

01. 데이터 모델의 이해

2017년 10월 30일 월요일 오후 11:57

1. 모델링의 이해

가. 모델링의 정의

모델: 모형(模型), 축소형(縮小型)의 의미로서 사람이 살아가면서 나타날 수 있는 다양한 현상에 대해서 일정한 표기법에 의해 표현해 놓은 모형
즉 커뮤니케이션의 효율성을 극대화한 고급화된 표현방법

모델링: 표기법에 의해 규칙을 가지고 표기하는 것. 즉, 모델을 만들어가는 일 자체

* 모델을 만든다 ----> 모델링



복잡한 현실세계를 일정한 표기법에 의해 표현하는 일

[그림 1-1-1] 모델링의 정의

모델링에 대한 다양한 정의

1) Webster 사전

- 가설적 또는 일정 양식에 맞춘 표현 (a hypothetical or stylized representation)
- 어떤 것에 대한 예비표현으로 그로부터 최종대상이 구축되도록 하는 계획으로서 기여하는 것
- 2) 복잡한 '현실세계'를 단순화시켜 표현하는 것
- 3) 모델이란 사물 또는 사건에 관한 양상(Aspect)이나 관점(Perspective)을 연관된 사람이나 그룹을 위하여 명확하게 하는 것
- 4) 모델이란 현실 세계의 추상화된 반영

나. 모델링의 특징

1) 추상화(모형화, 가설적) 표현

현실세계를 일정한 형식에 맞추어 표현을 한다는 의미. 즉, 다양한 현상을 일정한 양식인 표기법에 의해 표현

2) 단순화

복잡한 현실세계를 약속된 규약에 의해 제한된 표기법이나 언어로 표현하여 쉽게 이해할 수 있도록 하는 개념

3) 명확화

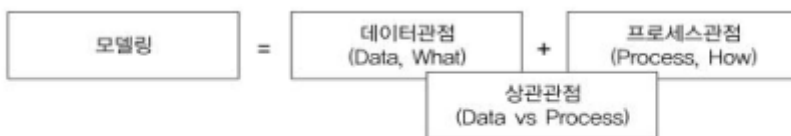
누구나 이해하기 쉽게 하기 위해 대상에 대한 애매모호함을 제거하고 정확(正確)하게 현상을 기술하는 것

(오답: 구체화 아님)

※ 정보시스템 구축에서 모델링 활용

- 계획/분석/설계 할 때 업무를 분석하고 설계하는데 이용
- 구축/운영 단계에서는 변경과 관리의 목적으로 이용

다. 모델링의 세 가지 관점



[그림 1-1-2] 모델링의 관점

- 데이터관점**: 업무가 어떤 데이터와 관련이 있는지 또는 데이터간의 관계는 무엇인지에 대해서 모델링하는 방법(What, Data)
- 프로세스관점**: 업무가 실제로 있는 일은 무엇인지 또는 무엇을 해야 하는지를 모델링하는 방법(How, Process)
- 데이터와 프로세스의 상관관점**: 업무가 처리하는 일의 방법에 따라 데이터는 어떻게 영향을 받고 있는지 모델링하는 방법(Interaction) (프로세스 관점 + 데이터관점)

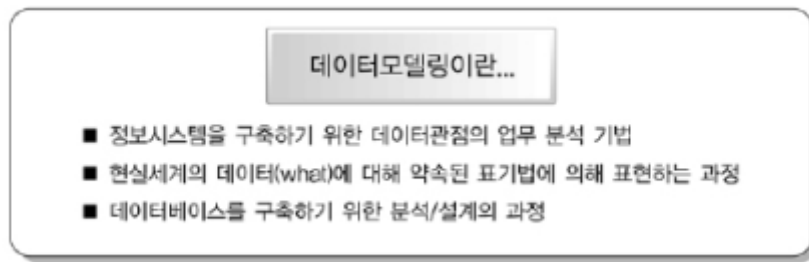
2. 데이터 모델의 기본 개념의 이해

가. 모델링의 정의

- 업무에서 필요로 하는 데이터를 시스템 구축 방법론에 의해 분석하고 설계하여 정보시스템을 구축하는 과정으로 정의할 수 있다.

- 일정한 표기법에 의해 표현함으로써 정보시스템 구축의 대상이 되는 업무내용을 정확하게 분석해야 한다.

- 단지 데이터베이스만을 구축하기위한 용도로만 쓰이는 것이 아니라, 데이터 모델링 자체로서 업무를 설명하고 분석하는 부분에도 매우 중요한 의미를 가지고있다.



나. 데이터 모델이 제공하는 기능

- 시스템을 현재 또는 원하는 모습으로 가시화하도록 도움
- 시스템의 구조와 행동을 명세화 할 수 있게 함
- 시스템을 구축하는 구조화된 틀을 제공
- 시스템을 구축하는 과정에서 결정한 것을 문서화
- 다양한 영역에 집중하기 위해 다른 영역의 세부 사항은 숨기는 다양한 관점을 제공
- 특정 목표에 따라 구체화된 상세 수준의 표현방법을 제공

3. 데이터 모델링의 중요성 및 유의점

가. **파급효과(Leverage)**

병행테스트, 통합테스트를 수행 중 만약, 이러한 시점에 **데이터 모델의 변경**이 불가피한 상황이 발생 할 경우

- 1) 이를 위해서 데이터 구조의 변경에 따른 표준 영향 분석, 응용 변경 영향 분석 등 많은 영향 분석 발생
- 2) 해당 분야의 실제적인 변경 작업이 발생

