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

7장. 릴레이션 정규화

- 릴레이션 정규화
  - ✓ 부주의한 데이터베이스 설계는 제어할 수 없는 데이터 중복을 야기하여 여러 가지 갱신 이상(update anomaly)을 유발함
  - ✓ 어떻게 좋은 데이터베이스 설계를 할 것인가? 데이터베이스에 어떤 릴레이션들을 생성할 것인가? 각 릴레이션에 어떤 애트리뷰트들을 둘 것인가?
  - ✓ 정규화(normalization)는 주어진 릴레이션 스키마를 함수적 종속성과 기본 키를 기반으로 분석하여, 원래의 릴레이션을 분해함으로써 중복과 세 가지 갱신 이상을 최소화함

7.1 정규화 개요(계속)

- 갱신 이상(update anomaly)
  - ✓ 수정 이상(modification anomaly)
    - 반복된 데이터 중에 일부만 수정하면 데이터의 불일치가 발생
  - ✓ 삽입 이상(insertion anomaly)
    - 불필요한 정보를 함께 저장하지 않고는 어떤 정보를 저장하는 것이 불가능
  - ✓ 삭제 이상(deletion anomaly)
    - 유용한 정보를 함께 삭제하지 않고는 어떤 정보를 삭제하는 것이 불가능

7.1 정규화 개요(계속)

- 수정 이상
  - 중복 데이터 중에서 일부만 갱신되어 정보의 모순이 발생하는 것

학 번	과 목 명	성적	이름
100	전자계산기구조	92	~~김사랑~~ 김소연 ←
101	데이터베이스	82	오지호
100	운영체제	90	김사랑 ←
101	데이터 통신	76	오지호
102	운영체제	82	이선균

7.1 정규화 개요(계속)

- 삽입 이상
  - 불필요한 정보를 함께 저장하지 않음
  - 어떤 정보를 저장하는 것이 불가능하기에 원하지 않는 정보를 강제로 삽입해야 하는 것

학 번	과 목 명	성적	이름
100	전자계산기구조	92	김사랑 김소연
101	데이터베이스	82	오지호
100	운영체제	90	김사랑
101	데이터 통신	76	오지호
102	운영체제	82	이선균
103	?	?	한예슬

기본키에는 널 값을 저장할 수 없기 때문에 한예슬이란 학생을 수강 릴레이션에 삽입하려고 한다면 수강 신청하지도 않은 가상의 과목명을 임시로라도 삽입해야 함

7.1 정규화 개요(계속)

- 삭제 이상
  - 유용한 정보를 함께 삭제하지 않고는 어떤 정보를 삭제하는 것이 불가능한 것

학 번	과 목 명	성적	이름
100	전자계산기구조	92	김사랑 김소연
101	데이터베이스	82	오지호
100	운영체제	90	김사랑
101	데이터 통신	76	오지호
~~102~~	~~운영체제~~	~~82~~	~~이선균~~

102번 학생의 이름이 이선균이라는 정보까지도 연쇄적으로 삭제되어 이 학생의 정보가 모두 손실

7.1 정규화 개요(계속)

- 정규형(normal form)의 종류
  - ✓ 제1정규형(first normal form), 제2정규형(second normal form), 제3정규형(third normal form), BCNF(Boyce-Codd normal form), 제4정규형(fourth normal form), 제5정규형(fifth normal form)
  - ✓ 일반적으로 산업계의 데이터베이스 응용에서 데이터베이스를 설계할 때 BCNF까지만 고려함

7.2 함수적 종속성

❑ 함수적 종속성의 개요

- ✓ 정규화 이론의 핵심
- ✓ 릴레이션의 애트리뷰트들의 의미로부터 결정됨
- ✓ 릴레이션 스키마에 대한 주장이지, 릴레이션의 특정 인스턴스에 대한 주장이 아님
- ✓ 릴레이션의 가능한 모든 인스턴스들이 만족해야 함
- ✓ 실세계에 대한 지식과 응용의 의미를 기반으로 어떤 함수적 종속성들이 존재하는가를 파악해야 함
- ✓ 함수적 종속성은 제2정규형부터 BCNF까지 적용됨

7.2 함수적 종속성(계속)

