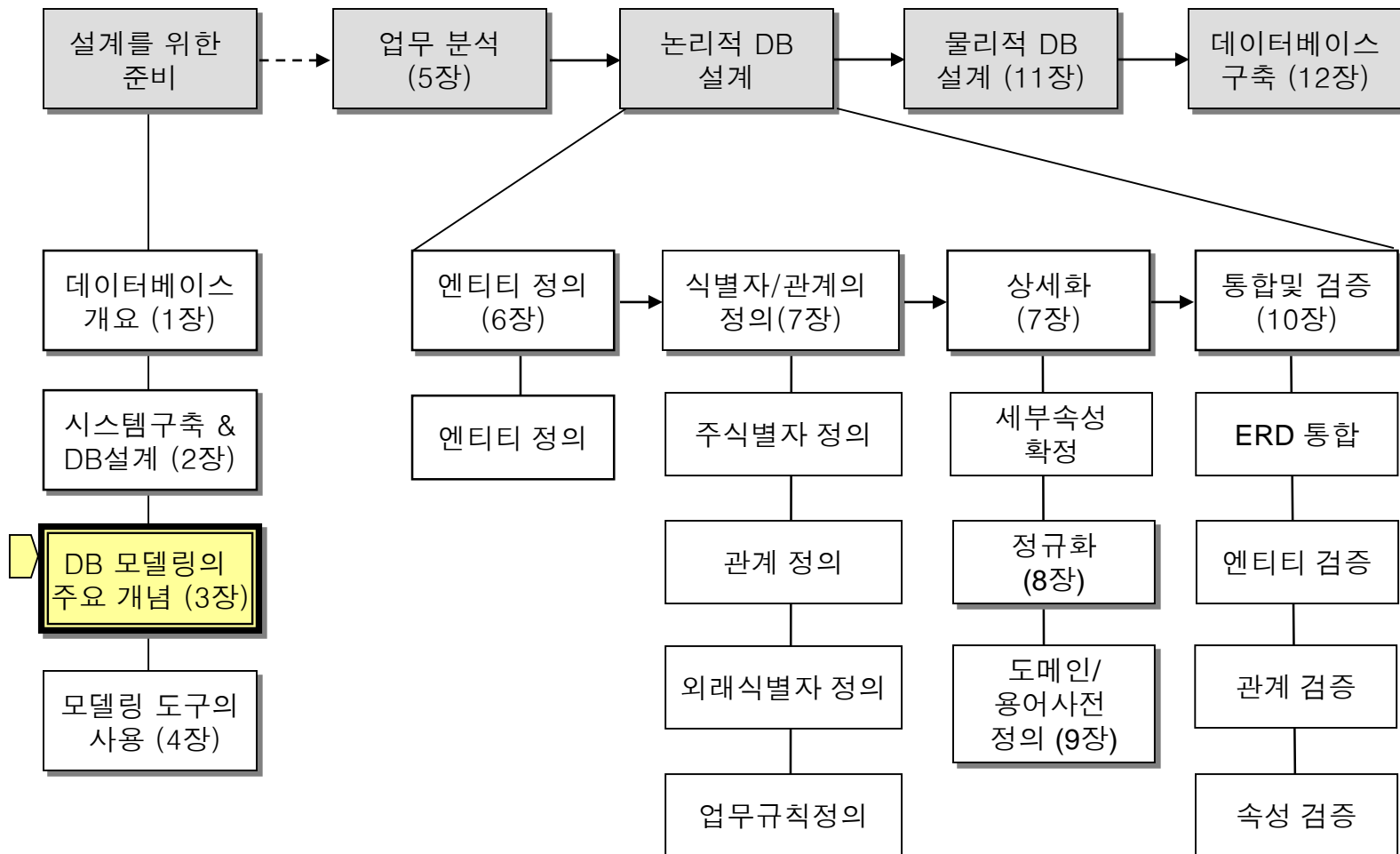




3장. 데이터모델링의 주요 개념

- ☐ 개요
- ☐ 엔티티
- ☐ 속성
- ☐ 관계
- ☐ 주식별자와 외래 식별자
- ☐ ERD 표기법



3.1 개요

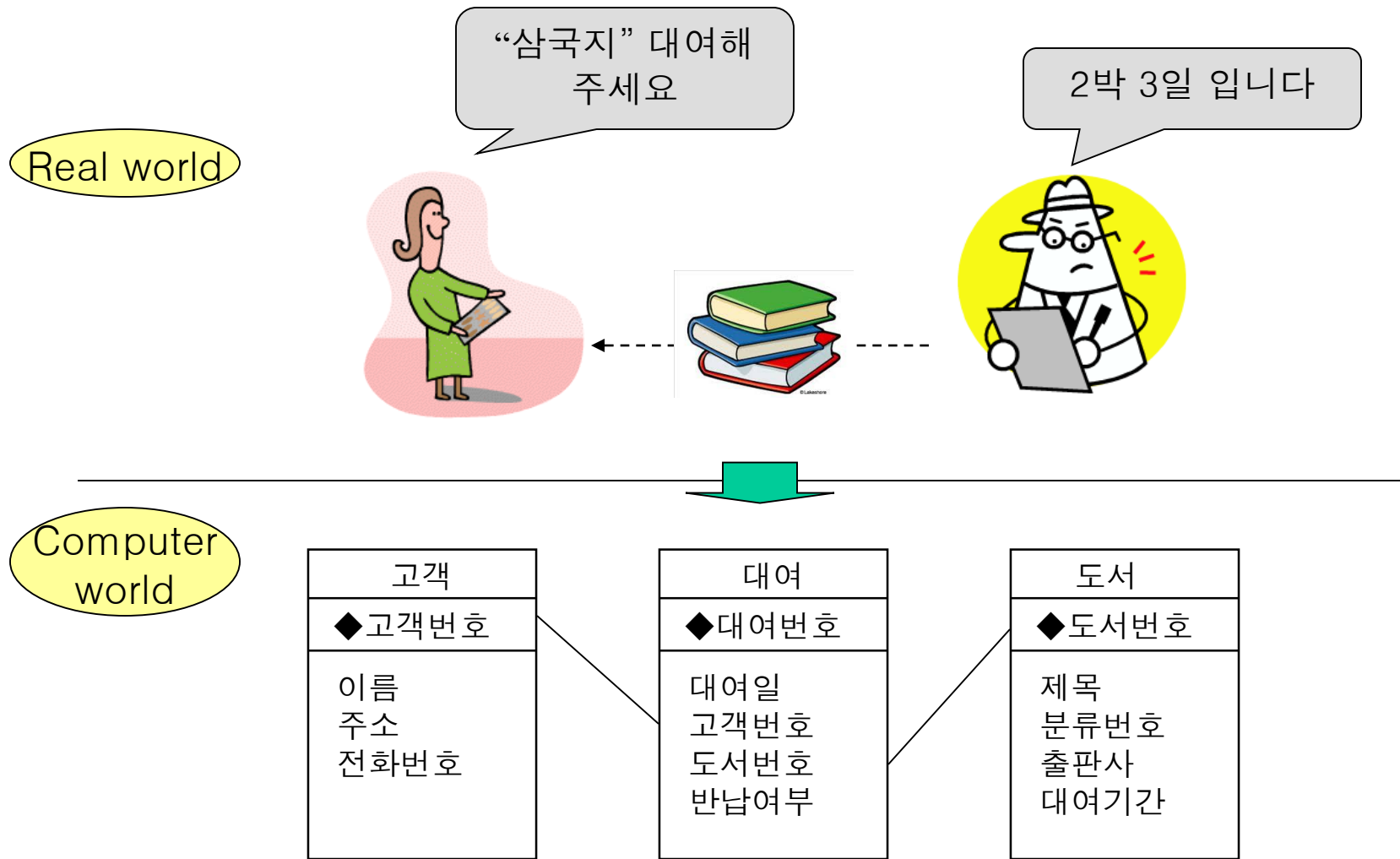
□ 데이터 모델링의 목적

- 정보 시스템을 구축하는데 필요한 정보(데이터)를 약속된 표기법에 의해 표현함으로써 시스템 구축 대상이 되는 업무 내용을 정확하게 분석하고자 함
- 분석된 모델을 가지고 실제 데이터베이스를 생성하여 개발 및 관리에 이용하고자 함



