

NoSQL Data Modeling



somkiat.cc



Workshop

NoSQL คืออะไร ?



คุณลักษณะของ NoSQL ?



คุณลักษณะของ NoSQL

Non-relational



คุณลักษณะของ NoSQL

Non-relational
Open-source



คุณลักษณะของ NoSQL

Non-relational
Open-source
Cluster-friendly



คุณลักษณะของ NoSQL

Non-relational
Open-source
Cluster-friendly
Schema-less



คุณลักษณะของ NoSQL

Non-relational
Open-source
Cluster-friendly
Schema-less
For new web app



แบ่งกลุ่มตาม Data Model ?





Cassandra

APACHE
HBASE



Couchbase



Workshop





APACHE
HBASE

Column Family



Couchbase



mongoDB

Document



Graph



Project Voldemort
A distributed database.

Key-Value

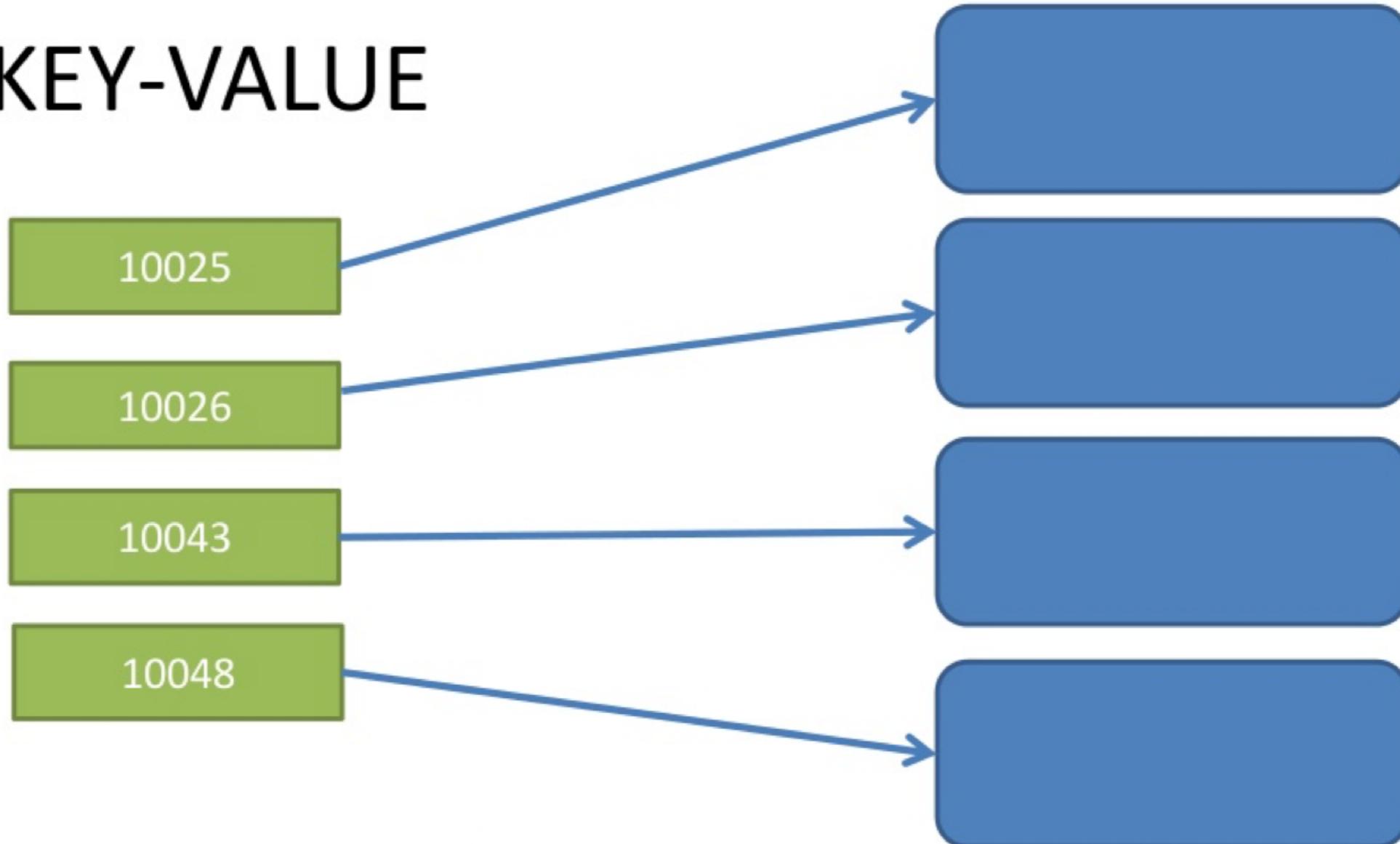


redis



Key-Value

KEY-VALUE



Document

```
<Key=CustomerID>
```

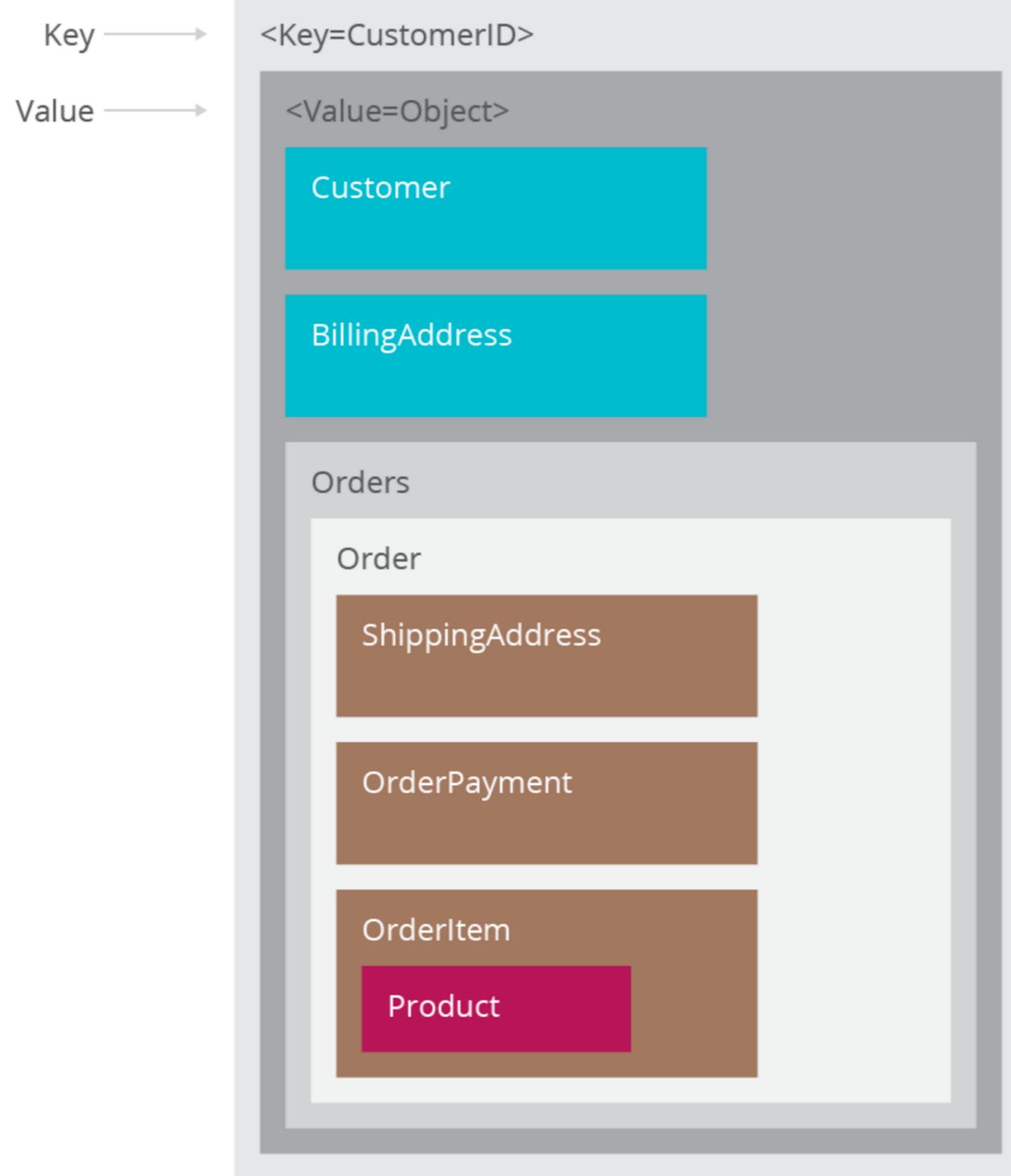
```
{  
  "customerId": "fc986e48ca6" ← Key  
  "customer":  
  {  
    "firstname": "Pramod",  
    "lastname": "Sadalage",  
    "company": "ThoughtWorks",  
    "likes": [ "Biking", "Photography" ]  
  }  
  "billingaddress":  
  { "state": "AK",  
    "city": "DILLINGHAM",  
    "type": "R"  
  }  
}
```

