

# 데이터 모델링 (Data Modeling)

라 가 영

# POINT 1. 데이터 모델링의 이해

- 데이터 모델링

- 현실세계를 데이터베이스로 표현하기 위해
- 주어진 개념으로부터 논리적인 데이터 모델을 구성하는 작업
- 고객의 요구를 데이터베이스에 반영하는 작업

## 데이터모델링이란...

- 정보시스템을 구축하기 위한 데이터관점의 업무 분석 기법
- 현실세계의 데이터(what)에 대해 약속된 표기법에 의해 표현하는 과정
- 데이터베이스를 구축하기 위한 분석/설계의 과정

# POINT 1. 데이터 모델링의 이해

## 데이터 모델링의 주요 특징

특징	설명
추상화(Abstraction)	현실세계를 간략하게 표현
단순화(Simple)	누구나 쉽게 이해할 수 있도록 표현
명확성(Clarity)	명확하게 의미가 해석되어야 하고 한 가지 의미를 가져야 함



복잡한 현실세계를 일정한 표기법에 의해 표현하는 일

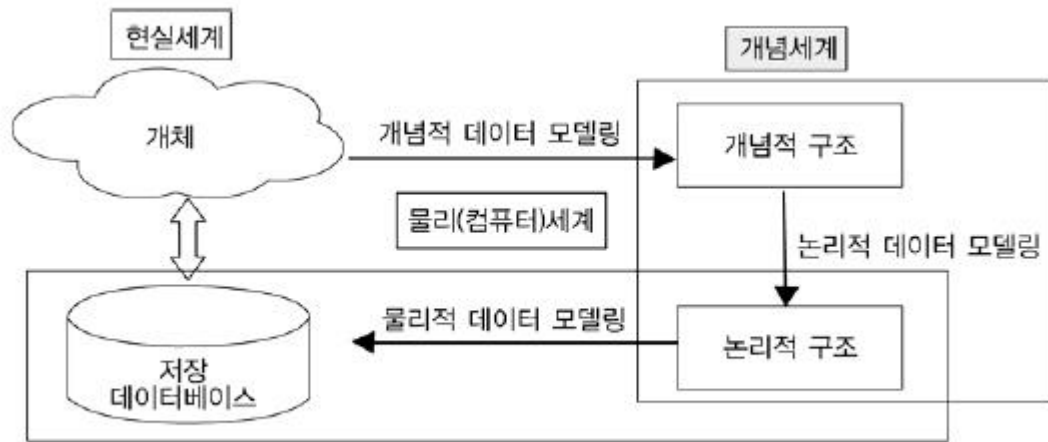
[그림 1-1-1] 모델링의 정의

# POINT 1. 데이터 모델링의 이해

## 데이터 모델링 단계

데이터 모델링 단계	설명
개념적 모델링	<ul style="list-style-type: none"><li>• <b>전사적 관점</b>에서 기업의 데이터를 모델링</li><li>• <b>추상화 수준이 가장 높은 수준으로 모델링</b></li><li>• 계층형 데이터 모델, 네트워크 모델, 관계형 모델에 관계없이 <b>업무적 측면에서 모델링</b></li><li>• 엔티티(Entity)와 속성(Attribute)을 도출하고 개념적 ERD(Entity Relationship Diagram)를 작성</li></ul>
논리적 모델링	<ul style="list-style-type: none"><li>• 특정 데이터베이스 모델에 종속</li><li>• 식별자를 정의하고 관계, 속성 등을 모두 표현</li><li>• 정규화를 통해서 <b>재사용성을 높임</b></li></ul>
물리적 모델링	<ul style="list-style-type: none"><li>• 구축할 데이터베이스 관리 시스템에 테이블, 인덱스 등을 생성하는 단계</li><li>• 성능, 보안, 가용성 등을 고려하여 데이터베이스를 구축</li></ul>

# POINT 1. 데이터 모델링의 이해



[그림 1-1-3] 현실세계와 데이터베이스 사이의 모델

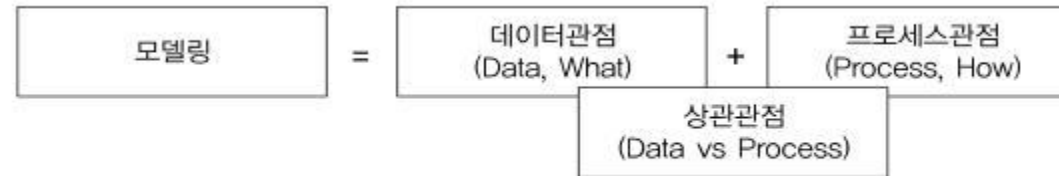
[표 1-1-1] 개념-논리-물리데이터 모델

데이터 모델링	내용	수준
개념적 데이터 모델링	추상화 수준이 높고 업무중심적이고 포괄적인 수준의 모델링 진행. 전사적 데이터 모델링, EA수립시 많이 이용	추상적
논리적 데이터 모델링	시스템으로 구축하고자 하는 업무에 대해 Key, 속성, 관계 등을 정확하게 표현, 재사용성이 높음	
물리적 데이터 모델링	실제로 데이터베이스에 이식할 수 있도록 성능, 저장 등 물리적인 성격을 고려하여 설계	구체적

# POINT 1. 데이터 모델링의 이해

## 데이터 모델링 관점

관점(View)	설명
데이터	<ul style="list-style-type: none"><li>• 비즈니스 프로세스에 사용되는 데이터를 의미</li><li>• 구조분석, 정적분석</li></ul>
프로세스	<ul style="list-style-type: none"><li>• 비즈니스 프로세스에 수행하는 작업을 의미</li><li>• 시나리오 분석, 도메인 분석, 동적 분석</li></ul>
데이터와 프로세스	<ul style="list-style-type: none"><li>• 프로세스와 데이터 간의 관계를 의미</li><li>• CRUD(Create Read Update Delete) 분석</li></ul>



[그림 1-1-2] 모델링의 관점

# POINT 1. 데이터 모델링의 이해

- 데이터 모델링을 위한 **ERD**  
(Entity Relationship Diagram)
- 엔터티(Entity)와 엔터티 간의 관계를 정의하는 모델링 방법

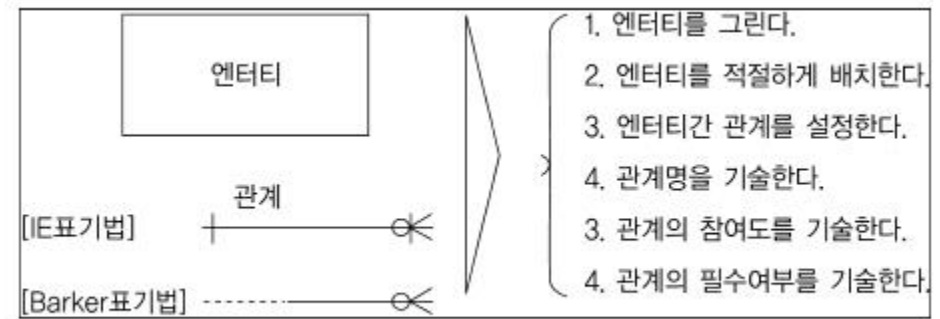
[표 1-1-6] 표기법

표기법	설명
<p>Chen</p>	<ul style="list-style-type: none"> <li>- 대학교재에서 가장 많이 이용하는 표기법</li> <li>- 실무적으로 사용안함</li> </ul>
<p>IDEF1X</p>	<ul style="list-style-type: none"> <li>- 마름모와 원을 이용한 표기법으로 실무현장에서는 소수 활용</li> <li>- ERWin</li> </ul>
<p>IE/Crow's Foot</p>	<ul style="list-style-type: none"> <li>- 까마귀발 모양의 표기법으로 가장 많이 사용함</li> <li>- ERWin, ERStudio</li> </ul>
<p>Min-Max/ISO</p>	<ul style="list-style-type: none"> <li>- 기수성을 좀 더 정교하게 표현한 방법으로 많이 활용안됨</li> </ul>
<p>UML</p>	<ul style="list-style-type: none"> <li>- 스테레오타입을 이용하여 엔터티 표현</li> <li>- UML로 표현하여 데이터 모델링 할 때 사용</li> <li>- Rational Rose</li> </ul>
<p>Case*Method/Barker</p>	<ul style="list-style-type: none"> <li>- Crow's Foot을 적용하면서 관계 표기법 등 일부 다름(Barker's Notation)</li> <li>- DA#</li> </ul>

# POINT 1. 데이터 모델링의 이해

## • ERD 작성 절차

- ① 엔터티를 도출하고 그린다.
- ② 엔터티를 배치한다.
- ③ 엔터티 간의 관계를 설정한다.
- ④ 관계명을 서술한다.
- ⑤ 관계 참여도를 표현한다.
- ⑥ 관계의 필수 여부를 표현한다.



[그림 1-1-9] ERD 작업순서



## POINT 1. 데이터 모델링의 이해

- 엔터티를 도출하고 그린다.
- 엔터티를 배치한다.



[그림 1-1-10] 엔터티 배치방법

## POINT 1. 데이터 모델링의 이해

- 엔터티 간의 관계를 설정한다.



