

MongoDB 개요

2 SQL vs NoSQL

Structure & Schema

1.
Relational
Database

USER TABLE

_id	first_name	last_name	email	zipcode	phone
1	Spring	Melon	spring123@lake.co	123456	123-1234-1324
2	Summer	Peach	summer_00@lake.co	123457	123-2345-2345
3	Fall	Apple	falling@lake.co	654312	123-4567-4567
4	Winter	Strawberry	cold@lake.co	374728	123-5678-5678
5	Lake	Moon	lm@lake.co	182948	234-4567-2435
6	River	Star	long@lake.co	584829	123-2345-4567
7	Ocean	Earch	oe@lake.co	478328	123-6534-7645



Structure & Schema

1.
Relational
Database

USER TABLE

_id	first_name	last_name	email	instagram_id	twitter_id	zipcode	phone
1	Spring	Melon	spring123@lake.co	in_spring	tw_spring	123456	123-1234-1324
2	Summer	Peach	summer_00@lake.co	in_summer		123457	123-2345-2345
3	Fall	Apple	falling@lake.co	in_fall		654312	123-4567-4567
4	Winter	Strawberry	cold@lake.co	in_winter		374728	123-5678-5678
5	Lake	Moon	lm@lake.co		tw_lake	182948	234-4567-2435
6	River	Star	long@lake.co		tw_river	584829	123-2345-4567
7	Ocean	Earch	oe@lake.co	in_ocean	tw_ocean	478328	123-6534-7645

Structure & Schema

1.

Relational Database

장단점

- 데이터 중복을 방지할 수 있다.
- Join의 성능이 좋다.
- 복잡하고 다양한 쿼리가 가능하다.
- 잘못된 입력을 방지할 수 있다.
- 하나의 레코드를 확인하기 위해 여러 테이블을 Join하여 가시성이 떨어진다.
- 스키마가 엄격해서 변경에 대한 공수가 크다.

ADDRESS TABLE			SNS TABLE			
_id	user_id	zipcode	_id	user_id	type	contact_id
			1	1	email	spring123@lake.co
1	1	123456	2	1	instagram	in_spring
2	2	123457	3	1	twitter	tw_spring
3	3	654312	4	2	email	summer_00@lake.co
4	4	374728	5	2	instagram	in_summer
5	5	182948	6	3	email	falling@lake.co
6	6	584829	7	3	instagram	in_fall
7	7	478328	8	4	email	cold@lake.co
PHONE TABLE			9	4	instagram	in_winter
_id	user_id	phone	10	5	email	lm@lake.co
1	1	123-1234-1324	11	5	twitter	tw_lake
2	2	123-2345-2345	12	6	email	long@lake.co
3	3	123-4567-4567	13	6	twitter	tw_river
4	4	123-5678-5678	14	7	email	oe@lake.co
5	5	234-4567-2435	15	7	instagram	in_ocean
6	6	123-2345-4567	16	7	twitter	tw_ocean
7	7	123-6534-7645				

