

## 11. 데이터 종속성과 정규화

## 11.1 데이터의 논리적 표현

---

### □ 관계 스키마(relational scheme)의 설계

- 관계 모델을 이용하여 어떻게 실세계를 정확히 표현할 것인가?
  - ◆ 어떤 릴레이션이 필요한가?
  - ◆ 그 릴레이션은 어떤 애트리뷰트들로 구성해야 하는가?

i. 애트리뷰트, 엔티티, 관계성을 파악

ii. 관련된 애트리뷰트들을 릴레이션으로 묶음

데이터 종속성 : 애트리뷰트들간의 관계성

효율적인 데이터 조작

데이터의 중복 방지

iii. 변칙적 성질의 예방

이상(anomaly)

## ▶ 이상 (Anomaly)

---

□ example : 등록 릴레이션

수강	학번	과목번호	성적	학년
	100	C413	A	4
	100	E412	A	4
	200	C123	B	3
	300	C312	A	1
	300	C324	C	1
	300	C413	A	1
	400	C312	A	4
	400	C324	A	4
	400	C413	B	4
	400	C412	C	4
	500	C312	B	2

기본키 : 학번, 과목 번호

---

□ 삭제이상 (deletion anomaly)

- 200번 학생이 'C123'의 등록을 취소  
⇒ 3학년이라는 정보도 함께 삭제됨
- 연쇄 삭제(triggered deletion)에 의한 정보의 손실(loss of information)

□ 삽입이상 (insertion anomaly)

- 600번 학생이 2학년이라는 사실을 삽입  
⇒ 어떤 과목을 등록하지 않는 한 삽입이 불가능  
(∵ 과목 번호가 기본키)
- 원하지 않는 정보의 강제 삽입

□ 수정이상 (update anomaly)

- 400번 학생의 학년을 4에서 3으로 변경  
⇒ 학번이 400인 4개의 튜플 모두를 갱신해야 함
- 중복데이터의 일부 갱신으로 정보의 모순성(inconsistency) 발생

## ▶ 이상의 원인과 해결책

---

### □ 이상의 원인

- 여러 종류의 사실을 하나의 릴레이션에 표현하기 때문
- 즉, 애트리뷰트들 간에 존재하는 여러 종속관계를 하나의 릴레이션에 표현

### □ 이상의 해결

- 애트리뷰트들 간의 종속관계를 분석하여 여러 개의 릴레이션으로 분해(decomposition)  
⇒ 정규화(normalization)

## ▶ 스키마 설계와 변환

---

### □ 스키마 설계 : 데이터베이스의 논리적 설계

- ① 애트리뷰트들을 수집
- ② 이들간에 존재하는 제약 조건 (데이터 종속성)들을 식별
- ③ 이 제약 조건을 기반으로 애트리뷰트들을 릴레이션으로 그룹짓.  
⇒ 스키마 변환 (schema transformation)

### □ 스키마 변환의 원리

- ◆ ① 정보의 무손실
- ◆ ② 데이터의 중복성 감소
- ◆ ③ 분리의 원칙

## ❖ 함수 종속(FD, Functional Dependency)

---

### □ 정의

- 어떤 릴레이션 R에서, 애트리뷰트 X의 값 각각에 대해 애트리뷰트 Y의 값이 하나만 연관
- “Y는 X에 함수 종속”  $X \rightarrow Y$
- 애트리뷰트 X는 Y를 (함수적으로) 결정  
즉, X는 Y의 결정자(determinant)
- X, Y는 복합 애트리뷰트일 수 있음

---

## □ 함수 종속의 의미

- 릴레이션  $R$ 에서 애트리뷰트  $X$ 가 키(후보키 포함)이면,  
 $R$ 의 모든 애트리뷰트  $Y$ 에 대해  $X \rightarrow Y$  성립
- 함수종속  $X \rightarrow Y$ 은, 애트리뷰트  $X$ 가 반드시 키어야 한다는 것을 요구하지 않음.
- 단,  $R$ 의 두 튜플에서 애트리뷰트  $X$ 의 값이 같으면, 이들에 연관된 애트리뷰트  $Y$ 의 값도 반드시 같아야 된다는 것을 의미.



---

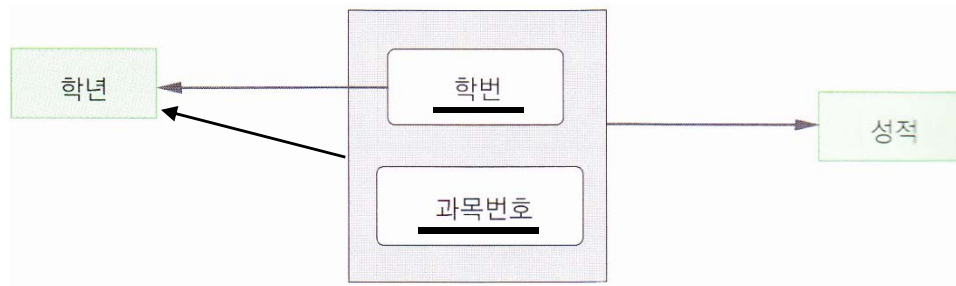
## □ 당연한 함수 종속 (Trivial FD)

- 함수 종속에서
  - ◆ 결정자와 종속자가 같거나
  - ◆ 결정자에 종속자가 포함된 경우는 함수 종속이 당연한 것이 됨.
- 예
  - ◆ 학번  $\rightarrow$  학번
  - ◆ {학번, 이름}  $\rightarrow$  이름
  - {학번, 과목번호}  $\rightarrow$  학번
- 함수 종속은 당연한 것을 제외한 비당연(nontrivial) 함수 종속만을 대상으로 함.

## ※ 함수 종속 다이어그램

### □ 수강지도 릴레이션

- 수강지도 (학번, 과목번호, 성적, 학년)
- ◆ FD : {학번, 과목번호} → 성적  
학번 → 학년



수강	<u>학번</u>	<u>과목번호</u>	성적	학년
	100	C413	A	4
	100	E412	A	4
	200	C123	B	3
	300	C312	A	1
	300	C324	C	1
	300	C413	A	1
	400	C312	A	4
	400	C324	A	4
	400	C413	B	4
	400	E412	C	4
	500	C312	B	2

## ▶ 완전 함수 종속과 부분 함수 종속

---

- 복합 애트리뷰트  $X$ 에 대하여  $X \rightarrow Y$ 가 성립할 때
- 완전 함수 종속 (full functional dependency)
  - $X' \subset X$  이고,  $X' \rightarrow Y$  를 만족하는 애트리뷰트  $X'$  가 존재하지 않으면
  - “ $Y$ 는  $X$ 에 완전 함수 종속”
  - 함수 종속은 완전 함수 종속을 의미
- 부분 함수 종속 (partial functional dependency)
  - $X' \subset X$  이고  $X' \rightarrow Y$  를 만족하는 애트리뷰트  $X'$  가 존재하면
  - “ $Y$ 는  $X$ 에 부분 함수 종속”
  - 부분 함수 종속의 경우는 반드시  $X$ 가 두 개 이상의 애트리뷰트로 구성됨.

## □ 예: 수강지도 릴레이션의 함수 종속

- {학년}은 {학번}에 완전 함수 종속,  
그러나 {학번, 과목번호}에는 부분 함수 종속



{학번, 과목번호} → 학년 :  $X = \{\text{학번}, \text{과목번호}\}$

학번 → 학년 :  $X' = \{\text{학번}\}$

- {성적}은 {학번, 과목번호}에 완전 함수 종속

## □ 추론 규칙

R1: (반사)  $A \supseteq B$ 이면,  $A \rightarrow B$ 이다.

R2: (첨가)  $A \rightarrow B$ 이면,  $AC \rightarrow BC$ 이고  $AC \rightarrow B$ 이다.

R3: (이행)  $A \rightarrow B$ 이고  $B \rightarrow C$ 이면,  $A \rightarrow C$ 이다.

R4: (분해)  $A \rightarrow BC$ 이면,  $A \rightarrow B$ 이다.

R5: (결합)  $A \rightarrow B$ 이고  $A \rightarrow C$ 이면,  $A \rightarrow BC$ 이다.

