

database-1

☰ 태그	
📅 날짜	@2023년 6월 8일

DB & DBMS & DB system

database(DB)

전자적으로(electronically) 저장되고 사용되는 **관련있는 (related) 데이터들의 조직화된 집합 (organized collection)**

관련있는 데이터들 : 인스타그램, 페이스북 에서 사진을 올리고 글을 쓰고 댓글을 달고 하는 일련의 정보들. 이런 데이터들이 모두 하나의 서비스와 관련된 데이터들이다. 그 외 로그인 하기 위해서 나의 개인정보를 입력했다면 그 개인정보 또한 이 서비스와 관련된 데이터들이 될 것이다.

관련있는 데이터라고 할때의 데이터의 의미는 같은 출처나 혹은 같은 목적 같은 서비스안에서 서비스 되는 데이터들을 **관련있는 데이터**라고 한다.

조직화된 집합 : 이 **관련있는 데이터들**을 **조직화된 집합**으로 묶어야 한다. 중요한 것은 **조직화**라는 개념이다. 데이터가 막 쌓여있다면 내가 찾는 데이터를 찾기가 힘들것이다. 또 같은 제목이 여러개라면 재고관리도 힘들 것이다. 하지만 잘 조직화 되어있는 데이터라면 내가 찾는 데이터를 좀 더 빨리 찾을 수 있으며, 불필요한 데이터가 중복해서 생성되는 것을 찾을 수 있고 데이터의 불일치 또한 막을 수 있다. 그러므로 **조직화된 집합**으로 데이터가 관리 되는것이 중요하다.

전자적으로 저장하고 사용 : **관련 있는 데이터들**을 **조직화된 집합**의 형태로 전자적으로 저장하고 사용할 때 이것을 데이터베이스라고 한다. 데이터가 컴퓨터에 저장이 되고 또 사용이 될 때 이것을 전자적으로 저장된다고 또 사용된다고 볼 수 있다.

