

# 210727 데이터모델링

🕒 생성일	@2021년 7월 27일 오후 5:10
👤 성함	나한주
📌 수업 유형	이론
🕒 수정일	@2021년 7월 28일 오전 2:02
👤 작성자	현동빈

0교시. Introduction

1교시. 데이터 모델링의 정의

2교시. 구성요소 및 개념

Chapter 1 : 엔티티(Entity)

Chapter2 : 관계(Relationship)

Chapter3 : 속성(Attribute)

Chapter4 : 식별자(Key)

Chapter5 : 주제영역(Subject Area)

Chapter6 : 표현방법(Notation)

논리 데이터모델링의 절차

3교시. 작업절차 및 기본기법

Chapter1 : 작업절차

Chapter2 : 데이터의 개념

## 0교시. Introduction

### ▼ 데이터 모델링 관련 기본 Skill

- 데이터 모델 관련 개념 및 정의
  - 데이터 모델은 현실 세계에 대한 우리가 관심있는 대상을 데이터베이스화하기 위한 개념적 도구
- 데이터 모델 설계 기법

### ▼ 데이터 모델링 관련 고급 Skill

- 참조 데이터 모델 개요
- 참조 문서 및 학습 방법

### ▼ 소프트웨어의 본질은 동작과 데이터

- 소프트웨어는 프로그래밍 언어를 이용해서 데이터를 제어하는 수단
- 소프트웨어의 가장 중요한 요소인 데이터를 얼마나 잘 이해하는 지가 중요

### ▼ 소프트웨어는 데이터를 제어하기 위한 언어나 도구 사용

- 다양한 프로그래밍 언어와 개발도구를 사용
- CRUD로 대표되는 다양한 기능들 구현

### ▼ 소프트웨어 → 데이터 제어

- 이 CRUD는 결국 데이터를 제어하는 과정

### ▼ 데이터모델링의 본질 → 데이터

- 추상화, 단순화, 명확하게 만들어가는 과정

▼ 소프트웨어 본질의 하나 → 데이터

- 데이터가 얼마나 잘 만들어졌는가가, 소프트웨어의 품질에 많은 영향을 미침

▼ 더 나은 소프트웨어 개발 → 데이터모델링

- 데이터의 품질 → 소프트웨어의 품질

## 1교시. 데이터 모델링의 정의

▼ 정의

- 사용자의 요구사항 분석 후 IT 시스템을 구축하기 위하여 '데이터'요건을 **형상화**하는 작업

▼ 필요성

- 클라이언트의 요구사항 : 애매한 것이 많음
- 개발자들의 시스템 이해
- 데이터는 여러 업무에서 사용됨
- 어떤 데이터가 필요한지를 정리하고 명확히 커뮤니케이션 할 필요가 있음
- 개별 업무 시각에서 벗어나, 전체 시각에서 일관된 개념하에 일관된 방법으로 분석/정리하여야 함
- 애플리케이션과 데이터의 통합 : 애플리케이션의 통합은 반드시 데이터 통합이 선행되어야 함

▼ 모델링 기본 원칙

- 커뮤니케이션 원칙
- 모델링 상세화 원칙
- 논리적 표현 원칙

▼ 중요성 1 : **파급효과(Leverage)**

- 데이터 모델 변경 → 표준 영향 분석, 응용 변경 영향 분석 등 → 해당 분야의 실제적인 변경 작업 → 전체 시스템 구축 프로젝트에서 큰 위험요소

⇒ 시스템 구축이 완성되어 가는 시점에서 데이터 모델을 불가피하게 변경 시 큰 위험 요소

▼ 중요성 2 : **복잡한 정보 요구사항의 간결한 표현(Cincuseness)**

- 구축 할 시스템의 정보 요구사항과 한계를 가장 명확하고 간결하게 표현할 수 있는 도구
- 데이터 모델은 시스템을 구축하는 많은 이해관계자들의 훌륭한 의사소통 도구가 될 수 있음
- 데이터 모델 == 건축물의 설계도면

▼ 중요성 3 : **데이터 품질**

- 데이터 품질의 문제가 야기되는 중대한 이유 중 하나가 데이터 구조의 문제
- 잘못 설계된 모델링 → 데이터의 정확성이 떨어짐
- 데이터 품질 향상을 위한 잘 구성된 데이터 모델은 필수

▼ 데이터모델의 Level

- 통상 개념(Conceptual), 논리(Logical), 물리(Physical)의 3단계 구분
- 데이터 개념 → 개념모델 → 논리모델 → 업무 논리모델 → 물리모델
- 개념 데이터 모델링 : 주제별로 분류 가능한 업무를 분석해서, 핵심 엔터티를 추출하고 그들 간의 관계를 정의하여 데이터 전체 골격을 생성