[그림 1-1-11] ERD 관계설정

## POINT 1. 데이터 모델링의 이해

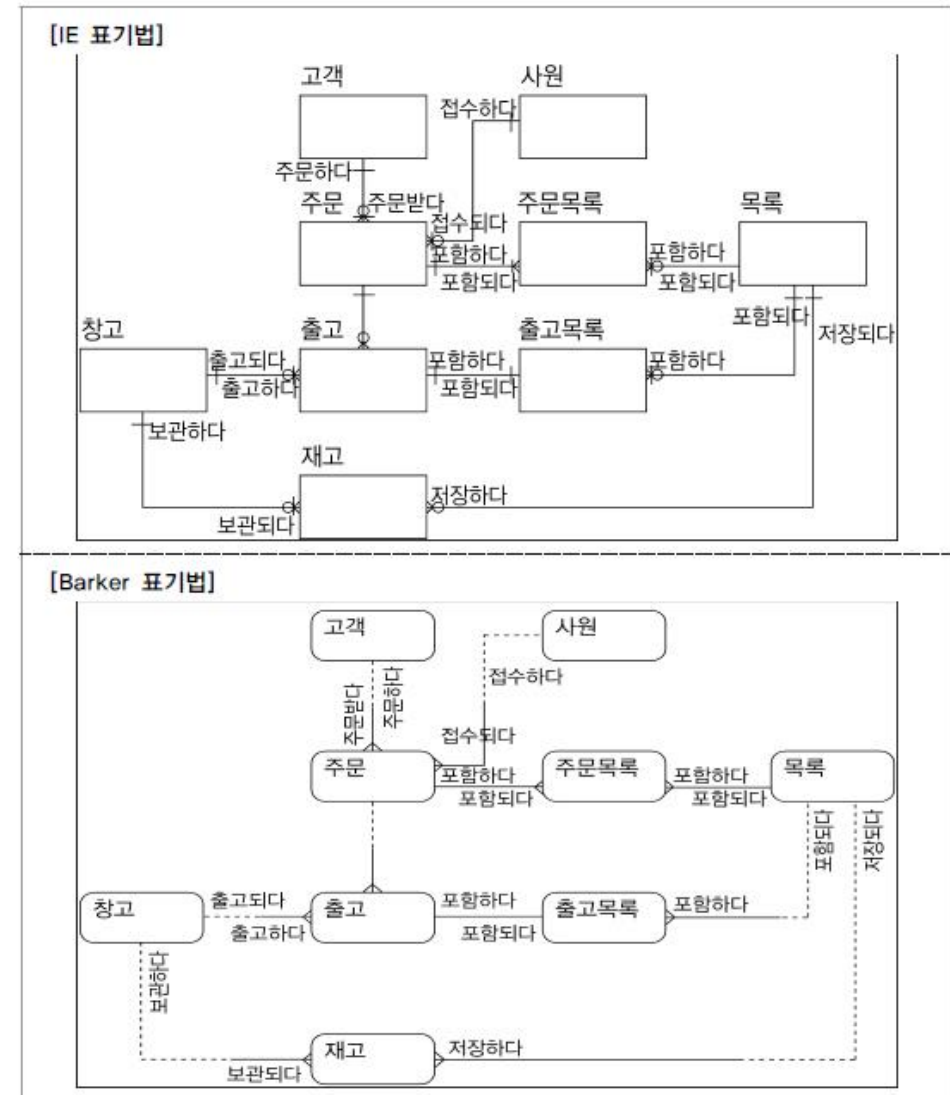
- 관계명을 설정한다.



[그림 1-1-12] ERD 관계명 표시

# POINT 1. 데이터 모델링의 이해

- 관계 참여도를 표현한다.
- 관계의 필수 여부를 표현한다.



[그림 1-1-13] 관계차수와 선택성 표시

# POINT 1. 데이터 모델링의 이해

- 데이터 모델링 고려사항

- 1) 데이터 모델의 독립성

- 독립성이 확보된 모델은 업무 변화에 능동적으로 대응 가능
- 독립성을 확보하기 위해서는 중복된 데이터를 제거해야 함 -> 정규화

- 2) 고객 요구사항의 표현

- 너무 복잡하지 않게 표현해야 함
- 간결하고 명확하게 표현해야 함

- 3) 데이터 품질 확보

- 데이터 표준을 정의하고 표준 준수율을 관리해야 함 -> 데이터 품질 향상

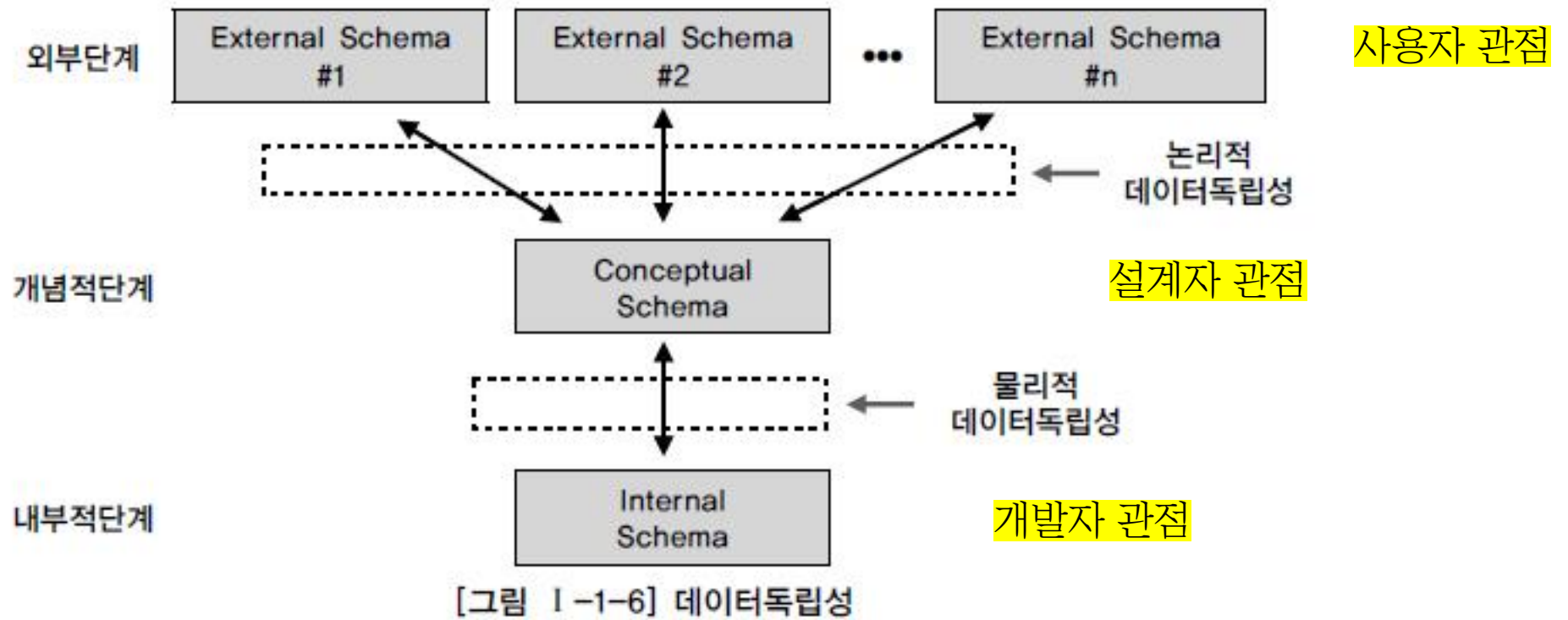
## POINT 2. 3층 스키마(3-Level Schema)

- 3층 스키마
  - 사용자, 설계자, 개발자가 데이터베이스를 보는 관점에 따라 데이터베이스를 기술하고 이들 간의 관계를 정의한 ANSI 표준
  - 독립성을 확보하기 위한 방법
- 데이터의 복잡도 증가, 데이터의 중복 제거, 사용자의 요구사항 변경에 따른 대응력 향상, 관리 및 유지보수 비용 절감 등의 장점

## POINT 2. 3층 스키마(3-Level Schema)

3층 스키마의 독립성	
독립성	설명
논리적 독립성	저장구조가 변경되어도 응용 프로그램 및 개념 스키마에 영향이 없음
물리적 독립성	데이터베이스 논리적 구조가 변경되어도 응용 프로그램에 변화가 없음

## POINT 2. 3층 스키마(3-Level Schema)





## POINT 2. 3층 스키마(3-Level Schema)

[표 1-1-2] 데이터독립성 구성요소

항목	내용	비고
외부스키마 (External Schema)	<ul style="list-style-type: none"><li>- View 단계 여러 개의 사용자 관점으로 구성, 즉 개개 사용자 단계로서 개개 사용자가 보는 개인적 DB 스키마</li><li>- DB의 개개 사용자나 응용프로그래머가 접근하는 DB 정의</li></ul>	사용자 관점 접근하는 특성에 따른 스키마 구성
개념스키마 (Conceptual Schema)	<ul style="list-style-type: none"><li>- 개념단계 하나의 개념적 스키마로 구성 모든 사용자 관점을 통합한 조직 전체의 DB를 기술하는 것</li><li>- 모든 응용시스템들이나 사용자들이 필요로 하는 데이터를 통합한 조직 전체의 DB를 기술한 것으로 DB에 저장되는 데이터와 그들간의 관계를 표현하는 스키마</li></ul>	통합관점
내부스키마 (Internal Schema)	<ul style="list-style-type: none"><li>- 내부단계, 내부 스키마로 구성, DB가 물리적으로 저장된 형식</li><li>- 물리적 장치에서 데이터가 실제로 저장되는 방법을 표현하는 스키마</li></ul>	물리적 저장구조

## POINT 2. 3층 스키마(3-Level Schema)

[표 1-1-3] 논리적, 물리적 데이터독립성

독립성	내용	특징
논리적 독립성	<ul style="list-style-type: none"><li>- 개념 스키마가 변경되어도 외부 스키마에는 영향을 미치지 않도록 지원하는 것</li><li>- 논리적 구조가 변경되어도 응용 프로그램에 영향 없음</li></ul>	<ul style="list-style-type: none"><li>- 사용자 특성에 맞는 변경가능</li><li>- 통합 구조 변경가능</li></ul>
물리적 독립성	<ul style="list-style-type: none"><li>- 내부스키마가 변경되어도 외부/개념 스키마는 영향을 받지 않도록 지원하는 것</li><li>- 저장장치의 구조변경은 응용프로그램과 개념스키마에 영향 없음</li></ul>	<ul style="list-style-type: none"><li>- 물리적 구조 영향 없이 개념구조 변경가능</li><li>- 개념구조 영향 없이 물리적인 구조 변경가능</li></ul>

## POINT 2. 3층 스키마(3-Level Schema)

[표 1-1-4] 사상(Mapping)

