

1. 데이터 모델링의 이해

A. 엔터티

i. 엔터티의 개념 : 대상간 동질성을 갖는 인스턴스나 행위의 집합

ii. 엔터티의 특징

1. 해당 업무에서 필요하고 "관리하고자 하는 정보"
2. "유일한 식별자"에 의해 인스턴스 식별 가능
3. 영속적으로 존재하는 "인스턴스의 집합 (2개 이상)"
4. 업무 프로세스에 의해 이용되어야 함. 즉, CRUD 발생하지 않는지 판단
5. 속성이 있어야 함(주 식별자만 존재해도 적절하지 않음)
6. 다른 엔터티와 최소 1개 이상의 "관계"가 있어야 함

[데이터 모델링을 하면서 관계를 생략해 표현해야 하는 경우]

- 통계를 위한 엔터티
- 코드를 위한 엔터티 : 너무 많은 엔터티와 엔터티간이— 고나계 설정으로 인해 데이터 모델의 읽기 효율성 저하
- 시스템 처리시 내부 필요에 의한 엔터티(ex, 트랙잭션 로그 테이블 등)

iii. 엔터티의 분류

1. 유무형에 따른 분류

- 유형 엔터티 : 물리적 형태, 업무로부터 구분 쉬움
- 개념 엔터티 : 물리적 존재 X, 관리해야 할 개념적 정보로 구분 가능
- 사건 엔터티 : 업무 수행함에 따라 발생하는 엔터티, 발생량이 많고 통계자료에 이용될 수 있음

2. 발생시점에 따른 분류

- 기본/키 엔터티 : "업무에 원래 존재하는" 정보, 다른 엔터티와의 관계에 의해 생성되지 않고 독립적으로 생성됨, 타 엔터티의 부모 역할, 다른 엔터티로부터 주식별자 상속받지 않고 자신의 고유 주식별자를 가짐
- 중심 엔터티 : 기본 엔터티로부터 발생됨, 업무에 있어 중심적인 역할

- 행위 엔터티 : 두 개 이상의 부모 엔터티로부터 발생, 자주 내용이 바뀌거나 데이터량이 증가함

3. 스스로 생성 여부 : 독립 엔터티, 의존 엔터티

iv. 엔터티의 명명

- 현업 업무에서 사용하는 용어
- 약어 X
- 단수 명사 사용
- "유일"하게 이름 부여
- 생성 의미대로 이름 부여(업무 목적에 따라 생성)

B. 속성 : 업무에서 필요로 하는 인스턴스의 구성요소, 관리하고자 하는 의미상 더 이상 분리되지 않는 최소의 데이터 단위

i. 엔터티, 인스턴스, 속성, 속성값의 관계

- 한 개의 엔터티는 두 개 이상의 인스턴스의 집합
- 한 개의 엔터티는 두 개 이상의 속성을 가짐
- 한 개의 속성은 한 개의 속성값을 가짐

ii. 속성의 표기법 : 엔터티 내에 이름을 포함해 표현

iii. 속성의 특징

- 해당 업무에서 필요로하고 관리하고자 하는 정보
- 정규화 이론에 근거해 정해진 주식별자에 함수적 종속성을 가져야 함
- 하나의 속성에는 한 개의 값만 가짐

iv. 속성의 분류

1. 속성의 특성에 따라

- 기본 속성 : 업무로부터 추출한 모든 속성이 해당

업무로부터 분석한 속성이라도 이미 업무 코드로 정의한 속성이 많지만

속성값이 원래 속성을 나타내지 않으므로 기본 속성이 아님

- 설계 속성 : 업무상 필요한 데이터 외 "데이터 모델링"을 위해, "업무를 규칙화"하기 위해 속성을 새로 만들거나 변형해 정의
- 파생 속성 : 다른 속성에 영향을 받아 발생하는 속성, 주로 계산값

