

Dokumentowe bazy danych – MongoDB

Ćwiczenie/zadanie

Imiona i nazwiska autorów: Wiktoria Zalińska, Magdalena Wilk

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": ...
```

```

    }
  }
]

```

Polecenia tworzące kolekcję **OrdersInfo**:

UWAGA: przy zapisie VS Code dodaje dodatkowe przecinki po nawiasach }

```

db.orders.aggregate([
  {
    $lookup: {
      from: "customers",
      localField: "CustomerID",
      foreignField: "CustomerID",
      as: "customer",
    },
  },
  { $unwind: "$customer" },

  {
    $lookup: {
      from: "orderdetails",
      localField: "OrderID",
      foreignField: "OrderID",
      as: "orderdetails",
    },
  },

  {
    $addFields: {
      Orderdetails: {
        $map: {
          input: "$orderdetails",
          as: "item",
          in: {
            UnitPrice: "$$item.UnitPrice",
            Quantity: "$$item.Quantity",
            Discount: "$$item.Discount",
            Value: {
              $multiply: [
                "$$item.UnitPrice",
                "$$item.Quantity",
                { $subtract: [1, "$$item.Discount"] },
              ],
            },
          },
          ProductID: "$$item.ProductID",
        },
      },
    },
  },
],
)

```

```
{
  $addFields: {
    OrderTotal: {
      $sum: {
        $map: {
          input: "$Orderdetails",
          as: "item",
          in: "$$item.Value",
        },
      },
    },
  },
},

{
  $lookup: {
    from: "employees",
    localField: "EmployeeID",
    foreignField: "EmployeeID",
    as: "employee",
  },
},
{ $unwind: "$employee" },

{
  $lookup: {
    from: "shippers",
    localField: "ShipVia",
    foreignField: "ShipperID",
    as: "shipper",
  },
},
{ $unwind: "$shipper" },

{
  $lookup: {
    from: "products",
    localField: "Orderdetails.ProductID",
    foreignField: "ProductID",
    as: "ProductData",
  },
},
{
  $lookup: {
    from: "categories",
    localField: "ProductData.CategoryID",
    foreignField: "CategoryID",
    as: "CategoryData",
  },
},
{
  $addFields: {
```

```

Orderdetails: {
  $map: {
    input: "$Orderdetails",
    as: "item",
    in: {
      UnitPrice: "$$item.UnitPrice",
      Quantity: "$$item.Quantity",
      Discount: "$$item.Discount",
      Value: "$$item.Value",
      Product: {
        $let: {
          vars: {
            prod: {
              $arrayElemAt: [
                {
                  $filter: {
                    input: "$ProductData",
                    as: "p",
                    cond: { $eq: ["$$p.ProductID", "$$item.ProductID"] },
                  },
                },
                0,
              ],
            },
          },
          in: {
            ProductID: "$$prod.ProductID",
            ProductName: "$$prod.ProductName",
            QuantityPerUnit: "$$prod.QuantityPerUnit",
            CategoryID: "$$prod.CategoryID",
            CategoryName: {
              $let: {
                vars: {
                  cat: {
                    $arrayElemAt: [
                      {
                        $filter: {
                          input: "$CategoryData",
                          as: "c",
                          cond: {
                            $eq: ["$$c.CategoryID", "$$prod.CategoryID"],
                          },
                        },
                      },
                      0,
                    ],
                  },
                },
                in: "$$cat.CategoryName",
              },
            },
          },
        },
      },
    },
  },
},

```

```

    },
  },
},
},
{
  $project: {
    _id: 1,
    OrderID: "$OrderID",
    Customer: {
      CustomerID: "$customer.CustomerID",
      CompanyName: "$customer.CompanyName",
      City: "$customer.City",
      Country: "$customer.Country",
    },

    Employee: {
      EmployeeID: "$employee.EmployeeID",
      FirstName: "$employee.FirstName",
      LastName: "$employee.LastName",
      Title: "$employee.Title",
    },

    Dates: {
      OrderDate: "$OrderDate",
      RequiredDate: "$RequiredDate",
    },