Key

<https://www.thoughtworks.com/insights/blog/nosql-databases-overview>



Workshop



<https://www.thoughtworks.com/insights/blog/nosql-databases-overview>



Column-family

Column Family

Row

Row KeyX

Column1

name1:value1

Column2

name2:value2

ColumnN

nameN:valueN

Row

Row KeyY

Column1

name1:value1

Column9

name9:value9

ColumnN

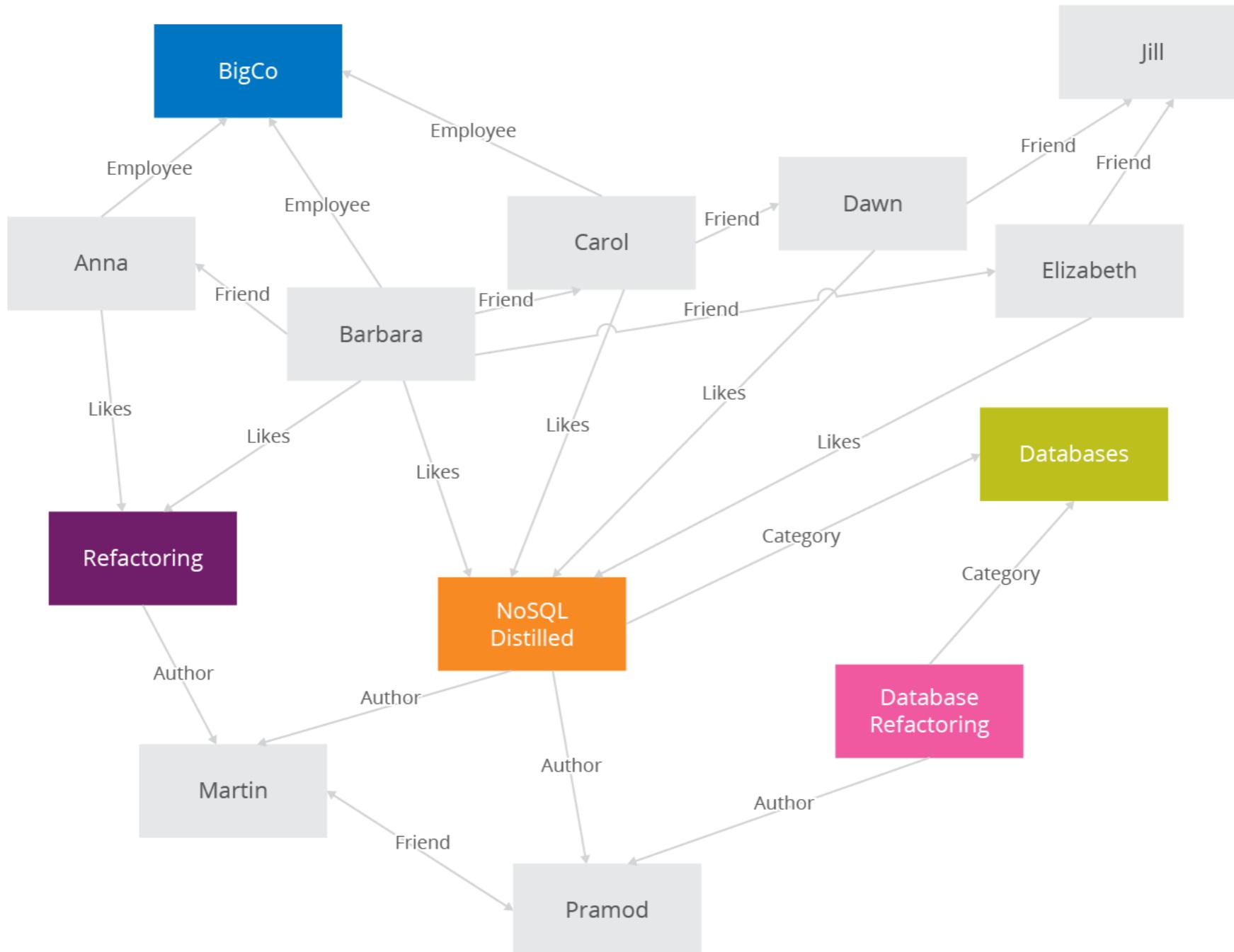
nameN:valueN

<https://www.thoughtworks.com/insights/blog/nosql-databases-overview>



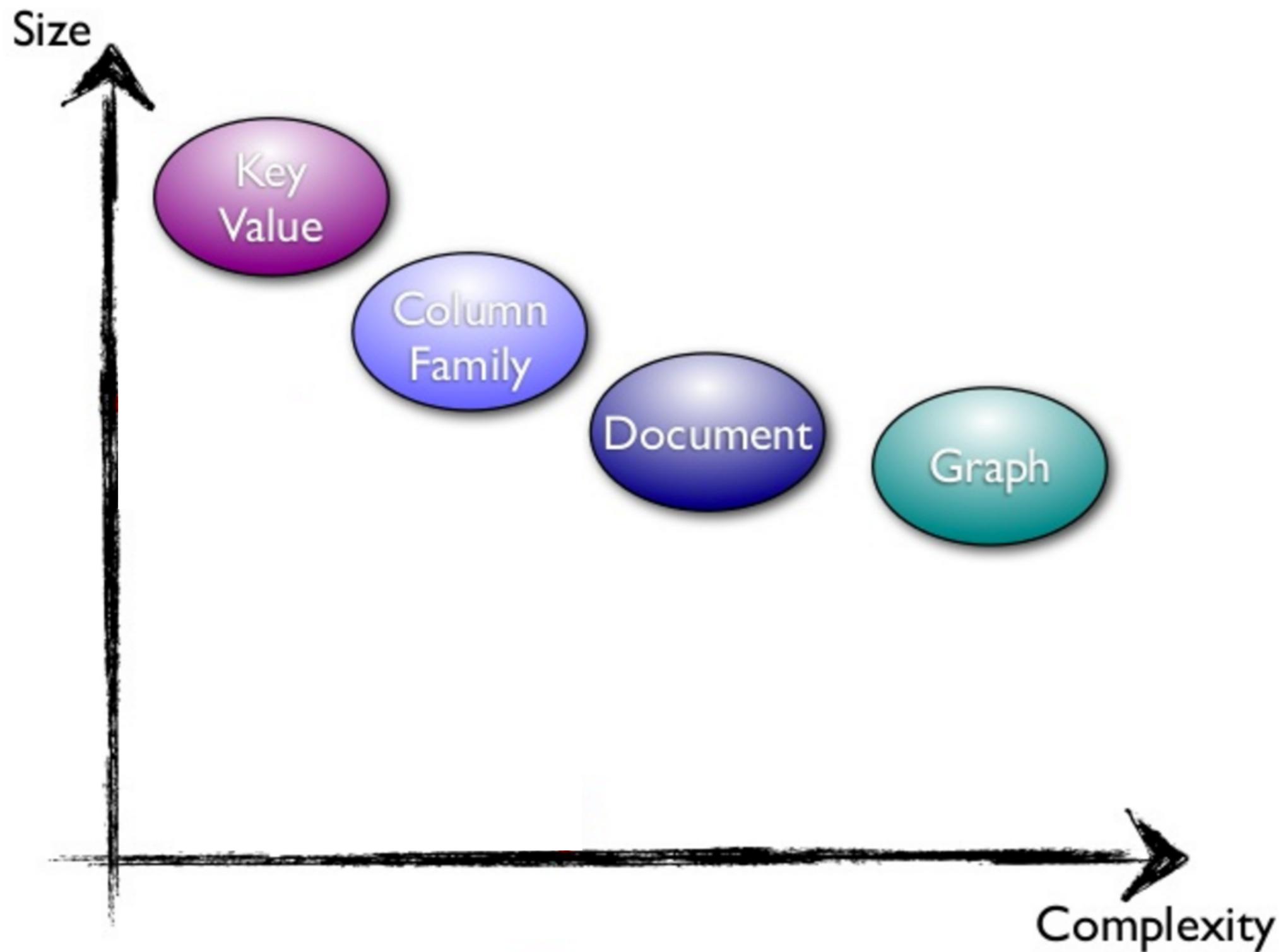
Workshop

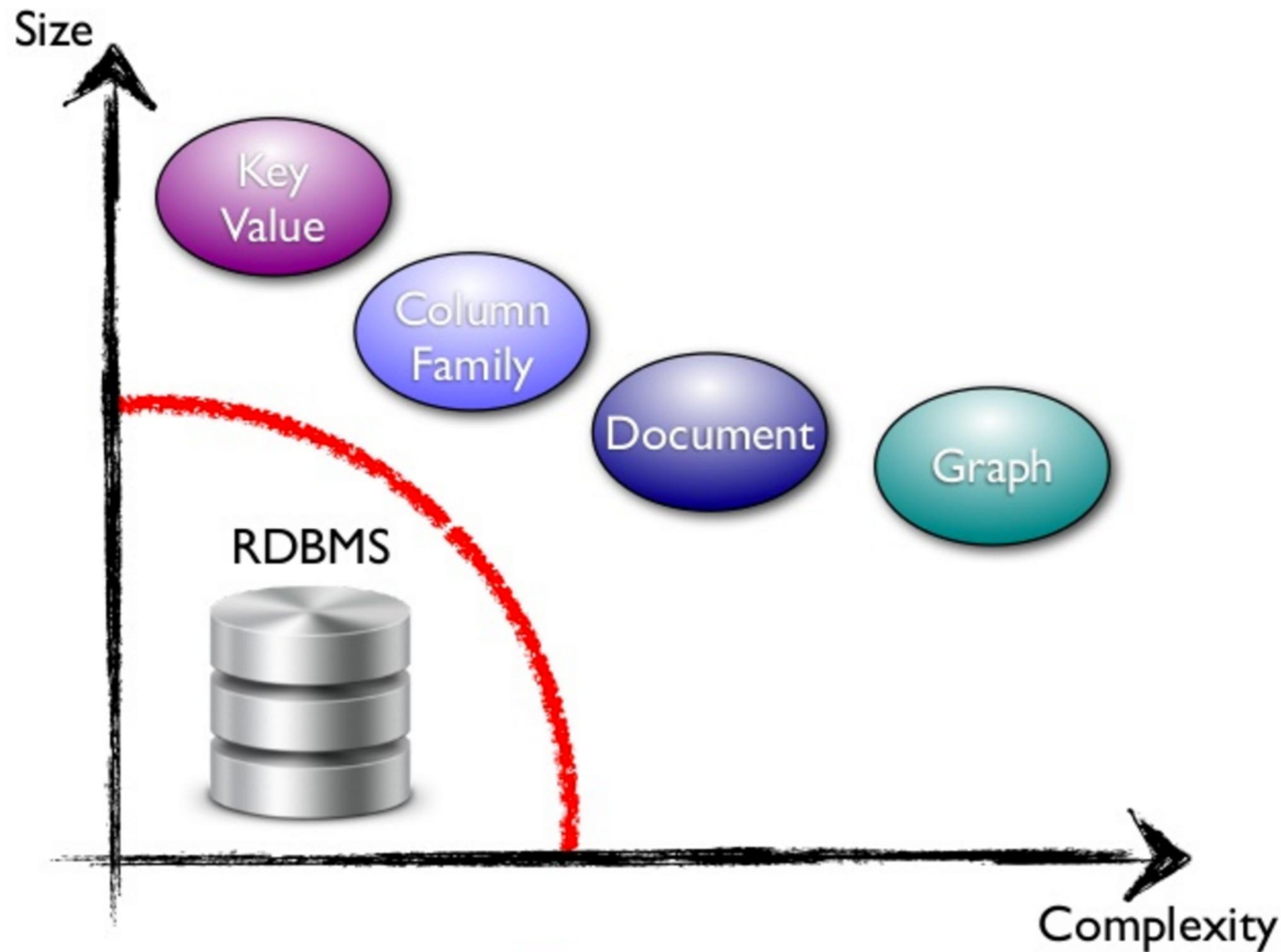
Graph



<https://www.thoughtworks.com/insights/blog/nosql-databases-overview>







Document

```
<Key=CustomerID>
```

```
{  
  "customerId": "fc986e48ca6" ← Key  
  "customer":  
  {  
    "firstname": "Pramod",  
    "lastname": "Sadalage",  
    "company": "ThoughtWorks",  
    "likes": [ "Biking", "Photography" ]  
  }  
  "billingaddress":  
  { "state": "AK",  
    "city": "DILLINGHAM",  
    "type": "R"  
  }  
}
```

Key

<https://www.thoughtworks.com/insights/blog/nosql-databases-overview>



Workshop

แนวคิด ในการออกแบบ



แนวคิด ในการออกแบบ

ความต้องการของระบบคืออะไร ?



แนวคิด ในการออกแบบ

What questions do i have ?



