

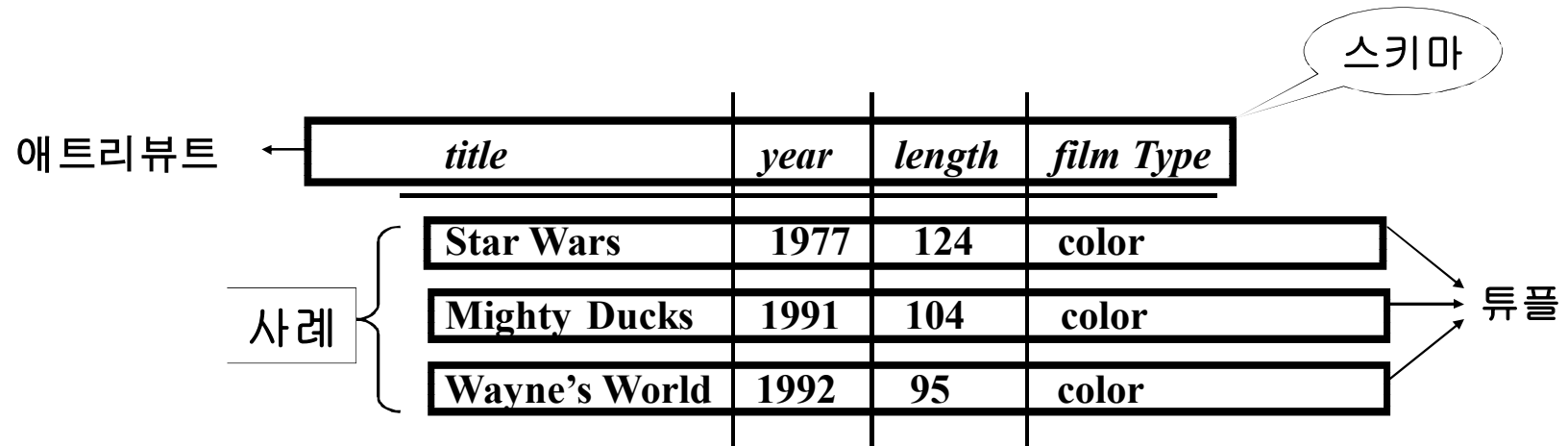
3장 . 관계 데이터 모델

- ◆ 관계 모델(**relational model**)의 기본
- ◆ E/R 다이어그램에서 관계 설계로의 변환
- ◆ 서브클래스 구조를 릴레이션으로 변환
- ◆ 함수적 종속성(**functional dependency**)
- ◆ 함수적 종속성에 관한 규칙들
- ◆ 관계 데이터베이스 스키마 설계
- ◆ 다중값 종속성(**multivalued dependency**)



관계 모델의 기본

◆ 릴레이션 Movie



관계 모델의 기본 (계속)

- ◆ 릴레이션 (사례:instance) : 2차원 테이블
 - 요소들이 원자적 값을 가지는 튜플들의 집합
- ◆ 애트리뷰트 : 릴레이션의 각 열에 대한 이름
- ◆ 튜플(tuple) : 릴레이션의 행
- ◆ 스키마 : 릴레이션의 이름과 릴레이션의 애트리뷰트들의 집합

[예] `Movie(title, year, length, filmType)`

- ◆ 도메인 : 릴레이션의 각 애트리뷰트에 연관된 타입

- ☞ 스키마와 사례(instance) : 스키마는 릴레이션에 대한 애트리뷰트들의 이름이며, 사례는 릴레이션에 대한 튜플들의 집합을 나타내는 것이다.
 - 사례는 시간의 흐름에 따라서 그 내용이 변한다.
 - 스키마는 거의 변하지 않는다.

관계 모델의 기본 (계속)

acctNo	type	balance
12345	Savings	12000
23456	Checking	2000
34567	Savings	5100

Accounts 릴레이션

firstName	lastName	idNo	account
Robbie	Banks	901-222	12345
Lena	Hand	805-333	12345
Lena	Hand	805-333	23456

Customers 릴레이션

관계 모델의 기본 (계속)

◆ 튜플의 예

- **Movie** 릴레이션의 한 사례

[예] (**Star Wars**, 1977, 124, color) : 각 애트리뷰트에 해당하는 값은 스키마에서 정의한 순서로 가정한다.

◆ OO, E/R, 관계 모델의 관계

- **Object = Entity = Tuple = Record(Structure)**
- **Class = Relation Schema**
- **Entity Set = Relation(Table) = File**
- **Attribute = Field**

◆ Single-Value Constraints

- 관계 모델에서 모든 값은 단일값을 가진다고 가정한다.

E/R에서 관계 모델로의 변환

◆ 엔티티 집합을 릴레이션으로 변환

- 강(**strong**) 엔티티 집합에 대하여 동일한 애트리뷰트 들을 갖는 같은 이름의 릴레이션을 생성한다.

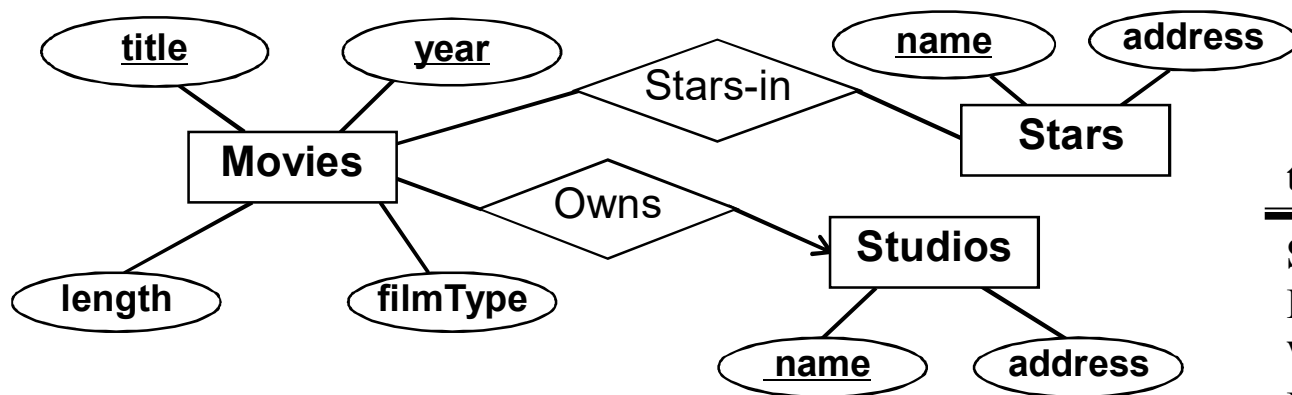
[예] **Movies(title,year,length,filmType), Studios(name,address)**

◆ 관계성을 릴레이션으로 변환

- 관계성 **R**에 연관된 엔티티 집합들의 키 애트리뷰트들을 **R**에 대한 릴레이션 스키마의 애트리뷰트들로 만든다.
- 관계성이 애트리뷰트를 가지면, 이 애트리뷰트도 릴레이션 **R**의 애트리뷰트로 만든다.

[예] **Stars-In(title,year,starName), Owns(title,year,studioName)**

E/R에서 관계 모델로의 변환 (계속)



title	year	studioName
Star Wars	1977	Fox
Mighty Ducks	1991	Disney
Wayne's World	1992	Paramount
Little Mermaid	1993	Disney

관계성 **Owns** 에 대한 릴레이션

title	year	starName
Star Wars	1977	Carrie Fisher
Star Wars	1977	Mark Hamill
Star Wars	1977	Harrison Ford
Mighty Ducks	1991	Emilio Estevez
Wayne's World	1992	Dana Carvey
Wayne's World	1992	Mike Meyers

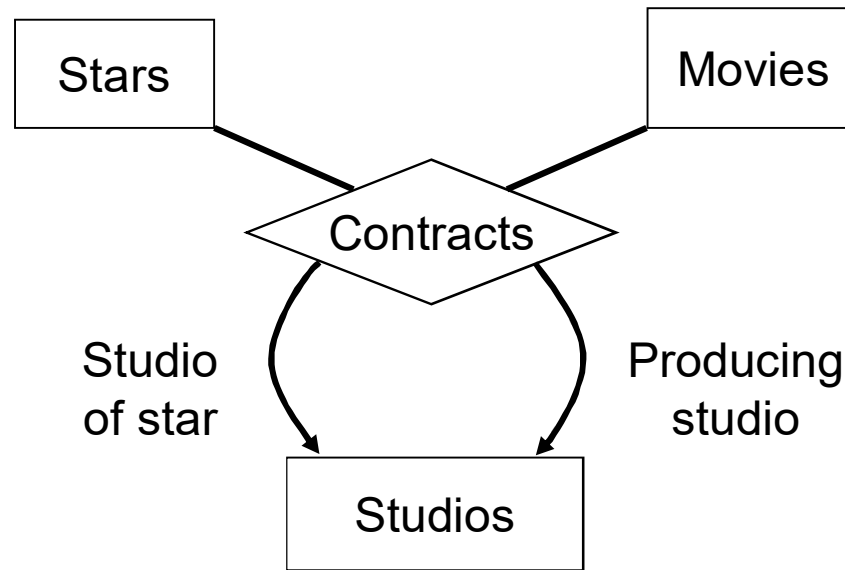
관계성 **Stars-in** 에 대한 릴레이션

name	address
Carrie Fisher	123 Maple St., Hollywood
Mark Hamill	456 Oak Rd., Brentwood
Harrison Ford	789 Palm Dr., Beverly Hills

엔티티 **Stars** 에 대한 릴레이션



E/R에서 관계 모델로의 변환 (계속)

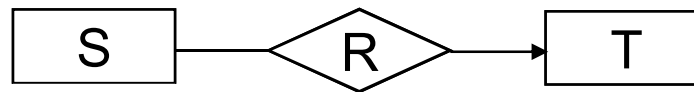


`Contracts(studioOfStar, producingStudio, starName, title, year)`

릴레이션들의 결합 (계속)

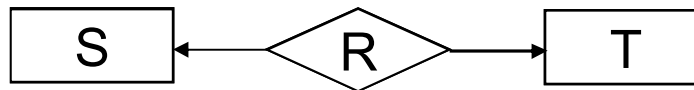
- 어떤 관계성에 대해서는 테이블을 생성할 필요가 없다.

- ◆ 다대일 관계성



- R에 대한 테이블을 생성하는 대신, S에 R의 모든 애트리뷰트들과 T의 키를 포함시켜도 된다.

- ◆ 일대일 관계성



- R에 대한 테이블을 생성하는 대신, S에 R의 모든 애트리뷰트와 T의 키를 포함시켜도 된다. 그 역도 가능하다.
- R에 대한 테이블을 생성하는 대신, 모든 것을 하나의 테이블로 표현해도 된다.

