



데이터 모델링의 이해

# 데이터 모델링이란?

논리적 데이터 모델링이란 데이터베이스 설계 프로세스의 *Input*으로써 **비즈니스 정보의 구조와 규칙을 명확하게 표현하는 기법**이다.

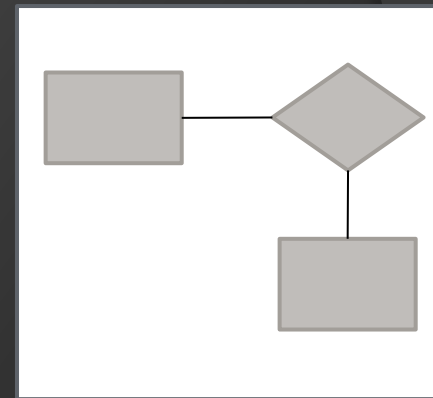
*Handbook of Relational Database Design - 1989  
Candace C. Fleming & Barbara von halle*

# 데이터 모델링이란?

현실세계



추상화(모형화)  
단순화  
명확화



모델 (MODEL)

**복잡한 현실세계를 일정한 표기법에 의해 표현하는 일**

# WHY ?

- 시스템의 구조(Data)와 업무 프로세스(Process)를 명세화 할 수 있다
- 현재 또는 원하는 시스템을 가시화하여 시스템을 구축하는 기본 틀을 제공한다
- 기업이 비즈니스의 목적을 바꾸지 않는 한 비즈니스에서 관리하는 데이터는 거의 영속적으로 변화하지 않는다

# WHY ?

- 가능성 실험 - 문제의 다양한 해결 방안을 실험
- 생산성 - 경제적인 효율과 효용성
- 이식성 - 비슷한 모델에 대해 바로 적용

# 데이터 베이스 시스템

# 스키마(Schema) - 데이터 구조의 표현

## 외부 스키마



기획자의  
뷰

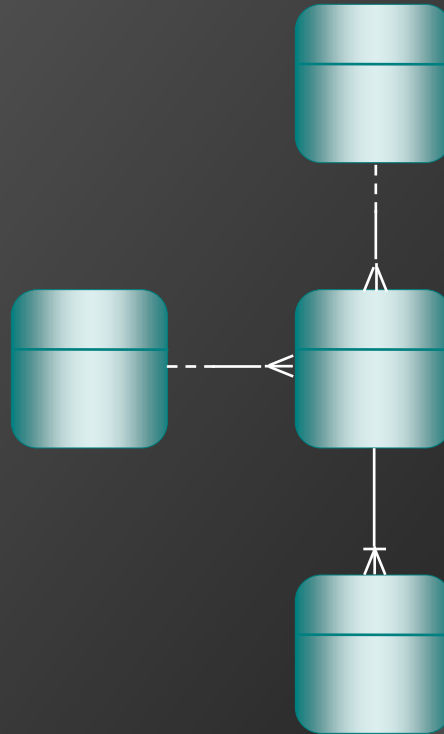


마케팅  
사원의 뷰



구매  
사원의 뷰

## 개념 스키마



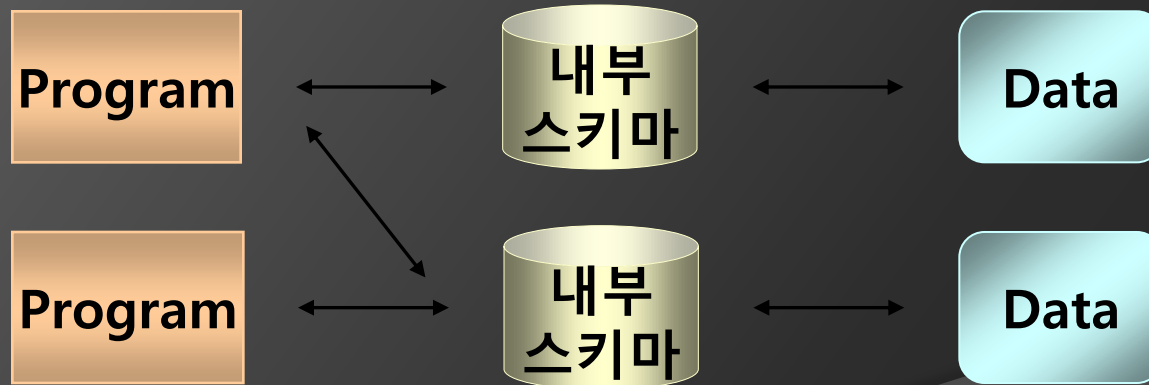
## 내부 스키마



# 스키마(Schema)의 표현

## 내부(Internal) 스키마

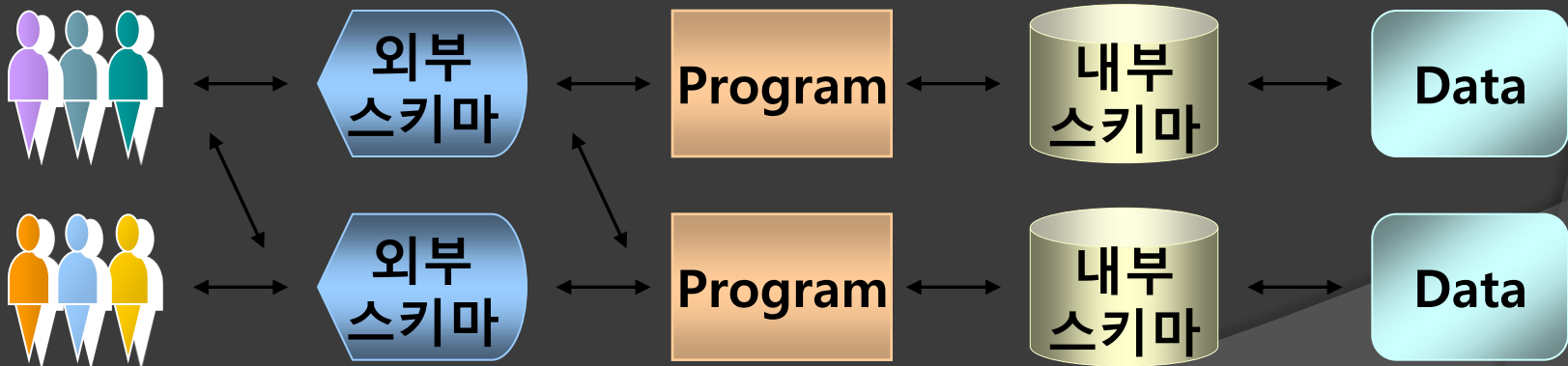
- 데이터의 물리적인 구조의 표현
- 코드가 데이터를 보는 관점
  - 파일 덤프





## 외부(External) 스키마

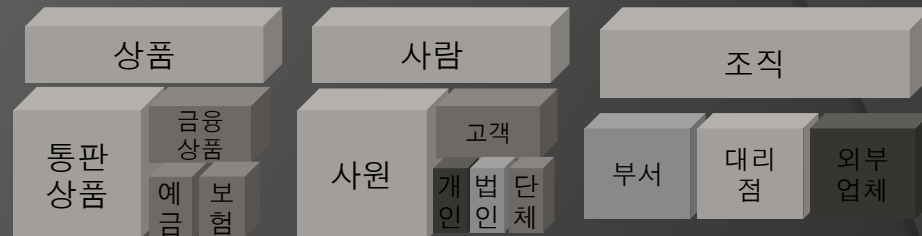
- **사용자에게 보여지는 데이터 구조의 표현**
- **사용자가 자신들만의 뷰로 데이터를 보는 방식**  
- form, report



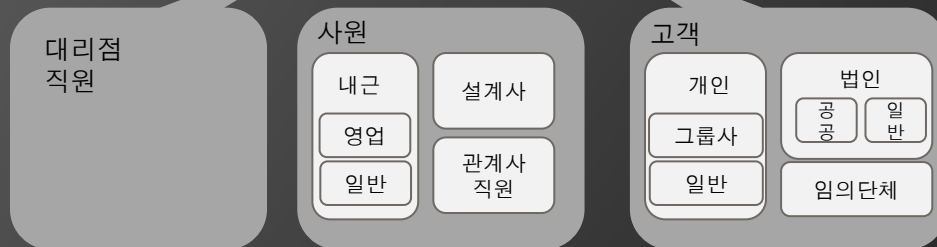
# 데이터 모델링 단계

# 데이터 모델링 단계

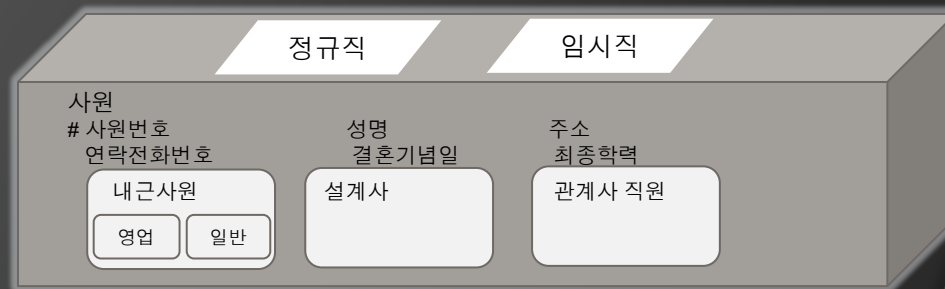
개괄적 모델



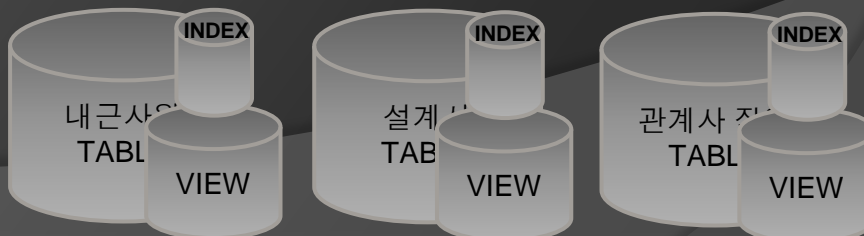
개념적 모델



논리적 모델



물리적 모델



# 논리 데이터 모델링

# 논리적 데이터 모델링이란?

- 데이터베이스 개발과정의 첫 단계
- 전략 수립 및 분석 단계에서 실시
- 요구되는 (분석한) 정보를 **실체(엔티티)**와 **관계**로 모델링하는 것

# 데이터 모델링의 3요소

1. **Entity** : 업무가 다루는 사물 (실체)
2. **Relation** : 사물들 사이에 존재하는 연관 (관계)
3. **Attribute** : 사물이 지니는 상세한 특성 (속성)

# 데이터 모델링의 3요소

개념(Concept)	타입(Type)	어커런스(Occurrence)
사물 사물들 사이의 연관 사물이 갖는 특성	실체(Entity) 관계(Relationship) 속성(Attribute)	인스턴스(Instance) 연관(association) 값(Value)

- 실체 : 사원, 부서 ..
- 인스턴스 : 이정화, 조인성, 데이터기술팀, MIS팀 ..
- 관계 : 부서와 사원 사이의 관계
- 연관 : 데이터기술팀 소속의 이정화 ..
- 속성 : 사원번호, 이름, 이메일 ..
- 값 : D1380, 이정화, tjdaum.daum@hanmail.net ..

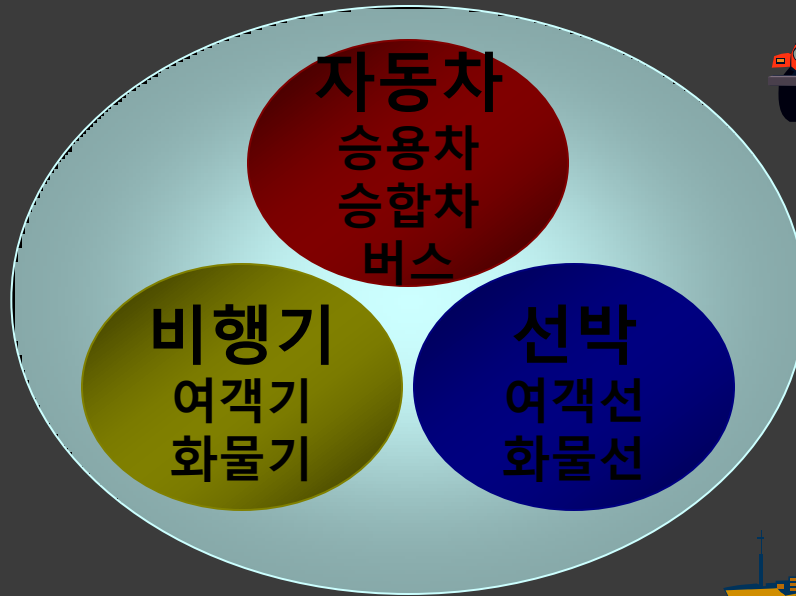
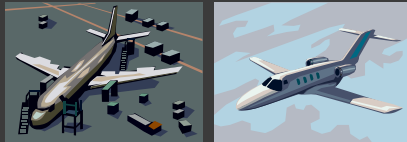
## Entity 란?

