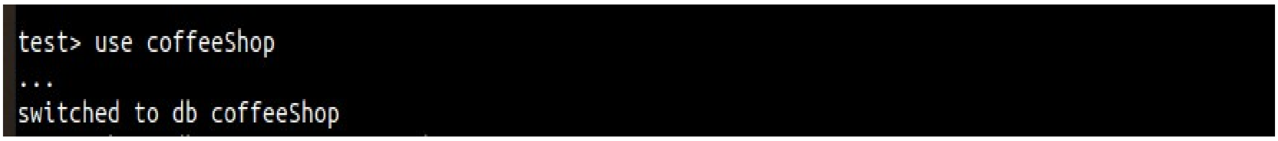
{ _id: 7, item: "Lattes", price: 25, size: "Tall", quantity: 30, date: ISODate("2022-02-21T10:08:00Z") },

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

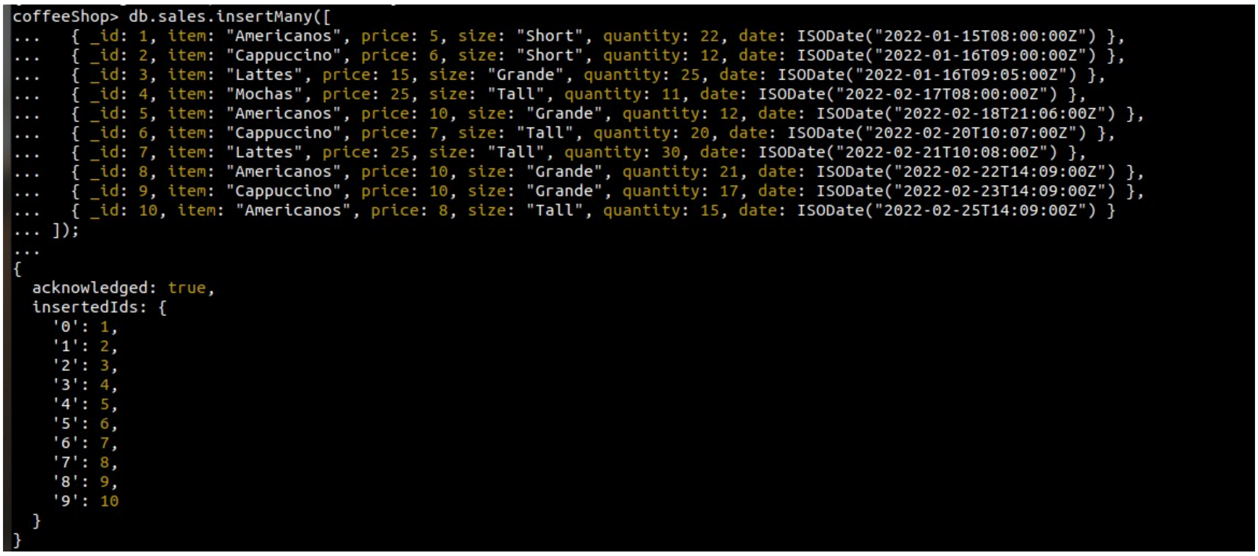
{ _id: 9, item: "Cappuccino", price: 10, size: "Grande", quantity: 17, date: ISODate("2022-02-23T14:09:00Z") },

