

# 데이터베이스론: 이론과 실제

경희대 박주석

1. 정보자원관리와 데이터관리
2. 정보시스템구축과 데이터접근방식
3. 정보공학방법론과 데이터모델링
4. 관계형 모델과 정규화 이론

# 데이터베이스 활용을 위한 3가지 관점

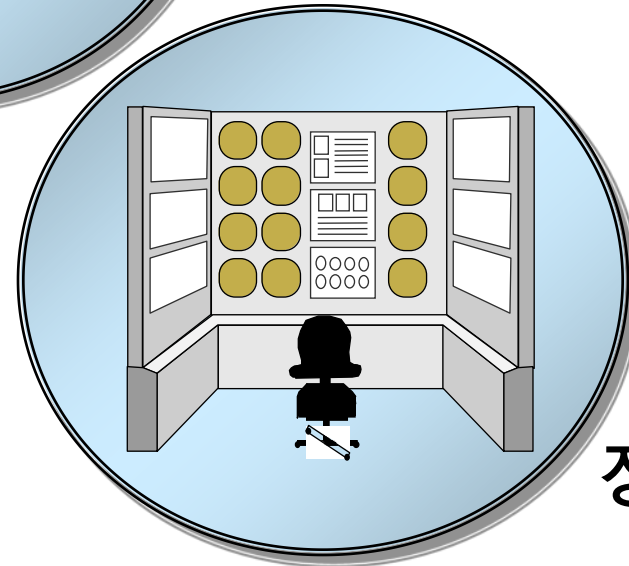
비즈니스



컨설팅



정보기술



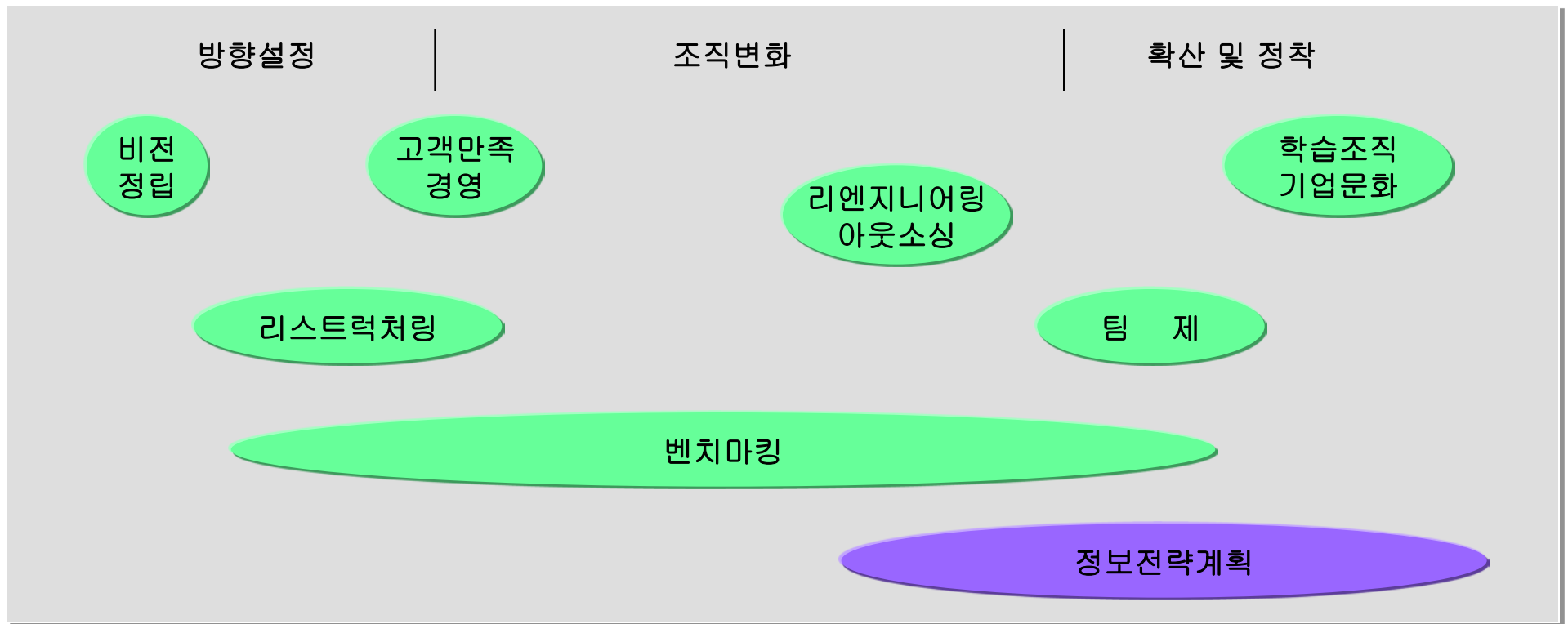
# 제 3 장

## 정보전략계획과 데이터모델링



컨설팅

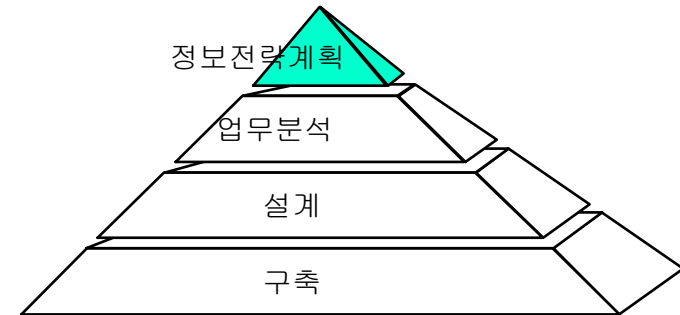
## 다양한 컨설팅방법론



## 정보공학 방법론과 정보전략계획

현재 세계적으로 가장 널리 검증되어 이용되어지는 방법론이 바로 정보공학 방법론(IE)인 것입니다. 이것은 미국의 James Martin이 1980년대 중반에 개발한 방법론으로 시스템 개발을 목적으로 개발되었으나, 오늘날에는 조직의 정보화 전략의 재정립 및 전사적인 통합시스템 개발의 방법론으로 이용되어지고 있습니다.

이 정보공학 방법론은 4개의 단계로 이루어지며, 첫 단계가 바로 정보전략계획(ISP) 단계인 것입니다. 이 단계에서 조직의 정보화 비전 설정, 목표 및 문제점 분석, 주요성공요인분석을 통하여 조직의 정보화 계획을 수립하며, 이와 더불어 현재 및 미래 정보기술 영향분석과 조직의 현 정보기술 현황을 분석하여 조직의 정보기술 체계를 도출합니다. 이러한 정보기술 체계를 바탕으로 기업은 필요한 요소기술을 파악할 수 있으며, 요소기술의 구축을 위한 전체적인 틀(framework)을 바탕으로 체계적으로 조직의 정보화를 발전시켜 나갈 수 있는 것입니다.



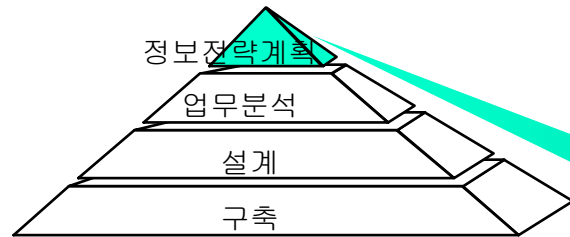
정보공학 방법론은 전체조직의 관점에서 출발하고 있으며 시스템 통합적인 관점이 강하므로 전사적인 통합시스템 구축에 있어서 필수적인 방법론입니다. 전체 시스템

의 통합을 위해서는 전체시스템이 하나로 연결되는 매개체가 필요한데 이 매개체가 바로 통합 데이터 모델링입니다. 전체적인 통합모델링을 기반으로 개별적인 시스템의 연계방안이 모색되어야 합니다. 또한 정보공학방법론은 조직목표와 전략으로부터

출발하여 이를 어떻게 정보화 계획으로 구현 시켜 나가는가 하는 방법을 제시합니다. 따라서 새로운 목표와 전략을 구현할 수 있는 프로세스의 변화는 자연스럽게 업무 재설계와 연계되어질 수 있으며 이를 바탕으로 새로운 관점의 프로세스 모델링이 만들어집니다.

이와 같이 정보공학 방법론(IE)은 앞에서 언급한 5가지 중 4가지를 자연스럽게 충족시켜 주며, 마지막 한가지인 프로젝트 관리 및 평가 방안의 마련은 정보공학 방법론의 첫번째 단계에서 추출된 다양한 정보를 바탕으로 추가적인 방법의 적용으로 가능할 수 있습니다. 따라서 정보공학 방법론을 통해 정보화계획 방안의 도출이 충분히 가능하며, 추가적인 방법의 병행실시로 과제를 완벽히 해결할 수 있습니다.

## 정보전략 계획 (ISP)



기업의 장기계획과  
정보화 계획의 연계

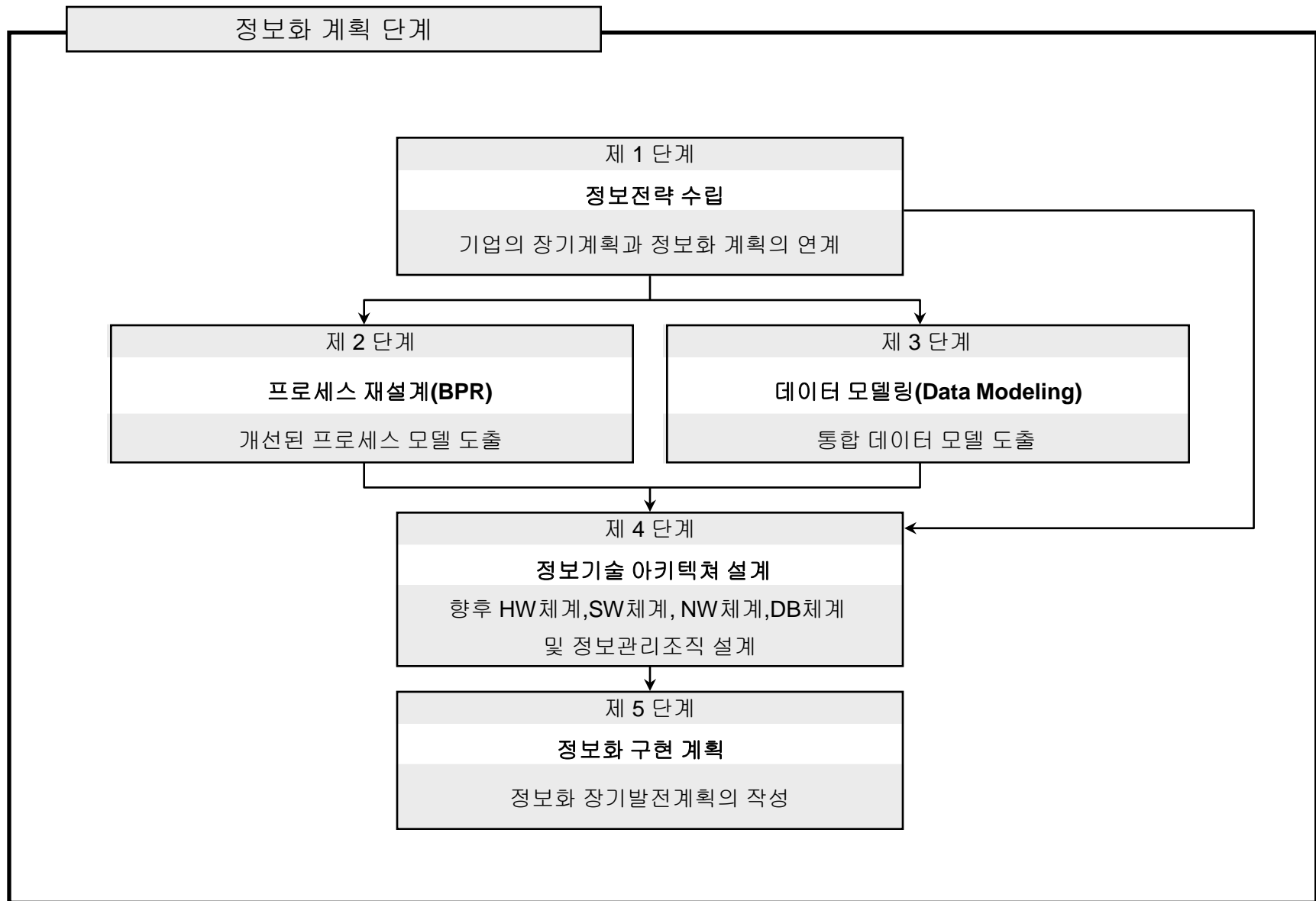
- ← 목표 및 문제점 분석
- ← 핵심성공요인 분석
- ← 기술동향 분석
- ← 정보시스템 비전

