

# Релационен модел

Таблици

Схеми

Преобразуване от E/R  
модел в релационен

# Релационен модел

- Днешните разработки на бази от данни почти винаги се базират на *релационния модел*
- Основен елемент в модела е *релацията* - двумерна таблица, в която се съхраняват данните
  - Основава се на математическото понятие **релация (relation)**
- Атрибутите служат за имена на колони
- Името на релацията и множеството от атрибутите ѝ формират схемата на релацията
- Множеството от всички схеми на релации в една БД - схема на БД - *relational database schema (database schema)*

# Защо релации?

- Простота на модела
- Съответства на начина на човешкото мислене
- Еднообразен начин за представяне на данните и връзките
- ...

# ОСНОВНИ ПОНЯТИЯ

- Атрибути
- Схеми
- Кортежи
- Домейни

# Представяне на същността Movie

<b>Movies</b>	<i>title</i>	<i>year</i>	<i>length</i>	<i>filmType</i>
	Star Wars	1977	124	color
	Mighty Ducks	1991	104	color
	Wayne's World	1992	95	color

# Схеми

- Името на релацията и множеството от атрибутите ѝ - схема на релацията.

*Movies(title, year, length, filmType)*

- Атрибутите в релационната схема са множество, а не списък. Независимо от това, ние трябва да укажем стандартна подредба на атрибутите един спрямо друг.

# Schema

- *Relation schema* =  
relation name + attributes
- *Database* = collection of relations.
- *Database schema* =  
set of all relation schemas in the database

# Атрибути

- Атрибутите са означенията (имената) на колоните на релацията.
- За предния пример атрибутите са:
  - *title, year, length* и *filmType*.
  - описват съдържанието на отделната колона на релацията.
    - Например *length* е атрибут за времетраенето на съответния филм.
- Атрибутите на една релация са практически атрибутите от съответната същност в Е/Р модела.



# Кортежи

- Редовете в релацията (без заглавната част - атрибутите) се наричат **кортежи (tuples)**.
  - един компонент за всеки от атрибутите (Star Wars, 1977, 124, color);
  - следва се вече определената последователност за атрибутите от релационната схема.
- Релацията е м-во от кортежи
- Релационният модел изисква всеки компонент на кортеж да бъде атомарен (неделим)
  - Прости типове - integer, string
  - Сложни типове като lists, arrays не са разрешени

# Домейни

- Асоциират се с всеки атрибут
- Компонентите на всеки кортеж принадлежат на домейна за съответния атрибут

- Пример: Movies

```
Movies(  
  title:string,  
  year:integer,  
  length:integer,  
  filmType:string)
```

# Екземпляри на релации

- Една реална релация не може да бъде статична, а се променя с времето
- Два типа промени:
  - insert, update, deletion of tuples (много често)
  - промени в схемата (по-рядко)
- Множеството от кортежи за дадена релация ще наричаме екземпляр на релацията
  - Екземплярите на релациите се променят с времето
  - Релациите съхраняват current instance
  - Cardinality of relation instance = number of tuples in relation
- Схема  $\neq$  Екземпляр (Schema  $\neq$  Instance)
- Relation instance, database instance

# От E/R диаграми към релации

Основни правила:

- Всяка същност се преобразува в отделна релация със същите атрибути
- Всяка връзка се преобразува в релация, чиито атрибути са:
  - Ключовете на свързаните същности
  - Атрибутите на връзката (ако има такива)

