데이터베이스: 강의노트 01

A. Silberschatz, H. Korth, S. Sudarshan
Database System Concepts,
Fourth Edition, McGraw-Hill, 2002.

1 소개

- **데이터베이스 관리시스템(DBMS**, DataBase Management System): 서로 관계있는 데이터의 모음과 그 데이터를 접근할 때 사용되는 프로그램의 집합을 말한다.
- **데이터베이스(database)**: 관련 있는 데이터의 모음
- DBMS의 목적: 데이터베이스에 정보를 효율적이고 편리하게 저장하고 검색하는 환경을 제공하는 것
- 데이터베이스는 대량의 정보를 관리할 수 있도록 설계되어 있다.
- 데이터의 관리
  - 정보의 저장 구조 정의
  - 정보를 조작하는 메커니즘을 제공
  - 저장된 정보를 시스템 오류나 불법적인 접근 등으로부터 보호
  - 여러 사용자가 정보를 공유할 경우 이것 때문에 발생하는 문제 방지

1.1 데이터베이스 시스템의 응용

- 데이터베이스 응용의 예
  - 은행 계좌 관리
  - 항공사 예약 시스템
  - 기업의 인사 관리 시스템
- 현재 우리는 우리가 잘 인식하지 못하지만 직접 여러 데이터베이스를 접근하고 사용하고 있다. 이것은 인터넷의 등장으로 더욱 늘어나고 있다.

1.2 데이터베이스 시스템 VS. 파일 시스템

- 파일 시스템의 문제점
  - 데이터의 중복성과 비일관성: 같은 데이터가 여러 파일에 중복될 수 있으며, 이 때문에 중복된 데이터가 서로 일치하지 않을 수 있을 뿐만 아니라 저장 공간 측면에서도 낭비이다.
  - 데이터 접근의 어려움: 초기 개발에서 고려하지 못한 데이터 접근 요구가 있으면 새롭게 프로그램을 작성해야 한다.
  - 데이터 고립: 데이터가 여러 파일에 분산되어 있을 수 있으며, 각각 다른 형태로 저장되어 있을 수 있어 새 프로그램을 작성하는 것이 어려울 수 있다.
  - 무결성(integrity) 문제: 데이터는 보통 어떤 특정한 종류의 일관성 제약 조건(consistency constraint)을 만족해야 한다. 미리 고려하지 못한 제약 조건이나 새 제약 조건이 발생하면 프로그램을 수정해야 한다. 만약 제약 조건이 여러 파일에 분산된 데이터에 걸쳐 있으면 문제가 더 복잡하다.
  - 원자성(atomicity) 문제: 컴퓨터 시스템에 오류가 발생하더라도 데이터가 일관성 상태를 유지하도록 해야 한다. 데이터베이스의 작업은 원자성을 만족해야 한다. 즉, 정상적으로 수행되거나 전혀 수행되지 않아야 한다.
  - 동시 접근의 문제: 여러 사용자가 같은 데이터를 동시에 갱신하면 결과의 일관성이 깨질 수 있다. 파일 시스템 방식에서는 여러 프로그램을 이용하여 접근할 수 있으므로 동시 접근을 효과적으로 제어하기가 어렵다.
  - 보안 문제: 사용자마다 접근할 수 있는 데이터가 제한되어야 한다. 파일 시스템 방식에서는 여러 응용 프로그램이 다른 프로그램의 고려 없이 추가되므로 이런 보안 제약 조건을 강요하기가 어렵다.

1.3 데이터의 관점

1.3.1 데이터의 추상화

- 데이터베이스는 사용자에게 데이터가 어떻게 저장되어 있고, 어떻게 유지되는지 자세한 내용을 숨겨준다.
- 데이터 추상화의 세 단계
  - 물리적 단계: 데이터가 어떻게 저장되어 있는지를 기술한다.
  - 논리적 단계: 어떤 데이터가 저장되어 있고 데이터 간에 어떤 관계가 있는지를 기술한다. 논리적 단계에서는 데이터베이스 전체를 기술한다. 데이터베이스 관리자나 프로그래머는 이 단계를 알아야 하지만 사용자는 그렇지 않다.
  - 뷰 단계(view level): 특정 사용자가 알고 싶은 데이터베이스의 일부분만을 기술한다. 따라서 하나의 데이터베이스는 여러 뷰를 가질 수 있다.
- 프로그래밍 언어에 비유하면
  - 물리적 단계: 실제 저장 구조(컴파일러는 이것을 프로그래머에게 추상화해준다.)

