

Part 1. Database

데이터 베이스를 사용하는 이유?

파일 시스템

- 컴퓨터에서 파일이나 자료를 쉽게 발견하고 접근할 수 있도록 저장하는 것
- 윈도우의 NTFS, 리눅스의 ext4
- 모든 응용프로그램은 개별적으로 파일 시스템 필요
- 데이터 종속성... 무결성.. 중복성..

데이터베이스 시스템

- 데이터베이스는 여러 사람들이 공유하여 사용할 목적으로 체계화해 통합, 관리하는 데이터의 집합
- DBMS를 통해 요구 처리, SQL 을 사용하여 데이터에 접근 하여 여러 작업 수행
- 통합된 환경을 제공함으로써 데이터 중복을 최소화하고 데이터의 논리적인 독립성을 추구

데이터 베이스 특징

장점

- 데이터의 독립성
- 데이터의 무결성
- 데이터의 보안성
- 데이터의 일관성
- 데이터 중복 최소화

단점

- 운영비용 증가
- 오버헤드의 가능성

데이터베이스의 성능을 높이는 법?

- 디스크 I/O를 잘~ 줄이면 된다!
- 디스크 I/O란
 - 데이터를 작성 하고 변경 할 때 디스크 즉 HDD 에 그것이 저장되는 것

Part 2. Index

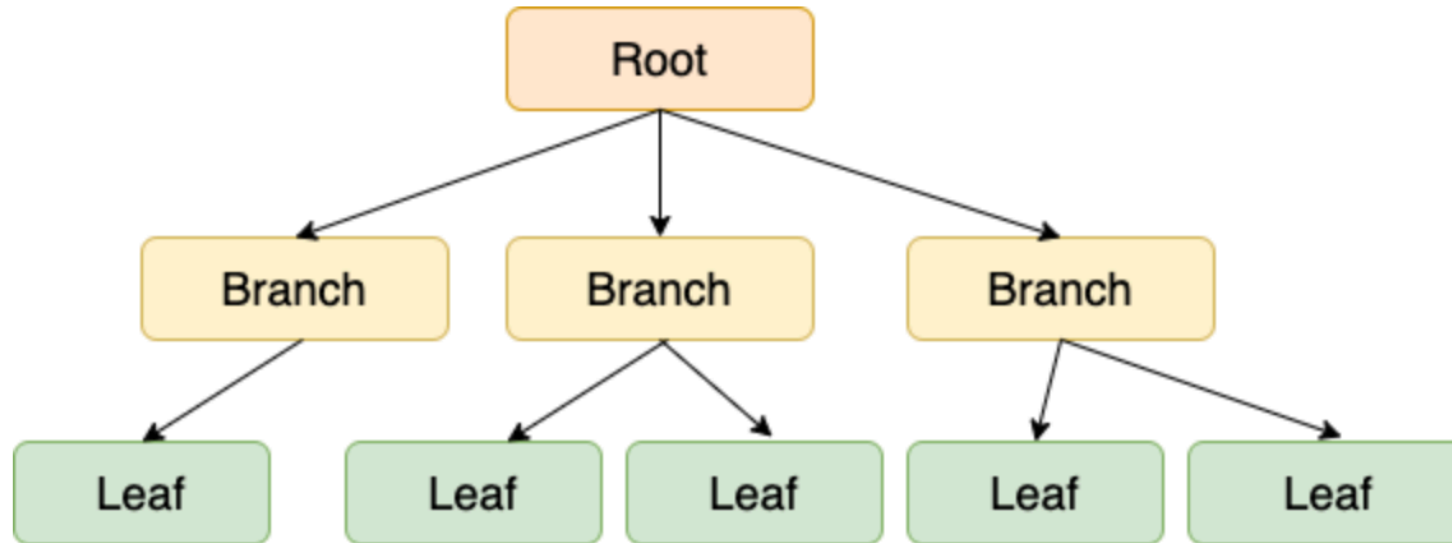
인덱스가 뭐예요?

- 간단하게 순번이죠 뭐
- 데이터 베이스에의 인덱스는
 - 데이터 = 책의 내용
 - 데이터가 저장된 레코드의 주소 = 인덱스 목록에 있는 페이지 번호
- DBMS의 인덱스는 항상 정렬되어있음
 - 저장성능 down, 데이터 읽는 속도 up

DBMS의 인덱스 자료구조

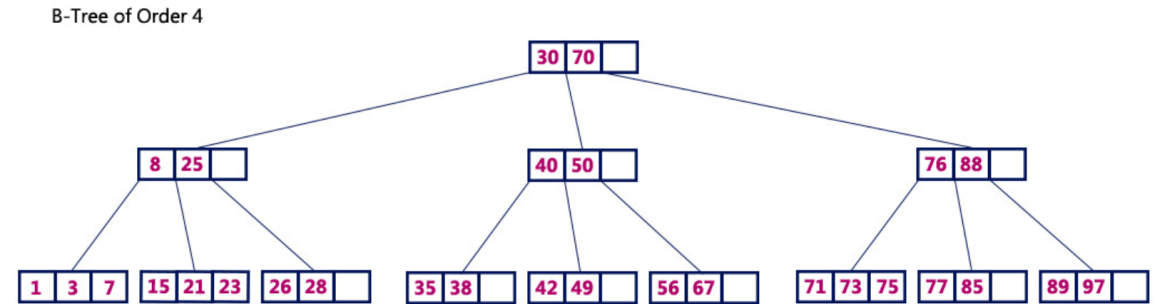
- B+-Tree (B-Tree, B+Tree)

