



데이터 모델링

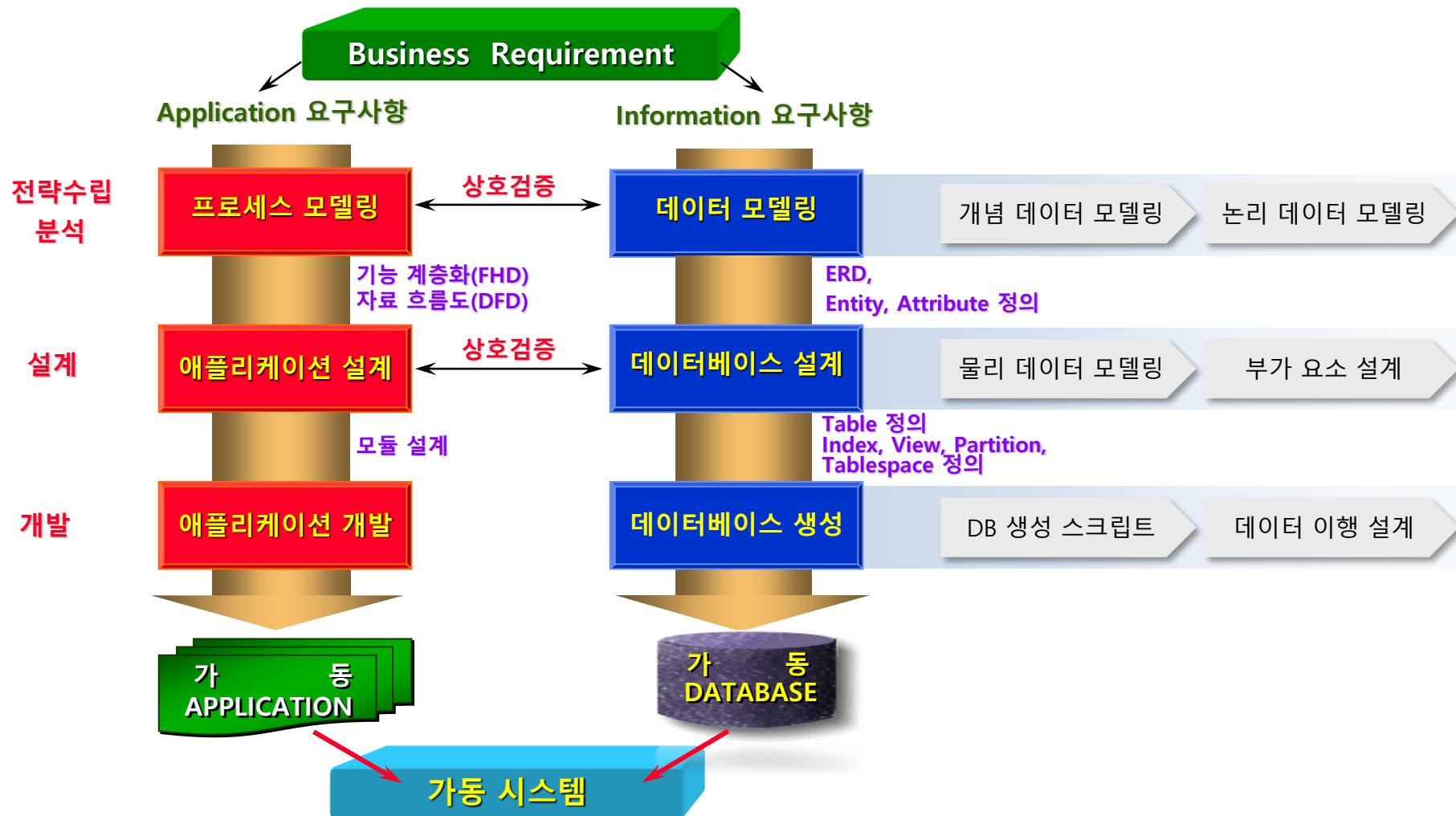
1장. 데이터 모델링 이해

1절. 데이터 모델링 개요

2절. 데이터 모델 구성요소

3절. 데이터 모델링 표기법 이해