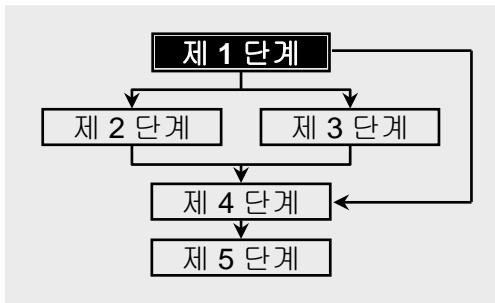
정보화 계획

기업의 정보화  
모형 완성

프로세스 모델

데이터 모델



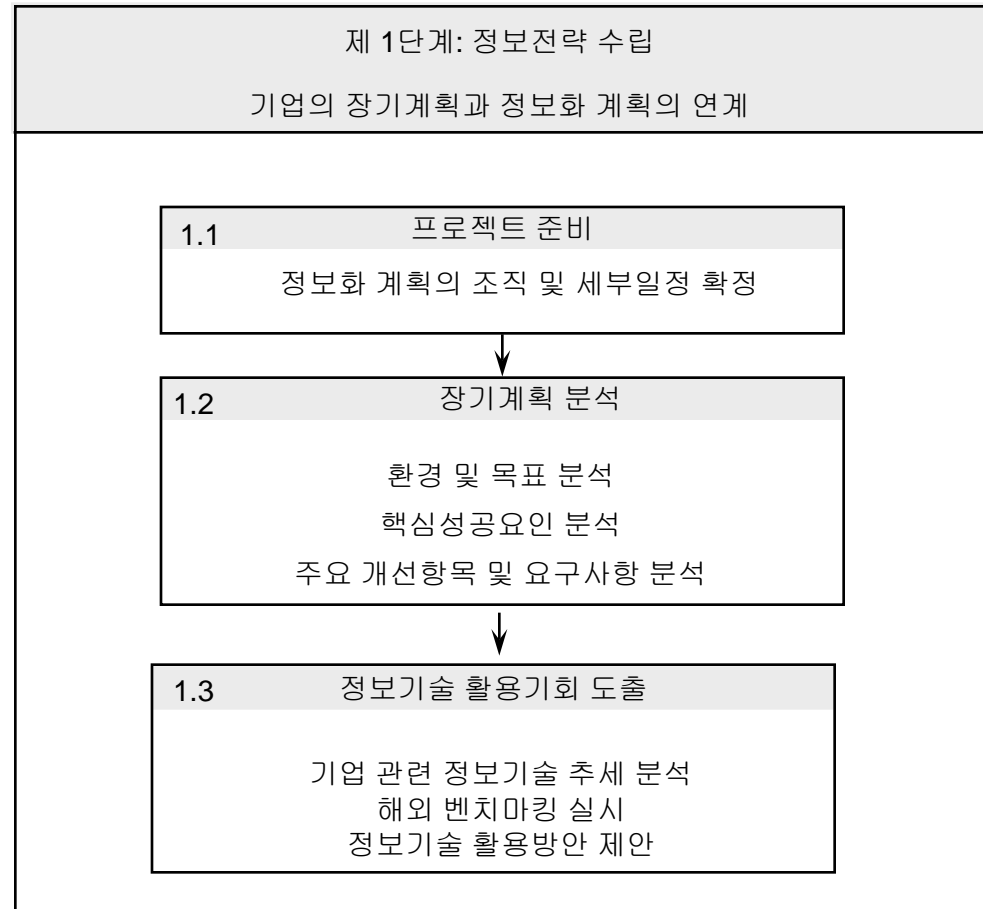


#### 필요성

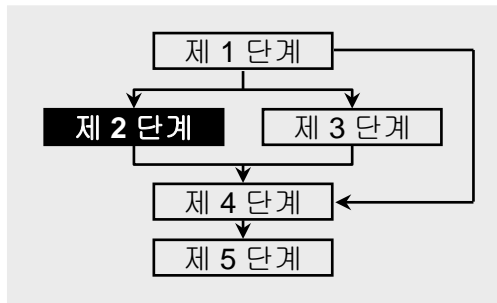
정보시스템의 활용목적은 최근 들어 내부관리업무의 단순한 지원에서 조직의 목적달성에 직접적으로 기여하는 무기로 변화하고 있습니다. 따라서 조직의 목적과 정보시스템을 연계시키는 정보전략계획의 수립이 매우 중요합니다.

#### 내용

정보전략계획은 정보화요구와 정보 기술의 활용기회를 규명 하는 것이 주된 내용입니다. 조직의 정보요구는 목적, 핵심 성공요인, 목적달성지표, 핵심 성공요인의 상태 등을 분석하며 규명됩니다. 정보기술의 활용 기회는 정보기술을 활용한 문제점 해결방안의 연구에서 발견될 수 있습니다.





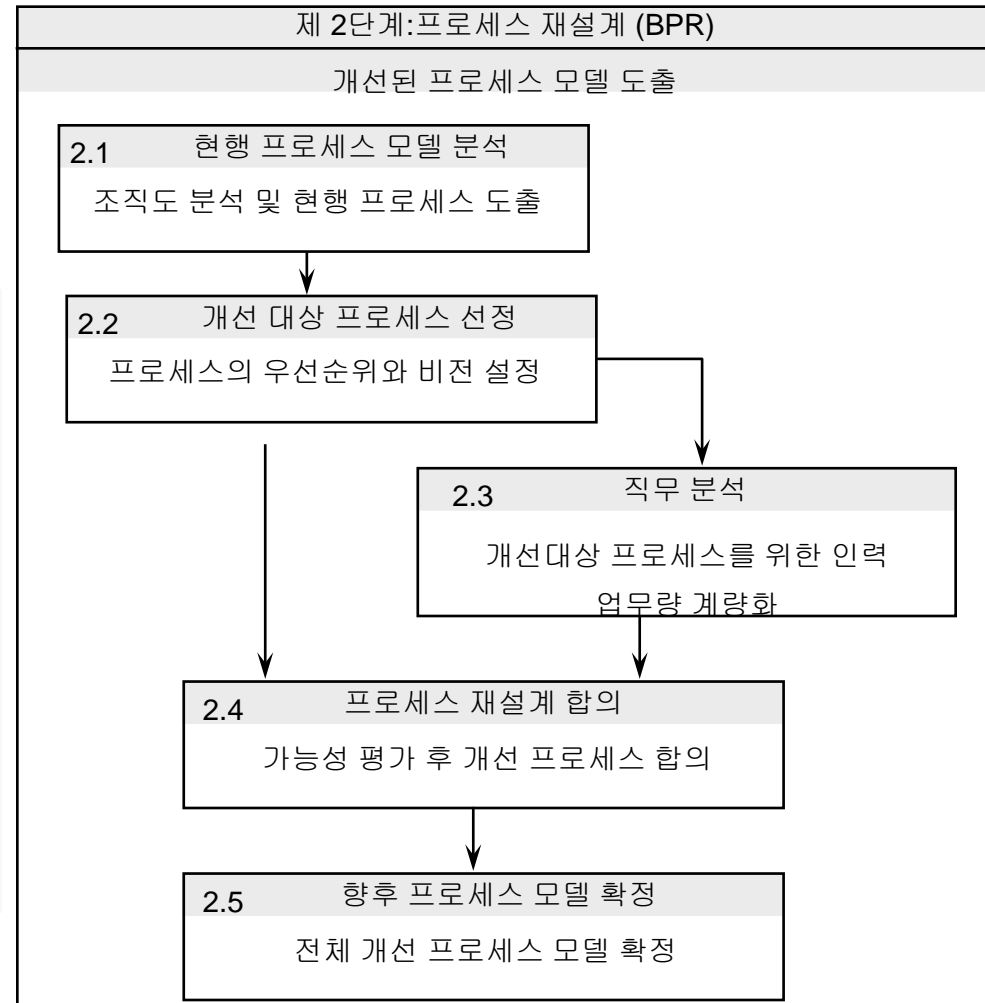


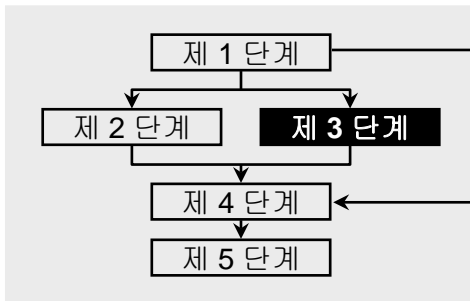
#### 필요성

오늘날 대부분의 조직은 분업화, 전문화에 근거하여 기능별 조직형태를 갖추고 있습니다. 기능별 조직구조에서는 부서간 정보의 공유가 어렵고, 의사 결정 시간이 지연되며, 프로세스의 최종산출물에 대한 책임소재가 불분명합니다. 또한 기능별 성과에 근거한 평가제도는 부처이기 주의를 조장하고 전체 조직의 성과를 저하시킵니다.

#### 내용

B.P.R은 프로세스 중심으로 업무를 구성하며 전체 조직의 성과를 향상시키자는 것입니다. 프로세스 중심의 구성은 정보 기술의 발달에 의하여 가능해진 것으로, 새로운 정보기술의 잠재력을 활용하여 정보의 흐름을 바꾸는 것입니다. 프로세스개선은 고정관념을 타파하고 강력한 개선의 의지가 반영되어야 성공할 수 있습니다.



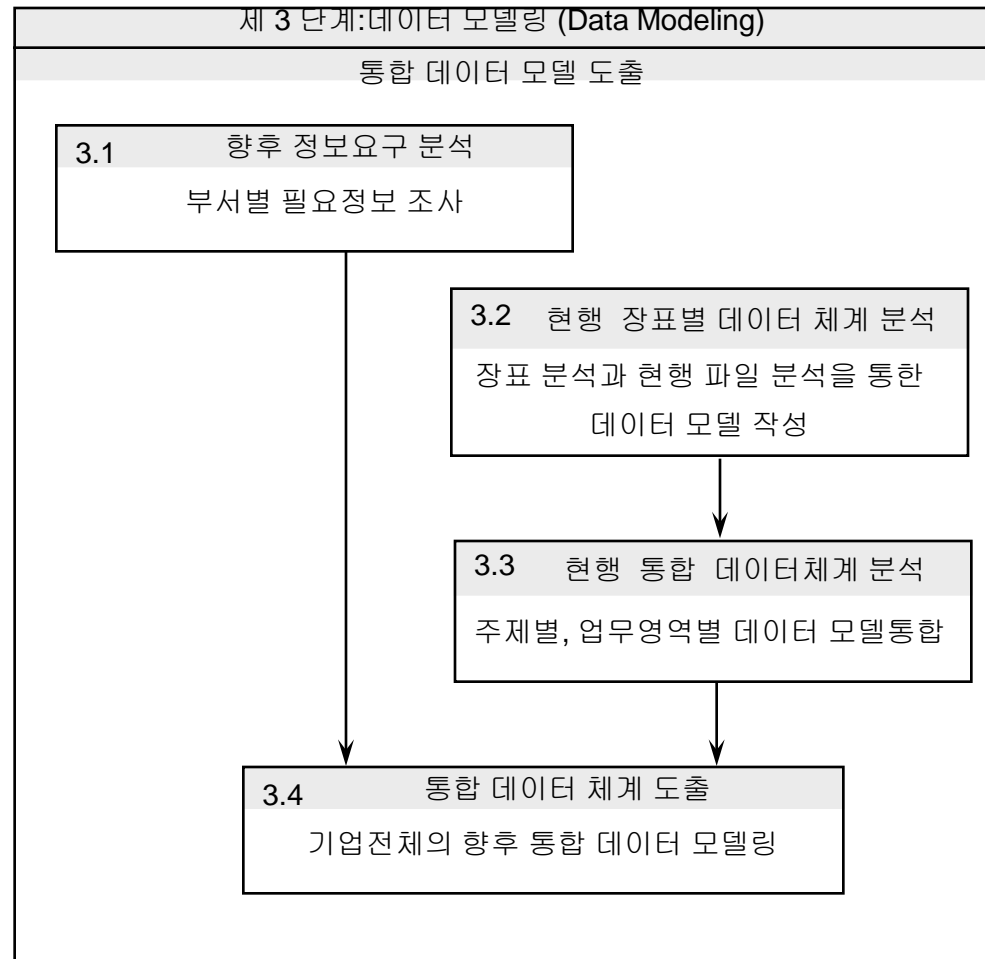


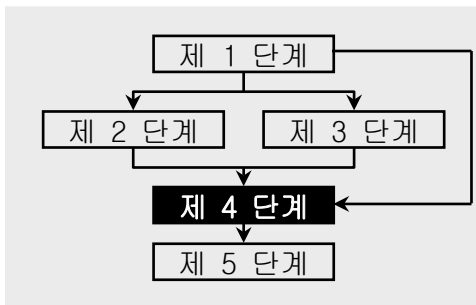
#### 필요성

각 부문의 정보시스템을 개별적으로 구축함으로써 데이터의 중복과 불일치가 발생하며 정보시스템의 유지 보수나 통합에 많은 문제를 초래할 수 있습니다. 이러한 문제점을 해소하기 위하여 모든 프로그램이 사용하는 데이터 베이스를 통합의 관점으로 설계 하는 것은 매우 중요합니다. 기업전체의 통합데이터 모델은 이러한 데이터베이스 설계의 틀이 되며 정보시스템 각부문의 연결고리를 보여줍니다.

#### 내용

데이터 모델링은 기업이 현재 사용하는 모든 데이터와 향후에 요구되는 데이터를 체계적으로 통합하는 것입니다. 현재 사용하는 데이터 분석은 장표와 현행 파일구조를 이용하여 분석하고 향후에 요구되는 데이터 분석은 향후 프로세스 모델과 사용자 인터뷰를 통하여 설계됩니다.



