Brewer's CAP Theorem

It is impossible for a distributed computer system to simultaneously provide all three of the following guarantees:

Consistency The same definition we've been using: every read returns the most recent data, or an error.

Availability Every request receives a response (no error) – but without a guarantee that it contains the most recent data

Partition-tolerance The system continues to operate despite an arbitrary number of messages being dropped (or delayed) by the network between nodes

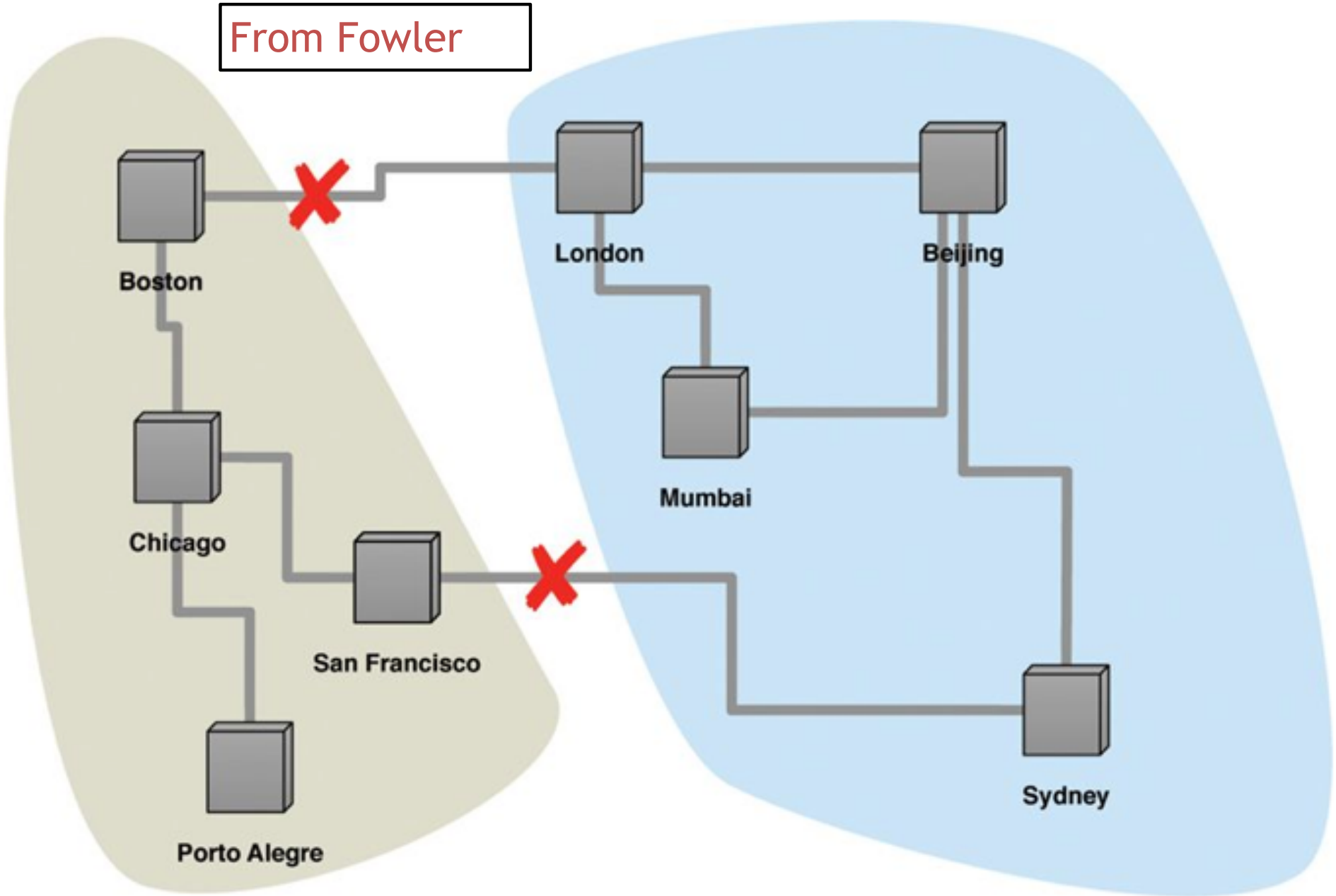


Figure 5.3. With two breaks in the communication lines, the network partitions into two groups.

```
// customer info collection
{
    "id":42,
    "fName":"ebenezer",
    "mi":"j",
    "lName":"coot",
    "addresses": [
     { "addrType":"billingAddress",
       “mailingAddr":"PO Box 99",
       "city":"Santa Fe",
       "St":"NM"
     },
     { "addrType": "shippingAddress",
       "streetAddr":"42 Wrinkled Way",
       "city":"Taos",
       "St":"NM"
