



BCNF 테스트

§ 사소하지 않은 종속성이 있는지 확인하려면

비 BCNF 위반 원인

1. a^+ (a 의 속성 클로저)를 계산 하고, 2. 그것이 R 의 모든 속성을 포

함하는지 확인합니다. 즉, R 의 수퍼키입니다. § 단순화된 테스트: 관계 스키마 R 이 BCNF에 있는지 확인하려면, F^+ 의 모든 종속성을 검사하는 대신 BCNF 위반에 대해 주어진 집합 F 의 종속성만 검사하는 것으로 충분합니다.

- F 의 종속성 중 어느 것도 BCNF를 위반하지 않으면 F^+ 의 종속성도 BCNF를 위반하지 않습니다.
§ 그러나 관계를 테스트할 때 F 만을 사용하는 단순화된 테스트는 올바르지 않습니다.

R 의 분해에서 • $R = (A, B, C, D, E)$

를 고려하고 $F = \{ A \xrightarrow{b} B, C \xrightarrow{b} D \}$

§ R 을 $R_1 = (A, B)$ 및 $R_2 = (A, C, D, E)$ 로 분해합니다. § F 의 종속성 중 어

느 것도 다음의 특성만 포함하지 않습니다.

$R_2 = (A, C, D, E)$ 그래서 우리는 R_2 가 BCNF를 만족한다고 오해할 수 있습니다.

§ 실제로 F^+ 의 종속성 $AC \xrightarrow{b} D$ 는 R_2 가 BCNF에 없음을 나타냅니다.

- $A \xrightarrow{b} B \Rightarrow$ 교류 $\xrightarrow{b} BC$ (의사 이행성 규칙에 따름)
- $AC \xrightarrow{b} BC$ 와 $BC \xrightarrow{b} D \Rightarrow AC \xrightarrow{b} D$



BCNF에 대한 테스트 분해

R 의 분해에서 관계 R_i 가 BCNF에 있는지 확인하려면

§ R_i 에 대한 F^+ 의 제한과 관련하여 BCNF에 대해 R_i 를 테스트합니다 (즉, 모든 R_i 의 속성만 포함하는 F^+ 의 FD) § 또는 R 을 유지하는 원래 종속

성 F 세트를 사용 하지만 다음 테스트를 사용합니다. § 모든 속성 $a \in R_i$ 세트에 대해 a^+ (a 의 속성 폐쇄) ($R_i - a$)의 속성을 포함

하지 않거나 R_i 의 모든 속성을 포함합니다. § R_i 는 BCNF를 만족합니다.

- R_i 의 일부 a 에 의해 조건이 위반되면 종속성 $a^+ \not\subseteq R_i$ 가 R_i 를 유지하는 것으로 표시될 수 있으며 R_i 는

BCNF를 위반합니다. • R_i 를 분해하기 위해 위의 종속성을 사용합니다. • 예:

§ $R_2 = (A, C, D, E)$, $F = \{ AD \}$ § $a = \{ A, C \}$ § R_2 , $(AC)^+ = \{ ABCD \}$ 는 다음을 포함합니다.

$(R_2 - a)$ 의 속성 $D = \{ D, E \}$ § $AC \xrightarrow{b} D$

§ $(\{ ABCD \} - \{ AC \}) = \{ BD \}$ 무엇 $R_2 = \{ D \}$ § D

§ $R_2 = (A, C, D, E)$ 는 $R_3 = \{ A, C, D \}$ 및 $R_4 = \{ A, C, E \}$ 로 분해됩니다.

- R_3 및 R_4 는 BCNF에 있습니다.



BCNF 분해 알고리즘

```

결과 := {R}; 완
료 := 거짓; F +를 계
산하십시오. while
(not done) do if ( 결과 에
BCNF에 없는 스키마 Ri가 있음) then begin let a가 Ri를 유지하는 중요 하
지 않은 기능적
종속성이 되도록 하여 a가 Ri를 포함하지 않고 a Ç = A; 결과 := (결과 -
Ri) È (Ri - b) È (a, b); 그렇지 않으면 완료됨 := true;
비

```

참고: 각 Ri는 BCNF에 있으며 분해는 무손실 조인입니다.