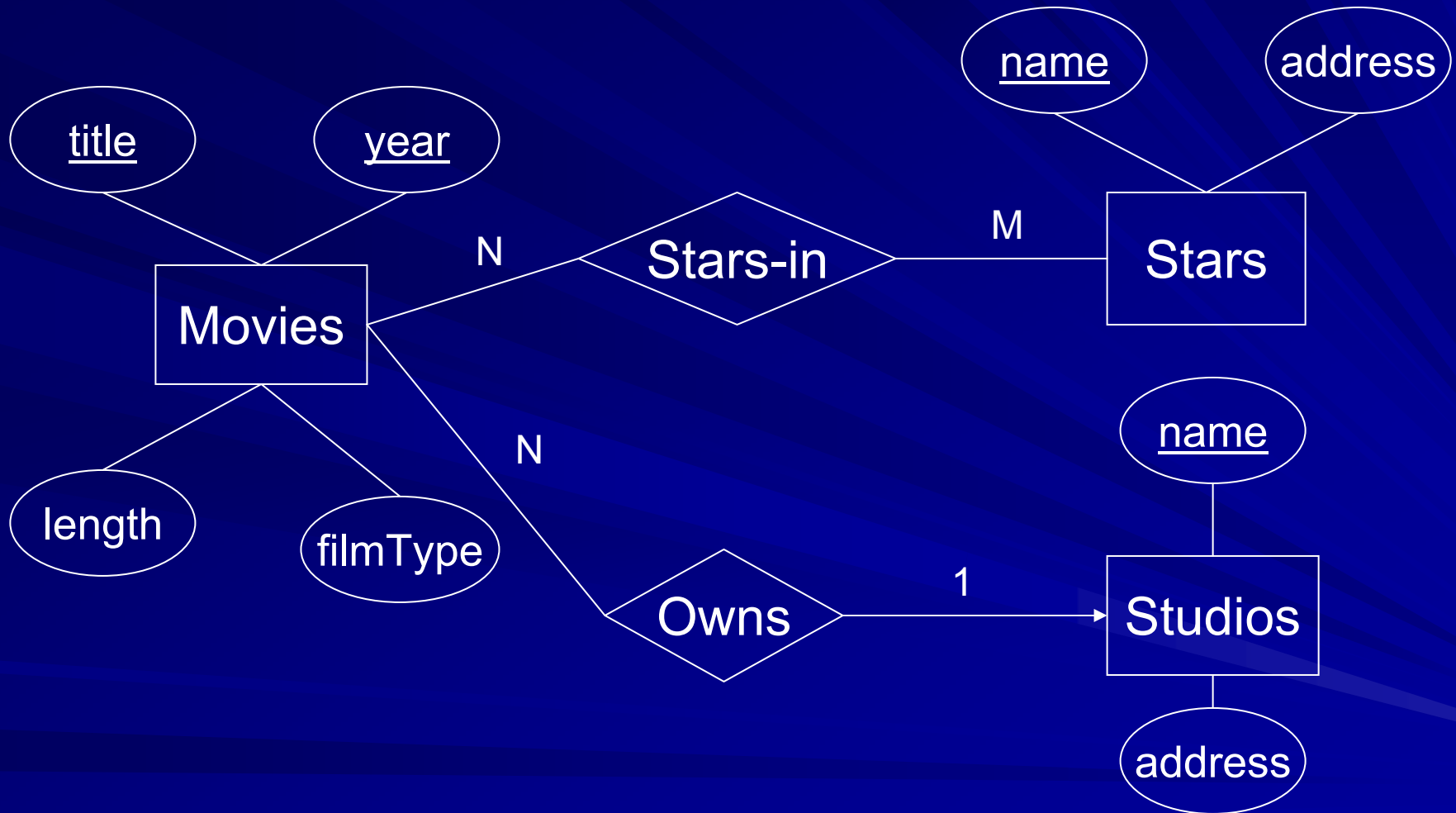
# От E/R диаграми към релации (2)

## Специални случаи:

- слабите м-ва същности не могат да се преобразуват директно в релации
- “Isa”-връзките и подкласовете изискват специална обработка
- понякога комбинираме 2 релации

Например, релацията за същност Е и релацията, която произлиза от връзката “много към един” (many-one) от Е към друга същност F.