     }]
}
```

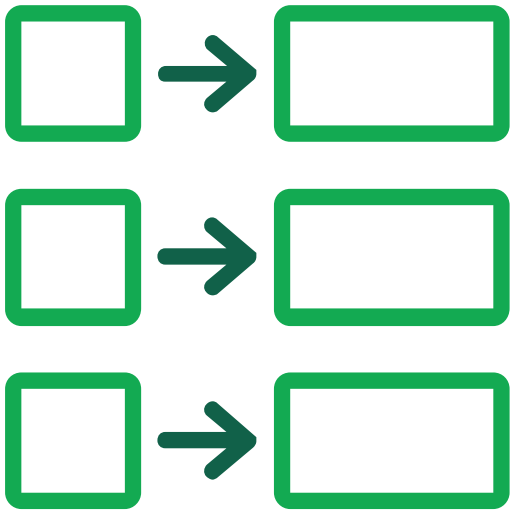
```
// carts
{
  "id":74829312,
  "customer":42,
  "itemList":[
        {
           "UPC":293012429,
           "price":79.95,
           "name":"apple 85w power adapter",
           "shippingSpeed":"2nd day"
        },
        {
           "UPC":829381427,
           "price":59.95,
           "name":"apple touchpad mouse"
        }
    ],
 "paymentInfo":[
        {
           "ccard":"1234-5678-9876-5432",
           "exp":"0115",
           "ccxact":"111111"
        }
    ]
}
```

Structure

- A unique key is paired with a collection of values, where the values can be anything from a string to a large binary object

Strength

- Simple data model



Key/Value Database

Key/Value: Example

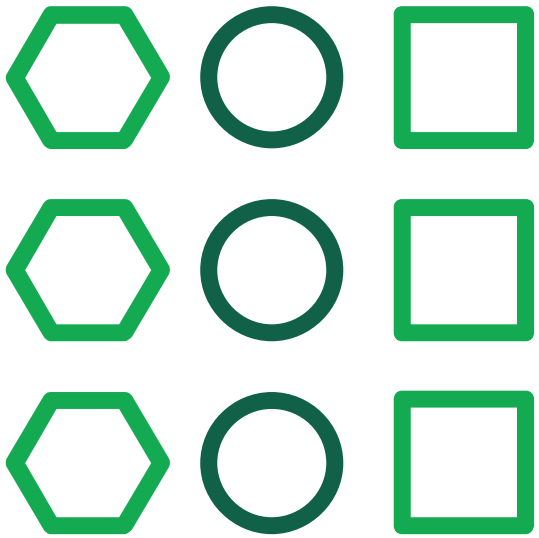
Key	Value
Name	Sherlock Holmes
Age	40
Address	221B Baker Street
Hobbies	[violin,crime, $C_{17}H_{21}NO_4$]

Structure

- Data is stored using key rows that can be associated with one or more dynamic columns