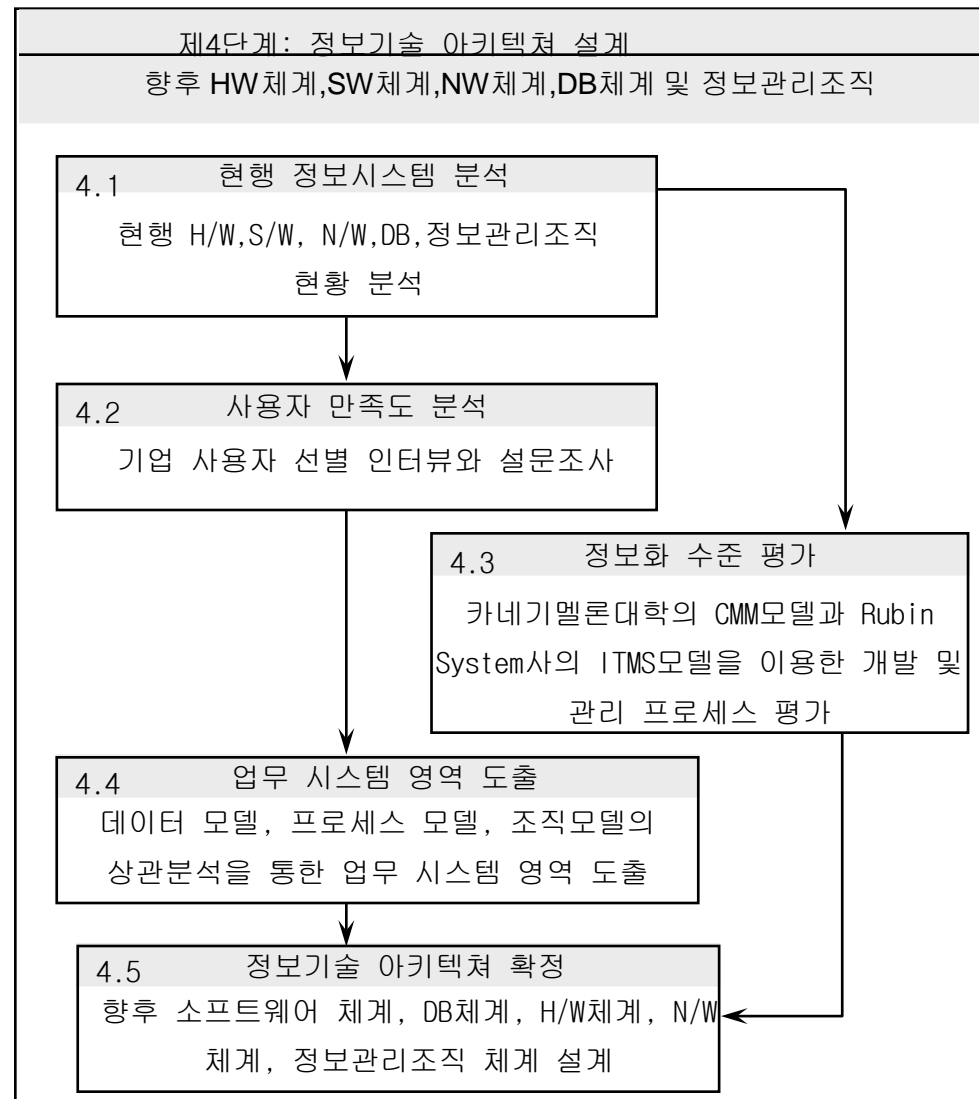


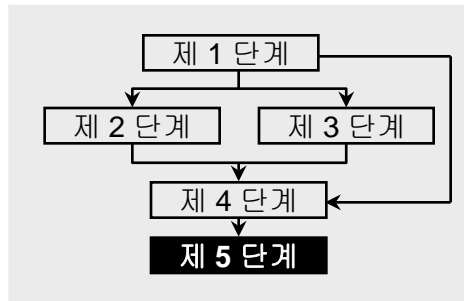
#### 필요성

목표와 요구사항을 충족시키며 현재의 정보시스템 환경과 새로운 정보기술 환경을 조화시키는 기업의 정보시스템 청사진을 보여줍니다. 만약 기업의 정보시스템 청사진이 없다면 앞으로 5년동안 정보시스템을 효과적으로 구축관리할 수 없습니다.

#### 내용

정보기술 아키텍처 설계는 제1단계 정보전략계획수립, 제2단계 프로세스 재설계, 제3단계 통합데이터 모델링을 종합하여 기업의 향후 정보기술 아키텍처를 설계하는 것입니다. 향후 정보기술 기반은 하드웨어 체계, 소프트웨어 체계, 데이터베이스 체계, 네트워크 체계, 정보관리 조직체계의 5가지로 구성되어 집니다.





#### 필요성

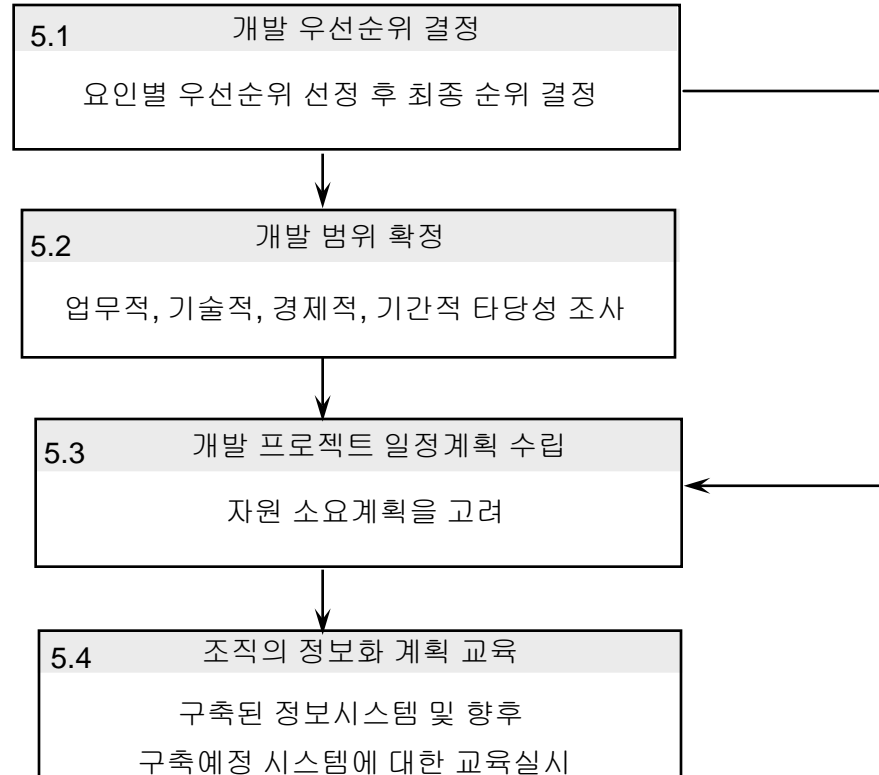
현행 정보시스템은 전 단계에서 규명된 정보화 요구를 충족시키기에 부족할 수도 있습니다. 또한 정보기술의 변화에 따라 재구축되어야 하는 경우도 있을 것입니다. 현행 시스템의 재구축은 조직전체의 정보시스템 모델, 즉 정보화 모델의 틀안에서 계획적으로 진행되어야 시스템 통합의 목적을 달성할 수 있습니다.

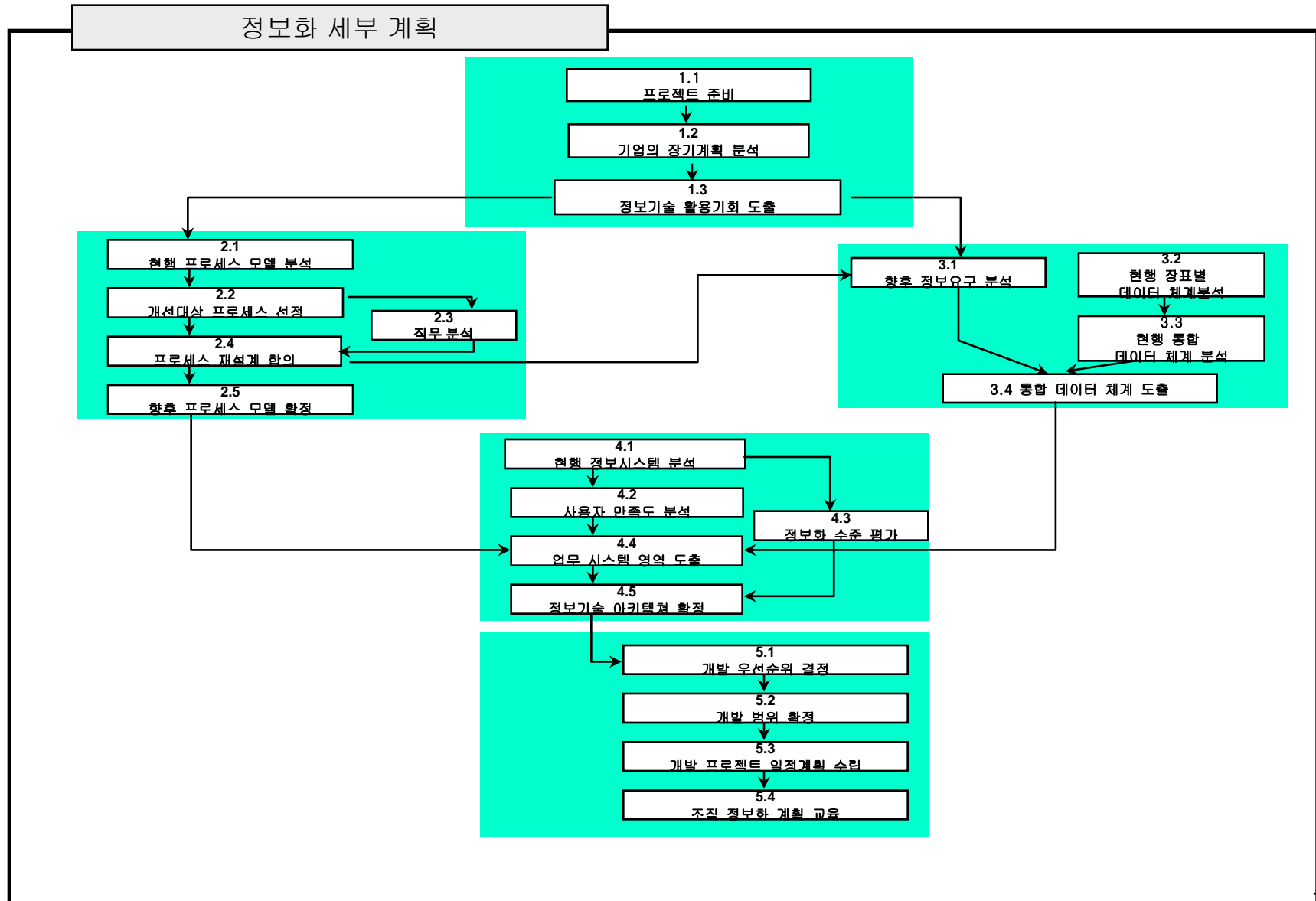
#### 내용

정보시스템 구현은 정보화 모델의 틀안에서 계획적으로 진행되고 통제되어야 합니다. 응용시스템간의 정보공유, 프로그램 재사용등은 잘 짜여진 계획하에서만 가능한 것입니다. 기존의 시스템은 수명이 다하는 시점에서 재개발되어야 하고, 현재 결여된 시스템은 우선 순위에 따라 최적의 방법으로 도입하여야 합니다. 이를 위해 서는 자원소요계획, 추진조직 등에 대한 철저한 준비가 필요 합니다.

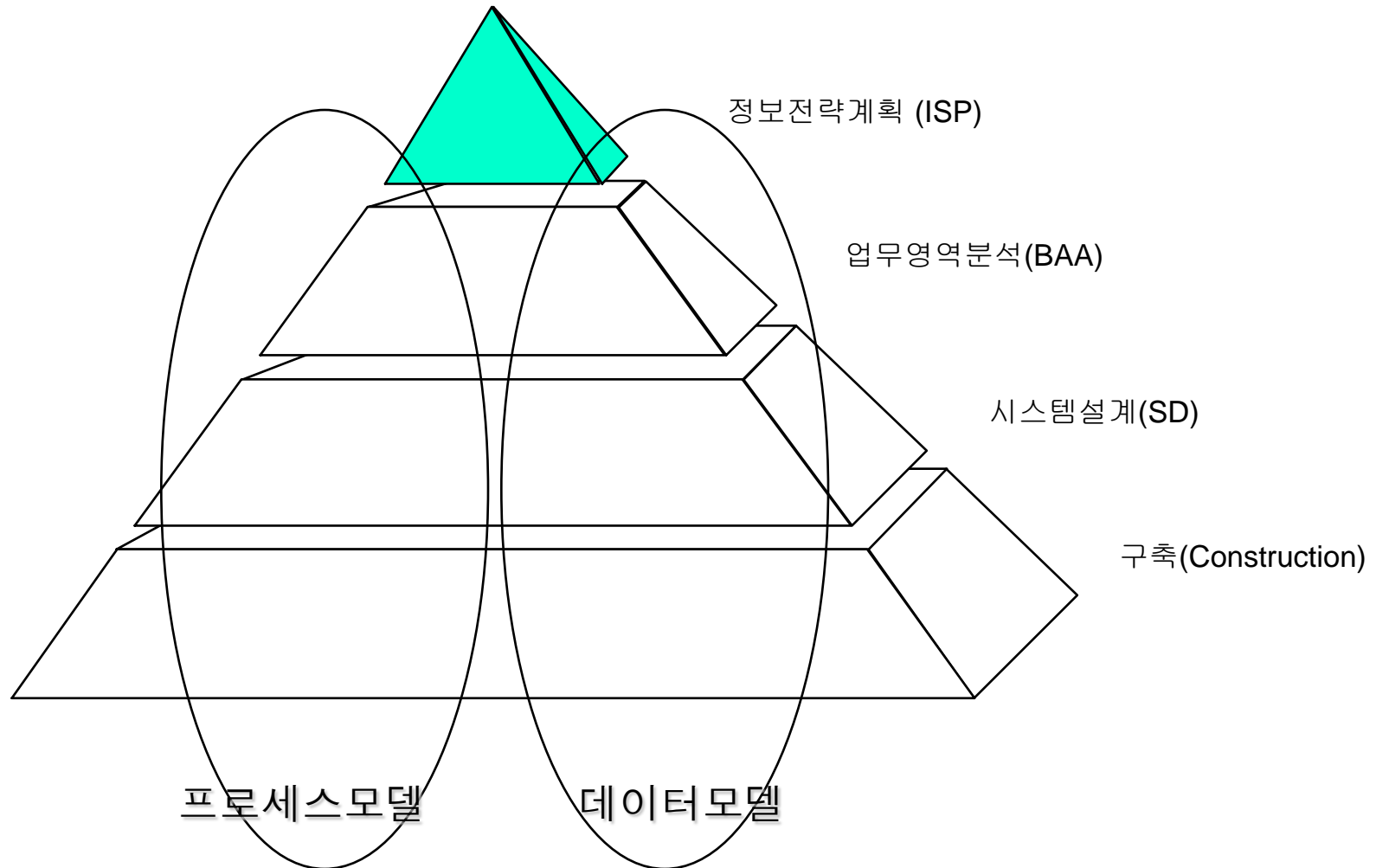
#### 제 5단계 : 정보화 구현 계획

##### 정보화 장기 발전 계획의 작성





# 프로세스모델과 데이터모델



## ☑ **Data Model** 은 왜 사용하는가 ?

- ✓ To Represent Data
- ✓ To Make the Data Understandable

## ☑ **Data Model의 종류**

- ✓ Relational Model
- ✓ Semantic Data Model
- ✓ Entity - Relationship Data Model
- ✓ Object - Oriented Model
- ✓ Logic - Based Model



## ☑ Entity - Relationship Data Model

" The Entity - Relationship Model - Toward a Unified View of Data. "  
by Chen, P., ACM - TODS 1976

1. Simple ER Model

2. Extended ER Model (Complicated ER Model )

# ☑ Today's ER DIAGRAM

## 1. Entity

- ① Entity Name
- ② Attribute
- ③ Class/Subclass

cf. Strong Entity  
Weak Entity  
Associative Entity

## 2. Relationship

- ① Role Name
- ② Degree
- ③ Connectivity
- ④ Membership
- ⑤ Attribute

## 업무설명: 인사관리

2001년도에 경희대 경영학과를 졸업한 오정보씨는 오랜 기간 동안의 백수생활을 마치고 마침내 한국담배인삼공사의 판매관리부에 입사하게 되었다. 한국담배인삼공사는 인사관리 데이터베이스를 구축하여 관리하고 있다.

