

Dokumentowe bazy danych – MongoDB

Ćwiczenie/zadanie

Imiona i nazwiska autorów: Kacper Cienkosz, Miłosz Dubiel

Odtwórz z backupu bazę **north0**

```
mongorestore --nsInclude='north0.*' ./dump/
```

```
use north0
```

Baza **north0** jest kopią relacyjnej bazy danych **Northwind**

- poszczególne kolekcje odpowiadają tabelom w oryginalnej bazie **Northwind**

Wprowadzenie

zapoznaj się ze strukturą dokumentów w bazie **North0**

```
db.customers.find();  
db.orders.find();  
db.orderdetails.find();
```

Operacje wyszukiwania danych, przetwarzanie dokumentów

Zadanie 1

stwórz kolekcję **OrdersInfo** zawierającą następujące dane o zamówieniach

- kolekcję **OrdersInfo** należy stworzyć przekształcając dokumenty w oryginalnych kolekcjach **customers**, **orders**, **orderdetails**, **employees**, **shippers**, **products**, **categories**, **suppliers** do kolekcji w której pojedynczy dokument opisuje jedno zamówienie

```
[  
  {
```

```

    "_id": ...

    "OrderID": ... numer zamówienia

    "Customer": { ... podstawowe informacje o kliencie składającym
        "CustomerID": ... identyfikator klienta
        "CompanyName": ... nazwa klienta
        "City": ... miasto
        "Country": ... kraj
    },

    "Employee": { ... podstawowe informacje o pracowniku obsługującym
zamówienie
        "EmployeeID": ... idntyfikator pracownika
        "FirstName": ... imie
        "LastName": ... nazwisko
        "Title": ... stanowisko
    },

    "Dates": {
        "OrderDate": ... data złożenia zamówienia
        "RequiredDate": data wymaganej realizacji
    }

    "Orderdetails": [ ... pozycje/szczegóły zamówienia – tablica takich
pozycji
        {
            "UnitPrice": ... cena
            "Quantity": ... liczba sprzedanych jednostek towaru
            "Discount": ... zniżka
            "Value": ... wartość pozycji zamówienia
            "product": { ... podstawowe informacje o produkcie
                "ProductID": ... identyfikator produktu
                "ProductName": ... nazwa produktu
                "QuantityPerUnit": ... opis/opakowanie
                "CategoryID": ... identyfikator kategorii do której należy
produkt
                "CategoryName" ... nazwę tej kategorii
            },
        },
        ...
    ],

    "Freight": ... opłata za przesyłkę
    "OrderTotal" ... sumaryczna wartosc sprzedanych produktów

    "Shipment" : { ... informacja o wysyłce
        "Shipper": { ... podstawowe inf o przewoźniku
            "ShipperID":
            "CompanyName":
        }
        ... inf o odbiorcy przesyłki
        "ShipName": ...

```

```
    "ShipAddress": ...
    "ShipCity": ...
    "ShipCountry": ...
  }
}
```

Wyniki:

Zapytanie

```
db.orders.aggregate([
  // Join customers
  {
    $lookup: {
      from: "customers",
      localField: "CustomerID",
      foreignField: "CustomerID",
      as: "customer",
    },
  },
  { $unwind: "$customer" },
  // Join employees
  {
    $lookup: {
      from: "employees",
      localField: "EmployeeID",
      foreignField: "EmployeeID",
      as: "employee",
    },
  },
  { $unwind: "$employee" },
  // Join shippers
  {
    $lookup: {
      from: "shippers",
      localField: "ShipVia",
      foreignField: "ShipperID",
      as: "shipper",
    },
  },
  { $unwind: "$shipper" },
  // Join orderdetails
  {
    $lookup: {
      from: "orderdetails",
      localField: "OrderID",
      foreignField: "OrderID",
      as: "order_details",
    },
  },
])
```

```
    },
    // Unwind orderdetails for further lookups
    { $unwind: "$order_details" },
    // Join products
    {
      $lookup: {
        from: "products",
        localField: "order_details.ProductID",
        foreignField: "ProductID",
        as: "product",
      },
    },
    { $unwind: "$product" },
    // Join categories
    {
      $lookup: {
        from: "categories",
        localField: "product.CategoryID",
        foreignField: "CategoryID",
        as: "category",
      },
    },
    { $unwind: "$category" },
    // Group by order to collect order details
    {
      $group: {
        _id: "$OrderID",
        OrderID: { $first: "$OrderID" },
        Customer: {
          $first: {
            CustomerID: "$customer.CustomerID",
            CompanyName: "$customer.CompanyName",
            City: "$customer.City",
            Country: "$customer.Country",
          },
        },
        Employee: {
          $first: {
            EmployeeID: "$employee.EmployeeID",
            FirstName: "$employee.FirstName",
            LastName: "$employee.LastName",
            Title: "$employee.Title",
          },
        },
        Dates: {
          $first: {
            OrderDate: "$OrderDate",
            RequiredDate: "$RequiredDate",
          },
        },
        Freight: { $first: "$Freight" },
        Shipment: {
          $first: {
            Shipper: {
```

```

        ShipperID: "$shipper.ShipperID",
        CompanyName: "$shipper.CompanyName",
    },
    ShipName: "$ShipName",
    ShipAddress: "$ShipAddress",
    ShipCity: "$ShipCity",
    ShipCountry: "$ShipCountry",
},
},
Orderdetails: {
  $push: {
    UnitPrice: "$order_details.UnitPrice",
    Quantity: "$order_details.Quantity",
    Discount: "$order_details.Discount",
    Value: {
      $multiply: [
        "$order_details.UnitPrice",
        "$order_details.Quantity",
        { $subtract: [1, "$order_details.Discount"] },
      ],
    },
  },
  product: {
    ProductID: "$product.ProductID",
    ProductName: "$product.ProductName",
    QuantityPerUnit: "$product.QuantityPerUnit",
    CategoryID: "$category.CategoryID",
    CategoryName: "$category.CategoryName",
  },
},
},
},
},
// Calculate OrderTotal
{
  $addFields: {
    OrderTotal: {
      $sum: "$Orderdetails.Value",
    },
  },
},
// Unset _id field so that it is autogenerated
{ $unset: "_id" },
// Save the results to `OrdersInfo` collection
{ $out: "OrdersInfo" },
});

```

Zadanie 2

stwórz kolekcję **CustomerInfo** zawierającą następujące dane o każdym kliencie

- pojedynczy dokument opisuje jednego klienta

```
[
  {
    "_id": ...
    "CustomerID": ... identyfikator klienta
    "CompanyName": ... nazwa klienta
    "City": ... miasto
    "Country": ... kraj
    "Orders": [
      ... tablica zamówień klienta (oczywiście bez informacji o kliencie)
      {
        "_id": ...
        "OrderID": ... numer zamówienia
        "Employee": { ... podstawowe informacje o pracowniku obsługującym
zamówienie
          "EmployeeID": ... idntyfikator pracownika
          "FirstName": ... imie
          "LastName": ... nazwisko
          "Title": ... stanowisko
        },
        "Dates": {
          "OrderDate": ... data złożenia zamówienia
          "RequiredDate": data wymaganej realizacji
        }
        "Orderdetails": [ ... pozycje/szczegóły zamówienia – tablica
takich pozycji
          {
            "UnitPrice": ... cena
            "Quantity": ... liczba sprzedanych jednostek towaru
            "Discount": ... zniżka
            "Value": ... wartość pozycji zamówienia
            "product": { ... podstawowe informacje o produkcie
              "ProductID": ... identyfikator produktu
              "ProductName": ... nazwa produktu
              "QuantityPerUnit": ... opis/opakowanie
              "CategoryID": ... identyfikator kategorii do której należy
produkt
              "CategoryName" ... nazwę tej kategorii
            },
          },
          ...
        ],
        "Freight": ... opłata za przesyłkę
        "OrderTotal" ... sumaryczna wartosc sprzedanych produktów
        "Shipment" : { ... informacja o wysyłce
          "Shipper": { ... podstawowe inf o przewoźniku
            "ShipperID":
            "CompanyName":
          }
          ... inf o odbiorcy przesyłki
          "ShipName": ...
          "ShipAddress": ...
          "ShipCity": ...
        }
      }
    ]
  }
]
```

```
        "ShipCountry": ...
      }
    }
  ]
}
```

Wyniki:

Zapytanie:

```
db.customers.aggregate([
  // Look up orders for each customer
  {
    $lookup: {
      from: "orders",
      localField: "CustomerID",
      foreignField: "CustomerID",
      as: "Orders",
    },
  },
  { $unwind: { path: "$Orders", preserveNullAndEmptyArrays: true } },
  // Look up employee for each order
  {
    $lookup: {
      from: "employees",
      localField: "Orders.EmployeeID",
      foreignField: "EmployeeID",
      as: "Employee",
    },
  },
  { $unwind: { path: "$Employee", preserveNullAndEmptyArrays: true } },
  // Look up orderdetails for each order
  {
    $lookup: {
      from: "orderdetails",
      localField: "Orders.OrderID",
      foreignField: "OrderID",
      as: "Orderdetails",
    },
  },
  // Unwind orderdetails to join products and categories
  { $unwind: { path: "$Orderdetails", preserveNullAndEmptyArrays: true } },
  {
    $lookup: {
      from: "products",
      localField: "Orderdetails.ProductID",
      foreignField: "ProductID",
    },
  },
])
```

```
        as: "Product",
      },
    },
    { $unwind: { path: "$Product", preserveNullAndEmptyArrays: true }
  },
  {
    $lookup: {
      from: "categories",
      localField: "Product.CategoryID",
      foreignField: "CategoryID",
      as: "Category",
    },
  },
  { $unwind: { path: "$Category", preserveNullAndEmptyArrays: true }
},
// Look up shipper for each order
{
  $lookup: {
    from: "shippers",
    localField: "Orders.ShipVia",
    foreignField: "ShipperID",
    as: "Shipper",
  },
},
{ $unwind: { path: "$Shipper", preserveNullAndEmptyArrays: true }
},
// Group orderdetails per order
{
  $group: {
    _id: {
      customerId: "$CustomerID",
      orderId: "$Orders.OrderID",
    },
    CustomerID: { $first: "$CustomerID" },
    CompanyName: { $first: "$CompanyName" },
    City: { $first: "$City" },
    Country: { $first: "$Country" },
    OrderID: { $first: "$Orders.OrderID" },
    Employee: {
      $first: {
        EmployeeID: "$Employee.EmployeeID",
        FirstName: "$Employee.FirstName",
        LastName: "$Employee.LastName",
        Title: "$Employee.Title",
      },
    },
    Dates: {
      $first: {
        OrderDate: "$Orders.OrderDate",
        RequiredDate: "$Orders.RequiredDate",
      },
    },
    Freight: { $first: "$Orders.Freight" },
    Shipment: {
```



```
    $first: {
      Shipper: {
        ShipperID: "$Shipper.ShipperID",
        CompanyName: "$Shipper.CompanyName",
      },
      ShipName: "$Orders.ShipName",
      ShipAddress: "$Orders.ShipAddress",
      ShipCity: "$Orders.ShipCity",
      ShipCountry: "$Orders.ShipCountry",
    },
  },
  Orderdetails: {
    $push: {
      UnitPrice: "$Orderdetails.UnitPrice",
      Quantity: "$Orderdetails.Quantity",
      Discount: "$Orderdetails.Discount",
      Value: {
        $multiply: [
          "$Orderdetails.UnitPrice",
          "$Orderdetails.Quantity",
          { $subtract: [1, "$Orderdetails.Discount"] },
        ],
      },
    },
    product: {
      ProductID: "$Product.ProductID",
      ProductName: "$Product.ProductName",
      QuantityPerUnit: "$Product.QuantityPerUnit",
      CategoryID: "$Category.CategoryID",
      CategoryName: "$Category.CategoryName",
    },
  },
},
},
},
},
// Calculate OrderTotal
{
  $addFields: {
    OrderTotal: { $sum: "$Orderdetails.Value" },
  },
},
// Group orders per customer
{
  $group: {
    _id: "$CustomerID",
    CustomerID: { $first: "$CustomerID" },
    CompanyName: { $first: "$CompanyName" },
    City: { $first: "$City" },
    Country: { $first: "$Country" },
    Orders: {
      $push: {
        OrderID: "$OrderID",
        Employee: "$Employee",
        Dates: "$Dates",
        Orderdetails: "$Orderdetails",
      },
    },
  },
},
}
```

```
        Freight: "$Freight",
        OrderTotal: "$OrderTotal",
        Shipment: "$Shipment",
    },
},
},
},
// Unset _id field so that it is autogenerated
{ $unset: "_id" },
// Save the results to `CustomerInfo` collection
{ $out: "CustomerInfo" },
]);
```

Zadanie 3

Napisz polecenie/zapytanie: Dla każdego klienta pokaż wartość zakupionych przez niego produktów z kategorii 'Confections' w 1997r

- Spróbuj napisać to zapytanie wykorzystując
- oryginalne kolekcje (`customers`, `orders`, `orderdetails`, `products`, `categories`)
- kolekcję `OrderInfo`
- kolekcję `CustomerInfo`

Porównaj zapytania/polecenia/wyniki.

