1. 데이터베이스 기본 개념

데이터와 정보
- 데이터: 단순히 관찰, 측정하여 수집한 값
- 정보: 유용하게 활용할 수 있게 데이터를 처리한 결과물

정보시스템과 데이터베이스
- 정보시스템: 데이터를 수집, 저장. 필요 시 유용한 정보 만드는 수단
- 데이터베이스: 정보 시스템 내, 데이터 저장/제공 역할
  - 특정 조직의 여러 사용자가 **공유**하여 사용할 수 있도록 **통합 저장**한 **운영** 데이터 집합

- 공유 데이터 — 여러 사용자가 함께 소유, 이용 가능
- 통합 데이터 — 최소한의 중복, 통제 가능한 중복만 허용
- 저장 데이터 — 컴퓨터가 접근 가능한 매체에 저장
- 운영 데이터 — 조직의 기능을 수행하기 위해 필요함

- 실시간 접근성 — 데이터 요구에 실시간 응답
- 내용 기반 참조 — 주소/위치 대신 내용으로 참조
- 계속 변화 – 계속해서 삽입, 삭제, 수정; 현재의 정확한 데이터 유지
- 동시 공유 – 데이터의 동시 사용 지원

데이터 과학 시대의 데이터
- 형태 기준: 정형 데이터 / 반정형 데이터 / 비정형 데이터
- 특성 기준: 범주형(명목형$_{서열X}$:순서형$_{서열O}$) / 수치형(이산형:연속형)

2. 데이터베이스 관리 시스템

등장 배경
- 파일 시스템
  - 데이터 중복성: 같은 내용의 데이터 여러 파일에 중복 저장
  - 데이터 종속성: 응용프로그램, 데이터 파일에 종속적
  - 동시 공유, 보안, 회복 기능 부족
  - 응용프로그램 개발 어려움

DBMS의 주요 기능
- 정의: DB 구조의 정의/수정
- 조작: 데이터 삽입, 삭제, 수정, 검색
- 제어: 항상 정확하고 안전하게 유지

- 데이터 중복 통제
- 데이터 독립성 확보
- 데이터 동시 공유
- 데이터 보안 향상
- 데이터 무결성 유지
- 표준화
- 장애 회복
- 응용 프로그램 개발 비용 절감

- 비용 증가
- 백업/회복 방법 복잡함
- 중앙 집중 관리 ⇒ 취약점

발전 과정
1. 네트워크 DBMS, 계층 DBMS
2. 관계 DBMS
3. 객체지향 DBMS, 객체관계 DBMS
4. NoSQL / NewSQL

3. 데이터베이스 시스템
- 데이터베이스에 데이터 저장/관리 ⇒ 조직에 필요한 정보 생성

스키마와 인스턴스
- 스키마: DB에 저장되는 구조와 제약조건 정의
- 인스턴스: 스키마에 맞춰 실제 저장된 값

3단계 DB 구조
- 외부 단계: 개별 사용자 관점
- 개념 단계: 조직 전체 관점
- 내부 단계: 저장 장치 관점

외부 스키마 (=Sub Schema)
- 하나의 DB에 여러 개 존재
- 개별 사용자 관점에서 이해/표현
- 사용자에게 필요한 DB 정의
- 각 사용자가 생각하는 DB 모습: 사용자마다 다른 스키마

개념 스키마
- 하나의 DB에 한 개 존재 (외부 스키마(소요) 전부 고려한 스키마이므로)
- 조직 전체 관점에서 이해/표현
- 전체 DB에 어떤 데이터 저장? 데이터 간 어떤 관계? 제약조건? 보안 정책? 접근 권한?

내부 스키마
- 저장 장치 관점에서 표현
- 하나의 DB에 한 개 존재
- 실제로 저장되는 방법 정의
- 레코드 구조, 필드 크기, 접근 경로 등 정의

3단계 구조에서의 매핑
- 외부/개념 사상 — 외부-개념 (응용 인터페이스)
- 개념/내부 사상 — 개념-내부 (저장 인터페이스)
- 3단계 구조화, 단계별 스키마 유지, 스키마 사이 대응 관계 정의 ⇒ 데이터 독립성 실현
  - 하위 스키마 변경 $\not\rightarrow$ 상위 스키마 영향
  - 논리적 데이터 독립성: 개념 스키마 변경 $\not\rightarrow$ 외부 스키마 영향
    변경 발생 시 관련된 외부/개념 사상만 수정
  - 물리적 스키마 독립성: 내부 스키마 변경 $\not\rightarrow$ 개념 스키마 영향
    변경 발생 시 관련된 개념/내부 사상만 수정

