

## 1과목 1장 데이터 모델링의 이해

### ● 데이터 모델링의 특징

- 추상화 : 현실세계를 간략하게 표현
- 단순화 : 누구나 쉽게 이해할 수 있도록 표현
- 명확성 : 명확하게 의미가 해석되어야 하고 한 가지 의미를 가져야 한다.

### ● 데이터 모델링의 중요성 및 유의점

- 중복 : 같은 시간 같은 데이터 제공
- 비유연성 : 사소한 업무변화에 데이터모델이 수시로 변경X
- 비밀관성 : 데이터와 데이터 간 상호 연관 관계에 대해 명확하게 정의해야 함(데이터 중복이 없더라도 비밀관성은 발생)

### ● 데이터 모델링의 3요소

어떤 것(Things), 성격(Attributes), 관계(Relationship)

### ● 데이터 모델링

#### (1) 개념적 모델링

전사적 관점 / 추상화 수준이 가장 높은 모델링  
업무 측면에서의 모델링

#### (2) 논리적 모델링

특정 데이터베이스 모델에 종속  
정규화를 통해 재사용성을 높음  
식별자를 정의하고 관계, 속성 등을 모두 표현

#### (3) 물리적 모델링

DBMS에 테이블, 인덱스 등을 생성하는 단계  
성능, 보안, 가용성 등을 고려하여 데이터베이스를 구축

### ● 데이터 모델링 관점

- 데이터 : 구조 분석, 정적 분석
- 프로세스 : 시나리오 분석, 도메인 분석, 동적 분석
- 데이터와 프로세스 : CRUD(create, read, update, delete)분석  
프로세스와 데이터 간의 관계

### ● 데이터 모델 표기법

1976년 피터첸이 Entity Relationship Model 개발

### ● 데이터 모델링의 중요성

파급효과, 복잡한 정보 요구사항의 간결한 표현, 데이터품질

### ● ERD 작성 절차

(1)엔터티 도출 → (2)엔터티 배치 → (3)엔터티 간의 관계설정 → (4)관계 참여도 표현 → (5)관계의 필수 여부 표현

### ● ERD 작성 시 고려사항

중요한 엔터티를 가급적 왼쪽 상단에 배치한다  
ERD는 이해가 쉬워야 하고 복잡하지 않아야 한다

### ● 데이터 독립성

- 논리적 독립성 : 논리적 구조가 변경되어도 응용 프로그램에는 영향 x / 개념 스키마가 변경되어도 외부 스키마에는 영향 x
- 물리적 독립성 : 저장장치의 구조변경을 응용프로그램과 개념스키마에 영향 x / 내부 스키마가 변경되어도 외부/개념 스키마에는 영향x

✓ Mapping(사상) : 상호 독립적인 개념을 연결해주는 다리

### ● 3층 스키마 구조

#### (1) 외부 스키마

사용자 관점, 업무상 관련이 있는 데이터  
관련 DB의 뷰(View)를 표시  
응용 프로그램이 접근하는 DB를 정의

#### (2) 개념 스키마

설계자 관점, 사용자 전체 집단의 DB 구조  
전체 DB 내의 규칙과 구조를 표현  
통합 DB 구조

#### (3) 내부 스키마

개발자 관점, DB의 물리적 저장구조  
데이터 저장구조, 인덱스, 레코드 구조, 필드 정의 등을 의미

### ● 좋은 데이터 모델의 요소

- 완전성 : 업무에 필요한 모든 데이터가 모델에 정의
- 중복배제 : 하나의 DB내에 동일한 사실은 한 번만
- 업무규칙 : 많은 규칙을 사용자가 공유하도록 제공
- 데이터 재사용 : 데이터가 독립적으로 설계되어야 함
- 의사소통 : 업무규칙은 Entity, Subtype, Attribute, Relationship 등의 형태로 최대한 자세히 표현
- 통합성 : 동일한 데이터는 한 번만 정의, 참조활용