- Peter Chen: 다른 것과 구별되어 **식별**될 수 있는 사물
- C.J. Date: 데이터베이스에서 표현 되어야 하는 **구별** 가능한 객체
- James Martin: 실체란 우리가 정보를 저장하는 모든 사물(예: 고객, 공급자, 도구, 직원, 항공좌석 등)을 의미한다. 각각의 실체에 대해 특정 **속성**들이 저장된다.



# 데이터 모델링의 3요소 : Entity

“운송수단”



- 엔티티는 업무가 관심을 갖는, 비즈니스와 관련이 있는 **인스턴스들의 집합(Set)**을 나타낸다.

## Entity 의 특성

- 구현 조직에서 **저장, 관리하고자 하는 의미 있는 정보**를 관리
- Entity 명은 현실 세계의 객체 및 정보를 **대표**
- 실체가 포함하는 정보에 대해 무엇, 왜, 범위에 대해 명확히 표현할 수 있어야 함
- **면적(집합) = 가로(속성) \* 세로(개체)**

## 예제 : Entity 명확화

- 법인도 회원인가?
- 정규직과 비정규직은 같은 사원에 포함되는가?
- 우리 상품을 구매(혹은 가입, 계약)하지 않은 사람도 고객인가?
- 연예인을 가수, 연기자, 개그맨으로 나눌 때 낸시랭은 연예인인가 아닌가?
- 우리 회사 직원들도 상품을 구매할 수 있다면 고객인가?

# Entity 작성 예

인터넷 도서구매 쇼핑몰을 구축하려 한다. 업무 담당자와 인터뷰 후 다음과 같은 업무 시나리오가 작성되었다.

- 인터넷에서 **도서**를 구입하기 위해서는 반드시 우리 회사에 **회원**으로 등록해야 구매가 가능하다
- 인터넷에 회원 등록하는 정보로는 회원번호, 주민번호, 주소, 전화번호, 전자메일, 휴대폰번호, 결제방법이 있다.
- 인터넷을 통해 등록된 회원은 구매하기 위한 도서목록을 선택하고 인터넷 주문서양식에 주문내용을 입력하여 **주문**한다.
- 주문서양식에는 **주문목록**에 대한 상세 정보와 주문일자, 배송지주소, 배송지전화번호, 배송방법, 결제방법이 있다.
- 주문목록에 대한 상세 정보로는 구매가격, 개수가 있다.
- 한번 등록된 회원에 대해서는 도서를 구매하는 실적에 따라 구매 포인트를 부여하여 다음 구매 시 혜택을 보게 한다.

## Entity 작성 예

도서

회원

주문

주문목록

## 실습 : Entity 선정하기

우리는 전산전문가과정을 중심으로 실시하는 전문강좌를 제공하는 교육기관의 관리자이다. 우리는 많은 강좌를 가르치고 있으며 각 강좌는 CODE, 강좌명, 수업료 등을 가지고 있다. RDBMS 기초와 DATA MODELING & DESIGN은 우리의 인기 강좌중의 하나이다.

강좌는 매년마다 개설기간이 정해지며 상황에 따라 약간의 변경을 하기도 한다. 강좌는 크게 초급과정, 고급과정, 전문가과정, 특별과정으로 나누며, 개설된 강좌의 수업일수는 2~5 일간으로 다양하다.

각 강사는 여러 개의 강좌를 가르친다. 이몽룡과 성춘향은 우리의 최고 강사중의 하나이다. 우리는 각 강사의 이름과 전화번호를 관리한다. 각 개설된 강좌는 단 한명의 강사에 의해 진행된다. 우리는 강좌를 개설하고 강사를 배정한다. 어떤 강사는 다른 강좌의 학생으로 참여하기도 한다.

한 학생이 동시에 여러 강좌를 수강할 수 있으며 많은 학생이 그렇게 한다. 예를 들어 ABC전자의 홍길동은 우리가 제공한 모든 강좌를 수강했다. 우리는 각 학생의 성명과 전화번호를 관리하고자 하며, 때로는 학생과 강사가 그들의 전화번호를 알려주지 않을 때도 있다. 학생들 중의 대부분의 직장을 가지고 있으며 가능한 우리는 그들의 근무처를 관리하기를 원한다.

대부분의 강좌는 평가를 실시하고 평가에는 출석사항을 반영하며 평가된 결과를 관리한다.

# 데이터 모델링의 3요소 : Entity

## 실습 : Entity 선정하기

교육과정

강좌

강사(교사)

학생

평가

수강

## Relation 이란?

- 엔티티 상호간에 어떻게 연관되어 있는지를 파악하여 연관관계를 표현
  - ① 관계의 식별성 (Identification)
  - ② 선택성 (Optionality)
  - ③ 관계형태 (Degree)

모든 Entity1 은      단 하나      의      반드시 관계 해야 한다.  
Entity2 와/에      하나 이상      관계 일수도 있다.



# 데이터 모델링의 3요소 : Relation

## ■ 식별성 (Identification)

### 1. 식별 관계 (Identification Relation)

- | (바) 표시
- 부모 엔티티의 식별자(UID)가 자식 엔티티의 식별자(UID)의 일부가 되는 관계

### 2. 비식별 관계 (Non-Identification Relation)

- 표시 없음
- 부모 엔티티의 식별자(UID)가 자식 엔티티의 식별자(UID)의 일부가 아닌 관계



# 데이터 모델링의 3요소 : Relation

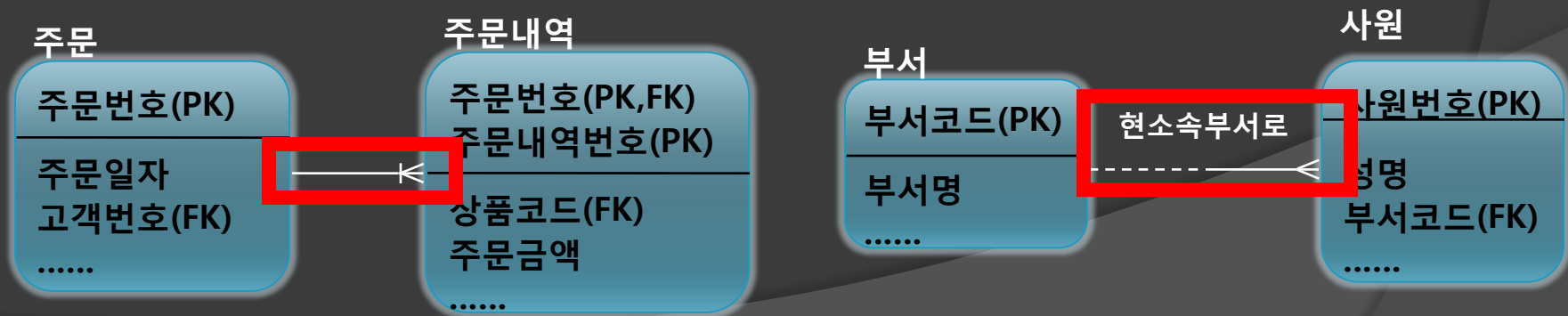
## ■ 선택성 (Optional)

### 1. 필수적 (Mandatory, Must)

- — (실선) 표시
- 자식 엔티티에 데이터를 입력하기 전에 부모 엔티티에 적어도 한 건 이상의 데이터가 있어야만 하는 경우

### 2. 선택적 (Optional, May)

- --- (점선) 표시
- 자식 엔티티에 데이터를 입력하기 전에 부모 엔티티에 데이터가 존재하지 않아도 상관이 없는 경우



# 데이터 모델링의 3요소 : Relation

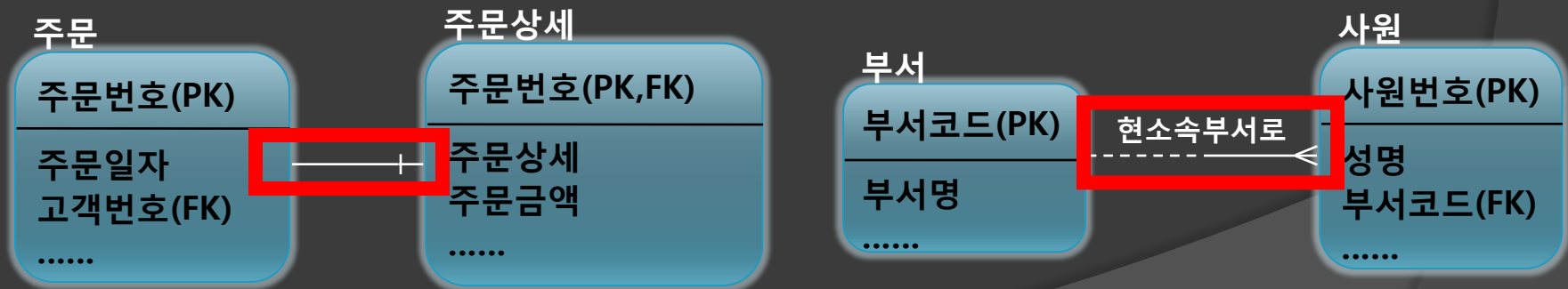
## ■ 기수성 (Degree)

### 1. 일

- — 로 표시
- 단 한 개 (One and Only One)

### 2. 여러 개

- ≪ 로 표시
- 하나 이상 (One or More)



# 데이터 모델링의 3요소 : Relation

## ■ 제 1자로서 보는 관계

- 관계명은 가능한 제1자의 입장에서 규정
- 다른 관계와 차별성 있게 구체적으로 지정



## ■ 제 3자로서 보는 관계

- 제 3자의 입장에서 관계의 내용을 보다 명확히 파악
- 관계를 규정할 때에는 반드시 제 1자의 입장에서

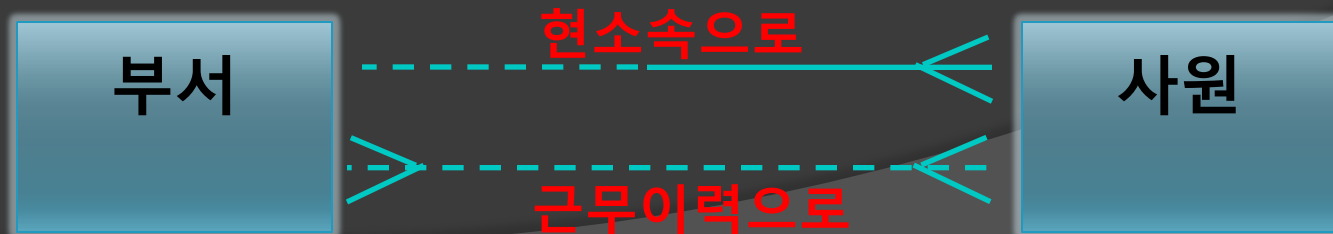


# 데이터 모델링의 3요소 : Relation

- 역할 위주로 정의하는 경우
  - 주는 쪽(One)의 특정 집합이 **서브타입**으로 지정되어 있지 않은 경우 주체들을 구체적으로 명시할 때 사용

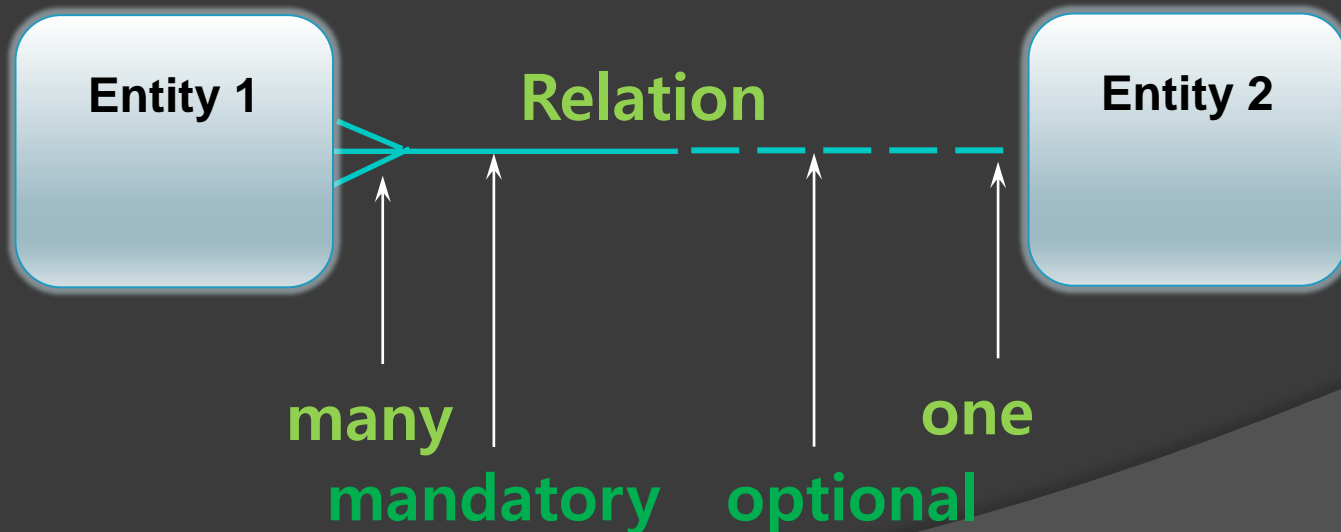


- 내용 위주로 정의하는 경우
  - 관계를 맺는 집합은 동일하나 관계를 맺는 내용이 다를 때 사용



# 데이터 모델링의 3요소 : Relation

- 릴레이션(Relation) 그리는 순서
  - ① 두 엔티티(Entity) 사이에 선을 그린다
  - ② 관계 명칭을 기록한다
  - ③ 선택사양(Optionality)를 표시한다
  - ④ 관계형태를 표시한다