DBMS - database management systems

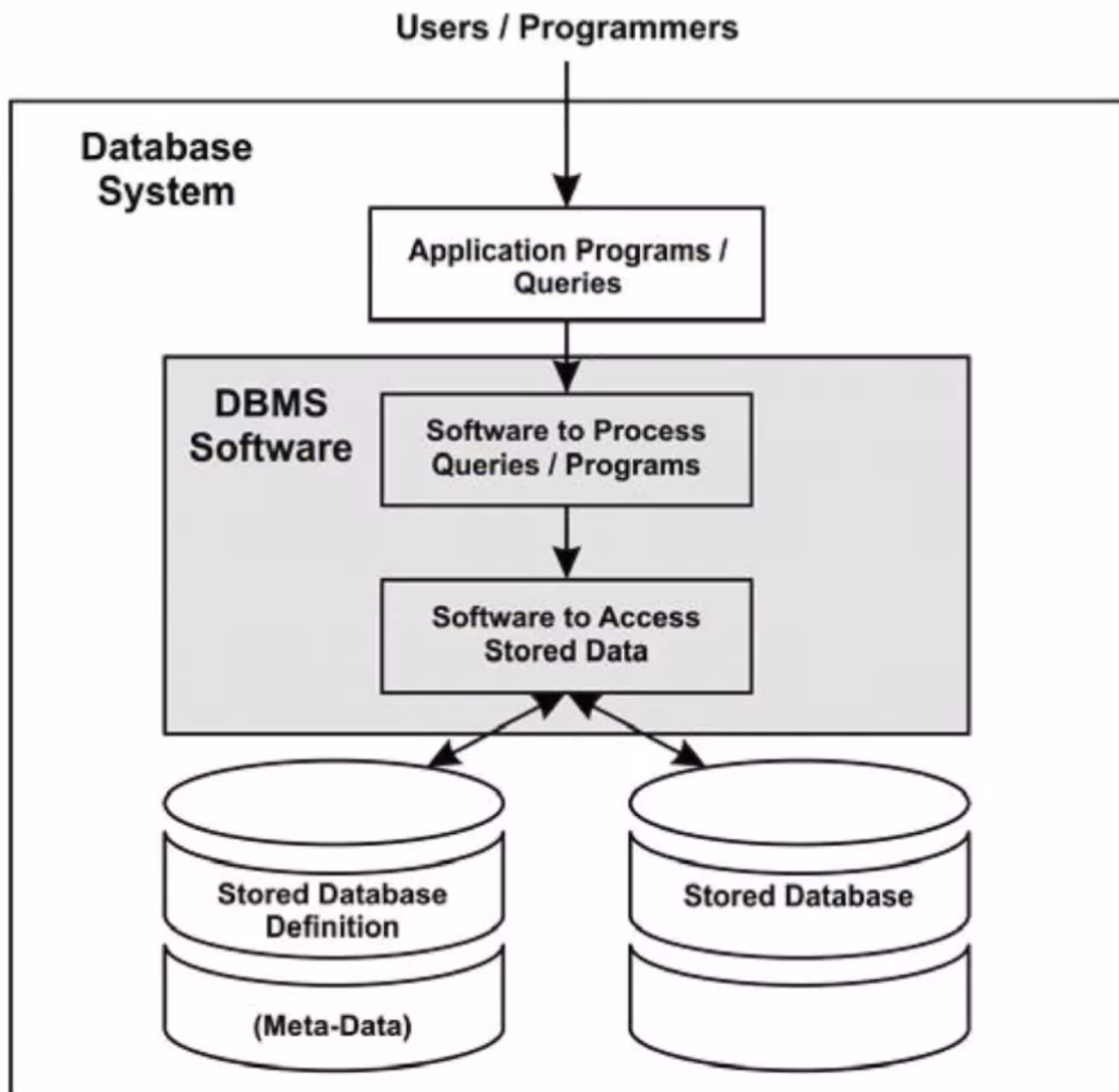
- 사용자에게 DB를 정의하고 만들고 관리하는 기능을 제공하는 소프트웨어 시스템
- 대표적인 DBMS - PostgreSQL, MySQL, oracleDB, SQLServer 가 있다.
- DB를 정의하다 보면 부가적인 데이터가 발생한다.

부가적인 데이터 : metadata

- database를 정의하거나 기술하는(descriptive) data
- catalog라고 부른다. 데이터베이스의 메타데이터가 저장되는 곳을 catalog라고도 한다.
- e.g.) 데이터 유형, 구조, 제약 조건, 보안, 저장, 인덱스, 사용자 그룹 등등
- metadata 또한 DBMS를 통해 저장/관리된다.
- 원래의 metadata란 data about data 라는 의미로 데이터를 설명하는 데이터란 의미

database system

- database + DBMS + 연관된 applications
- 줄여서 database라고도 부른다.
- 문맥에 따라서 잘 이해해야한다.