แนวคิดในการออกแบบ

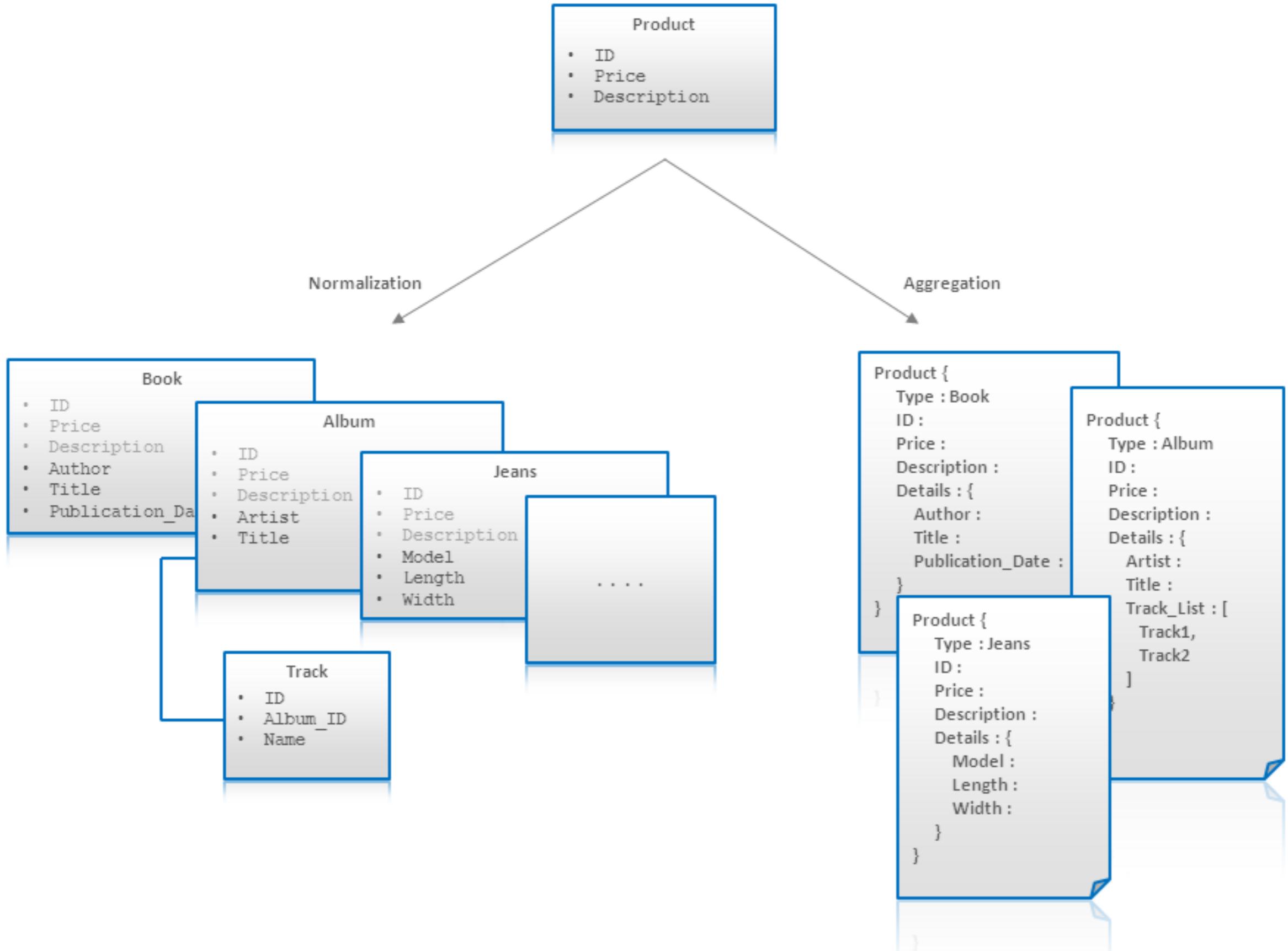
Data duplication & Denormalization



แนวคิดในการออกแบบ

Embedding over Referencing





มาอุ่นแบบกันเถอะ !!



All application development is
Schema design



**Success comes from
Proper data structure**



Schema design is
Flexible



Document schema design ?



1. Array

```
[  
  1, 2, 3, 4, "five",  
  6, "seven", [8, 9, 10]  
]
```



2. Embedded document

```
{  
    "name": "somkiat",  
    "age": 35,  
    "address" : {  
        "no": "122/64",  
        "province": "bangkok",  
        "zipcode": "10400"  
    }  
}
```



What is an Entity ?



What is an Entity ?

Object in your **model**
Associations with other entities



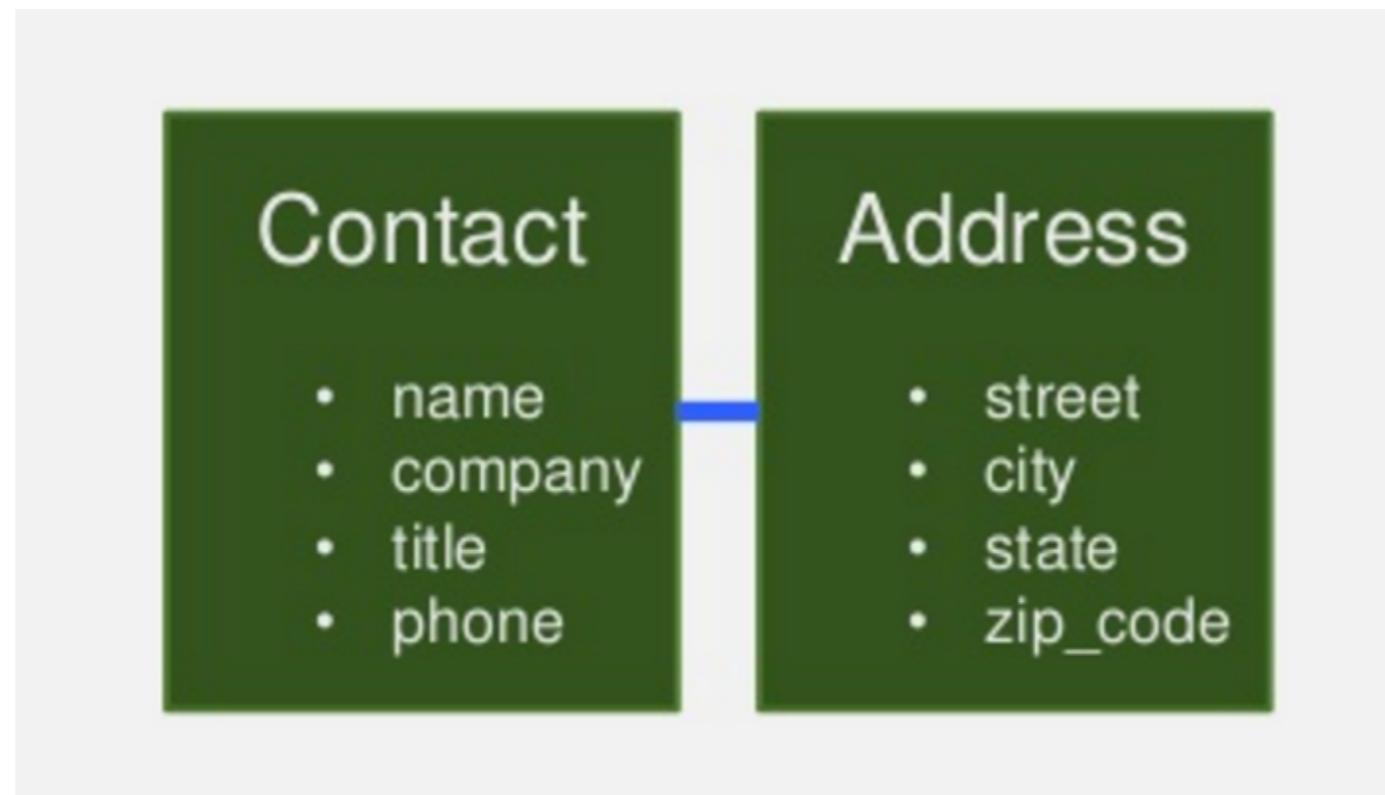
Association มี 2 แบบ

Referencing for relational database
Embedding for document database





Referencing



Embedding

Contact

- name
- company
- address
 - street
 - city
 - State
 - zip_code
- title
- phone



แตกต่างกันอย่างไร ? ทำไม ?

