DATE:31/07/2024

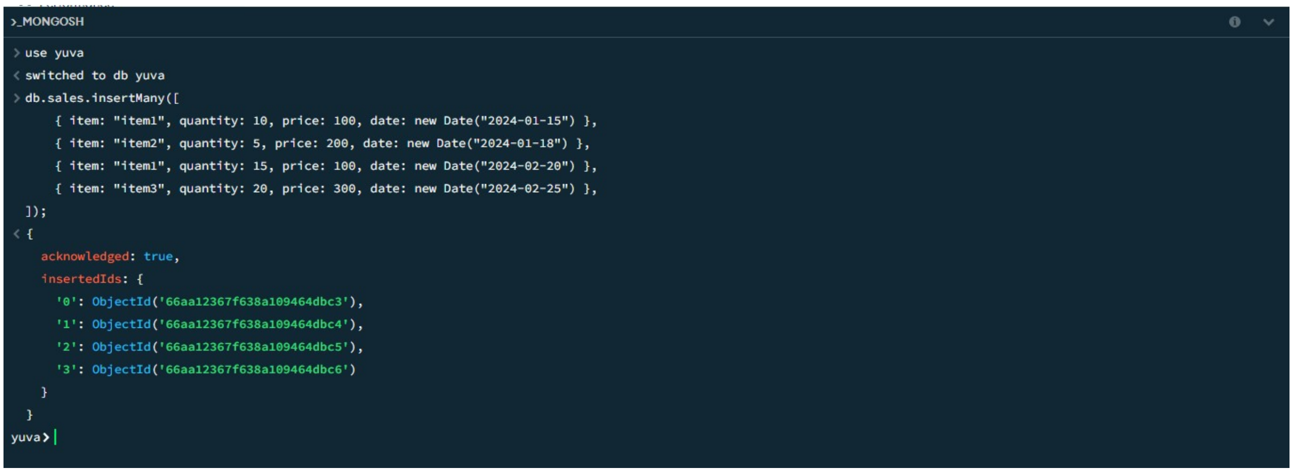
Aggregation Framework

1.Insert documents into a sales collection with fields such as item, quantity, price, and date:

Query:

```
>db.sales.insertMany([
 { item: "item1", quantity: 10, price: 100, date: new Date("2024-01-15") },
 { item: "item2", quantity: 5, price: 200, date: new Date("2024-01-18") },
 { item: "item1", quantity: 15, price: 100, date: new Date("2024-02-20") },
 { item: "item3", quantity: 20, price: 300, date: new Date("2024-02-25") }
 ]);
```

Output:



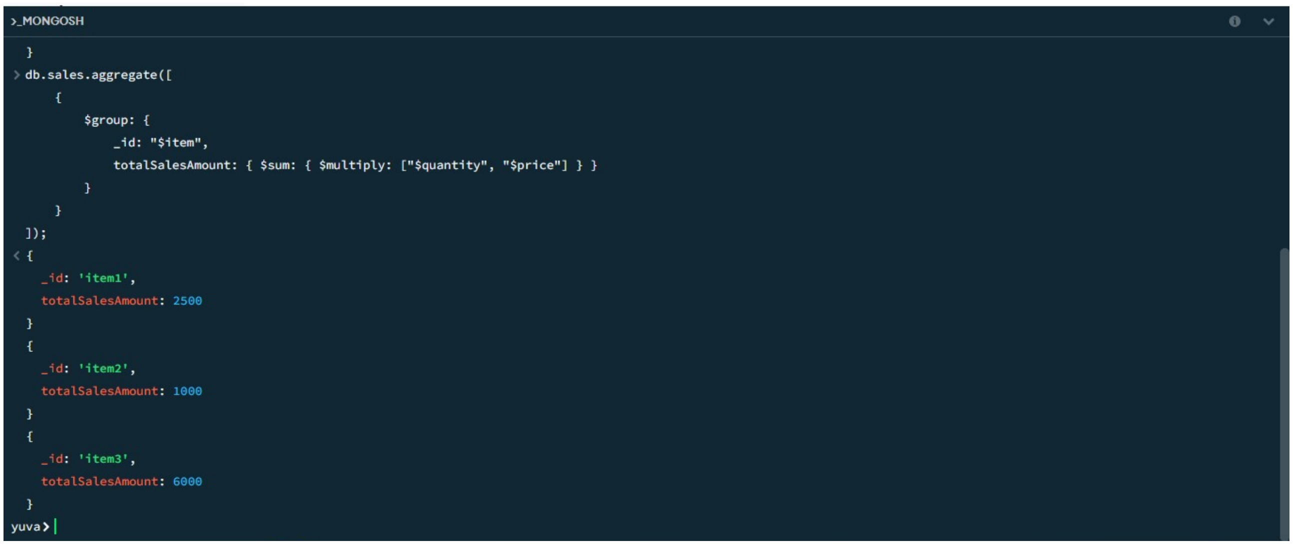
```
>_MONGOSH
> use yuva
< switched to db yuva
> db.sales.insertMany([
      { item: "item1", quantity: 10, price: 100, date: new Date("2024-01-15") },
      { item: "item2", quantity: 5, price: 200, date: new Date("2024-01-18") },
      { item: "item1", quantity: 15, price: 100, date: new Date("2024-02-20") },
      { item: "item3", quantity: 20, price: 300, date: new Date("2024-02-25") },
  ]);
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('66aa12367f638a109464dbc3'),
      '1': ObjectId('66aa12367f638a109464dbc4'),
      '2': ObjectId('66aa12367f638a109464dbc5'),
      '3': ObjectId('66aa12367f638a109464dbc6')
    }
  }
yuva>
```

