

Database

Database

1. RDB
2. 스키마
3. 관계
4. 1:N 모델링

Database

여러 사람이 공유하여 사용할 목적으로
체계화하여 관리하는 데이터의 집합이다.

wikipedia

RDB (관계형 데이터베이스)

관계형 모델을 기반으로 하는 전통적인 데이터베이스

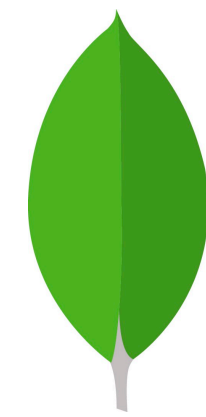
RDBMS

관계형 데이터베이스를 관리하는 시스템



NoSQL

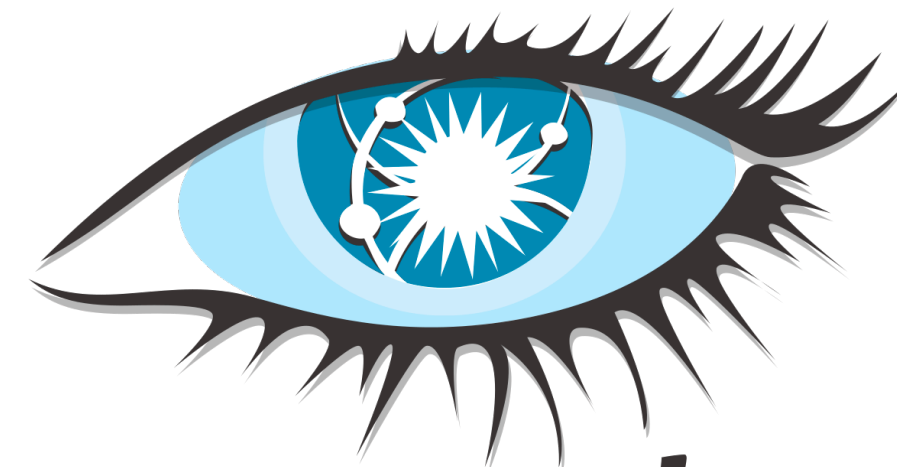
전통적인 RDBMS와 달리 유연한 스키마를 가진 데이터베이스



mongoDB



redis



cassandra

NoSQL

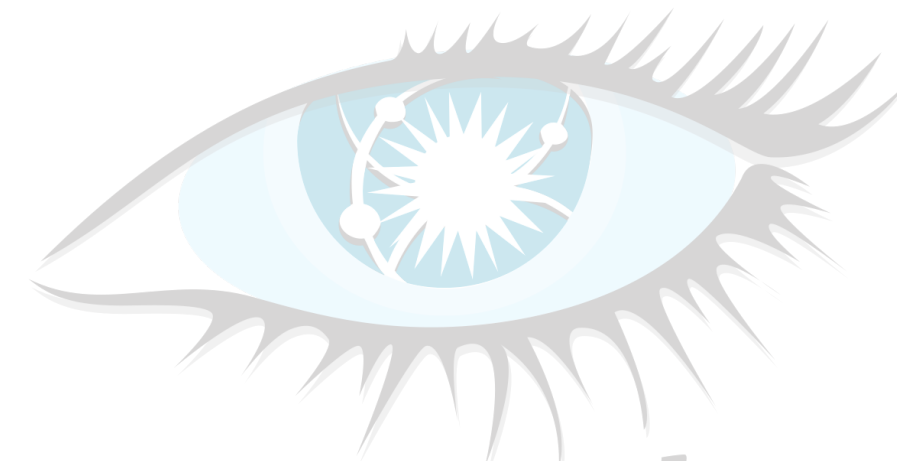
전통적인 RDBMS와 달리 유연한 스키마를 가진 데이터베이스

RDBMS부터 공부하고...!

 mongoDB





redis



cassandra

RDB (관계형 데이터베이스)

A1							
	A	B	C	D	E	F	
1							
2							
3							
4							
5							
6							
7							

여러 사람이 동시에 수정할 수 있는
거대한 스프레드시트의 집합

RDB (관계형 데이터베이스)

여러 사람이 동시에 수정할 수 있는
거대한 스프레드시트의 집합

A1						
	A	B	C	D	E	F
1						
2						
3						
4						
5						
7						

RDB (관계형 데이터베이스)

여러 사람이 동시에

수정할 수 있는

거대한 스프레드시트의 집합

평범한 스프레드시트

A1		<i>fx</i>					
	A	B	C	D	E	F	
1							
2							
3							
4							
5							
6							
7							

Table

A1		<i>fx</i>					
	A	B	C	D	E	F	
1							
2							
3							
4							
5							
6							
7							

또는 Relation 이라고 부름...