릴레이션들의 결합 (계속)

title	year	length	filmType	studioName
Star Wars	1977	124	color	Fox
Mighty Ducks	1991	104	color	Disney
Wayne's World	1992	95	color	Paramount

Movies와 Owns
릴레이션의 결합

title	year	length	filmType	studioName	starName
Star Wars	1977	124	color	Fox	Carrie Fisher
Star Wars	1977	124	color	Fox	Mark Hamill
Star Wars	1977	124	color	Fox	Harrison Ford
Mighty Ducks	1991	104	color	Disney	Emilio Estevez
Wayne's World	1992	95	color	Paramount	Dana Carvey
Wayne's World	1992	95	color	Paramount	Mike Meyers

Movies와 Owns, Stars-in 릴레이션의 결합

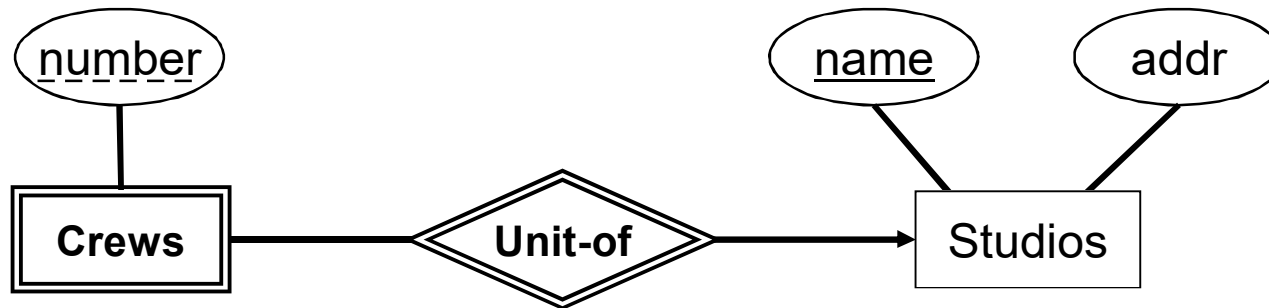


E/R에서 관계 모델로의 변환 (계속)

◆ 약 엔티티 집합의 변환

- 약 엔티티 집합 **W**에 대한 릴레이션은, **W**의 애트리뷰트들뿐만 아니라 다대일 관계성에 의해서 **W**와 연관되어 있는 **W**의 키를 형성하는데 도움을 주는 다른 엔티티 집합들(helping entity sets)의 키 애트리뷰트까지 포함해야 한다.
- **W**와 다른 엔티티 집합들을 연결시켜 주는 다대일 관계성에 대해서는 릴레이션을 생성하지 않는다.

E/R에서 관계 모델로의 변환 (계속)



Crews (number, studioName)

Studios (name, addr)

number	studioName
1	Fox
2	Fox
3	Fox
4	Fox
2	Disney
3	Disney
2	Paramount

Unit-Of 릴레이션의 사례

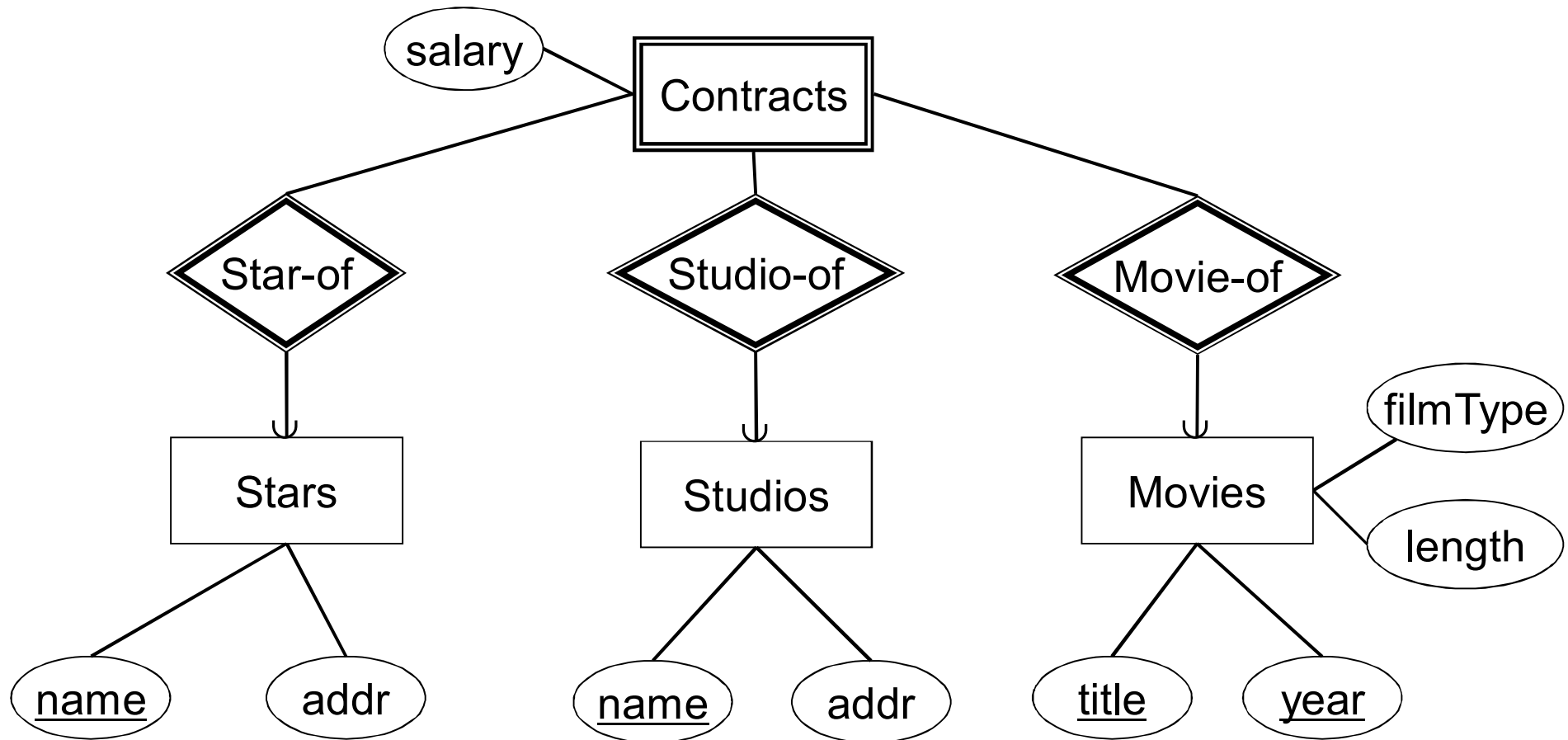
◆ 만일 Unit-of를 릴레이션으로 만든다면 ?

- Unit-of(number, studioName, studioName') : studioName과 studioName'는 동일하므로 하나로 표현해도 상관없으므로 ..

[예] (3, Disney, Disney), (5, Fox, Fox) ...

» Unit-of(number, studioName)으로 변형됨 : Crews 릴레이션과 동일하다.

E/R에서 관계 모델로의 변환 (계속)



Contracts (starName, studioName, title, year, salary)

서브클래스 구조를 릴레이션으로 변환

◆ E/R 모델에서의 서브클래스

– 가정들

- » 계층 구조에 루트 엔티티 집합이 있다.
- » 루트 엔티티 집합에는 키가 있다.
- » 특정 서브 엔티티 집합은 상위 엔티티 집합에 이르는 콤포넌트를 지닐 수 있다.

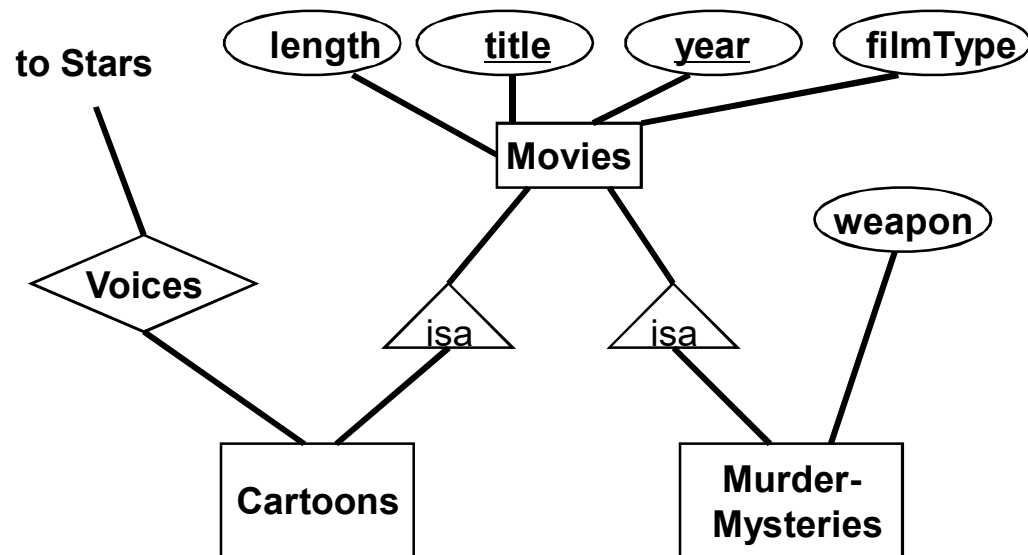
- E/R 모델에서 하나의 객체는 isa 관계성으로 연관된 여러 엔티티 집합에 속하는 엔티티들에 의해 표현된다. 연결되어 있는 엔티티들이 모여 하나의 객체를 표현한다.

① E/R 스타일 변환

◆ E/R 모델의 isa 계층 구조를 릴레이션 스키마로 변환

- 각 엔티티 집합에 대해 그 엔티티 집합에 정의된 모든 애트리뷰트를 포함하는 릴레이션을 만든다.
- 각 서브클래스에 대한 릴레이션에는 슈퍼클래스의 키 애트리뷰트를 포함시킨다.
- **ISA** 관계성은 각 서브클래스에 이미 표현되어 있으므로 만들 필요가 없다.

E/R 서브클래스를 릴레이션으로 표현(계속)



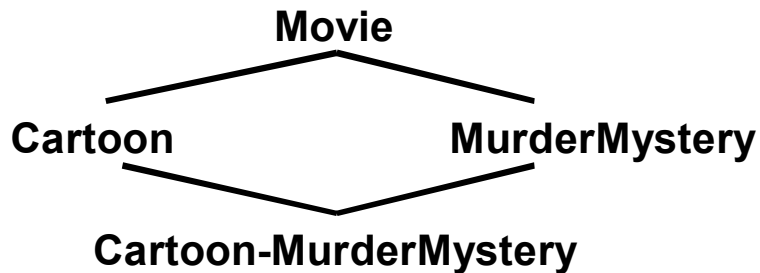
- ◆ 무성만화영화는 어떻게 표현되는가?
Voices 관계성에 starName이 없으므로 NULL로 표현할 수 있다.

```
Movies(title, year, length, filmType)
MurderMysteries(title, year, weapon)
Cartoons(title, year)
Voices(title, year, starName)
```


② 객체지향 접근 방법

◆ 서브클래스 계층 구조를 릴레이션 스키마로 변환

- 각 서브클래스는 그 클래스에 정의된 모든 특성과 상속 받는 모든 특성을 포함하는 하나의 릴레이션으로 변환된다.
- 엔티티는 특정 클래스 하나에만 존재한다.