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

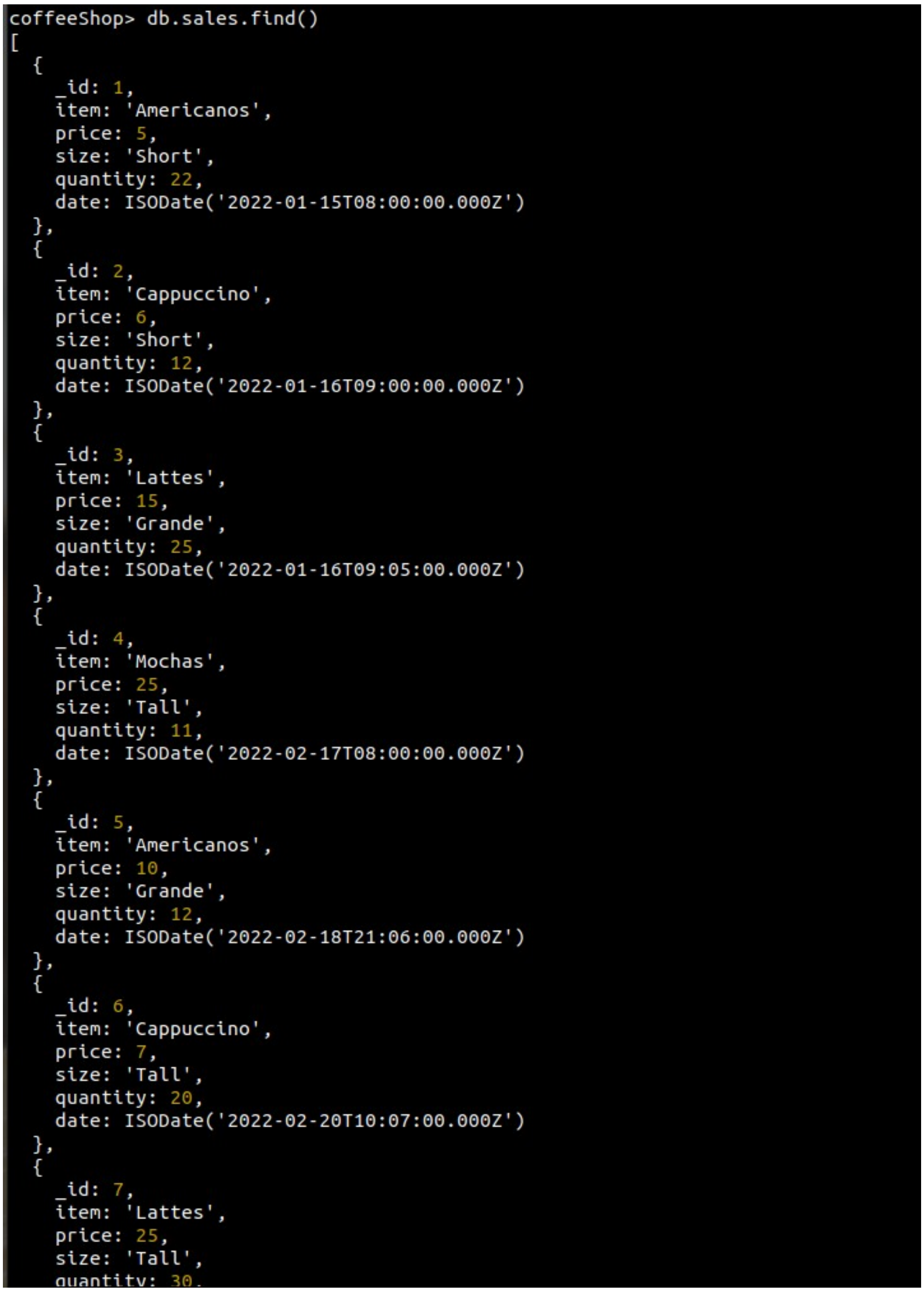
]);



```
test> use coffeeShop
...
switched to db coffeeShop
```