- 결정자(determinant)
  - ✓ 어떤 애트리뷰트의 값은 다른 애트리뷰트의 값을 고유하게 결정할 수 있음
  - ✓ 그림 7.4의 사원 릴레이션에서 사원번호는 사원이름을 고유하게 결정함
  - ✓ 주소는 사원이름을 고유하게 결정하지 못함
  - ✓ 결정자는 주어진 릴레이션에서 다른 애트리뷰트(또는 애트리뷰트들의 집합)를 고유하게 결정하는 하나 이상의 애트리뷰트를 의미
  - ✓ 결정자를 아래와 같이 표기하고, 이를 “A가 B를 결정한다”(또는 “A는 B의 결정자이다”)라고 말함

$$\mathbf{A} \rightarrow \mathbf{B}$$

7.2 함수적 종속성(계속)

사원

사원번호	사원이름	주소	전화번호	직책	부서번호	부서이름
4257	정미림	홍제동	731-3497	팀장	1	홍보
1324	이범수	양재동	653-7412	프로그래머	2	개발
1324	이범수	양재동	653-7412	웹 디자이너	1	홍보
3609	안명석	양재동	425-8520	팀장	3	홍보

[그림 7.4] 사원 릴레이션

사원번호 → 사원이름

사원번호 → 주소

사원번호 → 전화번호

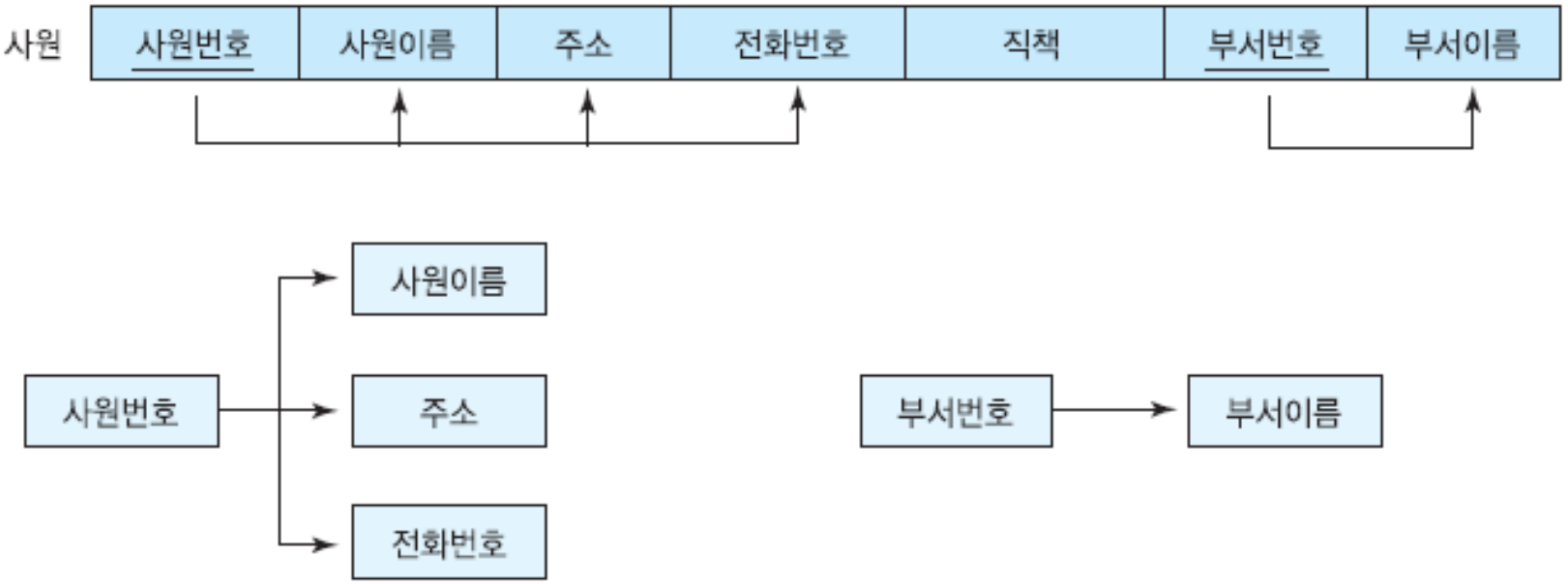
부서번호 → 부서이름

7.2 함수적 종속성(계속)