Zamieść odpowiedni komentarz

Które wersje zapytań były "prostsze"?

```
[
  {
    "_id":
      "CustomerID": ... identyfikator klienta
      "CompanyName": ... nazwa klienta
      "ConfectionsSale97": ... wartość zakupionych przez niego produktów z
kategorii 'Confections' w 1997r
  }
]
```

Wyniki:

Zapytanie w oparciu o oryginalne kolekcje (`customers`, `orders`, `orderdetails`, `products`, `categories`)

```
db.customers.aggregate([
  // Join orders for each customer
  {
    $lookup: {
      from: "orders",
      localField: "CustomerID",
      foreignField: "CustomerID",
      as: "orders",
    },
  },
  { $unwind: "$orders" },
  // Filter orders from 1997
  {
    $match: {
      "orders.OrderDate": {
        $gte: ISODate("1997-01-01T00:00:00Z"),
        $lt: ISODate("1998-01-01T00:00:00Z"),
      },
    },
  },
  // Join orderdetails for each order
  {
    $lookup: {
      from: "orderdetails",
      localField: "orders.OrderID",
      foreignField: "OrderID",
      as: "orderdetails",
    },
  },
  { $unwind: "$orderdetails" },
  // Join products for each orderdetail
  {
    $lookup: {
      from: "products",
      localField: "orderdetails.ProductID",
      foreignField: "ProductID",
      as: "product",
    },
  },
  { $unwind: "$product" },
  // Join categories for each product
  {
    $lookup: {
      from: "categories",
      localField: "product.CategoryID",
      foreignField: "CategoryID",
      as: "category",
    },
  },
  { $unwind: "$category" },
  // Filter only 'Confections' category
  {
    $match: {
```

```

        "category.CategoryName": "Confections",
    },
},
// Calculate value for each orderdetail
{
    $addFields: {
        value: {
            $multiply: [
                "$orderdetails.UnitPrice",
                "$orderdetails.Quantity",
                { $subtract: [1, "$orderdetails.Discount"] },
            ],
        },
    },
},
// Group by customer
{
    $group: {
        _id: "$CustomerID",
        CustomerID: { $first: "$CustomerID" },
        CompanyName: { $first: "$CompanyName" },
        ConfectionsSale97: { $sum: "$value" },
    },
},
// Unset _id field so that it is autogenerated
{ $unset: "_id" },
// Sort by ConfectionsSale97 descendingly
{ $sort: { ConfectionsSale97: -1 } },
]);

```

Zapytanie w oparciu o kolekcję **OrdersInfo**

```

db.OrdersInfo.aggregate([
    // Filter orders from 1997
    {
        $match: {
            "Dates.OrderDate": {
                $gte: ISODate("1997-01-01T00:00:00Z"),
                $lt: ISODate("1998-01-01T00:00:00Z"),
            },
        },
    },
    // Unwind order details
    { $unwind: "$Orderdetails" },
    // Filter only 'Confections' category
    {
        $match: {
            "Orderdetails.product.CategoryName": "Confections",
        },
    },
    // Group by customer

```

```
{
  $group: {
    _id: "$Customer.CustomerID",
    CustomerID: { $first: "$Customer.CustomerID" },
    CompanyName: { $first: "$Customer.CompanyName" },
    ConfectionsSale97: { $sum: "$Orderdetails.Value" },
  },
},
// Unset _id field so that it is autogenerated
{ $unset: "_id" },
// Sort by ConfectionsSale97 descendingly
{ $sort: { ConfectionsSale97: -1 } },
]);
```