- **변경이 불가능한 상황에 아주 큰 위험요소가 생기는 것을 막기위함**

☞ 변경을 해야 하는 데이터 모델의 형태에 따라서 그 영향 정도는 차이있지만 데이터 구조의 변경으로 인한 일련의 변경작업은 전체 시스템 구축 프로젝트에서 큰 위험요소

나. **복잡한 정보 요구사항의 간결한 표현(Conciseness)**

데이터 모델 : **구축할 시스템의 정보요구사항과 한계를 가장 명확하고 간결하게 표현할 수 있는 도구**

데이터 모델은 건축물로 비유하자면 설계 도면에 해당한다. 이것은 건축물의 설계 도면이 건축물을 짓는 많은 사람들이 공유하면서 설계자의 생각대로 일사불란하게 움직여서 아름다운 건축물을 만들어 내는 것에 비유할 수 있다. 데이터 모델은 시스템을 구축하는 많은 관련자들이 설계자의 생각대로 정보요구사항을 이해하고 이를 운용할 수 있는 애플리케이션을 개발하고 데이터 정합성을 유지할 수 있도록 하는 것이다. 이렇게 이상적으로 역할을 할 수 있는 모델이 갖추어야 할 가장 중요한 점은 정보 요구사항이 정확하고 간결하게 표현되어야 한다는 것이다. 우리가 활용하고 있는 데이터 모델이 이와 같은 요소들이 충족된 모델인지를 확인해 볼 필요가 있다.

다. **데이터 품질(Data Quality)**

- DB의 데이터는 기업의 중요한 자산, 시간이 오래되면 될수록 활용가치는 훨씬 높아짐
- But, 이러한 오래도록 저장되어진 데이터가 그저 그런 데이터, 정확성이 떨어지는 데이터?
☞ 해당 데이터로 얻을 수 있었던 소중한 비즈니스의 기회를 상실할 수도 있는 문제

데이터 품질의 문제가 중요한 이유

- 데이터 품질의 문제는 데이터 구조가 설계되고 초기에 데이터가 조금 쌓일 때에는 인지하지 못하는 경우가 대부분
- 데이터의 문제는 오랜 기간 축적된 데이터를 전략적으로 활용하려고 하는 시점에 문제가 대두됨

데이터 품질의 문제가 야기시키는 원인

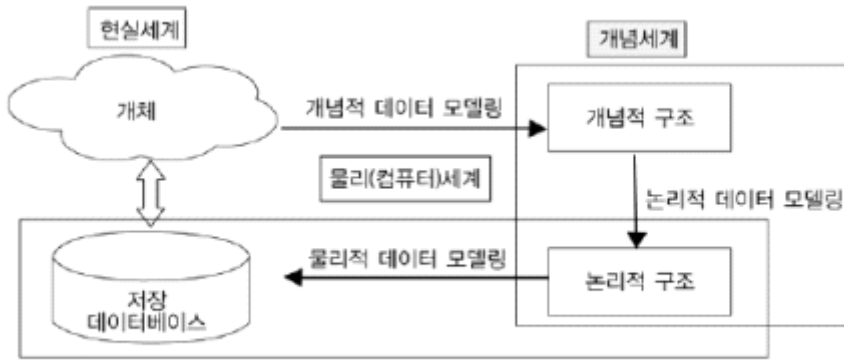
- 데이터 구조의 문제
☞ 중복 데이터의 미정의, 데이터 구조의 비즈니스 정의의 불충분, 동일한 성격의 데이터를 통합하지 않고 분리함으로써의 나타나는 데이터 불일치 등의 데이터 구조의 문제로 인한 데이터 품질의 문제는 치유하기에 불가능한 경우가 대부분

데이터 모델링을 할 때 유의점

- 1) **중복(Duplication)** 데이터베이스가 여러 장소에 같은 정보를 저장하는 것을 유의
- 2) **비유연성(Inflexibility)** 데이터 모델을 설계방법에 따라 사소한 업무변화에 데이터 모델이 수시로 변경됨으로써 유지보수의 어려움 가증 가능
☞ 해결 : 데이터의 정의를 데이터의 사용 프로세스와 분리
- 3) **비일관성(Inconsistency)** 데이터의 중복이 없어도 비일관성 발생
☞ 해결 : 데이터 모델링을 할 때 데이터와 데이터간 상호 연관 관계에 대한 명확한 정의는 사전에 예방 가능

4. 데이터 모델링의 3단계 진행

아래 그림과 같이 시간에 따라 진행되는 과정으로서 **추상화 수준에 따라** 개념적 데이터 모델, 논리적 데이터 모델, 물리적 데이터 모델로 정리



[그림 1-1-3] 현실세계와 데이터베이스 사이의 모델

- 1) 현실세계에서 추상화 수준이 높은 상위 수준을 형상화하기 위해 개념적 데이터 모델링을 전개
개념적 데이터 모델은 추상화 수준이 높고 업무중심적이고 포괄적인 수준의 모델링을 진행
참고로 EA기반의 전사적인 데이터 모델링을 전개할 때는 더 상위수준인 개괄적인 데이터 모델링을 먼저 수행하고 이후에 업무영역에 따른 개념적 데이터 모델링을 전개
- 2) 엔티티(Entity)중심의 상위 수준의 데이터 모델이 완성되면 업무의 구체적인 모습과 흐름에 따른 구체화된 업무중심의 데이터 모델을 만들어 냄 (논리적인 데이터 모델링)
- 3) 논리적인 데이터 모델링 이후 데이터베이스의 저장구조에 따른 테이블스페이스 등을 고려한 방식 (물리적인 데이터 모델링)

[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

가. 개념적 데이터 모델링(Conceptual Data Modeling)