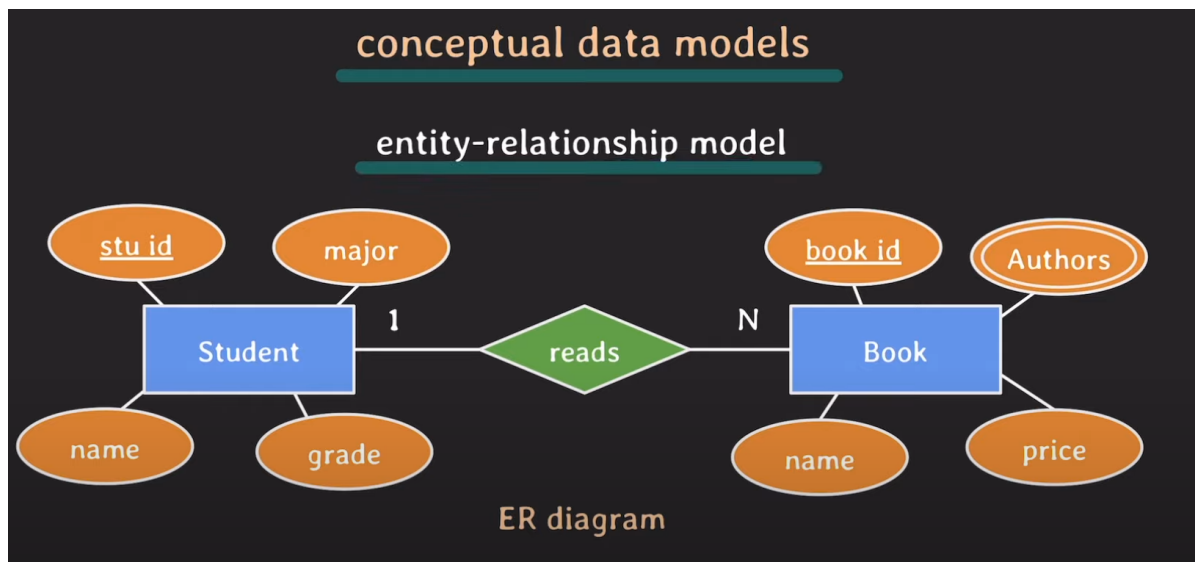
data models

- DB의 구조(structure)*를 기술하는데 사용될 수 있는 개념들이 모인 집합
- DB 구조*를 추상화해서 표현할 수 있는 수단을 제공한다.
- data model은 여러 종류가 있으며 추상화 수준과 DB구조화 방식이 조금씩 다르다.
- DB에서 읽고 쓰기 위한 기본적인 동작들(operations)도 포함한다.

*DB구조 : 데이터 유형, 데이터 관계(relationship), 제약 사항(constraints) 등등

data models 분류

- conceptual (or high-level) data models
 - 일반 사용자들이 쉽게 이해할 수 있는 개념들로 이루어진 모델
 - 추상화 수준이 가장 높음
 - 비즈니스 요구 사항을 추상화하여 기술 할 때 사용
 - 대표적인 모델 ER diagram



- logical (or representational) data models
 - 이해하기 어렵지 않으면서도 디테일하게 DB를 구조화 할 수 있는 개념들을 제공
 - 데이터가 컴퓨터에 저장될 때의 구조와 크게 다르지 않게 DB구조화를 가능하게 함
 - 특정 DBMS나 storage에 종속되지 않는 수준에서 DB를 구조화할 수 있는 모델
 - 백엔드 개발자가 가장 많이 사용하게 된다.
 - 대표적인 모델 relational data model
 - row는 데이터 각각을 의미
 - column은 데이터의 속성을 의미

- 그 외 종류
 - object data model
 - relational 과 object를 합친 object-relational data model
- MySQL, oracleDB, SQLServer 등이 relational data model 이다.
- PostgreSQL 는 object-relational data model이다.

logical data models 종류

relational data model

STUDENT

student id	이름	전공	학년
sid_2022022	홍진호	스타	4
sid_2022023	손흥민	축구	3
sid_2022032	최민정	쇼트트랙	2

- physical (or low-level) data models
 - 컴퓨터에 데이터가 어떻게 파일 형태로 저장되는지를 기술할 수 있는 수단을 제공
 - 데이터가 실제로 저장 장치에 저장된 형태의 가장 근접하게 데이터베이스 구조를 표현할 수 있는 데이터 구조이다.
 - data format, data orderings, access path 등등
 - access path : 데이터 검색을 빠르게 하기 위한 구조체. ex) index

database schema

- data model을 바탕으로 database의 구조를 기술(description)한 것
- schema는 database를 설계할 때 정해지며 한번 정해진 후에는 자주 바뀌지 않는다.

