1. 데이터베이스 시스템의 이해

1. 데이터베이스의 태동 & 2. 데이터베이스 시스템의 개요

1. 데이터베이스의 태동

- 정보 과부하(information overload)
- SCM(Supply Chain Management)
- CRM(Customer Relationship Management)
- 데이터의 양적 증가로 데이터 관리에 요구되는 시간과 비용이 증가하여 별도의 관리 장치가 필요하게 됨

2. 데이터베이스 시스템의 개요

- 데이터를 검색하고 이용하는 데 사용되는 정보 시스템의 구조 내부에는 반드시 DB라는 소프트웨어가 존재
- 데이터 베이스
  - '데이터의 집합'
- 데이터 관리 및 사용으로 업무를 자동화하는 프로그램은 기업의 비용을 최소화하고 의사결정에 필요한 시간을 단축할 수 있음
- 데이터베이스 관리 시스템 (DBMS: Database Management System)
  - 한 조직의 연관된 데이터 집합을 다수의 사용자가 공용으로 사용하기 위해 통합 저장하는 소프트웨어 패키지
- 데이터베이스 시스템
  - DBMS와 함께 사용자에게 서비스 형태로 제공되는 애플리케이션이 포함된 일체의 시스템

3. 데이터베이스 관리 시스템의 목적 [6]

개요

- 1960s, 데이터를 처리하고 분석하여 조직의 의사결정에 필요한 정보를 생산하기 위한 목적으로 도입된 컴퓨터 정보를 기록/관리하기 위해 당시 사용할 수 있었던 유일한 수단
  - 운영체제에서 지원하는 파일(FILE)
- 파일처리 시스템 (file processing system)
  - 파일을 사용하여 특정 업무에 해당하는 데이터를 관리하는 방식
  - 운영체제의 지원으로 조직에서 업무처리에 사용되는 데이터를 여러 파일에 나누어 영구 저장하고 운영하는 시스템
    - 운영체제에 의해 지원되며, 운영체제는 여러 파일에 레코드(데이터 단위)를 기록하고 레코드를 조작하기 위한 별도의 프로그램을 필요로 함
    - 프로그램에 지정된 파일에서 레코드를 추출하고 신규 레코드를 삽입하는 등의 작업을 수행
  - 애플리케이션 프로그램(application program)
    - if 신규 전공 개설 -> 학사 시스템은 새로운 학과에 해당하는 개별 파일 생성, 해당 학과에 등록되는 강사/학생, 과목 및 학위에 대한 정보와 내규를 기록
  - 데이터 문제 존재: 데이터의 종속, 데이터의 중복, 데이터의 무결성 훼손 및 동시 접근 이상 등

1. 데이터의 종속

- 파일 처리 프로그램에서는, 프로그램에 종속되는 파일이 생성됨
  - 데이터가 특정 프로그램에 종속될 경우, 데이터의 논리적 구조나 위치는 해당 프로그램만 알고 있어 다른 프로그램과의 데이터 공유도 불가능해지며, 데이터 구조의 변경이 프로그램의 변경으로 이어지기 때문에 프로그램 유지보수에 많은 비용이 소요
    - 데이터와 프로그램의 독립이 필요
- 데이터의 독립성 (data independency)
  - 논리적 데이터 독립성
    - 프로그래머가 생각하는 데이터의 논리적 구조가 변하더라도 그에 따른 프로그램의 구조가 변경되지 않는 것
  - 물리적 데이터 독립성
    - 보조기억장치와 같이 파일과 관련된 물리적인 시스템 구조가 변경되더라도 프로그램은 그대로 유지되는 것
  - 파일처리 시스템의 경우 데이터의 구조 및 위치 변경, 디스크 장치 또는 플랫폼 변경에 따른 데이터 표현 방식이 달라지기 때문에 프로그램 수정이 불가피함.
    - DBMS는 데이터를 프로그램과 완전히 독립시킴으로써 종속(dependency)으로 인한 문제를 근본적으로 배제시킴

2. 데이터의 중복

- 파일 처리 시스템, 프로그램들이 사용하는 정보가 여러 파일에 중복적으로 저장될 수 있다.
  - ex) 학생 (프로그램) 파일과 수강 (프로그램) 파일에 '학번', '이름', '전화번호'가 중복되어 나타날 수 있음. but 프로그램에서 요구하는 데이터 형식이 달라 파일 상호 공유 불가
- 데이터 중복성(data redundancy)
  - 하나의 사항에 대한 데이터가 여러 파일에 중복되어 저장되는 문제
  - 일관성(consistency) 문제

