

데이터베이스 시스템 개념과 아키텍처

데이터 모델

- 데이터 모델
 - 데이터 타입, 관계, 제약 조건들을 명시하기 위해 사용될 수 있는 개념들의 집합
 - 데이터베이스에서 검색과 갱신을 수행하는 기본 연산들의 집합을 포함
- 점차 DB 응용의 동적 측면 또는 행동이 데이터 모델에 포함됨
 - 사용자 정의 연산(user defined operation)

데이터 모델의 분류

- 저수준 또는 물리적 데이터 모델
 - 어떻게 데이터가 컴퓨터에 저장되는지 세부 사항을 명시하는 개념을 제공
- 고수준 또는 개념적 데이터 모델
 - 사용자들이 데이터를 인식하는 방식에 대한 개념을 제공
- 표현(또는 구현) 데이터 모델
 - 고수준 모델과 저수준 모델 사이에 존재
 - 일반 사용자들이 이해할 수 있는 개념을 제공
 - 데이터 저장 구조의 세부 사항을 은폐하지만 컴퓨터 상에서 직접 구현 가능
 - 상용 DBMS에서 많이 사용함.

스키마, 인스턴스, 데이터베이스 상태

- 데이터베이스 스키마 (또는 메타데이터)
 - 데이터베이스에 대한 기술
 - 데이터베이스 설계 과정에서 명시하며 자주 변경되지 않음

- 스키마 다이어그램

- 데이터베이스 스키마를 도식화한 것
- 레코드 타입의 이름, 데이터 항목의 이름, 일부 제약 조건 유형들과 같은 **스키마의 일부** **관점**만을 나타냄.

- 데이터베이스 상태(Database State)

- **어커런스(Occurrence)**나 **인스턴스(Instance)**들의 집합이라고도 함
- 어떤 **특정 시점**에 데이터베이스에 들어 있는 데이터
- 데이터베이스에 **갱신 연산**이 수행될 때마다 **새로운 다른 데이터베이스 상태**를 가짐
- **DBMS**는 데이터베이스 상태가 **스키마에 명시된 구조와 제약조건**을 만족하는 **유효한 상태**임을 보장하는 책임을 일부 가짐

- 내포와 외연

- 일반적으로 **스키마**는 **내포(intension)**라 하고, **데이터베이스 상태**는 **외연(extension)**이라 함

➤ 데이터베이스 스키마 다이어그램

DB schema

STUDENT

Name	Student_number	Class	Major
------	----------------	-------	-------

COURSE

Course_name	Course_number	Credit_hours	Department
-------------	---------------	--------------	------------

PREREQUISITE

Course_number	Prerequisite_number
---------------	---------------------

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
--------------------	---------------	----------	------	------------

GRADE_REPORT

Student_number	Section_identifier	Grade
----------------	--------------------	-------

Figure 2.1

Schema diagram for the database in Figure 1.2.

3단계 – 스키마 아키텍처

- 3단계 – 스키마 아키텍처의 목적

- 사용자의 응용과 물리적 데이터베이스의 분리가 목적

1. 내부 단계

- 내부 스키마를 가지며, 내부 스키마는 물리적 데이터 모델을 사용
- 데이터 저장구조의 세부 사항과 데이터베이스에 대한 접근 경로를 기술

2. 개념 단계

- 개념 스키마를 가지며, 이는 전체 사용자를 위한 데이터베이스의 구조를 기술함
- 엔티티, 데이터 타입, 관계, 사용자 연산, 제약 조건들을 나타내는데 중점