BCNF 분해의 예

§ class (course_id, 제목, dept_name, 학점, sec_id, 학기, 연도,
building, room_number, capacity, time_slot_id) § 기능 종

속성: • course_id → title, dept_name,
credits • building, room_number → capacity •
course_id, sec_id, term, year → building,
room_number, time_slot_id

§ 후보 키 {course_id, sec_id, 학기, 연도}. § BCNF 분해:

- course_id → title, dept_name, 학점 보유
§ 그러나 course_id는 수퍼키가 아닙니다.
- 수업을 다음으로 대체합니다.
§ 과목(과목_ID, 제목, 학과명, 학점) § 클래스-1(과목_ID, 초ID,
학기, 연도, 건물,
room_number, 수용 인원, time_slot_id)



BCNF 분해(계속) Composition (Cont.)

§ 과정은 BCNF에 있습니다.

- 우리는 이것을 어떻게 압니까?

§ 건물, 호실_번호→정원은 1등급 보유

- 그러나 {building, room_number}는 class-1의 수퍼키가 아닙니다. • 클래스 1을 다음으로 대체합니다.

§ 강의실(건물, 호실_번호, 수용인원) § 섹션 (강좌ID, 초
ID, 학기, 연도, 건물,
room_number, time_slot_id)

§ 교실과 섹션은 BCNF에 있습니다.



제3정규형 Third Normal Form

§ 다음과 같은 몇 가지 상황이 있습니다.

- BCNF는 종속성을 보존하지 않으며 • 업데이트 에서
FD 위반을 효율적으로 확인하는 것이 중요합니다.

§ 해결책: 3NF(Third Normal Form)라고 하는 더 약한 정규형을 정의 합니다 .

나중에 예시)

- 그러나 개별 관계에서 기능 종속성을 확인할 수 있습니다.
조인을 계산하지 않고.
- 3NF로의 무손실 조인 종속성 보존 분해가 항상 있습니다.



3NF 예제 -- 관계 dept_advisor

§ 부서_고문(s_ID, i_ID, 부서_이름)

$F = \{s_ID, dept_name \twoheadrightarrow i_ID, i_ID \twoheadrightarrow dept_name\}$ § 2개의

후보 키: s_ID, dept_name 및 i_ID, s_ID

§ R은 3NF에 있음

- s_ID, dept_name \twoheadrightarrow i_ID §
(s_ID, dept_name) 은 수퍼키입니다.
- i_ID \twoheadrightarrow 부서명
§ dept_name은 후보 키에 포함되어 있습니다.



3NF 테스트 for 3NF

§ F의 FD만 확인해야 하며 F+의 모든 FD를 확인할 필요는 없습니다. § 속성 클로저를 사용하여 a가 a인 경우 각 종속성 $a \twoheadrightarrow b$ 를 확인합니다. 슈퍼키.

§ a가 슈퍼키가 아닌 경우 b의 각 속성이 포함되어 있는지 확인해야 합니다. R의 후보 키에서

- 이 테스트는 후보자를 찾는 것과 관련되기 때문에 다소 비용이 많이 듭니다. 열쇠
 - 3NF에 대한 테스트는 NP-hard인 것으로 나타났습니다. • 흥
- 미롭게도 제3정규형(간단히 설명)으로의 분해는 다항식 시간 내에 수행될 수 있습니다.



3NF 분해 알고리즘

F_c 를 F 에 대한 정식 표지라고 하자. $\text{나는} :=$

0; **각**

가능 종속성에 대해 \otimes 시작

비 F_c 에서 $_$

$\text{나는} := \text{나는} + 1;$

$\text{리} := \text{비}$

끝

스키마 $R_j, 1 \leq j \leq i$ 중 어느 것도 R 에 대한 후보 키를 포함하지 않으면 i 시작 $:= i + 1;$

$R_i := R$ 에 대한 임의의 후보 키; **끝**

/* 선택적으로 중복 관계 제거 */

임의의 스

키 R_j 가 다른 스키마 R_k 에 포함되어 있으면 반복하고 /* R_j 삭제 */

$R_j = R_i ; i = i - 1;$ 반환 ($R_1,$

$R_2, \dots,$

R_i)



3NF 분해 알고리즘(계속)

위의 알고리즘은

\S 각 관계 스키마 R_i 는 3NF에 있음

\S 분해는 종속성 보존 및 무손실 조인입니다. \S 정확성 증명은 이 프레젠테이션의

끝에 있습니다(여기를 클릭).



