

Database

Partitioning / Sharding

파티셔닝 - Partitioning

파티셔닝이란?

파티셔닝

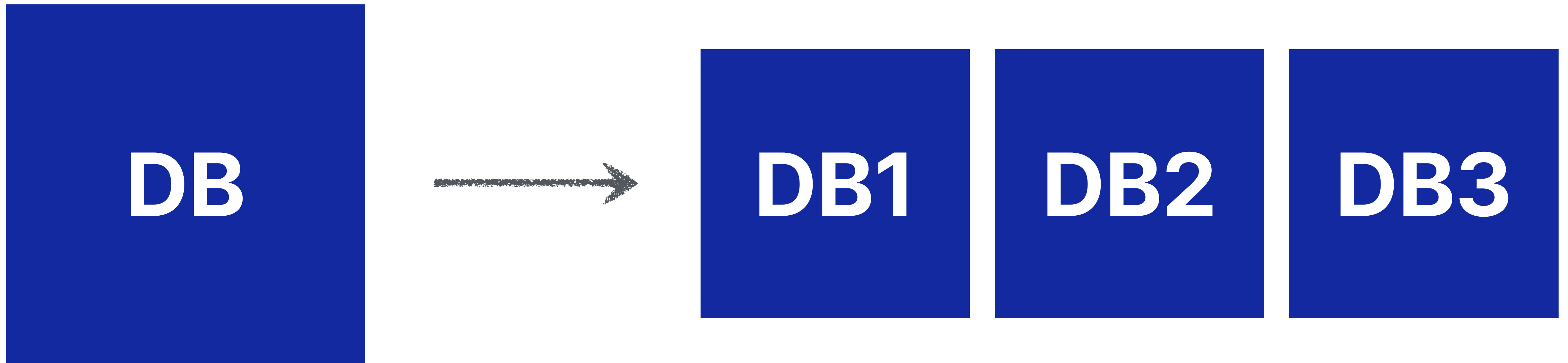
데이터베이스를
여러 부분으로 분할 하는 것



데이터베이스 테이블을
더 작은 테이블들로 나누는 것

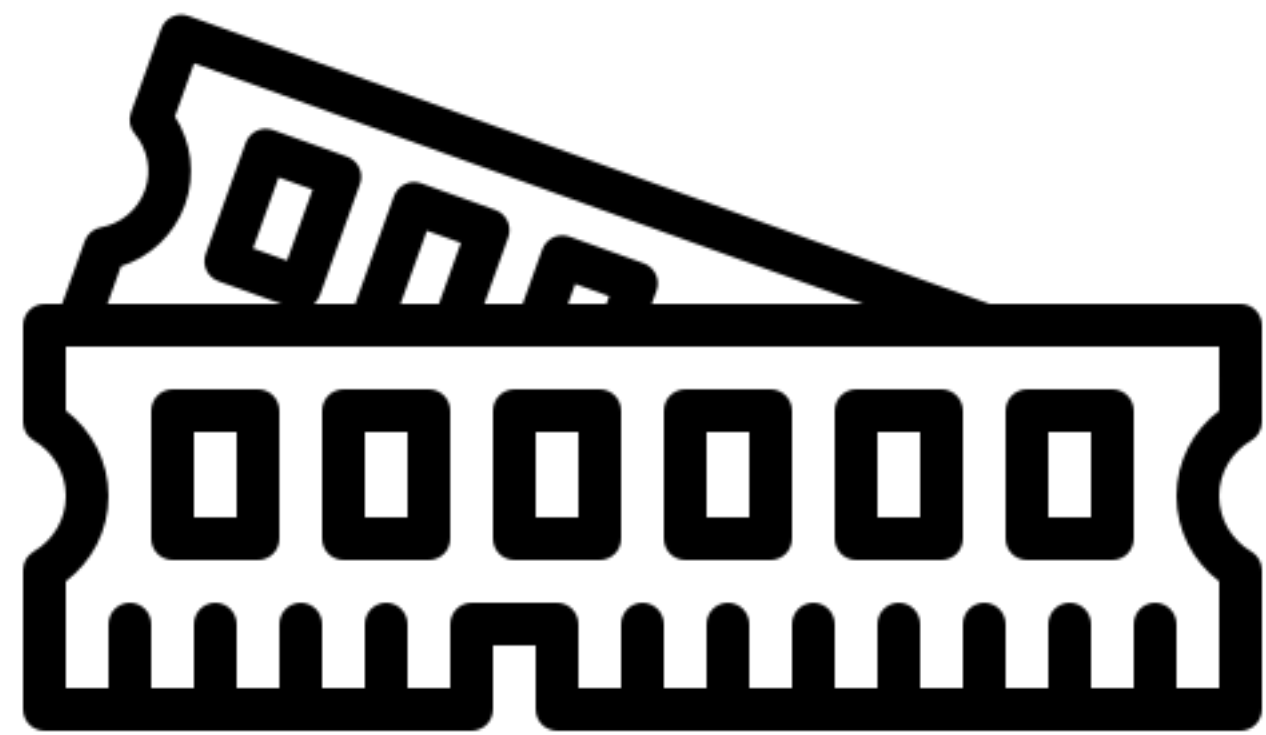
출처: 위키백과

파티셔닝이란?

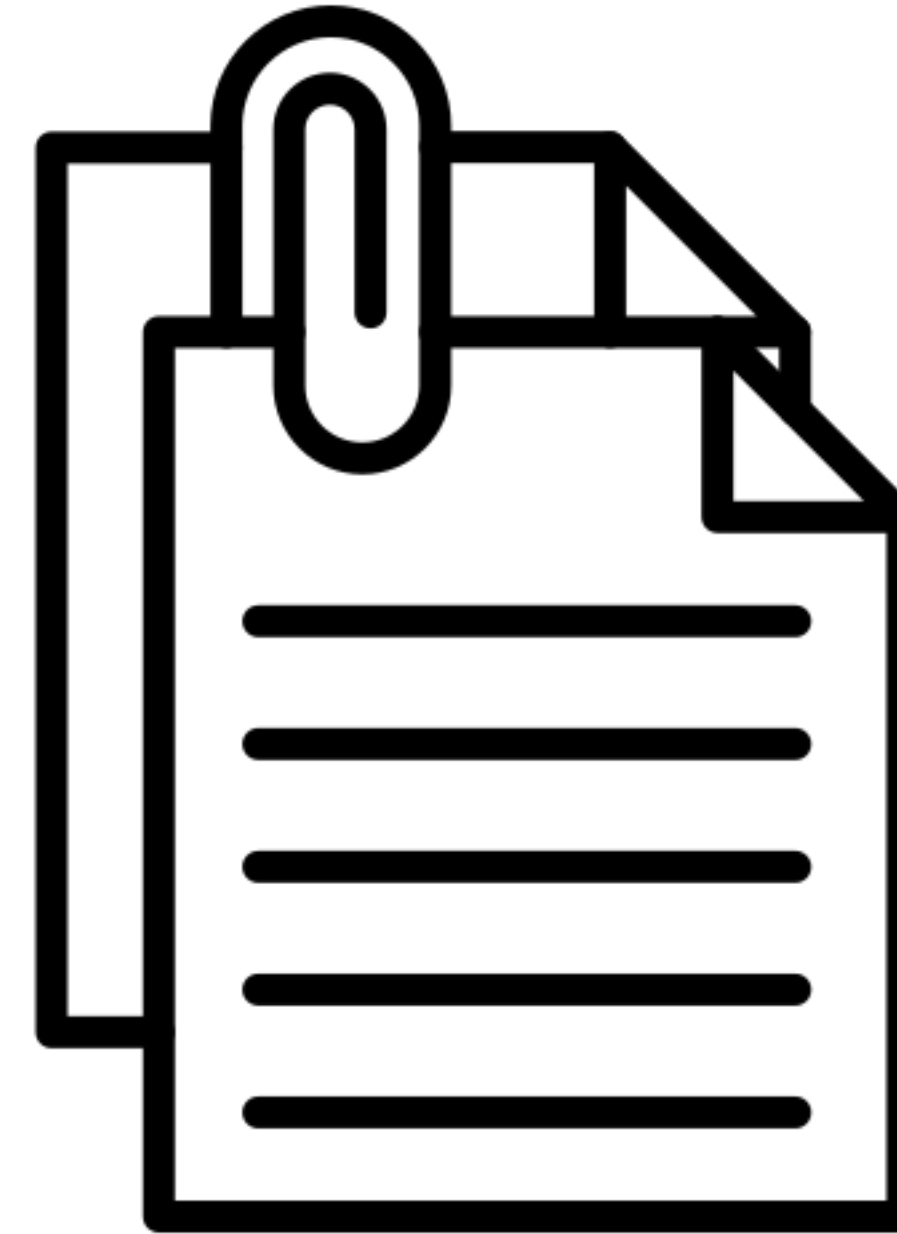
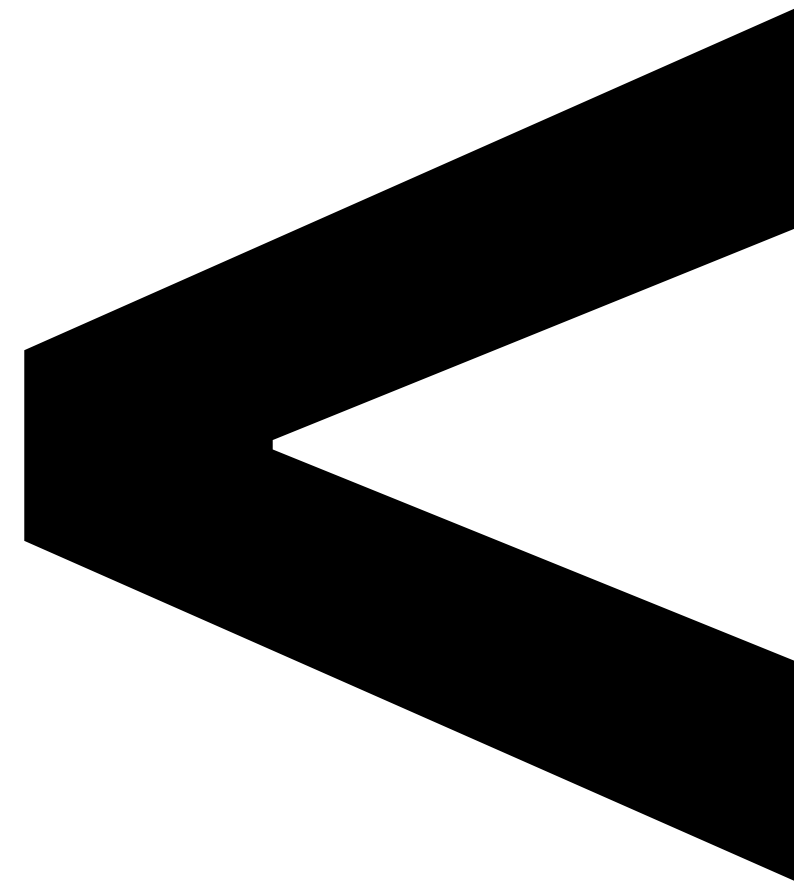


