

The background features a series of 3D cubes of varying heights arranged in a row, creating a skyline effect. A large, thin black circle is centered on the page, overlapping the cubes and the text area. The text is contained within a rectangular frame that is partially covered by the circle.

Chapter 03

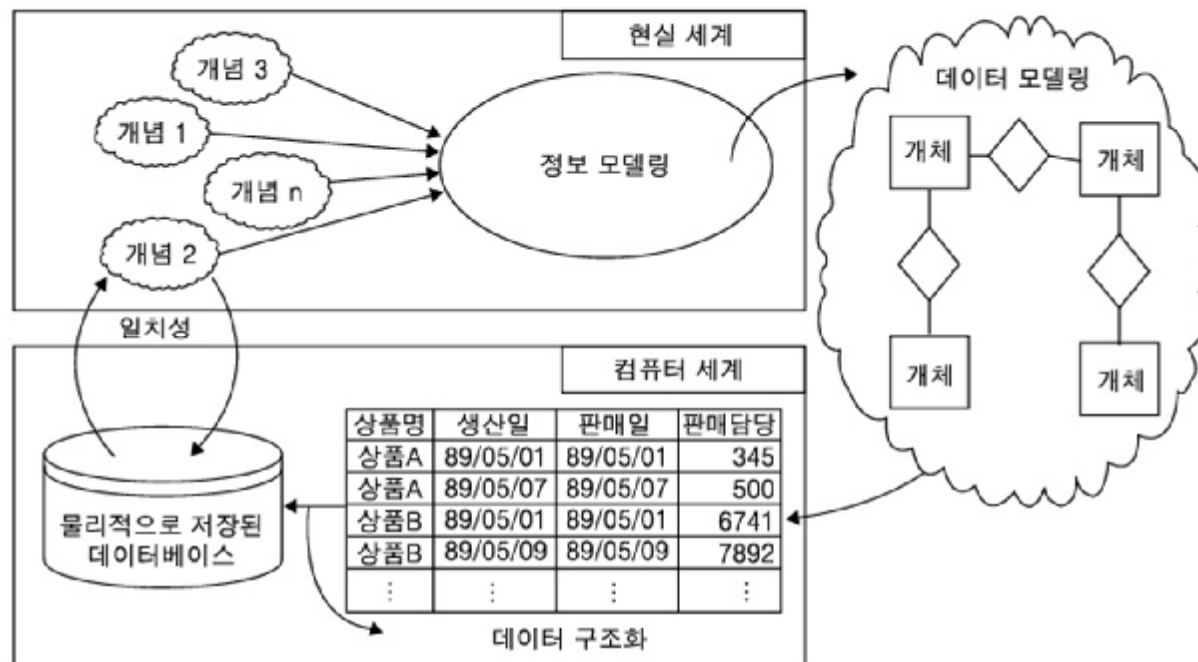
데이터 모델링

데이터베이스 응용

□ 데이터 모델링

● 데이터 모델링(Data Modelling)

- * 복잡한 실세계를 단순화하여 실세계에 존재하는 객체들을 식별하여 이들 객체와 객체 사이의 관계를 정의함으로써 컴퓨터상의 데이터베이스를 추상화된 개념으로 이해하기 쉽게 할 뿐만 아니라 사용자들 사이의 의사소통을 원활히 할 수 있도록 도와주는 도구

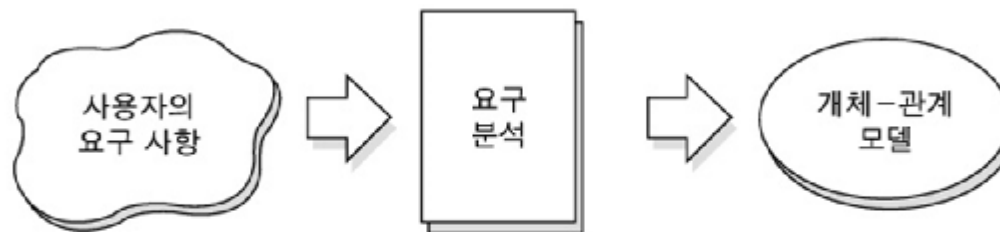


데이터 모델링

□ 데이터 모델링

● 엔터티-관계(Entity-Relationship) 모델

- * 현실 세계의 많은 데이터 중에서 관심의 대상이 되는 데이터를 언어보다 좀더 형식화된 다이어그램을 사용하여 표현한 것
- * 1976년 Peter Chen에 의해 제안
 - ☑ 데이터에 대해 관리자, 사용자, 프로그래머들이 서로 다르게 인식되고 있는 뷰들을 하나로 통합할 수 있는 단일화된 설계안을 만들기 위해서 사용.
 - ☑ 서로 다른 뷰들을 충족시킬 수 있는 데이터 처리와 제약조건 등의 요구사항들을 정의하기 위해 사용.



사용자의 요구로부터 개체-관계 모델의 설계

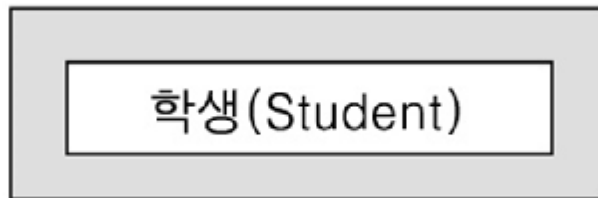
EP 모델링

□ 데이터 모델링

● 구성 요소

* 엔터티(Entity)

- ☑ 데이터 수집의 대상이 되는 정보 세계에 존재하는 사물(thing)
- ☑ 종류
 - ✍ 개념적 엔터티 : 장소, 사건 등과 같은 눈에 보이지 않는 것
 - ✍ 물리적 엔터티 : 물건 등과 같은 눈에 보이는 것. 즉 현실 세계에 존재하는 사물.
- ☑ 각 엔터티는 속성(attribute)으로 알려진 특성들로 정의.
- ☑ 엔터티-관계 다이어그램(ERD)에서 엔터티 집합은 직사각형으로 표시

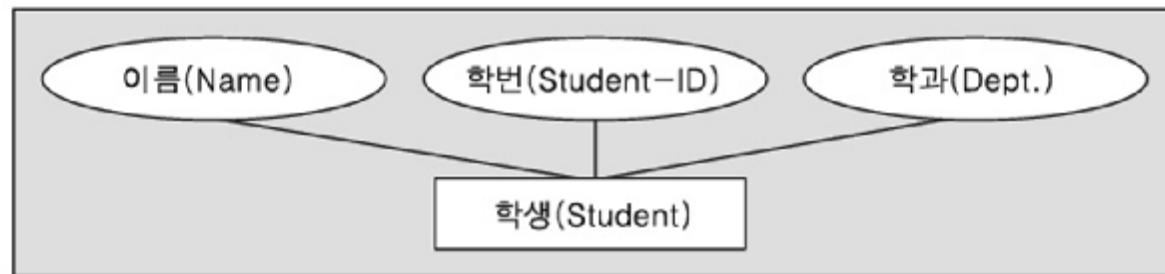


학생($\Sigma TY \Delta ENT$) 엔터티 집합

□ 데이터 모델링

* 속성(Attribute)

- ☑ 엔티티를 나타내는 특성
- ☑ ERD에서 속성은 엔티티 집합을 나타내는 직사각형에 실선으로 연결된 타원형으로 표현
- ☑ 도메인(domain)
 - ✎ 각 속성마다 가질 수 있는 값들의 범위