3. 외부 단계 또는 뷰 단계

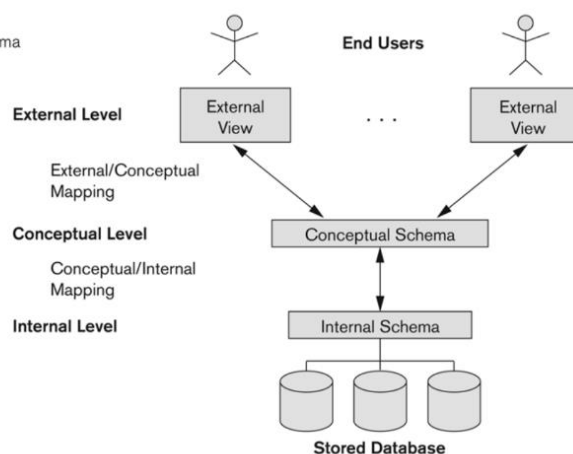
- 외부 스키마나 사용자 뷰들을 포함
- 특정 사용자 그룹이 관심을 갖는 부분을 나타내고 나머지는 은폐함

- 사상(Mapping)

- 외부 스키마를 참조하여 사용자가 데이터를 요구하면 이를 데이터베이스 내에서 개념 스키마에 대한 요구로 변환하고, 다시 내부 스키마에 대한 요구로 변환 과정을 거쳐 저장된 데이터베이스에 접근하여 데이터를 추출한 후 사용자의 뷰와 일치하도록 재구성하는 과정

➤ 3단계 스키마 아키텍처

Figure 2.2
The three-schema architecture.



DB – instance

OO – object

RDB – tuple, row

External view

- 표현 모델 (O)
- 물리적 모델 (X)
- 개념적 모델 (X)

데이터 독립성(Data Independence)

- 논리적 데이터 독립성 (Logical Data Independence)
 - 외부 스키마나 응용 프로그램을 변경하지 않으면서 개념 스키마를 변경할 수 있는 능력
- 물리적 데이터 독립성 (Physical Data Independence)
 - 개념 스키마를 변경하지 않으면서 내부 스키마를 변경할 수 있는 능력

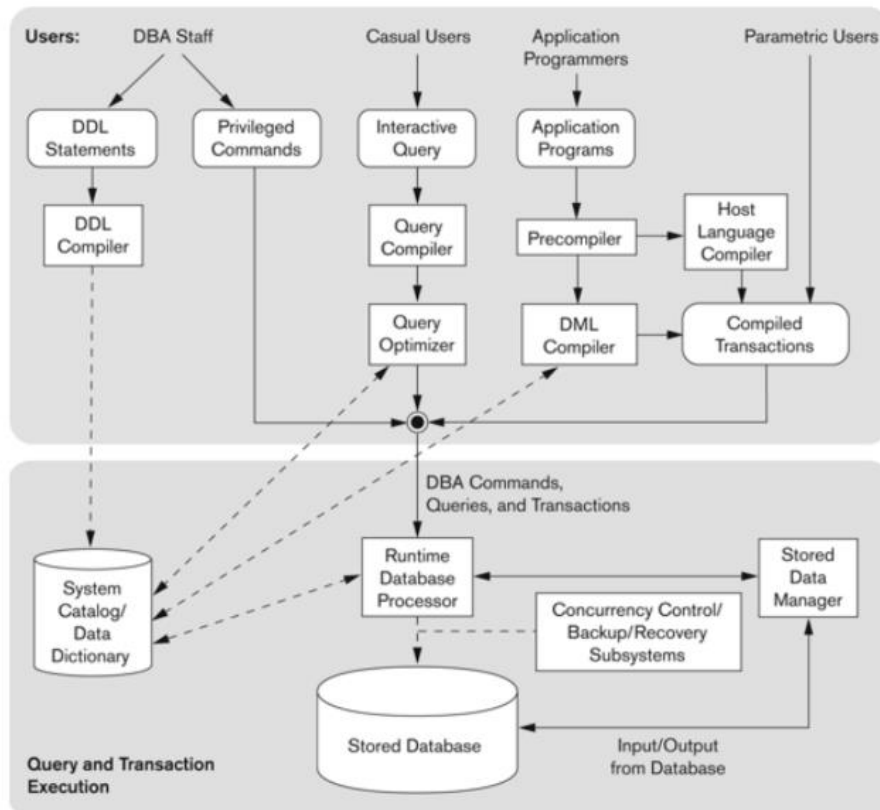
3단계 스키마 아키텍처를 가져가는 이유 – 데이터 독립성 보장

DBMS 언어

- 데이터 정의어(DDL : Data Definition Language)
 - 개념 스키마와 내부 스키마를 정의
- 저장구조 정의어와 뷰정의어
 - 어떤 DBMS에서는 저장구조 정의어(SDL : Storage Definition Language)를 사용하여 내부 스키마를 나타내고, 뷰 정의어(VDL : View Definition Language)를 사용하여 뷰를 명시하거나 개념 스키마 사이의 사상을 나타냄
- 데이터 조작어(DML : Data Manipulation Language)

