

제 7 장

릴레이션 정규화

- 7.1 정규화 개요
- 7.2 함수적 종속성
- 7.3 릴레이션의 분해(decomposition)
- 7.4 제1정규형, 제2정규형, 제3정규형, BCNF
- 7.5 역정규화
 - 연습문제

7.4 제1정규형, 제2정규형, 제3정규형, BCNF

□ 제1정규형

- ✓ 한 릴레이션 R이 제1정규형을 만족할 필요 충분 조건은 릴레이션 R의 모든 애트리뷰트가 원자값만을 갖는다는 것
- ✓ 즉 릴레이션의 모든 애트리뷰트에 **반복 그룹**(repeating group)이 나타나지 않으면 제1정규형을 만족함

학생	<u>학번</u>	이름	과목번호	주소
	11002	이홍근	{CS310, CS313}	우이동
	24036	김순미	{CS310, CS345}	양재동

[그림 7.13] 반복 그룹

7.4 제1정규형, 제2정규형, 제3정규형, BCNF(계속)

□ 제1정규형을 만족하지 않는 그림 7.13을 제1정규형으로 변환하는 방법

- ✓ 반복 그룹 애트리뷰트에 나타나는 집합에 속한 각 값마다 하나의 튜플로 표현

학생	<u>학번</u>	이름	<u>과목번호</u>	주소
	11002	이홍근	CS310	우이동
	11002	이홍근	CS313	우이동
	24036	김순미	CS345	양재동
	24036	김순미	CS310	양재동

[그림 7.14] 애트리뷰트에 원자값만 있는 릴레이션

7.4 제1정규형, 제2정규형, 제3정규형, BCNF(계속)

□ 제1정규형을 만족하지 않는 그림 7.13을 제1정규형으로 변환하는 방법(계속)

- ✓ 모든 반복 그룹 애트리뷰트들을 분리해서 새로운 릴레이션에 넣음. 원래 릴레이션의 기본 키를 새로운 릴레이션에 애트리뷰트로 추가함

학생1

<u>학번</u>	이름	주소
11002	이홍근	우이동
24036	김순미	양재동

수강

<u>학번</u>	<u>과목번호</u>
11002	CS310
11002	CS313
24036	CS345
24036	CS310

[그림 7.15] 두 릴레이션으로 분해

7.4 제1정규형, 제2정규형, 제3정규형, BCNF(계속)

□ 제1정규형에 존재하는 갱신 이상

- ✓ 그림 7.16의 학생 릴레이션은 모든 애트리뷰트가 원자값을 가지므로 제1정규형을 만족함
- ✓ 이 릴레이션의 기본 키는 (학번, 과목번호)

학생	학번	학과이름	학과전화번호	과목번호	학점
	11002	컴퓨터과학	210-2261	CS310	A0
	11002	컴퓨터과학	210-2261	CS313	B0
	24036	정보통신	210-2585	IC214	B+

[그림 7.16] 제1정규형을 만족하는 릴레이션

7.4 제1정규형, 제2정규형, 제3정규형, BCNF(계속)

□ 수정 이상

- ✓ 한 학과에 소속한 학생 수만큼 그 학과의 전화번호가 중복되어 저장되므로 여러 학생이 소속된 학과의 전화번호가 변경되었을 때 그 학과에 속한 모든 학생들의 튜플에서 전화번호를 수정하지 않으면 데이터베이스의 일관성이 유지되지 않음

□ 삽입 이상

- ✓ 한 명의 학생이라도 어떤 학과에 소속되지 않으면 이 학과에 관한 튜플을 삽입할 수 없음. 왜냐하면 학번이 기본 키의 구성요소인데 엔티티 무결성 제약조건에 따라 기본 키에 널값을 입력할 수 없기 때문

