

# 물리적 모델링

# 물리적 모델링

- 물리적 데이터베이스 모델링 단계는 우선 논리적 데이터베이스 모델링 단계에서 얻어진 데이터베이스 스키마를 좀더 효율적으로 구현하기 위한 작업과 함께 개발하려는 DBMS의 특성을 맞게 실제 데이터베이스 내의 개체들을 정의하는 단계임.
- 물리적 데이터베이스 모델링에서 특히나 중요한 점은 데이터 사용량 분석과 업무 프로세스 분석을 통해서 보다 효율적인 데이터베이스가 될 수 있도록 효과적인 인덱스를 정의하고 상황에 따른 적절한 역정규화를 수행하는 점임.
- 역정규화(De-Normalization)란 무엇인가?
  - 논리적 데이터베이스 모델링 단계에서 정규화 과정을 거치면 테이블들이 점점 분할되어 테이블 수가 증가하게 됨.
  - 테이블 수가 증가했다고 하는 것은 결국 원하는 정보를 위해 테이블들간의 조인이 자주 발생하고 데이터의 입력, 수정, 삭제 시에 데이터의 무결성을 검증(참조 무결성)하기 때문에 기본적으로 부하가 발생함.
  - 역정규화란 시스템의 퍼포먼스 향상을 위해서 정규화에 위배되는 행위를 말함.

# 물리적 모델링

- 역정규화가 완료되면 지금까지 정의했던 데이터베이스 스키마를 실제 데이터베이스 내의 개체로 생성한 후 본격적인 프로그램 개발 작업에 들어감.
- 물리적인 데이터베이스 모델링을 제대로 수행하기 위해서는 최소한 개발 대상 DBMS에 관해서 완벽한 이해를 하고 있어야함. 그렇지 않으면 효과적인 데이터베이스 모델링을 할 수 없음.

