

Dokumentowe bazy danych – MongoDB

Ćwiczenie/zadanie

Imiona i nazwiska autorów: Jakub Psarski, Dariusz Rozmus

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": ... 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": ...
    }
  }
]

```

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 poaje 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)

Załóżmy 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",
    },
  },
  {
    $unwind: "$Orderdetails",
  },
  {
    $lookup: {
      from: "shippers",
      localField: "ShipVia",
      foreignField: "ShipperID",
      as: "Shipment.Shipper",
    },
  },
  {
    $unwind: "$Shipment.Shipper",
  },
  {
    $lookup: {
      from: "products",
      localField: "Orderdetails.ProductID",
      foreignField: "ProductID",
      as: "Orderdetails.product",
    },
  },
  {
    $unwind: "$Orderdetails.product",
  },
  {
    $lookup: {
      from: "categories",
      localField: "Orderdetails.product.CategoryID",
      foreignField: "CategoryID",
      as: "Orderdetails.product.category",
    },
  },
  {
    $unwind: "$Orderdetails.product.category",
  },
  {
    $group: {
```

```

    _id: "$_id",
    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: "$RequiredDate",
        ShippedDate: "$ShippedDate",
      },
    },
    Freight: { $first: "$Freight" },
    Shipment: {
      $first: {
        Shipper: "$Shipment.Shipper",
        ShipName: "$ShipName",
        ShipAddress: "$ShipAddress",
        ShipCity: "$ShipCity",
        ShipCountry: "$ShipCountry",
      },
    },
    Orderdetails: {
      $push: {
        UnitPrice: "$Orderdetails.UnitPrice",
        Quantity: "$Orderdetails.Quantity",
        Discount: { $round: ["$Orderdetails.Discount", 2] },
        Value: {
          $round: [
            {
              $multiply: [
                "$Orderdetails.UnitPrice",
                "$Orderdetails.Quantity",
                {
                  $subtract: [1, "$Orderdetails.Discount"],
                },
              ],
            },
            2,
          ],
        },
      },
      product: {
        ProductID: "$Orderdetails.product.ProductID",
        ProductName: "$Orderdetails.product.ProductName",
        QuantityPerUnit: "$Orderdetails.product.QuantityPerUnit",
        CategoryID: "$Orderdetails.product.CategoryID",
        CategoryName: "$Orderdetails.product.category.CategoryName",
      },
    },
  },
},
{
  $project: {
    _id: 1,
    OrderID: 1,
    Customer: 1,
    Employee: 1,
    Dates: { OrderDate: 1, RequiredDate: 1 },
    Orderdetails: 1,
    Freight: 1,
    OrderTotal: { $sum: "$Orderdetails.Value" },
    Shipment: 1,
  },
}

```

```

    },
    {
      $out: "OrdersInfo",
    },
  ],
});

```

Działanie:

```
db.OrdersInfo.find().limit(2);
```

```

[
  {
    "_id": {
      "$oid": "63a060b9bb3b972d6f4e1fe2"
    },
    "OrderID": 10276,
    "Customer": {
      "CustomerID": "TORTU",
      "CompanyName": "Tortuga Restaurante",
      "City": "México D.F.",
      "Country": "Mexico"
    },
    "Employee": {
      "EmployeeID": 8,
      "FirstName": "Laura",
      "LastName": "Callahan",
      "Title": "Inside Sales Coordinator"
    },
    "Dates": {
      "OrderDate": {
        "$date": "1996-08-08T00:00:00Z"
      },
      "RequiredDate": {
        "$date": "1996-08-22T00:00:00Z"
      }
    },
    "Freight": 13.84,
    "Shipment": {
      "Shipper": {
        "_id": {
          "$oid": "63a05e60bb3b972d6f4e0abc"
        },
        "ShipperID": 3,
        "CompanyName": "Federal Shipping",
        "Phone": "(503) 555-9931"
      },
      "ShipName": "Tortuga Restaurante",
      "ShipAddress": "Avda. Azteca 123",
      "ShipCity": "México D.F.",
      "ShipCountry": "Mexico"
    },
    "Orderdetails": [
      {
        "UnitPrice": 24.8,
        "Quantity": 15,
        "Discount": 0,
        "Value": 372,
        "product": {
          "ProductID": 10,
          "ProductName": "Ikura",
          "QuantityPerUnit": "12 - 200 ml jars",
          "CategoryID": 8,
          "CategoryName": "Seafood"
        }
      },
      (...)
    ],
    "OrderTotal": 420
  },
  {
    "_id": {
      "$oid": "63a060b9bb3b972d6f4e208a"
    },

```

```

    },
    "OrderID": 10444,
    "Customer": {
      "CustomerID": "BERGS",
      "CompanyName": "Berglunds snabbköp",
      "City": "Luleå",
      "Country": "Sweden"
    },
    "Employee": {
      "EmployeeID": 3,
      "FirstName": "Janet",
      "LastName": "Leverling",
      "Title": "Sales Representative"
    },
    "Dates": {
      "OrderDate": {
        "$date": "1997-02-12T00:00:00Z"
      },
      "RequiredDate": {
        "$date": "1997-03-12T00:00:00Z"
      }
    },
    "Freight": 3.5,
    "Shipment": {
      "Shipper": {
        "_id": {
          "$oid": "63a05e60bb3b972d6f4e0abc"
        },
        "ShipperID": 3,
        "CompanyName": "Federal Shipping",
        "Phone": "(503) 555-9931"
      },
      "ShipName": "Berglunds snabbköp",
      "ShipAddress": "Berguvsvägen 8",
      "ShipCity": "Luleå",
      "ShipCountry": "Sweden"
    },
    "Orderdetails": [
      {
        "UnitPrice": 31.2,
        "Quantity": 10,
        "Discount": 0,
        "Value": 312,
        "product": {
          "ProductID": 17,
          "ProductName": "Alice Mutton",
          "QuantityPerUnit": "20 - 1 kg tins",
          "CategoryID": 6,
          "CategoryName": "Meat/Poultry"
        }
      },
      (...)
    ],
    "OrderTotal": 1031.7
  }
]

```

b)

```

db.OrdersInfo.aggregate([
  {
    $group: {
      _id: "$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",
        },
      },
    },
  },
  {
    $project: {
      _id: 0,
      CustomerID: "$_id",
      CompanyName: 1,
      City: 1,
      Country: 1,
      Orders: 1,
    },
  },
  {
    $out: "CustomerInfo",
  },
]);

```

Działanie:

```
db.CustomerInfo.find().limit(2);
```

```

[
  {
    "_id": {
      "$oid": "681a8945f19010a72fffbcb7"
    },
    "CompanyName": "Richter Supermarkt",
    "City": "Genève",
    "Country": "Switzerland",
    "Orders": [
      {
        "OrderID": 10758,
        "Employee": {
          "EmployeeID": 3,
          "FirstName": "Janet",
          "LastName": "Leverling",
          "Title": "Sales Representative"
        },
        "Dates": {
          "OrderDate": {
            "$date": "1997-11-28T00:00:00Z"
          },
          "RequiredDate": {
            "$date": "1997-12-26T00:00:00Z"
          }
        },
        "Orderdetails": [
          {
            "UnitPrice": 31.23,
            "Quantity": 20,

```



```

        "Discount": 0,
        "Value": 624.6,
        "product": {
            "ProductID": 26,
            "ProductName": "Gumbär Gummibärchen",
            "QuantityPerUnit": "100 - 250 g bags",
            "CategoryID": 3,
            "CategoryName": "Confections"
        }
    },
    (...)
],
"Freight": 138.17,
"OrderTotal": 1644.6,
"Shipment": {
    "Shipper": {
        "_id": {
            "$oid": "63a05e60bb3b972d6f4e0abc"
        },
        "ShipperID": 3,
        "CompanyName": "Federal Shipping",
        "Phone": "(503) 555-9931"
    },
    "ShipName": "Richter Supermarkt",
    "ShipAddress": "Starenweg 5",
    "ShipCity": "Genève",
    "ShipCountry": "Switzerland"
}
},
(...)
],
"CustomerID": "RICSU"
},
{
    "_id": {
        "$oid": "681a8945f19010a72fffbcb8"
    },
    "CompanyName": "Rattlesnake Canyon Grocery",
    "City": "Albuquerque",
    "Country": "USA",
    "Orders": [
        {
            "OrderID": 10401,
            "Employee": {
                "EmployeeID": 1,
                "FirstName": "Nancy",
                "LastName": "Davolio",
                "Title": "Sales Representative"
            },
            "Dates": {
                "OrderDate": {
                    "$date": "1997-01-01T00:00:00Z"
                },
                "RequiredDate": {
                    "$date": "1997-01-29T00:00:00Z"
                }
            },
            "Orderdetails": [
                {
                    "UnitPrice": 20.7,
                    "Quantity": 18,
                    "Discount": 0,
                    "Value": 372.6,
                    "product": {
                        "ProductID": 30,
                        "ProductName": "Nord-Ost Matjeshering",
                        "QuantityPerUnit": "10 - 200 g glasses",
                        "CategoryID": 8,
                        "CategoryName": "Seafood"
                    }
                },
                (...)
            ],
            "Freight": 12.51,
            "OrderTotal": 3868.6,
            "Shipment": {
                "Shipper": {

