스키마 정제와 정규형

Chapter 15

중복성에 따른 여러 문제

- 중복성(*Redundancy)*은 관계 스키마에서 발생되는 여러 문제의 근본 원인이다.
  - 중복 저장,삽입/삭제/갱신 이상
- 무결성 제약조건,특히 함수종속성(functional dependency)를 분석하면 이런 문제가 있는 스키마를 파악할 수 있고, 정제 방향도 알 수 있다.
- 주된 정제 기법: 분해(decomposition)(ABCD 를 가령 AB와 BCD나 ACD와 ABD로 대체).
- 분해는 신중하게:
  - 릴레이션을 분해할 이유가 있는가?
  - 분해때문에 무슨 문제라도 생길 가능성이 있는가?

함수 종속 (FD)

- 함수 종속 X⟶Y 가 릴레이션 r에 대하여 성립한다는 것은, R이 어떠한 인스턴스 r에 대해서도:
  - t1∈ r, t2∈ r, π_X(t1) =π_X (*t2*) 면 π_Y(*t1*) = π_Y (*t2*)이다.
  - i.e., *r*의 어떤 두 투플이, X 값이 같으면 Y의 값도 같아야 한다. (여기에서 X와 Y는 애트리뷰트의 ***집합***임)
- FD 는 모든 가능한 인스턴스에 대한 진술이다.
  - 응용의 내용에 근거하여 파악되어야 한다.
  - R의 어떤 적법한 인스턴스 r이 있으면, 어떤 FD f의위배 여부는체크할 수 있어도, f가 R에 대해 성립하는지는 알 수 없다 !
- K 가 R의 후보 키이면, 당연히 K⟶R
  - 그러나 K⟶ R이라고 K가 반드시 최소일 수는 없다!

예: 개체집합에 대한 제약조건

- "시간제_직원"으로부터 얻은 릴레이션을 보자:
  - 시간제_직원(ssn, 이름, 주차면, 등급, 시간당임금, 근무시간)
- 표기법: 이 릴레이션의 스키마를 애트리뷰트 대표문자의 리스트로 표현: SNLRWH
  - 실제로는 애트리뷰트의 집합 {S,N,L,R,W,H}.
  - 릴레이션의 이름으로 전체 애트리뷰트를 대신 표현하기도 할 것이다. (e.g., SNLRWH대신 시간제_직원)
- 시간제_직원의 일부 함수 종속:
  - ssn 이 키이다: S → SNLRWH
  - 등급이 시간당임금을 결정한다: R → W

예(계속.)

- R → W에 따른 문제:
  - 갱신 이상: SNLRWH의 첫 투플에서 W를 변경할 수 있는지?
  - 삽입 이상: 등급에 따른 시간당 임금은 모른 채 직원을 삽입한다면?
  - 삭제 이상: 등급이 5인 직원들을 모두 삭제해 버리면, 등급 5의 시간당임금 정보도 잃는다!

S	N	L	R	W	H
123-22-3666	Attishoo	48	8	10	40
231-31-5368	Smiley	22	8	10	30
131-24-3650	Smethurst	35	5	7	30
434-26-3751	Guldu	35	5	7	32
612-67-4134	Madayan	35	8	10	40

S	N	L	R	H
123-22-3666	Attishoo	48	8	40
231-31-5368	Smiley	22	8	30
131-24-3650	Smethurst	35	5	30
434-26-3751	Guldu	35	5	32
612-67-4134	Madayan	35	8	40

시간제_직원2

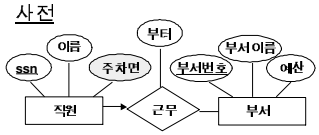
R	W
8	10
5	7

임금

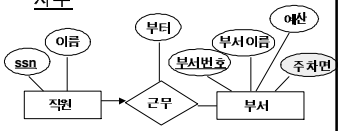
ER 다이어그램 정제

- 위의 그림을 변환하면:
  직원(S,N,L,D,S)
  부서(D,M,B)
  - 주차면을 직원에 연결.
- 한 부서의 직원들은 모두 같은 주차면을 사용한다면: D → L
- 중복성을 교정하면:
  직원2(S,N,D,S)
  부서별주차면(D,L)
- 더 정제 가능:
  직원2(S,N,D,S)
  부서(D,M,B,L)

사전



사후



FD에 대한 이해

- 얼마의 FD가 주어지면, 그로부터 새로운 FD를 추론할 수 있다.
  - ssn⟶부서번호, 부서번호⟶주차면 ⇨ ssn ⟶주차면
- FD의 집합 F가 성립할때 항상 FD f도 성립한다면, f는 F에 의해 암시(imply)된다.
  - F^+ = F의 폐쇄(closure). F에 의해 암시되는 모든 FD의 집합.
- Armstrong의 공리. (X, Y, Z 는 애트리뷰트 집합):
  - 반사(Reflexivity): If X ⊆ Y, then X ⟶Y
  - 첨가(Augmentation): If X⟶Y, then XZ ⟶YZ for any Z
  - 이행(Transitivity): If X⟶Y and Y⟶Z, then X ⟶Z
- 이들은 FD에 대한 정당(sound)하고 완전(complete)한 추론 법칙들이다.