3NF 분해: 예

§ 관계 스키마:

$\text{cust_banker_branch} = (\text{customer_id}, \text{employee_id}, \text{branch_name}, \text{type})$

§ 이 관계 스키마의 기능 종속성은 다음과 같습니다.

- $\text{customer_id}, \text{employee_id} \twoheadrightarrow \text{branch_name}, \text{type}$

- $\text{employee_id} \twoheadrightarrow \text{branch_name}$

$\text{customer_id}, \text{branch_name} \twoheadrightarrow \text{employee_id}$ § 먼저 표준

표지를 계산합니다.

- branch_name 은 첫 번째 종속성의 rhs에서 관련이 없습니다. • 다른 속성은 관련

이 없으므로 $\text{FC} = \text{customer_id}, \text{employee_id} \twoheadrightarrow \text{type}$

$\text{employee_id} \twoheadrightarrow \text{branch_name}$

$\text{customer_id}, \text{branch_name} \twoheadrightarrow$

employee_id 가 됩니다.



3NF 분해 예제(계속)

§ for 루프 는 다음 3NF 스키마를 생성합니다.

$(\text{customer_id}, \text{employee_id}, \text{type})$

$(\text{employee_id}, \text{branch_name})$

$(\text{customer_id}, \text{branch_name}, \text{employee_id})$ •

($\text{customer_id}, \text{employee_id}, \text{type}$)에 원래 스키마의 후보 키가 포함되어 있으므로 추가 관계 스키마를 추가할 필요가 없습니다.

§ for 루프의 끝에서 다른 스키마의 하위 집합인 $(\text{employee_id}, \text{branch_name})$ 과 같은 스키마를 검색하고 삭제합니다.

- 결과는 FD가 고려되는 순서에 따라 달라지지 않습니다. § 그 결과 단순화된 3NF 스키마는

다음과 같습니다.

$(\text{customer_id}, \text{employee_id}, \text{유형})$

$(\text{customer_id}, \text{branch_name}, \text{employee_id})$



BCNF와 3NF의 비교

§ 다음과 같이 관계를 3NF에 있는 일련의 관계로 분해하는 것이 항상 가능합니다.

- 분해는 무손실입니다. • 종속성이 보존됩니다.

§ 관계를 일련의 관계로 분해하는 것은 항상 가능합니다. 다음과 같이 BCNF에 있습니다.

- 분해는 무손실입니다.
- 종속성을 유지하는 것이 불가능할 수 있습니다.



디자인 목표

§ 관계형 데이터베이스 디자인의 목표는 다음과 같습니다.

- BCNF.
- 무손실 조인. • 종속성 보존. § 이를 달성할 수 없

는 경우 다음 중 하나를 수락합니다. • 종속성 보존 부족 • 3NF 사용으로 인한 중복성

§ 흥미롭게도 SQL은 기능을 지정하는 직접적인 방법을 제공하지 않습니다. 슈퍼 키 이외의 종속성.

어설션을 사용하여 FD를 지정할 수 있지만 테스트 비용이 많이 들고 현재 널리 사용되는 데이터베이스에서 지원하지 않습니다!) § 종속성 보존 분해가 있더라도 SQL

을 사용하면 기능적 종속성을 효율적으로 테스트할 수 없습니다. 그의 왼쪽은 열쇠가 아닙니다.



MVD(다중값 종속성) Dependencies (MVDs)

§ 자녀의 이름과 자녀의 전화번호를 기록한다고 가정합니다.
강사:

- inst_child(아이디, 아이이름)
- inst_phone(ID, 전화번호)

§ 이러한 스키마를 결합하여

- inst_info(ID, child_name, phone_number)

예시 데이터:

(99999, 데이빗, 512-555-1234)

(99999, 윌리엄, 512-555-4321)

(99999, 윌리엄, 512-555-1234)

(99999, 데이빗, 512-555-4321)

§ 이 관계는 BCNF에 있습니다.

- 왜?



다중값 종속성 Dependencies

§ R을 관계 스키마라고 하고 $a \twoheadrightarrow b$ 및 $b \twoheadrightarrow a$ 라고 합니다.

의존

$\twoheadrightarrow b$

$t1[a] = t2[a]$ 인 r 의 튜플 $t1$ 및 $t2$ 에 대한 모든 쌍에 대해 임의의 법칙 관계 $r(R)$ 에서 r 에 튜플 $t3$ 및 $t4$ 가 다음과 같이 존재하는 경우 R 을 유지합니다.