  - 논리적 단계: 타입 정보
  - 뷰 단계: 소프트웨어 사용자 입장에서 보는 단계

1.3.2 인스턴스와 스키마

- 데이터베이스는 시간이 흐름에 따라 그 내용이 변한다.
- **스키마(schema)**: 전체적인 데이터베이스의 설계
- **인스턴스(instance)**: 특정 순간에 데이터베이스에 저장되어 있는 정보의 모임
- 인스턴스는 시간에 흐름에 따라 변하지만 스키마는 그렇지 않다.
- 프로그래밍 언어에 비유하면
  - 스키마: 변수의 정의
  - 인스턴스: 프로그램 수행 중 특정 순간에 변수의 값
- 스키마는 추상화 단계에 따라 다음과 같은 스키마가 존재한다.
  - 물리적 스키마
  - 논리적 스키마: 가장 중요한 스키마이다.
  - 서브 스키마: 뷰 단계의 스키마로서 각 뷰마다 그 뷰를 설명하는 스키마가 존재한다.
- 만약 물리적 스키마가 변해도 논리적 스키마를 변경할 필요가 없으면 물리적 데이터 독립성이 있다고 한다.

1.4 데이터 모델

- 데이터베이스의 구조를 데이터 모델이라 한다.
- 데이터 모델: 데이터, 데이터의 상호관계, 데이터의 의미, 일관성 제약 조건 등을 기술하기 위한 개념적 도구의 집합

1.4.1 개체 관계 모델

- 개체 관계 모델(entity-relationship model)은 개체와 개체 간의 관계를 이용하여 실세계를 모델링하는 방법이다.
- 각 개체는 속성의 집합으로 기술된다. 예) 은행 계좌: 계좌번호, 잔액
- 관계는 여러 개체 간에 연관성을 말한다. 예) 고객과 계좌: 입금자 관계
- 개체 집합: 같은 종류의 개체의 모음
- 관계 집합: 같은 종류의 관계의 모음
- 데이베이스의 논리 스키마는 E-R 다이어그램을 이용하여 시각적으로 도식화한다.
  - 데이터베이스의 내용이 만족해야 제약 조건 중 하나인 **대응수(mapping cardinality)**도 E-R 다이어그램에 나타낼 수 있다.
  - 대응수: 어떤 개체가 하나의 관계 집합에 의해 연관될 수 있는 개체의 수

1.4.2 관계형 모델

- 관계형 모델(relational model)은 데이블을 이용하여 데이터와 데이터의 관계를 모두 나타낸다.
- 현재 가장 널리 사용되는 모델이다.
- 관계형 모델은 레코드 기반 모델의 예이다. 레코드 기반 모델이란 고정된 형태의 레코드로 구성된 데이터베이스를 설명하는 모델이다.
- 각 레코드는 여러 개의 필드(또는 속성)으로 구성되어 있다.
- 관계형 모델은 E-R 모델의 하위 추상화 모델로 볼 수 있다.

1.5 데이터베이스 언어

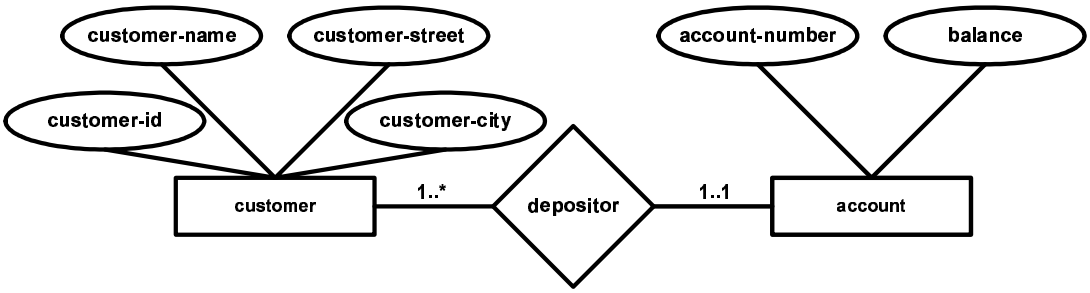
- 데이터베이스는 다음 두 가지 종류의 언어를 제공한다.
  - **데이터 정의 언어(DDL, Data Definition Language)**: 스키마를 정의할 때 사용한다.
  - **데이터 조작 언어(DML, Data Manipulation Language)**: 데이터베이스를 질의하고 갱신할 때 사용한다.