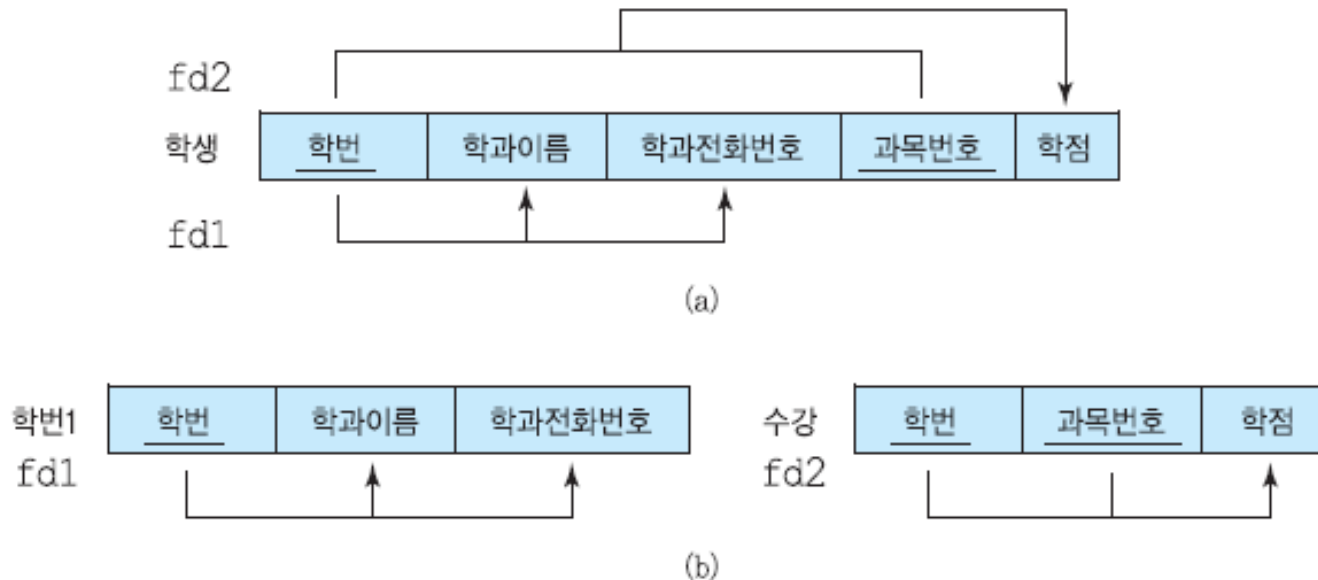
□ 삭제 이상

- ✓ 어떤 학과에 소속된 마지막 학생 튜플을 삭제하면 이 학생이 소속된 학과에 관한 정보도 삭제됨

7.4 제1정규형, 제2정규형, 제3정규형, BCNF(계속)

□ 갱신 이상이 생기는 이유

- ✓ 기본 키에 대한 부분 함수적 종속성이 학생 릴레이션에 존재함



[그림 7.17] (a) 부분 함수적 종속성이 존재하는 릴레이션(제1정규형)
(b) 부분 함수적 종속성이 존재하지 않도록 분해된 두 릴레이션(제2정규형)

7.4 제1정규형, 제2정규형, 제3정규형, BCNF(계속)

□ 제2정규형

- ✓ 한 릴레이션 R 이 제2정규형을 만족할 필요 충분 조건은 릴레이션 R 이 제1정규형을 만족하면서, 어떤 후보 키에도 속하지 않는 모든 애트리뷰트들이 R 의 기본 키에 완전하게 함수적으로 종속하는 것
- ✓ 기본 키가 두 개 이상의 애트리뷰트로 구성되었을 경우에만 제1정규형이 제2정규형을 만족하는가를 고려할 필요가 있음

7.4 제1정규형, 제2정규형, 제3정규형, BCNF(계속)

□ 제2정규형에 존재하는 갱신 이상

- ✓ 그림 7.18의 학생1 릴레이션의 기본 키는 한 애트리뷰트인 학번이므로 제2정규형을 만족함

학생1	<u>학번</u>	학과이름	학과전화번호
	11002	컴퓨터과학	210-2261
	24036	정보통신	210-2585
	11048	컴퓨터과학	210-2261

