데이터베이스 및 설계

Chap 7. 데이터 종속성과 정규화

2012.05.21.

오 병 우

컴퓨터공학과

kit 국립금오공과대학교
Kumoh National Institute of Technology

데이터베이스 구축

- 데이터베이스 구축
  - DB 설계 필요: 전체 attribute를 relation별로 grouping 필요
- (Logical) database design
  - DDL을 가지고 conceptual schema 정의
  - Given some body of data
    → suitable logical structure
    - (what relations & what attributes in Relational Database)
  - Application independent design (what, not how)
  - Conceptual “schema” design
- Physical database design
  - Given logical structure → suitable physical structure

관계형 데이터베이스인 경우에만 해당

데이터의 논리적 표현

- 관계형 데이터베이스에서는 관계 스킴(relational scheme)의 설계에 해당함
  - 관계 모델을 이용하여 어떻게 실세계를 정확히 표현할 것인가?

  i. attribute, entity, relationship 파악 필요

  ii. 관련된 attribute들을 relation으로 묶음
    - 데이터 종속성 : attribute들간의 관계성
    - 효율적인 데이터 처리
    - 데이터의 일관성 유지

  iii. 변칙적 성질의 예방
    - 데이터의 중복성 배제
    - 이상(anomaly)

이상 (anomaly)

- example : 수강 relation
  - High degree of redundancy → problems

일부러 만든 잘못된 예제. 학년을 수강에서만 가지고 있다고 가정

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	E412	C	4
500	C312	B	2

Primary key: 학번, 과목 번호

이상(2)

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

방아쇠, 작동시키다

- 삽입 이상 (insertion anomaly)
  - 600번 학생이 2학년이라는 사실을 삽입
    ⇒ 어떤 과목을 등록하지 않는 한 삽입이 불가능
    (∵ 과목 번호가 Primary key)
  - 원하지 않는 정보의 강제 삽입
- 갱신 이상 (update anomaly)
  - 400번 학생의 학년을 4에서 3으로 변경
    ⇒ 학번이 400인 4개의 투플 모두를 갱신시켜야 함
  - 중복데이터의 일부 갱신으로 정보의 모순성(inconsistency) 발생

이상의 원인과 해결책: Normalization

- 이상의 원인
  - 상이한 종류의 정보를 하나의 릴레이션으로 표현하려 하기 때문
  - Attribute들 간에 존재하는 여러 종속관계를 하나의 relation에 표현
- 이상의 해결
  - Attribute들 간의 종속관계를 분석하여 여러개의 relation으로 분해(decomposition)하는 과정 필요

    ⇒ 정규화(normalization)

- Normalization theory
  - Allow us to recognize certain undesirable properties and show how such relations can be converted to a more desirable form

스키마 설계와 변환

- 스키마 설계 : 데이터베이스의 논리적 설계
  - ① attribute들과 이들의 제약 조건 (종속성)들을 수집
  - ② 수집된 결과를 명시된 제약 조건에 따라 여러 개의 relation으로 분할
  - ⇒ 스키마 변환 (schema transformation)
- 스키마 변환의 원리
  - ① 정보의 무손실
  - ② 데이터의 중복성 감소
  - ③ 분리의 원칙
- Normalization procedure
  - The successive reduction of a given collection of relations to some more desirable form
  - 1NF → 2NF → 3NF → BCNF → 4NF → 5NF

연속적인

Normalization

- 정규형(Normal Form)
  - 어떤 일련의 제약 조건을 만족하는 relation

- 정규화(Normalization)의 원칙

  정규화 = 스키마 변환 (S → S')

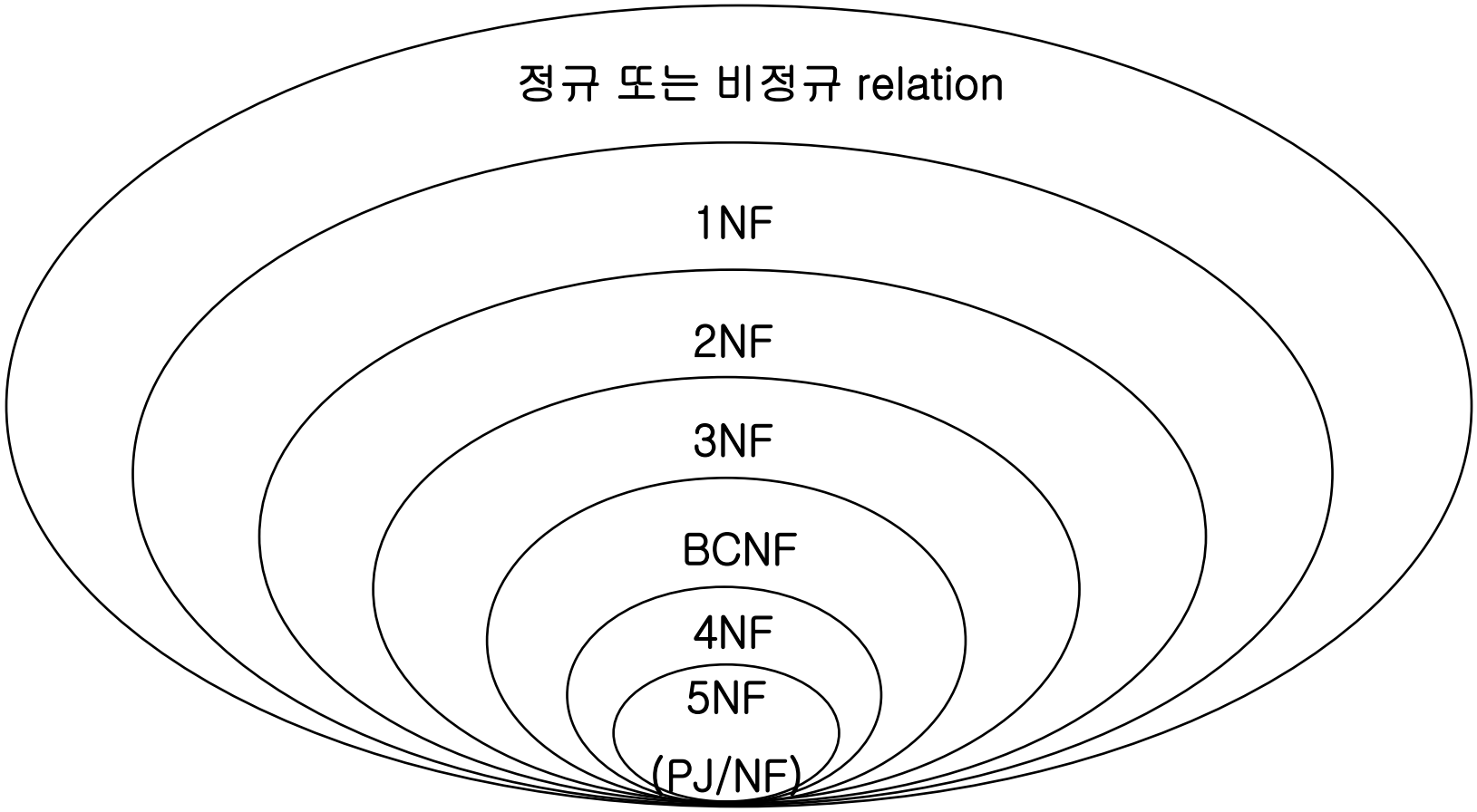
  ① 무손실 표현
  - 같은 의미의 정보 유지
  - 그러나 더 바람직한 구조

