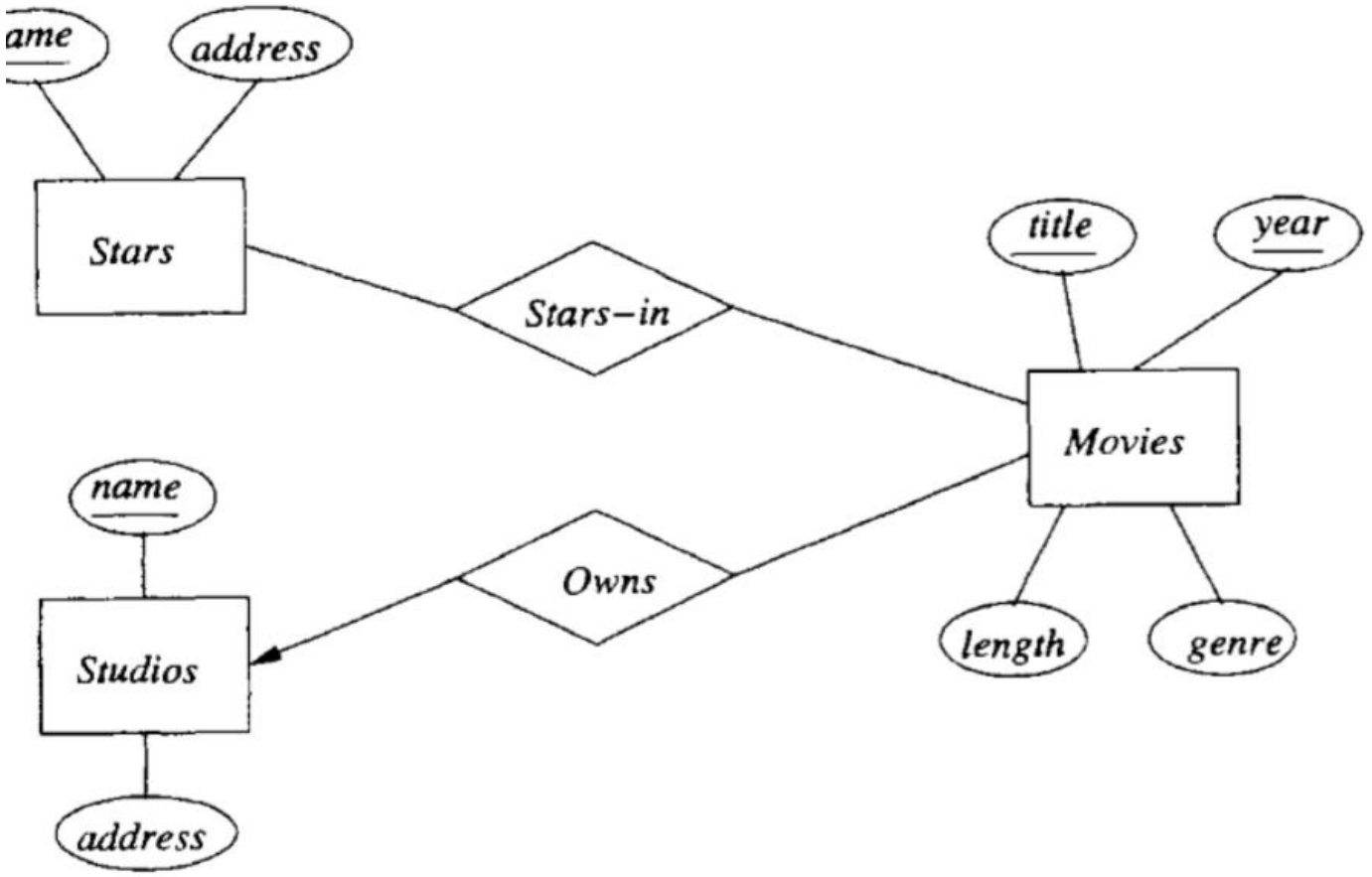
Mapiranje UML modela u relacioni

Primjer E/R dijagrama

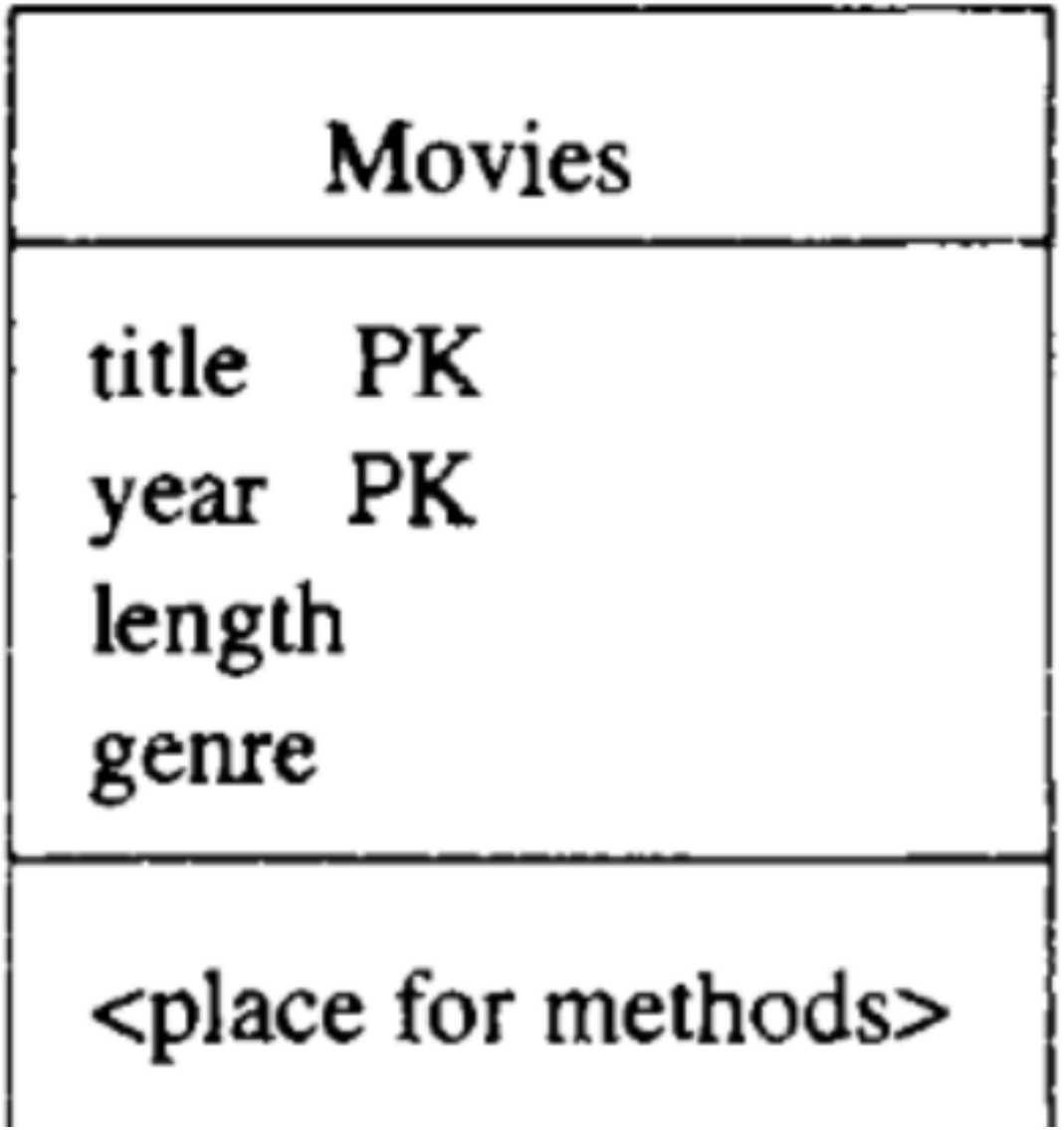


Terminologija

UML	E/R Model
Class	Entity set
Association	Binary relationship
Association Class	Attributes on a relationship
Subclass	Isa hierarchy
Aggregation	Many-one relationship
Composition	Many-one relationship with referential integrity