Scaling

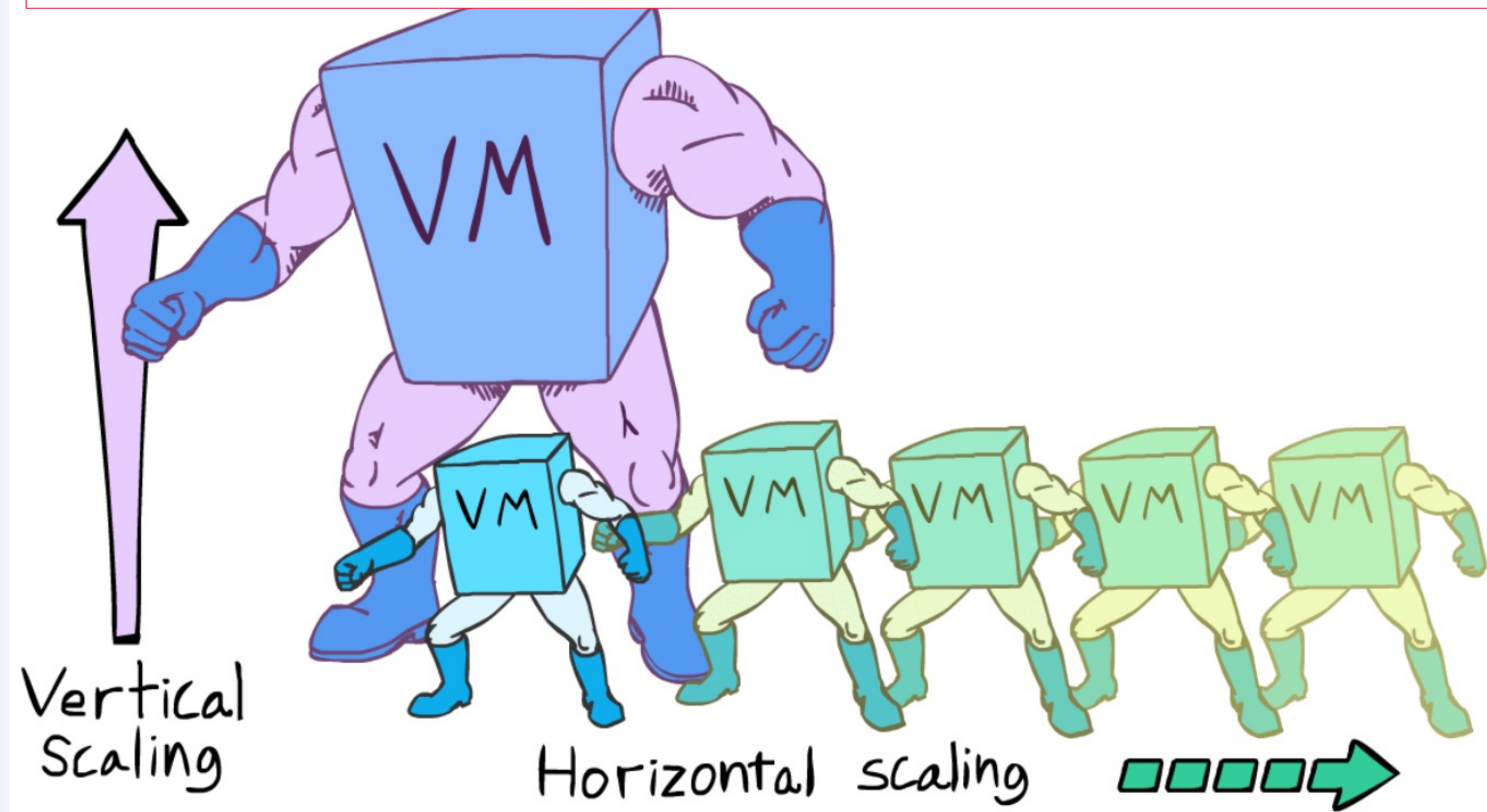
1.

Relational
Database

장단점

- Scale-Out이 가능하지만, 설정이 어렵다.
- 확장할 때마다 App단의 수정이 필요하다.
- 전통적으로 Scale-Up 위주로 확장했다.

왜 이렇게 제한되고 복잡할까요?



Scaling

1.

Relational
Database



Here's How Hard It Was to Move a 5MB IBM Hard Drive in 1956
<https://themindcircle.com/move-5mb-ibm-hard-drive-1956/>

Scaling



관계형 데이터베이스는 1970년대부터 사용되었는데, 당시 **Disk Storage**가
매우 고가의 제품으로 데이터 중복을 줄이는데 집중했다.