### ● 엔터티

업무에 필요하고 유용한 정보를 저장하고 관리하기 위한 집합적인 것, 보이지 않는 개념 포함 / 개념, 사건, 장소 등의 명사

### ● 엔터티 특징

- 유일한 식별자가 있어야 한다
- 2개 이상의 인스턴스가 있어야 한다
- 반드시 속성을 가지고 있어야 한다
- 다른 엔터티와 최소 한 개 이상 관계가 있어야 한다  
(단, 통계성 엔터티, 코드성 엔터티 같은 경우는 관계 생략 가능)
- 업무에서 관리되어야 하는 집합(관련있는 집합)이어야 한다

## ● 엔터티 분류(유형과 무형 엔터티)

### (1) 유형 엔터티

업무에서 도출, 안정적 사용, 지속적 사용, 물리적 형태  
ex) 사원, 물품, 강사

### (2) 개념 엔터티

개념적 정보, 물리적 형태X  
ex) 보험상품

### (3) 사건 엔터티

비즈니스 프로세스를 실행하면서 생성되는 엔터티  
통계자료 이용  
ex) 주문, 청구

## ● 엔터티 분류(발생시점에 따른 분류)

### (1) 기본 엔터티 (키 엔터티 라고도 불린다)

다른 엔터티로부터 영향 받지 않고 독립적으로 생성  
ex) 고객, 상품, 부서

### (2) 중심 엔터티

기본엔터티와 행위 엔터티 중간에 있는 것  
기본엔터티로부터 발생되고 행위엔터티를 생성하는 것  
ex) 계좌, 주문, 취소, 체결

### (3) 행위 엔터티

2개 이상의 엔터티로부터 발생  
자주 바뀌거나 양이 증가  
ex) 주문 이력, 체결 이력, 사원의 부서변경 이력

## ● 엔터티 이름 부여 시 주의 사항

업무에서 사용하는 용어 사용, 약어 금지, 단수명사 사용, 고유한 이름 사용, 생성의미대로 부여

## ● 속성

엔터티가 가지는 항목, 의미적으로 더 이상 분리되지 않는 단위

업무에 필요한 데이터 저장, 인스턴스의 구성 요소

## ● 속성의 특징

한 개의 속성은 하나의 속성값만 가진다

주식별자에게 함수적으로 종속된다. 즉, 기본키가 변경되면 속성의 값도 변경된다

## ● 속성의 종류

### (1) 기본속성

업무로부터 추출한 모든 일반적인 속성  
ex) 회원ID, 이름, 계좌번호, 주문일자

### (2) 설계속성

데이터 모델링 과정에서 발생하는 속성  
유일한 값을 부여  
ex) 상품코드, 지점코드

### (3) 파생 속성

다른 속성에 의해서 만들어지는 속성 ex)합계, 평균

✓ **도메인** : 속성이 가질 수 있는 값의 범위 ex)4글자

## ● 속성 이름 부여 시 주의사항

업무에서 사용하는 용어 사용, 서술식 속성명 금지, 약어 금지, 전체 데이터모델에서 유일성 확보

## ● 관계

엔터티 간의 관련성을 의미하여 존재 관계와 행위 관계로 분류

## ● 관계의 종류

존재관계 : 엔터티 간의 상태를 의미 ex)소속된다

행위관계 : 엔터티 간에 어떤 행위가 있는 것

ex) 계좌를 사용해서 주문을 발주하는 관계

ERD는 존재관계와 행위관계를 구분하지 않음

UML에서는 구분함(연관관계는 실선, 의존관계는 점선)

연관관계(=존재관계)는 항상 이용하는 관계

의존관계(=행위관계)는 상대방 행위에 의해 발생하는 관계

## ● 관계 차수

두 개의 엔터티 간에 관계에 참여하는 수

ex) 1:1 / 1:M / M:N

### (1) 1:1 관계

완전 1:1 관계 : 엔터티 간의 관계가 하나 있는 경우 (무조건)