[그림 7.18] 제2정규형을 만족하는 릴레이션

7.4 제1정규형, 제2정규형, 제3정규형, BCNF(계속)

□ 수정 이상

- ✓ 여러 학생이 소속된 학과의 전화번호가 변경되었을 때 그 학과에 속한 모든 학생들의 투플에서 전화번호를 수정하지 않으면 데이터베이스의 일관성이 유지되지 않음

□ 삽입 이상

- ✓ 어떤 학과를 신설해서 아직 소속 학생이 없으면 그 학과의 정보를 입력할 수 없다. 왜냐하면 학번이 기본 키인데 엔티티 무결성 제약조건에 따라 기본 키에 널값을 입력할 수 없기 때문

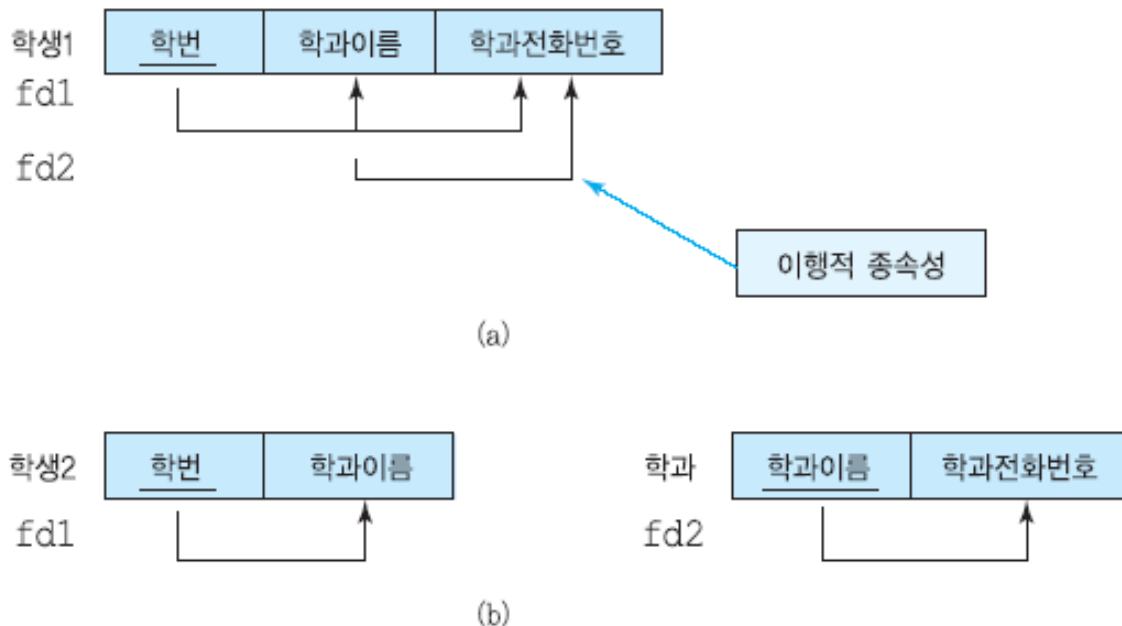
□ 삭제 이상

- ✓ 어떤 학과에서 마지막 학생의 투플이 삭제되면 그 학과의 전화번호도 함께 삭제됨

7.4 제1정규형, 제2정규형, 제3정규형, BCNF(계속)

□ 갱신 이상이 생기는 이유

- ✓ 학생1 릴레이션에 이행적 종속성이 존재하기 때문



[그림 7.19] (a) 이행적 종속성이 존재하는 릴레이션(제2정규형)
(b) 이행적 종속성이 존재하지 않도록 분해된 두 릴레이션(제3정규형)

7.4 제1정규형, 제2정규형, 제3정규형, BCNF(계속)

