

## T아카데미 온라인 강의

# 데이터베이스 프로그래밍 기초

# 01. DBMS 기초



# CONTENTS

1

## DBMS의 기초

2

## RDBMS

# 학습목표

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

# 1. DBMS의 기초



# 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
- 검색에 최적화

# 1. DBMS의 기초

## ● 정렬

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

# 1. DBMS의 기초

## ● 인덱스(Index)

### ▶ 인덱스 종류

#### ■ 이진검색(Binary Search)

- 최대  $\log_2(N)$ 번 내에 검색가능

#### ■ B-Tree계열

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

데이터 추가/수정/삭제할 때마다 정렬/인덱스 업데이트가 일어남

# 1. DBMS의 기초

## 이진탐색(Binary Search)

### ▶ 데이터를 정렬 후 “test” 단어를 검색하는 경우

- 한가운데 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

$$2^N > 1000 \text{인 값}(N=10)$$

- 데이터가 추가/삭제/변경될 때마다 한가운데/왼쪽 가운데/오른쪽 가운데값 등을 미리 계산해 놓음  
인덱스(INDEX)라고 통칭 함



# 1. DBMS의 기초

## ● 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, b값을 업데이트

# 1. DBMS의 기초

## DBMS의 종류

- 계층형 데이터베이스
- 네트워크형 데이터베이스
- 관계형 데이터베이스(RDBMS)
- 객체지향 데이터베이스
- 객체관계형 데이터베이스(ORDBMS)
- NoSQL(Not Only SQL)

## 2. RDBMS



## 2. RDBMS

### ● RDBMS란?

- 관계형(Relational) 데이터베이스 시스템
- 테이블(Table based)기반의 DBMS
  - ✓테이블- 컬럼형태의 데이터 저장 방식
  - ✓테이블과 테이블 간의 연관관계(주로 외래키형태)를 이용해 필요한 정보를 구하는 방식
- 모델링은 E-R(Entity Relationship) 모델을 사용
- 테이블을 엔티티(기본)와 릴레이션십(유도) 테이블로 구분하는 방식

## 2. RDBMS

### ● RDBMS란?

- 데이터를 테이블(Table)단위로 관리
  - ➔ 하나의 테이블은 여러 개의 컬럼으로 구성됨
- 테이블끼리의 중복정보는 최소화시킴
  - ➔ 동일한 데이터가 여러군데 중복되어 존재하면 데이터의 수정시 문제 발생 확률 높아짐
  - ➔ 정규화(Normalize) -> 정규형
- 사용방식
  - ➔ 여러 테이블을 합쳐 큰 테이블을 생성(조인:JOIN)해서 필요한 정보를 찾아내는 방식

## 2. RDBMS

### ● 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

## 2. RDBMS

### ● 기본용어

#### ▶ 테이블(Table)

- 정보들의 묶음단위
- 예) 학교, 학생, 교수 ...

#### ▶ 컬럼(Column)

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

## 2. RDBMS

### ● 기본용어

#### ▶ 레코드(Record)

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

#### ▶ 도메인값(Domain Value)

- 각 컬럼에서 나올 수 있는 후보값



# 학습정리



- 지금까지 'DBMS 기초'에 대해 살펴보았습니다.

## DBMS(Database Management System)

데이터를 CRUD(생성, 수정, 삭제, 검색)하는 시스템

검색에 최적화(인덱스)

업데이트가 많은 경우 대안이 필요

NoSQL

## RDBMS(Relational DBMS)

테이블 형태로 관리되는 DBMS의 한 형태

SQL이라는 질의언어를 사용