- 데이터를 검색, 삽입, 삭제, 수정하기 위한 조작 언어 (CRUD)
- DML 명령어는 범용 프로그래밍 언어에 삽입되어 사용될 수 있고, 이때 범용 프로그래밍 언어를 호스트 언어라 하고, 삽입된 DML 명령어를 데이터 부속어(DSL)라 함

➤ DBMS 구성모듈



데이터베이스 시스템 유틸리티

- 데이터베이스 유틸리티

- DBMS는 데이터베이스 관리자의 데이터베이스 시스템 운영을 도와줌

- 적재

- 데이터 파일을 자동적으로 데이터베이스 **파일의 형식으로 변환해서 저장함**

- 백업

- 전체 데이터베이스를 **테이프에 복사**하여 데이터베이스의 백업 사본을 만들

- 파일 재조직

- 성능 향상을 위해 데이터베이스 파일 구조를 다른 파일 구조로 재조직함

- 성능 모니터링

- 데이터베이스의 사용을 모니터해서 사용 통계를 데이터베이스 관리자에게 제공함
- 이 정보는 관리자가 데이터베이스 성능을 향상시키기 위해서 파일들을 재조직할 것인지를 결정하는데 사용됨

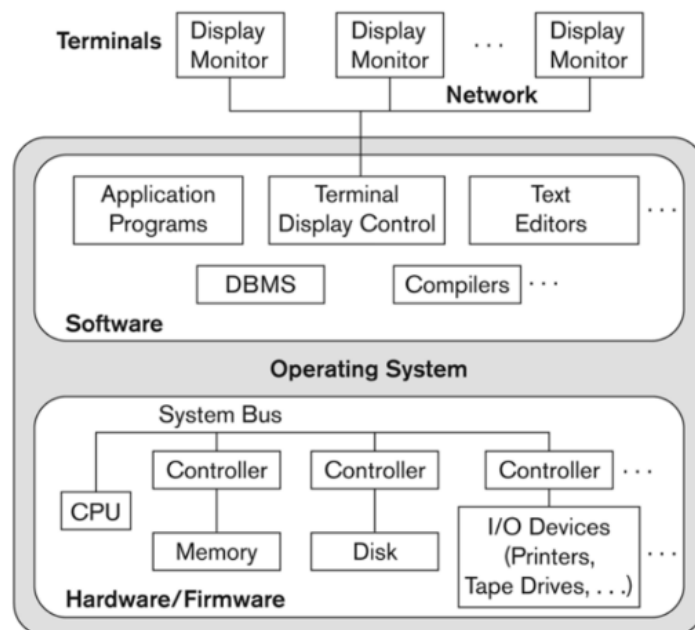
- 데이터 사전 시스템(data dictionary system)

- 스키마와 제약 조건들에 관한 카탈로그 정보와 설계 결정, 사용 표준, 응용 프로그램 기술, 사용자 정보 등과 같은 정보를 저장

DBMS를 위한 중앙집중식과 클라이언트/서버 아키텍처

- 중앙집중식 DBMS 아키텍처

- 메인프레임과 터미널을 사용
- 터미널이 PC와 워크스테이션으로 대체됨

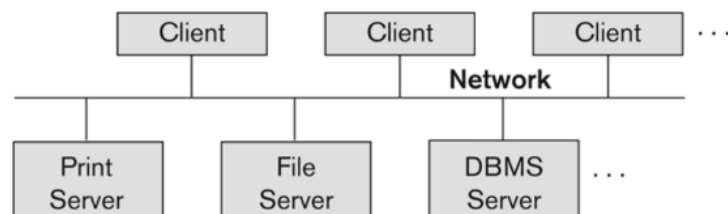


- 클라이언트-서버 아키텍처

- 특정 기능을 갖는 특별한 서버를 지정
- 파일 서버, 프린터 서버, 웹 서버, 전자 우편 서버 등
- 클라이언트 : 지역 응용들을 수행하기 위한 처리 기능뿐만 아니라 서버들을 이용하기 위한 인터페이스