□ 제3정규형

- ✓ 한 릴레이션 R이 제3정규형을 만족할 필요 충분 조건은 릴레이션 R이 제2정규형을 만족하면서, 키가 아닌 모든 애트리뷰트가 릴레이션 R의 기본 키에 이행적으로 종속하지 않는 것

7.4 제1정규형, 제2정규형, 제3정규형, BCNF(계속)

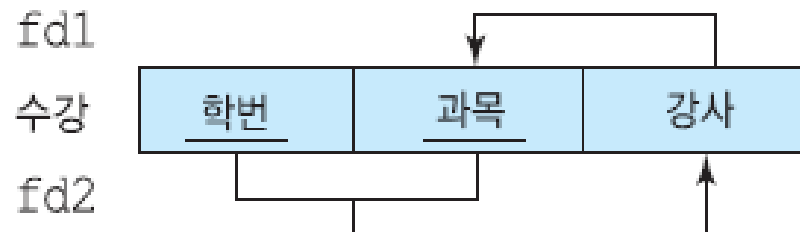
□ 제3정규형에 존재하는 갱신 이상

- ✓ 그림 7.20의 수강 릴레이션에서 각 학생은 여러 과목을 수강할 수 있고, 각 강사는 한 과목만 가르침. 이 릴레이션의 기본 키는 (학번, 과목)
- ✓ 키가 아닌 강사 애트리뷰트가 기본 키에 완전하게 함수적으로 종속하므로 제2정규형을 만족하고, 키가 아닌 강사 애트리뷰트가 기본 키에 직접 종속하므로 제3정규형도 만족함
- ✓ 이 릴레이션에는 아래와 같은 함수적 종속성들이 존재함
 - (학번, 과목) → 강사
 - 강사 → 과목

7.4 제1정규형, 제2정규형, 제3정규형, BCNF(계속)

수강	<u>학번</u>	<u>과목</u>	강사
	11002	데이터베이스	이영준
	11002	운영 체제	고성현
	24036	자료 구조	엄영지
	24036	데이터베이스	조민형
	11048	데이터베이스	이영준

[그림 7.20] 제3정규형을 만족하는 릴레이션



[그림 7.21] 수강 릴레이션에 존재하는 함수적 종속성

7.4 제1정규형, 제2정규형, 제3정규형, BCNF(계속)

□ 수정 이상

- ✓ 여러 학생이 수강 중인 어떤 과목의 강사가 변경되었을 때 그 과목을 수강하는 모든 학생들의 튜플에서 강사를 수정하지 않으면 데이터베이스의 일관성이 유지되지 않음

□ 삽입 이상

- ✓ 어떤 과목을 신설하여 아직 수강하는 학생이 없으면 어떤 강사가 그 과목을 가르친다는 정보를 입력할 수 없음
- ✓ 왜냐하면 학번이 기본 키를 구성하는 애트리뷰트인데 엔티티 무결성 제약조건에 따라 기본 키를 구성하는 애트리뷰트에 널값을 입력할 수 없기 때문

□ 삭제 이상

- ✓ 어떤 과목을 이수하는 학생이 한 명밖에 없는데 이 학생의 튜플을 삭제하면 그 과목을 가르치는 강사에 관한 정보도 함께 삭제됨

7.4 제1정규형, 제2정규형, 제3정규형, BCNF(계속)

□ 갱신 이상이 생기는 이유

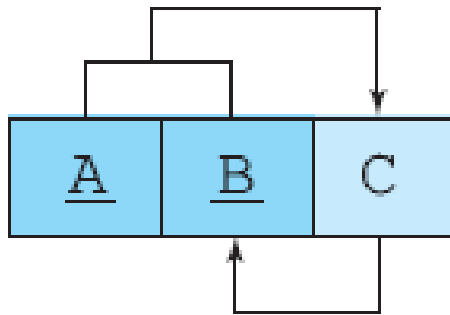
- ✓ 수강 릴레이션에서 키가 아닌 애트리뷰트가 다른 애트리뷰트를 결정하기 때문
- ✓ 이 릴레이션의 후보 키는 (학번, 과목)과 (학번, 강사)

