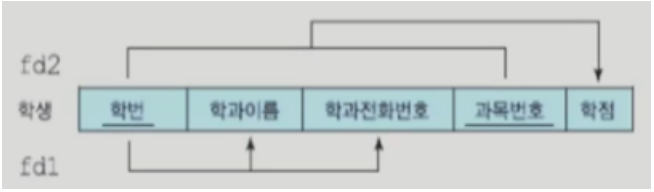
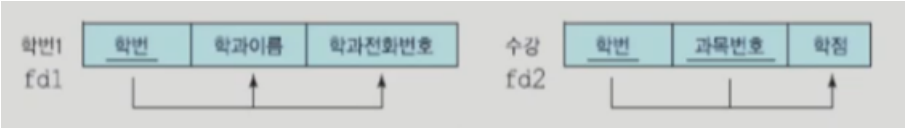
3
뷰와 시스템 카탈로그

제 1정규형으로 인한 갱신 이상



→ 기본키에 대한 부분 함수적 종속성이 학생 릴레이션에 존재



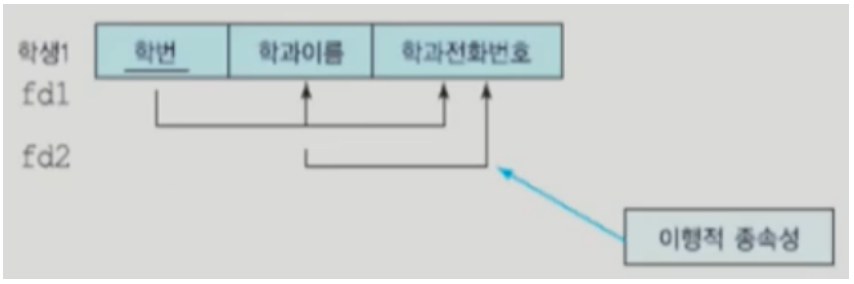
⇒ 제 2정규형을 만족

제 2정규형

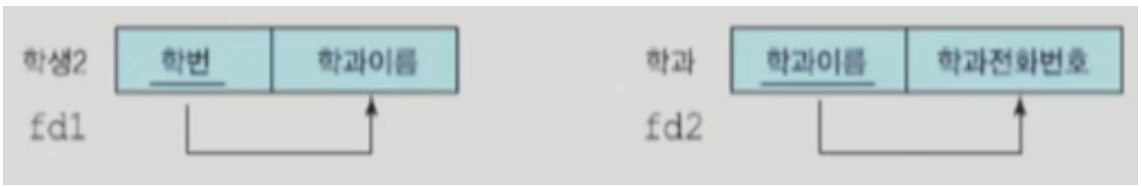
: 제 1정규형을 만족하면서 어떤 후보키에도 속하지 않는 모든 어트리뷰트들이 기본키에 완전하게 함수적으로 종속하는 것

→ 동일하게 수정, 삽입, 삭제 이상이 생김

갱신이상



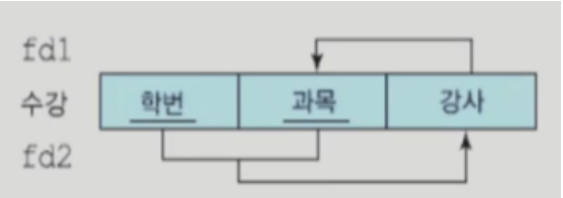
→ 학생1 릴레이션에 이행적 종속성이 존재



⇒ 제 3정규형을 만족

제3정규형

: 제2정규형을 만족하면서 키가 아닌 모든 애트리뷰트가 릴레이션의 기본키에 이행적으로 종속하지 않는 것



→ 동일하게 수정 이상, 삽입 이상, 갱신 이상이 생김

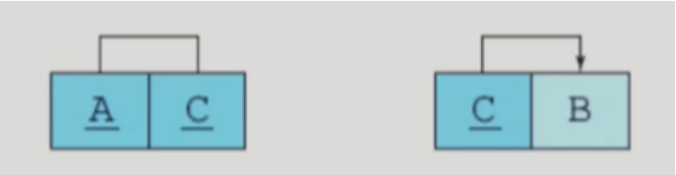
❓ 갱신 이상이 생기는 이유

수강 릴레이션에서 키가 아닌 애트리뷰트가 다른 애트리뷰트를 결정하기 때문

BCNF

: 제3정규형을 만족하고 모든 결정자가 후보키이어야 함

- 제3정규형을 만족하는 대부분의 릴레이션들은 BCNF도 만족함
- 제3정규형을 만족하는 릴레이션을 BCNF로 정규화하려면 키가 아니면서 결정자 역할을 하는 애트리뷰트와 그 결정자에 함수적으로 종속하는 애트리뷰트를 하나의 테이블에 넣음



제1정규형이 아님

반복 그룹 제거

제1정규형 → 모든 애트리뷰트가 원자값을 가짐

부분 함수적 종속성을 제거

제2정규형 → 키가 아닌 모든 애트리뷰트가 기본 키에 완전하게 함수적으로 종속

이행적 종속성 제거

제3정규형 → 키가 아닌 모든 애트리뷰트가 기본 키에 직접 종속

후보키가 아닌 결정자를 제거

BCNF → 모든 결정자가 후보키

역정규화

- 정규화 단계가 진행될수록 중복이 감소하고 갱신 이상도 감소됨
- 성능상의 관점에서만 보면 높은 정규형을 만족하는 스키마가 최적인 것이 아님
- 분해되기 전의 릴레이션을 대상으로 질의를 할 때는 조인이 필요없지만 분해된 릴레이션을 대상으로 질의를 할 때는 같은 정보를 얻기 위해서 보다 많은 릴레이션을 접근해야 하기 때문에 조인의 필요성 증가