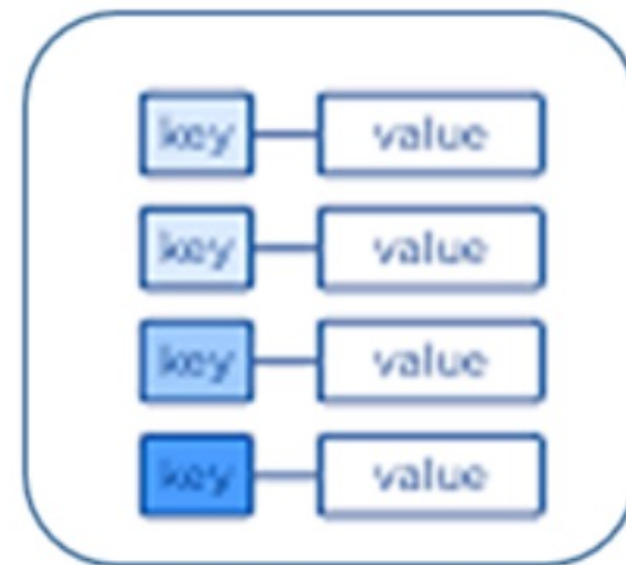


NoSQL

2. NoSQL



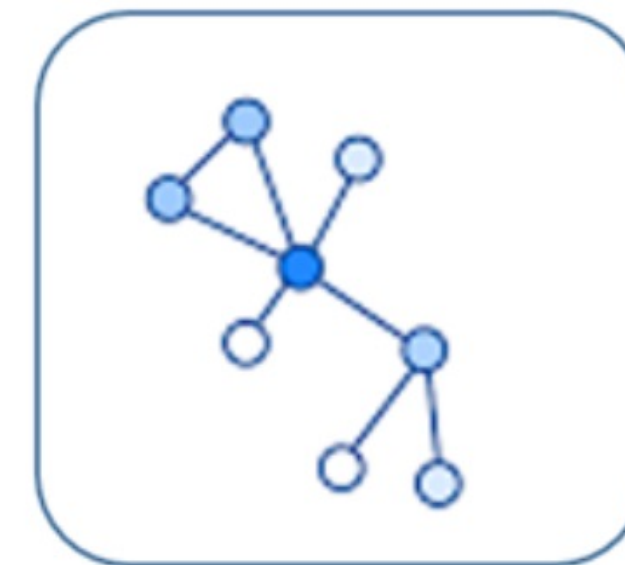
Document
Store



Key-Value
Store



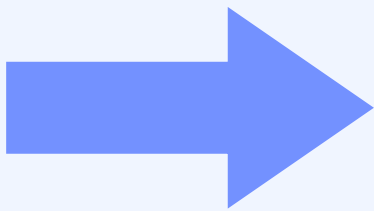
Wide-Column
Store



Graph
Store

Structure & Schema

USER TABLE			ADDRESS TABLE			SNS TABLE			
_id	first_name	last_name	_id	user_id	zipcode	_id	user_id	type	contact_id
1	Spring	Melon	1	1	123456	1	1	email	spring123@lake.co
2	Summer	Peach	2	2	123457	2	1	instagram	in_spring
3	Fall	Apple	3	3	654312	3	1	twitter	tw_spring
4	Winter	Strawberry	4	4	374728	4	2	email	summer_00@lake.co
5	Lake	Moon	5	5	182948	5	2	instagram	in_summer
6	River	Star	6	6	584829	6	3	email	falling@lake.co
7	Ocean	Earch	7	7	476328	7	3	instagram	in_fall
PHONE TABLE			_id	user_id	phone	8	4	email	cold@lake.co
			1	1	123-1234-1324	9	4	instagram	in_winter
			2	2	123-2345-2345	10	5	email	lm@lake.co
			3	3	123-4567-4567	11	5	twitter	tw_lake
			4	4	123-5678-5678	12	6	email	long@lake.co
			5	5	234-4567-2435	13	6	twitter	tw_river
			6	6	123-2345-4567	14	7	email	oe@lake.co
			7	7	123-6534-7645	15	7	instagram	in_ocean
						16	7	twitter	tw_ocean

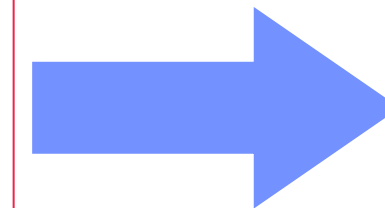


```
{
  _id: 1,
  first_name: "Spring",
  last_name: "Melon",
  zipcode: 123456,
  phone: "123-1234-1234",
  sns: [
    {type: "email", id: "spring123@lake.co"},
    {type: "instagram", id: "in_spring"},
    {type: "twitter", id: "tw_spring"}
  ]
}
```

Structure & Schema

장단점

- 데이터 접근성과 가시성이 좋다.
- Join없이 조회가 가능해서 응답 속도가 일반적으로 빠르다.
- 스키마 변경에 공수가 적다.
- 스키마가 유연해서 데이터 모델을 App의 요구사항에 맞게 데이터를 수용할 수 있다.
- 데이터의 중복이 발생한다.
- 스키마가 자유롭지만, 스키마 설계를 잘해야 성능 저하를 피할 수 있다.