    Orderdetails: "$Orderdetails",

    Freight: "$Freight",
    OrderTotal: "$OrderTotal",
    Shipment: {
      Shipper: {
        ShipperID: "$shipper.ShipperID",
        CompanyName: "$shipper.CompanyName",
      },
      ShipName: "$ShipName",
      ShipAddress: "$ShipAddress",
      ShipCity: "$ShipCity",
      ShipCountry: "$ShipCountry",
    },
  },
},
{
  $out: "OrdersInfo",
},
]);

```

Przykładowy wynik:

```
{
  "_id": {"$oid": "63a060b9bb3b972d6f4e1fcb"},
  "Customer": {
    "CustomerID": "HANAR",
    "CompanyName": "Hanari Carnes",
    "City": "Rio de Janeiro",
    "Country": "Brazil"
  },
  "Dates": {
    "OrderDate": {"$date": "1996-07-10T00:00:00.000Z"},
    "RequiredDate": {"$date": "1996-07-24T00:00:00.000Z"}
  },
  "Employee": {
    "EmployeeID": 3,
    "FirstName": "Janet",
    "LastName": "Leverling",
    "Title": "Sales Representative"
  },
  "Freight": 58.17,
  "OrderID": 10253,
  "OrderTotal": 1444.8000000000002,
  "Orderdetails": [
    {
      "UnitPrice": 10,
      "Quantity": 20,
      "Discount": 0,
      "Value": 200,
      "Product": {
        "ProductID": 31,
        "ProductName": "Gorgonzola Telino",
        "QuantityPerUnit": "12 - 100 g pkgs",
        "CategoryID": 4,
        "CategoryName": "Dairy Products"
      }
    },
    {
      "UnitPrice": 14.4,
      "Quantity": 42,
      "Discount": 0,
      "Value": 604.8000000000001,
      "Product": {
        "ProductID": 39,
        "ProductName": "Chartreuse verte",
        "QuantityPerUnit": "750 cc per bottle",
        "CategoryID": 1,
        "CategoryName": "Beverages"
      }
    },
    {
      "UnitPrice": 16,
      "Quantity": 40,
      "Discount": 0,
      "Value": 640,
```

```
    "Product": {
      "ProductID": 49,
      "ProductName": "Maxilaku",
      "QuantityPerUnit": "24 - 50 g pkgs.",
      "CategoryID": 3,
      "CategoryName": "Confections"
    }
  },
  "Shipment": {
    "Shipper": {
      "ShipperID": 2,
      "CompanyName": "United Package"
    },
    "ShipName": "Hanari Carnes",
    "ShipAddress": "Rua do Paço, 67",
    "ShipCity": "Rio de Janeiro",
    "ShipCountry": "Brazil"
  }
}
```

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 o strukturze takiej jak w punkcie a)
    (oczywiście bez informacji o kliencie)

  ]
]
```

Polecenie tworzące kolekcję **CustomerInfo**:

```
db.OrdersInfo.aggregate([
  {
```



```

    $group: {
      _id: "$Customer.CustomerID",
      CustomerID: { $first: "$Customer.CustomerID" },
      CompanyName: { $first: "$Customer.CompanyName" },
      City: { $first: "$Customer.City" },
      Country: { $first: "$Customer.Country" },
      Orders: {
        $push: {
          OrderID: "$OrderID",
          Employee: "$Employee",
          Dates: "$Dates",
          Orderdetails: "$Orderdetails",
          Freight: "$Freight",
          OrderTotal: "$OrderTotal",
          Shipment: "$Shipment",
        },
      },
    },
  },
},
{
  $project: {
    _id: 1,
    CustomerID: 1,
    CompanyName: 1,
    City: 1,
    Country: 1,
    Orders: 1,
  },
},
{
  $out: "CustomerInfo",
},
]);

```

Przykładowy wynik (a przynajmniej część - 1 zamówienie dla 1 klienta):

```

[
  {
    "_id": "BLONP",
    "City": "Strasbourg",
    "CompanyName": "Blondesddsl père et fils",
    "Country": "France",
    "CustomerID": "BLONP",
    "Orders": [
      {
        "OrderID": 10265,
        "Employee": {
          "EmployeeID": 2,
          "FirstName": "Andrew",
          "LastName": "Fuller",
          "Title": "Vice President, Sales"
        },