## ☞ Note

- 함수 종속은 데이터의 의미(data semantics)를 표현
  - ◆ 예: “학번  $\rightarrow$  학년”의 의미는 “학생은 하나의 학년값만 갖는다”
  - ◆ 의미적 제약 조건
- DBMS는 함수 종속을 유지하기 위하여
  - ◆ 함수 종속을 스키마에 명세하는 방법과 함수 종속을 보장하는 방법을 제공하여야 함

## ❖ 기본 정규형

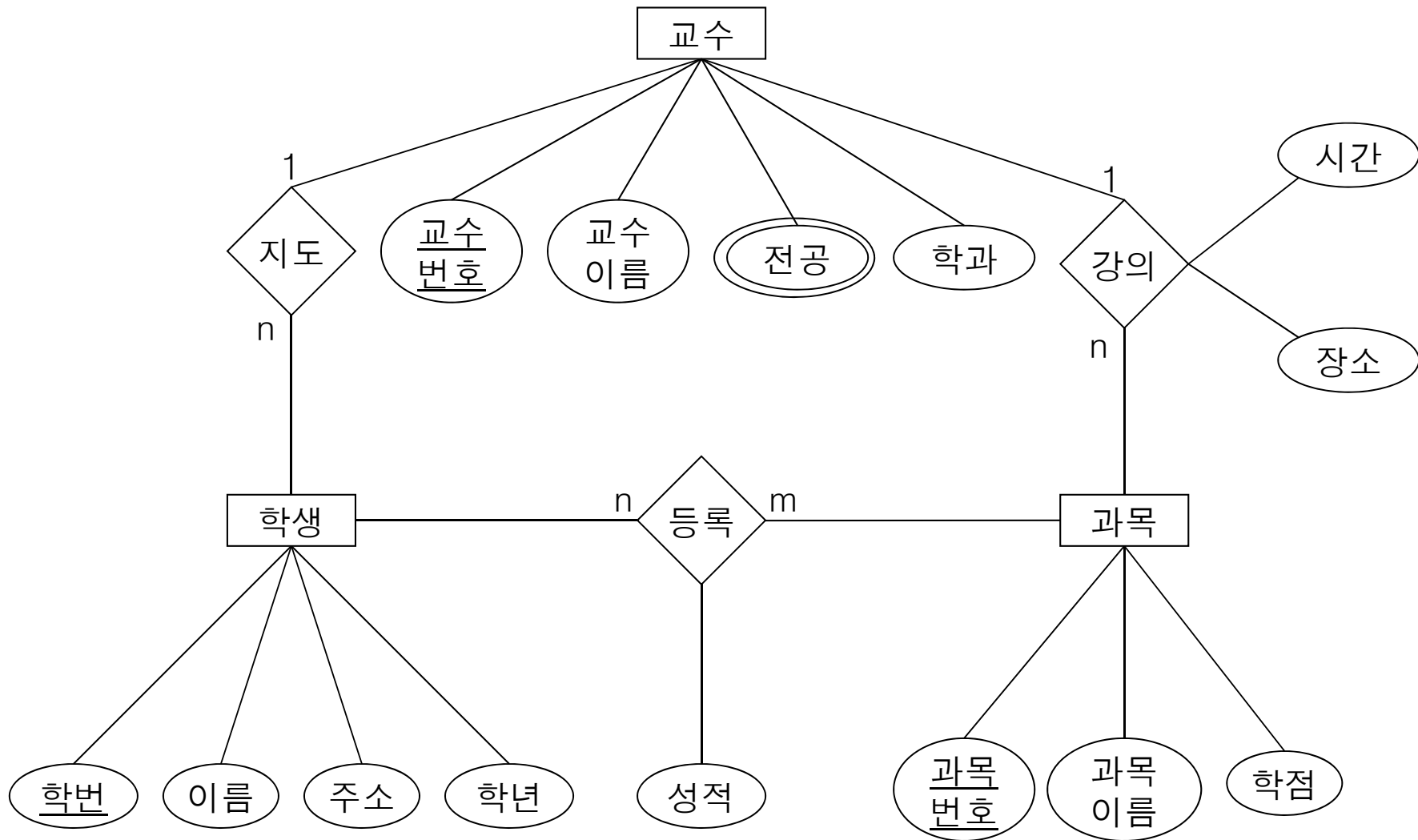
---

- 정규형(Normal Form)
  - 어떤 일련의 제약 조건을 만족하는 릴레이션
- 정규화(Normalization)의 원칙

정규화 = 스키마 변환 ( $S \rightarrow S'$ )

- ① 무손실 표현
  - ◆ 같은 의미의 정보 유지
  - ◆ 그러나 더 바람직한 구조
- ② 데이터의 중복성 감소
- ③ 분리의 원칙
  - ◆ 독립적인 관계는 별개의 릴레이션으로 표현
  - ◆ 릴레이션 각각에 대해 독립적 조작이 가능

## ► Example



---

□ Set-valued attribute “전공”의 처리

- 교수 (교수번호, 교수이름, 전공, 학과) 는 불가능

□ 방법 1: 별도의 테이블을 생성

- 교수 (교수번호, 교수이름, 학과)
- 교수전공 (교수번호, 전공)

□ 방법 2: “전공”의 허용 최대수를 알 수 있다면  
(예를 들어, 3개)

- 교수 (교수번호, 교수이름, 학과, 전공1, 전공2, 전공3)
- 일반적으로 가능한 방법이 아님.



## ▶ 제1정규형 (1NF)

---

### □ 정의

- 모든 도메인이 원자값(atomic value)만으로 된 릴레이션

□ 예 : 수강지도 릴레이션

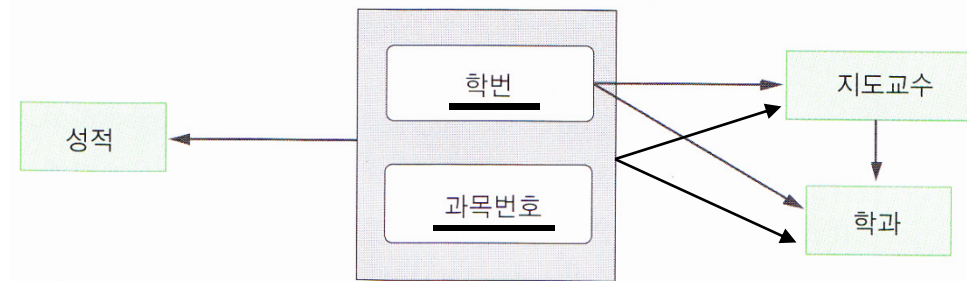
- 수강지도 (학번, 과목번호, 지도교수, 학과, 성적)

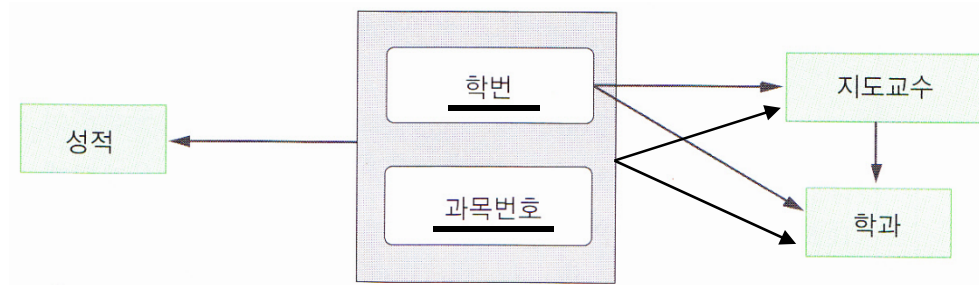
◆ FD : {학번, 과목번호} → 성적

학번 → 지도교수

학번 → 학과

지도교수 → 학과





수강지도

	<u>학번</u>	지도교수	학과	<u>과목번호</u>	성적
	100	P1	컴퓨터	C413	A
	100	P1	컴퓨터	E412	A
	200	P2	전기	C123	B
	300	P3	컴퓨터	C312	A
	300	P3	컴퓨터	C324	C
	300	P3	컴퓨터	C413	A
	400	P1	컴퓨터	C312	A
	400	P1	컴퓨터	C324	A
	400	P1	컴퓨터	C413	B
	400	P1	컴퓨터	E412	C