Column에는 데이터 타입을 지정합니다.

A1		<i>fx</i>					
	A	B	C	D	E	F	
1							
2							
3							
4							
5							
6							
7							

Column
(Field 또는 Attribute)

Row
(Tuple
또는
Record)

A1		fx					
	A	B	C	D	E	F	
1							
2							
3							
4							
5							
6							
7							

Row에는 각 Column에 맞는 데이터가 저장됩니다.

Table

Row
(Tuple
또는
Record)

A1		∇	f_x				
	A	B	C	D	E	F	
1							
2							
3							
4							
5							
6							
7							

Column
(Field 또는 Attribute)

스키마...?

1. Table 이름을 지정하고

A1		fx				
	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						
7						

2. Column의 이름과 함께 타입을 지정합니다.

게시글 (Article)

	A	B	C	D	E
1	pk (INT)	제목 (varchar)	내용 (text)	생성일 (datetime)	수정일 (datetime)
2					
3					
4					
5					
6					
7					

즉, 스키마는 테이블의 구조를 뜻하며
테이블의 이름과 컬럼의 집합으로 구성

게시글 (Article)

Row
(Tuple
또는
Record)

	A	B	C	D	E
1	pk (INT)	제목 (varchar)	내용 (text)	생성일 (datetime)	수정일 (datetime)
2	1	즐거운 장고	즐거워요...	2021. 03. 23.	2021. 03. 23.
3					
4					
5					
6					
7					

스키마를 기반으로 만들어지는
실제 데이터가 Row!

관계형 데이터베이스?

Database

E		F				

A1	▼	<i>fx</i>				
	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						
7						

A1	▼	<i>fx</i>				
	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						
7						

A1	▼	<i>fx</i>				
	A	B	C	D	E	
1						
2						
3						
4						
5						
6						
7						

D	E	F				

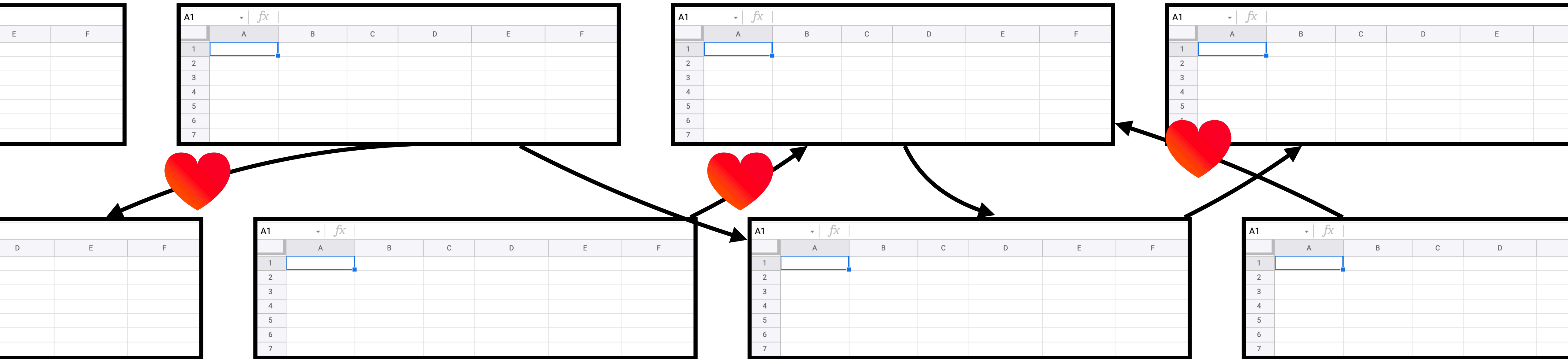
A1	▼	<i>fx</i>				
	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						
7						

A1	▼	<i>fx</i>				
	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						
7						