개념적 데이터 모델링

- 조직, 사용자의 데이터 요구사항을 찾고 분석하는데서 시작
- 이 과정은 어떠한 자료가 중요하며 또 어떠한 자료가 유지되어야 하는지를 결정하는 것도 포함

주요한 활동

- 핵심 엔티티와 그들 간의 관계를 발견 → 엔티티-관계 다이어그램 생성

※ 전사적 데이터 모델(Enterprise Data Model) : 전 조직에 걸쳐 이루어지는 데이터 모델링 과정

개념 데이터 모델을 통한 조직의 데이터 요구를 공식화의 의미

첫째, 개념 데이터 모델은 사용자와 시스템 개발자가 데이터 요구 사항을 발견하는 것 지원. 개념 데이터 모델은 추상적이다. 그렇기 때문에 그 모델은 상위의 문제에 대한 구조화를 쉽게 하며, 사용자와 개발자가 시스템 기능에 대해서 논의할 수 있는 기반 형성
둘째, 개념 데이터 모델은 현 시스템이 어떻게 변형되어야 하는가를 이해하는데 유용. 일반적으로 매우 간단하게 고립된(Stand Alone) 시스템도 추상적 모델링을 통해서 보다 쉽게 표현되고 설명

나. 논리적 데이터 모델링(Logical Data Modeling)

논리 데이터 모델링

"기초적인 업무조사를 하는 초기단계에서부터 인간이 결정해야 할 대부분의 사항을 모두 정의하는 시스템 설계의 전 과정을 지원하는 '과정의 도구'"

- 데이터베이스 설계 프로세스의 Input
- 비즈니스 정보의 논리적인 구조와 규칙을 명확하게 표현하는 기법 또는 과정
- 데이터 모델링 과정에서 가장 핵심이 되는 부분
- 단지 ERD라는 그림으로 그려내는 과정을 말하는 것이 아님
- 정규화 작업 포함

※ 참고

정규화는 논리 데이터 모델 상세화 과정의 대표적인 활동으로, 논리 데이터 모델의 일관성을 확보하고 중복을 제거하여 속성들이 가장 적절한 엔티티에 배치되도록 함으로써 보다 신뢰성있는 데이터구조를 얻는데 목적

논리 데이터 모델의 상세화

- 식별자 확정, 정규화, M:M 관계 해소, 참조 무결성 규칙 정의 등
- 추가적으로 이력 관리에 대한 전략을 정의하여 이를 논리 데이터 모델에 반영

다. 물리적 데이터 모델링(Physical Data Modeling)

물리 데이터 모델링

- 논리 데이터 모델이 데이터 저장소로서 어떻게 컴퓨터 하드웨어에 표현될 것인가를 다룸

물리적 스키마

- 데이터가 물리적으로 컴퓨터에 어떻게 저장될 것인가에 대한 정의
- 테이블, 칼럼 등으로 표현되는 물리적인 저장구조와 사용될 저장 장치, 자료를 추출하기 위해 사용될 접근 방법 등이 결정

※ 참고

계층적 데이터베이스 관리 시스템 환경에서는 데이터베이스 관리자가 물리적 스키마를 설계하고 구현하기 위해서 보다 많은 시간을 투자하여야 한다.

☞ But, 실질적인 현실 프로젝트에서는 개념적 데이터 모델링, 논리적 데이터 모델링, 물리적 데이터 모델링으로 수행하는 경우는 드물며 개념적 데이터 모델링과 논리적 데이터 모델을 한꺼번에 수행하여 논리적인 데이터 모델링으로 수행하는 경우가 대부분

5. 프로젝트 생명주기(Life Cycle)에서 데이터 모델링

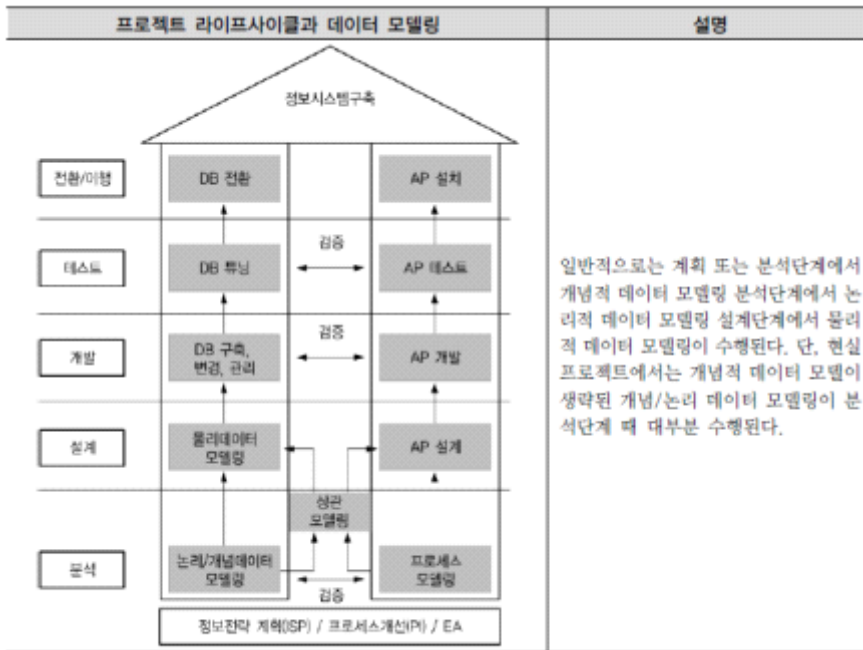
Waterfall 기반 : 데이터 모델링의 위치가 분석과 설계단계로 구분되어 명확하게 정의

정보공학이나 구조적 방법론

- 1) 보통 분석단계에서 업무중심의 논리적인 데이터 모델링을 수행
- 2) 설계단계에서 하드웨어와 성능을 고려한 물리적인 데이터 모델링을 수행

나선형 모델, 예를 들어 RUP(Rational Unified Process나 마르미)

- 업무구기에 따라 논리적 데이터 모델과 물리적 데이터 모델이 분석, 설계단계 양쪽에서 수행
- 비중은 분석단계에서 논리적인 데이터 모델이 더 많이 수행되는 형태
- 데이터축과 애플리케이션축으로 구분되어 프로젝트가 진행되면서 각각에 도출된 사항은 상호 검증을 지속적으로 수행하면서 단계별 완성도를 높게 된다.



