Schema Refinement and Normal Forms

- 스키마 설계와 변환
- 함수 종속(Functional Dependency)
- 정규화(Normalization)

데이타의 논리적 표현

- 관계 스킴(relational scheme)의 설계
  - 관계 모델을 이용하여 어떻게 실세계를 정확히 표현할 것인가?

i. 애트리뷰트, 엔티티, 관계성을 파악

ii. 관련된 애트리뷰트들을 릴레이션으로 묶음
데이터 종속성 : 애트리뷰트들 간의 관계성
효율적인 데이터 조작
데이터의 중복성 감소

iii. 변칙적 성질인 이상(anomaly) 예방

이상의 원인과 해결책

- 이상(anomaly)의 원인은 ?
  - 애트리뷰트들 간에 존재하는 여러 종속관계를 하나의 릴레이션에 표현

- 이상 (anomaly)의 해결 방법은 ?
  - 애트리뷰트들 간의 종속관계를 분석하여 여러 개의 릴레이션으로 분해(decomposition)

    ⇒ 정규화(normalization)

함수 종속(Functional Dependency)(1/2)

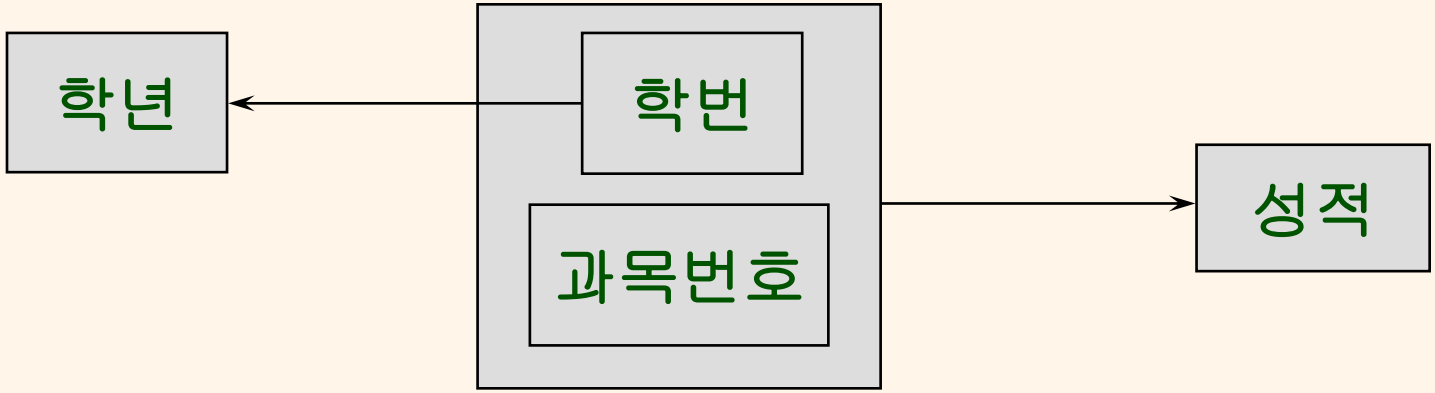
- 정의
  - 어떤 릴레이션 R에서, 애트리뷰트 X의 값 각각에 대해 애트리뷰트 Y의 값이 하나만 연관
    애트리뷰트 Y는 애트리뷰트 X에 함수 종속 : X → Y
    Ex) 학번 → 이름
  - 애트리뷰트 X는 Y를 함수적으로 결정
    X는 Y의 결정자(determinant)
    Y는 X의 종속자(dependent)
  - X, Y는 복합 애트리뷰트일 수 있음
    Ex) (학번, 과목) → 성적

함수 종속(Functional Dependency)(2/2)

- Note
  - 릴레이션 R에서 애트리뷰트 X가 후보 키이면,
    R의 모든 애트리뷰트 Y에 대해 X → Y 성립
  - 함수종속 X → Y의 경우
    애트리뷰트 X가 반드시 키라는 것을 요건으로 하지 않음

함수 종속 다이어그램

- 수강 릴레이션에서 기본 키는?
  (학번, 과목번호)



- 함수 종속은?