# Movie Database



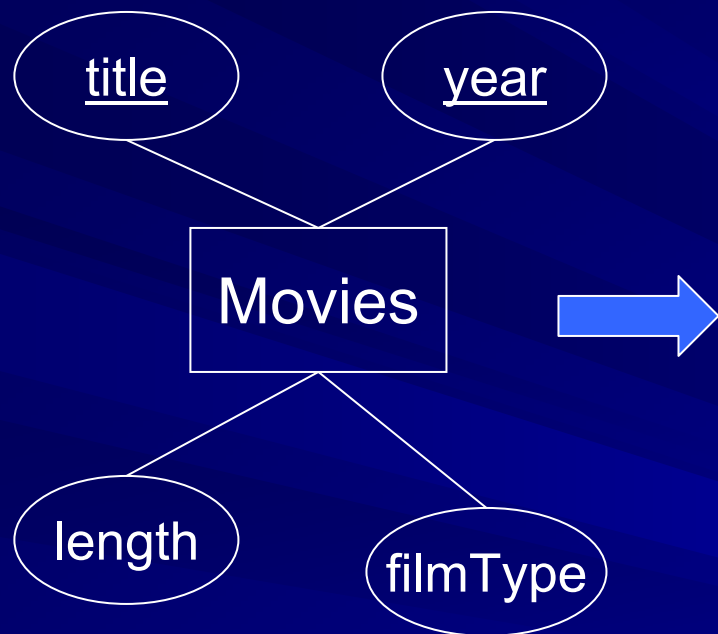
# Example

<b>Movies</b>	<i>title</i>	<i>year</i>	<i>length</i>	<i>filmType</i>
	Star Wars	1977	124	color
	Mighty Ducks	1991	97	color
	Wayne's World	1992	109	color

<b>Owns</b>	<i>title</i>	<i>year</i>	<i>studioName</i>
	Star Wars	1977	Fox
	Mighty Ducks	1991	Disney
	Wayne's World	1992	Paramount

<b>Studios</b>	<i>studioName</i>	<i>address</i>
	Fox	Fox Blvd.
	Disney	Main Str.
	Paramount	Av. of the Stars

# Преобразуване на същности



<b>Movies</b>			
<u><i>title</i></u>	<u><i>year</i></u>	<i>length</i>	<i>filmType</i>
Star Wars	1977	124	color
Mighty Ducks	1991	104	color
Wayne's World	1992	95	color

- 1 таблица за всяка същност (strong entity)
  - Ключът на същността става ключ на релацията