<그림 3.1> 데이터 모델링의 개념

3.1 개요



3.1 개요

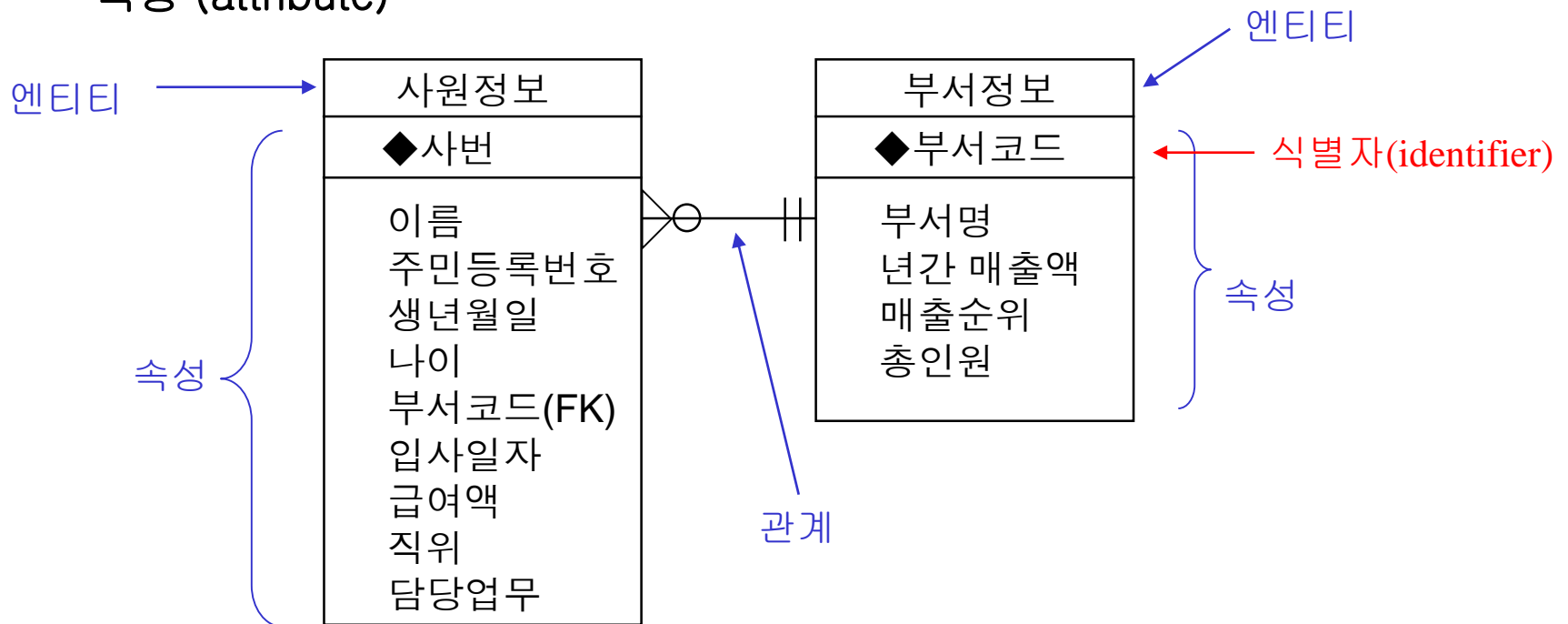
□ Note

- 논리적 데이터베이스 설계를 다른 말로 데이터 모델링 이라 한다.
 - ‘데이터 모델링’이 일반적으로 많이 쓰임
- 데이터 모델링은 전체 데이터베이스 설계에서 핵심적인 부분
- 데이터 모델링을 할 수 있기 위해서는 모델링에 사용되는 주요 개념들에 대해 알고 있어야 한다
- 데이터 모델링은 종이와 연필을 가지고도 진행할 수 있지만 효율적인 모델링을 위해 지원 도구를 사용하는 경우가 많다.
- 데이터 모델링을 수행하게 되면 최종적인 산출물(output)은 ERD 이다

3.1 개요

□ 데이터 모델링의 세가지 개념

- 엔티티 (Entity)
- 관계 (Relationship)
- 속성 (attribute)



3.1 개요

□ 데이터베이스 용어 vs 모델링 용어

데이터베이스 용어	모델링 용어
테이블(table)	엔티티(entity)
컬럼(column), 열	속성(attribute)
튜플(tuple), 행(row)	인스턴스(instance)
기본키(primary key)	주식별자(primary identifier)
외래키(foreign key)	외래 식별자(foreign identifier)

<표 3.1> 데이터베이스 용어 vs 모델링 용어

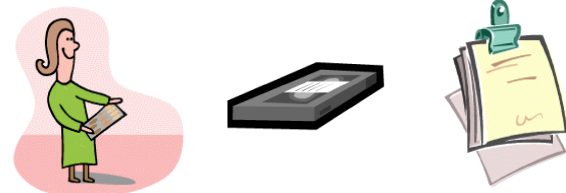
3.2 엔티티(Entity)

□ 엔티티란?

Entity란 업무의 관심 대상이 되는 정보를 갖고 있거나 그에 대한 정보를 알아야하는 유형, 무형의 사물이나 객체를 말한다.

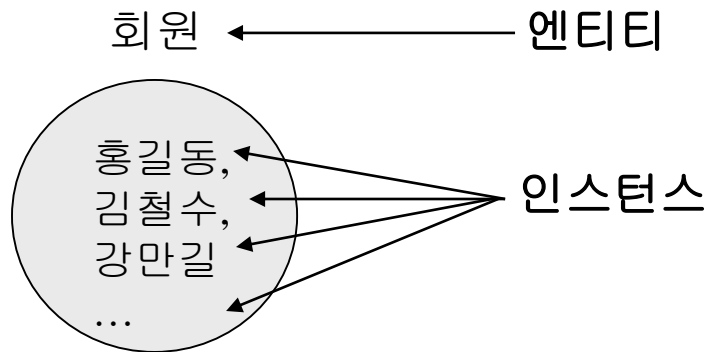
– 엔티티의 예

- 고객, 사원정보, 부서, 제품
- 주문서, 성적표, 입고전표, 금전출납부
- 생산계획, 공정



3.2 엔티티(Entity)

□ 엔티티의 표현



<그림 3.3> 엔티티와 인스턴스



<그림 3.4> ERD에서 엔티티의 표현

3.2 엔티티(Entity)

□ 엔티티의 분류

- 유형 엔티티 : 물리적인 형태가 있고 쉽게 엔티티임을 알 수 있다. (예: 고객, 사원, 상품, 거래처, 학생, 교수, ...)
- 무형 엔티티 : 물리적인 형태가 없고 개념적으로 존재하는 엔티티이다. (예: 생산계획, 부서조직, 색상별선호도, ...)
- 문서 엔티티 : 업무 절차상에서 사용되는 문서나 장부, 전표에 대한 엔티티이다. (예: 거래명세서, 입출금전표, 주문서, 금전출납부, ..)
- 이력 엔티티 : 업무상 반복적으로 이루어지는 행위나 사건의 내용을 일자별, 시간별로 저장하기 위한 엔티티이다. (예: 입고이력, 출고이력, ..)
- 코드 엔티티 : 무형 엔티티의 일종으로 각종 코드를 관리하기 위한 엔티티이다. (예: 국가코드, 색상코드, 직급분류코드, 상태코드, ...)

3.2 엔티티(Entity)

□ 엔티티의 특징 (1)

- 시스템 구축 대상이 되는 업무에서 필요하고 관리하고자 하는 정보이어야 한다.
 - 예) 환자 : 병원 정보 시스템에서는 꼭 필요한 엔티티
일반 회사의 정보 시스템에서는 필요하지 않음
- 일반적으로 엔티티는 2개 이상의 **인스턴스(instance)**가 존재해야 의미가 있다.
 - 예) 과목
 - » 영어, 수학, 과학,
 - » 수학만 가르치는 보습학원에서 과목이라는 엔티티를 만드는 것이 의미가 있을까?

3.2 엔티티(Entity)

□ 엔티티의 특징 (2)

- 엔티티는 반드시 하나 이상의 속성을 가져야 한다.
 - 예) 과목
 - » 과목 코드, 과목명, 학점,...
 - 만일 속성을 찾을 수 없다면 엔티티이기 보다는 다른 엔티티의 속성일 가능성이 높다.
 - » ‘이름’ 은 엔티티이기 보다는 속성임

3.2 엔티티(Entity)

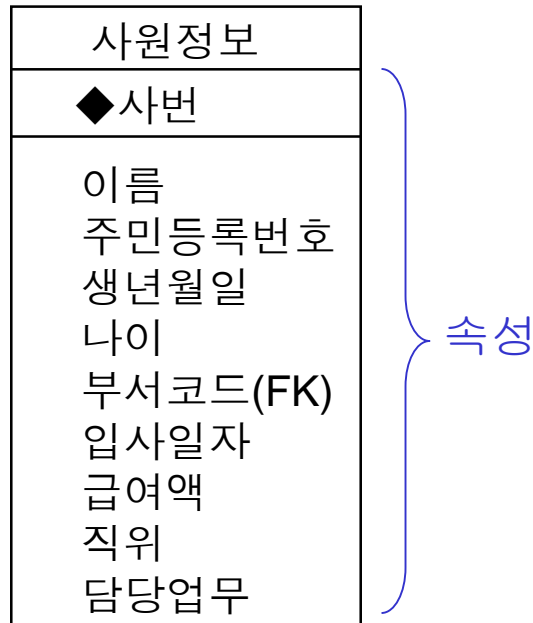
□ 엔티티의 명명(naming)

- 어떤 엔티티에 적절한 이름을 붙이는 것은 쉬운 일이 아니다.
- 예: 고객이 어떤 제품을 주문했는지를 관리하는 엔티티
 - 고객제품 : 고객이 주문한 제품? 고객의 제품 ?
- 일반적인 명명 기준
 - 현업 업무에서 사용하는 용어를 사용
 - 약어를 가능하면 사용하지 않는다
 - 단수명사를 사용
 - 모든 엔티티명은 유일해야
 - 엔티티 생성 의미대로 이름을 부여