문제점: “지도교수”와 “학과”에 많은 중복 데이터 포함

---

## □ 1NF(수강지도 릴레이션)에서의 이상

### ① 삽입이상

- ◆ 500번 학생의 지도교수가 P4라는 사실의 삽입은 어떤 교과목을 등록하지 않는 한 불가능

### ② 삭제이상

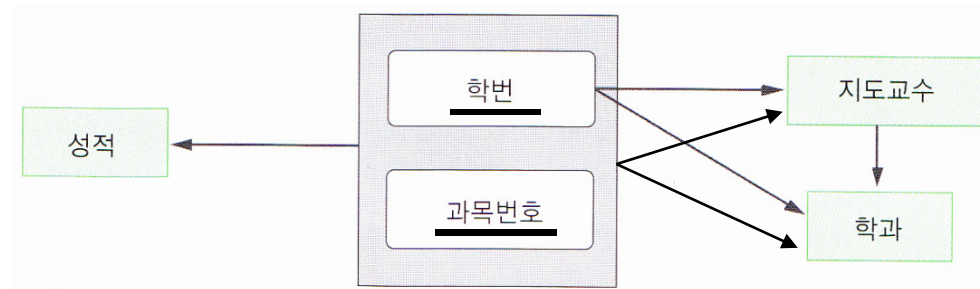
- ◆ 200번 학생이 C123의 등록을 취소하여 이 튜플을 삭제할 경우 지도교수가 P2라는 정보까지 손실됨

### ③ 갱신이상

- ◆ 400번 학생의 지도교수를 P1에서 P3로 변경할 경우 학번이 400인 4개 튜플의 지도교수 값을 P3로 변경해야 함

## □ 1NF 이상의 원인

- 기본키에 부분 함수 종속된 애트리뷰트(지도교수와 학과)가 존재 (그 결과 “지도교수”와 “학과”에 많은 중복 데이터가 포함)



## □ 1NF 이상의 해결

- 프로젝션으로 릴레이션을 분해하여, 부분 함수 종속을 제거  
⇒ 2NF

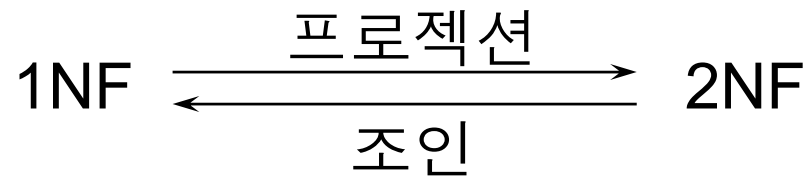
## ▶ 제2정규형 (2NF)

---

### □ 정의

- 1NF이고, 키가 아닌 모든 애트리뷰트들이 기본키에 완전 함수 종속

### □ 무손실 분해(nonloss decomposition)



- 프로젝션하여 분해된 릴레이션들은 자연 조인을 통해 원래의 릴레이션으로 복귀 가능
- 원래의 릴레이션에서 얻을 수 있는 정보는 분해된 릴레이션들로 부터도 얻을 수 있음. 그러나, 그 역은 성립하지 않음 (500번 학생의 지도교수가 P4라는 정보는 원래의 릴레이션에서 표현할 수 없음)

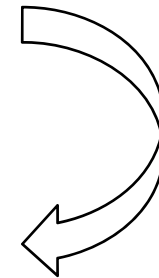
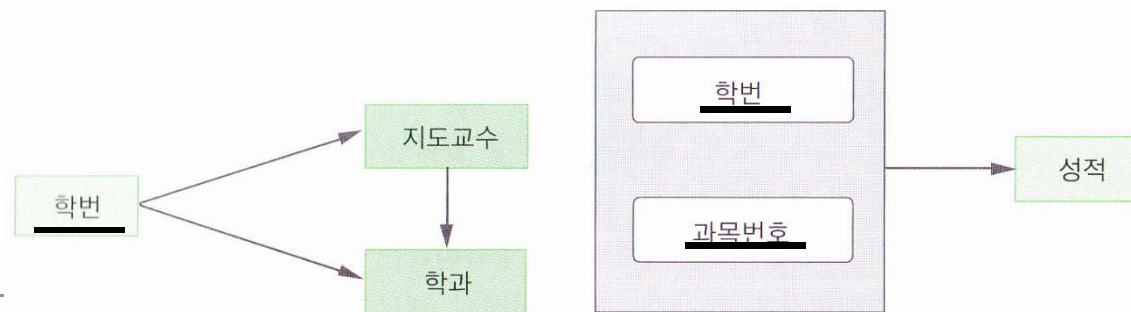
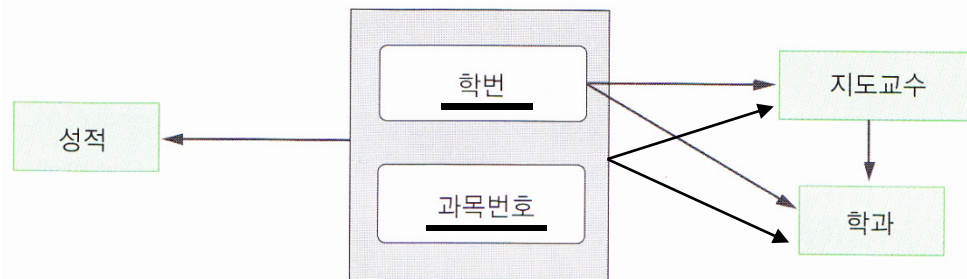
---

☞ Note: Heath의 무손실 분해

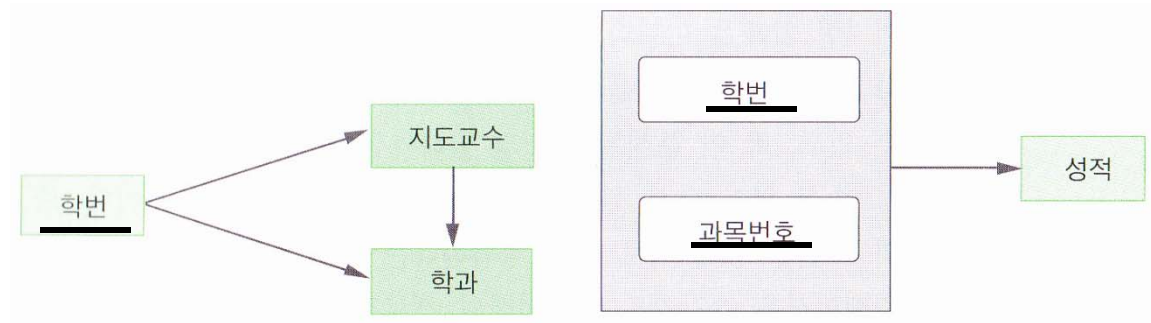
- 세 개의 애트리뷰트 집합  $A, B, C$ 로 구성된  $R(\underline{A}, B, C)$ 에서,  
함수 종속  $A \rightarrow B$ 가 성립하면,  
 $\Rightarrow R1(\underline{A}, B), R2(\underline{A}, C)$ 로 무손실 분해 가능

□ 예 : 수강지도  $\Rightarrow$  지도, 수강 릴레이션

- 지도 (학번, 지도교수, 학과)
  - ◆ FD : 학번  $\rightarrow$  지도교수  
학번  $\rightarrow$  학과  
지도교수  $\rightarrow$  학과
- 수강 (학번, 과목번호, 성적)
  - ◆ FD : (학번, 과목번호)  $\rightarrow$  성적







지도

학번	지도교수	학과
100	P1	컴퓨터
200	P2	전기
300	P3	컴퓨터
400	P1	컴퓨터

수강

학번	과목번호	성적
100	C413	A
100	E412	A
200	C123	B
300	C312	A
300	C324	C
300	C413	A
400	C312	A
400	C324	A
400	C413	B
400	E412	C

## □ 무손실 분해 (1NF $\Rightarrow$ 2NF)

- 지도 =  $\Pi_{\text{학번, 지도교수, 학과}}$  (수강지도)
- 수강 =  $\Pi_{\text{학번, 과목번호, 성적}}$  (수강지도)
- 수강지도 = 지도  $\bowtie$  수강

수강지도

학번	지도교수	학과	과목번호	성적
100	P1	컴퓨터	C413	A
100	P1	컴퓨터	E412	A
200	P2	전기	C123	B
300	P3	컴퓨터	C312	A
300	P3	컴퓨터	C324	C
300	P3	컴퓨터	C413	A
400	P1	컴퓨터	C312	A
400	P1	컴퓨터	C324	A
400	P1	컴퓨터	C413	B
400	P1	컴퓨터	E412	C

→

지도

학번	지도교수	학과
100	P1	컴퓨터
200	P2	전기
300	P3	컴퓨터
400	P1	컴퓨터

수강

학번	과목번호	성적
100	C413	A
100	E412	A
200	C123	B
300	C312	A
300	C324	C
300	C413	A
400	C312	A
400	C324	A
400	C413	B
400	E412	C

---

## □ 2NF(지도 릴레이션)에서의 이상

### ① 삽입이상

- ◆ 어떤 교수가 특정 학과에 속한다는 사실의 삽입 불가능  
(지도학생이 있어야만 교수의 추가가 가능)

