

C H A P T E R 04

데이터 모델링



Data Modeling

박상돈 조교수

대전대학교 컴퓨터공학과

목차

0

데이터 모델링의 개념

1

2단계 모델링, 데이터 모델 구성

0

개체-관계 모델 (ER Model)

2

개체, 속성, 관계, ERD

0

논리적 데이터 모델

3

관계형, 계층형, 네트워크 모델

01

데이터 모델링의 개념

Data Modeling Concepts

데이터 모델링이란?

데이터 모델링 (Data Modeling)

현실 세계에 존재하는 데이터를 컴퓨터 세계의 데이터베이스로 옮기는 과정.
데이터베이스 설계의 핵심 과정.

구조 정의

효율적 저장/검색을 위해
데이터 구조를 명확히 정의

요구사항 반영

사용자 요구사항을
체계적으로 DB에 반영

소통 도구

개발자와 사용자 간의
공통 언어 역할

비유: 건축가의 설계도 = 데이터 모델링 / 건물 = DB / 자재·기술 = DBMS

2단계 데이터 모델링



개념적 데이터 모델링

논리적 데이터 모델링

개념적 모델링

현실 세계 → 개념 세계

도구: ER 모델 (개체-관계 모델)

논리적 모델링

개념 세계 → 컴퓨터 세계

도구: 관계 데이터 모델 (테이블)

이 전체 과정을 추상화(Abstraction)라고 합니다

데이터 모델의 구성

데이터 모델의 구성 = 데이터 구조 (Structure) + 연산 (Operation) + 제약조건 (Constraint)

데이터 구조

데이터가 어떤 형태로 저장되는지 정의

연산

데이터를 어떻게 조작하는지 (삽입, 삭제, 수정, 검색)

제약조건

데이터가 만족해야 하는 조건과 규칙

02

개체-관계 모델

Entity-Relationship Model

개체-관계 모델 (E-R Model) 개요

개체-관계 모델 (Entity-Relationship Model)

Peter Chen(1976)이 제안. 현실 세계를 개체 간의 관계로 표현. ER 다이어그램(ERD)으로 시각화.

개체 (Entity)

사각형 □

현실의 구별되는 대상

속성 (Attribute)

타원 ○

개체의 고유 특성

관계 (Relationship)

마름모 ◇

개체 간 연관성

ER 모델 3요소 = 개체 + 속성 + 관계 → ERD로 시각화

개체 (Entity)

개체 (Entity)

현실 세계에서 구별되는 모든 것. 저장할 가치가 있는 데이터를 가진 사람·사물·개념·사건.

사람

고객, 학생, 교수

사물

상품, 책, 건물

개념

과목, 부서, 프로젝트

사건

주문, 수강신청, 예약

개체 타입 = 구조 정의(스키마) | 개체 인스턴스 = 실제 저장된 값 | ERD: 사각형(□)

속성 (Attribute)의 분류

속성 (Attribute)

개체의 고유한 특성. 의미 있는 데이터의 가장 작은 논리적 단위. ERD: 타원(o).

분류 기준	종류	설명	ERD 표현	예시
값의 개수	단일 값	값을 하나만 가짐	일반 타원	이름, 나이
	다중 값	여러 값 가능	이중 타원	연락처, 이메일
의미 분해	단순	더 분해 불가	일반 타원	적립금, 가격
	복합	하위 속성으로 분해	가지형	주소, 생년월일
유도 여부	유도	계산으로 유도	점선 타원	판매가격
식별 기능	key	인스턴스 식별	밑줄 타원	고객아이디
null 허용	null	값 미정 허용	—	등급(미정 시)

단일 값 속성 vs 다중 값 속성

단일 값 속성 (Single-Valued)