❑ 함수적 종속성

- ✓ 만일 애트리뷰트 A가 애트리뷰트 B의 결정자이면 B가 A에 함수적으로 종속한다고 말함
- ✓ 다른 말로 표현하면, 주어진 릴레이션 R에서 애트리뷰트 B가 애트리뷰트 A에 함수적으로 종속하는 필요 충분 조건은 각 A 값에 대해 반드시 한 개의 B 값이 대응된다는 것
- ✓ 예: 사원번호가 사원이름, 주소, 전화번호의 결정자이므로 사원이름, 주소, 전화번호는 사원번호에 함수적으로 종속
- ✓ 예: 직책은 (사원번호, 부서번호)에 함수적으로 종속하지, 사원번호에 함수적으로 종속하지는 않음

7.2 함수적 종속성(계속)



[그림 7.5] 사원 릴레이션의 함수적 종속성의 두 가지 다이어그램

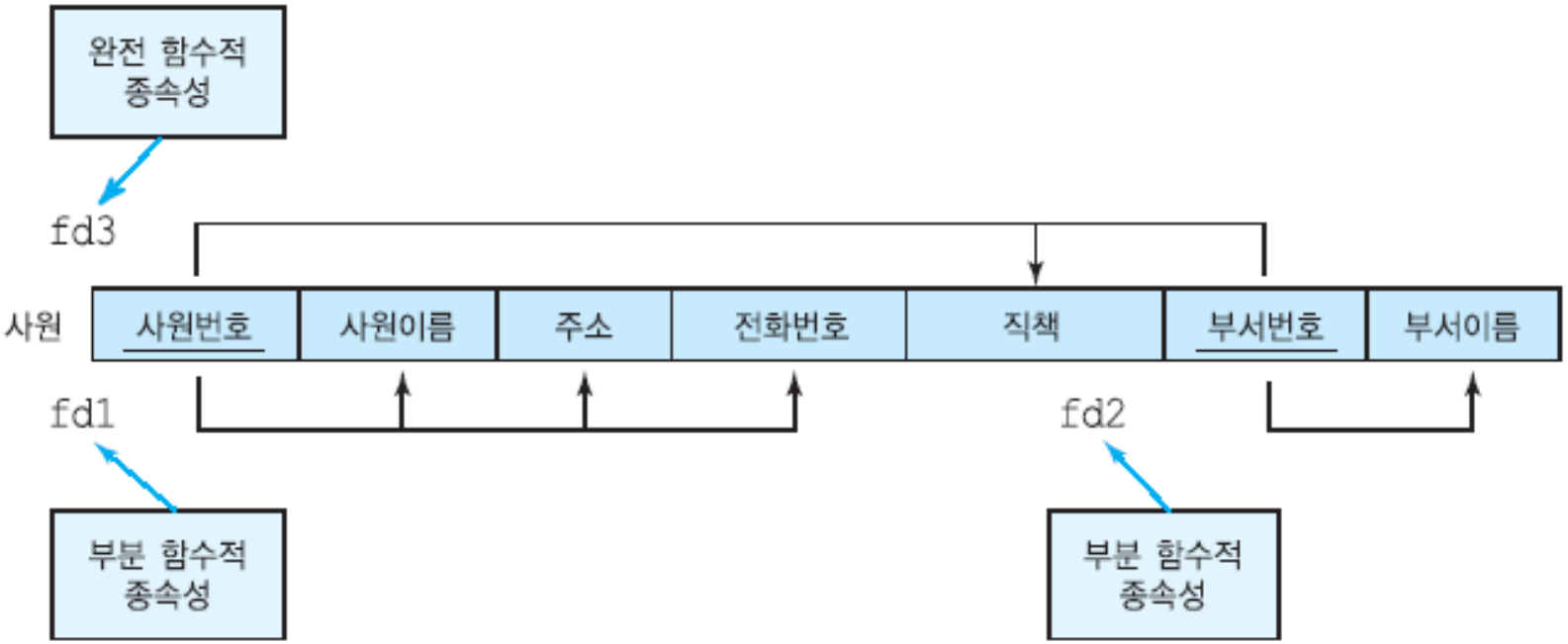
7.2 함수적 종속성(계속)

- 완전 함수적 종속성(FFD: Full Functional Dependency)
  - ✓ 주어진 릴레이션 R에서 애트리뷰트 B가 애트리뷰트 A에 함수적으로 종속하면서 애트리뷰트 A의 어떠한 진부분 집합에도 함수적으로 종속하지 않으면 애트리뷰트 B가 애트리뷰트 A에 완전하게 함수적으로 종속한다고 말함
  - ✓ 여기서 애트리뷰트 A는 복합 애트리뷰트