선택적 1:1 관계 : 관계가 하나이거나 없을 수도 있음

### (2) 1:N 관계

엔터티에 행이 하나 있을 때 다른 엔터티의 값이 여러 개 있는 관계

### (3) M:N 관계

두 개 엔터티가 서로 여러 개의 관계를 가지고 있는 것

관계형 DB에서 M:N 관계는 카테시안 곱이 발생한다

### (4) 필수적 관계와 선택적 관계

필수적 관계 : 고객은 반드시 있어야 계좌를 개설할 수 있음

선택적 관계 : 고객은 있지만 계좌는 없을 수도 있음

## ● 관계 체크사항

- 2개의 엔터티 사이에 관심있는 견관 규칙이 있는가?

- 2개의 엔터티 사이에 정보의 조합이 발생하는가?

- 업무기술서, 장표에 관계연결에 대한 규칙 서술이 있는가?

- 업무기술서, 장표에 관계연결을 가능케하는 동사가 있는가?

## ● 식별자

엔터티 내에서 인스턴스를 구분하는 구분자

식별자는 논리 데이터 모델링 단계에 사용

Key는 물리 데이터 모델링 단계에 사용

### ● 키의 종류

- (1)기본키 : 후보키 중에서 엔티티를 대표할 수 있는 키
- (2)후보키 : 유일성과 최소성을 만족하는 키
- (3)슈퍼키 : 유일성은 만족하나 최소성(not null)을 만족하지 않는 키
- (4)대체키 : 여러 개의 후보키 중에서 기본키를 선정하고 남은 키

### ● 식별자의 종류 - 식별자의 대표성

- (1) 주식별자 : 유일성과 최소성을 만족하면서 엔티티를 대표하는 식별자, 다른 엔티티와 참조 관계로 연결될 수 있다
- (2) 보조식별자 : 유일성과 최소성을 만족하지만 대표성을 만족하지 못하는 식별자

### ● 식별자의 종류 - 생성 여부

- (1) 내부식별자 : 스스로 생성되는 식별자  
ex) 부서코드, 주문번호
- (2) 외부식별자 : 다른 엔티티에서 가져오는 식별자  
ex) 계좌 엔티티의 회원ID (회원엔티티에서 가져옴)

### ● 식별자의 종류 - 속성의 수

- (1) 단일식별자 : 하나의 속성으로 구성
- (2) 복합식별자 : 두 개 이상의 속성으로 구성

### ● 식별자의 종류 - 대체 여부

- (1) 본질식별자 : 업무에 의해 만들어지는 식별자
- (2) 인조식별자 : 인위적으로 만들어지는 식별자

### ● 주식별자 도출 기준

- 해당 업무에서 자주 이용되는 속성
- 명칭, 내역 등과 같이 이름으로 기술되는 것들은 x
- 복합으로 주식별자를 구성할 경우 너무 많은 속성 x

### ● 식별 관계와 비식별 관계 - 식별 관계

- 부모로부터 받은 식별자를 자식엔티티의 주식별자로 이용
- 강한 연결관계 표현, 실선 표기
- 강한 개체는 누구에게도 지배되지 않는 독립적인 개체
- 강한 개체는 다른 엔티티에게 기본 키를 제공
- 강한 개체는 식별 관계로 표현
- 강한 개체의 기본키 값이 변경되면 식별 관계에 있는 자식 엔티티의 값도 변경된다.

### ● 식별 관계와 비식별 관계 - 비식별 관계

- 부모에게 받은 식별자를 자식엔티티에서 일반 속성으로 사용
- 부모 없는 자식이 생성될 수 있는 경우
- 부모와 자식의 생명주기가 다른 경우
- 여러개의 엔티티가 하나의 엔티티로 통합되어 표현되었는데 각각의 엔티티가 별도의 관계를 가진 경우
- 자식에 별도의 주식별자를 생성하는 것이 더 유리한 경우
- SQL 문장이 길어져 복잡성 증가되는 것 방지
- 약한 연결관계 표현, 점선 표기
- 자식의 주식별자 구성을 독립적으로 구성