Contact

- name
- company
- title
- phone

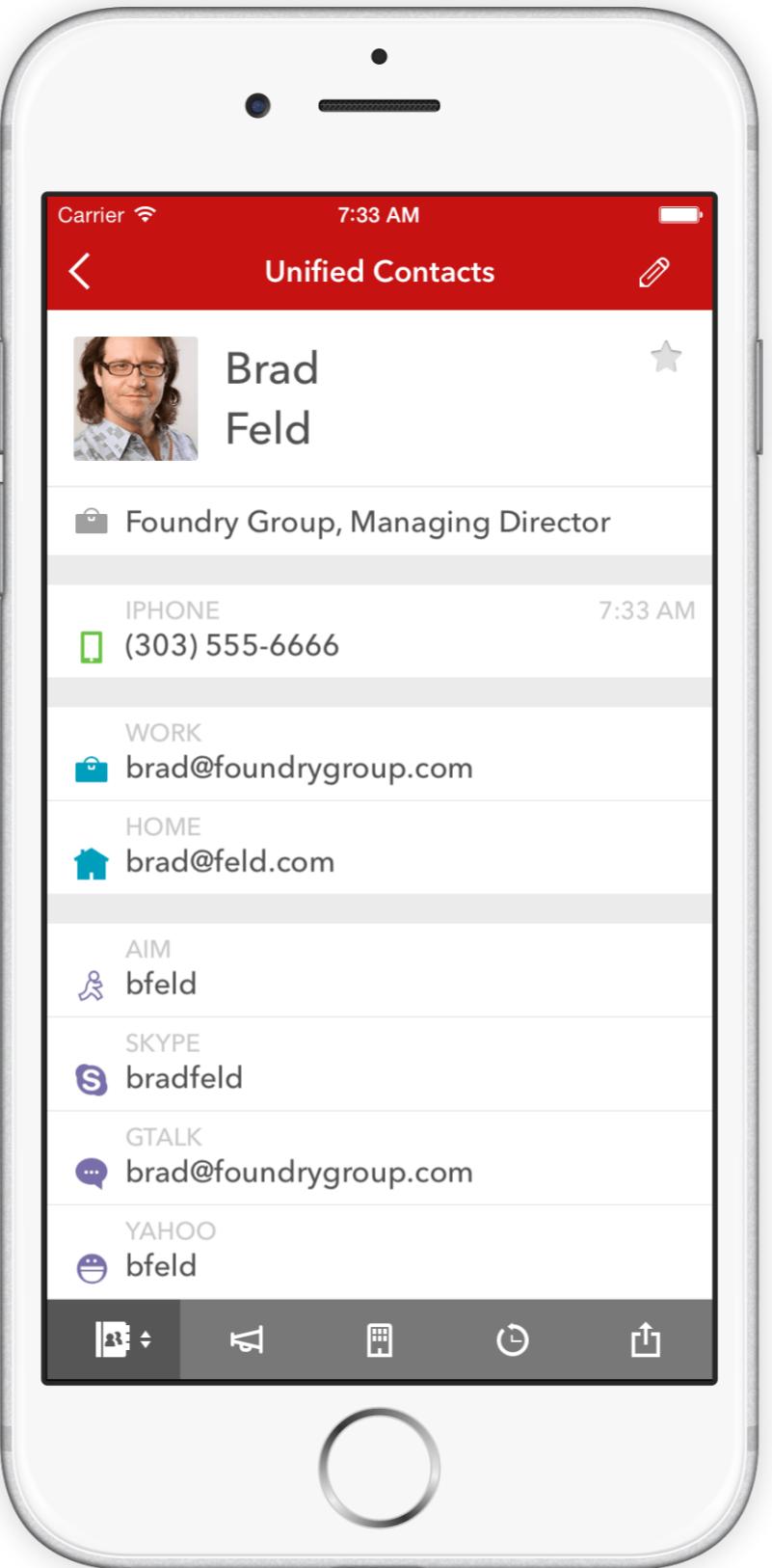
Address

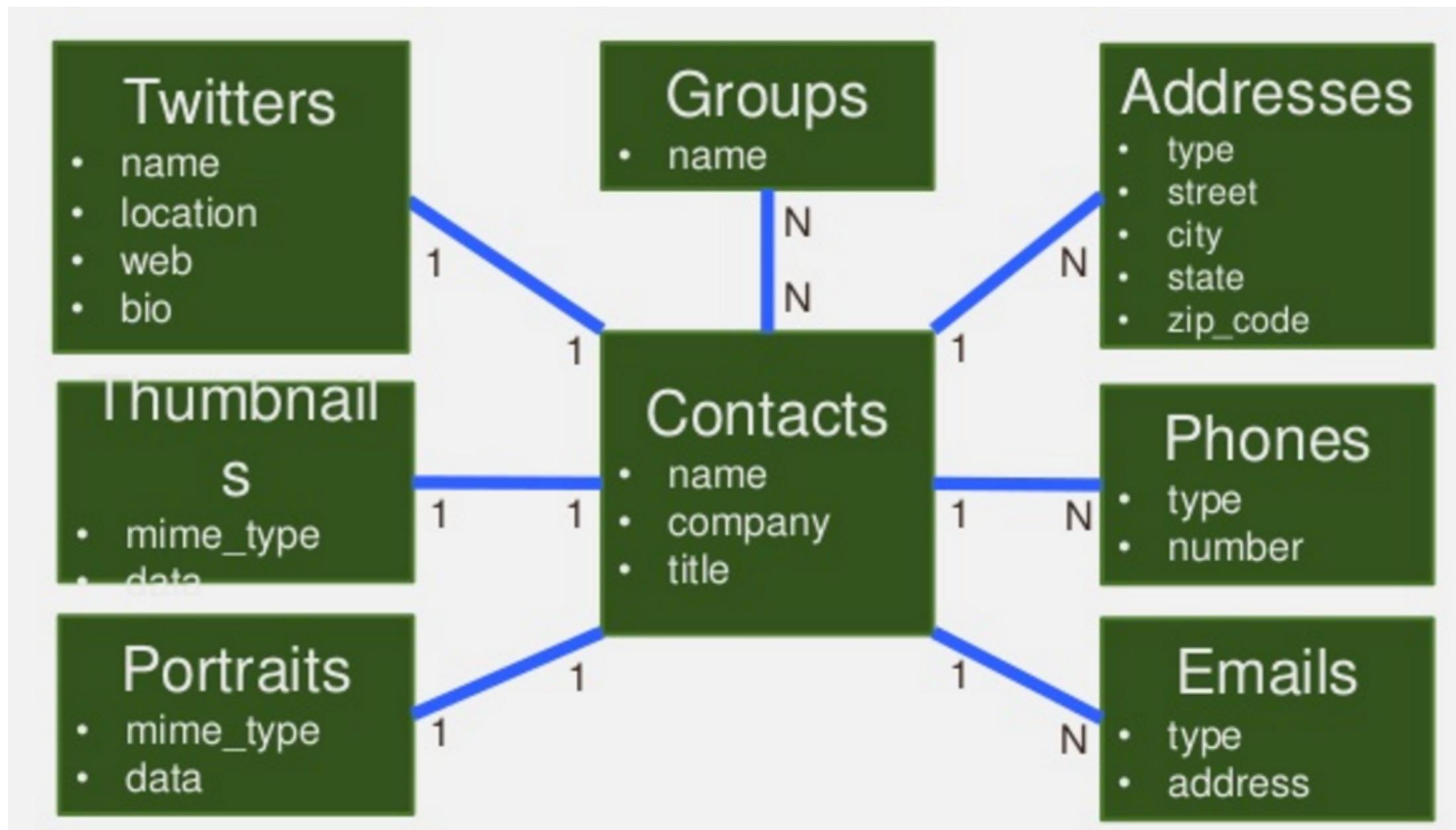
- street
- city
- state
- zip_code

Contact

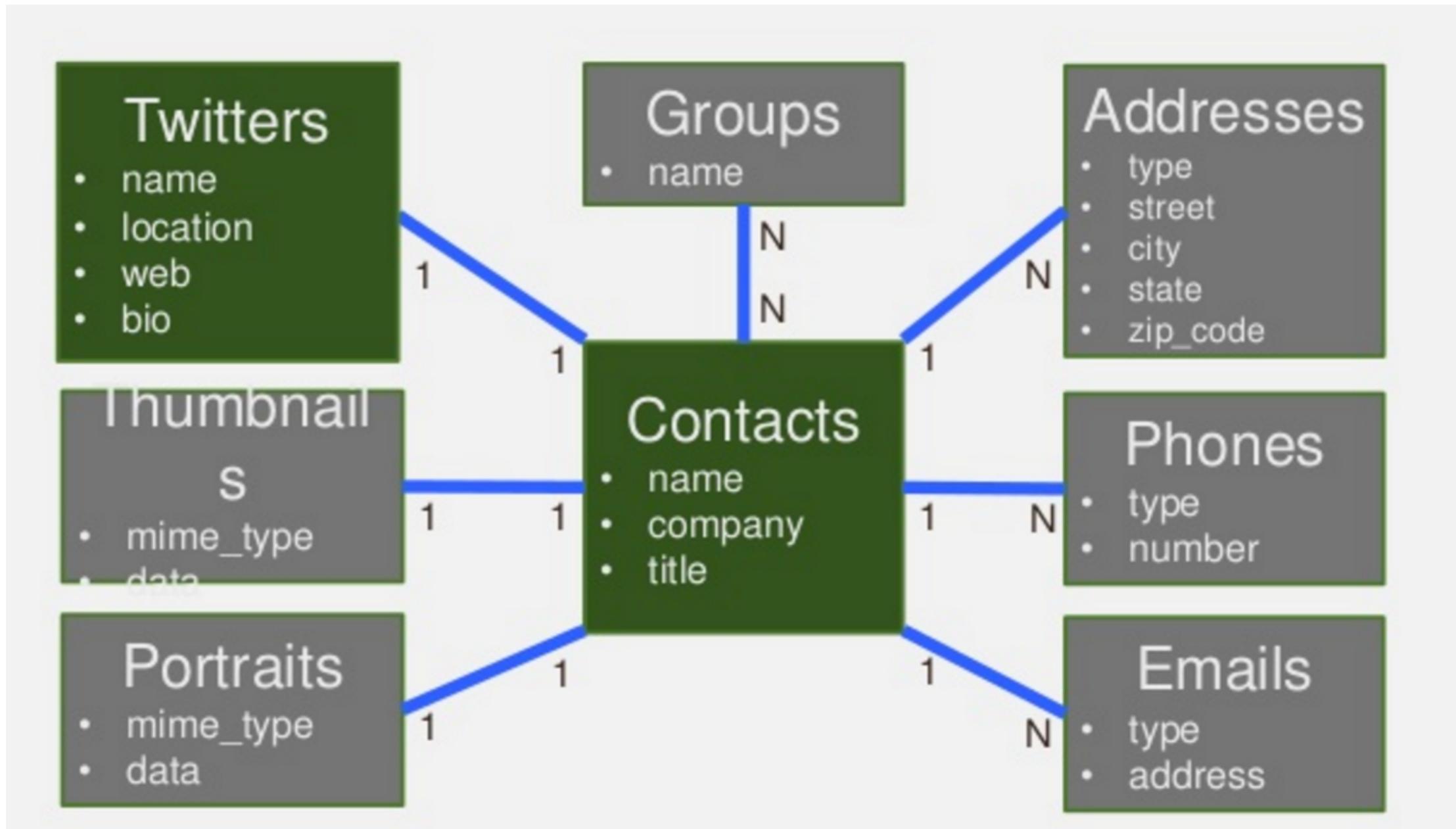
- name
- company
- address
 - street
 - city
 - state
 - zip_code
- title
- phone







One-to-One



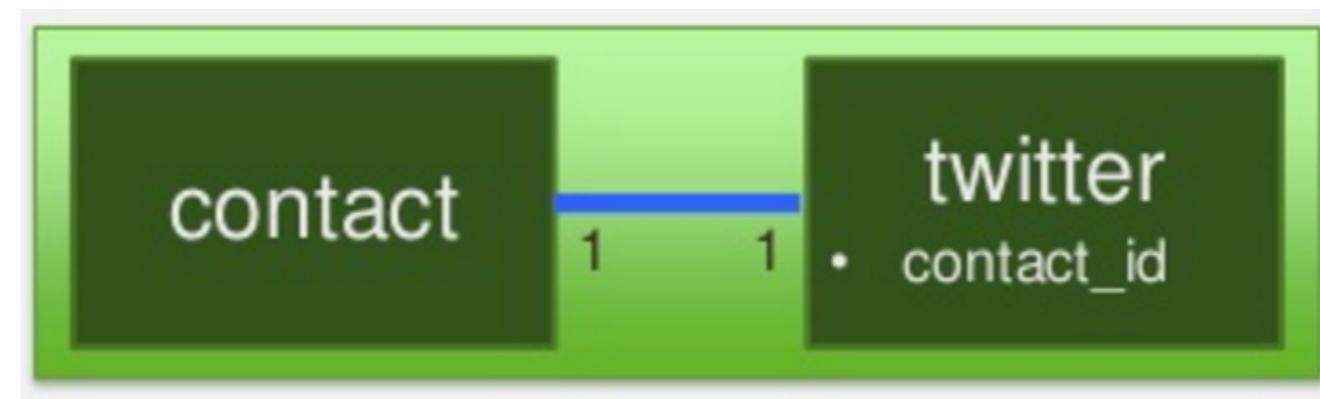
ออกแบบได้อย่างไรบ้าง ?