데이터 사전(= 시스템 카탈로그)
- 메타데이터를 유지하는 시스템 DB
- 스키마, 사상정보, 제약조건 저장
- DBMS가 스스로 생성, 유지
- 일반 사용자도 접근 가능, 저장 내용 검색만 가능

데이터 디렉토리
- 데이터 사전의 데이터를 접근하는데 필요한 위치 정보 DB
- 일반 사용자 접근 불가능

사용자 데이터베이스
- 일반 사용자가 이용하는 데이터가 저장된 DB

데이터베이스 사용자
- 데이터베이스 관리자, 최종 사용자, 응용 프로그래머
- 데이터베이스 관리자
  - DBMS를 운영 관리
  - 주로 DDL, DCL 사용
  - 업무
    - DB 구성 요소 선정
    - DB 스키마 정의
    - 물리적 저장 구조 / 접근 과정 결정
    - 무결성 유지: 제약조건 정의
    - 보안 / 접근 권한 정책 결정
    - 백업 / 회복 기법 정의
    - 시스템 DB 관리
    - 시스템 성능 감시 및 분석
    - DB 재구성
- 최종 사용자
  - DB에 접근하여 데이터 조작
  - DML 사용
- 응용 프로그래머
  - 데이터 언어 이용, 응용 프로그램 작성
  - DML 사용

데이터 언어
- 사용자-DBMS 간 통신 수단
- DDL / DML / DCL
  - DML: 절차적 데이터 조작어 / 비절차적 데이터 조작어(=선언적)
  - DCL: 무결성, 보안, 회복, 동시성 제어를 위해 사용

DBMS
- 주요 구성 요소
  - 쿼리 처리기
    - DDL 컴파일러 / DML (프리)컴파일러 / 런타임 데이터베이스 처리기 / 트랜잭션 관리자 등
  - 저장 데이터 관리자 – DB와 데이터 사전 관리/접근

4. 데이터 모델링
- 현실 세계의 데이터를 DB로의 변환 과정 (추상화)

2단계 데이터 모델링
- 개념적 데이터 모델링
  - 현실의 주요한 데이터를 추출하여 개념 세계에 마이그레이션
- 논리적 데이터 모델링
  - 개념 세계의 데이터를 DB에 마이그레이션

데이터 모델
- 모델링 결과물 표현
- 개념적 데이터 모델
  - 현실 세계를 개념적 모델링 ⇒ 데이터베이스의 개념적 구조로 표현
- 논리적 데이터 모델
  - 개념적 구조를 논리적 모델링 ⇒ 데이터베이스의 논리적 구조로 표현

데이터 모델의 구성
- 데이터 구조
  - 개념적 데이터 모델에서의 개념적 구조
  - 논리적 데이터 모델에서의 논리적 구조
- 연산: 값 처리 작업
- 제약조건: 데이터 무결성 유지
  - 구조적인 제약 사항
  - 연산 적용: 허용할 수 있는 의미적인 제약 사항

개체-관계 모델: E-R Model; Entity-Relationship model
- 개체-관계 다이어그램: E-R Diagram
  - 개체-관계 모델을 이용해 현실 세계를 개념적으로 모델링한 결과물
- 개체: Entity
  - 현실 세계에서 구별되는 모든 것
  - 저장할 가치가 있는 중요 데이터를 갖고 있음
  - 속성(개체만의 고유한 특성/상태)을 하나 이상 가짐
- 속성: Attribute
  - 개체/관계가 갖고 있는 고유한 특성
  - 의미 있는 데이터의 논리적 단위
  - 파일 구조: 필드(Field)
  - ERD: 타원으로 표현, 타원 안에 이름 표기
- 개체 타입: Entity Type
  - 개체를 고유한 이름과 속성으로 정의
  - 파일 구조: 레코드 타입(Record Type)
- 개체 인스턴스: Entity Instance, 개체 어커런스(occurence)
  - 실체화된 개체
  - 파일 구조: 레코드 인스턴스(Record Instance)
- 개체 집합: Entity Set

- 단일 값 속성과 다중 값 속성
  - 단일 값 속성: 하나만 가질 수 있는 속성
    - i.e. 고객 개체의 이름
  - 다중 값 속성: 여러개 가질 수 있는 속성
    - i.e. 책 개체의 저자 속성
    - ERD: 이중 타원으로 표현
- 단순 속성과 복합 속성
  - 단순 속성: Simple Attribute
    - 더 분해할 수 없음
  - 복합 속성: Composite Attribute
    - i.e. 주소(도, 시, 동, 우편번호로 분해) / 생년월일(연, 월, 일로 분해)
- 유도 속성: Derived Attribute
  - 다른 속성에서 유도되어 결정됨
  - 값이 별도로 저장되지 않음
  - ERD: 점선 타원으로 표현
  - i.e. 판매가격(가격 · 할인율 일 때)
- 널 속성: Null Attribute
  - 널 값이 허용되는 속성
- 키 속성: Key Attribute
  - 개체 인스턴스를 식별하는 데 사용되는 속성
  - 유일(Unique)한 값
  - 둘 이상의 속성으로 구성될 수 있음
  - ERD: 밑줄