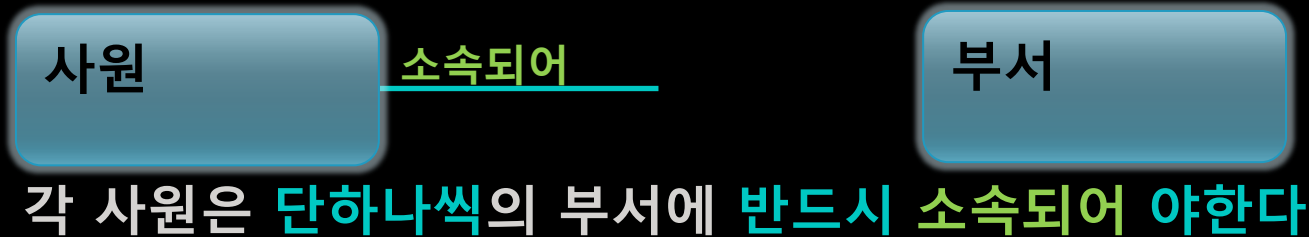


# 데이터 모델링의 3요소 : 릴레이션

## ■ 릴레이션(Relation) 읽기



좌에서 우로 읽기

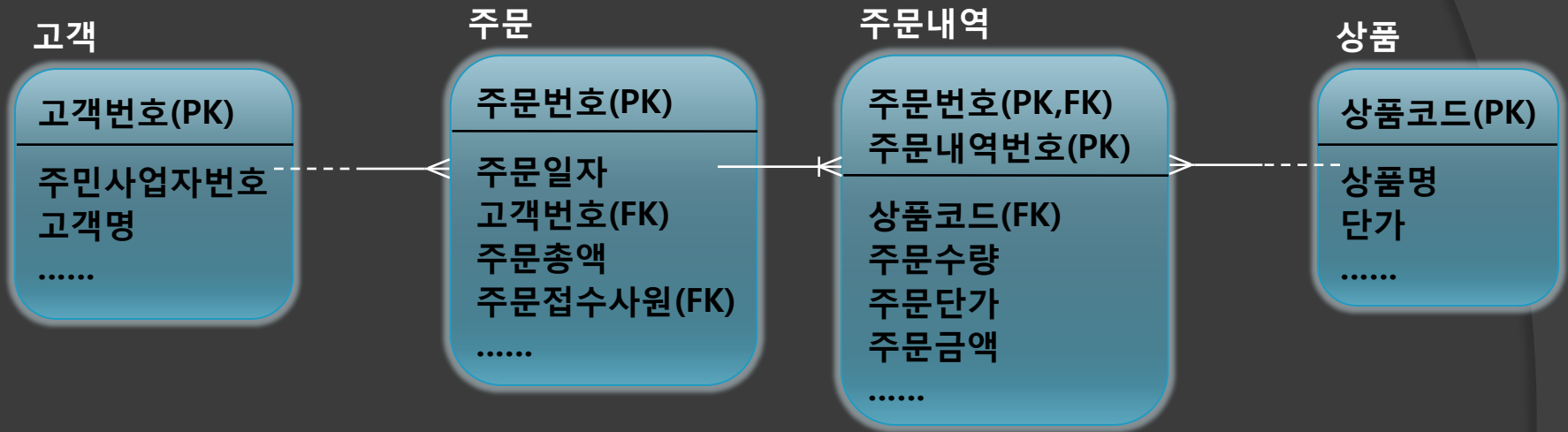


우에서 좌로 읽기



# 데이터 모델링의 3요소 : 릴레이션

## 릴레이션(Relation) 읽기 (Cont.)



“모든 고객은 **여러 개의** 주문을 **할 수도 있다.**”

“모든 주문은 **반드시 하나의** 고객에 **의하여 된다.**”

“모든 주문은 **여러 개의** 주문내역을 **반드시 포함한다.**”

“모든 주문내역은 **반드시 하나의** 주문에 **포함된다.**”

“모든 상품은 **여러 개의** 주문내역에 **포함 될 수도 있다.**”

“모든 주문내역은 **하나의** 상품을 **반드시 포함한다.**”



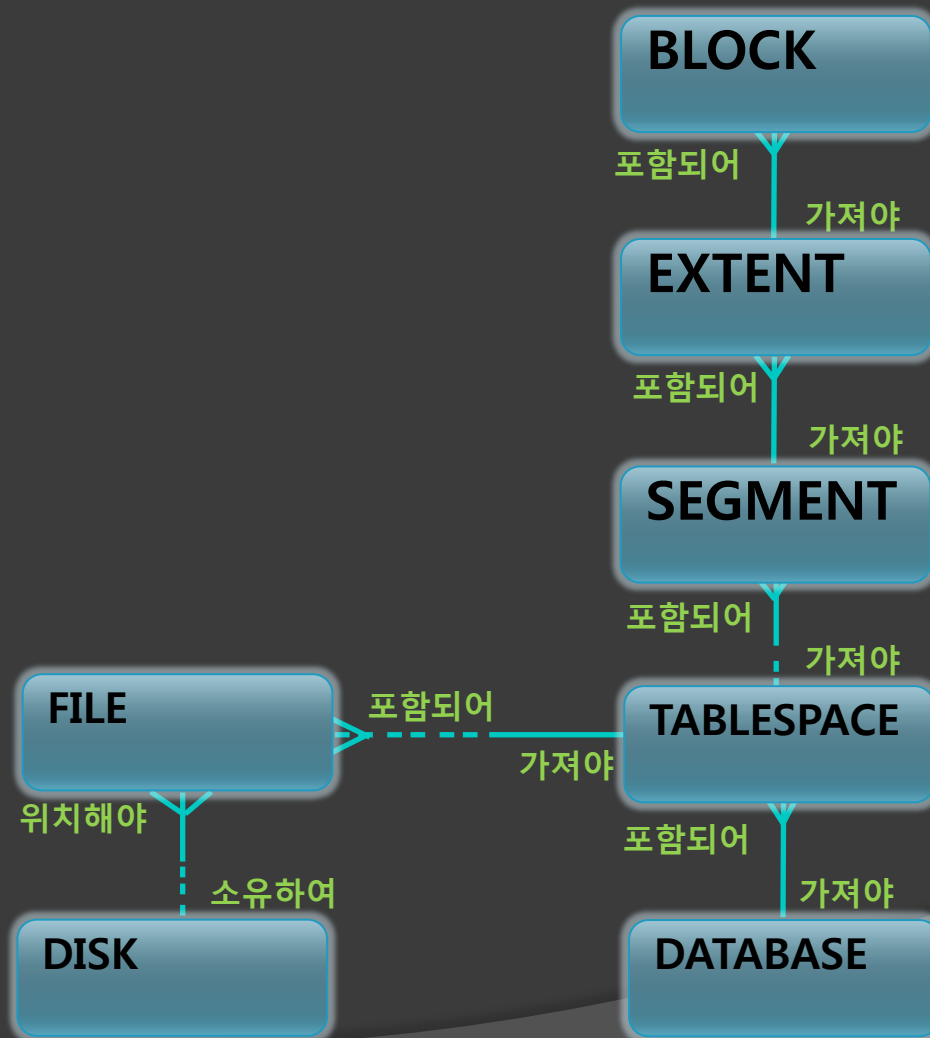
# 데이터 모델링의 3요소 : 릴레이션

## ■ 실습 : ER 작성하기

- 각 DATABASE는 **하나 이상의** TABLESPACE를 **반드시 가져야** 한다.
- 각 TABLESPACE는 **단 하나의** DATABASE에 **반드시 포함되어야** 한다.
- 각 TABLESPACE는 **하나 이상의** FILE을 **반드시 가져야** 한다.
- 각 FILE은 **단 하나의** TABLESPACE에 **포함될지도 모른다**.
- 각 TABLESPACE는 **하나 이상의** SEGMENT를 **가질지도 모른다**.
- 각 SEGMENT는 **단 하나의** TABLESPACE에 **반드시 포함되어야** 한다.
- 각 SEGMENT는 **하나 이상의** EXTENT를 **반드시 포함해야** 한다.
- 각 EXTENT는 **단 하나의** SEGMENT를 **반드시 가져야** 한다.
- 각 EXTENT는 **하나 이상의** BLOCK을 **반드시 포함해야** 한다.
- 각 BLOCK은 **단 하나의** EXTENT에 **반드시 속해야** 한다.
- 각 DISK는 **하나 이상의** FILE의 **소유자가 될지도 모른다**.
- 각 FILE은 **단 하나의** DISK에 **반드시 위치해야** 한다.

# 데이터 모델링의 3요소 : 릴레이션

## ■ 실습 : ER 모델 작성하기 (답안)



## 속성(Attribute) 이란?

속(屬) 성(性)

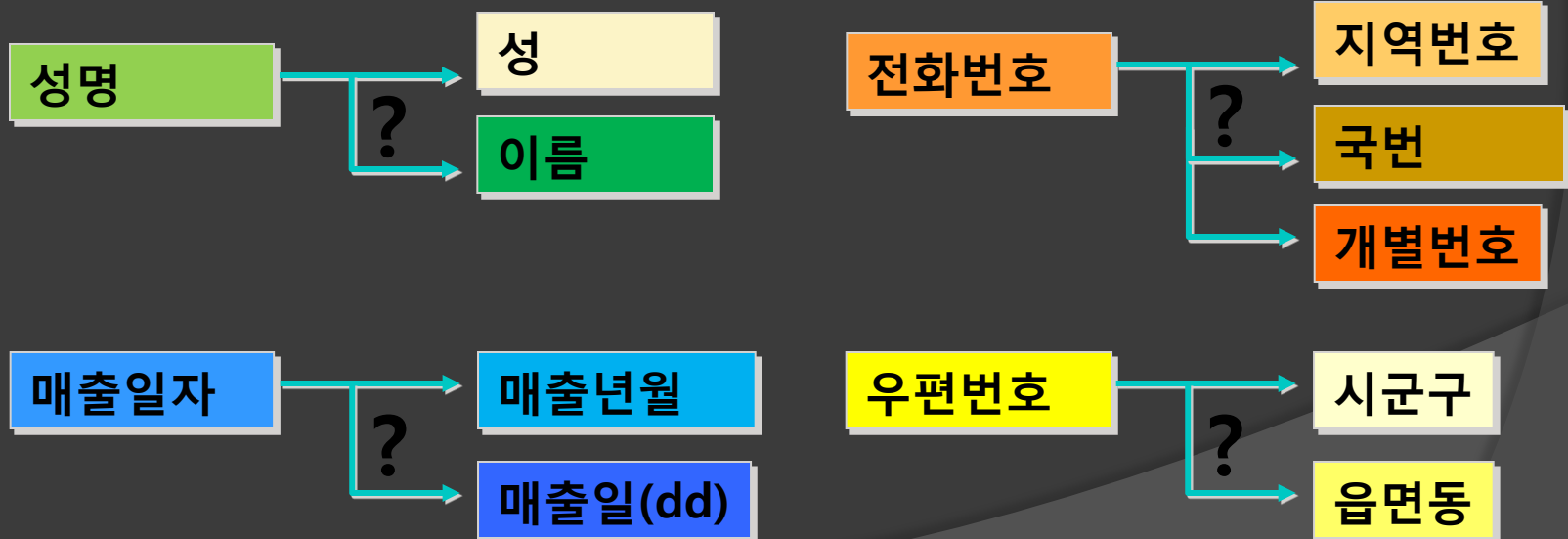
- 엔티티(실체,사물,개체)의 본질을 이루는 **고유한 성질이나 특성**으로서 우리가 관리하고자 하는 정보 항목들이다.