### ② 삭제이상

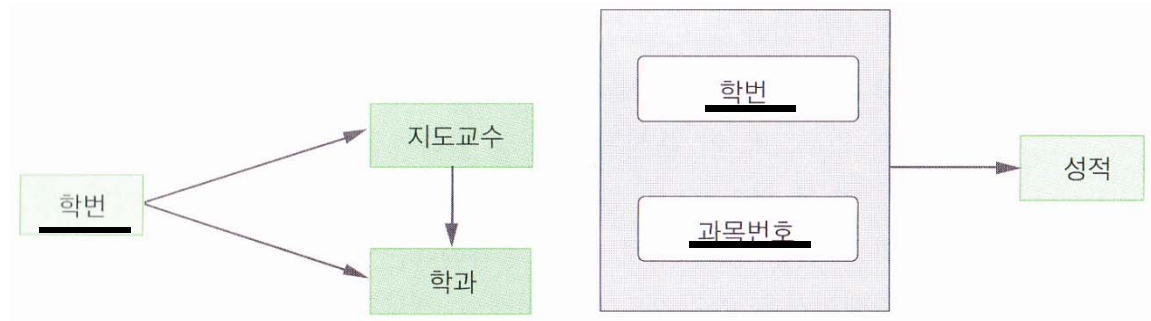
- ◆ 300번 학생의 투플을 삭제하면 지도교수 P3가 컴퓨터공학과에 속한다는 정보 손실

### ③ 갱신이상

- ◆ 지도교수 P1의 소속이 컴퓨터공학과에서 전자과로 변경된다면  
100과 400번 학생의 투플을 모두 변경하여야 함

## □ 2NF 이상의 원인

- 이행적 함수 종속(TD, Transitive Dependency)이 존재  
 $A \rightarrow B$ 와  $B \rightarrow C \Rightarrow A \rightarrow C$   
(즉, 애트리뷰트 C는 애트리뷰트 A에 이행적 함수 종속)  
- TD는 두 개의 상이한 정보(종속성)을 하나의 릴레이션으로 혼합해서 표현한다는 사실을 알려줌.



## □ 2NF 이상의 해결

- 프로젝션으로 릴레이션 분해 (이행적 함수 종속을 제거)  
 $\Rightarrow$  3NF

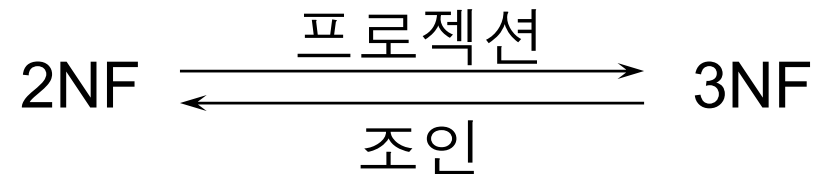
## ▶ 제3정규형 (3NF)

---

### □ 정의(3NF)

- 2NF이고, **키가 아닌 모든 애트리뷰트들이 기본키에 이행적 함수 종속되지 않음**

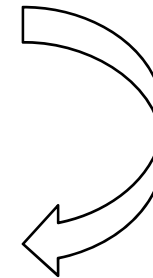
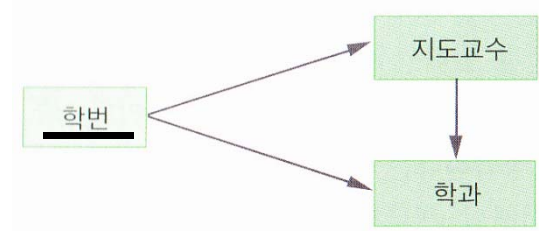
### □ 무손실 분해



- 원래의 릴레이션에서 얻을 수 있는 정보는 분해된 릴레이션들로부터도 얻을 수 있으나 그 역은 성립하지 않음.  
(교수 P4가 수학과에 속한다는 정보의 표현)

□ 예 : 지도  $\Rightarrow$  학생지도, 지도교수학과

- 학생지도 (학번, 지도교수)
  - ◆ FD : 학번  $\rightarrow$  지도교수
- 지도교수학과 (지도교수, 학과)
  - ◆ FD : 지도교수  $\rightarrow$  학과





학생지도

<u>학번</u>	지도교수
100	P1
200	P2
300	P3
400	P1

지도교수학과

<u>지도교수</u>	학과
P1	컴퓨터
P2	전기
P3	컴퓨터

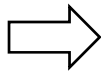
키가 아닌 애트리뷰트 값의 갱신시 불필요한 부작용(이상) 발생 없음

## □ 무손실 분해 (2NF $\Rightarrow$ 3NF)

- 학생지도 =  $\Pi_{\text{학번}, \text{지도교수}}(\text{지도})$   
지도교수학과 =  $\Pi_{\text{지도교수}, \text{학과}}(\text{지도})$
- 지도 = 학생지도  $\bowtie$  지도교수학과

지도

학번	지도교수	학과
100	P1	컴퓨터
200	P2	전기
300	P3	컴퓨터
400	P1	컴퓨터



학생지도

학번	지도교수
100	P1
200	P2
300	P3
400	P1

지도교수학과

지도교수	학과
P1	컴퓨터
P2	전기
P3	컴퓨터



---

## □ 1,2,3NF의 가정

- 기본적으로, 하나의 후보키로부터 기본키가 정의된 경우.
- 특히 3NF의 경우, 2개 이상의 후보키가 서로 중첩되는 경우는 여전히 anomaly가 발생.

## □ 3NF의 단점

- i . 복수의 후보키를 가지고 있고,
  - ii . 후보키들이 복합 애트리뷰트들로 구성되며,
  - iii . 후보키들이 서로 중첩되는 경우에는 적용이 불가능함.
- 3NF을 일반화한 BCNF (strong 3NF)

## ▶ 보이스/코드 정규형(BCNF, Boyce-Codd Normal Form)

---

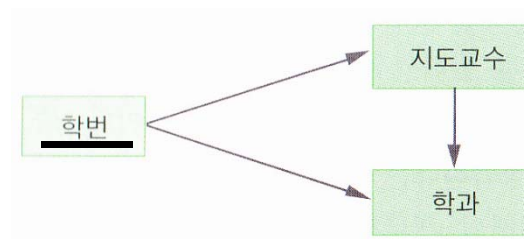
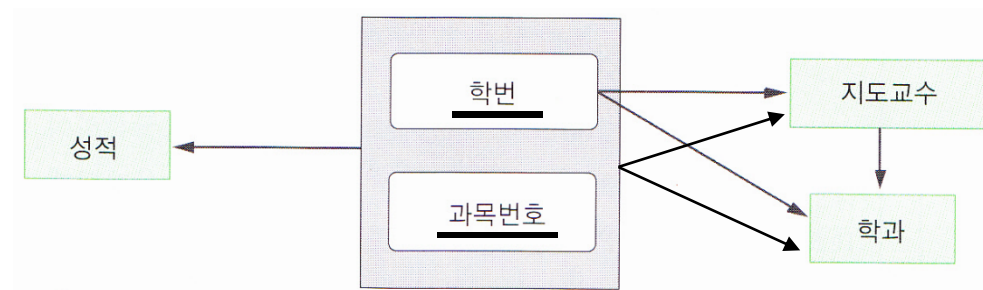
### □ 정의

- 릴레이션 R에서 모든 함수 종속의 결정자는 후보키 (기본키 포함)이다.

### □ 강한 제3정규형(strong 3NF)

- 릴레이션 R이 BCNF에 속하면 R은 제1, 제2, 제3 정규형에 속함. 그러나 역은 성립하지 않음.

## □ BCNF / 3NF 아님: 수강지도, 지도



## □ 예(3NF)

### - 수강과목 (학번, 과목, 교수)

#### ◆ 제약조건

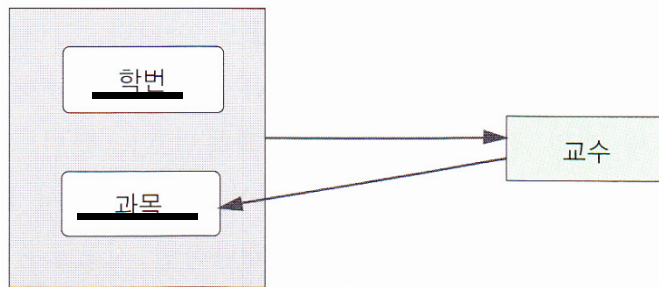
- 각 과목에 대해, 한 학생은 오직 한 교수의 강의만 수강한다.
- 각 교수는 한 과목만 담당한다.
- 한 과목은 여러 교수가 담당할 수 있다.