  ② 데이터의 중복성 감소

  ③ 분리의 원칙
  - 독립적인 관계는 별개의 relation으로 표현
  - relation 각각에 대해 독립적 처리가 가능

Normal Form

- 정규형들 간의 포함 관계



함수 종속 (FD)

- Functional dependence (or functional dependency): FD

  R(U): a relation R

  X, Y ⊆ U : attribute X, Y

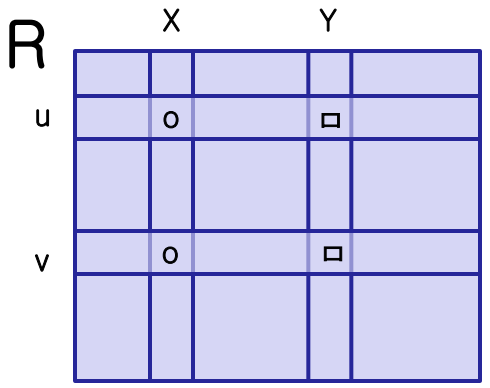
  R.X → R.Y: Y is functional dependent on X

  (or X functionally determines Y)

  Each X-value in R has associated with it precisely on Y-value in R

  (i.e., ∀u, v ∈ R, u[X] = v[X] → u[Y] = v[Y])

  - 어떤 relation R에서, attribute X의 값 각각에 대해 attribute Y의 값이 하나만 연관
    - attribute Y는 attribute X에 함수 종속 X → Y
  - attribute X는 Y를 (함수적으로) 결정
    - X를 결정자(determinant)
    - Y를 종속자(dependent)
  - X, Y는 복합 attribute일 수 있음

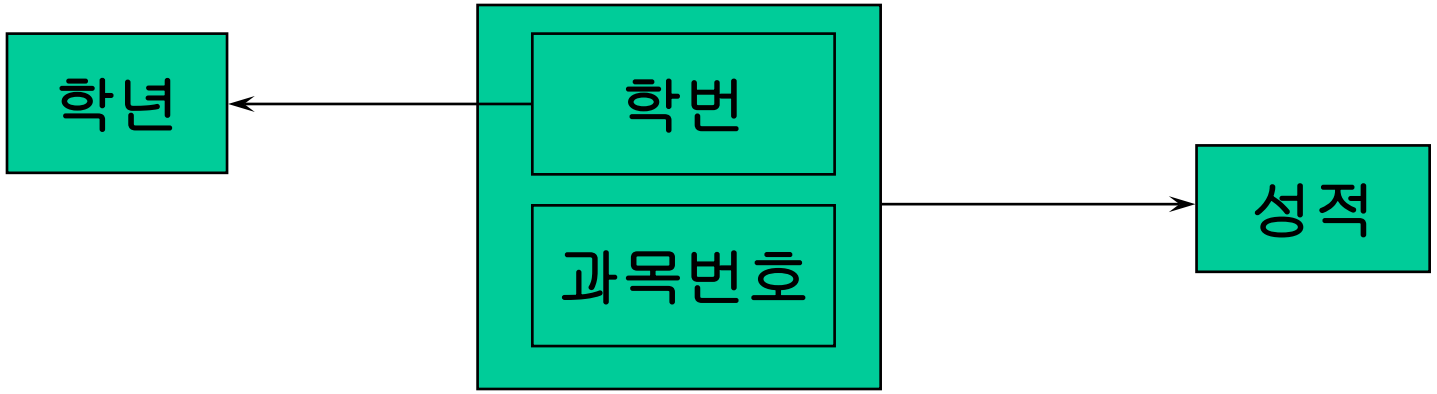


함수 종속(2)

- 후보키인 경우 정의와 일치함
  - relation R에서 attribute X가 후보키면,
    - R의 모든 attribute Y에 대해 $X \rightarrow Y$ 성립
  - 후보키(Candidate Key)의 정의
    - 유일성(Uniqueness): 모두 상이하고 유일함
    - 최소성(Minimality): 꼭 필요한 애트리뷰트들로만 구성
- FD에서는 후보키만을 다루는 것이 아님
  - 함수종속 $X \rightarrow Y$의 경우
    - attribute X가 반드시 키(유일 값)라는 것을 요건으로 하지 않음
    - 즉, attribute X값들이 하나 이상의 투플에서 같을 수 있음
    - 다만, X의 값이 같다면 Y의 값도 같음을 의미함

FD Diagram

- 수강 relation (Primary key: 학번, 과목번호)



{학번, 과목번호} → 성적

학번 → 학년

Full Functional Dependency

- Full functional dependency(default) ↔ partial FD
  - R.X → R.Y: Y is fully functional dependent on X
  - not∃ Z ⊂ X such that R.Z → R.Y
    - 복합 attribute X에 대하여 X → Y가 성립할 때 Z가 존재하지 않음
- 부분 함수 종속 (partial functional dependency)
  - Z가 존재함
- 예제 (수강 relation)
  - 학번 → 학년
  - {학번, 과목번호} → 성적
  - (학년)은 (학번)에 완전 함수 종속
    - 그러나 {학번,과목번호}에는 부분 함수 종속
  - (성적)은 {학번,과목번호}에 완전 함수 종속

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	E412	C	4
500	C312	B	2

추론 규칙

- 함수 종속에 대해 추론 규칙이 성립함

  R1: (반사, reflexive) A ⊇ B이면 A → B이다. 또한 A → A이다

  R2: (첨가, augmentation) A → B이면 AC → BC이고 AC → B이다.

  R3: (이행, transitive) A → B이고 B → C이면 A → C이다.