1.5.1 데이터 정의 언어

- DDL의 예

  create table account
  (account-number **char**(10),
  balance **integer**)

  - 위 DDL 문장은 account 데이블을 생성하고, 데이터 사전(data dictionary) 또는 데이터 디렉토리(data directory)라는 특수한 테이블을 갱신한다.
- 데이터 사전에는 메타데이터(metadata)라고 하는 데이터에 대한 데이터가 기록된다. 예) 테이블의 스키마
- DDL의 한 종류인 데이터 저장 및 정의 언어(data storage and definition language)를 이용하여 데이터의 저장구조와 접근 방법을 지정한다.
- 데이터베이스의 일관성 제약 조건도 DDL을 이용하여 정의한다.



<그림 1.1> E-R 다이어그램의 예

1.5.2 데이터 조작 언어

- 데이터 조작의 종류
  - 기존 정보 검색
  - 새 정보 추가
  - 기존 정보 삭제
  - 기존 정보 수정
- 데이터 조작 언어의 종류
  - **절차식(procedural) DML**: 사용자는 어떤 데이터를 어떻게 검색할지 지정해야 한다.
  - **선언적(declarative) DML**: 다른 말로 비절차식 DML이라 하며 사용자는 오직 어떤 데이터를 원하는지만 지정하면 된다.
- **질의(query)**: 정보 검색을 요청할 때 사용하는 문장
- **질의어(query language)**: DML 중 정보 검색을 담당하는 언어
- 질의어의 예

  select customer.customer-name
  from customer
  where customer.customer.id = '192-83-7465'
  - 위 질의 문장은 customer 데이블에서 고객 식별자가 192-83-7465인 고객의 이름을 검색한다.
  - 질의가 하나의 이상의 테이블을 검색해야 하는 경우도 있다.

1.5.3 응용 프로그램에서 데이터베이스 접근

- 자바, C와 같은 호스트 언어에서 DML을 사용하는 방법
  - 응용 프로그램과 데이터베이스 간에 인터페이스 제공
  - 호스트 언어를 확장하여 DML 기능을 포함하는 방법

1.6 데이터베이스 사용자와 관리자

1.6.1 데이터베이스 사용자와 사용자 인터페이스

- 데이터베이스 사용자의 종류
  - 일반 사용자: 응용 프로그램을 통해 데이터베이스를 사용하는 비 전문가
  - 응용 프로그램 프로그래머: 데이터베이스를 사용하는 응용 프로그램을 개발하는 컴퓨터 전문가
  - 능숙한 사용자: 응용 프로그램을 통해 데이터베이스를 접근하지 않고 데이터베이스 언어를 직접 이용하여 데이터베이스에 접근하는 데이터베이스 전문가
    - 온라인 분석 처리(OLAP, OnLine Analytical Processing) 도구를 사용하여 데이터베이스에 저장되어 있는 데이터를 분석하는 분석가
  - 특수 사용자: 능숙한 사용자로서 전통적인 데이터 처리 방식과 다른 특수한 데이터베이스 응용 프로그램을 개발하는 전문가
    - 특수한 응용 프로그램의 예: 컴퓨터 지원설계 시스템, 전문가 시스템

1.6.2 데이터베이스 관리자

- 데이터베이스 관리자(DBA, DataBase Administrator)의 역할
  - 스키마 정의
  - 저장 구조와 접근 방법 정의
  - 스키마 및 물리적 구조 수정: 요구 변화나 성능 향상을 위한 수정
  - 데이터 접근 권한 위임
  - 유지보수: 백업, 디스크 공간 관리, 데이터베이스 감시

1.7 트랜잭션 관리

- **트랜잭션(transaction)**: 데이터베이스 응용 프로그램에서 하나의 논리적 기능을 수행하는 연산들의 모임
- 트랜잭션의 요구사항: ACID

  - 원자성(atomicity): all-or-none 요구사항
  - 일관성(consistency): 정확성 요구사항
  - 고립성(isolation): 고립되어 실행하였을 때와 결과가 같아야 한다.
  - 지속성(durability): 시스템에 오류가 발생하여도 데이터는 그 값을 유지해야 한다는 요구사항

- 트랜잭션이 시작될 때 데이터베이스가 일관성 있는 상태였다면 끝난 후에도 데이터베이스는 일관성 있는 상태이어야 한다. 그러나 실행 중간에는 일관성 상태가 아닐 수 있다. 이것 때문에 시스템이 실패하면 문제가 발생할 수 있다. 보통 트랜잭션이 실행되는 도중에 중단되면 트랜잭션이 수행되기 전 상태로 데이터베이스를 되돌려야 한다. 이것을 **실패 복구(failure recovery)**라 한다.

