

2장. 데이터 모델

1 데이터 모델의 개요

● 정보화 시스템 구축

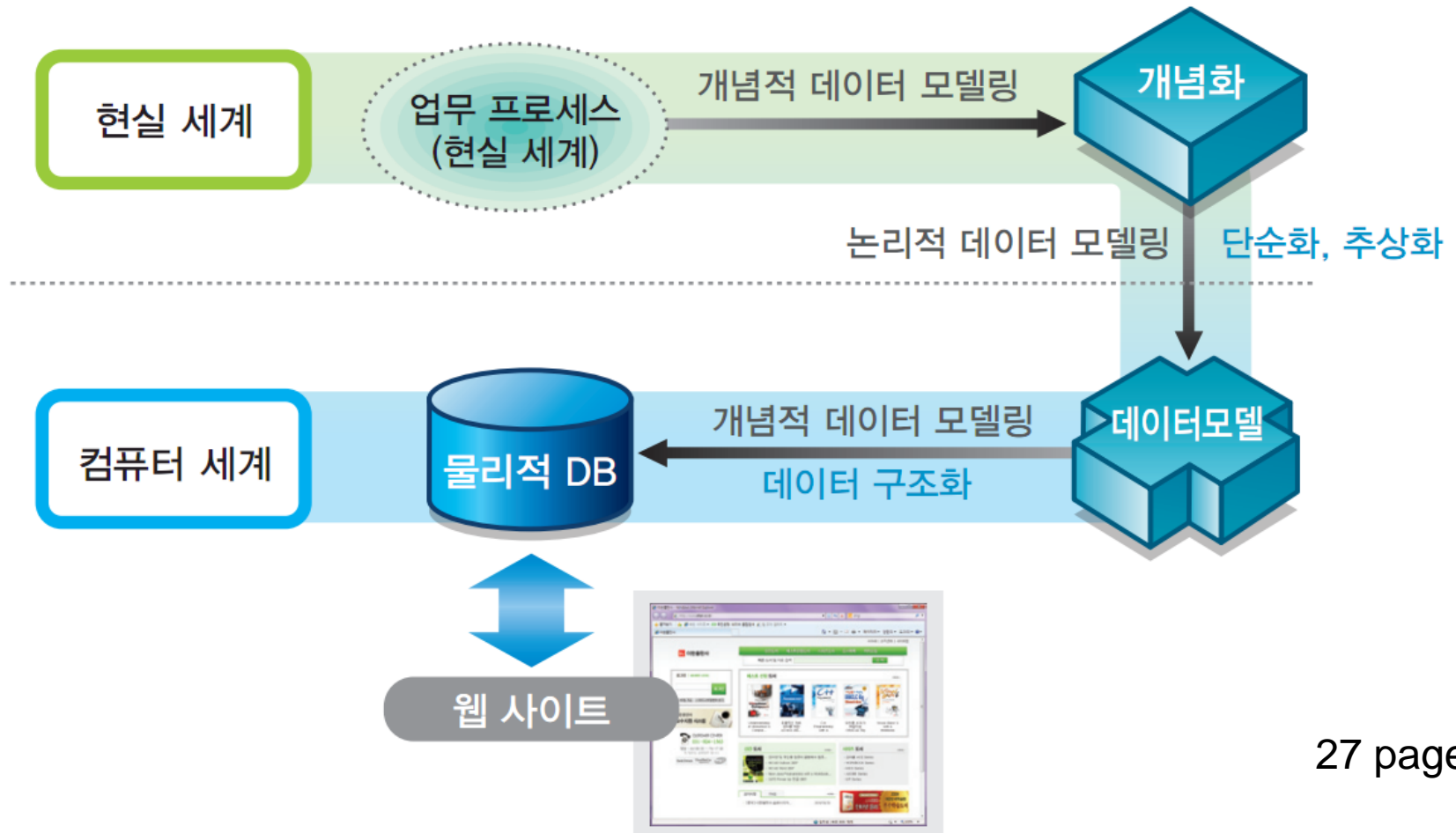
- 현실 세계에서 일어나는 사건들을 전산화하기 위한 것
- 정보화 시스템 과정에서 구축된 데이터베이스는 현실 세계의 특정 부분을 반영
- 현실 세계를 그대로 반영하지는 않음
- 현실 세계를 개념화, 단순화하여 가시적으로 표현

● 데이터 모델링(Data Modeling)

- 현실 세계의 업무적인 프로세스를 물리적으로 데이터베이스화하기 위한 과정
- 실체와 관계를 중심으로 체계적으로 표현하고 문서화하는 기법
- 정보 시스템의 중심을 데이터의 관점에서 접근하는 데이터 중심의 분석 방법
- 어떤 데이터가 존재하며 업무에서 필요로 하는 정보가 무엇인지를 분석하는 과정

1 데이터 모델의 개요

● 데이터 모델링



27 page

1 데이터 모델의 개요

1-1 데이터 모델의 구성요소

- 데이터 모델의 구성요소

- 구조
- 연산
- 제약조건

정의

$D = \langle S, O, C \rangle$

S:Structure, O:Operation, C:Constraint

1 데이터 모델의 개요

1-1 데이터 모델의 구성요소

- 구조(S : Structure)

- 데이터의 정적 성질(static properties)로서 데이터베이스에서 표현될 대상으로서의 엔티티 타입과 이들 간의 관계를 명시한 것

- 연산(O : Operation)

- 데이터의 동적 성질(dynamic properties)로서 연산은 엔티티 인스턴스에 적용 가능한 연산에 대한 명세

- 제약조건(C : Constraint)

- 데이터베이스에 저장하는 데이터가 무결하도록 하기 위해서 사용

1 데이터 모델의 개요

1-2 엔티티, 관계, 속성

- 데이터 모델링을 하기 위해서는 엔티티, 관계, 속성에 대한 이해가 필요

1. 업무가 관여하는 어떤 것 (ENTITY)
2. 업무가 관여하는 어떤 것 간의 관계성 (RELATIONSHIP)
3. 어떤 것이 가지는 성격 (ATTRIBUTE)

1 데이터 모델의 개요

1-2 엔티티, 관계, 속성

● 예문

- 이도령과 성춘향이 있고 둘은 사랑하는 사이다. 이도령은 키가 186이고 성격이 까칠하고, 성춘향은 키가 163이고 세심하고 활발한 성격이다.



30 page

개념	예문에 해당하는 내용	관계형 모델링
어떤 것		엔티티(ENTITY)
어떤 것 간의 관계성		관계(RELATION)
어떤 것의 성격		속성(ATTRIBUTE)

1 데이터 모델의 개요

1-2 엔티티, 관계, 속성



- 업무가 관여하는 어떤 것(ENTITY)

- "이도령", "성춘향"

- 업무가 관여하는 어떤 것 간의 관계성(RELATIONSHIP)

- "사랑하는 사이"

- 어떤 것이 가지는 성격(ATTRIBUTE)

- "이도령" : "키가 186", "까칠"
- "성춘향" : "키가 163", "세심, 활달"

1 데이터 모델의 개요

1-2 엔티티, 관계, 속성

● 엔티티(Entity)

- 데이터베이스에 자료로 표현하려는 것
- 사람이 생각하는 개념이나 정보단위 같은 현실 세계의 대상체
- 유형, 무형의 정보
- 서로 연관된 몇 개의 속성으로 구성

예

부서 엔티티, 사원 엔티티

1 데이터 모델의 개요

1-2 엔티티, 관계, 속성

● 속성(Attribute)

- 데이터의 가장 작은 논리적 단위
- 하나의 엔티티는 한 개 이상의 속성으로 구성
- 각 속성은 엔티티의 특성, 상태 등을 기술

예

부서 엔티티의 속성 : 부서 번호, 부서명, 위치

사원 엔티티의 속성 : 사원번호, 이름, 주소, 소속부서

1 데이터 모델의 개요

1-2 엔티티, 관계, 속성

● 관계성(Relationship)

- 엔티티와 엔티티 간의 연관성을 관계라고 함
- 일반적으로 동사형으로 표현

예

사원이 부서에 소속되어 있다.