사상	내용	예
외부적/개념적 사상 (논리적 사상)	- 외부적 뷰와 개념적 뷰의 상호 관련성을 정의함	사용자가 접근하는 형식에 따라 다른 타입의 필드를 가질 수 있음. 개념적 뷰의 필드 타입은 변화가 없음
개념적/내부적 사상 (물리적 사상)	- 개념적 뷰와 저장된 데이터베이스의 상호관련성 정의	만약 저장된 데이터베이스 구조가 바뀐다면 개념적/내부적 사상이 바뀌어야 함. 그래야 개념적 스키마가 그대로 남아있게 됨

## POINT 3. 엔터티(Entity) 도출

- 엔터티의 개념
  - 실체, 객체
  - 변별할 수 있는 사물 - Peter Chen(1976)
  - 데이터베이스 내에서 변별 가능한 객체 - C.J Date(1986)
  - 정보를 저장할 수 있는 어떤 것 - James Martin(1989)
  - 정보가 저장될 수 있는 사람, 장소, 물건, 사건 그리고 개념 등 - Thomas Bruce(1992)

## POINT 3. 엔터티(Entity) 도출

- 엔터티의 개념

- 사람, 장소, 물건, 사건, 개념 등의 명사에 해당
- 업무상 관리가 필요한 관심사
- 저장이 되기 위한 어떤 것(Thing)



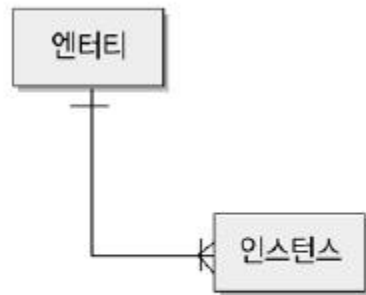
업무에 필요하고 유용한 정보를 저장하고 관리하기 위한 집합적인 것(thing)

[그림 1-1-14] 엔터티 종류

## POINT 3. 엔터티(Entity) 도출

- 엔터티와 인스턴스에 대한 내용과 표기법

엔터티-인스턴스 ERD

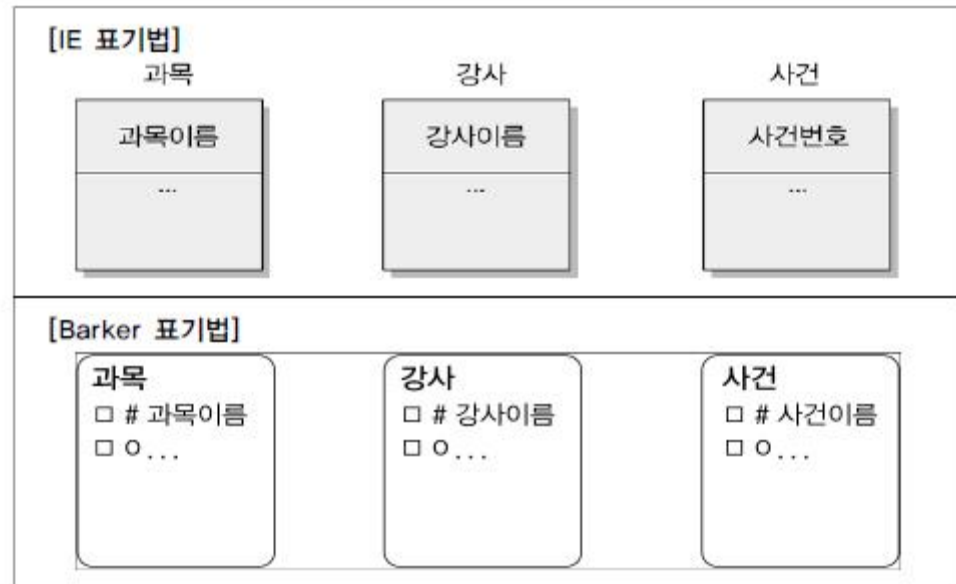


엔터티는 인스턴스의 집합

엔터티-인스턴스의 예

엔터티	인스턴스
과 목	수 학
	영 어
강 사	이춘식
	조시형
사 건	2010-001
	2010-002

[그림 1-1-15] 엔터티와 인스턴스



[그림 1-1-16] 엔터티에 대한 표기법

## POINT 3. 엔터티(Entity) 도출

- 엔터티의 특징

- 반드시 해당 업무에서 필요하고 관리하고자 하는 정보여야 함  
(ex. 환자, 토익 응시횟수, ...)
- 유일한 식별자에 의해 식별이 가능해야 함
- 영속적으로 존재하는 인스턴스의 집합이어야 함  
(한 개 x, 두 개 이상 o)
- 업무 프로세스에 의해 이용되어야 함
- 반드시 속성이 있어야 함
- 다른 엔터티와 최소 한 개 이상의 관계가 있어야 함

## POINT 3. 엔터티(Entity) 도출

- 업무에서 필요로 하는 정보





## POINT 3. 엔터티(Entity) 도출

- 식별이 가능해야 함



모두다 동일한 이름, 속성 관계

인스턴스 각각을 구분하기 위한 유일한 식별자가 존재해야 함

[그림 1-1-18] 엔터티 특성-유일성

## POINT 3. 엔터티(Entity) 도출

- 인스턴스의 집합

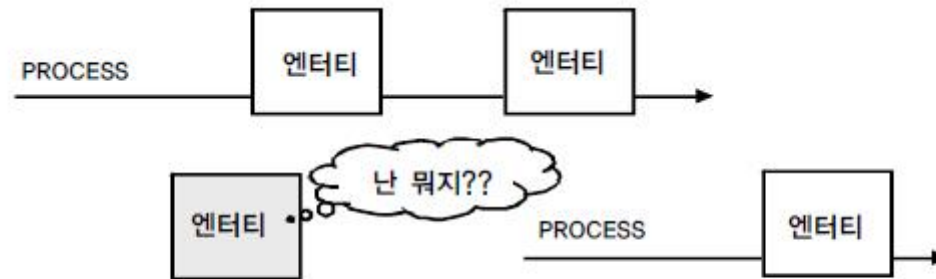


인스턴스가 한 개 밖에 없는 회사, 병원 엔터티는 집합이 아니므로 엔터티 성립이 안됨

[그림 1-1-19] 엔터티 특성-인스턴스 수

## POINT 3. 엔터티(Entity) 도출

- 업무프로세스에 의해 이용



업무 프로세스에 의해 이용되지 않는 엔터티는 그 업무의 엔터티가 아님

[그림 1-1-20] 엔터티 특성-프로세스 존재

## POINT 3. 엔터티(Entity) 도출

- 속성을 포함

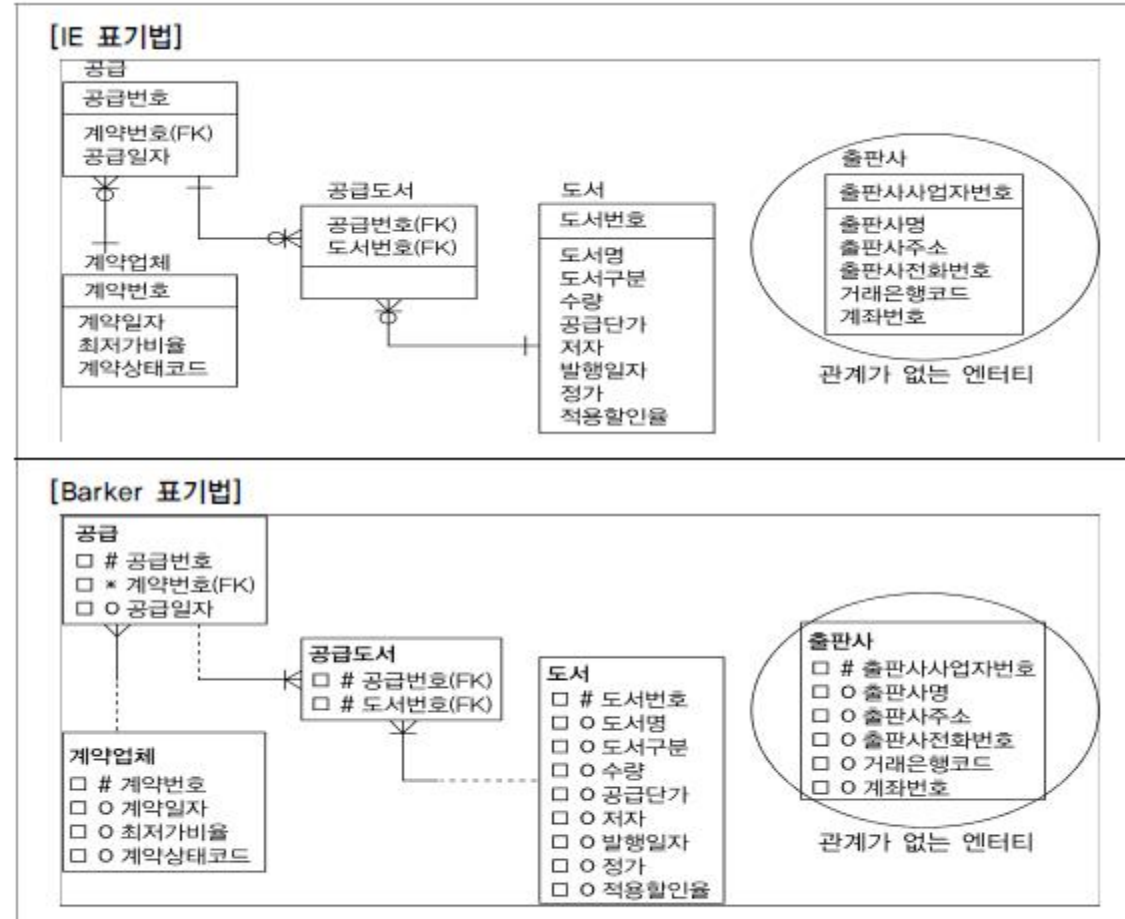


속성이 존재하지 않는 오브젝트는 엔터티가 될 수 없음

[그림 1-1-21] 엔터티 특성-속성의 존재

## POINT 3. 엔터티(Entity) 도출

- 관계의 존재



엔터티가 관계가 없으면, 잘못된 엔터티이거나 관계가 누락되었을 가능성이 큼

[그림 1-1-22] 엔터티 특성-관계의 존재

## POINT 4. 속성(Attribute)

- 사물의 성질, 특징 또는 본질적인 성질
- 업무에서 필요로 하는 인스턴스로 관리하고자 하는 의미상 더 이상 분리되지 않는 **최소의 데이터 단위**
- 엔터티의 한 분야



강사

엔터티는 속성들에 의해 설명된다.