(학번, 과목번호) → 성적
학번 → 학년

Reasoning About FDs(1/2)

- Armstrong's Axioms (X, Y, Z are sets of attributes):
  - *Reflexivity*: If $Y \subseteq X$, then $X \rightarrow Y$

    Ex) X: (학번, 과목번호), Y: 과목번호
  - *Augmentation*: If $X \rightarrow Y$, then $XZ \rightarrow YZ$ for any Z

    Ex) X: 학번, Y: 이름, Z: 학년

    If $XZ \rightarrow YZ$, then $X \rightarrow Y$ for any Z ?

    No, Ex) X: 학번, Y: 성적, Z: 과목번호
  - *Transitivity*: If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$

    Ex) X: 학번, Y: 지도교수, Z: 학과
  - *Union*: If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$

    Ex) X: 학번, Y: 이름, Z: 학년
  - *Decomposition*: If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$

    Ex) X: 학번, Y: 이름, Z: 학년

Reasoning About FDs(2/2)

- F^+(*closure(폐포) of F*) is the set of all FDs that are implied by *F*
- Ex) 어떤 공급자 S가 부품 P를 수량 Q만큼 부서 D가 수행하는 프로젝트 J에 대해 금액 V로 공급하기로 계약 C를 체결함
- *Contracts(cid,sid,jid,did,pid,qty,value)* 릴레이션에 대해 다음의 FD들이 성립한다고 가정
  - 계약번호 C는 후보 키 : C → CSJDPQV
  - 한 프로젝트에서 하나의 부품을 구매하는 것은 단 하나의 계약을 통해서만 이루어짐: JP → C
  - 각 부서는 각 공급자로부터 최대 하나의 부품만 구매할 수 있음: SD → P
- *F 의 Closure* F^+ ?
  - JP → C, C → CSJDPQV imply JP → CSJDPQV
  - SD → P implies SDJ → JP
  - SDJ → JP, JP → CSJDPQV imply SDJ → CSJDPQV

기본 정규형

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

- 정규화(Normalization) 의 과정
  ① 애트리뷰트들과 이들의 제약 조건 (종속성)들을 수집
  ② 수집된 결과를 명시된 제약 조건에 따라 여러 개의 릴레이션으로 분할 ⇒ 스키마 변환 (schema transformation)

- 정규화(Normalization)의 원칙
  ① 무손실 표현
    - 같은 의미의 정보 유지하면서 그러나 더 바람직한 구조
  ② 데이터의 중복성 감소
  ③ 분리의 원칙
    - 독립적인 관계는 별개의 릴레이션으로 표현
    - 릴레이션 각각에 대해 독립적 조작이 가능

제1 정규형 (1NF)(1/5)

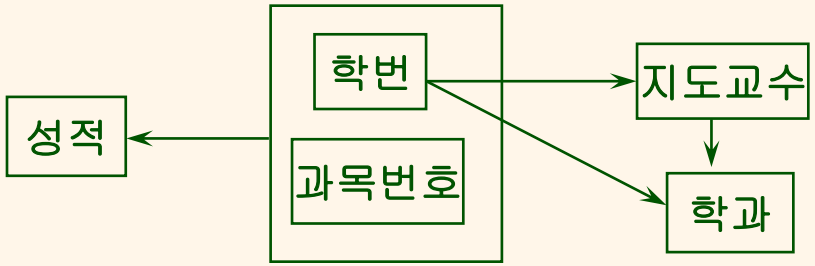
- 정의
  - 모든 도메인이 원자 값(atomic value)만으로 된 릴레이션

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	C412	C

- 기본 키는 ?
  (학번, 과목번호)

- 함수 종속은 ?
  (학번, 과목번호) → 성적
  학번 → 지도교수
  학번 → 학과
  지도교수 → 학과



제1 정규형(2/5)