$t1[a] = t2[a] = t3[a] = t4[a]$

$t3[b] = t1[b] \quad t3[R$

$- b] = t2[R - b] \quad t4[b]$

$= t2[b] \quad t4[R - b]$

$= t1[R - b]$

inst_info(ID, child_name, phone_number)

Example data:

(99999, David, 512-555-1234)

(99999, William, 512-555-4321)

(99999, William, 512-555-1234)

(99999, David, 512-555-4321)



MVD-표 형식 표현 representation

§ $a \multimap b$ 의 표 표현

	α	β	$R - \alpha - \beta$
t_1	$a_1 \dots a_i$	$a_{i+1} \dots a_j$	$a_{j+1} \dots a_n$
t_2	$a_1 \dots a_i$	$b_{i+1} \dots b_j$	$b_{j+1} \dots b_n$
t_3	$a_1 \dots a_i$	$a_{i+1} \dots a_j$	$b_{j+1} \dots b_n$
t_4	$a_1 \dots a_i$	$b_{i+1} \dots b_j$	$a_{j+1} \dots a_n$

$$t_1[\alpha] = t_2[\alpha] = t_3[\alpha] = t_4[\alpha]$$

$$t_3[\beta] = t_1[\beta]$$

$$t_3[R - \beta] = t_2[R - \beta]$$

$$t_4[\beta] = t_2[\beta]$$

$$t_4[R - \beta] = t_1[R - \beta]$$



MVD(계속) (nt.)

§ R 을 3개의 비어 있지 않은 하위 집합으로 분할된 속성 집합이 있는 관계 스키마라고 합니다.

Y, Z, W

§ 우리는 $Y \multimap Z$ (Y multitermines Z)라고 말합니다

$$\langle y_1, z_1, w_1 \rangle \in r \text{ 및 } \langle y_1, z_2, w_2 \rangle \in r$$

그 다음에

$$\langle y_1, z_1, w_2 \rangle \in r \text{ 및 } \langle y_1, z_2, w_1 \rangle \in r$$

§ Z 와 W 의 동작이 동일하므로 다음과 같습니다.

$Y \multimap Z$ 만약 $Y \multimap W$

inst info(ID, child name, phone number)

Example data:

(99999, David, 512-555-1234)

(99999, David, 512-555-4321)

(99999, William, 512-555-1234)

(99999, William, 512-555-4321)



예

§ 우리의 예에서:

아이디^{@@} child_name

ID^{@@} 전화번호

§ 위의 정식 정의는 Y(ID)의 특정 값이 주어지면 Z(자식 이름) 값 집합과 W(전화 번호) 값 집합과 연결된다는 개념을 공식화하기로 되어 있습니다. 이 두 집합은 어떤 의미에서 서로 독립적입니다.

§ 참고:

- $Y^@ Z$ 이면 $Y^{@@} Z$
- 실제로 우리는 (위 표기법에서) $Z1 = Z2$ 를 가집니다. 주장은 다음과 같습니다.



§ $R(A,B,C)$

§ $A \rightarrow B$

§ $(a,b1,c1), (a,b2,c2), (a,b3,c3)$

$(a,b1,c2) (a,b1,c3) (a,b2, c1) (a,b2, c3) (a,b3,c1) (a,b3,c2)$



다중값 종속성 사용

§ 다음 두 가지 방법으로 다중값 종속성을 사용합니다.

1. 주어진 관계에서 합법적인지 여부를 결정하기 위해 관계를 테스트합니다.
기능 및 다중값 종속성 집합
2. 법적 관계 집합에 대한 제약 조건을 지정합니다. 우리는 주어진 기능적 및 다중값 종속성을 만족시키는 관계에만 관심을 기울일 것입니다.

§ 관계 r 이 주어진 다중값 종속성을 충족하지 못하면 다음을 수행할 수 있습니다.
 r 에 튜플을 추가하는 관계 r' 다음과 같이 다중값 종속성을 충족합니다.
관계 r 를 구성합니다.



MVD 이론

§ 다중값 종속성의 정의에서 다음 규칙을 도출할 수 있습니다.

- $a \bowtie b$ 이면 $a \bowtie\bowtie b$

즉, 모든 기능 종속성은 다중값 종속성이기도 합니다.