3.3 속성(Attribute)

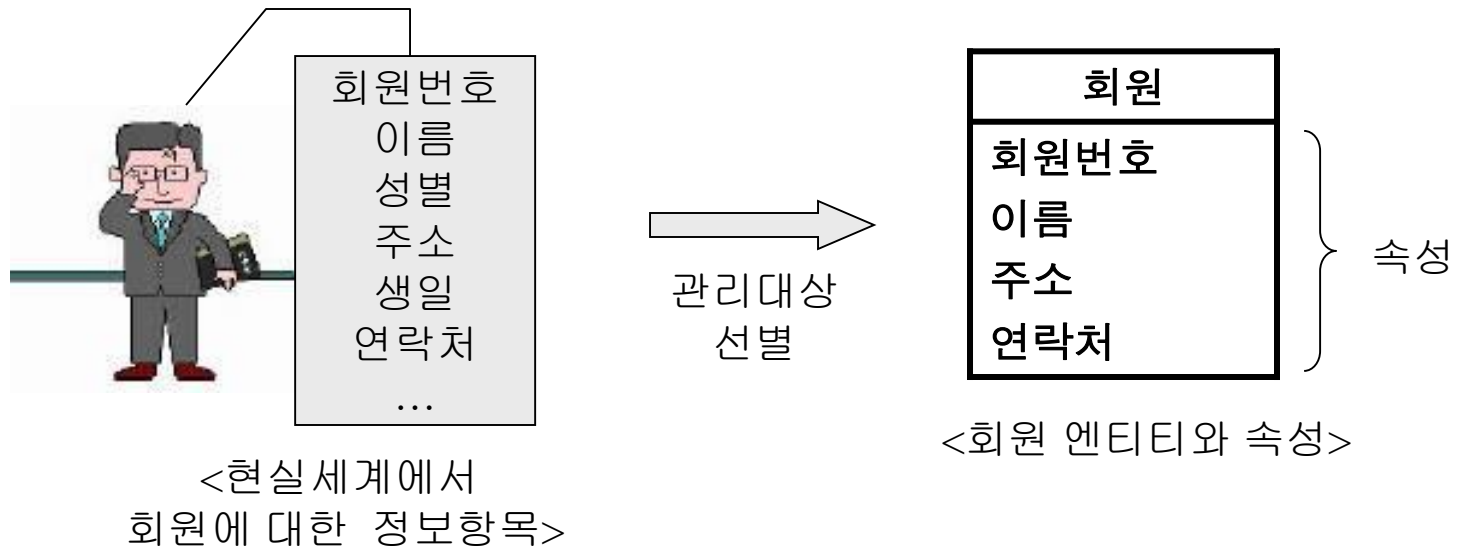
□ 속성이란

- 엔티티에서 관리하고자 하는 더 이상 분리되지 않는 최소 단위의 데이터
- 엔티티는 한 개 혹은 한 개 이상의 속성을 가진다
- 엔티티는 '속성들의 집합' 으로 정의될 수 있다.



3.3 속성(Attribute)

□ 속성이란



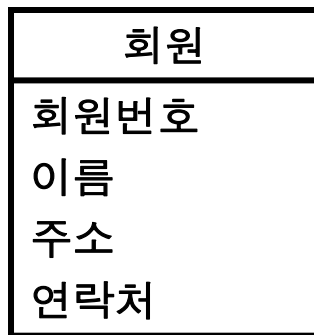
<그림 3.6> 속성의 결정

- 현실세계의 정보 항목중 업무에서 관심이 있는 정보 항목만을 속성으로 취한다

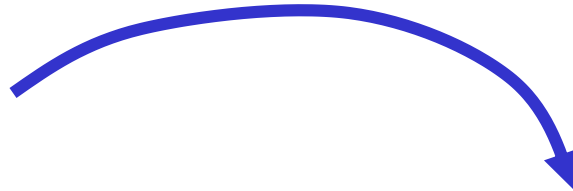
3.3 속성(Attribute)

□ 속성이란

- 엔터티의 속성은 나중에 테이블의 컬럼에 대응된다.



회원 엔티티



member

member_id	name	address	tel
1001	홍길동	서울	011-123-4561
1002	김우영	충남	019-555-1325
1003	김재일	부산	018-671-4435
1004	정태선	대전	011-801-1433

<그림 3.7> member 테이블

3.3 속성(Attribute)

□ 속성의 분류

- 기본 속성 : 업무분석을 통해 현실세계로 부터 얻어낸 속성
 - 제품이름, 제조년월, 원가
- 설계속성 : 원래 현실세계에는 존재하지 않지만 설계 과정에서 만들어진 속성
 - 국가코드, 색깔 코드, 일련번호
- 유도 속성 : 다른 속성으로 부터 계산이나 변형에 의해 나온 속성
 - 금액 (= 수량 x 단가)
 - 평균 (= 합계/인원수)

 7장에서 자세히 설명

3.3 속성(Attribute)

□ 속성의 명명 규칙

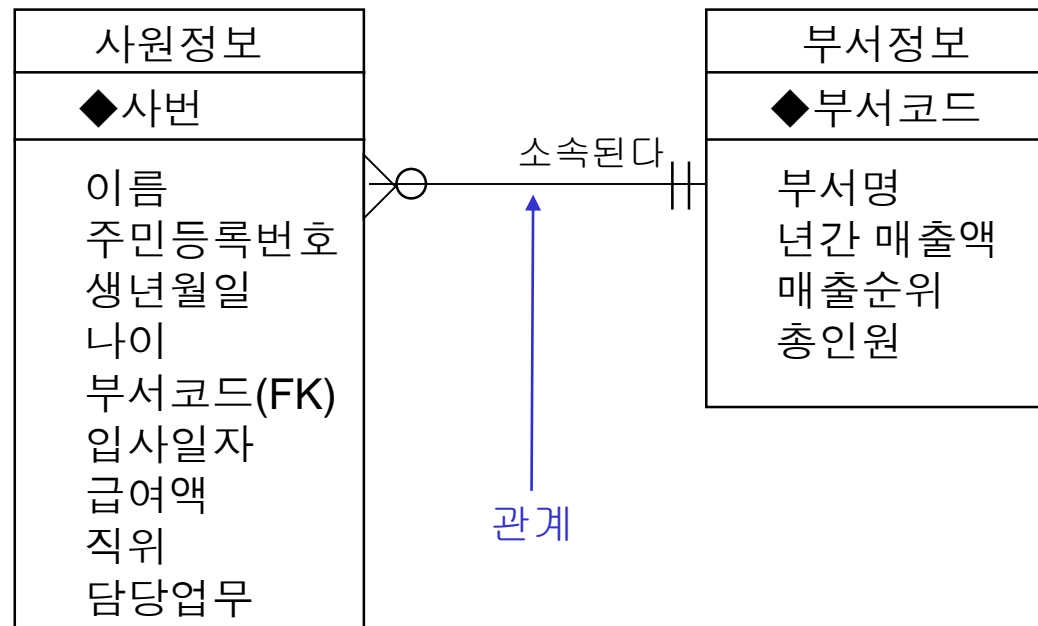
- 속성의 의미가 분명히 드러나게 이름을 부여 ***
- 해당 업무에서 사용하는 이름을 부여한다.
- 서술식 속성명은 사용하지 않는다. 수식어, 소유격 자제
- 약어의 사용은 가급적 피한다.
- 엔티티에서 유일하게 식별 가능하도록 지정
- 용어상의 혼란을 피하기 위해 사전에 용어사전(data dictionary)을 정의해 쓰는 경우도 많다.

9장에서 설명

3.4 관계(Relationship)

□ 관계란?

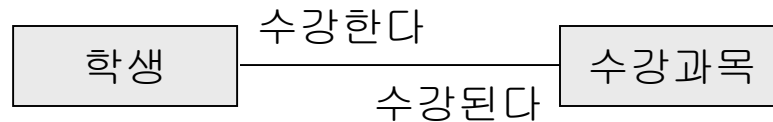
- 두개 혹은 그 이상의 엔티티들 간의 의미있는 연결
- 엔티티와 엔티티가 존재의 형태나 행위로 서로에게 영향을 주는 형태



3.4 관계(Relationship)

□ 관계란?