하나의 논리적인 테이블을 여러개의 물리적인 테이블로 나누는 것

사용하는 이유



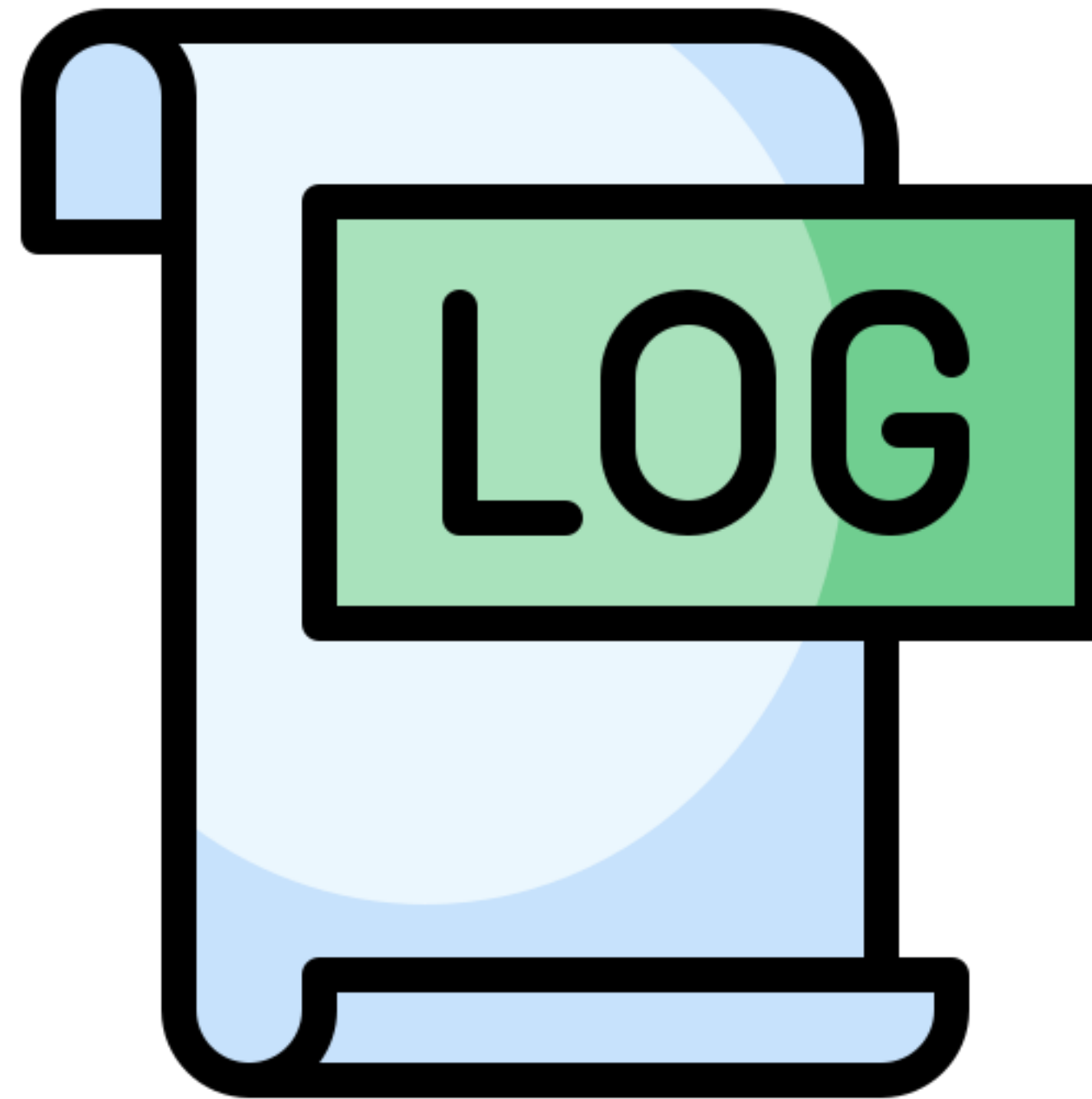
Memory



Index

인덱스 파일을 메모리에 한번에 올릴 수 없을때 물리적으로 나눈다면 효율적

사용하는 이유



파티셔닝 목적

[성능(Performance)]

- 특정 DML과 Query의 성능을 향상시킨다.
- 주로 대용량 Data WRITE 환경에서 효율적이다.
- 특히, Full Scan에서 데이터 Access의 범위를 줄여 성능 향상을 가져온다.
- 많은 INSERT가 있는 OLTP 시스템에서 INSERT 작업을 작은 단위인 partition들로 분산시켜 경합을 줄인다.

[가용성(Availability)]

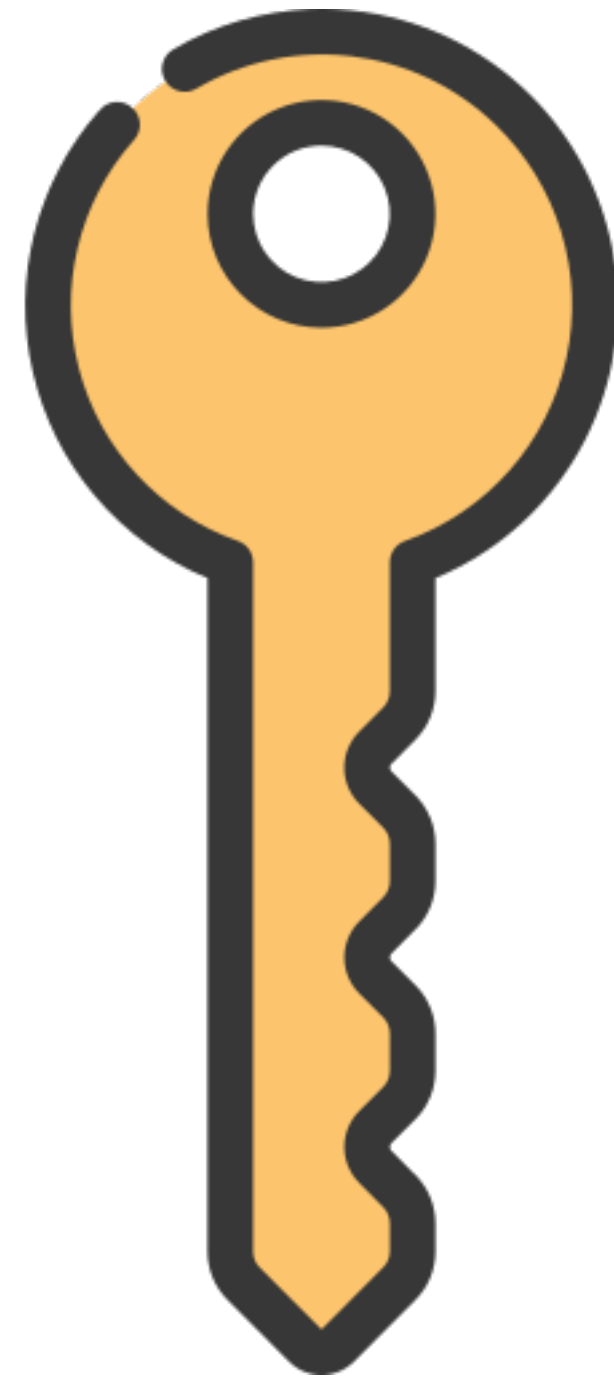
- 물리적인 파티셔닝으로 인해 전체 데이터의 훼손 가능성이 줄어들고 데이터 가용성이 향상된다.
각 분할 영역(partition별로)을 독립적으로 백업하고 복구할 수 있다.
table의 partition 단위로 Disk I/O를 분산하여 경합을 줄이기 때문에 UPDATE 성능을 향상시킨다.