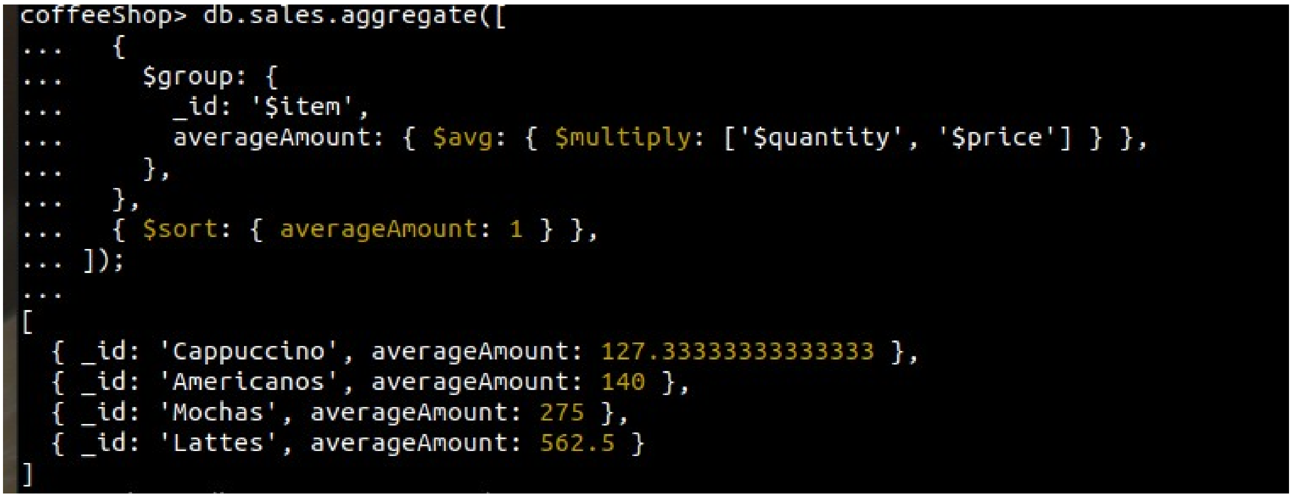
```
coffeeShop> db.sales.insertMany([
...   { _id: 1, item: "Americanos", price: 5, size: "Short", quantity: 22, date: ISODate("2022-01-15T08:00:00Z") },
...   { _id: 2, item: "Cappuccino", price: 6, size: "Short", quantity: 12, date: ISODate("2022-01-16T09:00:00Z") },
...   { _id: 3, item: "Lattes", 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
  }
}
```



```
coffeeShop> db.sales.find()
[
  {
    _id: 1,
    item: 'Americanos',
    price: 5,
    size: 'Short',
    quantity: 22,
    date: ISODate('2022-01-15T08:00:00.000Z')
  },
  {
    _id: 2,
    item: 'Cappuccino',
    price: 6,
    size: 'Short',
    quantity: 12,
    date: ISODate('2022-01-16T09:00:00.000Z')
  },
  {
    _id: 3,
    item: 'Lattes',
    price: 15,
    size: 'Grande',
    quantity: 25,
    date: ISODate('2022-01-16T09:05:00.000Z')
  },
  {
    _id: 4,
    item: 'Mochas',
    price: 25,
    size: 'Tall',
    quantity: 11,
    date: ISODate('2022-02-17T08:00:00.000Z')
  },
  {
    _id: 5,
    item: 'Americanos',
    price: 10,
    size: 'Grande',
    quantity: 12,
    date: ISODate('2022-02-18T21:06:00.000Z')
  },
  {
    _id: 6,
    item: 'Cappuccino',
    price: 7,
    size: 'Tall',
    quantity: 20,
    date: ISODate('2022-02-20T10:07:00.000Z')
  },
  {
    _id: 7,
    item: 'Lattes',
    price: 25,
    size: 'Tall',
    quantity: 30,
```

1. Group documents by the `item` field and calculate the average amount for each group.

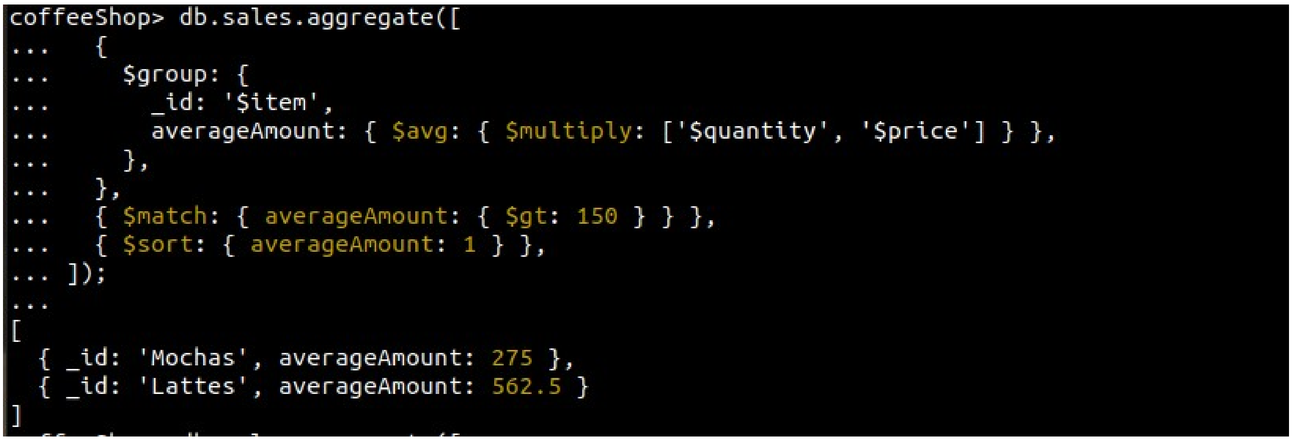
```
db.sales.aggregate([
  {
    $group: {
      _id: '$item',
      averageAmount: { $avg: { $multiply: ['$quantity', '$price'] } },
    },
  },
  { $sort: { averageAmount: 1 } },
]);
```



```
coffeeShop> db.sales.aggregate([
...   {
...     $group: {
...       _id: '$item',
...       averageAmount: { $avg: { $multiply: ['$quantity', '$price'] } },
...     },
...   },
...   { $sort: { averageAmount: 1 } },
... ]);
...
[
  { _id: 'Cappuccino', averageAmount: 127.33333333333333 },
  { _id: 'Americanos', averageAmount: 140 },
  { _id: 'Mochas', averageAmount: 275 },
  { _id: 'Lattes', averageAmount: 562.5 }
]
```