- 데이터 일관성
  - 하나의 사실을 나타내는 여러 개의 데이터가 모두 논리적으로 같은 값을 유지하는 것
- 비일관성(inconsistency)
  - but 하나의 사실에 대한 데이터가 물리적으로 서로 다른 장소에 위치하는 경우 일관성 유지가 매우 어려움 => 데이터 내용 간의 불일치가 발생하게 되는데, 이를 비일관성(inconsistency)라고 한다
- 보안성(security) 문제
  - 논리적으로 동등한 내용의 데이터에 대해서는 똑같은 수준의 보안이 유지되어야 함
  - but 같은 데이터가 중복되어 저장되는 경우, 동일한 수준의 보안을 유지하기 어려움
- 경제성(economy) 문제
  - 데이터를 중복 저장하기 위해서는 추가적인 저장 공간이 요구된다.
  - 중복 데이터가 존재하면 시스템 갱신 비용이 높아짐

3. 데이터 무결성 훼손
- 데이터 무결성(data integrity)
  - 데이터베이스에서 관리되는 데이터의 정확성을 보장하는 것
  - 데이터 무결성 훼손 발생
    - 하나의 사실을 표시하는 두 개 이상의 중복된 데이터가 서로 일치하지 않는 경우
    - 저장된 데이터 값이 일정한 형식의 조건들을 만족하지 않는 부정확한 데이터 저장으로 발생하는 경우
      - ex) 한 학기 최대 신청 학점 18학점 이하여야 하는 조건
        - 새로운 조건이 서로 다른 파일과 연관되는 경우 복수 개의 데이터 항목에 대한 무결성 유지 조건이 더욱 복잡해진다.

4. 동시 접근 이상
- 여러 사용자가 동시에 사용하는 시스템에서 한 데이터에 대한 수정 요구가 동시에 발생한 경우 비정상적인 데이터 수정이 일어날 수 있음
  - ex) 갱신 손실 등
  - ex) 은행의 입출금에서 예측할 수 없는 결과가 발생할 수 있음.
    - 데이터 동시 접근에 대한 통제 필요

4. 데이터베이스 관리 시스템의 특징
- DBMS는 파일 처리 시스템의 데이터 종속, 중복, 무결성 훼손 및 동시 접근 이상 등 데이터 관리 측면에서 문제 예방/해결을 위한 새로운 기능과 장치를 제공함.

1. 프로그램과 데이터의 독립성 및 추상화
- 파일 처리 시스템
  - 프로그램이 작업처리에 사용하는 데이터를 직접 데이터 파일에 접근하여 저장/사용하기 때문에 데이터 종속성이 발생
- DBMS
  - 프로그램-데이터 독립성 (program-data independency)
    - 데이터 사용과 데이터의 관리가 분리됨 데이터 파일의 구조가 프로그램으로부터 분리되어 DBMS 내부의 시스템 카탈로그에서 별도 관리되고 프로그램은 DBMS의 필요 데이터를 데이터의 의미만으로 요청할 수 있도록
  - 파일 처리 시스템에서 발생되는 데이터의 종속 및 중복 문제를 해결하고 공용으로 사용할 수 있도록 프로그램과 데이터를 중재하는 에이전트
  - 개념적인 표현(conceptual representation)
    - 데이터가 어떻게 물리적으로 저장장치에서 관리되는지 상세한 정보보다는 사용자가 데이터의 의미만으로 접근할 수 있도록 데이터에 대한 개념적인 표현(conceptual representation)을 제공함으로써 사용자가 좀 더 쉽게 데이터베이스를 사용할 수 있도록 한다.
    - 데이터 구조화를 위한 데이터 추성화(data abstraction) 작업으로 이루어짐

2. 자기 기술성
- 자기 기술성(self-describing)
  - 데이터 자체뿐만 아니라 데이터에 대한 정의나 의미까지 관리하는 것
  - 파일 처리 시스템
    - 데이터 정의가 프로그램의 일부 코드로 표현됨
  - 데이터베이스 시스템
    - DBMS는 메타데이터를 포함함
    - 메타 데이터(meta-data)
      - 각 파일의 구조, 각 데이터 항목의 타입과 저장 형식, 데이터의 다양한 제약조건
      - 시스템 카탈로그(system catalog) 또는 데이터 사전(data dictionary)에 저장됨

- 시스템 카탈로그는 특정 데이터베이스에 저장된 파일의 구조를 파악하기 위한 데이터의 타입이나 포맷과 같은 물리적 정보와 데이터의 의미, 설명 등의 논리적 정보를 관리함