#### ◆ 후보키 : {학번, 과목}, {학번, 교수} → 후보키가 중첩됨 !!

#### ◆ 기본키 : {학번, 과목}

#### ◆ FD : {학번, 과목} → 교수 교수 → 과목

#### ◆ “수강과목” 릴레이션에서 결정자인 {교수}가 후보키로 취급되지 않으므로, 이 릴레이션은 BCNF이 아님



수강과목

학번	과목	교수
100	프로그래밍	P1
100	자료구조	P2
200	프로그래밍	P1
200	자료구조	P3
300	자료구조	P3
300	프로그래밍	P4

---

## □ 3NF(수강과목 릴레이션)에서의 이상

### ① 삽입이상

- ◆ 교수 P5가 자료구조를 담당한다는 사실의 삽입은 수강 학생이 있어야 가능

### ② 삭제이상

- ◆ 100번 학생이 자료구조를 취소하여 투플을 삭제하면 P2가 담당교수라는 정보도 삭제됨

### ③ 갱신이상

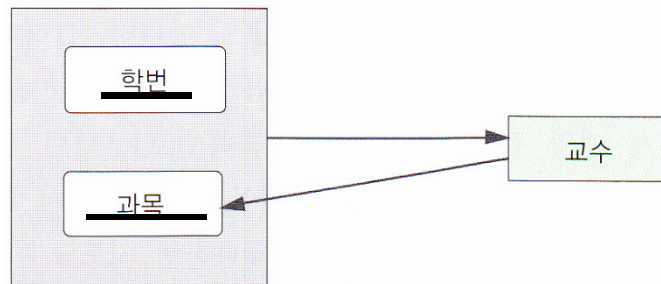
- ◆ P1이 프로그래밍 대신 자료구조를 담당하게 되면 P1이 나타난 모든 투플을 변경하여야 함

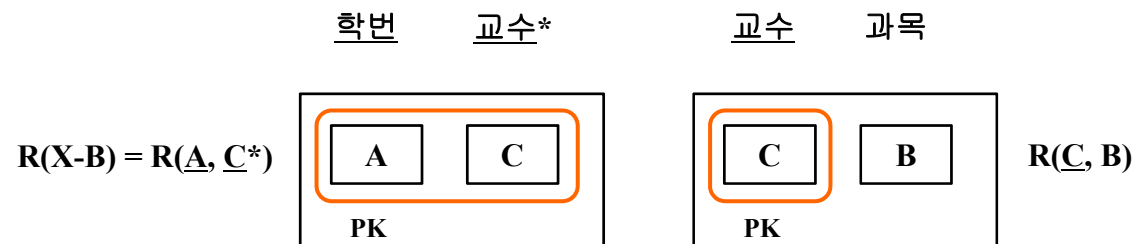
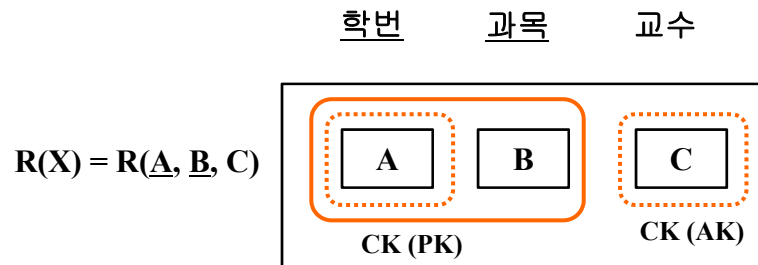
## □ 3NF 이상의 원인

- 결정자인 “교수”가 후보키(기본키)가 아님.

## □ 3NF 이상 원인의 해결

- 가정 (기본키와 후보키의 중첩)
  - ◆ B : 기본키 중 후보키에 포함되지 않은 부분
  - ◆ C : 후보키 중 기본키에 포함되지 않은 부분
  - ◆ 예에서,  $X = \{\text{학번}, \text{과목}, \text{교수}\}$ ,  $B = \{\text{과목}\}$ ,  $C = \{\text{교수}\}$
- 3NF 릴레이션  $R(X)$ 을 다음과 같이  $R_1$  및  $R_2$ 로 분해
  - ◆  $R_1(\underline{X-B})$ , 예에서  $\{\underline{\text{학번}}, \underline{\text{교수}}^*\}$
  - ◆  $R_2(\underline{C}, B)$ , 예에서  $\{\underline{\text{교수}}, \text{과목}\}$





수강과목(학번, 과목, 교수)

PK = {학번, 과목}

AK = {학번, 교수}

{학번, 과목} → 교수

교수 → 과목



수강교수(학번, 교수\*)

과목교수(교수, 과목)

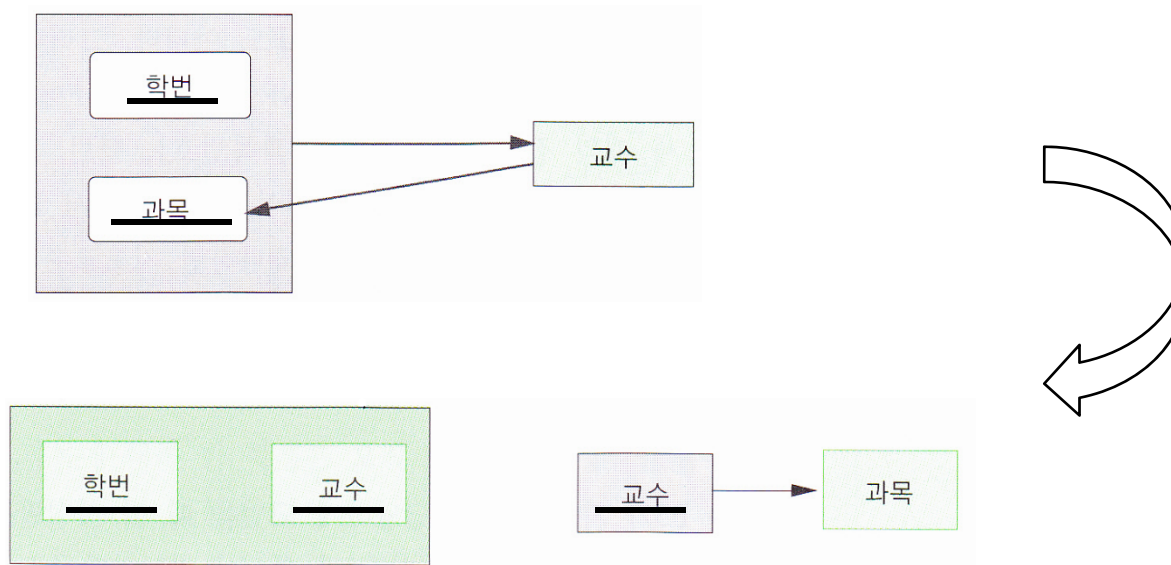
□ 예(BCNF) : 수강과목  $\Rightarrow$  수강교수, 과목교수

- 수강교수 (학번, 교수\*)

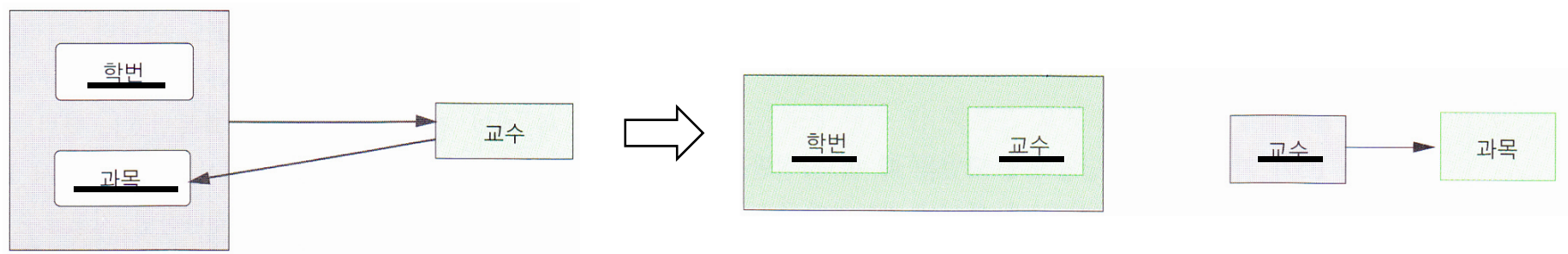
◆ FD : none

- 과목교수 (교수, 과목)

