

Dokumentowe bazy danych – MongoDB

Imiona i nazwiska autorów:

Radosław Szepielak, Kacper Wdowiak

Odtwórz z backupu bazę north0

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

```
use north0
```

Zadanie 1 - operacje wyszukiwania danych, przetwarzanie dokumentów

a)

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

- 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": ... identyfikator 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 wartość sprzedanych produktów

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

```

b)

stwórz kolekcję **CustomerInfo** zawierającą następujące dane kazdym klientie

- 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)

    ]

]

```

c)

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

```

[
  {
    "_id":

    "CustomerID": ... identyfikator klienta
    "CompanyName": ... nazwa klienta
    "ConfectionsSale97": ... wartość zakupionych przez niego produktów z kategorii
    'Confections' w 1997r

  }
]

```

d)

Napisz polecenie/zapytanie: Dla każdego klienta pokaż 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

```
[
  {
    "_id":
    "CustomerID": ... identyfikator klienta
    "CompanyName": ... nazwa klienta

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

e)

Założmy że pojawia się nowe zamówienie dla klienta 'ALFKI', zawierające dwa produkty 'Chai' oraz "Ikura"

- pozostałe pola w zamówieniu (ceny, liczby sztuk prod, inf o przewoźniku itp. możesz uzupełnić wg własnego uznania) Napisz polecenie które dodaje takie zamówienie do bazy
- aktualizując oryginalne kolekcje `orders`, `orderdetails`
- aktualizując kolekcję `OrderInfo`
- aktualizując kolekcję `CustomerInfo`

Napisz polecenie

- aktualizując oryginalną kolekcję `orderdetails``
- aktualizując kolekcję `OrderInfo`
- aktualizując kolekcję `CustomerInfo`

f)

Napisz polecenie które modyfikuje zamówienie dodane w pkt e) zwiększając zniżkę o 5% (dla każdej pozycji tego zamówienia)

Napisz polecenie

- aktualizując oryginalną kolekcję `orderdetails`
- aktualizując kolekcję `OrderInfo`
- aktualizując kolekcję `CustomerInfo`

UWAGA: W raporcie należy zamieścić kod poleceń oraz uzyskany rezultat, np wynik polecenia `db.kolekcka.find().limit(2)` lub jego fragment

Zadanie 1 - rozwiązanie

a)

```
db.orders.aggregate([
  { $lookup: {
    from: "customers",
    localField: "CustomerID",
    foreignField: "CustomerID",
    as: "Customer"
  }},
  { $unwind: "$Customer" },
  { $lookup: {
    from: "employees",
    localField: "EmployeeID",
    foreignField: "EmployeeID",
    as: "Employee"
  }},
  { $unwind: "$Employee" },
  { $lookup: {
    from: "orderdetails",
    localField: "OrderID",
    foreignField: "OrderID",
    as: "Orderdetails"
  }},
  { $lookup: {
    from: "shippers",
    localField: "ShipVia",
    foreignField: "ShipperID",
    as: "Shipper"
  }},
  { $unwind: { path: "$Shipper", preserveNullAndEmptyArrays: true } },
  { $unwind: { path: "$Orderdetails", preserveNullAndEmptyArrays: true } },
  { $lookup: {
    from: "products",
    localField: "Orderdetails.ProductID",
    foreignField: "ProductID",
    as: "Product"
  }},
  { $unwind: "$Product" },
  { $lookup: {
    from: "categories",
    localField: "Product.CategoryID",
    foreignField: "CategoryID",
    as: "ProdCat"
  }},
  { $unwind: "$ProdCat" },
  { $addFields: {
    "Orderdetails.product": {
      ProductID: "$Product.ProductID",
      ProductName: "$Product.ProductName",
      QuantityPerUnit: "$Product.QuantityPerUnit",
```

```

        CategoryID: "$Product.CategoryID",
        CategoryName: "$ProdCat.CategoryName"
    },
    "Orderdetails.Value": {
        $multiply: [
            "$Orderdetails.UnitPrice",
            "$Orderdetails.Quantity",
            { $subtract: [1, "$Orderdetails.Discount"] }
        ]
    }
}],
{ $group: {
    _id: "$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" },
    Orderdetails: { $push: {
        UnitPrice: "$Orderdetails.UnitPrice",
        Quantity: "$Orderdetails.Quantity",
        Discount: "$Orderdetails.Discount",
        Value: "$Orderdetails.Value",
        product: "$Orderdetails.product"
    }},
    Shipment: { $first: {
        Shipper: {
            ShipperID: "$Shipper.ShipperID",
            CompanyName: "$Shipper.CompanyName"
        },
        ShipName: "$ShipName",
        ShipAddress: "$ShipAddress",
        ShipCity: "$ShipCity",
        ShipCountry: "$ShipCountry"
    }
    }
}],
{ $addFields: {
    OrderTotal: { $sum: "$Orderdetails.Value" }
}],
{ $project: {
    _id: 1,

```

```

        OrderID: 1,
        Customer: 1,
        Employee: 1,
        Dates: 1,
        Orderdetails: 1,
        Freight: 1,
        OrderTotal: 1,
        Shipment: 1
    }},
    {
        $out: "OrderInfo"
    }
])

```

Wyniki polecenia `db.OrderInfo.find().limit(2):`

```

[
  {
    "_id": 10648,
    "Customer": {
      "CustomerID": "RICAR",
      "CompanyName": "Ricardo Adocicados",
      "City": "Rio de Janeiro",
      "Country": "Brazil"
    },
    "Dates": {
      "OrderDate": {"$date": "1997-08-28T00:00:00.000Z"},
      "RequiredDate": {"$date": "1997-10-09T00:00:00.000Z"}
    },
    "Employee": {
      "EmployeeID": 5,
      "FirstName": "Steven",
      "LastName": "Buchanan",
      "Title": "Sales Manager"
    },
    "Freight": 14.25,
    "OrderID": 10648,
    "OrderTotal": 372.37499959766865,
    "Orderdetails": [
      {
        "UnitPrice": 21,
        "Quantity": 15,
        "Discount": 0,
        "Value": 315,
        "product": {
          "ProductID": 22,
          "ProductName": "Gustaf's Knäckebröd",
          "QuantityPerUnit": "24 - 500 g pkgs.",
          "CategoryID": 5,
          "CategoryName": "Grains/Cereals"
        }
      }
    ]
  },

```

```

    {
      "UnitPrice": 4.5,
      "Quantity": 15,
      "Discount": 0.15000000596046448,
      "Value": 57.37499959766865,
      "product": {
        "ProductID": 24,
        "ProductName": "Guaraná Fantástica",
        "QuantityPerUnit": "12 - 355 ml cans",
        "CategoryID": 1,
        "CategoryName": "Beverages"
      }
    }
  ],
  "Shipment": {
    "Shipper": {
      "ShipperID": 2,
      "CompanyName": "United Package"
    },
    "ShipName": "Ricardo Adocicados",
    "ShipAddress": "Av. Copacabana, 267",
    "ShipCity": "Rio de Janeiro",
    "ShipCountry": "Brazil"
  }
},
{
  "_id": 10546,
  "Customer": {
    "CustomerID": "VICTE",
    "CompanyName": "Victuailles en stock",
    "City": "Lyon",
    "Country": "France"
  },
  "Dates": {
    "OrderDate": {"$date": "1997-05-23T00:00:00.000Z"},
    "RequiredDate": {"$date": "1997-06-20T00:00:00.000Z"}
  },
  "Employee": {
    "EmployeeID": 1,
    "FirstName": "Nancy",
    "LastName": "Davolio",
    "Title": "Sales Representative"
  },
  "Freight": 194.72,
  "OrderID": 10546,
  "OrderTotal": 2812,
  "Orderdetails": [
    {
      "UnitPrice": 30,
      "Quantity": 10,
      "Discount": 0,
      "Value": 300,
      "product": {
        "ProductID": 7,

```



```

        "ProductName": "Uncle Bob's Organic Dried Pears",
        "QuantityPerUnit": "12 - 1 lb pkgs.",
        "CategoryID": 7,
        "CategoryName": "Produce"
    },
    {
        "UnitPrice": 18,
        "Quantity": 30,
        "Discount": 0,
        "Value": 540,
        "product": {
            "ProductID": 35,
            "ProductName": "Steeleye Stout",
            "QuantityPerUnit": "24 - 12 oz bottles",
            "CategoryID": 1,
            "CategoryName": "Beverages"
        }
    },
    {
        "UnitPrice": 49.3,
        "Quantity": 40,
        "Discount": 0,
        "Value": 1972,
        "product": {
            "ProductID": 62,
            "ProductName": "Tarte au sucre",
            "QuantityPerUnit": "48 pies",
            "CategoryID": 3,
            "CategoryName": "Confections"
        }
    }
],
"Shipment": {
    "Shipper": {
        "ShipperID": 3,
        "CompanyName": "Federal Shipping"
    },
    "ShipName": "Victuailles en stock",
    "ShipAddress": "2, rue du Commerce",
    "ShipCity": "Lyon",
    "ShipCountry": "France"
}
}
]

```

b)

```
db.customers.aggregate([
  {
    $lookup: {
      from: "orders",
      localField: "CustomerID",
      foreignField: "CustomerID",
      as: "Orders"
    }
  },
  { $unwind: "$Orders" },
  {
    $lookup: {
      from: "employees",
      localField: "Orders.EmployeeID",
      foreignField: "EmployeeID",
      as: "Employee"
    }
  },
  { $unwind: "$Employee" },
  {
    $lookup: {
      from: "shippers",
      localField: "Orders.ShipVia",
      foreignField: "ShipperID",
      as: "Shipper"
    }
  },
  { $unwind: "$Shipper" },
  {
    $lookup: {
      from: "orderdetails",
      localField: "Orders.OrderID",
      foreignField: "OrderID",
      as: "Details"
    }
  },
  { $unwind: "$Details" },
  {
    $lookup: {
      from: "products",
      localField: "Details.ProductID",
      foreignField: "ProductID",
      as: "Product"
    }
  },
  { $unwind: "$Product" },
  {
    $lookup: {
      from: "categories",
      localField: "Product.CategoryID",
      foreignField: "CategoryID",
      as: "Category"
    }
  }
])
```