엔티티 (Entity, 실체)	속성 (Attribute)
<ul style="list-style-type: none"><li>사원</li><li>주문</li><li>제품</li></ul>	<ul style="list-style-type: none"><li>사원번호, 사원명, 이메일, 핸드폰번호</li><li>주문번호, 주문일, 주문상품, 결제여부</li><li>제품코드, 제품명, 가격, 색상</li></ul>

# 데이터 모델링의 3요소 : 속성

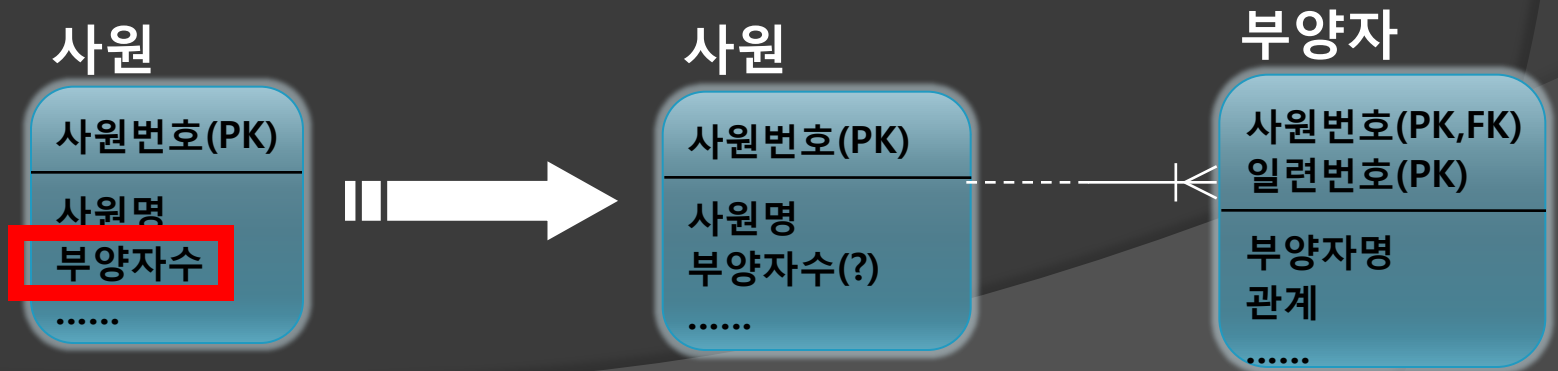
## ■ 속성(Attribute) 의 지정원칙

- ① 원자 단위인가?
- ② SINGLE VALUE 인가?
- ③ 추출값(Derived Value) 인가?
- ④ 보다 상세한 관리가 필요한가?



# 데이터 모델링의 3요소 : 속성

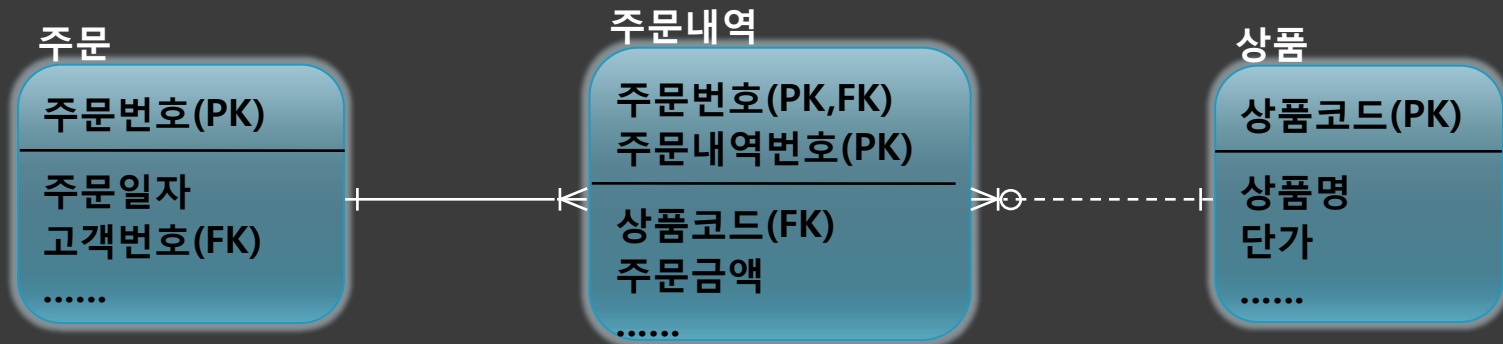
- **추출 속성 (Derived Attribute)**
  - 개수, 합계, 최대, 최소, 평균, 계산값 등
  - 추출 속성의 기본이 되는 속성값에 따라 변경된다
- **속성(Attribute) 과 엔티티(Entity)의 비교**
  - 속성이 자신의 속성을 가지면 엔티티(실체)
  - 실체는 속성을 갖지만 속성은 자신의 속성을 가질 수 없다



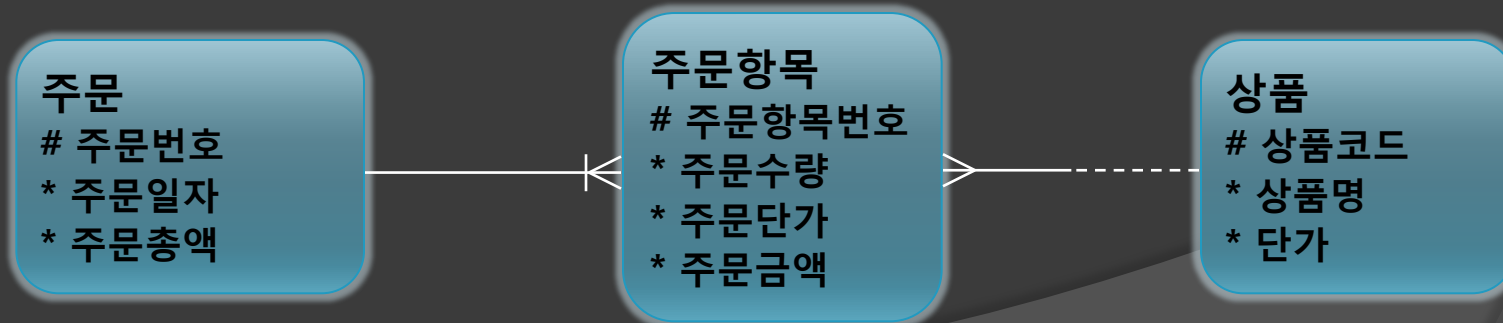
# 데이터 모델링의 3요소 : 속성

## ■ 속성(Attribute) 의 표현

### ■ IE Notation (정보 공학 방법론)



### ■ Barker Notation (CASE\*Method)



# 데이터 모델링의 3요소 : 속성

## ■ Notation 표기법 비교

IE Notation (정보공학방법론)	Many to One (M:1)			
	Many to Many (M:N)			
	One to One (1:1)			
Barker Notation (CASE* Method)	Many to One (M:1)			
	Many to Many (M:N)			
	One to One (1:1)			

# 데이터 모델링의 3요소 : 속성

## ■ 실습 : ER 모델 작성하기

나는 조그마한 비디오 가게를 운영하고 있으며 이 곳에서 관리할 테이프를 3000개 이상 보유하고 있다. 각 비디오 테이프는 테이프번호를 가지고 있으며, 각 영화마다 제목과 종류(예를 들어 코미디, 공포, 드라마, 액션, 전쟁 등)를 알 필요가 있다. 우리는 각 영화당 많은 테이프를 보유하고 있으며, 영화마다 특정번호를 부여하고 각 테이프가 어떤 영화를 포함하고 있는지 관리한다.

테이프는 Beta 혹은 VHS 방식일 수 있다. 우리는 각 영화를 위해 적어도 한 개 이상의 테이프를 보유하고 있으며, 각 테이프는 항상 한가지의 영화를 담고 있다. 보유한 테이프의 길이는 매우 길어 복수개의 테이프로 된 영화는 하나도 없다. 우리는 특정 배우가 출연한 영화를 자주 찾는다. 안성기나 강수연은 항상 인기가 좋다. 그래서 우리는 각 영화마다 주연배우를 알 필요가 있으며 본명 및 생년월일까지도 알고자 한다. 우리는 보유하고 있는 영화의 주연들에 대한 정보만 관리하고 싶다.

우리는 많은 고객을 보유하고 있으며 신용고객클럽에 가입한 회원들에게만 테이프를 대여한다. 이 클럽에 가입하기 위하여 고객은 좋은 신용을 가져야하며 각 회원들의 성명, 주소, 전화번호, 회원번호를 관리하고자 한다.

우리는 고객이 현재 어떤 테이프를 빌려갔는지를 관리하고자 하며 고객은 한번에 여러 개의 테이프를 빌려갈 지도 모른다. 또한, 우리는 현재의 대여정보만 관리하지 과거의 이력정보는 관리하지 않는다.



# 데이터 모델링의 3요소 : 속성

## ■ 실습 : ER 모델 작성하기 (답안)

	배우	고객	영화	테이프
배우	<u>          </u>	<u>          </u>	출연하여	<u>          </u>
고객	<u>          </u>	<u>          </u>	<u>          </u>	대여하여
영화	출연시켜	<u>          </u>	<u>          </u>	배포하여
테이프	<u>          </u>	대여되어	복사되어	<u>          </u>



# 데이터 모델링의 3요소 : 속성

## ■ 식별자(Identifier)란?

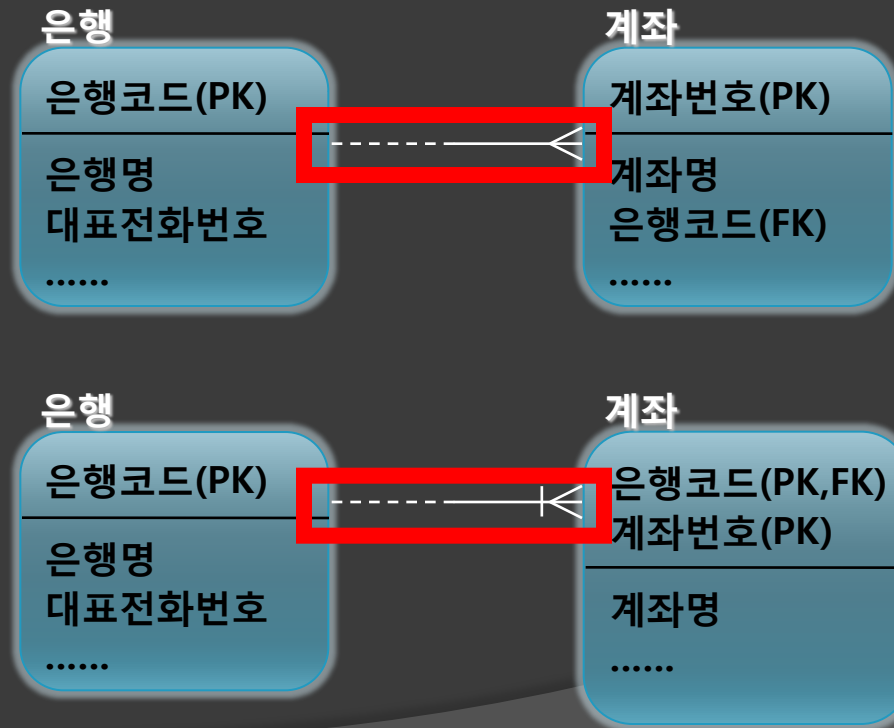
- **현실에서의 식별 문제**
  - 매우 유사한 특성을 가진 두 개의 사물을 어떻게 구분할 것인가
- **데이터베이스 내에서의 식별 문제**
  - 테이블내의 여러 개의 행(row)들을 어떻게 구별할 것인가
- **표현의 문제**
  - 테이블에서 실 세계의 사물이 표현된 것을 어떻게 알 수 있는가



# 데이터 모델링의 3요소 : 속성

## ■ 식별자(Identifier)의 검증

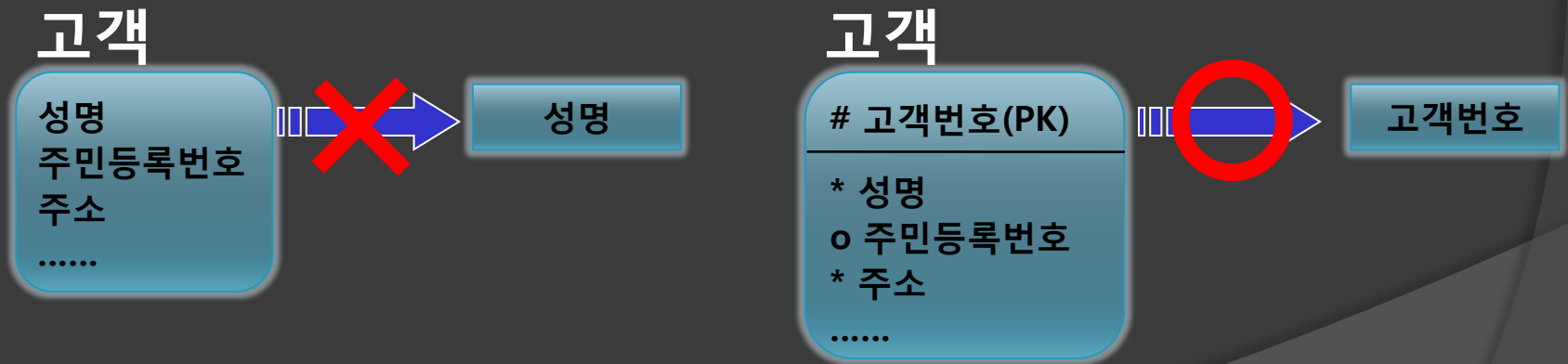
- 식별자를 구성하는 속성과 관계가 **NOT NULL**인가
- 식별자를 구성하는 속성과 관계가 **UNIQUE** 한가
- 식별자를 구성하는 속성과 관계가 **MINIMAL SET** 인가



# 데이터 모델링의 3요소 : 속성

## ■ 인위적인 식별자

- 식별자로 사용할 속성이 없거나 부적절한 경우 종종 인위적인 식별자가 사용됨
- 식별자의 길이가 너무 긴데 자주 사용해야 하는 경우
- 이미 실무에서 사용 중이거나 범용적인 속성을 사용할 것



# 데이터 모델링의 3요소 : 속성

## 사원

# 사원번호(PK)

- \* 성명
- \* 주소
- \* 부서코드(FK)
- .....