FD에 대한 이해(계속.)

- 그 밖의 규칙:
  - 결합(Union): If X ⟶Y and X ⟶Z, then X ⟶YZ
  - 분해(Decomposition): If X ⟶YZ, then X ⟶Y and X⟶Z
- 예: 계약(계약번호, 공급자번호, 과제번호, 부서번호, 부품번호, 수량, 가격)이고
  - (계약번호)가 키: C ⟶CSJDPQV
  - 과제는 각 부품을 단일 계약으로 일괄 구매: JP ⟶C
  - 부서는 한 공급자로부터 많아야 한 부품만 구매: SD ⟶P
- JP⟶C, C⟶ CSJDPQV ⇨ JP⟶ CSJDPQV
- SD⟶P ⇨ SDJ⟶JP
- SDJ ⟶JP, JP ⟶CSJDPQV ⇨ SDJ ⟶CSJDPQV

FD에 대한 이해 (계속.)

- FD 집합의 폐쇄를 구하는 부하는 크다.(폐쇄의 크기가 애트리뷰트 수에 지수 비례하기 때문!)
- 대개는 FD X⟶Y가 FD 집합 F의 폐쇄에 속하는지만 알면 되므로:
  - F에 대한 X의 애트리뷰트 폐쇄(X^+로 표기)를 구한다:
    - * F^+ 에 속한 X⟶A에 해당하는 모든 애트리뷰트 A의 집합임.
    - * 이것을 구하는 선형시간 알고리즘이 존재한다.
  - Y 가 X^+에 속하는지 검사하면 끝.
- F = {A⟶B, B⟶C, C D⟶E }가 A⟶E를 암시하는가?
  - 즉, F^+에 A ⟶ E가 속하는가? 곧, E가 A^+에 속하는가?

정규형(Normal Form)

- 스키마 정제 문제로 돌아가면, 첫번째로 더 이상의 정제가 필요한지를 판단하여야 한다.
- 릴레이션이 특정 정규형 (BCNF, 3NF etc.), 에 속한다는 것은, 어느 종류의 문제가 없다/적다는 것이 된다. 따라서 릴레이션을 분해할 필요가 있는지 판단하기 용이하다.
- 중복성 탐지에 FD가 유용하다:
  - 릴레이션 R에 애트리뷰트가 3개, ABC가 있다고 하자.
    - * FD가 없다면: 여기에는 중복성이 없다.
    - * A ⟶B가 성립한다면: A값이 같은 투플들이 여러개일 수는 있지만, 이들의 B값도 같게 된다!

Boyce-Codd 정규형 (BCNF)

- FD집합 F를 가진 릴레이션 R이, F^+에 속한 모든 X→A에 대해서 다음이 성립하면, R은 BCNF에 속한다.
  - $A \in X$ (당연한 FD), 또는
  - X 에 R의 어떤 키가 포함된다.
- 즉, R에서 성립하는 비당연 FD는 키 제약조건밖에 없다고 하면, R은 BCNF에 속한다.
  - FD로 찾을 수 있는 어떠한 중복성도 R에는 없다.
  - 일반적으로 두 투플의 X값이 같다고 해서 한 투플의 A값으로부터 다른 투플의 A값을 추론할 수는 없다.
  - 그런데 우측 릴레이션이 BCNF에 속한다면, 두 투플은 똑같아야 한다(X가 키이므로).

X	Y	A
x	y1	a
x	y2	?

제 3 정규형 (3NF)

- FD집합 F를 가진 릴레이션 R이, F+에 속한 모든 X→A 에 대해서 다음이 성립하면, R은 3NF에 속한다.
  - $A \in X$ (당연 FD), 또는
  - X에 R의 어떤 키가 포함된다. 또는
  - A가 R의 어떤 키의 일부이다.
- 위 세번째 조건에서 "키의 최소성"이 중요!
- R 이 BCNF, 에 속하면, 3NF에도 속한다..
- R이 3NF에 속하면 중복성이 잔존해 있을 가능성이 있다. BCNF까지 갈 수 없을 때(가령, 좋은 분해법이 없거나 성능상의 이유로) 3NF로 만족하게 된다.
  - R을 3NF들로 무손실-조인, 종속성유지 분해하는 것은 언제나 가능하다.

3NF의 의미는?