- 논리 데이터 모델링 : 핵심 엔티티와 관계를 바탕으로 상세 속성을 정의하고 식별자를 확정하며, 정규화와 같은 상세화 과정을 수행
- 물리 데이터 모델링 : DBMS의 특성 및 구현 환경 등을 감안한 스키마를 일정한 기준과 규칙에 의해 도출하고 컬럼의 데이터 타입과 크기를 정의
- 데이터 개념 : 정보를 추상화하여 단순화 된 개념으로 분류한 것
- 개념모델 : 핵심엔티티 및 핵심엔티티간의 관계를 표현한 모델(key 속성만 포함)
- 논리모델 : 어플리케이션에 의한 데이터 사용 형태 및 구현상의 제약 사항을 고려하지 않고 작성된 모델
- 업무 논리모델 : 어플리케이션에 의한 데이터 사용 형태 및 구현상의 제약사항을 고려하여, 도출속성의 추가, 관계의 고정 및 데이터의 중복 등과 같은 비정규화 작업이 수행된 모델
- 물리모델 : 시스템의 기술요건 및 운영 요건을 반영한 모델

#### ▼ 데이터 모델링 주의점

- 중복
- 유연성
- 일관성

#### ▼ 데이터 모델링 어프로치

- 탑다운
- 바텀업
- 하이브리드

#### ▼ 좋은 데이터 모델의 요소

- 완전성
- 중복 배제
- 비즈니스 룰
- 데이터 재사용
- 안전성 및 확장성
- 간결성
- 의사소통
- 통합성

- 참고 : <https://augustines.tistory.com/50?category=706549>

## 2교시. 구성요소 및 개념

### Chapter 1 : 엔티티(Entity)

#### ▼ 데이터 모델의 3요소

- Entity(엔티티)
- Relationship(관계)
- Attribute(속성)

#### ▼ 엔티티의 정의

- 데이터 관리 대상이 되는 것(사물)의 종류(유형)를 도출하는 작업

- 참고 :

[http://databaser.net/moniwiki/pds/\\_ec\\_a3\\_bc\\_ec\\_a0\\_9c\\_ec\\_98\\_81\\_ec\\_97\\_ad/data\\_modeling01.pdf](http://databaser.net/moniwiki/pds/_ec_a3_bc_ec_a0_9c_ec_98_81_ec_97_ad/data_modeling01.pdf)

#### ▼ 엔티티의 특징

- 업무에서 필요한 정보를 갖는 것
- 업무를 수행함에 있어서 지속적으로 중요한 의미를 가진 것
- 구성 엔티티 어커런스를 식별자에 의해 유일하게 구별할 수 있어야 함
- 궁극적으로 속성 값으로 표현되는 관찰 가능한 특성을 갖고 있어야 함

#### ▼ 엔티티의 명명규칙

- 현장에서 보편적으로 사용하는 업무용어
- 명사 또는 명사구
- 모델에서 유일한 이름
- 뜻이 보존되는 한 이름을 짧게
- "~기본/상세/내역/이력/관계/목록"등으로 엔티티의 성격에 따라 명칭을 통일하는 경우도 있음

#### ▼ 엔티티 어커런스(Entity Occurance)

- 엔티티에 소속된 사물의 예
- = 엔티티 인스턴스
- 간혹 Entity라고도 함. 이 경우에는 Entity Type이 Entity, Entity는 Entity Occurance

#### ▼ Super Type 엔티티와 Sub Type 엔티티 (1/3)

- 여러 개의 엔티티가 공통 속성이 있고 일부의 속성이나 관계만 다를 경우, 공통 속성을 위한 하나의 엔티티 (Super Type)와 개별 속성을 위한 복수개의 엔티티(Sub Type)를 정의
  - Super type Key = Sub Type Key
  - Sub type의 엔티티 어커런스는 반드시 Super에 속하여야 함
  - 잠재적으로 null값을 가지는 속성은 Sub type으로 정의됨

#### ▼ Super Type 엔티티와 Sub Type 엔티티 (2/3)

- Super-Sub 관계를 좀 더 확장하여 정의가능, 관계가 포함관계/중복여부에 따라 세분됨
- 포함관계 : Super Type의 어커런스가 반드시 Sub Type의 어커런스로 존재하는 지 여부
  - Exhaustive(Total) : 반드시 Sub Type에 있음
  - Non-Exhaustive(Partial) : Sub Type에 없을 수도 있음
- 중복여부 : Super Type의 어커런스가 2개 이상의 Sub Type 어커런스가 될 수 있는지 여부
  - Disjoint(Exclusive) : 중복 불가능 → Sub Type 판단을 위한 Attribute가 하나임
  - Overlap : 중복 가능 → Attribute가 복수