```

```
"Dates": {
  "OrderDate": {"$date": "1996-07-25T00:00:00.000Z"},
  "RequiredDate": {"$date": "1996-08-22T00:00:00.000Z"}
},
"Orderdetails": [
  {
    "UnitPrice": 31.2,
    "Quantity": 30,
    "Discount": 0,
    "Value": 936,
    "Product": {
      "ProductID": 17,
      "ProductName": "Alice Mutton",
      "QuantityPerUnit": "20 - 1 kg tins",
      "CategoryID": 6,
      "CategoryName": "Meat/Poultry"
    }
  },
  {
    "UnitPrice": 12,
    "Quantity": 20,
    "Discount": 0,
    "Value": 240,
    "Product": {
      "ProductID": 70,
      "ProductName": "Outback Lager",
      "QuantityPerUnit": "24 - 355 ml bottles",
      "CategoryID": 1,
      "CategoryName": "Beverages"
    }
  }
],
"Freight": 55.28,
"OrderTotal": 1176,
"Shipment": {
  "Shipper": {
    "ShipperID": 1,
    "CompanyName": "Speedy Express"
  },
  "ShipName": "Blondel père et fils",
  "ShipAddress": "24, place Kléber",
  "ShipCity": "Strasbourg",
  "ShipCountry": "France"
}
},
```

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

  }
]
```

1. Zapytanie do oryginalnych kolekcji:

```
db.orders.aggregate([
  {
    $match: {
      $expr: {
        $eq: [{ $year: "$OrderDate" }, 1997],
      },
    },
  },
  {
    $lookup: {
      from: "customers",
      localField: "CustomerID",
      foreignField: "CustomerID",
      as: "customer",
    },
  },
  { $unwind: "$customer" },
  {
    $lookup: {
      from: "orderdetails",
      localField: "OrderID",
```

```
        foreignField: "OrderID",
        as: "orderdetails",
    },
},
{ $unwind: "$orderdetails" },

{
    $lookup: {
        from: "products",
        localField: "orderdetails.ProductID",
        foreignField: "ProductID",
        as: "product",
    },
},
{ $unwind: "$product" },

{
    $lookup: {
        from: "categories",
        localField: "product.CategoryID",
        foreignField: "CategoryID",
        as: "category",
    },
},
{ $unwind: "$category" },

{
    $match: {
        "category.CategoryName": "Confections",
    },
},

{
    $group: {
        _id: "$customer.CustomerID",
        CustomerID: { $first: "$customer.CustomerID" },
        CompanyName: { $first: "$customer.CompanyName" },
        ConfectionsSale97: {
            $sum: {
                $multiply: [
                    "$orderdetails.UnitPrice",
                    "$orderdetails.Quantity",
                    { $subtract: [1, "$orderdetails.Discount"] },
                ],
            },
        },
    },
},
]);
```

Przykładowy wynik:

```
{
  "_id": "OTTIK",
  "CompanyName": "Ottilies K seladen",
  "ConfectionsSale97": 2314. 24998875335,
  "CustomerID": "OTTIK"
}
```

2. Zapytanie do kolekcji OrderInfo:

```
db.OrdersInfo.aggregate([
  {
    $match: {
      $expr: {
        $eq: [{ $year: "$Dates.OrderDate" }, 1997],
      },
    },
  },
  {
    $project: {
      CustomerID: "$Customer.CustomerID",
      CompanyName: "$Customer.CompanyName",
      Orderdetails: 1,
    },
  },
  {
    $addFields: {
      ConfectionsSale97: {
        $sum: {
          $map: {
            input: {
              $filter: {
                input: "$Orderdetails",
                as: "od",
                cond: {
                  $eq: ["$$od.Product.CategoryName", "Confections"],
                },
              },
            },
            as: "confection",
            in: "$$confection.Value",
          },
        },
      },
    },
  },
  {
    $group: {
      _id: "$CustomerID",
      CustomerID: { $first: "$CustomerID" },
      CompanyName: { $first: "$CompanyName" },
      ConfectionsSale97: { $sum: "$ConfectionsSale97" },
    },
  },
])
```