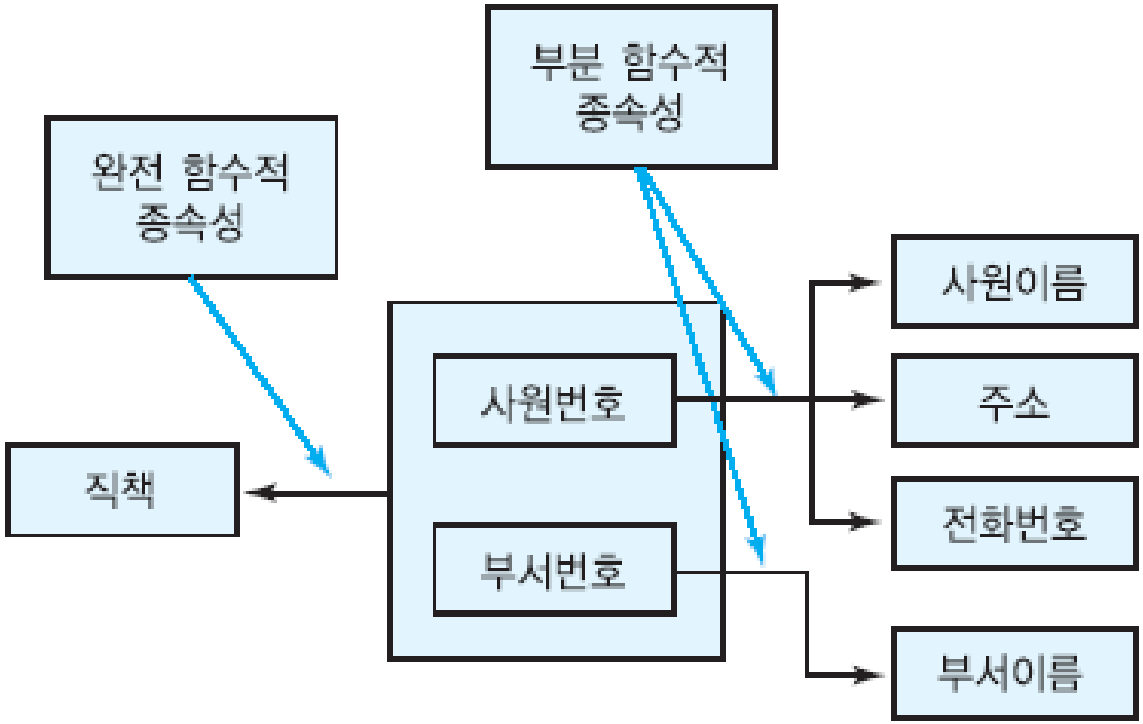
7.2 함수적 종속성(계속)

예 : 완전 함수적 종속성과 부분 함수적 종속성

그림 7.6에서 fd3은 완전 함수적 종속성을 나타내고, fd1과 fd2는 부분 함수적 종속성을 나타낸다.



7.2 함수적 종속성(계속)



[그림 7.6] 완전 함수적 종속성과 부분 함수적 종속성

7.2 함수적 종속성(계속)