◆ FD : 교수  $\rightarrow$  과목

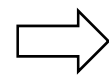






수강과목

학번	과목	교수
100	프로그래밍	P1
100	자료구조	P2
200	프로그래밍	P1
200	자료구조	P3
300	자료구조	P3
300	프로그래밍	P4



수강교수

학번	교수
100	P1
100	P2
200	P1
200	P3
300	P3
300	P4

과목교수

교수	과목
P1	프로그래밍
P2	자료구조
P3	자료구조
P4	프로그래밍

## □ 무손실 분해 (3NF $\Rightarrow$ BCNF)

- 수강교수 =  $\Pi_{\text{학번}, \text{교수}}(\text{수강과목})$   
과목교수 =  $\Pi_{\text{교수}, \text{과목}}(\text{수강과목})$
- 수강과목 = 수강교수  $\bowtie$  과목교수

수강과목

학번	과목	교수
100	프로그래밍	P1
100	자료구조	P2
200	프로그래밍	P1
200	자료구조	P3
300	자료구조	P3
300	프로그래밍	P4

수강교수

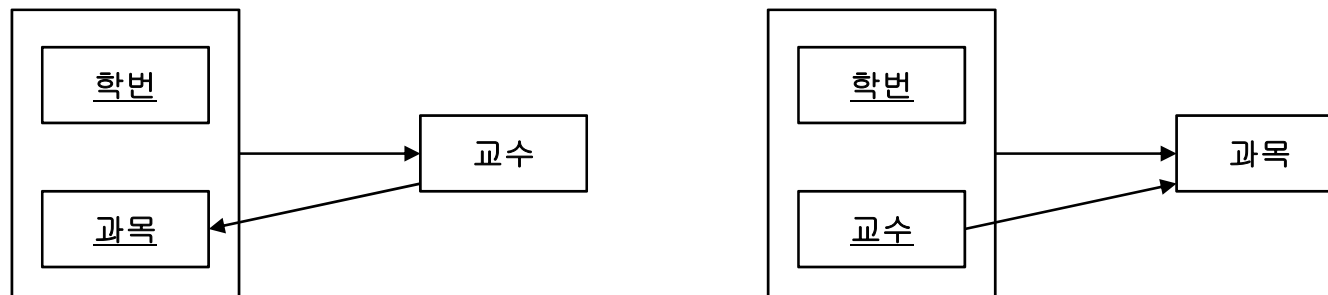
학번	교수
100	P1
100	P2
200	P1
200	P3
300	P3
300	P4

과목교수

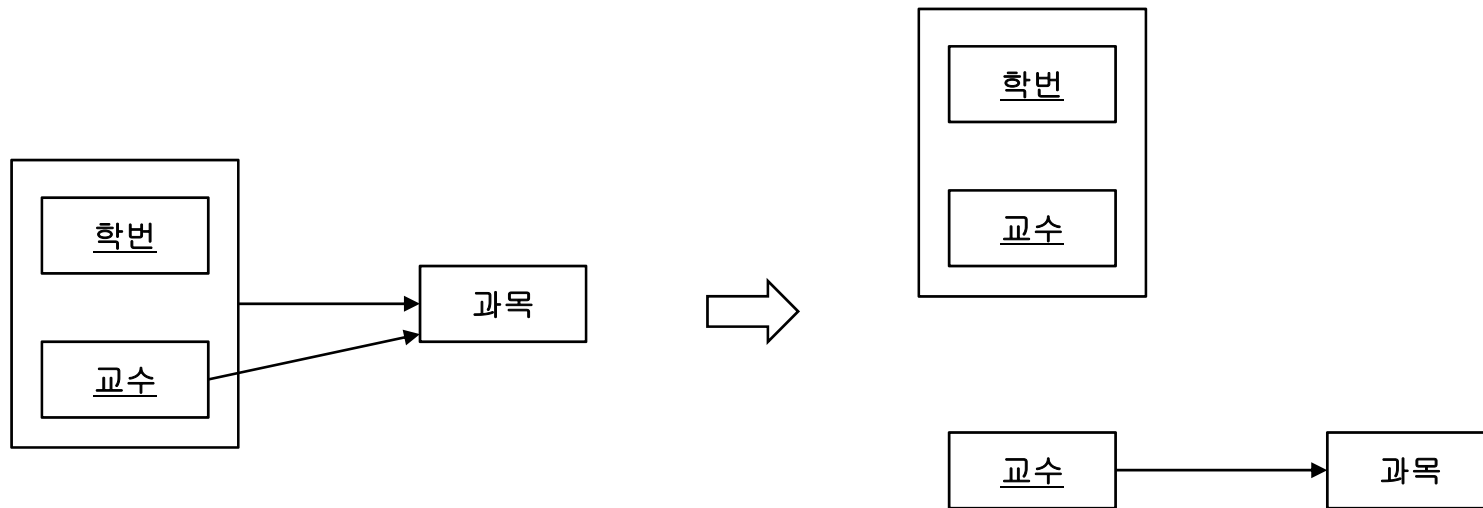
교수	과목
P1	프로그래밍
P2	자료구조
P3	자료구조
P4	프로그래밍

## □ 참고

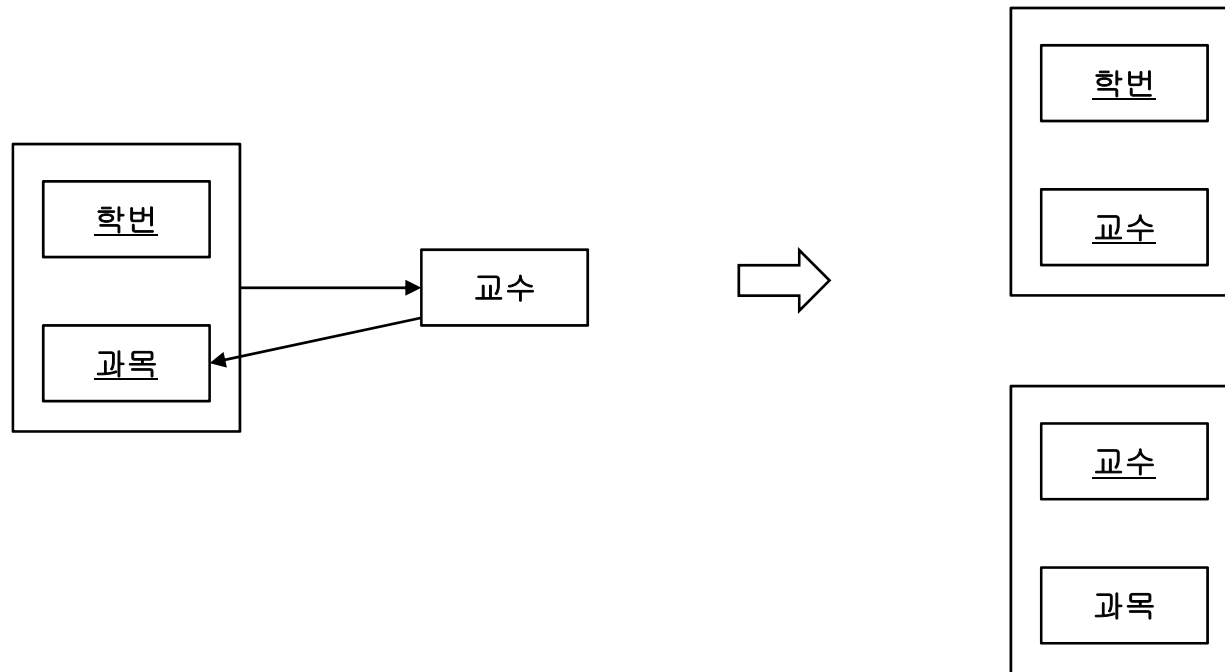
- 수강과목 (학번, 과목, 교수)에서
  - ◆ 제약조건
    - 각 과목에 대해, 한 학생은 오직 한 교수의 강의만 수강한다.
    - 각 교수는 한 과목만 담당한다.
    - 한 과목은 여러 교수가 담당할 수 있다.
  - ◆ 후보키 : {학번, 과목}, {학번, 교수} → 후보키가 중첩됨 !!
  - ◆ FD : {학번, 교수} → 과목  
교수 → 과목
  - ◆ 기본키 : {학번, 과목}으로 할 경우와 {학번, 교수}로 할 경우



- {학번, 교수}를 기본키로 할 경우
  - ◆ 과목이 기본키 {학번, 교수}에 부분함수종속 되므로, 2NF를 위반함. 따라서 2NF으로 변환



- {학번, 과목}을 기본키로 할 경우
  - ◆ 이 릴레이션은 2NF과 3NF을 만족
  - ◆ 그러나 두 후보키가 중첩되므로, BCNF을 위반함. 따라서 BCNF으로 변환.