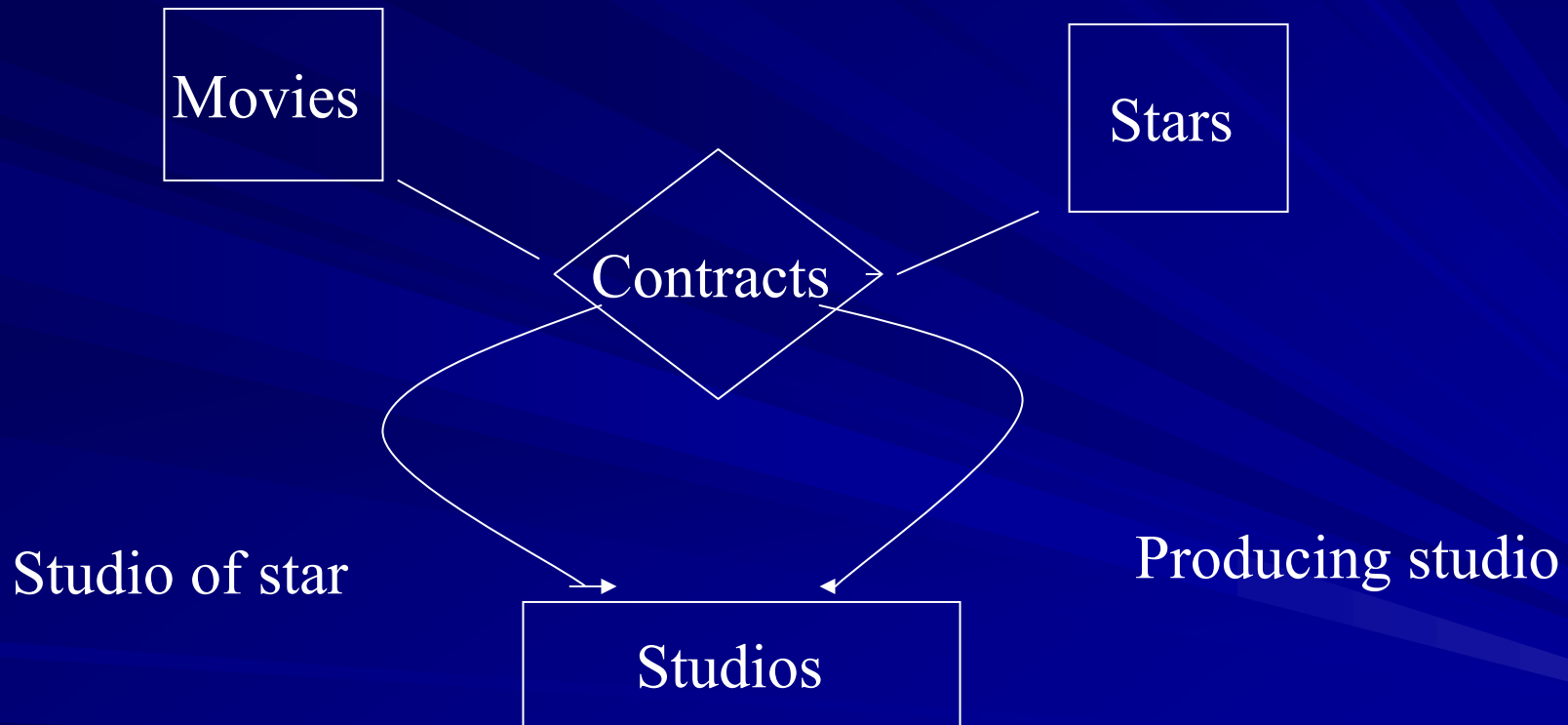
# От E/R връзки към релации



- Използвайте ключа на I същност и ключа на II същност
  - Name → starName

# От E/R връзки към релации

A four-way relationship - ?



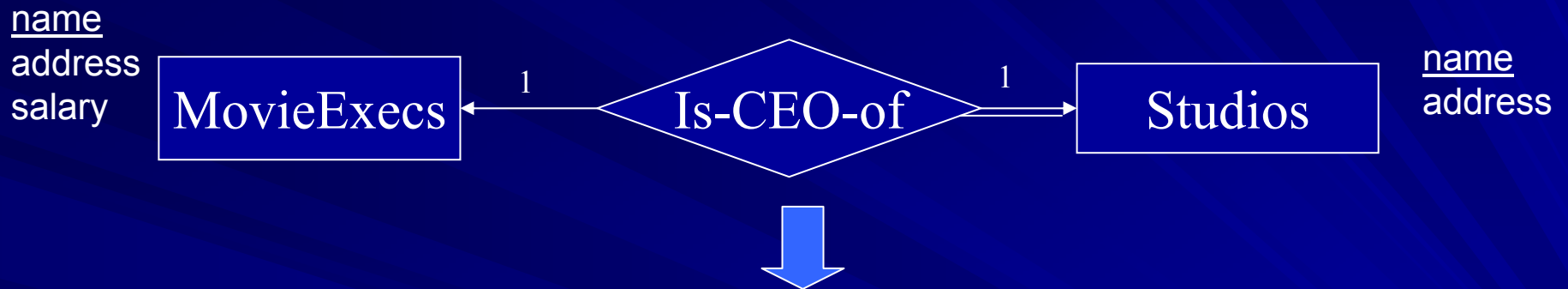
# Релация *Contracts*

- Релация *Contracts*, чиято схема се състои от атрибутите на ключовете на следните четири същности:
  - Ключът *starName* от *Stars*;
  - Ключът за *Movies*, състоящ се от *title* и *year*;
  - *studioOfStar* на *Studios*;
  - *producigStudio*, сочещ продуциращото студио.