다른 속성에 영향을 받기 때문에 "데이터 정합성 유지"하기 위해 유의점이 많음, 따라서 가급적 파생 속성을 적게 사용하는 것이 좋음

2. 엔터티 구성 방식에 따른 분류

- PK 속성 : 엔터티를 식별할 수 있음
- FK 속성 : 다른 엔터티와의 관계에서 포함된 속성
- 일반 속성 : 엔터티에 포함되고 PK, FK에 포함되지 않는 속성
- 복합 속성 : 세부 의미 쪼갤 수 있음
- 단순 속성 : 세부 의미 쪼갤 수 없음
- 단일 값 : 속성 1개에 하나의 값을 가지는 경우(ex, 주민번호)
- 다중 값 : 속성 1개에 여러 개의 값을 가지는 경우(ex, 전화번호: 집 전화, 휴대폰 번호 등) -> "1차 정규화" 필요

v. 도메인 : 각 속성이 가질 수 있는 값의 범위

vi. 속성의 명명 : client/server 환경이나 web 환경에서 속성명이 곧 UI에 나타나기 때문에 업무와 직결되는 항목, 따라서 속성 이름 정확하게 부여하기 위해 "용어 사전"이라는 업무 사전을 프로젝트에 사용 + 속성이 가지는 값의 종류와 범위를 명확하게 하기 위해 "도메인 정의" 미리 해 용어사전과 같이 사용

➔ 용어적 표준과 데이터 타입의 일관성 확보 가능

- 현업에서 사용하는 이름 부여
- 명사형 이용
- 약어 X
- 반정규화 적용시 속성명의 충돌을 방지하기 위해 유일하게 작성하는 것이 좋음

C. 관계 : 엔터티의 인스턴스 사이의 논리적인 연관성으로, 존재의 형태로서나 행위로서 서로에게 연관성이 부여된 상태

유의점 : 엔터티 안에 인스턴스가 개별적으로 관계를 가지는 것은 "페어링", 페어링의 집합을 관계로 표현함

각 엔터티의 인스턴스들이 자신이 관련된 인스턴스들과 관계의 어커런스로 참여하는 형태를 관계 페어링이라고 함

i. 관계의 분류 : 연결함에 있어 어떤 목적으로 연결되었느냐에 따라 분류

- 존재에 의한 관계(부서, 사원), 행위에 의한 관계(고객, 주문)
- UML : ERD에서는 존재적 관계와 행위에 의한 관계를 구분하지 않고 표현하지만, 클래스 다이어그램에서는 구분해 표현
 - 연관관계 : 항상 이용하는 관계, 존재적 관계, 실선으로 표시, 소스 코드에서 멤버변수로 선언해 사용
 - 의존관계 : 상대방 클래스의 행위에 의해 관계가 형성될 때 구분하여 표현, 점선으로 표현, 행위를 나타내는 코드인 method에서 파라미터 등으로 이용

ii. 관계의 표기법 : 관계명, 관계차수, 관계선택사양

1. 관계명 : 엔터티가 관계에 참여하는 형태 지칭

- 엔터티에서 관계가 시작되는 편 : 관계시작점
- 관계를 받는 편 : 관계끝점
- 관계 시작점과 끝점 모두 관계이름을 가져야 함
- 참여자의 관점에 따라 관계이름이 능동적이거나 수동적으로 명명됨
- 애매한 동사를 피하고, 현재형으로 표현

2. 관계차수 : 두 개의 엔터티간 관계에서 참여자의 수를 표현하는 것

- 1:1(one to one)
- 1:M(one to many)
- M:M(many to many)

3. 관계선택사양

- 필수참여 : 참여하는 모든 참여자가 반드시 관계를 가지는, 타 엔터티의

참여자 연결이 되어야 하는 관계(ex, 주문서는 반드시 주문목록을 가져야 함)