값을 하나만 가질 수 있는 속성

예) 이름, 나이, 학번

ERD: 일반 타원

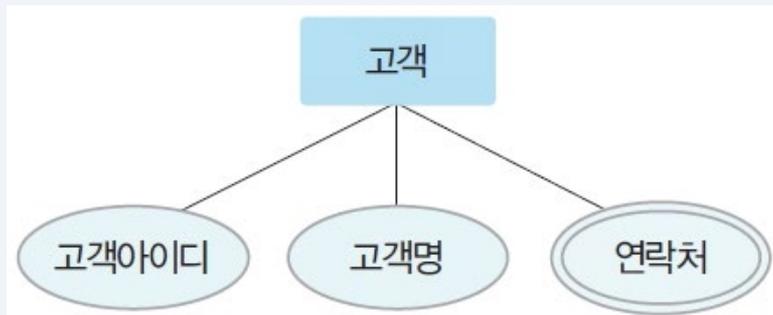
다중 값 속성 (Multi-Valued)

값을 여러 개 가질 수 있는 속성

예) 연락처, 이메일

ERD: 이중 타원 ◎

실무: 다중 값 속성은 관계형 DB에서 별도 테이블로 분리 (1NF 정규화)



단순/복합 속성 · 유도 속성

단순 vs 복합 속성

단순: 더 분해 불가 (예: 적립금)

복합: 하위 속성으로 분해 가능

예) 주소 → 시, 구, 동 / 생년월일 → 년, 월, 일

유도 속성 (Derived)

다른 속성에서 계산으로 유도

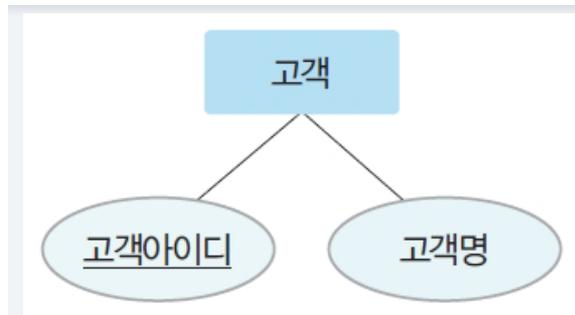
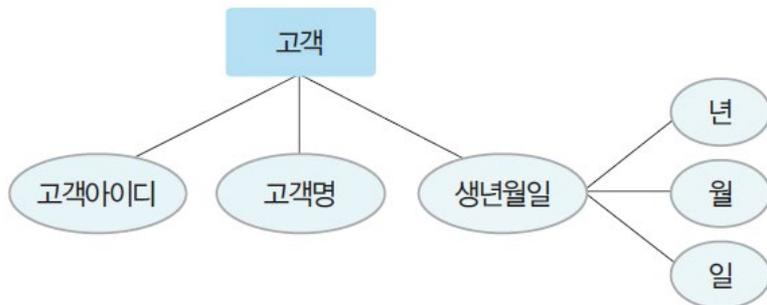
별도 저장하지 않음

예) 판매가격 = 정가 × (100 - 할인율) / 100

ERD: 점선 타원

null 속성 · key 속성

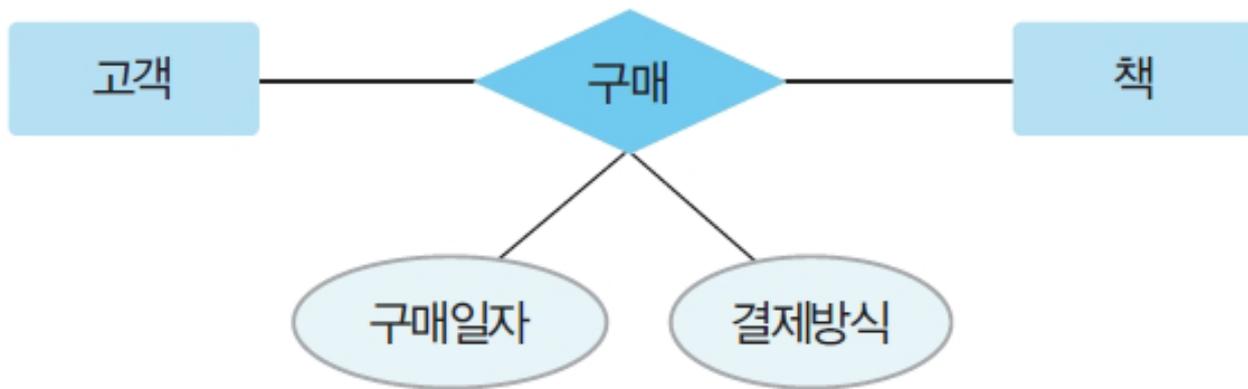
null: 아직 결정되지 않았거나 모르는 값 (≠ 공백, ≠ 0) | key: 인스턴스를 유일하게 식별하는 속성 (ERD: 밑줄)



관계 (Relationship)

관계 (Relationship)

개체와 개체가 맺고 있는 의미 있는 연관성. 개체 집합들 사이의 매핑(mapping). ERD: 마름모(◇).