= 보다 낮은 정규형으로 되돌아가는 것

뷰

: 다른 릴레이션으로부터 유도된 릴레이션

→ 복잡한 질의를 간단하게 표현하는 수단, 보안 매커니즘 (볼 수 있는 정보만 봄)

시스템 카탈로그

: 시스템 내의 객체(릴레이션, 뷰, 인덱스, 사용자 등)에 대한 정보를 포함

→ 적절히 활용하면 원하는 릴레이션을 데이터베이스에서 찾고, 그 릴레이션에 어떤 애트리뷰트들이 들어 있고, 각 애트리뷰트의 데이터 타입이 무엇인지를 쉽게 파악 가능

뷰

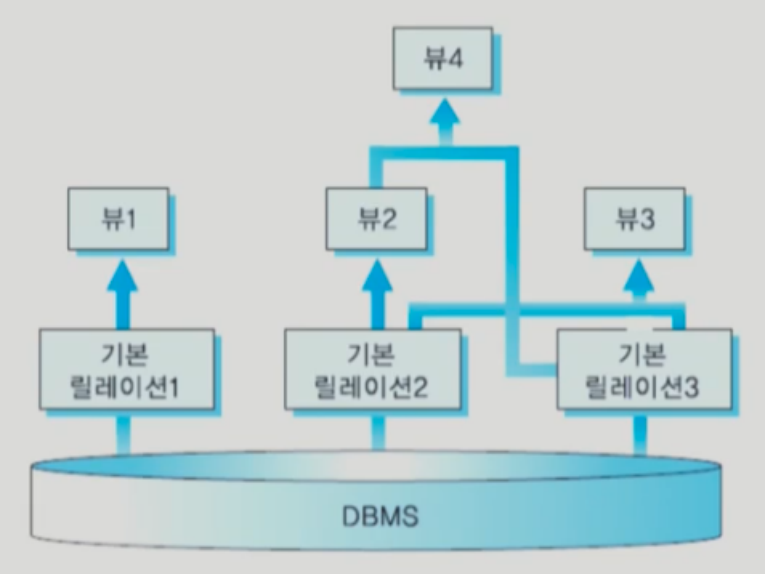
→ 기존의 기본 릴레이션에 대한 select문의 형태로 질의됨

→ 릴레이션으로부터 데이터를 검색하거나 갱신할 수 있는 동적인 창의 역할

스냅샷

어느 시점에 select문의 결과를 기본 릴레이션의 형태로 저장해 놓은 것

스냅샷을 정의하는 시점의 기본 릴레이션의 내용이 스냅샷에 반영됨



→ 뷰와 기본 릴레이션의 관계

```
create view 뷰 이름 [(애트리뷰트(들))]
as select문
[with check option]
```

뷰를 사용해 데이터 접근할 때 거치는 과정

- 시스템 카탈로그로부터 뷰의 정의, select문을 검색
- 기본 릴레이션에 대한 뷰의 접근 권한을 검사
- 뷰에 대한 질의를 기본 릴레이션에 대한 동등한 질의로 변환

뷰의 장점

- 뷰는 복잡한 질의를 간단하게 표현할 수 있게 함
- 뷰는 데이터 무결성을 보장하는데 활용됨
- 데이터 독립성을 제공 → 데이터베이스의 구조가 바뀌어도 기존의 질의를 다시 작성할 필요성을 줄이는데 사용될 수 있음
- 데이터 보안 기능 제공 → 뷰의 원본이 되는 기본 릴레이션에 직접 접근할 수 있는 권한을 부여하지 않고 뷰를 통해 데이터에 접근하도록 하기 때문에 보안 메커니즘으로 사용할 수 있음
- 동일한 데이터에 대한 여러 가지 뷰를 제공 → 사용자들의 그룹이 각자 특정한 기준에 따라 데이터를 접근하도록 함

뷰의 갱신

뷰에 대한 갱신도 기본 릴레이션에 대한 갱신으로 변환됨

시스템 카탈로그

= 메타데이터

사용자 및 질의 최적화 모듈 등 DBMS 자신의 구성요소에 의해서 사용됨

관계 DBMS마다 서로 다른 형태로 시스템 카탈로그 기능을 제공

데이터 사전 또는 시스템 테이블이라고도 부름

⇒ 질의에 활용 ~ select문의 문법이 정확한지 검사함

질의 최적화

DBMS가 질의를 수행하는 여러 가지 방법들 중에서 가장 비용이 적게 드는 방법을 찾는 과정

관계 DBMS의 시스템 카탈로그

사용자 릴레이션과 마찬가지 형태로 저장되어 사용자 릴레이션에 적용되는 회복 기법과 동시성 제어 기법을 동일하게 사용할 수 있음

릴레이션, 애트리뷰트, 인덱스, 사용자, 권한 등 각 유형마다 별도의 릴레이션이 유지됨

시스템 카탈로그의 갱신

어떤 사용자도 시스템 카탈로그를 직접 갱신할 수 없음

delete, update, insert문을 사용해 시스템 카탈로그를 변경할 수 없음

MS SQL Server의 시스템 카탈로그

- 통계 정보는 update statistics문을 사용해 수동으로 갱신
- SQL Server는 릴레이션의 데이터가 변경될 때 주기적으로 통계 정보를 자동으로 갱신
- 통계 정보를 갱신할 필요성이 있는가를 확인하기 위해 샘플링 방법 사용
- 모든 릴레이션에 대한 구성을 정의하는 데이터를 시스템 테이블이라고 부르는 특수한 테이블 집합에 저장