One-to-One :: 1



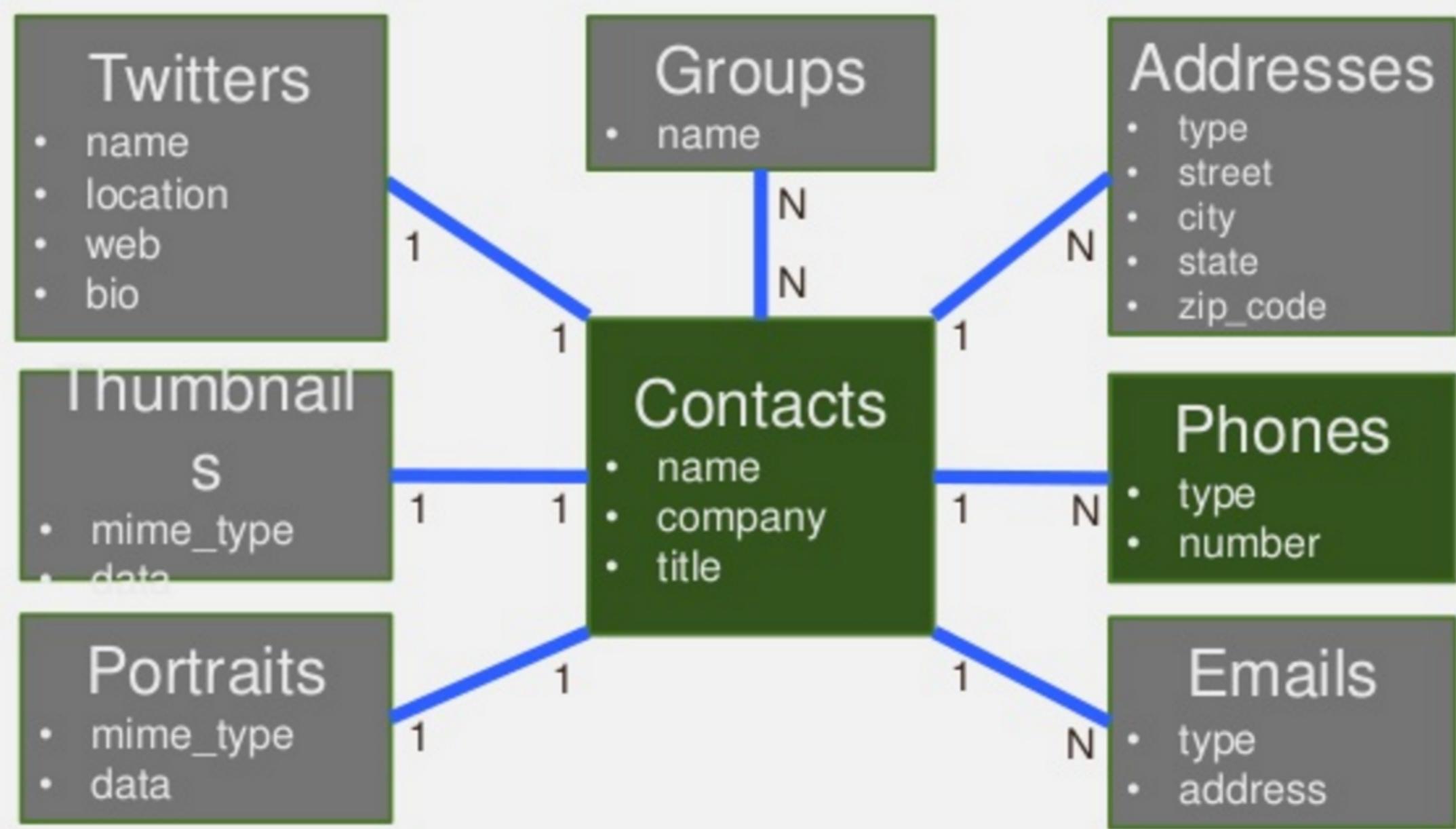
One-to-One :: 2



One-to-One :: 3



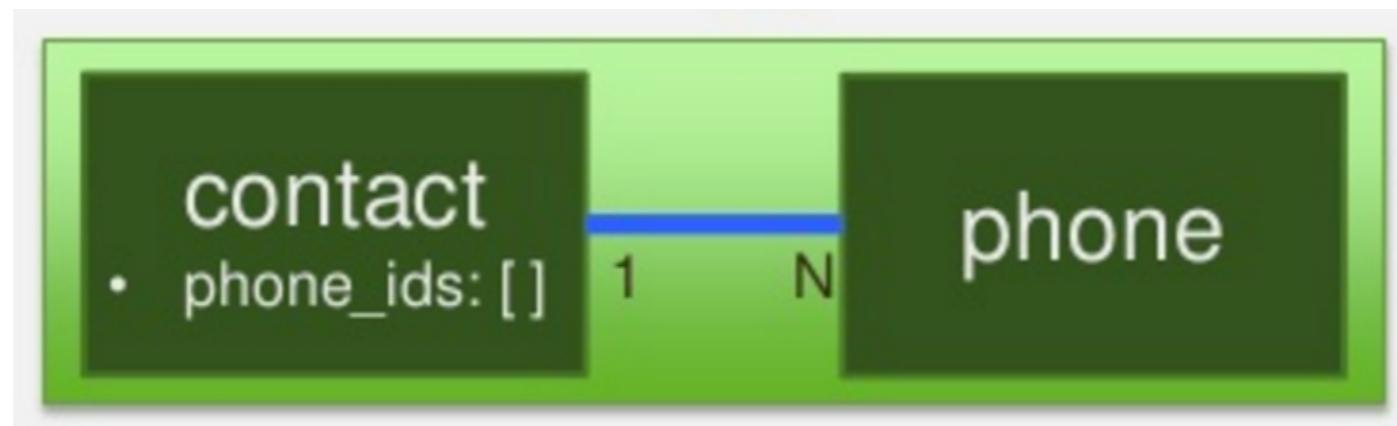
One-to-Many



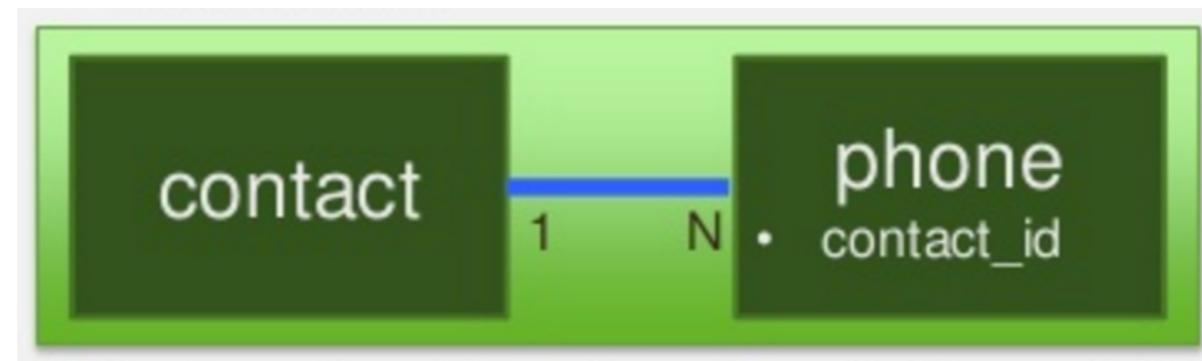
ออกแบบได้อย่างไรบ้าง ?



