용어 정리

논리적->물리적

엔티티->테이블

속성->컬럼

인스턴스->행

### 모델링

모델링 개념: 비즈니스 프로세스와 요구 사항을 추상적, 구조화된 형태로 표현하는 과정

DB의 구조, 관계를 정의해 데이터의 저장, 조작, 관리 방법을 명확하게 정의

모델링의 특징

* 단순화

핵심 요소에 집중하고 불필요한 세부 사항 제거

* 추상화

일정한 형식에 맞추어 간략하게 대략적으로 표현

* 명확화

애매모호함을 최대한 제거하고 정확하게 현상을 기술하는 과정

모델링 유의사항

* 중복

테이블에 같은 정보를 저장하지 않도록 설계

* 비유연성

잦은 모델 변경이 되지 않게 설계. 데이터 정의와 프로세스의 분리

* 비일관성

정보가 모순되거나 상반된 내용을 갖는 상태

관계를 명확히 정의, 품질 관리 필요

중복이 없더라도 비일관성은 발생할 수 있음

3가지 요소: 대상, 속성, 관계(Entity, Attribute, Relationship)

데이터 모델링의 3단계

* 개념적 모델링

업무 중심적, 포괄적인 수준의 모델링

추상화 수준이 가장 높음

업무 분석 후 핵심 엔티티를 추출하는 단계

관계를 표현하기 위해 ERD 작성

* 논리적 모델링

개념적 모델링을 토대로 세부속성, 식별자, 관계 등을 표현

구조를 정의하기 때문에 비슷한 업무, 프로젝트에서 동일한 형태의 데이터 사용 시 재사용 가능

데이터 정규화 수행

* 물리적 모델링

논리 모델링 후 직접 물리적으로 생성하는 과정

DB 성능, 디스크 저장구조, 하드웨어 보안성, 가용성을 고려

가장 구체적인 모델링

표기법: ERD(Entity Relationship Diagram)

엔티티, 관계를 시각적으로 표현한 다이어그램

엔티티 도출, 배치, 관계 설정, 관계명 서술, 참여도 기술, 관계 필수 여부 확인

### 엔티티

개념: 현실 세계에서 독립적으로 식별 가능한 객체, 사물

업무상 분석해야 하는 대상으로 이루어진 집합

인스턴스틑 특정한 속성 값으로 구성됨

특징

1. 유일한 식별자에 의해 식별
2. 업무에 필요, 관리하고자 하는 정보
3. 인스턴스의 집합
4. 반드시 속성을 가져야 함(2개 이상)
5. 업무 프로세스에 의해 이용됨
6. 다른 엔티티와 최소 1개 이상의 관계 성립

분류

유무형 분류

1. 유형 엔티티

물리적 형태 있음

안정적, 지속적 활용

사원, 물품 감사 등

1. 물리적 형태 없음

개념적 정보로부터 구분됨

조직, 보험상품 등

1. 사건 엔티티

업무 수행에 따라 발생하는 엔티티

통계자료에 이용

주문, 청구, 미납 등

발생 시점에 따른 분류

1. 기본 엔티티

업무에 원래 존재하는 정보

다른 관계에 의해 생성되지 않고 독립적으로 생성

다른 엔티티의 부모 역할

주식별자를 상속받지 않고, 고유한 주식별자를 가짐

1. 중심 엔티티

기본엔티티로부터 발생하는 업무에서 중심적인 역한

다른 엔티티와의 관계를 통해 많은 행위 엔티티 생성

계약, 사고, 청구 등

1. 행위 엔티티

2개 이상의 부모 엔티티로부터 발생

자주 내용이 바뀌거나, 데이터 양 증가

초기 단계보다 상세 설계, 상관 모델링을 진행하면서 도출됨

명명

* 현업 사용 용어
* 양자 사용 자제
* 단수 명사
* 유일
* 생성 의미대로

표기법

엔티티는 사각형으로 표기

속성은 표기법마다 다름

IE 표기법, Barker 표기법

IE는 맨 위에 PK가, Barker는 PK 속성에 #이 붙음



### 속성

업무에서 필요로 하는 고유한 성질, 특징

인스턴스로 관리하고자 하는 더 이상 분리되지 않는 최소의 데이터 단위

엔티티, 인스턴스, 속성, 속성값의 관계

엔티티는 2개 이상의 인스턴스의 집합

엔티티는 2개 이상의 속성을 가짐

속성은 1개의 속성값을 가짐

특징

반드시 업무에서 필요하고 관리하고자 하는 정보여야 함

정해진 주식별자에 함수적 종속성을 가져야 함(주 식별자에 따라 결정된다)