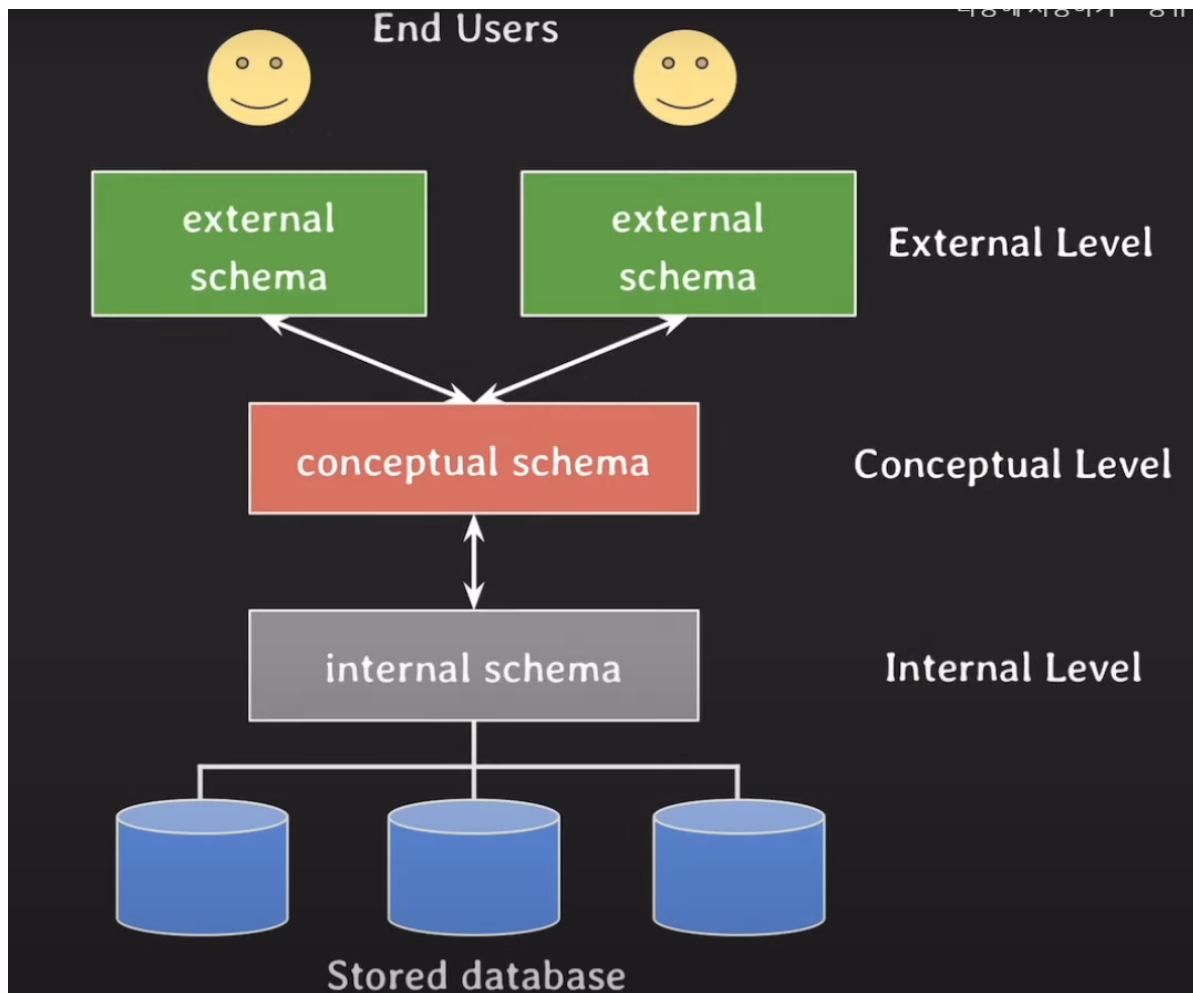
database state

- database에 있는 실제 데이터는 꽤 자주 바뀔 수 있다.
- 특정 시점에 database에 있는 데이터를 database state 혹은 snapshot이라고 한다.
- 혹은 database에 있는 현재 instances의 집합이라고도 한다.
- 시간의 흐름에 따라서 데이터가 바뀌는데 임의의 어떤 시점에서 그 때의 데이터베이스의 데이터들을 database state라고 한다. 여러 시점에서 database state는 같을 수도 있고 다를 수도 있다.