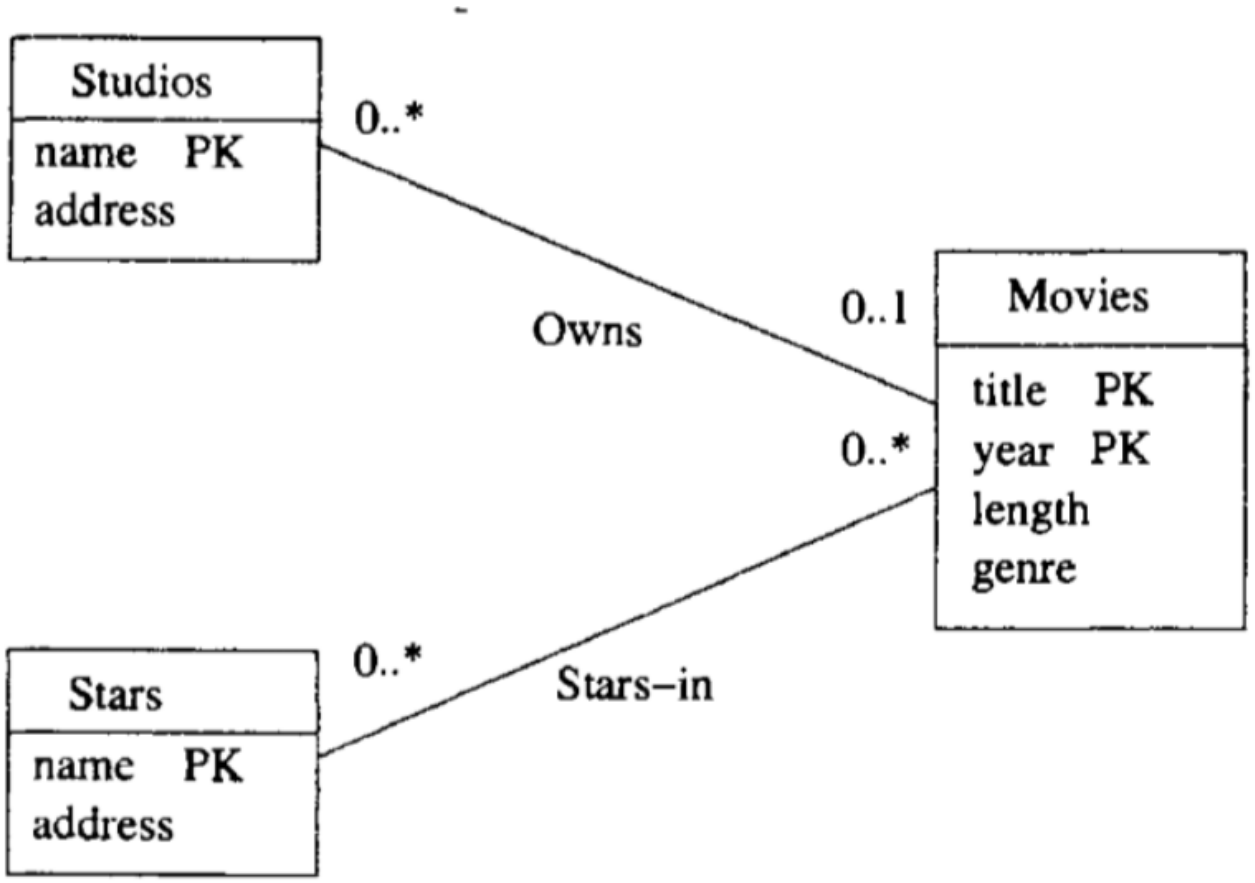
Klase

- U relacionom i E/R modelu nema metoda
- Ključevi
  - Primarni ključ - PK



Asocijacije

- Asocijacija je binarna relacija između klasa
  - Predstavlja skup parova objekata, po jedan iz klase koje su asocijacijom povezane
  - Ne postoji analogija u UML-u za n-arne relacije
- Multiplikatovnost *m..n*
  - Svaki objekat sa suprotnog kraja asocijacije povezan je sa najmanje *m* a najviše *n* objekata sa naznačene strane