학생 엔티티 집합의 속성들

□ 데이터 모델링

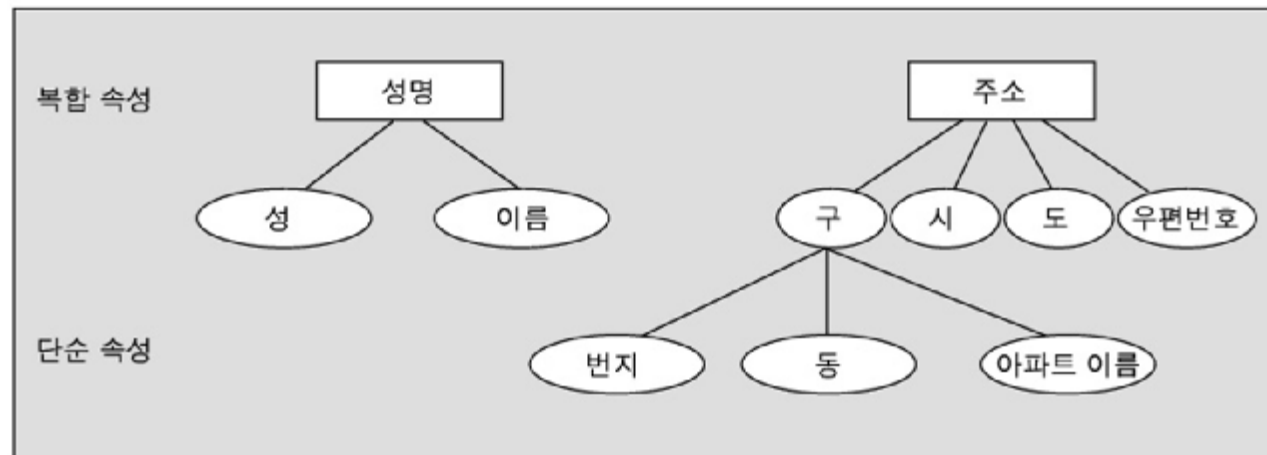
* 복합 속성(Composite Attribute)과 단순 속성(Simple Attribute)

☑ 단순 속성

더 이상 작은 구성 요소로 분해할 수 없는 속성

☑ 복합 속성

독립적인 의미를 좀더 기본적인 속성들로 분해할 수 있는 속성



복합 속성과 단순 속성

□ 데이터 모델링

● 단일값 속성(single-valued attribute)과 다중값 속성(multi-valued attribute)

* 단일값 속성(single-valued attribute)

엔터티의 속성 중 주민등록번호, 또는 학번과 같이 반드시 하나의 값만 존재하는 속성

* 다중값 속성(multi-valued attribute)

전화번호와 같이 집, 핸드폰, 회사 전화번호와 같이 여러 개의 값을 가질 수 있는 속성.

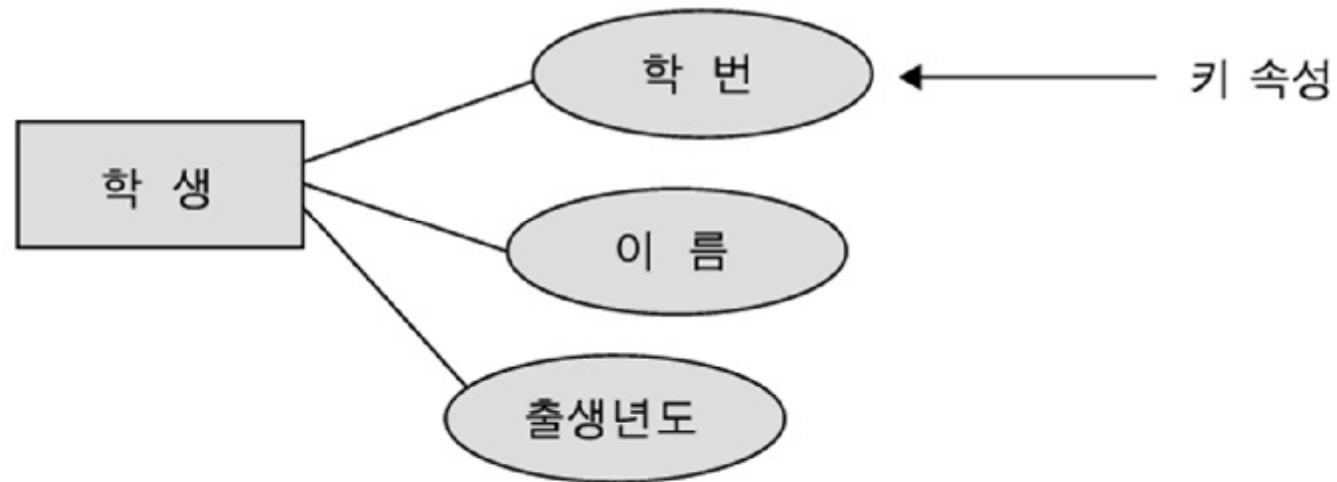
● 다중값 속성 변경 방법

1. 엔터티 집합 내에서 다중치 속성을 여러 개의 새로운 속성들로 분리.
2. 다중치 속성을 구성하는 속성들로 구성된 새로운 엔터티 집합을 생성하고,
3. 새로운 엔터티 집합과는 다대일 관계 설정.

□ 데이터 모델링

* 엔터티의 키(Key)

- ☑ 엔터티의 속성 중 하나 또는 그 이상의 속성이 엔터티를 다른 엔터티와 구별할 수 있는 속성



□ 데이터 모델링

* 유도 속성(Drived Attribute)과 저장 속성(Stored Attribute)

☑ 유도 속성

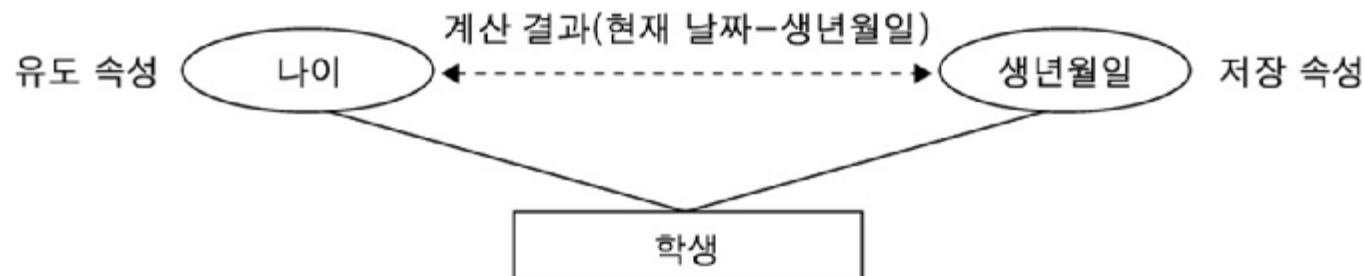
속성의 값이 다른 관련된 속성이나 엔터티가 가지고 있는 값으로부터 유도
되

어 결정되는 속성

☑ 저장 속성

유도 속성을 결정하기 위해 사용된 속성

ERD에서 유도 속성은 점선으로 표시



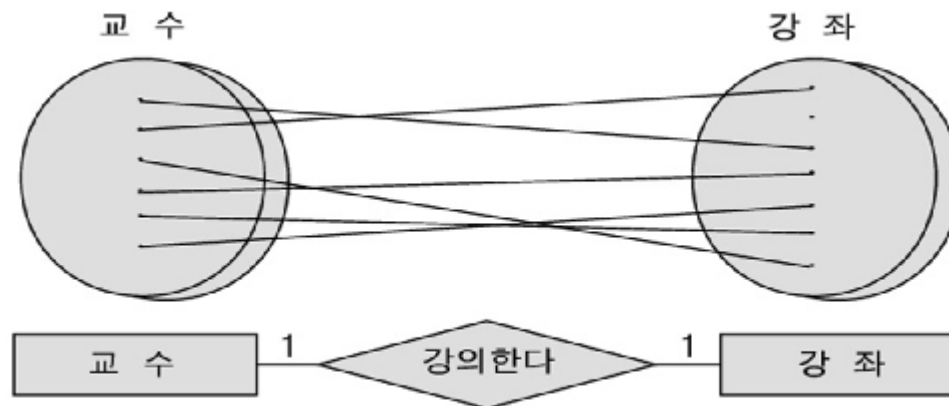
□ 데이터 모델링

● 관계(Relationship)

