

1주차 강의 요약

1. Data: 단순한 값.
2. Information: 인간에게 의미가 있는 정보. (가공된, 처리된 정보)
3. 정보 처리: 데이터에서 정보를 추출하는 과정 및 방법
4. 형태에 따른 데이터의 분류
 - A. 정형 데이터: 구조가 정해져 있다.
 - ① RDBMS의 테이블
 - B. 반정형 데이터: 내용 안에 구조에 대한 설명이 같이 있다.
 - ① HTML, XML, JSON 등
 - C. 비정형 데이터: 정해진 구조가 없다.
 - ① 영상, 이미지, 음성
 - ② 형식 없는 순수한 텍스트 문서
5. 특성에 따른 데이터의 분류
 - A. 범주형 데이터: 범주(category)로 구분 가능한 값 들.
 - ① 명목형 데이터: 서열이 없는 값.
 - ② 순서형 데이터: 서열이 있는 값.
 - B. 수치형 데이터: 크기 비교와 산술적인 연산이 가능한 숫자 값.
 - ① 이산형 데이터: 단절된 숫자 값.
 - ② 연속형 데이터: 연속된 숫자 값.

1주차 강의 요약 끝.

2주차 강의 요약

1. File system

- A. 데이터를 파일로 관리하기 위해 파일을 생성, 삭제, 수정, 검색하는 기능을 제공하는 소프트웨어를 말한다.
- B. 옛날엔 전문화된 DB 없이, File system을 직접 만들어서 사용했다.
- C. 문제점
 - ① 응용 프로그램이 데이터 파일에 종속적이다.
 - ② 동일 내용의 데이터가 여러 곳에 중복 저장될 수 있다.
 - ③ 데이터의 복구가 어렵고, 무결성을 유지하기 어렵다. (중복 저장된 데이터를 모든 장소에서 동일하게 유지하기 어렵다.)

2. DBMS

- A. 파일 시스템의 문제를 해결하기 위한 데이터베이스 관리 시스템.
- B. 'SQL'이란 표준에 따라서 만들어진다.
- C. API 형태로 프로그래머가 DB에 접근 가능하다. (응용 프로그램의 어느 부분에서도 DBMS와 통신 가능.)
- D. 주요 기능
 - ① 정의 기능: DB 구조를 정의, 수정 가능.
 - ② 조작 기능: 데이터를 삽입, 삭제, 수정, 검색하는 연산.
 - ③ 제어 기능: 데이터 무결성 유지, 장애의 회복, 사용자 권한 제어 등.
- E. 장점
 - ① 데이터 중복을 통제 가능
 - ② 데이터 독립성 확보
 - ③ 데이터를 동시 공유 가능 (동시 접근 제어 기술 적용)
 - ④ 데이터 보안 향상 (접근 제어 가능, 권한 기반의 요청 차단 기능)
 - ⑤ 데이터 무결성 유지 가능 (모든 연산에 대하여 유효성 검사 가능
-> 예: 주민번호가 숫자로만 입력되었는가?)
 - ⑥ 표준화된 방식으로 DB에 접근하도록 함
 - ⑦ 장애 발생 시 회복이 가능하다 (commit 이전으로 복귀)
 - ⑧ 응용 프로그램 개발 비용 감소 (File system을 처음부터 설계할 필요 없어짐 -> 비용 감소)
- F. 단점
 - ① 고비용(DBMS가 비싸다)
 - ② 백업, 회복 방법이 복잡하다
 - ③ 중앙집중식 구조로부터 기인하는 취약점 존재 (서버 다운시 모든 응용 프로그램 중단 가능)

3. DBMS 발전 과정

A. 1세대: 계층 DBMS / 네트워크 DBMS

- ① 계층: 트리로 구현됨
- ② 네트워크: 그래프로 구현됨

B. 2세대: 관계 DBMS (RDBMS)

- ① 테이블로 구현됨
- ② Access / MariaDB / Oracle ...

C. 3세대: 객체지향 DBMS / 객체관계 DBMS

- ① 객체지향: 객체로 구현됨
- ② 객체관계: 객체지향 + 관계형(테이블)
 - 1. Oracle이 대표적이다. 관계형인데, 객체개념도 포함되어 있는 것.

D. 4세대: NoSQL / NewSQL DBMS

- ① NoSQL의 경우
 - 1. 비정형 데이터 관리를 위해 사용하는 DB
 - 2. 여러 대의 컴퓨터에 분산 저장하고 처리하는 환경에서 주로 사용.
(빅데이터 연산,
영상 스트리밍 시스템-넷플릭스의 국가별 서버- 등)
- ② NewSQL DBMS = NoSQL + RDBMS <- 너무 복잡해서 잘 안쓴다.