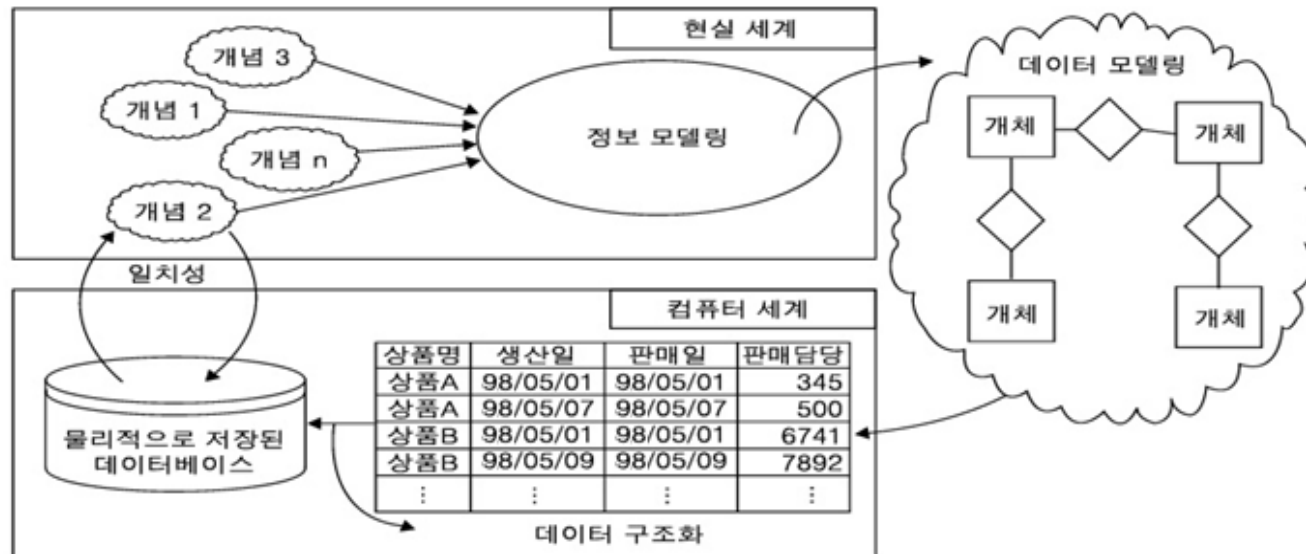
1-1 시스템 구축 절차



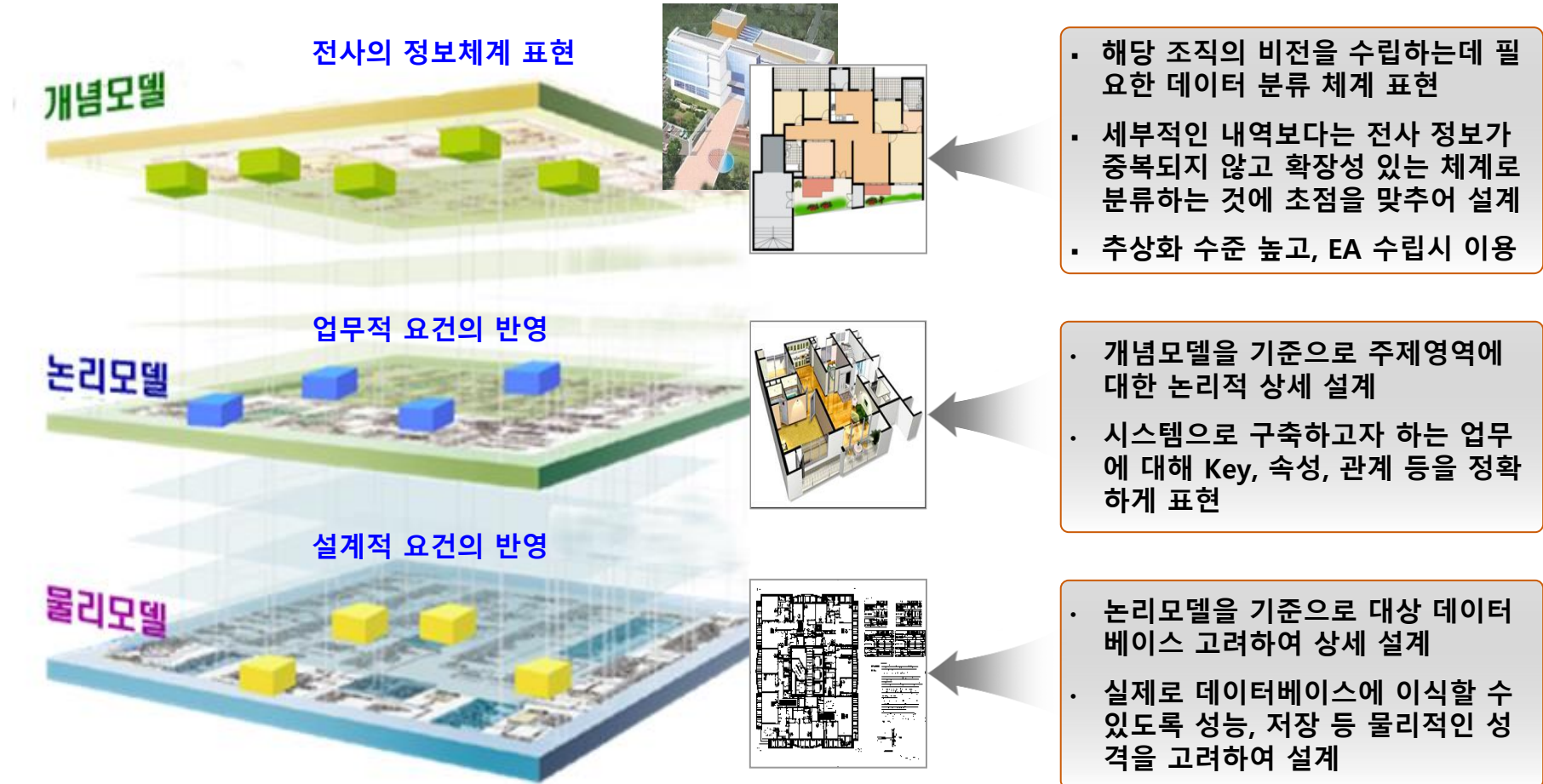
1-2 데이터 모델링 정의

❖ 데이터 모델링(Data Modeling)