3. 다중 뷰
   - 뷰(view)기능
     - DBMS는 사용자의 역할과 권한에 맞는 데이터에 접근할 수 있도록 데이터베이스에 대해 필요한 부분만을 추출해서 제공할 수 있는 뷰 기능을 지원
     - 파일 처리 시스템에서는 학생 파일-성적 파일 간 서로 중복되는 데이터를 별도로 관리하지만 DBMS는 중복 데이터 없이 데이터 일관성을 유지
4. 다수 사용자 요청 처리
   - 전체적인 처리성능을 향상시키고 빠른 응답시간을 얻을 수 있기 때문에 DBMS는 여러 사용자가 동시에 데이터베이스에 접근할 수 있는 기능을 제공
   - 트랜잭션(transaction)과 동시성 제어(concurrency control)
     - 단일 논리적인 작업을 수행하는 일련의 데이터베이스 명령의 집합
     - 동시 데이터 접근 이상과 같이 데이터에 대한 별도의 접근제어 없이 동일한 데이터를 대상으로 다수의 요청을 처리할 경우 데이터 일관성의 문제가 발생할 수 있는 문제를 해결
     - 다수 사용자가 동일한 데이터를 동시에 변경하는 경우에도 데이터 일관성을 보장할 수 있음

5. 데이터베이스 관리 시스템의 구조
   - 개요
     - DBMS는 다양한 유형의 사용자가 사용하는 시스템
       - 프로그래머: DB 구조, 데이터 의미, 타입, 길이 등 데이터와 메타데이터를 요구
       - 일반 사용자
         - 조건에 맞는 데이터 값을 요청
     - 일반사용자(=비전문가) 자신 목적에 부합하는 관점에서만 데이터 접근하는 것 선호
       - 내부 복잡한 구조를 감추어 데이터만을 보여 줄 필요성 있음
     - 데이터 독립성을 확보하기 위한 구조를 갖추어야 함
       - 사용자들이 필요로 하는 데이터 관점이 수시로 변경됨
   - 1. 데이터 추상화
     - 데이터 추상화(data abstraction)과 데이터 독립성(data independency)를 확보하기 위한 3단계 구조(3-level architecture)
       - 내부 단계(internal level)
         - 가장 낮은 추상화 단계, 내부 스키마에 의해 기술됨
         - 원시 수준(raw level)의 데이터 구조, 저장된 레코드 유형 정의, 인덱스(index)의 유무, 저장된 컬럼의 표현 방식, 저장된 레코드의 물리적 순서 등 구체적인 사항에 대해 정의함
         - DBMS는 하위 단계의 저장구조에 대한 구체적인 사항을 DB 프로그래머로부터 숨기는 역할
         - 학생, 과목, 또는 수강 레코드 등은 연속적으로 저장된 블럭(block)으로 표현됨
       - 개념 단계(conceptual level)
         - 데이터베이스 전체 구조를 추상화하는 단계, 개념 스키마(conceptual schema)를 통해 기술
         - 물리적 상세사항 등은 배제 DB에 무엇이 저장되어 있는지와 데이터 간의 관계만 기술
         - 여러 간단한 데이터의 구조로 전체 DB를 기술함 (ERD?)
         - 각 레코드의 타입과 레코드 간의 관계 정의로 표현 프로그래머는 이 단계의 추상화에서 프로그래밍 언어를 사용하여 작업
       - 외부 단계(external level)
         - 추상화의 최상위 단계로, 외부 스키마인 뷰(view)에 의해 기술
         - 사용자가 복잡한 시스템 지식 없이 시스템을 사용할 수 있도록 데이터를 추상화 한다.
         - 사용자는 데이터 타입의 구체성 없이 프로그램을 통해 데이터를 보게 됨
         - 뷰를 통한 보안 메커니즘 제공이 가능
   - 2. 단계 간 사상
     - 외부-개념 사상(external-conceptual mapping)
       - 논리적 데이터 독립성(logical data independence) 확보
       - 개념 스키마에 변화가 생기더라도 그 변화를 외부-개념 사상에만 반영시켜 주면 외부 스키마에 아무런 영향도 미치지 않는다.
     - 개념-내부 사상(conceptual-internal mapping)
       - 물리적 데이터 독립성(physical data independency)

- **- 물리적 저장 방식 변경: 인덱스 생성, 파일 구조 변경, 저장장치 변경 등 - 논리적 스키마 불변: 테이블 구조(컬럼, 관계 등)는 변경 없음 - 응용 프로그램 영향 없음: SQL이나 코드 수정이 필요 없음**
- **인덱스 추가, 데이터 파일 분산 저장, RAID 구성 변경**
- **다른 디스크로의 데이터 이동이나 파일 구조 변경 등의 물리적 변화가 발생해도 그 변화를 개념-내부 사상에만 반영시켜주면 개념 스키마에는 아무런 영향도 미치지 않음**