```

```

        "_id": {
            "$oid": "63a05e60bb3b972d6f4e0aba"
        },
        "ShipperID": 1,
        "CompanyName": "Speedy Express",
        "Phone": "(503) 555-9831"
    },
    "ShipName": "Rattlesnake Canyon Grocery",
    "ShipAddress": "2817 Milton Dr.",
    "ShipCity": "Albuquerque",
    "ShipCountry": "USA"
    }
},
(...)
],
"CustomerID": "RATTC"
}
]

```

c)

```

//1. Oryginalne kolekcje:
db.customers.aggregate([
{
    $lookup: {
        from: "orders",
        localField: "CustomerID",
        foreignField: "CustomerID",
        as: "Orders",
    },
},
{ $unwind: "$Orders" },
{
    $match: {
        "Orders.OrderDate": {
            $gte: new ISODate("1997-01-01"),
            $lt: new ISODate("1998-01-01"),
        },
    },
},
{
    $lookup: {
        from: "orderdetails",
        localField: "Orders.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" } },
{
    $group: {
        _id: "$CustomerID",
        CompanyName: { $first: "$CompanyName" },
        ConfectionsSale: {
            $sum: {

```

```

        $multiply: [
            "$Orderdetails.UnitPrice",
            "$Orderdetails.Quantity",
            { $subtract: [1, "$Orderdetails.Discount"] },
        ],
    },
},
{
    $project: {
        _id: 0,
        CustomerID: "$_id",
        CompanyName: 1,
        ConfectionsSale97: { $round: ["$ConfectionsSale", 2] },
    },
},
{
    $sort: { CustomerID: 1 },
},
]);

```

```

//2. OrdersInfo:
db.OrdersInfo.aggregate([
    {
        $match: {
            "Dates.OrderDate": {
                $gte: new ISODate("1997-01-01"),
                $lt: new ISODate("1998-01-01"),
            },
        },
    },
    { $unwind: "$Orderdetails" },
    {
        $match: {
            "Orderdetails.product.CategoryName": "Confections",
        },
    },
    {
        $group: {
            _id: "$Customer.CustomerID",
            CompanyName: { $first: "$Customer.CompanyName" },
            ConfectionsSale: { $sum: "$Orderdetails.Value" },
        },
    },
    {
        $project: {
            _id: 0,
            CustomerID: "$_id",
            CompanyName: 1,
            ConfectionsSale97: { $round: ["$ConfectionsSale", 2] },
        },
    },
    {
        $sort: { CustomerID: 1 },
    },
]);

```

```

// 3. CustomerInfo:
db.CustomerInfo.aggregate([
    { $unwind: "$Orders" },
    {
        $match: {
            "Orders.Dates.OrderDate": {
                $gte: new ISODate("1997-01-01"),
                $lt: new ISODate("1998-01-01"),
            },
        },
    },
    { $unwind: "$Orders.Orderdetails" },
    {
        $match: {
            "Orders.Orderdetails.product.CategoryName": "Confections",
        },
    },
    {
        $group: {

```

```

    _id: "$CustomerID",
    CompanyName: { $first: "$CompanyName" },
    ConfectionsSale: { $sum: "$Orders.Orderdetails.Value" },
  },
},
{
  $project: {
    _id: 0,
    CustomerID: "$_id",
    CompanyName: 1,
    ConfectionsSale97: { $round: ["$ConfectionsSale", 2] },
  },
},
{
  $sort: { CustomerID: 1 },
},
]);

```

Działanie:

```

[
  {
    "CompanyName": "Antonio Moreno Taquería",
    "CustomerID": "ANTON",
    "ConfectionsSale97": 958.92
  },
  {
    "CompanyName": "Around the Horn",
    "CustomerID": "AROUT",
    "ConfectionsSale97": 375.2
  },
  {
    "CompanyName": "Berglunds snabbköp",
    "CustomerID": "BERGS",
    "ConfectionsSale97": 561.96
  },
  {
    "CompanyName": "Blauer See Delikatessen",
    "CustomerID": "BLAUS",
    "ConfectionsSale97": 80
  },
  {
    "CompanyName": "Blondesddsl père et fils",
    "CustomerID": "BLONP",
    "ConfectionsSale97": 1379
  },
  {
    "CompanyName": "Bon app'",
    "CustomerID": "BONAP",
    "ConfectionsSale97": 462.41
  },
  (...)
]

```

(output identyczny dla wszystkich trzech wariantów)

Porównanie

- **Oryginalne kolekcje:** Wymaga wielu `$lookup` i `$unwind`, co zwiększa złożoność i czas wykonania.
- **OrdersInfo:** Prostsze zapytanie dzięki agregacji danych w jednej kolekcji.
- **CustomerInfo:** Zapytanie zbliżone złożonością do wersji wykorzystującej kolekcję **OrdersInfo**, wymaga obsługi zagnieżdżonych danych.

d)

```
// 1. Oryginalne kolekcje:
db.orders.aggregate([
  {
    $lookup: {
      from: "orderdetails",
      localField: "OrderID",
      foreignField: "OrderID",
      as: "details",
    },
  },
  { $unwind: "$details" },
  {
    $lookup: {
      from: "customers",
      localField: "CustomerID",
      foreignField: "CustomerID",
      as: "customer",
    },
  },
  { $unwind: "$customer" },
  {
    $project: {
      CustomerID: "$CustomerID",
      CompanyName: "$customer.CompanyName",
      year: { $year: "$OrderDate" },
      month: { $month: "$OrderDate" },
      orderValue: {
        $multiply: [
          "$details.UnitPrice",
          "$details.Quantity",
          { $subtract: [1, "$details.Discount"] },
        ],
      },
    },
  },
  {
    $group: {
      _id: {
        CustomerID: "$CustomerID",
        CompanyName: "$CompanyName",
        year: "$year",
        month: "$month",
      },
      totalSales: { $sum: "$orderValue" },
    },
  },
  {
    $group: {
      _id: {
        CustomerID: "$_id.CustomerID",
        CompanyName: "$_id.CompanyName",
      },
      Sale: {
        $push: {
          Year: "$_id.year",
          Month: "$_id.month",
          Total: { $round: ["$totalSales", 2] },
        },
      },
    },
  },
  {
    $project: {
      _id: 0,
      CustomerID: "$_id.CustomerID",
      CompanyName: "$_id.CompanyName",
      Sale: 1,
    },
  },
  { $sort: { CustomerID: 1 } },
]);

// 2. OrdersInfo:
```

```

db.OrdersInfo.aggregate([
  {
    $project: {
      CustomerID: "$Customer.CustomerID",
      CompanyName: "$Customer.CompanyName",
      year: { $year: "$Dates.OrderDate" },
      month: { $month: "$Dates.OrderDate" },
      orderValue: "$OrderTotal",
    },
  },
  {
    $group: {
      _id: {
        CustomerID: "$CustomerID",
        CompanyName: "$CompanyName",
        year: "$year",
        month: "$month",
      },
      totalSales: { $sum: "$orderValue" },
    },
  },
  {
    $group: {
      _id: {
        CustomerID: "$_id.CustomerID",
        CompanyName: "$_id.CompanyName",
      },
      Sale: {
        $push: {
          Year: "$_id.year",
          Month: "$_id.month",
          Total: { $round: ["$totalSales", 2] },
        },
      },
    },
  },
  {
    $project: {
      _id: 0,
      CustomerID: "$_id.CustomerID",
      CompanyName: "$_id.CompanyName",
      Sale: 1,
    },
  },
  { $sort: { CustomerID: 1 } },
]);

// 3. CustomerInfo:
db.CustomerInfo.aggregate([
  { $unwind: "$Orders" },
  {
    $project: {
      CustomerID: "$CustomerID",
      CompanyName: "$CompanyName",
      year: { $year: "$Orders.Dates.OrderDate" },
      month: { $month: "$Orders.Dates.OrderDate" },
      orderValue: "$Orders.OrderTotal",
    },
  },
  {
    $group: {
      _id: {
        CustomerID: "$CustomerID",
        CompanyName: "$CompanyName",
        year: "$year",
        month: "$month",
      },
      totalSales: { $sum: "$orderValue" },
    },
  },
  {
    $group: {
      _id: {
        CustomerID: "$_id.CustomerID",
        CompanyName: "$_id.CompanyName",
      },
      Sale: {

