



SQL 개발자 D 이론

모델링

- 데이터베이스의 모델링은 현실세계를 단순화하여 표현하는 기법이다.

모델링 특징

추상화

- 현실 세계를 일정한 형식으로 표현하는 것이다. 즉 아이디어나 개념을 간략하게 표현하는 과정

단순화

- 복잡한 현실 세계를 정해진 표기법으로 단순하고 쉽게 표현한다는 의미

명확화

- 불분명함을 제거. 명확하게 해석할 수 있도록 기술



모델링 이해: 현실세계를 추상화, 단순화, 명확화하기 위해 일정한 표기법에 의해 표현하는 기법

모델링 세가지 관점

데이터 관점

- 데이터 위주의 모델링. 어떤 데이터들이 업무와 얽혀있는지, 데이터간에 어떤 관계가 있는지에 대해서 모델링

프로세스 관점

- 프로세스 위주의 모델링. 업무가 실제로 처리하고 있는일이 무엇인지, 앞으로 처리해야하는 일은 무엇인지 모델링

데이터와 프로세스의 상관 관점

- 데이터와 프로세스의 관계 위주 모델링. 프로세스의 흐름에 따라 데이터가 어떤 영향을 받는지 모델링



데이터 품질 보장을위해 데이터 모델링시 유의사항

중복 : 같은 데이터가 여러 엔티티(테이블)에 중복으로 저장되는현상 지양해야함

비유연성 : 사소한 변경에도 데이터 모델이 수리로 변경해야하는 상황이 올 수 있음

유지보수에 어려움 증가. 데이터모델과 프로세스 분리 → 유연성 높임

비일관성 : 너무 중복이 없는 경우에도 비일관성 발생

다른 데이터와 연관 고려

데이터 간의 연관 관계 명확하게 정의할것.

모델링의 세가지 단계

개념적 데이터 모델링

- 전체적 데이터 모델링 수행시 행해짐 레벨이 가장 높은 모델링
업무중심적, 포괄적인 수준 모델링 진행

논리적 데이터 모델링

- 재사용이 가장 높은 모델링. Key, 속성, 관계 등을 모두 표현하는 단계

물리적 데이터 모델링

- 실제 데이터베이스로 구현할 수 있도록 성능, 가용성 등 물리적 성격 고려하여 모델을 표현하는 단계

데이터의 독립성

3단계 스키마 구조

- 외부 스키마
 - 사용자의 관점 : 각 사용자가 보는 데이터베이스의 스키마를 정의
- 개념 스키마
 - 통합된 관점 : 데이터베이스에 저장되는 데이터들을 표현하고 데이터들 간의 관계를 나타낸다.
- 내부 스키마
 - 물리적인 관점 : 실직적인 데이터의 저장구조나 컬럼 정의, 인덱스 등이 포함된다.