- 병행제어/동시성 제어 관리자(concurrency control manager): 여러 트랜잭션이 병행으로 실행되어도 데이터베이스가 일관성 상태를 유지할 수 있도록 하는 역할을 한다.

1.8 데이터베이스 시스템 구조

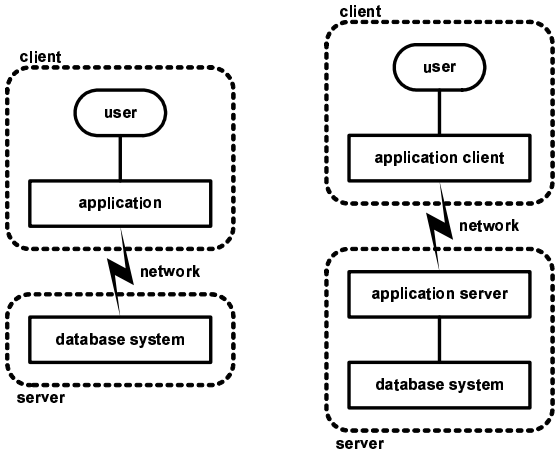
- 기능적 관점에서 데이터베이스의 중요 구성요소
  - 저장 관리자
  - 질의 처리기

1.8.1 저장 관리자

- 저장 관리자: 데이터베이스에 저장되는 있는 하위 레벨 데이터와 응용 프로그램 또는 질의 간에 인터페이스를 제공해 준다. 저장 관리자는 DML 문잘들을 하위 레벨 파일 시스템 명령으로 바꾸어 준다.

- 저장 관리자의 구성요소
  - 권한위임과 무결성 관리자
  - 트랜잭션 관리자
  - 파일 관리자: 디스크 공간 할당과 디스크에 저장될 정보의 데이터 구조를 관리한다.
  - 버퍼 관리자: 디스크 저장장치로부터 데이터를 인출하여 주기억장치에 적재하고, 주기억장치에 어떤 데이터를 캐시할지 결정하는 역할을 한다.

- 저장 관리자는 이를 위해 다음 데이터 구조를 구현한다.
  - 데이터 파일: 데이터베이스 자체를 저장하는데 사용된다.
  - 데이터 사전: 데이터베이스 구조에 대한 메타데이터를 저장하는데 사용된다. 특히 스키마가 저장된다.
  - 색인(index): 특정한 값을 지닌 데이터 항목에 빠르게 접근하기 위한 정보가 저장된다.



(a) 2-계층 구조 (b) 3-계층 구조

<그림 1.2> 2-계층과 3-계층 구조

1.8.2 질의 처리기

- 질의 처리기의 구성요소
  - **DDL 인터프리터**: DDL 문장을 해석하여 그것에 필요한 정의를 데이터 사전에 기록한다.
  - **DML 컴파일러**: DML 문장을 질의 수행 엔진이 이해할 수 있는 명령으로 구성한다.
  - **질의 수행 엔진(query evaluation engine)**: DML 컴파일러가 만든 하위 레벨 명령을 실행한다.

1.8.3 응용 프로그램 구조

- 보통 클라이언트-서버 구조를 사용한다.

- 데이터베이스 응용 프로그램 구조의 종류
  - 2-계층 구조(two-tier architecture): 응용 프로그램이 클라이언트 컴퓨터에 상주하는 부분과 서버에 상주하는 부분으로 나누어지며, 클라이언트 컴퓨터에 상주하는 부분은 데이터베이스를 직접 접근한다.
  - 3-계층 구조(three-tier architecture): 이 구조는 2-계층 구조와 달리 클라이언트 컴퓨터에 상주하는 부분은 데이터베이스를 직접 접근하지 않고 응용 서버를 통해 데이터베이스를 접근한다.
    - 응용 서버에만 비즈니스 로직이 구현되어 있다.
    - 비지니스 로직은 어떤 행동을 어떠한 조건 하에서 수행되는지를 결정한다.

- 규모가 큰 응용 프로그램이나 웹 응용의 경우에는 3-계층 구조가 더 적합한다.