- ✓ 복잡한 현실 세계를 단순화하여 현실 세계에 존재하는 개체들을 식별하고 이들 객체간에 관계를 정의함으로써 업무 데이터를 조직화하고 구조화하는 과정이며, 이 과정의 산출물이 데이터 모델이다.
- ✓ 데이터 모델링은 현재 업무를 파악하여 문제점을 인식하고 개선 사항을 도출하여 미래에 적합한 설계를 이끌어 내기 위해 인간이 해야 할 대부분의 결정들을 내리는 단계까지를 모두 포함한다.



1-3 데이터 모델링 단계



1-4 좋은 데이터 모델의 요소

❖ 완전성(Completeness)

- ✓ 업무에서 필요로 하는 모든 데이터가 데이터 모델에 정의되어 있어야 한다.

❖ 중복 배제(Non-Redundancy)

- ✓ 하나의 데이터베이스 내에 동일한 사실은 반드시 한 번만 기록하여야 한다. 데이터 중복으로 인해 저장 공간의 낭비, 중복 관리되고 있는 데이터의 일관성을 유지하기 위한 추가적인 데이터 조작 등이 대표적으로 낭비되고 있는 관리 비용이라고 볼 수 있다.

❖ 비즈니스 룰(Business Rules)

- ✓ 데이터 모델링 과정에서 도출되어 지고 규명되어지는 수많은 업무 규칙(Business Rules)을 데이터 모델에 표현하고 이를 해당 데이터 모델을 활용하는 모든 사용자가 그 규칙을 공유할 수 있게 제공하는 것이 중요하다. 이렇게 함으로써 모든 사용자(개발자, 관리자 등)가 해당 규칙에 대해 동일한 판단을 하고 데이터를 조작할 수 있게 되는 것이다.

1-4 좋은 데이터 모델의 요소

❖ 데이터 재사용(Data Reusability)

- ✓ 데이터의 재사용성을 향상시키고자 한다면 데이터의 통합성과 독립성에 대해서 충분히 고려해야 한다.
- ✓ 회사 전체 관점에서 공통 데이터를 도출하고 이를 전 영역에서 사용하기에 적절한 형태로 설계하여 시스템을 구축하는 통합 모델은 데이터 재사용성을 향상시킬 수 있다.
- ✓ 또한 과거 데이터 모델은 애플리케이션의 부속품 정도로 인식되어 데이터는 프로세스의 흐름에 따라 관리되었고, 데이터 중복이 많이 발생하고 데이터의 일관성 문제가 심각하게 초래되었다. 데이터가 애플리케이션에 대해 독립적으로 설계되어야만 데이터 재사용성을 향상시킬 수 있다.

❖ 안정성 및 확장성(Stability and Flexibility)

- ✓ 정보 시스템은 비즈니스의 변화에 대해 최적으로 적응하도록 끊임없이 요구 받고 있다. 현재의 데이터 구조를 거의 변화하지 않고도 또는 아주 적은 확장을 통해 이러한 변화에 대응하는 데이터 구조도 있을 것이다. 하지만 아주 많은 변화를 주어야만 한다면 변화의 대상이 되는 부분뿐만 아니라 정보시스템의 나머지 부분들도 많은 영향을 받게 될 것이다.
- ✓ 근래의 패키지 시스템들은 확장성을 강조하기 위해서 많은 부분을 통합한 데이터 모델 형태를 가지고 있는 것을 볼 수 있다. 특히 정보시스템에서의 '행위의 주체'가 되는 집합의 통합, '행위의 대상'이 되는 집합의 통합, '행위 자체'에 대한 통합 등은 전체 정보시스템의 안정성, 확장성을 좌우하는 가장 중요한 요소라 할 수 있다.

2-1 관계형 모델 이론

❖ Relational Theory 정의

- ✓ 관계형 모델은 1970년에 에드커 커드(Edgar Codd)에 의해서 최초로 제안되고 체계화되었으며, 비즈니스 데이터를 인식, 구성하고, 조작하는 잘 정돈되고, 예견가능하며, 직관적인 접근 방식이다.
- ✓ 관계형 모델은 데이터가 사용자에게 어떻게 보여지며, 사용자는 데이터에 어떤 조작을 하며, 데이터가 운영될 때 어떻게 유지되어야 하는가에 대한 템플릿이다

❖ 데이터베이스 구성 3요소

- ✓ 데이터 구조(Data Structure)
 - 사용자가 데이터를 인식하는 2차원의 관계테이블
- ✓ 데이터 조작(Data Manipulation)
 - 사용자가 관계테이블에 행하는 일련의 처리형태
- ✓ 데이터 무결성(Data Integrity)
 - 관계테이블에 존재하는 데이터의 정확성과 일관성을 유지하기 위한 업무규칙

2-1 관계형 모델 이론

❖ 데이터 구조(Data Structure)

사용자가 데이터를 인식하는 구성.