7.4 제1정규형, 제2정규형, 제3정규형, BCNF(계속)

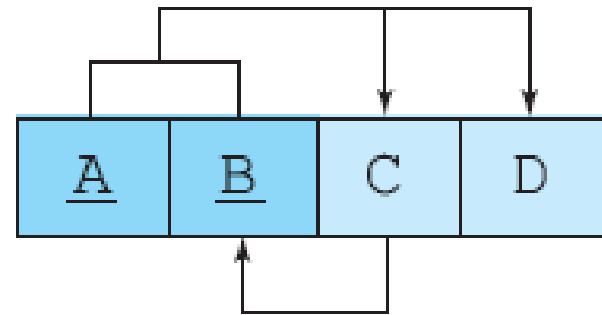
□ BCNF

- ✓ 한 릴레이션 R이 BCNF를 만족할 필요 충분 조건은 릴레이션 R이 제3정규형을 만족하고, 모든 결정자가 후보 키이어야 함
- ✓ 위의 수강 릴레이션에서 강사 애트리뷰트는 후보 키가 아님에도 불구하고 과목 애트리뷰트를 결정하기 때문에 BCNF가 아님
- ✓ 제3정규형을 만족하는 대부분의 릴레이션들은 BCNF도 만족함
- ✓ 하나의 후보 키만을 가진 릴레이션이 제3정규형을 만족하면 동시에 BCNF도 만족함
- ✓ 제3정규형을 만족하는 릴레이션을 BCNF으로 정규화하려면 키가 아니면서 결정자 역할을 하는 애트리뷰트와 그 결정자에 함수적으로 종속하는 애트리뷰트를 하나의 테이블에 넣음. 이 릴레이션에서 결정자는 기본 키가 됨
- ✓ 그 다음에는 기존 릴레이션에 결정자를 남겨서 기본 키의 구성요소가 되도록 함. 또한 이 결정자는 새로운 릴레이션에 대한 외래키 역할도 함

7.4 제1정규형, 제2정규형, 제3정규형, BCNF(계속)



(a) 애트리뷰트가 세 개



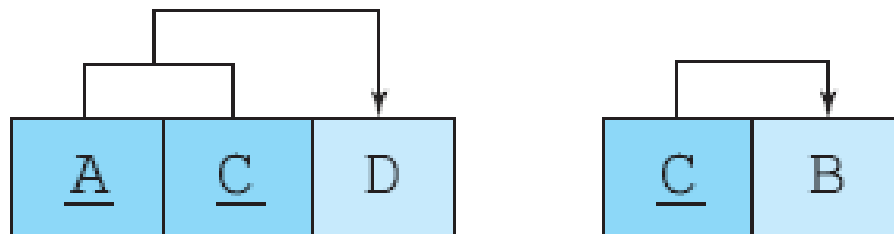
(b) 애트리뷰트가 네 개

[그림 7.22] 제3정규형을 만족하지만 BCNF는 만족하지 않는 릴레이션

7.4 제1정규형, 제2정규형, 제3정규형, BCNF(계속)