Multiplikatovnost asocijacije

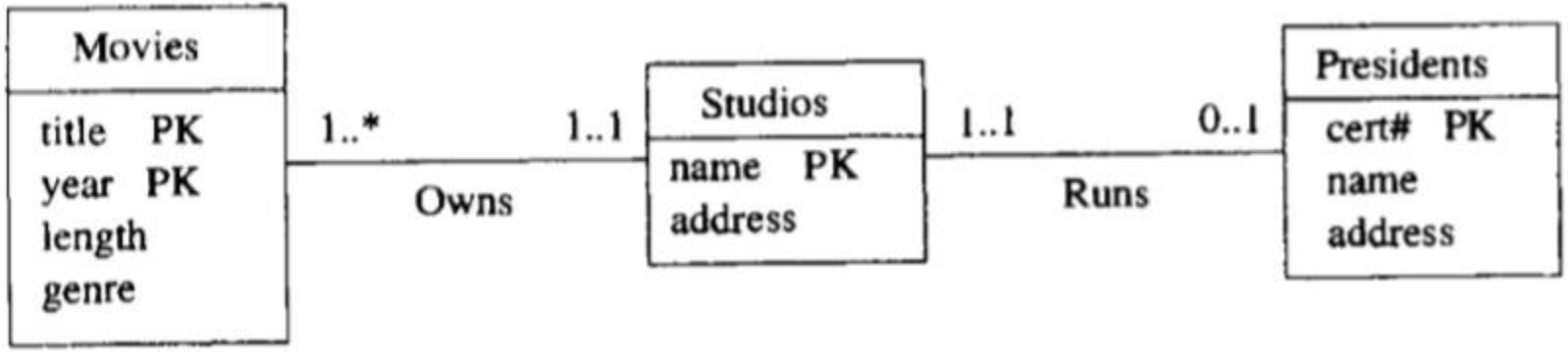
- Proširenja
  - *m..** - ne postoji gornje ograničenje
  - * - ekvivalentno sa *0..**, ne postoje ograničenja na broj objekata koji su povezani
  - Kada multiplikativnost nije naznačena podrazumijeva se 1-1

Referencijalni integritet

- Zaobljeni kraj strelice
  - Za svaki film moramo da znamo u kom studiju je snimljen
  - Svaki predsjednik mora da ima odgovarajući studio
  - Dozvoljeno je da studio nema predsjednika