“6가지 특성을 가진 행(Row)과 열(Column)로 구성된 2차원의 관계테이블로 표현”

- ✓ 각 열은 하나의 값만을 가진다.(Entries in Columns are single-valued) – 1차 정규화
- ✓ 각 열은 동일한 성격의 값을 가진다.(Entries in Columns are of the same kind) – Same Domain
- ✓ 각 행은 일련의 열의 값에 의해 유일하게 식별된다.(Each row is unique) – Primary Key
- ✓ 열의 순서는 의미가 없다.(Sequence of columns(left to right) is insignificant)
- ✓ 행의 순서는 의미가 없다.(Sequence of rows(top to bottom) is insignificant)
- ✓ 각 열은 유일한 이름을 가진다.(Each column has a unique name)

2-1 관계형 모델 이론

❖ 데이터 조작(Data Manipulation)

사용자가 관계테이블에 행하는 일련의 처리형태.

- ✓ **SET 처리** (set at a time, not one record at a time)
- ✓ **관계 연산자** : 조회
 - Select(or Restrict) : 열(Column)에 의거한 행(Row)의 Subset
 - Project : 열(Column) 의 subset
 - Product : 두 관계 테이블간 행(Row)의 조합을 묶음
 - Join : 열(Column)의 기준에 의거하여 각 행(Row)을 수평적으로 묶음
 - Union : 중복을 없이하여 각 행(Row)을 수직적으로 묶음
 - Intersection : 관계 테이블간의 공통된 행(Row)
 - Difference : 하나의 관계 테이블에만 있는 행(Row)
 - Division : 다른 관계 테이블의 모든 행에 대응하는 열을 제외한 열
- ✓ **처리 연산자** : 관계 테이블의 내용에 변화
 - Insert : 행의 입력
 - Update : 행의 수정
 - Delete : 행의 삭제

2-1 관계형 모델 이론

❖ 데이터 무결성

✓ 실체 무결성 규칙(entity integrity rule)

Primary Key는 Not Null이며, Unique(행 Row)은 하나 이상의 컬럼에 의해서 유일하게 식별되어야 한다.

- **Not NULL** : 값을 가져야 한다.
- **Unique** : 유니크(식별가능)해야 한다.
- **Minimal Set**
- 값의 변화가 없어야 한다.

2-1 관계형 모델 이론

❖ 데이터 무결성

✓ 참조 무결성 규칙(referential integrity rule)

관계 테이블의 모든 외부 식별자 값은 연관되어 있는 관계 테이블에 주식별자 값이 존재해야 한다.

• 입력규칙(Insert Rule)

부모와 자식 테이블 중에서 자식 테이블에 자료를 입력할 때 외부키(Foreign Key)가 되는 컬럼의 값이 부모 테이블에 존재하느냐 안 하느냐를 따져서 자식 테이블에 입력을 허용하거나 안 하거나 하는 규칙

• 삭제규칙>Delete Rule)

부모와 자식 테이블 중에서 부모 테이블에 자료를 삭제할 때 주키(Primary Key)로 선언된 컬럼의 값이 자식 테이블에서 사용되고 있는지를 따져서 부모 테이블 자료의 삭제를 허용하거나 안 하거나 하는 규칙

2-1 관계형 모델 이론

❖ 데이터 무결성

✓ 영역(속성) 무결성 규칙(domain integrity rule)

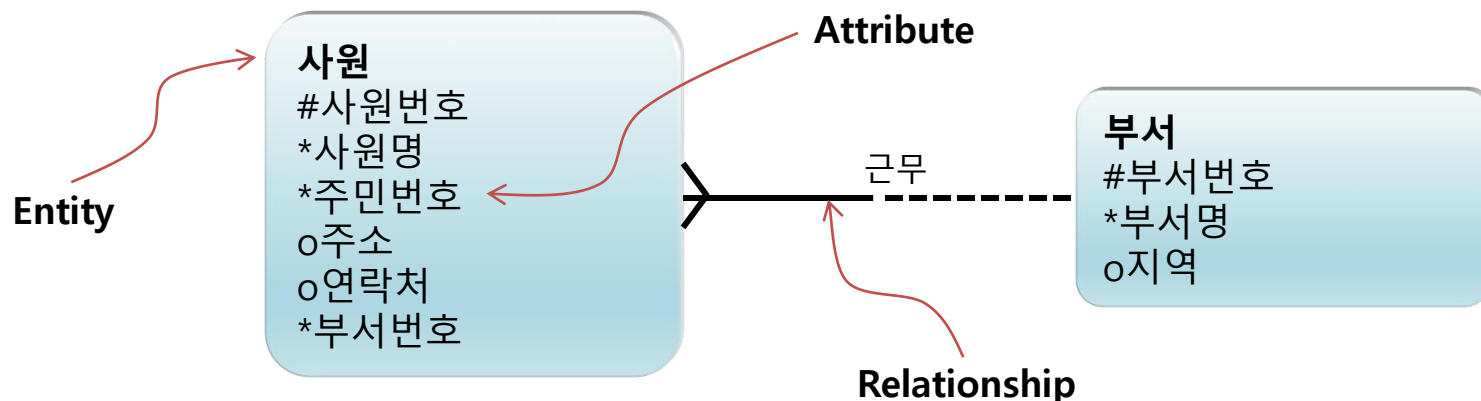
동일한 속성에 관한 무결성 규칙으로 데이터 타입, 길이, 허용 값, 기본값, 유일성, Null 여부 등에 관한 제한이다. (Oracle 데이터베이스의 경우 Physical하게는 PK Constraint와 FK Constraint를 제외한 각종 Constraint라는 것들이다)