- Cartoon은 Star 클래스에 대한 관계성 voice를 가진다.
- MurderMystery는 애트리뷰트 weapon을 가진다.

```
Movie(title,year,length,filmType,studioName,starName)
Cartoon(title,year,length,filmType,studioName,starName,voice)
MurderMystery(title,year,length,filmType,studioName,starName,weapon)
Cartoon-MurderMystery(title,year,length,filmType,studioName,starName,voice,weapon)
```

③ 클래스 계층구조의 결합에 NULL값 사용

■ NULL 값

- 튜플의 애트리뷰트에 대해서 적절한 값이 없는 경우에 사용한다.

◆ 클래스 계층에 대해서 하나의 릴레이션을 생성

- 하나의 릴레이션이 클래스 계층에 있는 모든 특성들에 대한 애트리뷰트들을 가진다.
- 한 객체는 하나의 튜플로 표현되며, 이 튜플은 그 객체의 클래스에 속하지 않는 특성에 대한 애트리뷰트에는 **NULL** 값을 가진다.

`Movie(title, year, length, filmType, studioName, starName, voice, weapon)`

[예 1] 영화 'Who Framed Roger Rabbit?' 은 NULL이 없다.

[예 2] 영화 'The Little Mermaid' 는 weapon이 NULL이다.

[예 3] 영화 'Gone with the Wind' 는 voice와 weapon이 모두 NULL이다.

여러 방식들의 비교 – 질의 처리 비용

◆ 객체지향 변환

- 한 영화의 모든 특성이 한 릴레이션에 모두 있다.
- 하나의 영화를 찾고자 할 때 여러 릴레이션들을 탐색해야 한다.

[예] “**Roger Rabbit**” 영화가 어느 릴레이션에 포함되어 있는가? 즉,
`Movie`, `Cartoon`, `MurderMystery`, `Cartoon-MurderMystery`
중 어느 릴레이션에 포함되어 있는가?

◆ E/R 변환

- 각 영화에 대한 키를 각 영화가 속하는 엔티티 집합들과 관계성들에 대해 반복한다.
 - » 해당 슈퍼 클래스에 대한 키가 반복된다: 공간 낭비
 - » 한 객체에 대한 정보를 얻기 위해 여러 릴레이션들을 찾아 보아야 한다.
- [예] 살인 미스터리 영화 ‘A’의 `length`와 `weapon`을 찾아라”

여러 방식들의 비교 - 공간 사용

◆ 많은 릴레이션을 생성

- E/R 변환 : 다중 상속성의 표현을 위한 릴레이션을 생성 안한다.
- 객체지향 방법 : 다중 상속성의 표현을 위해 더 많은 서브 릴레이션 생성
- 널값 사용 : 하나의 릴레이션만 생성하므로 유리

◆ 저장 공간의 최소화

- E/R 변환 : 키 애트리뷰트들의 반복, 상황에 따라서 다름
- 객체지향 방법 : 각 영화에 대해서 단 하나의 튜플만 저장하므로 공간 절약
- 널값 사용 : 해당 애트리뷰트를 사용하지 않는 튜플이 많을 수록 많은 공간 낭비

함수적 종속성

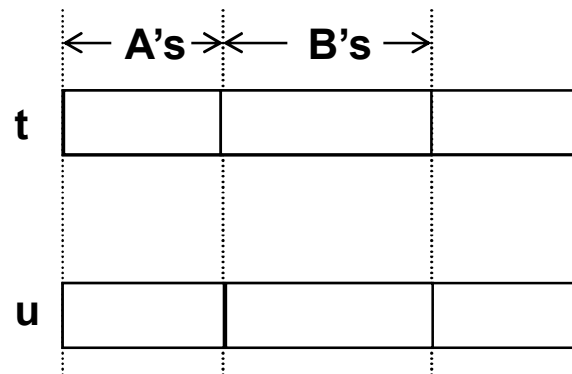
◆ 함수적 종속성(Functional Dependency)의 정의

- 릴레이션 R 에 A_1, A_2, \dots, A_n 과 B 라는 애트리뷰트가 있다고 하자.
- “ R 의 두 튜플이 A_1, A_2, \dots, A_n 에 대해 동일한 값을 가지면, B 에 대해서도 동일한 값을 가져야 한다.”
- $A_1A_2 \dots A_n \rightarrow B$: “ A_1, A_2, \dots, A_n 은 B 를 함수적으로 결정(determine)한다.”라고 말한다.
 - » 애트리뷰트의 집합 A_1, A_2, \dots, A_n 이 여러 애트리뷰트 B_1, B_2, \dots, B_m 을 함수적으로 결정하면,

$$A_1A_2 \dots A_n \rightarrow B_1B_2 \dots B_m$$

함수적 종속성(계속)

◆ 두 튜플에 대한 함수적 종속성의 의미



t와 u가
여기서
동일하다면,
여기에서도
반드시 동일해야
한다.

title year → length

title year → filmType

title year → studioName

title year → length filmType studioName

title year → starName (false) : many-to-many 관계

• 왜 “functional(함수적)”인가 ???

- length(Star Wars, 1977) = 124

- filmType(Star Wars, 1977) = color

함수적 종속성의 예

A	B	C	D
a ₁	b ₁	c ₁	d ₁
a ₁	b ₂	c ₁	d ₂
a ₂	b ₂	c ₂	d ₂
a ₂	b ₃	c ₂	d ₃
a ₃	b ₃	c ₂	d ₄

$A \rightarrow C$

$AB \rightarrow D$

$A \rightarrow A, AB \rightarrow A$

$ABCD \rightarrow A$

$AC \rightarrow D$ (?)

$BD \rightarrow A$ (?)

$A \rightarrow B$ (?)

함수적 종속성과 키

◆ 릴레이션의 키

- 애틀리뷰트들의 집합 $\{A_1, A_2, \dots, A_n\}$ 이 다음을 만족하면 릴레이션의 키가 된다.
 - ① 애틀리뷰트 A_1, A_2, \dots, A_n 는 릴레이션 R 에 있는 다른 모든 애틀리뷰트들을 함수적으로 결정한다. 즉, A_1, A_2, \dots, A_n 의 값에 대해서 동일한 값을 가지는 두 튜플이 존재할 수 없다. : **Uniqueness**
 - ② $\{A_1, A_2, \dots, A_n\}$ 의 어떤 진부분 집합도 R 의 모든 나머지 애틀리뷰트들을 함수적으로 결정할 수 없다. 즉 키는 최소(minimal)의 애틀리뷰트들로 이루어진다. : **Minimality**

[예] **Movie (title, year, length, filmType, studioName, starName)**

- » **$\{title, year, starName\}$** 이 키가 된다. : **$\{title, year\}$ 만으로는 안된다!!!**
- 키가 여러 개 존재하면 이 중 하나를 주(primary) 키로 명시한다.

수퍼키

◆ 수퍼키(superkey)

- 키를 포함하고 있는 애트리뷰트의 집합
 - » 키의 수퍼 집합(super set)
- 릴레이션의 다른 모든 애트리뷰트를 함수적으로 결정한다. 그러나 최소성(minimality)을 만족할 필요는 없다.
 - » {title, year, starName}은 수퍼키이다.
 - » {title, year, starName, length}도 수퍼키이다.

E/R로부터 변환된 릴레이션의 키

◆ 엔티티 집합에서 변환된 릴레이션

- 엔티티 집합의 키 애트리뷰트가 릴레이션의 키가 된다.

- » **Movie(title, year, length, filmType)**

- » **Stars(name, address)**

◆ 이진 관계성에서 변환된 릴레이션 **R**

- 다대다 관계성: 연결된 두 엔티티 집합 양쪽의 키가 함께 모여 **R**의 키 애트리뷰트가 된다.

- » **Stars-In(title, year, starName) : 다대다 관계**

E/R로부터 변환된 릴레이션의 키 (계속)

◆ 이진 관계성에서 변환된 릴레이션 R

① 엔티티 집합 E_1 으로부터 엔티티 집합 E_2 로의 다대일 관계성

E_1 의 키 애트리뷰트들만이 R의 키 애트리뷰트가 된다.

[예] **Owns(title, year, studioName)** : Movies와 Studios 사이 다대일 관계

② 일대일 관계성: 연결된 엔티티 집합 중 어느 한쪽의 키 애트리뷰트도 R의 키 애트리뷰트가 될 수 있다.

[예] **Runs(studioName, presidentName)** : Studios와 Presidents 사이 일대일 관계

❖ E/R에서 다대일 관계성과 일대일 관계성에 대해서는 관계성이 애트리뷰트를 가지고 있지 않으면 릴레이션을 생성할 필요가 없다.

함수적 종속성에 관한 규칙들

- ◆ 분할(splitting)/결합(combining) 규칙
- ◆ 단순(trivial) 종속성
- ◆ Armstrong의 공리(axiom)
- ◆ 애트리뷰트의 총합(closure)

함수적 종속성의 추론

◆ 추론 규칙(inference rule)

– 한 릴레이션이 만족하는 몇 개의 **FD**들이 있을 때 다른 **FD**들을 추론

» $A \rightarrow B, B \rightarrow C$ 가 만족된다면 $A \rightarrow C$ 도 만족함을 추론

◆ 두 튜플 $(a, b_1, c_1), (a, b_2, c_2)$ 가 $A \rightarrow B$ 를 만족 : $b_1=b_2$ 가 되어야 함

◆ 따라서 두 튜플은 $(a, b, c_1), (a, b, c_2)$ 가 되어야 하며 $B \rightarrow C$ 를 만족해야 하므로 $c_1=c_2$ 가 되어야 한다

– 즉, 두 튜플은 (a, b, c) 로 동일하다 : $A \rightarrow C$ 가 만족됨

◆ 함수적 종속성의 동등 (equivalent) 관계

– 릴레이션 **R**에 성립하는 **FD**들의 집합을 각각 **S**와 **T**라 하자

» **T**의 모든 **FD**들이 **R**에 대한 모든 instance들에 성립할 때, **S**의 모든 **FD**도 동일한 instance들에 성립하면 : **S**는 **T**로 부터 유도(follow)된다고 한다

» **S**와 **T**는 동등 : **S**가 **T**로 부터 유도되고 **T**도 **S**로 부터 유도된 경우..

분할/결합 규칙

◆ 분할 규칙(splitting rule)

- $A_1A_2 \dots A_n \rightarrow B_1B_2 \dots B_m$ 형태의 함수적 종속성은 $A_1A_2 \dots A_n \rightarrow B_i$ ($i = 1, 2, \dots, m$) 인 종속성들의 집합으로 대체할 수 있다.

◆ 결합 규칙(combining rule)

- $A_1A_2 \dots A_n \rightarrow B_i$ ($i = 1, 2, \dots, B_m$)인 종속성 집합들을 하나의 함수적 종속성 $A_1A_2 \dots A_n \rightarrow B_1B_2 \dots B_m$ 으로 대체할 수 있다.