하나의 속성은 하나의 값을 가진다

여러 값이 있을 경우 별도의 엔티티를 이용해 분리

함수적 종속성

한 속성의 값이 다른 속성의 값에 종속적인 관계를 갖는 특징

어떤 속성값에 의해 다른 속성도 유일하게 결정된다면 B는 A에 의해 함수적으로 종속됨. A -> B

1. 완전 함수적 종속

특정 컬럼이 기본키에 대해 완전히 종속될 때

PK 구성 속성이 2개 이상일 경우 모두에 의한 종속 관계를 나타낼 때 완전 함수 종속성 만족함

1. 부분 함수적 종속

기본키 전체가 아니라 기본키의 일부에 대해 종속될 때

학생번호|과목|강사

학생번호, 과목이 PK일 때 강사는 학생 번호가 아닌 과목에 종속되므로 부분 함수적 종속

분류

속성 특성에 따른 분류

1. 기본 속성

업무로부터 추출된 모든 특성

1. 설계 속성

기본 속성 외에 업무를 규칙화하기 위해 만들거나, 기본 속성을 변형해 만들어짐

상품코드, 지점코드 등

1. 파생 속성

다른 속성에 의해 만들어지는 속성

일반적으로 계산된 값

엔티티 구성방식에 따른 분류

1. PK
2. FK
3. 일반 속성

분해 여부에 따른 분류

1. 단일 속성

하나의 의미로 구성

1. 복합 속성

여러개의 의미로 구성

주소(시, 구 동 등 분해 가능)

1. 다중값 속성

속성에 여러개의 값을 가질 수 있는 경우

엔티티로 분해

명명

업무에서 사용하는 이름

서술식 속성명은 X

약어 사용은 제한

도메인: 각 속성이 가질 수 있는 값의 범위. 데이터 타입, 크기, 제약사항 지정

### 관계

엔티티간의 연관성을 나타낸 개념

인스턴스간의 논리적인 연관성을 파악하여 정의

종류

1. 존재적 관계

엔티티가 다른 엔티티의 존재에 영향을 미치는 관계

부서가 삭제 시 사원의 존재에 영향을 미침

1. 행위적 관계

엔티티간의 어떤 행위. 고객에 의해 주문 발생

구성

1. 관계명
2. 차수(Cardinality)
3. 선택성(Optionality)

차수

인스턴스가 다른 엔티티의 인스턴스와 어떻게 연결되는지

1:1, 1:N, N:M 으로 표현

1:1

완전 관계: 관계되는 엔티티가 반드시 하나 존재하는 경우

선택적 관계: 관계되는 엔티티가 없거나 하나인 경우

1:N

엔티티에 하나의 행에 여러 엔티티 값이 있는 관계

N:M

다대다 연결 관계

조인 시 카테시안 곱이 발생해 두 엔티티를 연결하는 연결엔티티의 추가로 1:N으로 해소할 필요가 있음

UserRole 테이블 만드는 그런거

페어링

엔티티 안에 인스턴스가 개별적으로 관계를 가지는 것

관계는 페어링의 집합을 의미

관계, 차수, 페어링 정리

엔티티는 관계를 가짐

차수는 엔티티의 관계가 어떻게 연결 되는지, 1:1, 1:N, M:N

인스턴스의 시점으로 보면 다른 엔티티의 인스턴스와 특정한 페어링이 형성

차수는 엔티티-엔티티의 연결 방식을 나타내고, 페어링은 엔티티간의 특정 연결을 설명

### 식별자

속성 중 엔티티를 대표할 수 있는 속성을 나타냄

유일해야 함

식별자는 논리 모델링에서, 물리 모델링에서는 키

주식별자

1. 유일성

모든 인스턴스를 유일하게 구분

1. 최소성

유일성을 만족하는 최소한의 속성으로 구성

1. 불변성

지정되면 값은 변하지 않아야 함

1. 존재성

반드시 값이 존재해야 함(NULL 금지)

분류

대표성 여부

주식별자

보조식별자: 인스턴스를 구분할 수 있지만, 대표성을 가지지 못해 참조 관계 연결 불가능. 유일성, 최소성은 만족하지만 대표성을 만족하지 못함. 주민번호, UID 등

생성 여부

내부 식별자: 다른 엔티티 참조 없이 스스로 생성되는 식별자

외부 식별자: 다른 엔티티와 관계로 인해 만들어지는 식별자

속성 수

단일

복합

대체 여부

본질 식별자: 업무 프로세스에서 만들어지는 식별자

인조 식별자: 인위적으로 생성. 자동 증가하는 Id 등. 업무적으로 필요하진 않음.

중복 데이터 발생 가능성, 불필요한 인덱스 생성할 수 있음

주 식별자 도출 기준

1. 업무에서 자주 이용되는 속성

같은 조건을 만족해도 업무적으로 더 많이 사용되는 속성을 지정

1. 명칭, 내역 등 이름은 피해야 함
2. 속성의 수를 최대한 적게 구성

조인으로 인한 성능 저하 방지

일반적으로 7개 이상의 주식별자는 인조 식별자를 생성해 단순화 하는게 좋음

관계간 구분

강한 개체: 독립적으로 존재할 수 있는 엔티티(고객->계좌에서 고객)

약한 개체: 독립적으로 존재할 수 없는 엔티티(고객->계좌에서 계좌)

