

ERD와 정규화 과정

ERD (Entity Relationship Diagram)

데이터베이스 구축 시 가장 기초적인 뼈대 역할

릴레이션 간의 관계들을 정의한 것

서비스 구축 시 가장 먼저 신경 써야 할 부분

ERD의 중요성

ERD는 시스템의 요구 사항을 기반으로 작성되며, 이를 기반으로 DB를 구축함

DB 구축 후에도 디버깅이나 비즈니스 프로세스 재설계가 필요한 경우에 설계도 역할을 담당

관계형 구조로 표현할 수 있는 데이터를 구성하는데 유용할 수 있으나, 비정형 데이터를 충분히 표현할 수 없다는 단점이 있음

- **비정형 데이터** : 비구조화 데이터. 미리 정의된 데이터 모델이 없거나 미리 정의된 방식으로 정리되지 않은 정보

ERD 예제

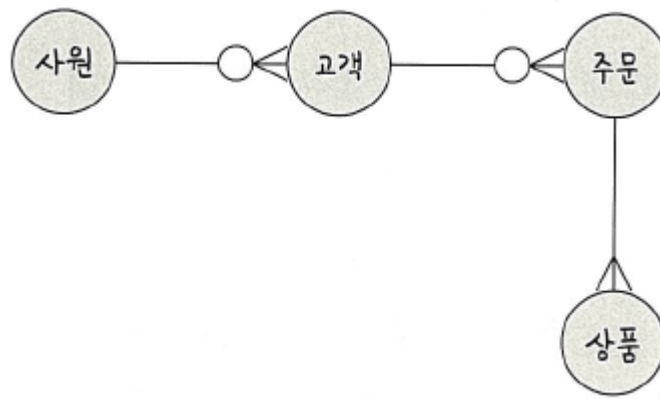
승원 영업부서의 ERD

요구사항

영업사원은 0~n명의 고객을 관리

고객은 0~n개의 주문 가능

주문에는 1~n개의 상품이 들어감



무무오브레전드의 ERD

| 요구사항

선수들은 1명의 챔피언을 고를 수 있음

챔피언은 한개 이상의 스킬을 가짐

스킬은 한개 이상의 특성을 가짐



정규화 과정

릴레이션 간의 잘못된 종속 관계로 인해 데이터베이스 이상현상이 발생해 이를 해결하거나, 저장공간을 효율적으로 사용하기 위해 릴레이션을 여러 개로 분리하는 과정