- X→A 가 3NF를 위배하는 경우는 다음 두 가지이다:
  - X가 어떤 키 K의 진부분집합이다.
    - * (X, A)가 중복 저장된다.
  - X가 어떤 키에도 속하지 않는다.
    - * K→X→A 라는 연쇄적인 FD가 생긴다. 따라서 어떤 K 값에 어떤 X값을 연관시킬 때에는 그 X값에 어떤 A값을 연관시켜 주어야 한다.
- 그러나: 3NF 릴레이션에서도 문제는 발생할 수 있다.
  - e.g., "예약" SBDC, S→ C, C→S는 3NF 에 속하지만, 뱃사람 S가 예약할 때마다 동일한 (S,C)가 중복저장된다
- 그렇기 때문에 3NF는 BCNF대신 사용하는 타협점인 것이다.

릴레이션 분해

- 릴레이션 R의 애트리뷰트가 A1 ... An이라고 하자 R의 분해(decomposition) 란 R을 다음과 같이 둘 이상의 릴레이션으로 대체하는 것이다.
  - 새 릴레이션들은 R의 애트리뷰트 중 일부씩만 가지고 (R에 없던 애트리뷰트는 갖지 않으며),
  - R의 애트리뷰트들은 모두 새 릴레이션 중 어느 하나에는 나타난다.
- R을 분해한다는 것은 R의 인스턴스를 저장하는 대신, 분해로 생성된 릴레이션들의 인스턴스를 저장하겠다는 것이다.
- E.g., SNLRWH를 SNLRH와 RW로 분해 가능.

분해 사례

- 꼭 필요할 때에만 분해하도록 한다.
  - SNLRWH에는 FD S→SNLRWH 와 R→W가 있다.
  - 두번째의 FD때문에 3NF가 못된다. W와 R의 값의 연관이 중복 표현된다. 가장 쉬운 해결책은, 이 정보용으로 RW릴레이션을 새로 만들고 원래 스키마에서는 W를 빼는 것이다.
    - * 즉, SNLRWH를 SNLRH와 RW로 분해하였다.
- 원래 저장할 정보는 SNLRWH 투플 형태이다. 이 투플들을 SNLRH 와 RW에 대해 프로젝션한 것들만 저장할 때, 무슨 문제라도 발생할 가능성은 없는지?

분해에 따르는 문제

- 다음 세 가지의 문제가 발생할 가능성이 있다:
  - ❶어떤 질의는 처리 비용이 더 높아진다.
    - * e.g., Joe 는 얼마나 버는가? (급여 = W*H)
  - ❷분해한 릴레이션 인스턴스들로부터 원래 릴레이션의 인스턴스를 재구성할 수 없는 경우도 생긴다!
    - * SNLRWH 사례는 다행히 아니다.
  - ❸어떤 종속성을 체크하려면 분해된 릴레이션의 인스턴스들을 죠인하여야 되는 경우도 있다.
    - * SNLRWH 사례는 다행히 아니다.
- Tradeoff: 이러한 문제와 중복성 간의 타협점을 찾아야 한다.

무손실-죠인 분해

- FD집합 F를 만족하는 모든 인스턴스 r에 대해 다음이 성립하면, R을 X와 Y로 분해하는 것이 F에 대해 무손실-죠인(lossless-join)이라고 한다:
  - $\pi_X(r) \bowtie \pi_Y(r) = r$
- $r \subseteq \pi_X(r) \bowtie \pi_Y(r)$ 은 항상 참.
  - 역은 참이 아닌 경우도 있다! 역도 참이면 무손실-죠인 분해이다.
- 이 정의를 3개 이상으로의 분해에 그대로 확장할 수있다.
- 중복성 제거를 위한 분해라면 반드시 무손실이어야 한다! (문제(2)방지.)

무손실-죠인 계속

- R을 X와 Y로 분해하는 것은 무손실-죠인이다.
  - $X \cap Y \longrightarrow X$, 또는
  - $X \cap Y \longrightarrow Y$
- 특히, $U \longrightarrow V$가 R에 대해 성립할 때, R을 UV와 R-V로 분해하면 무손실-죠인 분해가 된다.

A	B	C
1	2	3
4	5	6
7	2	8

A	B
1	2
4	5
7	2

B	C
2	3
5	6
2	8

A	B	C
1	2	3
4	5	6
7	2	8
1	2	8
7	2	3

종속성유지 분해

- CSJDPQV에서 C가 키이고 JP⟶C이며 SD⟶P라고 하자.
  - BCNF 분해: CSJDQV와 SDP
  - 문제점: JP⟶C를 체크하려면 죠인이 필요함!
- 종속성유지 분해란?
  - R이 X, Y, Z로 분해하고 X, Y, Z 각각에 대한 FD들을 집행하면 원래의 R에 대한 모든 FD들이 만족되어야 한다.(문제(2)방지)
- FD집합 F의 프로젝션: R이 X,…으로 분해되었을 때, F의 X에 대한 프로젝션(F_X로 표기)이란 X에 속한 U, V에 대하여 F^+ (F의 폐쇄)에 속하는 FD U⟶V들의 집합이다.