- ✓ 엔터티-관계 모델에서 엔터티 사이의 연관성을 표현하는 개념
- ✓ ERD에서 엔터티들 사이의 관계는 마름모를 사용하여 표현

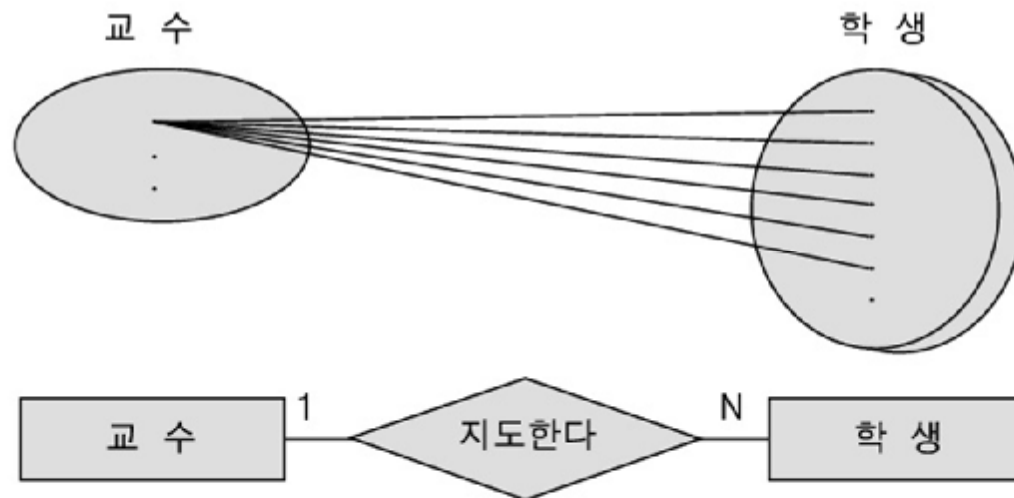


- ✓ 관계의 유형
 - ✍ 일대일 관계



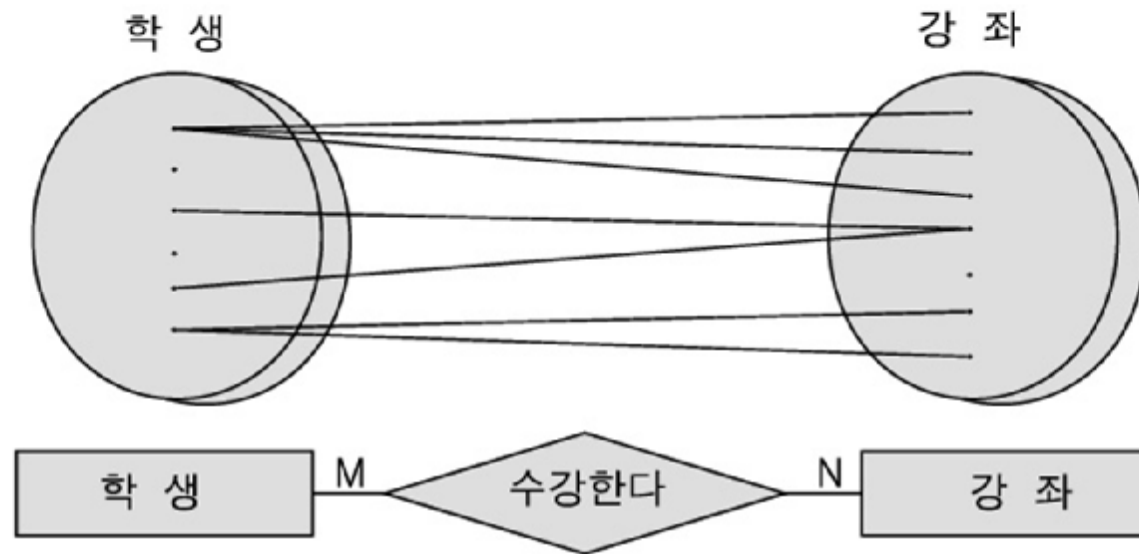
□ 데이터 모델링

✍ 일대다 관계



□ 데이터 모델링

✎ 다대다 관계



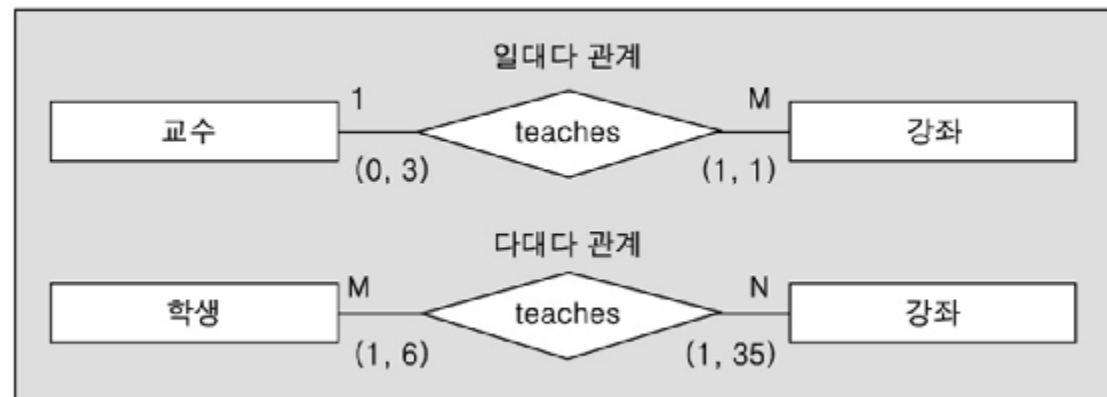
□ 데이터 모델링

● 관계의 대응 엔터티 수(Cardinality, 카디날리티)

※ 관계에 참여하는 하나의 엔터티에 대해 다른 엔터티 집합에서 몇 개의 엔터티가 참여하는지를 나타내는 것

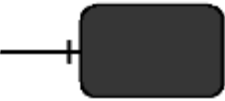
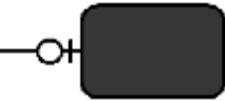



※ 예

- ☑ 한 명의 학생이 1개 이상 6개 이하의 과목에 등록할 수 있다면 대응 엔터티 수는 (1, 6)
- ☑ 대응 엔터티 수는 (min, max)의 한 쌍의 값으로 표현하는데 여기서 min은 관계에 참여하는 엔터티의 최소 개수, max는 관계에 참여하는 최대 개수 의미



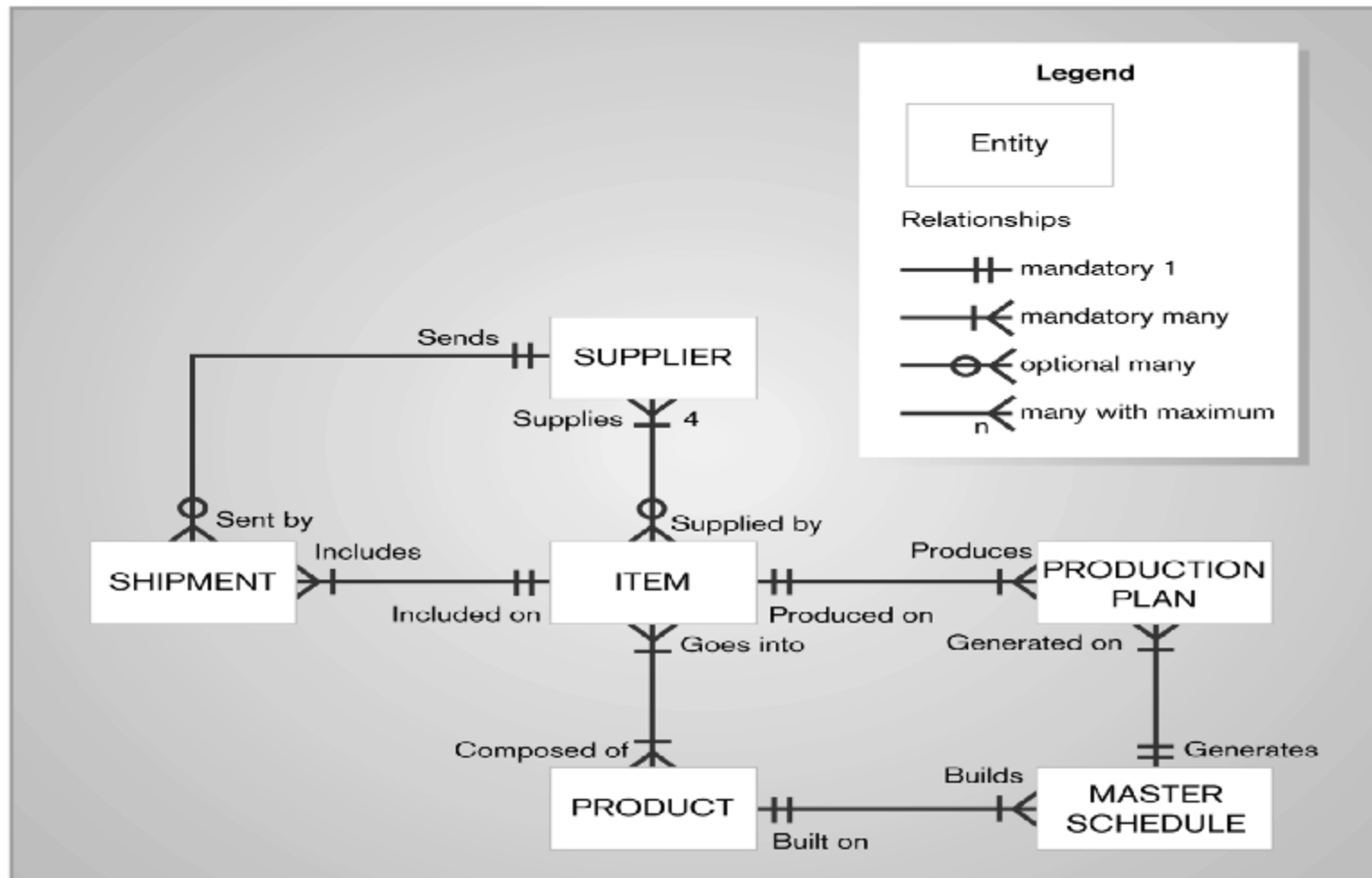
□ 데이터 모델링

● ERD의 Crow's Foot 표기 방법

Cardinality Interpretation	Minimum Instances	Maximum Instances	Graphic Notation
Exactly one	1	1	
Zero or one	0	1	
One or more	1	many (> 1)	
Zero, one, or more	0	many (> 1)	
More than one	> 1	> 1	