- ‘학생’과 ‘수강과목’의 예



<그림 3.8> 두 엔티티 사이의 관계의 표현

- 두 엔티티가 관계가 있다는 의미는 상호 공유하는 속성이 있다는 의미이다

3.4 관계(Relationship)

학생

학번	이름
21001	김철수
21002	양길현
21003	임영수
21004	박한나

수강과목

과목
전산학개론
이산수학
웹디자인

두 엔티티는
관계가 없음

<그림 3.9> 학생과 수강과목 테이블 (공유 속성 없음)

학생

학번	이름
21001	김철수
21002	양길현
21003	임영수
21004	박한나

수강과목

학번	과목
2001	전산학개론
2001	이산수학
2002	전산학개론
2003	웹디자인
2003	이산수학

두 엔티티는
관계 있음

공유하는 속성

<그림 3.10> 학생과 수강과목 테이블 (공유 속성 있음)

3.4 관계(Relationship)

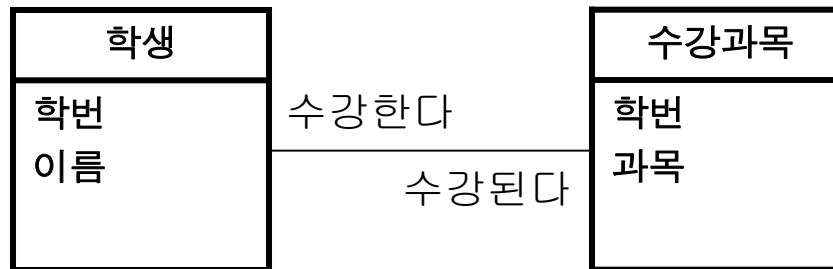
상호 관계가 있는 두 엔티티에서 공유하는 속성의 이름이 반드시 같을 필요는 없습니다



3.4 관계(Relationship)

□ 관계의 명명

- 두 엔티티 사이의 관계에 대해 이름을 붙이는 것
- 어느 엔티티의 관점에서 보느냐에 따라 이름이 다르다.

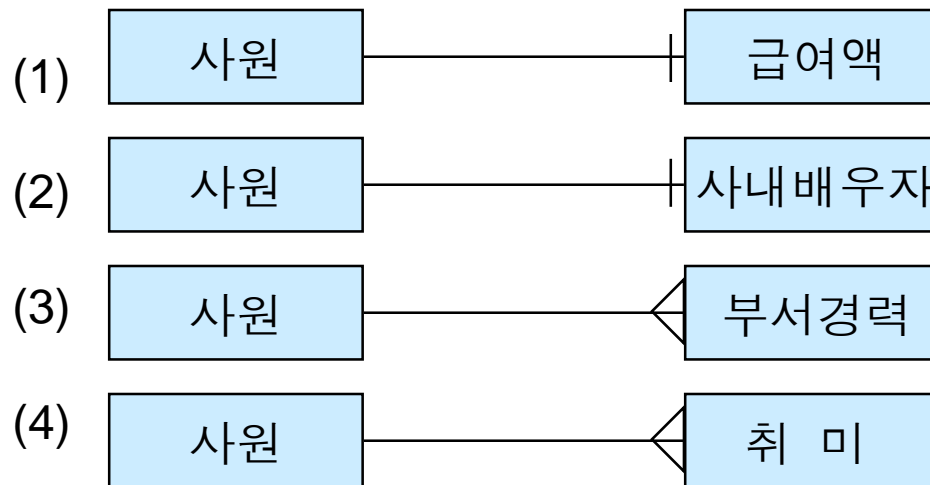


<그림 3.11> 학생과 수강과목 ERD

3.4 관계(Relationship)

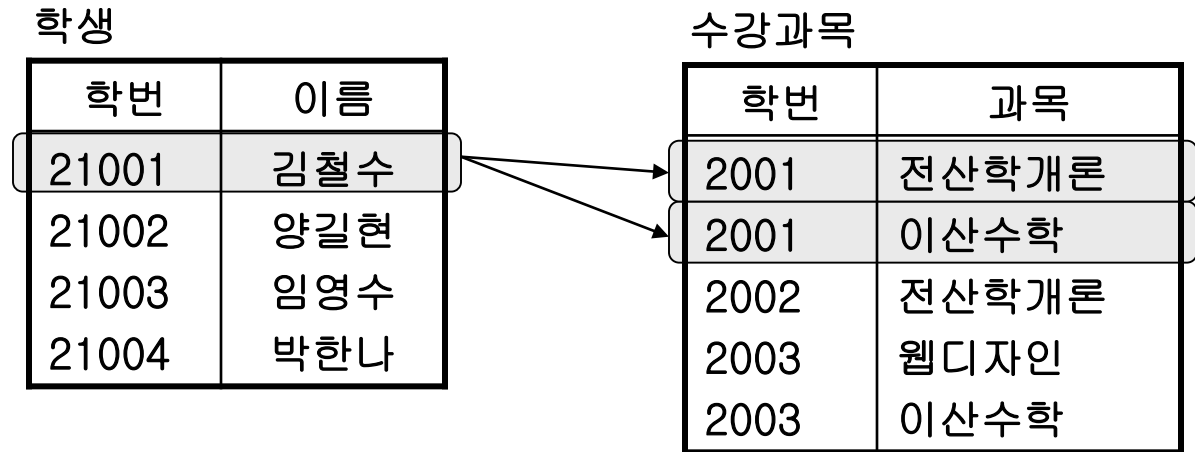
□ 관계의 카디널리티(Cardinality)

- 두개의 엔티티간의 관계에서 참여자(인스턴스)의 수를 표현한 것
 - 단일 : |
 - 다중 : <

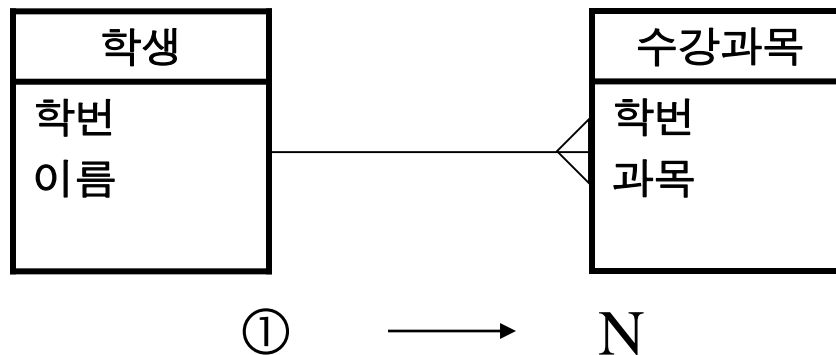


3.4 관계(Relationship)

□ 예제



<그림 3.12> 학생쪽에서 수강과목과의 카디널리티



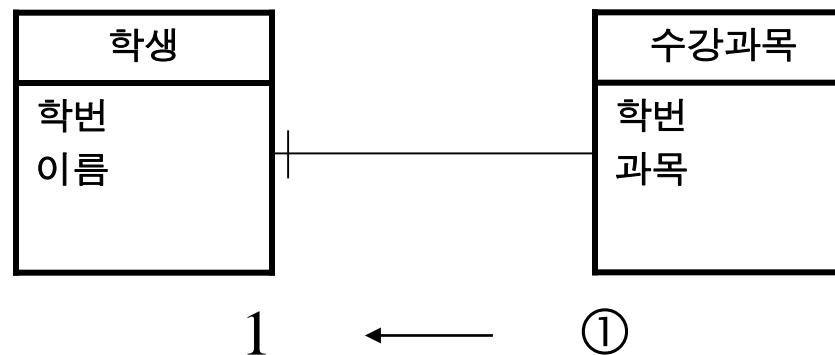
<그림 3.13> 학생 1명이 N개의 과목과 관련이 있음

3.4 관계(Relationship)

□ 예제

학생		수강과목	
학번	이름	학번	과목
21001	김철수	2001	전산학개론
21002	양길현	2001	이산수학
21003	임영수	2002	전산학개론
21004	박한나	2003	웹디자인
		2003	이산수학

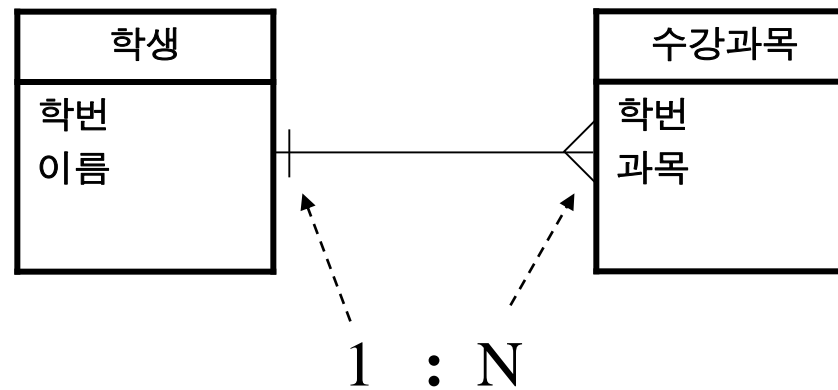
<그림 3.14> 수강과목쪽에서 학생과의 카디널리티



<그림 3.15> 수강과목 1개의 정보는 학생 1명과 관계를 가짐

3.4 관계(Relationship)

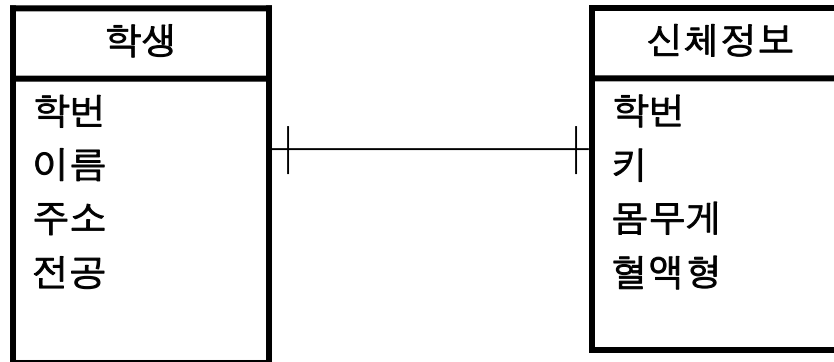
□ 예제



<그림 3.16> 학생~수강과목간의 카디널리티

3.4 관계(Relationship)

➤ 1:1 관계



1:1 관계에 있는 두 엔티티는 하나로 합칠 수 있다.