[그림 1-1-4] 프로젝트 생명주기에 따른 데이터 모델

단, 객체지향 개념은 데이터 모델링과 프로세스 모델링을 구분하지 않고 일체형으로 진행
(대표적인 예가 데이터(속성)과 프로세스(Method)가 같이 있는 클래스(Class))
∴ 데이터와 프로세스를 한꺼번에 바라보면서 모델링 전개

6. 데이터 모델링에서 데이터독립성의 이해

가. 데이터독립성의 필요성_데이터무결성 유지

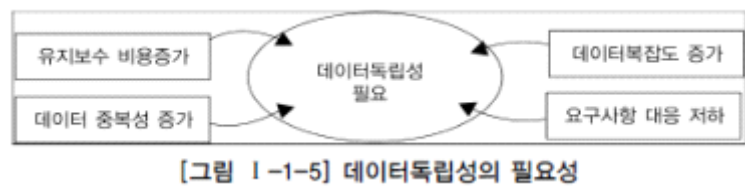
- 상호간 영향에서 벗어나 개별 형식이 가지는 고유의 기능을 유지시키며 그 기능을 극대화하는 것
- 컴포넌트 기반의 모듈 구성도 각각이 고유한 기능을 가지면서 다른 기능을 가지고 있는 컴포넌트와 인터페이스를 가지게 하는 모습으로 정의
- SOA의 '서비스'라고 하는 단위도 독립적인 비즈니스로 처리 가능한 단위를 서비스로 정의
그것이 다른 서비스에 비해 독립성을 구성하여 개별로도 의미
다른 서비스와 결합하여 프로세스로 제공해도 의미가 있는 단위(예, BPM)로 제공

이와 같이 어떤 단위에 대해 그 것을 효과적으로 구현독립적인 의미를 부여하게 되면 자신이 가지는 고유한 특징을 명확하게 할 뿐만 아니라 다른 기능의 변경으로부터 쉽게 변경되지 않고 자신의 고유한 기능을 가지고 기능을 제공하는 장점

데이터독립성 출현 배경

- 끊임없이 요구되는 사용자 요구사항에 대해 화면과 데이터베이스 간에 서로 독립성을 유지하기 위한 목적
- 데이터독립성의 반대말은 데이터종속성 (여기에서 종속의 주체는 보통 응용(Application)을 지칭하는 경우)
응용(Application)은 사용자 요구사항을 처리하는 사용자 접점의 인터페이스 오브젝트
과거에 파일방식으로 데이터를 구성할 때는 데이터가 있는 파일과 데이터에 접근하기 위한 인덱스를 별도로 구현하여 접근하게 하였는데 사용자가 접근하는 방법(트랜잭션의 유형)에 따라 파일의 정렬순서, 인덱스의 정렬순서, 파일 구성 등을 제공하기 쉽게 별도로 구성 ☞ 즉, 사용자 접근하는 유형에 따라 데이터를 구성하는 방법이 영향

- 데이터독립성은 지속적으로 증가하는 유지보수 비용을 절약
- 데이터 복잡도를 낮추며 중복된 데이터를 줄이기 위한 목적



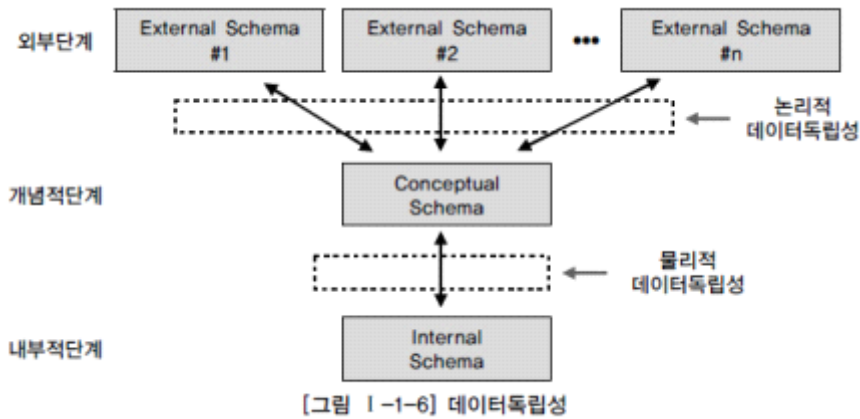
- 미국 표준 협회(ANSI) 산하의 X3 위원회(컴퓨터 및 정보 처리)의 특별연구분과위원회에서 1978년에 DBMS와 그 인터페이스를 위해 제안한 ‘three-schema architecture’로 정의

데이터 독립성의 효과

- 각 View의 독립성을 유지하고 계층별 View에 영향을 주지 않고 변경이 가능하다.
 - 단계별 Schema에 따라 데이터 정의어(DDL)와 데이터 조작어(DML)가 다름을 제공한다.
- 데이터독립성을 이해하기 위해서는 3단계로 표현된 ANSI 표준 모델을 살펴보면 되는데 특히 3단계인 구조, 독립성, 사상(Mapping) 3가지를 이해하면 된다.