- 먼저 개인이 회사로 입사하였을 경우 사원목록을 만들어야 하고,
- 정해진 회사 조직도에 따라 그 개인의 업무와 함께 부서를 배정하고
- 먼저 가족에 대한 사항을 입력하기 위해 가족목록을 작성하여야 한다.
- 개인에 대한 경력, 학력, 상벌, 자격사항 등도 관리되어야 한다.
- 그리고 월별 급여 사항도 있다.
- 회사에 포함되지 않는 개인의 자료는 개인목록을 만들어 저장할 수 있다.

## 인사관리 데이터모델

종업원, 부서, 학력, 경력, 가족, 정규직, 임시직, 프로젝트,

Independent Entity : Strong Entity, Class

Dependent Entity : Weak Entity, Subclass

Independent Relationship : Degree 1, 2, 3

Dependent Relationship : Associative, Is Weak, Is-A

cf. Logical Entity vs. Physical File:

External Model, Logical Model, Internal Model

## Relational Model vs. ER Model

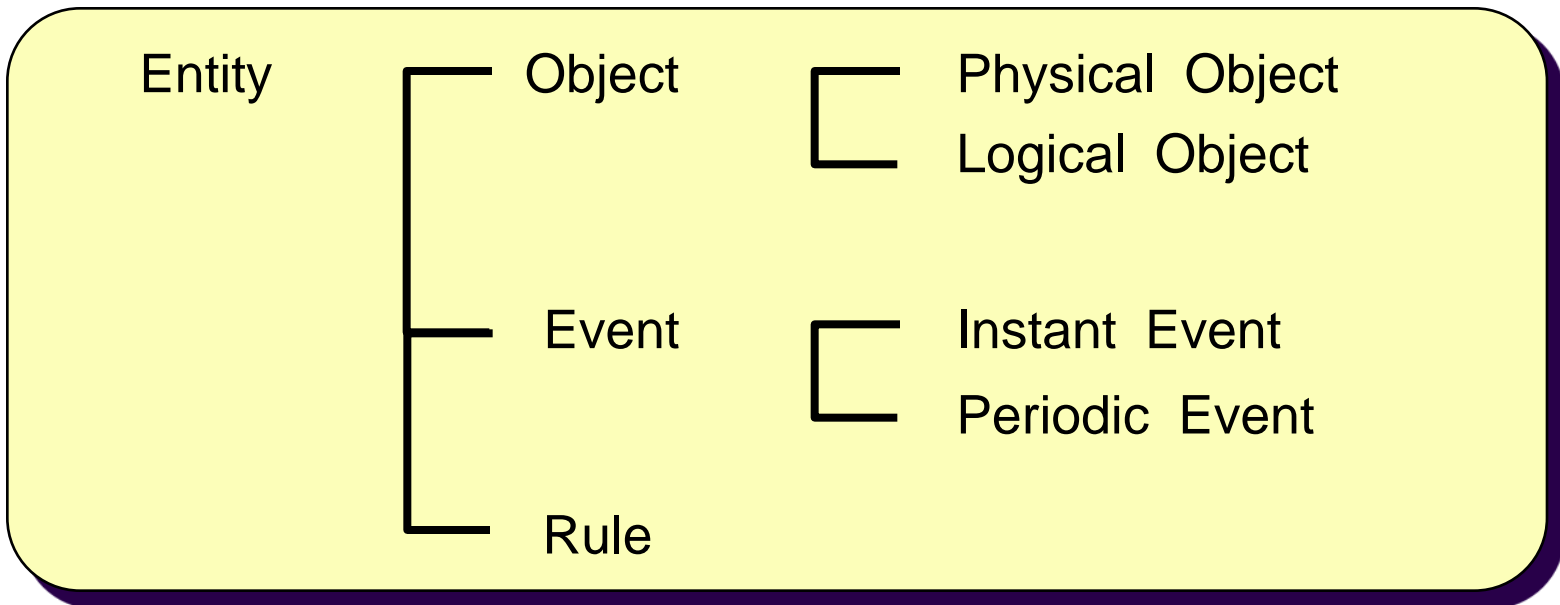
Entity-Relationship Model :

Relational Model:

## ☑ Entity의 의미

### □ Entity의 정확한 의미 :

" **Interesting** Object, Event or Rule"



### □ Entity 의 직관적 의미 :

" A set of Facts that you wish to be Recorded, so Be Careful ! "

## ☑ Relationship의 의미

### ❑ Relationship의 정확한 의미 :

" Interesting Association or Rules between Entities "

- Fixed
- Mandatory
- Optional Becomes Mandatory
- Optional

### ❑ Relationship의 직관적 의미 :

" An Implementable Rule between Primary & Primary Keys  
or between Primary & Foreign Keys "

## 업무설명: 판매관리

- 한국담배인삼공사의 판매관리부에 입사하게 된 신입사원 오정보씨에게 맡겨진 업무는 서울시내 동대문구, 종로구, 강남구에서 판매되는 담배판매현황을 정리하여 보고하는 일이다. 3개 구에 속해 있는 각 담배판매점들은 담배를 공급하는 직원에게 매달 초에 지난달의 담배판매실적표를 제출하고, 이 실적표가 오정보씨에게 전달된다. 3개구에 속해 있는 각 담배판매점들은 담배를 공급하는 직원에게 매달초에 지난달의 담배판매실적표를 제출하고, 이 실적표가 오정보씨에게 전달된다.
- 현재 3개구에 속해있는 담배판매점수는 각 구별로 6개점씩 총 18개점이다. 담배판매점의 형태는 슈퍼, 편의점, 가판점, 자동판매기의 4가지이다.
- 따라서 오정보씨는 담배인삼공사 판매관리부 입장에서 간단한 데이터베이스를 구축하고자 한다.



## 판매점정보

아래의 표는 오정보 신인사원이 관리해야 할 판매점들에 대한 정보로서 워드파일로 작성해둔 표이다.

판매점명	판매점형태	판매인명	구	동	연락처
광동슈퍼	슈퍼	최숙정, 이재욱	종로구	명륜동	768-2937
덕산골	가판점	최윤희	동대문구	회기동	946-2323
미니스톱	편의점	이보만	동대문구	전농동	989-2316
서남	편의점	박민현, 박주석, 박재홍	강남구	대치동	566-2845
서초	편의점	석재호	강남구	개포동	576-2385
선화	가판점	김이환, 엄재석	강남구	논현동	543-2373
세븐일레븐	편의점	최철	종로구	명륜동	765-7736
알프	슈퍼	박동혁	강남구	반포동	563-2837
역삼	슈퍼	채영환	동대문구	이문동	931-3233
용화	가판점	김기린	동대문구	전농동	977-3484
우도	슈퍼	김현모, 조재만	종로구	관철동	756-2319
우진	가판점	이양원	종로구	인사동	765-3635
진훈	슈퍼	정진화	동대문구	회기동	930-2393
하나로	편의점	김영학	동대문구	이문동	983-2938
한양유통	자판기	이수원, 이수연	강남구	반포동	556-9836
한우리	자판기	장용식	종로구	인사동	756-3540
현대	슈퍼	김태형	강남구	논현동	523-3245
LG25	편의점	권태욱	종로구	인사동	756-2391 - 25 -

## 판매내역 보고양식

각 판매점들은 아래와 같은 정해진 보고양식표에 매월말에 담배판매량을 작성하여 두었다가 다음달초에 담배를 공급받을 때 공급사원에게 제출한다. 공급사원은 각 판매점별로 모아진 월별 담배판매실적표를 오정보 신입사원에게 전달한다.

판매처명 :

집계월 :

상품명	판매량	상품명	판매량
88디럭스		레종	
던힐		마일드세븐	
디스		시마	
디스플러스		에쎬	

## 예제 : 판매처별 1월달 판매내역

판매처명 : 덕산골

집계월 : 2003년 1월

상품명	판매량	상품명	판매량
88디럭스	420	레종	440
던힐	550	마일드세븐	590
디스	450	시마	410
디스플러스	430	에세	330

판매처명 : 선화

집계월 : 2003년 1월

상품명	판매량	상품명	판매량
88디럭스	300	레종	380
던힐	590	마일드세븐	380
디스	350	시마	320
디스플러스	450	에세	580

## 예제 : 판매처리 화면

거래번호 : 1870211

거래일자: 2006/10/23

영업담당자: 11356 김운중

결제담당자: 13523 송희정

거래처번호: 210-20031120

판매처명: 덕산골

판매인: 최윤희

주소: 서울시 동대문구 회기동 33번지

상품번호	상품명	판매량
A212	레종	440
B114	마일드세븐	590
B356	시마	410
C145	에쎬	330

판매량 : 1870개

대금방식: 선불

총판매액: 53만8천원

## 예제 : 운송처리 화면

송장번호 : 20061022-0935    운송일자: 2006/10/30  
거래번호 : 1870211

운송담당자: 13523 송희정  
영업담당자: 11356 김운중  
거래담당자: aabbc 조용덕

거래처번호: 210-20031120  
판매처명: 덕산골  
판매인: 최윤희  
주소: 서울시 동대문구 회기동 33번지

상품번호	상품명	판매량	운송량
A212	레종	440	220
B114	마일드세븐	590	130
B356	시마	410	0
C145	에쎄	330	0

운송량 : 350개

## ☑ 왜 우리는 **ER Diagram**을 그리는가?

1. Organization
2. Function or Process
3. Data
4. Rule or Relationship



**Why Not DFD?**

## ☑ Why Not DFD (Data Flow Diagram) ?

1. DFD는 Process 관점.
2. Data Store는 임의로 만들어짐.
3. Data 사이의 Relationship 이나 Rule 은 보여 주지 않음.
4. Data 사이의 Generalization, Aggregation, Instantiation 등을 보여 주지 않음.



결국 데이터의 중복성이나 관련성을 정확히 설명할 수 없음.

## ☑ Entity- Relationship Diagram의 당위성

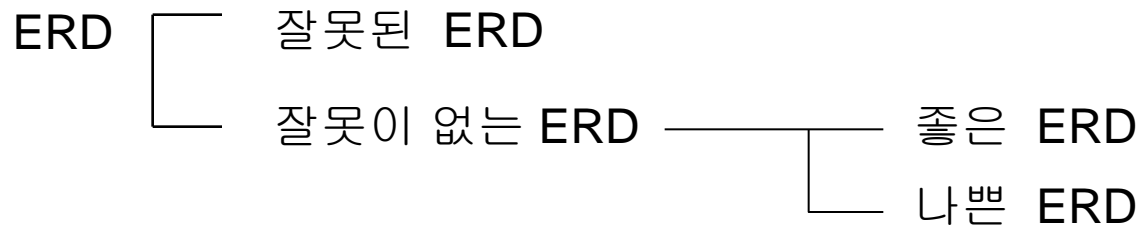
- ✓ 분석가, 개발자, 그리고 사용자 등 모든 정보시스템 관련자에게  
전체적인 데이터 체계를 보여 줄 수 있음.
- ✓ 또한 이와 관련된 자기관점의 데이터 체계를 보여 줄 수 있음.



## ☑ Right ERD, Wrong ERD, Best ERD or Worst ERD

모든 ERD가 똑같은지는 않다 !

그리고, ERD에 대해 어느 정도의 검증은 가능하지만  
완벽한 정답은 있을 수 없다.



# COMPUTERWORLD

## Data modeling: Tough but rewarding

By Johanna Ambrosio

Data modeling is rather like going to the doctor for wellness checkups: You know it is good for you, but you cannot necessarily quantify the benefits.

Although hard numbers are tough to come by, those who have done data modeling say it offers many advantages. Among them are getting data to users in a format that supports their business requirements and cleaning up conflicting and redundant data formats that are part of most large companies' information systems repertoires.

But data modeling is not easy or cheap. It can be a five- to 10-year effort fraught with political battles and difficulties justifying it to top management.

Data modeling, in its essence, is defining what the business does and how, and then figuring out what data is required to support those requirements. This information is then modeled and drives all application development efforts.

Donnette Bruno, a data administrator at Barnett Technologies, Inc. in Jacksonville, Fla., likened data modeling to "cleaning MIS' house." In her last job, at an insurance company, "We found 87 different varieties of region code. Why? Because programmers were doing what they were trained to do: being creative. And they were very good at it," she said.

Alan Kortesoja, a partner at Ernst & Young in Ann Arbor, Mich., said changing business requirements have forced insurance companies into the vanguard of data modeling. "Insurance companies are thinking now in terms of providing services, not in terms of individual policies. But the old batch systems just weren't set up that way."

Most companies, he added, are doing data modeling in fits and starts. That has been the case at Blue Cross/Blue Shield of Minnesota, which began data modeling about 2 1/2 years ago but has been doing it "intensely" for the past year, said Steve Hiller, manager of data resource management. "The

primary benefit has been clarifying terminology to the business people, so we all have one way of communicating with each other."

Mellon Bank Corp. has been involved with data modeling for about three years, according to Jim Stuber, first vice president. "It's been very successful. The quality of applications has improved," he said. The bank measures that quality by how stable applications are once they are in production.

"If you're building relational applications," Stuber said, "you need to be doing data modeling. If you don't, you'll build relational systems the same way you built the old ones."

Bruno said she has heard of returns on investment ranging from 20% to 80% from data modeling. Those numbers are difficult to substantiate, however, in part because of the challenges to data modeling. These include justification and political battles.

Kortesoja said, "The general problem with an architectural group is that it's hard to relate to cost reduction or revenue enhancement," the two primary motivators for any business decision. As a result, it is often difficult to convince management that a data modeling effort is needed.