속성들

- 이름
- 주소
- 생년월일
- 계약일자
- 전문분야

속성은 업무에서 필요로 하는 인스턴스에서 관리하고자 하는  
의미상 더 이상 분리되지 않는 최소의 데이터 단위

[그림 1-1-24] 속성의 정의

## POINT 4. 속성(Attribute)

- 속성의 정의
  - 업무에서 필요로 함
  - 의미상 더 이상 분리되지 않음
  - 엔티티를 설명하고 인스턴스의 구성요소가 됨

## POINT 4. 속성(Attribute)

- 속성의 특징
  - 업무에서 관리되는 정보
  - 하나의 값만 가짐
  - 주 식별자에게 함수적으로 종속됨 -> 기본키가 변경되면 속성값도 변경됨



## POINT 4. 속성(Attribute)

분해여부에 따른 속성의 종류	
종류	설명
단일 속성	<ul style="list-style-type: none"><li>• 하나의 의미로 구성된 것</li><li>• 회원 ID, 이름 등</li></ul>
복합 속성	<ul style="list-style-type: none"><li>• 여러 개의 의미가 있는 것으로 대표적으로 주소</li><li>• 시, 군, 동 등으로 분해 가능</li></ul>
다중값 속성	<ul style="list-style-type: none"><li>• 여러 개의 값을 가질 수 있는 것</li><li>• 상품 리스트 -&gt; 엔터티로 분해</li></ul>

## POINT 4. 속성(Attribute)

### 특성에 따른 속성의 종류

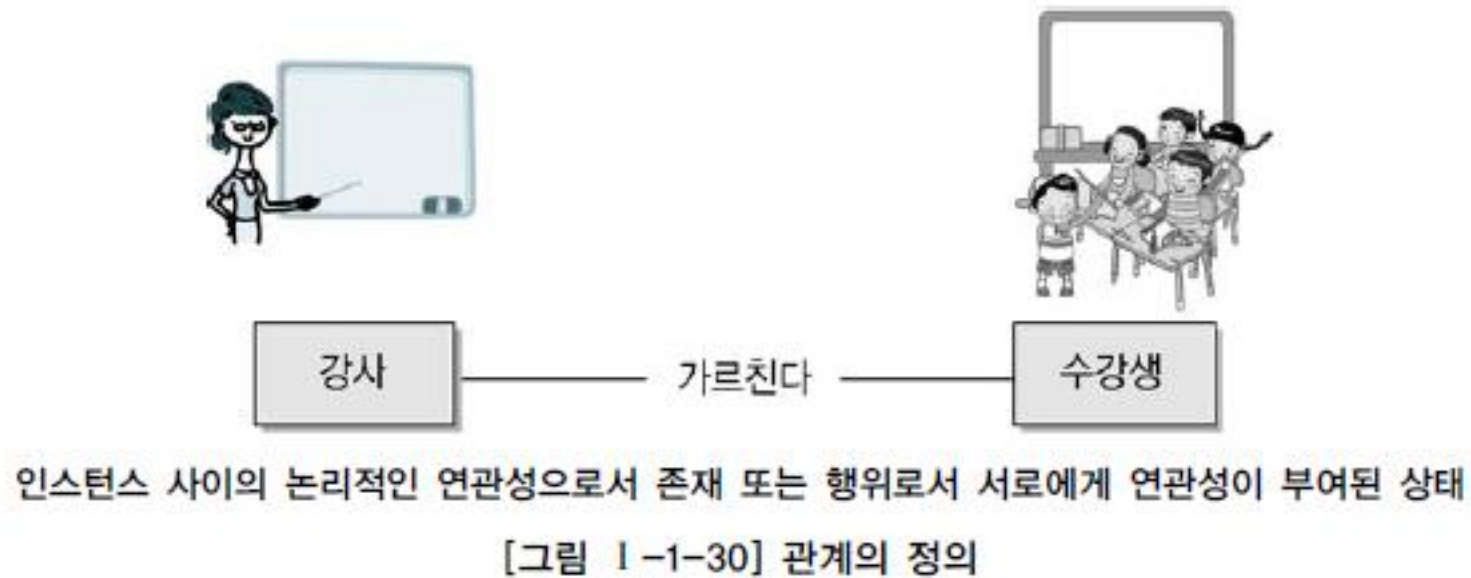
종류	설명
기본 속성	<ul style="list-style-type: none"><li>• 비즈니스 프로세스에서 도출되는 <b>본래의 속성</b></li><li>• 회원 ID, 이름, 계좌번호, 주문 일자 등</li></ul>
설계 속성	<ul style="list-style-type: none"><li>• <b>데이터 모델링 과정에서 발생하는 속성</b></li><li>• <b>유일한 값을 부여</b></li><li>• 상품코드, 지점 코드 등</li></ul>
파생 속성	<ul style="list-style-type: none"><li>• <b>다른 속성에 의해서 만들어지는 속성</b></li><li>• 합계, 평균 등</li></ul>

#### TIP. 도메인(Domain)

**속성이 가질 수 있는 값의 범위**  
성별이라는 속성의 도메인 값은 남성, 여성

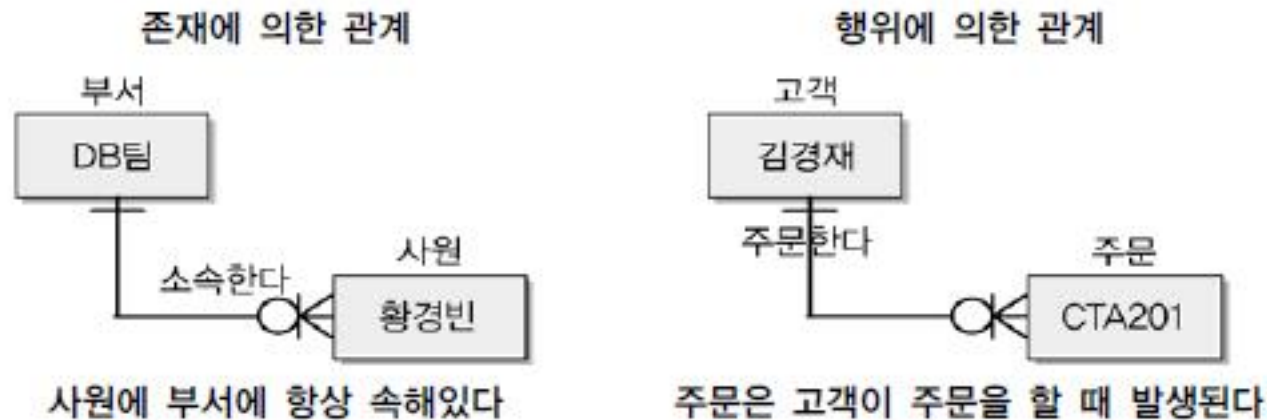
## POINT 5. 관계(Relationship)

- 상호 **연관성**이 있는 상태
- 엔터티와 인스턴스 사이의 논리적인 연관성으로서 존재의 형태 혹은 행위로서 서로에게 연관성이 부여된 상태



## POINT 5. 관계(Relationship)

- 관계의 종류 -> 어떤 **목적**으로 연결되었는지



[그림 1-1-32] 관계의 분류

## POINT 5. 관계(Relationship)

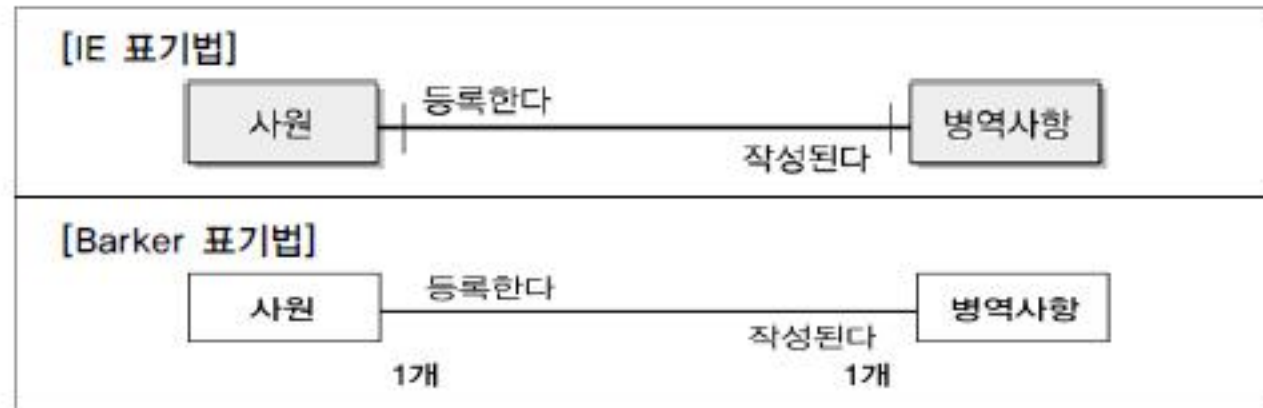
- 관계명(Membership) -> 관계에 참여하는 형태



[그림 1-1-33] 관계의 관계명

## POINT 5. 관계(Relationship)

- 관계차수(Degree/Cardianality) -> 참여자 수를 표현하는 것
  - 1:1(ONE TO ONE) 관계를 표시



[그림 1-1-34] 관계차수(1:1)