1 데이터 모델의 개요

1-3 개념적 데이터 모델과 논리적 데이터 모델

- 개념적 데이터 모델과 논리적 데이터 모델



32 page

1 데이터 모델의 개요

1-3 개념적 데이터 모델과 논리적 데이터 모델

● 개념적 데이터 모델(conceptual data model)

- 현실 세계에 존재하는 데이터를 의미 있는 엔티티와 엔티티 내의 공통된 속성과 엔티티들 사이의 관계를 정의하는 추상화 과정
- 현실 세계를 추상화하여 개념적으로 표현하여 이해하기 쉽게 할 뿐 아니라 의사소통을 원활하게 해주는 과정을 의미
- 정보 모델링이라고도 함
- 개체 관계 모델(E-R : Entity-Relationship Model) : 기술된 엔티티 타입과 이 엔티티 타입들 간의 관계를 이용하여 현실 세계를 표현하는 가장 대표적인 개념적 데이터 모델

● 논리적 데이터 모델(logical data model)

- 개념적 데이터 모델은 DBMS가 직접 이해할 수 없음
- 컴퓨터가 이해할 수 있도록 논리적 데이터 모델로 변환해야 함
- 논리적 데이터 모델은 데이터 모델링이라고도 함
- 관계 데이터 모델, 계층 데이터 모델, 네트워크 데이터 모델 등이 있음

2 개념적 데이터 모델

● 개념적 데이터 모델링

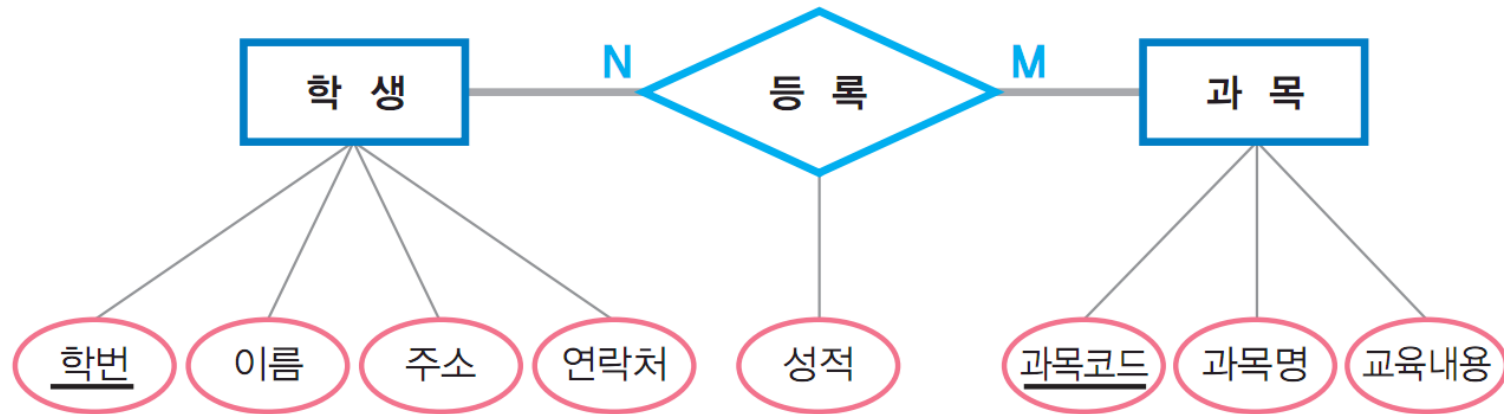
- 현실 세계에 존재하는 엔티티를 인간이 이해할 수 있는 정보 구조(Information Structure)로 표현하는 과정
- 정보 모델링(Information Modeling)이라고도 함
- 대표적인 것 : 1976년 Peter Chen에 의해 제안된 개체 관계 모델(E-R, Entity-Relationship Model)

● E-R 다이어그램(Entity-Relationship Diagram)

- 엔티티와 이들 간의 관계성을 알기 쉽게 미리 약속된 도형을 사용하여 일목요연하게 표현

2 개념적 데이터 모델

● E-R 다이어그램 예



34 page

2 개념적 데이터 모델

● ER 다이어그램 표기법

	엔티티 타입		속성
	약한 엔티티 타입		키 속성
	관계 타입		부분 키 속성
	식별 관계 타입		다중 값 속성
	ISA 관계		복합 속성
	전체 참여 엔티티 타입		유도 속성

35 page

2 개념적 데이터 모델

2-1 엔티티와 엔티티 타입

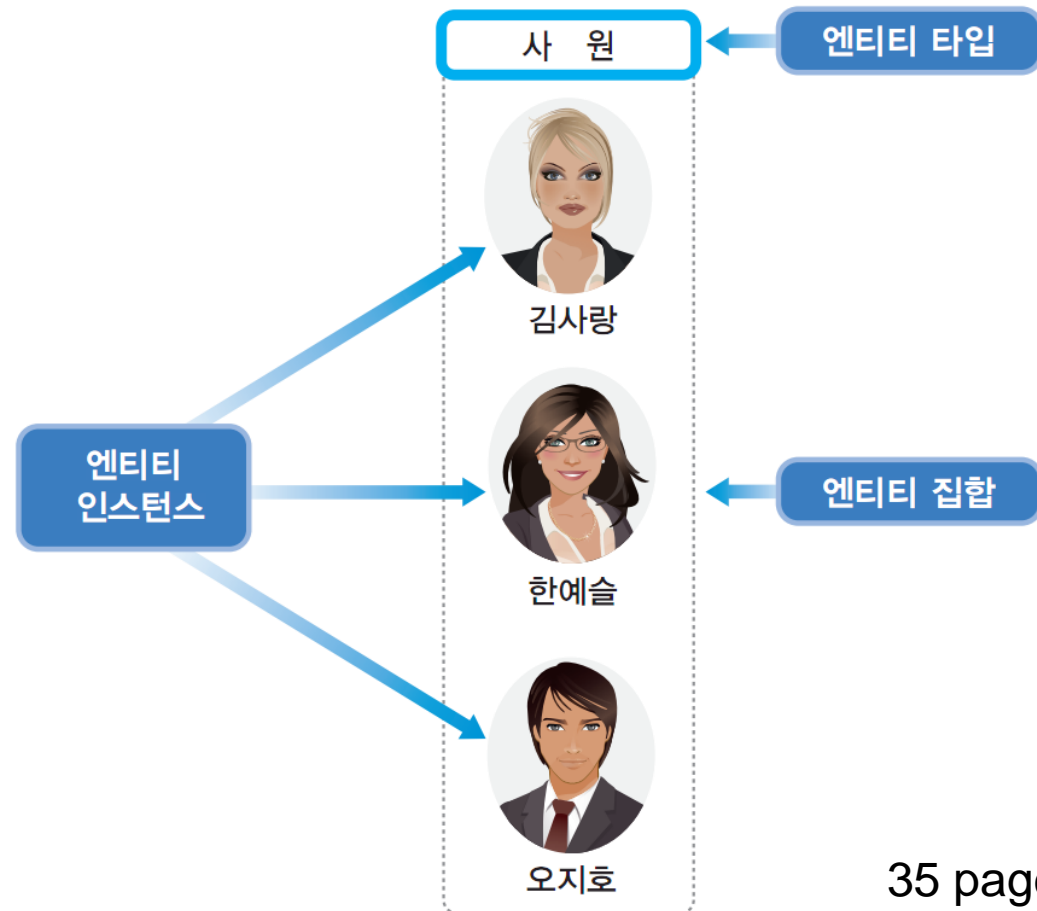
● 엔티티 타입(entity type)

- 정보가 저장될 수 있는 사람, 장소, 사물, 사건 등과 같이 독립적으로 존재하면서 고유하게 식별이 가능한 실세계의 엔티티
- 사원처럼 실체가 있는 것도 있지만 생각이나 개념과 같이 추상적인 것
- 동일한 속성들을 가진 엔티티들의 틀
- 엔티티들은 엔티티 타입으로 분류

2 개념적 데이터 모델

2-1 엔티티와 엔티티 타입

- 엔티티 타입, 엔티티 집합, 엔티티 인스턴스



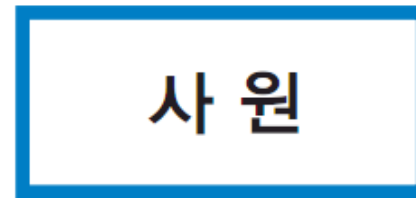
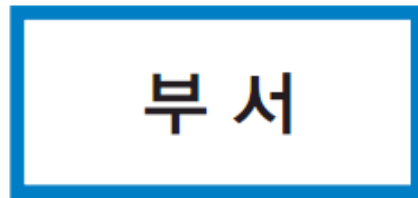
35 page

2 개념적 데이터 모델

2-1 엔티티와 엔티티 타입

- 엔티티의 집합(set of entities)

- 동일한 속성들을 가진 엔티티들의 모임
- 각 엔티티는 속성(attribute)으로 알려진 특성들로 정의되며 E-R 다이어그램에서 엔티티 타입은 직사각형으로 표시



- 엔티티 인스턴스(entity instance)