관계의 유형 — 매핑 카디널리티

매핑 카디널리티 (Mapping Cardinality)

각 개체 인스턴스가 상대 개체 집합의 인스턴스 몇 개와 대응되는지를 나타냄.

1:1 관계

일대일

양쪽 모두 하나씩 대응

예) 남편 ↔ 아내

1:n 관계

일대다

한 쪽이 여러 개와 대응

예) 부서 → 사원들

n:m 관계

다대다

양쪽 모두 여러 개와 대응

예) 학생 ↔ 과목

가장 중요한 세 가지: 1:1 (드물) / 1:n (가장 많음) / n:m (많음)

일대일 (1:1) 관계 예시

예시: 남편 ↔ 아내 (결혼)

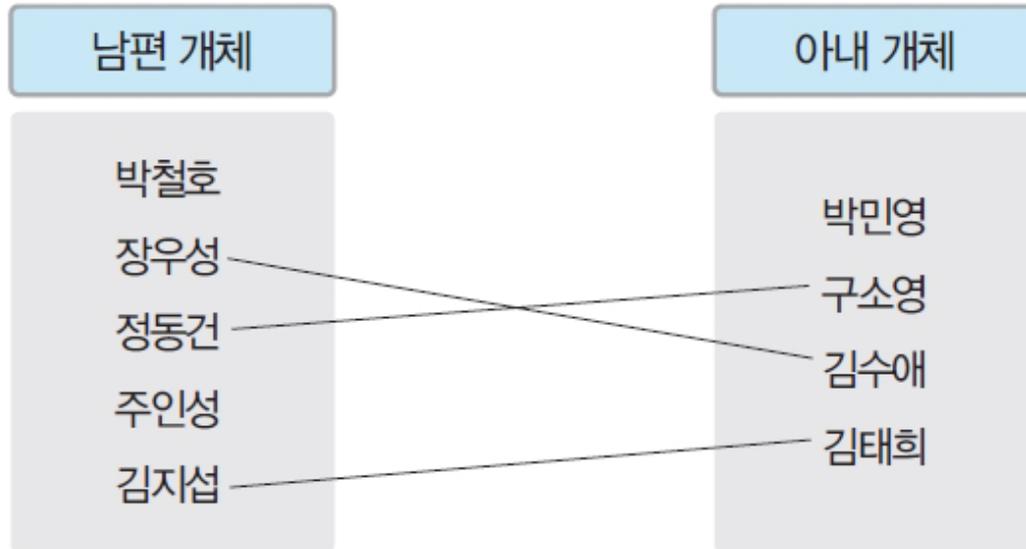


그림 4-14 일대일 관계의 예 : 남편과 아내 개체의 혼인 관계

일대다 (1:n) 관계 예시

예시: 부서 → 사원 (소속)