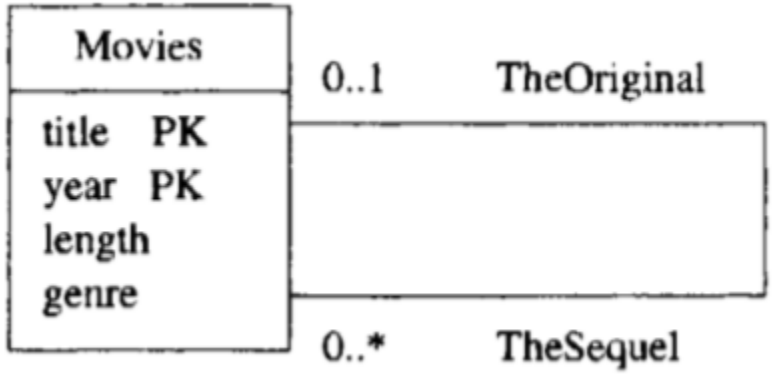
Referencijalni integritet - UML

- Owns
  - 1..* na strani Movies – svaki studio mora da posjeduje bar jedan film
  - 1..1 na strani Studios – film obavezno mora da pripada studiju i to tačno jednom
- Runs
  - 1..1 na strani Studios – predsjednik upravlja tačno jednim studijom
  - 0..1 na strani Presidents – studio može da ima najviše jednog predsjednika, ali je dozvoljeno i da nema predsjednika



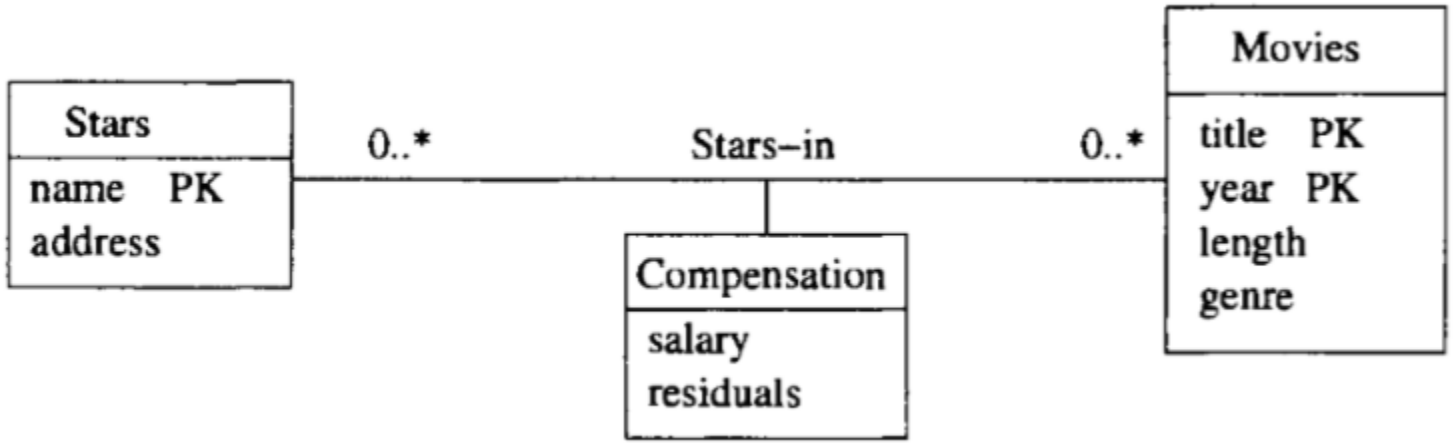
Self asocijacije

- Oba kraja asocijacije su na jednoj klasi
- Različite uloge
  - The Original, multiplikativnost 0..1, svaki nastavak ima najviše jedan original
  - TheSequel, multiplikativnost 0..*, original može da ima proizvoljan broj nastavaka



Asocijativne klase

- Asocijativna klasa ima naziv i attribute, koji su zapravo atributi veze
  - Svaki par film-glumac ima “svoju” novčanu nadoknadu