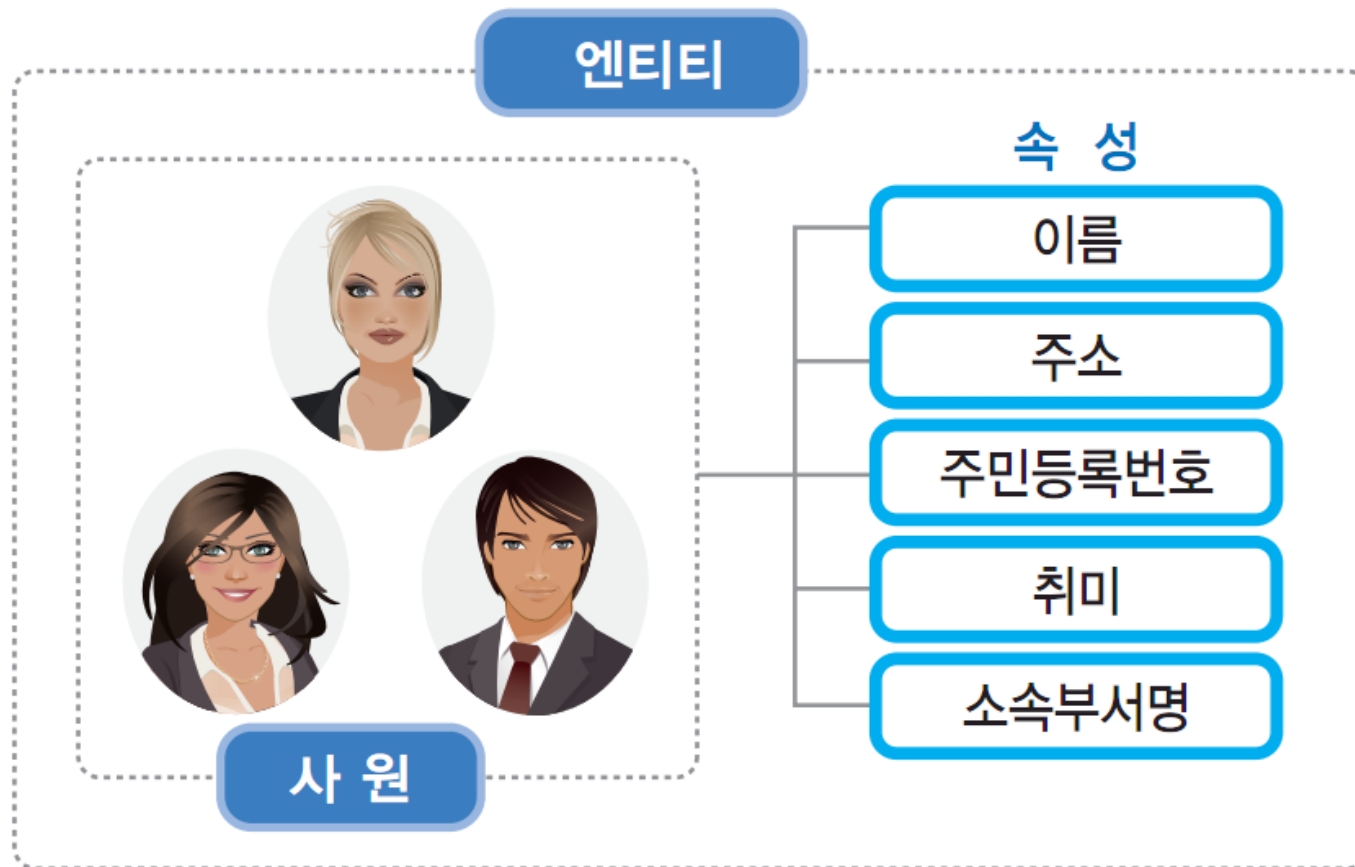
- 엔티티 타입을 구성하는 원소
- 엔티티는 엔티티 타입에 속하는 하나의 인스턴스

2 개념적 데이터 모델

2-2 속성

- 속성(attribute)

- 정보의 요소로서 관리되는 항목으로 엔티티의 성질, 분류, 수량, 상태, 특성 등을 구체적으로 나타내는 세부 항목



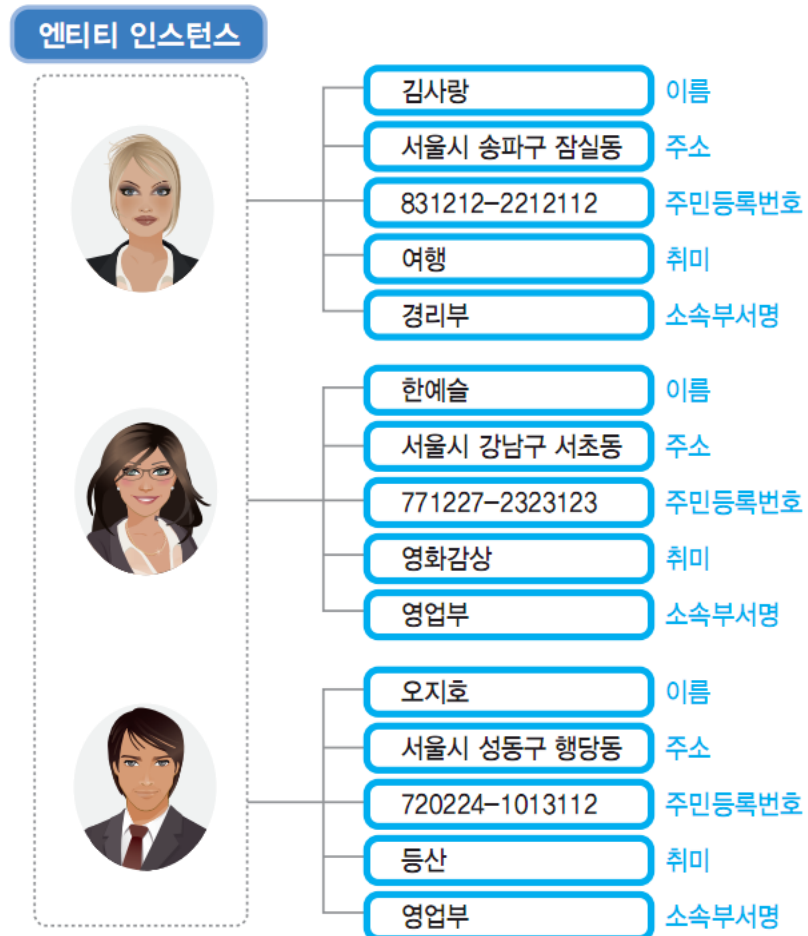
36 page

2 개념적 데이터 모델

2-2 속성

● 속성값(attribute value)

- 엔티티의 특성이나 상태가 현실화된 값



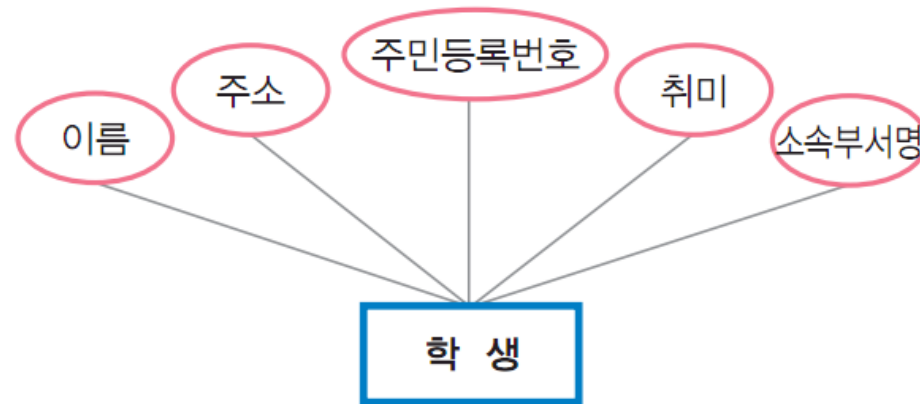
37 page

2 개념적 데이터 모델

2-2 속성

● 엔티티 타입과 속성

- ERD에서 속성은 엔티티 타입을 나타내는 직사각형에 실선으로 연결된 타원형으로 표현



37 page

● 속성(attribute)의 유형

- 단순 속성과 복합 속성
- 단일 값 속성과 다중 값 속성
- 유도 속성과 저장 속성

2 개념적 데이터 모델

2-2 속성

- 단순 속성(simple attribute)

- 더 이상 작은 구성요소로 분해할 수 없는 속성이고

- 복합 속성(composite attribute)

- 몇 개의 기본적인 단순 속성으로 분해할 수 있는 속성



38 page

2 개념적 데이터 모델

2-2 속성

- 단일 값 속성(single-valued attribute)