□ 데이터 모델링

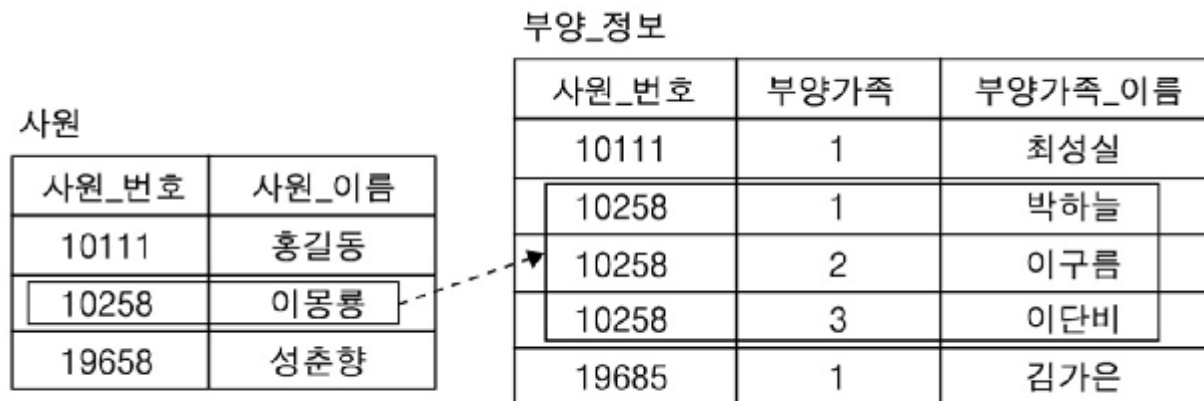
● Crows' Foot 표기법 예제



□ 데이터 모델링

● 존재 종속(existence-dependent)

- * 한 엔터티의 존재가 다른 엔터티(들)의 존재에 영향을 받는 경우 이를 존재 종속 (existence-dependent)이라 한다.



□ 데이터 모델링

● 관계 참여

* 선택적(optional) 관계

- ☑ 엔터티 집합의 엔터티는 관계를 이루는 다른 엔터티 집합에서의 엔터티와 연관이 없어도 되는 관계.
- ☑ 대응 엔터티 수(min, max)로도 표현하는 경우 min의 값이 0

* 의무적(mandatory) 관계

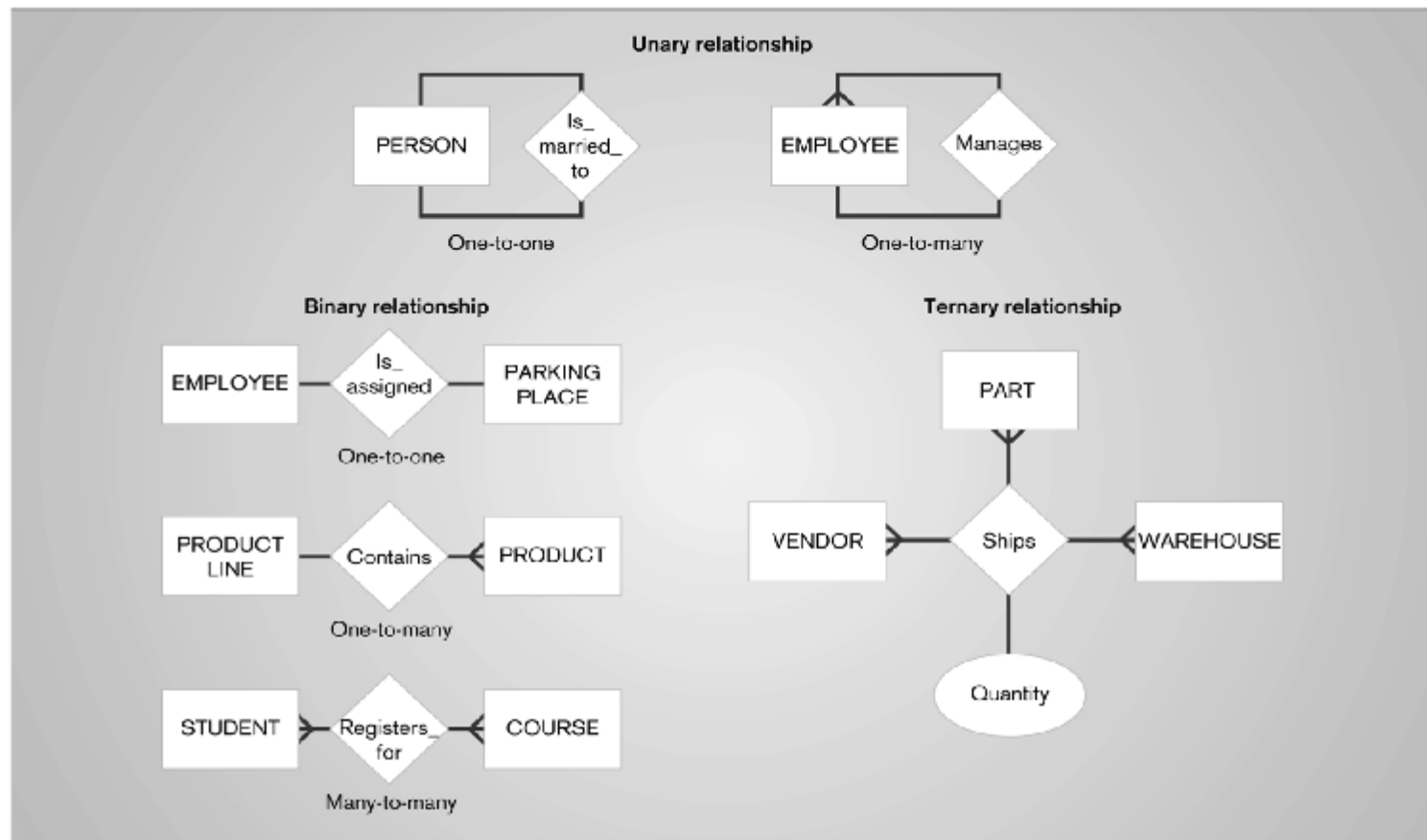
- ☑ 그 엔터티 집합의 모든 엔터티는 반드시 관계를 이루는 다른 엔터티 집합에서의 엔터티와 연관이 있어야 하는 것 의미
- ☑ 대응 엔터티 수(min, max)로도 표현하는 경우 min의 값이 1



□ 데이터 모델링

● 관계의 degree

- * 단일(unary) 관계 : 엔터티 자신과의 관계(recursive relationship)
- * 이진(binary) 관계 : 두개의 다른 엔터티 사이의 관계
- * 삼자(ternary) 관계 : 서로 다른 세개의 엔터티 사이의 관계



