**제 7장 릴레이션 정규화**

원래의 릴레이션을 분해함으로써 중복과 세 가지 갱신이상 최소화

**7.1 정규화 개요**



**갱신 이상**: 수정 이상, 삽입 이상, 삭제 이상

1. **수정 이상**: 중복된 것에 대해 일부분 수정 시 불일치 발생

부서 이름이 중복! -> 부서 이름 변경 시 수정이상 발생가능

\* 수정 이상을 해결하기 위해서는 관계 테이블을 만드는 것이 중요!

1. **삽입 이상**: 불필요한 정보 포함 저장 필수

박영권의 부서번호2, 부서이름2 null 저장 필수

1. **삭제 이상**: 필요한 정보도 같이 삭제됨

김창섭을 삭제할 때 영업부의 정보도 삭제됨 (부서 테이블이 따로 없을 경우를 가정)

**7.2 함수적 종속성**

결정자: 어떤 애트리뷰트 값을 고유하게 결정할 수 있는 애트리뷰트

A → B: A가 B를 결정한다. (A는 B의 결정자이다.)

= B가 A에 함수적으로 종속한다.

= 함수적으로 종속하는 필요 충분 조건은 각 A값에 대해 반드시 한 개의 B값 대응

**완전 함수적 종속성**

복합 애트리뷰트 A에 대하여 함수적으로 종속하는 경우.

A의 어떤 진부분 집합에도 종속하지 않는 것이 중요.

**부분 함수적 종속성**

키의 일부에만 종속적인 관계를 가짐. 즉, 결정자가 아닌 키가 존재함.

**이행적 함수적 종속성**

삼단 논법. A → C의 필요 충분 조건은 A → B ∩ B → C.

**7.3 릴레이션 분해**

하나의 릴레이션을 두 개 이상의 릴레이션으로 나누는 것.

장점: 중복 감소, 갱신이상이 줄어듬

단점: 조인을 필요하게 된다.

무손실 분해: 조인을 했을 때 원래 릴레이션으로 돌아갈 수 있음.

불필요한 분해: 중복되지않은 정보에 대하여 분해

나쁜 분해: 원래 릴레이션이 복원되지 않는 분해

\* **가짜 투플**: 분해했다가 다시 조인할 때 나타나는 투플. 가짜 투플이 생기면 나쁜 분해.

**7.4 정규형**

**제1정규형**: 모든 애트리뷰트가 원자 값만을 갖는다.

= 반복 그룹이 나타나지 않는다.

변환 방법: 반복 그룹 애트리뷰트에 나타나는 집합에 속한 값마다 하나의 투플로 표현

제2정규형: 제1정규형을 만족하고, 후보 키가 아닌 모든 애트리뷰트들이 기본 키에 완전하게 종속.

즉, 기본 키가 두 개 이상인 경우만 고려

변환 방법: 릴레이션 분해

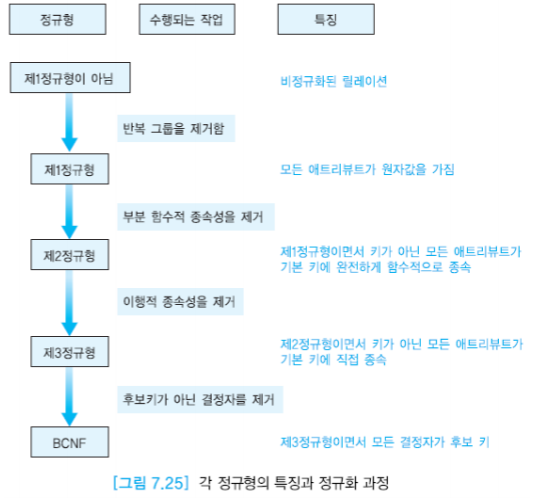
제3정규형: 제2정규형을 만족하고, 키가 아닌 모든 애트리뷰트들이 기본 키에 이행적으로 종속하지 않는 것.

변환 방법: 릴레이션 분해

BCNF: 제3정규형을 만족하고, 모든 결정자가 후보 키이어야 함.

변환 방법: 키가 아니면서 결정자인 애트리뷰트와 그 종속 관계인 애트리뷰트를 따로 분해

그리고 기존 릴레이션에 결정자를 남겨서 구성요소(외래키)로 만듦.



정규화 장점: 중복과 갱신이상 감소, 무결성 제약조건을 시행하기 위해 필요한 코드의 양 감소

정규화 단점: 릴레이션 개수 증가, 조인의 필요성 증가

역정규화: 낮은 정규형으로 돌아감.