2. Calculate the average amount per group and return groups with average amount greater than 150.

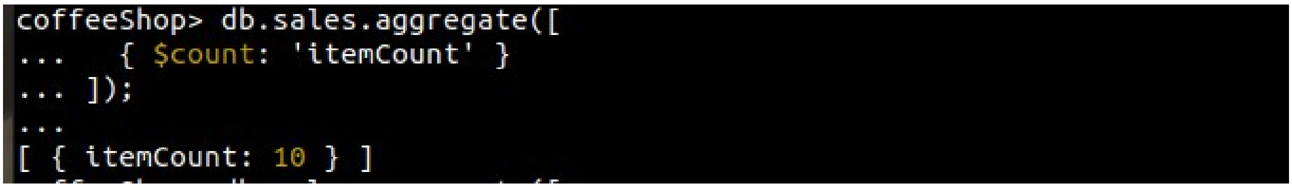
```
db.sales.aggregate([
  {
    $group: {
      _id: '$item',
      averageAmount: { $avg: { $multiply: ['$quantity', '$price'] } },
    },
  },
  { $match: { averageAmount: { $gt: 150 } } },
  { $sort: { averageAmount: 1 } },
]);
```



```
coffeeShop> db.sales.aggregate([
...   {
...     $group: {
...       _id: '$item',
...       averageAmount: { $avg: { $multiply: ['$quantity', '$price'] } },
...     },
...   },
...   { $match: { averageAmount: { $gt: 150 } } },
...   { $sort: { averageAmount: 1 } },
... ]);
...
[
  { _id: 'Mochas', averageAmount: 275 },
  { _id: 'Lattes', averageAmount: 562.5 }
]
```

3. Return the number of items (documents) in the sales collection.

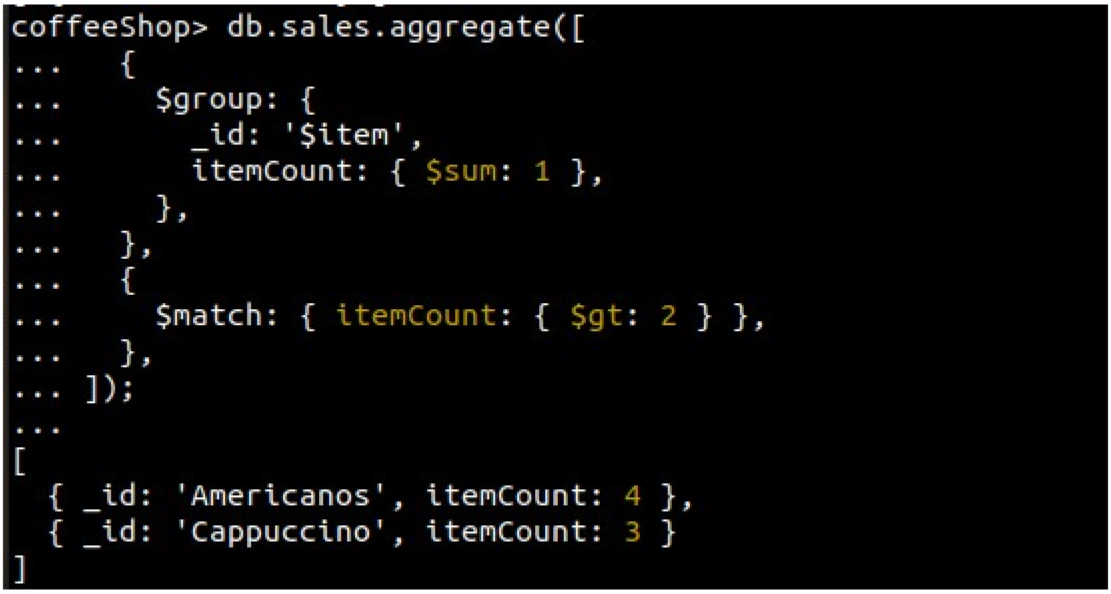
```
db.sales.aggregate([
  { $count: 'itemCount' }
]);
```



```
coffeeShop> db.sales.aggregate([
...   { $count: 'itemCount' }
... ]);
...
[ { itemCount: 10 } ]
```