## POINT 5. 관계(Relationship)

- 관계차수(Degree/Cardianality) -> 참여자 수를 표현하는 것
  - 1:M(ONE TO MANY) 관계를 표시

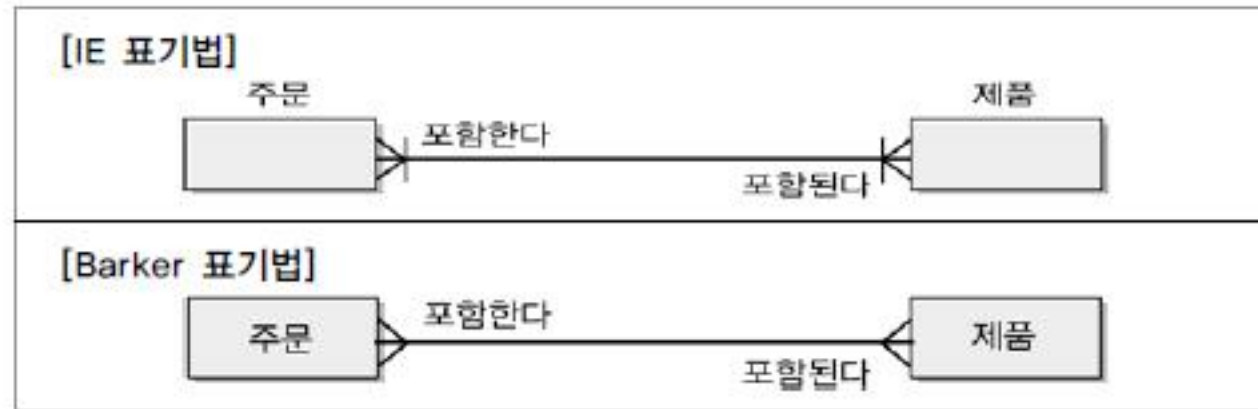


한 명의 사원은 한 부서에 소속되고 한 부서에는 여러 사원을 포함한다

[그림 1-1-35] 관계차수(1:M)

## POINT 5. 관계(Relationship)

- 관계차수(Degree/Cardianality) -> 참여자 수를 표현하는 것
  - **M:M**(MANY TO MANY) 관계를 표시

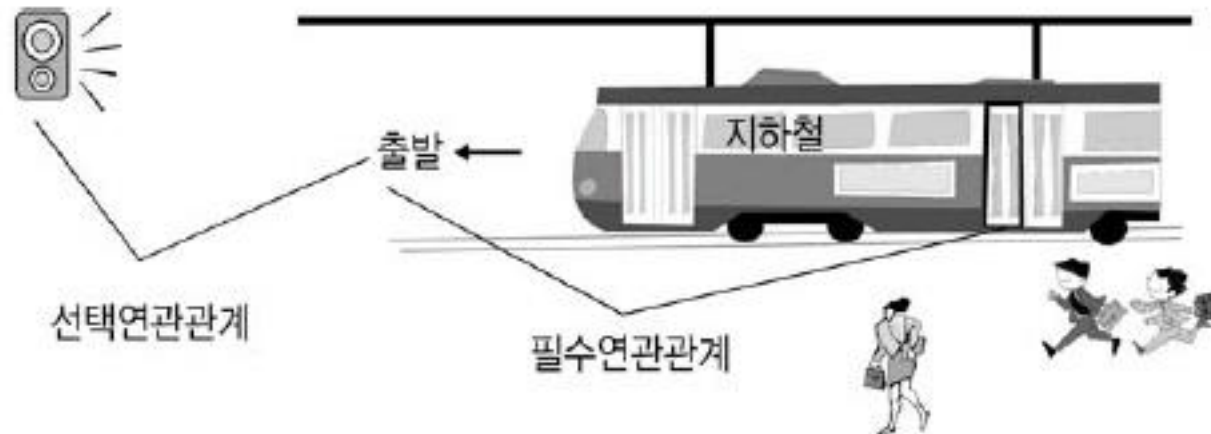


[그림 1-1-36] 관계차수(M:M)



## POINT 5. 관계(Relationship)

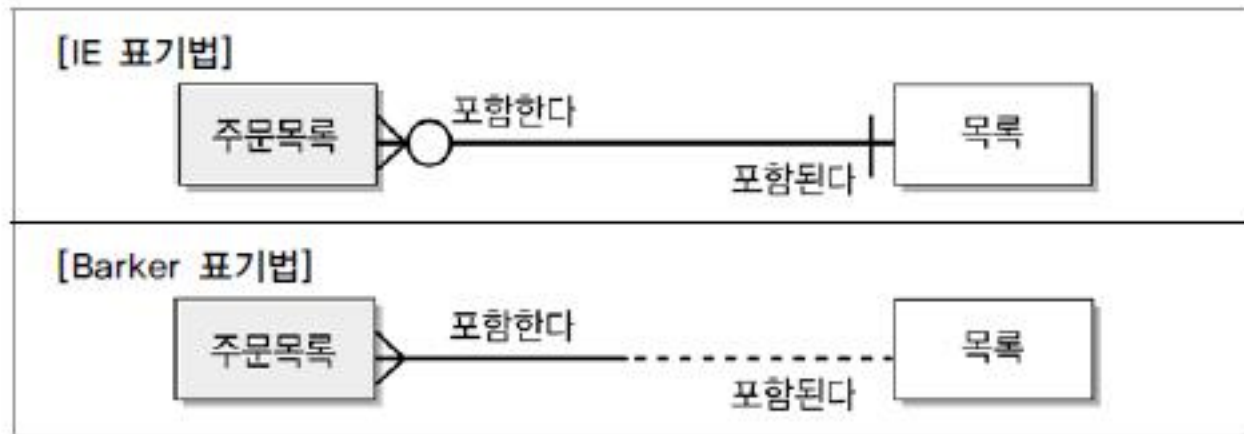
- 관계선택사항(Optionality) -> 필수적인(Mandatory) 관계, 선택적인(Optional) 관계



[그림 1-1-37] 관계선택사양

## POINT 5. 관계(Relationship)

- 관계선택사항(Optionality) -> 필수적인(Mandatory) 관계, 선택적인(Optional) 관계



하나의 주문목록에는 한 개의 목록을 항상 포함하고  
한 목록은 여러 개의 주문 목록에 의해 포함될 수 있다.

[그림 1-1-38] 관계선택참여

## POINT 5. 관계(Relationship)

- 관계 체크사항

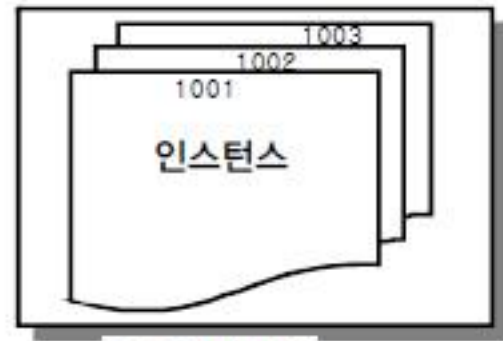
- 기준(Source) 엔터티를 한 개(One) 또는 각(Each)으로 읽는다.
- 대상(Target) 엔터티의 관계참여도 즉 개수(하나, 하나 이상)를 읽는다.
- 관계선택사양과 관계명을 읽는다.



[그림 I-1-39] 관계의 읽는 방법

## POINT 6. 엔터티 식별자(Entity Identifier)

- 식별자 -> 엔터티를 대표할 수 있는 **유일성**을 만족하는 속성  
ex. 회원ID, 계좌번호, 주민등록번호, 외국인등록번호, ...



주문 엔터티



식별자는 엔터티내에서 인스턴스들을  
구분할 수 있는 구분자이다.

[그림 1-1-40] 식별자의 정의

## POINT 6. 엔터티 식별자(Entity Identifier)

### 대표성 여부에 따른 식별자의 종류

종류	설명
주식별자	<ul style="list-style-type: none"><li>• 유일성과 최소성을 만족하면서 엔터티를 대표하는 식별자</li><li>• 다른 엔터티와 참조 관계로 연결 가능</li></ul>
보조 식별자	<ul style="list-style-type: none"><li>• 유일성과 최소성은 만족하지만 대표성을 만족하지 못하는 식별자</li></ul>

## POINT 6. 엔터티 식별자(Entity Identifier)

- 주식별자에 의해 엔터티 내에 모든 인스턴스들이 **유일하게 구분**
- 주식별자를 구성하는 속성의 수는 유일성을 만족하는 최소의 수
- 지정된 주식별자의 값은 **자주 변하지 않은 것이어야 함**
- 주식별자가 지정 되면 반드시 값이 입력되어야 함

[표 1-1-7] 주식별자의 특징

특징	내용	비고
유일성	주식별자에 의해 엔터티내에 모든 인스턴스들을 유일하게 구분함	예) 사원번호가 주식별자가 모든 직원들에 대해 개인별로 고유하게 부여됨
최소성	주식별자를 구성하는 속성의 수는 유일성을 만족하는 최소의 수가 되어야 함	예) 사원번호만으로도 고유한 구조인데 사원분류코드+사원번호로 식별자가 구성될 경우 부절한 주식별자 구조임
불변성	주식별자가 한 번 특정 엔터티에 지정되면 그 식별자의 값은 변하지 않아야 함	예) 사원번호의 값이 변한다는 의미는 이전기록이 말소되고 새로운 기록이 발생하는 개념임
존재성	주식별자가 지정되면 반드시 데이터 값이 존재 (Null 안됨)	예) 사원번호 없는 회사직원은 있을 수 없음

## POINT 6. 엔터티 식별자(Entity Identifier)

### 키의 종류

데이터베이스 키	설명
기본키(Primary key)	후보키 중에서 엔터티를 대표할 수 있는 키
후보키(Candidate key)	유일성과 최소성을 만족하는 키
슈퍼키(Super key)	유일성은 만족하지만 최소성(Not Null)을 만족하지 않는 키
대체키(Alternate key)	여러 개의 후보키 중에서 기본키를 선정하고 남은 키

STUDENT

sid	sname	dept	pid	age
2003001	김철수	국문학	831212-1213112	20
2003002	박선하	국문학	830823-2130121	20
2003003	안미희	컴퓨터공학	820224-2013112	21
2003004	유창식	컴퓨터공학	810509-1934142	22
2003005	임한솔	산업공학	831227-1324123	20