```

```

    $push: {
      Year: "$_id.year",
      Month: "$_id.month",
      Total: { $round: [ "$totalSales", 2 ] },
    },
  },
},
{
  $project: {
    _id: 0,
    CustomerID: "$_id.CustomerID",
    CompanyName: "$_id.CompanyName",
    Sale: 1,
  },
},
{ $sort: { CustomerID: 1 } },
]);

```

Działanie:

```

[
  {
    "Sale": [
      {
        "Year": 1998,
        "Month": 1,
        "Total": 845.8
      },
      {
        "Year": 1998,
        "Month": 3,
        "Total": 471.2
      },
      {
        "Year": 1998,
        "Month": 4,
        "Total": 933.5
      },
      {
        "Year": 1997,
        "Month": 10,
        "Total": 1208
      },
      {
        "Year": 1997,
        "Month": 8,
        "Total": 814.5
      }
    ],
    "CustomerID": "ALFKI",
    "CompanyName": "Alfreds Futterkiste"
  },
  {
    "Sale": [
      {
        "Year": 1997,
        "Month": 8,
        "Total": 479.75
      },
      {
        "Year": 1998,
        "Month": 3,
        "Total": 514.4
      },
      {
        "Year": 1996,
        "Month": 9,
        "Total": 88.8
      },
      {
        "Year": 1997,
        "Month": 11,
        "Total": 320
      }
    ]
  }
]

```

```

    }
  ],
  "CustomerID": "ANATR",
  "CompanyName": "Ana Trujillo Emparedados y helados"
},
(... )
]

```

(output identyczny dla wszystkich trzech wariantów)

Porównanie

- **Oryginalne kolekcje:** Wymaga wielu `$lookup` i `$unwind`, co zwiększa złożoność i czas wykonania.
- **OrdersInfo:** Prostsze zapytanie dzięki agregacji danych w jednej kolekcji.
- **CustomerInfo:** Zapytanie zbliżone złożonością do wersji wykorzystującej kolekcję **OrdersInfo**, wymaga obsługi zagnieżdżonych danych.

e)

```

// 1. Oryginalne kolekcje:
let maxOrderID = db.orders
  .find()
  .sort({ OrderID: -1 })
  .limit(1)
  .toArray()[0].OrderID;
let newOrderID = maxOrderID + 1;

db.orders.insertOne({
  OrderID: newOrderID,
  CustomerID: "ALFKI",
  EmployeeID: 5,
  OrderDate: new Date("2025-05-06"),
  RequiredDate: new Date("2025-05-15"),
  ShippedDate: null,
  ShipVia: 2,
  Freight: 25.5,
  ShipName: "Alfreds Futterkiste",
  ShipAddress: "Obere Str. 57",
  ShipCity: "Berlin",
  ShipRegion: null,
  ShipPostalCode: "12209",
  ShipCountry: "Germany",
});

db.orderdetails.insertMany([
  {
    OrderID: newOrderID,
    ProductID: 1,
    UnitPrice: 18.0,
    Quantity: 5,
    Discount: 0.0,
  },
  {
    OrderID: newOrderID,
    ProductID: 10,
    UnitPrice: 31.0,
    Quantity: 3,
    Discount: 0.05,
  },
]);

// 2. OrdersInfo:
let maxOrderID_ = db.OrdersInfo.find()
  .sort({ OrderID: -1 })
  .limit(1)
  .toArray()[0].OrderID;
let newOrderID_ = maxOrderID_ + 1;

let chaiProduct = db.products.findOne({ ProductID: 1 });
let ikuraProduct = db.products.findOne({ ProductID: 10 });
let chaiCategory = db.categories.findOne({
  CategoryID: chaiProduct.CategoryID,
});
let ikuraCategory = db.categories.findOne({
  CategoryID: ikuraProduct.CategoryID,
});

```



```

});

let customer = db.customers.findOne({ CustomerID: "ALFKI" });
let employee = db.employees.findOne({ EmployeeID: 5 });
let shipper = db.shippers.findOne({ ShipperID: 2 });

let chaiValue = 18.0 * 5 * (1 - 0);
let ikuraValue = 31.0 * 3 * (1 - 0.05);
let orderTotal = chaiValue + ikuraValue;

db.OrdersInfo.insertOne({
  OrderID: newOrderID_,
  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: new Date("2025-05-06"),
    RequiredDate: new Date("2025-05-15"),
  },
  Orderdetails: [
    {
      UnitPrice: 18.0,
      Quantity: 5,
      Discount: 0,
      Value: 90.0,
      product: {
        ProductID: chaiProduct.ProductID,
        ProductName: chaiProduct.ProductName,
        QuantityPerUnit: chaiProduct.QuantityPerUnit,
        CategoryID: chaiProduct.CategoryID,
        CategoryName: chaiCategory.CategoryName,
      },
    },
    {
      UnitPrice: 31.0,
      Quantity: 3,
      Discount: 0.05,
      Value: 88.35,
      product: {
        ProductID: ikuraProduct.ProductID,
        ProductName: ikuraProduct.ProductName,
        QuantityPerUnit: ikuraProduct.QuantityPerUnit,
        CategoryID: ikuraProduct.CategoryID,
        CategoryName: ikuraCategory.CategoryName,
      },
    },
  ],
  Freight: 25.5,
  OrderTotal: 178.35,
  Shipment: {
    Shipper: {
      _id: shipper._id,
      ShipperID: shipper.ShipperID,
      CompanyName: shipper.CompanyName,
      Phone: shipper.Phone,
    },
    ShipName: "Alfreds Futterkiste",
    ShipAddress: "Obere Str. 57",
    ShipCity: "Berlin",
    ShipCountry: "Germany",
  },
});

// 3. CustomerInfo:
let maxOrderID__ = db.orders
  .find()
  .sort({ OrderID: -1 })
  .limit(1)

```

```

.toArray()[0].OrderID;
let newOrderID__ = maxOrderID__ + 1;

let chaiProduct_ = db.products.findOne({ ProductID: 1 });
let ikuraProduct_ = db.products.findOne({ ProductID: 10 });
let chaiCategory_ = db.categories.findOne({
  CategoryID: chaiProduct.CategoryID,
});
let ikuraCategory_ = db.categories.findOne({
  CategoryID: ikuraProduct.CategoryID,
});

let customer_ = db.customers.findOne({ CustomerID: "ALFKI" });
let employee_ = db.employees.findOne({ EmployeeID: 5 });
let shipper_ = db.shippers.findOne({ ShipperID: 2 });

let chaiValue_ = 18.0 * 5 * (1 - 0);
let ikuraValue_ = 31.0 * 3 * (1 - 0.05);
let orderTotal_ = chaiValue + ikuraValue;

db.CustomerInfo.updateOne(
  { CustomerID: "ALFKI" },
  {
    $push: {
      Orders: {
        OrderID: newOrderID__,
        Employee: {
          EmployeeID: employee.EmployeeID,
          FirstName: employee.FirstName,
          LastName: employee.LastName,
          Title: employee.Title,
        },
        Dates: {
          OrderDate: new Date("2025-05-06"),
          RequiredDate: new Date("2025-05-15"),
        },
        Orderdetails: [
          {
            UnitPrice: 18.0,
            Quantity: 5,
            Discount: 0,
            Value: 90.0,
            product: {
              ProductID: chaiProduct.ProductID,
              ProductName: chaiProduct.ProductName,
              QuantityPerUnit: chaiProduct.QuantityPerUnit,
              CategoryID: chaiProduct.CategoryID,
              CategoryName: chaiCategory.CategoryName,
            },
          },
          {
            UnitPrice: 31.0,
            Quantity: 3,
            Discount: 0.05,
            Value: 88.35,
            product: {
              ProductID: ikuraProduct.ProductID,
              ProductName: ikuraProduct.ProductName,
              QuantityPerUnit: ikuraProduct.QuantityPerUnit,
              CategoryID: ikuraProduct.CategoryID,
              CategoryName: ikuraCategory.CategoryName,
            },
          },
        ],
        Freight: 25.5,
        OrderTotal: 178.35,
        Shipment: {
          Shipper: {
            _id: shipper._id,
            ShipperID: shipper.ShipperID,
            CompanyName: shipper.CompanyName,
            Phone: shipper.Phone,
          },
          ShipName: "Alfreds Futterkiste",
          ShipAddress: "Obere Str. 57",
          ShipCity: "Berlin",
          ShipCountry: "Germany",
        },
      },
    },
  }
);