three-schema architecture

- database system을 구축하는 architecture 중의 하나
- user application으로 부터 물리적인(physical) database를 분리시키는 목적
 - 물리적인 데이터베이스의 구조가 바뀌더라도 실제로 사용하는 유저 어플리케이션에는 영향을 끼치지 않도록 하기 위해 사용
- 세 가지 level이 존재하며 각각의 level마다 schema가 정의 되어 있다.
 - **external schemas** (or user views) at **external (or view) level**
 - external views, user views라고도 불림
 - 특정 유저들이 필요로 하는 데이터만 표현
 - 그 외 알려줄 필요가 없는 데이터는 숨김

- logical data model을 통해 표현
- 실제 사용자가 바라보는 schema
- dbsystem의 초창기 아키텍처는 external shemas와 internal schemas 밖에 없었는데 그로인해 생긴 문제가 각각의 유저마다 필요로 하는 데이터들이 달라지다 보니 이 internal level에 중복된 데이터가 생기게 되었다. needs에 맞춰서 데이터를 제공하려 보니 같은 데이터를 포함해도 조금씩 다른 internal schema들이 여러 개 생김. 점점 관리가 힘들어지고 데이터 불일치가 발생. 그래서 등장한것이 conceptual shemas
- conceptual schemas at conceptual level
 - 전체 database에 대한 구조를 기술
 - 물리적인 저장 구조에 관한 내용은 숨김
 - entities, data types, relationships, user operations, constraints에 집중
 - logical data model을 통해 기술
 - internal schemas를 한번 추상화 시켜서 표현한 schema라고 이해하면 된다.
- internal schemas at internal level
 - 물리적으로 데이터가 어떻게 저장되는지 physical data model을 통해 표현
 - data storage, data structure, access path 등등 실체가 있는 내용 기술
 - 물리적인 저장장치 가장 가까이에 위치



three-schema architecture

- 안정적으로 데이터시스템을 운영하기 위한 사용되는 아키텍처
- 각 레벨을 독립시켜서 어느 레벨에서의 변화가 상위 레벨에 영향을 주지 않기 위함
- 대부분의 DBMS가 three level을 완벽하게 혹은 명시적으로 나누지는 않음
- 데이터가 존재하는 곳은 internal level
- 가령 internal schema가 바뀌더라도 conceptual schema에 영향이 없다. 그 사이에 mapping만 바꿔주면 된다.
- internal schema가 바뀌더라도 conceptual은 바뀌지 않게 하는것은 상대적으로 쉬운데 conceptual가 바뀌어도 external이 바뀌지 않게 하는 것은 상대적으로 까다롭다.

database language

data definition language(DDL)

- **conceptual schema**를 정의하기 위해 사용되는 언어
- internal schema까지 정의할 수 있는 경우도 있다. (internal은 보통 파라미터로 정의한다.)

storage definition language(SDL)

- **internal schema**를 정의하는 용도로 사용되는 언어
- 요즘은 특히 relational DBMS에서는 SDL이 거의 없고 파라미터 등의 설정으로 대체됨

view definition language(VDL)

- **external schemas**를 정의하기 위해 사용되는 언어
- 대부분의 DBMS에서는 DDL이 VDL 역할 까지 수행

data manipulation language (DML)

- **database에 있는 실제 data**를 활용하기 위한 언어
- data 추가, 삭제, 수정, 검색 등등의 기능을 제공하는 언어

통합된 언어

- 오늘날의 DBMS는 DML, VDL, DDL이 따로 존재하기 보다는 통합된 언어로 존재

- 대표적인 예가 relational database language : SQL