Trần Ngọc Tuấn

21280058

# Lab 11

```
// Use aggregate framework

// 1. Display the first and last selling date
db.sales.aggregate([
  { $unwind: "$items" },
  { $group: {
      _id: null,
      firstSellingDay: { $min: '$saleDate' },
      lastSellingDay: { $max: '$saleDate' },
    }
  },
  { $project: {
      _id: 0,
      firstSellingDay: { $dateToString: { format: "%Y/%m/%d", date: {
$toDate: "$firstSellingDay" } } },
      lastSellingDay: { $dateToString: { format: "%Y/%m/%d", date: {
$toDate: "$lastSellingDay" } } }
    }
  }
]);

// 2. Display the nearest selling date that has sold the most items
db.sales.aggregate([
  { $unwind: "$items" },
  { $group: {
      _id: "$saleDate",
      totalQuantity: { $sum: "$items.quantity" }
    }
  },
  { $sort: { totalQuantity: -1, _id: 1 } },
  { $limit: 1 },
  { $project: { _id: 0, nearestSellingDate: { $dateToString: { format:
"%Y/%m/%d", date: { $toDate: "$_id" } } }, totalQuantity: 1 } }
]);

// 3. Display the product name and quantity sold for the product with the
highest quantity sold
db.sales.aggregate([
  { $unwind: "$items" },
  { $group: {
      _id: "$items.name",
      totalQuantity: { $sum: "$items.quantity" }
    }
  },
```

```
  { $sort: { totalQuantity: -1 } },
  { $limit: 1 },
  { $project: { _id: 0, productName: "$_id", totalQuantity: 1 } }
]);

// 4. Display 'storeLocation', number of customers ('no_of_customers') for
each 'storeLocation' and 'purchaseMethod', sorted by 'storeLocation' and
'purchaseMethod' alphabetically from A - Z
db.sales.aggregate([
  { $group: {
      _id: { storeLocation: "$storeLocation", purchaseMethod:
"$purchaseMethod" },
      no_of_customers: { $sum: 1 }
    }
  },
  { $sort: { "_id.storeLocation": 1, "_id.purchaseMethod": 1 } },
  { $project: { _id: 0, storeLocation: "$_id.storeLocation",
purchaseMethod: "$_id.purchaseMethod", no_of_customers: 1 } }
]);

// 5. Display the number of customers by age group: 15-29, 30-44, 45-59,
60-74, 75+
db.sales.aggregate([
  { $bucket: {
      groupBy: "$customer.age",
      boundaries: [15, 30, 45, 60, 75],
      default: "75+",
      output: {
          no_of_customers: { $sum: 1 }
      }
    }
  },
  { $project: { _id: 0, age_range: "$_id", no_of_customers: 1 } }
]);

// 6. Display the number of customers (no_of_customers), average age
(avg_age), and average satisfaction (avg_satisfaction) of customers by each
store location. Round avg_age up and avg_satisfaction to one decimal place.
Sort the result by no_of_customers in descending order
db.sales.aggregate([
  { $group: {
      _id: "$storeLocation",
      no_of_customers: { $sum: 1 },
      avg_age: { $avg: "$customer.age" },
      avg_satisfaction: { $avg: "$customer.satisfaction" }
    }
  },
  { $project: { _id: 0, storeLocation: "$_id", no_of_customers: 1, avg_age:
{ $ceil: "$avg_age" }, avg_satisfaction: { $round: ["$avg_satisfaction", 1]
} } },
  { $sort: { no_of_customers: -1 } }
]);

// 7. Display the number of customers, average age, and average
```

```
satisfaction of customers who shopped at the 'New York' store by gender.
Round avg_age up and avg_satisfaction to one decimal place
db.sales.aggregate([
  { $match: { storeLocation: "New York" } },
  { $group: {
      _id: "$customer.gender",
      no_of_customers: { $sum: 1 },
      avg_age: { $avg: "$customer.age" },
      avg_satisfaction: { $avg: "$customer.satisfaction" }
    }
  },
  { $project: {
      _id: 0,
      gender: "$_id",
      no_of_customers: 1,
      avg_age: { $ceil: "$avg_age" },
      avg_satisfaction: { $round: ["$avg_satisfaction", 1] }
    }
  }
]);

// 8. Display all distinct tags in the sales collection
db.sales.aggregate([
  { $unwind: "$items" },
  { $unwind: "$items.tags" },
  { $group: { _id: "$items.tags" } },
  { $project: { _id: 0, tag: "$_id" } }
]);

// 9. Display 'saleDate', 'items.name', 'items.price', 'items.quantity',
and add a field 'items.revenue' with 'items.revenue' = 'items.price' *
'items.quantity', sort the result by 'saleDate' in descending order and
only display the top 2 results
db.sales.aggregate([
  { $unwind: "$items" },
  { $project: {
      saleDate: 1,
      "items.name": 1,
      "items.price": 1,
      "items.quantity": 1,
      "items.revenue": { $multiply: ["$items.price", "$items.quantity"] }
    }
  },
  { $sort: { saleDate: -1 } },
  { $limit: 2 }
]);

// 10. Calculate the total sales amount (totalSalesAmount) by each
'items.name'. For example, binder has a totalSalesAmount of 511644.57
db.sales.aggregate([
  { $unwind: "$items" },
  { $group: {
      _id: "$items.name",
      totalSalesAmount: { $sum: { $multiply: ["$items.price",
```

```
"$items.quantity"] } }
    }
  },
  { $project: { _id: 0, name: "$_id", totalSalesAmount: 1 } }
]);

// 11. Calculate the total sales amount by each year
db.sales.aggregate([
  { $group: {
      _id: { $year: { $toDate: "$saleDate" } },
      totalSalesAmount: { $sum: { $reduce: { input: "$items", initialValue:
0, in: { $add: ["$$value", { $multiply: ["$$this.price", "$$this.quantity"]
}] } } } } }
    }
  },
  { $project: { _id: 0, year: "$_id", totalSalesAmount: 1 } }
]);

// 12. Calculate the total quantity sold and total revenue for the product
'laptop' at the New York store
db.sales.aggregate([
  { $match: { storeLocation: "New York" } },
  { $unwind: "$items" },
  { $match: { "items.name": "laptop" } },
  { $group: {
      _id: "$items.name",
      totalQuantity: { $sum: "$items.quantity" },
      totalRevenue: { $sum: { $multiply: ["$items.price",
"$items.quantity"] } } }
    }
  },
  { $project: { _id: 0, name: "$_id", totalQuantity: 1, totalRevenue: 1 } }
]);
```