## ❖ 고급 정규형

### 교과목 목록

과목(C)	교수(P)	교재(T)	← 비정규형 (Repeating Group)
화일처리	$\begin{Bmatrix} P1 \\ P2 \end{Bmatrix}$	$\begin{Bmatrix} T1 \\ T2 \end{Bmatrix}$	
데이터베이스	P3	$\begin{Bmatrix} T3 \\ T4 \\ T5 \end{Bmatrix}$	

### 개설 교과목

과목(C)	교수(P)	교재(T)	← BCNF ∴ (키에 속하지 않는 결정자 에트리뷰트가 없음)
화일처리	P1	T1	
화일처리	P1	T2	
화일처리	P2	T1	
화일처리	P2	T2	
데이터베이스	P3	T3	기본키: (과목, 교수, 교재)
데이터베이스	P3	T4	
데이터베이스	P3	T5	

---

## □ 개설교과목에서의 변경 이상

- P4가 데이터베이스를 담당한다는 정보삽입 시 3개의 교재에 대한 튜플을 삽입해야 함

## □ BCNF 이상의 원인

- 즉, 과목은 교수나 교재의 값 하나를 결정하는 것이 아니라 값의 집합(set of values)을 결정

과목  $\rightarrow$  교수 | 교재

(화일처리) { P1, P2 }

(화일처리) { T1, T2 }

## □ 다치 종속 (MVD, Multi-Valued Dependency)

– 정의:

릴레이션  $R(A,B,C)$ 에서 어떤  $(A, C)$ 값에 대응하는  $B$ 값의  
집합이  $A$ 값에만 종속되고  $C$ 값에 독립이면

다치 종속  $A \twoheadrightarrow B$ 가 성립  $(A,C) \quad \{ B \} \quad A \quad \{ B \}$

–  $A \twoheadrightarrow B$ 이면  $A \rightarrow C$ 도 성립 즉,  $A \rightarrow B|C$

– 모든 FD는 MVD이나, 역은 성립하지 않음  
즉,  $A \rightarrow B$ 이면  $A \twoheadrightarrow B$ 가 성립

## □ MVD를 가진 릴레이션의 분해

–  $R(A,B,C)$ 에서 MVD  $A \twoheadrightarrow B|C$ 이면  $R_1(A,B)$ 와  $R_2(A,C)$ 로  
무손실 분해 가능



## ▶ 제4정규형 (4NF)

---

### □ 정의

- 릴레이션 R에서 MVD  $A \twoheadrightarrow B$ 가 존재할 때 R의 모든 애트리뷰트들이 A에 함수 종속(FD)이면 R은 4NF  
(즉 R의 모든 애트리뷰트 X에 대해  $A \twoheadrightarrow X$  이고 A가 후보키)

### □ BCNF를 이용한 정의

- 릴레이션 R이 BCNF에 속하고 모든 MVD가 FD이면 R은 4NF

### □ 의미

- 어떤 릴레이션 R이 4NF이라면 MVD가 없거나,  $A \twoheadrightarrow B|C$ 가 있을 경우 A에 대응되는 B와 C의 값은 하나씩이어야 하며 이때 A는 후보키이어야 함

## ▶ 제4정규형(2)

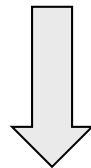
### □ 예      개설 교과목

과목(C)	교수(P)	교재(T)
화일처리	P1	T1
화일처리	P1	T2
화일처리	P2	T1
화일처리	P2	T2
데이터베이스	P3	T3
데이터베이스	P3	T4
데이터베이스	P3	T5

⇐ BCNF

∴ (키에 속하지 않는 결정자  
애트리뷰트가 없음)

기본키: (과목, 교수, 교재)  
MVD 과목 →→ 교수|교재



#### 교과목 교수

과목(C)	교수(P)
화일처리	P1
화일처리	P2
데이터베이스	P3

#### 교과목교재

과목(C)	교재(T)
화일처리	T1
화일처리	T2
데이터베이스	T3
데이터베이스	T4
데이터베이스	T5



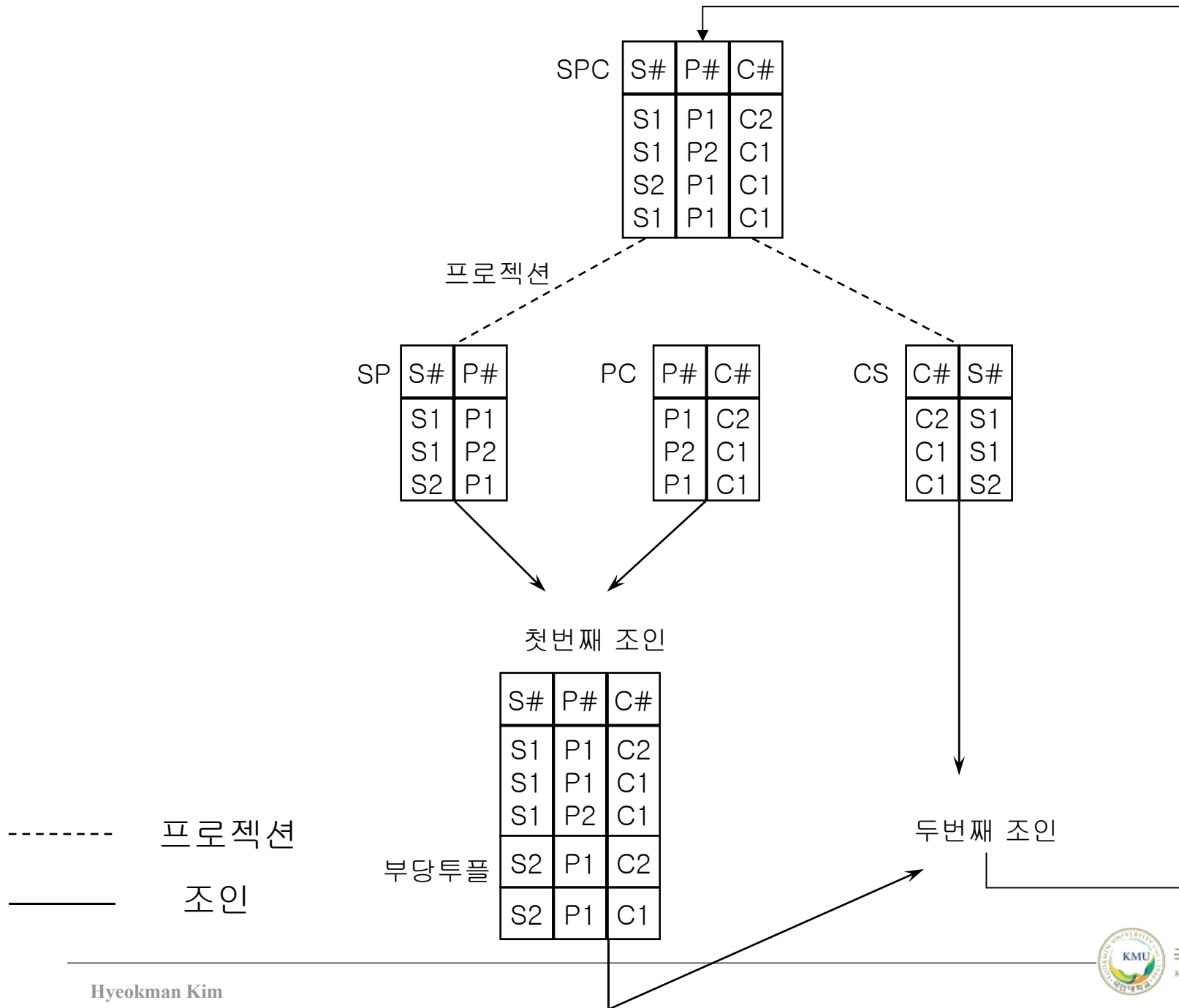
국민대학교  
KOOKMIN UNIVERSITY

## ▶ 제5정규형(5NF)

---

### □ 예 : 릴레이션 SPC(4NF)

- SPC를 프로젝션하여 SP,PC,CS를 생성
- SP,PC,CS를 조인하여 SPC의 재생성이 가능하나 그 어느 두 개의 조인만으로는 재생성 불가능



## ▶ 제5정규형(2)

### □ 3-분해 릴레이션

- 릴레이션 SPC가 세 개의 프로젝션 SP, PC, CS의 조인과 동등하다는 것은

(s1,p1)    SP  
(p1,c1)    PC    (s1,p1,c1)    SPC  
(c1,s1)    CS

- 즉 다음의 순환적 제약조건(3D)을 만족

(s1,p1,c2)    SPC  
(s2,p1,c1)    SPC    (s1,p1,c1)    SPC  
(s1,p2,c1)    SPC

- SPC : 3-분해 릴레이션  
          : 3D 제약조건을 만족

## ▶ 제5정규형(3)

---

### □ $n$ -분해 릴레이션( $n > 2$ )

–  $n$ 개의 프로젝션으로만 무손실 분해될 수 있으며

$m(m < n)$ 개의 프로젝션으로는 무손실 분해가 불가능한  
릴레이션

## ▶ 제5정규형(4)

---

### □ 조인 종속(JD, Join Dependency)

- 릴레이션 R이 그의 프로젝션 A, B, ..., Z의 조인과 동일하면 R은 JD  $\ast(A, B, \dots, Z)$ 을 만족.