3단계 스키마구조로 나누는 이유

데이터베이스에 대한 사용자들의 관점과 실제로 표현되는 물리적인 방식을 분리 독립적으로 보장하기 위한 것

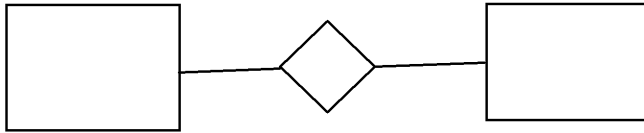
논리적 독립성 : 개념스키마가 변경되어도 외부스키마는 영향을 받지 않음

물리적 독립성 : 내부스키마가 변경되어도 외부/개념 스키마는 영향을 받지 않음

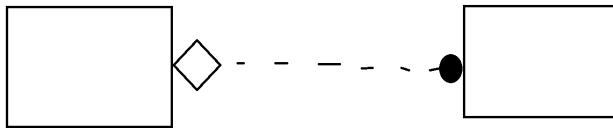
ERD

ERD 표기방식

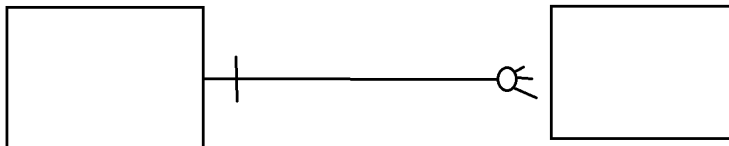
Peter chen: 실무에서 사용하는 경우는 드뭄



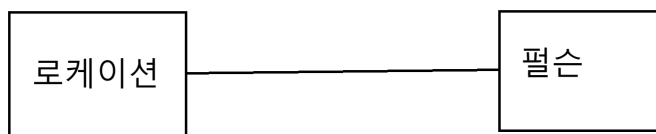
IDEF1X: 실무에서 사용하는 경우도 있음 ERWin에 사용되는 모델



IE/Crows'Foot: 까마귀발 표기법 가장 많이 사용 ERWin, ERStudio에서 사용되는 모델



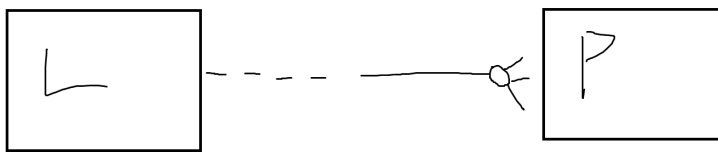
Min-Max/ISO : 각 엔터티의 참여도를 좀더 상세하게 나타내는 표현법



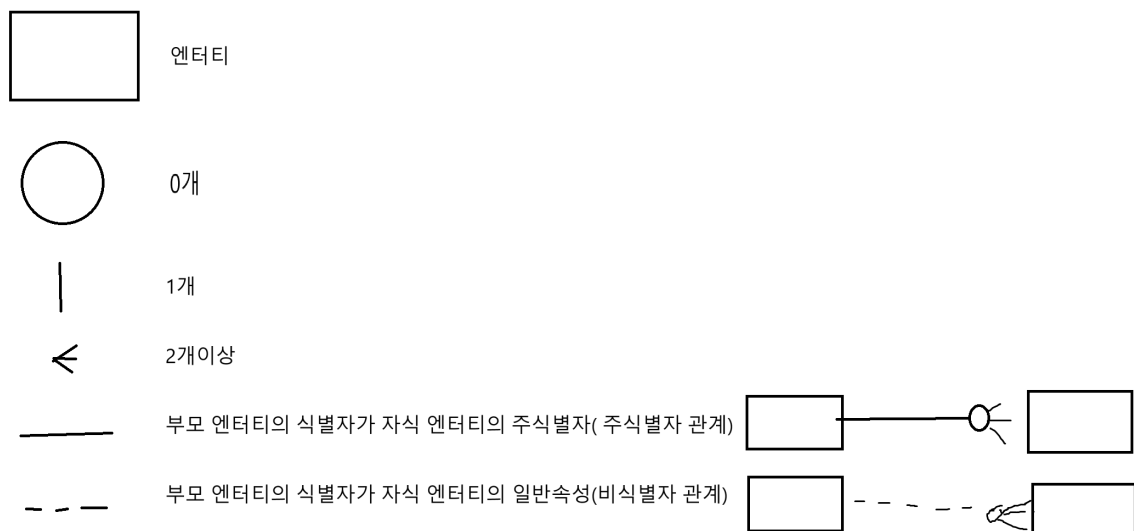
UML : 소프트웨어 공학에서 주로 사용되는 모델



Case Method/ Barker : Oracle에서 사용되는 모델 CrowsFoo과 비슷



IE/Crows'Foot



ERD 작성 순서

1. 엔터티를 도출하고 그린다
2. 엔터티를 적절하게 배치한다
3. 엔터티 간의 관계를 설정한다

4. 관계명을 기입한다
5. 관계의 참여도를 기입한다
6. 관계의 필수/선택 여부를 기입한다.

엔터티

- 엔터티 : 식별이 가능한 객체 ex) 주문(주문번호, 회원번호, 주문일자) 주문상품(주문번호FK), 상품번호, 주문수량)



엔터티 : 테이블 인스턴스: ROW(행) 속성: Column(열)

엔터티 특징

- 업무에 쓰이는 정보여야함
- 유니크함을 보장할 수 있는 식별자가 있어야함 식별이 모호하면 안됨.
- 2개 이상의 인스턴스(행)을 가지고 있어야함
- 속성을 가져야함 상세하게 나타낼 수 있는 속성 필요
- 다른 엔터티와 1개이상의 관계를 가지고 있어야함 연관성이 있어야함