Still, he said, some firms justify it "on the basis that they are going to lose revenue or miss opportunities" if they do not get their data act together.

Another problem is the political situation. "Systems people are our worst enemies," Bruno said. "They think we're taking their jobs away."

Kortesoja noted that part of the problem is that some data modelers become almost religious in their zeal for perfect data. "Other people look at them as a strange cult that is hooked on data. There can be a certain righteousness about people who are 'into' data," he added.

One way to avoid that, Kortesoja said, is to "help the developers understand that you're doing something for them, too — providing a better understanding of how their application works." For that reason, he

said, some companies assign a data modeler to every applications development team.

### Start where?

A big question at many companies is how to get started. It can be a catch-22 of modeling data for existing applications, which are presumably working and would be a hard sell, vs. a top-down effort that starts with the business needs.

Del Monte Foods in San Francisco struck an approach that may not sit well with data modeling purists. The company is implementing a corporate repository for decision-support applications.

"I've already warned the users that they will have conflicting data formats," said Les Bain, manager of data resources.

"When they start screaming, I'll begin data modeling to clean everything up."

Still, it can be worth the time and effort. "We started in 1985," said John Chatfield, senior principal analyst at Salt River Project in Phoenix. "It's led to rapid systems development and data standardization that has paid big benefits, and now it's pretty much routine."

### What is data modeling?

The process of matching business requirements to the data needed to support them. The resulting models are used to drive future applications development.

### Status

Insurance firms, banks and telecommunications concerns are the primary users of data modeling techniques. In most other companies, the effort is fraught with political and justification battles.

### Benefits

Faster systems development and standardization of data for use by varied applications.

### One user's comment

"If you're building relational applications, you need to be doing data modeling. If you don't, you'll build relational systems the same way you built the old ones."

## Guidelines for Right ERD and Best ERD

## ☑ **Entity** 인가 , **Attribute** 인가 ?

1. One or Many Instances ?
2. Additional Detailed Information ?

## ☑ 왜 우리는 **Is-A Relationship**을 싫어하는가?

## ☑ **Many- To-Many Relationship** ?

1. 숨겨진 **Attribute**를 찾아낼 수 있음.
2. 두개의 **Relationship**을 나타낼 지 모름.
3. Relational Model 은 Many - Many를 지원하지 않음.

## ☑ Associative Entity ?

1. Many - Many Relationship 을 없앨 수 있음.
2. Degree가 3 이상인 Relationship을 없앨 수 있음.
3. 완벽하게 Redundancy를 없앨 수 있음.

## ☑ Derived Attribute, Derived Entity, Derived Relationship ?

1. 자연스러움 (Naturalness)
2. 성능 (Performance)

## ☑ Right ERD or Wrong ERD

- ❑ 이 Relationship이 없어도 Entity와 다른 Relationship 에 의해 표현되는가? 즉, Query 만에 의해서 나타내질 수 있는가?
- ❑ Relationship을 정확히 알 수 있는 방법은 Instance와 Instance를 Check-Up한다.
- ❑ 이 Attribute는 알맞은 위치에 있는가?
- ❑ 이 Entity는 Fundamental Entity인가? 아니면, Derived Entity인가?

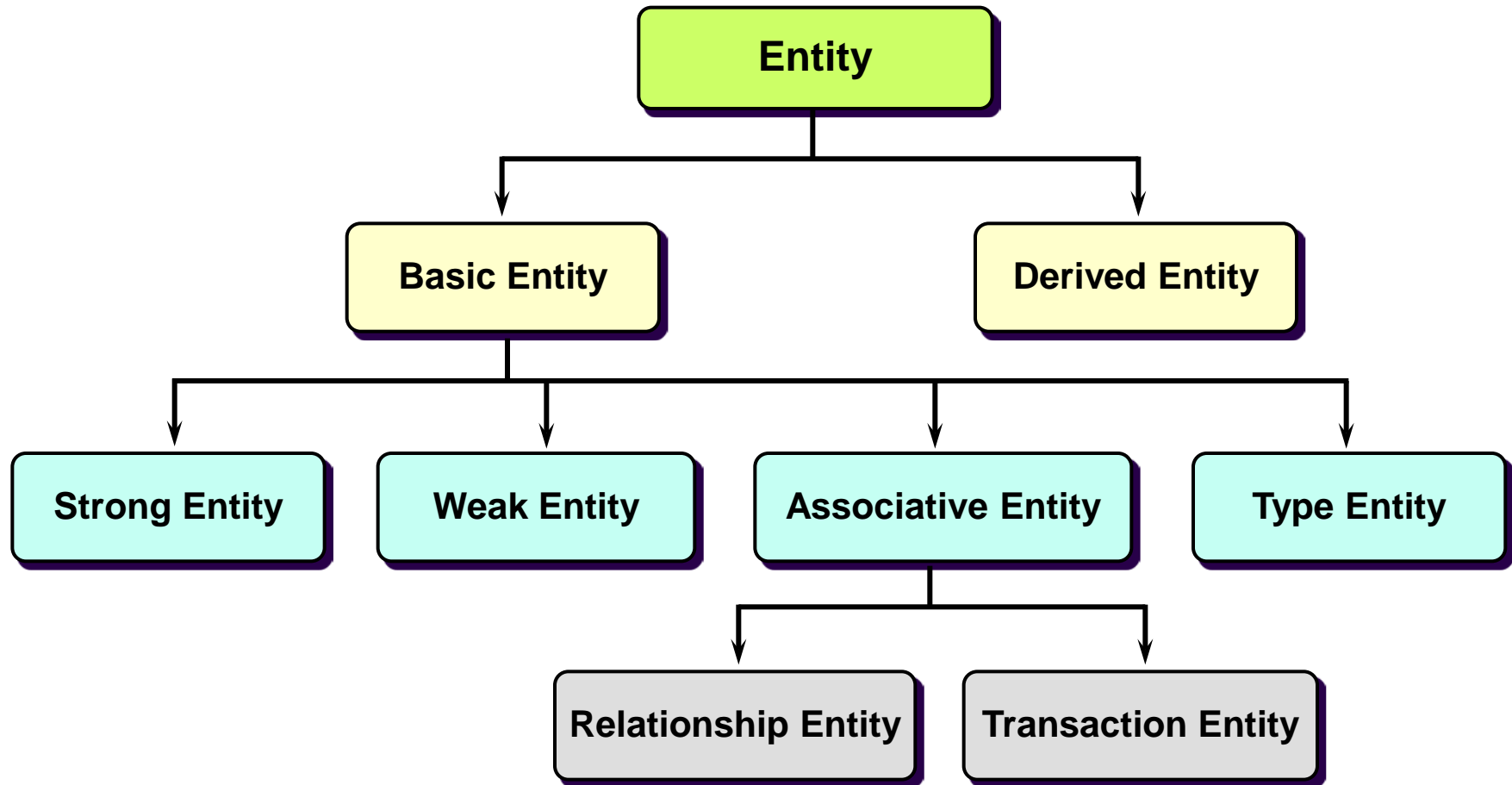
## ☒ **Best ERD or Worst ERD**

- ☐ 이 Relationship은 흥미로운가 ?
- ☐ 만약 이 Relationship이 Redundant 하다면 진정으로 이 Relationship이 불필요한가? 아니면, 다른 Relationship이 불필요한가?
- ☐ 이 Relationship 은 진정으로 알맞은 Entity들과 연결되어 있는가?
- ☐ Business Requirement 의 관점에서 Flexibility, User-Friendliness, Performance, Quality 중 무엇이 더 중요한가?

## ERD Notation



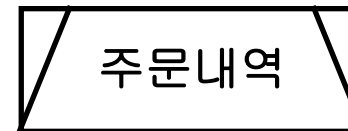
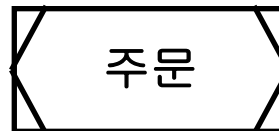
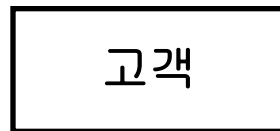
## ☑ Entity 의 분류



□ Entity : **Basic** / Derived , Strong / Weak , Associative/Transaction, Type/Class

## Basic Entity

- ◆ 자체 Entity 내에서 Insert,Update,Delete가 가능한 Entity
- ◆ 주로 Transaction과 관련 있는 데이터



❑ **Entity** : Basic / **Derived** , Strong / Weak , Associative/Transaction, Type/Class

**Derived Entity**

- ◆ 자체 **Entity** 내에서 **Insert, Update, Delete**가 불가능한 **Entity**
- ◆ 주로 정보계 or **Data Warehouse**와 관련 있는 데이터

+ 일별 주식보유 현황

+ 고객별 보험료 잔액

❑ **Entity** : Basic / Derived , **Strong** / Weak , Associative/Transaction, Type/Class

**Strong Entity**

◆ 가장 일반적인 **Entity**

◆ **Weak Entity**의 **Parent Entity**

직원

주문

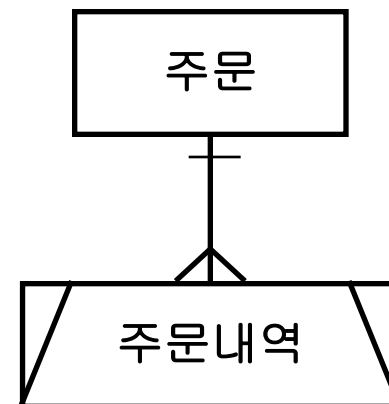
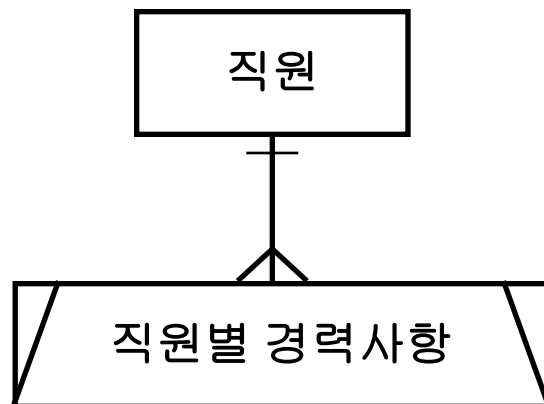
부서

□ **Entity** : Basic / Derived , Strong / Weak , Associative/Transaction, Type/Class

## Weak Entity

◆ **Strong Entity**에 강하게 종속된 **Entity**

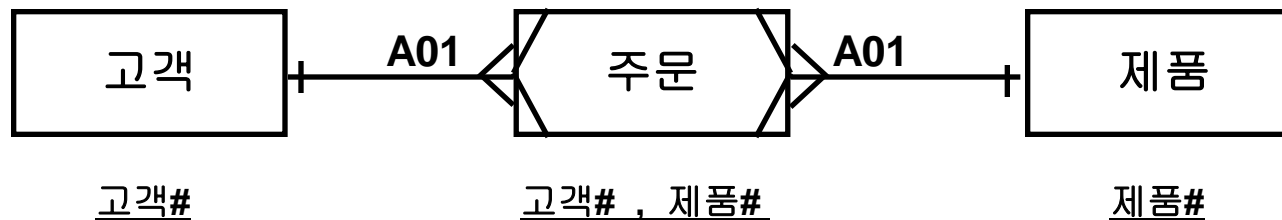
◆ 반드시 **Strong Entity**의 **Key**를 자신의 **Key**로 함



□ **Entity** : Basic / Derived , Strong / Weak , **Associative**/Transaction, Type/Class

## Associative Entity

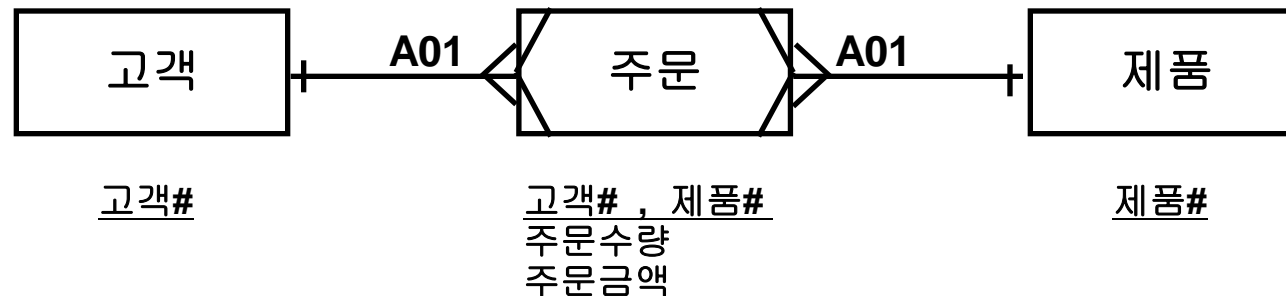
- ◆ 둘 이상의 **Entity** 사이에 관계를 데이터화 하고자 함
- ◆ **M:N Relationship**의 **Resolution**
- ◆ 반드시 2개 이상의 **Composite Key** 를 이룸



□ **Entity** : Basic / Derived , Strong / Weak , Associative/Transaction, Type/Class

## Transaction Entity

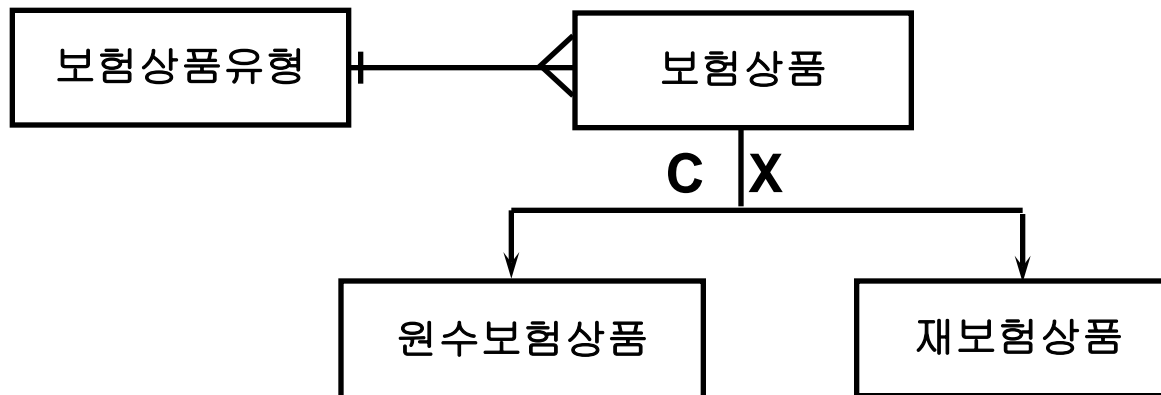
- ◆ Associative Entity의 한 유형
- ◆ M:N Relationship의 Resolution
- ◆ Key 속성 이외의 **Non-Key** 속성이 존재



