DBMS란

- Database Management System

- 데이터베이스를관리하는 시스템

- 통상적으로 DB로 줄여서 표현 데이터베이스(Database)란

- 위키

- 여러 사람에 의해 공유되어 사용될 목적으로 통합하여 관리되는 데이터의 집합

- A database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modeling techniques.

데이터베이스(Database)란

- 위키

- 여러 사람에 의해 공유되어 사용될 목적으로 통합하여 관리되는 데이터의 집합

- A database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modeling techniques.

- Merriam-Webster 정의

- “Database

- Definition of database by Merriam-Webster”. 《merriam-webster.com》.

- : a usually large collection of data organized especially for rapid search and retrieval (as by a computer)

관계형(Relational) DBMS란

- Relation(관계)란

- 동일한 구조로 이루어진 튜플(레코드)의 집합(위키)

- 테이블기반(Table based)으로 데이터 저장하는 시스템 - 테이블, 컬럼, 레코드, ....

- 여러 개의 테이블로 나누고 각 테이블 간의 관계를 정의하는 방식의 DBMS의 한 종류

- 각 테이블 간의 관계는 외래키라는 공통컬럼을 가짐

RDBMS란

–DBMS는

- 데이터를 CRUD(Create, Retrieve, Update, Delete)하는 시스템

- 데이터를 저장(수정, 삭제)하고 검색하는 시스템

- 검색에 좀 더 최적화

- DB(Database)란 - 여러 개의 테이블을 묶어서 관리하는 개념(예전. 카탈로그)

- 여러 개의 테이블이 하나의 DB를 구성 - DBMS는 여러 개의 DB를 관리하는 시스템

- DBMS-Database-Table

- 테이블은 하나의 관계(relation)를 표현

- 테이블은 여러 개의 레코드(튜플)이 들어있슴

- 하나의 레코드는 여러 개의 컬럼으로 구성됨

테이블의 예



스키마(Schema)

- 테이블의 구조 및 테이블간의 관계를 정의

- 테이블은 스키마에서 정의된 대로만 레코드를 저장/관리

- 이 형식 외에는 다른 값은 허용하지 않음 -> 정형데이터

예) 주민등록번호(13자리) -> 14자리 번호는 허용되지 않음

모델링(Modeling)

- 요구사항에서 DB스키마를 만드는 작업

- 개념모델링/논리모델링/물리모델링로 나뉨

정규화

- DB스키마는 고정되어 있지 않고 계속 변경(업데이트)

- DB스키마의 설계를 재구성

- 불필요한 데이터를 삭제, CUD에 발생할 수 있는 이상현상을 제거(Anamolies)

- 불필요한 데이터를 삭제

- 논리적인 데이터저장구조를 가지도록 변경

- 데이터의 중복을 정규화(Normalization)를 통해 중복성제거

- 보통 1,2,3, BC(Boyce-Codd) 정규형(Normal Form)까지 적용한 것을 정규화(Normalize)했다고 말함

- 정규화를 하면 보통 테이블이 큰 테이블에서 여러 개의 작은 테이블로 분할됨

검색

- 일반적인 DBMS는 레코드 추가/수정/삭제보다 검색을 빠르게 만드는 것이 주요한 목표

- CUD(Create/Update/Delete) vs. R(Retrieve:Search)

- Write vs. Read

- 검색을 빨리하기 위해서 주로 인덱스(index)라는 개념을 사용

- DBMS는 다양한 종류의 인덱스지원

- B-tree/Hash/R-tree/Full-text index/…

- 가장 기본되는 인덱스는 B-tree(정확하게는 B+트리)

인덱스(B-tree)

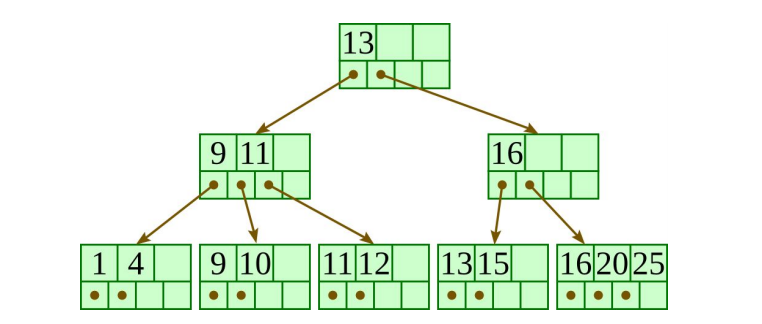
- 보통 B+ 트리를 말함(인덱스와 데이터 저장소 분리)

- 인덱스파일만별도로 구성하고 데이터 파일은 참조형태로

- 바이너리 검색(Binary Search)의 변형

- 인덱스는 CUD(Insert/Update/Delete)가 일어나면 재구성

- 정렬 후 인덱스 업데이트

- 인덱스를 사용하는 경우에도 느려지는 경우는?

- 저장되는 데이터가 많아지면 B-tree의 Depth(높이)가 늘어나게 되서 검색속도가 느려짐

- 해결책 -> 파티셔닝 / 샤딩

외래키(Foreign Key)

- 어떤 테이블의 기본키가 다른 테이블에 속해 있는 컬럼

- FK가 PK인 경우 - 식별관계(Identifying Relationship)

- FK가 일반 컬럼인 경우

- 비식별관계(Non-identifying Relationship)

- 두 테이블 간의 공통 컬럼 - 테이블 조인시 기준이 되는 컬럼

조인

- 여러 테이블을 합해서 큰 테이블로 만드는 연산

- 테이블이 여러 개로 나눠진 상태에서 원하는 데이터를 찾으려면 테이블을 합쳐서(조인:JOIN) 필요한 데이터 검색

조인은 테이블단위 연산(Operation)임

- 레코드 개별적으로 하는 연산이 아님

- 조인과정에서각 테이블 간의 공통컬럼이 있어야 함(표준조인)

- 공통컬럼을 외래키라고 함