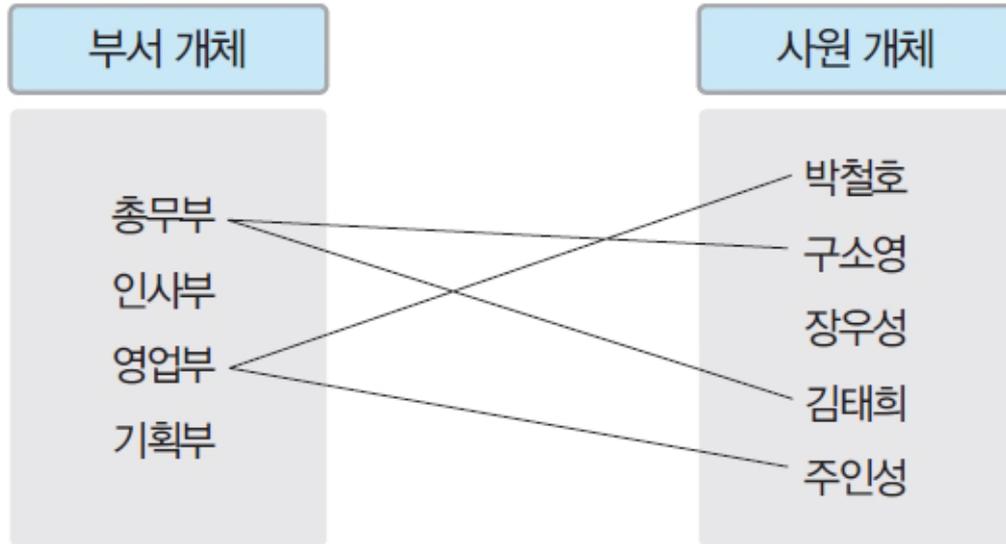


그림 4-15 일대다 관계의 예 : 부서와 사원 개체의 소속 관계

다대다 (n:m) 관계 예시

예시: 고객 ↔ 책 (구매)

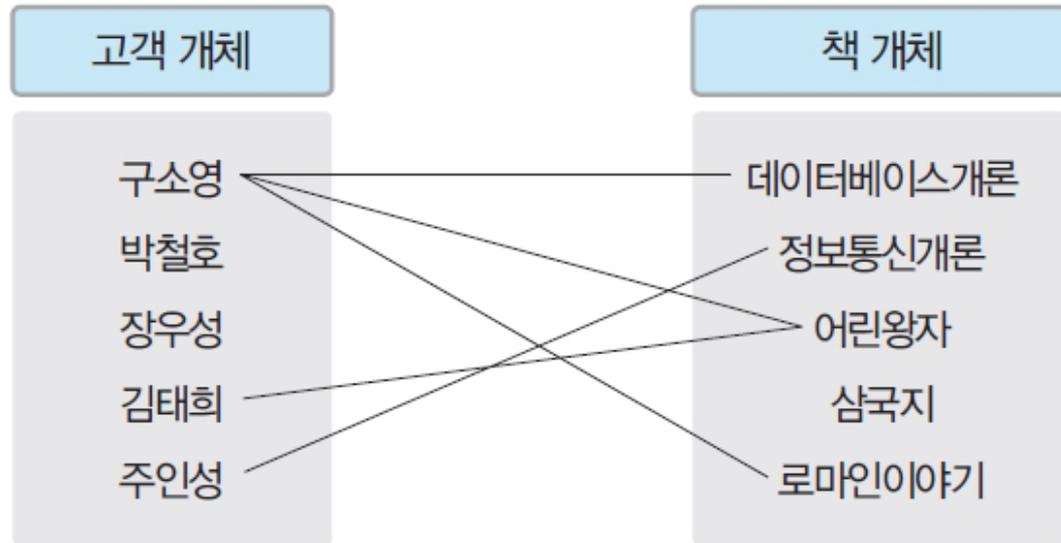


그림 4-16 다대다 관계의 예 : 고객과 책 개체의 구매 관계

관계의 참여 특성

필수적 참여 (전체 참여)

모든 개체 인스턴스가 관계에
반드시 참여해야 함

ERD: 이중선(=——)