학생

학번	이름	주소	전공
21001	김철수	서울	영문학
21002	양길현	인천	컴퓨터
21003	임영수	광주	화학
21004	박한나	부산	수학

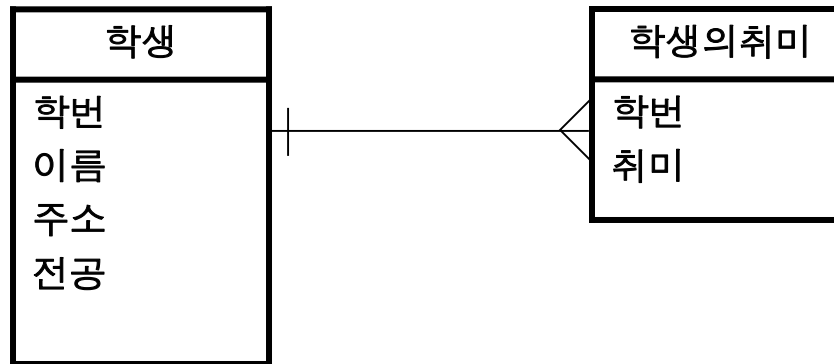
신체정보

학번	키	몸무게	혈액형
21001	175	70	A
21002	169	65	B
21003	180	60	O
21004	170	85	B

<그림 3.17> 1:1 카디널리티의 예

3.4 관계(Relationship)

➤ 1:N 관계



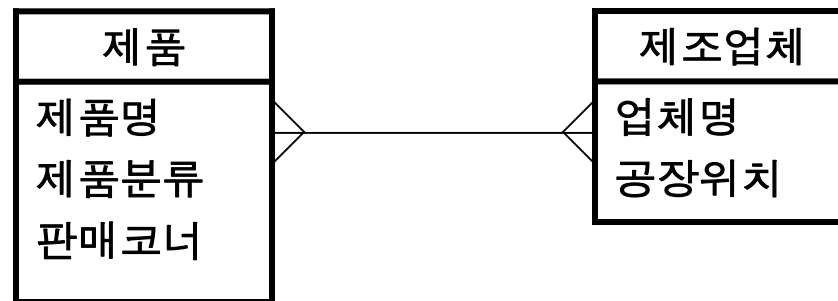
대부분의 관계는
1:N 관계이다

학생				학생의취미	
학번	이름	주소	전공	학번	취미
21001	김철수	서울	영문학	21002	낚시
21002	양길현	인천	컴퓨터	21002	등산
21003	임영수	광주	화학	21003	낚시
21004	박한나	부산	수학	21004	여행

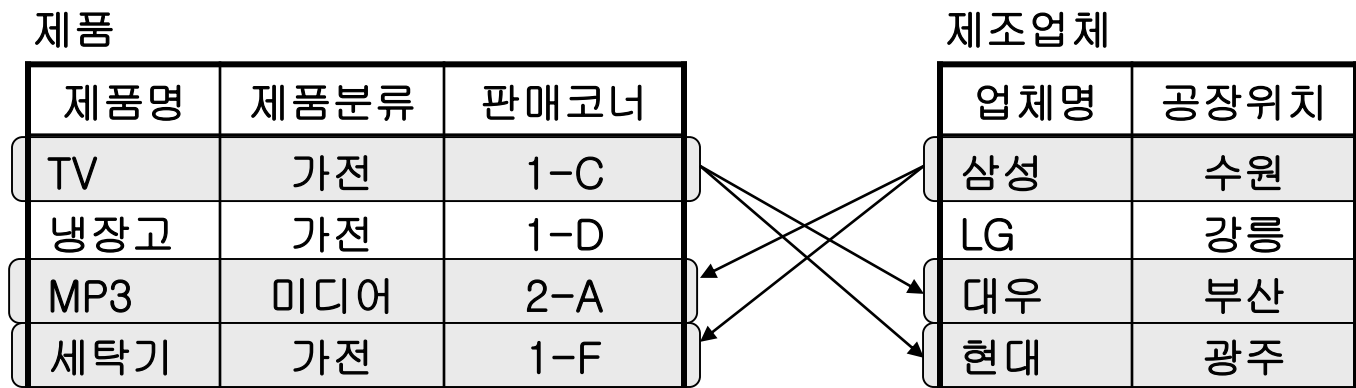
<그림 3.18> 1:N 카디널리티의 예

3.4 관계(Relationship)

➤ M:N 관계



M:N 관계는 아직 완성되지 않은 작업으로 간주되며, M:N 관계를 해소하기 위한 추가 작업이 필요하다

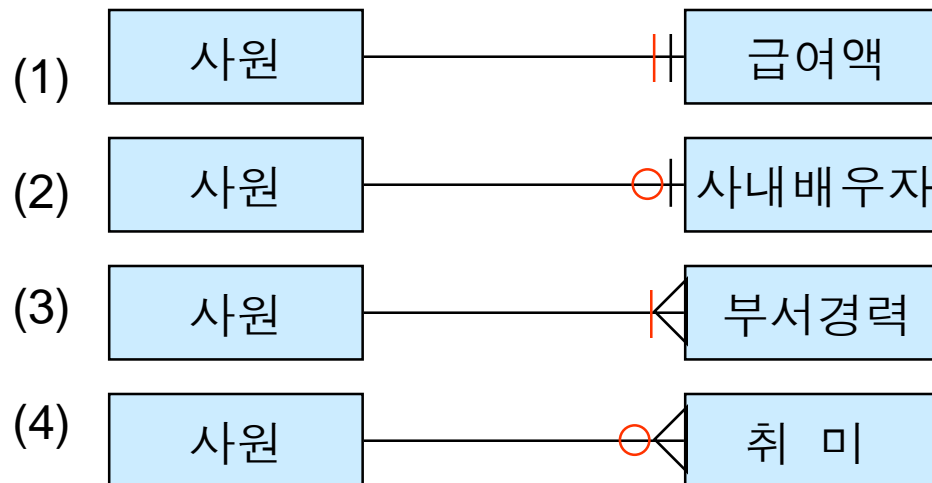


<그림 3.19> M:N 카디널리티의 예

3.4 관계(Relationship)

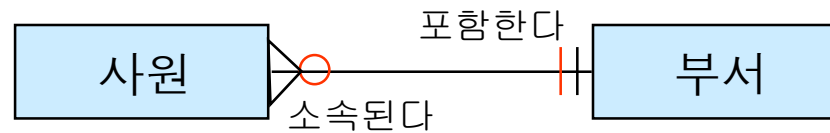
□ 관계의 참여도

- 관계가 있는 두 엔티티의 인스턴스들이 항상 관계에 참여 하는지, 아니면 경우에 따라 관계에 참여 하는지 여부
 - 필수 : |
 - 선택 : o



3.4 관계(Relationship)