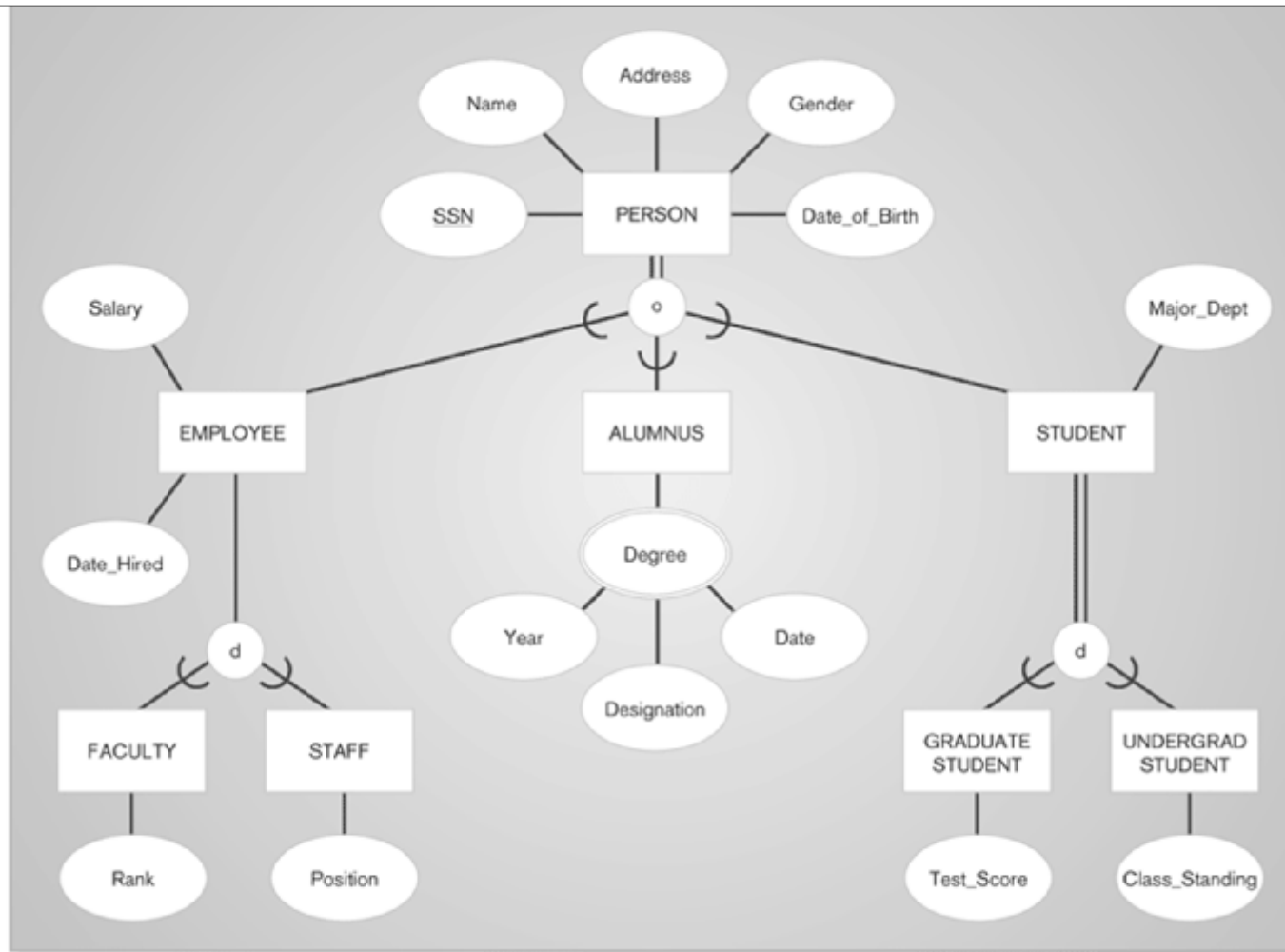
□ 데이터 모델링

● Subtype과 Supertype

- * 두개 이상의 엔터티들이 하나 이상의 속성이 관계를 공유하는 상황을 표현
- * subtype : 서로 구별되는 정보들로 세부적으로 분류될 수 있는 엔터티의 집합체
 - ☑ 학생은 학부생과 대학원생으로 분류
- * supertype : 하나 이상의 subtype을 가지는 엔터티의 집합체
 - ☑ 학부의 교수, 직원, 조교등은 교직원으로 일반화 됨
- * total specialization rule : supertype의 각 엔터티는 특정 subtype에 속해야 한다. (ERD에서 double line으로 표현)
- * partial specialization rule : supertype의 임의의 엔터티는 반드시 특정 subtype에 속할 필요는 없다. (ERD에서 single line으로 표현)
- * disjoint rule : supertype의 엔터티는 동시에 여러 subtype에 속할 수 없다.
- * overlap rule : supertype의 엔터티는 동시에 여러 subtype에 속할 수 있다.

□ 데이터 모델링

● subtype과 supertype의 예



□ 데이터 모델링

● Business rules

* 논리 데이터모델에서 일관성 표현

- ☑ entity integrity : 각 엔터티는 NULL이 아닌 유일한 ID를 가져야 한다.
- ☑ referential integrity : 엔터티 사이의 참조 관계에 대한 규칙
- ☑ domain : 각 속성값은 올바른 범위나 제약 조건을 가진다.
- ☑ triggering operation : 특정 이벤트의 발생으로 다른 작업의 실행을 촉발시킨다.

● Domain

* 속성값이 가져야 하는 자료형과 범위를 표현

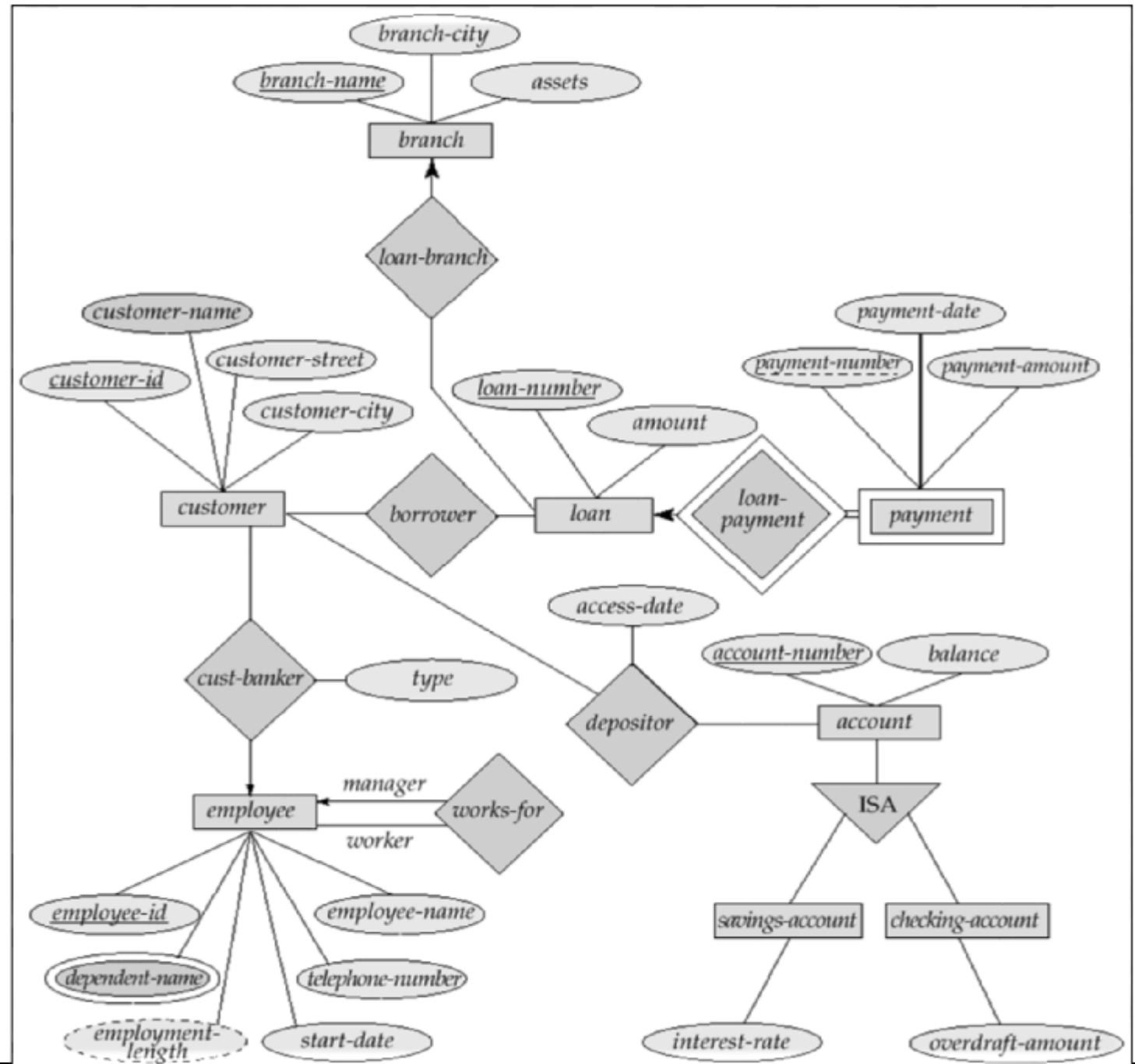
- ☑ 자료형, 저장공간 크기 또는 길이, 형식, 범위, 허용될 값들, 의미