- 데이터 타입(Data Type)
- 길이(Length)
- 허용 값(Permitted Value)
- 범위 체크(Range Check)

2-2 데이터 모델 3요소

❖ 구성요소 개념

- ✓ 업무가 다루는 사항(대상) ⇒ 엔티티(개체, 실체)(Entity)
(The things with which the business deals)
- ✓ 업무가 다루는 사항(대상)들 사이에 존재하는 연관 ⇒ 관계(Relationship)
(The associations which may exist between those things)
- ✓ 각 사항(대상)이 가지고 있는 상세한 특성 ⇒ 속성(Attribute)
(The detailed characteristics each thing may possess)



2-2 데이터 모델 3요소

❖ 엔티티(Entity)

- ✓ 개체 또는 실체로 해석하며, 업무 활동상 지속적인 관심을 가지고 있어야 하는 대상으로서 그 대상들 간에 동질성을 지닌 것으로 볼 수 있는 개체 집합이나 그들이 행하는 행위의 집합으로 정의할 수 있다.
- ✓ 학자별 실체 정의
 - Peter Chen : 유일하게 식별될 수 있는 사물 (1976)
 - C.J.Date : 데이터베이스 내에 나타내는 구별할 수 있는 객체 (1986)
 - James Martin : 우리가 그에 대한 정보를 보관하는 어떤 것이나 가능하다. (1989)
 - Thomas Bruce : 정보로 관리되어야 하는 식별 가능한 사람, 장소, 사물, 사건, 개념 (1992)



사람



장소



개념



사물

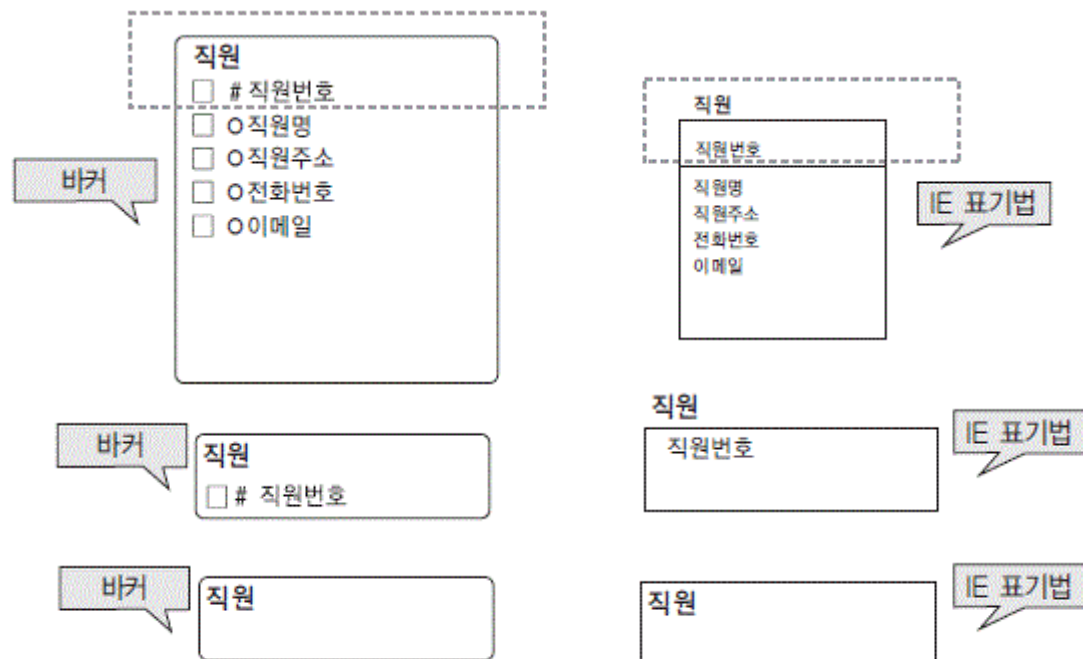


이벤트

2-2 데이터 모델 3요소

❖ 엔티티(Entity) 예

- ✓ 엔티티는 데이터 모델에서 3단계의 상세 수준으로 표현된다.
- ✓ 데이터 모델링의 초기 단계에서부터 모델을 지나치게 상세히 표현하면 개체-관계 다이어그램을 이해하고 다루기가 힘들어질 수도 있기 때문에 간단하게 식별자만 보이거나 직사각형 안에 개체 이름만 간결하게 보임으로써 많은 모델들 중에서 사용자가 필요한 것들을 손쉽게 식별하게 하기도 한다. 더 간결한 형식은 상위 수준의 개요와 전반적인 개체-관계를 보여주는 데 사용된다.