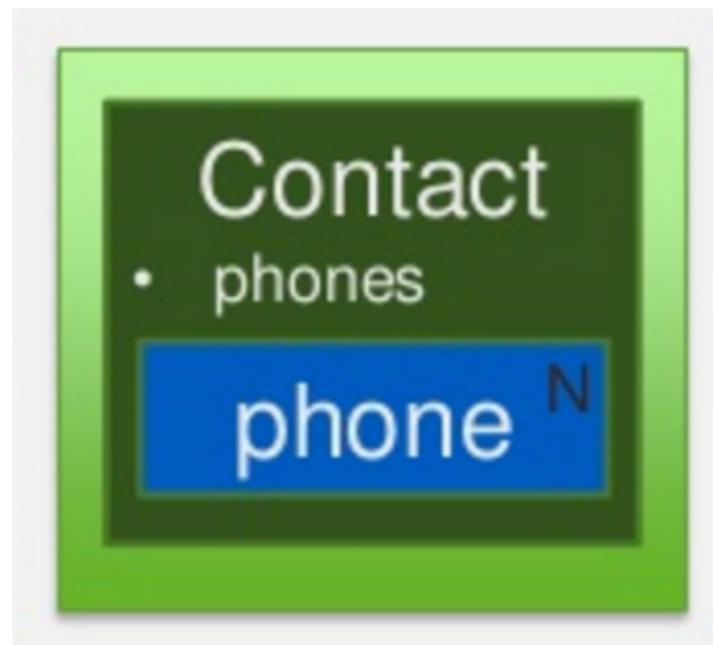
One-to-Many :: 1



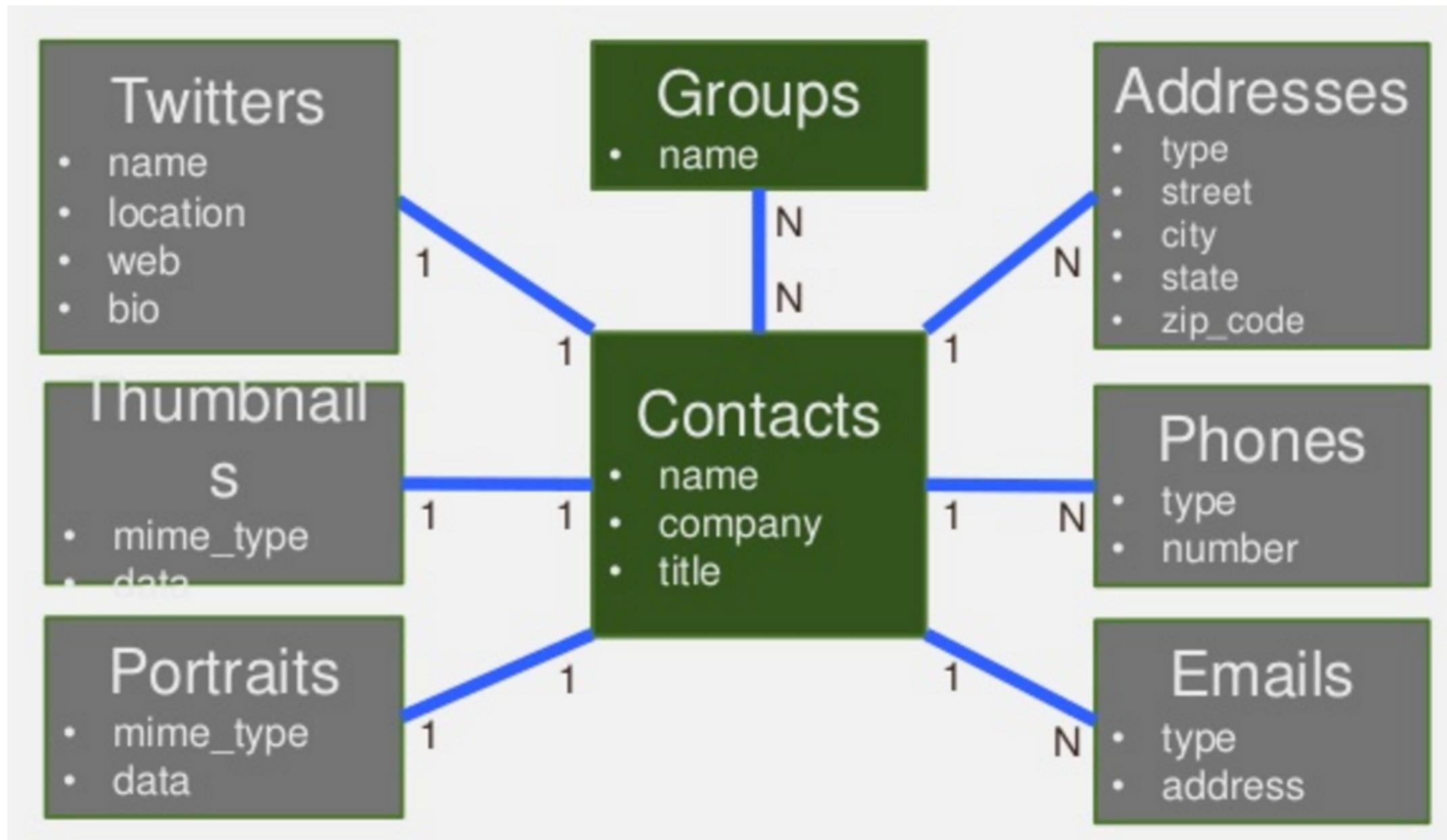
One-to-Many :: 2



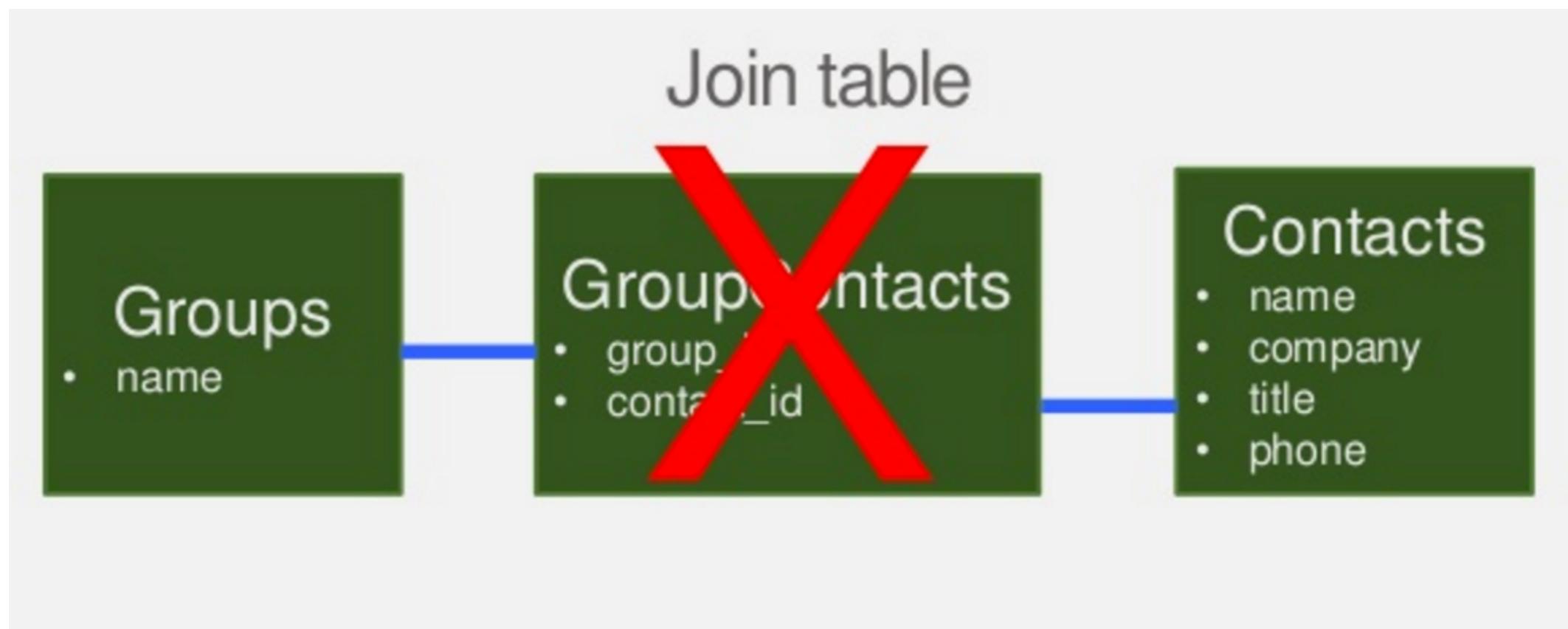
One-to-Many :: 3



Many-to-Many



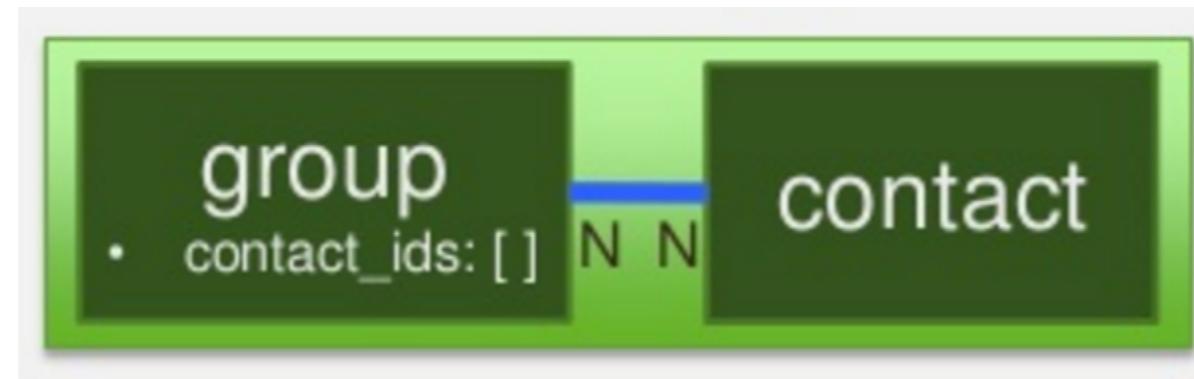
Many-to-Many



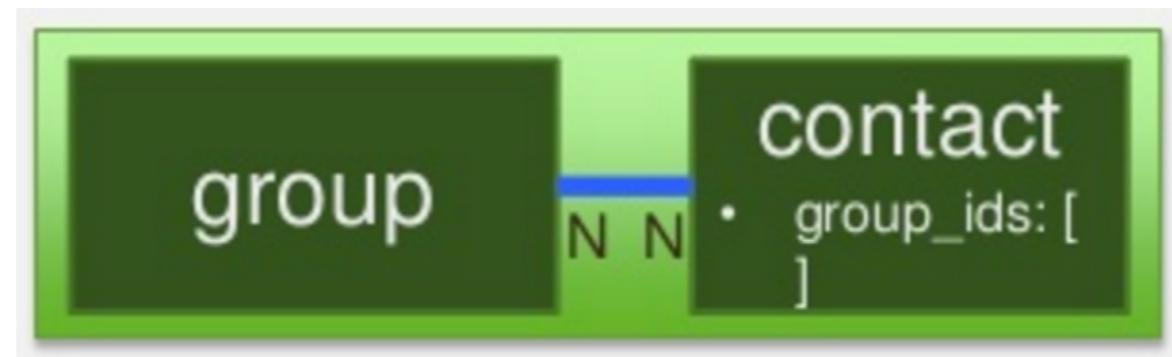
ออกแบบได้อย่างไรบ้าง ?



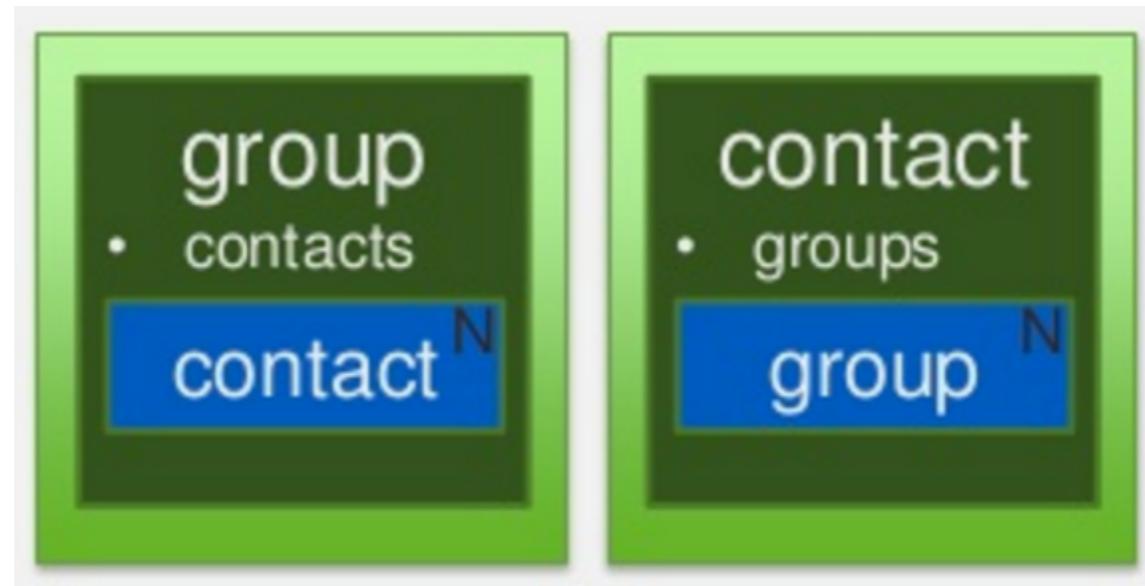
Many-to-Many :: 1



Many-to-Many :: 2

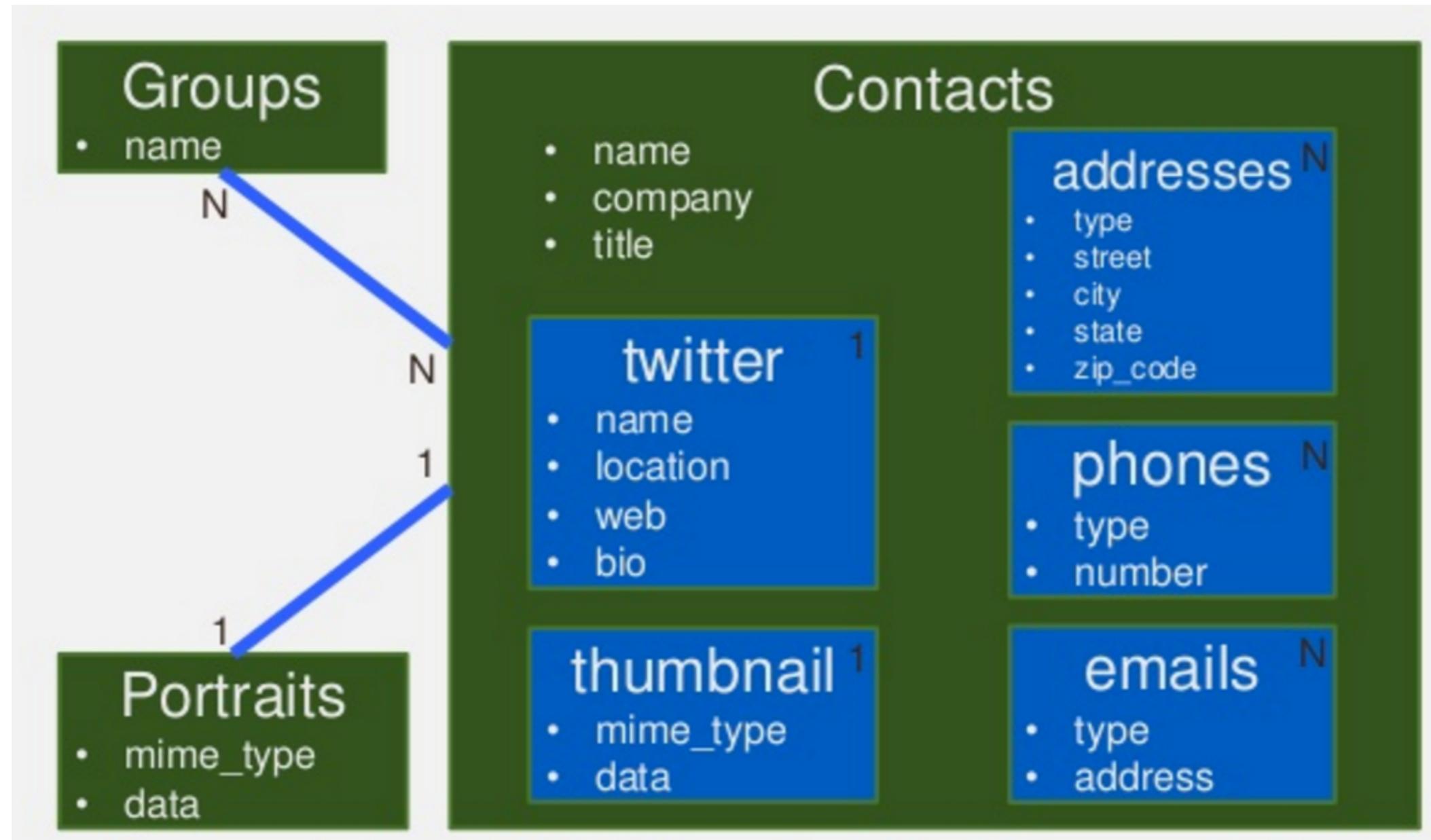


Many-to-Many :: 3



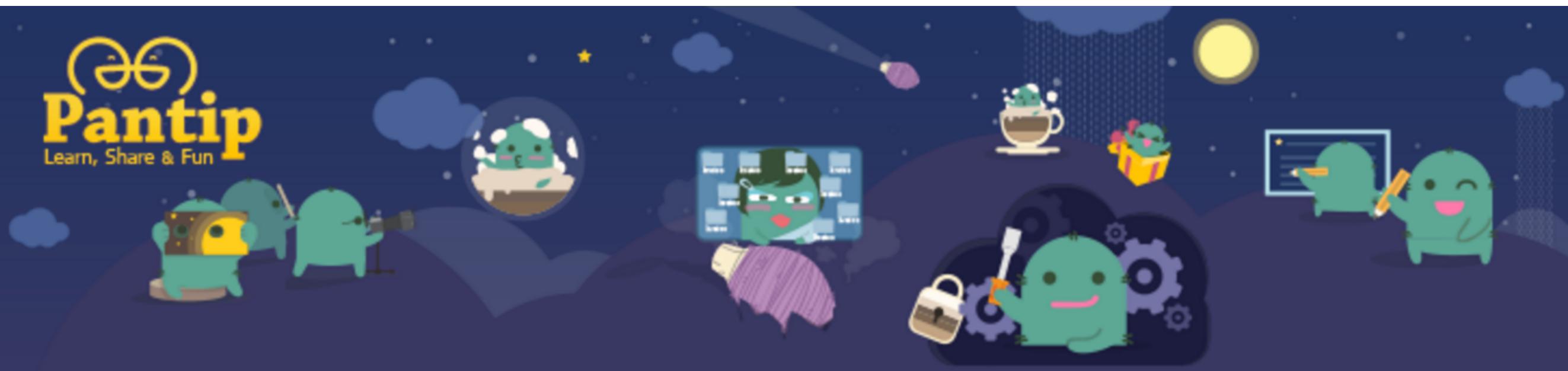
ສຸດທ້າຍແລ້ວ





ระบบตัวอย่าง





หน้าแรก

เลือกห้อง ▼

แท็ก

กิจกรรม N

อื่นๆ ▼

Smart Search



Workshop

ออกแบบด้วยแนวคิดของ RDBMS ?