Strengths

- Highly performant queries



Column Oriented or Wide Column

Column Oriented Database Example

Name	ID
Sherlock	001
John	002
Irene	003

Age	ID
40	001
45	002
43	003

Height	ID
6’2	001
5’9	002
5’7	003

Structure

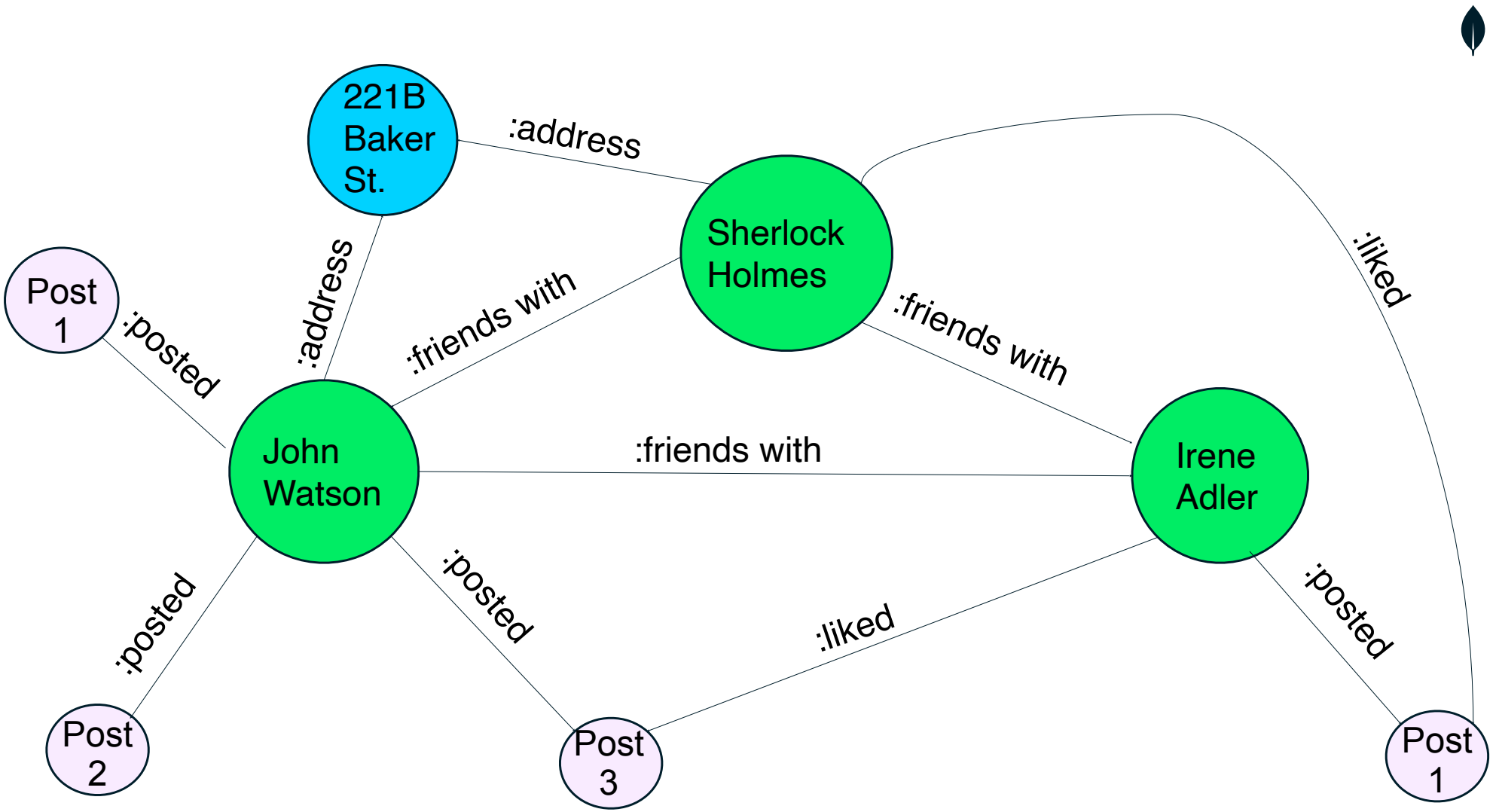
- Captures connected data
- Each element is stored as a node
- Connections between nodes are called links or relationships