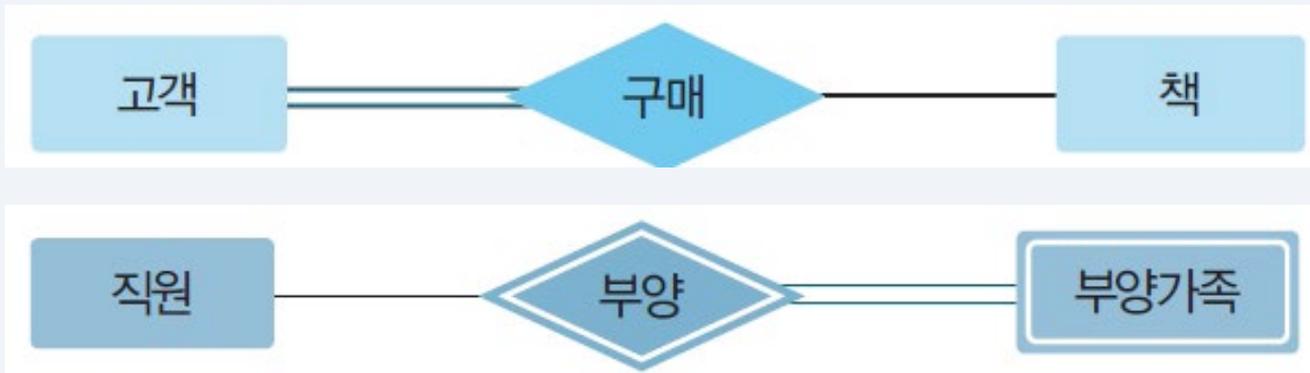
예) 모든 고객은 반드시 구매

선택적 참여 (부분 참여)

일부 인스턴스만
관계에 참여해도 됨

ERD: 일반 단선(—)

예) 모든 책이 구매될 필요 없음



관계의 종속성 — 강한 개체와 약한 개체

강한 개체 (Strong Entity)

독립적으로 존재 가능

ERD: 일반 사각형 □

예) 직원

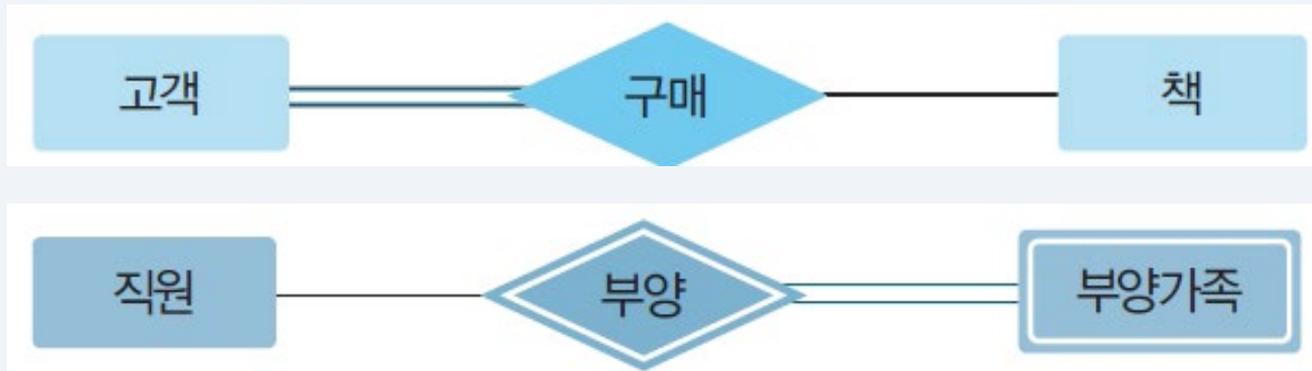
약한 개체 (Weak Entity)

다른 개체에 종속적 (강한 개체 없이 존재 불가)

ERD: 이중 사각형 □ + 이중 마름모

예) 부양가족 (직원에 종속)

