

인덱스

인덱스 Index

색인 : 쉽게 찾아볼 수 있도록 **일정한 순서**에 따라 놓은 **목록**

A - Z

apply	79
arguments	41
array-like object	80
arrow function	74
asynchronous	104
bind	85
call	78
callback function	92
callback hell	104

출처 : 코어 자바스크립트 목차

원하는 값을 빠르게 찾는다!

SELECT INSERT UPDATE DELETE

데이터베이스 인덱스

데이터가 특정 **기준**으로 **정렬**되어 있다면 **검색**을 **빠르게** 할 수 있다

```
SELECT * FROM member  
WHERE email = "abc@gmail.com"
```

인덱스가 적용된 대상(email로 정렬된 데이터)을
WHERE 절을 통해 검색

```
SELECT * FROM member
```

WHERE 절을 통해 검색 X
인덱스가 사용되지 않음


데이터베이스에서 인덱스란?

데이터베이스 테이블에 대한 **검색 성능을 향상**시키는 자료 구조이며 **WHERE** 절 등을 통해 활용된다.

- 인덱스는 항상 최신의 정렬 상태를 유지
- 인덱스도 하나의 데이터베이스 객체
- 데이터베이스 크기의 약 10% 정도의 저장공간 필요

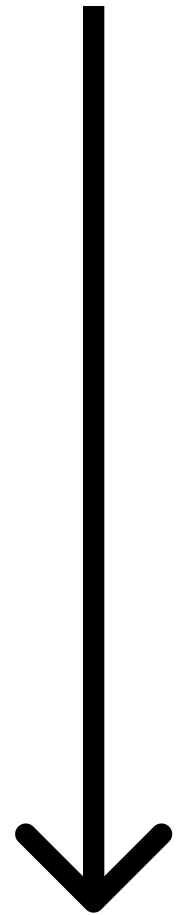
인덱스 알고리즘 : Full Table Scan

페이지 1



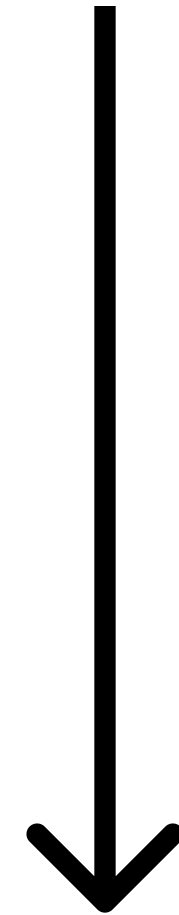
1	A
2	B
3	C
4	D

페이지 2



5	G
6	H
7	J
8	K

페이지 3



9	L
10	M
11	N
12	O

페이지 : 데이터가 저장되는 단위 (16Kbytes)

- 적용 가능한 인덱스 없는 경우
- 인덱스 처리 범위가 넓은 경우
- 크기가 작은 테이블에 액세스 하는 경우

인덱스 알고리즘 : B-Tree

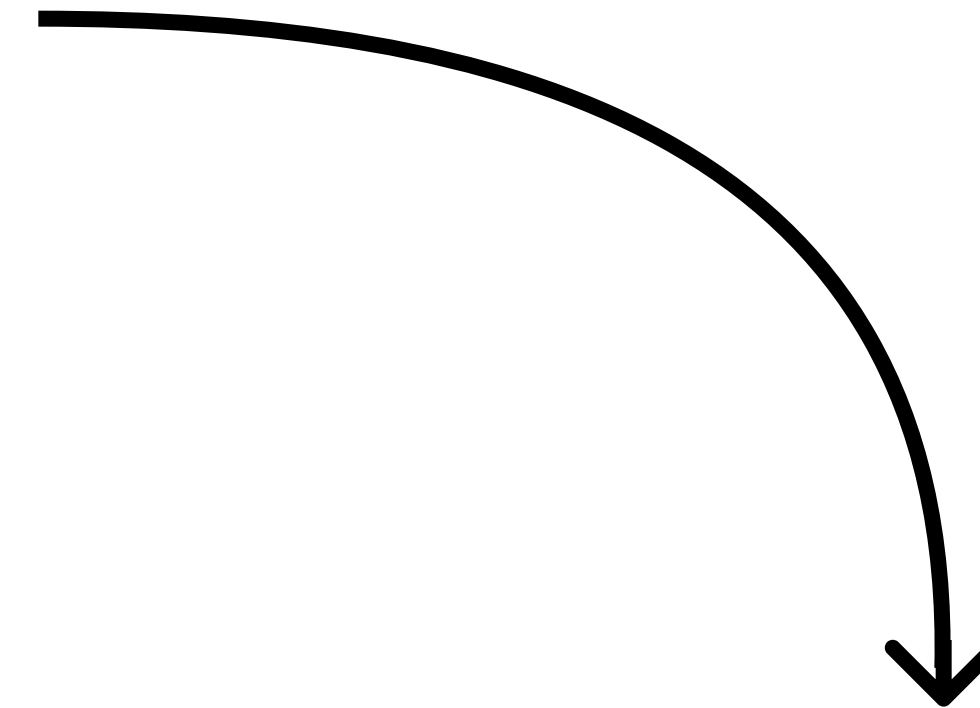
- 트리 높이가 같음
- 자식 노드를 2개 이상 가질 수 있음
- 기본 데이터베이스 인덱스 구조