식별 관계: 엔티티의 기본키를 다른 엔티티가 기본키 중 하나로 공유하는 관계

UserRole 엔티티가 UserId, RoleId를 주식별자로 가지고 있을때

비식별 관계: 강한 개체의 기본키를 다른 엔티티의 일반 속성으로 가지는 것

Blog.UserId

### 정규화

최소한의 데이터를 엔티티에 넣는 식으로 데이터를 분해하는 과정

일관성, 최소한의 중복, 최대한의 유연성을 위한 과정

모델의 독립성 확보. 이상 현상을 줄이기 위함

상세화 하는 과정으로 논리 데이터 모델링 수행 시점에 고려됨

1~5까지 존재, 보통 3까지 수행

이상현상:

정규화를 하지 않아 발생하는 현상(삽입이상, 갱신이상, 삭제이상)

단계

1. 제1정규화(1NF)

컬럼이 원자성을 갖도록 분해하는 단계

하나의 행과 컬럼의 값이 반드시 한개의 값만 입력되도록 행을 분리

1. 제2정규화

1정규화 진행 후 완전 함수 종속을 만들도록 테이블을 분해

기본키를 구성하는 모든 컬럼 값이 다른 컬럼을 결정지어야 함

기본키의 부분집합이 다른 컬럼과 1:1 대응 관계를 갖지 않아야 함

PK가 2개 이상일때 발생, 일부와 종속되면 분리

1. 제3정규화

2정규화 진행 후 이행적 종속을 없애도록 분리

이행적 종속이란 A -> B, B -> C의 관계 성립 시 A -> C가 성립되는 것을 의미

이걸 (A,B), (B,C)로 테이블을 분리하는 것이 제3정규화

BNCF, 4, 5는 거의 이론적으로 발생함

반정규화

성능 향상을 위해 중복 허용, 조인을 줄이는 DB 성능 향상 방법

성능 향상, 개발, 운영 단순화를 위해 정규화된 모델을 중복, 통합, 분리하는 모델링 기법

### 관계, 조인

엔티티->인스턴스의 논리적 연관성

엔티티의 정의, 속성 정의 및 관계 정의에 따라 변할 수 있음

관계를 맺는다는 것은 부모의 식별자를 자식에 상속, 상속된 속성을 매핑키로 활용

조인은 분리되면서 테이블이 관계를 갖고, 두 테이블의 데이터를 동시에 출력하거나 참조하기 위해 연결해야 하는 과정

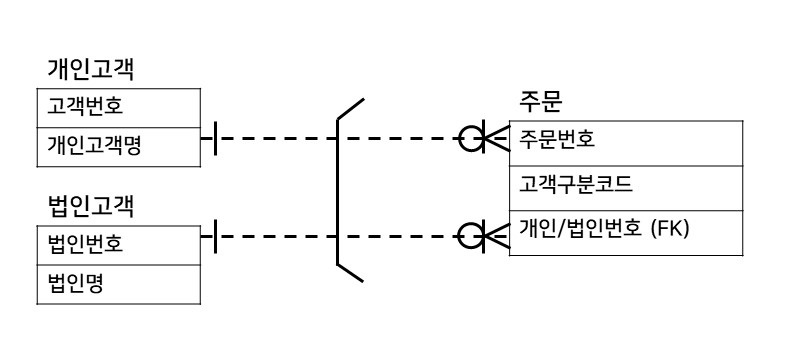
계층형 데이터 모델: 자기 자신끼리 관계가 발생, 즉 하나의 엔티티 내의 인스턴스끼리 계층 구조를 가지는 경우. 셀프 조인이라고 하며 같은 테이블을 여러번 조인하는 것

상호 배타적 관계:

두 테이블 중 하나만 가능한 관계

한 테이블이 A, B 테이블 중 하나만 상속될 수 있는 경우를 상호배타적 관계라고 함

표기법:



### 트랜젝션

하나의 연속적인 업무 단위

트랜잭션에 대한 관계는 필수적인 관계형태를 가짐

여러 DML이 포함될 수 있음

서로 독립적으로 발생할 수 없음

부분 커밋 X

필수적, 선택적 관계의 ERD

관계가 서로 필수적일 때 하나의 트랜잭션 형성

서로 독립적 수행 가능하다면 선택적 관계로 표현

IE 표기법

선택적 관계는 끝에 원으로 표현

필수적 관계는 원 없음

Barker 표기법

필수적 관계는 실선

선택적 관계는 점선

### 

### NULL

DBMS에서 아직 정하지 않은 값

NULL을 포함한 연산 결과는 항상 NULL

산술 연산(+-\*/ 등)시 주의

오라클에서는 NVL(column, value)로 null일 시 value 반환

집계함수는 NULL 제외함

count, sum, avg, min, max 등

COUNT(nullable\_column), COUNT(\*)의 값이 다를 수 있음

AVG의 경우 null이 아닌 값의 합/null이 아닌 갯수를 반환. null까지 포함하기를 원할 시 적절히 바꿔야 함

IE 표기법에서는 nullable을 알 수 없음

Barker 표기법에는 속성 앞에 동그라미를 써 nullable을 의미