```

```

    },
  },
},
}
);

```

Działanie:

```
db.orders.find().sort({ OrderID: -1 }).limit(1);
```

```

[
  {
    "_id": {
      "$oid": "681a8eba0f6c6cd3d09170a7"
    },
    "OrderID": 11078,
    "CustomerID": "ALFKI",
    "EmployeeID": 5,
    "OrderDate": {
      "$date": "2025-05-06T00:00:00Z"
    },
    "RequiredDate": {
      "$date": "2025-05-15T00:00:00Z"
    },
    "ShippedDate": null,
    "ShipVia": 2,
    "Freight": 25.5,
    "ShipName": "Alfreds Futterkiste",
    "ShipAddress": "Obere Str. 57",
    "ShipCity": "Berlin",
    "ShipRegion": null,
    "ShipPostalCode": "12209",
    "ShipCountry": "Germany"
  }
]

```

```
db.orderdetails.find().sort({ OrderID: -1 }).limit(2);
```

```

[
  {
    "_id": {
      "$oid": "681a8eba0f6c6cd3d09170a9"
    },
    "OrderID": 11078,
    "ProductID": 10,
    "UnitPrice": 31,
    "Quantity": 3,
    "Discount": 0.05
  },
  {
    "_id": {
      "$oid": "681a8eba0f6c6cd3d09170a8"
    },
    "OrderID": 11078,
    "ProductID": 1,
    "UnitPrice": 18,
    "Quantity": 5,
    "Discount": 0
  }
]

```

```
db.OrdersInfo.find().sort({ OrderID: -1 }).limit(1);
```

```
[
  {
    "_id": {
      "$oid": "681a8f441b059bfff07bf4284"
    },
    "OrderID": 11078,
    "Customer": {
      "CustomerID": "ALFKI",
      "CompanyName": "Alfreds Futterkiste",
      "City": "Berlin",
      "Country": "Germany"
    },
    "Employee": {
      "EmployeeID": 5,
      "FirstName": "Steven",
      "LastName": "Buchanan",
      "Title": "Sales Manager"
    },
    "Dates": {
      "OrderDate": {
        "$date": "2025-05-06T00:00:00Z"
      },
      "RequiredDate": {
        "$date": "2025-05-15T00:00:00Z"
      }
    },
    "Orderdetails": [
      {
        "UnitPrice": 18,
        "Quantity": 5,
        "Discount": 0,
        "Value": 90,
        "product": {
          "ProductID": 1,
          "ProductName": "Chai",
          "QuantityPerUnit": "10 boxes x 20 bags",
          "CategoryID": 1,
          "CategoryName": "Beverages"
        }
      },
      {
        "UnitPrice": 31,
        "Quantity": 3,
        "Discount": 0.05,
        "Value": 88.35,
        "product": {
          "ProductID": 10,
          "ProductName": "Ikura",
          "QuantityPerUnit": "12 - 200 ml jars",
          "CategoryID": 8,
          "CategoryName": "Seafood"
        }
      }
    ],
    "Freight": 25.5,
    "OrderTotal": 178.35,
    "Shipment": {
      "Shipper": {
        "_id": {
          "$oid": "63a05e60bb3b972d6f4e0abb"
        },
        "ShipperID": 2,
        "CompanyName": "United Package",
        "Phone": "(503) 555-3199"
      },
      "ShipName": "Alfreds Futterkiste",
      "ShipAddress": "Obere Str. 57",
      "ShipCity": "Berlin",
      "ShipCountry": "Germany"
    }
  }
]
```

```
db.CustomerInfo.find({ CustomerID: "ALFKI" });
```

```
[
  {
    "_id": {
      "$oid": "681a8945f19010a72ffbcf2"
    },
    "CompanyName": "Alfreds Futterkiste",
    "City": "Berlin",
    "Country": "Germany",
    "Orders": [
      (...)
      {
        "OrderID": 11078,
        "Employee": {
          "EmployeeID": 5,
          "FirstName": "Steven",
          "LastName": "Buchanan",
          "Title": "Sales Manager"
        },
        "Dates": {
          "OrderDate": {
            "$date": "2025-05-06T00:00:00Z"
          },
          "RequiredDate": {
            "$date": "2025-05-15T00:00:00Z"
          }
        },
        "Orderdetails": [
          {
            "UnitPrice": 18,
            "Quantity": 5,
            "Discount": 0,
            "Value": 90,
            "product": {
              "ProductID": 1,
              "ProductName": "Chai",
              "QuantityPerUnit": "10 boxes x 20 bags",
              "CategoryID": 1,
              "CategoryName": "Beverages"
            }
          },
          {
            "UnitPrice": 31,
            "Quantity": 3,
            "Discount": 0.05,
            "Value": 88.35,
            "product": {
              "ProductID": 10,
              "ProductName": "Ikura",
              "QuantityPerUnit": "12 - 200 ml jars",
              "CategoryID": 8,
              "CategoryName": "Seafood"
            }
          }
        ],
        "Freight": 25.5,
        "OrderTotal": 178.35,
        "Shipment": {
          "Shipper": {
            "_id": {
              "$oid": "63a05e60bb3b972d6f4e0abb"
            },
            "ShipperID": 2,
            "CompanyName": "United Package",
            "Phone": "(503) 555-3199"
          },
          "ShipName": "Alfreds Futterkiste",
          "ShipAddress": "Obere Str. 57",
          "ShipCity": "Berlin",
          "ShipCountry": "Germany"
        }
      }
    ]
  }
]
```

```

    ],
    "CustomerID": "ALFKI"
  }
]

```

Porównanie

- **Oryginalne kolekcje:** Wymaga aktualizacji kilku kolekcji, co zwiększa ryzyko niespójności danych, jednak nie wymaga ręcznego obliczenia wartości zamówienia.
- **OrdersInfo:** Jedna operacja, ale wymaga ręcznego obliczenia wartości zamówienia.
- **CustomerInfo:** Podobnie złożona operacja, ale dane są zagnieżdżone, co może utrudniać późniejsze analizy.

f)

```

// 1. Oryginalne kolekcje:
let maxOrderID = db.orders
  .find()
  .sort({ OrderID: -1 })
  .limit(1)
  .toArray()[0].OrderID;

db.orderdetails.updateMany({ OrderID: maxOrderID }, [
  {
    $set: {
      Discount: { $round: [{ $add: ["$Discount", 0.05] }, 2] },
    },
  },
]);

// 2. OrdersInfo:
let maxOrderID_ = db.OrdersInfo.find()
  .sort({ OrderID: -1 })
  .limit(1)
  .toArray()[0].OrderID;

db.OrdersInfo.updateOne({ OrderID: maxOrderID_ }, [
  {
    $set: {
      Orderdetails: {
        $map: {
          input: "$Orderdetails",
          as: "detail",
          in: {
            UnitPrice: "$$detail.UnitPrice",
            Quantity: "$$detail.Quantity",
            Discount: {
              $round: [{ $add: ["$$detail.Discount", 0.05] }, 2],
            },
            Value: {
              $round: [
                {
                  $multiply: [
                    "$$detail.UnitPrice",
                    "$$detail.Quantity",
                    {
                      $subtract: [
                        1,
                        {
                          $add: ["$$detail.Discount", 0.05],
                        },
                      ],
                    },
                  ],
                },
                2,
              ],
            },
            product: "$$detail.product",
          },
        },
      },
    },
  },
]);