Zapytanie w oparciu o kolekcję **CustomerInfo**

```
db.CustomerInfo.aggregate([
  // Unwind orders
  { $unwind: "$Orders" },
  // Filter orders from 1997
  {
    $match: {
      "Orders.Dates.OrderDate": {
        $gte: ISODate("1997-01-01T00:00:00Z"),
        $lt: ISODate("1998-01-01T00:00:00Z"),
      },
    },
  },
  // Unwind orderdetails
  { $unwind: "$Orders.Orderdetails" },
  // Filter only 'Confections' category
  {
    $match: {
      "Orders.Orderdetails.product.CategoryName": "Confections",
    },
  },
  // Group by customer
  {
    $group: {
      _id: "$CustomerID",
      CustomerID: { $first: "$CustomerID" },
      CompanyName: { $first: "$CompanyName" },
      ConfectionsSale97: { $sum: "$Orders.Orderdetails.Value" },
    },
  },
  // Unset _id field so that it is autogenerated
  { $unset: "_id" },
  // Sort by ConfectionsSale97 descendingly
  { $sort: { ConfectionsSale97: -1 } },
]);
```

Rezultat zapytań

Wszystkie trzy zapytania dały taki sam wynik:

CompanyName	ConfectionsSale97	CustomerID
QUICK-Stop	11648.599999657274	QUICK
Ernst Handel	9829.757463981212	ERNSH
Save-a-lot Markets	6351.084993118047	SAVEA
Old World Delicatessen	2758.375	OLDWO
Rattlesnake Canyon Grocery	2562.5	RATTC
Ottilies Käseladen	2314.024998875335	OTTIK
Folies gourmandes	2232	FOLIG
Maison Dewey	2132	MAISD
Königlich Essen	2119.399995714426	KOENE
Victuailles en stock	1972	VICTE
La maison d'Asie	1798.3999991714954	LAMAI
Mère Paillarde	1715.1249987158924	MEREP
Hungry Coyote Import Store	1701	HUNGC
Queen Cozinha	1691.4374993396923	QUEEN
Frankenversand	1678.9749984033406	FRANK
Lehmanns Marktstand	1509.1424948142842	LEHMS
Morgenstern Gesundkost	1449	MORGK
Blondesddsl père et fils	1379	BLONP
Magazzini Alimentari Riuniti	1300.0499990858139	MAGAA
Richter Supermarkt	1254.3839989572764	RICSU
Que Delícia	1235.1999998259544	QUEDE
Seven Seas Imports	1161.4999977126718	SEVES
Vaffeljernet	1064.5	VAFFE
White Clover Markets	1001.9749971002341	WHITC
LINO-Delicatesses	998.3	LINOD
Antonio Moreno Taquería	958.927493275702	ANTON
Suprêmes délices	900.3	SUPRD

CompanyName	ConfectionsSale97	CustomerID
B's Beverages	875	BSBEV
Ricardo Adocicados	855.9999993294477	RICAR
HILARION-Abastos	824.6639980454743	HILAA
Chop-suey Chinese	823.6999997019768	CHOPS
Bottom-Dollar Markets	809.399999499321	BOTTM
LILA-Supermercado	740	LILAS
Piccolo und mehr	718.65	PICCO
Simons bistro	694.8749991059303	SIMOB
France restauration	667.5	FRANR
Furia Bacalhau e Frutos do Mar	662.1499957256019	FURIB
Berglunds snabbköp	561.9599996879697	BERGS
Wellington Importadora	551.759999679029	WELLI
Hungry Owl All-Night Grocers	520	HUNGO
Trail's Head Gourmet Provisioners	493	TRAIH
Eastern Connection	480	EASTC
Bon app'	462.41249980498105	BONAP
Reggiani Caseifici	443.69999926537275	REGGC
Folk och fä HB	442.5	FOLKO
Great Lakes Food Market	407.6999993249774	GREAL
Die Wandernde Kuh	402.7999988079071	WANDK
Tradição Hipermercados	394.3999985307455	TRADH
Wartian Herkku	390.29999999999995	WARTH
Around the Horn	375.19999977201223	AROUT
Split Rail Beer & Ale	308.9999997764826	SPLIR
Rancho grande	199.39999999999998	RANCH
Wolski Zajazd	187.5	WOLZA
Familia Arquibaldo	173.27999956235288	FAMIA
Toms Spezialitäten	166.4399995431304	TOMSP
Godos Cocina Típica	145.99999991059303	GODOS
Island Trading	144.6	ISLAT

CompanyName	ConfectionsSale97	CustomerID
Franchi S.p.A.	143.3	FRANS
Gourmet Lanchonetes	142.5	GOURL
Princesa Isabel Vinhos	126	PRINI
Lonesome Pine Restaurant	125	LONEP
Océano Atlántico Ltda.	96	OCEAN
Blauer See Delikatessen	80	BLAUS
Cactus Comidas para llevar	75	CACTU
Tortuga Restaurante	64.39999999999999	TORTU
Du monde entier	60	DUMON
Spécialités du monde	52.349999999999994	SPECD
Wilman Kala	52.349999999999994	WILMK

Porównanie zapytań

Zapytanie w oparciu o oryginalne kolekcje jest oczywiście najdłuższe i najtrudniejsze do napisania, ponieważ trzeba było połączyć w nim dane z wielu kolekcji (tzn. z wszystkich wymienionych w poleceniu, czyli pięciu).