: 최상위에 root 노드, 최하위에 leaf 노드, 중간에 branch 노드로 구성



B-Tree

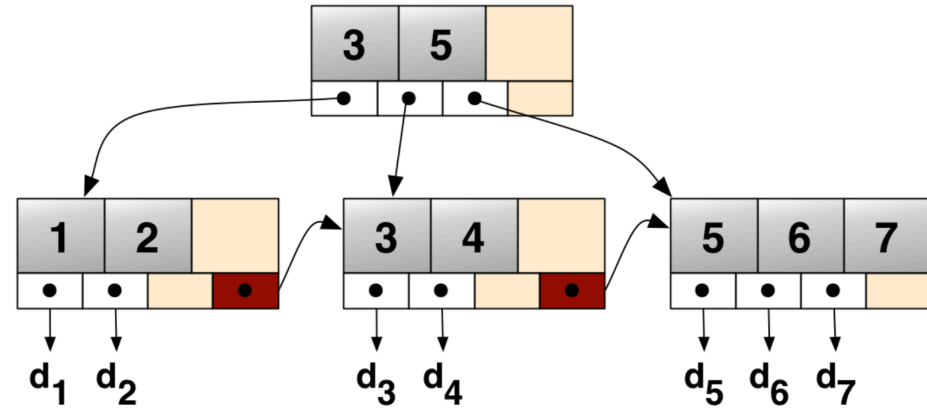
- 한 노드당 자식노드가 2개 이상 가능
- Key값을 이용해 데이터를 찾음
- 데이터를 찾는속도가 같음(=비슷함)
- 균형트리
- 브랜치 노드에 key, data 담기 가능



출처: http://www.btechsmartclass.com/data_structures/b-trees.html

B+Tree(B-Tree의 확장)

- 오직 리프노드에만 key, data 담기 가능
- 브랜치 노드에서는 key만
- 리프노드는 링크드 되어있음
- 리프노드만 데이터를 담아 메모리 손실 줄여 더 많은key 수용



B-tree Vs B+tree



아래서 말하는 데이터는 자료구조 상 value를 가리킴. (실제 DB 데이터 X)

구분	B-tree	B+tree
데이터 저장	리프 노드, 브랜치 노드 모두 데이터 저장 가능	오직 리프 노드에만 데이터 저장 가능
트리의 높이	높음	낮음(한 노드 당 key를 많이 담을 수 있음)
풀 스캔 시, 검색 속도	모든 노드 탐색	리프 노드에서 선형 탐색
키 중복	없음	있음(리프 노드에 모든 데이터가 있기 때문)
검색	자주 access 되는 노드를 루트 노드 가까이 배치할 수 있고, 루트 노드에서 가까울 경우, 브랜치 노드에도 데이터가 존재하기 때문에 빠름	리프 노드까지 가야 데이터 존재
링크드 리스트	없음	리프 노드끼리 링크드 리스트로 연결

Hash

- $O(1)$ 겁나빠름 but...잘 안쓰임 왜?
- Select문에 비교연산자(< >)는 hash가 추구하는 = 과 맞지 않음.
=> 그래서 메모리기반 데이터베이스에서 사용된다고 합니다!
- <https://gaeggu.tistory.com/7>

인덱스 설정할 때 주의점

- 중복이 많은 데이터에는 인덱스 추가를 안하는게 좋음
- 문제! 주민등록 vs 성별 어느 column에 index를 추가하는것이 좋을까?

인덱스 설정할 때 주의점

- Primary vs secondary idx
 - 클러스터란?
 - 여러 개를 하나로 묶는것
 - 클러스터드 인덱스란?
 - 비슷한 데이터를 동시에 조회하는 경우가 많아 비슷한 값을 물리적으로 인접한 장소에 저장함.
 - Primary만 영향을 주기때문에 신중하게 설정해야함
 - 문제! 클러스터드 인덱스는 테이블 당 몇개 생성 가능할까?

인덱스 설정할 때 주의점

- Composite index(결합 인덱스)
 - 인덱스를 생성할 때 두 개 이상의 컬럼을 합쳐서 인덱스를 만드는 것
 - AND 조건으로 검색되는 경우만

```
CREATE INDEX idx_emp_comp ON emp(title, author);
```

vs

```
CREATE INDEX idx_emp_comp ON emp(author, title);
```

- Select질의를 어떻게 할것인가는 인덱스 생성에 영향을 줌