2.Calculate the total sales amount for each item:

Query:

```
>db.sales.aggregate([
  {
          $group: {
               _id: "$item",
               totalSalesAmount: { $sum: { $multiply: ["$quantity", "$price"] } }
          }
 }
 ]);
```

Output:



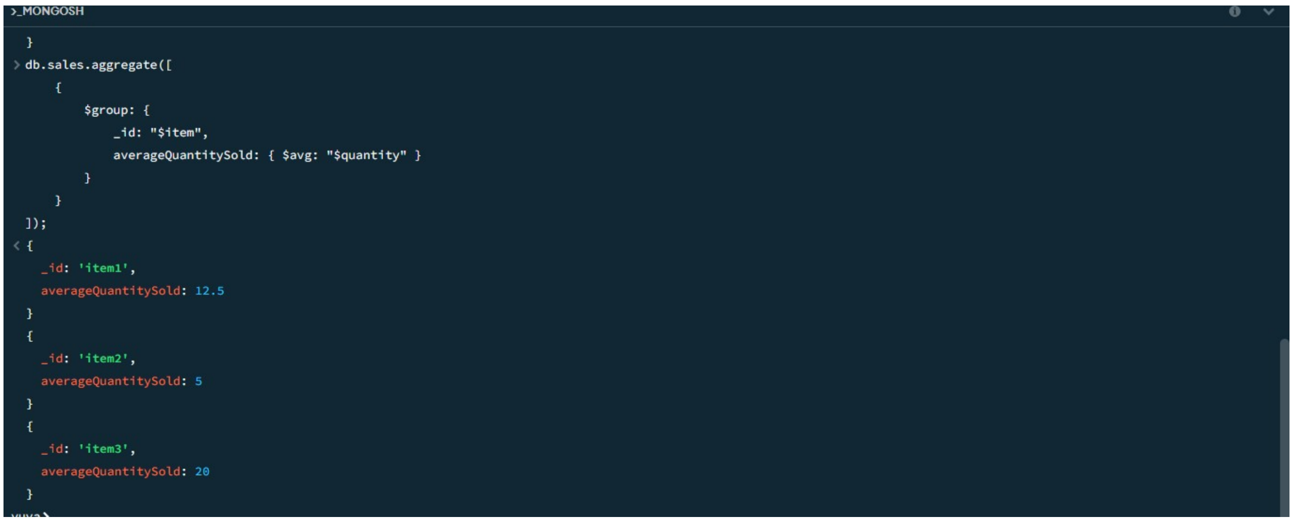
```
  }
> db.sales.aggregate([
      {
          $group: {
              _id: "$item",
              totalSalesAmount: { $sum: { $multiply: ["$quantity", "$price"] } }
          }
      }
  ]);
< {
    _id: 'item1',
    totalSalesAmount: 2500
  }
  {
    _id: 'item2',
    totalSalesAmount: 1000
  }
  {
    _id: 'item3',
    totalSalesAmount: 6000
  }
yuva>
```

3.Find the average quantity sold per item:

Query:

```
>db.sales.aggregate([
 {
        $group: {
            _id: "$item",
            averageQuantitySold: { $avg: "$quantity" }
        }
}
]);
```

Output:



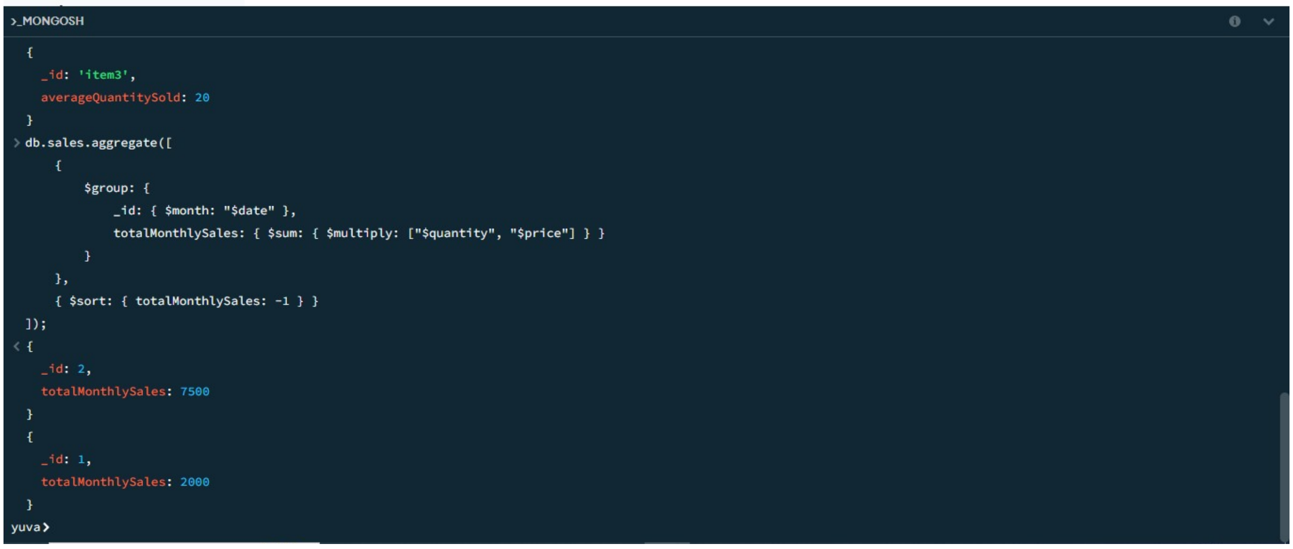
```
  }
> db.sales.aggregate([
      {
          $group: {
              _id: "$item",
              averageQuantitySold: { $avg: "$quantity" }
          }
      }
  ]);
< {
    _id: 'item1',
    averageQuantitySold: 12.5
  }
  {
    _id: 'item2',
    averageQuantitySold: 5
  }
  {
    _id: 'item3',
    averageQuantitySold: 20
  }
```

4.Group sales by month and calculate the total sales for each month and sort from the largest value:

Query:

```
>db.sales.aggregate([
 {
        $group: {
               _id: { $month: "$date" },
              totalMonthlySales: { $sum: { $multiply: ["$quantity",
                   "$price"] } }
         }
 },
  { $sort: { totalMonthlySales: -1 } }
]);
```

Output:



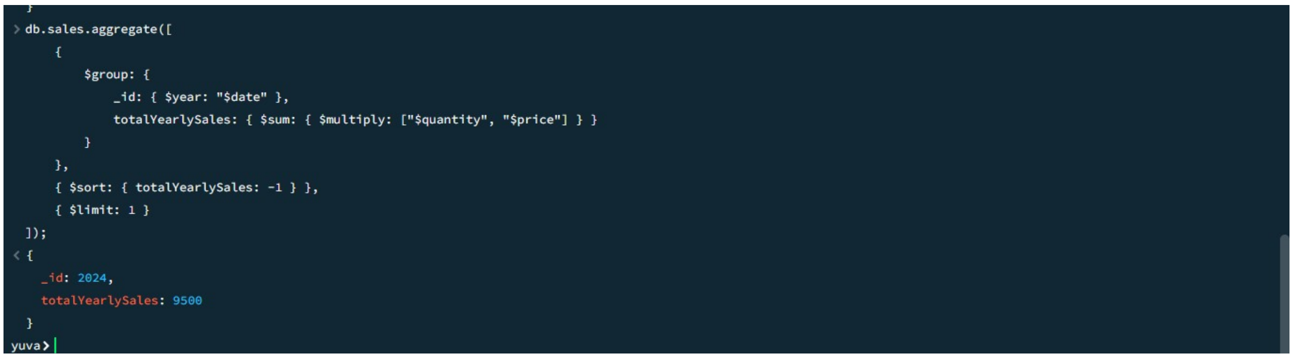
```
>_MONGOSH
  {
    _id: 'item3',
    averageQuantitySold: 20
  }
> db.sales.aggregate([
      {
          $group: {
              _id: { $month: "$date" },
              totalMonthlySales: { $sum: { $multiply: ["$quantity", "$price"] } }
          }
      },
      { $sort: { totalMonthlySales: -1 } }
  ]);
< {
    _id: 2,
    totalMonthlySales: 7500
  }
  {
    _id: 1,
    totalMonthlySales: 2000
  }
yuva>
```

5.Display which year has the maximum sales:

Query:

```
>db.sales.aggregate([
   {
            $group: {
                  _id: { $year: "$date" },
                  totalYearlySales: { $sum: { $multiply: ["$quantity", "$price"] } }
   }
  },
  { $sort: { totalYearlySales: -1 } },
 { $limit: 1 }
]);
```

Output:



```
> db.sales.aggregate([
      {
          $group: {
              _id: { $year: "$date" },
              totalYearlySales: { $sum: { $multiply: ["$quantity", "$price"] } }
          }
      },
      { $sort: { totalYearlySales: -1 } },
      { $limit: 1 }
  ]);
< {
    _id: 2024,
    totalYearlySales: 9500
  }
yuva>
```