# 데이터 모델과 성능

라 가 영



## POINT 1. 정규화

- 데이터베이스의 **설계를 재구성**하는 테크닉
- **불필요한 데이터(redundancy)**를 제거
- 삽입, 갱신, 삭제 시 발생할 수 있는 **이상현상들을 방지**
- 1~6 정규화 까지 있지만, 실무에서는 대체로 1~3 정규화 과정을 거침
- 정규화의 목적
  - ① 불필요한 데이터(data redundancy)를 제거
  - ② 데이터 저장을 “논리적”으로

## POINT 1. 정규화

- 제 1정규화(First Normal Form, 1NF)
  - 어떤 relation에 속한 모든 domain이 원자 값(atomic value)만므로 되어 있음
  - 모든 attribute에 반복되는 그룹(repeating group)이 나타나지 않음
  - 기본 키를 사용하여 관련 데이터의 각 집합을 고유하게 식별할 수 있음

## POINT 1. 정규화

- 제 1정규화(First Normal Form, 1NF)

Customer

Customer ID	First Name	Surname	Telephone Number
123	Pooja	Singh	555-861-2025, 192-122-1111
456	San	Zhang	(555) 403-1659 Ext. 53; 182-929-2929
789	John	Doe	555-808-9633

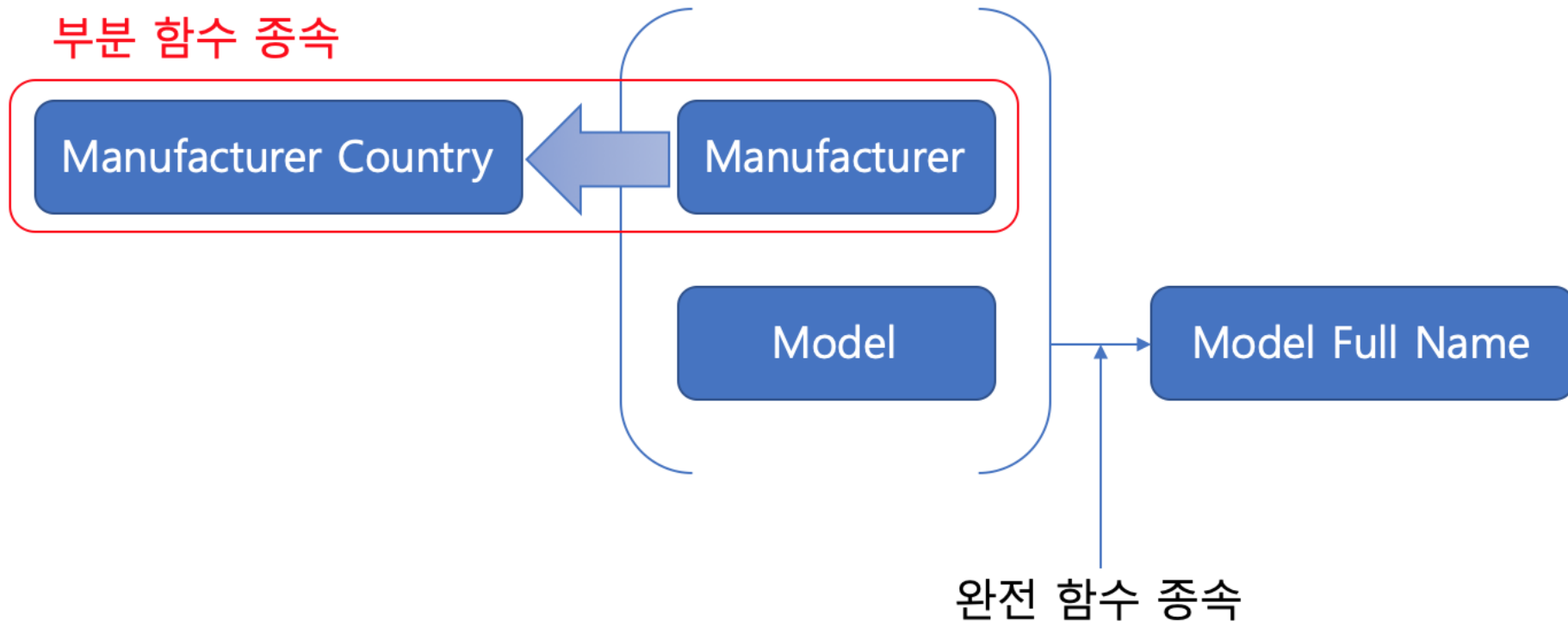


Customer

Customer ID	First Name	Surname	Telephone Number1	Telephone Number2
123	Pooja	Singh	555-861-2025	192-122-1111
456	San	Zhang	(555) 403-1659 Ext. 53	182-929-2929
789	John	Doe	555-808-9633	

## POINT 1. 정규화

- 제 2정규화(Second Normal Form, 2NF)
  - 모든 컬럼이 완전 함수적 종속을 만족함(부분 함수적 종속을 모두 제거)



## POINT 1. 정규화

- 제 2정규화(Second Normal Form, 2NF)

Electric Toothbrush Models

<u>Manufacturer</u>	<u>Model</u>	Model Full Name	Manufacturer Country
Forte	X-Prime	Forte X-Prime	Italy
Forte	Ultraclean	Forte Ultraclean	Italy
Dent-o-Fresh	EZbrush	Dent-o-Fresh EZbrush	USA
Kobayashi	ST-60	Kobayashi ST-60	Japan
Hoch	Toothmaster	Hoch Toothmaster	Germany
Hoch	X-Prime	Hoch X-Prime	Germany



Electric Toothbrush Manufacturers

<u>Manufacturer</u>	Manufacturer Country
Forte	Italy
Dent-o-Fresh	USA
Kobayashi	Japan
Hoch	Germany

Electric Toothbrush Models

<u>Manufacturer</u>	<u>Model</u>	Model Full Name
Forte	X-Prime	Forte X-Prime
Forte	Ultraclean	Forte Ultraclean
Dent-o-Fresh	EZbrush	Dent-o-Fresh EZbrush
Kobayashi	ST-60	Kobayashi ST-60
Hoch	Toothmaster	Hoch Toothmaster
Hoch	X-Prime	Hoch X-Prime

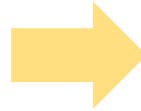
## POINT 1. 정규화

- 제 3정규화(Third Normal Form, 3NF)
  - Relation이 제 2정규화 되었음(The relation is in second normal form)
  - 기본 키(primary key)가 아닌 속성(attribute)들은 기본 키에만 의존해야 함

## POINT 1. 정규화

- 제 3정규화(Third Normal Form, 3NF)

Tournament Winners			
<u>Tournament</u>	<u>Year</u>	Winner	Winner Date of Birth
Indiana Invitational	1998	Al Fredrickson	21 July 1975
Cleveland Open	1999	Bob Albertson	28 September 1968
Des Moines Masters	1999	Al Fredrickson	21 July 1975
Indiana Invitational	1999	Chip Masterson	14 March 1977



Tournament Winners			Winner Dates of Birth	
<u>Tournament</u>	<u>Year</u>	Winner	<u>Winner</u>	<u>Date of Birth</u>
Indiana Invitational	1998	Al Fredrickson	Chip Masterson	14 March 1977
Cleveland Open	1999	Bob Albertson	Al Fredrickson	21 July 1975
Des Moines Masters	1999	Al Fredrickson	Bob Albertson	28 September 1968
Indiana Invitational	1999	Chip Masterson		

## POINT 2. 정규화의 성능

- 정규화의 문제점 -> 데이터 조회(SELECT) 시에 조인(Join)을 유발하기 때문에 CPU와 메모리를 많이 사용함
- 반정규화를 통해 해결 -> 데이터가 중복되기 때문에 또 다른 문제점 발생



## POINT 3. 반정규화(De-Normalization)

- 성능 향상 및 개발과 운영의 단순화를 위해 역으로 정규화를 수행
- 일반적으로 join을 많이 사용해야 할 경우, 대량의 범위를 자주 처리하는 경우 등

## POINT 4. 분산 데이터베이스

- 여러 곳으로 분산되어 있는 데이터베이스를 하나의 가상 시스템으로 사용할 수 있도록 한 데이터베이스
- 논리적으로 동일한 시스템에 속하지만, 컴퓨터 네트워크를 통해 물리적으로 분산되어 있는 모임
- 물리적 Site 분산, 논리적으로 사용자 통합 및 공유