나. 데이터베이스 3단계 구조

ANSI/SPARC의 3단계 구성의 데이터독립성 모델 : 외부단계, 개념적 단계, 내부적 단계로 서로 간섭되지 않는 모델을 제시



다. 데이터독립성 요소

[표 1-1-2] 데이터독립성 구성요소

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

데이터베이스 스키마 구조

- 3단계로 구분, 각각은 상호 독립적인 의미와 고유한 기능 소유
- ※ 참고 : 데이터 모델링은 통합관점의 뷰를 가지고 있는 개념 스키마를 만들어가는 과정

라. 두 영역의 데이터독립성

논리적인 독립성 & 물리적인 독립성 : 3단계로 개념의 각 영역에 대한 독립성을 지정하는 용어

[표 1-1-3] 논리적, 물리적 데이터독립성

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

논리적인 데이터독립성 : 외부의 변경에도 개념스키마가 변하지 않는 특징

마. 사상(Mapping)

영어로 'Mapping' 은 우리말로 '사상'이라고 번역되는데 이것은 상호 독립적인 개념을 연결시켜주는 다리를 뜻함
데이터독립성에는 크게 2가지의 사상 도출

[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

논리적 사상 : 외부 화면이나 사용자에게 인터페이스하기 위한 스키마 구조는 전체가 통합된 개념적 스키마와 연결된다는 것
물리적 사상 : 통합된 개념적 스키마 구조와 물리적으로 저장된 구조의 물리적인 테이블스페이스와 연결되는 구조

7. 데이터 모델링의 중요한 세 가지 개념

가. 데이터 모델링의 세 가지 요소데이터 모델링을 구성하는 중요한 개념 세 가지가 있는데 이것은 데이터 모델에 대한 이해의 근간이 되므로 반드시 기억할 필요가 있다.

- 1) 업무가 관여하는 어떤 것(Things)
- 2) 어떤 것이 가지는 성격(Attributes)
- 3) 업무가 관여하는 어떤 것 간의 관계(Relationships)

이 세 가지는 데이터 모델링을 완성해 가는 핵심 개념 **결국 엔터티, 속성, 관계로 인식**

사물이나 사건 등을 바라 볼 때 전체를 지칭하는 용어를 어떤 것(Things)이라 하고, 그 어떤 것이 가지는 세부적인 사항을 성격(Attributes)이라고 할 수 있다. 또한 각각의 어떤 것은 다른 어떤 것과 연관성을 가질 수 있는데 이것을 관계(Relationship)라고 표현한다. 예를 들어 '이주일과 심순애가 존재하고 둘 사이는 서로 사랑하는 연인사이이다. 이주일은 키가 180cm에 성격은 친절하고 심순애는 키가 165cm에 세심하며 활발한 성격을 가지고 있다'는 시나리오를 살펴보자. 여기에서 '이주일, 심순애'는 어떤 것(Things)에 해당하고 '사랑하는 연인사이'가 어떤 것 간의 관계(Relationships)에 해당하며 '180cm에 성격은 친절, 세심하며 활발함'이 어떤 것이 가지는 성격(Attributes)에 해당한다. 위의 예와 같이 이 세상의 모든 사람, 사물, 개념 등은 어떤 것, 어떤 것 간의 관계, 성격의 구분을 통해서 분류할 수 있다. 바로 이러한 원리, 즉 자연계에 존재하는 모든 유형의 정보들을 세 가지 관점의 접근 방법을 통해 모델링을 진행하는 것이다.

나. 단수와 집합(복수)의 명명

[표 1-1-5] 용어의 구분정의

개념	복수/집합개념 타입/클래스	개별/단수개념 어커런스/인스턴스
어떤 것 (Thing)	엔터티 타입(Entity Type)	엔터티(Entity)
	엔터티(Entity)	인스턴스(Instance), 어커런스(Occurrence)
어떤 것간의 연관 (Association between Things)	관계(Relationship)	페어링(Pairing)
어떤 것의 성격 (Characteristic of a Thing)	속성(Attribute)	속성값(Attribute Value)

- 어떤 것 : 실제 실무 현장에서는 복수/집합개념도 엔터티로 짧게 명명
엔터티를 집합개념으로 사용하는 경우, 개별요소에 대해서는 인스턴스/어커런스를 단수의 개념으로 사용
- 관계(Relationship) : 일반적으로 단수든 복수든 대부분 관계라고 표현
- 성격(Attribute) : 속성과 속성값으로 구분

※ 교재에서의 용어 구분 사용 **Thing : 복수 - 엔터티, 단수 - 인스턴스**라 표기

8. 데이터 모델링의 이해관계자

가. 이해관계자의 데이터 모델링 중요성 인식

- 실질적으로 DBA보다 업무시스템을 개발하는 응용시스템 개발자가 데이터 모델링도 같이하는 경우가 많음
- 원인 데이터 모델링이라는 과정의 특징
 - 1) 단지 데이터베이스를 설계한다는 측면 < 업무를 이해하고 분석하여 표현하는 것
 - 2) 표현된 내용을 바탕으로 프로젝트 관련자와 의사소통
 - 3) 프로그램이나 다른 표기법과 비교 검증하는 일을 수행

∴ 많은 시간을 업무를 분석하고 설계하는데 할애하기 때문에 업무영역별 개발팀에서 보통 데이터 모델링을 진행
단, 시스템이 대형화되면 모델링만을 전문적으로 담당하는 모델러를 투입하여 진행
but, 실제 모델링 작업은 응용개발을 하는 사람이나 업무분석가(역할분담이 잘되어 있을 경우)가 담당
모델러나 DBA는 정확하게 모델링이 진행될 수 있도록 교육하고 제시하며 현안별로 직접 모델링을 진행하는 역할을 수행