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	C412	C

1NF에서의 이상

① 삽입이상

500번 학생의 지도교수가 P4라는 사실을 삽입할 경우 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 P3로 변경해야 함

제1정규형(3/5)

- <u>복합 애트리뷰트</u> X에 대하여 X → Y가 성립할 때

- *완전 함수 종속* (full functional dependency)
  - X’⊂ X 이고 X’→Y 를 만족하는 애트리뷰트 X'이 존재하지 않음
    Ex) (A, B) → C 일 때, A → C or B → C 가 성립하지 않아야 됨

- *부분 함수 종속* (partial functional dependency)
  - X’⊂ X 이고 X’→Y 를 만족하는 애트리뷰트 X'이 존재함

제1정규형(4/5)

- 예: 수강 릴레이션의 함수 종속

  (학번, 과목번호)→성적

  학번→학년

  (학년)은 (학번)에 완전 함수 종속인가?

  YES

  (학년)은 (학번, 과목번호)에는 완전 함수 종속인가?

  NO - 부분 함수 종속

  (성적)은 (학번, 과목번호)에 완전 함수 종속인가?

  YES

제1정규형(5/5)

- 1NF 이상의 원인은 ?
  - 기본 키에 부분 함수 종속된 애트리뷰트가 존재

- 1NF 이상의 해결은 ?
  - 릴레이션을 분해 (부분 함수 종속을 제거)

  ⇒ 2NF

제2 정규형 (2NF)(1/6)

- 정의
  - **1NF**이고, 키에 속하지 않는 모든 애트리뷰트들이 기본 키에 완전 함수 종속

- 무손실 분해(nonloss decomposition)

$$1NF \underset{\text{조인}}{\overset{\text{분해}}{\rightleftarrows}} 2NF$$

  - 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)

제2 정규형(2/6)

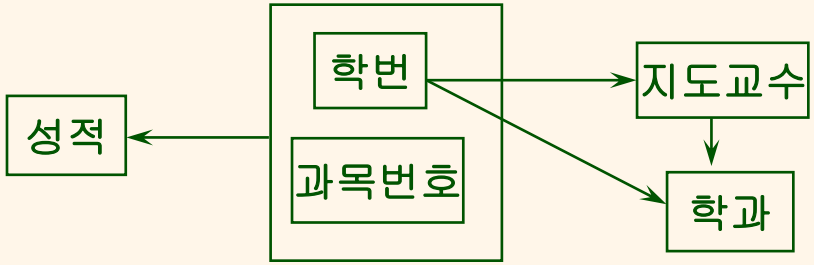
- Note: Heath의 무손실 분해
  - R(A, B, C)에서 함수 종속 A → B가 성립하면 ?

    ⇒ R1(A, B), R2(A, C) 로 무손실 분해 가능

제2 정규형(3/6)

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	C412	C



➔

지도

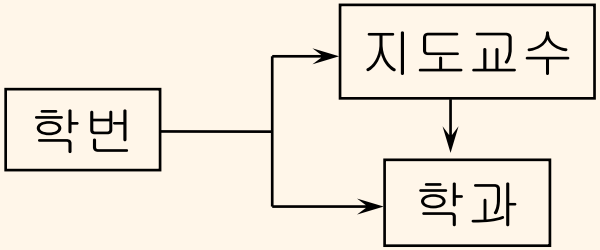
학번	지도교수	학과
100	P1	컴퓨터
200	P2	전기
300	P3	컴퓨터
400	P1	컴퓨터

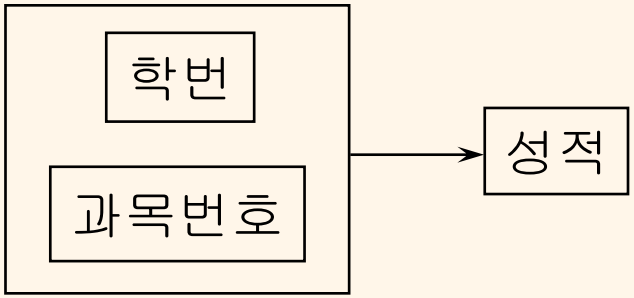
수강

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	C412	C

제2 정규형(4/6)