## 당초계약

# 당초계약번호(PK)

- \* 계약일자
- o 자금원
- .....

## 주문

# 주문번호(PK)

- \* 주문일자
- \* 고객번호(FK)
- \* 주문총액
- \* 주문접수사원(FK)
- .....

## 고객

# 고객번호(PK)

- \* 고객명
- \* 주소
- o 전화번호

### 고객구분

#### 개인고객

- \* 주민등록번호

#### 법인고객

- \* 사업자번호

## 실행계약

# 실행계약번호(PK)

- \* 순번
- \* 실행일자
- \* 개시일자
- \* 만기일자
- \* 당초계약번호(FK)
- .....

## 상품

# 상품코드(PK)

- \* 상품명
- \* 단가
- .....

## 주문내역

# 주문번호(PK,FK)  
# 주문내역번호(PK)

- \* 상품코드(FK)
- \* 주문수량
- \* 주문단가
- \* 주문금액
- .....

# 실전 데이터 모델링 1

## - 정규화

## 정규화(Normalization) 란?

논리적인 데이터 설계의 목적인 **정확성, 일관성, 단순성, 비중복성, 안정성**을 만족시키는 최적의 개념적 데이터

- 데이터 구조의 안정성을 최대화한다
- 불필요한 데이터 중복을 최소화한다
- 업무 규칙을 보다 세밀하게 체크할 수 있다
- 데이터 저장공간을 최소화한다
- 모형 관리 및 통합에 도움이 된다

## ■ 제1 정규형

- 모든 속성은 반드시 하나의 값을 가져야 한다
- 반복 또는 복수 값을 갖는 속성을 제거

## ■ 제2 정규형

- 모든 속성은 반드시 UID 전부에 종속되어야 한다
- UID의 일부에만 종속되는 속성을 제거

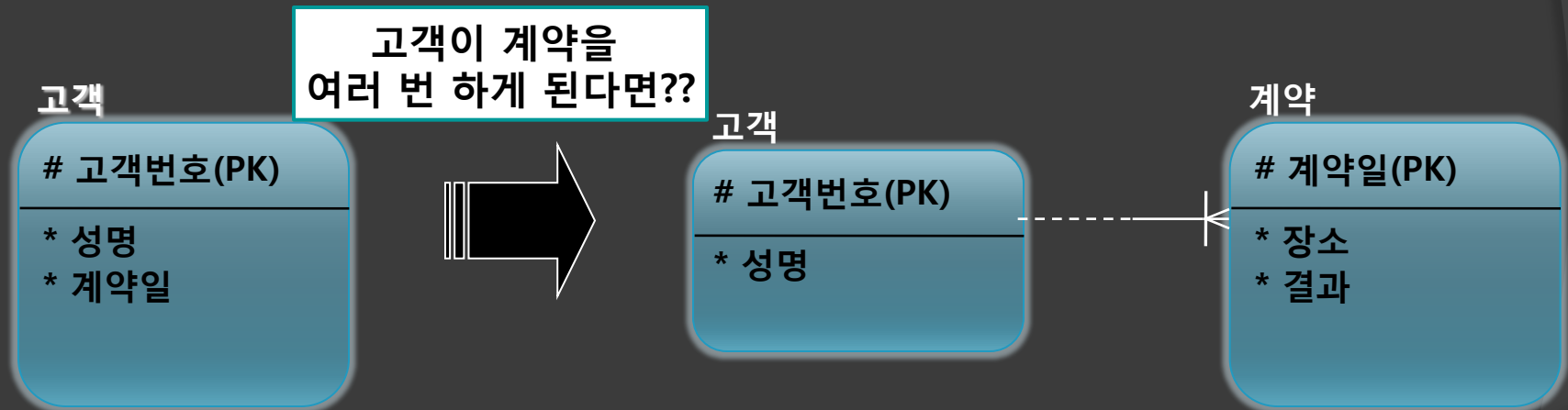
## ■ 제3 정규형

- UID가 아닌 속성간에는 서로 종속될 수 없다
- UID가 아닌 일반 속성에 종속적인 속성을 제거
- 데이터 중복을 제거하기 위한 데이터베이스 설계의 일반적인 목표



## ■ 제1 정규형

- 어떤 속성이 여러 개의 값을 갖고 있다면 1 : N 관계의 새로운 엔티티를 추가한다.



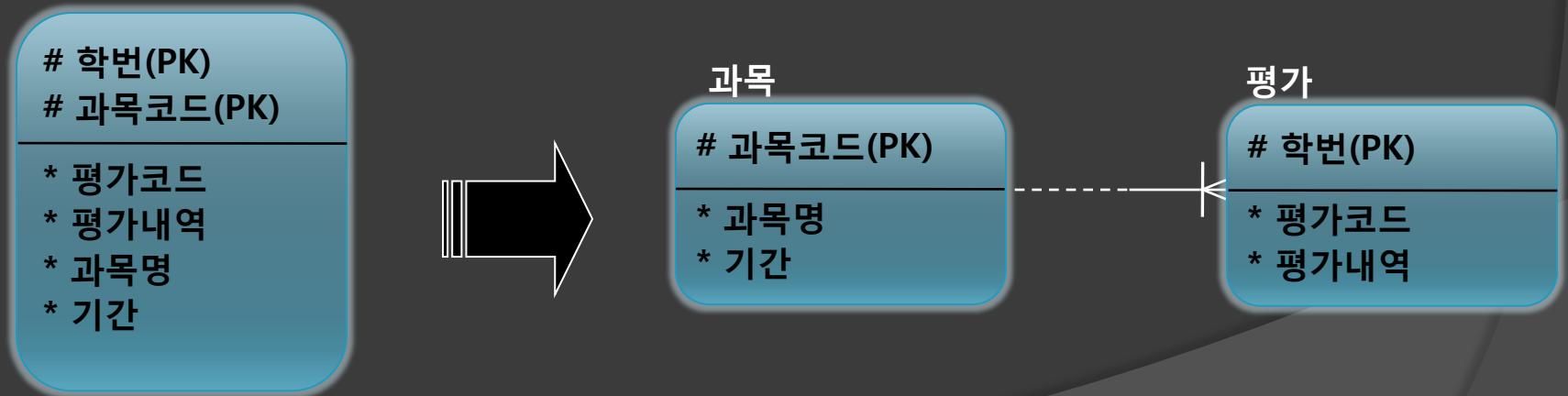
## ■ 제2 정규형

- 어떤 속성이 UID 전체에 종속되어 있지 않다면 새로운 엔티티를 만들어서 추가한다

평가코드,평가내역,과목명,기간 → 학번+과목코드 에 종속

과목명, 기간 → 과목코드 에도 종속

과목평가

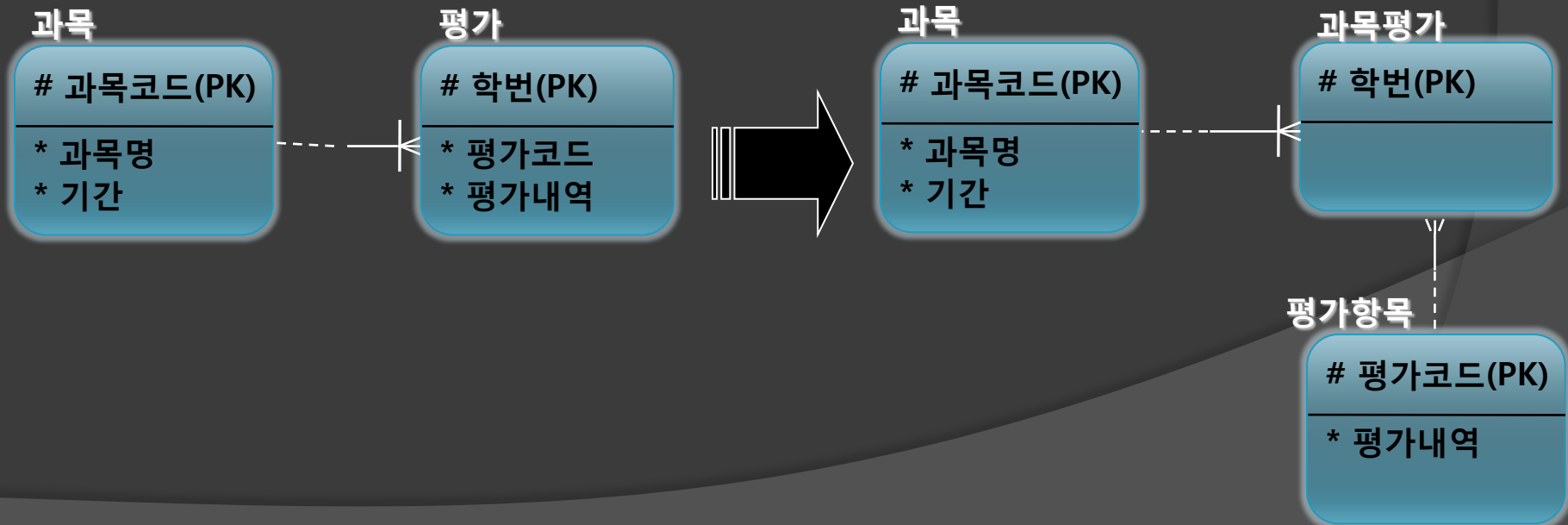


## ■ 제3 정규형

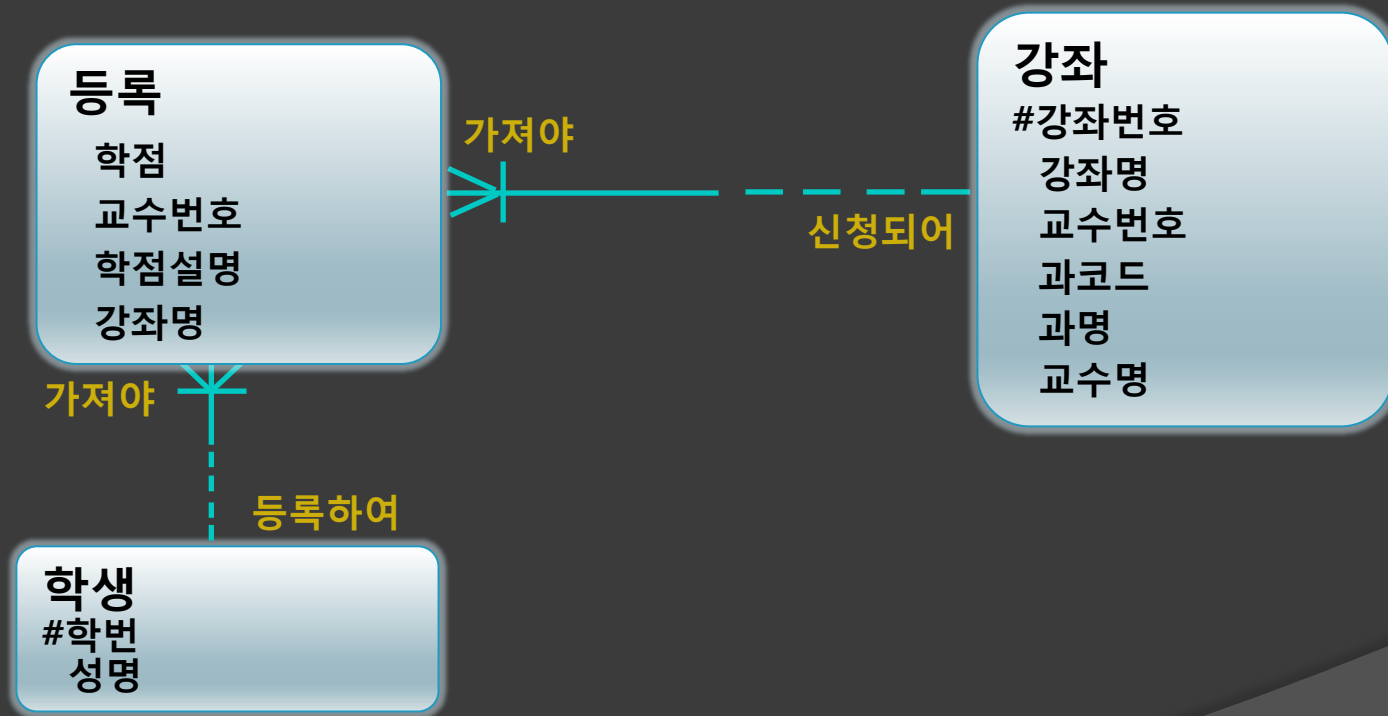
- 속성간에 종속관계가 존재하면 새로운 엔티티로 분리하여 속성간 종속을 제거한다

