* **Documentation on How the structure would be in a NO SQL database and contrasting that with the RELATIONAL.**

In a NoSQL database like MongoDB, data organization differs from traditional SQL databases, as it does not rely on tables and relationships. MongoDB, being a document-oriented database, utilizes BSON documents, which are a binary representation of JSON documents.

For our Hospital database, instead of maintaining separate tables for Patients, Diseases, Doctors, and related entities, we can adopt a consolidated approach.

A single "Treatments" collection may be employed, wherein each treatment document encapsulates embedded documents representing associated patient, disease, medication, doctor, nurse, vaccine, and appointment details. This denormalized approach offers advantages such as expedited data retrieval by minimizing the necessity for intricate, join-intensive queries.

For the Hospital database, "Treatments"  has a collection where each document represents a treatment event and includes embedded documents for associated entities. Here's a conceptual example:

{

  "\_id": ObjectId("5fe4eae0472a3b1d5c3e2536"),

  "patient": {

    "first\_name": "John",

    "last\_name": "Doe",

    "dob": ISODate("1990-05-15"),

    "gender": "M",

    "contact": "123-456-7890",

    "address": "123 Main St"

  },

  "disease": {

    "disease\_name": "Flu",

    "disease\_type": "Viral"

  },

  "medication": {

    "medication\_name": "Ibuprofen"

  },

  "doctor": {

    "first\_name": "Dr. Smith",

    "last\_name": "Johnson",

    "specialty": "Cardiology",

    "contact": "987-654-3210"

  },

  "nurse": {

    "first\_name": "Mary",

    "last\_name": "Johnson",

    "contact": "555-123-4567"

  },

  "vaccine": {

    "vaccine\_name": "COVID-19 Vaccine",

    "manufacturer": "Pfizer"

  },

  "appointment": {

    "appointment\_time": ISODate("2023-01-10T10:00:00"),

    "appointment\_date": ISODate("2023-01-10")

  }

}

In the above example:

* Each treatment is a document.
* Patient, disease, medication, doctor, nurse, vaccine, and appointment information is embedded within the treatment document.

This de-normalized approach can indeed lead to faster data retrieval because it minimizes the need for joins, which is especially beneficial for read-heavy operations. However, it's important to note that it comes with trade-offs, and the structure of the data should align with the specific use cases and query patterns of your application.

Queries in MongoDB are generally focused on retrieving document-based content.