```

```

    $set: {
      OrderTotal: { $sum: "$Orderdetails.Value" },
    },
  },
]);

// 3. CustomerInfo:
let maxOrderID__ =
  db.orders.find().sort({ OrderID: -1 }).limit(1).toArray()[0].OrderID + 1;

db.CustomerInfo.updateOne({ "Orders.OrderID": maxOrderID__ }, [
  {
    $set: {
      Orders: {
        $map: {
          input: "$Orders",
          as: "order",
          in: {
            $cond: [
              { $eq: [ "$$order.OrderID", maxOrderID__ ] },
              {
                OrderID: "$$order.OrderID",
                Employee: "$$order.Employee",
                Dates: "$$order.Dates",
                Orderdetails: {
                  $map: {
                    input: "$$order.Orderdetails",
                    as: "detail",
                    in: {
                      UnitPrice: "$$detail.UnitPrice",
                      Quantity: "$$detail.Quantity",
                      Discount: {
                        $round: [
                          {
                            $add: [ "$$detail.Discount", 0.05 ],
                          },
                          2,
                        ],
                      },
                    },
                  },
                  Value: {
                    $round: [
                      {
                        $multiply: [
                          "$$detail.UnitPrice",
                          "$$detail.Quantity",
                        ],
                        $subtract: [
                          1,
                          {
                            $add: [ "$$detail.Discount", 0.05 ],
                          },
                        ],
                      },
                      2,
                    ],
                  },
                  product: "$$detail.product",
                },
              },
            ],
          },
          Freight: "$$order.Freight",
          OrderTotal: {
            $sum: {
              $map: {
                input: "$$order.Orderdetails",
                as: "detail",
                in: {
                  $round: [
                    {
                      $multiply: [
                        "$$detail.UnitPrice",
                        "$$detail.Quantity",
                      ],
                      $subtract: [
                        1,

```

```

        {
            $add: ["$$detail.Discount", 0.05],
        },
    ],
},
],
},
2,
],
},
},
},
},
Shipment: "$$order.Shipment",
},
"$$order",
],
},
},
},
},
});

```

Działanie:

```
db.orderdetails.find().sort({ OrderID: -1 }).limit(2);
```

```

[
  {
    "_id": {
      "$oid": "681a8eba0f6c6cd3d09170a9"
    },
    "OrderID": 11078,
    "ProductID": 10,
    "UnitPrice": 31,
    "Quantity": 3,
    "Discount": 0.1
  },
  {
    "_id": {
      "$oid": "681a8eba0f6c6cd3d09170a8"
    },
    "OrderID": 11078,
    "ProductID": 1,
    "UnitPrice": 18,
    "Quantity": 5,
    "Discount": 0.05
  }
]

```

```
db.OrdersInfo.find().sort({ OrderID: -1 }).limit(1);
```

```

[
  {
    "_id": {
      "$oid": "681a8f441b059bff07bf4284"
    },
    "OrderID": 11078,
    "Customer": {
      "CustomerID": "ALFKI",
      "CompanyName": "Alfreds Futterkiste",
      "City": "Berlin",
      "Country": "Germany"
    },
    "Employee": {
      "EmployeeID": 5,
      "FirstName": "Steven",

```



```

    "LastName": "Buchanan",
    "Title": "Sales Manager"
  },
  "Dates": {
    "OrderDate": {
      "$date": "2025-05-06T00:00:00Z"
    },
    "RequiredDate": {
      "$date": "2025-05-15T00:00:00Z"
    }
  },
  "Orderdetails": [
    {
      "UnitPrice": 18,
      "Quantity": 5,
      "Discount": 0.05,
      "Value": 85.5,
      "product": {
        "ProductID": 1,
        "ProductName": "Chai",
        "QuantityPerUnit": "10 boxes x 20 bags",
        "CategoryID": 1,
        "CategoryName": "Beverages"
      }
    },
    {
      "UnitPrice": 31,
      "Quantity": 3,
      "Discount": 0.1,
      "Value": 83.7,
      "product": {
        "ProductID": 10,
        "ProductName": "Ikura",
        "QuantityPerUnit": "12 - 200 ml jars",
        "CategoryID": 8,
        "CategoryName": "Seafood"
      }
    }
  ],
  "Freight": 25.5,
  "OrderTotal": 169.2,
  "Shipment": {
    "Shipper": {
      "_id": {
        "$oid": "63a05e60bb3b972d6f4e0abb"
      },
      "ShipperID": 2,
      "CompanyName": "United Package",
      "Phone": "(503) 555-3199"
    },
    "ShipName": "Alfreds Futterkiste",
    "ShipAddress": "Obere Str. 57",
    "ShipCity": "Berlin",
    "ShipCountry": "Germany"
  }
}
]

```

```
db.CustomerInfo.find({ CustomerID: "ALFKI" });
```

```

[
  {
    "_id": {
      "$oid": "681a8945f19010a72fffbcf2"
    },
    "CompanyName": "Alfreds Futterkiste",
    "City": "Berlin",
    "Country": "Germany",
    "Orders": [
      (...)
      {
        "OrderID": 11078,

```

```

    "Employee": {
      "EmployeeID": 5,
      "FirstName": "Steven",
      "LastName": "Buchanan",
      "Title": "Sales Manager"
    },
    "Dates": {
      "OrderDate": {
        "$date": "2025-05-06T00:00:00Z"
      },
      "RequiredDate": {
        "$date": "2025-05-15T00:00:00Z"
      }
    },
    "Orderdetails": [
      {
        "UnitPrice": 18,
        "Quantity": 5,
        "Discount": 0.05,
        "Value": 85.5,
        "product": {
          "ProductID": 1,
          "ProductName": "Chai",
          "QuantityPerUnit": "10 boxes x 20 bags",
          "CategoryID": 1,
          "CategoryName": "Beverages"
        }
      },
      {
        "UnitPrice": 31,
        "Quantity": 3,
        "Discount": 0.1,
        "Value": 83.7,
        "product": {
          "ProductID": 10,
          "ProductName": "Ikura",
          "QuantityPerUnit": "12 - 200 ml jars",
          "CategoryID": 8,
          "CategoryName": "Seafood"
        }
      }
    ],
    "Freight": 25.5,
    "OrderTotal": 169.2,
    "Shipment": {
      "Shipper": {
        "_id": {
          "$oid": "63a05e60bb3b972d6f4e0abb"
        },
        "ShipperID": 2,
        "CompanyName": "United Package",
        "Phone": "(503) 555-3199"
      },
      "ShipName": "Alfreds Futterkiste",
      "ShipAddress": "Obere Str. 57",
      "ShipCity": "Berlin",
      "ShipCountry": "Germany"
    }
  },
  "CustomerID": "ALFKI"
}
]

```

Porównanie

- **Oryginalne kolekcje:** Najprostsza aktualizacja.
- **OrdersInfo:** Bardziej złożona aktualizacja kolekcji, wymaga przeliczenia wartości zamówienia.
- **CustomerInfo:** Najbardziej złożona operacja ze względu na zagnieżdżoną strukturę danych, również wymaga ręcznego przeliczenia.

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

Wybrany problem: Wypożyczalnia sprzętu narciarskiego

W ramach zadania porównujemy trzy różne podejścia do modelowania danych dla wypożyczalni sprzętu narciarskiego.