- 각 엔티티에 대해 하나의 값만 갖는 것

- 다중 값 속성(multi-valued attribute)

- 한 엔티티에 대해서 여러 개의 값을 갖는 것



38 page

2 개념적 데이터 모델

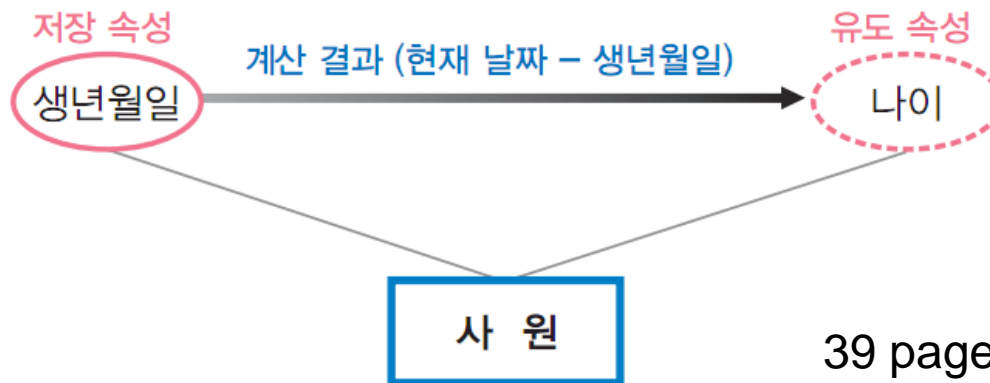
2-2 속성

- 유도 속성(derived attribute)

- 속성의 값이 다른 속성이나 엔티티가 가지고 있는 값으로부터 유도되어 결정되는 경우

- 저장 속성(stored attribute)

- 유도 속성을 생성하는데 사용된 속성



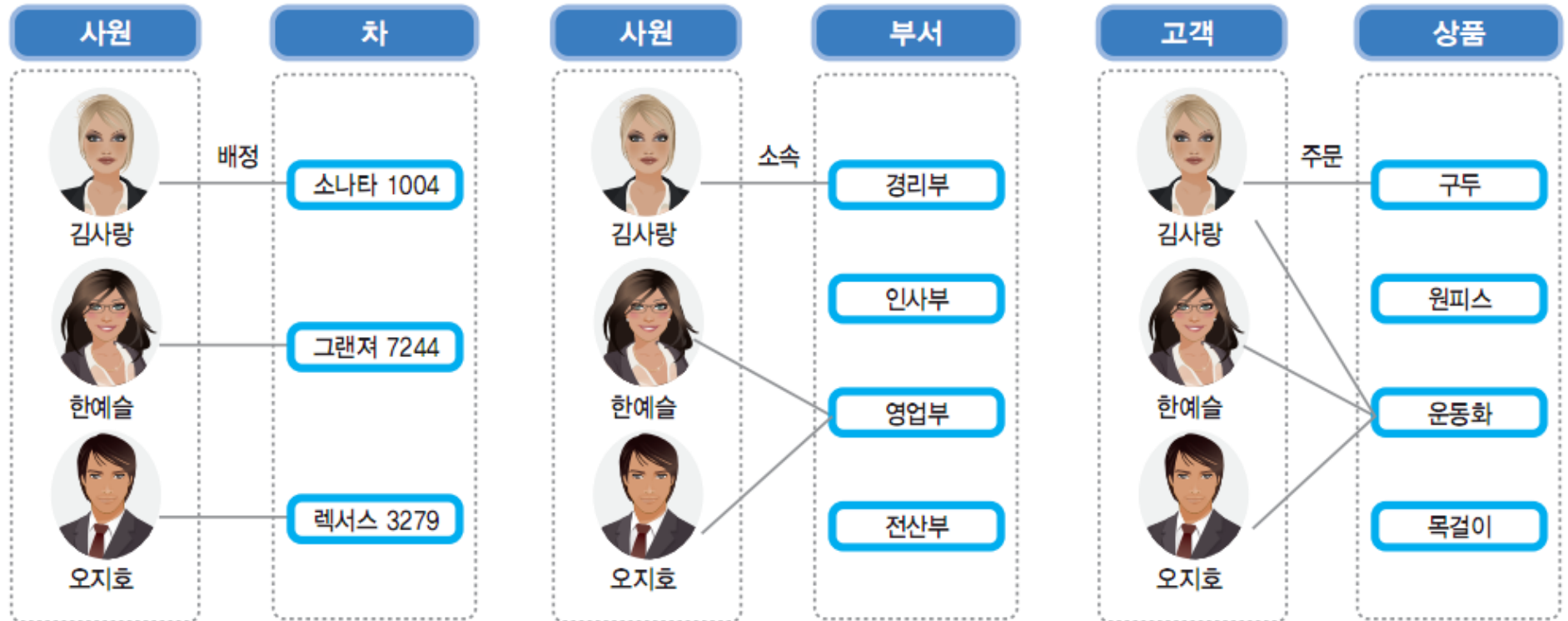
- 널(null) 값

- 어느 특정 속성에 대한 값을 가지고 있지 않을 때 이를 명시적으로 표시하기 위해 사용

2 개념적 데이터 모델

2-3 관계성 타입

- 엔티티 사이의 관계



40 page

2 개념적 데이터 모델

2-3 관계성 타입

● 관계성 타입(relationship type)

- 엔티티-관계 모델에서 엔티티 사이의 연관성을 표현하는 개념
- 두 개의 엔티티 타입 사이의 업무적인 연관성을 논리적으로 표현한 것
- 관계성을 정의할 때 동사구로 관계를 정의
- 엔티티 집합(타입)들 사이의 대응(correspondence), 즉 사상(mapping)을 의미
- ERD에서 엔티티들 사이의 관계 타입은 마름모를 사용하여 표현한 후 그 관계성에 연관된 엔티티에 선으로 연결하여 표시

● 관계성 집합(relationship set)

- 한 관계 타입에 속하는 모든 관계 인스턴스(relationship instance)들의 집합을 말함



40 page

2 개념적 데이터 모델

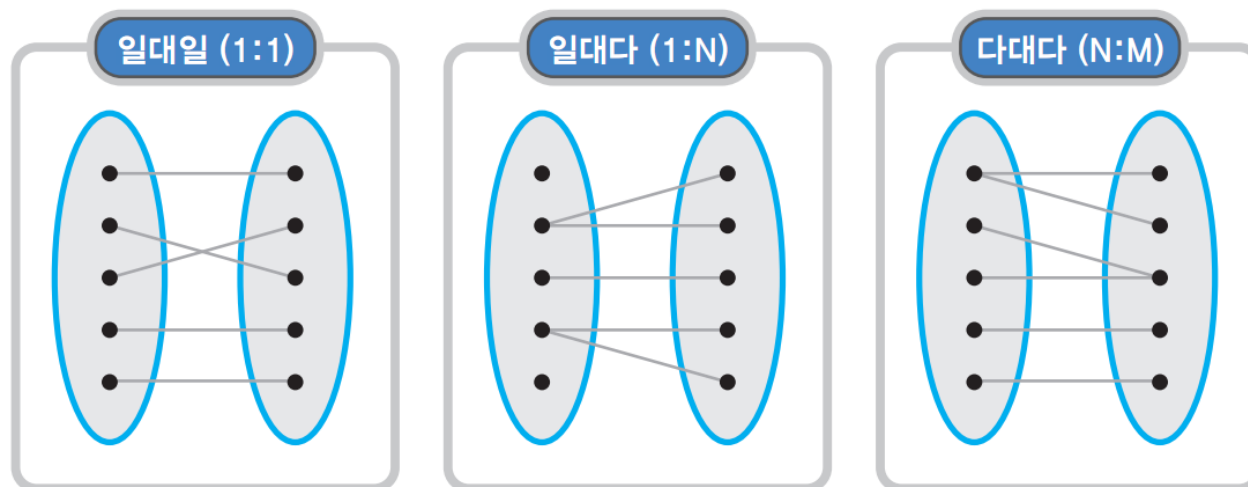
2-3 관계성 타입

● 관계성의 유형

- 관계에 참여하는 하나의 엔티티에 대해 다른 엔티티가 몇 개 참여하는지를 나타내는 것
- 카디널리티(Cardinality : 관계의 대응 엔티티 수)라고도 함

● 카디널리티

- 하나의 관계에 실제로 참여할 수 있는 인스턴스의 수를 의미
- 표현방법 : 일대일(1:1), 일대다(1:N), 다대다(N:M)