- 수강지도 ⇒ 지도, 수강 릴레이션으로 분해
  - 지도 (학번, 지도교수, 학과)
    학번 → 지도교수
    학번 → 학과
    지도교수 → 학과
  - 수강 (학번, 과목번호, 성적)
    (학번, 과목번호) → 성적





제2 정규형(5/6)

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	C412	C

- 2NF(지도 릴레이션)에서의 이상

① 삽입이상

어떤 지도교수가 지도하는 학생이 없다면 특정 학과에 속한다는 사실의 삽입 불가능

② 삭제이상

300번 학생의 투플을 삭제하면 지도교수 P3가 컴퓨터공학과에 속한다는 정보도 손실

③ 갱신이상

지도교수 P1의 소속이 컴퓨터공학과에서 전자과로 변경된다면 100번과 400번 학생의 투플을 모두 변경하여야 함

- 2NF 이상의 원인은 ?
  - 이행적 함수 종속이 존재

제2 정규형(6/6)

- Note :
  - 이행적 함수 종속 (TD, Transitive Dependency) ?

    A → B 이고 B → C

    ⇒ A → C

    (애트리뷰트 C는 애트리뷰트 A에 이행적 함수 종속)

- 2NF 이상의 해결은 ?
  - 릴레이션 분해 (이행적 함수 종속을 제거)

    ⇒ 3NF

제3정규형 (3NF)(1/2)

- 정의(3NF)
  - 2NF이고, 후보 키에 속하지 않는 모든 애트리뷰트들이 후보 키에 이행적 함수 종속되지 않음

- 무손실 분해

$$2NF \underset{\text{조인}}{\overset{\text{분해}}{\rightleftarrows}} 3NF$$

제3정규형(2/2)

- 지도 릴레이션은 3정규형인가?

  ➔ NO

- 지도 ⇒ 학생지도, 지도교수학과 릴레이션으로 분해

지도

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

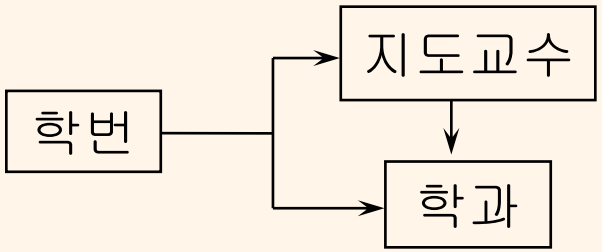
➔

학생지도

학번	지도교수
100	P1
200	P2
300	P3
400	P1

지도교수학과

지도교수	학과
P1	컴퓨터
P2	전기
P3	컴퓨터



학번 → 지도교수

지도교수 → 학과

보이스/코드 정규형(BCNF)(1/4)

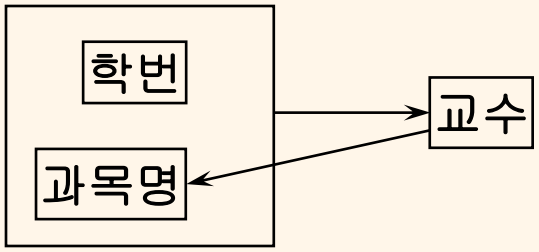
- 정의
  - 릴레이션 R의 모든 결정자가 *후보 키*이면 릴레이션 R은 BCNF에 속함
- 릴레이션 R이 BCNF에 속하면 R은 제1, 제2, 제3 정규형에 속함
- 강한 제3정규형(strong 3NF)

보이스/코드 정규형(2/4)

- 수강과목 릴레이션에서 다음과 같은 제약조건을 가정
  - 각 과목명에 대해 한 학생은 오직 한 교수의 강의만 수강함
  - 각 교수는 한 과목명만 담당함
  - 한 과목명은 여러 교수가 담당할 수 있음

수강과목

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



함수종속은 ?