[예 1] $\text{title year} \rightarrow \text{length filmType studioName}$

$\Leftrightarrow \text{title year} \rightarrow \text{length}, \text{title year} \rightarrow \text{filmType}, \text{title year} \rightarrow \text{studioName}$

[예 2] $\text{title year} \rightarrow \text{length} \Leftrightarrow \text{title} \rightarrow \text{length}, \text{year} \rightarrow \text{length} (X)$

단순 종속성

◆ 종속성 $\alpha \rightarrow \beta$ 은

- $\alpha \supseteq \beta$ 이면, 단순(trivial) 종속성이다 : $ABC \rightarrow BC$
- $(\beta - \alpha) \neq \emptyset$ 이면, 비단순(nontrivial) 종속성이다 : $ABC \rightarrow BCD$
- $\alpha \cap \beta = \emptyset$ 이면, 완전 비단순(completely nontrivial) 종속성이다 : $AB \rightarrow CD$

◆ 단순 종속성 규칙

- 우변에는 있으나 좌변에는 없는 애트리뷰트들을 $C_1C_2 \dots C_k$ 라고 하면, 함수적 종속성 $A_1A_2 \dots A_n \rightarrow B_1B_2 \dots B_m$ 은 $A_1A_2 \dots A_n \rightarrow C_1C_2 \dots C_k$ 와 동등(equivalent)하다. : $\alpha \rightarrow \beta \Leftrightarrow \alpha \rightarrow \beta - \alpha$ (if $\alpha \cap \beta \neq \emptyset$)
 - » $ABCD \rightarrow CDEF \Leftrightarrow ABCD \rightarrow EF, \text{ ~~ABCD} \rightarrow \text{CD (trivial FD)}~~$

애트리뷰트의 총합

- ◆ 애트리뷰트의 집합 $\{A_1, A_2, \dots, A_n\}$ 과 함수적 종속성 집합 S 가 있을 때, S 의 종속성을 기반으로 한 $\{A_1, A_2, \dots, A_n\}$ 의 총합(closure)은 집합 S 의 모든 종속성을 만족하는 릴레이션이 $A_1 A_2 \dots A_n \rightarrow B$ 도 만족하는 모든 애트리뷰트들의 집합인 B 이다.

» 총합에는 단순 종속성도 허용된다.

- $\{A_1, A_2, \dots, A_n\}$ 의 총합을 $\{A_1, A_2, \dots, A_n\}^+$ 로 표시한다.

[예] **Movie (title, year, studioName, studioAddr)**

함수적 종속성: **title year \rightarrow studioName**

studioName \rightarrow studioAddr

**$\{title, year\}^+$: $\{title, year\} \Rightarrow \{title, year, studioName\} \Rightarrow$
 $\{title, year, studioName, studioAddr\}$**

transitive rule에 의해
title year \rightarrow studioAddr이
성립

애트리뷰트의 총합(계속)

◆ 총합 계산

1. X 를 최종적으로 얻게 되는 답이라고 하자. X 의 초기 상태를 $\{A_1, A_2, \dots, A_n\}$ 로 설정한다.
 2. B_1, B_2, \dots, B_m 은 애트리뷰트 집합 X 에 존재하며 C 는 그렇지 않은 함수적 종속성 $B_1 B_2 \dots B_m \rightarrow C$ 를 찾는다. 이러한 함수적 종속성이 존재할 경우 C 를 집합 X 에 추가한다.
 3. X 집합에 새로운 애트리뷰트가 추가되지 않을 때까지 단계 2를 반복한다.
 - 더 이상의 애트리뷰트 추가가 없을 때의 집합 X 가 $\{A_1, A_2, \dots, A_n\}^+$ 가 된다.
- ☞ X 는 단지 증가할 수만 있으며 릴레이션에 있는 애트리뷰트들의 수는 유한하므로 단계 2가 무한히 반복되는 일은 없다.

애트리뷰트의 총합 (계속)

[예] 릴레이션: $R(A, B, C, D, E, F)$

함수적 종속성: $AB \rightarrow C, BC \rightarrow AD, D \rightarrow E, CF \rightarrow B$

$\{A, B\}^+ : \{A, B\} \Rightarrow \{A, B, C\} \Rightarrow \{A, B, C, D\} \Rightarrow \{A, B, C, D, E\}$

$AB \rightarrow C \quad BC \rightarrow AD \quad D \rightarrow E$

◆ $A_1A_2 \dots A_n \rightarrow B$ 가 함수적 종속성의 집합 S 로부터 유도되는(follow) 지를 검사

- $\{A_1, A_2, \dots, A_n\}^+$ 를 계산
- B 가 $\{A_1, A_2, \dots, A_n\}^+$ 에 있으면 $A_1A_2 \dots A_n \rightarrow B$ 이다.

(예) 위의 예제에서 $AB \rightarrow D$ 가 유도되는 지를 검사해 보자.

- $\{A, B\}^+$ 가 D 를 포함하므로 $AB \rightarrow D$ 는 유도가능 하다.

총합과 키

◆ 총합 알고리즘의 증명

- **soundness** : **closure algorithm**은 올바르게 **attribute**의 **closure**를 구한다
 - » **closure algorithm**에 의해서 구한 **closure**는 정확하다
- **completeness** : **closure algorithm**이 구한 **closure**는 해당 **attribute**로 부터 유도되는 모든 **attribute**를 포함하고 있다

◆ A_1, A_2, \dots, A_n 가 릴레이션 R 에 대한 키인가를 검사

- $\{A_1, A_2, \dots, A_n\}^+$ 는 R 의 모든 애트리뷰트를 포함한다
- $\{A_1, A_2, \dots, A_n\}$ 의 어떤 진부분 집합 S 도 S^+ 가 R 의 모든 애트리뷰트들을 포함하지 않는다. ► **minimality**를 위반하는가 ?

[예] **Movies(title,year,length,filmType,studioName)** :

title year → **length filmType studioName**이 성립하므로 **{title,year}**는 키

Armstrong의 공리

- Armstrong의 공리
 - 함수적 종속성들에 관한 올바르고(sound) 완전한(complete) 추론들의 집합이다.
- ◆ 재귀성(reflexivity)
 - $\beta \subseteq \alpha$ 이면, $\alpha \rightarrow \beta$ 이 성립한다. 이는 단순 종속성에 해당한다.
- ◆ 증가성(augmentation)
 - $\alpha \rightarrow \beta$ 가 성립하면, $\alpha\gamma \rightarrow \beta\gamma$ 이 성립한다.
- ◆ 이행성(transitivity)
 - $\alpha \rightarrow \beta$ 와 $\beta \rightarrow \gamma$ 가 성립하면, $\alpha \rightarrow \gamma$ 도 성립한다.

릴레이션의 기저

◆ 릴레이션의 기저(basis)

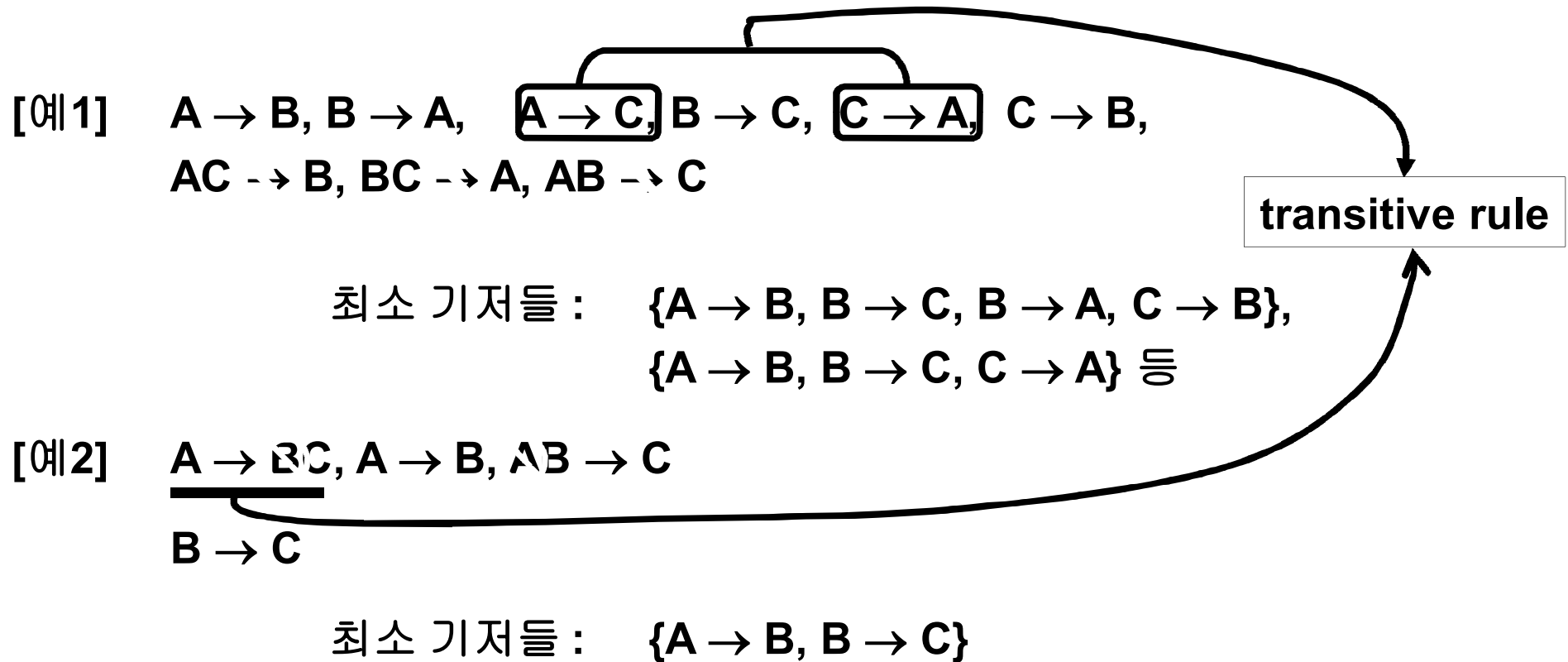
- 어떤 릴레이션에 있는 모든 함수적 종속성들을 유도할 수 있는 함수적 종속성들의 집합
- 최소 기저(minimal basis) : 한 릴레이션에 있는 모든 종속성들을 유도할 수 있는 진부분 집합을 가지고 있지 않은 기저
- $\alpha \rightarrow \beta$ 가 릴레이션 **R**에 대한 **FD**들의 집합 **F**에 속할 때 다음을 만족하면 애트리뷰트 **A**는 중복이다.

❶ 만일 **F**가 $(F - \{\alpha \rightarrow \beta\}) \cup \{(\alpha - A) \rightarrow \beta\}$ 를 유도하면 **A** ($A \in \alpha$)는 중복

❷ 만일 **F**가 $(F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - A)\}$ 를 유도하면 **A** ($A \in \beta$)는 중복

» **minimal basis**가 최소의 **FD**들의 집합을 가지려면 중복된 **FD**들을 모두 제거 해야 한다.