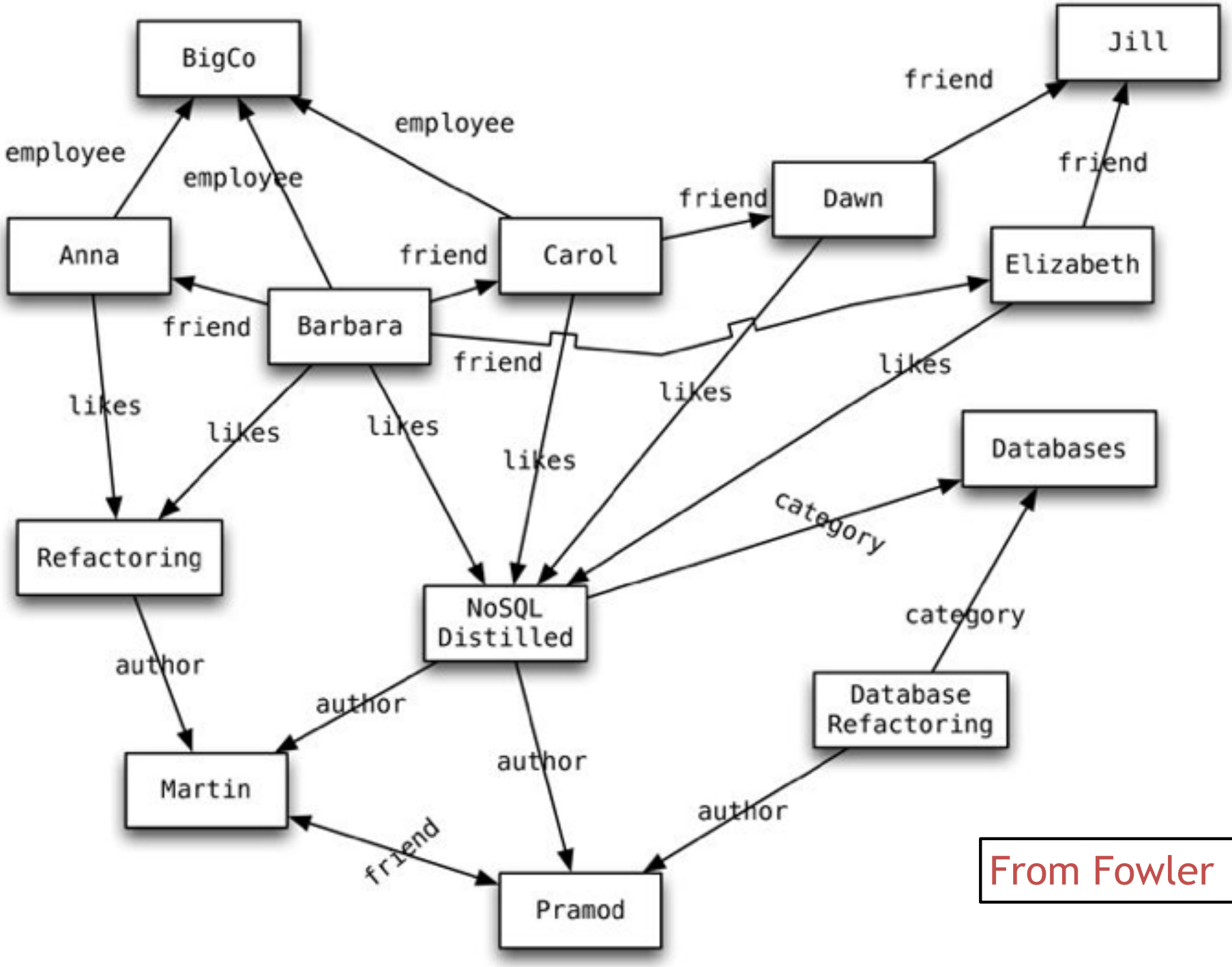
Strength

- Traverses the connections between data rapidly

Graph Database

Graph Database: Example





From Fowler

Figure 3.1. An example graph structure

From Robinson

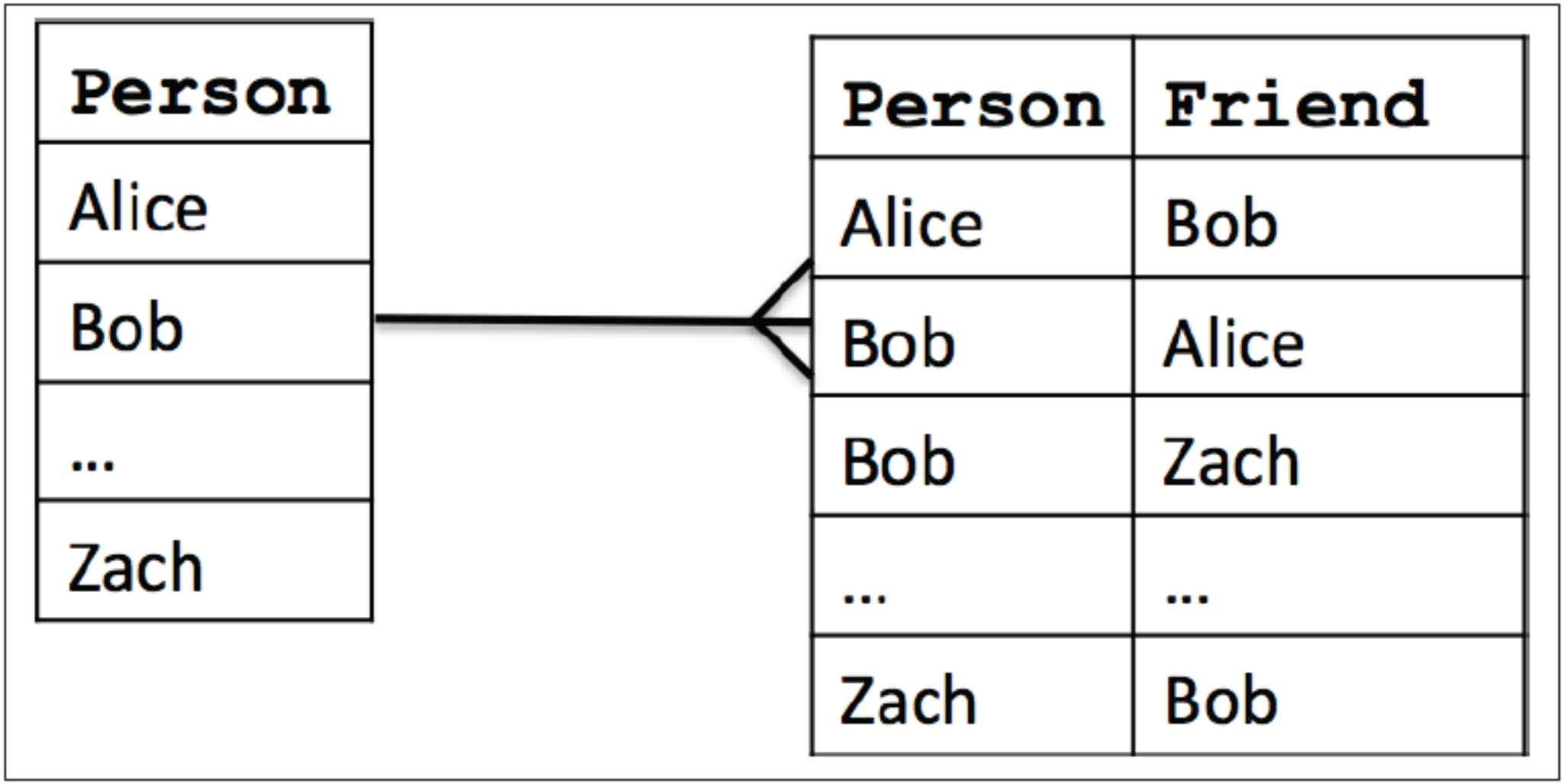


Figure 3-4. Modeling friends and friends-of-friends in a relational database

```
SELECT PersonFriend.friend
FROM Person
JOIN PersonFriend ON Person.name = PersonFriend.name
WHERE Person.name = 'Bob';
```

```
SELECT Person.name
FROM Person
JOIN PersonFriend ON Person.name = PersonFriend.name
WHERE PersonFriend.friend = 'Bob';
```

