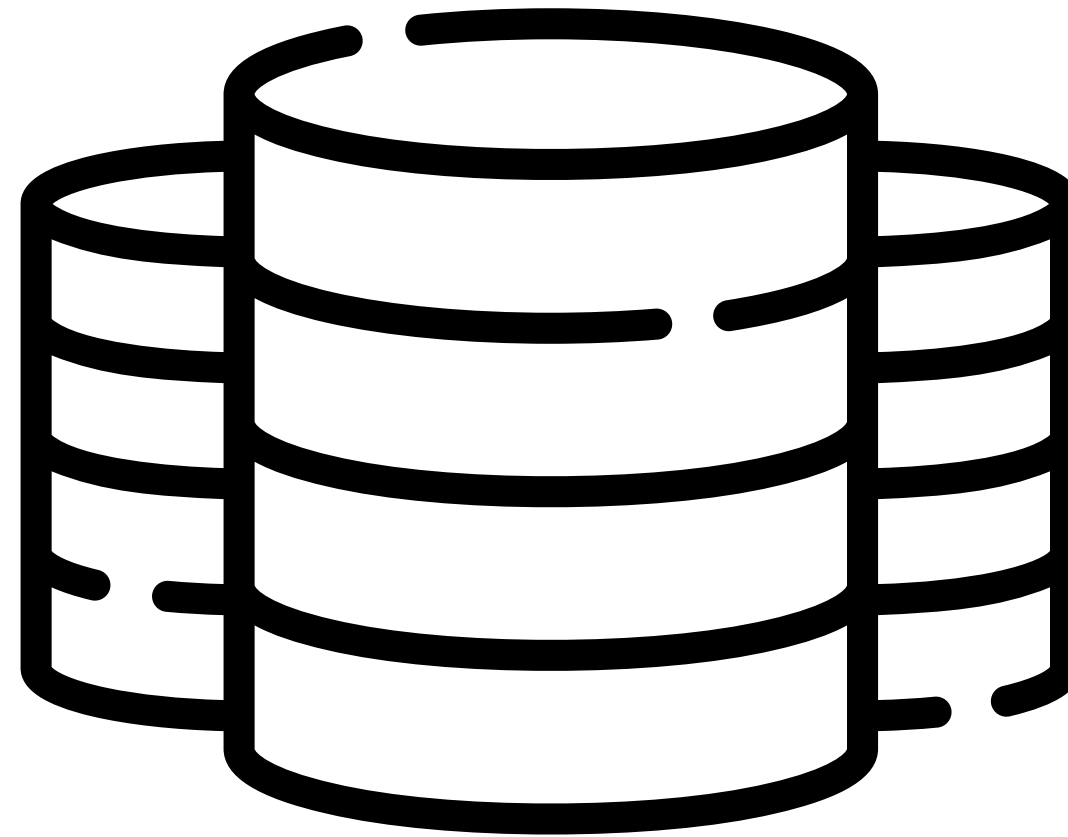
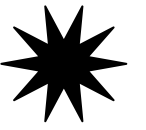


# Database Index

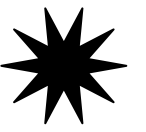


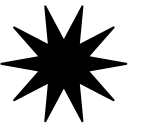


1. 정의
2. 관리
3. 장단점
4. 어느 경우에 사용하는가
5. 인덱스(Index)의 자료구조

01

# 인덱스(Index)란?

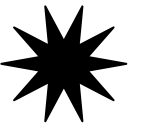




**INSERT** : 새로운 데이터에 대한 인덱스를 추가한다.

**DELETE** : 삭제하는 데이터의 인덱스를 사용하지 않는다는 작업을 진행한다.

**UPDATE** : 기존의 인덱스를 사용하지 않음 처리하고, 갱신된 데이터에 대해 인덱스를 추가한다.

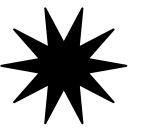


### 장점

1. 테이블을 조회하는 속도와 그에 따른 성능을 향상시킬 수 있다.
2. 중복 데이터를 방지하거나 특정 컬럼의 유일성(Unique)을 보장할 수 있다.
3. ORDER BY 절과 GROUP BY 절, WHERE 절 등이 사용되는 작업이 더욱 효율적으로 처리된다.