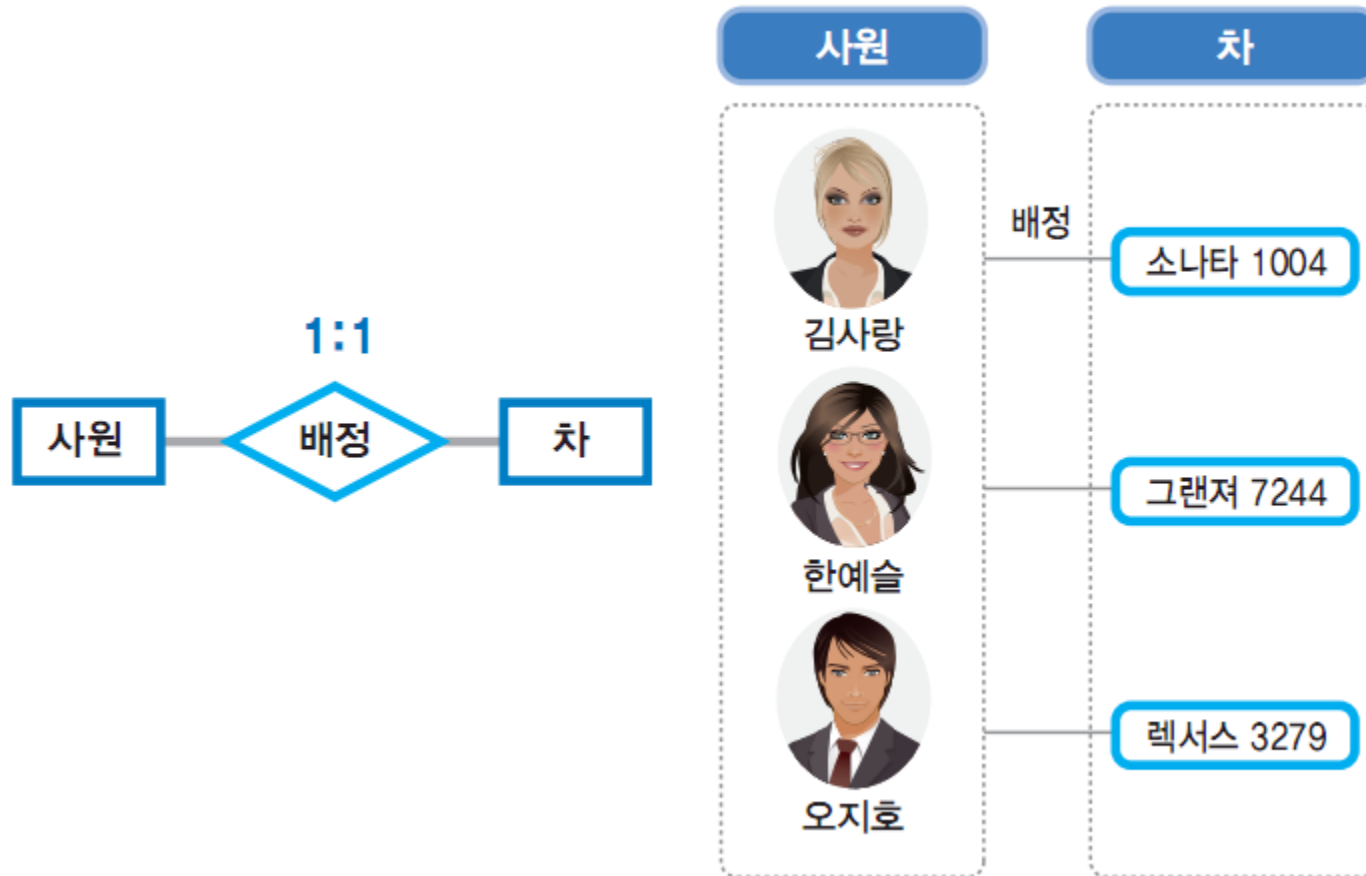


41 page

2 개념적 데이터 모델

2-3 관계성 타입

- 일대일(One To One) 관계(1:1)

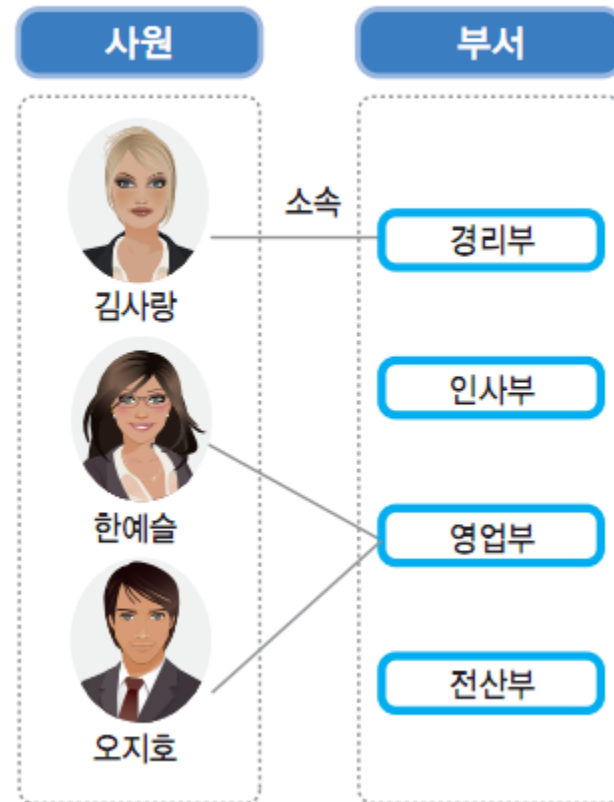


41 page

2 개념적 데이터 모델

2-3 관계성 타입

- 일대다(One To Many) 관계(1:N)

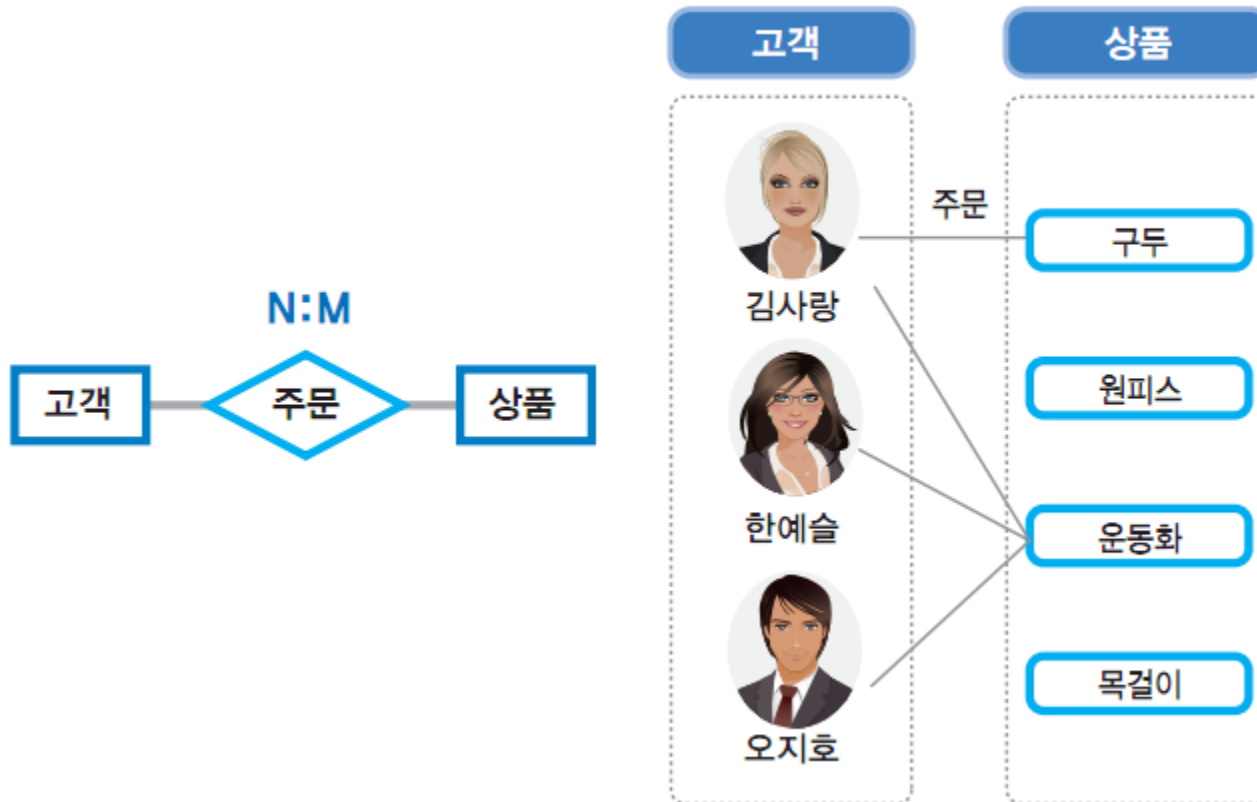


42 page

2 개념적 데이터 모델

2-3 관계 타입

- 다대다(Many To Many) 관계(N : M)