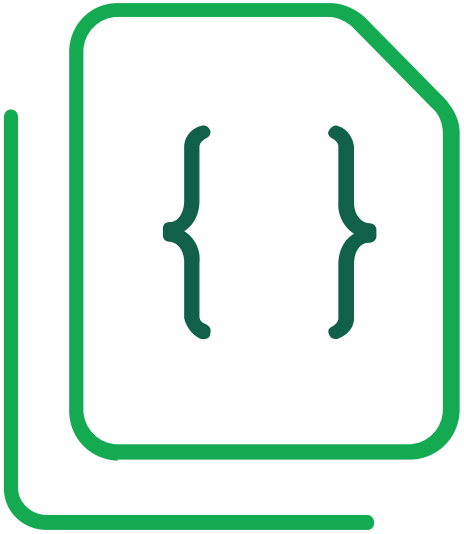
```
SELECT pf1.name AS PERSON,
       pf3.name AS FRIEND_OF_FRIEND
FROM PersonFriend pf1
    JOIN Person ON pf1.name = Person.name
    JOIN PersonFriend pf2 ON pf1.friend = pf2.friend
    JOIN PersonFriend pf3 ON pf2.friend = pf3.friend
WHERE pf1.name = ‘Alice’
  AND pf3.name <> ‘Alice’;
```

Structure

- Polymorphic data models
- Each document contains markup that identifies fields and values

Strengths

- Obvious relationships using embedded arrays



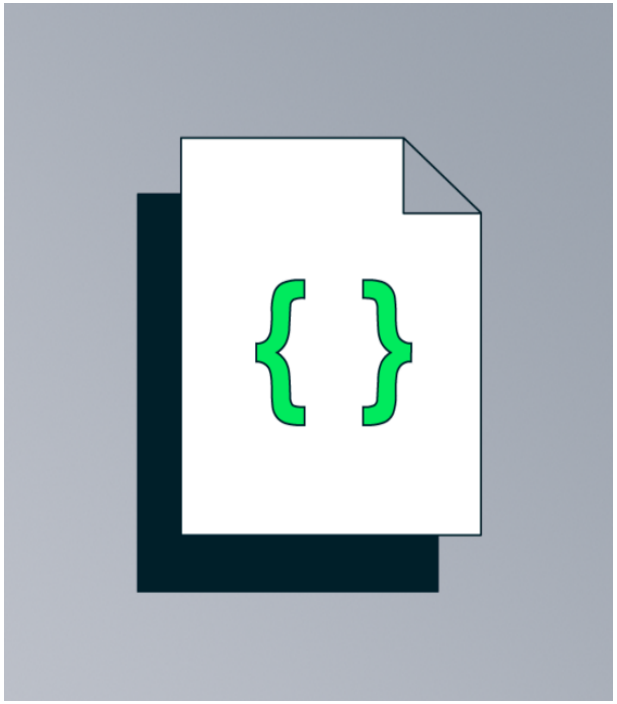
Document Database

Document Model Example

```
{
  "_id":
ObjectId(“5ef2d4b45b7”),
  "user_id":
      “Sherlock Holmes",
  "age": 40,
  "address":
     {
      “Country: “England”
      “City”: “London”,
      “Street”: “221B
    Baker St.”
    },
  “Hobbies”:[ violin,
         crime,C17H21NO4]
}
```