4. Calculate the number of documents per item and return only items with a count greater than 2.

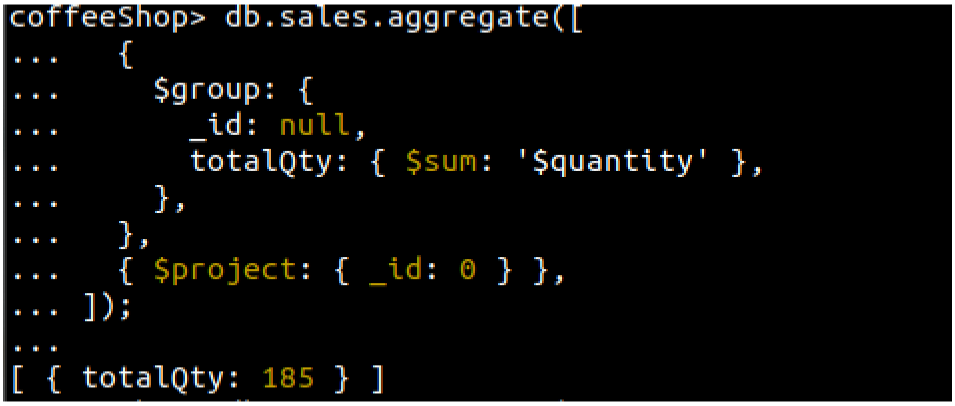
```
db.sales.aggregate([
  {
    $group: {
      _id: '$item',
      itemCount: { $sum: 1 },
    },
  },
  {
    $match: { itemCount: { $gt: 2 } },
  },
]);
```



```
coffeeShop> db.sales.aggregate([
...   {
...     $group: {
...       _id: '$item',
...       itemCount: { $sum: 1 },
...     },
...   },
...   {
...     $match: { itemCount: { $gt: 2 } },
...   },
... ]);
...
[
  { _id: 'Americanos', itemCount: 4 },
  { _id: 'Cappuccino', itemCount: 3 }
]
```