평가코드, 평가내역 → 학번 에 종속

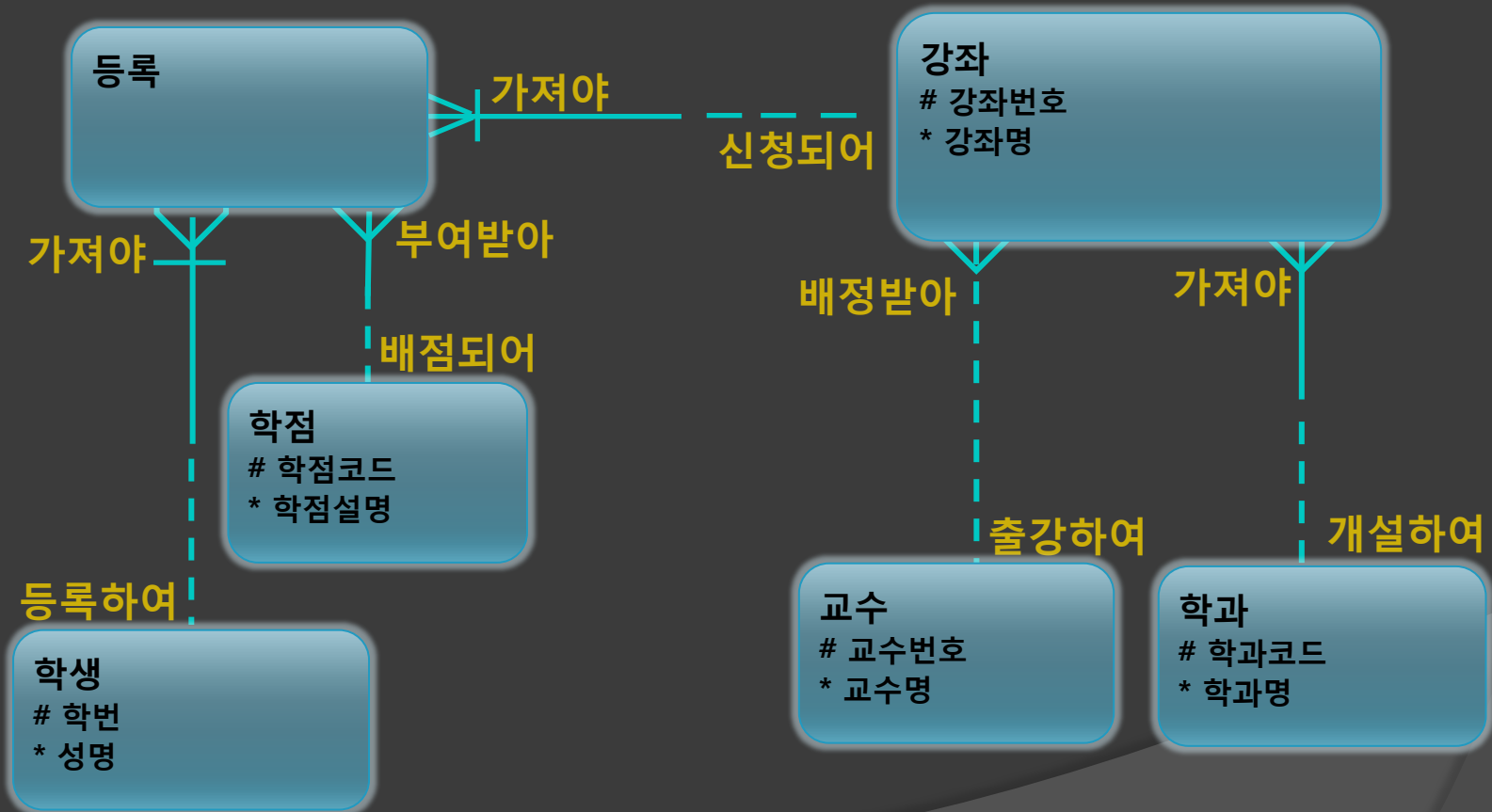
평가내역 → 평가코드 에도 종속



- 실습 : 제 3 정규형으로 정규화하기
  - 아래 모델을 제3 정규형에 맞게 다시 작성하시오.



## ■ 실습 : 제 3 정규형 (답안)



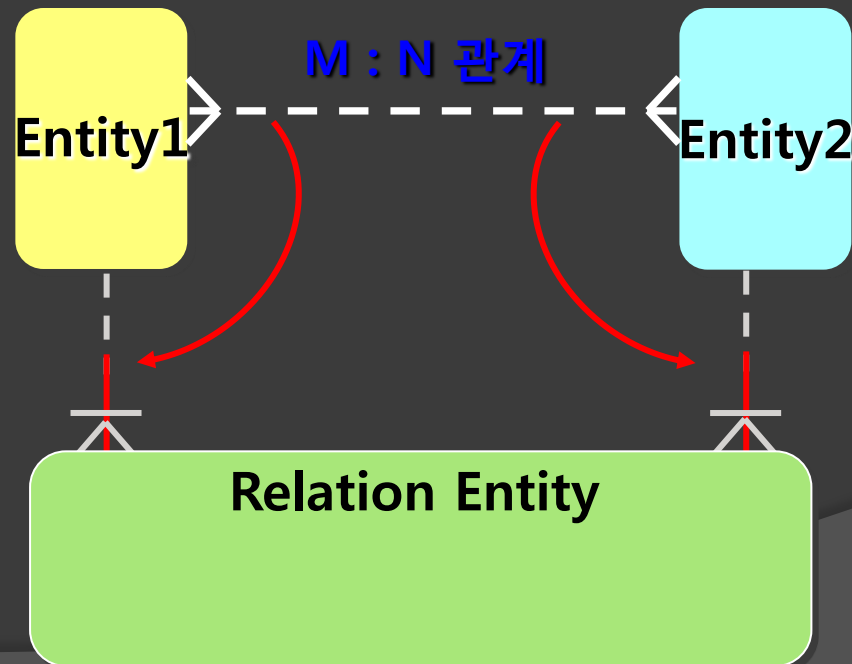
# 실전 데이터 모델링 2

## - M:N 관계

# M:N 관계

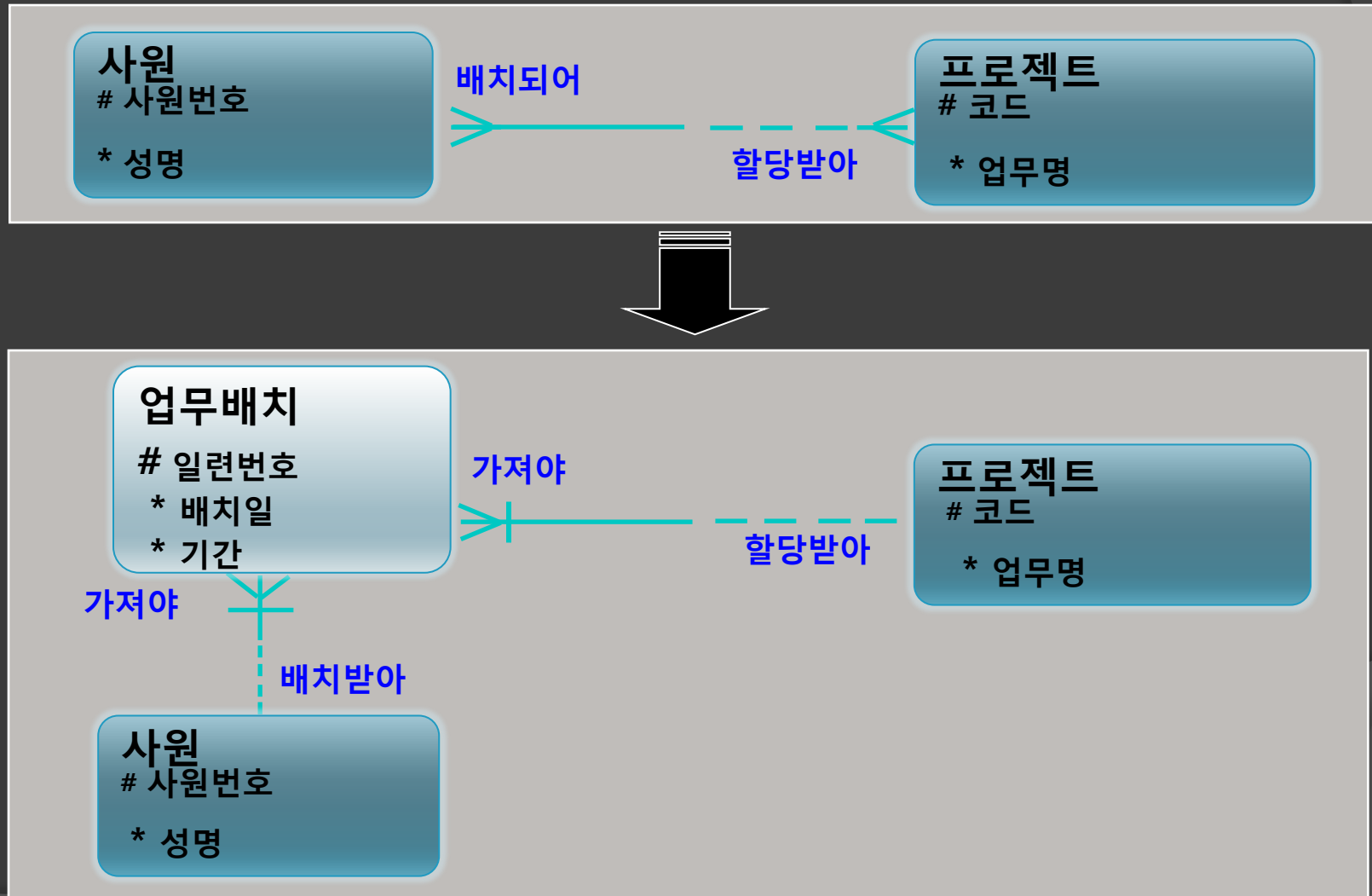
## ■ M:N 관계의 해결

- 실제 대부분의 업무/비즈니스는 M:N 관계
- 만약 어떤 속성이 관계를 나타내고 있다면 이 관계는 분해되어야 한다.
- M:N 관계는 새로운 엔티티를 추가하여 1:N 관계로 변경한다.