릴레이션의 최소 기저의 예



» 최소 기저를 찾는 방법은 교재에 언급되어 있지 않다.

관계 데이터베이스 설계

- 너무 많은 정보를 하나의 릴레이션에 저장하고자 할 때 발생하는 중복과 같은 문제를 이상(anomaly)이라 한다.

◆ 이상

- 중복(redundancy): 정보들이 불필요하게 여러 튜플에 반복되는 현상(space wastes)
- 갱신 이상(update anomaly): 여러 튜플에 중복되어 나타나는 정보가 하나의 튜플에서만 변경되고 다른 튜플에서는 갱신되지 않은 상태로 남아 있는 문제 (data inconsistency)
- 삭제 이상(deletion anomaly): 어떤 값들의 집합이 공집합이 되는 경우 부수적인 결과로 다른 정보를 잃게 되는 경우
 - » 릴레이션 `Movie`에서 스타 이름이 “**Emlio Estevez**”인 튜플을 삭제하면 영화 “**Mighty Ducks**”에 대한 모든 정보도 잃어버리게 된다.

관계 데이터베이스 설계 (계속)

- 다중값 관계성으로 인한 중복성
 - **starName** 애트리뷰트는 다중값 관계성으로부터 생성되었다.
 - **length, filmType** 정보 등이 반복된다.

title	year	length	filmType	studioName	starName
Star Wars	1977	124	color	Fox	Carrie Fisher
Star Wars	1977	124	color	Fox	Mark Hamill
Star Wars	1977	124	color	Fox	Harrison Ford
Mighty Ducks	1991	104	color	Disney	Emilio Esteves
Wayne's World	1992	95	color	Paramount	Dana Carvey
Wayne's World	1992	95	color	Paramount	Mike Meyers

릴레이션 **Movie**



관계 데이터베이스 설계 (계속)

- 이상 문제들을 해결하기 위해 릴레이션을 분해한다.

- ◆ 릴레이션의 분해(decomposition)

스키마가 $\{A_1, A_2, \dots, A_n\}$ 인 릴레이션 R 을 스키마가 각각 $\{B_1, B_2, \dots, B_m\}$ 과 $\{C_1, C_2, \dots, C_k\}$ 인 릴레이션 S 와 T 로 다음과 같이 분해할 수 있다.

- $\{A_1, A_2, \dots, A_n\} = \{B_1, B_2, \dots, B_m\} \cup \{C_1, C_2, \dots, C_k\}$

1. 릴레이션 S 의 튜플들은 R 의 모든 튜플들을 $\{B_1, B_2, \dots, B_m\}$ 에 대해 프로젝션한 결과이다. (S 에 중복된 튜플은 허용되지 않는다.)
2. 같은 방법으로 릴레이션 T 의 튜플들은 R 의 모든 튜플들을 $\{C_1, C_2, \dots, C_k\}$ 에 대해 프로젝션한 결과이다.

관계 데이터베이스 설계 (계속)

- 릴레이션 Movie (title, year, length, filmType, studioName, starName)를 다음 두 릴레이션들로 분해한다.
 - ❖ Movie1(title, year, length, filmType, studioName)
 - ❖ Movie2(title, year, starName)

title	year	length	filmType	studioName
Star Wars	1977	124	color	Fox
Mighty Ducks	1991	104	color	Disney
Wayne's World	1992	95	color	Paramount

Movie1

title	year	starName
Star Wars	1977	Carrie Fisher
Star Wars	1977	Mark Hamill
Star Wars	1977	Harrison Ford
Mighty Ducks	1991	Emilio Esteves
Wayne's World	1992	Dana Carvey
Wayne's World	1992	Mike Meyers

Movie2



보이스-코드 정규형 (BCNF)

◆ 보이스-코드 정규형(Boyce-Codd Normal Form: BCNF)

- 릴레이션 R 이 BCNF에 있다(iff): 비단순 종속성 $A_1A_2 \dots A_n \rightarrow B$ 가 릴레이션 R 에 있으면, $\{A_1, A_2, \dots, A_n\}$ 은 R 의 수퍼키이다.
 - » 모든 비단순 종속성의 좌변은 수퍼키이어야 한다.
- 릴레이션 R 이 BCNF에 있다(iff) : 비단순 종속성 $A_1A_2 \dots A_n \rightarrow B_1B_2 \dots B_m$ 이 릴레이션 R 에 있으면, $\{A_1, A_2, \dots, A_n\}$ 은 R 의 수퍼키이다.
- 릴레이션 R 이 BCNF에 있다(iff) : 어떤 비단순 종속성도 존재하지 않는다.

BCNF (계속)

Movie는 **BCNF** 형태가 아니다. 키: {title, year, starName}

title	year	length	filmType	studioName	starName
Star Wars	1977	124	color	Fox	Carrie Fisher
Star Wars	1977	124	color	Fox	Mark Hamill
Star Wars	1977	124	color	Fox	Harrison Ford
Mighty Ducks	1991	104	color	Disney	Emilio Esteves
Wayne's World	1992	95	color	Paramount	Dana Carvey
Wayne's World	1992	95	color	Paramount	Mike Meyers

릴레이션 **Movie**

Movie1은 **BCNF** 형태이다. 키: {title, year}

title	year	length	filmType	studioName
Star Wars	1977	124	color	Fox
Mighty Ducks	1991	104	color	Disney
Wayne's World	1992	95	color	Paramount

title year → length filmType
studioName

릴레이션 **Movie1**

BCNF (계속)

◆ 두개의 애트리뷰트로 구성된 릴레이션

■ A와 B로 구성된 릴레이션에 대해서 BCNF임을 증명

① non-trivial FD가 없는 경우 : trivial FD만 있는 경우엔 BCNF에 있다.

[예] Stars-In(title, year, starName) : non-trivial FD가 없다.

② $A \rightarrow B$ 만 있는 경우 : A는 수퍼키이므로 BCNF에 있다.

③ $B \rightarrow A$ 만 있는 경우 : B는 수퍼키이므로 BCNF에 있다.

④ $A \rightarrow B$ 와 $B \rightarrow A$ 가 있는 경우 : A와 B 모두 수퍼키이므로 BCNF에 있다.

◆ 애트리뷰트가 두 개인 모든 릴레이션은 BCNF에 있다.

BCNF (계속)

◆ BCNF로의 분해

- 적절한 분해를 반복하여 한 릴레이션 스키마를 다음의 특성을 만족하는 애트리뷰트들의 부분 집합들로 분리할 수 있다.
 - » 이 부분 집합들은 **BCNF**에 있는 릴레이션 스키마들이다.
 - » 본래의 릴레이션에 있는 데이터들은 분해 후의 릴레이션에 정보의 손실없이 나타난다. 즉, 분해된 릴레이션들로부터 원래의 릴레이션을 다시 복원할 수 있다.

BCNF (계속)

◆ BCNF로 분해하는 알고리즘

1. α 가 릴레이션 R 의 수퍼키가 아닌 각각의 완전 비단순 종속성 $\alpha \rightarrow \beta$ 에 대해, R 을 $R_1 = (\alpha, \beta)$ 와 $R_2 = (R - \beta)$ 로 분해한다
» 여기서, β 에는 비단순 종속성이 없다고 가정한다
2. 이러한 함수적 종속성이 존재하지 않을 때까지 단계 1을 반복한다

[예] **Movie(title, year, length, filmType, studioName, starName)**

키: {title, year, starName}

title year \rightarrow length filmType studioName에서 좌변이 수퍼키가 아니므로
다음 두 릴레이션으로 분해한다

Movie1(title, year, length, filmType, studioName),

Movie2(title, year, starName)



BCNF (계속)

■ 이행 종속성(transitive dependency)으로 인한 중복성

	title	year	length	filmType	studioName	studioAddr
MovieStudio 키: {title, year}	Star Wars	1977	124	color	Fox	Hollywood
	Mighty Ducks	1991	104	color	Disney	Buena Vista
	Wayne's World	1992	95	color	Paramount	Hollywood
	Addams Family	1991	102	color	Paramount	Hollywood

title year → studioName
studioName → studioAddr

/* 문제의 근원: 이행 종속성 */

studioName
→ studioAddr

title	year	length	filmType	studioName
Star Wars	1977	124	color	Fox
Mighty Ducks	1991	104	color	Disney
Wayne's World	1992	95	color	Paramount
Addams Family	1991	102	color	Paramount

MovieStudio1

studioName	studioAddr
Fox	Hollywood
Disney	Buena Vista
Paramount	Hollywood

MovieStudio2

BCNF (계속)

(예) 이행 종속성이 연달아 있는 경우

`MovieMovieExec(title, year, studioName, president, presAddr)`

키: {title, year}

`title year → studioName, studioName → president,`
`president → presAddr`

◆ `president → presAddr`

$R_1 = \{\text{president, presAddr}\}, R = \{\text{title, year, studioName, president}\}$

◆ `studioName → president`의 좌변이 키가 아니므로 R은 BCNF가 아니다.

$R_2 = \{\text{studioName, president}\}, R_3 = \{\text{title, year, studioName}\}$

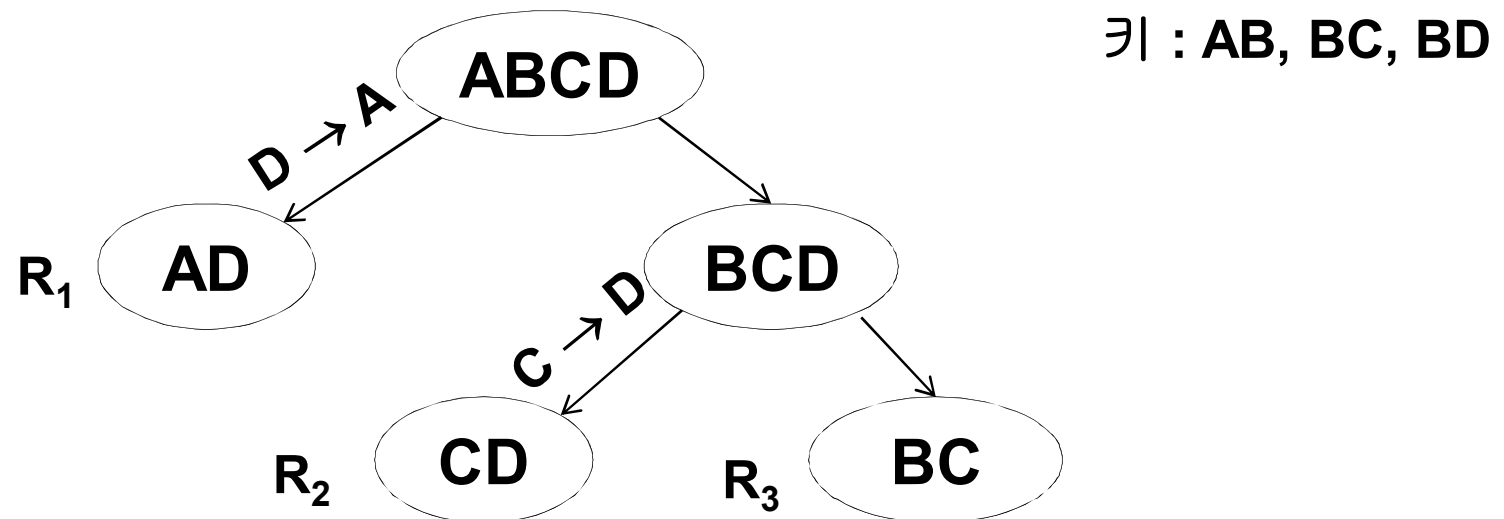
◆ 최종 분해 결과: R_1, R_2, R_3

☞ 참고: BCNF 분해의 결과는 여러 개 있을 수도 있다.