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

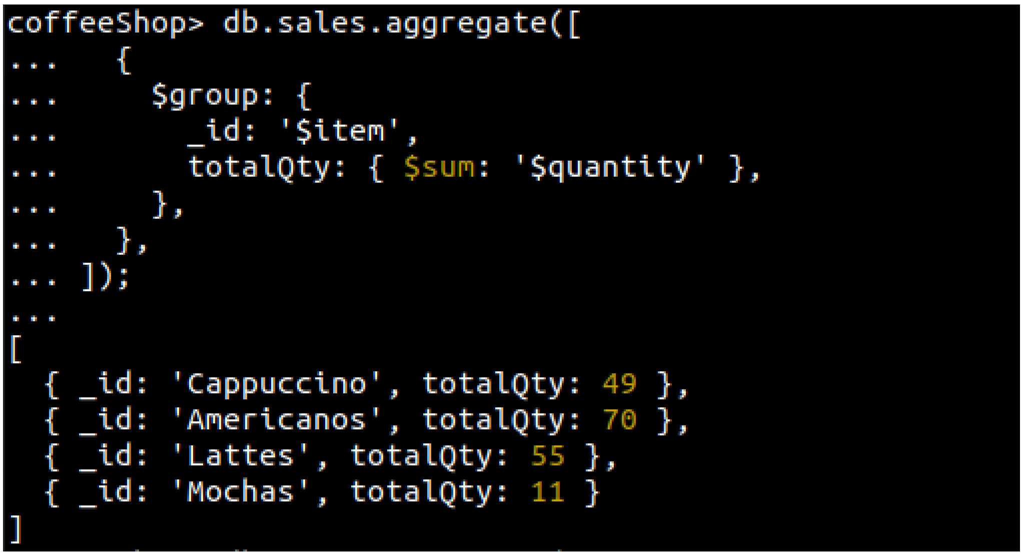
```
db.sales.aggregate([
  {
    $group: {
      _id: null,
      totalQty: { $sum: '$quantity' },
    },
  },
  { $project: { _id: 0 } },
]);
```



```
coffeeShop> db.sales.aggregate([
...   {
...     $group: {
...       _id: null,
...       totalQty: { $sum: '$quantity' },
...     },
...   },
...   { $project: { _id: 0 } },
... ]);
...
[ { totalQty: 185 } ]
```

6. Calculate the sum of quantity grouped by items.

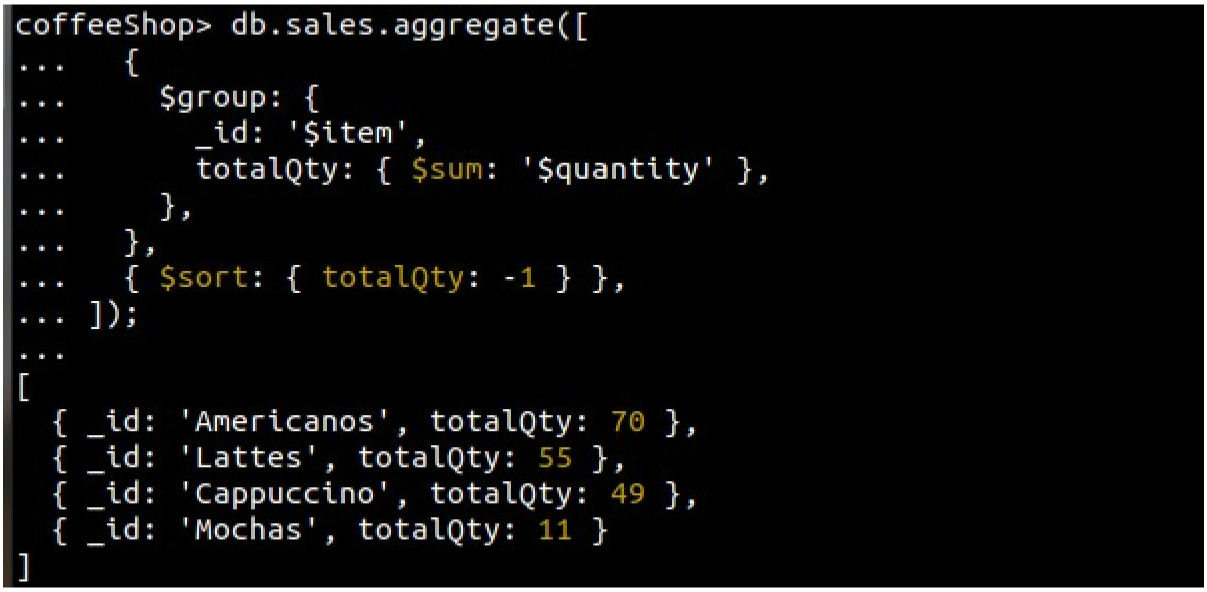
```
db.sales.aggregate([
  {
    $group: {
      _id: '$item',
      totalQty: { $sum: '$quantity' },
    },
  },
]);
```



```
coffeeShop> db.sales.aggregate([
...   {
...     $group: {
...       _id: '$item',
...       totalQty: { $sum: '$quantity' },
...     },
...   },
... ]);
...
[
  { _id: 'Cappuccino', totalQty: 49 },
  { _id: 'Americanos', totalQty: 70 },
  { _id: 'Lattes', totalQty: 55 },
  { _id: 'Mochas', totalQty: 11 }
]
```