4. 데이터베이스 시스템

A. DBS(DataBase System): DBMS + 사용자 + 하드웨어

- ① 사용자를 포함하는 DB 시스템인 것!
- ② DBS에 DBMS가 포함된다는 것이 이상해 보일 수 있으나, DBMS는 어디까지나 DB와 사용자 사이를 매개하는 소프트웨어이므로, 넓은 의미의 DBS에 포함된다.

B. 스키마(Schema): DB에 저장되는 데이터 구조와 제약조건을 정의한 것.

C. 인스턴스(Instance): 스키마에 따라 DB에 실제로 저장된 값.

5. 3단계 DB 구조

- = DB를 효과적으로 관리하기 위하여, 3개의 관점으로 시스템을 바라보는 것.
- = 각 단계의 주체에게 서로 다른 수준으로 추상화된 인터페이스를 제공 가능.
- A. 외부 단계: 개별 사용자의 관점에서 바라보는 DB
 - > 외부 스키마. 자신이 생각하는 DB의 모습 정의.
 - = 프로그래머가 보는 인터페이스의 기반.
- B. 개념 단계: 조직 전체의 관점에서 바라보는 DB
 - > 개념 스키마. 모든 사용자들의 관점을 통합하여, 조직 전체의 관점인 '개념 스키마' 정의.
 - = DB 관리자가 보는 인터페이스의 기반.
- C. 내부 단계: 물리적인 저장 장치의 관점에서 바라보는 DB
 - > 내부 스키마. 물리적인 저장 장치에 어떻게 저장될 지를 정의.
 - = DB 내부.