[관리용이성(Manageability)]

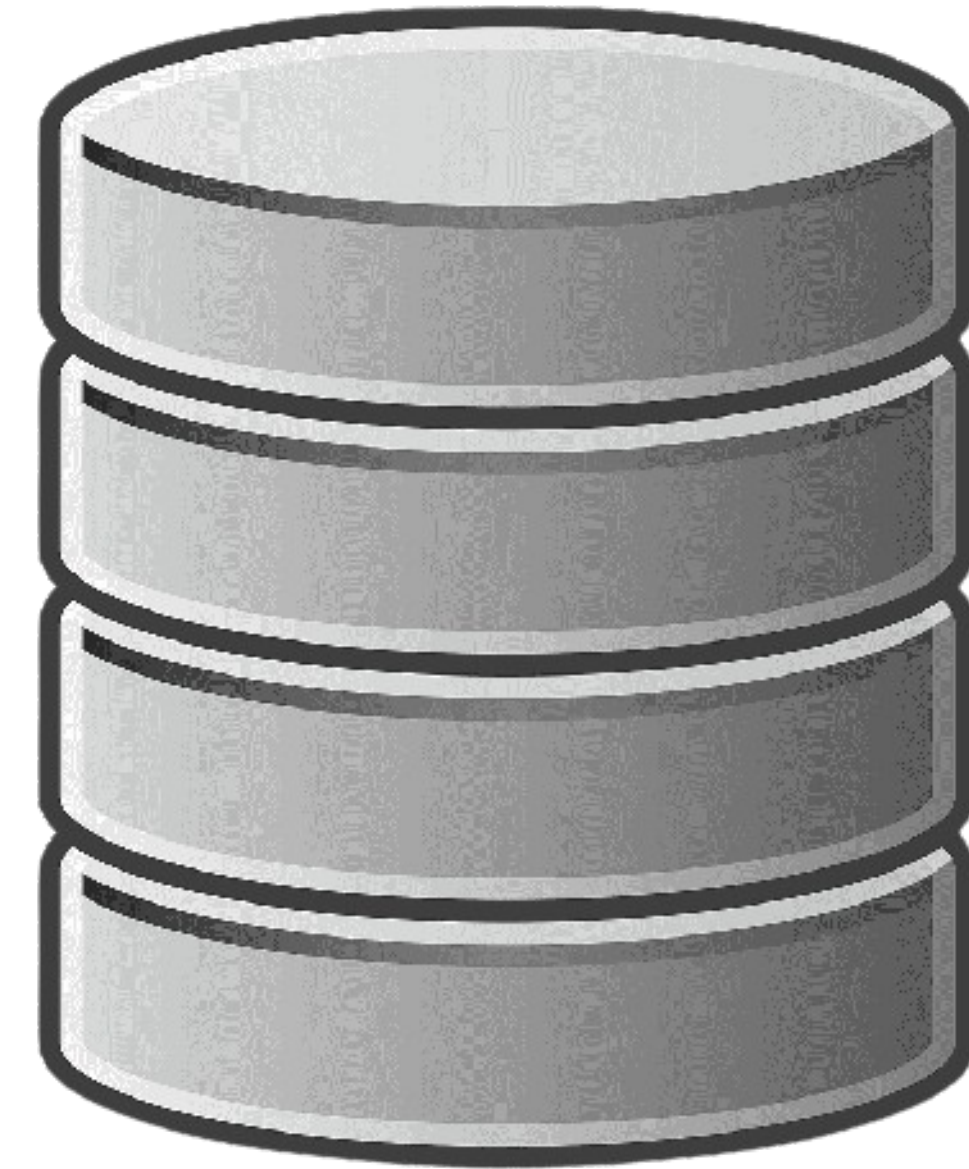
- 큰 table들을 제거하여 관리를 쉽게 해준다.