### 단점

1. 인덱스 생성에 따른 추가적인 저장 공간이 필요하다. (인덱스 사용 시 해당 정보를 담은 MYI 파일 생성)
2. CREATE(삽입), DELETE(삭제), UPDATE(수정) 작업 시에도 인덱스를 업데이트해야 하므로 성능 저하가 발생할 수 있다.
3. 한 페이지를 동시에 수정할 수 있는 병행성이 줄어든다.
4. 인덱스 생성 시간이 오래 걸릴 수 있다.
5. 인덱스를 잘못 사용할 경우, 오히려 성능이 저하되는 역효과가 발생할 수 있다.



#### 대량의 데이터 검색

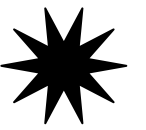
대량의 데이터를 전체 스캔하는 것은 매우 느리고 부하가 발생하기 때문에 이 경우에 인덱스를 사용하여 검색하는 것이 효율적입니다.

#### 정렬된 결과 출력

인덱스를 사용하여 데이터를 정렬하면 매우 빠르게 정렬된 결과를 출력할 수 있습니다.

#### 조인 연산 수행

데이터베이스에서 조인(join)은 두 개 이상의 테이블을 결합하여 필요한 정보를 가져오는 작업을 말합니다. 이때 인덱스를 사용하면 연산 속도를 향상시킬 수 있습니다.



