

Chap3 The Relational Model of Data

③ 복습

E/R 모델 : 현실세계 구조를 피는 단계

Relational 모델 : DB 설계 과정의 ~~1차~~ 모델

* Data model (데이터 모델 or 개념적 데이터 모델)

- 자료를 추상적으로 나타내는 기술
- 일정한 제약이 있는 일반적인 구조와 연산으로 구성
- 데이터와 데이터들간의 관계를 기술하는 개념적 도구. (DB의 설계 구조를 명시)

④ 릴레이션 모델의 기초

* Relation (릴레이션)

- 2차원적 테이블로 각 요소들이 원자적 값을 가지는 튜플의 집합.

(ex) Movies

title	year	length	genre
Gone With...	1939	231	drama
Star Wars	1977	124	SciFi
Wayne's ...	1992	95	comedy

← tuples (rows)

④ 데이터뷰트 속성

* 데이터뷰트

- 릴레이션의 각 열에 대한 이름. (title, year, length, genre)

* 튜플

- 릴레이션의 행으로, 릴레이션은 튜플들의 집합 (리스트 x)으로 정의된다. (Star wars, 1977, 124, ...)

* 도메인

- 릴레이션의 각 데이터뷰트에 연관된 타입 : 자료형

* 스키마 (Schema) ⇒ 데이터가 무엇인지 설명하는 것 (관계 스키마) (= relational database schema) (= database schema)

- 릴레이션의 이름과 릴레이션의 데이터뷰트들의 집합. 데이터뷰트의 순서는 무관하다.

(예) Movies (title, year, length, genre)

* 릴레이션 Instance (사례)

- 릴레이션 스키마에 따라 튜플들의 집합

* 릴레이션의 동일한 표현

(예) Movies

year	title	genre	length
1939	Gone With...	drama	231
1992	Wayne's	Comedy	95
1977	Star Wars	SciFi	124

① ② ③

- 튜플들의 순서는 무관 : ③

- 데이터뷰트의 순서도 무관 : ①, ②

* 관계의 기

- 같은 값을 가지면 안되고 식별하기 위한 독특한 값들로 구성되어야 한다.
- 릴레이션에 표시할 때 밑줄을 같이 표시한다.
- 키는 제약이 있으며 여러개의 선택이 가능하다.

* 스키마 (Schema) 표기법

- (예) Movies (title : string, year : integer, length : integer, genre : string)
- (예) Movies (title, year, length, genre)

② 쿼리 언어의 Algebraic (대수)

- 릴레이션의 대수, 프로그래밍 이론이 가깝다.

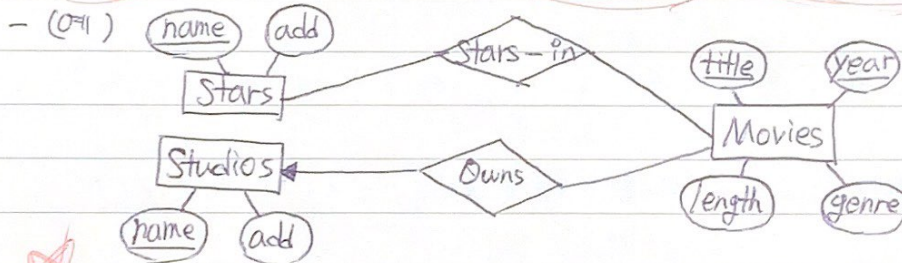
SQL: 관계형 DB를 설명하고
프로그래밍에 사용하는 언어

DDL: 데이터 정의 언어

DML: 데이터 조작 언어

③ E/R 다이어그램에서 관계 모델로

- 엔티티 집합을 릴레이션으로 변환
- 관계성을 릴레이션으로 변환: 연결된 엔티티의 키값과 자신의 애포리뷰드 값을 애포리뷰드 값이다.



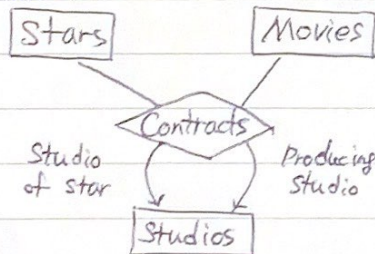
Owns (relation)

<u>title</u>	year	StudiosName
...

Stars-in (relation)

<u>title</u>	year	StarsName
...