```

    },
    { $unwind: "$Category" },
    {
      $addFields: {
        "Details.Value": {
          $multiply: [
            "$Details.UnitPrice",
            "$Details.Quantity",
            { $subtract: [1, "$Details.Discount"] }
          ]
        },
        "Details.Product": {
          ProductID: "$Product.ProductID",
          ProductName: "$Product.ProductName",
          QuantityPerUnit: "$Product.QuantityPerUnit",
          CategoryID: "$Category.CategoryID",
          CategoryName: "$Category.CategoryName"
        }
      }
    },
    {
      $group: {
        _id: {
          CustomerID: "$CustomerID",
          OrderID: "$Orders.OrderID"
        },
        CompanyName: { $first: "$CompanyName" },
        City: { $first: "$City" },
        Country: { $first: "$Country" },
        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: { $push: "$Details" },
    OrderTotal: {
      $sum: "$Details.Value"
    }
  }
},
{
  $group: {
    _id: "$_id.CustomerID",
    CustomerID: { $first: "$_id.CustomerID" },
    CompanyName: { $first: "$CompanyName" },
    City: { $first: "$City" },
    Country: { $first: "$Country" },
    Orders: {
      $push: {
        OrderID: "$_id.OrderID",
        Employee: "$Employee",
        Dates: "$Dates",
        Freight: "$Freight",
        Shipment: "$Shipment",
        Orderdetails: "$OrderDetails",
        OrderTotal: "$OrderTotal"
      }
    }
  }
},
{
  $project: {
    _id: 1,
    CustomerID: 1,
    CompanyName: 1,
    City: 1,
    Country: 1,
    Orders: 1
  }
},
{
  $out: "CustomerInfo"
}
])

```

Wyniki polecenia `db.CustomerInfo.find().limit(1):`

```

[
  {
    "_id": "LETSS",
    "City": "San Francisco",
    "CompanyName": "Let's Stop N Shop",
    "Country": "USA",
    "CustomerID": "LETSS",

```

```

"Orders": [
  {
    "OrderID": 10579,
    "Employee": {
      "EmployeeID": 1,
      "FirstName": "Nancy",
      "LastName": "Davolio",
      "Title": "Sales Representative"
    },
    "Dates": {
      "OrderDate": {"$date": "1997-06-25T00:00:00.000Z"},
      "RequiredDate": {"$date": "1997-07-23T00:00:00.000Z"}
    },
    "Freight": 13.73,
    "Shipment": {
      "Shipper": {
        "ShipperID": 2,
        "CompanyName": "United Package"
      },
      "ShipName": "Let's Stop N Shop",
      "ShipAddress": "87 Polk St. Suite 5",
      "ShipCity": "San Francisco",
      "ShipCountry": "USA"
    },
    "Orderdetails": [
      {
        "_id": {"$oid": "63a06016bb3b972d6f4e1a33"},
        "OrderID": 10579,
        "ProductID": 15,
        "UnitPrice": 15.5,
        "Quantity": 10,
        "Discount": 0,
        "Value": 155,
        "Product": {
          "ProductID": 15,
          "ProductName": "Genen Shouyu",
          "QuantityPerUnit": "24 - 250 ml bottles",
          "CategoryID": 2,
          "CategoryName": "Condiments"
        }
      },
      {
        "_id": {"$oid": "63a06016bb3b972d6f4e1a34"},
        "OrderID": 10579,
        "ProductID": 75,
        "UnitPrice": 7.75,
        "Quantity": 21,
        "Discount": 0,
        "Value": 162.75,
        "Product": {
          "ProductID": 75,
          "ProductName": "Rhönbräu Klosterbier",
          "QuantityPerUnit": "24 - 0.5 l bottles",
          "CategoryID": 1,

```

```

        "CategoryName": "Beverages"
    }
}
],
"OrderTotal": 317.75
},
{
    "OrderID": 10719,
    "Employee": {
        "EmployeeID": 8,
        "FirstName": "Laura",
        "LastName": "Callahan",
        "Title": "Inside Sales Coordinator"
    },
    "Dates": {
        "OrderDate": {"$date": "1997-10-27T00:00:00.000Z"},
        "RequiredDate": {"$date": "1997-11-24T00:00:00.000Z"}
    },
    "Freight": 51.44,
    "Shipment": {
        "Shipper": {
            "ShipperID": 2,
            "CompanyName": "United Package"
        },
        "ShipName": "Let's Stop N Shop",
        "ShipAddress": "87 Polk St. Suite 5",
        "ShipCity": "San Francisco",
        "ShipCountry": "USA"
    },
    "Orderdetails": [
        {
            "_id": {"$oid": "63a06016bb3b972d6f4e1b99"},
            "OrderID": 10719,
            "ProductID": 18,
            "UnitPrice": 62.5,
            "Quantity": 12,
            "Discount": 0.25,
            "Value": 562.5,
            "Product": {
                "ProductID": 18,
                "ProductName": "Carnarvon Tigers",
                "QuantityPerUnit": "16 kg pkg.",
                "CategoryID": 8,
                "CategoryName": "Seafood"
            }
        },
        {
            "_id": {"$oid": "63a06016bb3b972d6f4e1b9a"},
            "OrderID": 10719,
            "ProductID": 30,
            "UnitPrice": 25.89,
            "Quantity": 3,
            "Discount": 0.25,
            "Value": 58.2525,

```

```

        "Product": {
          "ProductID": 30,
          "ProductName": "Nord-Ost Matjeshering",
          "QuantityPerUnit": "10 - 200 g glasses",
          "CategoryID": 8,
          "CategoryName": "Seafood"
        }
      },
      {
        "_id": {"$oid": "63a06016bb3b972d6f4e1b9b"},
        "OrderID": 10719,
        "ProductID": 54,
        "UnitPrice": 7.45,
        "Quantity": 40,
        "Discount": 0.25,
        "Value": 223.5,
        "Product": {
          "ProductID": 54,
          "ProductName": "Tourtière",
          "QuantityPerUnit": "16 pies",
          "CategoryID": 6,
          "CategoryName": "Meat/Poultry"
        }
      }
    ],
    "OrderTotal": 844.2525
  },
  {
    "OrderID": 10884,
    "Employee": {
      "EmployeeID": 4,
      "FirstName": "Margaret",
      "LastName": "Peacock",
      "Title": "Sales Representative"
    },
    "Dates": {
      "OrderDate": {"$date": "1998-02-12T00:00:00.000Z"},
      "RequiredDate": {"$date": "1998-03-12T00:00:00.000Z"}
    },
    "Freight": 90.97,
    "Shipment": {
      "Shipper": {
        "ShipperID": 2,
        "CompanyName": "United Package"
      },
      "ShipName": "Let's Stop N Shop",
      "ShipAddress": "87 Polk St. Suite 5",
      "ShipCity": "San Francisco",
      "ShipCountry": "USA"
    },
    "Orderdetails": [
      {
        "_id": {"$oid": "63a06016bb3b972d6f4e1d3c"},
        "OrderID": 10884,

```

```

        "ProductID": 21,
        "UnitPrice": 10,
        "Quantity": 40,
        "Discount": 0.05000000074505806,
        "Value": 379.9999997019768,
        "Product": {
            "ProductID": 21,
            "ProductName": "Sir Rodney's Scones",
            "QuantityPerUnit": "24 pkgs. x 4 pieces",
            "CategoryID": 3,
            "CategoryName": "Confections"
        }
    },
    {
        "_id": {"$oid": "63a06016bb3b972d6f4e1d3d"},
        "OrderID": 10884,
        "ProductID": 56,
        "UnitPrice": 38,
        "Quantity": 21,
        "Discount": 0.05000000074505806,
        "Value": 758.0999994054437,
        "Product": {
            "ProductID": 56,
            "ProductName": "Gnocchi di nonna Alice",
            "QuantityPerUnit": "24 - 250 g pkgs.",
            "CategoryID": 5,
            "CategoryName": "Grains/Cereals"
        }
    },
    {
        "_id": {"$oid": "63a06016bb3b972d6f4e1d3e"},
        "OrderID": 10884,
        "ProductID": 65,
        "UnitPrice": 21.05,
        "Quantity": 12,
        "Discount": 0.05000000074505806,
        "Value": 239.96999981179835,
        "Product": {
            "ProductID": 65,
            "ProductName": "Louisiana Fiery Hot Pepper Sauce",
            "QuantityPerUnit": "32 - 8 oz bottles",
            "CategoryID": 2,
            "CategoryName": "Condiments"
        }
    }
],
"OrderTotal": 1378.0699989192187
},
{
    "OrderID": 10735,
    "Employee": {
        "EmployeeID": 6,
        "FirstName": "Michael",
        "LastName": "Suyama",

```



```

    "Title": "Sales Representative"
  },
  "Dates": {
    "OrderDate": {"$date": "1997-11-10T00:00:00.000Z"},
    "RequiredDate": {"$date": "1997-12-08T00:00:00.000Z"}
  },
  "Freight": 45.97,
  "Shipment": {
    "Shipper": {
      "ShipperID": 2,
      "CompanyName": "United Package"
    },
    "ShipName": "Let's Stop N Shop",
    "ShipAddress": "87 Polk St. Suite 5",
    "ShipCity": "San Francisco",
    "ShipCountry": "USA"
  },
  "Orderdetails": [
    {
      "_id": {"$oid": "63a06016bb3b972d6f4e1bc1"},
      "OrderID": 10735,
      "ProductID": 61,
      "UnitPrice": 28.5,
      "Quantity": 20,
      "Discount": 0.10000000149011612,
      "Value": 512.9999991506338,
      "Product": {
        "ProductID": 61,
        "ProductName": "Sirop d'érable",
        "QuantityPerUnit": "24 - 500 ml bottles",
        "CategoryID": 2,
        "CategoryName": "Condiments"
      }
    },
    {
      "_id": {"$oid": "63a06016bb3b972d6f4e1bc2"},
      "OrderID": 10735,
      "ProductID": 77,
      "UnitPrice": 13,
      "Quantity": 2,
      "Discount": 0.10000000149011612,
      "Value": 23.39999996125698,
      "Product": {
        "ProductID": 77,
        "ProductName": "Original Frankfurter grüne Soße",
        "QuantityPerUnit": "12 boxes",
        "CategoryID": 2,
        "CategoryName": "Condiments"
      }
    }
  ],
  "OrderTotal": 536.3999991118908
}
]

```

```
}  
]
```

c)

