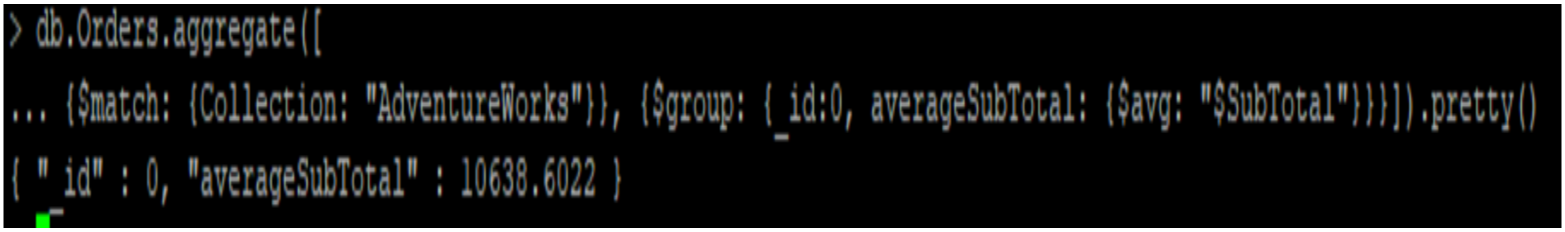
MongoDB Queries

David Chehet

Query 1:

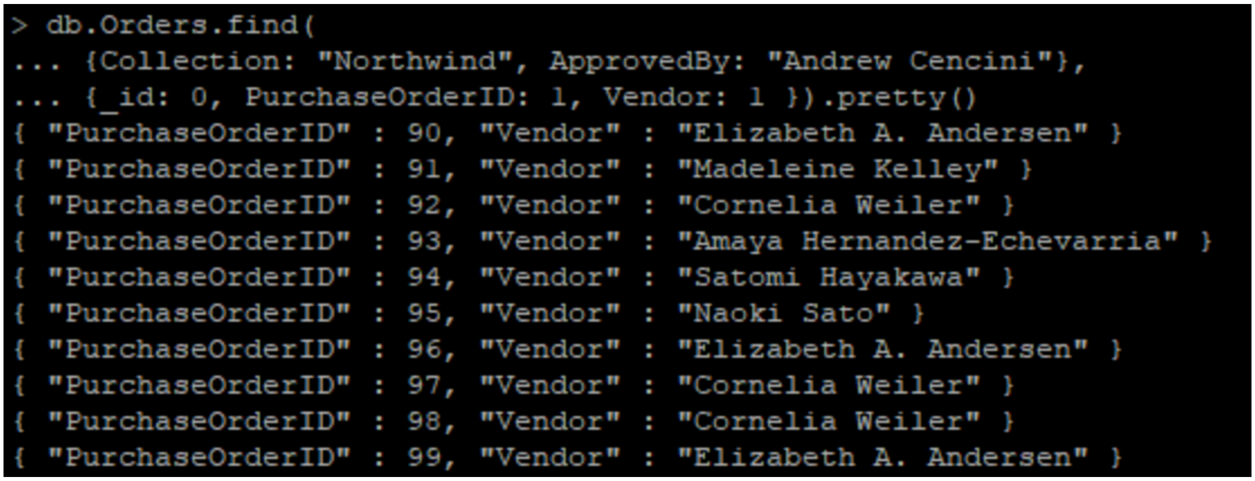
I want to see what the average ‘Subtotal’ for my orders in AdventureWorks is.



```
> db.Orders.aggregate([
... {$match: {Collection: "AdventureWorks"}}, {$group: {_id:0, averageSubTotal: {$avg: "$SubTotal"}}}]).pretty()
{ "_id" : 0, "averageSubTotal" : 10638.6022 }
```

Query 2:

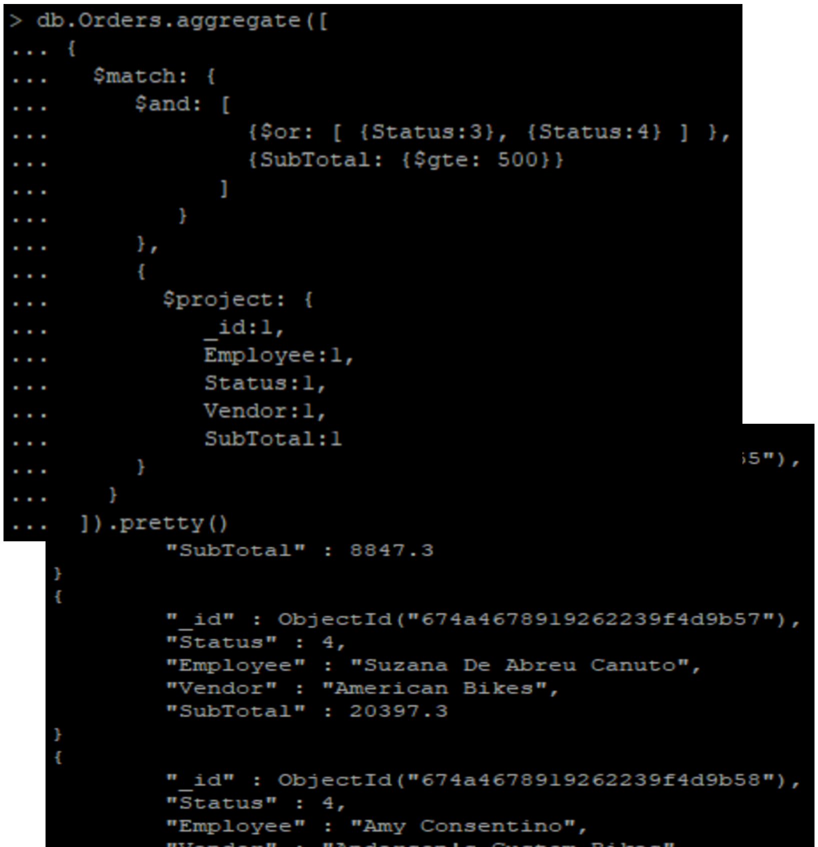
The employee from the Northwind DB, “Andrew Cencini”, sent out an order incorrectly. I’d like to see the other orders that he approved using the “ApprovedBy” value. However, I don’t want to see all the information, just the order ID and vendor.



```
> db.Orders.find(
... {Collection: "Northwind", ApprovedBy: "Andrew Cencini"},
... {_id: 0, PurchaseOrderID: 1, Vendor: 1 }).pretty()
{ "PurchaseOrderID" : 90, "Vendor" : "Elizabeth A. Andersen" }
{ "PurchaseOrderID" : 91, "Vendor" : "Madeleine Kelley" }
{ "PurchaseOrderID" : 92, "Vendor" : "Cornelia Weiler" }
{ "PurchaseOrderID" : 93, "Vendor" : "Amaya Hernandez-Echevarria" }
{ "PurchaseOrderID" : 94, "Vendor" : "Satomi Hayakawa" }
{ "PurchaseOrderID" : 95, "Vendor" : "Naoki Sato" }
{ "PurchaseOrderID" : 96, "Vendor" : "Elizabeth A. Andersen" }
{ "PurchaseOrderID" : 97, "Vendor" : "Cornelia Weiler" }
{ "PurchaseOrderID" : 98, "Vendor" : "Cornelia Weiler" }
{ "PurchaseOrderID" : 99, "Vendor" : "Elizabeth A. Andersen" }
```

Query 3:

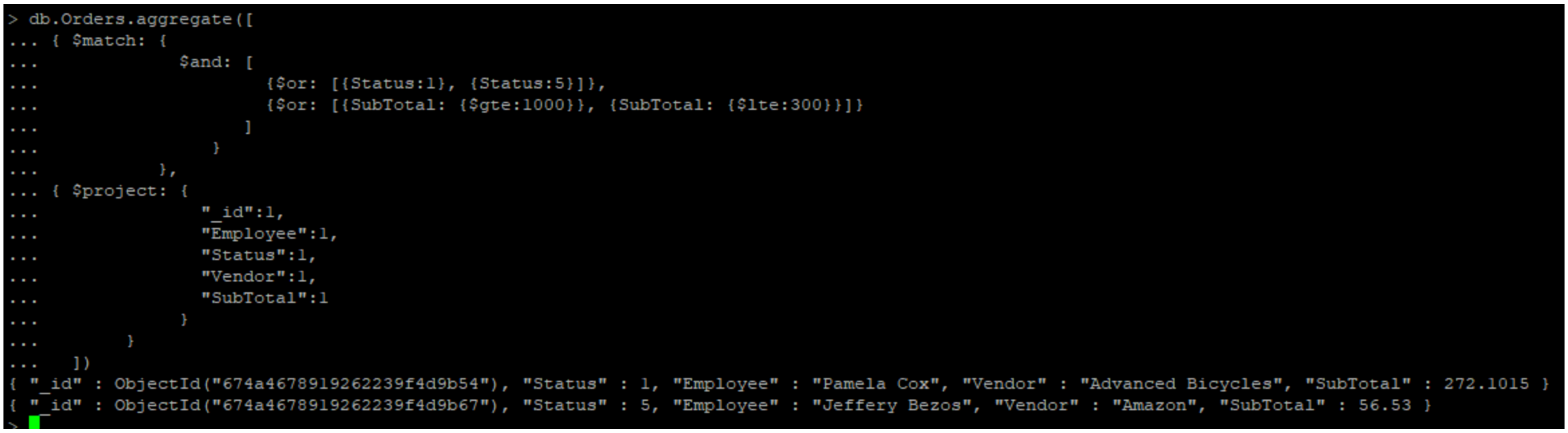
I want to make sure that we stay on top of our orders that are close to being done (Status 3 or 4). I also care more about orders with Subtotals greater than or equal to $500.



```
> db.Orders.aggregate([
... {
...   $match: {
...      $and: [
...              {$or: [ {Status:3}, {Status:4} ] },
...              {SubTotal: {$gte: 500}}
...           ]
...         }
...      },
...      {
...        $project: {
...           _id:1,
...           Employee:1,
...           Status:1,
...           Vendor:1,
...           SubTotal:1
...      }                                                     5"),
...    }
...  ]).pretty()
          "SubTotal" : 8847.3
}
{
          "_id" : ObjectId("674a4678919262239f4d9b57"),
          "Status" : 4,
          "Employee" : "Suzana De Abreu Canuto",
          "Vendor" : "American Bikes",
          "SubTotal" : 20397.3
}
{
          "_id" : ObjectId("674a4678919262239f4d9b58"),
          "Status" : 4,
          "Employee" : "Amy Consentino",
          "Vendor" : "Andersen's Custom Bikes",
```

Query 4:

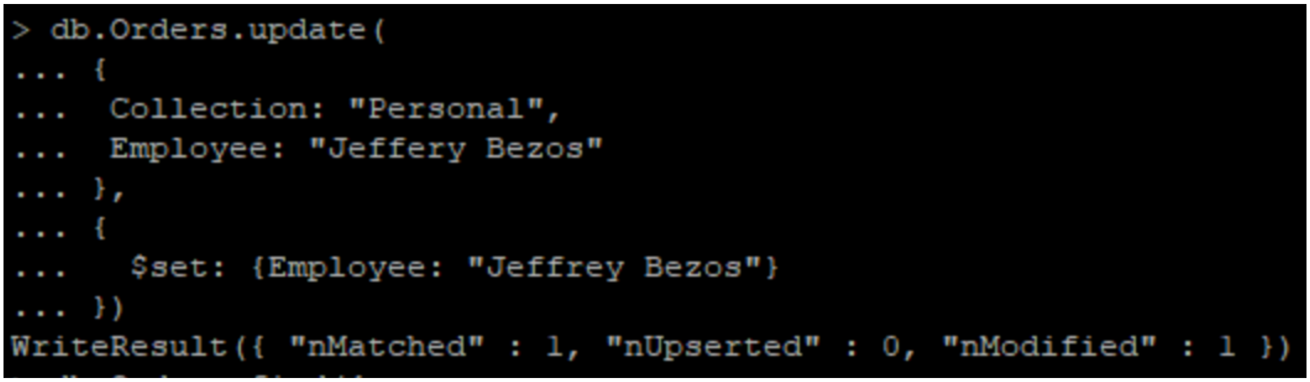
Now I want to observe the extremes of my Purchase Orders, Status 1 and Status 5. Additionally, I want to see Subtotals that are either greater than or equal to $1000, or less than or equal to $300.



```
> db.Orders.aggregate([
... { $match: {
...             $and: [
...                     {$or: [{Status:1}, {Status:5}]},
...                     {$or: [{SubTotal: {$gte:1000}}, {SubTotal: {$lte:300}}]}
...                   ]
...                }
...            },
... { $project: {
...               "_id":1,
...               "Employee":1,
...               "Status":1,
...               "Vendor":1,
...               "SubTotal":1
...             }
...        }
...    ])
{ "_id" : ObjectId("674a4678919262239f4d9b54"), "Status" : 1, "Employee" : "Pamela Cox", "Vendor" : "Advanced Bicycles", "SubTotal" : 272.1015 }
{ "_id" : ObjectId("674a4678919262239f4d9b67"), "Status" : 5, "Employee" : "Jeffery Bezos", "Vendor" : "Amazon", "SubTotal" : 56.53 }
>
```