리프 페이지

A
B
C
D

루트 페이지

1	A
2	H
3	O



G
H
J
K

4	L
5	M
6	N
7	O

인덱스와 DML

SELECT

성능 향상

INSERT

UPDATE

DELETE

페이지 분할과 사용 안함 표시로
인덱스의 조각화가 심해져

성능 저하

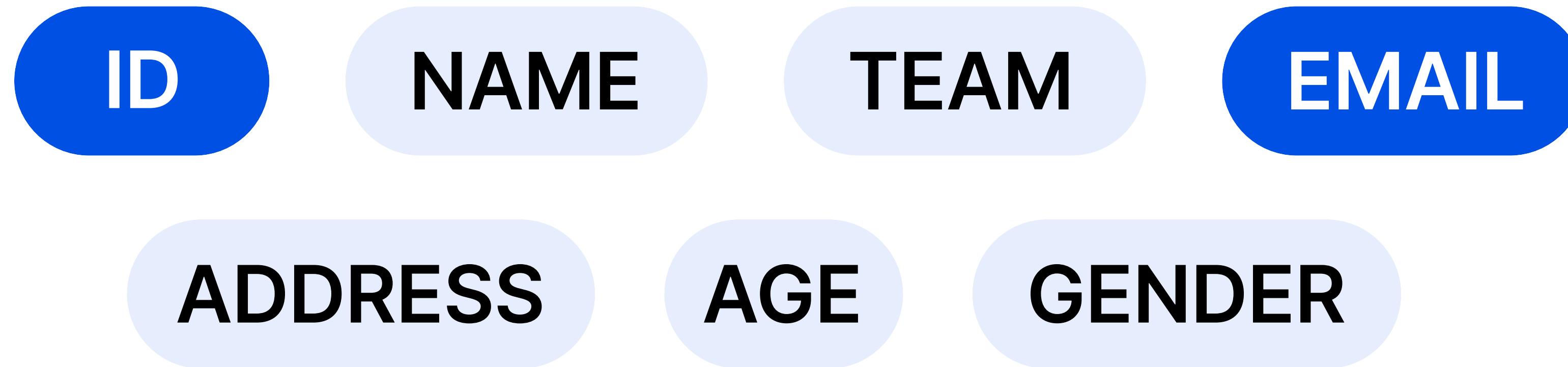
B-Tree

- 루트 페이지 : 자식 페이지 정보 담겨 있음
- 브랜치 페이지 : 자식 페이지 정보 담겨 있음
- 리프 페이지 : 실제 데이터 페이지 (**클러스터링 인덱스**)

실제 데이터의 주소 페이지 (**논-클러스터링 인덱스**)

카디널리티 Cardinality

: 그룹 내 요소의 개수



카디널리티가 높은 것
= 중복도가 낮은 것

인덱스 적용 기준

- 카디널리티가 높은 (중복도가 낮은) 칼럼
- WHERE, JOIN, ORDER BY 절에 자주 사용되는 칼럼
- INSERT, UPDATE, DELETE가 자주 발생하지 않는 칼럼
- 규모가 작지 않은 테이블