#### 유니크한 값을 가져오는 경우

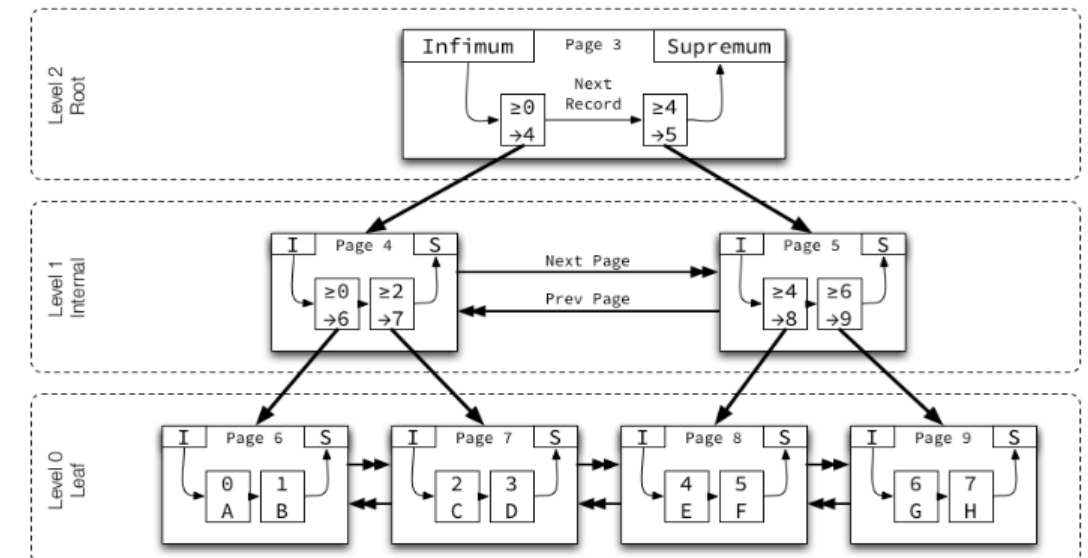
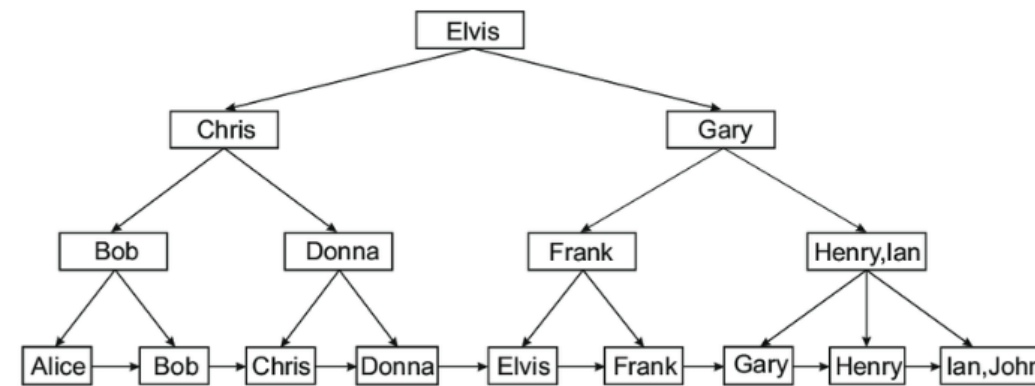
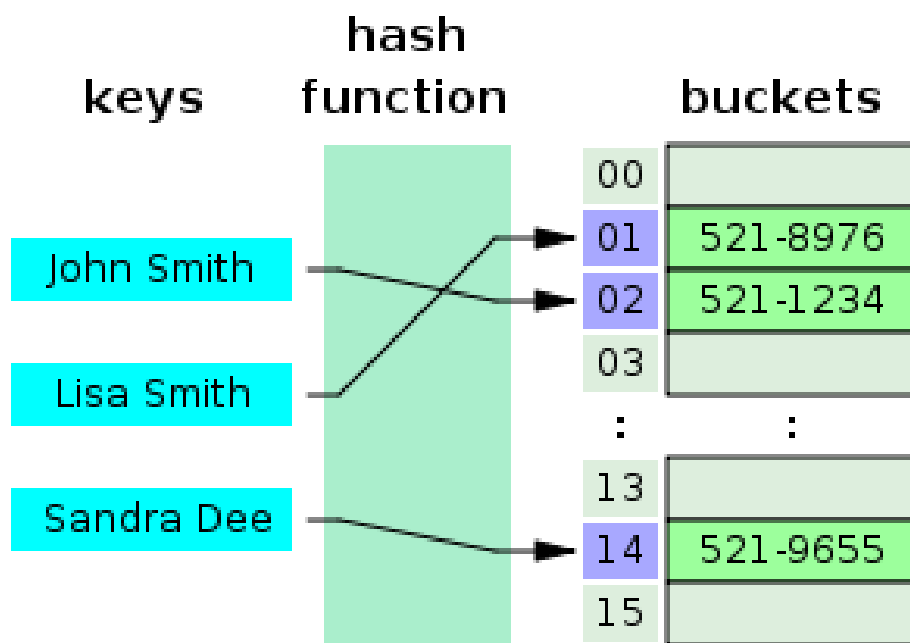
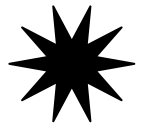
인덱스는 유니크한 값을 가지고 있는 필드에 대해 중복되지 않는 값을 빠르게 검색할 수 있습니다. 이러한 경우 인덱스를 사용하여 검색 속도를 빠르게 할 수 있습니다.

#### 검색 빈도가 높은 경우

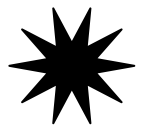
검색 빈도가 높은 필드에 대해서 인덱스를 생성하여 검색 속도를 향상시키는 것이 좋습니다. 자주 검색되는 필드에 인덱스를 적용하면 검색 작업이 빠르고 효율적으로 수행됩니다.