Query 5:

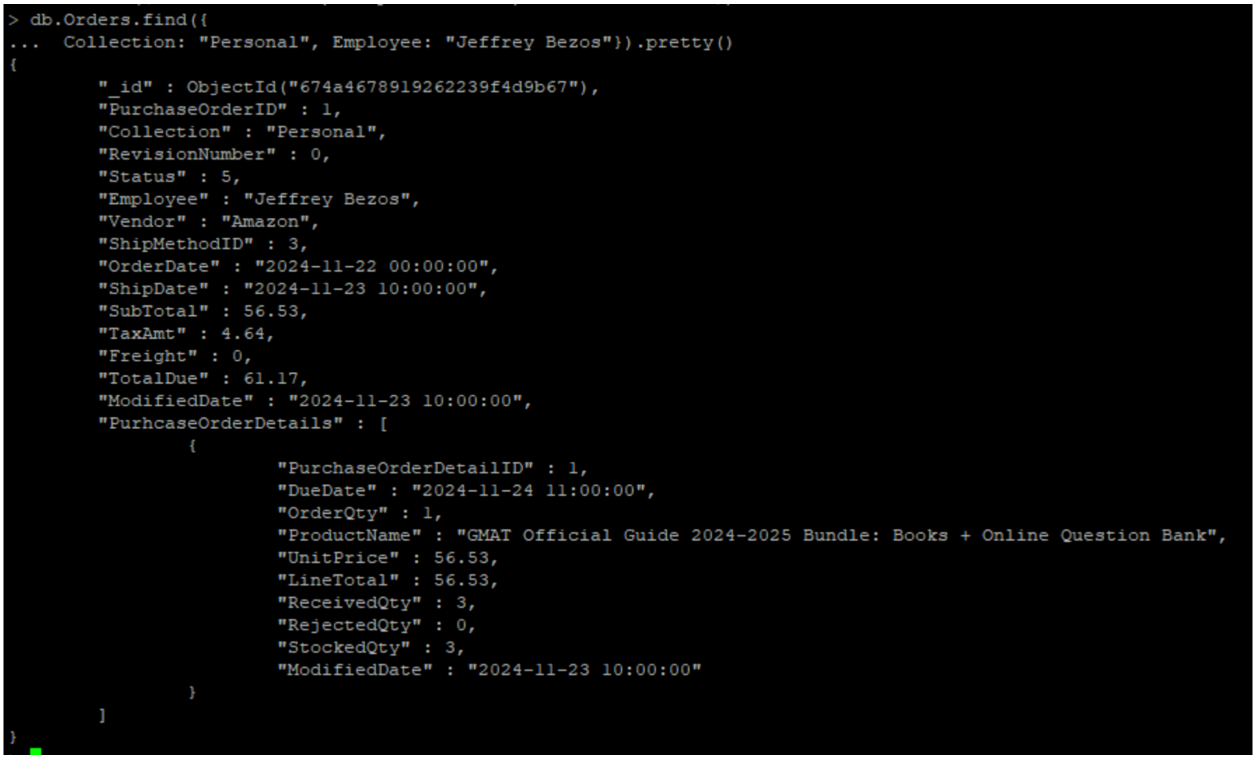
I realized that I spelled “Jeffrey Bezos” as “Jeffery Bezos”, wrong spelling for the first time. We will fix that with the query below.



```
> db.Orders.update(
... {
...  Collection: "Personal",
...  Employee: "Jeffery Bezos"
... },
... {
...   $set: {Employee: "Jeffrey Bezos"}
... })
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

Query 6:

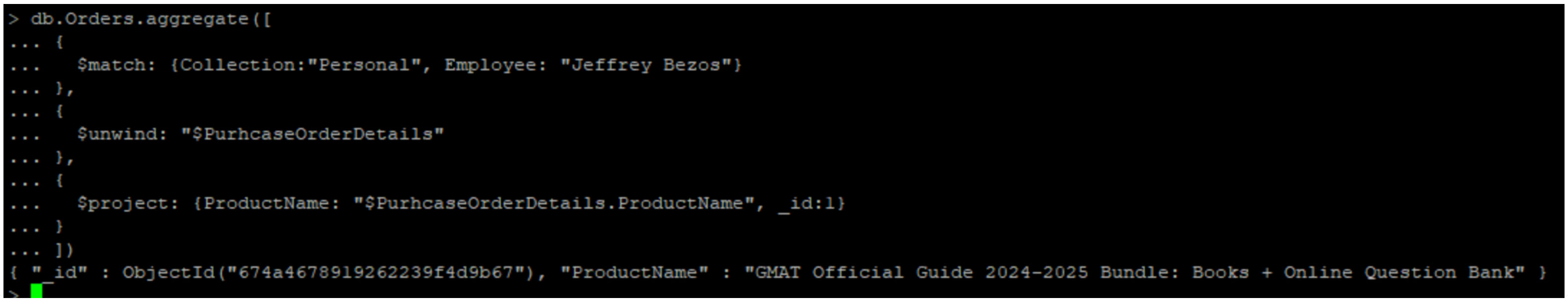
I am paranoid and want to confirm that it changed it. Perform a query to find that order and confirm the name is now correct.