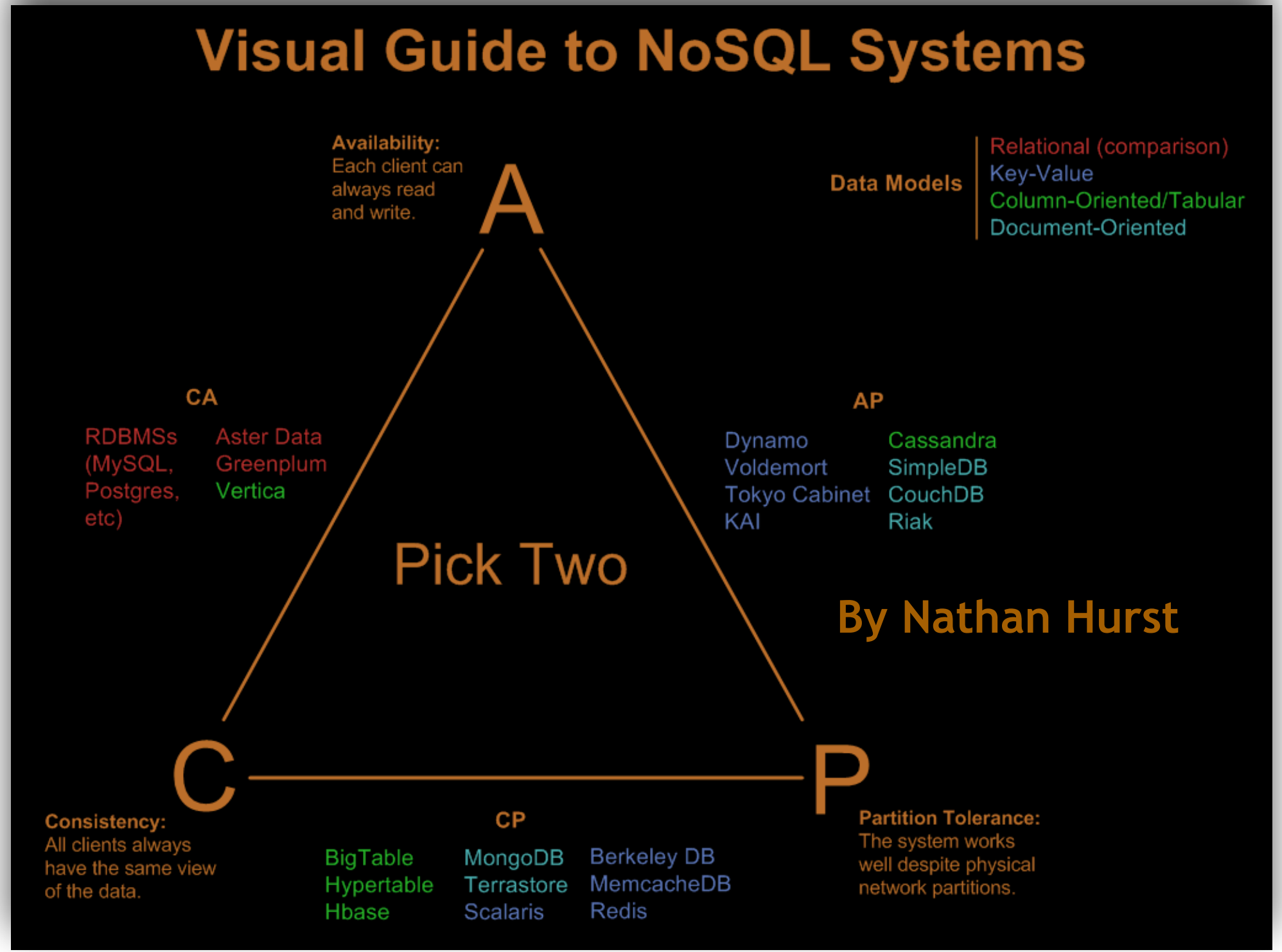
```
{
  "_id":
ObjectId("6ef8d4b32c9f"),
  "user_id":
           "John Watson",
  "age": 45,
  "address":
     {
      “Country: “England”
      “City”: “London”,
      “Street”:
         “221B Baker St.”
    },
  “Medical license”:
                  “Active”
}
```

The Document Model



For general purpose use, the document model prevails as the preferred model by developers and database administrators.

Visual Guide to NoSQL Systems



What's the best NoSQL DB for …

- Lottery
- Snapchat
- High-speed data recorder
- Large-scale organized crime investigations
- Stock market transactions
- Statistical analysis
- Discord
- Economic modeling
- Mostly unstructured, interrelated data