- 선택참여 : 물리속성에서 FK로 연결될 경우 NULL을 허용할 수 있는 항목이 됨, ERD에서 관계를 나타내는 선에서 선택참여하는 엔터티 쪽을 원으로 표시(ex, 목록은 주문이 될 수도 있고 주문이 되지 않을 수도 있음, 따라서 목록과 주문목록과의 관계는 선택참여)

iii. 관계 체크사항

- 두 개의 엔터티 사이에 관심있는 연관규칙이 존재?
- 두 개의 엔터티 사이에 정보의 조합이 발생?
- 업무기술서, 장표에 관계연결에 대한 규칙이 서술되어 있는가?
- 업무기술서, 장표에 관계연결을 가능하게 하는 동사가 있는가?

iv. 관계 읽기

- 기준 엔터티를 한 개 또는 각으로 읽음
- 대상 엔터티의 관계참여도 개수를 읽음
- 관계선택사양과 관계명을 읽음

D. 식별자 : 하나의 엔터티에 구성되어 있는 여러 개의 속성 중 엔터티를 대표하는 수 있는 속성을 의미, 하나의 엔터티는 "반드시" 하나의 유일한 식별자가 존재해야 함

(식별자는 업무적으로 구분이 되는 정보로 생각할 수 있으므로 "논리 데이터 모델링" 단계에서 사용하고, 키는 데이터베이스 테이블에 접근을 위한 매개체로서 "물리 데이터 모델링" 단계에서 사용함)

i. 식별자의 특징

1. 주식별자의 특징

유일성 : 주식별자에 의해 엔터티내에 모든 인스턴스들을 "유일하게 구분"

최소성 : 주식별자를 구성하는 속성의 수는 유일성을 만족하는 "최소의 수"가 되어야 함

불변성 : 주식별자가 한 번 특정 엔터티에 지정되면 그 식별자의 값은 "변하지 않아야 함"

존재성 : 주식별자가 지정되면 반드시 데이터 값이 존재(NULL 안됨)

2. 외부식별자의 경우 주식별자 특징과 일치하지 않으며 “참조무결성 제약조건”에 따른 특징을 가짐

ii. 식별자 분류

1. 자신의 엔터티 내에서 대표성을 가지는가 : 주식별자, 보조식별자
 - 주식별자 : 엔터티 내에서 각 어커런스를 구분할 수 있는 구분자, 타 엔터티와 참조관계를 연결할 수 있는 식별자
 - 보조식별자 : 엔터티 내에서 각 어커런스를 구분할 수 있는 구분자이나 대표성을 가지지 못해 참조관계 연결을 못함
2. 엔터티 내에서 스스로 생성되었는가 : 내부식별자, 외부식별자
 - 내부식별자 : 엔터티 내부에서 스스로 만들어지는 식별자
 - 외부식별자 : 타 엔터티와의 관계를 통해 타 엔터티로부터 받아오는 식별자
3. 단일 속성으로 식별이 되는가 : 단일식별자, 복합식별자
 - 단일식별자 : 하나의 속성으로 구성된 식별자
 - 복합식별자 : 둘 이상의 속성으로 구성된 식별자
4. 원래 의미있던 식별자 속성을 대체하였는가 : 본질식별자, 인조식별자
 - 본질식별자 : 업무에 의해 만들어지는 식별자
 - 인조식별자 : 업무적으로 만들어지지는 않지만 원조식별자가 복잡한 구성을 가지고 있기 때문에 인위적으로 만든 식별자

iii. 주식별자 도출기준

1. 해당 업무에서 자주 이용되는 속성을 지정
2. 명칭, 내역 등과 같이 이름으로 기술되는 것들은 가능하면 주식별자로 지정하지 않음(조건절에 정확한 부서이름을 기술하기 어렵기 때문)
3. 복합으로 주식별자를 구성할 경우 너무 많은 속성이 포함되지 않도록 함

iv. 식별자관계와 비식별자관계에 따른 식별자

v.