MySQL에서의 처리 방식

INSERT 작업



1. 파티션 표현식 평가



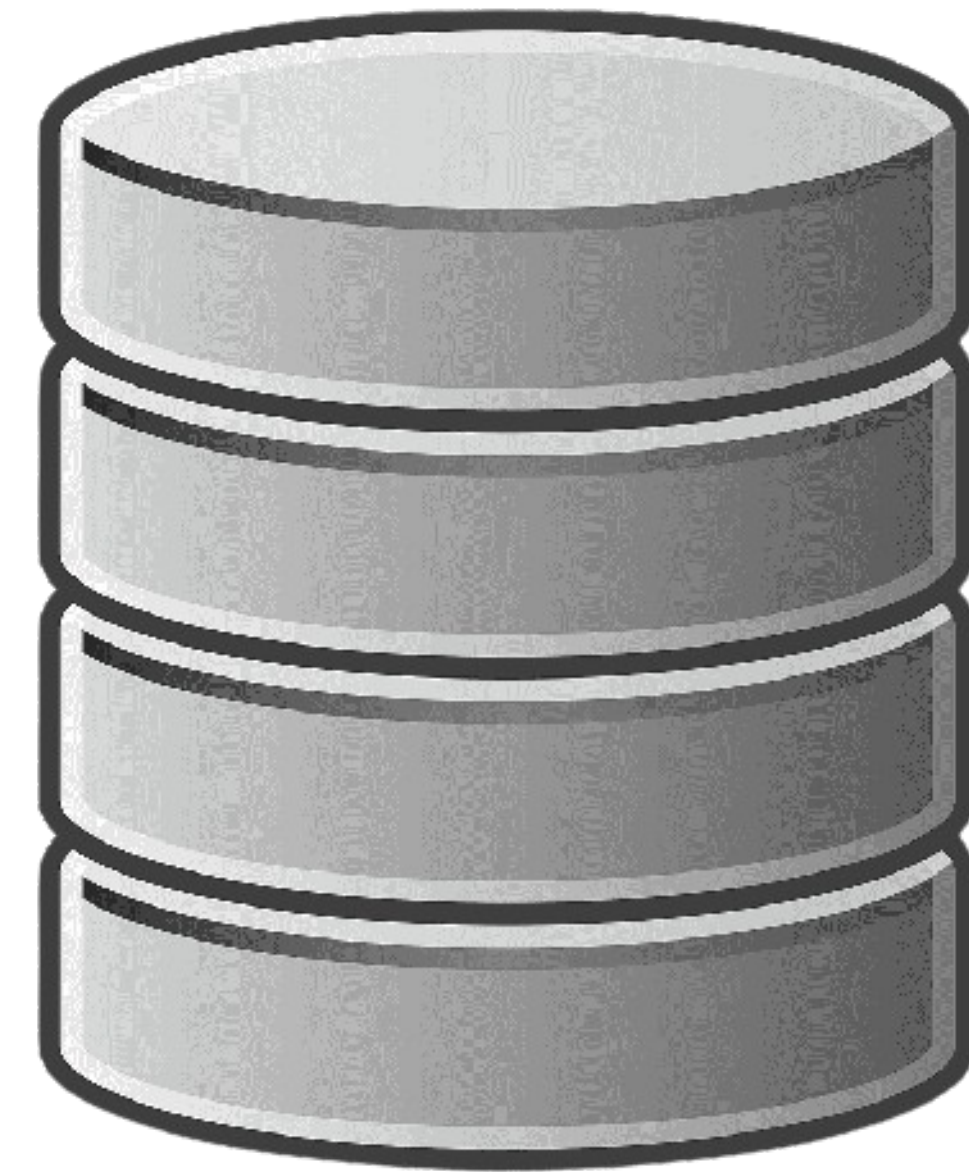
2. 저장

MySQL에서의 처리 방식

UPDATE 작업 - 파티션 키 이외의 컬럼이 변경



1. 파티션 표현식 평가



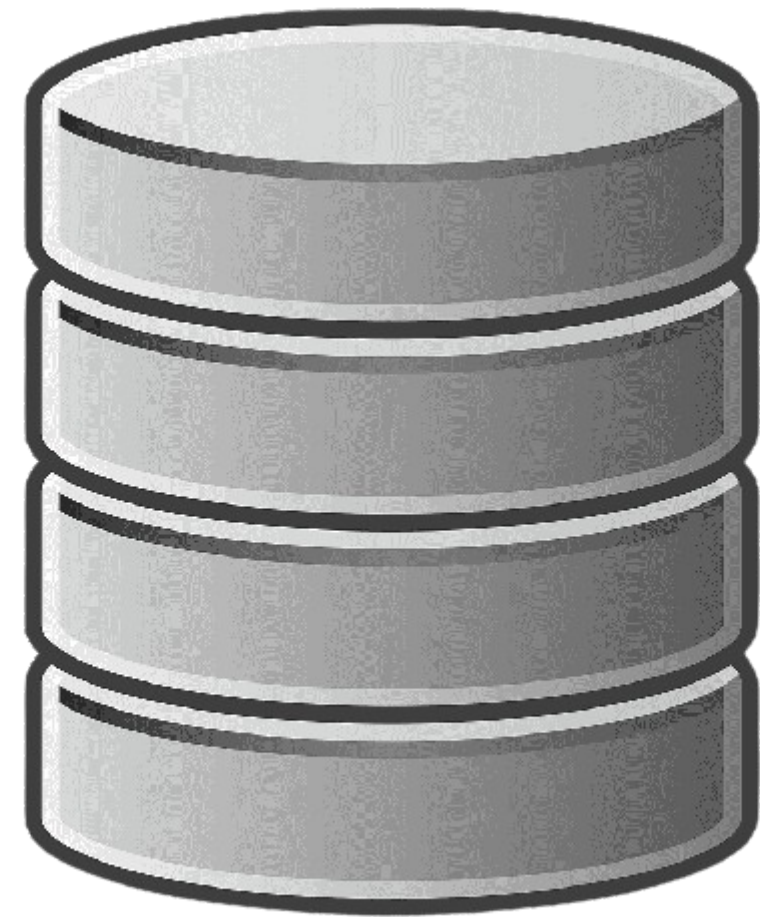
2. 수정

MySQL에서의 처리 방식

UPDATE 작업 - 파티션 키 컬럼이 변경



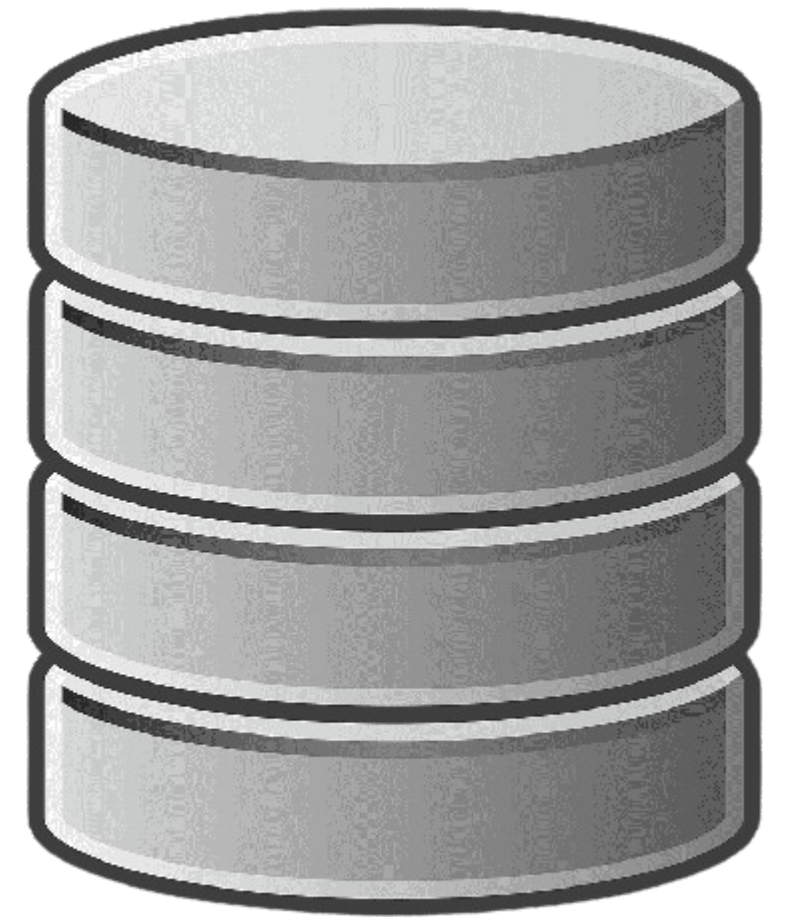
1. 파티션 표현식 평가



2. 삭제



3. 새 표현식 평가



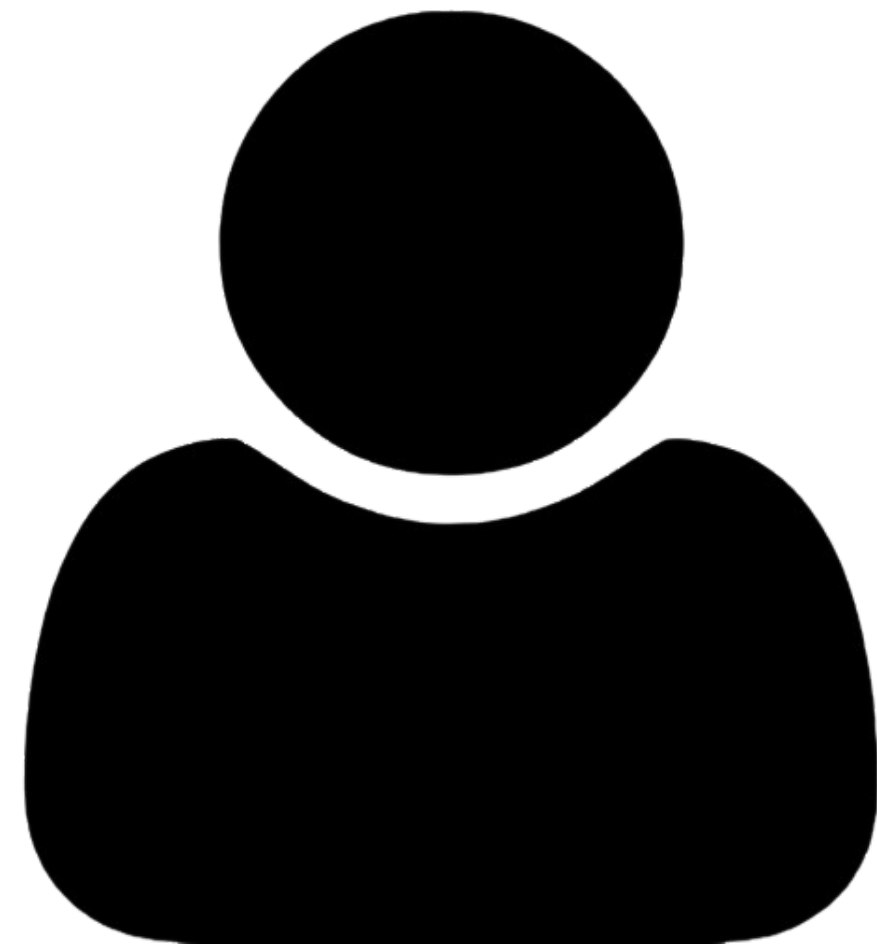
4. 저장

MySQL에서의 처리 방식

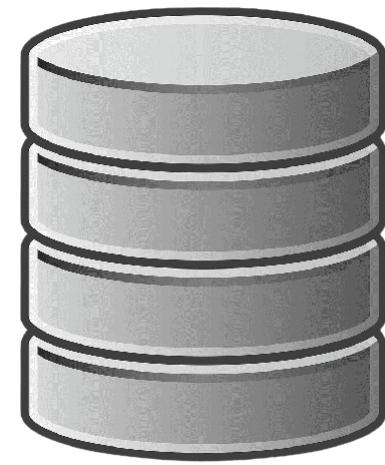


1. WHERE 절에서 파티션을 결정할 수 있는가?
2. WHERE 절에서 인덱스를 효율적으로 사용할 수 있는가?

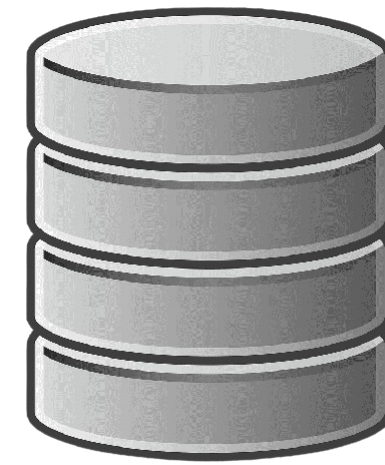
MySQL에서의 처리 방식



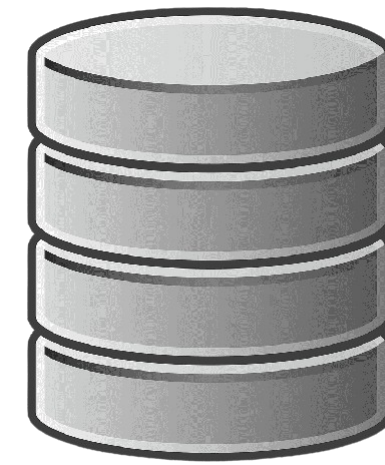
이름 (Index)
출생년도
...