6. 데이터베이스 언어

- **개요**
  - **DBMS는 사용자가 데이터가 저장된 파일에 직접 접근할 수 없도록 데이터의 사용과 관리를 분리함으로써 파일 처리 시스템의 문제를 극복함**
  - **DBMS에 요청하기 위한 언어 형태의 인터페이스를 제공**
    - **데이터 정의 언어(DDL: Data Definition Language)**
    - **데이터 조작 언어(DML: Data Manipulation Language)**
- **1. 데이터 정의 언어**
  - **데이터 정의 언어(DDL)**
    - **DB 스키마와 데이터에 대한 부가적 특징을 표현**
  - **3가지 요건**
    - **프로그램이 요구하는 데이터 논리적 구성이나 특징을 데이터 모델에 의거하여 규정**
    - **데이터가 기억장치(storage)에 저장되도록 데이터 물리적 구성을 규정**
    - **DBMS가 물리적 구성을 논리적 구성으로 변환할 수 있도록 데이터 물리적 구성과 논리적 구성 간의 사상을 규정함**
  - **DDL 명령의 결과**
    - **메타데이터를 저장하는 시스템 카탈로그(또는 데이터 사전)에서 관리**
  - **시스템 카탈로그**
    - **특수한 형태의 테이블로 DBMS에 의해서만 사용/수정 된다.**
  - **데이터 사전**
    - **DBMS는 실제 데이터를 읽고 쓰기 전에 항상 데이터 사전을 참조**
- **2. 데이터 조작 언어**
  - **데이터 조작 언어(DML)**
    - **데이터 조작 언어(DML)는 DDL에 의해 구조화된 데이터에 사용자가 데이터를 삽입/수정/삭제하고 저장된 데이터를 검색할 수 있도록 지원하는 언어**
    - **유형**
      - **절차적(procedural) DML**
        - **사용자가 필요한 데이터를 어떻게 구할 것인지를 구체적으로 명시**
        - **반복문/변수 선언 등 일반적 프로그래밍 언어의 기능**
      - **선언적(declarative) 또는 비절차적(nonprocedural) DML**
        - **사용자가 요구하는 데이터가 무엇인지만 기술하는 유형**
        - **처리 절차를 명시하지 않기 때문에 비효율적으로 처리될 수 있다는 단점이 있음**

7. 데이터베이스 시스템 아키텍처 [19]

- **개요**
  - **데이터베이스 시스템**
    - **데이터를 관리하는 DBMS와 데이터를 사용하는 프로그램으로 구성되는 DBMS를 포함한 전체적인 시스템**
    - **유형**
      - **중앙 집중 방식**
      - **클라이언트-서버 방식**
      - **분산 데이터베이스 시스템 방식**
        - **병렬 컴퓨팅 기술 활용**
- **1. 중앙집중식 데이터베이스 관리 시스템 구조**
  - **메인프레임 컴퓨터가 DBMS의 모든 기능뿐만 아니라 사용자의 프로그램, 사용자 인터페이스 등을 모두 처리하는 구조**
  - **대부분 사용자가 자체 데이터 처리 능력이 없는 터미널(terminal) 사용**
  - **단점**
    - **중앙 서버에 너무 많은 부하가 집중되어 병목현상으로 인한 전체적인 성능 저하 및 하드웨어 또는 소프트웨어적 오류 발생 시 전체 시스템이 중단되는 문제**
- **2. 클라이언트-서버 구조**