Podklase

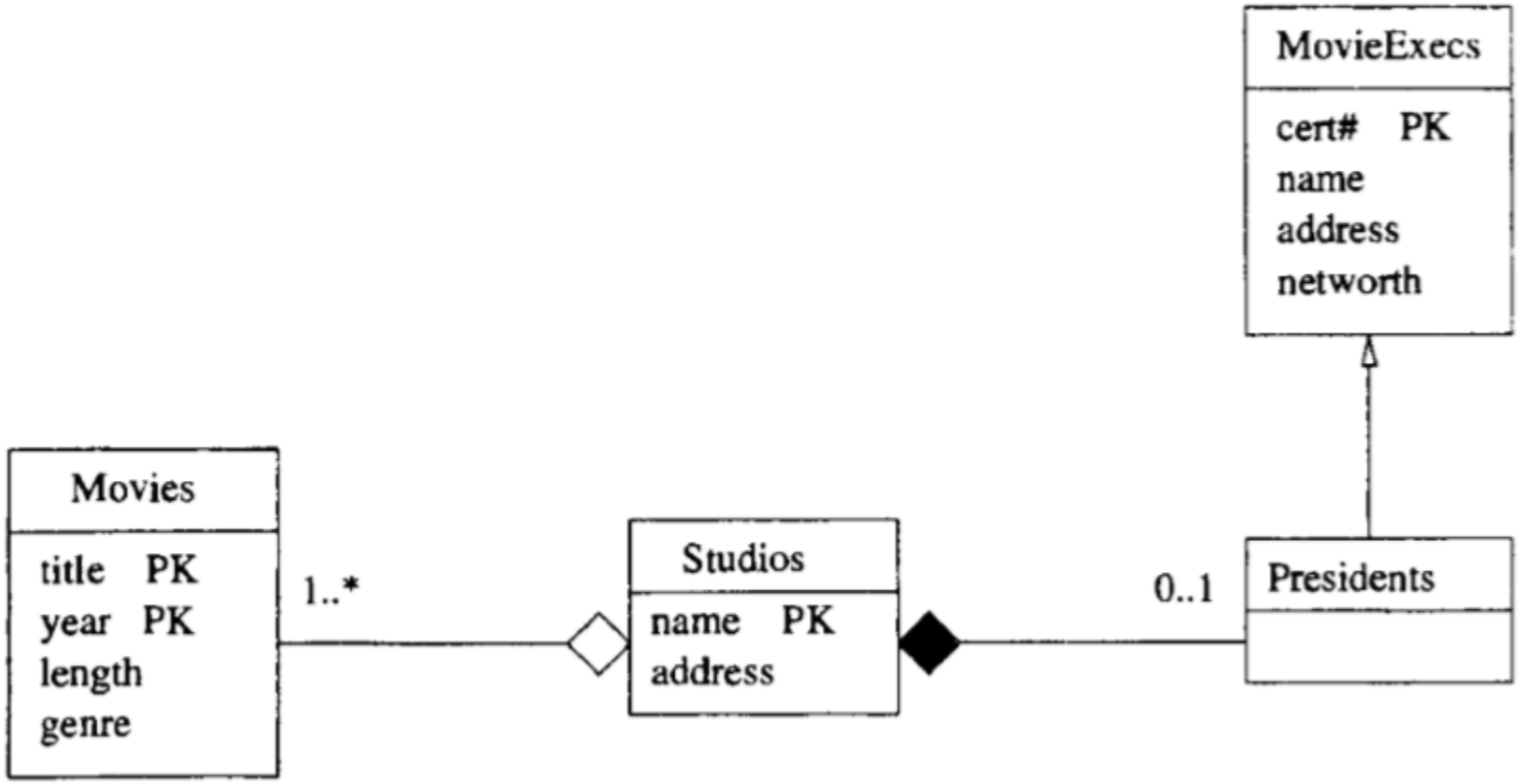
- Complete versus Partial
  - Svaki objekat korjenske klase pripada odnosno ne pripada nekoj od podklasa
- Disjoint versus Overlapping
  - Svaki objekat pripada isključivo jednoj podklasi
- Prethodna svojstva odnose se na svaki čvor u hijerarhiji
- Tipično je da su u OO sistemima podklase disjunktne, podrazumijeva se parcijalna hijerarhija
- ER model podrazumijevano dozvoljava preklapanja u hijerarhiji, podrazumijeva se parcijalna hijerarhija