엔터티 분류

유형 vs 무형

유형 엔터티	물리적인 형태 존재 안정적 지속적 ex) 상품 회원 등
개념 엔터티	물리적인 형태 없음 개념적 ex) 부서, 학과
사건 엔터티	행위를 함으로 써 발생, 빈번함 통계 자료로 이용가능 ex) 주문, 이벤트응 모

발생시점

기본 엔터티	업무에 원래 존재하는 정보 독립적으로 생성되며, 자식 엔터 티를 가질 수 있음 ex 상품,사원,부서등
중심 엔터티	기본 엔터티로부터 파생되고 행 위 엔터티 생성 업무에 있어서 중심적인 역할을 하며 데이터의 양이 많이 발생 ex 주문,매출계약
행위 엔터티	2개 이상의 엔터티로부터 파생 데이터가 자주 변경되거나 증가 할 수있음 ex 주문내역 이벤트응모이력등



엔터티 이름 정할때 주의할 점

- 업무에서 실제로 쓰이는 용어 사용
- 한글은 약어 사용 금지 영문은 대문자로 표기
- 단수 명사로 표현 띄어쓰기 안함
- 다른엔터티와 의미상중복X
- 해당엔터티가 갖고있는 데이터가 무엇인지 명확하게 표현

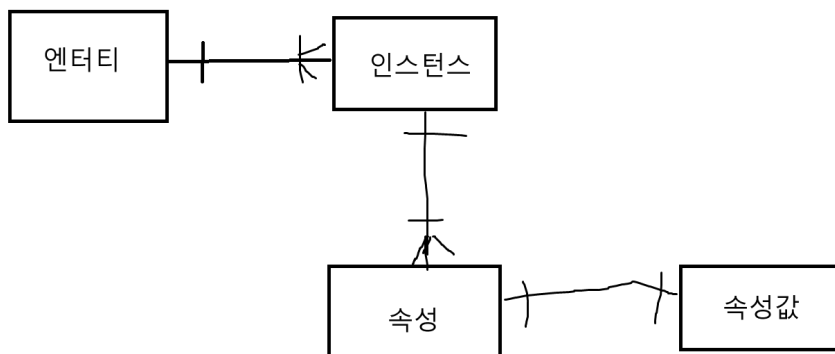
속성 (Attribute 열)

- 엔터티의 특징을 나타내는 최소의 데이터 단위

속성값

- 인스턴스를 구체적으로 나타내주는 데이터 (여러개 속성값을 갖는경우 별도의 엔터티로 분리)

엔터티(테이블),인스턴스(행),속성(열),속성값의 관계



- 한개의 엔터티 = 두개이상의 인스턴스
- 한개의 인스턴스 = 두개 이상의 속성
- 한 개의 속성= 하나의 속성값

분류

- 특성에 따른 분류

- 기본속성
 - 업무 프로세스 분석 바로 정의 가능한 속성
- 설계속성
 - 설계하다보니 필요하다고 판단되어 도출해낸 속성 유니크함부여
- 파생속성
 - 특정한 규칙으로 변형하여 생성한 속성 계산값 가공된 값
- 구성방식에 따른 분류
 - PK 속성
 - 엔터티의 인스턴스들을 식별할 수 있는 속성
 - FK 속성
 - 다른 엔터티의 속성에서 가져온 속성
 - 일반속성
 - PK,FK를 제외한 속성

도메인

- 속성이 가질 수 있는 속성값의 범위



용어 사전 : 어떤 시스템이든 속성명은 업무와 직결되는 항목. 속성이름 정확, 직관적으로 부여 용어 혼란 없앰

시스템 카탈로그 : 시스템 자체 데이터 관련 데이터베이스 조회만 가능

관계

- 엔터티와 엔터티와의 관계를 의미. 연관성 타입분류 존재관계 행위관계
- 존재관계
 - 존재자체로 연관성 학생,학과
- 행위관계
 - 특정 행위를 하으로써 연관성이 생기는 관계 학생, 출석부 출석

표기법

- 관계명
 - 관계의 이름. 어떠한 관계인지 각엔터티 관점 관계명 하나씩 명확한 문장
- 관계차수
 - 관계에 참여하는 수 1:1 1:M M:N
- 관계선택사양
 - 필수인지 선택인지의 여부 필수적관계(반드시 존재해야하는 관계), 선택적 관계(없을 수도 있는 관계)