```
db.customers.aggregate([  
  {  
    $lookup: {  
      from: "orders",  
      localField: "CustomerID",  
      foreignField: "CustomerID",  
      as: "Orders"  
    }  
  },  
  { $unwind: "$Orders" },  
  {  
    $match: {  
      "Orders.OrderDate": {  
        $gte: ISODate("1997-01-01"),  
        $lt: ISODate("1998-01-01")  
      }  
    }  
  },  
  {  
    $lookup: {  
      from: "orderdetails",  
      localField: "Orders.OrderID",  
      foreignField: "OrderID",  
      as: "Details"  
    }  
  },  
  { $unwind: "$Details" },  
  {  
    $lookup: {  
      from: "products",  
      localField: "Details.ProductID",  
      foreignField: "ProductID",  
      as: "Product"  
    }  
  },  
  { $unwind: "$Product" },  
  {  
    $lookup: {  
      from: "categories",  
      localField: "Product.CategoryID",  
      foreignField: "CategoryID",  
      as: "Category"  
    }  
  }  
])
```

```

    },
    { $unwind: "$Category" },
    {
      $match: {
        "Category.CategoryName": "Confections"
      }
    },
    {
      $group: {
        _id: "$CustomerID",
        CustomerID: { $first: "$CustomerID" },
        CompanyName: { $first: "$CompanyName" },
        ConfectionsSale97: {
          $sum: {
            $multiply: [
              "$Details.UnitPrice",
              "$Details.Quantity",
              { $subtract: [1, "$Details.Discount"] }
            ]
          }
        }
      }
    }
  ]
})

-----

db.OrderInfo.aggregate([
  {
    $match: {
      "Orderdetails.product.CategoryName": "Confections",
      "Dates.OrderDate": {
        $gte: ISODate("1997-01-01"),
        $lt: ISODate("1998-01-01")
      }
    }
  },
  { $unwind: "$Orderdetails" },
  {
    $match: {
      "Orderdetails.product.CategoryName": "Confections"
    }
  },
  {
    $group: {
      _id: "$Customer.CustomerID",
      CustomerID: { $first: "$Customer.CustomerID" },
      CompanyName: { $first: "$Customer.CompanyName" },
      ConfectionsSale97: { $sum: "$Orderdetails.Value" }
    }
  }
])

-----

```

```

db.CustomerInfo.aggregate([
  { $unwind: "$Orders" },
  {
    $match: {
      "Orders.Dates.OrderDate": {
        $gte: ISODate("1997-01-01"),
        $lt: ISODate("1998-01-01")
      }
    }
  },
  { $unwind: "$Orders.Orderdetails" },
  {
    $match: {
      "Orders.Orderdetails.Product.CategoryName": "Confections"
    }
  },
  {
    $group: {
      _id: "$CustomerID",
      CustomerID: { $first: "$CustomerID" },
      CompanyName: { $first: "$CompanyName" },
      ConfectionsSale97: {
        $sum: "$Orders.Orderdetails.Value"
      }
    }
  }
])

```

Fragment wyniku, każde polecenia zwraca to samo, w trzecim podejściu różni się tylko kolejność:

```

[
  {
    "_id": "MAGAA",
    "CompanyName": "Magazzini Alimentari Riuniti",
    "ConfectionsSale97": 1300.0499990858139,
    "CustomerID": "MAGAA"
  },
  {
    "_id": "PRINI",
    "CompanyName": "Princesa Isabel Vinhos",
    "ConfectionsSale97": 126,
    "CustomerID": "PRINI"
  },
  {
    "_id": "RANCH",
    "CompanyName": "Rancho grande",
    "ConfectionsSale97": 199.39999999999998,
    "CustomerID": "RANCH"
  },
  {

```

```

    "_id": "SAVEA",
    "CompanyName": "Save-a-lot Markets",
    "ConfectionsSale97": 6351.084993118047,
    "CustomerID": "SAVEA"
  },
  {
    "_id": "SPECD",
    "CompanyName": "Spécialités du monde",
    "ConfectionsSale97": 52.349999999999994,
    "CustomerID": "SPECD"
  },
  {
    "_id": "SUPRD",
    "CompanyName": "Suprêmes délices",
    "ConfectionsSale97": 900.3,
    "CustomerID": "SUPRD"
  },
  {
    "_id": "FOLIG",
    "CompanyName": "Folies gourmandes",
    "ConfectionsSale97": 2232,
    "CustomerID": "FOLIG"
  },
  {
    "_id": "OCEAN",
    "CompanyName": "Océano Atlántico Ltda.",
    "ConfectionsSale97": 96,
    "CustomerID": "OCEAN"
  },
  {
    "_id": "BSBEV",
    "CompanyName": "B's Beverages",
    "ConfectionsSale97": 875,
    "CustomerID": "BSBEV"
  },
  {
    "_id": "EASTC",
    "CompanyName": "Eastern Connection",
    "ConfectionsSale97": 480,
    "CustomerID": "EASTC"
  }
]

```

d)

```

db.orders.aggregate([ //originalne kolekcje
  {
    $lookup: {

```

```

        from: "customers",
        localField: "CustomerID",
        foreignField: "CustomerID",
        as: "Customer"
    }
},
{ $unwind: "$Customer"},

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

{
    $project: {
        CustomerID: "$Customer.CustomerID",
        CompanyName: "$Customer.CompanyName",
        OrderDate: "$OrderDate", //z orders
        TotalSale: {
            $multiply: [
                "$Orderdetails.UnitPrice",
                "$Orderdetails.Quantity",
                { $subtract: [1, "$Orderdetails.Discount"]}
            ]
        }
    }
},

{
    $addFields: {
        Year: { $year: "$OrderDate"},
        Month: { $month: "$OrderDate"},
    }
},

{
    $group: {
        _id: { CustomerID: "$CustomerID", Year: "$Year", Month: "$Month"},
        CustomerID: { $first: "$CustomerID"},
        CompanyName: { $first: "$CompanyName"},
        TotalSale: { $sum: "$TotalSale" }
    }
},

{
    $group: {
        _id: { CustomerID: "$CustomerID" } ,
        CompanyName: { $first: "$CompanyName" },
    }
}

```

```

        Sales: {
            $push: {
                Year: "$_id.Year",
                Month: "$_id.Month",
                Total: "$TotalSale"
            }
        }
    },
    {
        $project: {
            _id: 1,
            CustomerID: "$_id.CustomerID",
            CompanyName: "$CompanyName",
            Sales: 1
        }
    }
])

```

```

-----

db.OrderInfo.aggregate([
    {
        $unwind: "$Orderdetails"
    }, //niepotrzebne, $sum działa na tablicach
    {
        $addFields: {
            Year: { $year: "$Dates.OrderDate" },
            Month: { $month: "$Dates.OrderDate" }
        }
    },
    {
        $group: {
            _id: { CustomerID: "$Customer.CustomerID", Year: "$Year", Month: "$Month"
        },
        CustomerID: { $first: "$Customer.CustomerID" },
        CompanyName: { $first: "$Customer.CompanyName" },
        TotalSale: { $sum: "$Orderdetails.Value" }
    },
    {
        $group: {
            _id: {CustomerID: "$CustomerID"},//z zakresu wyżej dostępny
            CompanyName: { $first: "$CompanyName" },
            Sales: {
                $push: {
                    Year: "$_id.Year",
                    Month: "$_id.Month",
                    Total: "$TotalSale"
                }
            }
        }
    }
])

```

```

    }
  }
},
{
  $project: {
    _id: 0,
    CustomerID: "$_id.CustomerID", //wyżej tylko w _id
    CompanyName: "$CompanyName",
    Sales: 1
  }
}
])

```

```

-----

db.CustomerInfo.aggregate([
  {
    $unwind: "$Orders"
  }, // -,,- jak wyżej

  {
    $addFields: {
      Year: { $year: "$Orders.Dates.OrderDate" },
      Month: { $month: "$Orders.Dates.OrderDate" }
    }
  },
  {
    $group: {
      _id: { CustomerID: "$CustomerID", Year: "$Year", Month: "$Month" },
      CustomerID: { $first: "$CustomerID" },
      CompanyName: { $first: "$CompanyName" },
      TotalSale: { $sum: "$Orders.OrderTotal" }
    }
  },
  {
    $group: {
      _id: { CustomerID: "$CustomerID",
      CompanyName: { $first: "$CompanyName" },
      Sales: {
        $push: {
          Year: "$_id.Year",
          Month: "$_id.Month",
          Total: "$TotalSale"
        }
      }
    }
  },
  {
    $project: {

```



```
    _id: 0,  
    CustomerID: "$_id.CustomerID",  
    CompanyName: "$CompanyName",  
    Sales: 1  
  }  
}  
])
```

Wynik polecenia (jedno z podanych powyżej, wszystkie zwracają taki sam format)

```
{  
  Sales: [  
    {  
      Year: 1997,  
      Month: 1,  
      Total: 224.829999990209936  
    },  
    {  
      Year: 1996,  
      Month: 12,  
      Total: 166  
    },  
    {  
      Year: 1997,  
      Month: 10,  
      Total: 287.8  
    },  
    {  
      Year: 1997,  
      Month: 6,  
      Total: 309.99999884516  
    },  
    {  
      Year: 1997,  
      Month: 8,  
      Total: 1779.1999993890522  
    },  
    {  
      Year: 1997,  
      Month: 4,  
      Total: 525.299996316433  
    },  
    {  
      Year: 1996,  
      Month: 11,  
      Total: 814.4199954867363  
    }  
  ]  
}
```

```

    ],
    CustomerID: 'FAMIA',
    CompanyName: 'Familia Arquibaldo'
  }
  {
    Sales: [
      {
        Year: 1997,
        Month: 2,
        Total: 631.6
      },
      {
        Year: 1997,
        Month: 3,
        Total: 156.000000000000003
      },
      {
        Year: 1998,
        Month: 1,
        Total: 931.5
      }
    ],
    CustomerID: 'CONSH',
    CompanyName: 'Consolidated Holdings'
  }
}

```

e)

Wstawianie danych:

```

db.orders.insertOne({
  OrderID: 11078,
  EmployeeID: 5,
  OrderDate: new Date("2025-04-16"),
  RequiredDate: new Date("2025-04-23"),
  ShippedDate: null,
  ShipVia: 2,
  Freight: 20,
  ShipName: "Alfreds Futterkiste",
  ShipAddress: "Obere Str. 57",
  ShipCity: "Berlin",
  ShipRegion: null,
  ShipPostalCode: "12209",
  ShipCountry: "Germany"
})

```