7. Return the total quantity of each item and sort the result by `totalQty` in descending order.

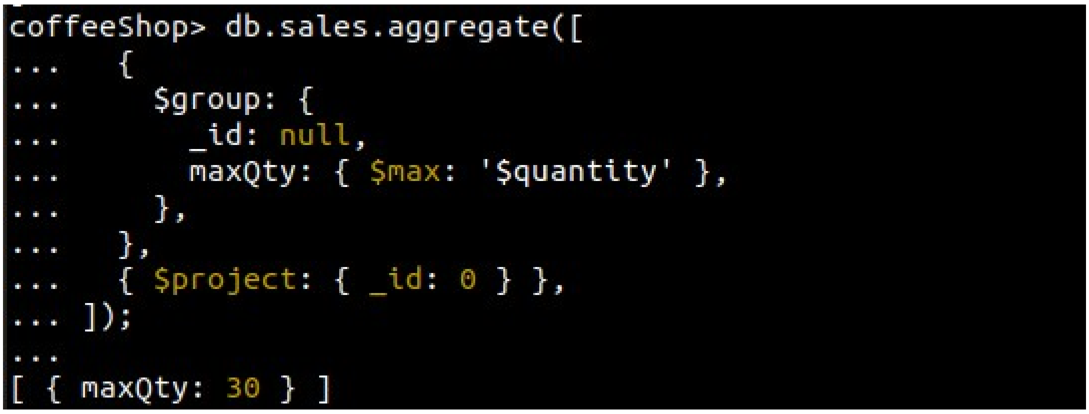
```
db.sales.aggregate([
  {
    $group: {
      _id: '$item',
      totalQty: { $sum: '$quantity' },
    },
  },
  { $sort: { totalQty: -1 } },
]);
```



```
coffeeShop> db.sales.aggregate([
...   {
...     $group: {
...       _id: '$item',
...       totalQty: { $sum: '$quantity' },
...     },
...   },
...   { $sort: { totalQty: -1 } },
... ]);
...
[
  { _id: 'Americanos', totalQty: 70 },
  { _id: 'Lattes', totalQty: 55 },
  { _id: 'Cappuccino', totalQty: 49 },
  { _id: 'Mochas', totalQty: 11 }
]
```

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

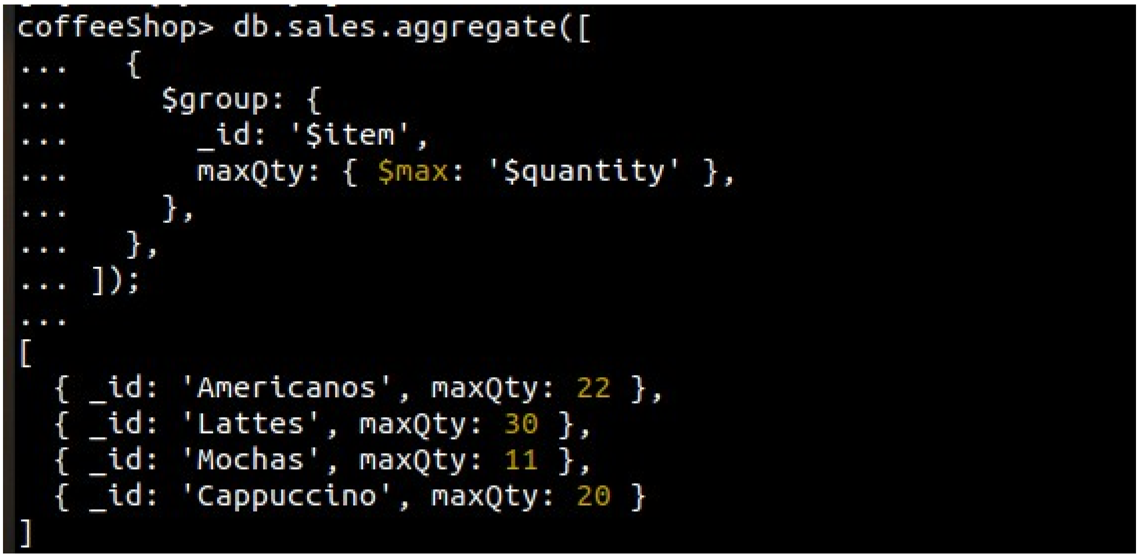
```
db.sales.aggregate([
  {
    $group: {
      _id: null,
      maxQty: { $max: '$quantity' },
    },
  },
  {
    $project: {
      _id: 0,
    },
  },
]);
```



```
coffeeShop> db.sales.aggregate([
...   {
...     $group: {
...       _id: null,
...       maxQty: { $max: '$quantity' },
...     },
...   },
...   { $project: { _id: 0 } },
... ]);
...
[ { maxQty: 30 } ]
```