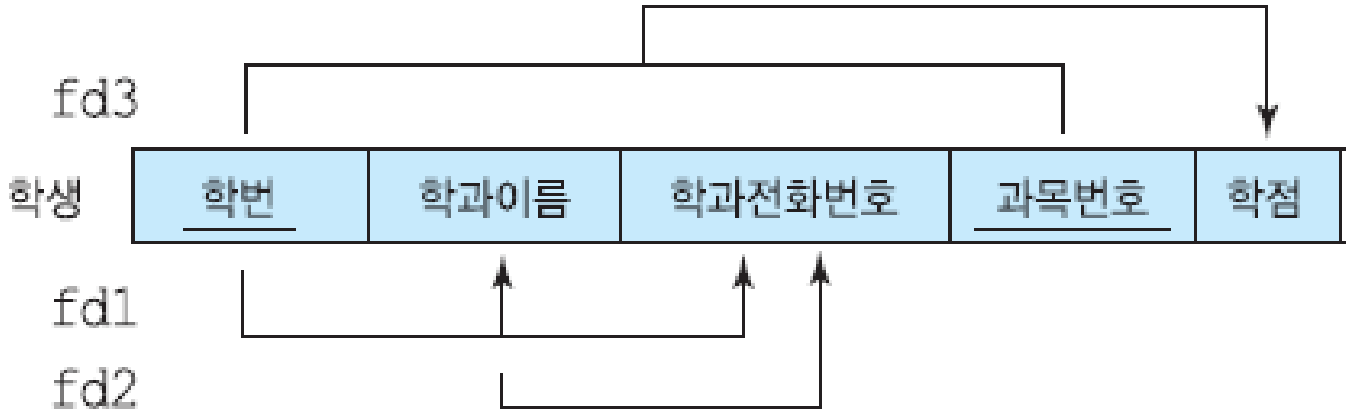
- 이행적 함수적 종속성(transitive FD)
  - ✓ 한 릴레이션의 애트리뷰트 A, B, C가 주어졌을 때 애트리뷰트 C가 이행적으로 A에 종속한다(A→C)는 것의 필요 충분 조건은

    A→B ∧ B→C

    가 성립하는 것

  - ✓ A가 릴레이션의 기본 키라면 키의 정의에 따라 A→B와 A→C가 성립. 만일 C가 A외에 B에도 함수적으로 종속한다면 C는 A에 직접 함수적으로 종속하면서 B를 거쳐서 A에 이행적으로 종속

7.2 함수적 종속성(계속)



[그림 7.7] 이행적 함수적 종속성

7.3 릴레이션 분해

❑ 릴레이션 분해

- ✓ 하나의 릴레이션을 두 개 이상의 릴레이션으로 나누는 것
- ✓ 릴레이션을 분해하면 중복이 감소되고 갱신 이상이 줄어드는 장점이 있는 반면에, 바람직하지 않은 문제들을 포함하여 몇 가지 잠재적인 문제들을 야기할 수 있음
  - ▪ 릴레이션이 분해되기 전에는 조인이 필요 없는 질의가 분해 후에는 조인을 필요로 하는 질의로 바뀔 수 있음
  - ▪ 분해된 릴레이션들을 사용하여 원래 릴레이션을 재구성하지 못할 수 있음

7.3 릴레이션 분해(계속)

- 무손실 분해(lossless decomposition)
  - ✓ 분해된 두 릴레이션을 조인하면 원래의 릴레이션에 들어 있는 정보를 완전하게 얻을 수 있음
  - ✓ 여기서 손실이란 정보의 손실을 뜻함
  - ✓ 정보의 손실은 원래의 릴레이션을 분해한 후에 생성된 릴레이션들을 조인한 결과에 들어 있는 정보가 원래의 릴레이션에 들어 있는 정보보다 적거나 많은 것을 모두 포함

7.3 릴레이션 분해(계속)

학생

학번	이름	이메일	과목번호	학점
11002	이홍근	sea@hanmail.net	CS310	A0
11002	이홍근	sea@hanmail.net	CS313	B+
24036	김순미	smkim@venus.uos.ac.kr	CS345	B0
24036	김순미	smkim@venus.uos.ac.kr	CS310	A+

