Wariant 3 - Hybrydowy

Schemat bazy danych

```
db.createCollection("companies", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["name", "description", "contactInfo", "foundedYear",
"licenseNumber"],
      properties: {
        name: { bsonType: "string" },
        description: { bsonType: "string" },
        contactInfo: {
          bsonType: "object",
          required: ["email", "phone", "address"],
          properties: {
            email: { bsonType: "string" },
            phone: { bsonType: "string" },
            address: {
              bsonType: "object",
              required: ["street", "city", "postalCode"],
              properties: {
                street: { bsonType: "string" },
                city: { bsonType: "string" },
                postalCode: { bsonType: "string" }
              }
            }
          }
        },
        foundedYear: { bsonType: "int" },
        licenseNumber: { bsonType: "string" }
      }
    }
  }
});

db.createCollection("tours", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["companyId", "companyName", "title", "startDate", "endDate",
"price", "currency", "capacity", "reviews"],
      properties: {
        companyId: { bsonType: "objectId" },
        companyName: { bsonType: "string" },
        title: { bsonType: "string" },
        description: { bsonType: "string" },
        startDate: { bsonType: "date" },
        endDate: { bsonType: "date" },
```

```
        price: { bsonType: "double" },
        capacity: { bsonType: "int" },
        reservationsCount: { bsonType: "int" },
        destination: { bsonType: "string" },
        itinerary: {
          bsonType: "array",
          items: {
            bsonType: "object",
            required: ["day", "description"],
            properties: {
              day: { bsonType: "int" },
              description: { bsonType: "string" }
            }
          }
        },
        tags: {
          bsonType: "array",
          items: { bsonType: "string" }
        },
        reviews: {
          bsonType: "array",
          items: {
            bsonType: "object",
            required: ["userId", "userName", "rating", "createdAt"],
            properties: {
              userId: { bsonType: "objectId" },
              userName: { bsonType: "string" },
              rating: { bsonType: "double" },
              comment: { bsonType: "string" },
              createdAt: { bsonType: "date" }
            }
          }
        },
        averageRating: { bsonType: "double" }
      }
    }
  }
});

db.createCollection("users", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["firstName", "lastName", "email", "phone", "dateOfBirth",
"registrationDate"],
      properties: {
        firstName: { bsonType: "string" },
        lastName: { bsonType: "string" },
        email: { bsonType: "string" },
        phone: { bsonType: "string" },
        dateOfBirth: { bsonType: "date" },
        registrationDate: { bsonType: "date" },
        reservationCount: { bsonType: "int" },
        reviewCount: { bsonType: "int" }
```

```
      }
    }
  }
});

db.createCollection("reservations", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["userId", "userName", "tourId", "tourTitle", "companyId",
"companyName", "bookingDate", "tourDates", "status", "participantsCount",
"totalPrice", "paymentStatus"],
      properties: {
        userId: { bsonType: "objectId" },
        userName: { bsonType: "string" },
        tourId: { bsonType: "objectId" },
        tourTitle: { bsonType: "string" },
        companyId: { bsonType: "objectId" },
        companyName: { bsonType: "string" },
        bookingDate: { bsonType: "date" },
        tourDates: {
          bsonType: "object",
          required: ["startDate", "endDate"],
          properties: {
            startDate: { bsonType: "date" },
            endDate: { bsonType: "date" }
          }
        },
        status: { enum: ["confirmed", "pending", "cancelled"] },
        participantsCount: { bsonType: "int" },
        totalPrice: { bsonType: "double" },
        paymentStatus: { enum: ["paid", "unpaid", "refunded"] },
        specialRequests: { bsonType: "string" }
      }
    }
  }
});
```

Prezentacja wad i zalet:

ZALETA 1: Zbalansowana wydajność – efektywny odczyt bez dużej redundancji

W poniższym zapytaniu nie musimy używać kosztownego $lookup tylko wystarczy zapytanie na jednej kolekcji.

```
db.reservations.find({}, {
  userName: 1,
  tourTitle: 1,
  companyName: 1,
  bookingDate: 1,
```

```
  totalPrice: 1
});
```

ZALETA 2: Ograniczona duplikacja – denormalizacja tylko istotnych pól

Duplikujemy tylko nazwy, co zmniejsza wielkość dokumentu i zwiększa wydajność.

```
db.tours.find({}, {
  companyName: 1,
  title: 1
});
```

ZALETA 3: Łatwiejsza aktualizacja – główne dane w jednym miejscu

```
db.companies.updateOne(
  { name: "Podróże Marzeń" },
  { $set: { licenseNumber: "TUR/456/2025" } }
);
```

WADA 1: Umiarkowana złożoność – trudniejszy w utrzymaniu model

Jeśli potrzebujemy bardziej szczegółowych danych (których nie zdenormalizowaliśmy), to i tak trzeba użyć $lookup.

```
db.reservations.aggregate([
  {
    $lookup: {
      from: "companies",
      localField: "companyId",
      foreignField: "_id",
      as: "companyData"
    }
  },
  { $unwind: "$companyData" },
  {
    $project: {
      userName: 1,
      tourTitle: 1,
      companyName: "$companyData.name",
      companyEmail: "$companyData.contactInfo.email"
    }
  }
]);
```

WADA 2: Częściowa duplikacja – wymaga aktualizacji zduplikowanych danych

Aktualizacja danych powielonych wiąże się z większą liczbą poleceń aby ją zmienić.

```
db.users.updateOne(
  { _id: user1Id },
  { $set: { lastName: "Nowy" } }
);

// Musimy zaktualizować także userName w tours.reviews
db.tours.updateMany(
  { "reviews.userId": user1Id },
  { $set: { "reviews.$[elem].userName": "Jan Nowy" } },
  { arrayFilters: [ { "elem.userId": user1Id } ] }
);

// I w reservations
db.reservations.updateMany(
  { userId: user1Id },
  { $set: { userName: "Jan Nowy" } }
);
```