□ 예제

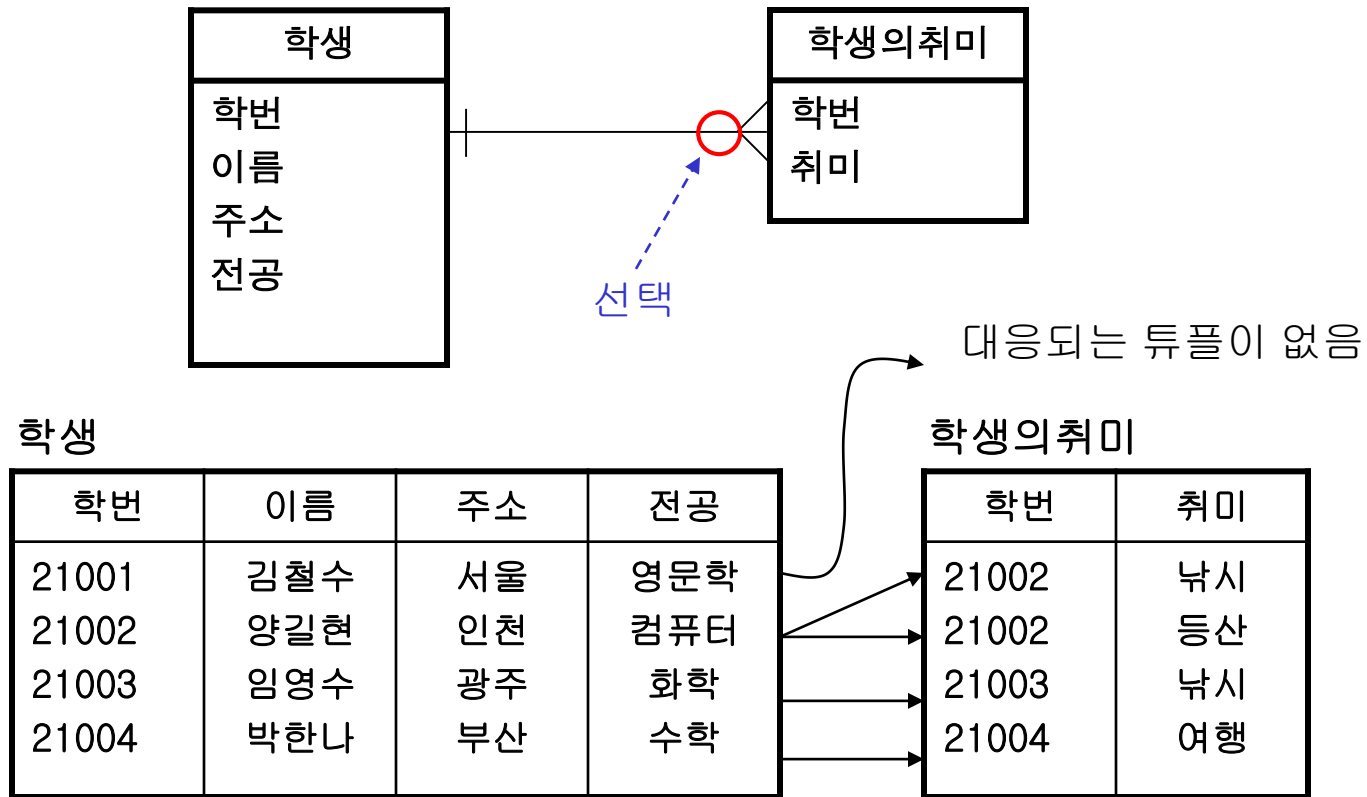


- 모든 사원은 반드시 부서를 가지며, 단 하나의 부서에 소속된다.
- 어떤 부서는 특정 시점에서 사원이 없을수도 있고, 있는 경우 여러 사원을 포함할 수 있다.

** 두 엔티티 사이의 관계가 어떠한지는 전적으로
현실세계의 의미를 따른다.

3.4 관계(Relationship)

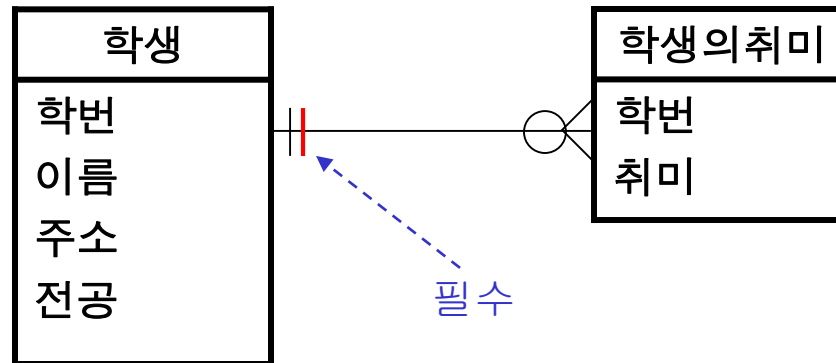
□ 예제



<그림 3.20> 학생은 취미를 선택적으로 갖는다.

3.4 관계(Relationship)

□ 예제



학번	이름	주소	전공
21001	김철수	서울	영문학
21002	양길현	인천	컴퓨터
21003	임영수	광주	화학
21004	박한나	부산	수학

학번	취미
21002	낚시
21002	등산
21003	낚시
21004	여행

<그림 3.21> 학생의 취미정보는 관련된 학생정보를 필수적으로 갖는다.

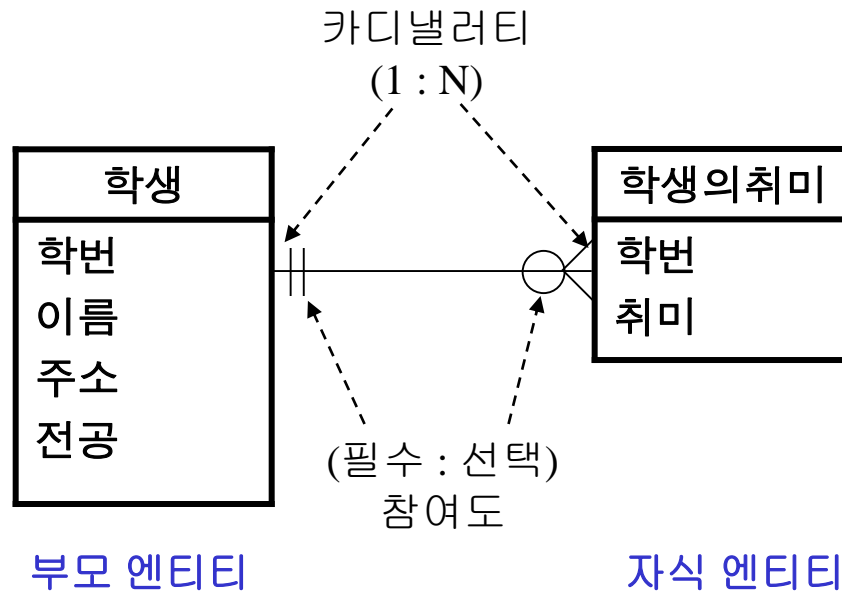
3.4 관계(Relationship)

□ 부모 엔티티와 자식 엔티티

- 상호 관계가 있는 두 엔티티는 부모-자식의 관계에 있는 경우가 많다
- 부모, 자식 여부는 어느쪽에 정보가 먼저 생성이 되는가에 따라 결정 된다
 - 정보가 먼저 생성되는 쪽이 부모, 가져다 쓰는 쪽이 자식
- 두 엔티티가 부모-자식의 관계가 있다면 일반적으로 부모 엔티티와 자식 엔티티의 카디널러티는 1:N 이고 참여도는 부모쪽이 필수, 자식쪽이 선택으로 나타난다

3.4 관계(Relationship)