# M:N 관계

## ■ 예제1 : M:N 관계의 해소

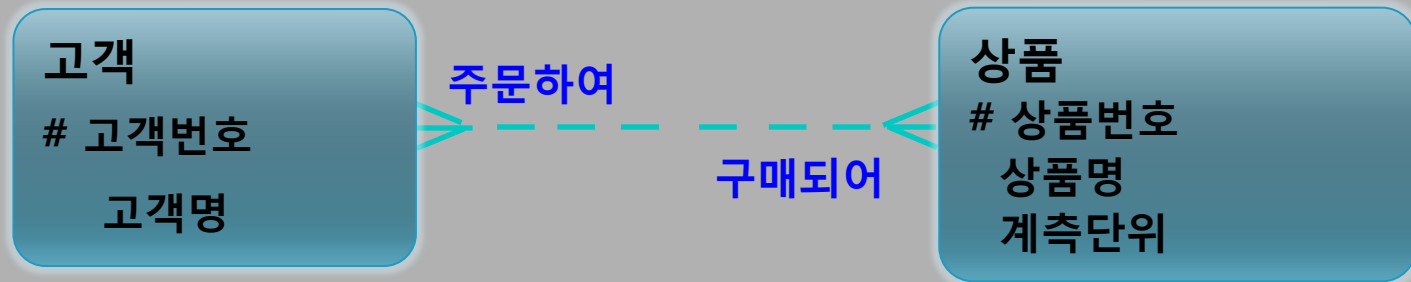




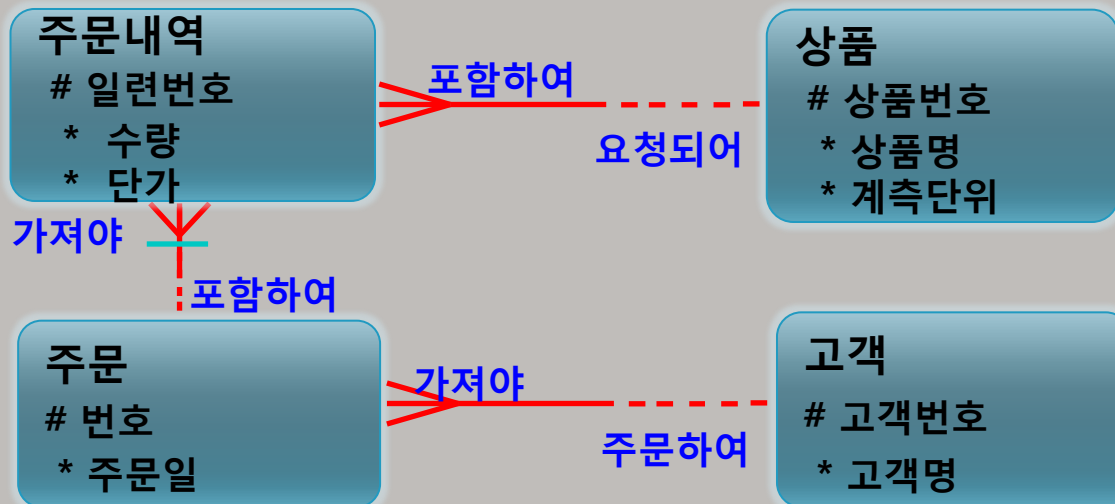
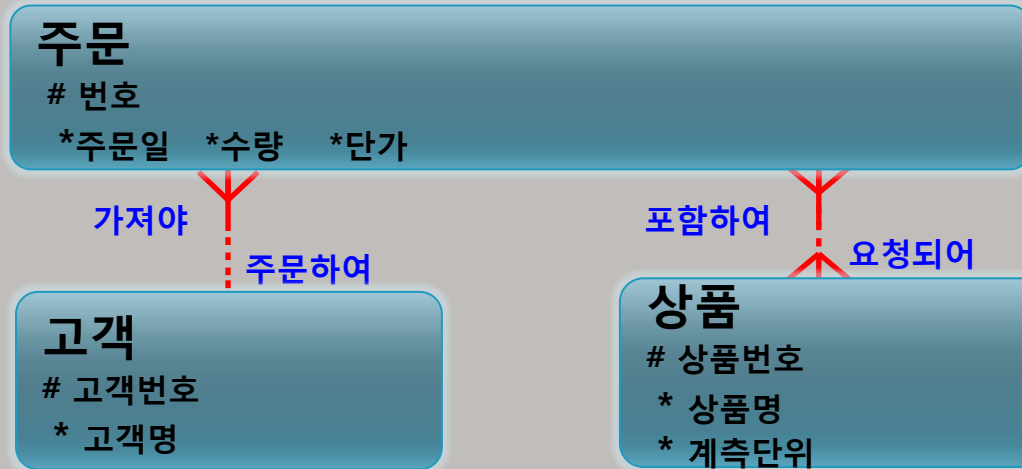
## M:N 관계

### ■ 실습 : M:N관계의 해소

1. 고객과 상품 엔티티 사이의 M:N관계를 분해하고, 속성으로 주문일, 주문수량, 주문가격을 추가하시오.
2. 이를 주문내역 이란 엔티티로 다시 분해하시오.



## ■ 실습 : M:N관계의 해소 (답안)



# 실전 데이터 모델링 3

## - 반정규화(역정규화)

# 반정규화(DeNormalization) 란?

정규화된 엔티티, 속성, 관계를  
시스템의 **성능향상, 개발과 운영의 단순화**를  
위해 데이터 모델을 통합하는 프로세스

- 데이터의 **정합성과 무결성**을 우선?
- 데이터베이스 구성의 **단순화와 성능**을 우선?

# 반정규화(역정규화) 절차

## ■ 반정규화(역정규화) 대상 조사

- 범위 처리 빈도수 조사
- 대량의 범위 처리 조사
- 통계성 프로세스 조사
- 테이블 조인개수

## ■ 다른 방법 유도 검토

- 뷰 테이블
- 클러스터링 또는 인덱스 적용
- 애플리케이션

## ■ 반정규화(역정규화) 적용

- 테이블 반정규화
- 속성의 반정규화
- 관계의 반정규화

# 테이블 반정규화(역정규화)

## ■ 테이블 병합

- 1:1 관계의 테이블 병합
- 1:M 관계의 테이블 병합
- 슈퍼타입 서브타입 테이블 병합

## ■ 테이블 분할

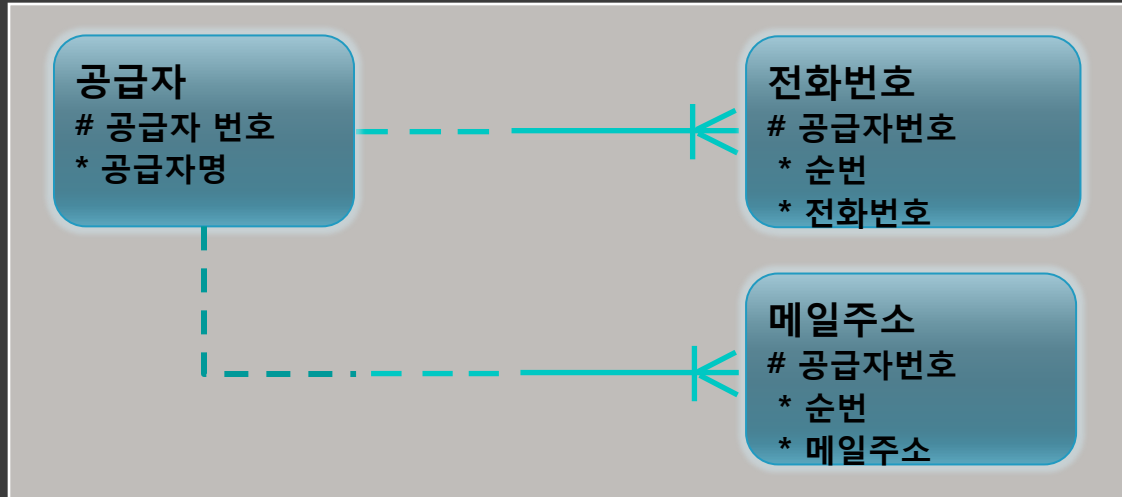
- 수직 분할
- 수평 분할

## ■ 테이블 추가

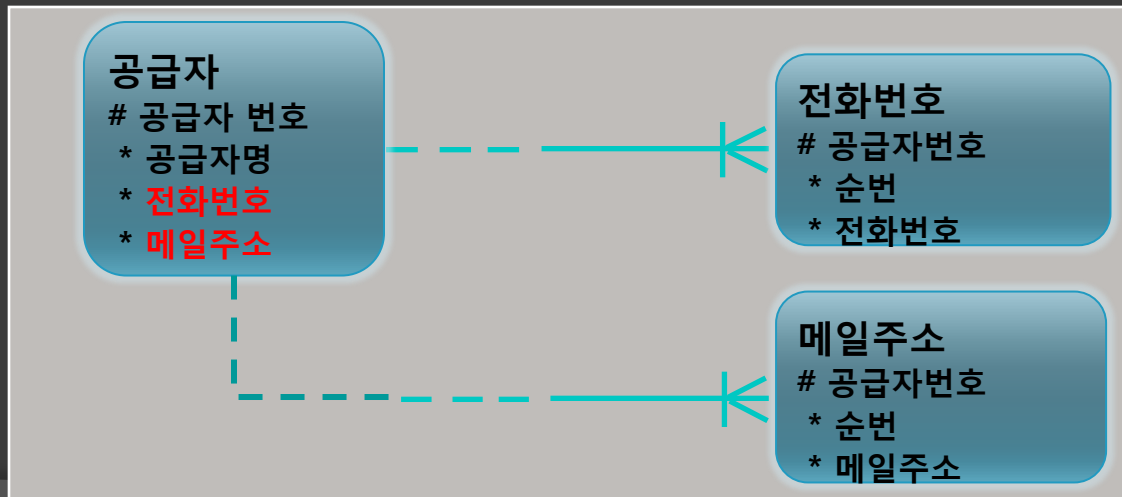
- 중복 테이블 추가
- 통계 테이블 추가
- 이력 테이블 추가
- 부분 테이블 추가

# 속성의 반정규화(역정규화)

## ■ 컬럼중복



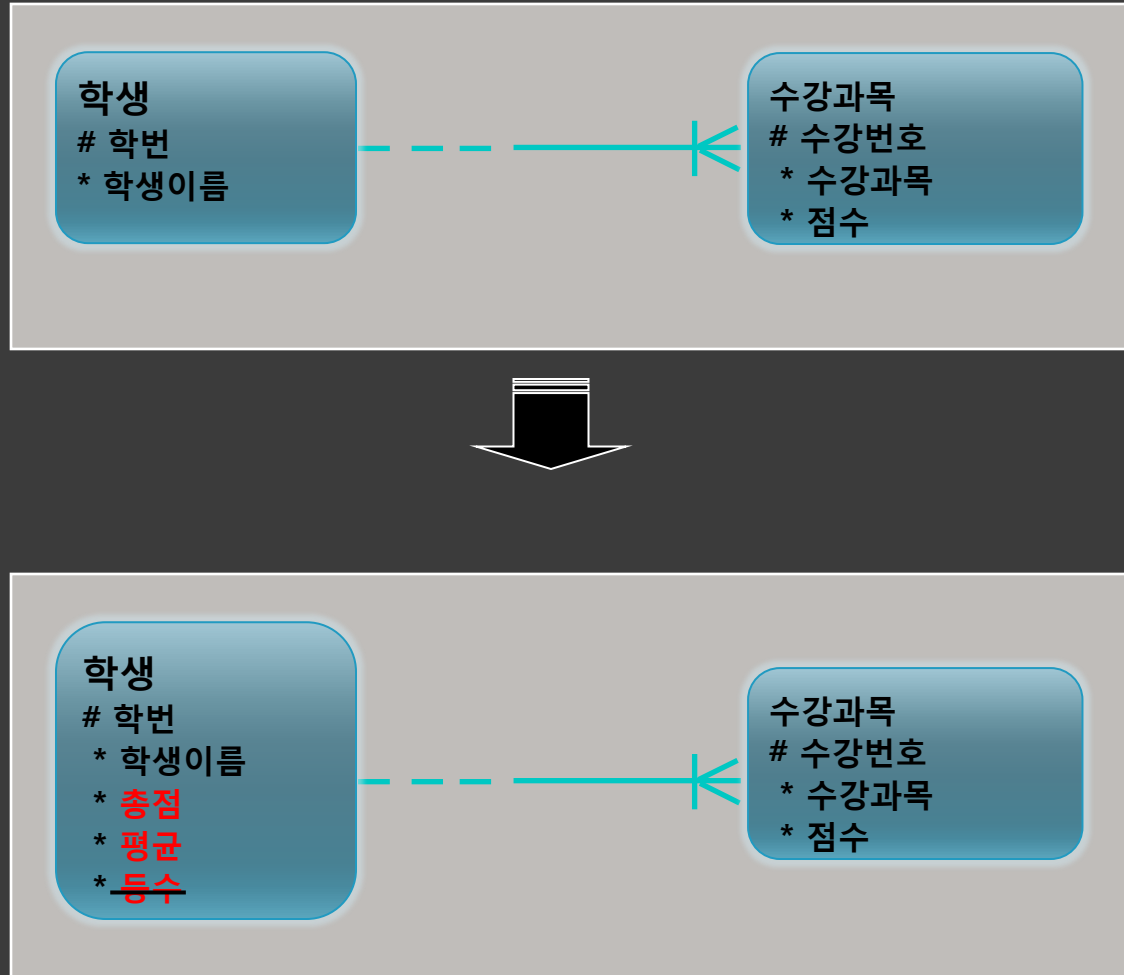
```
SELECT a.공급자명, b.전화번호, c.메일주소
FROM 공급자명 a,
(SELECT x.공급자번호, x.전화번호
FROM 전화번호 x,
(SELECT 공급자번호, MAX (순번) 순번
FROM 전화번호
WHERE 공급자번호 BETWEEN '1001' AND '1005'
GROUP BY 공급자번호) y
WHERE x.공급자번호 = y.공급자번호 AND x.순번 = y.순번) b,
(SELECT x.공급자번호, x.메일주소
FROM 메일주소 x,
(SELECT 공급자번호, MAX (순번) 순번
FROM 메일주소
WHERE 공급자번호 BETWEEN '1001' AND '1005'
GROUP BY 공급자번호) y
WHERE x.공급자번호 = y.공급자번호 AND x.순번 = y.순번) c
WHERE a.공급자번호 = b.공급자번호
AND a.공급자번호 = c.공급자번호
AND a.공급자번호 BETWEEN '1001' AND '1005'
```



```
SELECT 공급자명, 전화번호, 메일주소, 위치
FROM 공급자
WHERE 공급자번호 BETWEEN '1001' AND '1005'
```