~ 1997



1998



1999 ~

```
SELECT name FROM member WHERE year = 1998;
```

```
SELECT name FROM member;
```

```
SELECT * FROM member WHERE year = 1998;
```

```
SELECT * FROM member;
```

제약 사항 및 주의 사항

파티션 표현식은 기본 산술 연산자[+, -, * 등]와 MySQL 내장 함수만 쓸 수 있다.

PK와 UK들은 모두 파티션 키 컬럼을 포함해야 한다.

파티션의 최대 갯수는 8192개이다.

`sql_mode`를 변경하면 파티션의 일관성이 깨질 수 있다.

외래키를 사용할 수 없다.

Full Text Search를 적용할 수 없다.

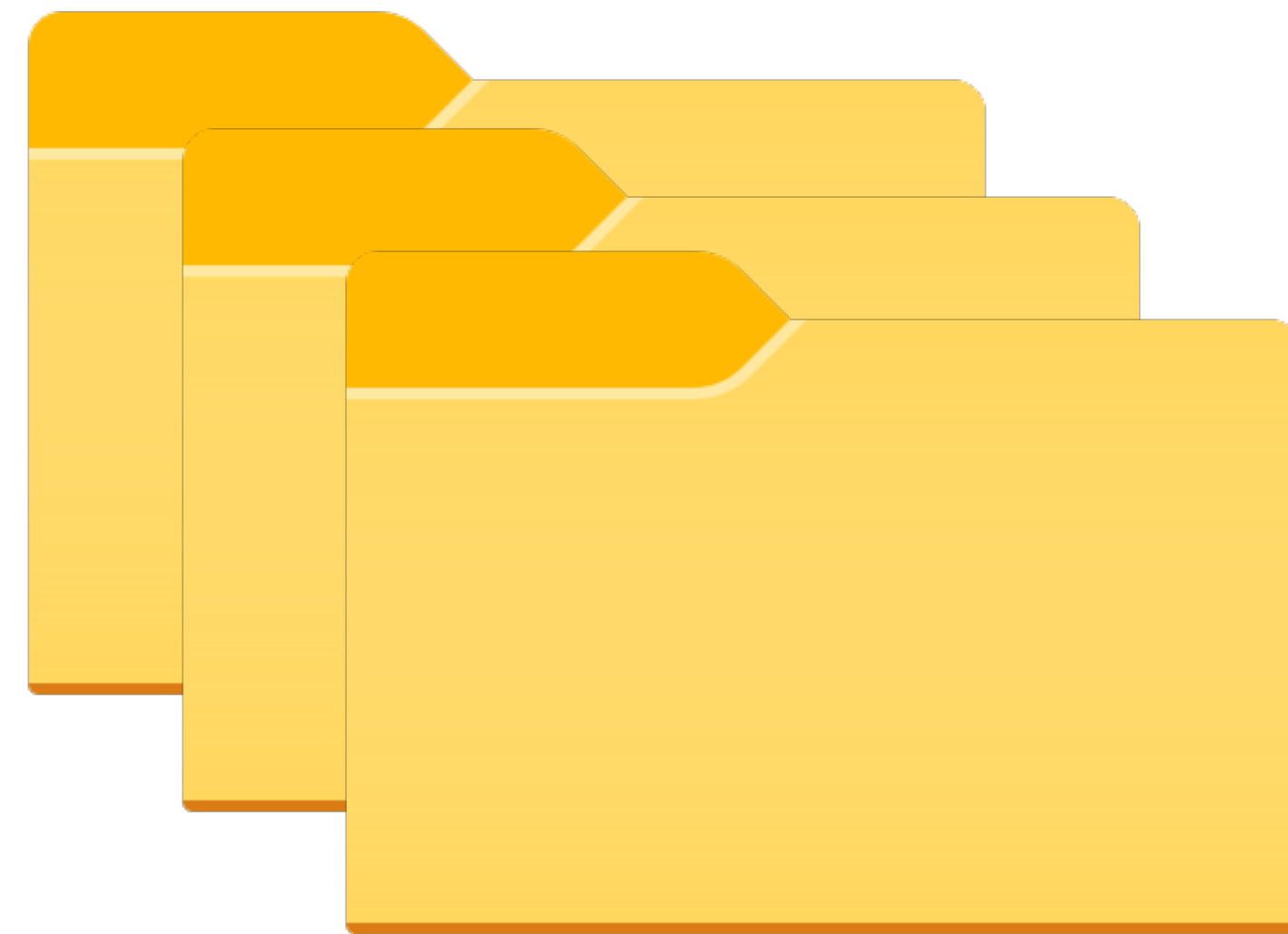
공간 데이터 컬럼을 사용할 수 없다.

Temporary Table은 파티션 불가능하다.

제약 사항 및 주의 사항



MySQL

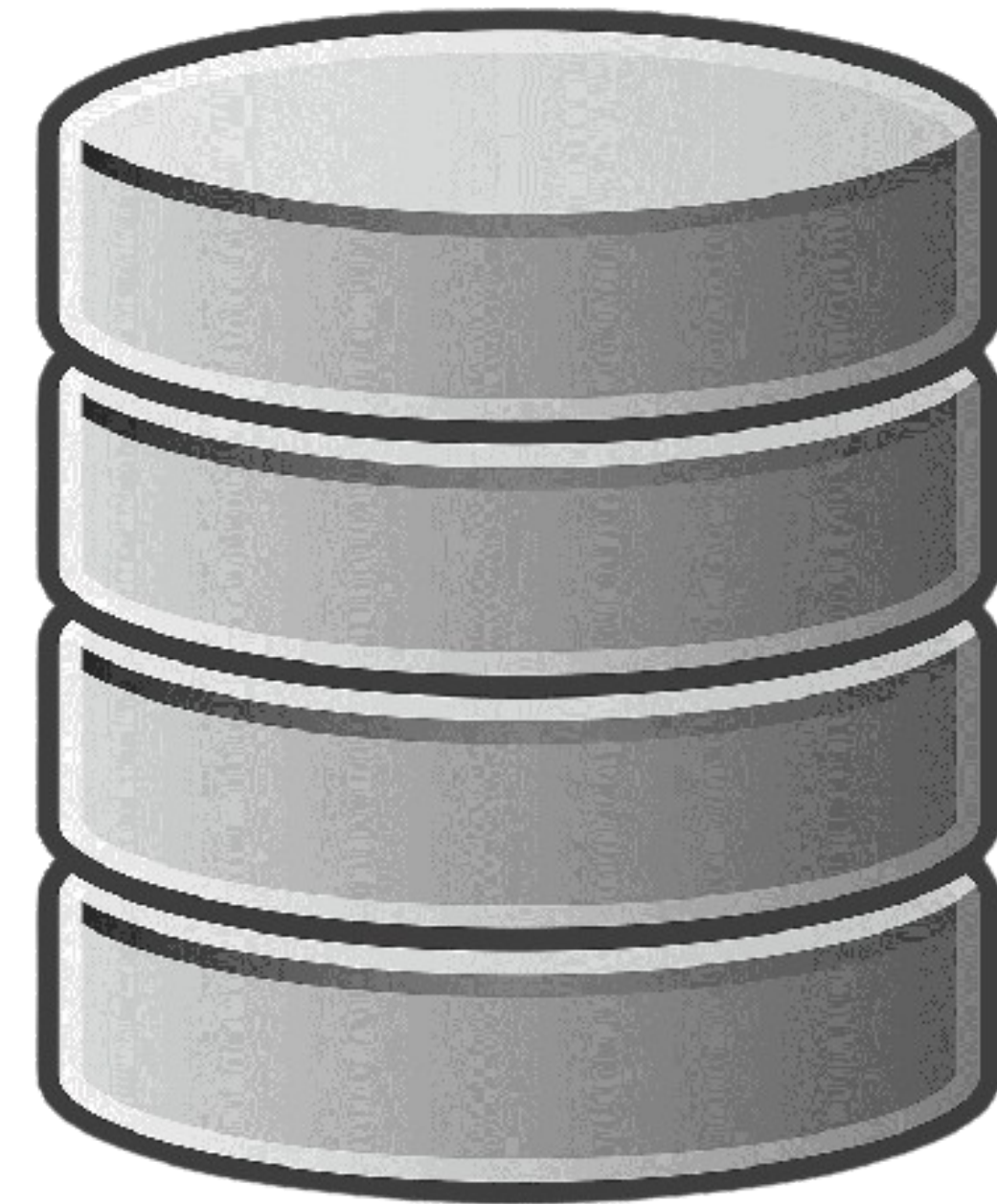
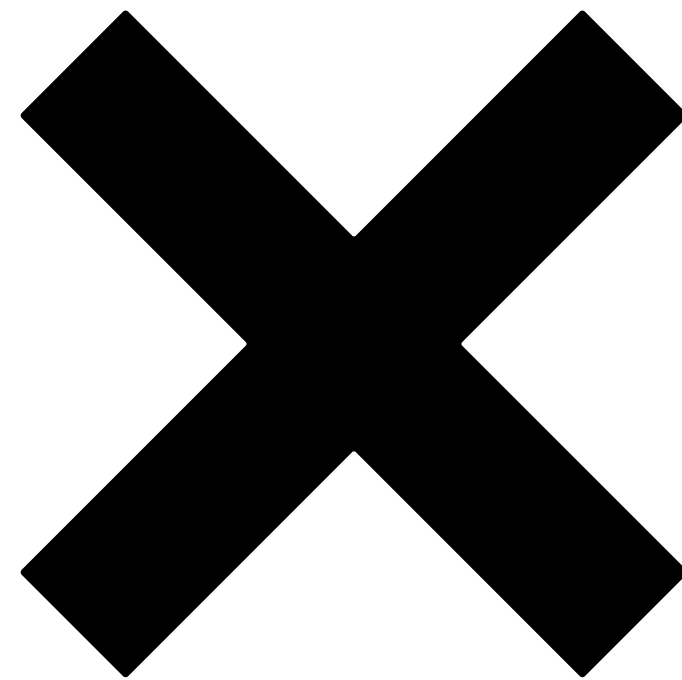