Propozycje struktur bazy danych

Wariant 1: Struktura znormalizowana z referencjami

Kolekcje:

- **equipment** - sprzęt dostępny do wypożyczenia
- **clients** - dane klientów
- **reservations** - rezerwacje z referencjami do klientów i sprzętu

Przykładowe dokumenty:

```
// Kolekcja: equipment
{
  "_id": ObjectId("60d21b4667d0d8992e610c1a"),
  "name": "Rossignol Experience 76",
  "type": "ski",
  "size": "170cm",
  "dailyRate": 25.0,
  "isAvailable": true
}

// Kolekcja: clients
{
  "_id": ObjectId("60d21b4667d0d8992e610c1e"),
  "name": "John Smith",
  "email": "john.smith@example.com",
  "phone": "555-123-4567"
}

// Kolekcja: reservations
{
  "_id": ObjectId("60d21b4667d0d8992e610c20"),
  "clientId": ObjectId("60d21b4667d0d8992e610c1e"),
  "equipmentIds": [
    ObjectId("60d21b4667d0d8992e610c1a"),
    ObjectId("60d21b4667d0d8992e610c1c")
  ],
  "startDate": ISODate("2025-01-15"),
  "endDate": ISODate("2025-01-18"),
  "totalCost": 120.0,
  "status": "completed"
}
```

Wariant 2: Podejście z dokumentami zagnieżdżonymi

Kolekcje:

- **clients** - dane klientów z zagnieżdżonymi rezerwacjami i danymi sprzętu

Przykładowy dokument:

```
// Kolekcja: clients
{
  "_id": ObjectId("60d21b4667d0d8992e610c22"),
  "name": "John Smith",
  "email": "john.smith@example.com",
  "phone": "555-123-4567",
  "reservations": [
    {
      "equipment": [
        {
          "equipmentId": ObjectId("60d21b4667d0d8992e610c1a"),
          "name": "Rossignol Experience 76",
          "type": "ski",

```

```

        "size": "170cm",
        "dailyRate": 25.0
    },
    {
        "equipmentId": ObjectId("60d21b4667d0d8992e610c1c"),
        "name": "Lange XT3",
        "type": "boots",
        "size": "27.5",
        "dailyRate": 15.0
    }
],
"startDate": ISODate("2025-01-15"),
"endDate": ISODate("2025-01-18"),
"totalCost": 120.0,
"status": "completed"
}
]
}

```

Wariant 3: Podejście hybrydowe

Kolekcje:

- **equipment** - przechowuje dane o sprzęcie
- **clients** - przechowuje dane o klientach
- **reservations** - przechowuje dane o rezerwacjach z częściowo zagnieżdżonymi danymi sprzętu i klientów

Przykładowy dokument:

```

// Kolekcja: reservations
{
  "_id": ObjectId("60d21b4667d0d8992e610c24"),
  "clientId": ObjectId("60d21b4667d0d8992e610c1e"),
  "clientName": "John Smith",
  "equipment": [
    {
      "equipmentId": ObjectId("60d21b4667d0d8992e610c1a"),
      "name": "Rossignol Experience 76",
      "type": "ski"
    },
    {
      "equipmentId": ObjectId("60d21b4667d0d8992e610c1c"),
      "name": "Lange XT3",
      "type": "boots"
    }
  ],
  "startDate": ISODate("2025-01-15"),
  "endDate": ISODate("2025-01-18"),
  "totalCost": 120.0,
  "status": "completed"
}

```

Porównanie zapytań

1. Znalezienie wszystkich rezerwacji dla klienta

- Wariant 1:

```
db.reservations.find({
  clientId: ObjectId("60d21b4667d0d8992e610c1e"),
});
```

Wynik:

```
[
  {
    "_id": {
      "$oid": "60d21b4667d0d8992e610c20"
    },
    "clientId": {
      "$oid": "60d21b4667d0d8992e610c1e"
    },
    "equipmentIds": [
      {
        "$oid": "60d21b4667d0d8992e610c1a"
      },
      {
        "$oid": "60d21b4667d0d8992e610c1c"
      }
    ],
    "startDate": {
      "$date": "2025-01-15T00:00:00Z"
    },
    "endDate": {
      "$date": "2025-01-18T00:00:00Z"
    },
    "totalCost": 120,
    "status": "cancelled"
  },
  {
    "_id": {
      "$oid": "681acc86ad09c722b955c56b"
    },
    "clientId": {
      "$oid": "60d21b4667d0d8992e610c1e"
    },
    "equipmentIds": [
      {
        "$oid": "60d21b4667d0d8992e610c1b"
      },
      {
        "$oid": "60d21b4667d0d8992e610c1d"
      }
    ],
    "startDate": {
      "$date": "2025-06-01T00:00:00Z"
    },
    "endDate": {
      "$date": "2025-06-03T00:00:00Z"
    },
    "totalCost": 80,
    "status": "pending"
  }
]
```

Komentarz: Wymaga tylko jednego zapytania, ale zwraca tylko identyfikatory sprzętu, bez szczegółowych informacji.

- **Wariant 2:**

```
db.clients.findOne(
  { _id: ObjectId("60d21b4667d0d8992e610c22") },
  { reservations: 1 }
);
```

Wynik:

```
{
  "_id": {
    "$oid": "60d21b4667d0d8992e610c22"
  },
  "reservations": [
    {
      "equipment": [
        {
          "equipmentId": {
            "$oid": "60d21b4667d0d8992e610c1a"
          },
          "name": "Rossignol Experience 76",
          "type": "ski",
          "size": "170cm",
          "dailyRate": 25
        },
        {
          "equipmentId": {
            "$oid": "60d21b4667d0d8992e610c1c"
          },
          "name": "Lange XT3",
          "type": "boots",
          "size": "27.5",
          "dailyRate": 15
        }
      ],
      "startDate": {
        "$date": "2025-01-15T00:00:00Z"
      },
      "endDate": {
        "$date": "2025-01-18T00:00:00Z"
      },
      "totalCost": 120,
      "status": "cancelled"
    },
    {
      "equipment": [
        {
          "equipmentId": {
            "$oid": "60d21b4667d0d8992e610c1b"
          },
          "name": "Burton Custom",
          "type": "snowboard",
          "size": "158cm",
          "dailyRate": 30
        },
        {
          "equipmentId": {
            "$oid": "60d21b4667d0d8992e610c1d"
          },
          "name": "Smith Vantage",
          "type": "helmet",
          "size": "M",
          "dailyRate": 10
        }
      ],
      "startDate": {
        "$date": "2025-06-01T00:00:00Z"
      },
      "endDate": {
        "$date": "2025-06-03T00:00:00Z"
      },
      "totalCost": 80,
      "status": "pending"
    }
  ]
}
```

```
}  
]  
}
```

Komentarz: Jednym zapytaniem pobiera wszystkie rezerwacje klienta wraz z pełnymi szczegółami sprzętu.

- **Wariant 3:**

```
db.reservations.find({  
  clientId: ObjectId("60d21b4667d0d8992e610c1e"),  
});
```

Wynik:

```
[  
  {  
    "_id": {  
      "$oid": "60d21b4667d0d8992e610c24"  
    },  
    "clientId": {  
      "$oid": "60d21b4667d0d8992e610c1e"  
    },  
    "clientName": "John Smith",  
    "equipment": [  
      {  
        "equipmentId": {  
          "$oid": "60d21b4667d0d8992e610c1a"  
        },  
        "name": "Rossignol Experience 76",  
        "type": "ski"  
      },  
      {  
        "equipmentId": {  
          "$oid": "60d21b4667d0d8992e610c1c"  
        },  
        "name": "Lange XT3",  
        "type": "boots"  
      }  
    ],  
    "startDate": {  
      "$date": "2025-01-15T00:00:00Z"  
    },  
    "endDate": {  
      "$date": "2025-01-18T00:00:00Z"  
    },  
    "totalCost": 120,  
    "status": "cancelled"  
  },  
  {  
    "_id": {  
      "$oid": "681acf10417e5b43ec12f103"  
    },  
    "clientId": {  
      "$oid": "60d21b4667d0d8992e610c1e"  
    },  
    "clientName": "John Smith",  
    "equipment": [  
      {  
        "equipmentId": {  
          "$oid": "60d21b4667d0d8992e610c1b"  
        },  
        "name": "Burton Custom",  
        "type": "snowboard"  
      },  
      {  
        "equipmentId": {  
          "$oid": "60d21b4667d0d8992e610c1d"  
        },  
        "name": "Smith Vantage",  
        "type": "helmet"  
      }  
    ],  
    "startDate": {
```



```
    "$date": "2025-06-01T00:00:00Z"
  },
  "endDate": {
    "$date": "2025-06-03T00:00:00Z"
  },
  "totalCost": 80,
  "status": "pending"
}
]
```

Komentarz: Zapytanie proste jak w wariacie 1, ale zwraca także podstawowe informacje o sprzęcie bez dodatkowych zapytań.