BCNF (계속)

- ❖ 릴레이션 $R(A,B,C,D)$ 와 다음과 같은 FD들에 대해서 BCNF를 만족하는 릴레이션으로 decomposition하라.

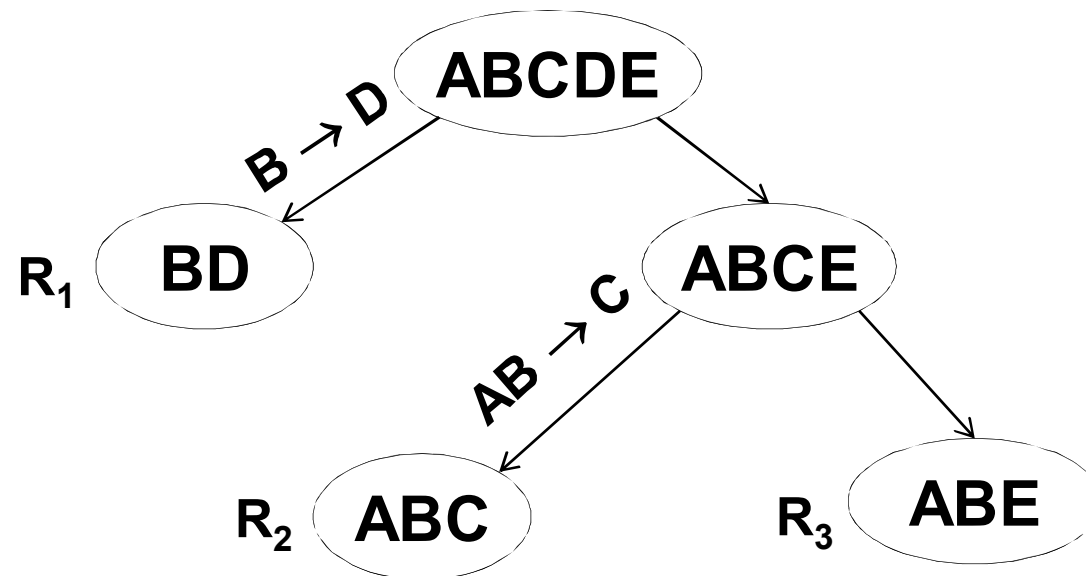
$$AB \rightarrow C, C \rightarrow D, D \rightarrow A$$



BCNF (계속)

- ❖ 릴레이션 $R(A,B,C,D,E)$ 와 다음과 같은 FD들에 대해서 BCNF를 만족하는 릴레이션으로 decomposition하라.

$$AB \rightarrow C, DE \rightarrow C, B \rightarrow D$$

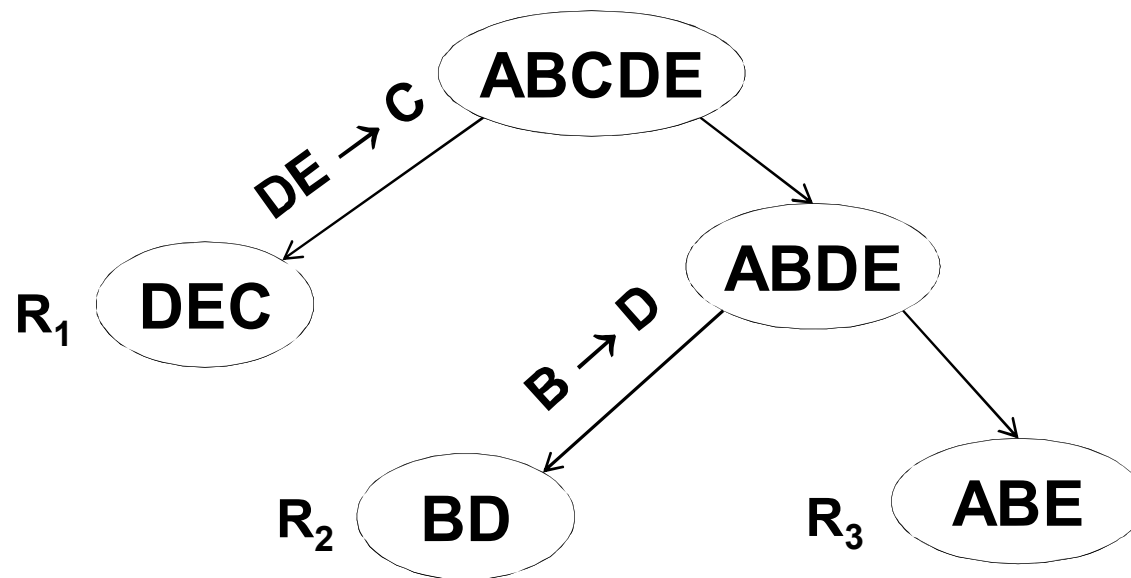


키 : **ABE**

BCNF (계속)

- ❖ 릴레이션 $R(A,B,C,D,E)$ 와 다음과 같은 FD들에 대해서 BCNF를 만족하는 릴레이션으로 decomposition하라.

$AB \rightarrow C, DE \rightarrow C, B \rightarrow D$

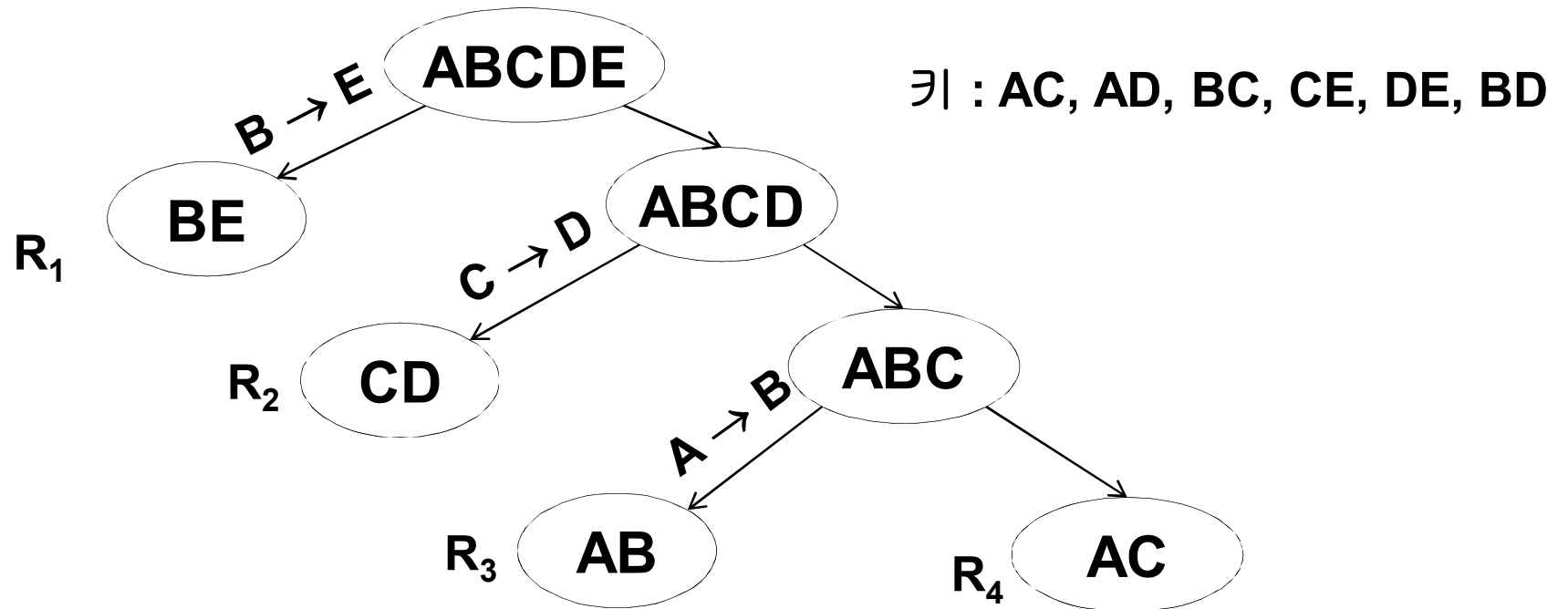


키 : ABE

BCNF (계속)

- ◆ 릴레이션 $R(A,B,C,D,E)$ 와 다음과 같은 FD들에 대해서 BCNF를 만족하는 릴레이션으로 decomposition하라.

$A \rightarrow B, B \rightarrow E, C \rightarrow D, DE \rightarrow AC$



함수적 종속성의 추출

◆ 분해된 결과에서 함수적 종속성 찾기

- 릴레이션 **R** 이 분해되어 릴레이션 **S**와 다른 릴레이션이 생성되었으며, **F**는 릴레이션 **R** 에서 성립하는 함수적 종속성들의 집합이라고 하자.

■ 릴레이션 **S**에서 성립하는 종속성 찾기

- **S**의 애트리뷰트 집합의 각 부분집합 **X**에 대해, 애트리뷰트 **B**가 다음의 조건을 만족하면 함수적 종속성 $X \rightarrow B$ 가 릴레이션 **S**에서 성립한다.
 - **B**는 **S**의 애트리뷰트이다
 - **B**가 **X**에는 없으나 X^+ 에는 포함된다.

함수적 종속성의 추출 (계속)

[예] $R(A, B, C, D)$ 에서 함수적 종속성 $A \rightarrow B$ 와 $B \rightarrow C$ 가 성립한다.

$S(A, C)$ 가 릴레이션을 분해하여 생기는 릴레이션 스키마의 하나일 때

S 에서 성립하는 종속성을 찾아보자.

- $\{A\}^+ = \{A, B, C\}$. C 가 S 와 $\{A\}^+$ 에 있으므로, $A \rightarrow C$ 가 성립한다.
- $\{C\}^+ = \{C\}$ /* 사실상 이 경우는 고려할 필요가 없다. */
- 따라서 $A \rightarrow C$ 가 S 에서 성립하는 유일한 종속성이다.

[예] $R(A, B, C, D, E)$ 에서 함수적 종속성 $A \rightarrow D$, $B \rightarrow E$, $DE \rightarrow C$ 가 성립한다.

$S(A, B, C)$ 에서 성립하는 종속성을 찾아보자.