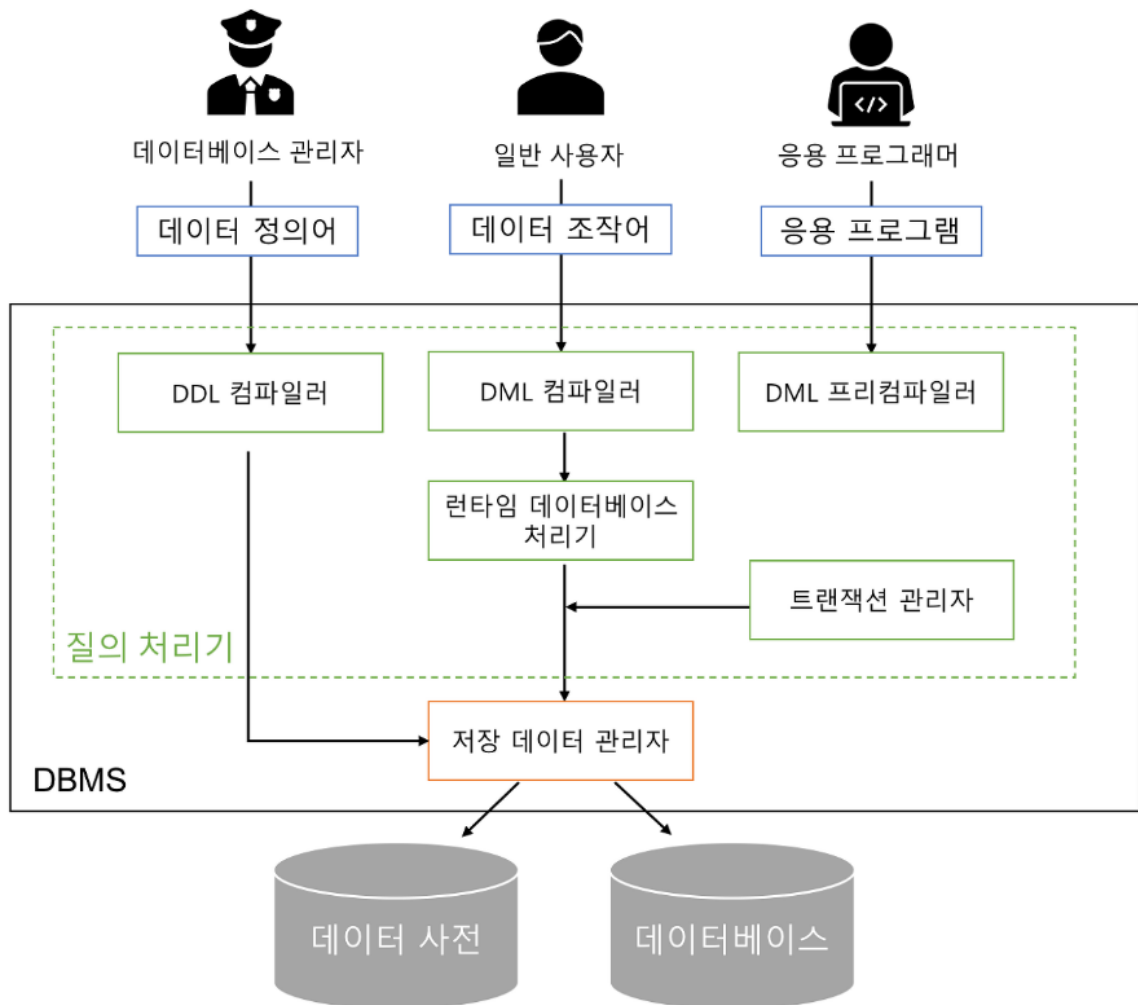
6. 3단계 DB 구조의 사상

- A. 각 단계의 주체 간에는 서로의 구조를 모른다. 그러므로 간섭할 수 없고, 간섭할 이유도 없다. <- 독립성 유지.
 - ① 외부 <-> 개념: API로만 통신함.
 - ② 개념 <-> 내부: 저장 인터페이스로 통신. (즉, 이 인터페이스를 수정하는 것으로, 내부 스키마가 어떻게 변해도 개념 단계에서는 어떤 변화도 인지하지 못함.)
- B. 데이터 독립성이 보장된다. (하위 스키마를 변경해도 상위 스키마의 변동 없음. 이때 상위는 외부, 하위는 내부이다.)
 - ① 외부 <-> 개념: 논리적 독립성 보장.
(전체 DB 구조가 바뀌어도, 외부 사용자는 인지 불가.)
 - ② 개념 <-> 내부: 물리적 독립성 보장.
(하드웨어가 바뀌어도 개념 스키마의 변동 X)
- C. 데이터 사전(Data dictionary, System catalog)
 - ① DB에 저장되는, 수정 불가능한, "DB에 저장되는 데이터"에 대한 정보를 저장하는 장소. (데이터 디렉터리에 저장됨.)
 - ② 시스템 DB라고도 한다. (대응되는 용어: 사용자 DB)

7. 데이터 언어

- A. DDL(데이터 정의어): 스키마 정의/수정/삭제를 수행하는 언어.
- B. DML(데이터 조작어): 데이터 삽입/삭제/수정/검색 등의 처리를 요구하는 언어. (절차적 언어-처리방식 명시-와, 비절차적 언어-처리방식 명시 X-로 나뉜다.)
- C. DCL(데이터 제어어): 규칙이나 기법을 적용하는 언어.
 - ① 무결성: 제약조건 검사.
 - ② 보안: 접근권한 제어.
 - ③ 회복: 회복 방법 명령.
 - ④ 동시성: 동시 접근 가능 사용자 수 제한 명령.

8. DBS의 구성 (DBMS + 사용자 + 하드웨어)



< DBMS 구성 >

- A. (<https://noahlogs.tistory.com/39> 에서 가져옴)

2주차 강의 요약 끝. (3장까지 정리 끝.)

3주차 강의 요약

1. 데이터 모델링

A. 현실에 존재하는 데이터를 컴퓨터의 DB로 옮기는 변환 과정을 의미함.

B. 과정:

- ① 데이터 모델링의 대상을 추상화. (개념적 모델링)
- ② 추상화된 특징으로 DB 구조 결정. (논리적 모델링)
- ③ DB에 저장.

2. 데이터 모델의 구성요소

A. 데이터 구조

- ① 현실 세계를 추상화 했을 때, 어떤 요소로 이루어져 있는지를 표현하는 논리적 구조이다.
- ② 자주 변하지 않는다: 정적이다.

B. 제약조건

- ① 무결성 유지를 위한 구조적 제약조건
- ② 연산 적용시 허용 가능한 범위를 정하는 제약 사항

C. 연산

- ① 값들을 처리하는 작업.
- ② 연산에 의해 값이 계속 변한다: 동적이다.

3. E-R 모델

A. 현실세계를 개념적 구조로 표현하는 방법 중 하나이다.

B. 구성요소:

- ① 개체(entity)
- ② 속성(attribute)
- ③ 관계(relationship)

더 자세한 내용은 3주차 강의 기록물을 참고.

교재 p.94~152의 내용은 생략하겠다.

3주차 강의 요약 끝.

4주차: 휴강.

5주차: 관계 대수 <- 정리 생략. (SQL에서 사용할 질의문의 수학적 표현 중심)

6주차 ~ 7주차: SQL 질의 + 뷰.

개념 정리는 종료되었다.

관계 대수는 이해만 하고 넘어가면 됨.

6~7주차는 교재 기반으로 실제 SQL 쿼리문을 작성할 수 있어야 하므로, 정리가 무의미.