2-2 데이터 모델 3요소

❖ 속성(Attribute)

- ✓ 속성은 엔티티 내에서 관리하고자 하는 정보들의 항목이다. 엔티티를 명확하고 구체적으로 정의했다 하더라도 이들의 특성을 설명할 수 있는 보다 구체적인 항목(속성)이 없으면 이 집합을 명쾌하게 객관화할 수 없다.
- ✓ “더 이상 분리되지 않는 단위 값(Atomic Value)”
- ✓ 실체(Entity)를
 - 서술하고
 - 양을 계수화하고
 - 자격을 부여하고
 - 분류를 하고
 - 구체적으로 기입하는 정보항목

2-2 데이터 모델 3요소

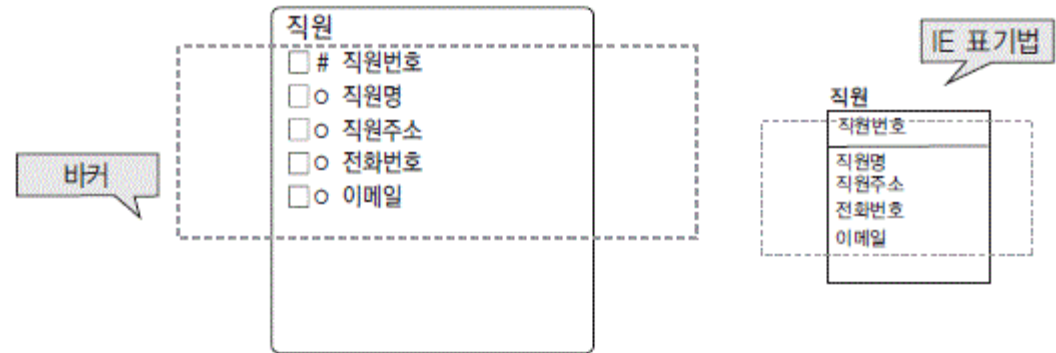
❖ 속성(Attribute) 예

✓ 실체(Entity)

- 사원
- 자동차
- 주문

✓ Instance

- 사원번호, 사원명, 생년월일, 주소, 직급, 이메일, ...
- 모델, 중량, 가격, ...
- 주문번호, 주문일자, 선적일자, ...



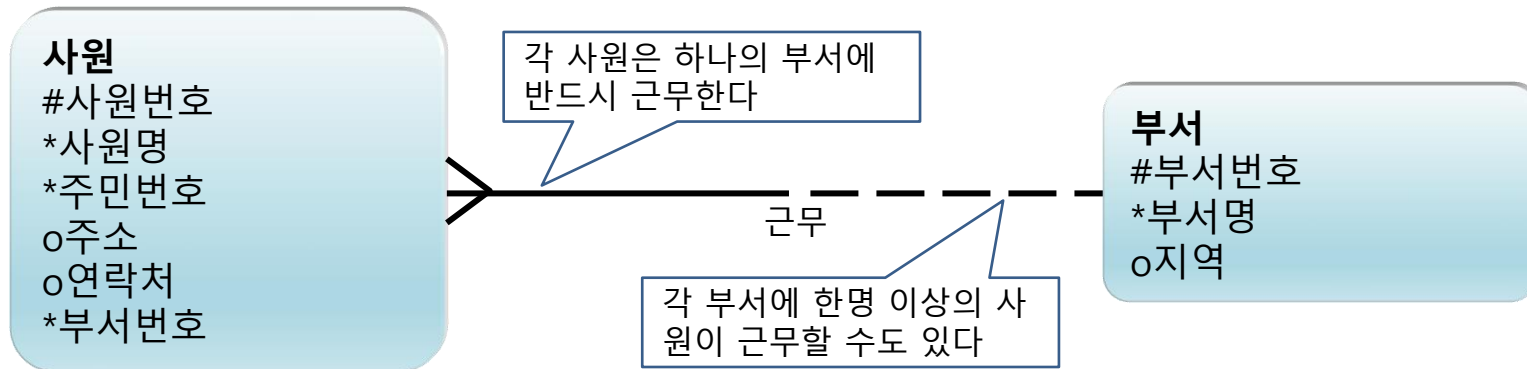
2-2 데이터 모델 3요소

❖ 관계(Relationship)