A1	▼	<i>fx</i>				
	A	B	C	D		
1						
2						
3						
4						
5						
6						
7						

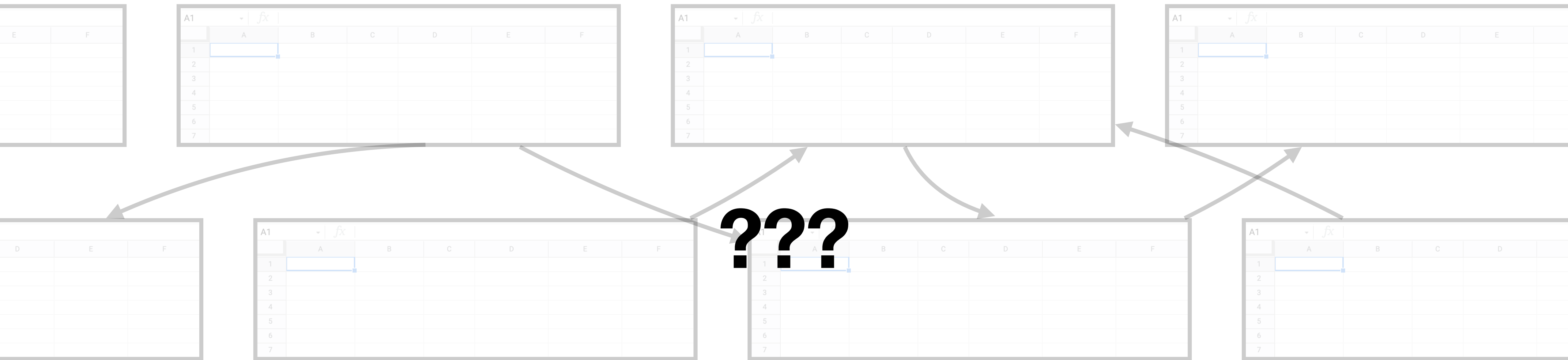
하나의 데이터베이스에는
수 많은 테이블이 존재할 수 있다.

Database



그리고 그 테이블 사이에는
관계가 제어될 수 있다.

Database



그리고 그 테이블 사이에는
관계가 지어질 수 있다.

왜?

현실 세계를 표현하기 위해!

DB 모델링

현실 세계를 표현해보자.

게시글도 작성하고

게시글에 댓글도 작성하고 싶다면?

게시글 (Article)

	A	B	C	D	E
1	pk (INT)	제목 (varchar)	내용 (text)	생성일 (datetime)	수정일 (datetime)
2					
3					
4					
5					
6					
7					

1. 게시글 테이블을 만든다.

게시글 (Article)

	A	B	C	D	E
1	pk (INT)	제목 (varchar)	내용 (text)	생성일 (datetime)	수정일 (datetime)
2					
3					
4					
5					
6					
7					

2. 댓글도 저장하자. 그런데 어떻게?

게시글 (Article)

	A	B	C	D	E	F
1	pk (INT)	제목 (varchar)	내용 (text)	생성일 (datetime)	수정일 (datetime)	댓글 (text)
2	1	즐거운 장고	즐거워요...	2021. 03. 23.	2021. 03. 23.	유저1: 저도 즐거워용 유저2: 노잼 유저3: 이놈의 디장고...
3						
4						
5						
6						
7						

아하, **필드를 하나 추가**해서 거기에 다 저장하자!

게시글 (Article)

약속 1.

	A	B	C	D	E	F
1	pk (INT)	제목 (varchar)	내용 (text)	생성일 (datetime)	수정일 (datetime)	댓글 (text)
2	1	즐거운 장고	즐거워요...	2021. 03. 23.	2021. 03. 23.	유저1: 저도 즐거워용 유저2: 노잼 유저3: 이놈의 디장고...
3						
4						
5						
6						
7						

하나의 필드에는
하나의 값만 저장할 것.

필드를 하나 추가해서 거기에 다 저장하자!

게시글 (Article)

	A	B	C	D	E	F	G	H
1	pk (INT)	제목 (varchar)	내용 (text)	생성일 (datetime)	수정일 (datetime)	댓글1 (text)	댓글2 (text)	댓글3 (text)
2	1	즐거운 장고	즐거워요...	2021. 03. 23.	2021. 03. 23.	유저1: 저도 즐거워용	유저2: 노잼	유저3: 이놈의 디장고...
3								
4								
5								
6								
7								

그러면... **필드를 몇 개 더 추가**해서 거기에 저장하자!

게시글 (Article)

약속 2.