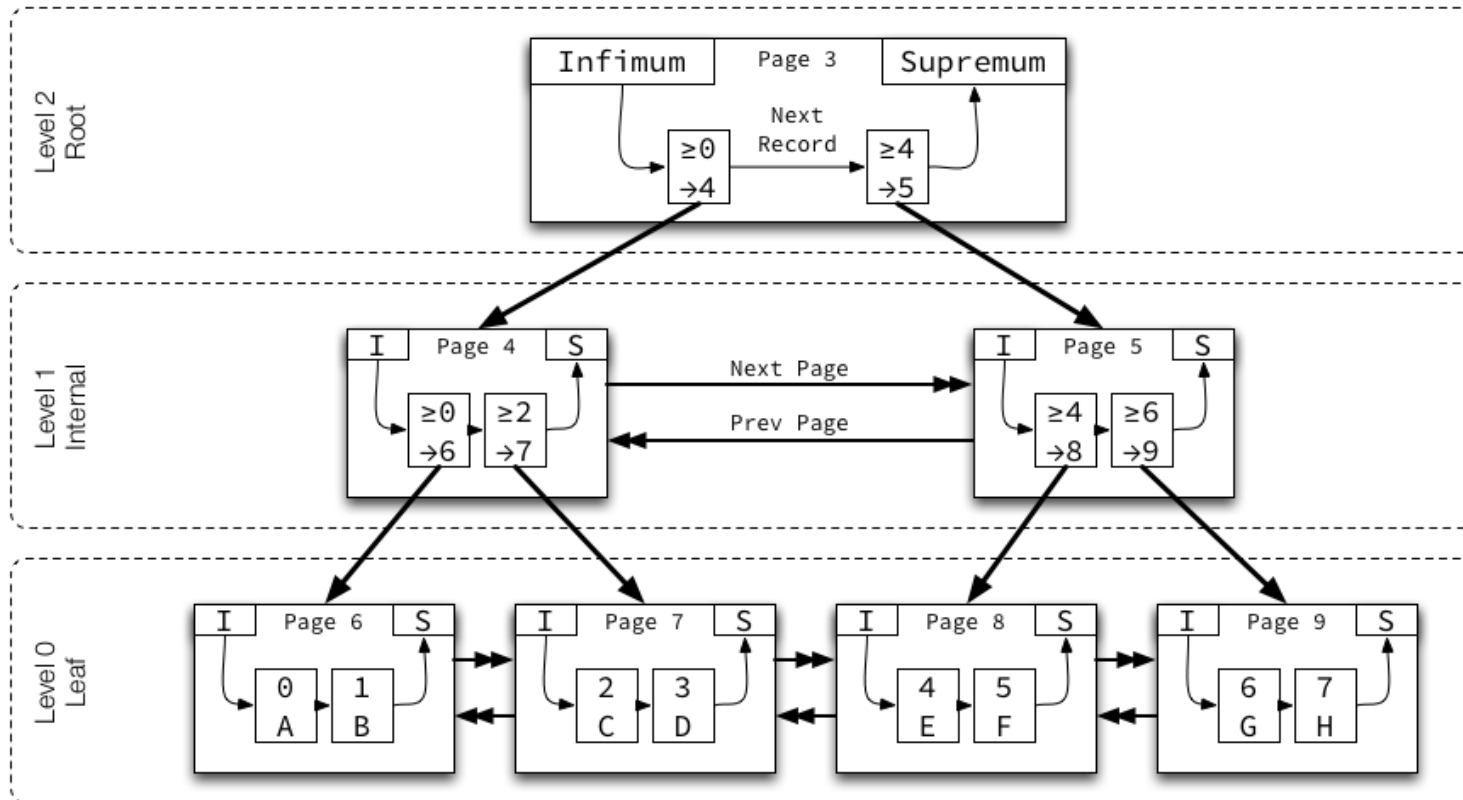
# 인덱스에 대한 정의

- 인덱스란 무엇인가?
  - 인덱스는 항상 정렬된 상태로 만들어지고 유지됨.
  - 정렬되어 있기 때문에 원하는 데이터를 인덱스를 이용해서 빠르게 검색할 수 있음.
  - 정렬 상태는 기본이 오름차순(Ascending)이고 내림차순(Descending)으로 인덱스의 정렬 상태를 정의할 수도 있음.

# 인덱스 정의를 위해 참고 할 사항

- 데이터 저장 단위
  - 데이터 베이스는 페이지(Page)라는 공간을 할당하고 이 페이지에 실제 데이터가 저장됨.
    - 예) Sql Server는 데이터 페이지(Page)가 8kb임
    - Page는 Page Header(96byte), Data Rows(8060Byte), Row Offset등으로 정의됨.
- 데이터 페이지(Data Page)
  - 데이터가 저장되기 위해 할당되는 페이지를 데이터 페이지라고 함. 사용자가 입력하는 모든 데이터가 실제로 저장되고 관리되는 곳
- 인덱스 페이지(Index Page)
  - 인덱스 페이지란 인덱스와 관련된 데이터를 저장하는 페이지임.
  - 일반적인 책들을 보면 책의 내용과 함께 책 후미에 인덱스를 위한 추가적인 공간이 할당되어 있는 것을 볼 수 있음.

## B+Tree Structure



Levels are numbered starting from 0 at the leaf pages, incrementing up the tree.

Pages on each level are doubly-linked with previous and next pointers in ascending order by key.

Records within a page are singly-linked with a next pointer in ascending order by key.

Infimum represents a value lower than any key on the page, and is always the first record in the singly-linked list of records.

Supremum represents a value higher than any key on the page, and is always the last record in the singly-linked list of records.

Non-leaf pages contain the minimum key of the child page and the child page number, called a "node pointer".

# 역정규화

- 역정규화란 무엇인가?
  - 시스템의 성능을 향상을 위해서 정규화에 위배되는 행위를 하는 것
- 역정규화 고려사항은 업무에 따른 데이터 사용량(입력, 수정, 삭제, 조회)와 업무 프로세스 분석임.

# 역정규화 유형

- 데이터 중복(컬럼 역정규화)
  - 조인 프로세스를 줄이기 방법.
  - 조인을 해서 다른 테이블에 있는 컬럼의 데이터를 조회하는 경우, 아예 해당 컬럼을 중복해서 정의하는 방법.
- 파생컬럼 생성
  - 데이터 중복과 비슷한 유형으로 테이블에 없는 컬럼을 숫자 연산이나 데이터 조작 등을 통해서 조회 할때 새로운 정보를 보여주는 경우.
- 테이블 분리
  - 컬럼을 기준으로 분리
  - 레코드를 기준으로 분리
- 운영 중인 시스템이라면 더더욱 신중으로 고려해서 진행 할 것.



# 역정규화 유형

- 요약 테이블 생성
  - 판매 테이블과 같이 비교적 많은 데이터가 저장되는 테이블에서 요약된 정보를 얻기 위해서 Group By나 Sum등을 이용하여 가공된 결과를 얻는 질의를 자주 실행하게 되는 경우 해당 질의를 수행하기 위해 많은 프로세스가 발생하게 되는 경우
- 관계제거
  - 권장할 만한 방법은 아님. 무결성을 보장할 수 없음.
  - 자식 테이블의 데이터를 입력하거나 수정하는 경우 부모 테이블을 검증함. 부모 테이블에 데이터를 수정하거나 삭제할때 자식 테이블의 데이터를 참조하는지 검증. 이러한 프로세스를 허용하지 않으려면 관계를 제거하고 사용함.

# 역정규화 유형

- 테이블 통합
  - 두 테이블은 원래 하나였지만 이유는 효율적으로 데이터를 저장하기 위해서였음. 그러나 데이터를 입력, 수정, 삭제할때마다 부모 자식 테이블과 자식 테이블의 데이터를 참조하는 경우가 많이 발생함. 이러한 경우 데이터를 합쳐 처리하는 것을 말함.