□ **Entity** : Basic / Derived , Strong / Weak , Associative/Transaction, **Type**/Class

## Type Entity

- ◆ 하나의 **Entity**를 분류하는 **Supertype/Subtype** 들을 의미함
- ◆ **Physical Design**시에는 **Type Entity**들의 통합이 필요

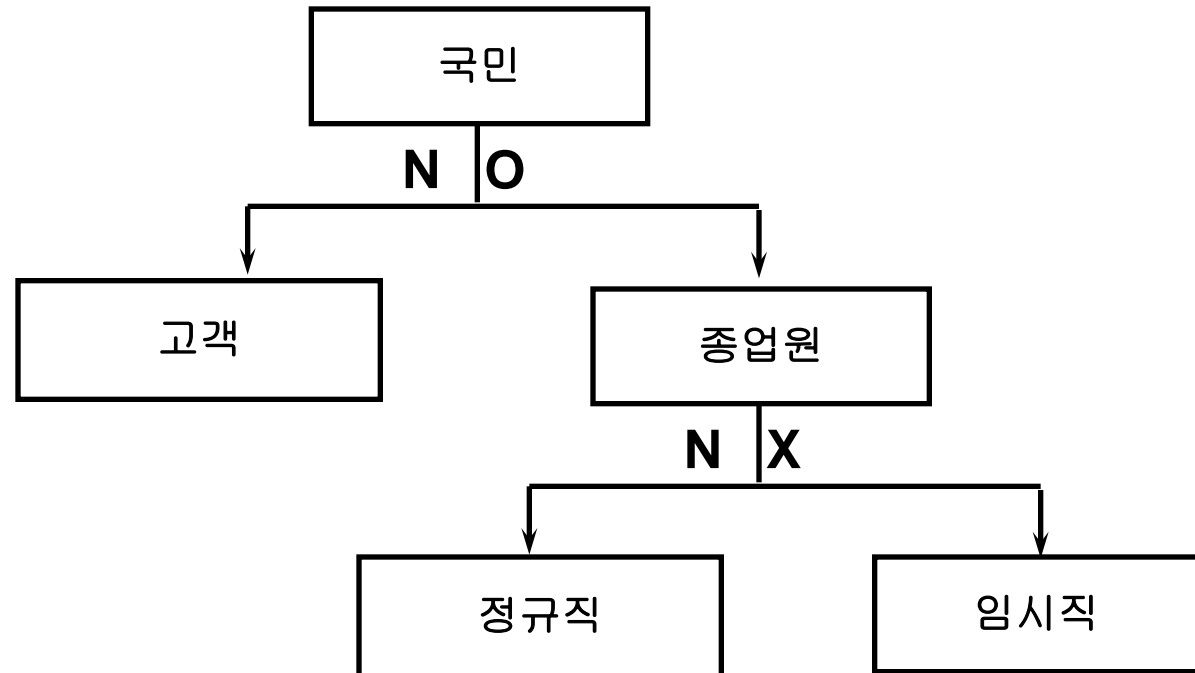




□ **Entity** : Basic / Derived , Strong / Weak , Associative/Transaction, Type/**Class**

## Class Entity

- ◆ 여러 개의 **Entity**를 분류하는 **Superclass/Subclass** 들에 의미함.
- ◆ **Logical Design**시에는 다양한 **Entity**로 표현됨



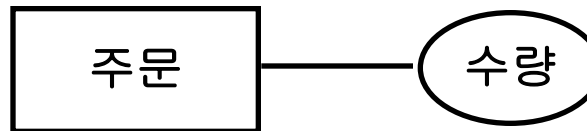
## □ **Attribute** : Basic or Derived

- Candidate Key
- Primary Key
- Alternate Key
- Foreign Key
- Composite Attribute
- Secondary Key

## □ Attribute : Basic or Derived

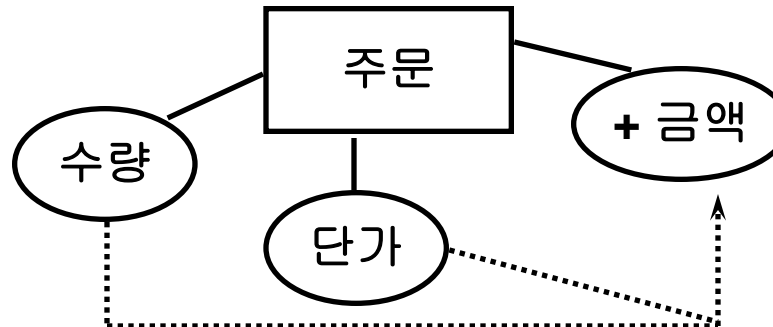
### Basic Attribute

- ◆ 자체 내에서 Insert, Update, Delete 가 가능한 Attribute



### Derived Attribute

- ◆ 자체 내에서 Insert, Update, Delete 가 불가능한 Attribute



## □ Attribute : Candidate Key

Candidate Key

◆ PK 가 될 수 있는 2개 이상의 Attribute

사번

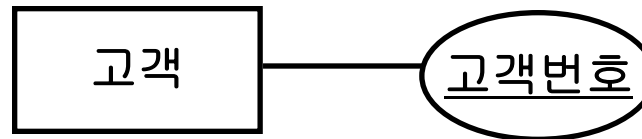
주민번호

의료보험번호

## □ Attribute : Primary Key

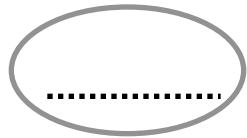


- ◆ Entity의 식별자로서 유일성(Uniqueness) 보장
- ◆ PK는 하나 이상의 Attribute로 구성

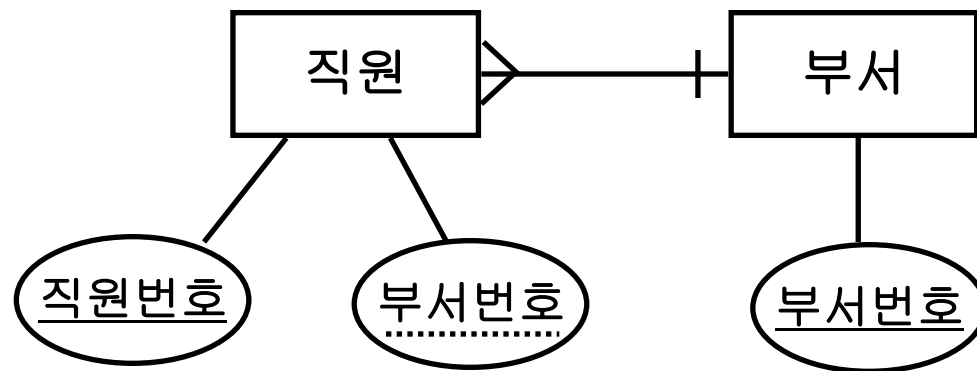


## □ Attribute : Foreign Key

### Foreign Key



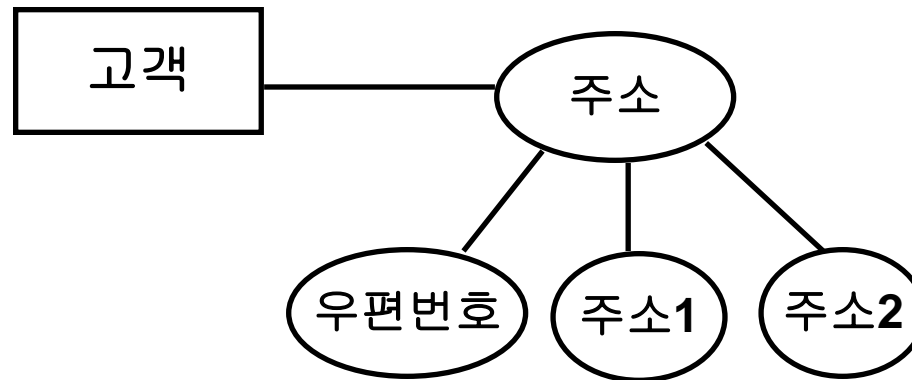
- ◆ Entity 사이의 Relationship을 표현
- ◆ FK는 반드시 다른 Entity의 PK이어야 함
- ◆ 참조 무결성(Referential Integrity)을 보장
- ◆ Entity 사이의 참조경로를 알 수 있음



## ❑ Attribute : Composite Attribute

Composite Attribute

◆ Attribute가 여러 개의 속성(Attribute)으로 나뉘지는 경우



## □ Attribute : Secondary Key

Secondary Key

- ◆ **Uniqueness** 는 보장하지 않지만 중요한 관리 속성일 경우
- ◆ **Index Key** 선정 시 후보가 되는 **Attribute (Non-Unique Index)**

고객명

주소



## ❑ **Relationship** : Basic or Derived

### ➤ Connectivity

❶ 1:1

❷ M:N

❸ 1:M

### ➤ Membership

❶ Optional

❷ Mandatory

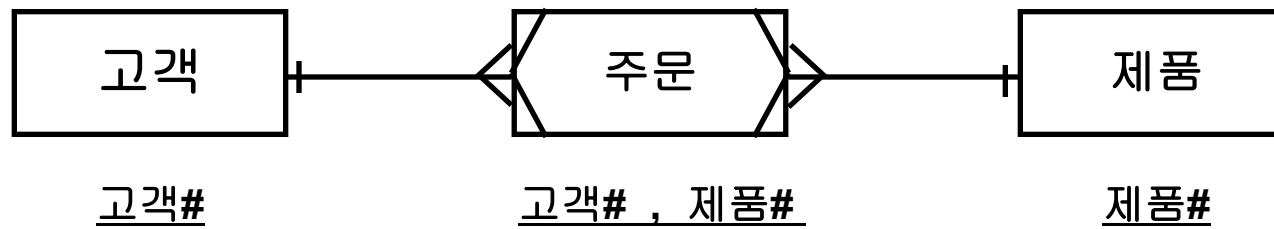
❸ Optional Becoming Mandatory

### ➤ Role Name

## ❑ Relationship : Basic or Derived

### Basic Relationship

- ◆ Basic Entity 사이에 존재하는 Relationship
- ◆ Relationship이 존재한다는 것은 Foreign Key가 존재한다는 의미