```

    },
  },
  {
    $project: {
      _id: 1,
      CustomerID: 1,
      CompanyName: 1,
      ConfectionsSale97: 1,
    },
  },
]);

```

Przykładowy wynik:

```

{
  "_id": "PRINI",
  "CompanyName": "Princesa Isabel Vinhos",
  "ConfectionsSale97": 126,
  "CustomerID": "PRINI"
}

```

3. Zapytanie do kolekcji CustomerInfo:

```

db.CustomerInfo.aggregate([
  {
    $project: {
      CustomerID: 1,
      CompanyName: 1,
      Orders: {
        $filter: {
          input: "$Orders",
          as: "order",
          cond: {
            $eq: [{ $year: "$$order.Dates.OrderDate" }, 1997],
          },
        },
      },
    },
  },
  {
    $addFields: {
      ConfectionsSale97: {
        $sum: {
          $map: {
            input: "$Orders",
            as: "order",
            in: {
              $sum: {
                $map: {

```

```

        input: {
          $filter: {
            input: "$$order.Orderdetails",
            as: "od",
            cond: {
              $eq: ["$$od.Product.CategoryName", "Confections"],
            },
          },
        },
      ],
    },
  ],
},
{
  $project: {
    _id: 1,
    CustomerID: 1,
    CompanyName: 1,
    ConfectionsSale97: 1,
  },
},
]);
```

Przykładowy wynik:

```
{
  "_id": "BOTTM",
  "CompanyName": "Bottom-Dollar Markets",
  "ConfectionsSale97": 809.399999499321,
  "CustomerID": "BOTTM"
}
```

Oryginalne kolekcje:

- najdłuższe zapytanie, ale wydaje się stosunkowo proste
- wymaga dołączenia wszystkich potrzebnych tabel
- jasno widać co sumujemy

Użycie OrderInfo lub CustomerInfo:

- zapytania są krótsze, bo wszystkie dane są w jednym dokumencie

- niestety tworzenie pola ConfectionsSale97 jest dość skomplikowane, nie jest przejrzyste tak jak przy oryginalnych kolekcjach
- trudniejsze do zrozumienia dla początkujących

Zadanie 4

Napisz polecenie/zapytanie: Dla każdego klienta podaje 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": ...
      }
      ...
    ]
  }
]
```

1. Użycie oryginalnych kolekcji:

```
db.customers.aggregate([
  // Dołączenie orders
  {
    $lookup: {
      from: "orders",
      localField: "CustomerID",
      foreignField: "CustomerID",
      as: "Orders",
    },
  },
  { $unwind: "$Orders" },

  // Dołączenie orderdetails
```



```
{
  $lookup: {
    from: "orderdetails",
    localField: "Orders.OrderID",
    foreignField: "OrderID",
    as: "OrderDetails",
  },
},
{ $unwind: "$OrderDetails" },

// Obliczenie wartości jednej pozycji zamówienia
{
  $addFields: {
    OrderValue: {
      $multiply: [
        "$OrderDetails.UnitPrice",
        "$OrderDetails.Quantity",
        { $subtract: [1, "$OrderDetails.Discount"] },
      ],
    },
  },
},

// Dodanie informacji o roku i miesiącu
{
  $addFields: {
    Year: { $year: "$Orders.OrderDate" },
    Month: { $month: "$Orders.OrderDate" },
  },
},

// Grupowanie po kliencie, roku i miesiącu
{
  $group: {
    _id: {
      CustomerID: "$CustomerID",
      CompanyName: "$CompanyName",
      Year: "$Year",
      Month: "$Month",
    },
    TotalSales: { $sum: "$OrderValue" },
  },
},

// Grupowanie wyników dla każdego klienta
{
  $group: {
    _id: "$_id.CustomerID",
    CustomerID: { $first: "$_id.CustomerID" },
    CompanyName: { $first: "$_id.CompanyName" },
    Sales: {
      $push: {
        Year: "$_id.Year",
        Month: "$_id.Month",

```

```

        Total: "$TotalSales",
      },
    },
  },
},
{
  $project: {
    _id: 1,
    CustomerID: 1,
    CompanyName: 1,
    Sales: 1,
  },
},
]);