이때 A,B,...,Z는 R의 애트리뷰트들에 대한 부분 집합.

- 릴레이션 R(A,B,C)가 JD  $\ast(AB,AC)$ 을 만족하면, 한쌍의 MVD A B|C도 성립 $\Rightarrow$
- JD는 MVD의 일반형, MVD는 JD의 특별한 경우(2-분해)
- SPC 릴레이션은
  - ◆ JD  $\ast(SP, PC, CS)$ 를 만족
  - ◆ 3-분해 릴레이션
- JD를 만족하는 n-분해 릴레이션은 n개의 프로젝션으로 분해해야 함.

## ▶ 제5정규형(5)

---

### □ 릴레이션에서의 갱신이상

#### – ① 삽입이상

- ◆ 릴레이션 SPC'에서 (S2,P1,C1)의 삽입시 (S1,P1,C1)의 삽입 필요
- ◆ 역은 성립 않음

SPC'

SK	PK	CK
S1	P1	C2
S1	P2	C1



## ▶ 제5정규형(6)

### □ 릴레이션의 갱신이상(con't)

#### ② 삭제이상

- ◆ 릴레이션 SPC에서 (S1,P1,C1)의 삭제시 다른 튜플 중 어느 하나를 함께 삭제하여야 함
- ◆ (S2,P1,C1)의 삭제는 이상 없이 가능

SPC	SK	PK	CK
	S1	P1	C2
	S1	P2	C1
	S2	P1	C2
	S1	P1	C1

이상의 원인 : SPC는 3-분해 릴레이션

이상의 해결 : 릴레이션 SPC를 3-분해함

## ▶ 제5정규형(7)

---

### □ 제5정규형(5NF)

#### – 정의

- ◆ 릴레이션 R에 존재하는 모든 조인 종속이 R의 후보키를 통해 성립되면, R은 5NF

#### – 프로젝션-조인 정규형(PJ/NF)

#### – 예1:

- ◆ SPC : JD  $*(SP, PC, CS)$ 는 후보키 (S,P,C)를 통하지 않으므로 5NF이 아님
- ◆ SP, PC, SC : 5NF

## ▶ 제5정규형(8)

---

### □ 제5정규형 (5NF) (con't)

– 예2:

학생(학번,이름,학과,학년) 릴레이션의 후보키가 학번과 이름일 경우

JD \*((학번,이름,학과), (학번,학년))

JD \*((학번,이름), (학번,학년), (이름,학과))

위의 JD는 모두 후보키를 통해 성립되므로 5NF

## ❖ 정규형들 간의 관계

### □ 정규화 과정 (무손실 분해)

비정규 릴레이션

원자값이 아닌  
도메인을 분해

1NF

부분 함수종속 제거

2NF

이행 함수 종속 제거

3NF

결정자가 후보키가  
아닌 함수종속(FD) 제거

BCNF

함수종속이 아닌  
다치종속(MVD) 제거

4NF

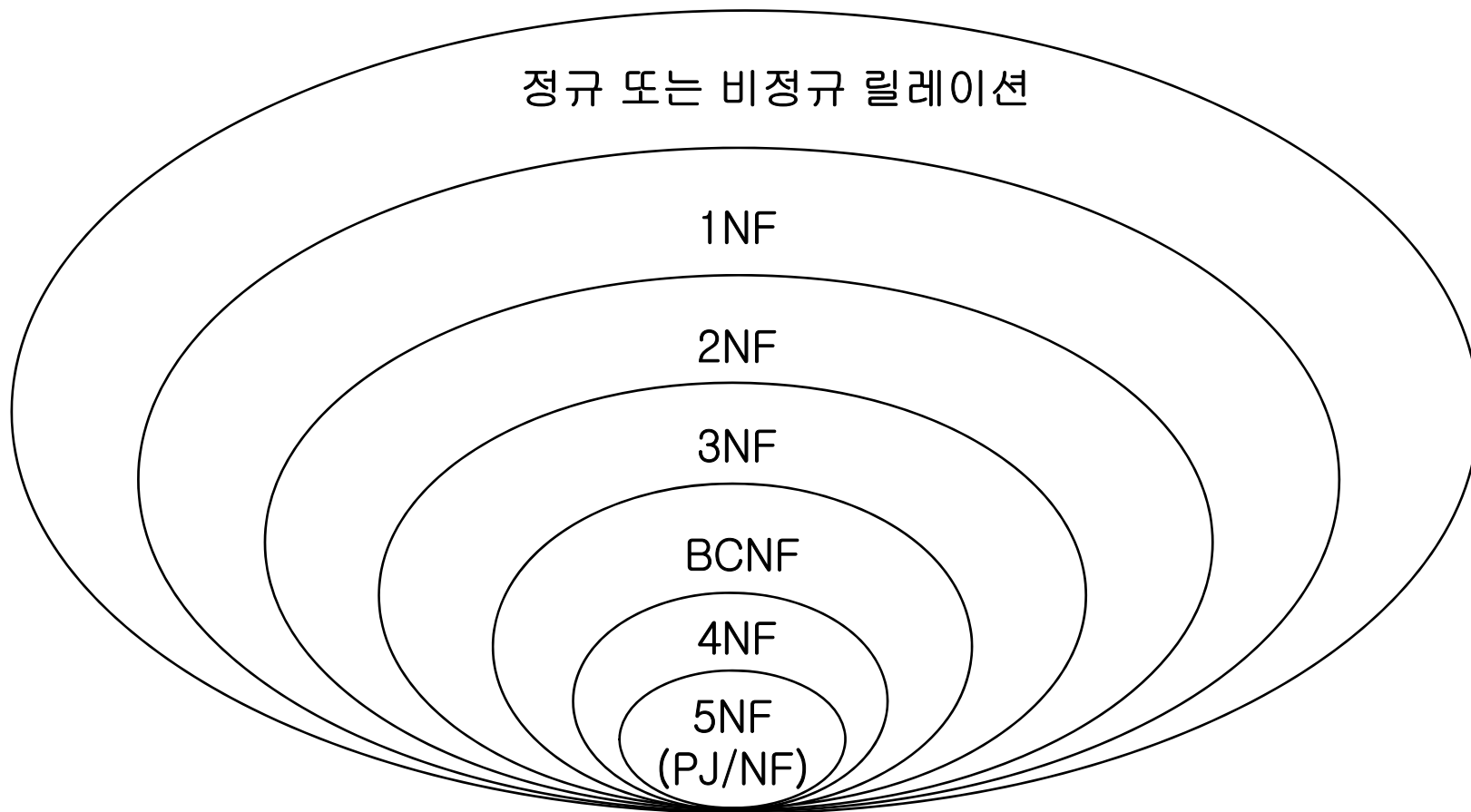
후보키를 통하지 않은  
조인종속(FD) 제거

5NF

## ❖ 정규형들 간의 관계(2)

---

### □ 정규형들 간의 포함 관계



## ❖ 정규형들 간의 관계(3)

---

### 👉 Note

- 릴레이션의 정규화는 실제 데이터 값이 아니라 개념적인 측면에서 다루어져야 함
- 실제 정규화 과정은 정규형의 순서와 다를 수 있음

## ❖ 정규형들 간의 관계(4)

---

### 👉 Note

- 모든 릴레이션을 반드시 5NF에 속하도록 분해할 필요는 없음
- 학생주소(학번,이름,주소,전화번호) : 5NF이 아님

FD : 전화번호 → 주소



학생전화(학번,이름,전화번호) : 5NF

전화주소(전화번호,주소) : 5NF

이름, 전화번호, 주소는 분리하지 않고 사용하는 것이 일반적이므로 위의 5NF으로의 분해는 무의미함