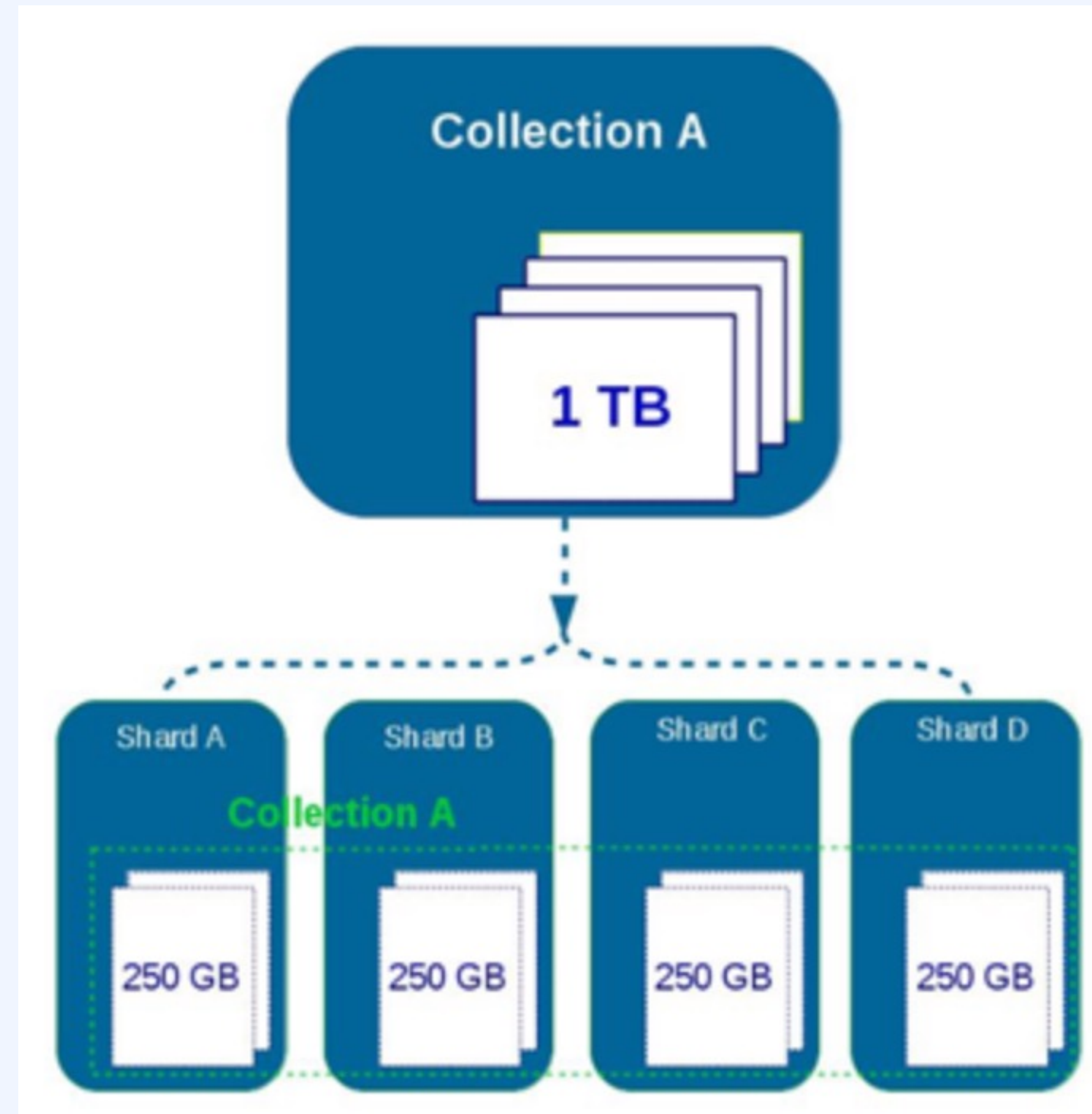
```
{
  _id: 1,
  first_name: "Spring",
  last_name: "Melon",
  zipcode: 123456,
  phone: "123-1234-1234",
  sns: [
    {type: "email", id: "spring123@lake.co"},
    {type: "instagram", id: "in_spring"},
    {type: "twitter", id: "tw_spring"}
  ]
}
```

Scaling

2. NoSQL

장단점

- HA와 Sharding에 대한 솔루션을 자체적으로 지원하고 있어 Scale-Out이 간편하다.
- 확장 시, Application의 변경사항이 없다.



Summary

3.

요약

- MongoDB는 Document 지향 Database이다.
- 데이터 중복이 발생할 수 있지만, 접근성과 가시성이 좋다.
- 스키마 설계가 어렵지만, 스키마가 유연해서 Application의 요구사항에 맞게 데이터를 수용할 수 있다.
- 분산에 대한 솔루션을 자체적으로 지원해서 Scale-Out이 쉽다.
- 확장 시, Application을 변경하지 않아도 된다.

MongoDB는 유연하고 확장성 높은 Opensource Document 지향 Database이다.