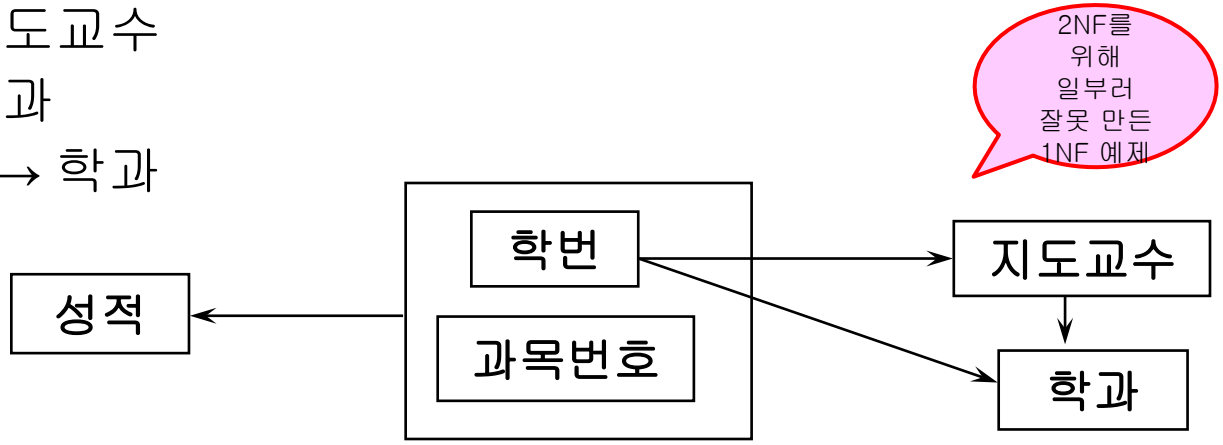
  R4: (분해, decomposition) A → BC이면 A → B이다.

  R5: (결합, union) A → B이고 A → C이면 A → BC이다.

- 함수 종속은 데이터의 의미(data semantics) 표현
  - 데이터베이스가 현실 세계를 표현할 때 적용해야 할 의미적 제약 조건
    - 예: "학번 → 학년"의 의미는 "학생은 하나의 학년에만 속한다"
      - 학번이 하나인데 학년이 두 가지 값이라면 현실 세계를 잘못 표현한 것이 됨

1NF: First Normal Form

- A relation: first normal form (1NF)
  - All underlying simple domains contain atomic values only, any normalized relation is in 1NF
  - 모든 도메인이 원자값(atomic value)만으로 된 relation
- 예 : 수강지도 relation
  - 수강지도 (학번,지도교수,학과,과목번호,성적)
  - Primary key: {학번,과목번호}
  - FD : {학번,과목번호} → 성적
    학번 → 지도교수
    학번 → 학과
    지도교수 → 학과



1NF 예제

1NF
설명 끝

수강 지도

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

2NF를 위한 Anomalies in 1NF

- 1NF에서의 이상
  - ① 삽입이상
    - 500번 학생의 지도교수가 P4라는 사실의 삽입은 어떤 교과목을 등록하지 않는 한 삽입 불가능
  - ② 삭제이상
    - 200번 학생이 C123의 등록을 취소하여 이 투플을 삭제할 경우 지도교수가 P2라는 정보까지 손실됨
  - ③ 갱신이상
    - 400번 학생의 지도교수를 P1에서 P3로 변경할 경우 학번이 400인 4개 투플의 지도교수 값을 모두 P3로 변경해야 함

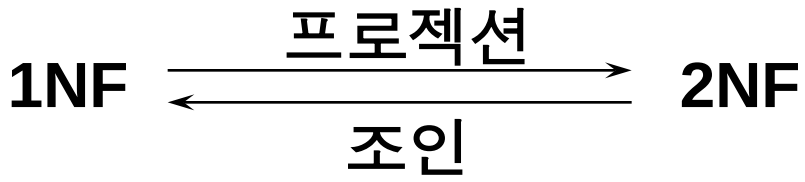
Solution by Suitable Projections

- 1NF 이상(anomaly)의 원인
  - Primary key에 partial FD(부분 함수 종속)된 attribute가 존재
  - → Primary key로 식별되는 개체와 무관한 attribute가 존재
  - → 두 가지 상이한 정보가 포함

- 1NF 이상(anomaly)의 해결
  - Projection으로 relation을 decomposition(분해)
    - Partial FD (부분 함수 종속) 제거
  - ⇒ 2NF

2NF: Second Normal Form

- A relation: second normal form (2NF)
  - 1) 1NF
  - 2) Every non key attributes is fully dependent on the primary key
    - Non key attribute: any attribute that does not participate in the (primary) key
  - 1NF이고, 키에 속하지 않는 attribute들은 모두 Primary key에 full FD (완전 함수 종속)
  - A relation: 1NF, not 2NF → collection of 2NF relations (suitable projections)
- Nonloss decomposition (무손실 분해)



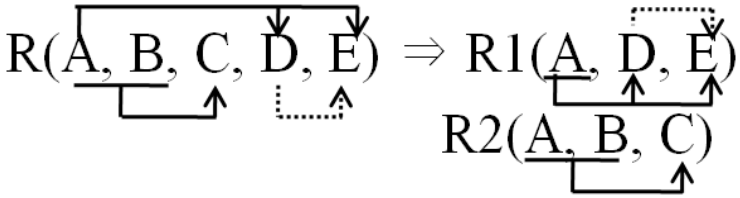
  - Original relation: recovered by taking the natural join (∴ no information is lost)
  - 프로젝션하여 분해된 relation들은 자연 조인을 통해 원래의 relation으로 복귀 가능
  - 원래의 relation에서 얻을 수 있는 정보는 분해된 relation들로부터도 얻을 수 있음 그러나, 그 역은 성립하지 않음

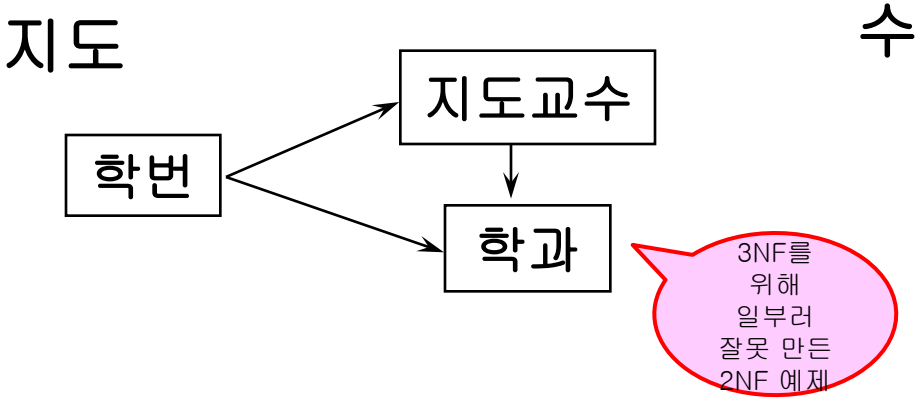
(삽입 이상의 예제에서 500번 학생의 지도교수가 P4라는 정보는 원래의 relation에서 표현할 수 없음)