# 속성의 반정규화(역정규화)

## ■ 파생컬럼의 생성



학번	총점	평균	순위
20130111	890	4.0	9
20130112	950	4.3	3

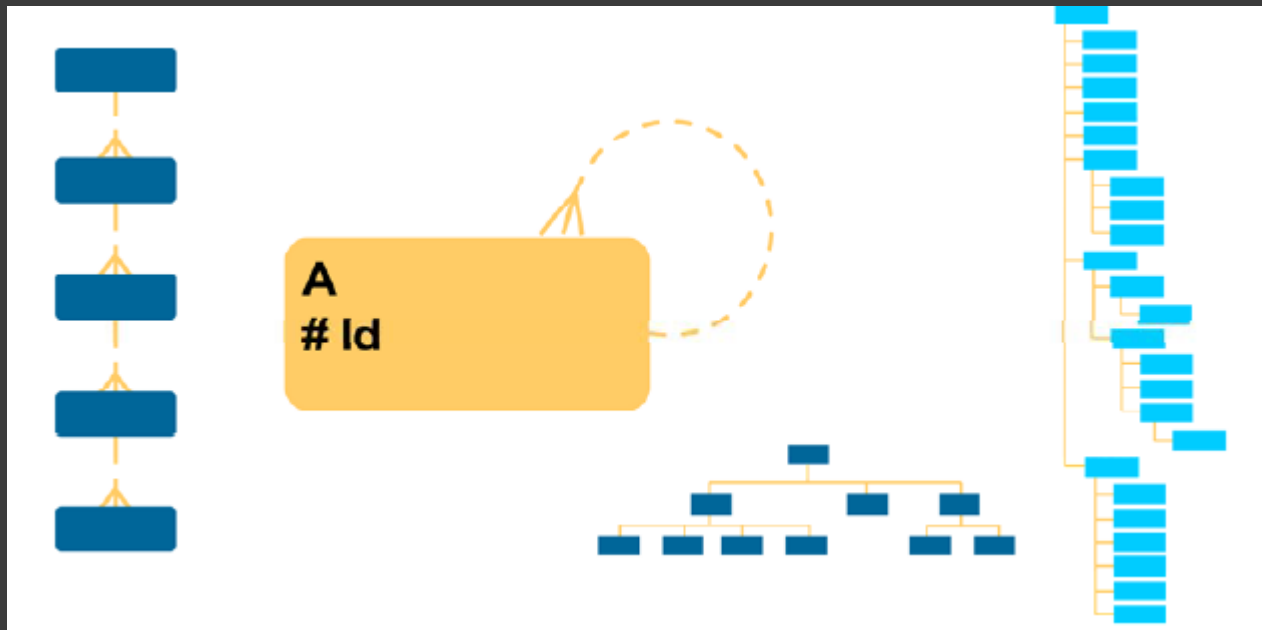


# 실전 데이터 모델링 4

## - 계층구조

## 계층구조(Parent – Child ) 란?

레벨을 가진 **종속관계형 데이터**들의 집합  
예) 카테고리, 분류, 메뉴 ..



# 계층구조 모델링

## ■ 순환 관계를 가진 구조

- 구조에 알려진 **레벨 제한이 없을 경우**
- 구조에 레벨 한계가 있지만 그 **한계가 높을 경우**
- 구조의 인스턴스가 **쉽게 위치를 변경할 수 있고** 그로 인해 레벨이 변경되는 경우
- 사용자가 제약 조건 유지를 선호할 경우

Id	Parent	Name
Id1	0	category1
id12	id1	category12
Id13	id1	category13
Id123	Id12	category123
Id124	id12	category124

# 계층구조 모델링

- 비순환 관계를 가진 구조
  - 구조의 레벨 한계가 낮을 경우
  - 상위 레벨 변경 빈도가 낮을 경우
  - 데이터가 많고, 성능개선이 우선될 경우



Id	Level1	Level2	Level3
id1	id1		
id12	id1	id12	
id13	id1	id13	
id123	id1	id2	id3
id124	id1	id2	id4

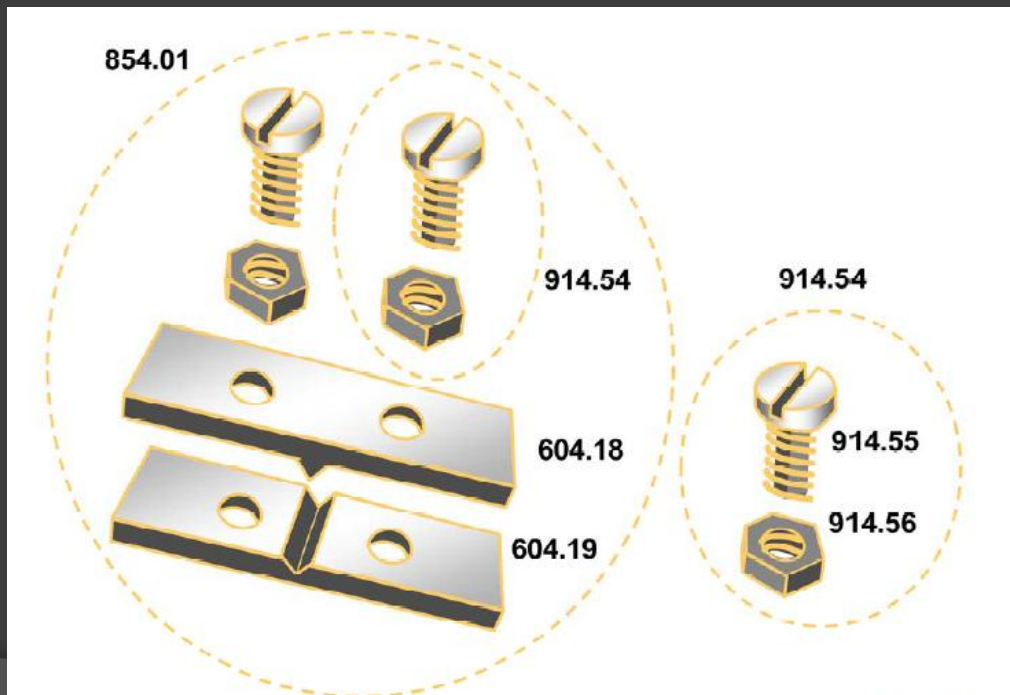
# 실전 데이터 모델링 5

## - 네트워크 구조

## 네트워크 구조( Network , BOM ) 란?

동일한 유형으로 된 사물의 쌍.

예) 결혼, 철로, 동의어 등..



### PRODUCT

Code	Name
914.53	AAAAA
914.54	AAA
914.55	BBBBB

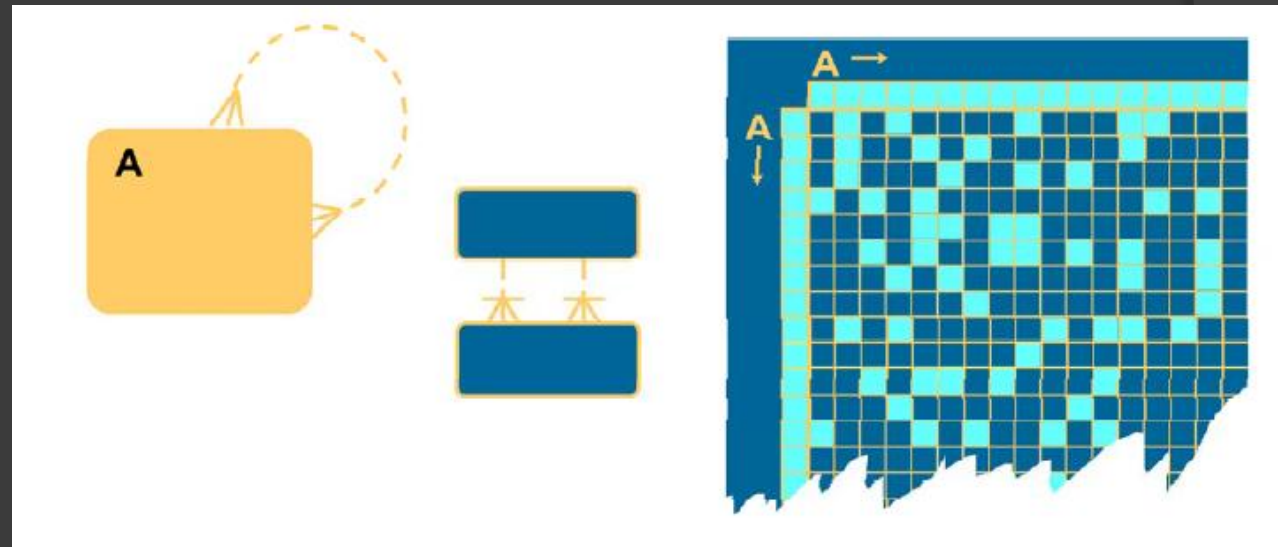
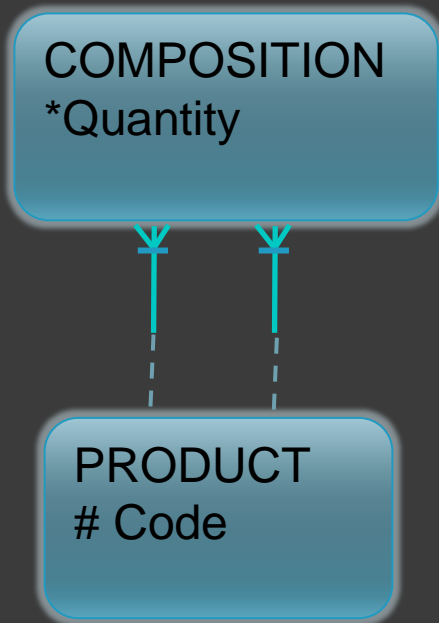
### COMPOSITION

Prod_code	Part_code	Quntity
854.01	604.18	1
854.01	604.19	1
854.01	914.54	2
914.54	914.55	1
914.54	914.56	1

# 네트워크 구조 모델링

## ■ 특징

- m:m 관계는 쌍에 대한 특정 정보를 보유
- 교차 엔티티의 두 관계는 고유 식별자를 구성
- 두 개의 관계는 상이한 Sub-type 의 엔티티를 참조



# 과제

1. 다음회원이면 누구나 카페를 개설 할 수 있고, 카페에 가입 할 수 있다.
2. 카페 개설시 카페명, 카페주소, 소속카테고리, 공개여부, 소개글을 반드시 입력해야 하며 개설일 및 폐쇄일을 별도로 관리하고 있어야 한다. 한번 개설한 카페는 중복 개설할 수 없으며 카페주소로 그 여부를 판단한다.
3. 카페 개설자는 카페를 꾸미기 위해 여러 배경화면 중 한가지를 선택할 수 있다. 배경음악은 음악샵에서 무료로 선택하여 목록으로 관리하고 원하는 곡을 카페 방문자들이 들을 수 있도록 일부만 선택할 수 있다.
4. 카페 회원은 개설자, 운영진, 정회원, 준회원으로 구분하여 관리하며 회원들은 카페 가입시 별도의 별명을 만들어 활동을 할 수 있다.
5. 카페 회원의 가입일 및 탈퇴일을 별도로 관리해야 하며 이미 탈퇴했던 회원이라도 다시 가입이 가능하다. 단 과거 가입기록은 계속 남아 있어야 하며 탈퇴 후 최소 10일 이후에 가입할 수 있다.
6. 카페 개설자는 게시판 생성할 수 있으며 게시판 생성시 게시판명을 정하고 읽기/쓰기 권한을 준회원이상, 정회원이상, 운영진이상 공개 권한을 설정할 수 있다.
7. 게시판 종류는 일반게시판과 사진게시판으로 나뉘며 파일을 여러개 첨부할 수 있고 첨부파일의 크기, 종류, 파일명을 저장한다. 각 게시글별로 댓글을 달 수 있으며 배경음악은 한개만 정할 수 있다.
8. 카페 개설자가 카페끼리 친구 신청을 할 수 있으며, 카페 개설자는 다른 카페로부터의 신청건에 대해 수락 및 거절을 할 수 있고, 카페 회원들에게는 친구로 등록된 카페들을 보여준다.



# Q & A