เทคนิคในการออกแบบ NoSQL



1. Denormalization

Query data volume หรือ I/O per query

vs.

Total data volume



1. Denormalization

Processing complexity

vs.

Total data volume



2. Aggregation

Minimization of one-to-many relationship
Reduction of joins



2. Aggregation

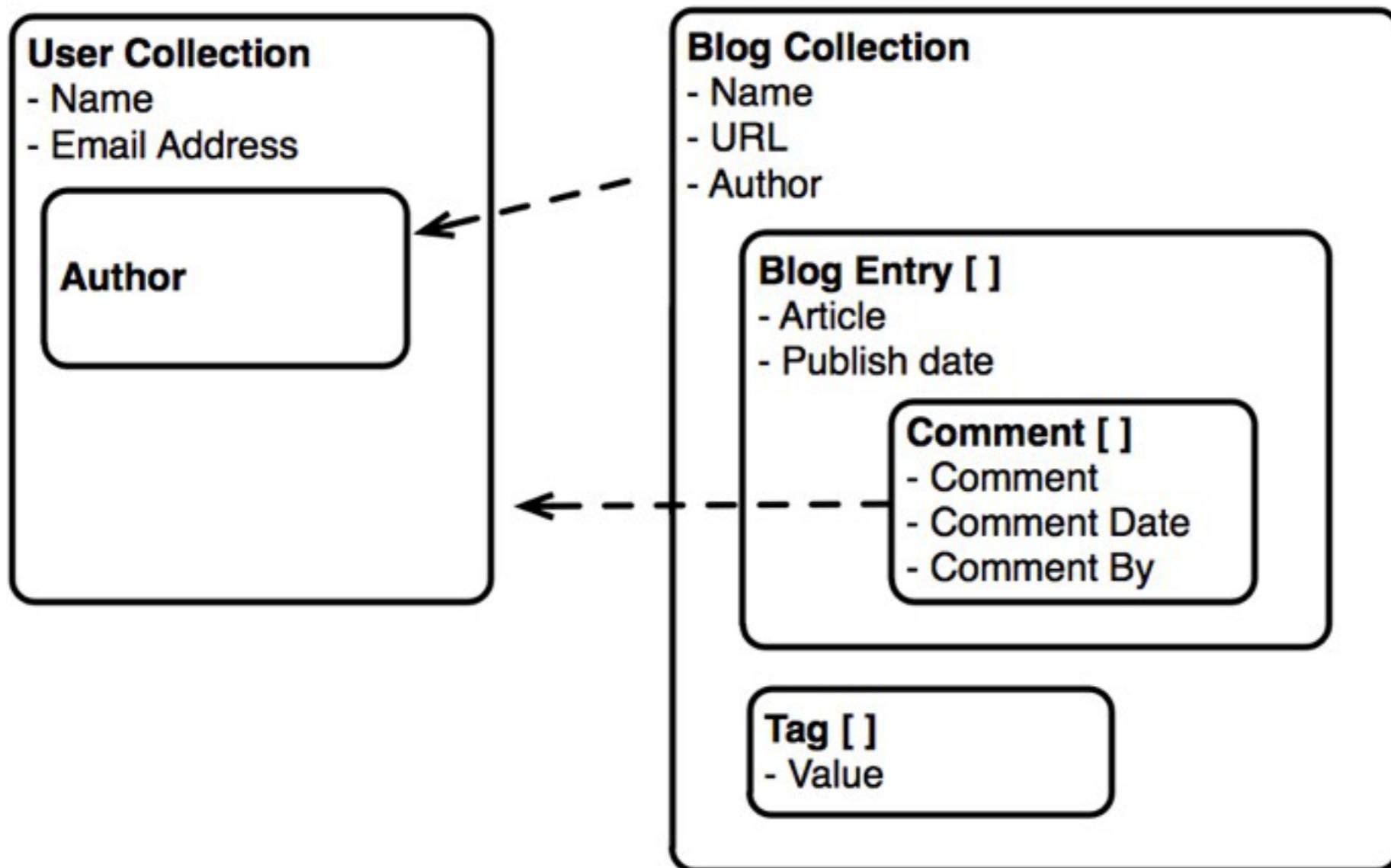
Minimization of one-to-many relationship

Reduction of joins

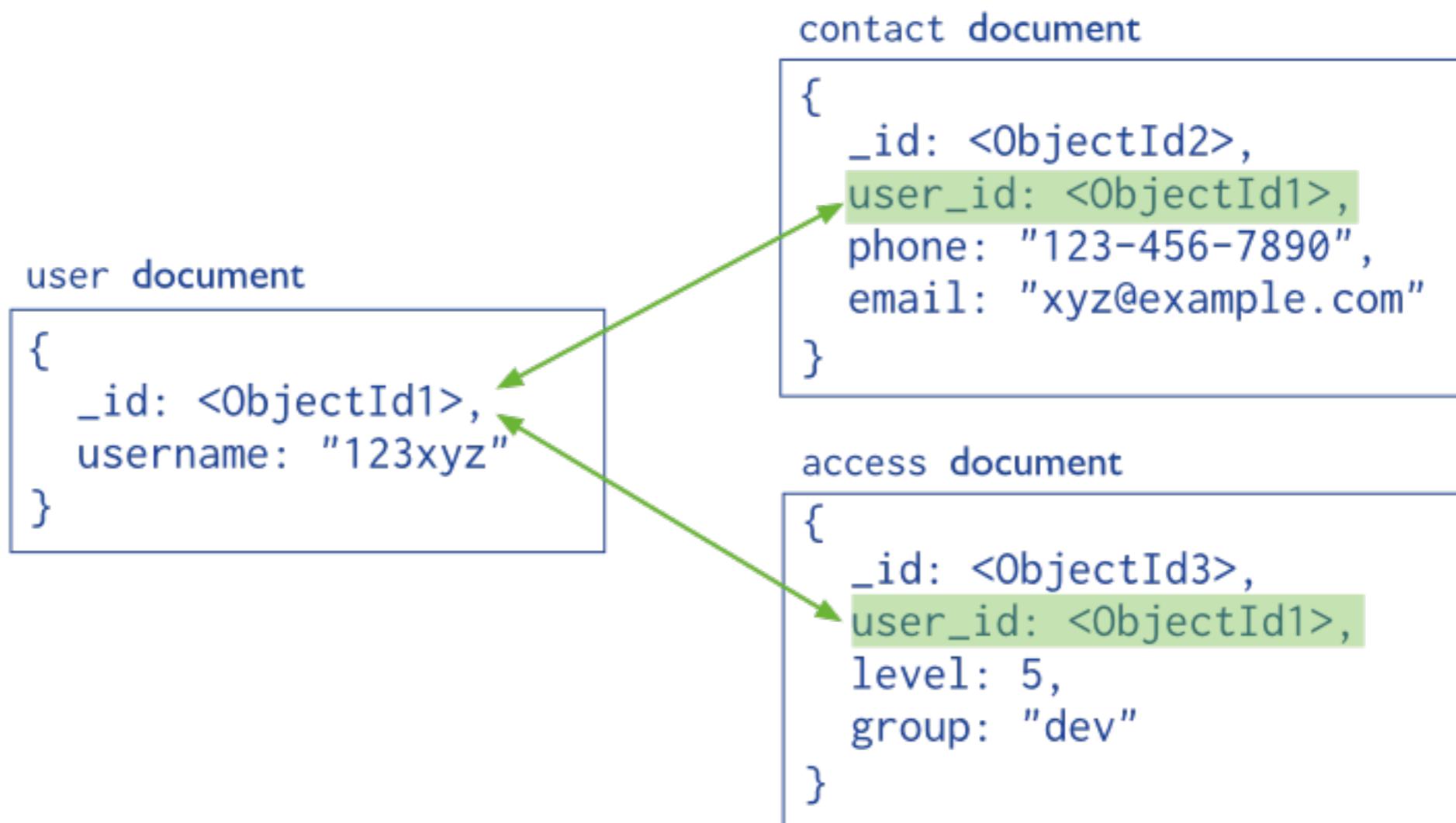
Nested or Embedded entity



Embedded document



Normalized data model



Embedded document

```
{  
  _id: <ObjectId1>,  
  username: "123xyz",  
  contact: {  
    phone: "123-456-7890",  
    email: "xyz@example.com"  
  },  
  access: {  
    level: 5,  
    group: "dev"  
  }  
}
```