* domain 사용의 장점

- ☑ 속성값의 일관성 보장
- ☑ 속성값의 논리적 수정 보장
- ☑ 속성값의 특성을 표현하는 체계적 수단

□ 데이터 모델링

● 은행 DB를 위한 ERD 예



□ 데이터 모델링

● 관계 데이터 구조

※ 1970년 Codd에 의해 개발

※ 관계형 데이터 구조를 구성하는 용어

- ☑ 릴레이션(Relation) : 정보 저장의 기본 형태가 2차원 구조의 테이블
- ☑ 속성(attribute) : 테이블의 각 열을 의미
- ☑ 도메인(Domain) : 속성이 가질 수 있는 값들의 집합
- ☑ 튜플(Tuple) : 테이블이 한 행을 구성하는 속성들의 집합

The diagram shows a table representing a relation. Above the table, the word '학생' (Student) is written, with dashed lines connecting it to each of the four columns. Above the first three columns, the word '속성' (Attribute) is written, with dashed lines connecting it to each of the three columns. Below each column, the word '도메인' (Domain) is written, with dashed lines connecting it to the column. To the right of the table, the word '튜플' (Tuple) is written, with dashed lines connecting it to each of the five rows.

학번	이름	학년	학과
100	홍길동	4	컴퓨터
200	이하늘	3	전기
300	임걱정	1	컴퓨터
400	송호준	4	컴퓨터
500	박현진	2	음악

□ 데이터 모델링

* 릴레이션(Relation)

☑ 정의

- ✍ 릴레이션은 행(row)과 열(column)로 구성되는 2차원 구조
- ✍ 릴레이션의 각 행은 하나의 엔터티를 나타내고, 릴레이션의 각 열은 엔터티의 각 속성 의미
- ✍ 관계형 모델에서는 행은 튜플(tuple), 열은 속성(attribute)이라는 이름 사용

☑ 특징

- 1. 하나의 릴레이션에 있는 튜플들은 모두 상이(distinct)해야 한다.
- 2. 하나의 릴레이션에서 튜플들의 순서와 속성들의 순서는 아무런 의미가 없다.
- 3. 하나의 릴레이션에서 같은 이름을 가진 속성들이 있을 수 없다
- 4. 각 속성이 가질 수 있는 값들의 범위(도메인,domain)를 벗어나는 값을 가진 튜플들이 존재할 수 없다.
- 5. 행과 열이 교차되는 곳은 원자값(atomic value)으로만 표현된다. 원자값은 논리적으로 더 이상 쪼개질 수 없는 값으로서, 여러 개의 값을 갖는 속성을 직접 표현하는 것이 불가능함을 의미한다.
 - 수치형 (Numeric Type), 문자형 (Character Type) , 날짜형 (Date Type), 논리형 (Logical Type)

□ 데이터 모델링

☑ 키 (key)

- ✍ 릴레이션을 구성하는 각 튜플들을 데이터 값들에 의해 유일하게 식별할 수 있는 속성
- ✍ 후보키(Candidate Key)
 - 릴레이션의 한 속성 집합(K)이 릴레이션이 전체 속성 집합 A의 부분 집합이면서 유일성(uniqueness)과 최소성(minimality)을 만족하는 경우 속성 집합(K)
 - 학번 속성
 - {이름, 학과} 속성

학생

학번	이름	학년	학과
100	홍길동	4	컴퓨터
200	이하늘	3	전기
300	임꺽정	1	컴퓨터
400	송호준	4	컴퓨터
500	박현진	2	음악

□ 데이터 모델링

✎ 기본 키(Pimary Key)

- 후보 키 중 하나의 속성만으로 릴레이션의 튜플들을 유일하게 식별할 수 있는 키

✎ 대체키(Alternate Key)

- 기본 키를 제외한 나머지 후보키(Candidate Key)

✎ 외래키(Foreign Key)

- 하나 이상의 테이블을 연결하여 사용하는 경우 필요한 키
- 한 테이블의 속성들의 집합으로 그 값이 다른 테이블의 주키와 일치하거나 null 값인 키를 의미

□ 데이터 모델링

☑ 릴레이션의 연결

공통된 속성을 공유하여 관계형 데이터베이스 내의 릴레이션들을 연결



□ 데이터 모델링

* 도메인(Domain)과 속성(Attribute)

☑ 속성 값(Attribute Value)

- ✍ 개개의 속성이 가지는 데이터 값
- ✍ 관계형 모델에서 이러한 데이터 값들은 더 이상 분해할 수 없는 원자 값만을 허용

☑ 도메인 (Domain)

- ✍ 하나의 속성이 가질 수 있는 같은 타입의 모든 원자 값의
- ✍ 각 속성은 하나의 도메인에 대해서만 값을 사용할 수 있다.

□ 데이터 모델링

● 제약 조건

* 엔터티 무결성(Entity Integrity)

- ☑ 릴레이션의 엔터티들을 식별할 기본 키로 사용되는 속성이 튜플들을 유일하게 식별할 수 있도록 널 값(null)을 가질 수 없는 성질

* 참조 무결성(Referential Integrity)

- ☑ 릴레이션은 참조할 수 없는 외래키의 값을 가져서는 안 된다는 것을 의미.
- ☑ 참조할 수 없는 외래키 값이란 널이 아니면서 참조된 릴레이션의 어떤 기본키의 값과도 일치하지 않는 값 의미.

□ 데이터 모델링

● 관계 데이터 연산

※ 데이터의 구조와 제약을 정의하는 것 뿐만 아니라 데이터를 조작하기 위한 연산

※ 종류

- ☑ 관계 대수(relational algebra)
- ☑ 관계 해석(relational calculus)

※ 관계 대수(relation algebra)

- ☑ 관계 대수는 E. F. Codd가 관계 데이터 모델을 처음 제안할 때 정의
- ☑ 이후 여러 형태로 확장되어 왔다.
- ☑ 관계 대수는 연산자를 사용하여 주어진 릴레이션들로부터 필요한 새로운 릴레이션을 생성한다.
- ☑ 관계 대수는 질의에 대한 결과를 생성하기 위해 수행해야 할 연산의 순서를 명시하는 절차적 언어다.
- ☑ 연산자의 종류
 - ✍ 일반 집합 연산(set operations)
 - 합집합(UNION), 교집합(INTERSECTION), 차집합(DIFFERENCE), 카티션 프로덕트(CARTESIAN PRODUCT)
 - ✍ 순수 관계 연산
 - 셀렉트(SELECT), 프로젝트(PROJECT), 조인(JOIN), 디비전(DIVISION)

□ 데이터 모델링

☑ 카티션 프로덕트(CARTESIAN PRODUCT, X)

✎ 두 릴레이션의 조합 가능한 모든 릴레이션

A		B		(A TIMES B)	
A#		B#		A#	B#
A1		B1		A1	B1
A2		B4		A1	B4
A3		B5		A1	B5
				A2	B1
				A2	B4
				A#	B#
				A2	B5
				A3	B1
				A3	B4
				A3	B5

□ 데이터 모델링

☑ 선택(SELECT, σ)

- ✍ 릴레이션으로부터 조건에 만족된 튜플들을 선택하는 연산자
- ✍ 선택 연산 결과 구성되는 릴레이션의 수평 부분 집합(horizontal subset)으로 데이터베이스 조작용의 조건절에 지정된 조건식(predicate)을 만족하는 릴레이션 내의 튜플의 집합이 된다.