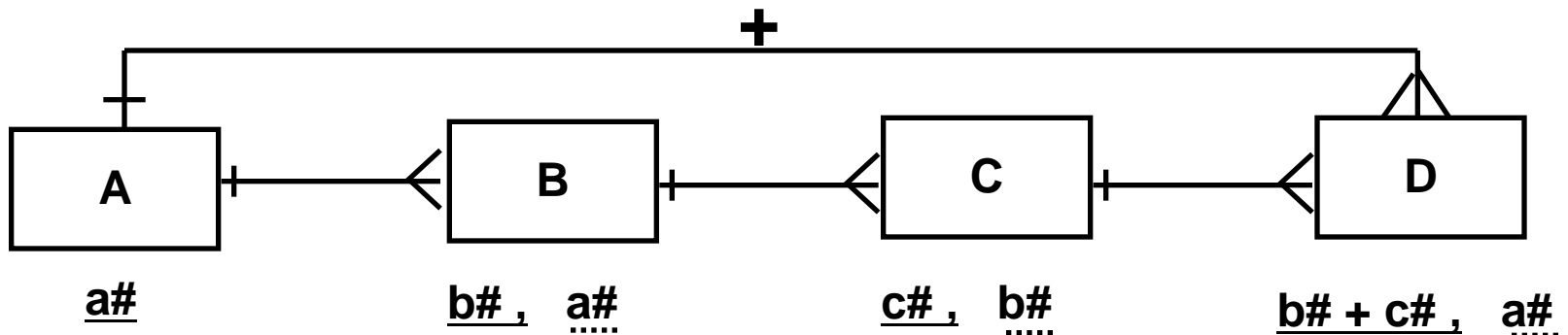


## ❑ Relationship : Basic or Derived

### Derived Relationship

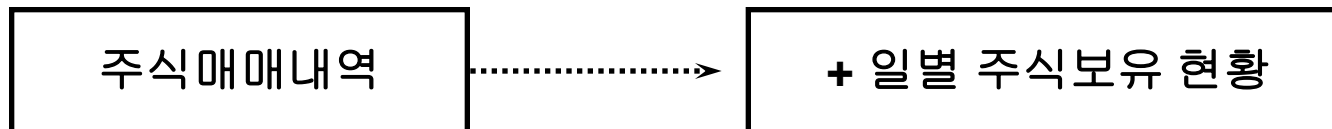
◆ Performance 를 고려한 Relationship

◆ Physical ERD 에서 나타나는 Relationship으로서 빠른 참조경로



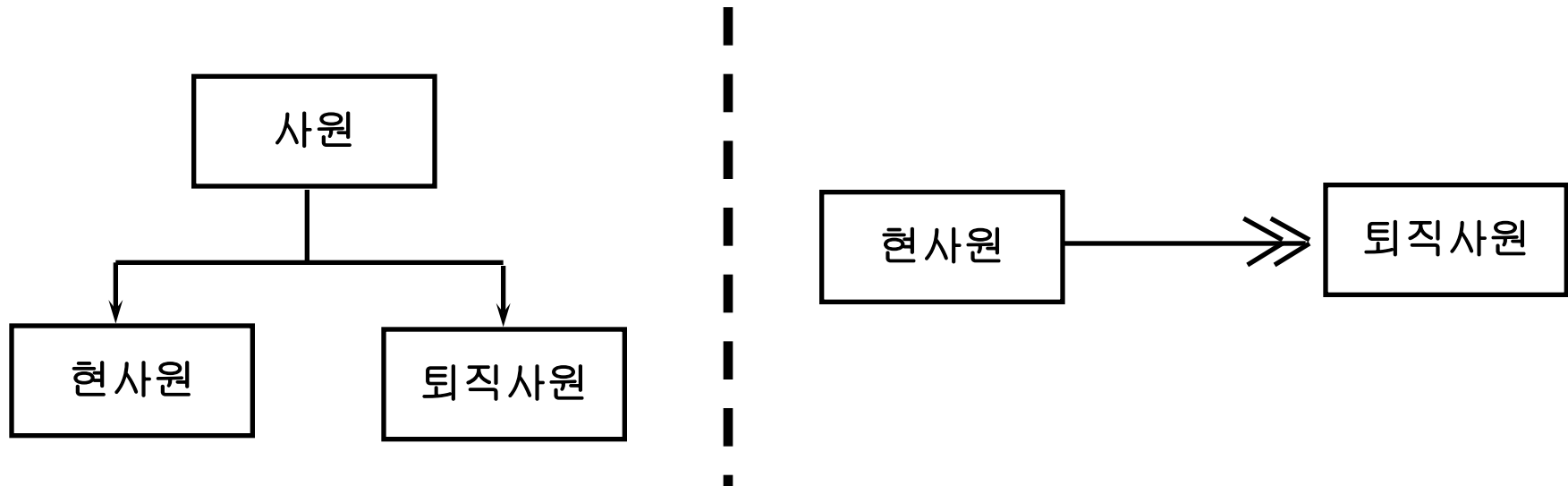
## □ Derivation

- ◆ Basic Entity 와 Derived Entity 사이에 존재
- ◆ Foreign Key가 존재하지 않음

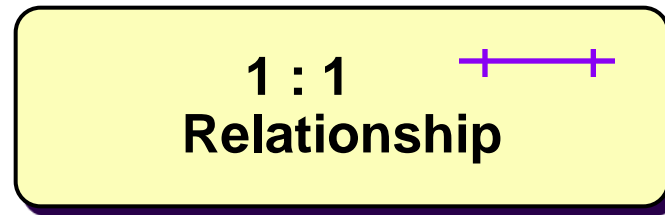


## □ Move

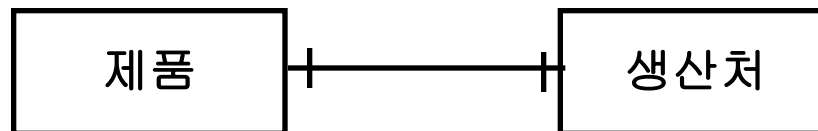
- ◆ Basic Entity 와 Basic Entity 사이에 존재
- ◆ Foreign Key가 존재하지 않음
- ◆ Class/Subclass 의 Physical Resolution



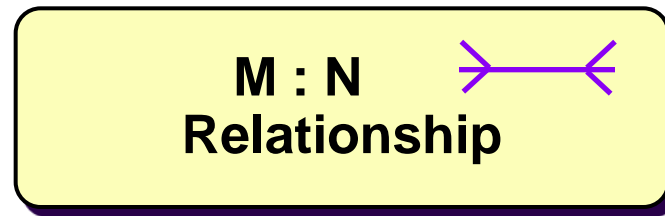
## ❑ Relationship : Connectivity



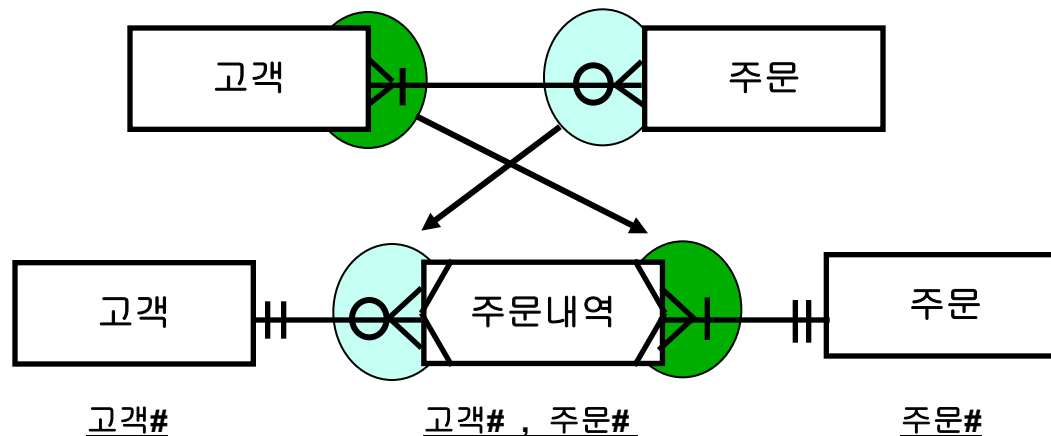
- ◆ 관리의 관점에서 또 하나의 **Entity**를 가지는 경우
- ◆ 하나의 **Entity**일수 도 있다는 의심이 필요.
- ◆ 별로 나타나지 않는 **Relationship**



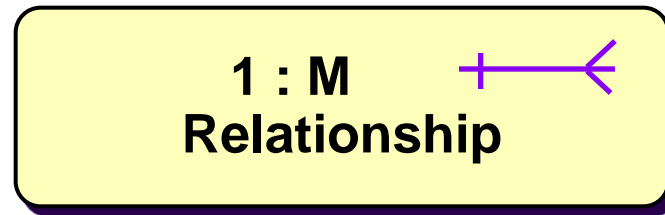
## ❑ Relationship : Connectivity



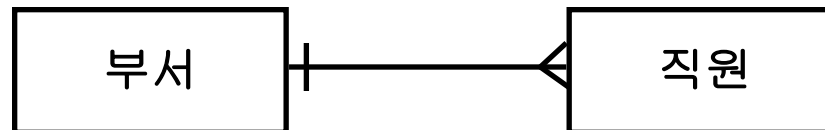
- ◆ 2개의 1:M Relationship로 Resolution
- ◆ 숨어 있는 속성을 놓치게 되는 Relationship
- ◆ 잘못된 ERD일 수 있는 Relationship -> Associative Entity



## ❑ Relationship : Connectivity



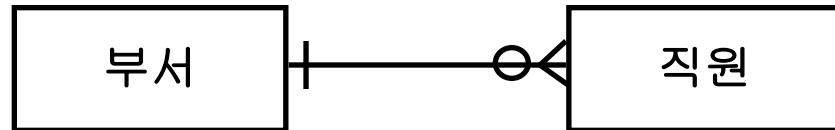
- ◆ ERD 에서 가장 많이 나타나는 Relationship
- ◆ 반드시 **Many** 쪽의 **Entity**는 상대 **Entity**의 **Key**를 Foreign Key.



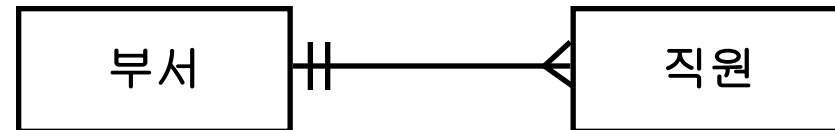


## ❑ Relationship : Membership

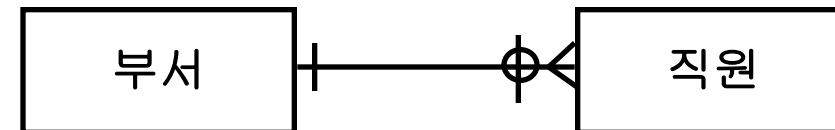
**Optional**



**Mandatory**

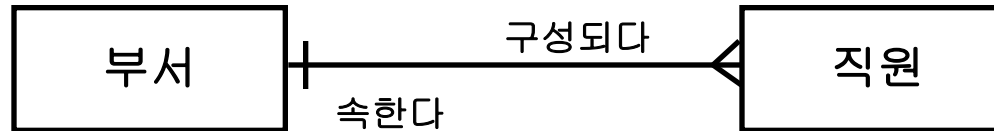


**Optional Becoming  
Mandatory**



## □ Relationship : Role Name

- ◆ Relationship의 역할을 **Comment** 로 표현
- ◆ **Physical Level**에서는 큰 의미가 없음
- ◆ 현실적 제한

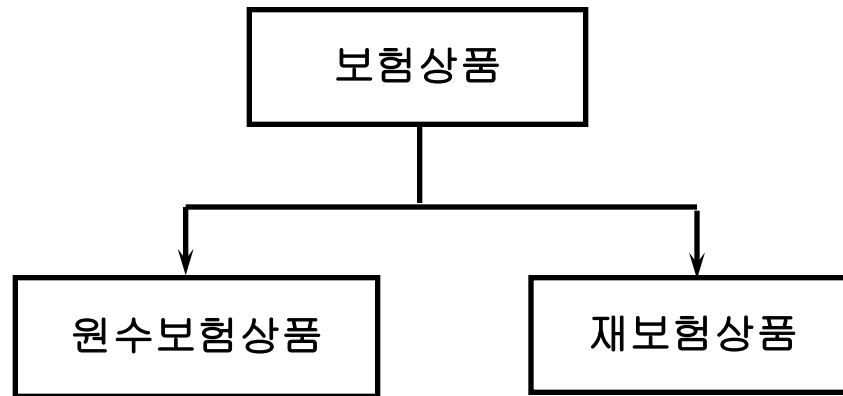


## ❑ **Class / Subclass** : Tree or Box Representation

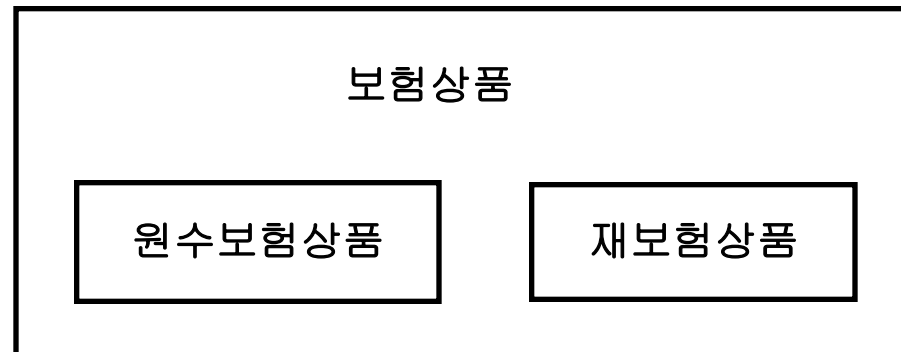
- Covering / Non Covering
- Exclusive / Non Exclusive

## ❑ Class / Subclass : Tree or Box Representation

**Tree  
Representation**



**Box  
Representation**



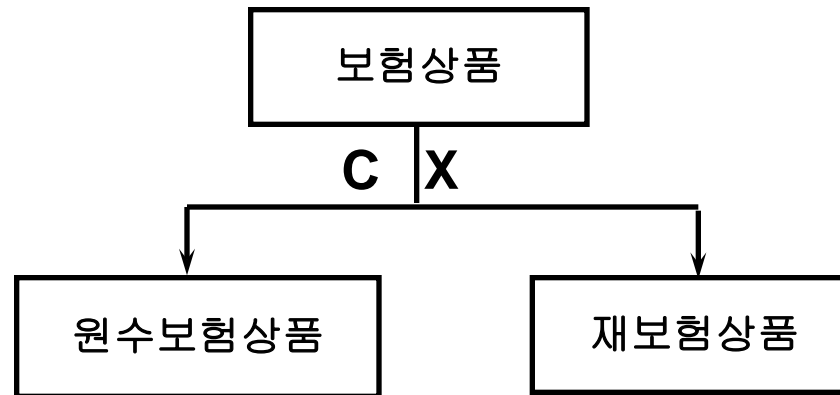
## ❑ Class / Subclass : Covering & Exclusive

Covering /  
Non Covering

◆ 더 이상의 Subclass가 없음/있음 ( C / N )

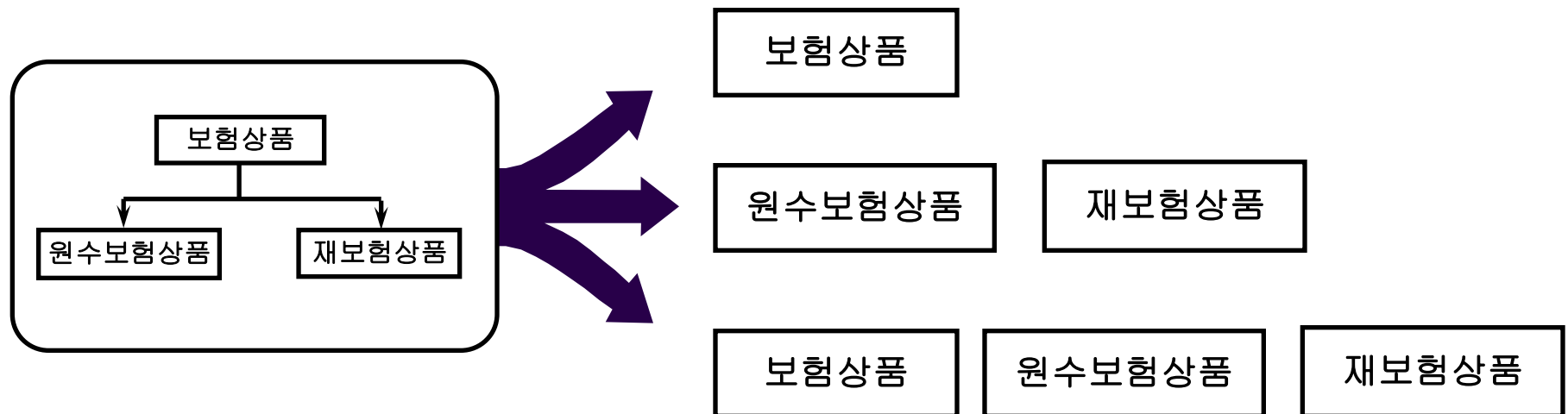
Exclusive /  
Non Exclusive

◆ Subclass를 동시에 불가능/가능 ( X / N )