정규형 원칙을 기반으로 정규형을 만들어가는 과정

정규화된 정도는 정규형(NF, Normal Form)으로 표현함

데이터베이스 이상현상

아래와 같은 경우를 데이터베이스 이상현상이라고 함

회원이 한 개의 등급을 가져야하는데, 세개의 등급을 갖는 경우

삭제 시 필요한 데이터가 같이 삭제되는 경우

데이터 삽입 시 하나의 필드 값이 NULL 이면 안되어서 삽입하기 어려운 경우

정규형 원칙

같은 의미를 표현하는 릴레이션이지만 좀 더 좋은 구조로 만들어야함

자료의 중복성은 감소해야함

독립적인 관계는 별개의 릴레이션으로 표현해야함

각각의 릴레이션은 독립적인 표현이 가능해야함

제 1정규형

릴레이션의 모든 도메인이 더 이상 분해될 수 없는 원자값만으로 구성되어야함

릴레이션의 속성 값 중 한 개의 기본키에 대해 두 개 이상의 값을 가지는 반복 집합이 있어서는 안됨

반복 집합이 있는 경우, 제거해야함

유저번호	유저ID	수강명	성취도
1	홍철	{C++코딩테스트, 프런트특강}	{90%, 10%}
2	범석	{코드포스특강, DS특강}	{7%, 8%}



유저번호	유저ID	수강명	성취도
1	홍철	C++코딩테스트	90%
1	홍철	프런트특강	10%
2	범석	코드포스특강	7%
2	범석	DS특강	8%

제2정규형

릴레이션이 제 1정규형이면서 함수의 종속성을 제거한 형태

부분 함수의 종속성 제거: 기본키가 아닌 모든 속성이 기본키에 완전 함수 종속적인 것

유저번호	유저ID	수강명	성취도
1	홍철	C++코딩테스트	90%
1	홍철	프런트특강	10%
2	범석	코드포스트특강	7%
2	범석	DS특강	8%



유저번호	유저ID
1	홍철
2	범석

유저ID	수강명	성취도
홍철	C++코딩테스트	90%
홍철	프런트특강	10%
범석	코드포스트특강	7%
범석	DS특강	8%

제3정규형

제2정규형이면서 기본키가 아닌 모든 속성이 이행적 함수 종속(transitive FD)을 만족하지 않는 상태

| 이행적 함수 종속

$A \rightarrow B$ 와 $B \rightarrow C$ 가 존재하면 $A \rightarrow C$ 가 논리적으로 성립하는데, 이 때 C 가 A 에 이행적으로 함수 종속이 되었다고 함

유저ID	등급	할인율
홍철	플래티넘	30%
범수	다이아	50%
가영	마스터	70%



유저ID	등급
홍철	플래티넘
범수	다이아
가영	마스터

등급	할인율
플래티넘	30%
다이아	50%
마스터	70%

보이스/코드 정규형

제3정규형이면서 결정자가 후보키가 아닌 함수 종속 관계를 제거하여 릴레이션의 함수 종속 관계에서 모든 결정자가 후보키인 상태

결정자: 함수 종속 관계에서 특정 종속자를 결정짓는 요소. $X \rightarrow Y$ 일 때 X는 결정자, Y는 종속자

요구사항이 아래와 같은 경우

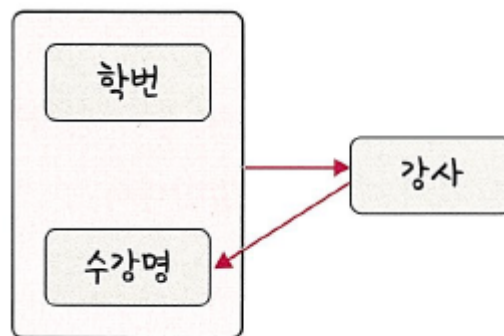
- 각 수강명에 대해 한 학생은 오직 한 강사의 강의만 수강함
- 각 강사는 한 수강명만 담당함
- 한 수강명은 여러 강사가 담당할 수 있음

학번	수강명	강사
12010	코딩테스트	큰돌
12010	MEVN	재엽
12011	코딩테스트	큰돌
12011	MEVN	가영
NULL	롤	범석

↑
삽입 이상

위 릴레이션의 경우, {학번, 수강명} 또는 {학번, 강사}가 후보키가 되며, 만약 범석이라는 강사가 '롤'이라는 수강명을 담당한다고 했을 경우 이를 삽입하면 학번이 NULL 이 되는 문제점 발생

아래와 같은 함수 종속 다이어그램을 가짐



즉, 강사 속성이 결정자이나, 후보키가 아니므로 이 강사 속성을 분리해야함

학번	수강명	강사
12010	코딩테스트	큰돌
12010	MEVN	재엽
12011	코딩테스트	큰돌
12011	MEVN	가영
NULL	롤	범석

↑ 삽입 이상



학번	강사
12010	큰돌
12010	재엽
12011	큰돌
12011	가영

수강명	강사
코딩테스트	큰돌
MEVN	재엽
MEVN	가영
롤	범석

테이블이 잘 분해되었으며, NULL 값이 없음

정규화 과정을 거쳐 테이블을 분리한다고 해서 성능이 무조건적으로 좋아지는 것은 아님

⇒ 서비스 요구사항에 따라 정규화 또는 비정규화 과정을 진행해야함