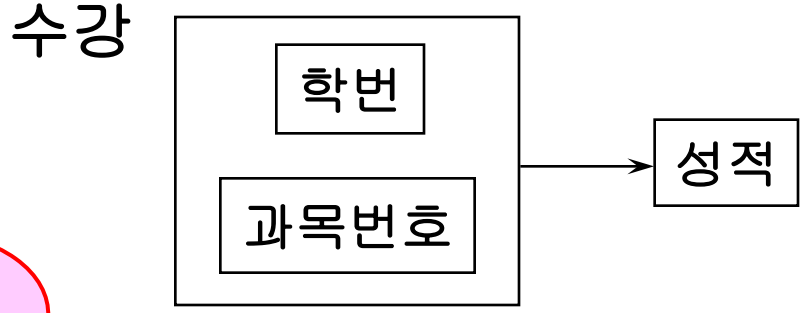
  - cf.) loss decomposition: original relation ⊂ natural join of those projections

2NF 예제

- 예 : 수강지도 ⇒ 지도, 수강 relation
  - 지도 (학번, 지도교수, 학과)
    Primary key : {학번}
  - 수강 (학번, 과목번호, 성적)
    Primary key : {학번, 과목번호}
    Foreign key : {학번} 참조 : 지도







2NF 예제

2NF
설명 끝

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

3NF를 위한 Anomalies in 2NF

- 2NF(지도 relation)에서의 이상 (anomaly)
  - ① 삽입이상
    - 어떤 지도교수가 특정 학과에 속한다는 사실의 삽입 불가능
    - 예제: 지도교수 P4가 수학과에 속한다
  - ② 삭제이상
    - 300번 학생의 투플을 삭제하면 지도교수 P3가 컴퓨터공학과에 속한다는 정보 손실
  - ③ 갱신이상
    - 지도교수 P1의 소속이 컴퓨터공학과에서 전자과로 변경된다면 학번이 100과 400번인 두개의 투플을 모두 변경하여야 함

TD: Transitive (Functional) Dependence

- 2NF 이상(anomaly)의 원인
  - Transitive FD (TD)가 존재
  - R(U): a relation
  - A, B, C ⊂ U
  - R.A → R.B, R.B → R.C ⇒ R.A → R.C
  - 이행적 함수 종속 (TD, Transitive Dependency)

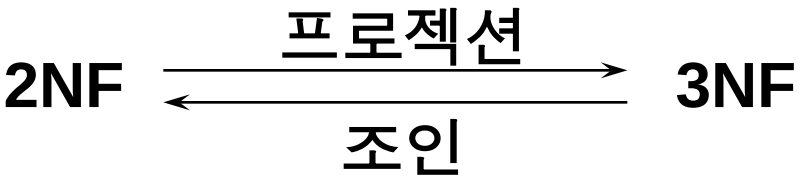
    A → B이고 B → C ⇒ A → C

    (즉, attribute C는 attribute A에 이행적 함수 종속)
- 2NF 이상(anomaly)의 해결
  - Projection으로 relation을 decomposition
    - TD 제거

  ⇒ 3NF

3NF: Third Normal Form

- A relation: third normal form(3NF)
  - 2NF
  - Every nonkey attribute is nontransitively dependent on the primary key
    - The non key attributes (if any) are (i) mutually independent, (ii) fully dependent on the primary key (not FD each other)
  - 2NF이고, 키가 아닌 모든 attribute들은 Primary key에 transitive FD(이행적 함수 종속)되지 않음
- Nonloss decomposition (무손실 분해)



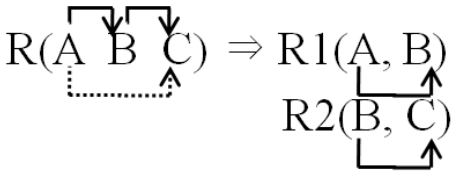
  - 원래의 relation에서 얻을 수 있는 정보는 분해된 relation들로부터도 얻을 수 있으나 그 역은 성립하지 않음

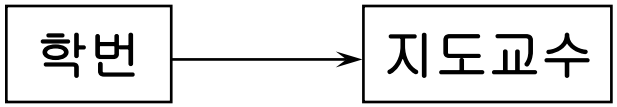
(삽입 이상 예제에서 지도교수 P4가 수학과에 속한다는 정보 표현)

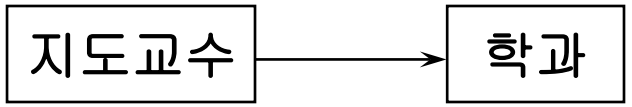
3NF 예제

3NF
설명 끝

- 예 : 지도 ⇒ 학생지도, 지도교수학과 relation
  - 학생지도 (학번, 지도교수)
    Primary key : {학번}
    Foreign key : {지도교수} 참조 : 지도교수학과
  - 지도교수학과 (지도교수, 학과)
    Primary key : {지도교수}







학생지도

학번	지도교수
100	P1
200	P2
300	P3
400	P1

지도교수학과

지도교수	학과
P1	컴퓨터
P2	전기
P3	컴퓨터

BCNF: Boyce/Codd Normal Form

- 3NF의 약점

  i . 복수의 후보키를 가지고 있고

  ii . 후보키들이 복합 attribute들로 구성되고

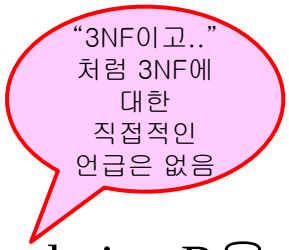
  iii . 후보키들이 서로 중첩되는 경우

  ⇒ 적용 불가능

  ⇒ 보다 일반적인 Boyce/Codd Normal Form(BCNF)을 제안

BCNF

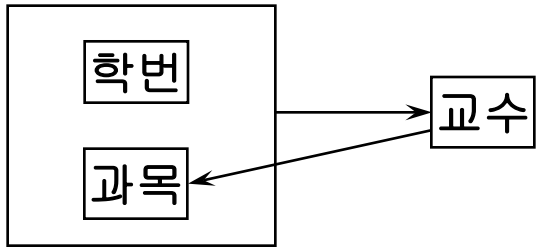
- A relation: Boyce/Codd normal form (BCNF)
  - Every determinant is a “candidate key”
    - Determinant
      - Any attribute on which some other attribute is (fully) functional dependent
  - Relation R의 모든 determinant(결정자)가 후보키이면 relation R은 BCNF에 속한다.