- 대부분의 기업에 있어서 정보시스템의 데이터베이스 구조는 사용자에게 숨겨진 형태로 구축되어 왔다.
→ 정보의 고립화



- 프로그램은 6가지 유형의 데이터베이스 유지절차 - C. Finkelstein
- Programmer is the Navigator in the sea of data. - Bachmann

[그림 1-1-7] 자료 처리의 중심은 데이터

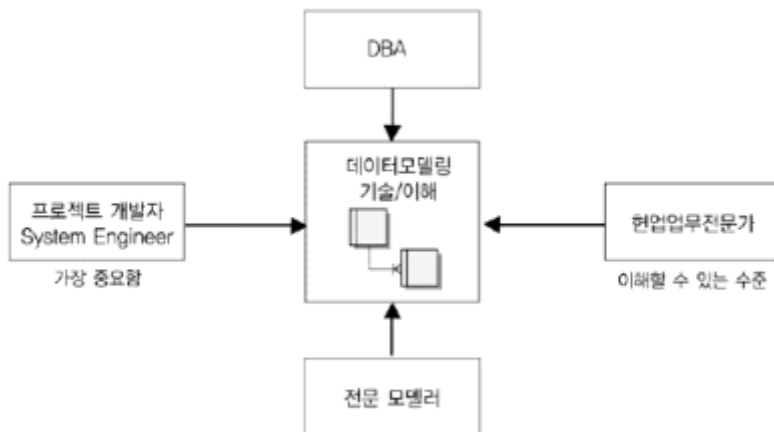
우리가 구축하려는 시스템 대부분을 데이터에 기반한, 데이터가 중심에 있는 정보시스템을 구축하기 때문에 정보시스템의 핵심에 있는 데이터베이스 설계를 잘못했을 때 미치는 영향력은 모든 프로그램, 시간에 따라 입력되는 모든 데이터, 그리고 그 데이터베이스에 발생하는 모든 트랜잭션에 영향을 미칠 수 밖에 없게 된다. Bachmann은 '프로그래머는 데이터집합의 탐색자이다'라고 하였다. 그만큼 데이터에 대한 중요성을 높게 평가하는 것이다.

나. 데이터 모델링의 이해관계자

데이터 모델링에 대해 연구 및 학습 담당자

- 첫 번째는 정보시스템을 구축하는 모든 사람 (전문적으로 코딩만하는 사람 포함)
- 데이터 모델링도 전문적으로 할 수 있거나 적어도 완성된 모델을 정확하게 해석가능 해야 함
 - ⇒ 프로젝트에 참여한 모든 IT기술자들은 데이터 모델링에 대해 정확하게 알고 있어야 함
- 두 번째는 IT기술에 종사하거나 전공하지 않았더라도 해당 업무에서 정보화를 추진하는 위치에 있는 사람
- 데이터 모델링에 대한 개념 및 세부사항에 대해 어느 정도 지식이 있어야 함

⇒ 서로가 프로젝트 수행 중에 의사소통을 잘 할 수 있고 업무를 잘못 해석하여 잘못된 시스템을 구축하는 위험(Risk)을 줄일 수 있음. 업무를 가장 잘 알고 있는 사람이 가장 훌륭한 모델러가 될 수 있음



[그림 1-1-8] 데이터 모델 이해관계자

9. 데이터 모델의 표기법인 ERD의 이해

가. 데이터 모델 표기법

1976년 피터첸(Peter Chen)이 Entity-relationship model(E-R Model)이라는 표기법

[표 1-1-6] 표기법

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

※ 교재의 표기법

데이터아키텍처 전문가(DAP) 관련 자격에서는 바커(Barker) 표기법을 적용하여 설명
본 가이드에서는 범용적인 **Information Engineering(이하 IE) 표기법**과 **바커 표기법** 모두 적용

나. **ERD**(Entity Relationship Diagram) 표기법을 이용하여 모델링하는 방법

ERD

- 각 업무분석에서 도출된 엔티티와 엔티티간의 관계를 이해하기 쉽게 도식화된 다이어그램으로 표시하는 방법

- 해당 업무에서 데이터의 흐름과 프로세스와의 연관성을 이야기하는 데 가장 중요한 표기법이자 산출물

UML 표준 표기법을 사용하는 오브젝트 모델링

해당 업무에 가장 적절한 클래스다이어그램을 그려내는 것이 가장 중요

정보공학을 기반으로 하는 모델링

해당 업무에 가장 적절한 ERD를 그려내는 것이 프로젝트의 지상과제

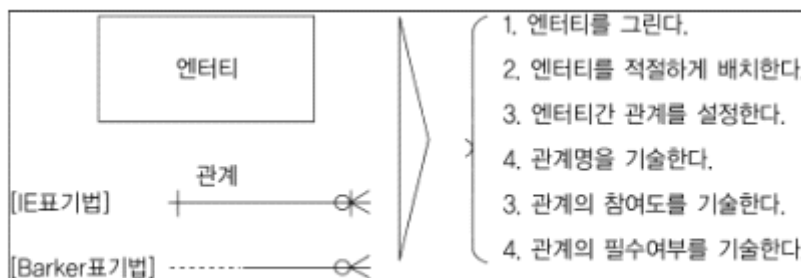
데이터베이스를 생성할 수 있는 데이터 모델 생성이 프로젝트에서 아주 중요한 작업

이론 : 데이터분석이 어느 정도 완료되면 즉 엔티티, 관계, 속성 등이 데이터사전이나 각종 산출물에 의해 분석된 상태에서 ERD 그리기

현실 : 프로젝트에서는 분석된 엔티티와 관계, 속성 정보가 바로 ERD에 표현되며 내부 프로젝트 인원이나 해당 업무고객과 대화할 때 핵심 업무산출물로 이용

1) ERD 작업순서 ERD를 작성하는 작업순서

① 엔티티를 그린다. ② 엔티티를 적절하게 배치한다. ③ 엔티티간 관계를 설정한다. ④ 관계명을 기술한다. ⑤ 관계의 참여도를 기술한다. ⑥ 관계의 필수여부를 기술한다.