Ważną rzeczą pomagającą nam zrozumieć co się dzieje w tych zapytaniach było pisanie komentarza opisującego każdy krok, ponieważ pozwalało to patrzeć na zapytania wysokopoziomowo i "bardziej SQL-owo".

Pozostałe dwa zapytania w oparciu o kolekcje **OrdersInfo** oraz **CustomerInfo** były bardzo zbliżone do siebie, różniły się tylko jednym poziomem zagnieżdżenia danych w **CustomerInfo**, ze względu na to jakie struktury zostały zaproponowane w zadaniach 1 i 2.

Odpowiadając na pytanie *Które wersje zapytań były "prostsze"?*: zdecydowanie skłaniamy się w kierunku zapytania z **OrdersInfo**.

Zadanie 4

Napisz polecenie/zapytanie: Dla każdego klienta podaj wartość sprzedaży z podziałem na lata i miesiące

Spróbuj napisać to zapytanie wykorzystując

- oryginalne kolekcje (**customers**, **orders**, **orderdetails**, **products**, **categories**)
- kolekcję **OrderInfo**
- kolekcję **CustomerInfo**

Porównaj zapytania/polecenia/wyniki.

Zamieść odpowiedni komentarz

Które wersje zapytań były "prostsze"?

```
[
  {
    "_id":

    "CustomerID": ... identyfikator klienta
    "CompanyName": ... nazwa klienta

    "Sale": [ ... tablica zawierająca inf o sprzedaży
      {
        "Year": ....
        "Month": ....
        "Total": ...
      }
      ...
    ]
  }
]
```

Wyniki:

Zapytanie w oparciu o oryginalne kolekcje (**customers**, **orders**, **orderdetails**)

```
db.customers.aggregate([
  // Join orders for each customer
  {
    $lookup: {
      from: "orders",
      localField: "CustomerID",
      foreignField: "CustomerID",
      as: "orders",
    },
  },
  { $unwind: "$orders" },
  // Join orderdetails for each order
  {
    $lookup: {
      from: "orderdetails",
      localField: "orders.OrderID",
      foreignField: "OrderID",
      as: "orderdetails",
    },
  },
  { $unwind: "$orderdetails" },
  // Calculate value for each orderdetail
  {
```

```
$addField: {
  value: {
    $multiply: [
      "$orderdetails.UnitPrice",
      "$orderdetails.Quantity",
      { $subtract: [1, "$orderdetails.Discount"] },
    ],
  },
},
// Group by customer, year, and month
{
  $group: {
    _id: {
      CustomerID: "$CustomerID",
      CompanyName: "$CompanyName",
      Year: { $year: "$orders.OrderDate" },
      Month: { $month: "$orders.OrderDate" },
    },
    Total: { $sum: "$value" },
  },
},
// Group by customer and collect sales per year/month
{
  $group: {
    _id: { CustomerID: "$_id.CustomerID", CompanyName:
"$_id.CompanyName" },
    Sale: {
      $push: {
        Year: "$_id.Year",
        Month: "$_id.Month",
        Total: "$Total",
      },
    },
  },
},
// Project the required fields
{
  $project: {
    _id: 0, // this makes the _id autogenerated
    CustomerID: "$_id.CustomerID",
    CompanyName: "$_id.CompanyName",
    Sale: 1,
  },
},
// Sort by CustomerID
{ $sort: { CustomerID: 1 } },
]);
```