```
db.orderdetails.insertMany([
  {
    OrderID: 11078,
    ProductID: 1,
    UnitPrice: 18,
    Quantity: 10,
    Discount: 0.1
  },
  {
    OrderID: 11078,
    ProductID: 10,
    UnitPrice: 31,
    Quantity: 5,
    Discount: 0
  }
])
```

Aktualizacja kolekcji OrderInfo:

```
db.OrderInfo.createIndex({ OrderID: 1 }, { unique: true })

db.orders.aggregate([
  { $match: { OrderID: 11078 } },
  { $lookup: {
    from: "customers",
    localField: "CustomerID",
    foreignField: "CustomerID",
    as: "Customer"
  }},
  { $unwind: "$Customer" },
  { $lookup: {
    from: "employees",
    localField: "EmployeeID",
    foreignField: "EmployeeID",
    as: "Employee"
  }},
  { $unwind: "$Employee" },
  { $lookup: {
    from: "orderdetails",
    localField: "OrderID",
    foreignField: "OrderID",
    as: "Orderdetails"
  }},
  { $lookup: {
```

```

        from: "shippers",
        localField: "ShipVia",
        foreignField: "ShipperID",
        as: "Shipper"
    }},
    { $unwind: { path: "$Shipper", preserveNullAndEmptyArrays: true } },
    { $unwind: { path: "$Orderdetails", preserveNullAndEmptyArrays: true } },
    { $lookup: {
        from: "products",
        localField: "Orderdetails.ProductID",
        foreignField: "ProductID",
        as: "Product"
    }},
    { $unwind: "$Product" },
    { $lookup: {
        from: "categories",
        localField: "Product.CategoryID",
        foreignField: "CategoryID",
        as: "ProdCat"
    }},
    { $unwind: "$ProdCat"},
    { $addFields: {
        "Orderdetails.product": {
            ProductID: "$Product.ProductID",
            ProductName: "$Product.ProductName",
            QuantityPerUnit: "$Product.QuantityPerUnit",
            CategoryID: "$Product.CategoryID",
            CategoryName: "$ProdCat.CategoryName"
        },
        "Orderdetails.Value": {
            $multiply: [
                "$Orderdetails.UnitPrice",
                "$Orderdetails.Quantity",
                { $subtract: [1, "$Orderdetails.Discount"] }
            ]
        }
    }},
    { $group: {
        _id: "$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" },
    Orderdetails: { $push: {
        UnitPrice: "$Orderdetails.UnitPrice",
        Quantity: "$Orderdetails.Quantity",
        Discount: "$Orderdetails.Discount",
        Value: "$Orderdetails.Value",
        product: "$Orderdetails.product"
    }},
    Shipment: { $first: {
        Shipper: {
            ShipperID: "$Shipper.ShipperID",
            CompanyName: "$Shipper.CompanyName"
        },
        ShipName: "$ShipName",
        ShipAddress: "$ShipAddress",
        ShipCity: "$ShipCity",
        ShipCountry: "$ShipCountry"
    }}
    }},
    { $addFields: {
        OrderTotal: { $sum: "$Orderdetails.Value" }
    }},
    { $merge: {
        into: "OrderInfo",
        on: "OrderID",
        whenMatched: "replace",
        whenNotMatched: "insert"
    }}
    ]);

```

Wynik polecenia: db.OrderInfo.find({OrderID: 11078})

```

{
  _id: 11078,
  OrderID: 11078,
  Customer: {
    CustomerID: 'ALFKI',
    CompanyName: 'Alfreds Futterkiste',
    City: 'Berlin',
    Country: 'Germany'
  },
  Employee: {
    EmployeeID: 5,
    FirstName: 'Steven',
    LastName: 'Buchanan',

```

```
    Title: 'Sales Manager'
  },
  Dates: {
    OrderDate: 2025-04-16T00:00:00.000Z,
    RequiredDate: 2025-04-23T00:00:00.000Z
  },
  Freight: 20,
  Orderdetails: [
    {
      UnitPrice: 18,
      Quantity: 10,
      Discount: 0.1,
      Value: 162,
      product: {
        ProductID: 1,
        ProductName: 'Chai',
        QuantityPerUnit: '10 boxes x 20 bags',
        CategoryID: 1,
        CategoryName: 'Beverages'
      }
    },
    {
      UnitPrice: 31,
      Quantity: 5,
      Discount: 0,
      Value: 155,
      product: {
        ProductID: 10,
        ProductName: 'Ikura',
        QuantityPerUnit: '12 - 200 ml jars',
        CategoryID: 8,
        CategoryName: 'Seafood'
      }
    }
  ],
  Shipment: {
    Shipper: {
      ShipperID: 2,
      CompanyName: 'United Package'
    },
    ShipName: 'Alfreds Futterkiste',
    ShipAddress: 'Obere Str. 57',
    ShipCity: 'Berlin',
    ShipCountry: 'Germany'
  },
  OrderTotal: 317
}
```

```
db.CustomerInfo.createIndex({ CustomerID: 1 }, { unique: true })

db.customers.aggregate([
  {
    $match: {
      CustomerID: "ALFKI"
    }
  },
  {
    $lookup: {
      from: "orders",
      let: { customerId: "$CustomerID" },
      pipeline: [
        {
          $match: {
            $expr: { $eq: ["$CustomerID", "$$customerId"] }
          }
        }
      ],
      as: "Orders"
    }
  },
  { $unwind: "$Orders" },
  {
    $lookup: {
      from: "employees",
      let: { empId: "$Orders.EmployeeID" },
      pipeline: [
        {
          $match: {
            $expr: { $eq: ["$EmployeeID", "$$empId"] }
          }
        }
      ],
      { $limit: 1 }
    },
    as: "Employee"
  },
  {
    $unwind: "$Employee" },
  {
    $lookup: {
      from: "shippers",
      let: { shipperId: "$Orders.ShipVia" },
      pipeline: [
        {
          $match: {
            $expr: { $eq: ["$ShipperID", "$$shipperId"] }
          }
        }
      ],
      { $limit: 1 }
    },
    as: "Shipper"
  }
])
```

```

    }
  },
  { $unwind: "$Shipper" },
  {
    $lookup: {
      from: "orderdetails",
      let: { orderId: "$Orders.OrderID" },
      pipeline: [
        {
          $match: {
            $expr: { $eq: ["$OrderID", "$$orderId"] }
          }
        }
      ],
      as: "Details"
    }
  },
  { $unwind: "$Details" },
  {
    $lookup: {
      from: "products",
      let: { productId: "$Details.ProductID" },
      pipeline: [
        {
          $match: {
            $expr: { $eq: ["$ProductID", "$$productId"] }
          }
        },
        { $limit: 1 }
      ],
      as: "Product"
    }
  },
  { $unwind: "$Product" },
  {
    $lookup: {
      from: "categories",
      let: { categoryId: "$Product.CategoryID" },
      pipeline: [
        {
          $match: {
            $expr: { $eq: ["$CategoryID", "$$categoryId"] }
          }
        },
        { $limit: 1 }
      ],
      as: "Category"
    }
  },
  { $unwind: "$Category" },
  {
    $addFields: {
      "Details.Value": {
        $multiply: [

```



```

        "$Details.UnitPrice",
        "$Details.Quantity",
        { $subtract: [1, "$Details.Discount"] }
    ]
},
"Details.Product": {
    ProductID: "$Product.ProductID",
    ProductName: "$Product.ProductName",
    QuantityPerUnit: "$Product.QuantityPerUnit",
    CategoryID: "$Category.CategoryID",
    CategoryName: "$Category.CategoryName"
}
},
{
    $group: {
        _id: {
            CustomerID: "$CustomerID",
            OrderID: "$Orders.OrderID"
        },
        CompanyName: { $first: "$CompanyName" },
        City: { $first: "$City" },
        Country: { $first: "$Country" },
        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: { $push: "$Details" },
        OrderTotal: {
            $sum: "$Details.Value"
        }
    }
}

```

```

    }
  },
  {
    $group: {
      _id: "$_id.CustomerID",
      CustomerID: { $first: "$_id.CustomerID" },
      CompanyName: { $first: "$CompanyName" },
      City: { $first: "$City" },
      Country: { $first: "$Country" },
      Orders: {
        $push: {
          OrderID: "$_id.OrderID",
          Employee: "$Employee",
          Dates: "$Dates",
          Freight: "$Freight",
          Shipment: "$Shipment",
          Orderdetails: "$OrderDetails",
          OrderTotal: "$OrderTotal"
        }
      }
    }
  },
  {
    $project: {
      _id: 1,
      CustomerID: 1,
      CompanyName: 1,
      City: 1,
      Country: 1,
      Orders: 1
    }
  },
  {
    $merge: {
      into: "CustomerInfo",
      on: "CustomerID",
      whenMatched: "replace",
      whenNotMatched: "insert"
    }
  }
}
])

```

Wynik polecenia: db.CustomerInfo.find({"Orders.OrderID": 11078})

```

{
  _id: 'ALFKI',
  CustomerID: 'ALFKI',
  CompanyName: 'Alfreds Futterkiste',

```

```
City: 'Berlin',
Country: 'Germany',
Orders: [
  {
    OrderID: 10835,
    Employee: {
      EmployeeID: 1,
      FirstName: 'Nancy',
      LastName: 'Davolio',
      Title: 'Sales Representative'
    },
    Dates: {
      OrderDate: 1998-01-15T00:00:00.000Z,
      RequiredDate: 1998-02-12T00:00:00.000Z
    },
    Freight: 69.53,
    Shipment: {
      Shipper: {
        ShipperID: 3,
        CompanyName: 'Federal Shipping'
      },
      ShipName: "Alfred's Futterkiste",
      ShipAddress: 'Obere Str. 57',
      ShipCity: 'Berlin',
      ShipCountry: 'Germany'
    },
    Orderdetails: [
      {
        _id: ObjectId('63a06016bb3b972d6f4e1cb9'),
        OrderID: 10835,
        ProductID: 59,
        UnitPrice: 55,
        Quantity: 15,
        Discount: 0,
        Value: 825,
        Product: {
          ProductID: 59,
          ProductName: 'Raclette Courdavault',
          QuantityPerUnit: '5 kg pkg.',
          CategoryID: 4,
          CategoryName: 'Dairy Products'
        }
      },
      {
        _id: ObjectId('63a06016bb3b972d6f4e1cba'),
        OrderID: 10835,
        ProductID: 77,
        UnitPrice: 13,
        Quantity: 2,
        Discount: 0.20000000298023224,
        Value: 20.799999922513962,
        Product: {
          ProductID: 77,
          ProductName: 'Original Frankfurter grüne Soße',
```