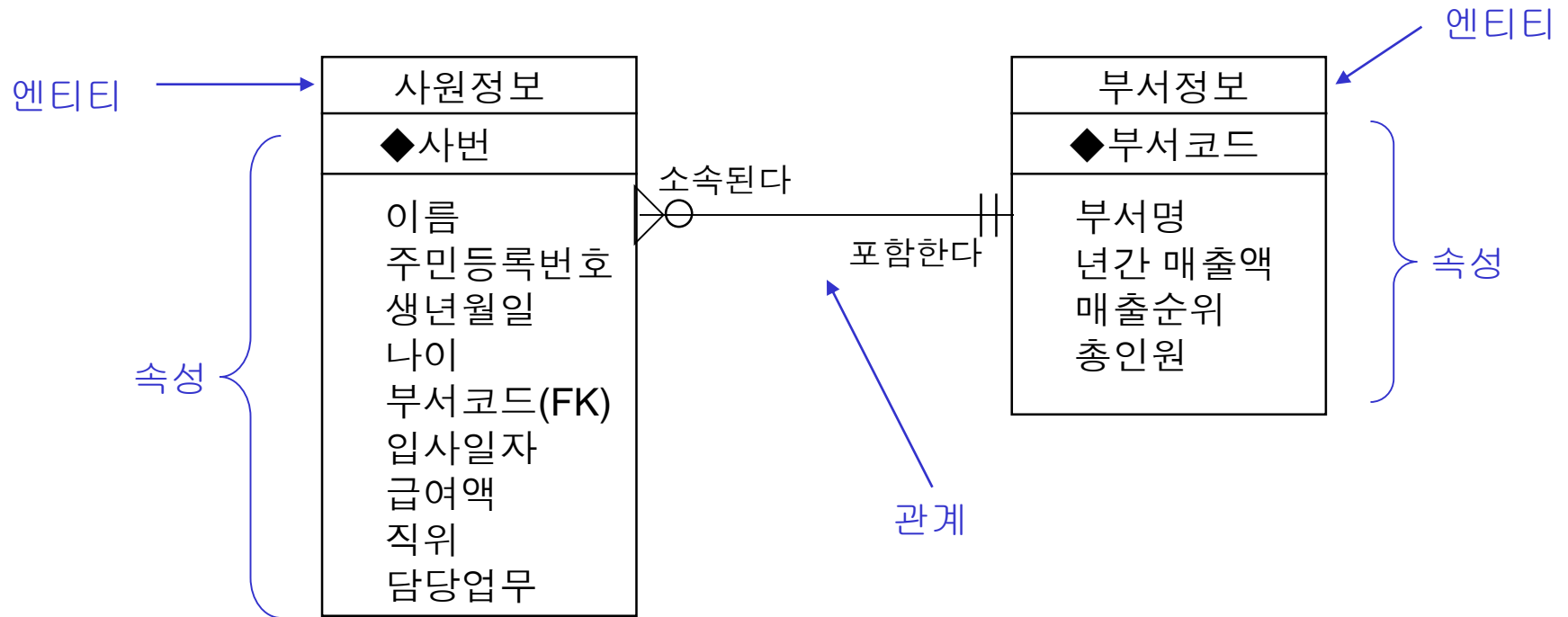
□ 부모 엔티티와 자식 엔티티



<그림 3.22> 부모-자식 관계에서 일반적인 카디널리티와 참여도

중간 정리

□ 엔티티 / 관계 / 속성

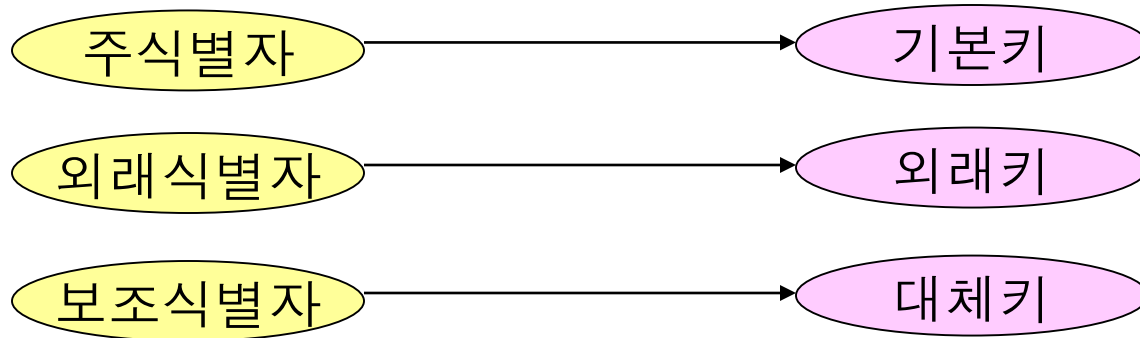


3.5 주식별자와 외래 식별자

□ 식별자 vs 키

Data Modeling 단계

물리적DB 설계 단계



3.5 주식별자와 외래 식별자

□ 주식별자의 표현 방법

사원정보
◆ 사번
이름 주민등록번호 생년월일 나이 부서코드

사원정보
사번(PK)
이름 주민등록번호 생년월일 나이 부서코드

사원정보
사번(PK)
이름 주민등록번호 생년월일 나이 부서코드

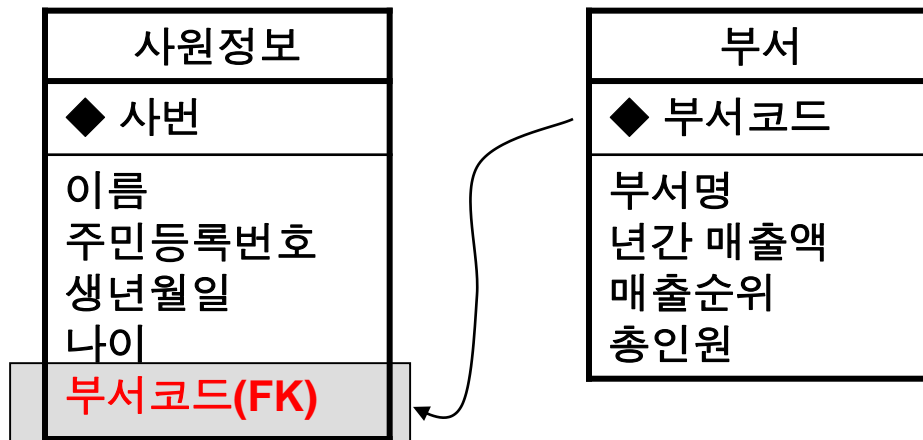


이것을 사용

엔티티 내에서 인스턴스와 인스턴스를 구별하는 기준

3.5 주식별자와 외래 식별자

□ 외래 식별자의 표현



엔티티와 엔티티를 연결해 주는 고리 역할

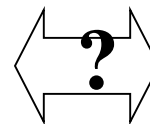
3.5 주식별자와 외래 식별자

□ 외래 식별자의 역할

사원정보
◆ 사번
이름 주민등록번호 생년월일 나이

부서정보
◆ 부서코드
부서명 년간 매출액 매출순위 총인원

98001	홍길동	770316-1203123	77.3.16	25
98002	김철수	760519-1205431	76.5.19	26

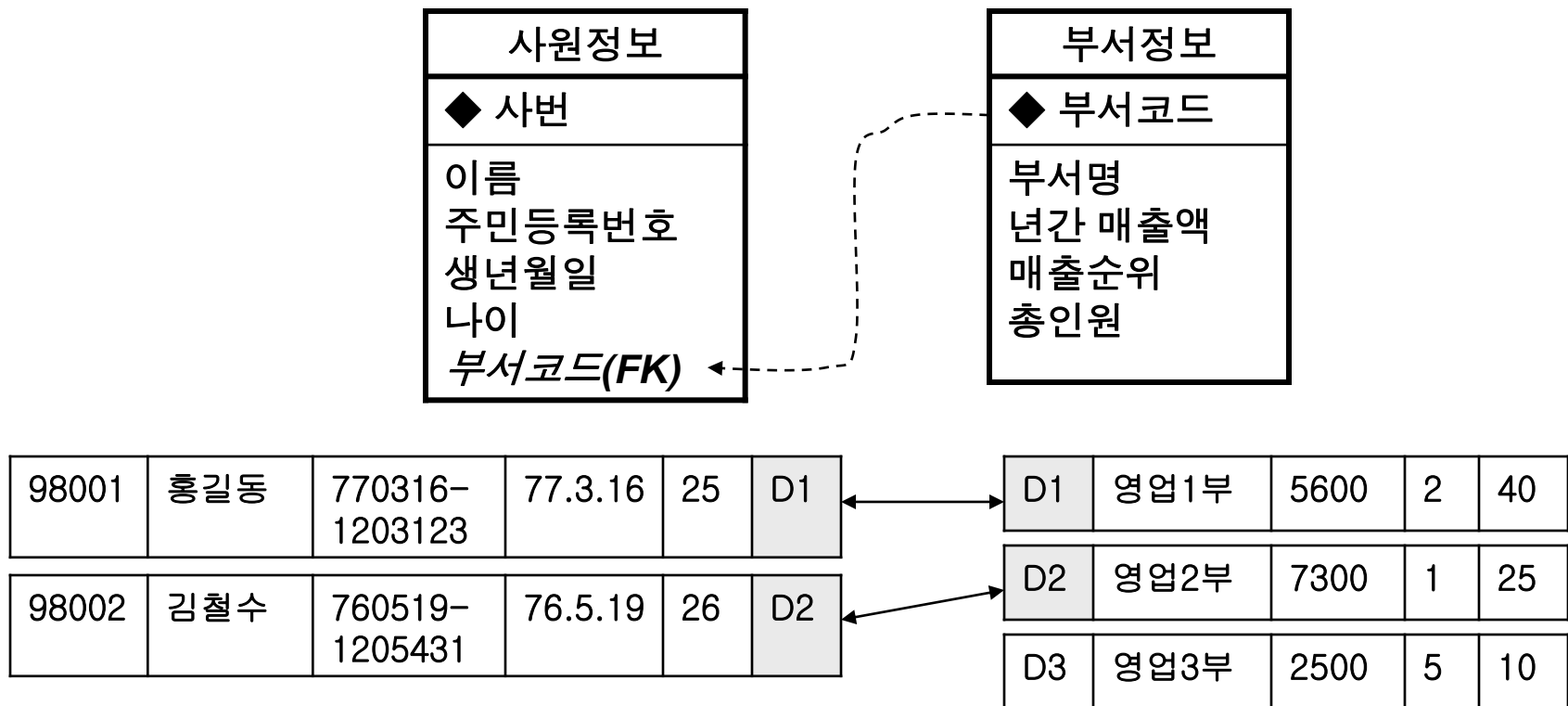


D1	영업1부	5600	2	40
D2	영업2부	7300	1	25
D3	영업3부	2500	5	10

<그림 3.24> 데이터 관점에서는 상호 관련성이 없는 두 엔티티

3.5 주식별자와 외래 식별자

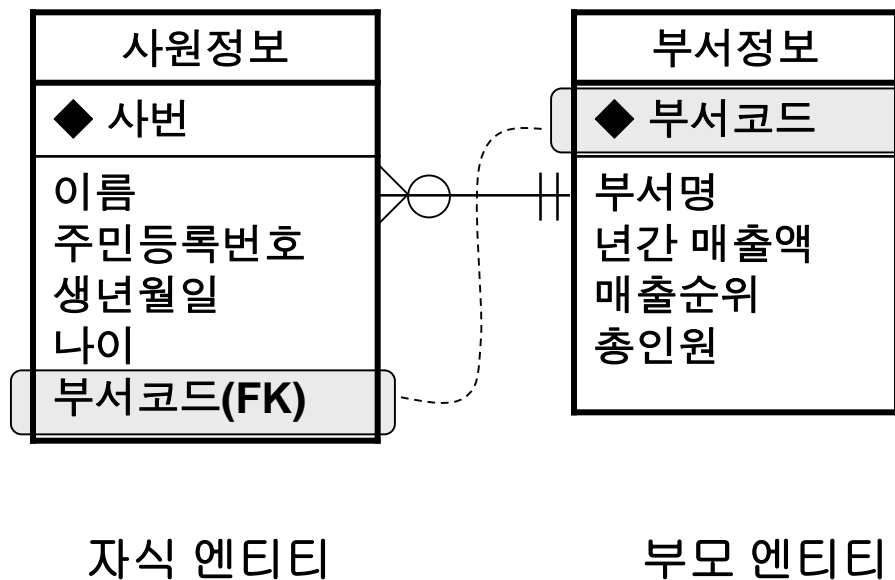
□ 외래 식별자의 역할



<그림 3.25> 외래식별자에 의해 연결된 두 엔티티

3.5 주식별자와 외래 식별자

□ 부모 엔티티와 자식 엔티티에서의 식별자



부모 엔티티의 **주식별자**는 자식 엔티티의 **외래식별자**와 연결된다.