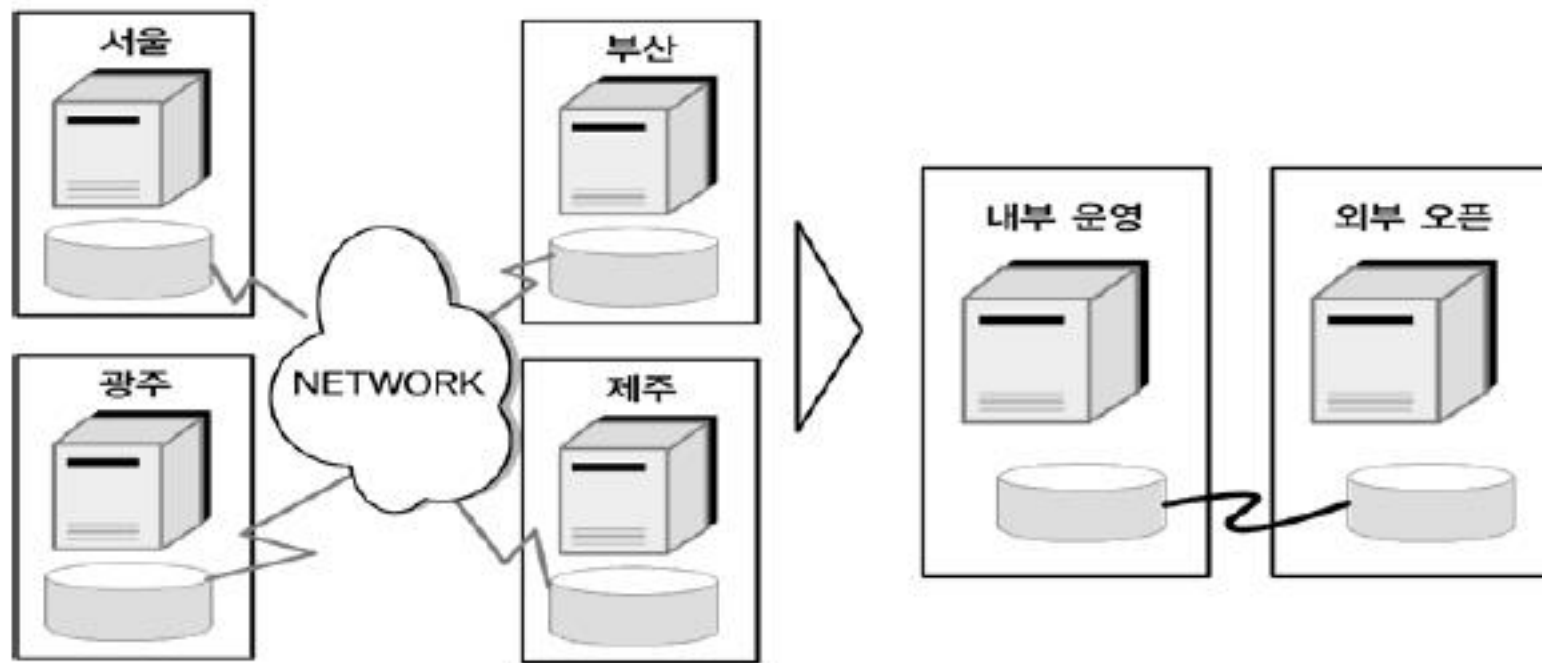
## POINT 4. 분산 데이터베이스

- 1) **분할 투명성**(단편화) : 하나의 논리적 relation이 여러 단편으로 분할되어 각 사본이 여러 site에 저장
- 2) **위치 투명성** : 위치정보가 system catalo에 유지되어야 함
- 3) **지역사상 투명성** : 지역DBMS와 물리적 DB 사이의 Mapping 보장
- 4) **중복 투명성** : 중복되어 있는지 알 필요가 없는 성질
- 5) **장애 투명성** : 장애에 무관한 transaction의 원자성 유지
- 6) **병행 투명성** : 동시 수행 시 결과의 일관성 유지

## POINT 4. 분산 데이터베이스

장점	단점
<ul style="list-style-type: none"><li>- 지역 자치성, 점증적 시스템 용량 확장</li><li>- 신뢰성과 가용성</li><li>- 효율성과 융통성</li><li>- 빠른 응답 속도와 통신비용 절감</li><li>- 데이터의 가용성과 신뢰성 증가</li><li>- 시스템 규모의 적절한 조절</li><li>- 각 지역 사용자의 요구 수용 증대</li></ul>	<ul style="list-style-type: none"><li>- 소프트웨어 개발 비용</li><li>- 오류의 잠재성 증대</li><li>- 처리 비용의 증대</li><li>- 설계, 관리의 복잡성과 비용</li><li>- 불규칙한 응답 속도</li><li>- 통제의 어려움</li><li>- 데이터 무결성에 대한 위협</li></ul>

## POINT 4. 분산 데이터베이스

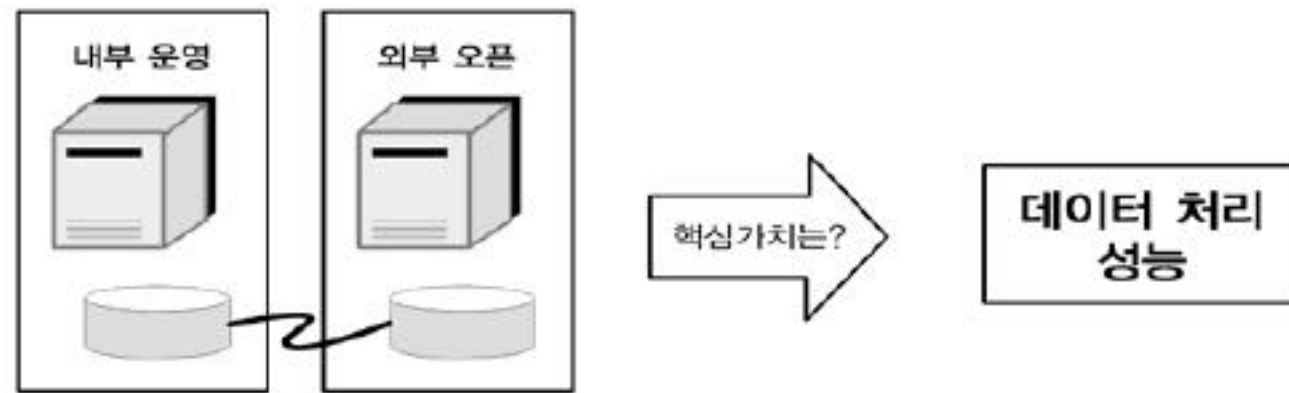


〈이전〉위치 중심의 분산설계

〈현재〉업무필요에 의한 분산설계

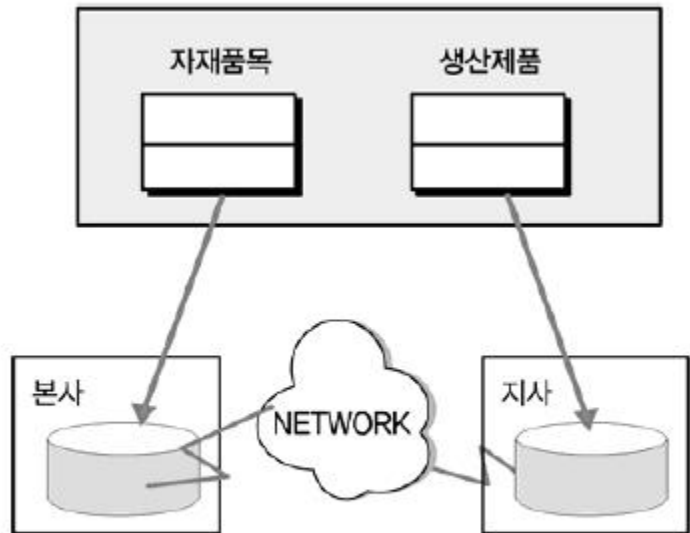
[그림 1-2-39] 분산설계 방향성

## POINT 4. 분산 데이터베이스



[그림 1-2-40] 분산 데이터베이스 핵심가치

POINT 4. 분산 데이터베이스



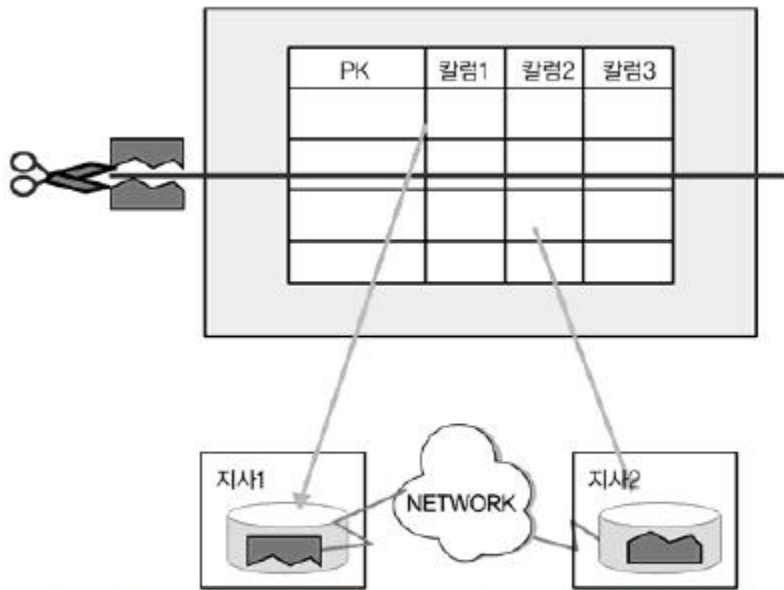
[그림 1-2-41] 테이블별 위치 분산

테이블 위치	자재품목	생산제품	협력회사	사원	부서
본사	•		•		•
지사		•		•	

[그림 1-2-42] 테이블별 위치 분산

## POINT 4. 분산 데이터베이스

- 수평분할 (Horizontal Fragmentation)



[그림 1-2-43] 테이블 분할 분산 - 수평분할

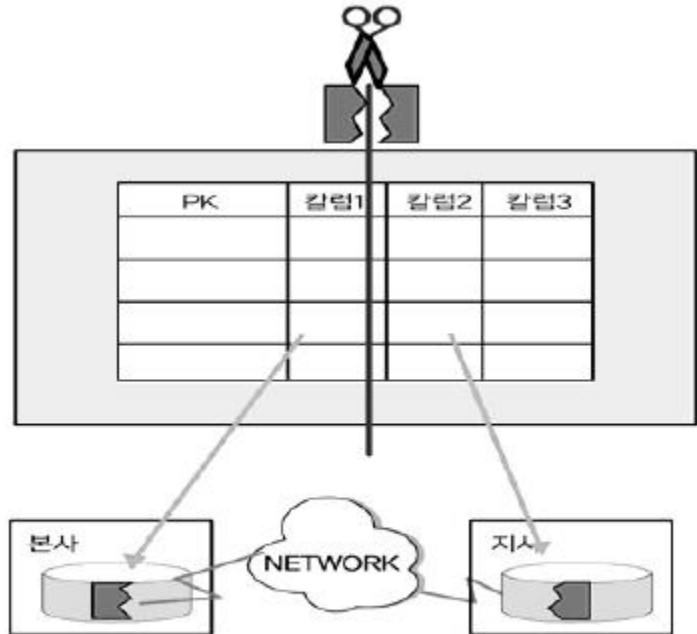
테이블 위치	고객	생산제품	협력회사	사원	부서
지사 1	●	●	●	●	●
지사 2	●	●	●	●	●

[그림 1-2-44] 테이블 수평분할



## POINT 4. 분산 데이터베이스

- 수직분할 (Vertical Fragmentation)