- 클라이언트-서버(CS: Client-Server) 환경
  - 1) 프로그램 부하를 분산시키고 2) 시스템의 성능을 향상시킬 수 있으며, 3) 프로그램의 유지보수 비용을 절감하고 이식성이 증가하는 등의 효과를 기대
- 클라이언트는 단일 사용자 컴퓨터로서 표현, 계산, 연결, 데이터베이스 서비스 요청 등을 자체적으로 수행
- 서버는 다중 사용자 컴퓨터로서 계산, 연결, 데이터베이스 서비스 등을 제공
- DBMS 서버와 클라이언트 컴퓨터 간의 기능에 따라 2계층과 3계층의 두 가지 유형이 사용
  - (1) 2계층 클라이언트-서버 구조
    - 소프트웨어 구성요소들이 클라이언트와 서버에 분산되어 있기 때문에 2계층 구조(2-tier architecture)라고 한다.
    - 클라이언트에서 DBMS에 대한 접근이 필요하면 프로그램은 서버 쪽에 있는 DBMS의 연결을 설정하고 클라이언트 프로그램이 DBMS와 통신하는 방식으로 동작
    - 질의처리와 트랜잭션 기능은 서버 측에 남아 있어, DBMS 서버를 흔히 질의 서버(query server) 또는 트랜잭션 서버(transaction server)라고 한다. 또한 대부분의 서버가 SQL 언어를 표준으로 사용하므로 흔히 SQL 서버(SQL Server)라고도 한다.
  - (2) 3계층 클라이언트-서버 구조
    - 클라이언트와 데이터베이스 서버 사이 중간 단계인 애플리케이션 서버(Application Server)
    - 웹상에서 동작하는 서버의 경우 웹 애플리케이션 서버(WAS)라고 부르기도 한다.
      - WAS는 비즈니스 규칙(프로시저 또는 제약조건)들을 저장하는 중간 역할
      - 클라이언트로부터 요청을 받아 처리
      - DB 명령을 DB 서버로 보낸 후 DB 서버에서 처리된 데이터를 가시화 등의 작업 후 클라이언트로 보내는 통로 역할을 한다.
    - 클라이언트는 일부 비즈니스 규칙들을 포함

8. 데이터베이스 사용자 및 관리자
   1. 데이터베이스 사용자
      - 유형
        - 일반 사용자
          - 작성된 프로그램을 사용하여 DB 시스템에 접근하는 비전문 사용자
        - 애플리케이션 프로그래머
          - 프로그램을 작성하는 컴퓨터 전문가
        - 전문 사용자
          - SQL이나 데이터 분석 소프트웨어를 사용하여 DBMS를 조작할 수 있는 사용자
        - 특수 사용자
          - 전문 사용자 중 전통적인 데이터 처리 프레임워크에 속하지 않는 CAD 시스템, 지식기반 전문가 시스템, 복합 데이터 타입 등을 저장하는 시스템과 환경 모델링 시스템 등 특수한 데이터베이스 애플리케이션을 작성하는 사용자
   2. 데이터베이스 관리자
      - 데이터베이스 관리자(DBA: Database Administrator)
        - 여러 사용자(프로그램 사용자)가 필요로 하는 정보에 대한 요건을 결정하고, 그들이 필요로 하는 뷰를 제공
        - 데이터 처리의 경제적 효율성을 극대화
        - 임무
          - 설계
            - DB 설계: 데이터의 내부 스키마, 개념-내부 사상 및 외부-개념 사상을 명시하고 스키마와 데이터 사전 구성
            - 보안 기능
            - 백업(backup)과 회복(recovery) 절차
          - 관리
            - 데이터 표현 방법과 시스템 문서화에 대한 기준을 정하기
          - 운용 및 통제
            - 최적의 물리적 저장구조뿐만 아니라 저장매체를 선택, 유지
            - 무결성을 유지하기 위한 DB에 대한 접근과 이 접근을 통제하는 방법도 정의
          - 성능 측정
            - 시스템 내 자원 사용 분석, 병목 현상 조사, 데이터 이용 형태 조사, 각 장치별 성능 측정 등을 수행
            - 물리적 저장구조 재구성, 데이터 접근 방법 갱신
9. MySQL과 MySQL 워크벤치[24]
   1. MySQL Server

- **MySQL : 무료 사용, 다중 스레드, 다중 사용자 지원**
- **MariaDB**
- **상업 라이센스와 오픈 소스 라이센스를 동시에 제공하는 이중 라이센스 정책을 시행**
  - **오픈 소스 라이센스는 GPL에 기반한 소스 코드 공개를 의무화하는 라이센스**
- **Edition**
  - **MySQL Standard Edition**
    - **MyISAM 엔진**
    - **InnoDB엔진 : 트랜잭션과 테이블 제공**
  - **MySQL Enterprise Edition**
    - **대용량 DB 성능 향상을 위한 테이블 파티셔닝 기능 제공, MySQL Enterprise Monitor 기능을 통해 실시간 모니터링 툴로 효과적 DB 관리**
  - **MySQL Cluster Edition**
  - **MySQL Community Edition**
    - **GPL에 의거한 오픈 소스로 배포되는 버전**

2. MySQL 워크벤치

- **SQL 개발과 관리, DB 설계, 생성 및 유지를 위한 단일 개발 통합 환경을 제공하는 비주얼 데이터베이스 설계 도구**
- **주요 기능**
  - **GUI DB 모델링 지원**
  - **Forward & Reverse Engineer 기능 제공**
  - **변경 이력 추적 관리**