식별자

- 인스턴스를 구분가능하게 만들어주는 대표격인 속성
- 주식별자
 - 기본키 PK 해당하는 속성
 - 유일성
 - 유니크함 부여 식별
 - 최소성
 - 유일성 보장 최소 개수의 속성
 - 불변성
 - 속성값 변경 X
 - 존재성
 - 속성값 NULL X

분류

- 대표성여부
 - 주식별자 : 다른 엔터티와 참조관계로 연결
 - 보조식별자 : 다른 엔터티와 참조 관계로 연결되지 않음
- 스스로 생성되었는지 여부
 - 내부 식별자
 - 엔터티 내부에서 스스로 생성된 식별자

- 외부식별자
 - 다른 엔터티에서 온 식별자. 다른 엔터티와의 연결고리 역할
- 단일 속성의 여부
 - 단일식별자
 - 하나의 속성구성된 식별자
 - 복합 식별자
 - 두 개 이상의 속성으로구성된 식별자
- 대체 여부
 - 원조 식별자
 - 업무프로세스에 존재하는 식별자 가공되지않은 식별자(본질식별자)
 - 대리 식별자
 - 주식별자의 속성이 두 개 이상인 경우 하나로 묶어서 사용 (인조식별자)

식별자 관계 vs 비식별자 관계

- 식별자 관계
 - 부모 엔터티의 식별자가 자식 엔터티의 주식별자가 되는 관계
- 비식별자 관계
 - 부모 엔터티의 식별자가 자식 엔터티의 주식별자가 아닌 일반속성이 되는 관계



챕터 1 끝

정규화

- 데이터 정합성(데이터 정확성, 일관성 유지보장)을 위해 엔터티를 작은 단위로 분리하는 과정이다.

제1 정규형

- 모든 속성은 반드시 하나의 값만 가져야 한다.

제 2정규형

- 모든 일반속성은 반드시 모든 주식별자에 종속되어야 한다.

제 3정규형

- 주식별자가 아닌 모든 속성 간에는 서로 종속될 수 없다.



주의사항 !!! 지나친 정규화는 오히려 성능을 저하 시킬 수 있다.

반 정규화

- 데이터의 조회 성능을 향상시키기 위해 데이터의 중복을 허용하거나 데이터를 그룹핑하는 과정이다.



입력, 수정, 삭제 성능은 저하 될 수 있음. 정규화가 끝난 후 실행

테이블 반 정규화

테이블 병합	1:1 관계 테이블 병합 1:M 관계 테이블 병합 슈퍼 서브타입 테이블 병합
테이블 분할	테이블 수직 분할(속성) 테이블 수평 분할(인스턴스 분할, 파티셔닝)
테이블 추가	중복 테이블 추가 통계 테이블 추가 이력 테이블 추가 부분 테이블 추가

- 테이블 병합

- Join이 필요한 경우가 많아 테이블을 통합하는것이 성능 측면에서 유리할 때 고민한다.
- 테이블 수직 분할
 - 엔터티 일부 속성을 별도의 엔터티로 분할(1:1관계성립)한다.
- 테이블 수평 분할
 - 엔터티의 인스턴스를 특정기준으로 별도의 엔터티로 분할 한다.
- 테이블 추가
 - 중복 테이블 추가 : 성능상 필요시 엔터티 추가
 - 통계 테이블 추가
 - 이력 테이블 추가
 - 부분 테이블 추가

컬럼 반정규화

- 중복 컬럼 추가
 - 업무 프로세스상 Join이 필요한 경우
- 파생 컬럼 추가
 - 프로세스 수행시 부하가 염려되는 계산값을 미리 컬럼으로 추가
- 이력 테이블 컬럼 추가
 - 조회 기준이 될 것으로 판단되는 컬럼을 미리 추가

관계 반정규화

- 업무 프로세스상 Join이 필요한 경우가 많아 중복관계를 추가하는것 성능측면 유리

트랜잭션(Transaction)

- 데이터를 조작하기 위한 하나의 논리적인 작업 단위이다.

NULL

- NULL은 존재하지 않음. 즉 값이 없음을 의미한다.