Zapytanie w oparciu o kolekcję **OrdersInfo**

```
db.OrdersInfo.aggregate([
  // Unwind order details
  { $unwind: "$Orderdetails" },
  // Group by customer, year, and month
  {
    $group: {
      _id: {
        CustomerID: "$Customer.CustomerID",
        CompanyName: "$Customer.CompanyName",
        Year: { $year: "$Dates.OrderDate" },
        Month: { $month: "$Dates.OrderDate" },
      },
      Total: { $sum: "$Orderdetails.Value" },
    },
  },
  // Group by customer and collect sales per year/month
  {
    $group: {
      _id: { CustomerID: "$_id.CustomerID", CompanyName:
"$_id.CompanyName" },
      Sale: {
        $push: {
          Year: "$_id.Year",
          Month: "$_id.Month",
          Total: "$Total",
        },
      },
    },
  },
  // Project the required fields
  {
    $project: {
      _id: 0, // this makes the _id autogenerated
      CustomerID: "$_id.CustomerID",
      CompanyName: "$_id.CompanyName",
      Sale: 1,
    },
  },
  // Sort by CustomerID
  { $sort: { CustomerID: 1 } },
]);
```

Zapytanie w oparciu o kolekcję **CustomerInfo**

```
db.CustomerInfo.aggregate([
  // Unwind orders
  { $unwind: "$Orders" },
  // Unwind orderdetails
  { $unwind: "$Orders.Orderdetails" },
  // Group by customer, year, and month
```

```
{
  $group: {
    _id: {
      CustomerID: "$CustomerID",
      CompanyName: "$CompanyName",
      Year: { $year: "$Orders.Dates.OrderDate" },
      Month: { $month: "$Orders.Dates.OrderDate" },
    },
    Total: { $sum: "$Orders.Orderdetails.Value" },
  },
},
// Group by customer and collect sales per year/month
{
  $group: {
    _id: { CustomerID: "$_id.CustomerID", CompanyName:
"$_id.CompanyName" },
    Sale: {
      $push: {
        Year: "$_id.Year",
        Month: "$_id.Month",
        Total: "$Total",
      },
    },
  },
},
// Project the required fields
{
  $project: {
    _id: 0,
    CustomerID: "$_id.CustomerID",
    CompanyName: "$_id.CompanyName",
    Sale: 1,
  },
},
// Sort by CustomerID
{ $sort: { CustomerID: 1 } },
]);
```

Rezultat zapytań

Wszystkie trzy zapytania dały taki sam wynik – nie będziemy go tutaj wklejać w pełnej formie, ponieważ zajmuje za dużo miejsca.

Wkleimy tylko pierwszy wynik posortowany alfabetycznie po **CustomerID**.

```
{
  "CompanyName": "Alfreds Futterkiste",
  "CustomerID": "ALFKI",
  "Sale": [
    {
      "Year": 1997,
```

```
    "Month": 10,
    "Total": 1208
  },
  {
    "Year": 1998,
    "Month": 3,
    "Total": 471.19999970197676
  },
  {
    "Year": 1997,
    "Month": 8,
    "Total": 814.5
  },
  {
    "Year": 1998,
    "Month": 1,
    "Total": 845.799999922514
  },
  {
    "Year": 1998,
    "Month": 4,
    "Total": 933.4999996051192
  }
]
```

Porównanie zapytań

Tak jak w poprzednim zadaniu, zapytanie w oparciu o oryginalne kolekcje (tym razem tylko trzy: `customers`, `orders`, `orderdetails`) jest najdłuższe i najbardziej skomplikowane, ponieważ trzeba wykonać dużo więcej operacji - a szczególnie łączenia kolekcji - aby uzyskać dostęp do interesujących nas danych.

Jeżeli chodzi o zapytania z użyciem kolekcji `OrdersInfo` oraz `CustomerInfo`, to znowu możemy powtórzyć wniosek z poprzedniego zadania – te dwa zapytania były prawie identyczne, różniły się tylko jednym poziomem, którym jest wyciągnięcie danych z atrybutu `Orders` w `CustomerInfo`.

Warto tutaj zaznaczyć, że etapy:

- grupowania danych najpierw po `CustomerID`, `CompanyName`, `Year`, `Month` a następnie po `CustomerID`, `CompanyName`
- obliczania miesięcznej wartości sprzedaży
- projekcji przetworzonych danych

Powtarzają się w niezmienionej formie we wszystkich trzech zapytaniach. Natomiast główną różnicą pomiędzy nimi jest sposób na wydobycie wszystkich potrzebnych danych do właśnie tego "wspólnego" etapu agregacji.

Odpowiadając na pytanie *Które wersje zapytań były "prostsze"*?: w tym przypadku również twierdzimy, że najprostsze było zapytanie z użyciem kolekcji `OrdersInfo`.

Punktacja:

zadanie	pkt
1	3
2	3
3	3
4	3
razem	12