42 page

2 개념적 데이터 모델

2-3 관계성 타입

- 카디널리티 비율의 최소값과 최대값

- 비율에 최소값과 최대값을 표시하여 관계의 유형을 좀 더 명확하게 나타낼 수 있음.
- ER 다이어그램에서 최소값과 최대값은 관계성 타입과 엔티티 타입을 연결하는 실선 위에 (min, max) 형태로 표기



43 page

2 개념적 데이터 모델

2-3 관계 타입

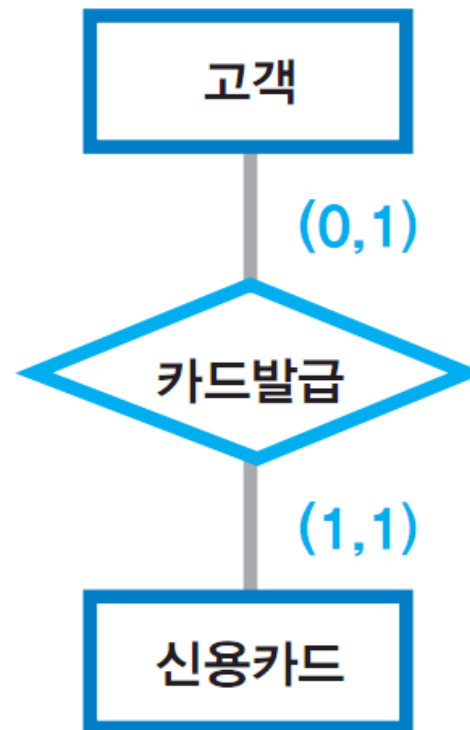
- 카디널리티 유형

(min1, max1)	(min2, max2)	관계
(0, 1)	(0, 1)	1 : 1
(0, *)	(0, 1)	1 : N
(0, *)	(0, *)	N : M

2 개념적 데이터 모델

2-3 관계 타입

- 카디널리티 비율의 최소값과 최대값의 예



44 page

2 개념적 데이터 모델

2-4 엔티티의 키

● 키(Key) 속성

- 간단히 키라고 함
- 각 엔티티 인스턴스를 유일하게 식별하는 데 사용

● 예

- “한예슬”이란 사원의 주소가 필요한데 “한예슬”이란 이름을 갖고 있는 사원이 2명이라면 이름만 가지고는 둘 중에 어느 사원을 지칭하는 것인지 알 수 없으므로, 다른 키(예: 사번)가 필요하다.



2 개념적 데이터 모델

2-4 엔티티의 키

- 엔티티 키의 예



2 개념적 데이터 모델

2-5 강한 엔티티 타입과 약한 엔티티 타입

- 강한 엔티티 타입(strong entity type)

- 엔티티 타입 내에서 자신의 키를 사용하여 고유하게 엔티티들을 식별할 수 있는 엔티티 타입
- 독립 엔티티, 부모 엔티티라, 정규 엔티티 타입이라고도 함
- 독자적으로 존재

- 약한 엔티티 타입(weak entity type)

- 키를 형성하기에 충분한 속성들을 갖지 못한 것
- 자기 자신의 속성만으로는 키를 명세할 수 없는 엔티티
- 종속 엔티티, 자식 엔티티라고도 함
- 소유 엔티티 타입이 있어야 존재
- 소유 엔티티 타입을 식별 엔티티 타입(identifying entity type)이라고도 부름

2 개념적 데이터 모델

2-5 강한 엔티티 타입과 약한 엔티티 타입

- 약한 엔티티의 예

- 아래 그림의 엔티티에서 이름이 "이지수"인 사람을 조회해 보자.



46 page

2 개념적 데이터 모델

2-5 강한 엔티티 타입과 약한 엔티티 타입

- 구별자(discriminator)

- 약한 엔티티 타입을 구별하기 위해서 사용되는 이름
- 단독으로 엔티티를 구분할 수는 없다,
- 부분키(partial key)라고도 한다.
- 구별자는 강한 엔티티와 연관된 약한 엔티티 집합 내에서만 서로 구별할 수 있다.

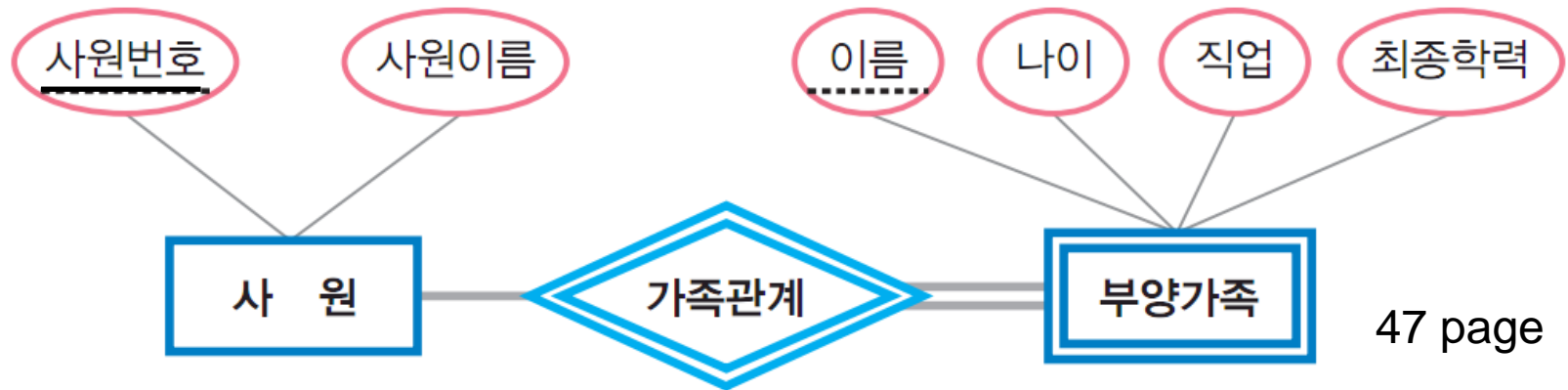
2 개념적 데이터 모델

2-5 강한 엔티티 타입과 약한 엔티티 타입

- 부양가족 엔티티(약한 엔티티 타입) 키는 사원 엔티티(강한 엔티티 타입) 키와 부양가족 부분키(구별자)로 구성

예

{사원번호, 이름}



47 page

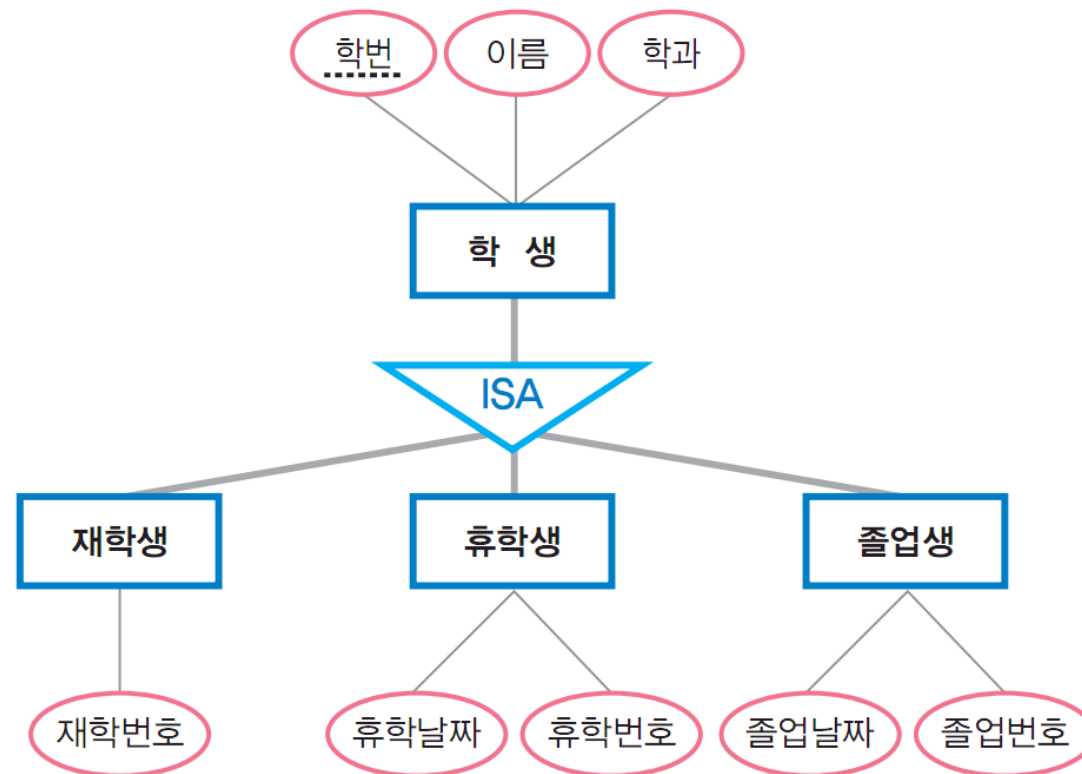
- 부양가족 엔티티가 가족 관계에 이중 링크로 연결된 것은 부양가족 엔티티가 전체 참여(total participation)를 하고 있다는 것을 나타내기 위한 것