# 05

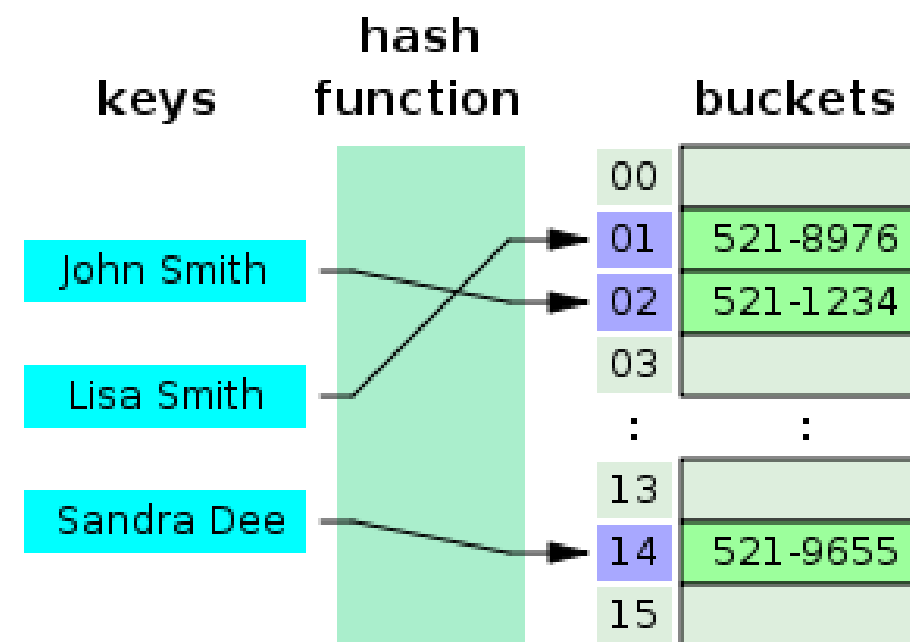
## 인덱스(Index)의 자료구조

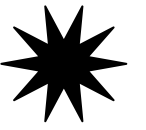




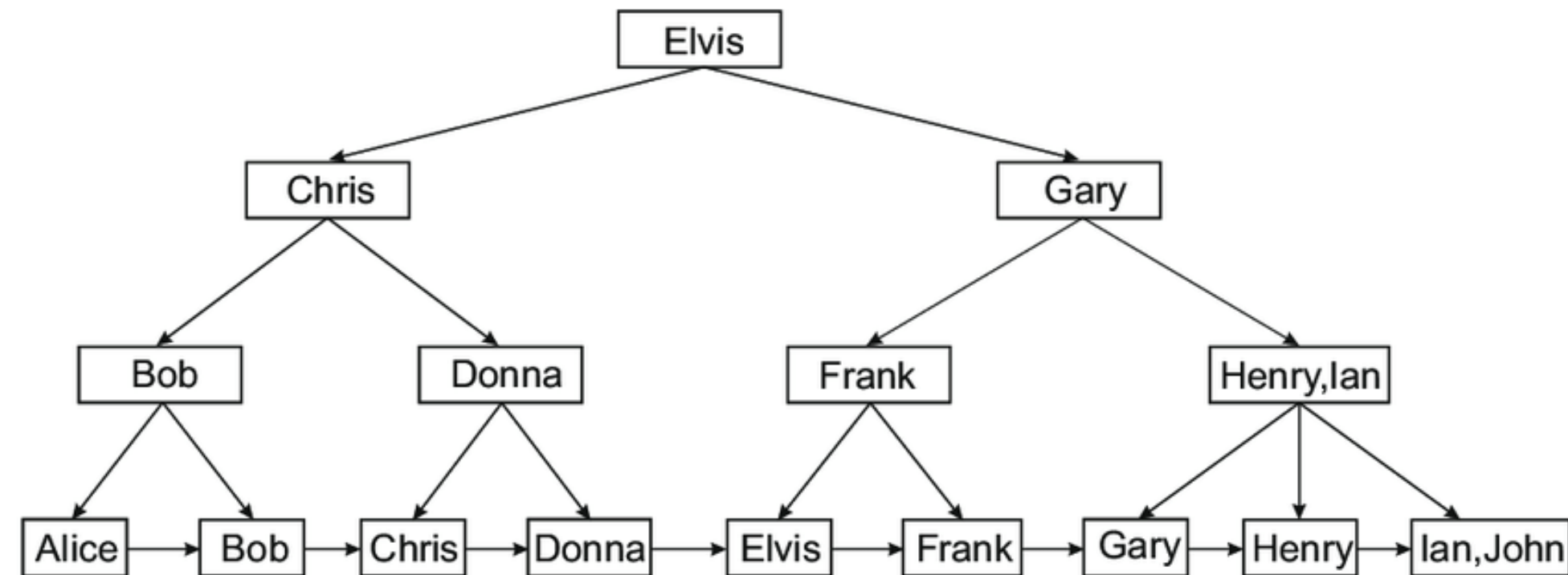


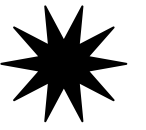
## 1. 해시 테이블(Hash table)