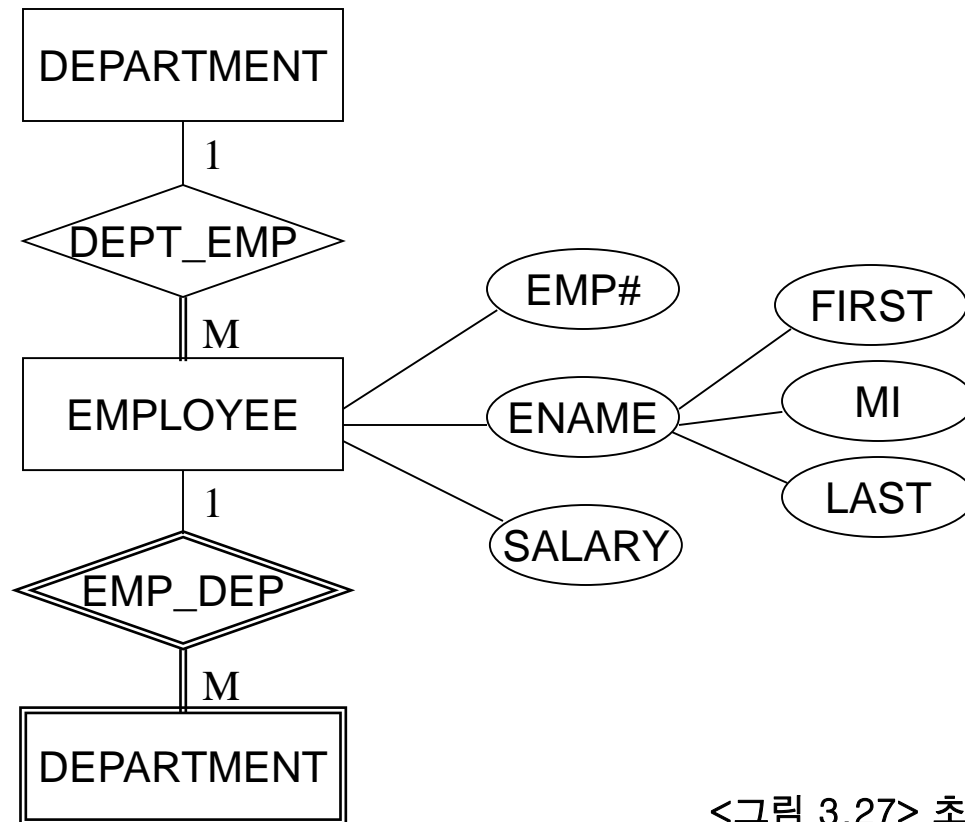
3.5 주식별자와 외래 식별자

□ *Note*

- 관계형 데이터베이스에서
 - 주식별자 속성은 null 값을 가질 수 없다.
 - 외래식별자 속성은 null 값을 가질 수 있다.

3.6 ERD 표기법

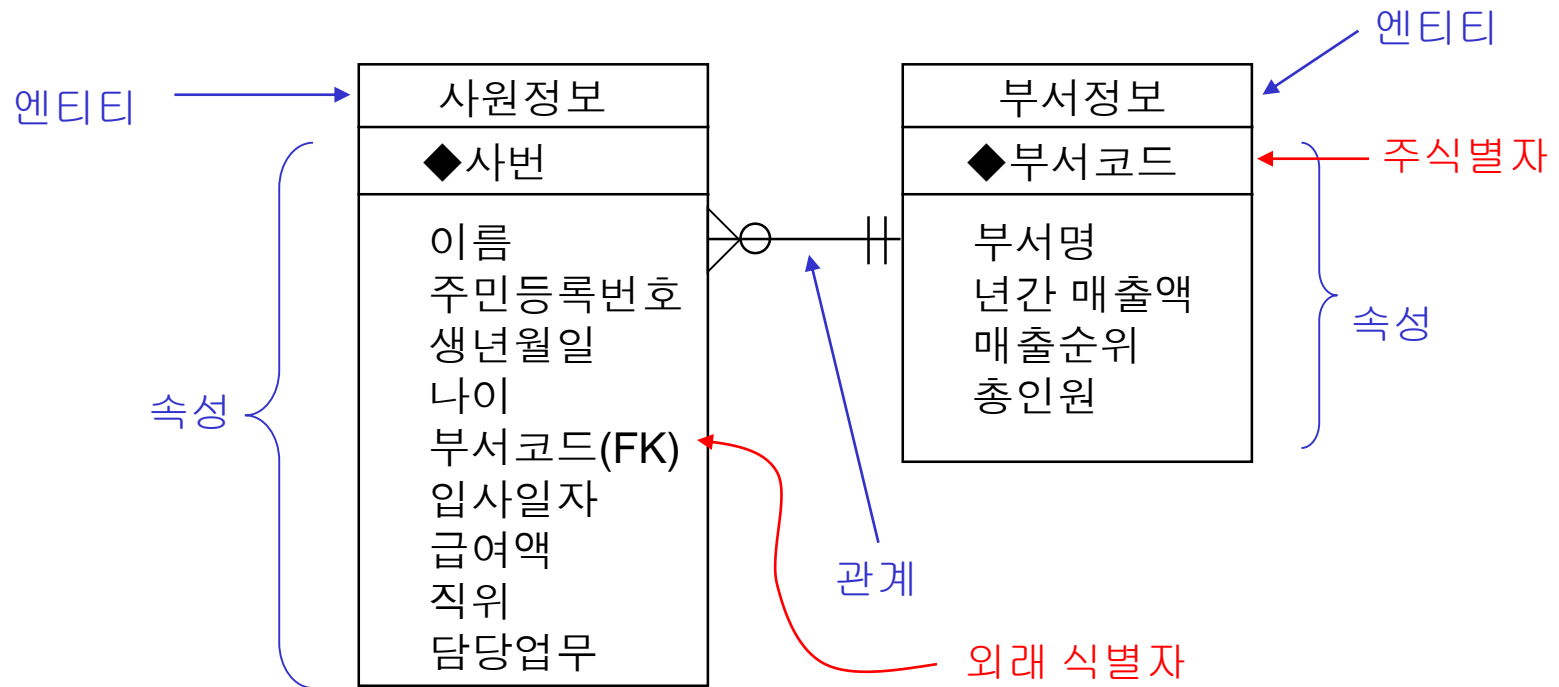
□ 초기의 ERD



<그림 3.27> 초기의 ERD

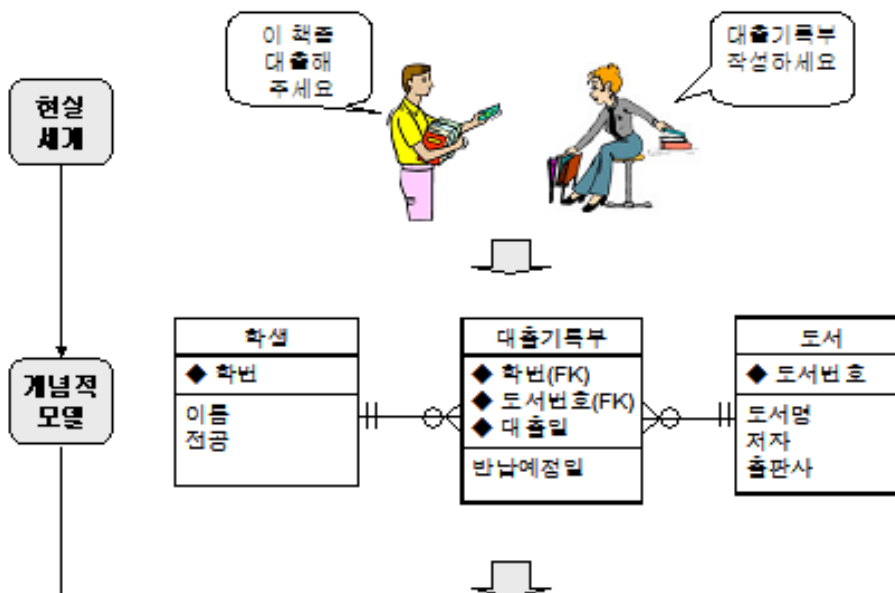
3.6 ERD 표기법

□ 일반적으로 사용되는 ERD



단원정리

- 데이터 모델링



학생

학번	이름	전공
21001	김철수	영문학
21002	양길현	컴퓨터
21003	임영수	화학
21004	박한나	수학

대출기록부

학번	도서번호	대출일	반납예정일
21001	B001	20050512	20050526
21001	B004	20050512	20050526
21004	B001	20050601	20050615
21004	B003	20050601	20050615

도서정보

도서번호	도서명	저자	출판사
B001	자바 프로그래밍	정용주	글벗
B002	컴퓨터 교육론	이원규	C미디어
B003	운영체제론	강길만	홍익
B004	인터넷 윤리	오예인	좋은씨앗

데이터
베이스