- $\{A\}^+ = \{A, D\}$ /* S 에 대한 종속성은 만들어지지 않는다. */
- $\{B\}^+ = \{B, E\}$, $\{C\}^+ = \{C\}$ /* S 에 대한 종속성은 만들어지지 않는다. */
- $\{A, B\}^+ = \{A, B, C, D, E\}$: $AB \rightarrow C$
- $AB \rightarrow C$ 가 S 에서 성립하는 유일한 종속성이다.

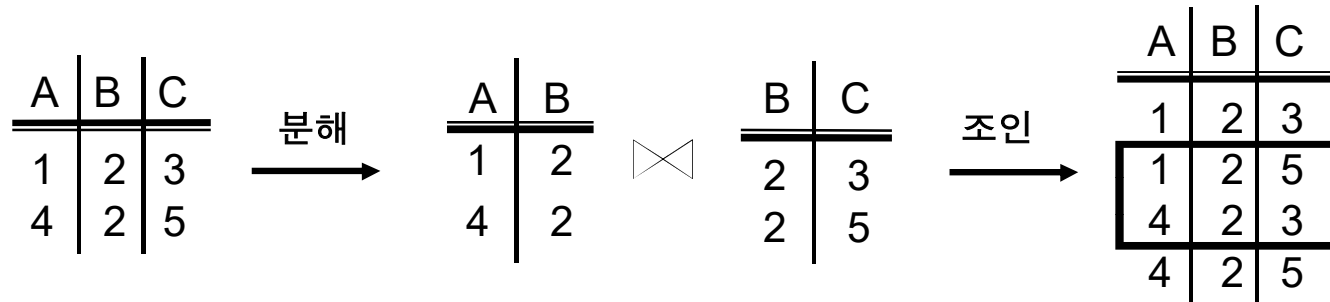
분해된 결과로부터 정보를 복원

- 릴레이션이 분해되었을 때, 본래 릴레이션은 분해된 릴레이션들의 조인(join)에 의해서만 복원될 수 있다.

☞ 조인에 대한 자세한 설명은 다음 장에서 기술됨

◆ FD를 무시한 부주의한 분해

- 다음과 같은 스키마 {A, B, C}를 가진 릴레이션 R에서 **non-trivial FD**가 없을 때 ..



- 정보의 손실(information loss) : 조인 후에 불필요한 데이터가 추가되었다.

분해된 결과로부터 정보를 복원 (계속)

◆ FD를 고려한 릴레이션의 분해

- $B \rightarrow C$ 가 $R(A, B, C)$ 에 대해서 성립할 때 릴레이션 R 이 $R_1(A, B)$ 와 $R_2(B, C)$ 로 분해된다고 가정.

- ① R 의 두 튜플들을 $t=(a, b, c)$, $v=(d, b, e)$ 라 하자.
- ② R_1 에 대해서 프로젝션시키면 $u=(a, b)$ 와, R_2 에 대해서 프로젝션시키면 $w=(b, e)$ 를 얻을 수 있다.
- ③ u 와 w 를 조인하면 $x=(a, b, e)$ 를 얻지만 R 에 존재하는지 확인해야 한다.
- ④ $B \rightarrow C$ 이므로 t 와 v 에서 $c=e$ 가 만족되어야 한다.
- ⑤ 결론적으로, $x=(a, b, e)=(a, b, c)$ 이며 t 와 일치하므로 R_1 와 R_2 는 정보의 손실이 없는 분해이다.

무손실 조인(Lossless Join) 분해

◆ $\rho = (R_1, \dots, R_n)$ 를 R 의 분해라고 하자. 릴레이션 R_1, \dots, R_n 의 조인이 릴레이션 R 과 항상 같으면, ρ 를 R 의 무손실 조인 분해(lossless join decomposition)라고 한다 → BCNF 분해 알고리즘은 무손실 조인이다

◆ 무손실 조인 분해 검사

- R_1 과 R_2 가 R 의 분해라고 하자.
- $R_1 \cap R_2 \rightarrow R_1$ 또는 $R_1 \cap R_2 \rightarrow R_2$ 가 성립하면(iff), 무손실 조인 분해이다.

[예] $R = \{A, B, C\}$, 함수적 종속성: $A \rightarrow B$

- » $R_1 = \{A, B\}$, $R_2 = \{A, C\}$: 무손실(lossless)
- » $R_1 = \{A, B\}$, $R_2 = \{B, C\}$: 손실(lossy)
- » $R_1 = \{A, C\}$, $R_2 = \{B, C\}$: 손실(lossy)

A	B	C
1	2	3
4	2	5
5	3	1
5	3	3



제 3 정규형 (Third Normal Form: 3NF)

🍷 BCNF 분해에서 종속성이 보존(preserve)되지 않을 수도 있다.

[예] **Bookings (title, theater, city):**

한 영화가 현재 어떤도시의 어떤극장에서 상영중이다.

theater → **city** /* 한 영화관은 하나의 도시에 위치한다. */

title city → **theater** /* 같은 도시에 있는 두 영화관이 동일한 영화를 상영하지 않는다. */

키: {title, city}, {theater, title}

BCNF 분해: {theater, city}, {title, theater} : theater → city 만이 보존된다.

theater	city	theater	title		theater	city	title
Guild	Menlo Park	Guild	The Net	조인 ➔	Guild	Menlo Park	The Net
Park	Menlo Park	Park	The Net		Park	Menlo Park	The Net

“title city → theater”를 만족시키지 않는다.

제 3 정규형 (계속)

◆ 종속성 보존 분해(Dependency Preserving Decomposition)

– 스키마 R 에 대한 FD들의 집합 F 에 대해서 :

» R 의 decomposition을 $R_1 \cup R_2 \cup \dots \cup R_k$ 이라 하고,

» $F' = F_1 \cup F_2 \cup \dots \cup F_k$ 를 F_i 가 R_i 의 애트리뷰트만 포함한 FD들의 집합이라 하자.

» 만일 $F^+ = F'^+$ 이면 $R = \{R_1 \cup R_2 \cup \dots \cup R_k\}$ 를 종속성 보존 분해라 한다.

◆ 다음의 조건을 만족하는 릴레이션 R 은 제 3 정규형에 있다.

– 비단순 종속성 $\alpha \rightarrow \beta$ 가 있으면, α 가 수퍼키이거나 또는 β 가 어떤 키의 멤버이다. (이 때 β 를 **prime** 또는 키 애트리뷰트라고 한다.)

» 무손실 조인이며 모든 종속성을 보존하는 제 3 정규형의 릴레이션들로 항상 분해될 수 있다.

» 릴레이션들이 **BCNF** 조건을 만족하지 않을 때에는 스키마에 약간의 중복성이 남아 있다는 것을 의미한다.

제 3 정규형 (계속)

◆ BCNF를 위반하는 3NF에 있는 릴레이션 : 중복 허용

theater	city	title
A	P	123
A	P	456
A	P	678
A	P	789

theater \rightarrow city, title city \rightarrow theater

제 3 정규형 (계속)

◆ 무손실 조인이며 모든 종속성을 보존하는 3NF 분해

■ FD를 릴레이션 R에 있는 함수적 종속성들의 **minimal basis**라고 하자.

1. FD에 있는 각 $\alpha \rightarrow \beta$ 에 대해, 릴레이션 스키마 $R_i = \{\alpha\beta\}$ 를 생성한다.
2. R의 키를 포함하는 릴레이션 스키마가 없으면, R의 키 애트리뷰트만으로 구성된 릴레이션 스키마를 추가하라.
3. F' 를 $F_1 \cup F_2 \cup \dots \cup F_k$ 라 할 때, 만일 $F^+ = F'^+$ 이면 성공적으로 종속성 보존 분해가 성공한 것이다.

제 3 정규형 (계속)

[예1] $R = \{C, T, H, R, S, G\}$

함수적 종속성: $C \rightarrow T$, $HS \rightarrow R$, $CS \rightarrow G$, $HR \rightarrow C$, $HT \rightarrow R$

키: $\{H, S\}$

3NF: $\{C, T\}$, $\{H, R, S\}$, $\{C, S, G\}$, $\{H, R, C\}$, $\{H, T, R\}$

[예2] $R = \{B, O, I, S, Q, D\}$

함수적 종속성: $S \rightarrow D$, $I \rightarrow B$, $IS \rightarrow Q$, $B \rightarrow O$

키: $\{I, S\}$

3NF: $\{S, D\}$, $\{I, B\}$, $\{I, S, Q\}$, $\{B, O\}$

BCNF: $\{S, D\}$, $\{I, B\}$, $\{I, S, Q\}$, $\{O, I, S\}$ /* 종속성 $B \rightarrow O$ 가 보존되지 않는다. */

BCNF: $\{S, D\}$, $\{B, O\}$, $\{I, B\}$, $\{I, S, Q\}$ /* 종속성이 보존된다. */

다중값(Multivalued) 종속성

- ◆ **BCNF** 에 있는 릴레이션이지만 함수적 종속성과는 관련이 없는 중복이 발생하는 경우가 있다.
 - 1:M의 관계성을 가지는 애트리뷰트들은 다중값 형태를 가질 수 있다.
 - 스타의 주소가 하나 이상이라면 스타와 주소 사이의 관계성은 1:M이다.
- ◆ **BCNF** 스키마에서 중복들이 발생하는 가장 큰 원인은 둘 이상의 서로 독립적인 다중값 애트리뷰트들이 있기 때문이다.

