

# NoSQL

정제희



SQL

RDBMS



## SQL

diary - Table ×

Table Name:  Schema: **diary**

Filter Rows:  Edit: Export/Import: Wrap Cell Content: Apply changes:

COLUMN

SCHEMA

ROW

	NUM	TIME	TITLE	CONTENTS	STEP2
	1	2017-04-27	3_3 코딩 더럽게 힘들다...	근데 난 이걸 왜 하고 앉았냐...	237
	2	2017-05-01	3x3	수요일부터 휴일이라 뭔가 실험하기...	236
	3	2017-05-01	고해 갔다음 ㅇㅇ	헐 존맛 ㅇㅇ	235
	4	2017-05-02	와 내일 쉰다	와 와  예이	234
	5	2017-05-02	뭐지 이 프로그램이 안 돌아가	근데 내가 이런 아난 것 같다고 말했...	233
	6	2017-05-02	음아아앙아아아아아아아아	기지개 한 번 피고!  gold chip에 ...	232
	7	2017-05-08	아 피곤타	휴일모드 들어가니까 아주 잉여롭게 ...	231
	8	2017-05-08	아	오늘 팀장님 생신이시다  ㅇㅇ  <...>	230
	9	2017-05-08	점심먹었는데추네	점심 르구떼데추네 ㅇㅇ  존맛...	229
	10	2017-05-10	오늘도 출근이니라	ㅇㅇ 아침에 뉴스보니까 문재인이 당...	228
	11	2017-05-10	아침부터 할일 투성이	근데 나 왜 이룩린스만 잔고 있지	???

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

출처: okky

## SQL JOIN



[그림 II-1-13] EQUI JOIN을 설명하기 위한 선수-팀 테이블 관계도

## SQL JOIN

선수(PLAYER) 테이블			팀(Team) 테이블		
PLAYER-NAME	BACK_NO	TEAM_ID	TEAM_ID	TEAM NAME	REGION NAME
이 고 르	21	K06	K05	현대모터스	전부
오 비 나	26	K10	K08	일화천마	성남
윤 원 일	45	K02	K03	스틸러스	포항
페 르 난 도	44	K04	K07	드래곤즈	전남
레 오	45	K03	K09	FC서울	서울
실 바	45	K07	K04	유나이티드	인천
무 스 타 파	77	K04	K11	경남FC	경남
에 디	7	K01	K01	울산현대	울산
알리송	14	K014	K10	시티즌	대전
차스민	33	K08	K02	삼성블루윙즈	수우 js
디디	8	K06	K12	광주상무	광주
⋮	⋮	⋮	K06	아이파크	부산
			K13	강원FC	강원
			K14	제주유나이티드FC	제주
			K15	대구FC	대구

[그림 II-1-13] EQUI JOIN을 설명하기 위한 선수-팀 테이블 관계도

선수(PLAYER) 테이블			팀(Team) 테이블		
PLAYER-NAME	BACK_NO	TEAM_ID	TEAM_ID	TEAM NAME	REGION NAME
이 고 르	21	K06	K06	아이파크	부산
오 비 나	26	K10	K10	시티즌	대전
윤 원 일	45	K02	K02	삼성블루윙즈	수원
페 르 난 도	44	K04	K04	유나이티드	인천
레 오	45	K03	K03	스틸러스	포항
실 바	45	K07	K07	드래곤즈	전남
무 스 타 파	77	K04	K04	유나이티드	인천
에 디	7	K01	K01	울산현대	울산
알리송	14	K014	K01	울산현대	울산
차스민	33	K08	K08	일화천마	성남
디디	8	K06	K06	아이파크	부산
⋮	⋮	⋮	⋮	⋮	⋮

[그림 II-1-14] EQUI JOIN을 설명하기 위한 데이터 재배열 후

## SQL JOIN

선수(PLAYER) 테이블			팀(Team) 테이블		
PLAYER-NAME	BACK_NO	TEAM_ID	TEAM_ID	TEAM NAME	REGION NAME
이 고 르	21	K06	K05	현대모터스	전부
오 비 나	26	K10	K08	일화천마	성남
윤 원 일	45	K02	K03	스틸러스	포항
페 르 난 도	44	K04	K07	드래곤즈	전남
레 오	45	K03	K09	FC서울	서울
실 바	45	K07	K04	유나이티드	인천
무 스 타 파	77	K04	K11	경남FC	경남
에 디	7	K01	K01	울산현대	울산
알리송	14	K014	K10	시티즌	대전
자스민	33	K08	K02	삼성블루윙즈	수원
디디	8	K06	K12	광주상무	광주
⋮	⋮	⋮	K06	아이파크	부산
			K13	강원FC	강원
			K14	제주유나이티드FC	제주
			K15	대구FC	대구