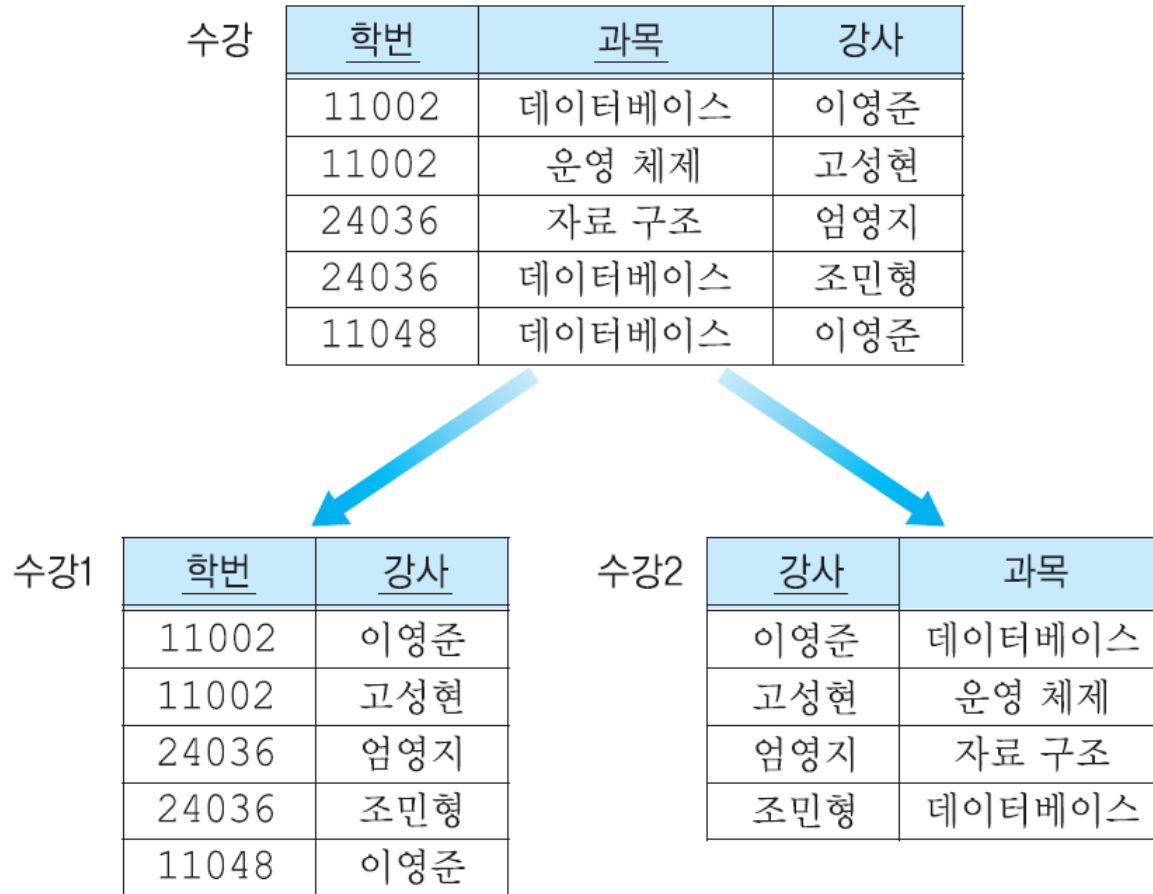
(a) 애트리뷰트가 세 개



(b) 애트리뷰트가 네 개

[그림 7.23] 제3정규형을 BCNF로 분해

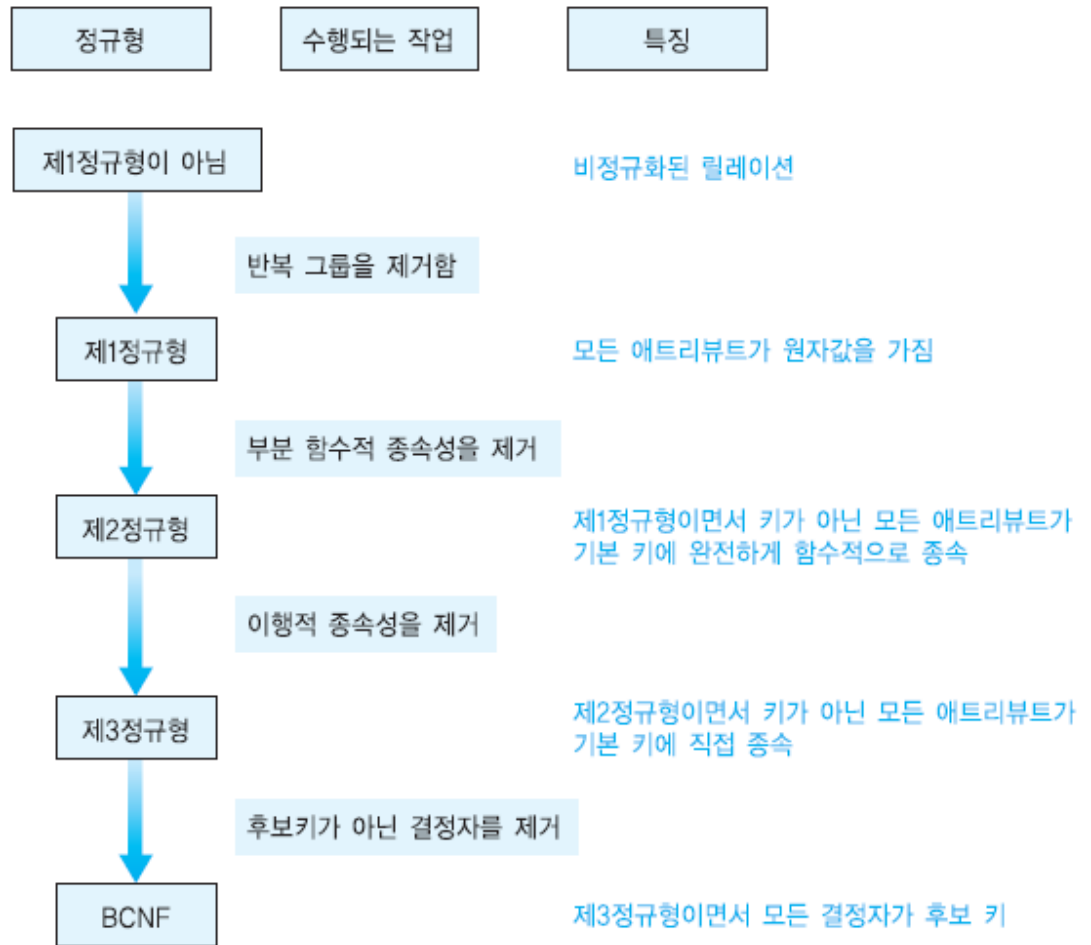
7.4 제1정규형, 제2정규형, 제3정규형, BCNF(계속)



[그림 7.24] 제3정규형을 BCNF로 정규화

7.4 제1정규형, 제2정규형, 제3정규형, BCNF(계속)

□ 여러 정규형의 요약



[그림 7.25] 각 정규형의 특징과 정규화 과정

7.5 역정규화(denormalization)

□ 역정규화

- ✓ 정규화 단계가 진행될수록 중복이 감소하고 갱신 이상도 감소됨
- ✓ 정규화가 진전될수록 무결성 제약조건을 시행하기 위해 필요한 코드의 양도 감소됨
- ✓ 정규화가 데이터베이스 설계의 중요한 요소이지만 성능상의 관점에서만 보면 높은 정규형을 만족하는 릴레이션 스키마가 최적인 것은 아님
- ✓ 한 정규형에서 다음 정규형으로 진행될 때마다 하나의 릴레이션이 최소한 두 개의 릴레이션으로 분해됨
- ✓ 분해되기 전의 릴레이션을 대상으로 질의를 할 때는 조인이 필요 없지만 분해된 릴레이션을 대상으로 질의를 할 때는 같은 정보를 얻기 위해서 보다 많은 릴레이션들을 접근해야 하므로 조인의 필요성이 증가함

7.5 역정규화(계속)

예 : 조인의 필요성

제2정규형을 만족하는 그림 7.18의 학생1 릴레이션에서 “학번이 11002인 학생이 속한 학과의 이름과 전화번호를 검색하라”는 질의를 아래와 같은 SELECT문으로 표현한다. 한 릴레이션에서 필요한 정보를 모두 찾을 수 있으므로 조인이 필요 없다.

```
SELECT      학과이름, 학과전화번호
FROM        학생1
WHERE       학번 = '11002';
```