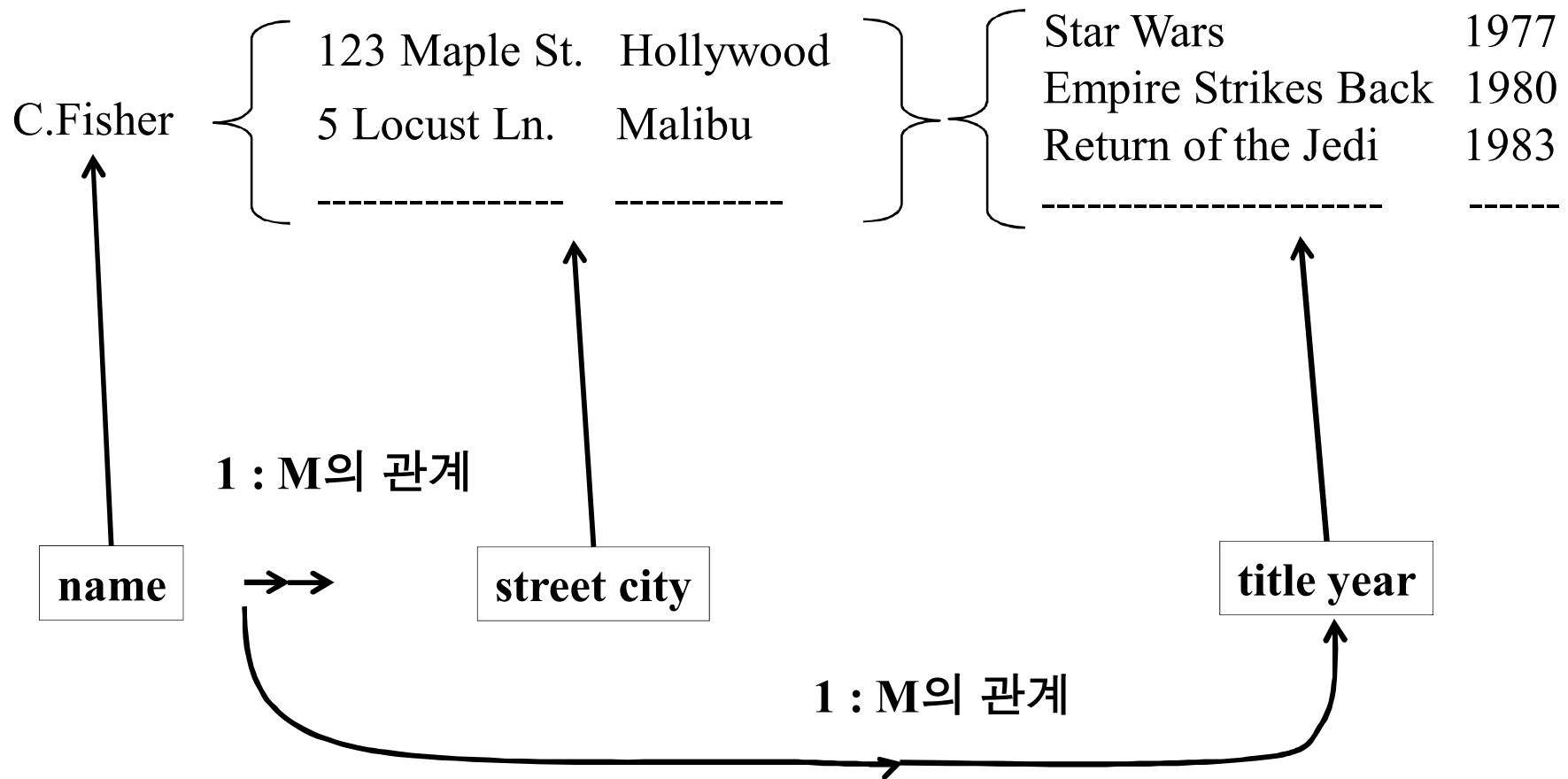
다중값 종속성 (계속)

name	street	city	title	year
C.Fisher	123 Maple St.	Hollywood	Star Wars	1977
C.Fisher	123 Maple St.	Hollywood	Empire Strikes Back	1980
C.Fisher	123 Maple St.	Hollywood	Return of the Jedi	1983
C.Fisher	5 Locust Ln.	Malibu	Star Wars	1977
C.Fisher	5 Locust Ln.	Malibu	Empire Strikes Back	1980
C.Fisher	5 Locust Ln.	Malibu	Return of the Jedi	1983

영화와 독립적인 주소들의 집합

- 각 영화마다 주소 정보가 나타난다.
 - 주소와 영화의 모든 조합이 나타난다.
- **BCNF**에 있지만 중복성을 가진다.
 - 다섯 개의 모든 애트리뷰트가 모여 유일한 키를 형성한다.

다중값 종속성 (계속)



다중값 종속성 (계속)

- ◆ 다중값 종속성 $A_1A_2 \dots A_n \twoheadrightarrow B_1B_2 \dots B_m$ 이 릴레이션 R 에서 성립하기 위한 조건은 다음과 같다.
 - 릴레이션 R 의 A 애트리뷰트 값이 일치하는 임의의 두 튜플 t 와 u 가 있을 때, 다음을 만족하는 튜플 v 도 있다.
 1. v 의 A 애트리뷰트 값이 t 와 u 의 A 값과 일치한다.
 2. v 의 B 애트리뷰트 값이 t 의 B 와 일치한다.
 3. A 나 B 에 속하지 않는 릴레이션 R 의 나머지 모든 애트리뷰트에 대하여 v 는 u 와 일치한다.

	A's	B's	others
t			
u			
v			

다중값 종속성 (계속)

- 다중값 종속성이란 릴레이션 **R**에서 애트리뷰트 집합 **A** (예를 들어, **name**)의 값을 고정시킬 때, 다른 특정 애트리뷰트 집합 **B**(예를 들어, **street, city**)의 값이 **A** 나 **B** 에 속하지 않는 릴레이션 **R**의 나머지 모든 애트리뷰트들(예를 들어, **title year**)의 값에 독립이라는 것을 의미한다.

name	street	city	title	year
C.Fisher	123 Maple St.	Hollywood	Star Wars	1977
C.Fisher	5 Locust Ln.	Malibu	Empire Strikes Back	1980

name →→ **street city** 이면 다음의 두 튜플도 반드시 있어야 한다.

(C.Fisher, 123 Maple St., Hollywood, Empire Strikes Back, 1980)

(C.Fisher, 5 Locust Ln., Malibu, Star Wars, 1977)

다중값 종속성에 대한 추론

◆ 단순 종속성 규칙

- 다중값 종속성 $A_1A_2 \dots A_n \twoheadrightarrow B_1B_2 \dots B_m$ 이 어떤 릴레이션에서 성립하면, B 에 A 의 어떤 애트리뷰트(들)를 추가한 C 에 대해서 $A_1A_2 \dots A_n \twoheadrightarrow C_1C_2 \dots C_k$ 도 성립한다.
 - » A 에 속하지 않는 모든 B 애트리뷰트들을 D 라고 하면 $A_1A_2 \dots A_n \twoheadrightarrow D_1D_2 \dots D_r$ 도 성립한다.
- $\alpha \cup \beta = R$ 인 경우 $\alpha \twoheadrightarrow \beta$ 가 성립하면

◆ 이행 규칙

- 어떤 릴레이션에서 다중값 종속성 $A_1A_2 \dots A_n \twoheadrightarrow B_1B_2 \dots B_m$ 과 $B_1B_2 \dots B_m \twoheadrightarrow C_1C_2 \dots C_k$ 가 성립하면, $A_1A_2 \dots A_n \twoheadrightarrow C_1C_2 \dots C_k$ 도 성립한다.
 - » 단, B_i 와 C_i 는 intersection이 공집합이어야 함.
- $\alpha \twoheadrightarrow \beta$ 와 $\beta \twoheadrightarrow \gamma$ 가 성립하면 $\alpha \twoheadrightarrow (\gamma - \beta)$ 가 성립한다.

다중값 종속성 (계속)

◆ 보완(complementation) 규칙

- 릴레이션 R 에서 다중값 종속성 $\alpha \twoheadrightarrow \beta$ 이 성립하면, $\alpha \twoheadrightarrow (R - \beta - \alpha)$ 도 만족한다.
- $\text{name} \twoheadrightarrow \text{street city}$ 가 성립하면 $\text{name} \twoheadrightarrow \text{title year}$ 도 성립한다.

◆ 모든 함수적 종속성은 다중값 종속성이다.

- $A_1A_2 \dots A_n \rightarrow B_1B_2 \dots B_m$ 이면 $A_1A_2 \dots A_n \twoheadrightarrow B_1B_2 \dots B_m$ 이다.

☞ 다중값 종속성에서 분할/결합 규칙은 성립하지 않는다.

[예] $\text{name} \twoheadrightarrow \text{street city}$ 는 성립하나 $\text{name} \twoheadrightarrow \text{street}$ 는 성립하지 않는다.

제 4 정규형 (Fourth Normal Form: 4NF)

◆ 비단순(non-trivial) 다중값 종속성

– $\alpha \twoheadrightarrow \beta$ 이 다음 두 조건을 만족하면 비단순이다.

1. β 에는 α 의 어떤 애트리뷰트도 없다.

2. $\alpha \cup \beta \neq R$

◆ 제 4 정규형(Fourth Normal Form:4NF)

– 다음을 만족하면 릴레이션 R 은 4NF이다: $\alpha \twoheadrightarrow \beta$ 이 R 에 있는 비단순 다중값 종속성이면, α 는 R 의 수퍼키이다.

☞ 두개의 독립적인 다중값 애트리뷰트가 한 릴레이션에 있어서는 안된다.

☞ 4NF 조건은 BCNF 조건과 본질적으로 같은 형태이나, 함수적 종속성 대신 다중값 종속성을 사용한다.

제 4 정규형 (계속)

◆ 4NF으로 분해하는 알고리즘

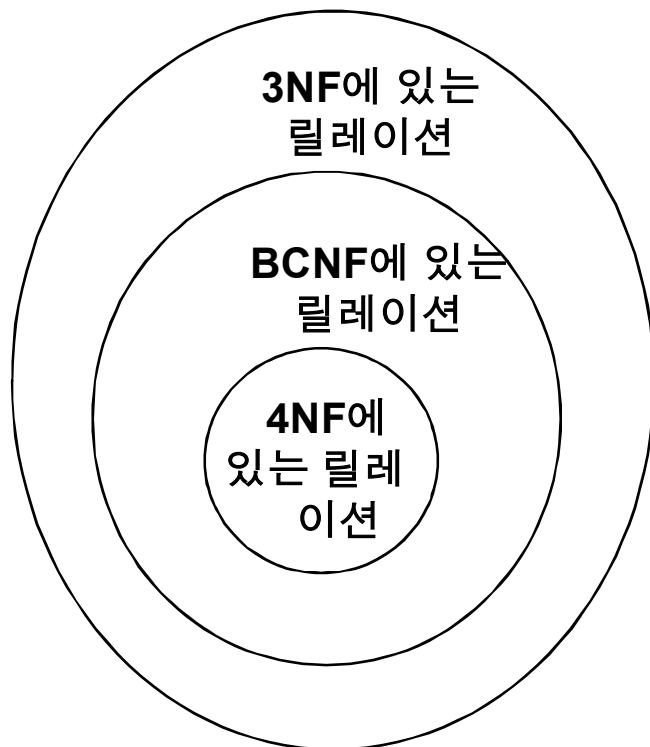
1. X 가 릴레이션 R 의 수퍼키가 아닌 각 비단순 다중값 종속성 $X \twoheadrightarrow A$ 에 대해, R 을 $R_1 = (X, A)$, $R_2 = (R - A)$ 인 R_1 과 R_2 로 분해한다.
» A 내부에는 비단순 다중값 종속성이 없다고 가정한다.
2. 이러한 비단순 다중값 종속성이 없을 때까지 단계 1을 반복한다.

name	street	city	title	year
C.Fisher	123 Maple St.	Hollywood	Star Wares	1977
C.Fisher	5 Locust Ln.	Malibu	Star Wares	1977
C.Fisher	123 Maple St.	Hollywood	Empire Strikes Back	1980
C.Fisher	5 Locust Ln.	Malibu	Empire Strikes Back	1980

“name \twoheadrightarrow street city” 이 4NF 조건을 위반하는 것이므로 {name, street, city}와 {name, title, year}로 분해한다.

제 4 정규형 (계속)

◆ 정규형들 사이의 관계



특성	3NF	BCNF	4NF
함수적 종속성에 따른 중복성의 제거	Most	Yes	Yes
다중값 종속성에 따른 중복성의 제거	No	No	Yes
함수적 종속성의 보존	Yes	Maybe	Maybe
다중값 종속성의 보존	Maybe	Maybe	Maybe