- Relation R이 BCNF에 속하면 R은 1NF, 2NF, 3NF에 속함
  - 강한 제3정규형(strong 3NF)이라고도 함
- 앞의 예제 Relation은 BCNF인가?
  - 각 Relation의 기본키가 모두 유일한 결정자인가?
  - YES: 수강, 학생지도, 지도교수학과
  - NO (3NF도 아님) : 수강지도, 지도

BCNF를 위한 3NF 예제

- 예(3NF) : 수강과목 relation
  - 제약 조건
    - 각 과목에 대한 한 학생은 오직 한 교수의 강의만 수강
    - **각 교수는 한 과목만 담당**
    - 한 과목은 여러 교수가 담당할 수 있음
  - 수강과목 (학번,과목,교수)
  - 후보키 : {학번,과목}, {학번,교수}
  - Primary key : {학번,과목}
  - 함수종속 : {학번,과목} → 교수
    교수 → 과목



BCNF를 위해 일부러 잘못 만든 3NF 예제

수강과목

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

BCNF를 위한 Anomalies in 3NF

- 3NF(수강과목 relation)에서의 이상
  - ① 삽입이상
    - 교수 P5가 자료구조를 담당한다는 사실의 삽입은 학번(수강 학생)이 있어야 가능
  - ② 삭제이상
    - 100번 학생이 자료구조를 취소하여 투플을 삭제하면 P2가 담당교수라는 정보도 삭제됨
  - ③ 갱신이상
    - P1이 프로그래밍 과목 대신 자료구조를 담당하게 되면 P1이 나타난 모든 투플을 변경하여야 함

⇒ 원인 : 교수가 결정자이지만 후보키가 아님

BCNF 예제

- 예(BCNF) : 수강과목 ⇒ 수강교수, 과목교수

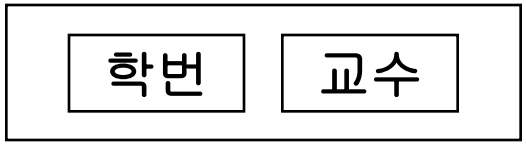
수강교수(학번, 교수)
Primary key : {학번, 교수}
Foreign key : {교수} 참조 : 과목교수
과목교수(교수, 과목)
Primary key : {교수}

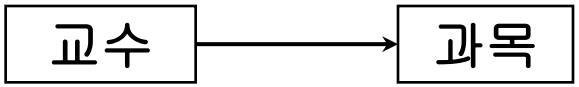
수강교수

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

과목교수

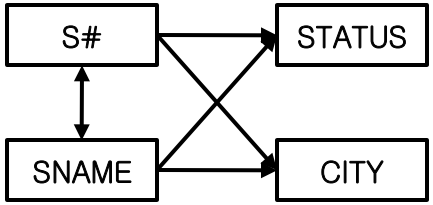
교수	과목
P1	프로그래밍
P2	자료구조
P3	자료구조
P4	프로그래밍





BCNF 예제

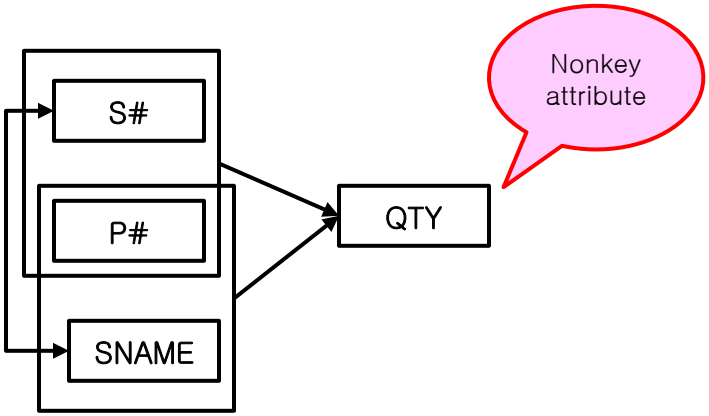
- S(S#, SNAME, STATUS, CITY)
  - S#, SNAME: two disjoint (nonoverlapping) candidate keys
    - Primary key: S#
    - Alternate key: SNAME
  - BCNF



Not BCNF 예제

BCNF 설명 끝

- SSP(S#, SNAME, P#, QTY)
  - (S#, P#), (SNAME, P#): two overlapping candidate keys
  - 3NF, but not BCNF (∵ S#, SNAME: determinants)



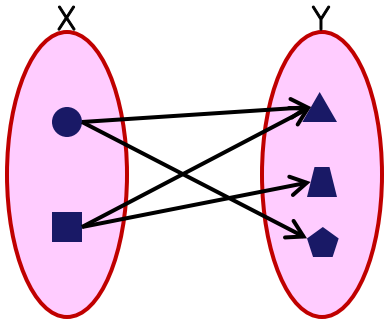
- BCNF
  - S(S#, SNAME), SP(S#, P#, QTY) or SP(P#, SNAME, QTY)
  - 앞의 예제를 고려하면 S(S#, SNAME, STATUS, CITY), SP(S#, P#, QTY) or SP(P#, SNAME, QTY)

Multi-valued Fact

과목 목록(UCPT)

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

⇐ 비정규형



Cartesian product

개설 과목(CPT)

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

⇐ BCNF

∵Primary key에 속하지 않는 determinant attribute가 없음

Primary key: (과목, 교수, 교재)

4NF를 위한 Anomalies in BCNF

- 개설과목에서의 변경 이상
  - P4가 데이터베이스를 담당한다는 정보삽입 시 3개의 교재에 대한 투플을 삽입해야 함
    - ∵ Primary key에 null을 넣을 수 없으므로…
    - ∵ Cartesian product 이므로…
      - 모두 primary key로 하지 않고 null을 넣는 방법도 있으나 교재에선 Cartesian product 방법 사용

- BCNF 이상의 원인
  - 즉, 과목은 교수나 교재의 값 하나를 결정하는 것이 아니라 몇 개의 값, 즉 multi-value(set of values)를 결정

    과목 ↠ 교수|교재 (or 과목 →→ 교수|교재)

    (화일구조) → { P1, P2 }

    (화일구조) → { T1, T2 }

MVD: Multi-valued Dependency

애트리뷰트가 적어도 3개 이상

- 다치 (다중값) 종속 (MVD, Multi-Valued Dependency)
  - Relation R(A,B,C)에서 어떤 (A, C)값에 대응하는 B값의 집합이 A값에만 종속되고 C값에 독립이면 다치 종속 $A \twoheadrightarrow B$가 성립
    - $(A,C) \rightarrow \{ B \} \equiv A \twoheadrightarrow \{ B \}$
  - A multi-determines B
  - B is multi-dependent on A
  - $A \twoheadrightarrow B$이면 $A \twoheadrightarrow C$도 성립 즉, $A \twoheadrightarrow B|C$