그러나 정규화 과정을 거쳐 그림 7.18의 릴레이션이 그림 7.19(b)처럼 두 개의 릴레이션으로 분해되면 동일한 정보를 찾기 위해 아래와 같이 조인을 포함한 SELECT문이 사용된다.

```
SELECT      학과이름, 학과전화번호
FROM        학생2, 학과
WHERE       학번 = '11002'
AND 학생2.학과이름 = 학과.학과이름;
```



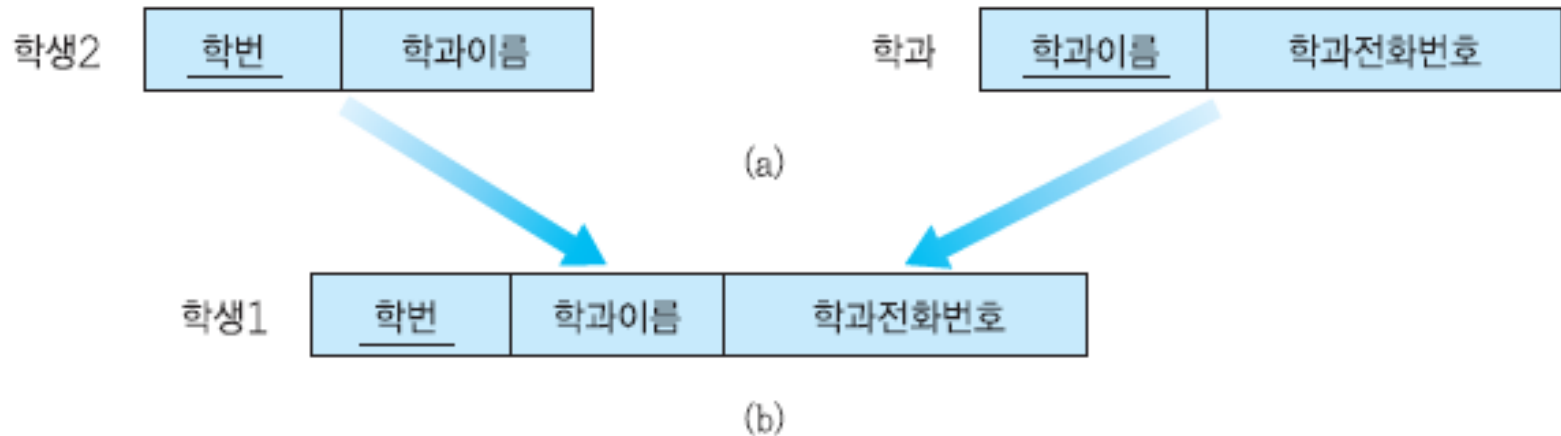
조인조건

7.5 역정규화(계속)

□ 역정규화(계속)

- ✓ 때로 데이터베이스 설계자는 응용의 요구 사항에 따라 데이터베이스 설계의 일부분을 역정규화함으로써 데이터 중복 및 갱신 이상을 대가로 치르면서 성능상의 요구를 만족시키기도 함
- ✓ 많은 데이터베이스 응용에서 검색 질의의 비율이 갱신 질의의 비율보다 훨씬 높음. 역정규화는 주어진 응용에서 빈번하게 수행되는 검색 질의들의 수행 속도를 높이기 위해서 이미 분해된 두 개 이상의 릴레이션들을 합쳐서 하나의 릴레이션으로 만드는 작업
- ✓ 즉 역정규화는 보다 낮은 정규형으로 되돌아가는 것

7.5 역정규화(계속)



[그림 7.26] (a) 제3정규형을 만족하도록 분해된 두 릴레이션
(b) 역정규화된 릴레이션