2 ~ 3 Files Per Table

제약 사항 및 주의 사항



2 ~ 3 Files Per Table



1024 Partioned

파티셔닝 종류

수직 파티셔닝
Vertical Partitioning

Column을 기준으로
Table을 나누는 방식

수평 파티셔닝
Horizontal Partitioning

Row를 기준으로
Table을 나누는 방식

MySQL 파티션 종류

레인지 파티션

리스트 파티션

해시 파티션

키 파티션

MySQL 파티션 종류

레인지 파티션

```
CREATE TABLE member
(
    id          BIGINT PRIMARY KEY,
    name        VARCHAR(20),
    year_of_birth DATE
) PARTITION BY RANGE (YEAR(year_of_birth)) (
    PARTITION p0 VALUES LESS THAN (1998),
    PARTITION p0 VALUES LESS THAN (1999),
    PARTITION p0 VALUES LESS THAN MAXVALUE
)
```

사용처

시간 개념 단위로 분석 or 삭제
범위 기반으로 균등한 파티션

파티션 재구성

파티션의 레코드가 많을 수록 오래 걸린다
최소 읽기 잠금은 필요하다

파티션 추가 및 삭제

가장 마지막 파티션만 추가할 수 있다
삭제는 매우 빠르다

MySQL 파티션 종류

리스트 파티션

```
MySQL

CREATE TABLE item
(
    id          BIGINT PRIMARY KEY,
    name        VARCHAR(20),
    category    VARCHAR(20)
) PARTITION BY LIST (category) (
    PARTITION p_computer VALUES IN ('Notebook', 'Desktop'),
    PARTITION p_sports VALUES IN ('Tennis', NULL)
);
```

레인지 파티션과 대부분 흡사

키 값 하나하나를 리스트로 나열해야 한다
`MAXVALUE`를 사용할 수 없다

사용처

파티션 키 값이 카테고리처럼 고정적
키 값이 연속되지 않고 정렬 순서와 관계 없다

MySQL 파티션 종류

해시 파티션

```
MySQL

CREATE TABLE member
(
    id          BIGINT PRIMARY KEY,
    name        VARCHAR(20),
    year_of_birth DATE
) PARTITION BY HASH ( id ) PARTITIONS 4;
```

레인지 파티션과 대부분 흡사

키 값 하나하나를 리스트로 나열해야 한다
`MAXVALUE`를 사용할 수 없다

사용처

파티션 키 값이 카테고리처럼 고정적
키 값이 연속되지 않고 정렬 순서와 관계 없다

MySQL 파티션 종류

키 파티션

```
CREATE TABLE member
(
    id          BIGINT PRIMARY KEY,
    name        VARCHAR(20),
    year_of_birth DATE
) PARTITION BY KEY ( ) PARTITIONS 4;
```

해시 파티션과 대부분 흡사
주어진 컬럼들 MD5 해싱
대부분의 데이터 타입 사용 가능

예시 및 정리



보통 1 ~ 3개의 테이블만 접근



예시 및 정리



모두 균등하게 접근



**MySQL을 물리적으로 여러개를
사용하는 것이 효율적**

샤딩 - Sharding

샤딩이란?

데이터베이스 샤드

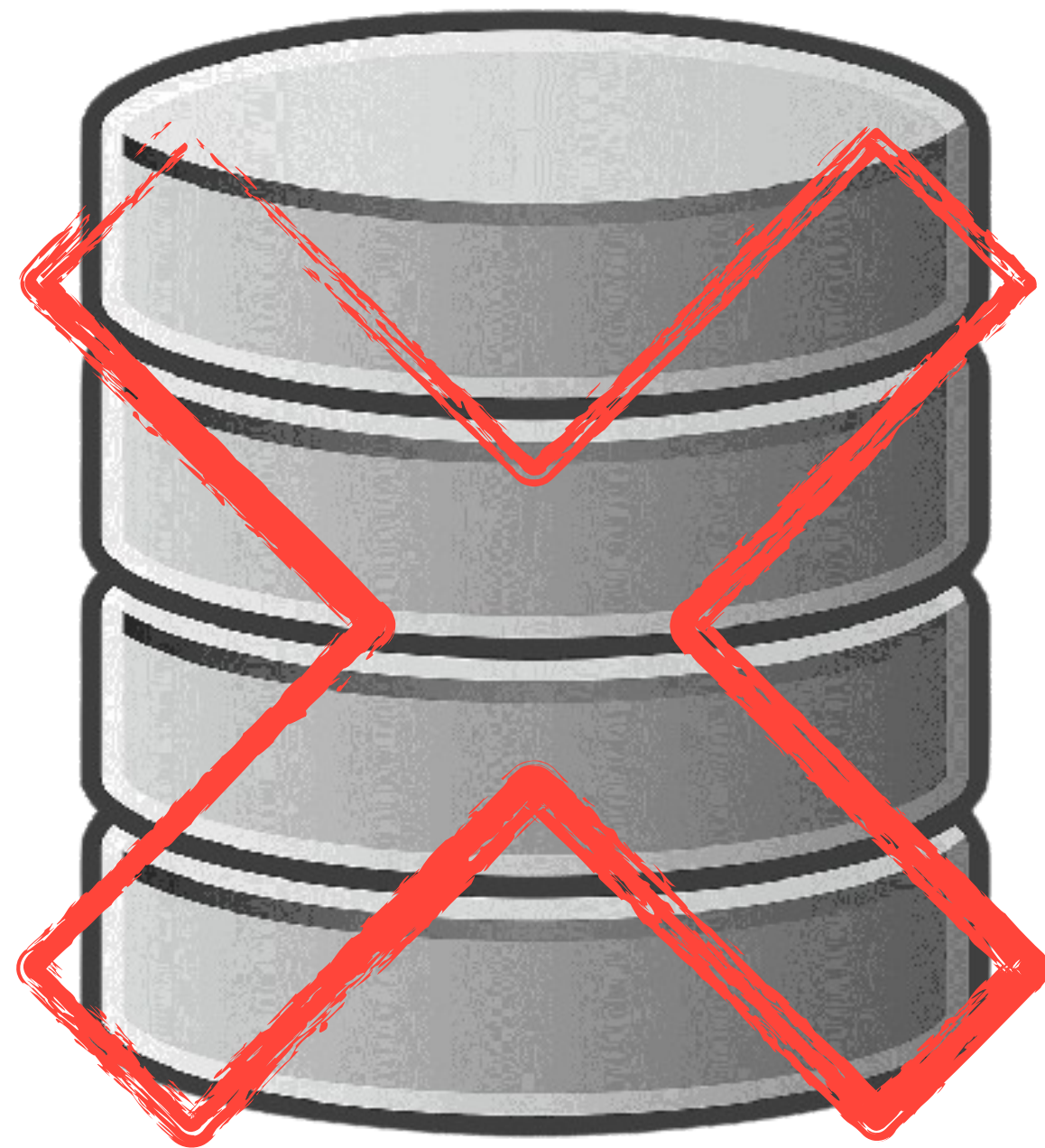
데이터베이스나 웹 검색 엔진의
데이터의 수평 분할이다.