	A	B	C	D	E	F	G	H
1	pk (INT)	제목 (varchar)	내용 (text)	생성일 (datetime)	수정일 (datetime)	댓글1 (text)	댓글2 (text)	댓글3 (text)
2	1	즐거운 장고	즐거워요...	2021. 03. 23.	2021. 03. 23.	유저1: 저도 즐거워용	유저2: 노잼	유저3: 이놈의 디장고...
3								
4								
5								
6								
7								

컬럼 이름은
고유하게 하나만 만들 것

그러면... 필드를 몇 개 더 추가해서 거기에 저장하자!

게시글 (Article)

	A	B	C	D	E	F
1	pk (INT)	제목 (varchar)	내용 (text)	생성일 (datetime)	수정일 (datetime)	댓글 (text)
2	1	즐거운 장고	즐거워요...	2021. 03. 23.	2021. 03. 23.	유저1: 저도 즐거워용
3	1	즐거운 장고	즐거워요...	2021. 03. 23.	2021. 03. 23.	유저2: 노잼
4	1	즐거운 장고	즐거워요...	2021. 03. 23.	2021. 03. 23.	유저3: 이놈의 디장고...
5						
6						
7						

마지막으로... 그러면 댓글만 바꿔서 여러개를 저장하자!

게시글 (Article)

약속 3.

	A	B	C	D	E	F
1	pk (INT)	제목 (varchar)	내용 (text)	생성일 (datetime)	수정일 (datetime)	댓글 (text)
2	1	즐거운 장고	즐거워요...	2021. 03. 23.	2021. 03. 23.	유저1: 저도 즐거워용
3	1	즐거운 장고	즐거워요...	2021. 03. 23.	2021. 03. 23.	유저2: 노잼
4	1	즐거운 장고	즐거워요...	2021. 03. 23.	2021. 03. 23.	유저3: 이놈의 디장고...
5						
6						
7						

가능하면
레코드 값의 중복은 피할 것

마지막으로... 그러면 댓글만 바뀌서 여러개를 저장하자!

테이블을 쪼개고

관계를 지어주자!

1:N 관계

게시글과 댓글의 관계

하나의 게시글은 여러개의 댓글을 가질 수 있을까?

하나의 댓글은 여러 개의 게시글에 속할 수 있을까?

게시글과 댓글의 관계

하나의 게시글은 여러개의 댓글을 가질 수 있다?

1

N

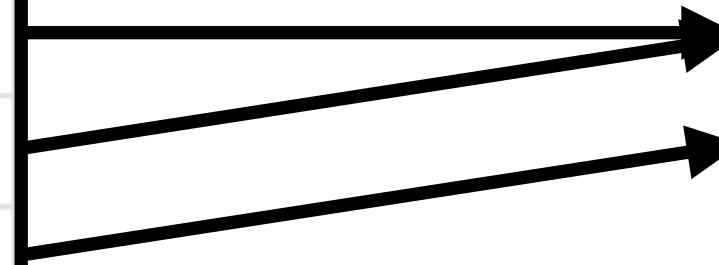
하나의 댓글은 여러 개의 게시글에 속할 수 있을까?

댓글 (N)

	A	B	C
1	pk (int)	댓글 (text)	게시글 pk (int)
2	1	유저1: 저도 즐거워용	1
3	2	유저2: 저는 알고가 더...	1
4	3	유저3: 자바스크립트 좋죠!	2
5			
6			
7			

게시글 (1)

	A	B	C
1	pk (INT)	제목 (varchar)	내용 (text)
2	1	즐거운 장고	즐거워요...
3	2	뷰가 기대돼요	자바스크립트도!
4	3	알고리즘	저는 알고도 좋아요!
5			
6			
7			



하나의 게시글은 여러개의 댓글을 가질 수 있다.

ForeignKey

자식이 부모 테이블을 참조하기 위해 저장한
부모 테이블의 PK를 말한다.

댓글 (N)

	A	B	C
1	pk (int)	댓글 (text)	게시글 pk (int)
2	1	유저1: 저도 즐거워용	1
3	2	유저2: 저는 알고가 더...	1
4	3	유저3: 자바스크립트 좋죠!	2
5			
6			
7			

게시글 (1)

	A	B	C
1	pk (INT)	제목 (varchar)	내용 (text)
2	1	즐거운 장고	즐거워요...
3	2	뷰가 기대돼요	자바스크립트도!
4	3	알고리즘	저는 알고도 좋아요!
5			
6			
7			

ForeignKey (FK)

외래키는 자식 테이블에 추가한다!

ERD (Entity Relationship Diagram)