(학번, 과목명) → 교수

교수 → 과목명

후보 키는 ?

(학번, 과목명), (학번, 교수)

3NF인가 ?

YES

BCNF인가 ?

NO

보이스/코드 정규형(3/4)

수강과목

학번	과목명	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 교수 P2가 자료구조를 담당한다는 정보도 삭제됨

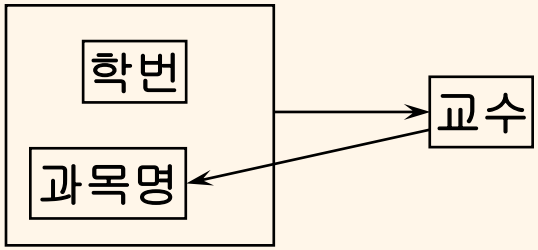
③ 갱신이상

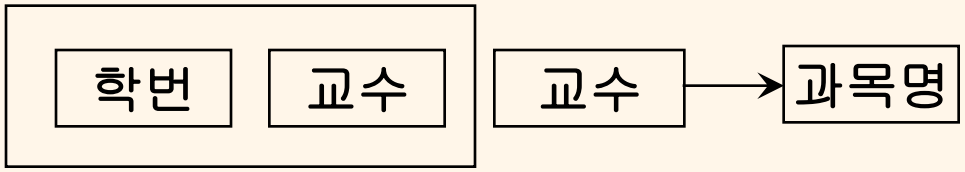
P1이 프로그래밍 대신 자료구조를 담당하게 되면 P1이 나타난 모든 투플을 변경하여야 함

- 원인은 ?
  - 교수가 결정자이나 후보 키가 아니므로

보이스/코드 정규형(4/4)

- 예(BCNF) : 수강과목 ⇒ 수강교수, 과목교수





수강과목

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

➔

수강교수

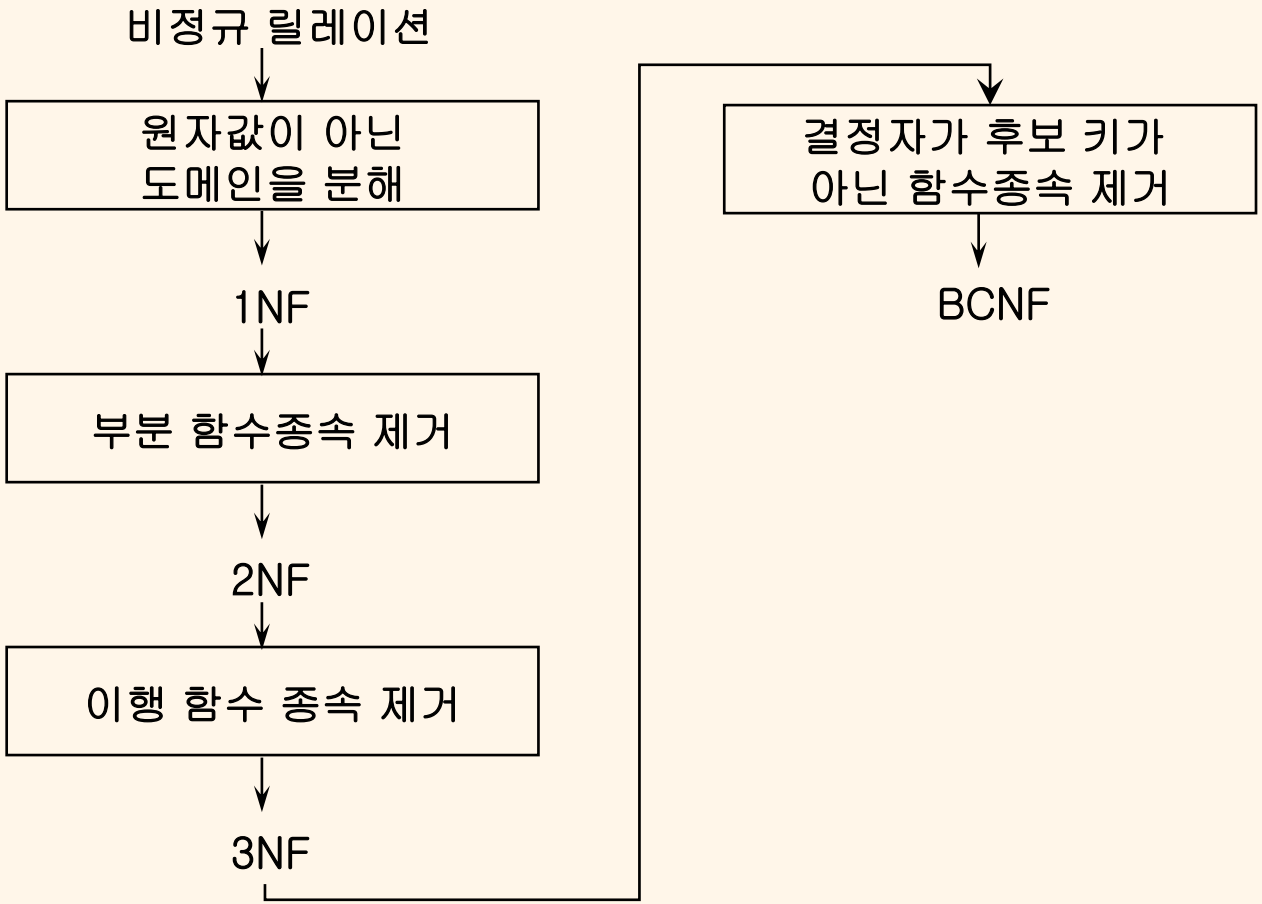
학번	교수
100	P1
100	P2
200	P1
200	P3
300	P3
300	P4