종속성유지 분해(계속.)

- R을 X와 Y로 분해할 때 (F_X 합집합 F_Y)$^+$ = F^+가 되면 종속성 유지라고 한다.
  - 즉, 폐쇄 F^+에 속하는 종속성 중 X로만 체크할 수 있는 것들과 Y로만 체크할 수 있는 것들을 모으면 F^+의 모든 종속성들이 암시(imply)된다.
- F가 아니라 F^+인 점에 주의
  - ABC, A⟶B, B⟶C, C⟶A 이고, AB와 BC로 분해.
  - 이것이 종석성유지인가? C⟶A 가 유지되는가?????
- 종속성유지라고 무손실-죠인인 것은 아니다:
  - ABC, A⟶B, AB와 BC로 분해하는 경우.
- 역도 마찬가지! (예는?)

BCNF로 분해

- 릴레이션 R이 있다고 하자. X⟶Y때문에 BCNF가 안된다면, R을 R - Y 와 XY로 분해한다.
  - 이 방식을 반복적용하면 BCNF 릴레이션 집단을 얻게 된다. 이 분해는 무손실-죠인이며, 언젠가는 반드시 끝난다.
  - e.g., CSJDPQV, 키는 C, JP⟶C, SD⟶P, J⟶S
  - SD⟶P때문에, SDP와 CSJDQV로 분해.
  - J⟶S때문에, CSJDQV를 JS와 CJDQV로 분해.
- 일반적으로는 BCNF를 위배하는 종속성이 여러개 일 수 있다. 그 중에 어느 것을 먼저 보느냐에 따라 다른 릴레이션 집단을 얻게 된다.

BCNF 와 종속성유지

- 일반적으로, BCNF 분해는 종속성유지를 보장하지 못한다.
  - e.g., CSZ, CS → Z, Z → C
  - 첫 FD를 유지하면서 BCNF 분해는 불가능하다.
- CSJDQV를 SDP, JS와 CJDQV 로 분해하는 것도 종속성 유지가 아니다. (FD들이 JP → C, SD → P, J → S일때.)
  - 그러나 무손실-조인은 된다.
  - 이 경우, JPC를 추가하면 종속성유지 분해로 만들 수 있다.
    * JPC 의 투플들은 FD 검사에만 이용! (중복!)

3NF로 분해

- BCNF로 무손실-조인 분해하는 알고리즘을 3NF 무손실-조인 분해에도 당연히 적용할 수 있다.(대개는 더 빨리 끝난다)
- 종속성유지를 확보하는 한가지 아이디어:
  - X → Y가 유지되지 않으면, 릴레이션 XY를 추가한다.
  - XY가 3NF가 아닐 수도 있는 문제! e.g., JP → C를 유지하려고 CJP를 추가하였는데, J → C도 있다면 ?
- 개선: 원래의 FD집합 F 대신, F의 최소 포괄(miniml cover)을 이용한다.

FD 집합의 최소 포괄

- FD 집합 F에 대한 최소 포괄 G:
  - F의 폐쇄 = G의 폐쇄.
  - G에서는 어떤 FD의 오른쪽도 단일 애트리뷰트.
  - G에서 FD 하나를 제거하든지 FD의 애트리뷰트를 제거하면, 폐쇄가 바뀐다.
- 즉 F와 동일한 폐쇄를 얻기 위하여, G의 모든 FD는 필요하면서도 "가급적 최소" 이다.
- e.g., A → B, ABCD → E, EF → GH, ACDF → EG 이면 최소 포괄은 다음과 같다:
  - A → B, ACD → E, EF → G, EF → H
- 최소포괄 → 무손실-조인, 종속성유지 분해!!!(책 참조)

스키마 정제 요약

- BCNF에 속하는 릴레이션에는 FD로 탐지할 수 있는 중복성이 하나도 없다. 따라서 가급적 모든 릴레이션이 BCNF에 속하도록 하는 것이 바람직하다.
- BCNF가 아닌 릴레이션은, BCNF 릴레이션 집단으로 분해해 볼 수 있다.
  - FD가 모두 유지되는지 보아야 한다. BCNF로 무손실-조인, 종속성유지 분해가 불가능하면(또는 통상적인 질의에 부적절하면) 3NF까지로만 분해할 수도 있다.
  - 분해할 때에나 그 후에 주어진 "성능 요건"을 따져 보아야 한다.