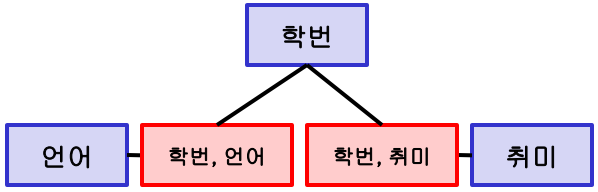
학번	언어	취미
100	C++, PHP	게임, 사진
200	Java, C#	사진, 트위터

- 모든 FD는 MVD이나, 역은 성립하지 않음

  - MVD: A generalization of the functional dependence
  - 즉, $A \rightarrow B$이면 $A \twoheadrightarrow B$가 성립 (MVD는 여러 개, FD는 딱 한 개만)
- MVD를 가진 relation의 분해(Fagin의 정리, 1977년)
  - R(A,B,C)에서 MVD $A \twoheadrightarrow B|C$이면 R1(A,B)와 R2(A,C)로 nonloss decomposition (무손실 분해) 가능

4NF: Fourth Normal Form

- A relation : fourth normal form (4NF)
  - i) BCNF
  - ii) All MVDs in R are in fact FDs
  - Relation R이 BCNF에 속하고 **모든 MVD가 FD**이면 R은 4NF
    - MVD가 없으면 4NF
- Relation R에서 MVD A ↠ B가 존재할 때 R의 모든 attribute들이 A에 함수 종속(FD)이면 R은 4NF
  - 즉 R의 모든 attribute X에 대해 A → X 이고 A가 후보키
- 의미
  - 어떤 relation R이 4NF이라면 MVD가 없거나,
  - MVD A ↠ B|C가 있을 경우
    - A에 대응되는 B와 C의 값은 하나씩 이어야 하며
    - 이때 A는 후보키라는것을 의미한다.



4NF 예제

- 예제

개설 교과목

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

⇐ BCNF

∵(key에 속하지 않는 determinant attribute가 없음)

Primary key: (과목, 교수, 교재)

MVD 과목 ↠ 교수|교재

과목 교수

과목(C)	교수(P)
화일구조	P1
화일구조	P2
데이터베이스	P3

과목교재

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

⇐ 4NF

참고

4NF
설명 끝

- Rissanen's work
  - R(A, B, C) : a relation
  - A ↠ B, B ↠ C (or A → B, B → C)
  - R1(A, B) , R2(B, C) : independent
    (R1(A, B) , R2(A, C) : not independent)
    - (A, B)(A, C)보다 (A, B)(B, C)로 projection함이 바람직하다.

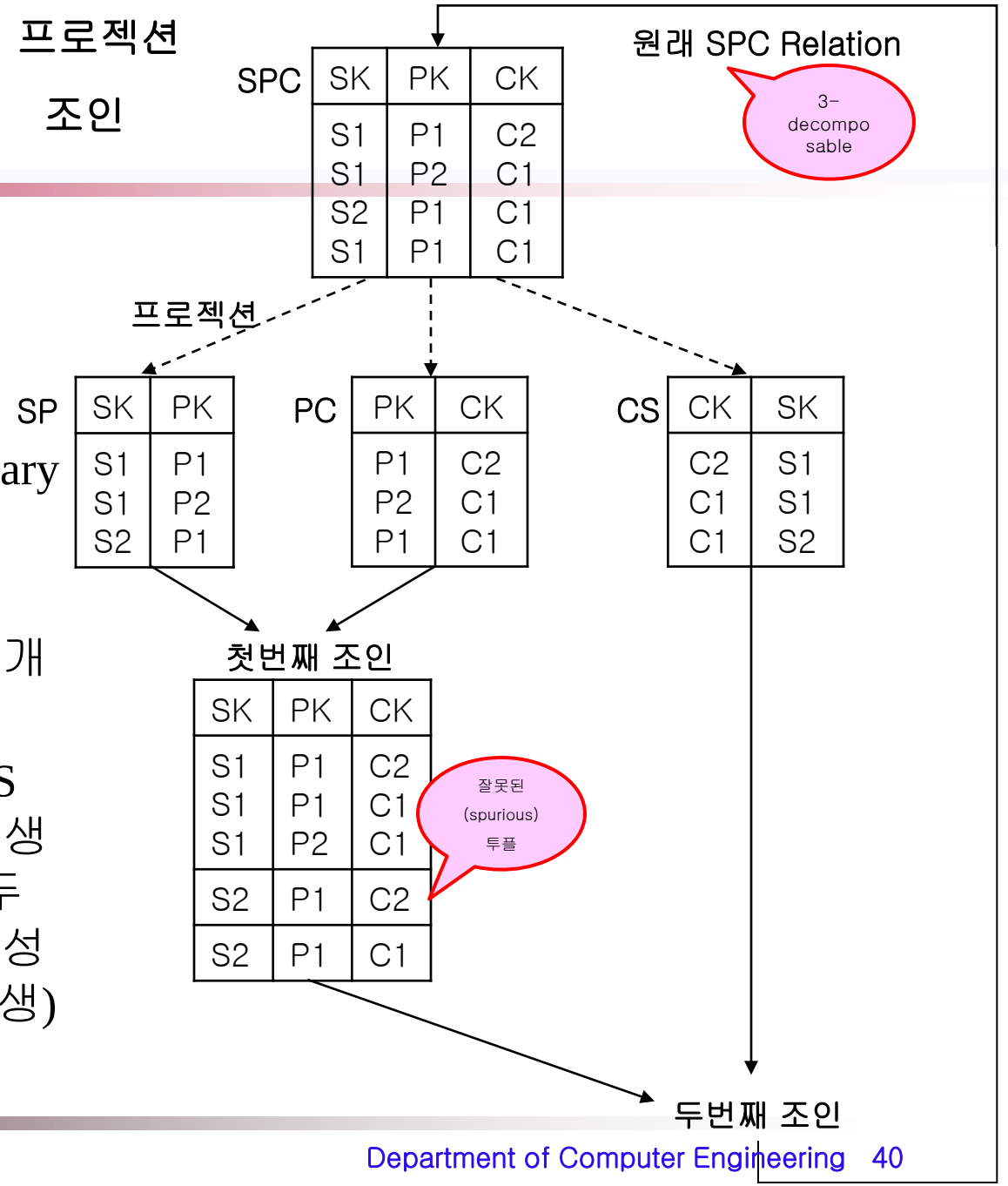
n–decomposable (n-분해) Relation

- Nonloss decomposition
  - A relation ⇒ two projections (1NF → … → 4NF)
    - 4NF까지는 2개의 relation으로 decomposition(분해)하면 문제 해결
    - 2-decomposable relation이라고 함
  - three or more projections (4NF → 5NF)
    - 5NF는 n개로 decomposition 필요
- A relation: n–decomposable (n > 2)
  - The relation can be nonloss-decomposed into n projections but not into m projections for any m < n
    - n개의 projection으로만 nonloss decomposition (무손실 분해)될 수 있으며 m (m<n)개의 projection으로는 nonloss decomposition이 불가능한 relation