과목교수

교수	과목명
P1	프로그래밍
P2	자료구조
P3	자료구조
P4	프로그래밍

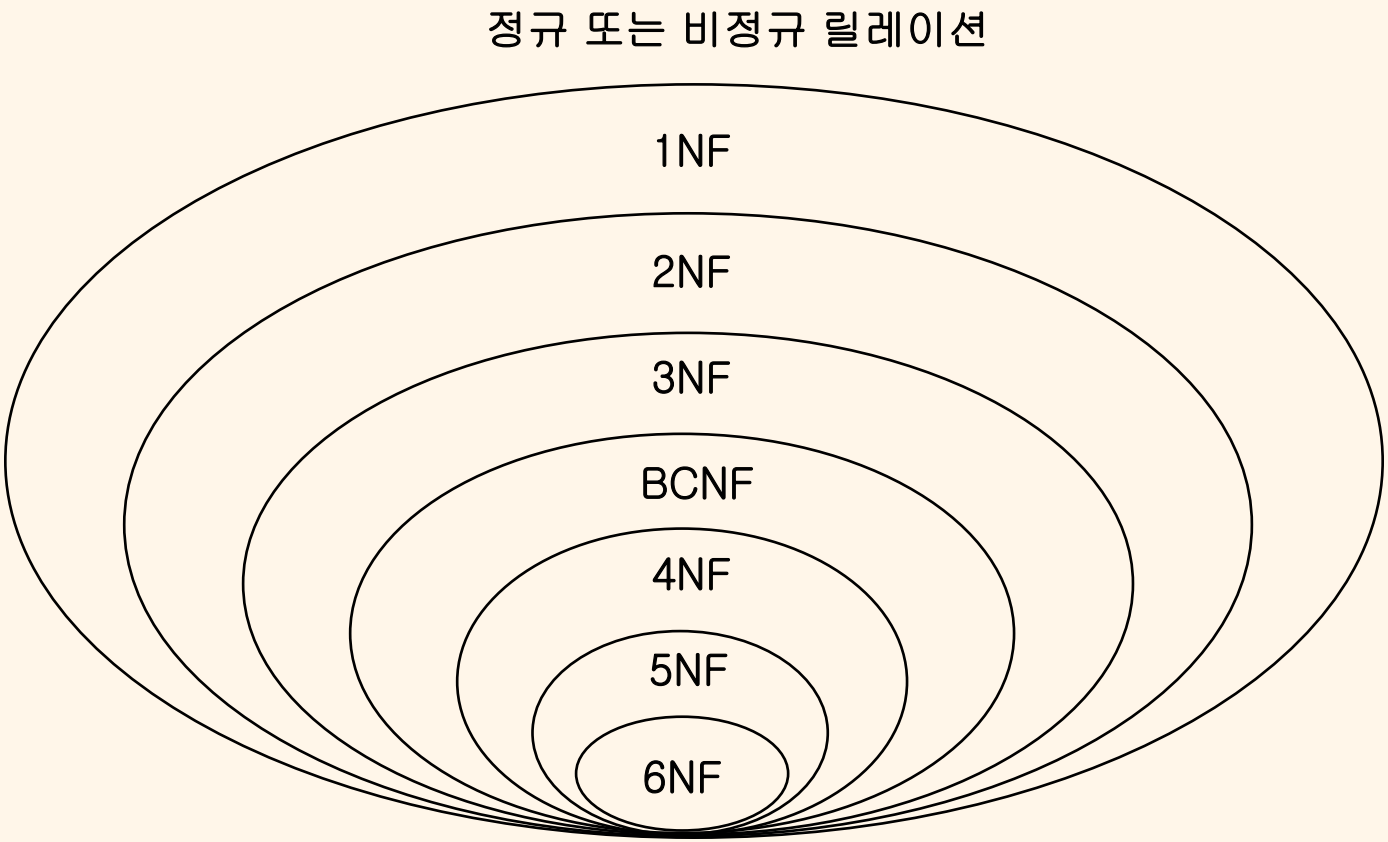
정규형들 간의 관계(1/2)

- 정규화 과정 (무손실 분해)



정규형들 간의 관계(2/2)

- 정규형들 간의 포함 관계



정규화(Normalization)

- 정규화: normalization ≒ decomposition
- 정규화의 단점은?

  테이블 수가 많아질수록 조건결합, 즉 join의 부담으로 인해 답은 더 늦게 나올 수 있음

  -> 역정규화(비정규화 또는 반정규화) : 정규화된 릴레이션을 물리적 데이터 모델링 과정에서 성능을 향상시키기 위해 중복을 허용하고라도 다시 통합하거나 분할하여 구조를 재조정하는 것

- 정규화와 테이블 수 최적화라는 두 마리의 토끼를 한꺼번에 잡기는 어려움
- 4NF, 5NF, 6NF는 테이블 수가 너무 많아지므로 실무에서 실제로 활용할 가치가 가장 높은 것은 3NF 혹은 BCNF라는 결론에 도달함

Summary(1/2)

- 1 정규형(1 NF)
  - 모든 도메인이 원자 값(atomic value)만으로 된 릴레이션
- 2 정규형(2 NF)
  - 1NF이고, 키에 속하지 않는 모든 애트리뷰트들이 기본 키에 완전 함수 종속
- 3 정규형(3 NF)
  - 2NF이고, 후보 키에 속하지 않는 모든 애트리뷰트들이 후보 키에 이행적 함수 종속되지 않음
- 보이스/코드 정규형(BCNF)
  - 릴레이션 R의 모든 결정자가 후보 키이면 릴레이션 R은 BCNF에 속함

Summary(2/2)

- If a relation is in BCNF, it is free of redundancies that can be detected using FDs. Thus, trying to ensure that all relations are in BCNF is a good heuristic.

- If a relation is not in BCNF, we can try to decompose it into a collection of BCNF relations.
  - Must consider whether all FDs are preserved.
  - Decompositions should be carried out and/or re-examined while keeping *performance requirements* in mind.