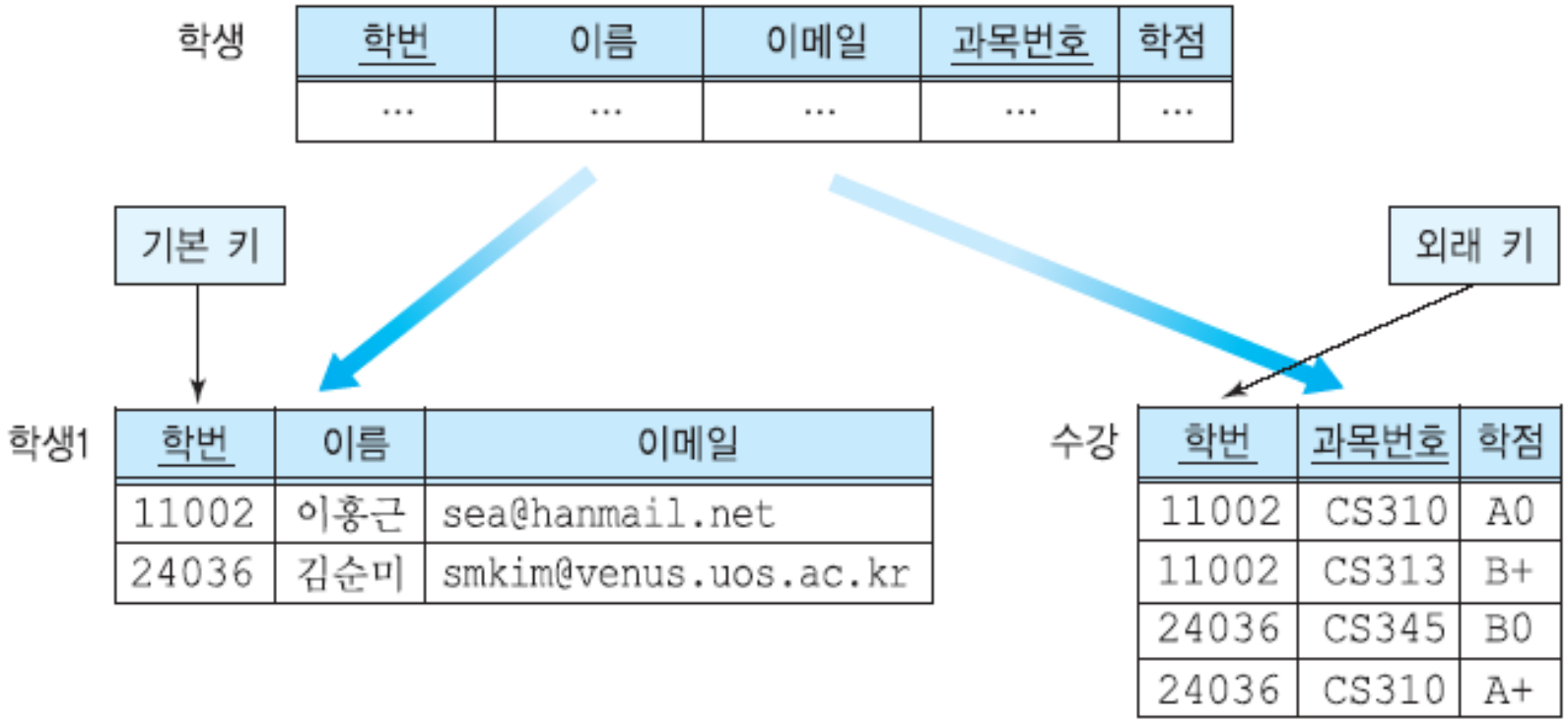
[그림 7.8] 학생 릴레이션

학번 → 이름, 이메일

이메일 → 학번, 이름

(학번, 과목번호) → 학점

7.3 릴레이션 분해(계속)



[그림 7.9] 학생 릴레이션을 두 릴레이션으로 분해

7.3 릴레이션 분해(계속)

학생1

학번	이름	이메일
11002	이홍근	sea@hanmail.net
24036	김순미	smkim@venus.uos.ac.kr

학생2

학번	이름
11002	이홍근
24036	김순미

학생3

학번	이메일
11002	sea@hanmail.net
24036	smkim@venus.uos.ac.kr

[그림 7.10] 불필요한 분해

7.3 릴레이션 분해(계속)

수강

학번	과목번호	학점
11002	CS310	A0
11002	CS313	B+
24036	CS345	B0
24036	CS310	A+

수강1

학번	과목번호
11002	CS310
11002	CS313
24036	CS345
24036	CS310

수강2

학번	학점
11002	A0
11002	B+
24036	B0
24036	A+

[그림 7.11] 나쁜 분해

7.3 릴레이션 분해(계속)

예 : 가짜 투플

그림 7.11의 수강1 릴레이션과 수강2 릴레이션을 학번 애트리뷰트를 사용하여 자연 조인하면 그림 7.12와 같은 결과를 얻는다. 이 릴레이션에서 파란색으로 표시한 투플들은 그림 7.11의 원래 릴레이션인 수강 릴레이션에 존재하지 않는 투플들이므로 가짜 투플에 해당한다.

학번	과목번호	학점
11002	CS310	A0
11002	CS310	B+
11002	CS313	A0
11002	CS313	B+
24036	CS345	B0
24036	CS345	A+
24036	CS310	B0
24036	CS310	A+

[그림 7.12] 가짜 투플