**Ouput**:

- **Question 1:**

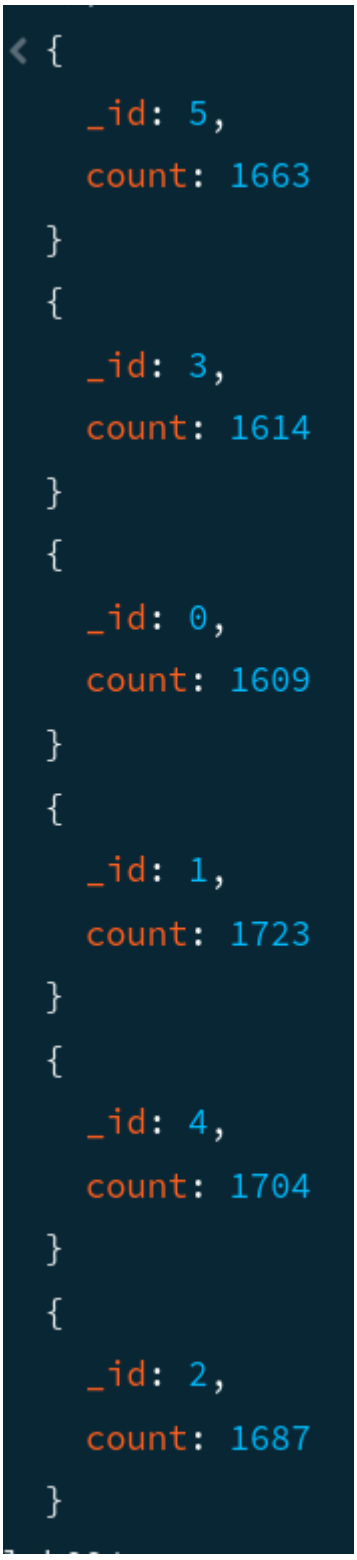
```
{
  name: "Harriet'S Kitchen",
  stars: 5
}
{
  name: 'Carvel Ice Cream',
  stars: 5
}
{
  name: 'Golden Pavillion',
  stars: 5
}
{
  name: 'P and S Deli Grocery',
  stars: 5
}
{
  name: 'Riviera Caterer',
  stars: 5
}
```

- **Question 2:**

```
{
  name: '18 Bakery',
  stars: 5
}
{
  name: '21 Bar',
  stars: 5
}
{
  name: '3 Guys Resturant',
  stars: 5
}
{
  name: '310 - Exelsior',
  stars: 5
}
{
  name: '318 - Two Boots',
  stars: 5
}
```

- **Question 8:**


{
name: 'Luigis Pizza',
stars: 0,
categories: [
'Soups and Sandwiches',
'Vietnamese',
'Chinese',
'Russian',
'Polish'
]
}


{
_id: 5,
count: 1663
}
{
_id: 3,
count: 1614
}
{
_id: 0,
count: 1609
}
{
_id: 1,
count: 1723
}
{
_id: 4,
count: 1704
}
{
_id: 2,
count: 1687
}

- **Question 15:**