2. Znaleźnienie sprzętu wypożyczonego przez klienta

- **Wariant 1:**

```
db.reservations.aggregate([
  { $match: { clientId: ObjectId("60d21b4667d0d8992e610c1e") } },
  {
    $lookup: {
      from: "equipment",
      localField: "equipmentIds",
      foreignField: "_id",
      as: "equipment",
    },
  },
  { $unwind: "$equipment" },
  {
    $project: {
      _id: 0,
      name: "$equipment.name",
      type: "$equipment.type",
      size: "$equipment.size",
    },
  },
]);
```

Wynik:

```
[
  {
    "name": "Rossignol Experience 76",
    "type": "ski",
    "size": "170cm"
  },
  {
    "name": "Lange XT3",
    "type": "boots",
    "size": "27.5"
  },
  {
    "name": "Burton Custom",
    "type": "snowboard",
    "size": "158cm"
  },
  {
    "name": "Smith Vantage",
    "type": "helmet",
    "size": "M"
  }
]
```

Komentarz: Wymaga złożonej agregacji z operacją \$lookup do połączenia z kolekcją equipment.

- **Wariant 2:**

```
db.clients.aggregate([
  { $match: { _id: ObjectId("60d21b4667d0d8992e610c22") } },
  { $unwind: "$reservations" },
  { $unwind: "$reservations.equipment" },
  {
    $project: {
      _id: 0,
      name: "$reservations.equipment.name",
      type: "$reservations.equipment.type",
      size: "$reservations.equipment.size",
    },
  },
]);
```

Wynik:

```
[
  {
    "name": "Rossignol Experience 76",
    "type": "ski",
    "size": "170cm"
  },
  {
    "name": "Lange XT3",
    "type": "boots",
    "size": "27.5"
  },
  {
    "name": "Burton Custom",
    "type": "snowboard",
    "size": "158cm"
  },
  {
    "name": "Smith Vantage",
    "type": "helmet",
    "size": "M"
  }
]
```

Komentarz: Wymaga rozwinięcia zagnieżdżonych tablic, ale dane są już dostępne bez łączenia kolekcji.

- **Wariant 3:**

```
db.reservations.aggregate([
  { $match: { clientId: ObjectId("60d21b4667d0d8992e610c1e") } },
  { $unwind: "$equipment" },
  {
    $project: {
      _id: 0,
      name: "$equipment.name",
      type: "$equipment.type",
    },
  },
]);
```

Wynik:

```
[
  {
    "name": "Rossignol Experience 76",
    "type": "ski"
  },
  {
    "name": "Lange XT3",
    "type": "boots"
  },
  {

```

```
    "name": "Burton Custom",
    "type": "snowboard"
  },
  {
    "name": "Smith Vantage",
    "type": "helmet"
  }
]
```

Komentarz: Prostsze niż w wariancie 1, wymaga tylko rozwinięcia tablicy bez operacji łączenia.

3. Dodanie nowej rezerwacji

- **Wariant 1:**

```
db.reservations.insertOne({
  clientId: ObjectId("60d21b4667d0d8992e610c1e"),
  equipmentIds: [
    ObjectId("60d21b4667d0d8992e610c1b"),
    ObjectId("60d21b4667d0d8992e610c1d"),
  ],
  startDate: new Date("2025-05-10"),
  endDate: new Date("2025-05-15"),
  totalCost: 150.0,
  status: "pending",
});
```

Komentarz: Proste dodanie dokumentu do kolekcji reservations.

- **Wariant 2:**

```
db.clients.updateOne(
  { _id: ObjectId("60d21b4667d0d8992e610c22") },
  {
    $push: {
      reservations: {
        equipment: [
          {
            equipmentId: ObjectId("60d21b4667d0d8992e610c1b"),
            name: "Burton Custom",
            type: "snowboard",
            size: "158cm",
            dailyRate: 30.0,
          },
          {
            equipmentId: ObjectId("60d21b4667d0d8992e610c1d"),
            name: "Smith Vantage",
            type: "helmet",
            size: "M",
            dailyRate: 10.0,
          },
        ],
        startDate: new Date("2025-06-01"),
        endDate: new Date("2025-06-03"),
        totalCost: 80.0,
        status: "pending",
      },
    },
  },
);
```

Komentarz: Wymaga wcześniejszego pobrania danych o sprzęcie i ich zagnieżdżenia w dokumencie klienta.

- **Wariant 3:**

```
db.reservations.insertOne({
  clientId: ObjectId("60d21b4667d0d8992e610c1e"),
  clientName: "John Smith",
  equipment: [
    {
      equipmentId: ObjectId("60d21b4667d0d8992e610c1b"),
      name: "Atomic Redster",
      type: "ski",
    },
    {
      equipmentId: ObjectId("60d21b4667d0d8992e610c1d"),
      name: "Head Kore",
      type: "helmet",
    },
  ],
  startDate: new Date("2025-05-10"),
  endDate: new Date("2025-05-15"),
  totalCost: 150.0,
  status: "pending",
});
```

Komentarz: Wymaga wcześniejszego pobrania danych o sprzęcie i ich zagnieżdżenia w dokumencie klienta.

4. Aktualizacja statusu rezerwacji

- **Wariant 1:**

```
db.reservations.updateOne(
  { _id: ObjectId("60d21b4667d0d8992e610c20") },
  { $set: { status: "cancelled" } }
);
```

Komentarz: Proste i wydajne - bezpośrednia aktualizacja dokumentu.

- **Wariant 2:**

```
db.clients.updateOne(
  {
    _id: ObjectId("60d21b4667d0d8992e610c22"),
    "reservations.startDate": new Date("2025-01-15"),
  },
  { $set: { "reservations.$.status": "cancelled" } }
);
```

Komentarz: Nieco bardziej złożone - wymaga identyfikacji konkretnej rezerwacji w tablicy po dacie i użycia operatora pozycyjnego \$.

- **Wariant 3:**

```
db.reservations.updateOne(
  { _id: ObjectId("60d21b4667d0d8992e610c24") },
  { $set: { status: "cancelled" } }
);
```

Komentarz: Identycznie jak w wariantcie 1.

5. Znajdzenie rezerwacji w określonym zakresie dat

- **Wariant 1:**

```
db.reservations.find({
  startDate: { $gte: new Date("2025-01-01") },
  endDate: { $lte: new Date("2025-01-31") },
});
```

Wynik:

```
[
  {
    "_id": {
      "$oid": "60d21b4667d0d8992e610c20"
    },
    "clientId": {
      "$oid": "60d21b4667d0d8992e610c1e"
    },
    "equipmentIds": [
      {
        "$oid": "60d21b4667d0d8992e610c1a"
      },
      {
        "$oid": "60d21b4667d0d8992e610c1c"
      }
    ],
    "startDate": {
      "$date": "2025-01-15T00:00:00Z"
    },
    "endDate": {
      "$date": "2025-01-18T00:00:00Z"
    },
    "totalCost": 120,
    "status": "cancelled"
  }
]
```

Komentarz: Proste wyszukiwanie po datach w jednej kolekcji.

- **Wariant 2:**

```
db.clients.aggregate([
  { $unwind: "$reservations" },
  {
    $match: {
      "reservations.startDate": { $gte: new Date("2025-01-01") },
      "reservations.endDate": { $lte: new Date("2025-01-31") },
    },
  },
  {
    $project: {
      name: 1,
      email: 1,
      reservations: 1,
    },
  },
]);
```

Wynik:

```
[
  {
    "_id": {
      "$oid": "60d21b4667d0d8992e610c22"
    },
    "name": "John Smith",
    "email": "john.smith@example.com",
    "reservations": {
      "equipment": [
        {
          "equipmentId": {
            "$oid": "60d21b4667d0d8992e610c1a"
          },
          "name": "Rossignol Experience 76",
          "type": "ski",
          "size": "170cm",
          "dailyRate": 25
        },
        {

```

```

    "equipmentId": {
      "$oid": "60d21b4667d0d8992e610c1c"
    },
    "name": "Lange XT3",
    "type": "boots",
    "size": "27.5",
    "dailyRate": 15
  }
],
"startDate": {
  "$date": "2025-01-15T00:00:00Z"
},
"endDate": {
  "$date": "2025-01-18T00:00:00Z"
},
"totalCost": 120,
"status": "cancelled"
}
}
]

```

Komentarz: Wymaga użycia agregacji i rozwinięcia tablicy rezerwacji.