- ✓ 관계는 엔티티와 엔티티 간 연관성을 표현하는 것으로, 엔티티의 정의에 따라 영향을 받기도 하고, 속성 정의 및 관계 정의에 따라서도 다양하게 변할 수 있다.
- ✓ 실체(Entity) 상호간 어떻게 연관되어 있는지를 파악하여 표현
 - 식별성 (Identification)
 - 선택성 (Optionality)
 - 기수성 (Degree, Cardinality)
- ✓ 관계는 항상 2개의 실체(또는 한 개가 두 번) 사이에 존재(이항 관계 모델)
- ✓ 관계는 두 개의 관점(관계 Membership)을 가짐
- ✓ 관계의 양쪽 끝에 관계 명을 적음(관계가 상당히 복잡한 경우). 대부분은 관계선 가운데에 관계 명을 적음. 관계 명을 적지 않아도 알 수 있는 경우(교차실체, 1차 정규화 실체)에는 적지 않아도 무방함

2-2 데이터 모델 3요소

❖ 관계(Relationship) 예



주어부	관계비	목적부	선택사항
각 사원은	단 하나의	부서를	반드시 가져야 한다
각 부서는	한명 이상의	사원을	가질 수도 있다

실습1. 데이터 모델링 구성 3요소 작성

❖ 업무처리규정

- ✓ 각각의 고객은 하나 이상의 계약에 가입할 수도 있다.
- ✓ 고객에 관해서는 고객번호, 고객명, 주소 등의 정보가 유지됩니다.
- ✓ 계약에 관해서는 계약번호, 계약일자, 상품에 대한 정보가 유지됩니다.

❖ 개체, 관계, 속성을 검출하여 작성하시오.

- ✓ 개체 :
- ✓ 관계 :
- ✓ 속성 :

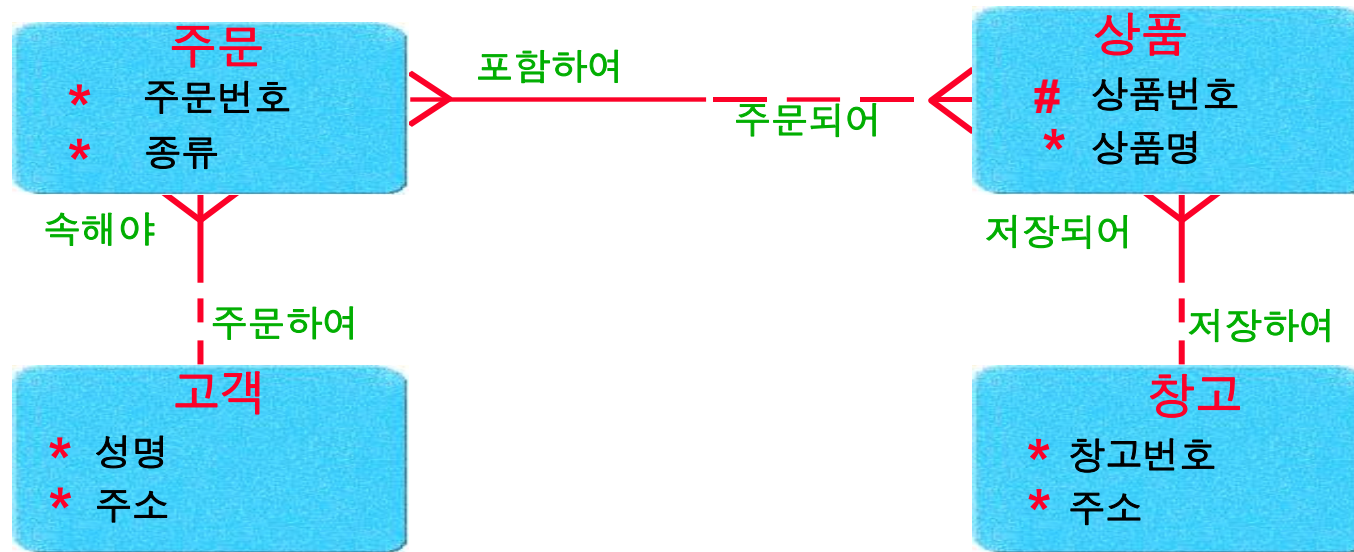
실습2. 관계 기수성

예시5) 한 명의 남자는 단 한 명의 여자와 법적인 결혼관계를 유지할 수 있다.

예시6) 한 공급자는 여러 번의 계약을 실시하며, 한 계약은 한 공급자에 의해 이루어진다.
또, 한 개의 계약에 대하여 여러 번의 주문이 이루어진다.

예시7) 한 계약에 TV, VTR, 오디오 등의 품목이 포함될 수 있다.
TV 품목 또한 계약1, 계약2 등 여러 개의 계약 내에 포함될 수 있다.