# Contracts

<i>Movies</i>		<i>Stars</i>	<i>&lt;Stars-Studios&gt;</i>	<i>&lt;Movies-Studios&gt;</i>
<i>title</i>	<i>year</i>	<i>StarName</i>	<i>studioOfStar</i>	<i>Prod.Studio</i>
Star Wars	1977	Carrie Fisher	Fox	Fox
Star Wars	1977	Mark Hamill	Col.Tristar	Fox
Star Wars	1977	Harrison Ford	Fox	Fox
Mighty Ducks	1991	Emilio Estevez	Warner Bros	Disney
Wayne's World	1992	Dana Carvey	Fox	Paramount
Wayne's World	1992	Mike Meyers	MGM	Paramount

# Relationship Types - 1:1



- **MovieExecs**(MovieExecName, address, salary)
- **Studios**(studioAddress, studioName)
- **IsCEOOf**(MovieExecName, studioName)

# Relationship Types - 1:1



MovieExecs			Studios		
<u>name</u>	address	salary	<u>name</u>	address	ceoName
Eisner	Newport Beach	2.8M	Fox	Hollywood	Big
Lee	Brentwood	1M	Paramount	Hollywood	Lee
Big	Beverly Hills	3.8M	Disney	Buena Vista	Eisner
Smith	Malibu	0.9M			

- Не е необходимо връзката да се представя като отделна релация

# От E/R връзки към релации



Owns		
<i>title</i>	<i>year</i>	<i>studioName</i>
Star Wars	1977	Fox
Mighty Ducks	1991	Disney
Wayne's World	1992	Paramount

- Използвайте ключа на I същност и ключа на II същност
  - Name → studioName

# Комбиниране на релации (1)

■ Пример:

■ `Movies(title,year,length,filmType)` и  
`Owns(title,year,studioName)`

се комбинират

`Movies1(title,year,length,filmType, studioName)` .



# Комбиниране на релации (2)

Movies1(title,year,length,filmType, studioName)

<b>Movies1</b>	<i>title</i>	<i>year</i>	<i>length</i>	<i>filmType</i>	<i>studioName</i>
	Star Wars	1977	124	color	Fox
	Mighty Ducks	1991	97	color	Disney
	Wayne's World	1992	109	color	Paramount

# Комбиниране на релации (3)

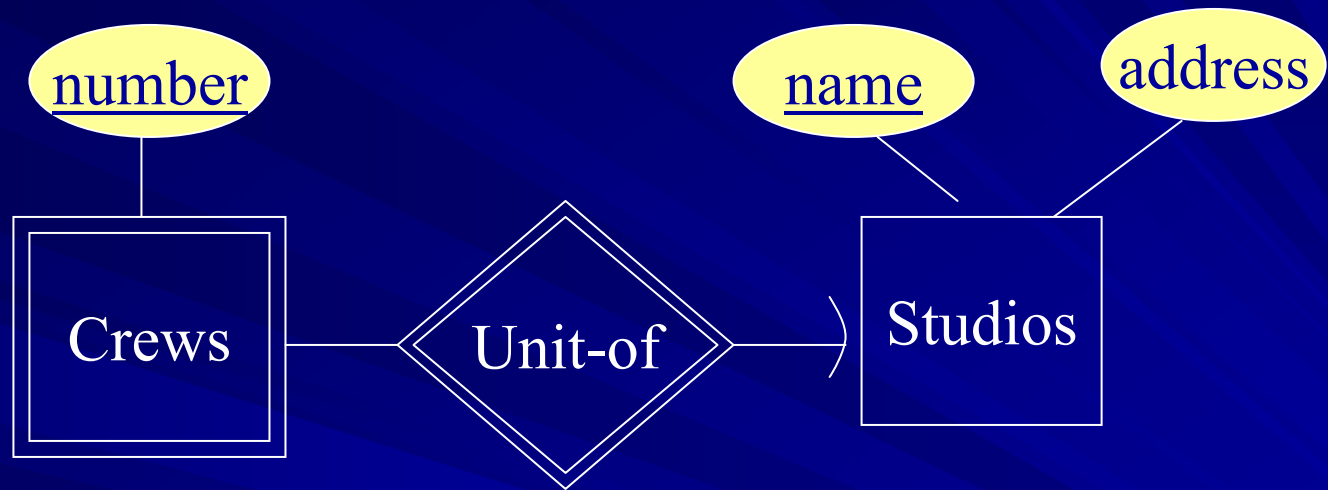
- Същност  $E$  с връзка  $R$  “М:1” от  $E$  към  $F$ 
  - > една релация със схема, състояща се от:
    - Всички атрибути на  $E$
    - Ключовите атрибути за  $F$
    - Атрибутите на връзката  $R$  (ако има такива)