- **Wariant 3:**

```

db.reservations.find({
  startDate: { $gte: new Date("2025-01-01") },
  endDate: { $lte: new Date("2025-01-31") },
});

```

Wynik:

```

[
  {
    "_id": {
      "$oid": "60d21b4667d0d8992e610c24"
    },
    "clientId": {
      "$oid": "60d21b4667d0d8992e610c1e"
    },
    "clientName": "John Smith",
    "equipment": [
      {
        "equipmentId": {
          "$oid": "60d21b4667d0d8992e610c1a"
        },
        "name": "Rossignol Experience 76",
        "type": "ski"
      },
      {
        "equipmentId": {
          "$oid": "60d21b4667d0d8992e610c1c"
        },
        "name": "Lange XT3",
        "type": "boots"
      }
    ],
    "startDate": {
      "$date": "2025-01-15T00:00:00Z"
    },
    "endDate": {
      "$date": "2025-01-18T00:00:00Z"
    },
    "totalCost": 120,
    "status": "cancelled"
  }
]

```

Komentarz: Identyczne jak w wariantcie 1.

6. Ilości wypożyczeń różnych rodzajów sprzętu

- **Wariant 1:**

```
db.reservations.aggregate([
  { $unwind: "$equipmentIds" },
  {
    $lookup: {
      from: "equipment",
      localField: "equipmentIds",
      foreignField: "_id",
      as: "equipment",
    },
  },
  { $unwind: "$equipment" },
  {
    $group: {
      _id: "$equipment.type",
      count: { $sum: 1 },
    },
  },
]);
```

Wynik:

```
[
  {
    "_id": "snowboard",
    "count": 3
  },
  {
    "_id": "ski",
    "count": 1
  },
  {
    "_id": "boots",
    "count": 1
  },
  {
    "_id": "helmet",
    "count": 3
  }
]
```

Komentarz: Wymaga agregacji z operacją \$lookup i złączeniem kolekcji.

- **Wariant 2:**

```
db.clients.aggregate([
  { $unwind: "$reservations" },
  { $unwind: "$reservations.equipment" },
  {
    $group: {
      _id: "$reservations.equipment.type",
      count: { $sum: 1 },
    },
  },
]);
```

Wynik:

```
[
  {
    "_id": "snowboard",
    "count": 2
  },
  {
    "_id": "ski",
```

```

    "count": 1
  },
  {
    "_id": "boots",
    "count": 1
  },
  {
    "_id": "helmet",
    "count": 2
  }
]

```

Komentarz: Wymaga rozwinięcia zagnieżdżonych tablic, ale bez łączenia kolekcji.

- **Wariant 3:**

```

db.reservations.aggregate([
  { $unwind: "$equipment" },
  {
    $group: {
      _id: "$equipment.type",
      count: { $sum: 1 },
    },
  },
]);

```

Wynik:

```

[
  {
    "_id": "ski",
    "count": 2
  },
  {
    "_id": "snowboard",
    "count": 2
  },
  {
    "_id": "boots",
    "count": 1
  },
  {
    "_id": "helmet",
    "count": 3
  }
]

```

Komentarz: Najprostszy wariant - rozwinięcie jednej tablicy.

7. Podsumowanie przychodów według miesiąca

- **Wariant 1:**

```

db.reservations.aggregate([
  {
    $project: {
      month: { $month: "$startDate" },
      year: { $year: "$startDate" },
      totalCost: 1,
    },
  },
  {
    $group: {
      _id: { month: "$month", year: "$year" },
      totalRevenue: { $sum: "$totalCost" },
    },
  },
  {
    $sort: { "_id.year": 1, "_id.month": 1 },
  }
]);

```



```
    },
  ]);
```

Wynik:

```
[
  {
    "_id": {
      "month": 1,
      "year": 2025
    },
    "totalRevenue": 120
  },
  {
    "_id": {
      "month": 5,
      "year": 2025
    },
    "totalRevenue": 80
  },
  {
    "_id": {
      "month": 6,
      "year": 2025
    },
    "totalRevenue": 160
  }
]
```

Komentarz: Standardowa agregacja na kolekcji reservations.

- **Wariant 2:**

```
db.clients.aggregate([
  { $unwind: "$reservations" },
  {
    $project: {
      month: { $month: "$reservations.startDate" },
      year: { $year: "$reservations.startDate" },
      totalCost: "$reservations.totalCost",
    },
  },
  {
    $group: {
      _id: { month: "$month", year: "$year" },
      totalRevenue: { $sum: "$totalCost" },
    },
  },
  {
    $sort: { "_id.year": 1, "_id.month": 1 },
  },
]);
```

Wynik:

```
[
  {
    "_id": {
      "month": 1,
      "year": 2025
    },
    "totalRevenue": 120
  },
  {
    "_id": {
      "month": 5,
      "year": 2025
    },
    "totalRevenue": 80
  },
  {
    "_id": {
      "month": 6,
      "year": 2025
    },
    "totalRevenue": 160
  }
]
```

```
{
  "_id": {
    "month": 6,
    "year": 2025
  },
  "totalRevenue": 80
}
```

Komentarz: Podobna agregacja, ale z koniecznością rozwinięcia zagnieżdżonej tablicy.

- **Wariant 3:**

```
db.reservations.aggregate([
  {
    $project: {
      month: { $month: "$startDate" },
      year: { $year: "$startDate" },
      totalCost: 1,
    },
  },
  {
    $group: {
      _id: { month: "$month", year: "$year" },
      totalRevenue: { $sum: "$totalCost" },
    },
  },
  {
    $sort: { "_id.year": 1, "_id.month": 1 },
  },
]);
```

Wynik:

```
[
  {
    "_id": {
      "month": 5,
      "year": 2023
    },
    "totalRevenue": 150
  },
  {
    "_id": {
      "month": 1,
      "year": 2025
    },
    "totalRevenue": 120
  },
  {
    "_id": {
      "month": 5,
      "year": 2025
    },
    "totalRevenue": 80
  },
  {
    "_id": {
      "month": 6,
      "year": 2025
    },
    "totalRevenue": 80
  }
]
```

Komentarz: Identyczne jak w wariantcie 1.

8. Aktualizacja dostępności sprzętu

- **Wariant 1 oraz 3:**

```
db.equipment.updateOne(  
  { _id: ObjectId("60d21b4667d0d8992e610c1a") },  
  { $set: { isAvailable: false } }  
);
```

Komentarz: Prosta aktualizacja dokumentu.

- **Wariant 2:** W tym wariantcie informacja o dostępności sprzętu nie jest przechowywana bezpośrednio w dokumencie.

Wnioski

Na podstawie analizy trzech wariantów modeli danych i porównania zapytań można wyciągnąć następujące wnioski:

Wariant 1: Struktura znormalizowana

- **Zalety:**
 - Brak duplikacji danych
 - Łatwość niezależnych aktualizacji informacji o klientach i sprzęcie
 - Najlepsze dopasowanie do częstych zmian w katalogu sprzętu
- **Wady:**
 - Wolniejsze odczyty wymagające złączenia kolekcji
 - Bardziej złożone zapytania dla uzyskania pełnych danych
 - Wyższe koszty obliczeniowe przy pobieraniu złożonych relacji

Wariant 2: Dokumenty zagnieżdżone

- **Zalety:**
 - Najszybsze odczyty kompletnych danych o rezerwacjach klienta
 - Proste zapytania dla historii klienta
 - Dobra wydajność dla operacji zorientowanych na klienta
- **Wady:**
 - Znaczna duplikacja danych
 - Trudność w utrzymaniu spójności danych
 - Skomplikowane aktualizacje danych sprzętu (wymagane w wielu dokumentach)
 - Potencjalnie duże dokumenty dla aktywnych klientów
 - Brak bezpośredniej informacji o dostępności sprzętu

Wariant 3: Podejście hybrydowe

- **Zalety:**
 - Dobry balans między wydajnością a umiarkowaną duplikacją
 - Szybsze odczyty niż w wariancie 1 bez nadmiernej duplikacji jak w wariancie 2
 - Łatwiejsze utrzymanie spójności danych niż w wariancie 2
 - Najlepsze ogólne wyniki dla różnorodnych zapytań
- **Wady:**
 - Nadal pewien poziom duplikacji danych
 - Konieczność aktualizacji niektórych danych w wielu miejscach
 - Większa złożoność projektowa niż w modelach czysto znormalizowanych lub zagnieżdżonych

Wniosek końcowy:

Wariant 3 (podejście hybrydowe) wydaje się najlepszym kompromisem. Wybór optymalnej struktury powinien jednak uwzględniać konkretne wymagania biznesowe, proporcje operacji odczytu do zapisu oraz przewidywany wzrost danych w systemie.

Punktacja:

zadanie	pkt
1	1
2	1
razem	2