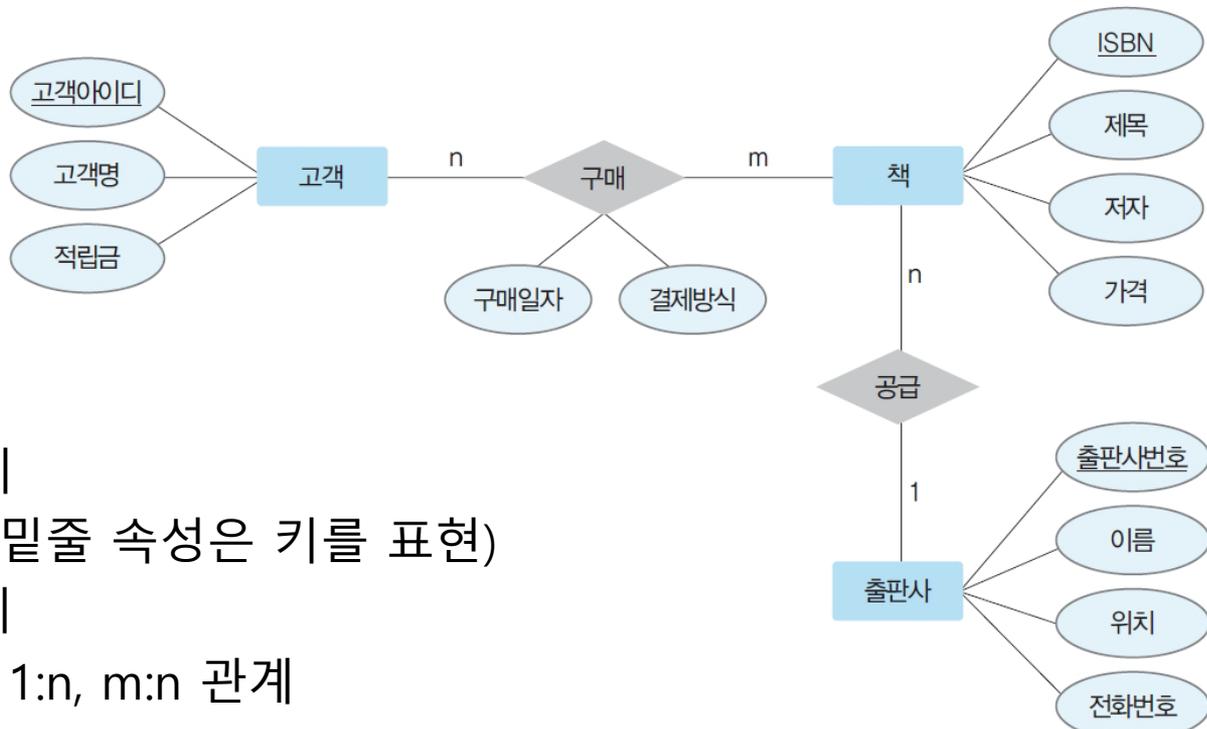
강한 개체 ↔ 약한 개체: 일반적으로 1:n 관계. 약한 개체는 강한 개체의 키를 포함하여 자신의 키를 구성.



ERD 기호 — 종합 정리

기호 이름	ERD 표현	의미
개체	사각형 □	현실의 구별되는 대상
속성	타원 ○	개체의 고유 특성
관계	마름모 ◇	개체 간 연관성
키 속성	밑줄 타원	인스턴스를 유일하게 식별
다중 값 속성	이중 타원 ◎	여러 값 가능
유도 속성	점선 타원	다른 속성에서 계산
약한 개체	이중 사각형 □	다른 개체에 종속적
필수적 참여	이중선 —	모든 인스턴스가 반드시 참여

ERD 종합 예제 — 서점 데이터베이스



- 사각형 : 개체
- 타원 : 속성 (밑줄 속성은 키를 표현)
- 마름모 : 관계
- 레이블 : 1:1, 1:n, m:n 관계

ERD 작성 단계

개체 식별

요구사항에서 명사를 찾아 개체 후보 도출

관계 파악

개체 간 연관성(동사)을 찾아 관계 도출

속성 정의

각 개체의 특성(속성)을 결정

카디널리티 결정

1:1 / 1:n / n:m 관계 유형 판별

키 속성 결정

인스턴스를 유일하게 식별하는 속성 선택

ERD 완성

기호를 사용하여 다이어그램 그리기

팁: 개체=명사(고객,상품) / 관계=동사(구매하다,수강하다)

03

논리적 데이터 모델

Logical Data Models

논리적 데이터 모델의 개념

논리적 데이터 모델

ERD(개념적 구조)를 DB에 저장할 형태로 표현한 논리적 구조 = 데이터베이스 스키마

관계 데이터 모델

Relational

2차원 테이블 형태
Oracle, MySQL, PostgreSQL
가장 많이 사용됨

계층 데이터 모델

Hierarchical

트리(Tree) 형태
부모-자식 관계 (1:n만)
n:m 표현 불가

네트워크 데이터 모델

Network

그래프(Graph) 형태
1:n 관계만 허용
구조가 복잡

현재 대부분의 DBMS는 관계 데이터 모델 사용 → 5장에서 본격 학습!