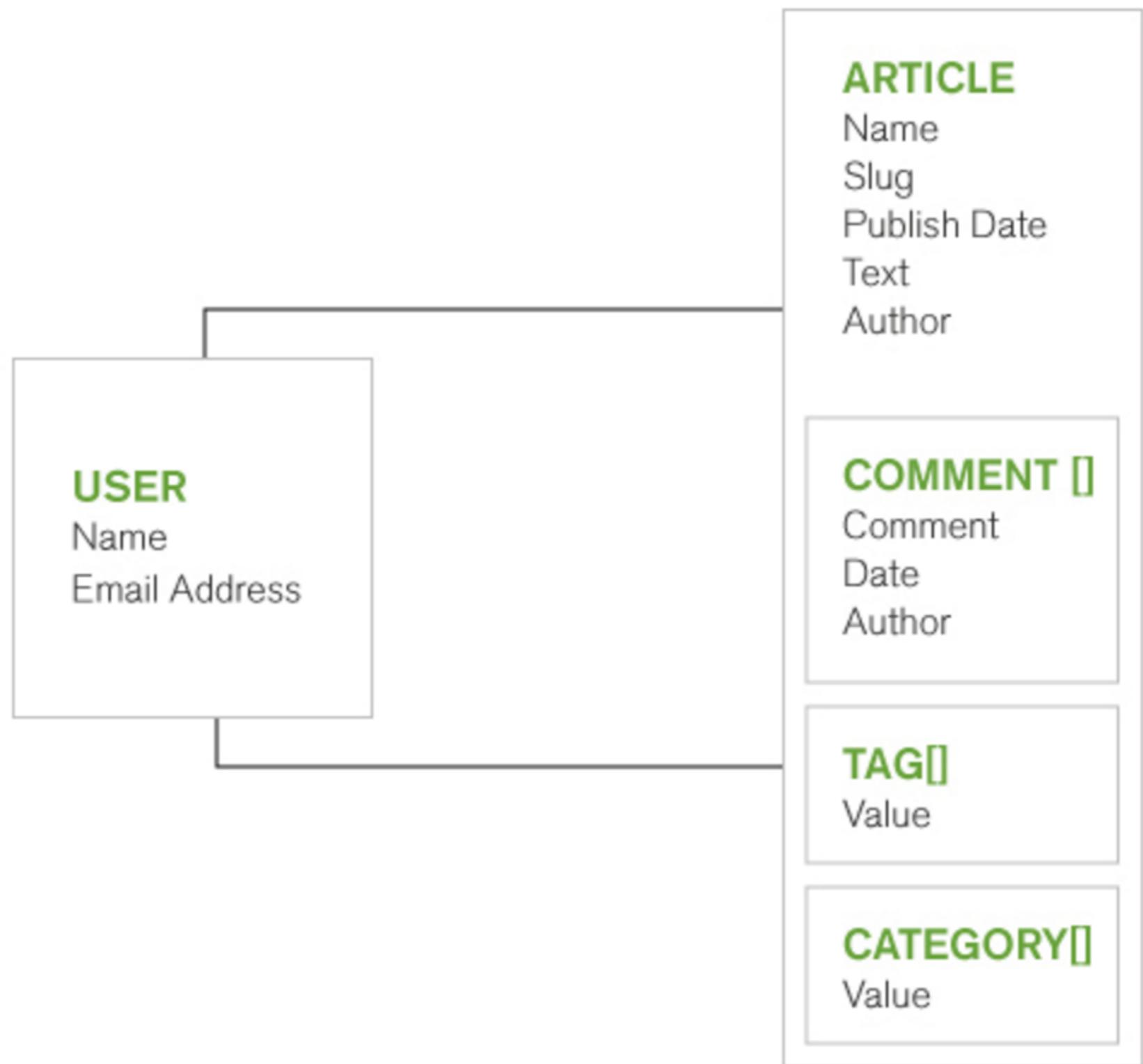
Embedded sub-document

Embedded sub-document



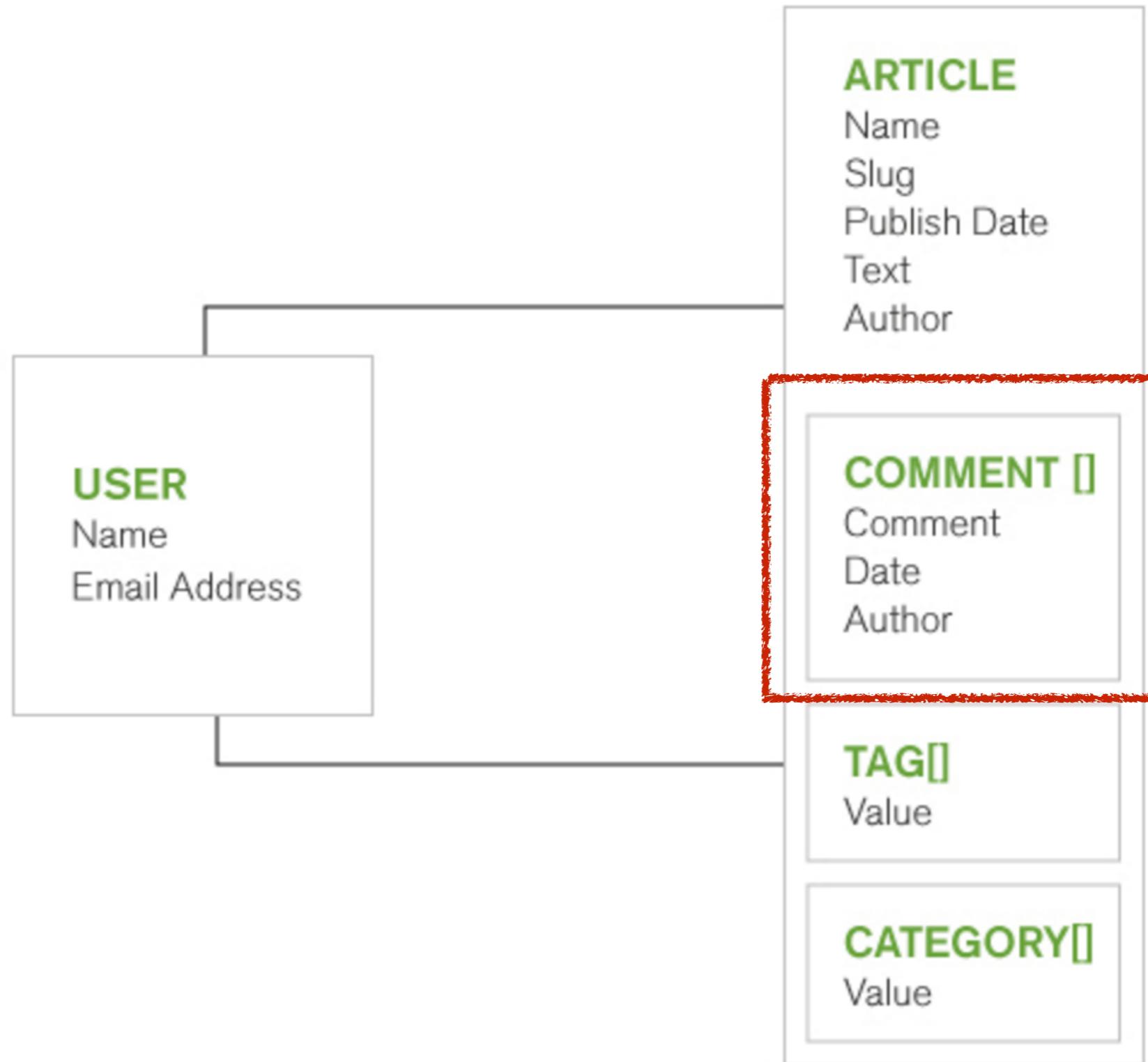
ออกแบบด้วยแนวคิดของ NoSQL





ปัญหาที่อาจจะเกิดขึ้น ?





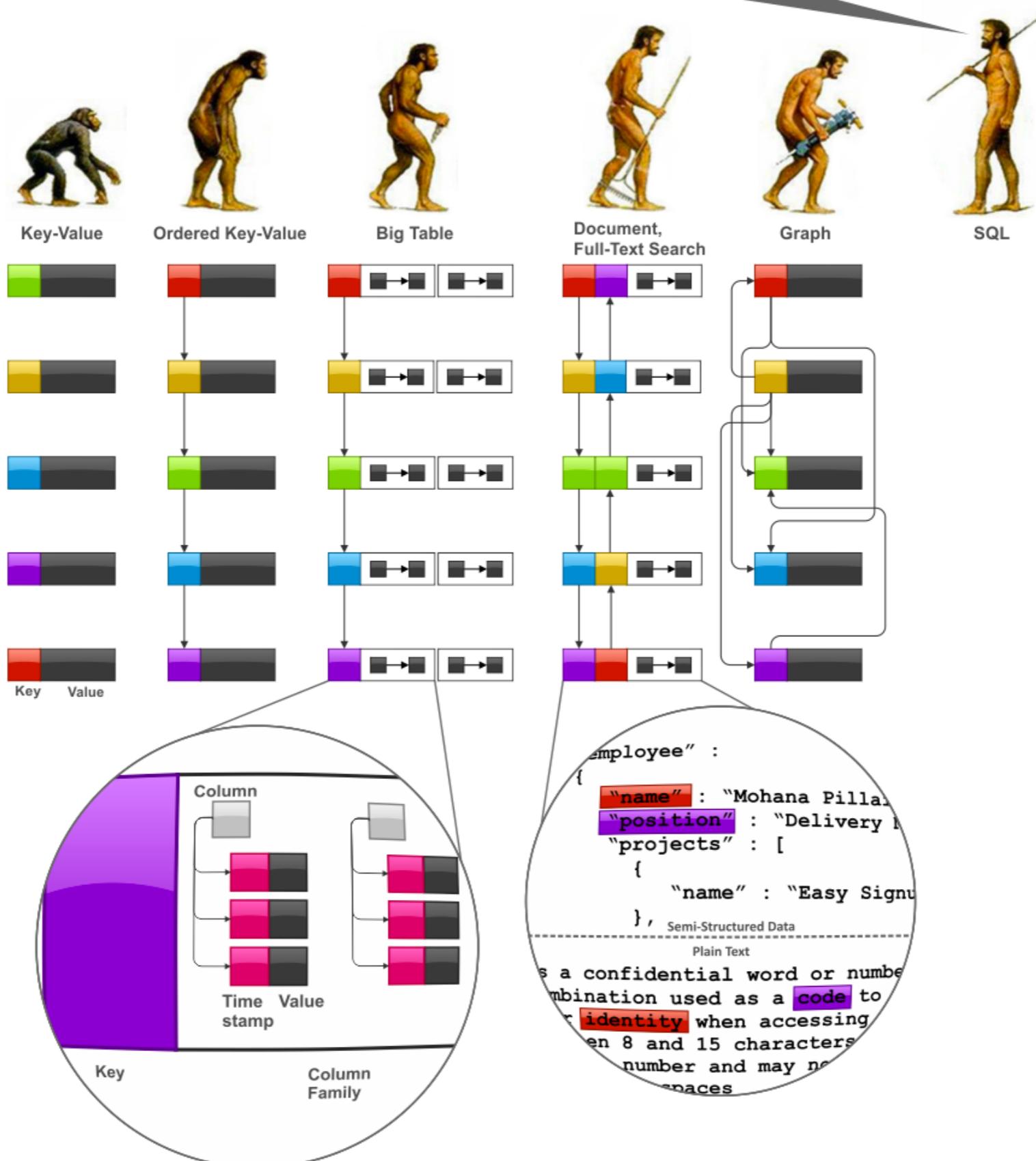
แก้อย่างไรดี ?



Embedding vs Referencing



Stop following me, you fucking freaks!



คำแนะนำสำหรับการออกแบบ



Iterate schema design development



Measure performance



Find bottlenecks



Embedding by default (90/10)



Referencing when you need more scaling



Normalize vs Denormalize



Flexible for development and change



Workshop



**KEEP
CALM**

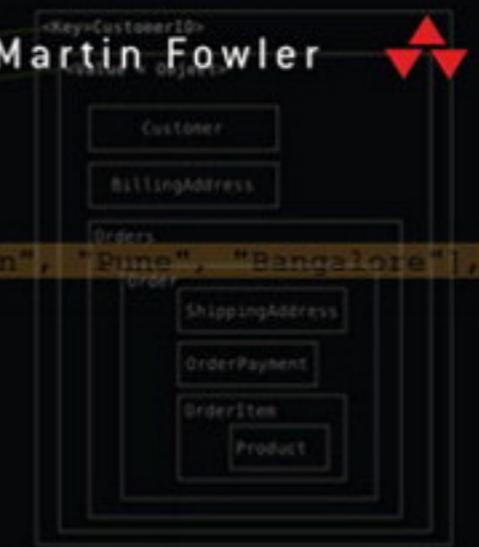
AND

**use the right tool
for the right job**



Pramodkumar J Sadalage & Martin Fowler

```
( "firstname": "Pramod",
  "citiesvisited": [ "Chicago", "London", "Pune", "Bangalore" ],
  "addresses": [
    { "state": "AK",
      "city": "DILLINGHAM",
      "type": "R"
    },
    { "state": "MH",
      "city": "PUNE",
      "type": "R"
    }
  ],
  "lastcity": "Chicago"
)
```



NoSQL

A Brief Guide to the Emerging World of Polyglot Persistence

Distilled

