정규화

관계형 데이터베이스 설계란?

데이터 중복의 문제점:

불일치 – 갱신이상(Update anomaly)

1. 수정 이상

반복된 데이터 중에서 일부만 수정하는 현상 (신뢰도가 떨어진다.)

1. 삽입 이상

필요한 정보를 저장하기 위해서 불필요한 정보를 반드시 함께 저장해야 하는 현상

(넣고 싶은데 무언가가 없어서 넣지 못하는 상황)

1. 삭제 이상

내가 어떤 것을 지우려고 할 때 원하지 않는 값도 지우게 되는 현상

그루핑이 잘못된 예

표에서 동일한 값이 여러 번 나온다고 해서 중복이 아니다.

연관성을 살펴보고 중복을 따져봐야 한다.

수정 이상 => 이홍근이 이메일 주소를 바꿨다. 그럼 두 군데를 수정해야한다. 그렇지 않으면 제대로된 이메일 정보가 포함되지 않을 수 있다.

삽입 이상 => 신상정보를 수정하고 싶었는데 수강정보가 없어서 삽입을 할 수 없다거나

삭제 이상 => 신상정보 or 수강정보를 삭제하고 싶었는데 다른 정보들도 같이 삭제 된다거나

데이터 중복의 문제점 : NULL

그룹화를 통한 널값 피하기

* 근데 그룹화를 하면 차지하는 용량이 더 많을텐데? 하지만 그것으로 인한 단점보다 얻는 이점이 더 많다!
* 막 잘라서 그룹화 시켜도 괜찮은 걸까?
* 릴레이션의 분해

: 하나의 릴레이션을 두개 이상으로 나누는 것

무손실 분해 (lossless decomposition)

조인하면 원래의 릴레이션에 들어 있는 정보를 완전하게 얻을 수 있음

정보의 손실은 원래의 릴레이션을 분해한 후에 생성된 릴레이션들을 조인한 결과에 들어 있는 정보가 원래의 릴레이션에 들어 있는 정보보다 적거나 많은 것을…

손실 분해를 가짜 튜플들이 생길 수 있다.

그러면 무손실 분해를 하려면 어떻게 잘라야 할까??

* 기본키와 외래키 관계 (참조 관계)를 있을 수 있도록 조인 애트리뷰트를 설계해서 넣어야 한다!

외부조인 => 매칭이 되는 부분이 없을 때 그냥 NULL을 채워 넣어 합칠 수 있다.

좋은 관계형 데이터 설계란?

잘못 그룹화된 릴레이션 발견 => 무손실 분해 => 잘 그룹화된 릴레이션으로 변환 => 반복

잘 그룹화된 릴레이션은

* 이해하기 쉽다.
* 중복이 없어서 갱신이상이 최소화.
* 널 값이 별로 없다.
* 가짜 튜플이 생기지 않는다.

함수적 종속성 : 좋은 릴레이션 설계의 기준