동일한 스키마를 가지고 있는 여러대의
데이터베이스 서버들에 데이터를
작은 단위로 나누어 분산 저장하는 기법

이때, 작은 단위를 샤드(shard)라고 부른다

샤딩이란?



Database

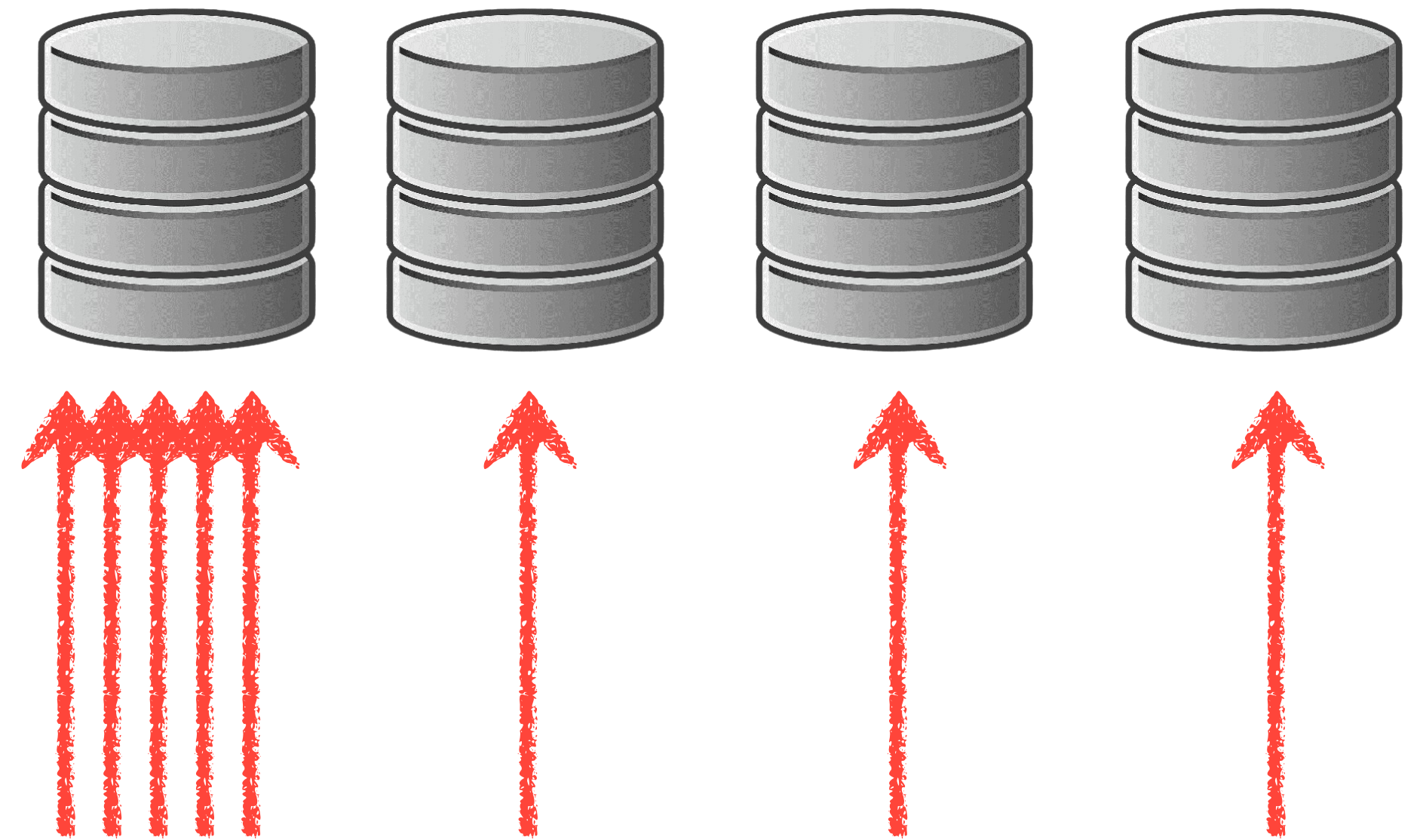


Application Level

주의점



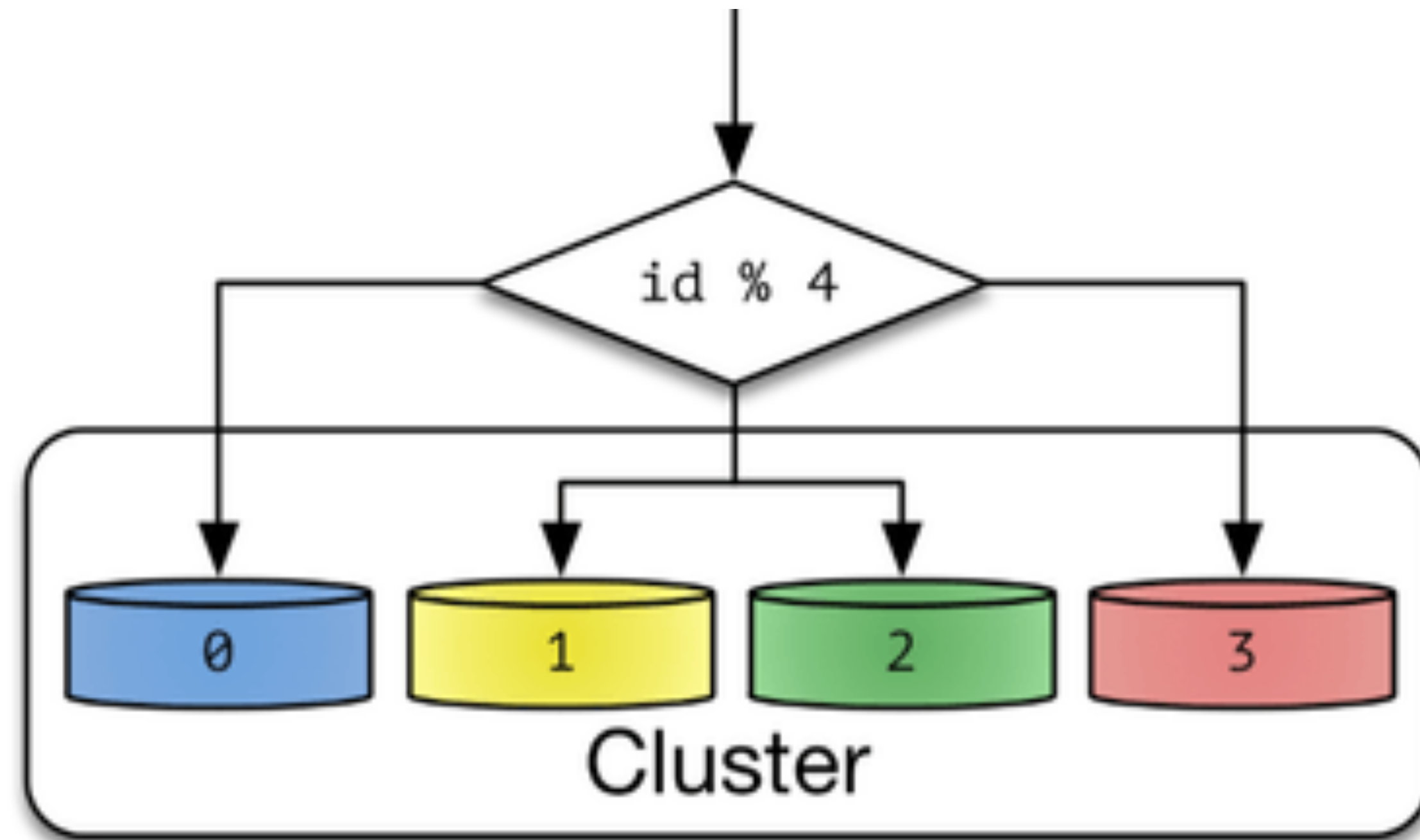
JOIN



INSERT

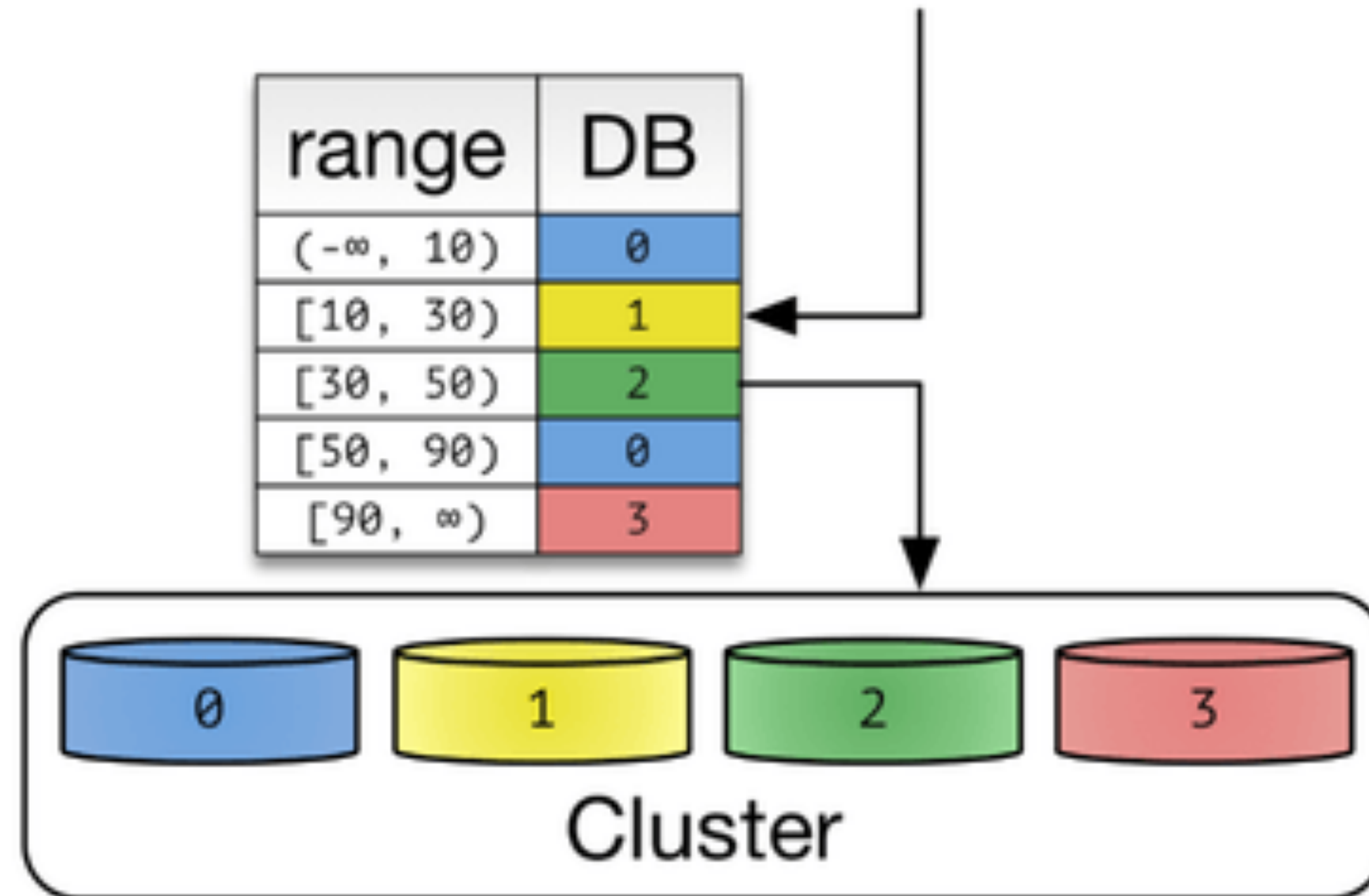
샤딩 종류

해쉬 샤딩 Hash Sharding



샤딩 종류

레인지 샤딩 Range Sharding