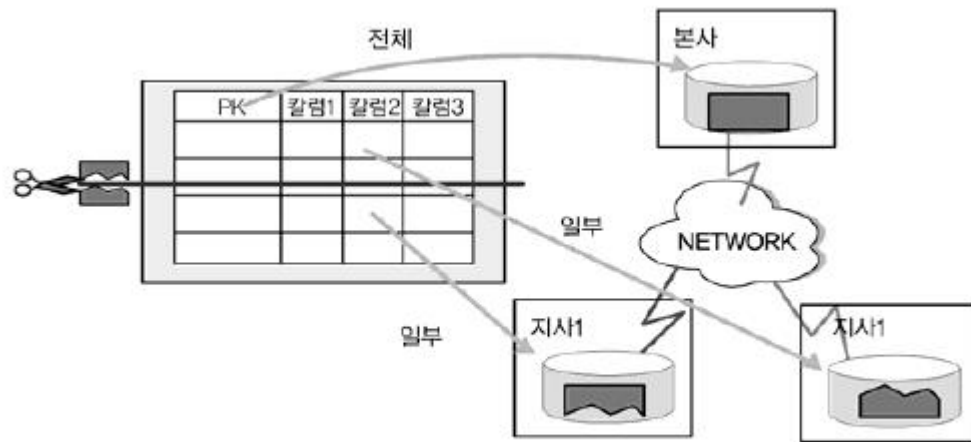
[그림 1-2-45] 테이블 수직분할

위치 \ 테이블	테이블	
	제품	분할 칼럼
본사	●	제품번호, 단가
지사	●	제품번호, 재고량

[그림 1-2-46] 테이블 수직분할 모델링

## POINT 4. 분산 데이터베이스

- 부분복제 (Segment Replication)



[그림 1-2-47] 테이블 복제 분산 - 부분복제

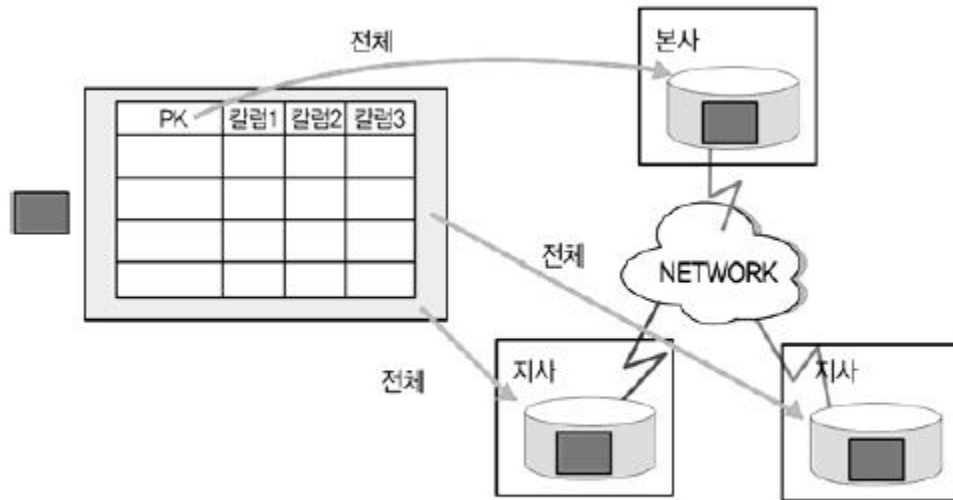
위치 \ 테이블	고객
본사	●
지사 1	○
지사 2	○

본사에는 전국의 고객정보를 관리하고 지사 1의 고객테이블에서는 지사 1에 속한 고객정보를 지사 2의 고객테이블에서는 지사 2에 속한 고객정보를 관리한다.

[그림 1-2-48] 테이블 복제 분산 - 부분복제

## POINT 4. 분산 데이터베이스

- **광역복제** (Segment Replication)



[그림 1-2-49] 테이블 복제 분산 - 광역복제

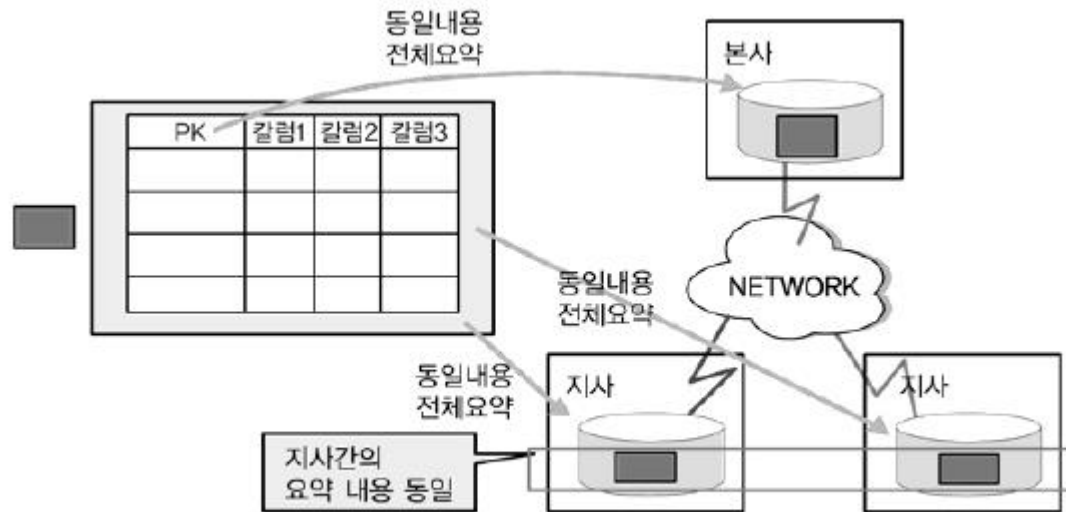
위치 \ 테이블	코드
본사	●
지사 1	●
지사 2	●

본사, 지사1, 지사2 모두 동일한 양의 코드 테이블의 데이터를 가지고 있다.

[그림 1-2-50] 테이블 복제 분산 - 광역복제

## POINT 4. 분산 데이터베이스

- 분석요약(Rollup Replication)



[그림 1-2-51] 테이블 요약 분산 - 분석요약

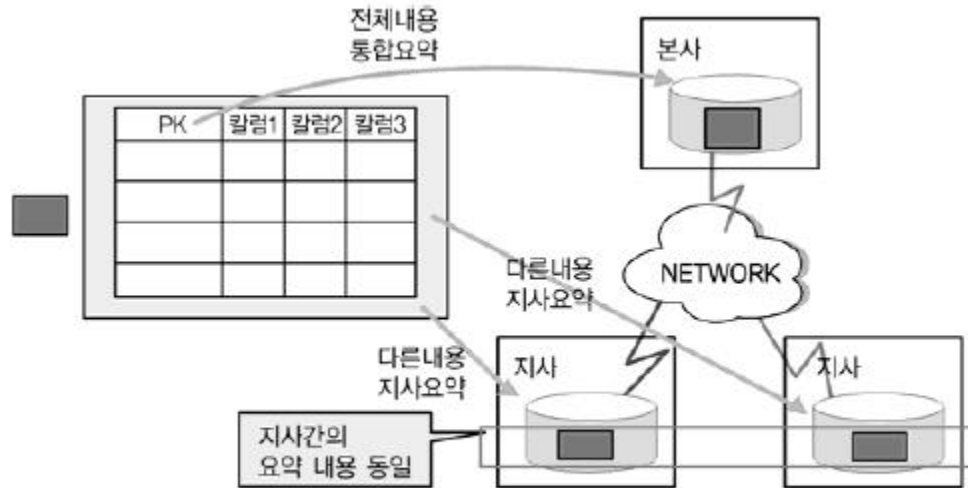
위치 \ 테이블	판매실적
본사	●
지사 1	●
지사 2	●

지사 1에서는 지사 1의 판매실적이 있고  
지사 2에서는 지사 2의 판매실적이 존재  
한다. 본사에서는 모든 지사의 판매실적  
을 통합한 실적 데이터가 생성된다.

[그림 1-2-52] 테이블 요약 분산 - 분석요약

## POINT 4. 분산 데이터베이스

- **통합요약**(Consolidation Replication)



[그림 1-2-53] 테이블 요약 분산 - 통합요약

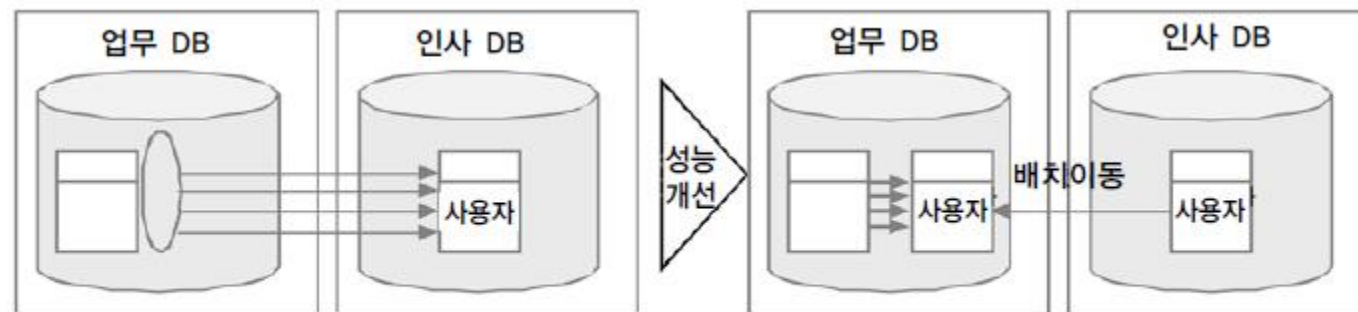
테이블 위치	테이블	판매실적
본사		●
지사 1		○
지사 2		○

지사 1과 지사 2에 판매실적이 존재하지만 서로 다른 내용으로 존재한다. 본사에서 는 모든 지사의 판매실적을 통합한 실적 데이터가 생성된다.

[그림 1-2-54] 테이블 요약분산 - 통합요약

## POINT 4. 분산 데이터베이스

- 성능이 향상된 사례



트랜잭션 개별적으로 원격지 조인

트랜잭션 내부 조인

[그림 1-2-55] 업무 특성에 따른 분산환경 구성