2 개념적 데이터 모델

2-6 ISA 관계

● ISA 관계

- 상위 엔티티와 하위 엔티티 간의 관계를 의미
- 특정 엔티티는 서로 구별되는 여러 하위 엔티티로 나누어질 수 있음



47 page

2 개념적 데이터 모델

ER 도표 예

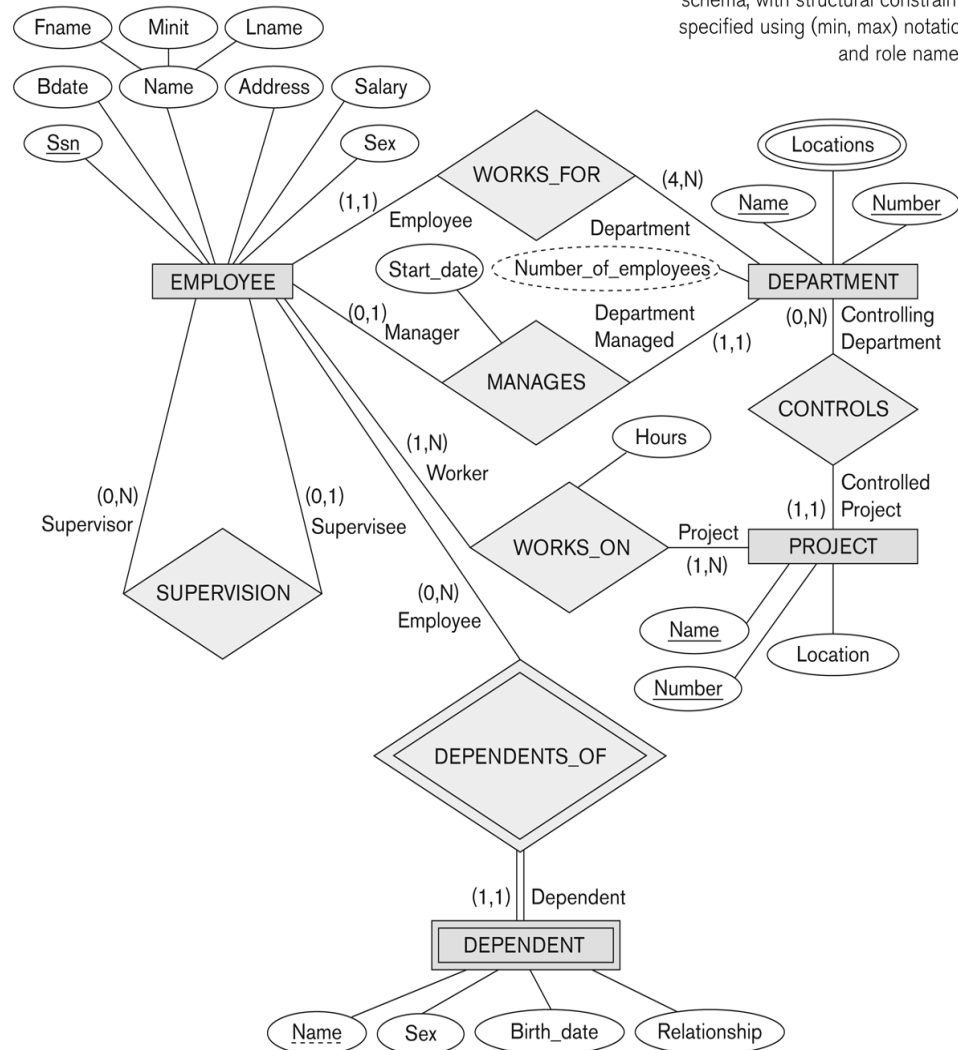


Figure 3.15

ER diagrams for the company schema, with structural constraints specified using (min, max) notation and role names.

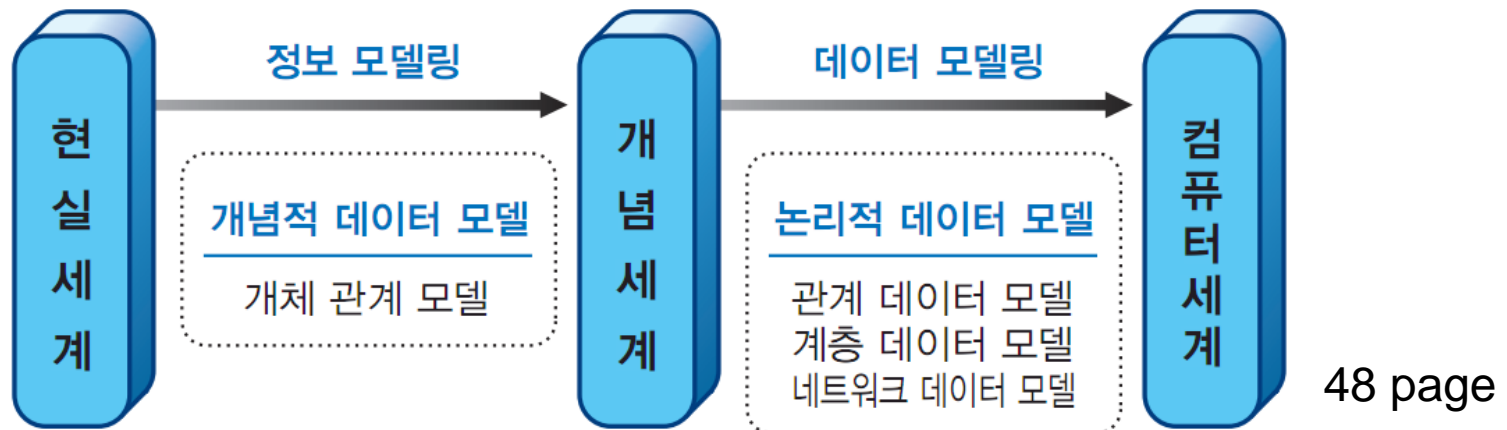
3 논리적 데이터 모델

● 논리적 데이터 모델링

- 개념적 모델링 과정에서 얻은 개념적 구조를 컴퓨터 환경에 맞도록 변환하는 과정
- 데이터 모델링(Data Modeling)이라고도 함

● 논리적 데이터 모델링 목적

- 개념적 설계에서 추출된 실체와 속성들의 관계를 관계형 데이터베이스 이론에 맞게 구조적으로 설계하는 단계(스키마의 설계)
- 정확한 업무 분석을 통한 자료의 흐름을 분석하여 실체와 속성들의 관계를 구조적으로 설계