§ D 의 클로저 D^+ 는 모든 기능 및 다중값의 집합입니다.

D 에 의해 논리적으로 암시된 종속성. • 함수적 종속

성과 다중값 종속성의 공식 정의를 사용하여 D 에서 D^+ 를 계산할 수 있습니다.

- 매우 간단한 다중값 종속성에 대한 이러한 추론으로 관리할 수 있습니다. 이는 실제로 가장 일반적으로 보입니다. • 복잡한 종속성의 경우 추론 규칙 시스템을 사용하여 종속성 집합에 대해 추론하는 것이 좋습니다(부록 C).



네 번째 정규형 Normal Form

§ 관계 스키마 R은 기능 및

$a \twoheadrightarrow b$ 형식의 D^+ 에서 모든 다중값 종속성에 대해 $a \in R$ 및 $b \in R$ 인 경우 다음 중 적어도 하나가 유지됩니다. $a \twoheadrightarrow b$ 는 사소한(즉, $b \twoheadrightarrow a$) a 는 스키마 R의 수퍼키

-

-

§ 관계가 4NF에 있으면 BCNF에 있음



다중값 종속성의 제한 Multivalued Dependencies

§ R_i 에 대한 D의 제한은 다음으로 구성된 집합 D_i 입니다.

- R_i 의 속성만 포함하는 D^+ 의 모든 기능적 종속성
- 형식의 모든 다중값 종속성

$a \twoheadrightarrow (b \in R_i)$ 여기서

$a \in R_i$ 및 $a \twoheadrightarrow b$ 는 D^+ 에 있음



4NF 분해 알고리즘 Decomposition Algorithm

결과: = {R}; 완

료 := 거짓; **D+ 계산** ;

D_i 가 R_i 에 대한 D^+ 의 제한을 나타내도록 하고 (not

done) if (4NF에 없는

스키마 R_i 가 결과에 있는 경우) then

시작하다

$a \twoheadrightarrow b$ 는 $a \twoheadrightarrow R_i$ 가 D_i 에 없고 $a \not\rightarrow b$ 가 되도록 R_i 를 유지하는 중요하지 않은 다중값 종속성
입니다. 결과 := (결과 - R_i) \cup ($R_i - b$) \cup (a, b); 종료 완료:= 참;

참고: 각 R_i 는 4NF이며 분해는 무손실 조인입니다.



예 kample

$\Sigma R = (A, B, C, G, H, I)$

$F = \{ A \twoheadrightarrow B$

$B \twoheadrightarrow H \}$

ΣR

은 $A \twoheadrightarrow B$ 및 A 가 R 의 슈퍼키가 아니므로 4NF에 속하지 않습니다. Σ 분해 a) $R_1 = (A, B)$ b)

$R_2 = (A, C, G, H, I)$ c) R_3

$= (C, G, H)$ d) R_4

$= (A, C, G, I)$

$\bullet A \twoheadrightarrow B$ 및 $B \twoheadrightarrow H$

대한 MVD 제한) e) $R_5 = (A, I)$

음) $R_6 = (A, C, G)$

- A relation schema R is in **4NF** with respect to a set D of functional and multivalued dependencies if for all multivalued dependencies in D^+ of the form $\alpha \twoheadrightarrow \beta$, where $\alpha \subseteq R$ and $\beta \subseteq R$, at least one of the following hold:
 - $\alpha \twoheadrightarrow \beta$ is trivial (i.e., $\beta \subseteq \alpha$ or $\alpha \cup \beta = R$)
 - α is a superkey for schema R
- If a relation is in 4NF it is in BCNF

그리고 $A \twoheadrightarrow H$, (MVD 이행성) 및 \bullet 따라서 $A \twoheadrightarrow I$ (R_4 에

(R_1 은 4NF에 있음)

(R_2 는 4NF가 아니므로 R_3 과 R_4 로 분해)

(R_3 는 4NF에 있음)

(R_4 는 4NF가 아니므로 R_5 와 R_6 으로 분해)

(R_5 는 4NF에 있

(R_6 은 4NF에 있음)



추가 정규형 Normal Forms

§ 결합 종속성은 다중 값 종속성을 일반화합니다.

- PJNF(Project-Join Normal Form) 로 이어짐 (제5 정규형 이라고도 함) 형태)

§ 훨씬 더 일반적인 제약 조건 클래스는
도메인 키 일반 형식.