#### ▼ Super Type 엔티티와 Sub Type 엔티티 (3/3)

- Super-Sub 관계 표기법은 다양
- 참고 : <https://gregorio78.tistory.com/162>

## Chapter2 : 관계(Relationship)

#### ▼ 정의

- 엔티티 간에 존재하는 상호간의 연관성을 "Cardinality(기수)"와 "Relationship Name(관계명)"으로 표기(행 위에 의한 관계, 소유에 의한 관계)
- 관계는 업무규칙(Business Rule)을 기술
- 관계는 데이터 상으로 관리 대상이 되는 관계 의미
- 두 엔티티 간의 관계는 2개 이상일 수 있음

#### ▼ 관계명(Relationship Name)

- 두 엔티티 사이의 관계 : 선으로 표시
- 관계선의 좌우 or 상하에 관계명 기록(시계 방향)

#### ▼ 관계명의 명명규칙

- 애매한 동사 사용 X
- 현재형 표현

#### ▼ 최대 기수(Cardinality) - 3가지 유형

- 두 개의 엔티티 간 관계에서, 한 엔티티의 한 어커런스를 기준으로 상대 엔티티의 관계된 어커런스가 최대 몇 개까지 있는가를 표현
- 1 : 1 관계 (one-to-one) -|—————|-
- 1 : 다 관계 (one-to-many) -|—————<-
- 다 : 다 관계 (many-to-many) : 물리적 구현 불가능, 엔티티를 추가하여 1:n 관계로 변형하여 구현 many-to-many 관계를 지양 →————←

#### ▼ 최소 기수 - 3가지 유형

- 두 개의 엔티티 간 관계에서, 한 엔티티의 한 어커런스를 기준으로 상대 엔티티의 관계된 어커런스가 적어도 몇 개까지 있을 수 있는가를 표현
- 0 : 발생할 수도 있고 발생 안 할 수 도 있는 경우 (Optional)
- 1(>0) : 꼭 발생해야 하는 경우 (Mandatory)
- 양방향 필수 (Fully Mandatory) -|—————|-
- 단방향 선택 (Partly Optional) -|—————o-
- 양방향 선택 (Fully Optional) -o—————o-

#### ▼ 관계 검증 1 - 중복 제거

- 원형관계 제거
- 사슬관계 제거

#### ▼ 관계 검증 2 - M:N 관계 단순화

- 관계를 별도 엔티티로 분리 - 숨겨진 정보로부터 연관 엔티티를 추출

#### ▼ 관계 검증 3 - 1:1 관계 정제

#### ▼ 데이터 모델링 과정에서 1:1로 대응 → 해당 엔티티들의 기본키가 동일하다는 의미 → 3가지 방법 표현가능

- 하나로 통합
  - Mandatory 관계
  - 기본키가 동일

- 대부분의 속성과 관계가 동일
- 그대로 유지
  - Optional 관계이면서 대부분의 속성 및 관계가 상이한 경우
  - 대응되는 속성의 값이 서로 상이한 경우
  - 새로운 Super Type 엔티티 추가
- 새로운 Super Type 엔티티 추가
  - 기본키가 동일하고 속성의 일부만 상이한 경우
- ▼ 관계별 엔티티 종류 (관계를 기준으로 한 엔티티 종류)
  - 기본형 엔티티 (Fundamental Entity)
  - 속성형 엔티티 (Attributive Entity)
  - 관련형 엔티티 (Associative Entity)
- ▼ 다 대 다 관계 해소
  - 개념적 설계에서 허용되는 M:M 관계와 다중값 속성이, 논리적 설계에서의 관계형 데이터베이스에서는 하나의 속성에 오직 하나의 값만 허용하기 때문에 논리 데이터 모델에서는 M:M관계가 해결되어야 함