Review

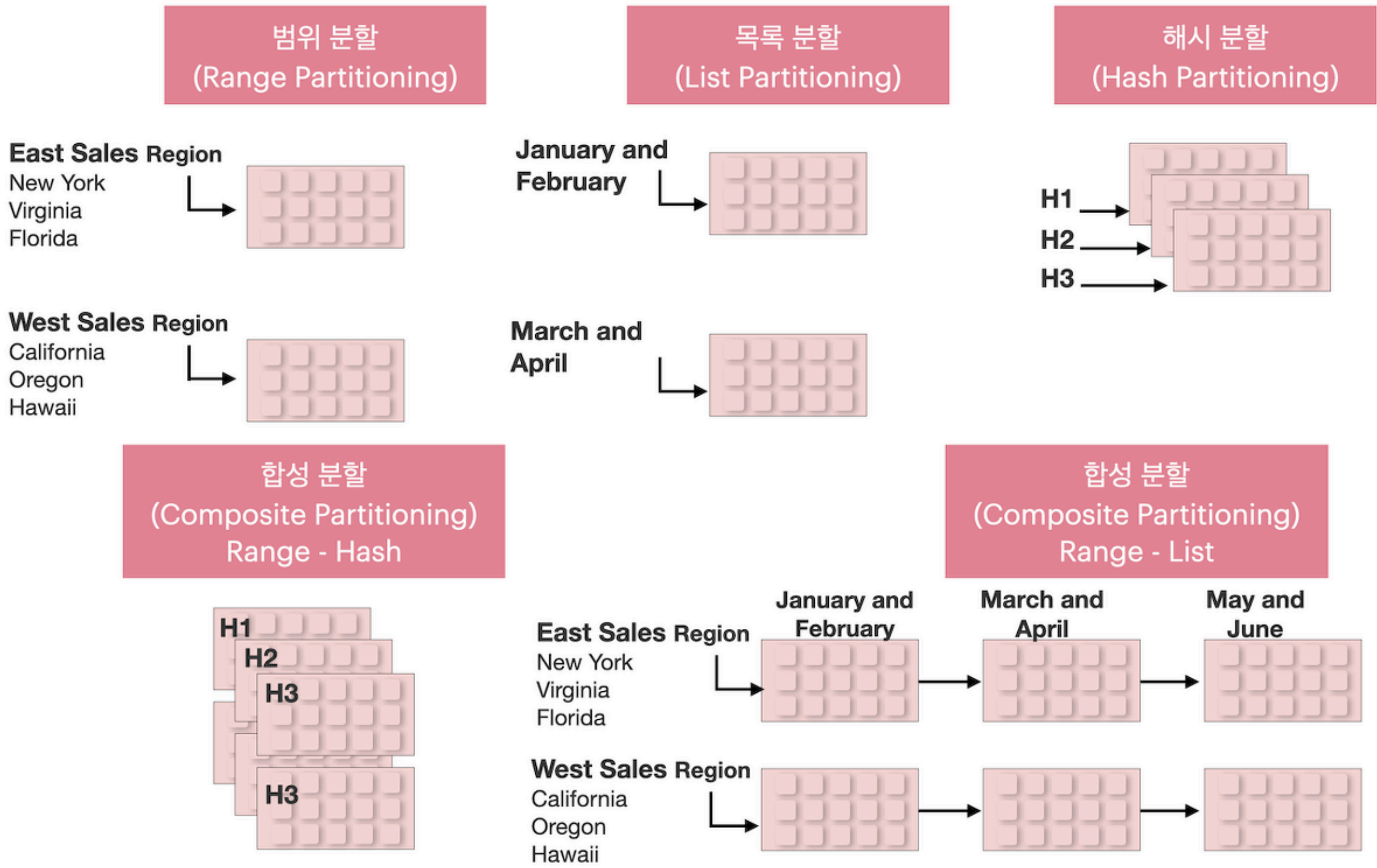
파티셔닝

성능, 가용성, 관리용이성

수직, 수평적 파티셔닝

샤딩

데이터베이스 x, 애플리케이션 레벨 o



Ref

1. Real My SQL 8.0.13 파티션
2. 데이터베이스 파티셔닝과 샤딩
3. [10분 테코톡] 히브리의 Sharding, Clustering, Replication
4. [DB] 파티셔닝? 샤딩? 레플리케이션?
5. [데이터베이스] 파티셔닝(Partitioning)이란?

수직 파티셔닝은 언제 가능할까?

구글 캘린더를 파티셔닝한다면 어떤 방식을 사용하는게 좋을까요?