```

        QuantityPerUnit: '12 boxes',
        CategoryID: 2,
        CategoryName: 'Condiments'
    }
}
],
OrderTotal: 845.799999922514
},
{
    OrderID: 10952,
    Employee: {
        EmployeeID: 1,
        FirstName: 'Nancy',
        LastName: 'Davolio',
        Title: 'Sales Representative'
    },
    Dates: {
        OrderDate: 1998-03-16T00:00:00.000Z,
        RequiredDate: 1998-04-27T00:00:00.000Z
    },
    Freight: 40.42,
    Shipment: {
        Shipper: {
            ShipperID: 1,
            CompanyName: 'Speedy Express'
        },
        ShipName: "Alfred's Futterkiste",
        ShipAddress: 'Obere Str. 57',
        ShipCity: 'Berlin',
        ShipCountry: 'Germany'
    },
    Orderdetails: [
        {
            _id: ObjectId('63a06016bb3b972d6f4e1ddd'),
            OrderID: 10952,
            ProductID: 6,
            UnitPrice: 25,
            Quantity: 16,
            Discount: 0.05000000074505806,
            Value: 379.9999997019768,
            Product: {
                ProductID: 6,
                ProductName: "Grandma's Boysenberry Spread",
                QuantityPerUnit: '12 - 8 oz jars',
                CategoryID: 2,
                CategoryName: 'Condiments'
            }
        },
        {
            _id: ObjectId('63a06016bb3b972d6f4e1dde'),
            OrderID: 10952,
            ProductID: 28,
            UnitPrice: 45.6,
            Quantity: 2,

```

```

        Discount: 0,
        Value: 91.2,
        Product: {
            ProductID: 28,
            ProductName: 'Rössle Sauerkraut',
            QuantityPerUnit: '25 - 825 g cans',
            CategoryID: 7,
            CategoryName: 'Produce'
        }
    },
    OrderTotal: 471.19999970197676
},
{
    OrderID: 10692,
    Employee: {
        EmployeeID: 4,
        FirstName: 'Margaret',
        LastName: 'Peacock',
        Title: 'Sales Representative'
    },
    Dates: {
        OrderDate: 1997-10-03T00:00:00.000Z,
        RequiredDate: 1997-10-31T00:00:00.000Z
    },
    Freight: 61.02,
    Shipment: {
        Shipper: {
            ShipperID: 2,
            CompanyName: 'United Package'
        },
        ShipName: "Alfred's Futterkiste",
        ShipAddress: 'Obere Str. 57',
        ShipCity: 'Berlin',
        ShipCountry: 'Germany'
    },
    Orderdetails: [
        {
            _id: ObjectId('63a06016bb3b972d6f4e1b4a'),
            OrderID: 10692,
            ProductID: 63,
            UnitPrice: 43.9,
            Quantity: 20,
            Discount: 0,
            Value: 878,
            Product: {
                ProductID: 63,
                ProductName: 'Veggie-spread',
                QuantityPerUnit: '15 - 625 g jars',
                CategoryID: 2,
                CategoryName: 'Condiments'
            }
        }
    ]
},

```

```
OrderTotal: 878
},
{
  OrderID: 11011,
  Employee: {
    EmployeeID: 3,
    FirstName: 'Janet',
    LastName: 'Leverling',
    Title: 'Sales Representative'
  },
  Dates: {
    OrderDate: 1998-04-09T00:00:00.000Z,
    RequiredDate: 1998-05-07T00:00:00.000Z
  },
  Freight: 1.21,
  Shipment: {
    Shipper: {
      ShipperID: 1,
      CompanyName: 'Speedy Express'
    },
    ShipName: "Alfred's Futterkiste",
    ShipAddress: 'Obere Str. 57',
    ShipCity: 'Berlin',
    ShipCountry: 'Germany'
  },
  Orderdetails: [
    {
      _id: ObjectId('63a06016bb3b972d6f4e1e6f'),
      OrderID: 11011,
      ProductID: 58,
      UnitPrice: 13.25,
      Quantity: 40,
      Discount: 0.05000000074505806,
      Value: 503.4999996051192,
      Product: {
        ProductID: 58,
        ProductName: 'Escargots de Bourgogne',
        QuantityPerUnit: '24 pieces',
        CategoryID: 8,
        CategoryName: 'Seafood'
      }
    },
    {
      _id: ObjectId('63a06016bb3b972d6f4e1e70'),
      OrderID: 11011,
      ProductID: 71,
      UnitPrice: 21.5,
      Quantity: 20,
      Discount: 0,
      Value: 430,
      Product: {
        ProductID: 71,
        ProductName: 'Flotemysost',
        QuantityPerUnit: '10 - 500 g pkgs.',
```

```

        CategoryID: 4,
        CategoryName: 'Dairy Products'
    }
}
],
OrderTotal: 933.4999996051192
},
{
    OrderID: 11078,
    Employee: {
        EmployeeID: 5,
        FirstName: 'Steven',
        LastName: 'Buchanan',
        Title: 'Sales Manager'
    },
    Dates: {
        OrderDate: 2025-04-16T00:00:00.000Z,
        RequiredDate: 2025-04-23T00:00:00.000Z
    },
    Freight: 20,
    Shipment: {
        Shipper: {
            ShipperID: 2,
            CompanyName: 'United Package'
        },
        ShipName: 'Alfreds Futterkiste',
        ShipAddress: 'Obere Str. 57',
        ShipCity: 'Berlin',
        ShipCountry: 'Germany'
    },
    Orderdetails: [
        {
            _id: ObjectId('6813aa513d52ef96860fd52c'),
            OrderID: 11078,
            ProductID: 1,
            UnitPrice: 18,
            Quantity: 10,
            Discount: 0.1,
            Value: 162,
            Product: {
                ProductID: 1,
                ProductName: 'Chai',
                QuantityPerUnit: '10 boxes x 20 bags',
                CategoryID: 1,
                CategoryName: 'Beverages'
            }
        },
        {
            _id: ObjectId('6813aa513d52ef96860fd52d'),
            OrderID: 11078,
            ProductID: 10,
            UnitPrice: 31,
            Quantity: 5,
            Discount: 0,

```

```

        Value: 155,
        Product: {
          ProductID: 10,
          ProductName: 'Ikura',
          QuantityPerUnit: '12 - 200 ml jars',
          CategoryID: 8,
          CategoryName: 'Seafood'
        }
      ],
      OrderTotal: 317
    },
    {
      OrderID: 10643,
      Employee: {
        EmployeeID: 6,
        FirstName: 'Michael',
        LastName: 'Suyama',
        Title: 'Sales Representative'
      },
      Dates: {
        OrderDate: 1997-08-25T00:00:00.000Z,
        RequiredDate: 1997-09-22T00:00:00.000Z
      },
      Freight: 29.46,
      Shipment: {
        Shipper: {
          ShipperID: 1,
          CompanyName: 'Speedy Express'
        },
        ShipName: 'Alfreds Futterkiste',
        ShipAddress: 'Obere Str. 57',
        ShipCity: 'Berlin',
        ShipCountry: 'Germany'
      },
      Orderdetails: [
        {
          _id: ObjectId('63a06016bb3b972d6f4e1acb'),
          OrderID: 10643,
          ProductID: 28,
          UnitPrice: 45.6,
          Quantity: 15,
          Discount: 0.25,
          Value: 513,
          Product: {
            ProductID: 28,
            ProductName: 'Rössle Sauerkraut',
            QuantityPerUnit: '25 - 825 g cans',
            CategoryID: 7,
            CategoryName: 'Produce'
          }
        },
        {
          _id: ObjectId('63a06016bb3b972d6f4e1acc'),

```



```

      OrderID: 10643,
      ProductID: 39,
      UnitPrice: 18,
      Quantity: 21,
      Discount: 0.25,
      Value: 283.5,
      Product: {
        ProductID: 39,
        ProductName: 'Chartreuse verte',
        QuantityPerUnit: '750 cc per bottle',
        CategoryID: 1,
        CategoryName: 'Beverages'
      }
    },
    {
      _id: ObjectId('63a06016bb3b972d6f4e1acd'),
      OrderID: 10643,
      ProductID: 46,
      UnitPrice: 12,
      Quantity: 2,
      Discount: 0.25,
      Value: 18,
      Product: {
        ProductID: 46,
        ProductName: 'Spegesild',
        QuantityPerUnit: '4 - 450 g glasses',
        CategoryID: 8,
        CategoryName: 'Seafood'
      }
    }
  ],
  OrderTotal: 814.5
},
{
  OrderID: 10702,
  Employee: {
    EmployeeID: 4,
    FirstName: 'Margaret',
    LastName: 'Peacock',
    Title: 'Sales Representative'
  },
  Dates: {
    OrderDate: 1997-10-13T00:00:00.000Z,
    RequiredDate: 1997-11-24T00:00:00.000Z
  },
  Freight: 23.94,
  Shipment: {
    Shipper: {
      ShipperID: 1,
      CompanyName: 'Speedy Express'
    },
    ShipName: "Alfred's Futterkiste",
    ShipAddress: 'Obere Str. 57',
    ShipCity: 'Berlin',

```

```

    ShipCountry: 'Germany'
  },
  Orderdetails: [
    {
      _id: ObjectId('63a06016bb3b972d6f4e1b68'),
      OrderID: 10702,
      ProductID: 3,
      UnitPrice: 10,
      Quantity: 6,
      Discount: 0,
      Value: 60,
      Product: {
        ProductID: 3,
        ProductName: 'Aniseed Syrup',
        QuantityPerUnit: '12 - 550 ml bottles',
        CategoryID: 2,
        CategoryName: 'Condiments'
      }
    },
    {
      _id: ObjectId('63a06016bb3b972d6f4e1b69'),
      OrderID: 10702,
      ProductID: 76,
      UnitPrice: 18,
      Quantity: 15,
      Discount: 0,
      Value: 270,
      Product: {
        ProductID: 76,
        ProductName: 'Lakkalikööri',
        QuantityPerUnit: '500 ml',
        CategoryID: 1,
        CategoryName: 'Beverages'
      }
    }
  ],
  OrderTotal: 330
}
]
}

```

f)