# Слаби множества същности

- Релацията за слаба същност  $W$  трябва да включва не само неключовите атрибути на  $W$ , но също и ключовите атрибути на “поддържащата” същност
- Поддържащата връзка създава излишество и не се преобразува в отделна релация

# Пример



- Studios(name, address)
- Crews(number, studioName)
  - add non-key attributes
- Unit-of(number, studioName, name)

# Пример



Hosts(hostName)

Logins(loginName, hostName, time)

~~At(loginName, hostName, hostName2)~~

# ИЗВОДИ

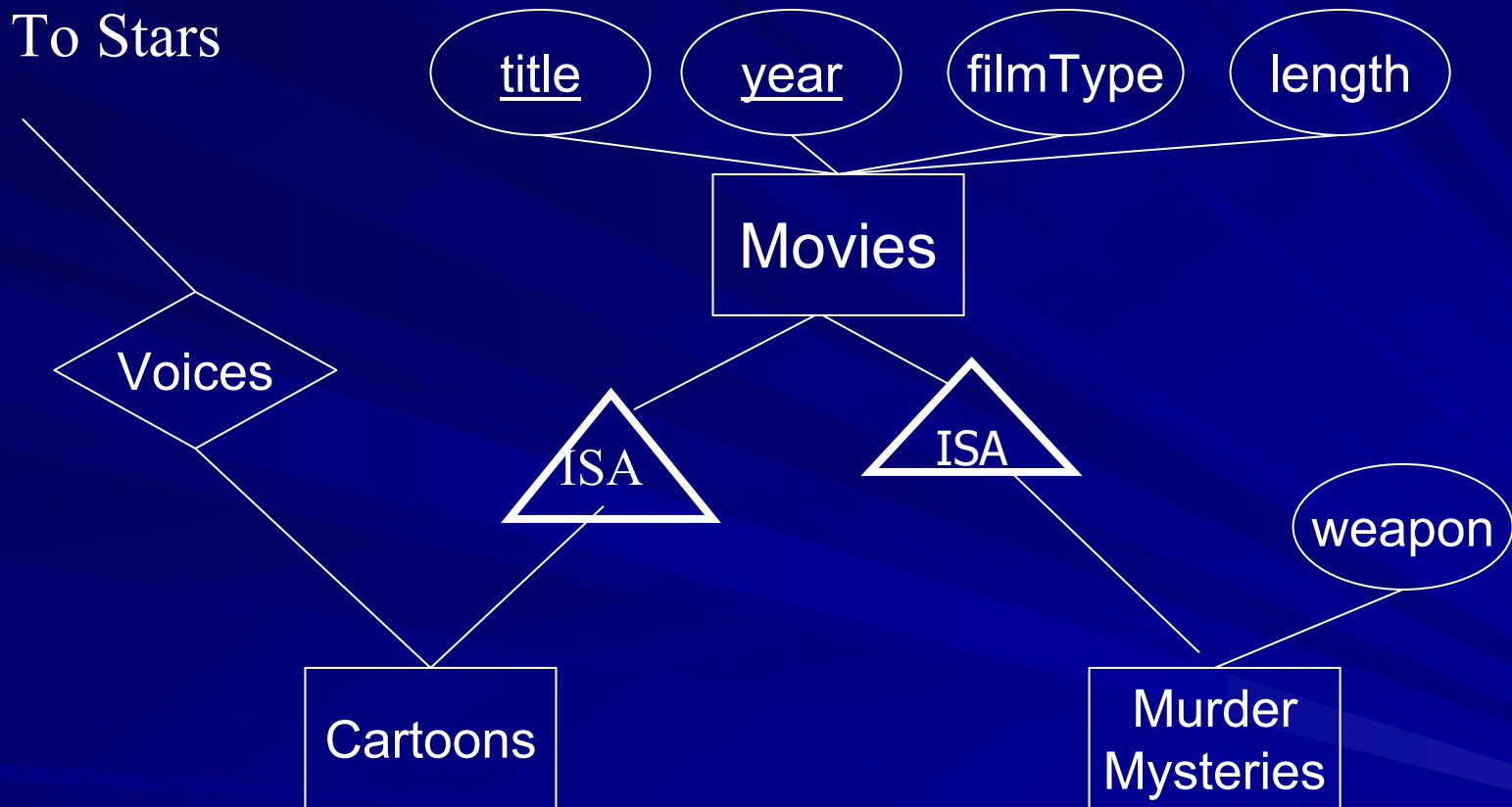
- Ако  $W$  е слаба същност, съставянето на релация за  $W$  ще бъде със схема от:
  - Всички атрибути на  $W$ ;
  - Всички атрибути на поддържащите връзки;
  - За всяка поддържаща връзка за  $W$ , която е “много към един” (М:1) от  $W$  към същност  $E$ , всички ключови атрибути на  $E$ .
- Не се съставя релация за никоя от поддържащите връзки към  $W$ !