```
> db.Orders.find({
...  Collection: "Personal", Employee: "Jeffrey Bezos"}).pretty()
{
        "_id" : ObjectId("674a4678919262239f4d9b67"),
        "PurchaseOrderID" : 1,
        "Collection" : "Personal",
        "RevisionNumber" : 0,
        "Status" : 5,
        "Employee" : "Jeffrey Bezos",
        "Vendor" : "Amazon",
        "ShipMethodID" : 3,
        "OrderDate" : "2024-11-22 00:00:00",
        "ShipDate" : "2024-11-23 10:00:00",
        "SubTotal" : 56.53,
        "TaxAmt" : 4.64,
        "Freight" : 0,
        "TotalDue" : 61.17,
        "ModifiedDate" : "2024-11-23 10:00:00",
        "PurhcaseOrderDetails" : [
                {
                        "PurchaseOrderDetailID" : 1,
                        "DueDate" : "2024-11-24 11:00:00",
                        "OrderQty" : 1,
                        "ProductName" : "GMAT Official Guide 2024-2025 Bundle: Books + Online Question Bank",
                        "UnitPrice" : 56.53,
                        "LineTotal" : 56.53,
                        "ReceivedQty" : 3,
                        "RejectedQty" : 0,
                        "StockedQty" : 3,
                        "ModifiedDate" : "2024-11-23 10:00:00"
                }
        ]
}
```

<u>Query 7:</u>

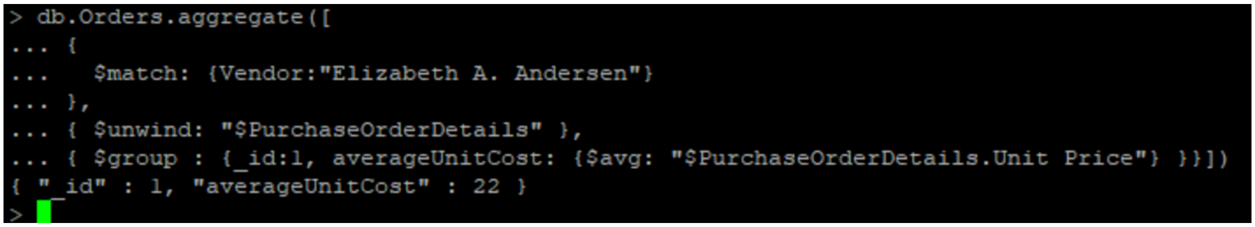
I want to run a query to confirm the ProductName of what I ordered in my personal order.