## □ Class / Subclass : Physical Level

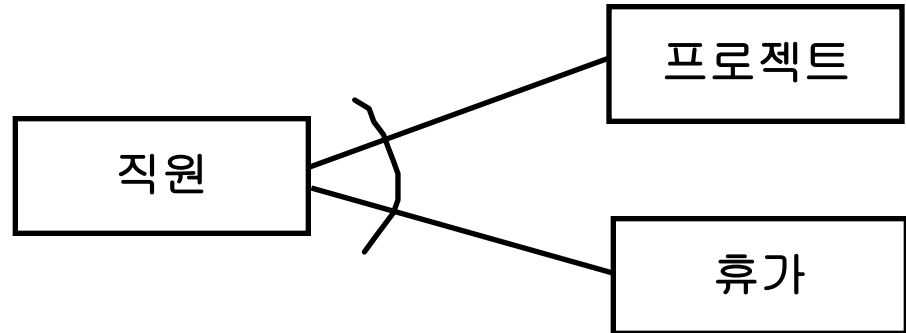
- ◆ One Table or
- ◆ Two Table or
- ◆ Three Table



## ❑ Exclusive Relationship or Instance

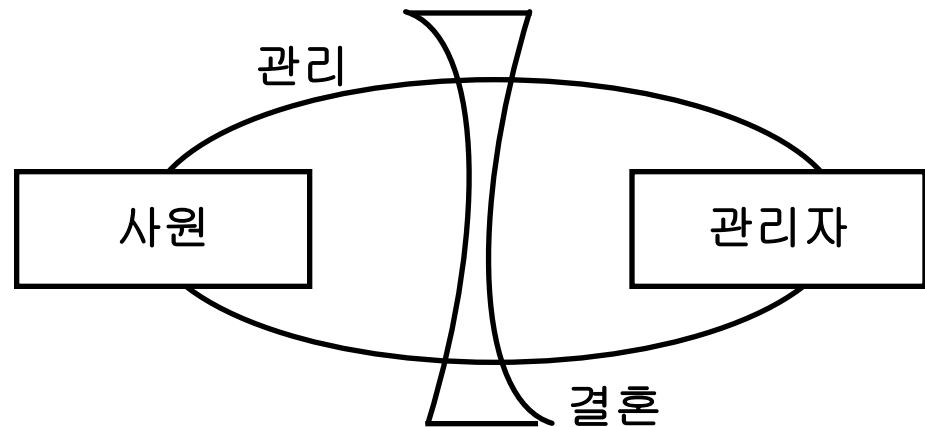
### Exclusive Relationship

- ◆ 직원은 프로젝트와 휴가를 동시에 행할 수 없다.



### Exclusive Instance

- ◆ 관리자는 부하사원과 결혼 할 수 있지만, 자기 부하사원과는 결혼 할 수 없다.



## □ Exclusive Relationship - Tip

✓ **Exclusive RLSP**은 **Child Entity**의 **RLSP**에서만 적용된다.

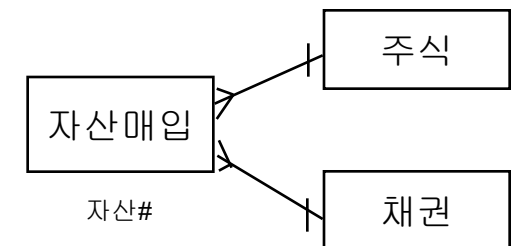
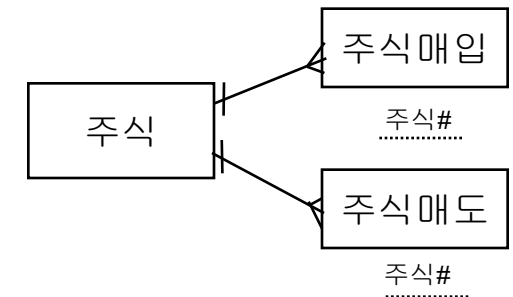
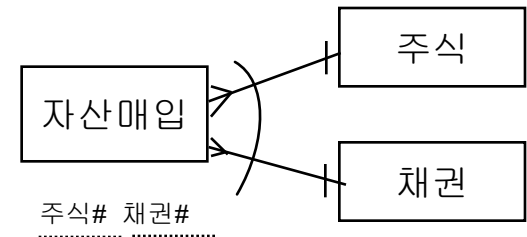
즉 **Child Entity**의 **Instance**가 다른 2개 이상의 **Parent Entity**의 **Instance**와 상호배타적(**Exclusive**)일 때 만 적용된다.

✓ **Parent Entity**에서 나타나는 **Exclusive**한 관계는 **Exclusive RLSP**이 아님.

즉 **Parent Entity**의 **Instance**가 다른 2개 이상의 **Child Entity**의 **Instance**와 상호배타적(**Exclusive**)일 때는 적용되지 않는다.

✓ **Exclusive RLSP**의 경우 **Child Entity**의 **Foreign Key**는 **RLSP**수와 동일해야 한다.

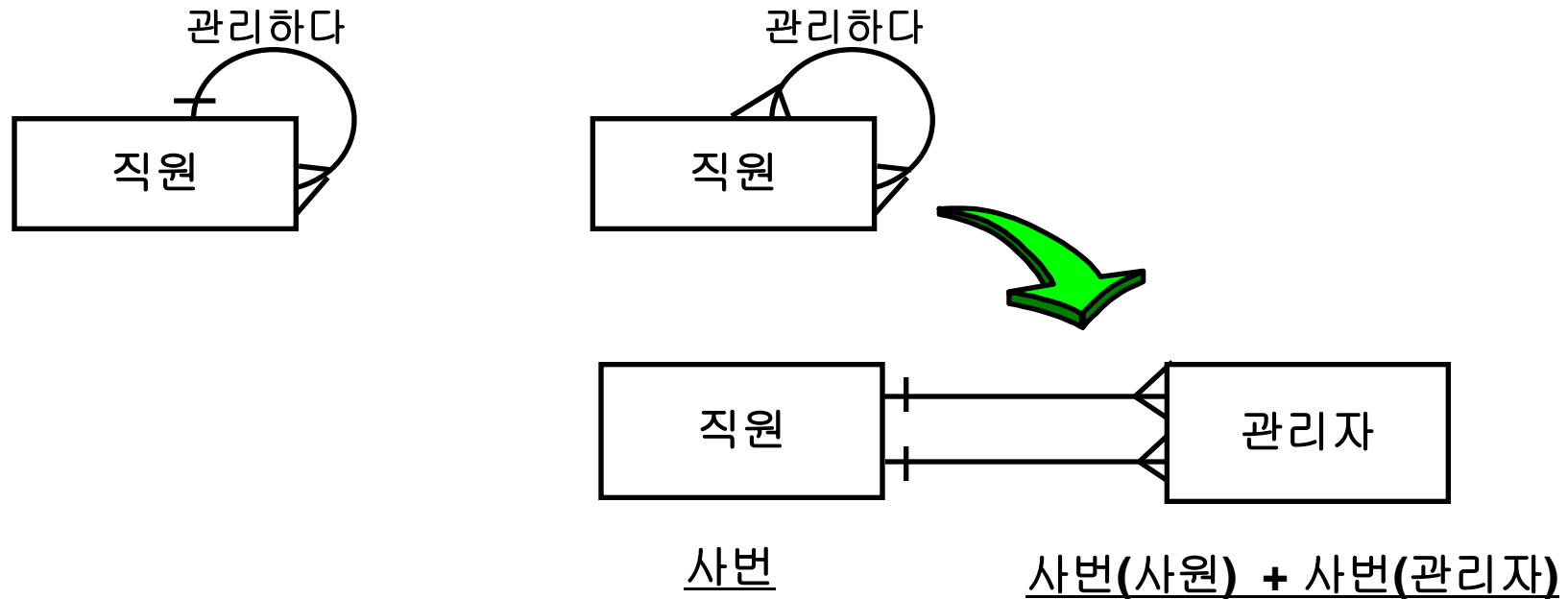
즉 여러 개의 **RLSP**을 하나의 **Attribute**로 나타내는 것은 **Exclusive RLSP**이 아님.





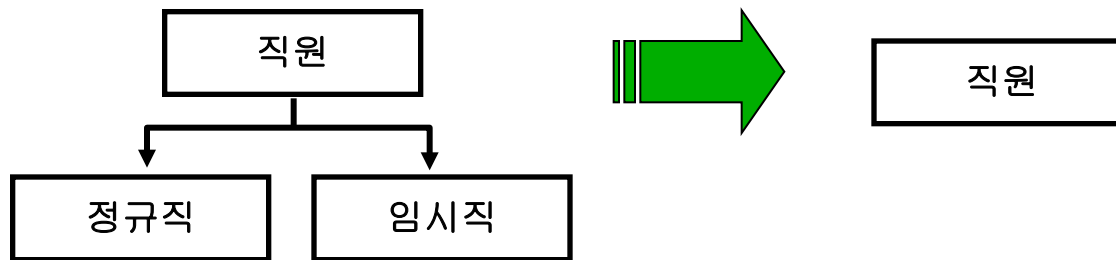
## □ Recursive Relationship

- ◆ Entity내의 Instance사이에 일어나는 Relationship
- ◆ M : N Relationship 이라면 Resolution 이 필요

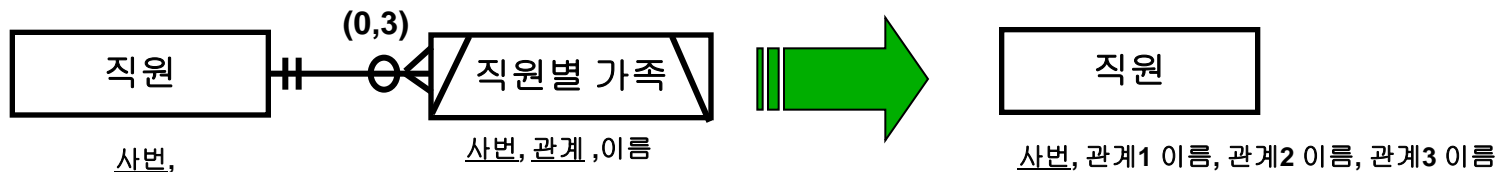


## □ Clustering & Attribution-Physical Level

### ◆ Clustering : SubClass Entity Merge



### ◆ Attribution : Weak Entity를 Strong Entity의 Attribute화 하는 것

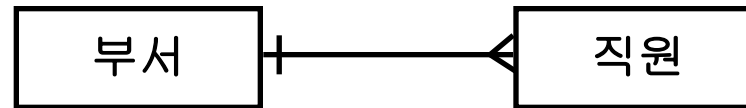


## ❑ **Constraint / Rule : Physical Level**

- Non FK
- FK
- F(D)
- Trigger

## □ Constraint / Rule : Physical Level

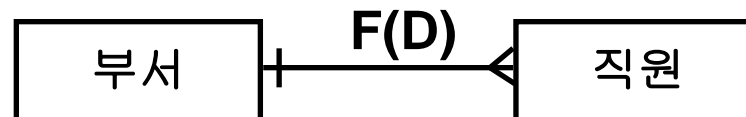
Non FK



FK



FK - Delete on Cascade



Trigger



## ☑ 일반적인 데이터 모델링 방법론

### One..

1. Conceptual Data Modeling
2. Logical Data Modeling
3. Physical DB Design

### Two..

1. View Design
2. View Integration
3. Database Construction
4. DBMS Implementation

- 하향식(Top-Down) 접근 방법과 상향식(Bottom-Up) 접근 방법론
- Top-Down Approach : 기업의 전략,기능,주제영역으로부터 도출되는 데이터 모델
- Bottom-Up Approach : 장표,화면 등의 업무처리 양식으로부터 도출되는 데이터 모델

구분	Top-Down Approach	Bottom-Up Approach
분석의 기점	전체	부분 또는 단위
분석의 종착	부분 또는 단위	전체
중심 분석대상	기업의 사명, 목표, 전략,	일상 데이터의 속성
분석의 범위	전사적	전사적
속성의 정확성	모호하다.	구체적이고 정확하다.
전체구조와 틀	전체의 틀 중심	전체보다는 부분 중심
전사구조 왜곡의 정도	심하지 않다.	심하다.
통합의 관점	강하다.	약하다.
새로운 정보요구	나타난다.	없다.

## □ Top-Down & Bottom-Up Approach

### ❖ Top-Down Approach의 특징

- ➡ 하향식 접근법의 기본철학은 먼저 전체의 구조와 틀을 만든 후, 이를 구체화하고 세분화 하는 모델링 과정.
- ➡ 시스템기획단계(ISP : Information Strategy Planning 등)에서 정의된 기업의 사명(Mission), 목표(Objective), 전략(Strategy), 주요성공요인(Critical Success Factor), 문제점(Problem) 등을 분석한 후 이를 구체적인 주제 영역 또는 업무영역으로 세분화하여 분석하여 모델링이 이루어짐.
- ➡ 하향식 접근법은 해당 산업의 선진 모델 템플리트(Best Practice)를 벤치마킹하여 To Be 모델의 관점을 적용.
- ➡ 하향식 접근법에서는 위와 같은 기업의 전략적 요소들을 토대로 전사적 구조와 틀을 규정한 뒤 몇 개의 단위 영역으로 구분하여, 이 단위 영역별로 점차 자세하고 구체적인 분석을 하게 됨.
- ➡ 하향식 데이터모델링 접근법에서는 마치 나무에서 큰 줄기가 먼저 뻗은 후 잔 가지가 생기는 경우나 전화선을 깔 때 먼저 주 선(Back-bone)을 먼저 깔고 각 가정으로의 지선을 내리듯이 구조와 틀이 확정된 후 하위의 구체적인 영역으로 확장해 나가는 방식이라고 할 수 있음

## □ Top-Down & Bottom-Up Approach

### ❖ Bottom-Up Approach의 특징

- ➡ 하향식 접근법에 반해 상향식 접근법은 부분으로부터 전체를 이루어 가는 것이 기본 철학.
- ➡ 데이터모델링 방법에 있어서는 우선 기업 내에서 사용되어지거나 운용되고 있는 장표나 문서(Documentation)의 수집, 혹은 기존 시스템의 분석을 통하여 기업 내의 사용자들이 업무 일상에서 취급하는 구체적이고 세부적인 필요정보를 먼저 추출한 후, 이를 점차 전체 모델로 통합해 나아가는 과정임.
- ➡ 상향식 접근법에서는 하향식접근에서 놓치기 쉬운 데이터의 속성(Attribute)들을 추출하는 데 유용.