실습3. 관계(Relationship) 읽기



- ✓ 각 주문은 **하나 이상의**
- ✓ 각 상품은 **하나 이상의**
- ✓ 각 주문은
- ✓ 각 고객은
- ✓ 각 상품은
- ✓ 각 창고는

상품을 반드시 포함해야 한다.
주문에 **주문될** 수도 있다.
고객을(에)
주문을
창고를(에)
상품을

3-1 표기법 기본 개념

❖ 개체-관계 모델의 표기법 사례

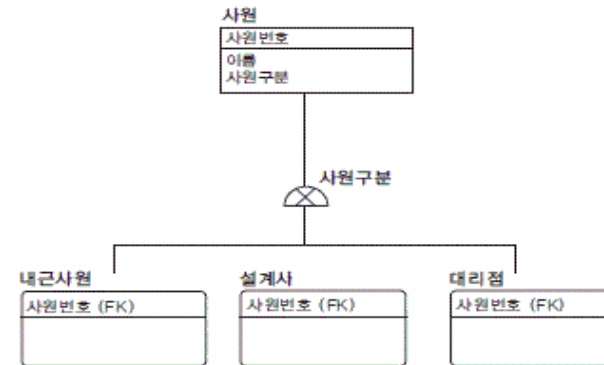
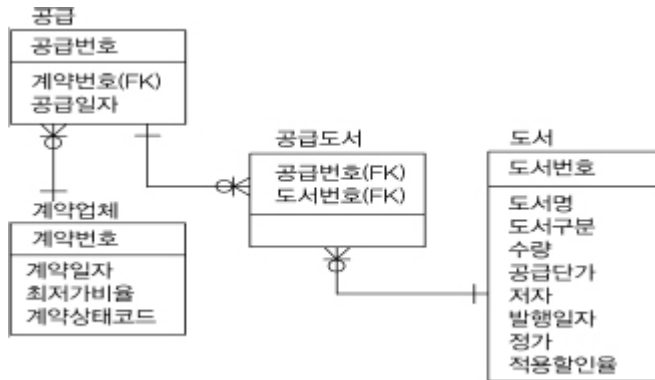
- ✓ 정보공학 표기법 (IE Notation)
 - ERWin

- ✓ 바커 표기법 (Barker's Notation)
 - DA#

3-1 표기법 기본 개념

❖ 정보공학 표기법 (IE Notation)

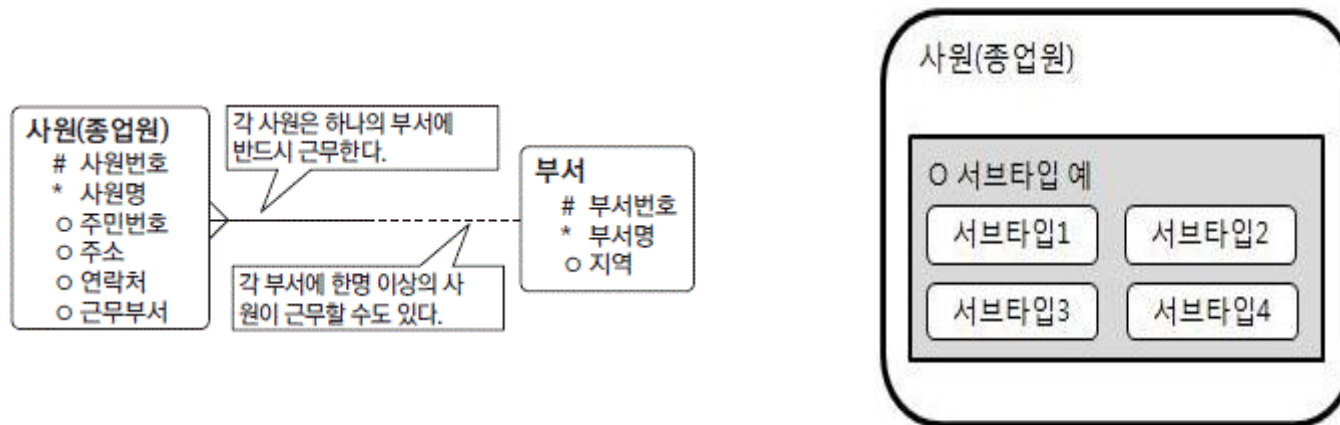
- ✓ 제임스 마틴(James Martin)이 주창한 Information Engineering에서 사용하는 데이터 모델 표기법
- ✓ 까마귀발 모양과 같은 형태의 관계 표현 때문에 Crow's Foot Model로 불리기도 한다.
- ✓ 가장 널리 사용되는 표기법의 하나이지만, 서브타입과 같은 개체 집합의 상세 표현에 있어서 공간을 많이 차지하는 단점이 있다.



3-1 표기법 기본 개념

❖ 바커 표기법 (Barker's Notation)

- ✓ 1986년, 영국 컨설팅 회사 CACI에 근무하던 Richard Barker, Lan Palmer, Harry Ellis 등에 의해 처음 개발. Original Chen style 에서 사용되던 Crow's Foot' style을 변형하여 개발
- ✓ 후에 오라클로 이직한 리차드 바커에 의해 오라클에서 Case Method로 채택하여 사용되고, Oracle CASE modeling tools에 적용
- ✓ 오늘날 널리 사용되는 표기법의 하나로 개체 집합과 그 관계의 표현에 있어서 직관적 이해성이 뛰어나고 구체적이고 상세한 표현성 및 공간 활용의 장점 등으로 인해 사용자가 지속적으로 증가



실습4. DA# 실습

- 설치파일 다운로드 : <http://www.en-core.com/board/download>
- DAV5010204_OpenSetup.exe
- DATAWARE™ DA 5 설치 매뉴얼.pdf