- 관계: Relationship
  - 개체 사이에 맺고 있는 의미 있는 연관성, 대응 관계(=mapping)
  - ERD: 마름모
- 관계 유형
  - 관계 참여자 수 기준
    - 이항 관계 – 개체 타입 2개가 맺음
    - 삼항 관계 – 개체 타입 3개가 맺음

  - 순환 관계 – 개체 타입 1개가 자기 자신과 맺음
- 매핑 카디널리티 기준
  - 1:1 / 1:n / n:m
  - 매핑 카디널리티: 관계를 맺는 두 집합에서, 각 개체 인스턴스가 연관성을 맺고 있는 상대 개체 집합의 인스턴스 수
- 관계의 참여 특성
  - 필수 참여 (전체 참여) / 선택 참여(부분 참여)
  - ERD: (필수 참여) 이중선
- 관계의 종속성
  - 약한 개체 — 다른 개체의 존재에 의존
    - 강한 개체의 키를 포함하여 키 구성
    - ERD: 이중 사각형으로 표현
  - 강한 개체 — 다른 개체의 존재를 결정

- E-R 다이어그램
  - 사각형: 개체 // 마름모: 관계 // 타원: 속성
  - 링크(연결선): 각 요소 연결 // 레이블: 1:1 1:n n:m 관계 표기

논리적 데이터 모델
- E-R 다이어그램으로 표현된 개념 구조 ⇒ DB에 저장할 형태로 표현 (=schema)
- 관계 데이터 모델, 계층 데이터 모델, 네트워크 데이터 모델

- 관계 데이터 모델: 일반적으로 널리 사용
  - 논리적 구조 — 2차원 테이블 형태
- 계층 데이터 모델
  - 논리적 구조 — 트리 형태(루트 존재, 사이클 없음)
  - 개체 간 상하 관계: 부모(1)/자식(n) – n:m 불가능
  - 두 개체 간 하나의 관계만 정의 가능
  - 모델링 어려움 ⇒ 구조 복잡해질 수 있음
  - 삽입/삭제/수정/검색 어려움
- 네트워크 데이터 모델
  - 논리 구조: 네트워크(=그래프 형태)
  - 1:n 관계만 허용 – n:m 불가능
  - 관계 여러개 정의 가능 – 관계를 이름으로 구별
  - 삽입/삭제/수정/검색 어려움, 구조 복잡

5. 관계 데이터 모델
관계 데이터 모델
- 개념
  - 논리 데이터 모델: 개념적 구조 ⇒ 논리적 구조로 표현
  - 하나의 개체 정보 ⇒ 하나의 릴레이션
- 용어
  - 릴레이션: 개체에 대한 2차원 데이터 집합
  - 속성: 릴레이션의 열
  - 튜플: 릴레이션의 행
  - 도메인: 속성이 가질 수 있는 모든 값의 집합
    - 실제 값의 데이터 타입은 도메인 이용하여 정의
  - 차수: 릴레이션 당 속성 개수
  - 카디널리티: 릴레이션에서 튜플 개수
- 예시:

수강신청

신청번호	학번	강의코드	신청일자
20230001	213892	STT2939	2023-03-01 10:00:44
20230002	231933	STT2939	2023-03-01 10:00:45
20230003	213892	AIC4934	2023-03-01 10:00:45
20230004	231933	AIC4934	2023-03-01 10:00:45
20230005	213892	JLL4934	2023-03-01 10:00:45

  - 차수 4, 카디널리티: 5

릴레이션
- 릴레이션 스키마: 릴레이션의 논리적 구조
  - (= 릴레이션 내포)
  - 릴레이션의 이름 + 모든 속성의 이름
    - i.e. 고객(고객아이디, 고객이름, 나이, 등급, 직업)
  - 정적임: 자주 변하지 않음
- 릴레이션 인스턴스: 릴레이션에 존재하는 튜플들의 집합
  - (= 릴레이션 외연)
  - 삽입/삭제/수정 빈번
- 특성
  - 유일성: 동일한 튜플 존재 불가
  - 튜플/속성의 무순서
  - 속성의 원자성

키: 릴레이션에서 튜플들을 유일하게 구별하는 속성(/집합)
- 성질
  - 유일성: 모든 튜플은 서로 다른 키 값 가짐
  - 최소성: 최소한의 속성들로 키 구성
- 종류