[그림 1-1-9] ERD 작업순서

ERD는 엔티티와 엔티티 사이의 관계가 있는 정보를 나타내므로 두 개를 이용하여 작성

⇨ Primary Key와 Foreign Key를 ERD 규칙에 따라 기술, 엔티티는 사각형으로 표기하여 기술

2) 엔티티 배치 엔티티를 처음에 어디에 배치하는지는 데이터 모델링 툴을 사용하지 않던 중요한 문제이다. 일반적으로 사람의 눈은 왼쪽에서 오른쪽, 위 쪽에서 아래쪽으로 이동하는 경향이 있다. 따라서 데이터 모델링에서도 가장 중요한 엔티티를 왼쪽상단에 배치하고 이것을 중심으로 다른 엔티티를 나열하면서 전개하면 사람의 눈이 따라가기에 편리한 데이터 모델링을 전개할 수 있다. 해당 업무에서 가장 중요한 엔티티는 왼쪽 상단에서 조금 아래쪽 중앙에 배치하여 전체 엔티티와 어울릴 수 있도록 하면 향후 관계를 연결할 때 선이 꼬이지 않고 효과적으로 배치할 수 있게 된다.



[그림 1-1-10] 엔터티 배치방법

"그림 1-1-10"의 데이터 모델에서도 가장 중요한 엔터티인 고객과 주문을 왼쪽 상단에 배치하여 다른 엔터티를 연결하는 방식으로 엔터티를 배치하였다. 주문에 따라 출고가 이루어졌으므로 주문이 위에 출고가 아래에 위치해 있다. 두 번째 업무흐름에 중심이 되는 엔터티, 보통 업무 흐름에 있어서 중심이 되는 엔터티는 타 엔터티와 많은 관계를 가지고 있으므로 중앙에 배치하도록 한다. "그림 1-1-10"에서는 주문, 출고, 주문목록, 출고목록이 업무의 중심엔터티에 해당한다. 세 번째는 업무를 진행하는 중심엔터티와 관계를 갖는 엔터티들은 중심에 배치된 엔터티를 주위에 배치하도록 한다. "그림 1-1-10"에서는 창고, 고객, 직원, 재고가 이에 해당한다.

3) ERD 관계식서를 보고 서로 관련있는 엔터티간에 관계를 설정하도록 한다. 초기에는 모두 Primary Key로 속성이 상속되는 식별자 관계를 설정하도록 한다. 중복되는 관계가 발생되지 않도록 하고 Circle 관계도 발생하지 않도록 유의하여 작성하도록 한다.



[그림 1-1-11] ERD 관계설정

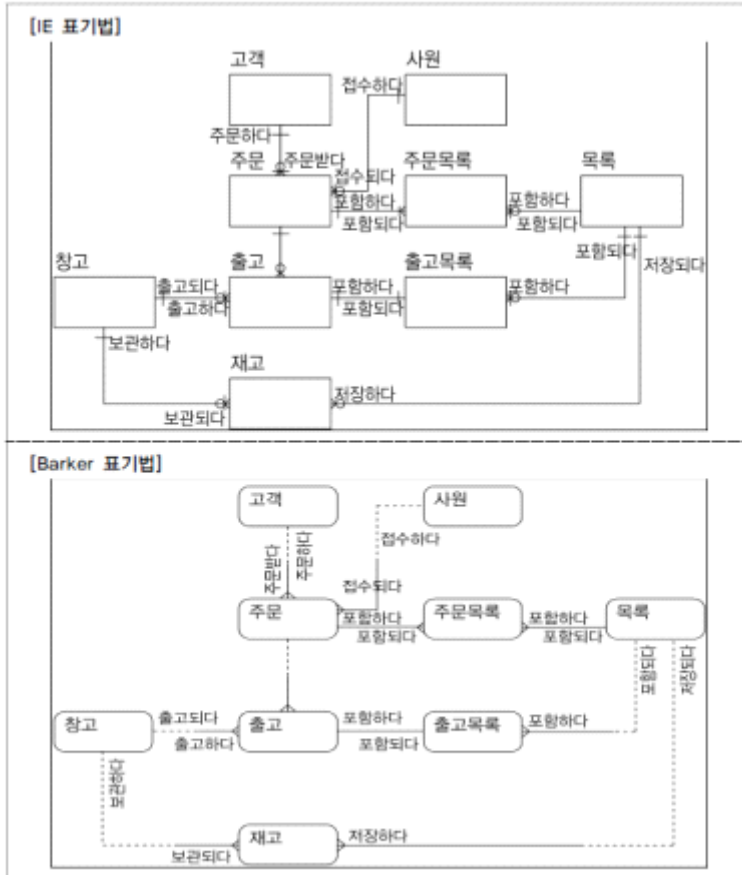
4) ERD 관계명의 표시 관계설정이 완료되면 연결된 관계에 관계이름을 부여하도록 한다. 관계이름은 현재형을 사용하고 지나치게 포괄적인 용어(예, 이다, 가진다 등)는 사용하지 않도록 한다.



[그림 1-1-12] ERD 관계명 표시

실제 프로젝트에서는 관계의 명칭을 크게 고려하지 않아도 무방하다. 왜냐하면 관계의 명칭이 나타나지 않아도 ERD의 흐름이 명확하게 드러나기 때문이다. 대부분의 관계는 엔터티의 성질과 주 식별자를 보고 유추가 가능하다.