논리적 데이터 모델 — 3가지 비교

구분	관계 데이터 모델	계층 데이터 모델	네트워크 데이터 모델
구조	2차원 테이블	트리	그래프
1:1 / 1:n	O	O	O
n:m	O(가능)	X(불가)	X(직접 불가)
유연성	매우 높음	낮음	보통
데이터 조작	SQL(쉬움)	어려움	어려움
현재 사용	대부분의 DBMS	거의 사용 안 함	거의 사용 안 함
대표	Oracle, MySQL	IMS	IDMS

관계 데이터 모델 (Relational Data Model)

관계 데이터 모델 (Relational Data Model)

가장 많이 사용되는 논리적 데이터 모델. 데이터베이스의 논리적 구조를 2차원 테이블 형태로 표현. 1970년 에드거 코드(E.F. Codd)가 제안. 5장에서 본격 학습.

- 데이터를 행(row)과 열(column)로 구성된 테이블로 표현
- SQL(Structured Query Language)로 데이터 조작 – 배우기 쉽고 직관적
- n:m 관계도 연결 테이블을 통해 표현 가능
- 데이터 독립성이 높아 구조 변경에 유연
- 대표 DBMS: Oracle, MySQL, PostgreSQL, SQL Server, MariaDB

현재 시장의 거의 모든 DBMS가 관계 데이터 모델 기반 → 5장에서 릴레이션, 키, 무결성 제약조건 학습

계층 데이터 모델 (Hierarchical Data Model)

계층 데이터 모델 (Hierarchical Data Model)

데이터베이스의 논리적 구조를 트리(Tree) 형태로 표현.
루트(root) 역할을 하는 개체가 있으며, 사이클(cycle)이 존재하지 않음.

- 부모-자식 관계(상하 관계)로 구성 — 1:n 관계(1:1 포함)만 허용
- n:m 관계는 직접 표현 불가 (가장 큰 한계)
- 두 개체 사이에는 하나의 관계만 정의할 수 있음
- 데이터 삽입·삭제·수정·검색이 쉽지 않음
- 대표 DBMS: IBM IMS (1960년대) — 현재 거의 사용되지 않음

쇠퇴 원인: n:m 표현 불가 + 데이터 조작 어려움 + 구조 변경 비유연 → 관계 모델에 대체됨

네트워크 데이터 모델 (Network Data Model)

네트워크 데이터 모델 (Network Data Model)

데이터베이스의 논리적 구조를 네트워크(그래프) 형태로 표현.
개체 간에 일대다(1:n) 관계만 허용 (owner : member).

- 그래프(Graph) 형태 — 계층 모델보다 유연한 구조
- n:m 관계를 직접 표현할 수 없음 (1:n으로 분해해야 함)
- 두 개체 간의 관계를 여러 개 정의할 수 있으며, 이름으로 구별
- 구조가 복잡하며, 데이터 삽입·삭제·수정·검색이 쉽지 않음
- 대표 DBMS: IDMS — 현재 거의 사용되지 않음

계층 모델의 단점을 일부 보완했지만, 구조 복잡성 문제로 관계 모델에 대체됨

4장 전체 요약



데이터 모델링 = 현실 세계 → DB로 옮기는 과정 (개념적 + 논리적 모델링)



ER 모델 3요소: 개체(□) + 속성(o) + 관계(◇) → ERD로 시각화



속성 분류: 단일/다중값, 단순/복합, 유도, null, key



관계 유형: 1:1, 1:n, n:m / 참여: 필수/선택 / 종속: 강한/약한 개체



논리적 모델: 관계형(테이블) > 계층형(트리) > 네트워크(그래프)

다음 시간: 4장 연습문제 풀이. ER 모델과 ERD를 복습하고 오세요!



감사합니다

Thank You

박상돈 조교수 | 대전대학교 컴퓨터공학과