[그림 II-1-13] EQUI JOIN을 설명하기 위한 선수-팀 테이블 관계도

1. Inner Join
2. Left Outer Join
3. Right Outer Join
4. Full Outer Join

선수(PLAYER) 테이블			팀(Team) 테이블		
PLAYER-NAME	BACK_NO	TEAM_ID	TEAM_ID	TEAM NAME	REGION NAME
이 고 르	21	K06	K06	아이파크	부산
오 비 나	26	K10	K10	시티즌	대전
윤 원 일	45	K02	K02	삼성블루윙즈	수원
페 르 난 도	44	K04	K04	유나이티드	인천
레 오	45	K03	K03	스틸러스	포항
실 바	45	K07	K07	드래곤즈	전남
무 스 타 파	77	K04	K04	유나이티드	인천
에 디	7	K01	K01	울산현대	울산
알리송	14	K014	K01	울산현대	울산
자스민	33	K08	K08	일화천마	성남
디디	8	K06	K06	아이파크	부산
⋮	⋮	⋮	⋮	⋮	⋮

[그림 II-1-14] EQUI JOIN을 설명하기 위한 데이터 재배열 후

NoSQL  
?

NOT ONLY SQL



한 가지를 선택한다기 보다는 상호보완적 존재  
그때 그때 필요에 따라 선택

## 종류

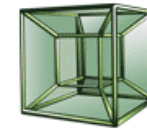
1. Key-Value DB
2. Wide Columnar DB
3. KDocument DB
4. Graph DB

APACHE  
HBASE

 Cassandra

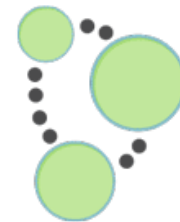
  
CouchDB  
relax

 riak



mongoDB

HYPERTABLE<sup>INC</sup>



Neo4j



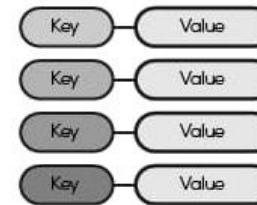
redis

## 1. Key-Value DB

Key와 Value의 쌍으로 데이터가 저장되는 유형  
Amazon의 Dynamo Paper에서 유래되었다. Riak,  
Voldemort, Tokyo 등의 제품이 알려져 있다.

읽기와 쓰기에 최적화

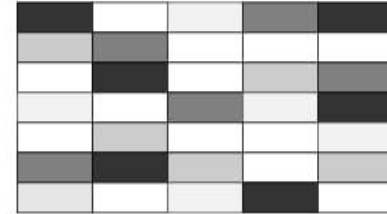
### Key-Value



## 2. Wide Columnar DB

Big Table DB라고도 하며, Google의 BigTable Paper에서 유래되었다. Column Family 데이터 모델을 사용하고 있고, HBase, Cassandra, Hypertable이 이에 해당된다.

Wide-Column



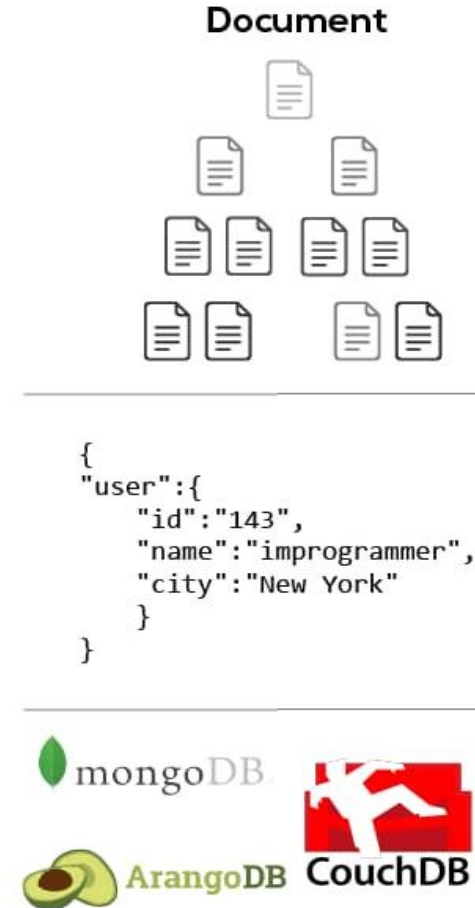
1	Fruit	A Foo	B Baz	
2	City	E DC	D PLA	G FLD
3	State	A NZ	C CL	