✍ $\sigma_{\langle \text{선택조건} \rangle}$ (테이블이름)

학생

학번	이름	학번	학과	점수
100	홍길동	4	컴퓨터	80
200	이하늘	3	전기	90
300	임꺽정	1	컴퓨터	85
400	송호준	4	컴퓨터	70
500	박현진	2	음악	79

$\sigma_{\text{점수} \geq 80}$ (학생)

학번	이름	학번	학과	점수
100	홍길동	4	컴퓨터	80
200	이하늘	3	전기	90
300	임꺽정	1	컴퓨터	85

□ 데이터 모델링

☑ 프로젝트(PROJECT, π)

- ✎ 프로젝트 연산은 릴레이션의 특정 속성만으로 구성된 새로운 릴레이션을 구하기 위한 연산
- ✎ 결과 릴레이션은 릴레이션을 수직으로 절단한 열(column)의 집합
- ✎ 프로젝트는 릴레이션의 수직적 부분 집합(vertical subset)
- ✎ 프로젝트 연산 결과로 만들어진 릴레이션에 똑같은 튜플이 중복되어 존재하는 경우 시스템은 그 중 하나만 제외하고 나머지는 모두 삭제

✎ π 속성리스트 (테이블 이름)

학번	이름	학년	학과	점수
100	홍길동	4	컴퓨터	80
200	이하늘	3	전기	90
300	임찍정	1	컴퓨터	85
400	송호준	4	컴퓨터	70
500	박현진	2	음악	79

π 이름, 학과 (학생)

이름	학과
홍길동	컴퓨터
이하늘	전기
임찍정	컴퓨터
송호준	컴퓨터
박현진	음악

□ 데이터 모델링

☑ 조인(JOIN, ⋈)

- ✍ 두 릴레이션과 관련된 튜플을 하나의 튜플로 결합하는 연산
- ✍ 카티션 프로덕트(cartisian product) 연산의 결과에서 얻어진 릴레이션으로부터 조건에 맞는 튜플의 집합을 구하기 위한 연산
- ✍ 조인 연산은 연산자를 θ 로 표현하여 일반화하므로 θ 로 표현될 수 있는 조인을 세타 조인(θ -join)이라 한다.
- ✍ θ 가 "="인 조인 : 동일 조인 또는 이취 조인(equijoin)

학생

학번	이름	학점	학과
100	홍길동	4	컴퓨터
200	이하늘	3	전기
300	임꺽정	1	컴퓨터

성적

학번	과목번호	등급
100	C123	B
100	C234	A
200	C236	B
300	C156	A
300	C234	A

학생 ▷◁ 학번-학점 성적

학생 · 학번	학생 · 이름	학생 · 학점	학생 · 학과	수강과목 · 과목 번호	수강과목 · 등급
100	홍길동	4	컴퓨터	C123	B
100	홍길동	4	컴퓨터	C234	A
200	이하늘	3	전기	C236	B
300	임꺽정	1	컴퓨터	C156	A
300	임꺽정	1	컴퓨터	C234	A

□ 데이터 모델링

✎ θ 가 “N”인 조인 : 자연 조인(Natural Join)

학생 \bowtie_N 등록

학생 · 학번	학생 · 이름	학생 · 학년	학생 · 학과	수강과목 · 과목 번호	수강과목 · 등급
100	홍길동	4	컴퓨터	C123	B
100	홍길동	4	컴퓨터	C234	A
200	이하늘	3	전기	C236	B
300	임꺽정	1	컴퓨터	C156	A
300	임꺽정	1	컴퓨터	C234	A

□ 데이터 모델링

☑ 디비전(DIVISION, ÷)

- ✎ 두 릴레이션에 대해 A, B에 대해 B 릴레이션의 튜플에 관련된 모든 튜플들을 A 릴레이션으로부터 구하는 것

학과목(SC)		과목_1(C1)	과목_2(C2)	과목_3(C3)
학번	과목 번호	과목 번호	과목 번호	과목 번호
100	C123	C123	C234	C236
100	C234			C156
200	C236			
200	C234			
200	C123			
300	C156			
300	C234			
300	C236			
300	C123			
400	C156			
400	C234			

(디비전의 결과)		
SC ÷ C1	SC ÷ C2	SC ÷ C3
학번	학번	학번
100	100	300
200	200	
300	400	

□ 데이터 모델링

* 관계 해석(Relational Calculus)

☑ 특징

- ✎ 주어진 릴레이션에 의해 요구되는 릴레이션의 정의를 형식화하기 위한 표기법 제공
- ✎ 이를 사용하여 원하는 정보만을 선언하는 비절차적인 특징을 가진다.

☑ 종류

✎ 튜플 관계 해석(Tuple Relational Calculus)

원하는 릴레이션에 대한 튜플 관계 해석식은 다음과 같이 정의

$$\{ t_1.A_1, t_2.A_2, \dots, t_n.A_n \mid F(t_1, t_2, \dots, t_{n+1}, t_{n+2}, \dots, t_{n+m}) \}$$

✎ 도메인 관계 해석(Domain Relational Calculus)

사용자가 원하는 정보를 도메인 해석식으로 표현

튜플 해석식에서의 튜플 변수 대신 도메인 변수를 사용하는 것을 제외하

$$\{ d_STD_NO \mid \text{수강}(d_STD_NO, d_GRADE) \wedge d_CNO='G413' \wedge d_GRADE='A' \}$$

□ 데이터 모델링

● 함수적 종속성 (Functional Dependency: FD)

* 의미

- ☑ 속성들 사이의 관계(relationship)에 대한 제약 조건
- ☑ 속성 X의 값이 속성 Y의 값을 결정지으면, "속성 Y는 속성 X에 함수적으로 종속된다"고 함

* 표기

학생

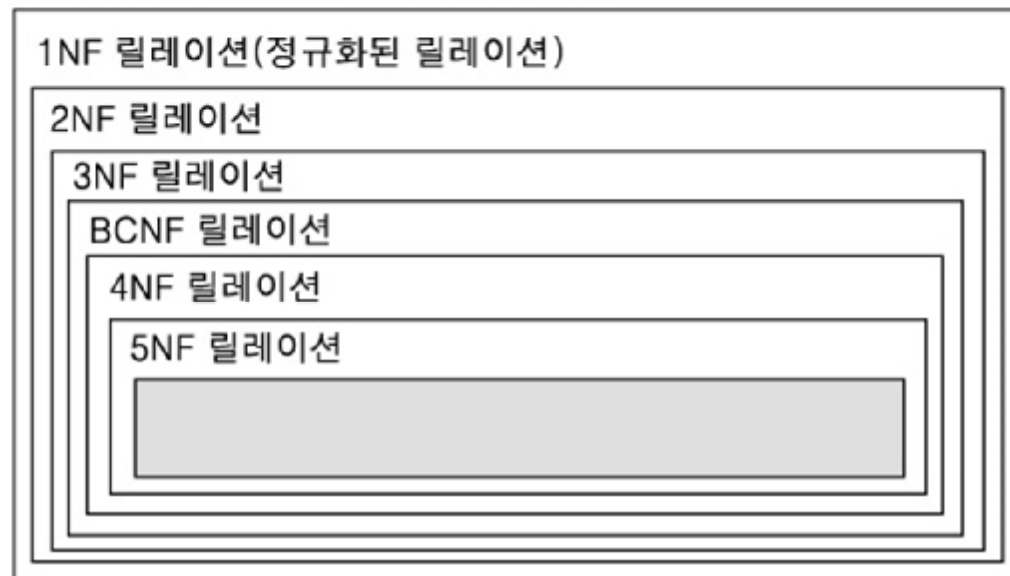
학번	이름	학년	학과
100	홍길동	4	컴퓨터
101	이하늘	3	전기
102	임꺽정	1	컴퓨터
103	이하늘	2	음악

□ 데이터 모델링

● 정규화 형태

- * 릴레이션의 정규화는 실제 데이터 값이 아니라 개념적인 측면에서 다루어져야 함
- * 실제 정규화 과정은 정규형의 순서와 다를 수 있음

전체 릴레이션(정규화 또는 비정규화된)

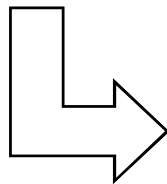


□ 데이터 모델링

* 제1정규형(1NF)

- ☑ 릴레이션에서 속성의 값은 속성의 도메인에 속하는 단일 값(원자 값, atomic Value)이어야 한다는 제약을 의미
- ☑ 다차값을 가지는 비정규형 릴레이션을 제1정규형으로 변환하는 방법
 - ✍ 1. 각 다치값을 위해서 하나의 레코드를 생성하여 새로운 릴레이션 생성

학번	이름	학과	동아리
20013426	박하늘	컴퓨터학과	{영어회화반, 검도부}
20025914	홍길동	영문학과	{수화반, 합창반}
20038540	홍길순	음악학과	미술반
99590264	이몽룡	사회복지학과	검도부
97456123	최푸름	국어국문학과	축구부