## Chapter3 : 속성(Attribute)

- ▼ 정의
  - 엔티티가 관리해야 되는 데이터 항목으로 더 이상 분리될 수 없는 최소의 데이터 보관 단위
  - 향후 물리 데이터 모델링 작업을 통하여 컬럼으로 변환됨
- ▼ 명명규칙
  - 해당 업무에서 사용하는 이름 부여
  - 엔티티에서 유일하게 식별가능 하도록 지정
  - 가능하면 약어 사용 X → 실제로는 약어가 많이 쓰임
  - "수식어 + 도메인 명" 형식을 사용하여 표준화
- ▼ 속성 타입
  - 근본적으로는 속성과 같은 의미
  - 엔티티가 갖는 속성의 유형 의미
- ▼ 속성 인스턴스
  - 속성 타입의 값
- ▼ 도메인
  - 속성이 가질 수 있는 값에 대한 업무적 제약요건으로부터 파악된 특성
  - 속성에 대한 데이터 타입과 크기, 제약사항을 지정한 것
  - 활용(중요성)
    - 속성 값의 비교 기준
    - 공통 검증 규칙
  - 어딘 한 속성에 허용되는 업무적으로 의미 있는 인스턴스 값의 집합
  - 열거형 도메인 : 코드와 같이 인스턴스 값이 정해져 있으며, 개수가 한정되어 있는 것으로 업무의 관리체제가 변하지 않는 한 인스턴스는 일정함

- 제약형 도메인 : 값의 범주가 정해져 있음, 특정한 값으 정할 수 없고 개수 제한 없음
- 구성요소 : 도메인명, 데이터 타입, 자릿수, 인스턴스 기술서(열거형 도메인)

#### ▼ 속성의 배치 1

- 기본키에 종속되는 속성을 해당 엔티티에 배치
- 가능한 범위 내에서 부모 엔티티 쪽으로 배치
- 속성이 여러 개의 값을 가질 수 있는 경우, 새로운 자식 엔티티를 만들어 자식 엔티티의 속성으로 배치

#### ▼ 속성의 배치 2

- 관계를 구체화 시켜주는 속성이 존재 → 관계를 엔티티로 전환시켜 해당 속성 배치

#### ▼ 속성의 배치 3

- Super Type 엔티티, Sub Type 엔티티 내의 속성 배치
  - Super Type 엔티티에 Sub Type 엔티티의 구분항목을 속성으로 등록
  - Sub Type 엔티티 모두에 포함되는 속성을 Super Type 엔티티로 이동

## Chapter4 : 식별자(Key)

#### ▼ 정의

- 한 엔티티 내에서 각각의 어커런스를 유일하게 구분할 수 있는 단일 속성 또는 속성 그룹
- 식별자는 데이터베이스에서 조건에 만족하는 관계의 행을 찾거나 순서대로 정렬할 때 다른 행과 구별할 수 있는 유리한 기준이 되는 속성의 집합

#### ▼ 후보키(Candidate Key)

- 엔티티 내에서 각각의 어커런스를 유일하게 구분할 수 있는 속성으로 하나 또는 하나 이상의 속성으로 구성
- 기본키(PK)가 될 수 있는 후보 속성 → 유일성 & 최소성 동시에 만족

#### ▼ 기본키(Primary Key)

- 어느 엔티티 어커런스를 식별하기 위한 속성
- 후보키들 중에서 하나를 선택한 키로 최소성(짧은)과 유일성(Unique)을 만족하는 속성
- 기본키는 NULL 값을 절대 가질수 없음(Not Null)
- 가능하면 업무적으로 활용도가 높은 것으로 정의
- 가능하면 변하지 않아야 함(Never Change)
- 기본키는 여러개 지정 가능(굳이 복수 지정 필요 없음)

#### ▼ 대체키(Alternate Key)

- 후보키 중 기본키로 선정되지 않은 속성

#### ▼ 복합키(Concatenate Key)

- 2개 이상의 속성으로 기본키(Primary Key)를 구성한 식별자(Key)

#### ▼ 외부키(Foreign Key)

- 어떤 엔티티 A가 다른 엔티티 B와 관계를 갖는 경우, B 엔티티와 관계때문에 속성으로 갖게 되는 B엔티티의 기본키(Primary Key)
- Super Type 먼저 생성 → Sub Type
- Sub Type 삭제 → Super Type 삭제(Super Type 삭제 → Sub Type 삭제 불가)

▼ 합성키(Compound Key)

- 엔티티와 엔티티의 관계가 Many to Many 관계일 때 그 관계를 나타내는 엔티티의 Key

## Chapter5 : 주제영역(Subject Area)

▼ 정의

- 엔티티를 도출, 관리하는 단위 영역
- 일반적으로 데이터 중심으로 정의하나, 경우에 따라서 업무 흐름 중심으로 정의
- 계층 존재
- 최하위 주제영역은 ER Diagram의 작도 단위