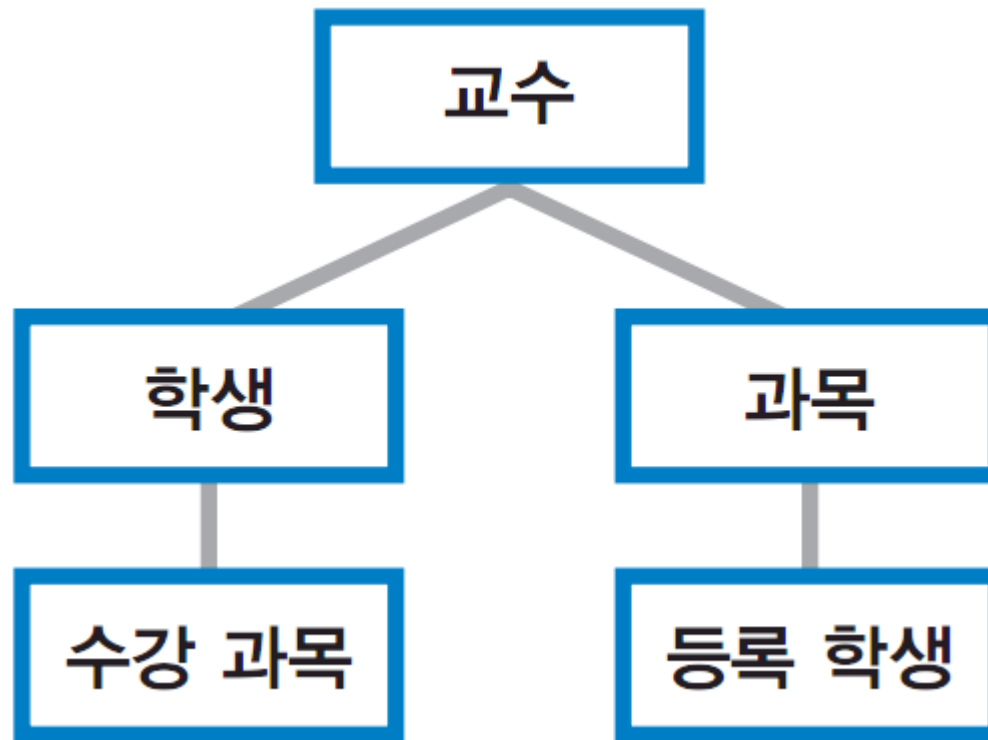


3 논리적 데이터 모델

3-1 계층 데이터 모델

- 계층(Hierarchical) 데이터 모델

- 데이터를 저장하는 단위(Entity)의 구조가 상하 종속적인 관계로 구성



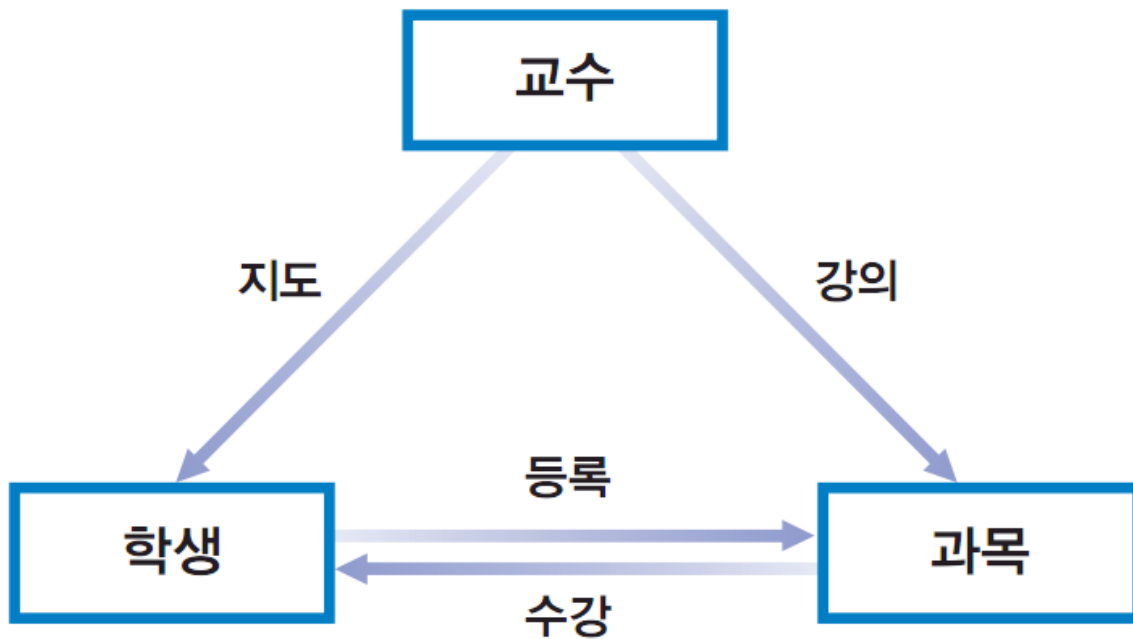
49 page

3 논리적 데이터 모델

3-2 네트워크 데이터 모델

● 네트워크(network) 데이터 모델

- CODASYL이 제안(CODASYL DBTG 모델이라고도 함)
- 그래프를 이용해서 데이터 논리구조를 표현한 데이터 모델
- 상위와 하위 레코드 사이에서 다대다(N:M) 대응 관계를 만족하는 구조



50 page

3 논리적 데이터 모델

3-3 관계 데이터 모델

● 관계(Relational) 데이터 모델

- 계층 모델과 망 모델의 복잡한 구조를 단순화시킨 모델
- 표(Table)를 이용해서 데이터 상호관계를 정의하는 구조

● 관계 스키마

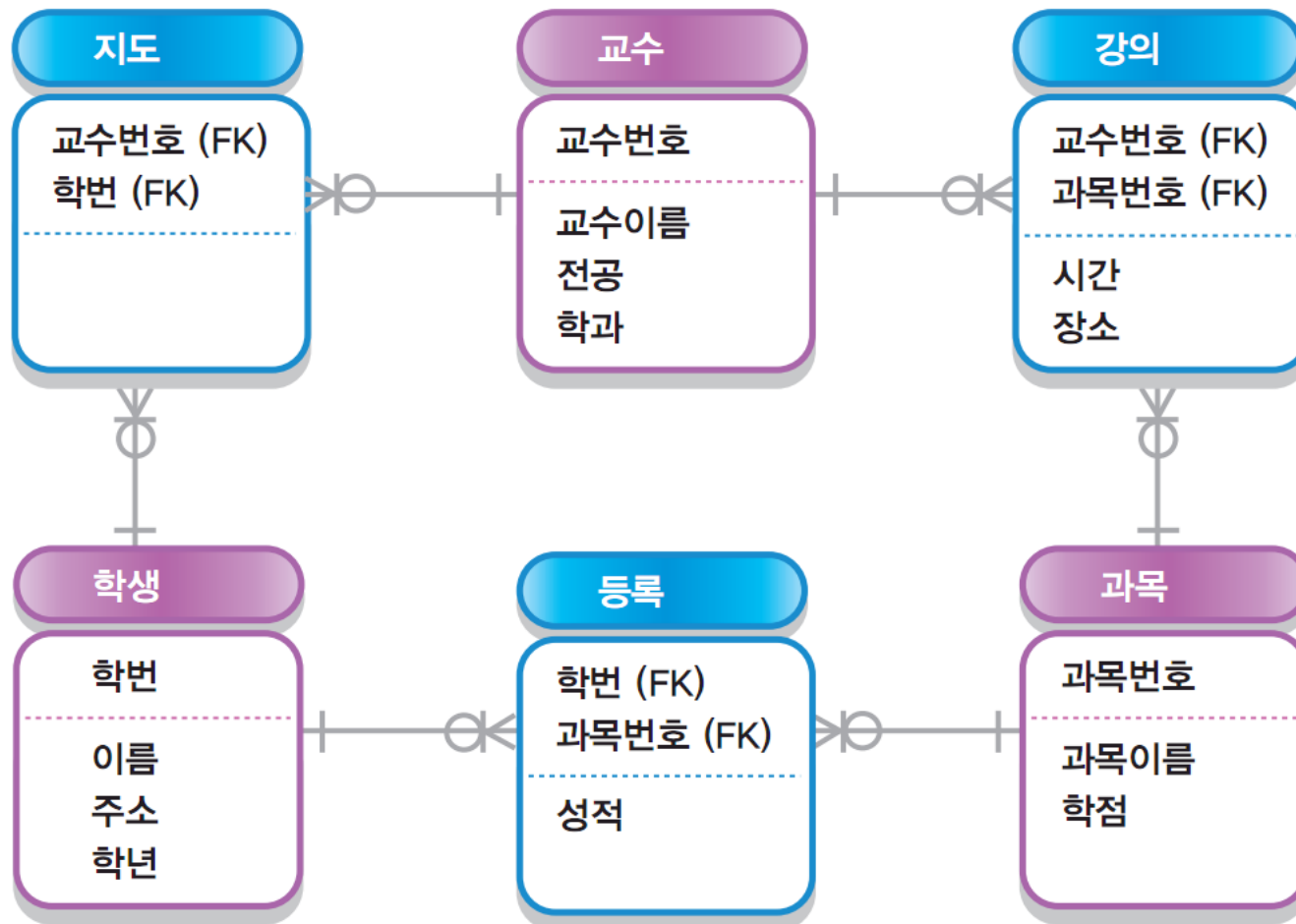
학생	학번	이름	주소	학년
교수	교수 번호	교수 이름	전공	학과
과목	과목 번호	과목 이름	학점	
지도	교수번호		학번	
등록	학번	과목 번호	성적	
강의	교수 번호	과목 번호	시간	장소

51 page

3 논리적 데이터 모델

3-3 관계 데이터 모델

- 다음은 관계 데이터 모델을 ERD 표기법 중 정보 공학 표기법으로 나타낸 것



51 page