```

db.orderdetails.updateMany(
  { OrderID: 11078 },
  { $inc: { Discount: 0.05 } }
)

```

```
-----

db.orders.aggregate([
  { $match: { OrderID: 11078 } },
  { $lookup: {
    from: "customers",
    localField: "CustomerID",
    foreignField: "CustomerID",
    as: "Customer"
  }},
  { $unwind: "$Customer" },
  { $lookup: {
    from: "employees",
    localField: "EmployeeID",
    foreignField: "EmployeeID",
    as: "Employee"
  }},
  { $unwind: "$Employee" },
  {
    $lookup: {
      from: "orderdetails",
      localField: "OrderID",
      foreignField: "OrderID",
      as: "Orderdetails"
    }
  },
  {
    $unwind: "$Orderdetails"
  },
  { $lookup: {
    from: "shippers",
    localField: "ShipVia",
    foreignField: "ShipperID",
    as: "Shipper"
  }},
  { $unwind: { path: "$Shipper", preserveNullAndEmptyArrays: true } },
  {
    $lookup: {
      from: "products",
      localField: "Orderdetails.ProductID",
      foreignField: "ProductID",
      as: "Product"
    }
  },
  {
    $unwind: "$Product"
  },
  {
    $lookup: {
      from: "categories",
      localField: "Product.CategoryID",
      foreignField: "CategoryID",
      as: "Category"
    }
  },
  { $unwind: "$Category" },
```

```

{
  $set: {
    "Orderdetails.product": {
      ProductID: "$Product.ProductID",
      ProductName: "$Product.ProductName",
      QuantityPerUnit: "$Product.QuantityPerUnit",
      CategoryID: "$Product.CategoryID",
      CategoryName: "$ProdCat.CategoryName"
    },
    "Orderdetails.Value": {
      $multiply: [
        "$Orderdetails.UnitPrice",
        "$Orderdetails.Quantity",
        { $subtract: [1, "$Orderdetails.Discount"] }
      ]
    }
  }
},
{
  $group: {
    _id: "$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: { $push: "$Orderdetails" },
    OrderTotal: { $sum: "$Orderdetails.Value" }
  }
}

```

```

    },
    {
      $merge: {
        into: "OrderInfo",
        on: "OrderID",
        whenMatched: "replace",
        whenNotMatched: "insert"
      }
    }
  ]
])

-----

db.OrderInfo.aggregate([
  {
    $group: {
      _id: "$Customer.CustomerID",
      CustomerID: { $first: "$Customer.CustomerID" },
      CompanyName: { $first: "$Customer.CompanyName" },
      City: { $first: "$Customer.City" },
      Country: { $first: "$Customer.Country" },
      Orders: { $push: "$$ROOT" }
    }
  },
  {
    $unset: ["Orders.Customer"]
  },
  {
    $merge: {
      into: "CustomerInfo",
      on: "CustomerID",
      whenMatched: "replace",
      whenNotMatched: "insert"
    }
  }
])

```

Wyniki do f z podejściem z **OrderInfo**:

przed:

Bazy2 > north0 > OrderInfo Open MongoDB shell

Documents 831 Aggregations Schema Indexes 2 Validation

{ _id: 11078 }

Generate query Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE 100 1 - 1 of 1

```
_id: 11078
OrderID: 11078
Customer: Object
Employee: Object
Dates: Object
Freight: 20
Orderdetails: Array (2)
Shipment: Object
OrderTotal: 317
```

po:

Bazy2 > north0 > OrderInfo Open MongoDB shell

Documents 830 Aggregations Schema Indexes 2 Validation

{ _id: 11078 }

Generate query Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE 100 1 - 1 of 1

```
_id: 11078
OrderID: 11078
Customer: Object
Employee: Object
Dates: Object
Freight: 20
Orderdetails: Array (2)
Shipment: Object
OrderTotal: 300.25
```

Wyniki do f z podejściem z CustomerInfo:

przed:

Bazy2 > north0 > CustomerInfo Open MongoDB shell

Documents 89 Aggregations Schema Indexes 2 Validation

{ "Orders.OrderID": 11078 }

Generate query Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE INSIGHT 100 1 - 1 of 1

```
4: Object
  OrderID: 11078
  Employee: Object
  Dates: Object
  Freight: 20
  Shipment: Object
  Orderdetails: Array (2)
    0: Object
      _id: ObjectId('6813aa513d52ef96860fd52c')
      OrderID: 11078
      ProductID: 1
      UnitPrice: 18
      Quantity: 10
      Discount: 0.1
      Value: 162
      Product: Object
        ProductID: 1
        ProductName: "Chai"
        QuantityPerUnit: "10 boxes x 20 bags"
        CategoryID: 1
        CategoryName: "Beverages"
    1: Object
      _id: ObjectId('6813aa513d52ef96860fd52d')
      OrderID: 11078
      ProductID: 10
      UnitPrice: 31
      Quantity: 5
      Discount: 0
      Value: 155
      Product: Object
        ProductID: 10
        ProductName: "Ikura"
        QuantityPerUnit: "12 - 200 ml jars"
        CategoryID: 8
        CategoryName: "Seafood"
  OrderTotal: 317
```

po:

The screenshot shows the MongoDB Compass interface. At the top, the database is 'Bazy2', the collection is 'north0', and the document is 'CustomerInfo'. The query is `{ "Orders.OrderID": 11078 }`. The result shows a document with the following structure:

```

{
  OrderTotal: 330,
  _id: 11078,
  OrderID: 11078,
  Employee: Object,
  Dates: Object,
  Freight: 20,
  Shipment: Object,
  Orderdetails: Array (2)
}

```

The 'Orderdetails' array contains two objects:

- Object 0:
 - _id: ObjectId('6813aa513d52ef96860fd52c')
 - OrderID: 11078
 - ProductID: 1
 - UnitPrice: 18
 - Quantity: 10
 - Discount: 0.15000000000000002
 - product:
 - ProductID: 1
 - ProductName: "Chai"
 - QuantityPerUnit: "10 boxes x 20 bags"
 - CategoryID: 1
 - Value: 153
- Object 1:
 - _id: ObjectId('6813aa513d52ef96860fd52d')
 - OrderID: 11078
 - ProductID: 10
 - UnitPrice: 31
 - Quantity: 5
 - Discount: 0.05
 - product:
 - ProductID: 10
 - ProductName: "Ikura"
 - QuantityPerUnit: "12 - 200 ml jars"
 - CategoryID: 8
 - Value: 147.25

The total 'OrderTotal' is 300.25.

Zadanie 2 - modelowanie danych

Zaproponuj strukturę bazy danych dla wybranego/przykładowego zagadnienia/problemu

Należy wybrać jedno zagadnienie/problem (A lub B lub C)

Przykład A

- Wykładowcy, przedmioty, studenci, oceny
 - Wykładowcy prowadzą zajęcia z poszczególnych przedmiotów
 - Studenci uczęszczają na zajęcia
 - Wykładowcy wystawiają oceny studentom
 - Studenci oceniają zajęcia

Przykład B

- Firmy, wycieczki, osoby
 - Firmy organizują wycieczki
 - Osoby rezerwują miejsca/wykupują bilety
 - Osoby oceniają wycieczki

Przykład C

- Własny przykład o podobnym stopniu złożoności

a) Zaproponuj różne warianty struktury bazy danych i dokumentów w poszczególnych kolekcjach oraz przeprowadź dyskusję każdego wariantu (wskazać wady i zalety każdego z wariantów)

- zdefiniuj schemat/reguły walidacji danych
- wykorzystaj referencje
- dokumenty zagnieżdżone
- tablice

b) Kolekcje należy wypełnić przykładowymi danymi

c) W kontekście zaprezentowania wad/zalet należy zaprezentować kilka przykładów/zapytań/operacji oraz dla których dedykowany jest dany wariant

W sprawozdaniu należy zamieścić przykładowe dokumenty w formacie JSON (pkt a) i b)), oraz kod zapytań/operacji (pkt c)), wraz z odpowiednim komentarzem opisującym strukturę dokumentów oraz polecenia ilustrujące wykonanie przykładowych operacji na danych

Do sprawozdania należy kompletny zrzut wykonanych/przygotowanych baz danych (taki zrzut można wykonać np. za pomocą poleceń `mongoexport`, `mongodump` ...) oraz plik z kodem operacji/zapytań w wersji źródłowej (np. plik .js, np. plik .md), załącznik powinien mieć format zip

Zadanie 2 - rozwiązanie

Wybraliśmy przykład B i zaimplementowaliśmy dwie struktury bazy danych.

Wariant 1 - wiele kolekcji

Tworzenie kolekcji wraz z walidacją:

```
db.createCollection("companies", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["name", "address", "email"],
      properties: {
        name: { bsonType: "string" },
        address: {
          bsonType: "object",
          required: ["street", "city", "postalCode", "country"],
          properties: {
            street: { bsonType: "string" },
            city: { bsonType: "string" },
            postalCode: { bsonType: "string" },
            country: { bsonType: "string" }
          }
        },
        phone: { bsonType: "string" },
        email: {
          bsonType: "string",
          pattern: "^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$"
        }
      }
    }
  }
})
```



```

    }
  }
}
});

```

```

db.createCollection("users", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["firstName", "lastName", "email"],
      properties: {
        firstName: { bsonType: "string" },
        lastName: { bsonType: "string" },
        email: {
          bsonType: "string",
          pattern: "^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$"
        },
        phone: { bsonType: "string" },
        dateOfBirth: { bsonType: "date" },
      }
    }
  }
});

```

```

db.createCollection("trips", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["companyId", "name", "destination", "startDate", "endDate",
"price"],
      properties: {
        companyId: { bsonType: "objectId" },
        name: { bsonType: "string" },
        description: { bsonType: "string" },
        destination: { bsonType: "string" },
        startDate: { bsonType: "date" },
        endDate: { bsonType: "date" },
        price: { bsonType: "decimal" },
        maxParticipants: { bsonType: "int", minimum: 1 },
        availableSpots: { bsonType: "int", minimum: 0 },
        ratings: {
          bsonType: "array",
          items: {
            bsonType: "object",
            required: ["userId", "rating", "createdAt"],
            properties: {
              userId: { bsonType: "objectId" },
              rating: { bsonType: "int", minimum: 1, maximum: 5 },
              comment: { bsonType: "string" },
              createdAt: { bsonType: "date" }
            }
          }
        }
      }
    }
  }
});

```