§ 이러한 일반화된 제약 조건의 문제점: 추론하기 어렵고 건전하고 완전한 추론 규칙 집합이 존재하지 않습니다.

§ 따라서 거의 사용되지 않음



전체 데이터베이스 설계 프로세스

우리는 스키마 R이 주어진다고 가정했습니다.

§ ER 다이어그램을 집합으로 변환할 때 R이 생성되었을 수 있음 테이블의.

§ R은 다음에 해당하는 모든 속성을 포함하는 단일 관계일 수 있습니다.
관심(보편적 관계라고 함).

§ 정규화는 R을 더 작은 관계로 나눕니다.

§ R은 관계의 임시 설계의 결과일 수 있으며, 그런 다음 정규 형식으로 테스트/변환합니다.



ER 모델 및 정규화 Normalization

§ 모든 엔터티를 식별하는 ER 다이어그램을 신중하게 설계한 경우
올바르게, ER 다이어그램에서 생성된 테이블은 추가 정규화가 필요하지 않습니다.

§ 그러나 실제(불완전한) 설계에서는 기능적
엔터티의 키가 아닌 속성에서 엔터티의 다른 속성으로의 종속성 • 예: 직원 엔터티

§ 속성 부서_이

름 및 건물, § 기능적 종속성 부서_이름 건

물 § 좋은 설계는 부서를 엔터티로

만들 수 있음 § 관계 집합의 키가 아닌

속성으로부터의 기능적 종속성은 가능하지만 드물지만 --- 대부분의 관계는 이

진 관계입니다.



성능을 위한 비정규화 Normalization for Performance

§ 성능을 위해 정규화 되지 않은 스키마를 사용하고자 할 수 있습니다 .

물론 prereq와 함께 가입

§ 대안 1: 과정의 속성을 포함하는 비정규화된 관계 사용

위의 모든 속성을 가진 prereq 뿐만 아니라

• 더 빠른 조회 • 업

데이트를 위한 추가 공간 및 추가 실행 시간 • 프로그래머를 위한

추가 코딩 작업 및 추가 오류 가능성

암호

§ 대안 2: 과정 선행 조건으로 정의된 구체화된 보기 사용

✕

• 장단점은 프로그래머가 별도의 코딩 작업을 할 필요가 없고 가능한 오류를 피할 수 있다는 점을 제외하면 위와 동일합니다.



기타 설계 문제 Other Design Issues

§ 데이터베이스 디자인의 일부 측면은 정규화로 포착되지 않습니다. § 피해야 할 잘못된 데이터베이스 디자인의 예:

수입(company_id, year, amount) 대신 다음을 사용하십시오.

- income_2004, income_2005, income_2006 등 모두 스키마(company_id, 수입).

§ 위의 내용은 BCNF에 있지만 연도별 쿼리가 어렵고 매년 새 테이블이 필요함

- company_year(company_id, income_2004, income_2005, income_2006)

§ BCNF에서도 마찬가지

가지이지만 연도 간 쿼리가 어렵고 매년 새 속성이 필요합니다.

§ 하나의 속성에 대한 값이 있는 교차 분석의 예입니다.
열 이름이 됨

§ 스프레드시트 및 데이터 분석 도구에서 사용



임시 데이터 모델링 Temporal Data

§ 임시 데이터에는 데이터가 연결되는 연결 시간 간격이 있습니다.
유효합니다.

§ 스냅 샷은 특정 시점의 데이터 값입니다. § 유효한 시간을 추가하여 ER 모델을 확장하기 위한 몇 가지 제안

- 속성(예: 서로 다른 시점의 강사 주소) • 개체(예: 학생 개체가 존재하는 기간) • 관계(예: 강사가 관련된 시간)

학생을 고문으로.

§ 그러나 승인된 표준이 없음

§ 시간적 구성 요소를 추가하면 다음과 같은 기능적 종속성이 발생합니다.

ID^o 거리, 도시

시간이 지남에 따라 주소가 달라지기 때문에 보유하지 않음

§ 시간적 기능 종속성 $X \twoheadrightarrow Y$ 는 다음과 같은 경우 스키마 R에 유지됩니다.

기능 종속성 $X \twoheadrightarrow Y$ 는 모든 적절한 인스턴스 $r(R)$ 에 대한 모든 스냅샷을 보유합니다.