▼ 명명규칙

- 현업에서 보편적으로 사용하는 업무용어
- 데이터의 그룹을 의미하는 이름 부여
- 계층표현 "xxx\_yyy"형식으로 사용하기도 함

## Chapter6 : 표현방법(Notation)

▼ 데이터 모델의 표현

- Martin/IE/Crow's Foot
- Min-Max/ISO
- UML

## 논리 데이터모델링의 절차

▼ 주제영역 정의

- 주제영역 도출
- 주제영역 명확화

▼ 실체 정의

- 실체 도출
- 실체 명확화
- 식별자 정의

▼ 속성 정의

- 속성 도출
- 속성 명확화
- 정규화

▼ 관계 정의

- 관계 도출
- 관계 명확화
- 특수유형 관계 정의

▼ 업무규칙 정의

- 키 업무규칙 정의



- 속성 업무규칙 정의
- ▼ 이력 관리
  - 시점 이력 관리
  - 기간 이력 관리
- ▼ 모델 통합
  - 수평적 통합
  - 수직적 통합
  - 전사 모델 통합

## 3교시. 작업절차 및 기본기법

### Chapter1 : 작업절차

#### ▼ 데이터 모델링 작업 흐름

- 데이터 모델링 : 주어진 개념으로부터 논리적인 데이터 모델을 구성하는 작업을 말하며 일반적으로 이를 물리적인 데이터베이스 모델로 환원하여 고객의 요구에 따라 특정 정보 시스템의 데이터베이스에 반영하는 작업 포함 (후자의 의미 : 데이터베이스 모델링)
- 데이터 개념 정의 : 전사 통합 데이터 모델과 같은 대규모의 데이터모델을 개발 할 때 선행 필요
- 전사 데이터 모델 참고 : <https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=kevinjung423&logNo=221457829407>
- 개념 모델 개발 : 대규모 통합 데이터 모델 개발 시, 전체 모델의 구조 및 핵심 엔티티간 관계 정의 필요
- 논리 데이터 모델 개발 - 개념 모델을 상세화 & As-Is 모델을 통한 보완작업 과정을 거쳐 완성
- 논리 데이터 모델 참고 : <https://dataonair.or.kr/db-tech-reference/d-guide/da-guide/?pageid=5&mod=document&uid=285>

#### ▼ 개념 데이터 모델링 작업 절차

개념 데이터 모델 : 주제별로 분류 가능한 업무를 분석해서, 핵심 엔티티를 추출하고 그들 간의 관계를 정의하여 데이터의 골격을 생성하는 것 (참고 : <https://augustines.tistory.com/51>)

##### 1. Sub-Type의 도출

- 데이터 개념별로 Sub-Type 도출

##### 2. 핵심 엔티티 도출

- Sub-Type을 핵심 엔티티로 도출

##### 3. 핵심 엔티티 간 관계 정의

- 핵심 엔티티간 관계 분석
- 필요시 Relationship Entity 도출

##### 4. ERD(Entity Relationship Diagram) 작성

- 데이터 개념별로 Sub-Type Entity, Relationship Entity 간 관계를 ERD로 작성
- ERD
  - ERD는 말로서 되어있는 요구분석사항을 그림으로 그려내어 그 관계를 도출하는 것

- ERM(ERM: Entity-Relationship Modelling : 개체 관계 모델링) 참고 : <https://huskdoll.tistory.com/541>

#### 5. 주요속성 도출 및 Entity 정제

- 엔티티의 Key 및 주요 속성 정의
- 필요시 속성형 엔티티 도출(주요 속성형 엔티티로 한정)

#### ▼ 논리 데이터 모델링 작업 절차

논리 데이터 모델링 목적 및 효과 (참고 : <https://augustines.tistory.com/54?category=706549>)

- 해당 비즈니스에 대한 데이터 관점에서의 **명확한 이해**를 도움
- **전사적인** 통합 데이터 체계 확립
- 데이터의 **일관성** 및 **정확성** 유지를 위한 규칙을 도출
- **안정적인** 데이터베이스 설계의 토대를 마련
- 사용자와의 **명확한 의사소통**을 위한 수단으로 활용

#### 1. 속성 도출

- 상세 속성 도출
- 필요시 속성형 엔티티 도출

#### 2. 정규화 검토(참고 : <https://mangkyu.tistory.com/110>)

- 제 1, 2, 3 정규화 수행 (정규화 : 쪼개기)
- 필요시 엔티티 분리

#### 3. 속성 정의

- 속성의 의미를 정의하고 도메인을 지정 또는 생성함

#### 4. 현행 시스템 정보 분석

- 현행 시스템 정보 반영 여부 검토
- 필요시 엔티티/속성 보완

## Chapter2 : 데이터의 개념

#### ▼ 데이터 개념의 필요성

- 개념 데이터를 정의하지 않고 엔티티를 도출해서는 고품질의 데이터 모델을 작성할 수 없음
- Ex) FSDM(Financial Services Data Model)