```
> db.Orders.aggregate([
... {
...   $match: {Collection:"Personal", Employee: "Jeffrey Bezos"}
... },
... {
...   $unwind: "$PurhcaseOrderDetails"
... },
... {
...   $project: {ProductName: "$PurhcaseOrderDetails.ProductName", _id:1}
... }
... ])
{ "_id" : ObjectId("674a4678919262239f4d9b67"), "ProductName" : "GMAT Official Guide 2024-2025 Bundle: Books + Online Question Bank" }
>
```

<u>Query 8</u>

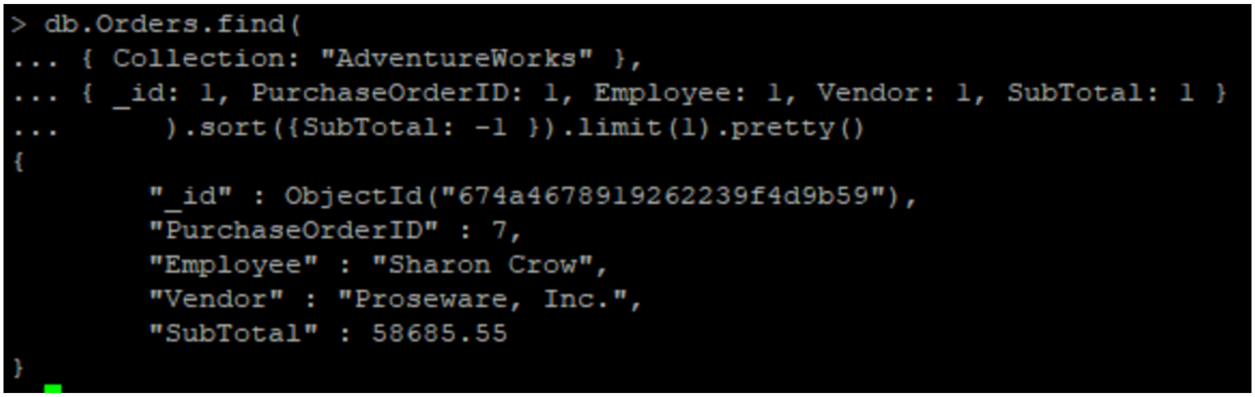
I want to calculate the average unit price of the Purchase Order Details from the Elizabeth A. Andersen vendor order.