```

    }
  },
},
}
}
});

db.createCollection("bookings", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["userId", "tripId", "bookingDate", "status", "noParticipants",
"totalPrice", "createdAt", "updatedAt"],
      properties: {
        userId: { bsonType: "objectId" },
        tripId: { bsonType: "objectId" },
        bookingDate: { bsonType: "date" },
        status: {
          bsonType: "string",
          enum: ["confirmed and paid", "cancelled", "new"]
        },
        participants: { bsonType: "int", minimum: 1 },
        totalPrice: { bsonType: "decimal" },
        createdAt: { bsonType: "date" },
        updatedAt: { bsonType: "date" }
      }
    }
  }
});

```

Indeksy:

```

db.companies.createIndex({ name: 1 });
db.companies.createIndex({ email: 1 }, { unique: true });

db.users.createIndex({ email: 1 }, { unique: true });
db.users.createIndex({ lastName: 1, firstName: 1 });

db.trips.createIndex({ companyId: 1 });
db.trips.createIndex({ destination: 1 });
db.trips.createIndex({ startDate: 1, endDate: 1 });
db.trips.createIndex({ "ratings.userId": 1 });

db.bookings.createIndex({ userId: 1 });

```

```
db.bookings.createIndex({ tripId: 1 });
db.bookings.createIndex({ status: 1 });
db.bookings.createIndex({ bookingDate: 1 });
```

Przykładowe dane:

```
db.companies.insertMany([
  {
    _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5a"),
    name: "Adventure Tours",
    address: {
      street: "Main St 123",
      city: "Warsaw",
      postalCode: "00-001",
      country: "Poland"
    },
    phone: "+48123456789",
    email: "contact@adventure.com"
  },
  {
    _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5b"),
    name: "Sunny Holidays",
    address: {
      street: "Beach Ave 456",
      city: "Gdansk",
      postalCode: "80-001",
      country: "Poland"
    },
    phone: "+48987654321",
    email: "info@sunnyholidays.com"
  },
  {
    _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5c"),
    name: "Mountain Explorers",
    address: {
      street: "Alpine St 789",
      city: "Zakopane",
      postalCode: "34-500",
      country: "Poland"
    },
    phone: "+48111222333",
    email: "office@mountainexp.com"
  }
]);
```

```
db.users.insertMany([
  {
    _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5d"),
    firstName: "Anna",
    lastName: "Kowalska",
    email: "anna.kowalska@example.com",
    phone: "+48555666777",
    dateOfBirth: new Date("1985-05-15")
  },
  {
    _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5e"),
    firstName: "Jan",
    lastName: "Nowak",
    email: "jan.nowak@example.com",
    phone: "+48666777888",
    dateOfBirth: new Date("1990-11-20")
  },
  {
    _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5f"),
    firstName: "Marek",
    lastName: "Wiśniewski",
    email: "marek.wisniewski@example.com",
    phone: "+48777888999",
    dateOfBirth: new Date("1982-03-10")
  },
  {
    _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f6a"),
    firstName: "Katarzyna",
    lastName: "Lewandowska",
    email: "k.lewandowska@example.com",
    phone: "+48888999000",
    dateOfBirth: new Date("1995-07-22")
  },
  {
    _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f6b"),
    firstName: "Piotr",
    lastName: "Wójcik",
    email: "piotr.wojcik@example.com",
    phone: "+48999000111",
    dateOfBirth: new Date("2004-09-30")
  }
]);

db.trips.insertMany([
  {
    _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f6c"),
    companyId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5a"), // Adventure Tours
    name: "Italian Dream",
    description: "7-day tour through Italy's most beautiful cities",
    destination: "Italy",
    startDate: new Date("2023-06-01"),
    endDate: new Date("2023-06-07"),
  }
]);
```

```
    price: NumberDecimal("2500.00"),
    maxParticipants: 20,
    availableSpots: 15,
    ratings: [
      {
        userId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5d"), // Anna Kowalska
        rating: 5,
        comment: "Amazing experience!",
        createdAt: new Date("2022-12-10")
      }
    ]
  },
  {
    _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f6d"),
    companyId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5b"), // Sunny Holidays
    name: "Greek Islands Cruise",
    description: "10-day cruise around Greek islands",
    destination: "Greece",
    startDate: new Date("2023-07-15"),
    endDate: new Date("2023-07-25"),
    price: NumberDecimal("3500.00"),
    maxParticipants: 15,
    availableSpots: 10,
    ratings: []
  },
  {
    _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f6e"),
    companyId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5c"), // Mountain Explorers
    name: "Alpine Adventure",
    description: "5-day hiking in the Alps",
    destination: "Switzerland",
    startDate: new Date("2023-08-10"),
    endDate: new Date("2023-08-15"),
    price: NumberDecimal("1800.00"),
    maxParticipants: 12,
    availableSpots: 8,
    ratings: [
      {
        userId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5e"), // Jan Nowak
        rating: 4,
        comment: "Great views but challenging trails",
        createdAt: new Date("2022-11-15")
      }
    ]
  },
  {
    _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f6f"),
    companyId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5a"), // Adventure Tours
    name: "Spanish Fiesta",
    description: "8-day tour of Spain with flamenco shows",
    destination: "Spain",
    startDate: new Date("2023-09-05"),
    endDate: new Date("2023-09-13"),
    price: NumberDecimal("2800.00"),
```

```

    maxParticipants: 18,
    availableSpots: 12,
    ratings: []
  },
  {
    _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f7a"),
    companyId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5b"), // Sunny Holidays
    name: "Egyptian Wonders",
    description: "6-day tour of pyramids and Nile cruise",
    destination: "Egypt",
    startDate: new Date("2023-10-20"),
    endDate: new Date("2023-10-26"),
    price: NumberDecimal("2200.00"),
    maxParticipants: 16,
    availableSpots: 16,
    ratings: []
  }
]);

```

```

db.bookings.insertMany([
  {
    _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f7b"),
    userId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5d"), // Anna Kowalska
    tripId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f6c"), // Italian Dream
    bookingDate: new Date("2023-01-15"),
    status: "confirmed and paid",
    noParticipants: 2,
    totalPrice: NumberDecimal("5000.00"),
    createdAt: new Date("2023-01-15T10:30:00Z"),
    updatedAt: new Date("2023-01-15T10:30:00Z")
  },
  {
    _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f7c"),
    userId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5e"), // Jan Nowak
    tripId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f6e"), // Alpine Adventure
    bookingDate: new Date("2023-02-20"),
    status: "confirmed and paid",
    noParticipants: 1,
    totalPrice: NumberDecimal("1800.00"),
    createdAt: new Date("2023-02-20T14:15:00Z"),
    updatedAt: new Date("2023-02-20T14:15:00Z")
  },
  {
    _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f7d"),
    userId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5f"), // Marek Wiśniewski
    tripId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f6d"), // Greek Islands Cruise
    bookingDate: new Date("2023-03-05"),
    status: "new",
    noParticipants: 3,
    totalPrice: NumberDecimal("10500.00"),
    createdAt: new Date("2023-03-05T09:45:00Z"),

```

```

    updatedAt: new Date("2023-03-05T09:45:00Z")
  },
  {
    _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f7e"),
    userId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f6a"), // Katarzyna Lewandowska
    tripId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f6f"), // Spanish Fiesta
    bookingDate: new Date("2023-04-10"),
    status: "confirmed and paid",
    noParticipants: 2,
    totalPrice: NumberDecimal("5600.00"),
    createdAt: new Date("2023-04-10T16:20:00Z"),
    updatedAt: new Date("2023-04-10T16:20:00Z")
  },
  {
    _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f7f"),
    userId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f6b"), // Piotr Wójcik
    tripId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f7a"), // Egyptian Wonders
    bookingDate: new Date("2023-05-15"),
    status: "cancelled",
    noParticipants: 4,
    totalPrice: NumberDecimal("8800.00"),
    createdAt: new Date("2023-05-15T11:10:00Z"),
    updatedAt: new Date("2023-05-20T09:30:00Z")
  }
]);

```

Wariant 2 - zdenormalizowana wersja z jedną kolekcją - trips

```

db.createCollection("trips", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["company", "name", "destination", "startDate", "endDate",
"price"],
      properties: {
        company: {
          bsonType: "object",
          required: ["name", "email"],
          properties: {
            name: { bsonType: "string" },
            address: {
              bsonType: "object",
              properties: {
                street: { bsonType: "string" },
                city: { bsonType: "string" },
                postalCode: { bsonType: "string" },

```

```

        country: { bsonType: "string" }
    },
    phone: { bsonType: "string" },
    email: {
        bsonType: "string",
        pattern: "^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$"
    }
},
name: { bsonType: "string" },
description: { bsonType: "string" },
destination: { bsonType: "string" },
startDate: { bsonType: "date" },
endDate: { bsonType: "date" },
price: { bsonType: "decimal" },
maxParticipants: { bsonType: "int", minimum: 1 },
availableSpots: { bsonType: "int", minimum: 0 },
bookings: {
    bsonType: "array",
    items: {
        bsonType: "object",
        required: ["userId", "user", "bookingDate", "status",
            "noParticipants", "totalPrice"],
        properties: {
            userId: { bsonType: "objectId" },
            user: {
                bsonType: "object",
                required: ["firstName", "lastName", "email"],
                properties: {
                    firstName: { bsonType: "string" },
                    lastName: { bsonType: "string" },
                    email: { bsonType: "string" },
                    phone: { bsonType: "string" },
                    dateOfBirth: { bsonType: "date" }
                }
            },
            bookingDate: { bsonType: "date" },
            status: {
                bsonType: "string",
                enum: ["confirmed and paid", "cancelled", "new"]
            },
            noParticipants: { bsonType: "int", minimum: 1 },
            totalPrice: { bsonType: "decimal" },
            createdAt: { bsonType: "date" },
            updatedAt: { bsonType: "date" }
        }
    },
},
ratings: {
    bsonType: "array",
    items: {
        bsonType: "object",
        required: ["userId", "user", "rating"],

```