9. Group by `item` and return the maximum quantity per group.

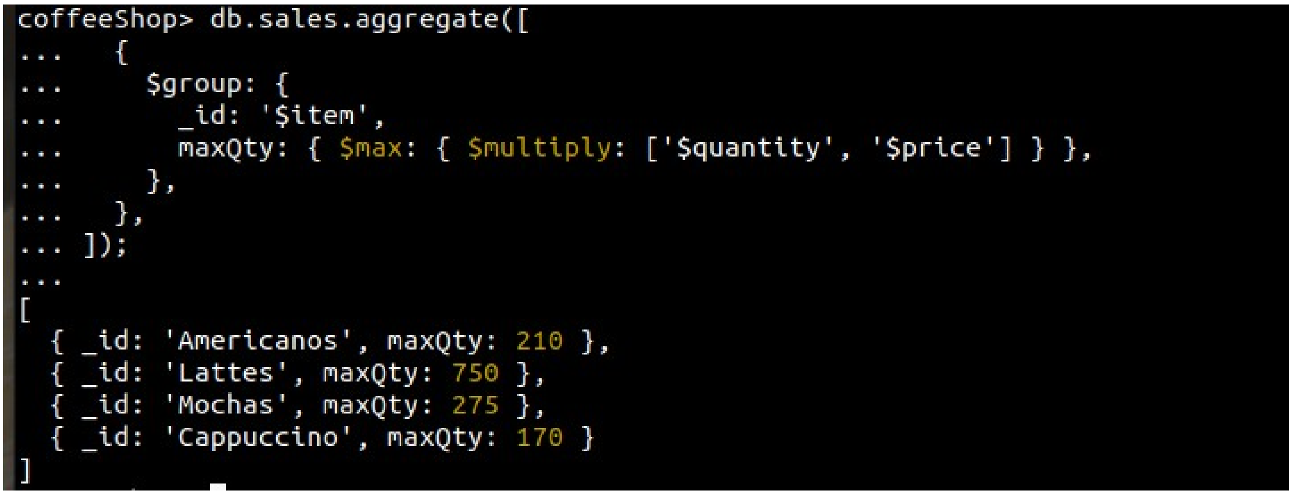
```
db.sales.aggregate([
  {
    $group: {
      _id: '$item',
      maxQty: { $max: '$quantity' },
    },
  },
]);
```



```
coffeeShop> db.sales.aggregate([
...   {
...     $group: {
...       _id: '$item',
...       maxQty: { $max: '$quantity' },
...     },
...   },
... ]);
...
[
  { _id: 'Americanos', maxQty: 22 },
  { _id: 'Lattes', maxQty: 30 },
  { _id: 'Mochas', maxQty: 11 },
  { _id: 'Cappuccino', maxQty: 20 }
]
```

10. Group by `item` and return the maximum total amount (`price * quantity`) for each item.

```
db.sales.aggregate([
  {
    $group: {
      _id: '$item',
      maxQty: { $max: { $multiply: ['$quantity', '$price'] } },
    },
  },
]);
```



```
coffeeShop> db.sales.aggregate([
...   {
...     $group: {
...       _id: '$item',
...       maxQty: { $max: { $multiply: ['$quantity', '$price'] } },
...     },
...   },
... ]);
...
[
  { _id: 'Americanos', maxQty: 210 },
  { _id: 'Lattes', maxQty: 750 },
  { _id: 'Mochas', maxQty: 275 },
  { _id: 'Cappuccino', maxQty: 170 }
]
```