```
> db.Orders.aggregate([
... {
...   $match: {Vendor:"Elizabeth A. Andersen"}
... },
... { $unwind: "$PurchaseOrderDetails" },
... { $group : {_id:1, averageUnitCost: {$avg: "$PurchaseOrderDetails.Unit Price"} }}])
{ "_id" : 1, "averageUnitCost" : 22 }
>
```

<u>Query 9</u>

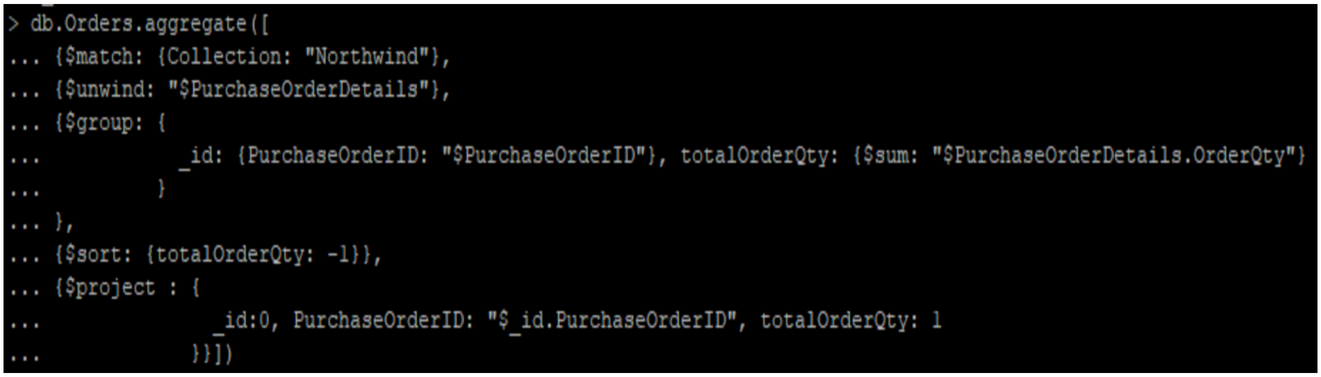
I'd like to find the largest order by Subtotal in the AdventureWorks database.



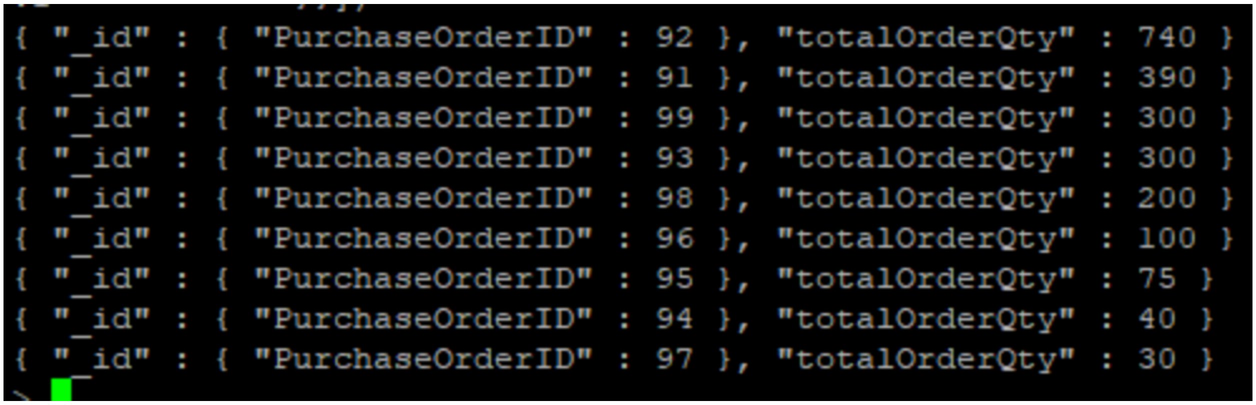
```
> db.Orders.find(
... { Collection: "AdventureWorks" },
... { _id: 1, PurchaseOrderID: 1, Employee: 1, Vendor: 1, SubTotal: 1 }
...       ).sort({SubTotal: -1 }).limit(1).pretty()
{
        "_id" : ObjectId("674a4678919262239f4d9b59"),
        "PurchaseOrderID" : 7,
        "Employee" : "Sharon Crow",
        "Vendor" : "Proseware, Inc.",
        "SubTotal" : 58685.55
}
```

Query 10

I would like to sort all of the orders in Northwind based on total number of items ordered across all PurchaseOrderDetails descending.



```
> db.Orders.aggregate([
... {$match: {Collection: "Northwind"},
... {$unwind: "$PurchaseOrderDetails"},
... {$group: {
...            _id: {PurchaseOrderID: "$PurchaseOrderID"}, totalOrderQty: {$sum: "$PurchaseOrderDetails.OrderQty"}
...          }
... },
... {$sort: {totalOrderQty: -1}},
... {$project : {
...               _id:0, PurchaseOrderID: "$_id.PurchaseOrderID", totalOrderQty: 1
...             }}])
```



```
{ "_id" : { "PurchaseOrderID" : 92 }, "totalOrderQty" : 740 }
{ "_id" : { "PurchaseOrderID" : 91 }, "totalOrderQty" : 390 }
{ "_id" : { "PurchaseOrderID" : 99 }, "totalOrderQty" : 300 }
{ "_id" : { "PurchaseOrderID" : 93 }, "totalOrderQty" : 300 }
{ "_id" : { "PurchaseOrderID" : 98 }, "totalOrderQty" : 200 }
{ "_id" : { "PurchaseOrderID" : 96 }, "totalOrderQty" : 100 }
{ "_id" : { "PurchaseOrderID" : 95 }, "totalOrderQty" : 75 }
{ "_id" : { "PurchaseOrderID" : 94 }, "totalOrderQty" : 40 }
{ "_id" : { "PurchaseOrderID" : 97 }, "totalOrderQty" : 30 }
>
```