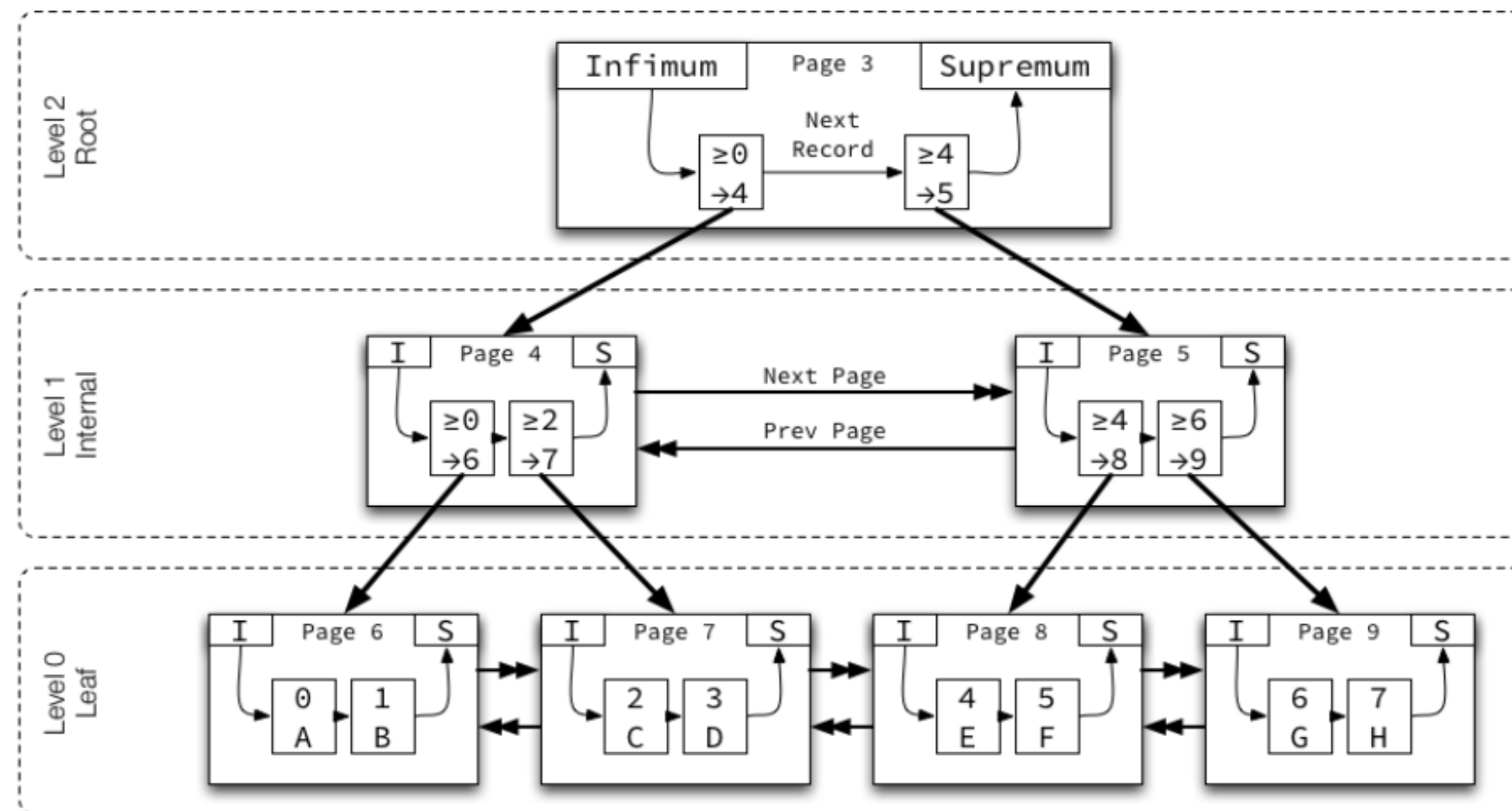


## 2. B-Tree





## 3. B+Tree



감사합니다.