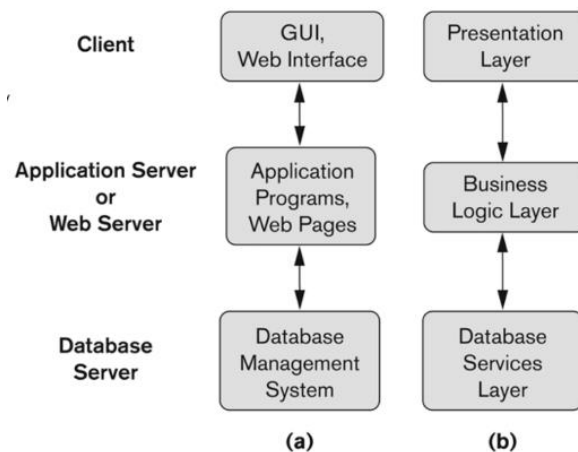
- 2층 클라이언트-서버 아키텍처

- 중앙집중식 RDBMS가 클라이언트-서버 아키텍처로 바뀌고 있음
- 서버 : 질의와 트랜잭션 기능. 질의 서버(트랜잭션 서버), SQL 서버(RDBMS의 경우)
- 클라이언트 : User interface program과 App, Program 수행
- ODBC(Open Database Connectivity) API, JDBC
- 서버 단계 : 데이터 저장, 동시성 제어와 회복, 버퍼링(캐싱) 등
- 클라이언트 단계 : 사용자 인터페이스, 데이터 사전, 버퍼 내의 데이터로부터 복잡한 객체를 구성



- 웹 응용들을 위한 3층 클라이언트-서버 아키텍처

- 클라이언트와 데이터베이스 서버 사이에 응용 서버 또는 웹 서버를 추가
- 이 서버는 데이터베이스 서버에 저장된 비즈니스 규칙(프로시저 또는 제약조건)들을 저장함으로써 중간 역할을 수행함
- 비즈니스 규칙은 일반적으로 데이터를 접근하는데 사용됨



DBMS의 분류

● DBMS의 분류 기준

■ 데이터 모델 : 관계, 객체지향, 객체관계 등

◆ Legacy : 네트워크, 계층

◆ Currently : 관계, 객체지향, 객체관계

◆ Recently, NOSQL systems :

- document-based : JSON을 데이터 모델로 사용
- column-based : 레코드의 컬럼 단위로 저장
- graph-based : 객체와 관계를 저장
- key-value based : 키와 객체로 구성. 키가 주어지면 빠르게 객체에 접근
- Native XML DBMS : XML 모델을 데이터 모델로 사용

■ 사용자의 수 : 단일 사용자, 다수 사용자 시스템

■ 사이트의 수 : 중앙집중식, 분산 DBMS (동질 분산 DBMS 또는 이질 분산 DBMS)

■ 범용 또는 특수 목적용

● 관계 모델

- 데이터베이스는 테이블들의 모임으로 구성

- 고급 질의어를 제공하고 제한된 형태의 사용자 뷰를 지원

- 네트워크 모델

- 데이터를 레코드 타입들로 나타냄

- 계층 모델

- 데이터를 계층적 트리 구조로 나타냄

- 객체지향 모델

- 객체, 객체의 속성, 연산으로 데이터베이스를 정의
- 같은 구조와 행위를 갖는 객체들은 한 클래스에 속하고 클래스들은 계층 또는 비순환 그래프로 조직됨
- 메소드라고 하는 미리 정의된 프로시저들이 클래스의 연산을 나타냄

- 객체-관계 모델

- 관계 모델에 객체지향 모델의 개념을 도입하여 확장함