학습목표

1. DBMS의 기초
2. RDBMS

학습내용

- DBMS에 대한 개념을 이해할 수 있습니다.
- RDBMS(관계형 데이터베이스 시스템)의 기본 개념을 이해할 수 있습니다.

사전퀴즈

1. RDBMS는 데이터의 CRUD(Create, Retrieve, Update, Delete)가 모두 빠른 시스템을 말한다.

정답: X

RDBMS는 데이터의 검색(Retrieve)를 빠르게 하기 위해서 데이터의 변경(추가/업데이트/삭제)할 때 인덱스를 미리 만들어서 데이터의 검색을 빨리 되도록 한다. 하지만 데이터의 변경이 많을 경우에는 인덱스때문에 오히려 성능이 떨어지기도 한다.

수업

1. DBMS의 기초

* DataBase Management System
  - 데이터베이스(DB)를 관리하는(Management) 시스템(System)

* DB : 테이블들이 모여 이루는 데이터 단위
  - 데이터를 저장하고 유지보수(수정, 삭제, 추가)하고 이를 검색하는 시스템

* CRUD(Create, Retrieve, Update, Delete)
  - 대량의 데이터를 처리하는 시스템
  - 다양한 자료구조와 검색구조(소팅, 인덱싱, ...) 사용해 "빠른" 검색 가능
  - 대부분의 시스템은 R(검색) >>>> CUD(업데이트)의 빈도수가 많음

  예) 주민등록 데이터베이스 시스템의 경우 통상적으로 5,100만 ~ 200만 사이의 레코드가 있는데

* 이 레코드 내용 중에서 CUD가 차지하는 비율은 1% 미만이지만 데이터베이스 시스템 검색의 경우 수백만건이 됨
  - 검색에 최적화

정렬

* 빠른 검색을 위해서는 데이터가 반드시 정렬(Sorting)되어 있어야 함
  - 정렬되어 있지 않다면 평균적으로 전체 데이터의 절반 필요
  (최선:1, 최악:N, 평균: N/2)
* 정렬되어 있을 경우 데이터를 빠른 시간 안에 찾을 수 있음
* O(NlogN)-O(N^2)
  - 퀵정렬/힙정렬 계열이 주로 사용됨

인덱스(Index)

* 인덱스종류

1) 이진검색(Binary Search)
   - 최대 $\log_2(N)$번 내에 검색가능

2) B-Tree 계열
   - 최대 $\log_3(N)$번 내에 검색가능
   - 상용 DBMS에서 가장 일반적으로 많이 사용됨

데이터 추가/수정/삭제할 때마다 **정렬/인덱스 업데이트**가 일어남

이진탐색(Binary Search)

* 데이터를 정렬 후 "test" 단어를 검색하는 경우
  - 한 가운데 값을 확인 → "sample" → 뒤쪽 절반
  - 뒤쪽 중 한가운데 확인 → "zeal" → 앞쪽 절반
  - 계속 반복해 "test" 단어가 나올 때까지 계속
  예) 1,000개의 데이터가 있을 경우에 10번만 찾으면 데이터를 찾는 것이 이론적으로 보장됨

$2^N > 1000$인 값(N=10)

  - 데이터가 추가/삭제/변경될 때마다 한가운데/왼쪽 가운데/오른쪽 가운데값 등을 미리 계산해 놓음
  → 인덱스(Index)라고 통칭함

B-트리(B-Tree)

* 이진 검색과 유사하지만 한 번에 비교를 2번(a,b: $a < b$)
  - 작은 값 보다 작은 경우 ($x < a$)
  - 큰 값과 작은 값 사이인 경우 ($a < x < b$)

- 큰 값보다 큰 경우 (x > b)

* B-트리 계열 > 이진검색 계열

  $O(\log_3 N) > O(\log_2 N)$

데이터가 추가/삭제/변경 될 때마다 a,b 값을 업데이트

DBMS의 종류

* 역사적으로 여러 형태의 DB가 존재했음
  - 계층형 데이터베이스
  - 네트워크형 데이터베이스
  - 관계형 데이터베이스(RDBMS) : 80년대부터 주류
  - 객체지향 데이터베이스
  - 객체관계형 데이터베이스(ORDBMS)
  - NoSQL(Not Only SQL)
* RDBMS와 NoSQL의 차이점?