```

        properties: {
            userId: { bsonType: "objectId" },
            user: {
                bsonType: "object",
                properties: {
                    firstName: { bsonType: "string" },
                    lastName: { bsonType: "string" }
                }
            },
            rating: { bsonType: "int", minimum: 1, maximum: 5 },
            comment: { bsonType: "string" }
        }
    }
}
}
}
});

```

Indeksy:

```

db.trips.createIndex({ "company.name": 1 });
db.trips.createIndex({ destination: 1 });
db.trips.createIndex({ startDate: 1, endDate: 1 });
db.trips.createIndex({ "bookings.userId": 1 });
db.trips.createIndex({ "ratings.userId": 1 });

```

Przykładowe dane:

```

db.trips.insertMany([
    {
        _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f6c"),
        company: {
            name: "Adventure Tours",
            address: {
                street: "Main St 123",
                city: "Warsaw",
                postalCode: "00-001",
                country: "Poland"
            },
            phone: "+48123456789",

```

```

    email: "contact@adventure.com"
  },
  name: "Italian Dream",
  description: "7-day tour through Italy's most beautiful cities",
  destination: "Italy",
  startDate: new Date("2023-06-01"),
  endDate: new Date("2023-06-07"),
  price: NumberDecimal("2500.00"),
  maxParticipants: 20,
  availableSpots: 15,
  bookings: [
    {
      _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f7b"),
      userId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5d"),
      user: {
        firstName: "Anna",
        lastName: "Kowalska",
        email: "anna.kowalska@example.com",
        phone: "+48555666777",
        dateOfBirth: new Date("1985-05-15")
      },
      bookingDate: new Date("2023-01-15"),
      status: "confirmed and paid",
      noParticipants: 2,
      totalPrice: NumberDecimal("5000.00"),
      createdAt: new Date("2023-01-15T10:30:00Z"),
      updatedAt: new Date("2023-01-15T10:30:00Z")
    }
  ],
  ratings: [
    {
      userId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5d"),
      user: {
        firstName: "Anna",
        lastName: "Kowalska"
      },
      rating: 5,
      comment: "Amazing experience!"
    }
  ]
},
{
  _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f6d"),
  company: {
    name: "Sunny Holidays",
    address: {
      street: "Beach Ave 456",
      city: "Gdansk",
      postalCode: "80-001",
      country: "Poland"
    },
    phone: "+48987654321",
    email: "info@sunnyholidays.com"
  },

```

```

    name: "Greek Islands Cruise",
    description: "10-day cruise around Greek islands",
    destination: "Greece",
    startDate: new Date("2023-07-15"),
    endDate: new Date("2023-07-25"),
    price: NumberDecimal("3500.00"),
    maxParticipants: 15,
    availableSpots: 10,
    bookings: [
      {
        _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f7c"),
        userId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5e"),
        user: {
          firstName: "Jan",
          lastName: "Nowak",
          email: "jan.nowak@example.com",
          phone: "+48666777888",
          dateOfBirth: new Date("1990-11-20")
        },
        bookingDate: new Date("2023-02-20"),
        status: "confirmed and paid",
        noParticipants: 1,
        totalPrice: NumberDecimal("3500.00"),
        createdAt: new Date("2023-02-20T14:15:00Z"),
        updatedAt: new Date("2023-02-20T14:15:00Z")
      }
    ],
    ratings: []
  },
  {
    _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f6e"),
    company: {
      name: "Mountain Explorers",
      address: {
        street: "Alpine St 789",
        city: "Zakopane",
        postalCode: "34-500",
        country: "Poland"
      },
      phone: "+48111222333",
      email: "office@mountainexp.com"
    },
    name: "Alpine Adventure",
    description: "5-day hiking in the Alps",
    destination: "Switzerland",
    startDate: new Date("2023-08-10"),
    endDate: new Date("2023-08-15"),
    price: NumberDecimal("1800.00"),
    maxParticipants: 12,
    availableSpots: 8,
    bookings: [
      {
        _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f7d"),
        userId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5f"),

```

```
    user: {
      firstName: "Marek",
      lastName: "Wiśniewski",
      email: "marek.wisniewski@example.com",
      phone: "+48777888999",
      dateOfBirth: new Date("1982-03-10")
    },
    bookingDate: new Date("2023-03-05"),
    status: "new",
    noParticipants: 3,
    totalPrice: NumberDecimal("10500.00"),
    createdAt: new Date("2023-03-05T09:45:00Z"),
    updatedAt: new Date("2023-03-05T09:45:00Z")
  }
],
ratings: [
  {
    userId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f5e"),
    user: {
      firstName: "Jan",
      lastName: "Nowak"
    },
    rating: 4,
    comment: "Great views but challenging trails"
  }
]
},
{
  _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f6f"),
  company: {
    name: "Adventure Tours",
    address: {
      street: "Main St 123",
      city: "Warsaw",
      postalCode: "00-001",
      country: "Poland"
    },
    phone: "+48123456789",
    email: "contact@adventure.com"
  },
  name: "Spanish Fiesta",
  description: "8-day tour of Spain with flamenco shows",
  destination: "Spain",
  startDate: new Date("2023-09-05"),
  endDate: new Date("2023-09-13"),
  price: NumberDecimal("2800.00"),
  maxParticipants: 18,
  availableSpots: 12,
  bookings: [
    {
      _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f7e"),
      userId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f6a"),
      user: {
        firstName: "Katarzyna",
```

```
        lastName: "Lewandowska",
        email: "k.lewandowska@example.com",
        phone: "+48888999000",
        dateOfBirth: new Date("1995-07-22")
    },
    bookingDate: new Date("2023-04-10"),
    status: "confirmed and paid",
    noParticipants: 2,
    totalPrice: NumberDecimal("5600.00"),
    createdAt: new Date("2023-04-10T16:20:00Z"),
    updatedAt: new Date("2023-04-10T16:20:00Z")
  }
],
ratings: []
},
{
  _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f7a"),
  company: {
    name: "Sunny Holidays",
    address: {
      street: "Beach Ave 456",
      city: "Gdansk",
      postalCode: "80-001",
      country: "Poland"
    },
    phone: "+48987654321",
    email: "info@sunnyholidays.com"
  },
  name: "Egyptian Wonders",
  description: "6-day tour of pyramids and Nile cruise",
  destination: "Egypt",
  startDate: new Date("2023-10-20"),
  endDate: new Date("2023-10-26"),
  price: NumberDecimal("2200.00"),
  maxParticipants: 16,
  availableSpots: 16,
  bookings: [
    {
      _id: ObjectId("5f8d8a7b2f4d4a1d2c3e4f7f"),
      userId: ObjectId("5f8d8a7b2f4d4a1d2c3e4f6b"),
      user: {
        firstName: "Piotr",
        lastName: "Wójcik",
        email: "piotr.wojcik@example.com",
        phone: "+48999000111",
        dateOfBirth: new Date("2004-09-30")
      },
      bookingDate: new Date("2023-05-15"),
      status: "cancelled",
      noParticipants: 4,
      totalPrice: NumberDecimal("8800.00"),
      createdAt: new Date("2023-05-15T11:10:00Z"),
      updatedAt: new Date("2023-05-20T09:30:00Z")
    }
  ]
}
```

```
    ],  
    ratings: []  
  }  
]);
```

Wariant 1 - przykładowa operacja:

```
db.trips.find({"name" : "Greek Islands Cruise"})  
{  
  _id: ObjectId('5f8d8a7b2f4d4a1d2c3e4f6d'),  
  companyId: ObjectId('5f8d8a7b2f4d4a1d2c3e4f5b'),  
  name: 'Greek Islands Cruise',  
  description: '10-day cruise around Greek islands',  
  destination: 'Greece',  
  startDate: 2023-07-15T00:00:00.000Z,  
  endDate: 2023-07-25T00:00:00.000Z,  
  price: Decimal128('3500.00'),  
  maxParticipants: 15,  
  availableSpots: 10,  
  ratings: []  
}  
  
db.companies.find({_id: ObjectId('5f8d8a7b2f4d4a1d2c3e4f5b')})  
{  
  _id: ObjectId('5f8d8a7b2f4d4a1d2c3e4f5b'),  
  name: 'Sunny Holidays',  
  address: {  
    street: 'Beach Ave 456',  
    city: 'Gdansk',  
    postalCode: '80-001',  
    country: 'Poland'  
  },  
  phone: '+48987654321',  
  email: 'info@sunnyholidays.com'  
}
```

Wariant 2 - przykładowa operacja:

```
> db.trips.find({"name" : "Greek Islands Cruise"})
< {
  _id: ObjectId('5f8d8a7b2f4d4a1d2c3e4f6d'),
  company: {
    name: 'Sunny Holidays',
    address: {
      street: 'Beach Ave 456',
      city: 'Gdansk',
      postalCode: '80-001',
      country: 'Poland'
    },
    phone: '+48987654321',
    email: 'info@sunnyholidays.com'
  },
  name: 'Greek Islands Cruise',
  description: '10-day cruise around Greek islands',
  destination: 'Greece',
  startDate: 2023-07-15T00:00:00.000Z,
  endDate: 2023-07-25T00:00:00.000Z,
  price: Decimal128('3500.00'),
  maxParticipants: 15,
  availableSpots: 10,
```

Porównanie wariantów:

1. Struktura danych

- **Znormalizowany:** Rozbicie na wiele kolekcji (*companies*, *users*, *trips*, *bookings*) z relacjami przez *ObjectId*.
- **Zdenormalizowany:** Wszystkie dane osadzone w jednej kolekcji *trips*.

2. Wydajność odczytu

- **Znormalizowany:** Wymaga *\$lookup* - wolniejsze zapytania.
- **Zdenormalizowany:** Dane od razu dostępne - szybsze odczyty.

3. Aktualizacja danych

- **Znormalizowany:** Łatwiejsze, zmiana w jednym miejscu .
- **Zdenormalizowany:** Trudniejsze, dane powielone w wielu dokumentach.

4. Spójność danych

- **Znormalizowany:** Lepsza spójność, brak duplikatów.
- **Zdenormalizowany:** Ryzyko niespójności, np. różne wersje danych użytkownika (patrz 3)).

Punktacja:

| zadanie | pkt |
|---------|-----|
| 1 | 1 |
| 2 | 1 |
| razem | 2 |