프로젝션과 조인

-------- 프로젝션

———— 조인



- 예제 : Relation SPC
  - 4NF
    - 모든 attribute가 primary key에 속함
    - FD나 MVD 없음
  - SPC를 프로젝션하여 세 개의 SP,PC,CS를 생성
  - 세 개의 relation SP,PC,CS를 조인해서는 SPC의 재생성이 가능하나 그 어느 두 개의 조인만으로는 재생성 불가능(Spurious tuple 발생)

3–decomposable Relation 예제

- 3-decomposable relation
  - Relation SPC가 세 개의 프로젝션 SP,PC,CS의 조인과 동등하다는 것은 다음을 의미

$$\left.\begin{array}{l}(S1,P1) \in SP \\ (P1,C1) \in PC \\ (C1,S1) \in CS\end{array}\right\} \Rightarrow (S1,P1,C1) \in SPC$$

  - 즉 다음의 순환적 제약조건(3D: 3-decomposable)을 만족

$$\left.\begin{array}{l}(\underline{S1,P1},C2) \in SPC \\ (S2,\underline{P1,C1}) \in SPC \\ (\underline{S1},P2,\underline{C1}) \in SPC\end{array}\right\} \Rightarrow (S1,P1,C1) \in SPC$$

  - SPC : 3-decomposable relation
    : 3D 제약조건을 만족

3D 제약조건을 만족하는 것을 JD라고 함

JD: Join Dependency

- 조인 종속(JD, Join Dependency)
  - R(U) : a relation
  - X, Y, … Z ⊆ U (X, Y, …, Z는 R의 attribute들에 대한 subset)
  - *(X, Y, … , Z): R satisfies the join dependence (JD)
    - R = (natural) join of its projections on X, Y, … , Z
      - relation R이 그의 프로젝션 X, Y, …, Z의 join과 동일하면 R은 JD*(X, Y, … , Z) 만족
  - A generalization of FD and MVD (JD는 MVD의 일반형)
    - MVD는 JD의 특별한 경우 (2-분해)
- R(A,B,C)가 JD *(AB,AC)을 만족하면, 한 쌍의 MVD A ↠ B|C도 성립
  - R(A, B, C) satisfies the JD*(AB,AC) ↔
    R(A, B, C) satisfies the MVDs A ↠ B | C
    (∵ ↔ R(A, B, C) = R1(A, B) JOIN R2(A, C))
- JD를 만족하는 n-decomposable relation은 n개의 projection으로 분해해야 함
  - SPC relation은 JD *(SP, PC, CS)를 만족
    - 3-decomposable relation
    - 그러므로 5NF 아님 (3개로 분해해야 함)

Anomalies

- relation에서의 갱신이상
  - ① 삽입이상
    - relation 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

Anomalies

- relation의 갱신이상(con't)
  - ② 삭제이상
    - relation SPC에서 (S1,P1,C1)의 삭제시 다른 투플 중 어느 하나를 함께 삭제하여야 함
    - (S2,P1,C1)의 삭제는 이상 없이 가능

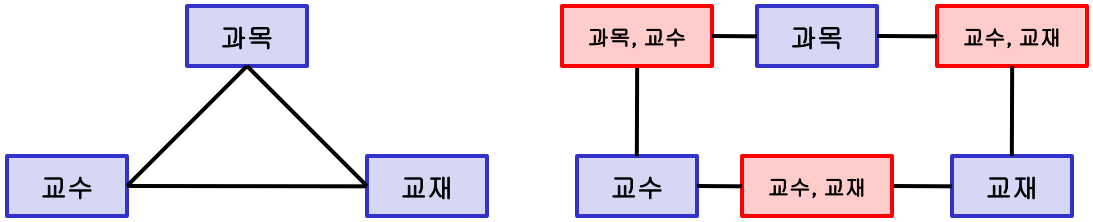
SPC

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

- 이상의 원인 : SPC는 3-분해 relation
- 이상의 해결 : relation SPC를 3-분해함

5NF: Fifth Normal Form
PJ/NF: Projection-Join Normal Form

- A relation: fifth normal form (5NF) or projection-join normal form(PJ/NF)
  - Every join dependency in R is a consequence of the candidate keys of R
  - Relation R에 존재하는 모든 조인 종속이 R의 후보키를 통해 성립되면, R은 5NF
- 예제 SPC : 5NF 아님
  - JD를 가지고 있고, JD *(SP,PC,CS)는 후보키 (SK,PK,CK)를 통하지 않고 후보키가 아닌 (SK, PK), (PK, CK), (CK, SK)를 통해서 JD 성립하므로 5NF 아님
  - SP,PC, CS 각각은 5NF (∵ no JDs)



조인 종속이 R의 후보키를 통해 성립된다는 의미

5NF
설명 끝

- 예제

  학생(학번,이름,학과,학년) relation의 후보키가 학번과 이름일 경우

  JD *((학번,이름,학과), (학번,학년))