	슈퍼키	후보키	기본키	대체키
유일성	O	O	O	O
최소성	X	O	O	O
디폴트	X		O	X

  - 후보키 = 기본키 ∪ 대체키
  - 외래키: 다른 릴레이션의 기본키

관계 데이터 모델의 제약
- 무결성 제약조건
  - 개체 무결성 제약조건: 기본키를 구성하는 모든 속성은 Not Nullable
  - 참조 무결성 제약조건: 참조 불가능한 외래키 존재 X

8. DB 설계
DB 설계 단계
1. 요구사항 분석†
2. 개념적 설계†
3. 논리적 설계†
4. 물리적 설계
5. 구현

- 설계 과정 중 오류 발견 시 이전 단계로 돌아가 수정
- †: 핵심 단계

1. 요구사항 분석 ⇒ 요구 사항 명세서
- 요구사항 수집, 분석 ⇒ DB 용도 파악
- 작업
  - DB를 실제로 사용할 사용자 범위 결정
  - 사용자가 수행하는 업무 분석
  - 요구 사항 수집 ⇒ 분석 결과를 명세서로 작성

2. 개념적 설계 ⇒ ER 다이어그램
- DBMS에 독립적인 개념적 스키마 설계
- 개념적 모델링: 1단계 결과물을 개념적 구조로 표현
- 작업
  1. 개체/속성 추출: “명사 찾기”
  2. 관계 추출: “동사 찾기”
  3. ERD 작성

3. 논리적 설계 ⇒ 릴레이션 스키마
- DBMS에 맞는 논리적 스키마 설계
- 논리적 모델링: 개념적 스키마 ⇒ 논리적 구조로 표현
- 규칙
  1. 모든 개체는 릴레이션으로 변환
  2. 다대다: 릴레이션으로 변환
  3. 일대다: 외래키
  4. 일대일: 외래키
  5. 다중 값 속성: 릴레이션으로 변환

9. 정규화
정규화와 이상현상
- 이상현상: 불필요한 데이터 중복으로 연산 수행 중 발생 가능한 부작용
  - 삽입 이상: 삽입 시 추가적인 데이터 삽입 필요
  - 갱신 이상: 중복 튜플 일부만 갱신 ⇒ 데이터 갱신 미완
  - 삭제 이상: 필요한 데이터까지 제거
- 정규화: 함수 종속성 이용, 이상 현상 제거
  - ⇒ 이상 현상 없는 릴레이션으로

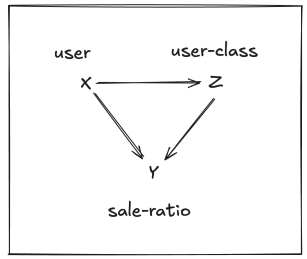
함수 종속
- 함수 종속: $X \to Y$
  - X: 결정자 / Y: 종속자
  - X에 대한 Y 값이 항상 한 개
  - 기본키/후보키: 다른 속성들을 함수적으로 결정
- 완전 함수 종속: FFD(Full Functional Dependency)
  - $Y \not\to \sim X$ (=$Y \to X$의 여집합 없음)
  - 남는 Y 없음
- 부분 함수 종속: PFD(Partial Functional Dependency)
  - 남는 Y 있음

정규화
- 릴레이션은 무손실 분해되어야 함

1. 제1정규형
   - 모든 속성이 원자값이어야 함
   - i.e. 만족하지 않음: `(apple, (e001, e004, e010), (t, f, t), 0.1)`
   - 이상 발생: 기본키에 완전 함수 종속되지 않은 경우 이상 발생
   - i.e.

사용자	이벤트	등급
user1	e001	gold
user1	e004	gold
user1	e010	gold

     - `user1`의 등급을 변경하려면 세 행 모두 변경해야함
2. 제2정규형
   - 제1정규형 + 모든 속성이 기본키에 완전 함수 종속
   - 테이블 분해
   - 이상 발생: 이행적 함수 종속
   - 한 테이블 내에서...



     - $X \to Z$; $Z \to Y$가 한 테이블에 들어있어서, $Z \to Y$ 변경 시 $z_t \to y_t$가 설정된 모든 열을 바꿔야 함
3. 제3정규형
   - 제2정규형 + 이행적 함수 종속 제거
   - 이상 발생:

username	lecture	lecturerer-id
username1	lecture1	p004
username2	lecture2	p004

     - 설명: `lecturer-id`가 `lecture`를 결정

- 강한 제3정규형 (= BCNF; Boyce/Codd Normal Form)
  - 릴레이션의 함수 종속 관계에서 모든 결정자가 후보키임
  - 후보키가 아닌 결정자 제거 목표로 테이블 분해