```

Przykładowy wynik:

```

[
  {
    "_id": "ALFKI",
    "CompanyName": "Alfreds Futterkiste",
    "CustomerID": "ALFKI",
    "Sales": [
      {
        "Year": 1998,
        "Month": 4,
        "Total": 933.4999996051192
      },
      {
        "Year": 1998,
        "Month": 1,
        "Total": 845.799999922514
      },
      {
        "Year": 1997,
        "Month": 8,
        "Total": 814.5
      },
      {
        "Year": 1998,
        "Month": 3,
        "Total": 471.19999970197676
      },
      {
        "Year": 1997,
        "Month": 10,
        "Total": 1208
      }
    ]
  }
],

```

2. Użycie `OrderInfo`:

```
db.OrdersInfo.aggregate([
  {
    $addFields: {
      Year: { $year: "$Dates.OrderDate" },
      Month: { $month: "$Dates.OrderDate" },
    },
  },
  {
    $group: {
      _id: {
        CustomerID: "$Customer.CustomerID",
        CompanyName: "$Customer.CompanyName",
        Year: "$Year",
        Month: "Month",
      },
      Total: { $sum: "$OrderTotal" },
    },
  },
  {
    $group: {
      _id: "$_id.CustomerID",
      CustomerID: { $first: "$_id.CustomerID" },
      CompanyName: { $first: "$_id.CompanyName" },
      Sale: {
        $push: {
          Year: "$_id.Year",
          Month: "$_id.Month",
          Total: "$Total",
        },
      },
    },
  },
  {
    $project: {
      _id: 1,
      CustomerID: 1,
      CompanyName: 1,
      Sale: 1,
    },
  },
]);
```

Przykładowy wynik:

```
[
  {
    "_id": "LILAS",
```

```

"CompanyName": "LILA-Supermercado",
"CustomerID": "LILAS",
"Sale": [
  {
    "Year": 1996,
    "Month": "Month",
    "Total": 5394.079985570907
  },
  {
    "Year": 1997,
    "Month": "Month",
    "Total": 5175.199989449978
  },
  {
    "Year": 1998,
    "Month": "Month",
    "Total": 5507.319994567037
  }
]
},

```

3. Użycie CustomerInfo:

```

db.CustomerInfo.aggregate([
  {
    $unwind: "$Orders",
  },
  {
    $addFields: {
      Year: { $year: "$Orders.Dates.OrderDate" },
      Month: { $month: "$Orders.Dates.OrderDate" },
      OrderTotal: "$Orders.OrderTotal",
    },
  },
  {
    $group: {
      _id: {
        CustomerID: "$CustomerID",
        CompanyName: "$CompanyName",
        Year: "$Year",
        Month: "$Month",
      },
      Total: { $sum: "$OrderTotal" },
    },
  },
  {
    $group: {
      _id: "$_id.CustomerID",
      CustomerID: { $first: "$_id.CustomerID" },
      CompanyName: { $first: "$_id.CompanyName" },
      Sale: {

```

```
        $push: {
          Year: "$_id.Year",
          Month: "$_id.Month",
          Total: "$Total",
        },
      },
    },
  },
  {
    $project: {
      _id: 1,
      CustomerID: 1,
      CompanyName: 1,
      Sale: 1,
    },
  },
]);
```

Przykładowy wynik:

```
[
  {
    "_id": "GROSR",
    "CompanyName": "GROSELLA-Restaurante",
    "CustomerID": "GROSR",
    "Sale": [
      {
        "Year": 1997,
        "Month": 12,
        "Total": 387.5
      },
      {
        "Year": 1996,
        "Month": 7,
        "Total": 1101.2
      }
    ]
  }
],
```

Oryginalne kolekcje:

- najbardziej złożone zapytanie
- mamy tutaj najwięcej kontroli nad obliczeniami
- wymaga ręcznego przeliczania wartości zamówień

Użycie OrderInfo:

- zapytanie dużo prostsze niż w wersji z oryginalnymi kolekcjami

- używa gotowego pola OrderTotal

Użycie CustomerInfo:

- proste i czytelne jak przy użyciu OrderInfo

Punktacja:

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