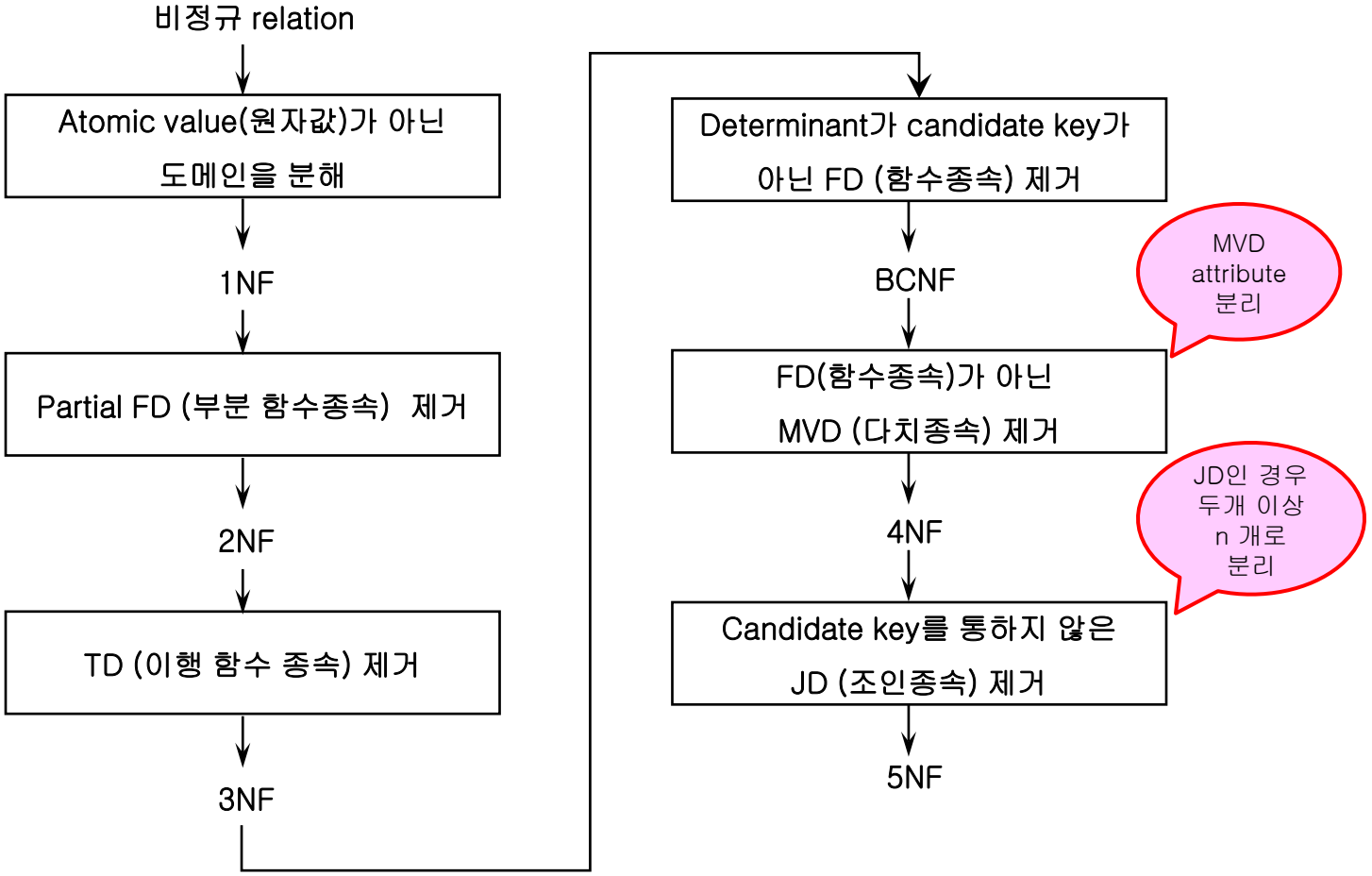
  JD *((학번,이름), (학번,학년), (이름,학과))

  위의 JD는 모두 후보키를 통해 성립됨 (JD가 2개밖에 없다면 5NF)

- 문제는 JD를 모두 찾아내는 것이 어려움
  - ◈분리하는 것이 유익한 것인지 분명치 않음

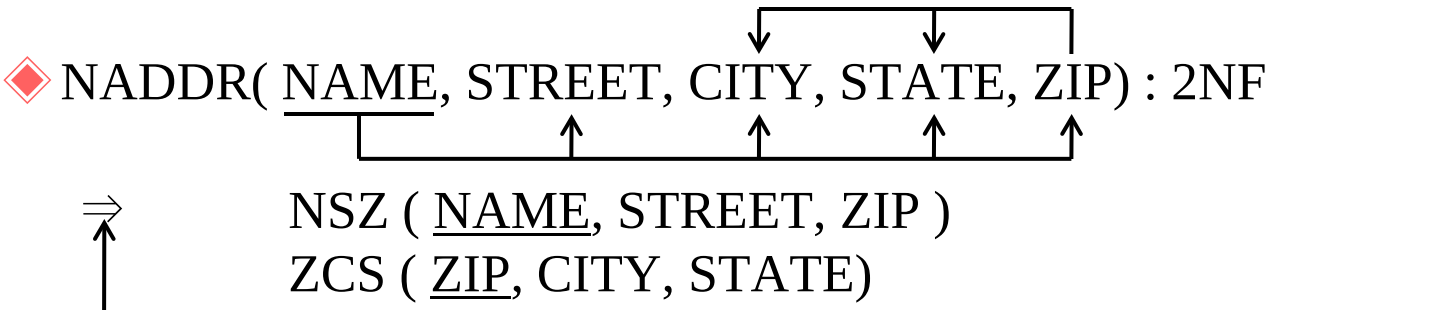
Normalization Process

- 정규화 과정 (무손실 분해)
  - 실제 정규화 과정은 정규형의 순서와 다를 수 있음



실제 문제의 예제

- 일반적으로 3NF, BCNF 정도까지 고려
  - Anomaly에 의한 inconsistency를 방지할 수 있으면 됨
    - 현실적으로 모든 relation을 반드시 5NF에 속하도록 분해할 필요는 없음



Concluding Remarks

- A relation : BCNF (4NF, 5NF)
  - every FD (MVD, JD) is a consequence of the candidate keys of R
  - anomalies: caused by FDs or MVDs or JDs that are not consequence of the candidate keys
- Normalization
  - Relation scheme을 분석하여 바람직한 구조로 재구축
    - 대충 정의한 relation을 잘~ 정리하는 과정
    - 필요한 attribute들을 단순히 한 개의 relation으로 만들어 놓았다면 중복을 피하고 inconsistency를 방지할 수 있도록 normalization을 통해 정리 해야함
      - FD, MVD, JD를 가지고 Entity 및 relationship 추출 필요
  - E-R diagram을 그려서 설계한 후에 relation으로 만드는 것이 좋음
- Objectives of normalization process
  1. to eliminate certain kinds of redundancy
  2. to avoid certain update anomalies
  3. to produce a good representation of the real world
  4. to simply the enforcement of certain integrity constraints

Concluding Remarks

- Dependency, normalization: data semantics
  Relational algebra, relational calculus, data language: data value
  - relation의 정규화는 실제 데이터 값이 아니라 개념적인 측면에서 다루어져야 함
- Decomposition operator
  - projection
  - restriction

  Recomposition operator
  - natural join
  - union
- Beyond the 5NF
  - 6NF
  - Optimal Normal Form
  - Domain-key Normal Form
  - Etc.