# Конвертиране на йерархии от подкласове в релации

■ При “ISA” йерархия от същности:

- съществува същност, корен на йерархията;
- тази същност има ключ, който служи за идентифициране на всеки елемент, представен от йерархията;
- даден елемент може да има *компоненти*, които принадлежат на същностите на кое да е поддърво на йерархията, стига това поддърво да включва корена.

# Преобразуване на подкласове в релации



■ The Movie hierarchy



# М-ва същности с подкласове: 3 подхода

*Е/R подход*: за всяко м-во същности Е от йерархията създаваме релация, която включва ключовите арибути на корена и атрибитите на Е

*Обектно-ориентиран подход* : за всяко възможно поддърво, включително корена, се създава релация, която включва всички атрибути на всички м-ва същности от поддървото

*Null стойности* : създаваме 1! релация с всички атрибути на всички м-ва същности от йерархията

# Е/R подход

Създаване на релация за всяко м-во същности

- ако м-то същности не е корен, добавя се ключът на корена

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

# ОО подход

Има 4 възможни поддървета, вкл. корена:

*Само Movies*

Movies(title, year, length, filmType)

*Movies и Cartoons*

MoviesC(title, year, length, filmType)

*Movies и Murder-Mysteries*

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

Всички 3 м-ва същности

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

Връзка Voices – M:M

Voices(title, year, starName)

# Използване на Nulls

Ако ни е позволено да използваме null-стойности (the NULL value as in SQL) като стойности в кортежите, можем да пресъздадем йерархията на същностите с единствена релация

- всички атрибути от всички същности

Movies(title, year, length, filmType, weapon)

- Voices

Voices(title, year, starName)

# Сравнение на подходите

- Кой филми от 1999 са по-дълги от 150 мин.?
- Какви оръжия са използвани в cartoons, по-дълги от 150 мин.?

# Сравнение

- E/R подходът е добър при заявки от вида  
“What films of 1999 were longer than 150 minutes?”
  - only Movies relation.
- O-O подходът е добър при заявки от вида  
“What weapons were used in cartoons of over 150 minutes in length?”
  - MoviesCMM relation.
- Използването на nulls спестява място, освен в случаите на много атрибути, чиито стойности обичайно са *null*.