  RDBMS는 읽기 최적화, NoSQL 쓰기 최적화 기술

2. RDBMS

RDBMS란?

* 관계형(Relational) 데이터베이스 시스템
* 테이블(Table based) 기반의 DBMS
  - 테이블/ 컬럼 형태의 데이터 저장 방식
  - 테이블과 테이블 간의 연관관계(주로 외래키 형태)를 이용해 필요한 정보를 구하는 방식
* 모델링은 E-R(Entity Relationship) 모델을 사용
* 테이블을 엔티티(기본)와 릴레이션(유도) 테이블로 구분하는 방식

  예) 엔티티(기본) 테이블 - 학생, 교수

  릴레이션(유도) 테이블 - 수업 : 학생과 교수를 연결시키는 유도된 테이블
* 데이터를 테이블(Table) 단위로 관리
  - 하나의 테이블은 여러 개의 컬럼으로 구성됨
* 테이블기리의 중복정보는 최소화시킴
  - 동일한 데이터가 여러 군데 중복되어 존재하면 데이터의 수정 시 문제 발생 확률 높아짐
  - 정규화(Normalize) → 정규형
* 사용방식
  - 여러 테이블을 합쳐 큰 테이블을 생성(조인:JOIN)해서 필요한 정보를 찾아내는 방식

기본용어

스키마(Schema)

DB, 테이블 정의 내역

SQL쿼리(SQL Query)

- 관계형 DBMS를 사용하는 전용 질의언어
- 대소문자 가리지 않음

기본키(Primary Key: PK)

- 테이블에서 하나의 레코드를 지정할 수 있는 하나 이상의 컬럼집합
- 예) 주민등록번호, SSN(Social Security Number)

외래키(Foreign Key: FK)

- 어떤 테이블의 기본키가 다른 테이블의 컬럼에 들어 있을 경우
- 테이블과 테이블을 조인할 때 보통 많이 쓰임

테이블(Table)

- 정보들의 묶음단위
- 예) 학교, 학생, 교수 ...

컬럼(Column)

- 테이블을 구성하는 정보들
- 예) 학생테이블 - 이름, 주소, 전화, 번호, 나이, 성별 ...

레코드(Record)

- 테이블에 들어 있는 여러가지 인스턴스(개체) 하나하나를 지정
- 대학교의 학과테이블
- 예) 경영학과, 미술학과, 수학과, 컴퓨터공학과 ...
- 기본키(PK)로 구별가능

도메인값(Domain Value)

- 각 컬럼에서 나올 수 있는 후보값
- 예) "계절" 컬럼의 도메인값은 봄, 여름, 가을, 겨울

DBMS가 데이터처리에 적합한 이유?

전문가 의견

데이터가 적다면 굳이 DBMS에서 관리할 필요가 없는 경우도 있습니다.
데이터가 많지 않으면 엑셀이나 간단한 표로도 관리가 가능합니다.
DBMS가 의미를 가지려면 충분히 큰 데이터(보통 수 만 개의 레코드 이상)를 관리하는 경우이어야 합니다.
많은 데이터를 관리하기 위해서 DBMS에서는 저장된 데이터를 필요한 컬럼 순서대로 정렬합니다. (보통 기본 키 컬럼) 그리고 데이터가 추가, 수정, 삭제될 때마다 해당 컬럼 기반으로 새롭게 정렬합니다.
그리고 찾는 데이터의 유무를 빠르게 판단하기 위해서 데이터 구조를 만들어 놓습니다.
데이터 구조로는 다양한 형태의 인덱스가 있는데 가장 많이 사용되는 것은 B-트리 인덱스입니다.
B-트리 인덱스는 alpha, beta값(alpha < beta)을 기준으로 alpha보다 작은 경우, alpha보다 크고 beta보다 작은 경우, beta보다 큰 경우 이렇게 세 가지로 나누고 자료를 트리에 따라가면서 검색하는 방법을 사용합니다. 그리고 이 외에 다양한 형태의 인덱스를 제공합니다.
결론적으로, DBMS는 빠른 데이터 검색에 최적화된 시스템이라고 볼 수 있고 이를 위해 데이터가 추가, 수정, 삭제될 때마다 정렬이 내부적으로 일어나고 인덱스가 업데이트(alpha, beta 값)되는 시스템으로 이해할 수 있습니다.