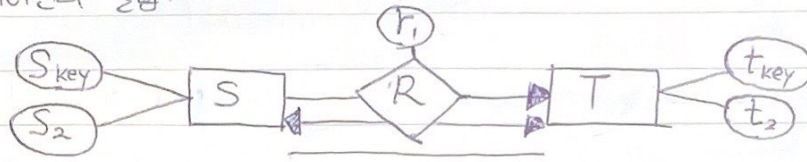
다중방향 관계성을 릴레이션으로 변환



Contracts

StarName	title	Year	Star Studios	ProducingStudio
...

* 릴레이션의 결합.



- M:1 관계 (M:1 관계)

S에 R의 모든 애틀리뷰트와 T의 key를 포함.

S(Skey, S2)	⇒	S(Skey, S2, tkey, r1)
R(Skey, tkey, r1)		T(tkey, t2)
T(tkey, t2)		

(예) Studio 와 Cuns 를 Movies 이 합치면

title	year	length	genre	studioName
...

- 1:1 관계

S에 R의 모든 애틀리뷰트와 T의 key를 포함, 그 역도 가능.

S에 R과 T에 대한 테이블을 생성하려면 모든 것을 하나로 표현

$S(S_{key}, S_2)$	\Rightarrow	$S(S_{key}, S_2 \cdot t_{key}, r_1)$	\Rightarrow	$ST(S_{key}, S_2 \cdot t_{key}, r_1, t_2)$
$R(S_{key}, t_{key}, r_1)$		$T(t_{key}, t_2)$		
$T(t_{key}, t_2)$				

- 다:다 관계

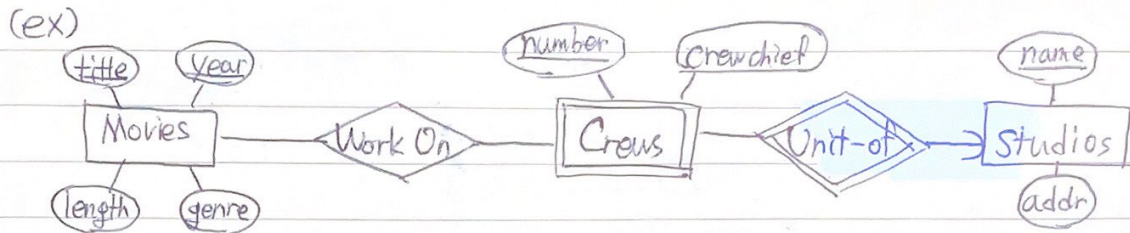
하나의 릴레이션은 중복이 발생하므로 좋은 설계는 아니다. (관계는 양쪽의 키값을 가지고 있다)

(ex) Chap3 - 31 page 참고.

* 애틀리뷰티를 릴레이션으로 변환

- 애틀리뷰티 W에 릴레이션은 W의 애틀리뷰트들, 지원관계가 가리키는 애틀리뷰트의 키값, 지원관계에 있는 애틀리뷰트들을 포함한다.

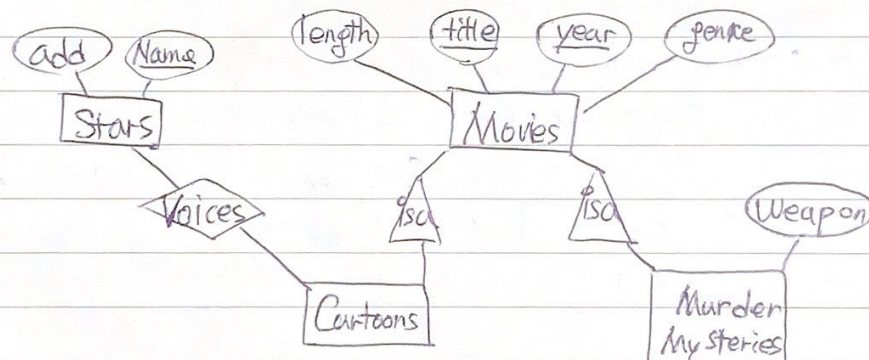
- 지원관계성 릴레이션은 생성하지 않는다.



Studios		Crews		
name	addr	number	Crewchief	StudiosName
...

Unit of		Work-On			
StudiosName	CrewsNumber	title	Year	CrewsNumber	StudiosName
...

* E/R 모델의 하위클래스를 관계모델로 변환



- isa 계층을 관계로 변환 (스키마로 표현)

movies (title , year , length , genre)

MurderMyStories (title , year , weapon)

Cartoons (title , year)

Voices (title , year , StarsName)

- 하위클래스에 합쳐진다면 (다른 분류를 할 필요가 없다면)

movies (title , year , length , genre)

MoviesC (title , year , length , genre)

MoviesMM (title , year , length , genre , weapon)

MoviesCMM (title , year , length , genre , weapon)