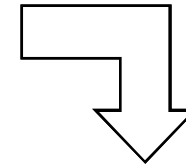


학번	이름	학과	동아리
20013426	박하늘	컴퓨터학과	영어회화반
20013426	박하늘	컴퓨터학과	검도부
20025914	홍길동	영문학과	수화반
20025914	홍길동	영문학과	합창반
20038540	홍길순	음악학과	미술반
99590264	이몽룡	사회복지학과	검도부
97456123	최푸름	국어국문학과	축구부

□ 데이터 모델링

✎ 2. 키가 되는 속성(학번)을 이용하여 두 개의 테이블로 분리

학번	이름	학과	동아리
20013426	박하늘	컴퓨터학과	{영어회화반, 검도부}
20025914	홍길동	영문학과	{수화반, 합창반}
20038540	홍길순	음악학과	미술반
99590264	이몽룡	사회복지학과	검도부
97456123	최푸름	국어국문과	축구부



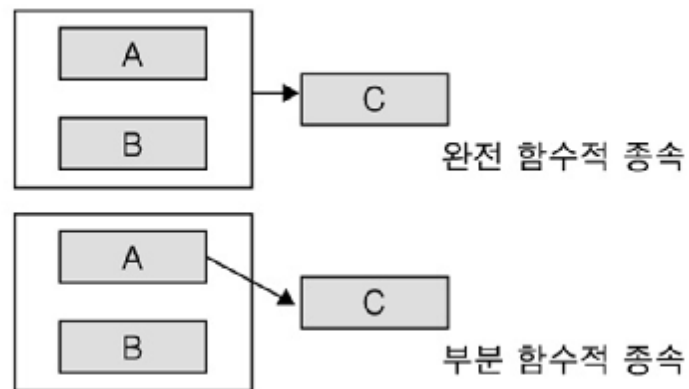
학번	이름	학과
20013426	박하늘	컴퓨터학과
20025914	홍길동	영문학과
20038540	홍길순	음악학과
99590264	이몽룡	사회복지학과
97456123	최푸름	국어국문과

학번	동아리
20013426	영어회화반
20013426	검도부
20025914	수화반
20025914	합창반
20038540	미술반
99590264	검도부
97456123	축구부

□ 데이터 모델링

* 제 2정규형(2NF)

- ☑ 완전 함수적 종속성(full functional dependency)의 개념에 기반을 둔 것으로, 릴레이션 R이 1NF이고 키가 아닌 모든 속성이 기본키에 완전 함수적 종속이면 릴레이션 R은 2NF임.
- ☑ 완전 함수적 종속성
 - ✍ 두 속성 A와 B 사이에 $A \rightarrow B$ 의 함수적 종속성이 존재할 때, B가 A의 부분집합 A'에 함수적으로 종속되지 않는 것 의미
 - ✍ 즉, $A' \rightarrow B$ 가 성립되지 않아야 하며, 만약 $A' \rightarrow B$ 가 성립하면 부분종속(partial dependency)이라고 부른다.



□ 데이터 모델링

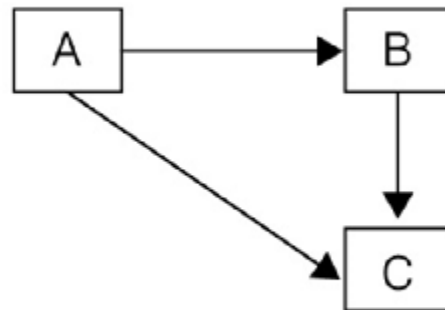
* 제 3정규형(3NF)

☑ 특징

- ✍ 이행 종속성(transitive dependency)의 개념에 기반을 둔 정규형
- ✍ 키가 아닌 속성 값을 갱신하는 경우 불필요한 부작용(이상) 발생 없음
- ✍ 모든 이진 릴레이션(2NF)는 3NF에 속함

☑ 이행 종속성

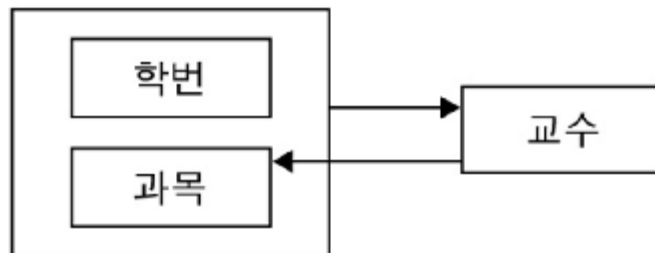
- ✍ 세 속성 사이에 존재하는 함수적 종속성
- ✍ A, B, C가 한 테이블 내의 세 속성이고 $A \rightarrow B$ 와 $B \rightarrow C$ 의 함수적 종속성이 존재하면, 함수적 종속성 $A \rightarrow C$ 가 성립하는 것 의미



□ 데이터 모델링

* BCNF

- ☑ 기본 키가 둘 이상의 속성으로 구성된 합성키이고, 합성키가 둘 이상 존재할 경우에 발생하는 이상현상을 방지하기 위해서 고안
- ☑ 모든 결정자가 후보키일 경우 릴레이션은 BCNF형
- ☑ 릴레이션 R이 BCNF에 속하면 R은 제1, 제2, 제3 정규형에 속함
- ☑ 강한 제3정규형(strong 3NF)
- ☑ 예(3NF) : 과목 릴레이션
 - ✍ 과목 (학번,과목,교수)
 - ✍ 후보키 : (학번,과목), (학번,교수)
 - ✍ 기본키 : (학번,과목)
 - ✍ 함수종속 : (학번,과목) → 교수 , 교수 → 과목



과목

학번	과목	교수
100	프로그래밍	P1
100	자료구조	P2
200	프로그래밍	P1
200	자료구조	P3
300	자료구조	P3
300	프로그래밍	P4

□ 데이터 모델링

☑ 3NF(수강과목 릴레이션)에서의 이상

삽입이상

교수 P5가 자료구조를 담당한다는 사실의 삽입은 수강 학생이 있어야 가능

삭제이상

100번 학생이 자료구조를 취소하여 튜플을 삭제하면 P2가 담당교수라는 정보도 삭제됨

갱신이상

P1이 프로그래밍 대신 자료구조를 담당하게 되면 P1이 나타난 모든 튜플을 변경하여야 함

⇒ 원인 : 교수가 결정자이나 후보키가 아님

⇒ 해결 : 과목 ⇒ 교수, 지도과목 릴레이션으로 분해

수강	학번	교수
	100	P1
	100	P2
	200	P1
	200	P3
	300	P3
	300	P4

지도과목	교수	과목
	P1	프로그래밍
	P2	자료구조
	P3	자료구조
	P4	프로그래밍

□ 데이터 모델링

* 제 4정규형

☑ 의미

- ✍ 릴레이션 R에서 MVD $A \twoheadrightarrow B$ 가 존재할 때 R의 모든 속성들이 A에 함수적 종속(FD)이면 R은 4NF (즉, R의 모든 애트리뷰트 X에 대해 $A \rightarrow X$ 이고 A가 후보키)
- ✍ 릴레이션 R이 BCNF에 속하고 모든 MVD(다치종속성)가 FD이면 R은 4NF
- ✍ 릴레이션 R이 4NF이라면 MVD가 없거나, MDV $A \twoheadrightarrow B|C$ 가 있을 경우 A에 대응되는 B와 C의 값은 하나씩 이어야 하며 이때 A는 후보키라는것 의미

□ 데이터 모델링

학생등록

학생	수강과목	동아리
홍길동	심리학	타임반
홍길동	심리학	영화감상반
홍길동	컴퓨터	타임반
홍길동	컴퓨터	영화감상반
홍길동	교양영어	타임반
홍길동	교양영어	영화감상반
이몽룡	교양영어	영어회화반
이몽룡	교양영어	유도반
이몽룡	프로그래밍 기초	영어회화반
이몽룡	프로그래밍 기초	유도반

← BXNΦ

5(키에 속하지 않는 결정자 속성이 없음)

기본키: (학생, 수강과목, 동아리)

MϕΔ 학생 —————> ϕ수|교재



수강등록

학생	수강과목
홍길동	심리학
홍길동	컴퓨터
홍길동	교양영어
이몽룡	교양영어
이몽룡	프로그램의기초

동아리등록

학생	동아리
홍길동	타임반
홍길동	영화감상반
이몽룡	타임반
이몽룡	영화감상반