SCYLLA

### 3. KDocument DB

Lotus Notes에서 유래  
JSON, XML과 같은 Collection 데이터 모델 구조를 채택하고 있다. Mongo DB, Couch DB가 이 종류에 해당된다.

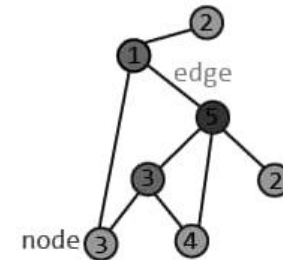
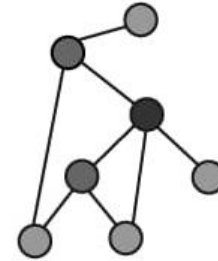


## 4. Graph DB

Euler & Graph Theory에서 유래한 DB이며, Nodes, Relationship, Key-Value 데이터 모델을 채용하고 있다. Neo4J 등의 제품이 있다.

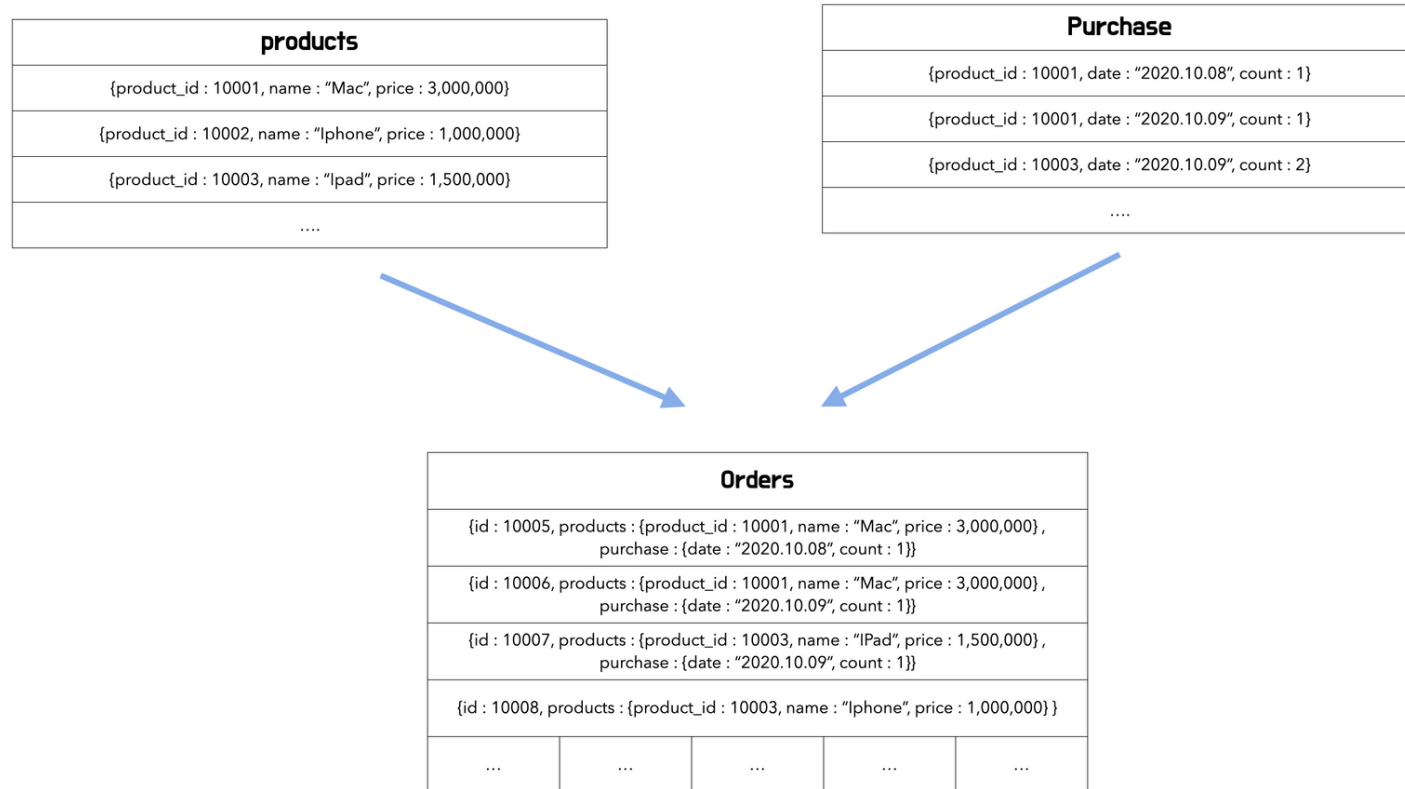
페이스북

Graph

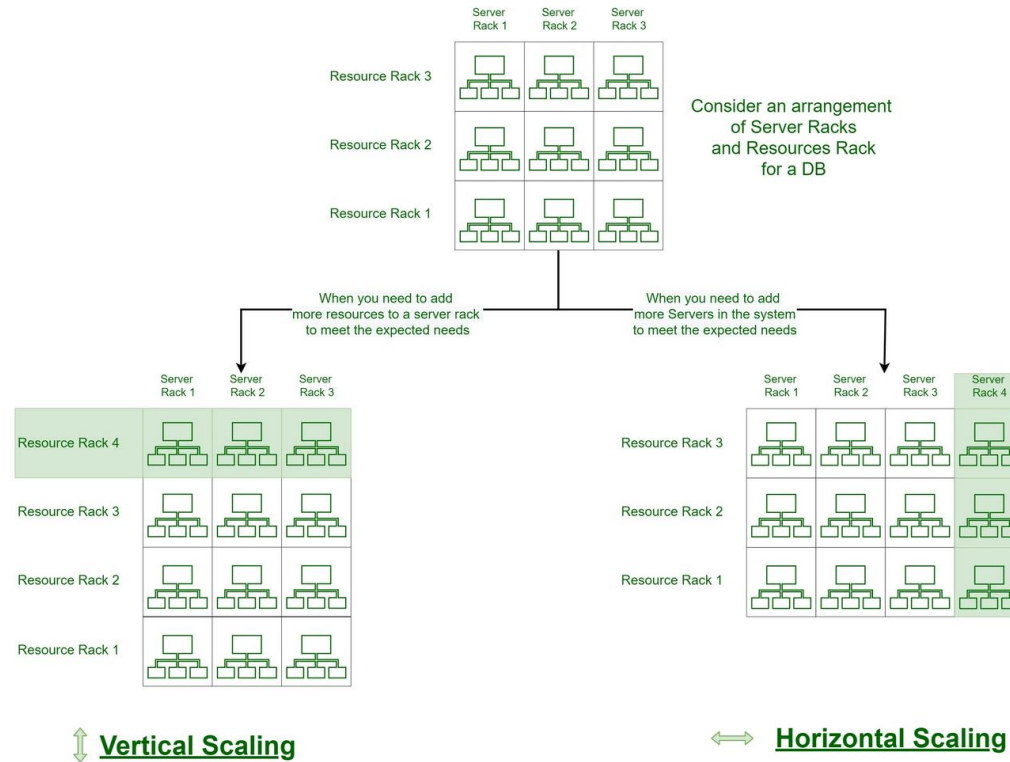


## NoSQL

스키마와 관계가 없고 조인이 필요없다.



## Horizontal and Vertical Scaling In Databases



언제 사용하는것이 좋을까?

- SQL

관계를 맺고 있는 데이터가 자주 변경(수정)되는 애플리케이션일 경우

(NoSQL에서라면 여러 컬렉션을 모두 수정해주어야 함.)

변경될 여지가 없고, 명확한 스키마가 사용자와 데이터에게 중요한 경우

- NoSQL의 경우

정확한 데이터 구조를 알 수 없거나 변경 / 확장 될 수 있는 경우

읽기(read)처리를 자주하지만, 데이터를 자주 변경(update)하지 않는 경우 (즉, 한번의 변경으로 수십 개의 문서를 업데이트 할 필요가 없는 경우)

데이터베이스를 수평으로 확장해야 하는 경우 (막대한 양의 데이터를 다뤄야 하는 경우)