5) ERD 관계 관계차수와 선택성 표시 관계에 대한 이름을 모두 지정하였으면 관계가 참여하는 성격 중 엔터티내에 인스턴스들이 얼마나 관계에 참여하는 지를 나타내는 관계차수(Cardinality)를 표현한다. "그림 1-1-13"은 관계의 관계차수를 지정한 ERD의 모습을 보여준다. 관계설명에서도 언급하겠지만 IE표기법으로는 하나(1, One)의 관계는 실선으로 표기하고 Barker표기법으로는 점선과 실선을 혼합하여 표기한다. 다수참여(Many)의 관계는 까마귀발과 같은 모양으로 그려준다. 또한 관계의 필수/선택표시는 관계선에 원을 표현하여 ERD를 그리도록 한다.



[그림 1-1-13] 관계차수와 선택성 표시

10. 좋은 데이터 모델의 요소

가. **완전성**(Completeness)

업무에서 필요로 하는 모든 데이터가 데이터 모델에 정의

이 기준이 충족되지 못하면 다른 어떤 평가 기준도 의미가 없어짐

만약, 보험사의 데이터 모델에 고객의 직업을 관리하기 위한 속성이 존재하지 않는다면 어떨까?

나. **중복배제**(Non-Redundancy)

하나의 데이터베이스 내에 **동일한 사실은 반드시 한 번만 기록**

저장공간의 낭비, 중복 관리되고 있는 데이터의 일관성을 유지하기 위한 추가적인 데이터 조작 등이 대표적으로 낭비되는 비용

다. **업무규칙**(Business Rules)

업무규칙(Business Rules)을 데이터 모델에 표현하고 이를 해당 데이터 모델을 활용하는 모든 사용자가 공유할 수 있도록 제공

특히, 데이터 아키텍처에서 언급되는 논리 데이터 모델(Logical Data Model)에서 이러한 요소들이 포함되어야 함

예를 들면, 보험사의 직원들은 매월 여러 가지 항목에 대해서 급여를 지급받고 있고 이를 데이터로 관리하고 있다. 각 직원들은 월별로 하나 이상의 급여 항목(기본급, 상여금, 수당, 수수료, 등등)에 대해서 급여를 지급받는다. 여기에 더 나아가 각 직원은 사원 구분별(내근, 설계사, 계약직, 대리점 등)로 위의 급여 항목을 차등적으로 지급받는다는 업무규칙이 있다. 이러한 내용을 데이터 모델에 나타내야 한다.

☞ 해당 데이터 모델을 사용하는 모든 사용자(개발자, 관리자 등)가 해당 규칙에 대해서 동일한 판단을 하고 데이터를 조작 가능

라. **데이터 재사용**(Data Reusability)

데이터의 통합성과 독립성에 대해서 충분히 고려

과거 정보시스템의 데이터 구조의 가장 큰 특징은 데이터 모델이 별도로 존재하지 않고 애플리케이션의 부속품 정도로 인식

☞ 데이터 중복이 많이 발생하고 데이터의 일관성 문제가 심각하게 초래

해결 : 데이터가 애플리케이션에 대해 독립적으로 설계되어야만 데이터 재사용성을 향상

데이터 구조의 확장성, 유연성에 많은 노력을 기울이고 있음 → 업무 환경 변화에 대해서 유연하게 대응

근래의 많은 패키지 시스템들이 가지고 있는 데이터 모델들은 확장성을 강조하기 위해서 많은 부분을 통합한 데이터 모델의 형태

확장성을 담보하기 위해서는 데이터 관점의 통합이 불가피함

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 요소

기업이 관리하고자 하는 데이터를 합리적으로 균형이 있으면서도 단순하게 분류하는 것

마. **의사소통**(Communication)

데이터를 분석 과정에서는 자연스럽게 많은 업무 규칙들 도출

☞ 도출되는 많은 업무 규칙들은 데이터 모델에 엔터티, 서브타입, 속성, 관계 등의 형태로 최대한 자세하게 표현되어야 함

예를 들면, '사원' 테이블에는 어떠한 '사원구분'을 가지는 사원들이 존재하는지, '정규직', '임시직' 사원들이 같이 존재하는지, 아니면 또 다른 형태의 사원들이 존재하는지를 표현해야 한다. 더 나아가서 '호봉'이라는 속성은 '정규직'일 때에만 존재하는 속성인데 이러한 업무 규칙이 데이터 모델에 표현되어야 한다. 또한, 우리가 관리하는 사원들 중에서 '정규직' 사원들만이 '급여' 테이블

과 관계를 가진다. 이러한 부분은 개별 관계로 데이터 모델에 표현되어야 한다.

표현된 많은 업무 규칙들을 해당 정보시스템을 운용, 관리하는 많은 관련자들이 설계자가 정의한 업무 규칙들을 동일한 의미로 받아들이고 정보시스템을 활용할 수 있게 하는 역할
즉, 데이터 모델이 진정한 의사소통(Communication)의 도구로서의 역할

바. 통합성(Integration)

동일한 데이터는 조직의 전체에서 한번만 정의되고 이를 여러 다른 영역에서 참조, 활용하는 것
단, 물론 이 때에 성능 등의 부가적인 목적으로 의도적으로 데이터를 중복시키는 경우 존재가능

해결 : 동일한 성격의 데이터를 한 번만 정의하기 위해서는 공유 데이터에 대한 구조를 여러 업무 영역에서 공동으로 사용하기 용이하게 정의. 즉, 데이터 아키텍처의 중요성