Agregacija i kompozicija

- Agragacija je više u jedan veza, multiplikativnost na strani „romba“ je 0..1
- Kompozicija ima

multiplikativnost 1..1

na strani „romba“



Agregacija i kompozicija 2

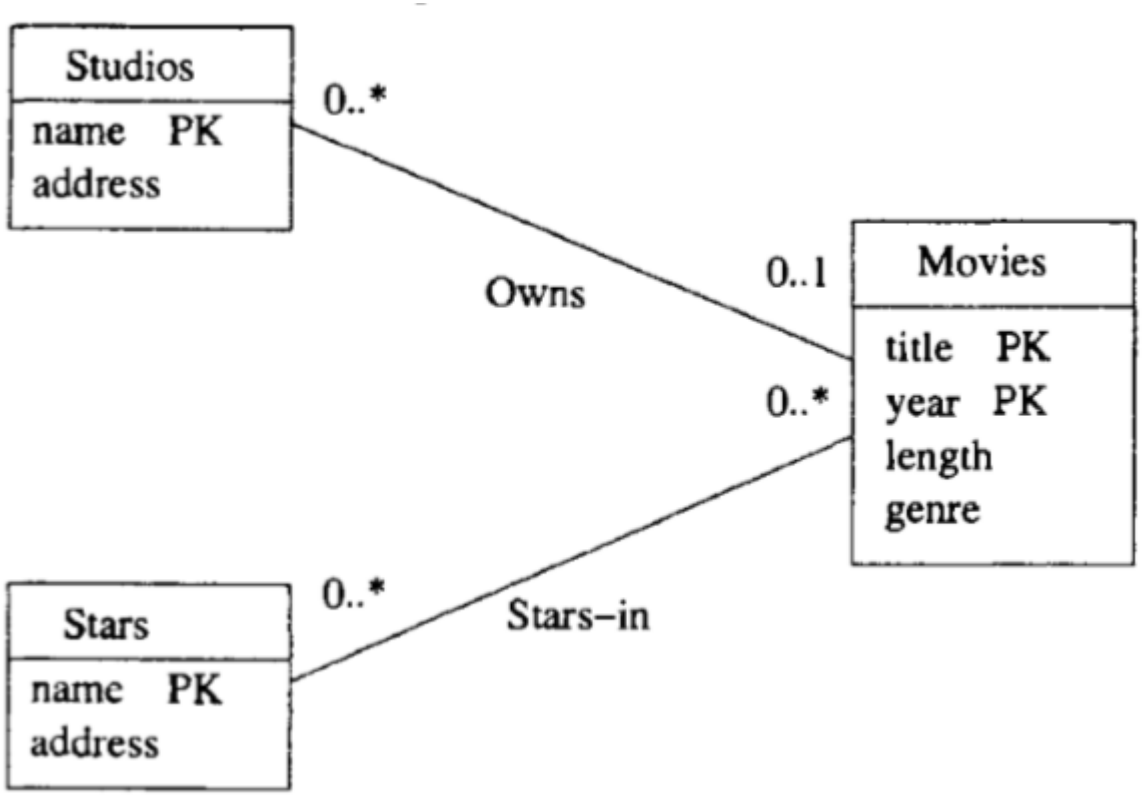
- Movies – Studios
  - „prazni romb“ – multiplikativnost 0..1
  - Objekti klase Movies sadržaće referencu na objekat klase Studios, može li ova referenca da bude NULL?
- President – Studios
  - Multiplikatovnost na strani Studios je 1..1 (ispunjeni romb)
  - Objekti klase President sadržaće referencu na Studios
  - Može li ova referenca da bude NULL ?

Prevođenje UML u relacioni model

- Mapiranje klasa u relacije
  - Za svaku klasu kreira se posebna relacija

- Mapiranje asocijacija u relacije
  - Za svaku asocijaciju kreira se posebna relacija
  - Atributi ove relacije su primarni ključevi povezanih klasa, zajedno sa atributima asocijativne klase ako ona postoji
  - Dozvoljeno je preimenovanje atributa

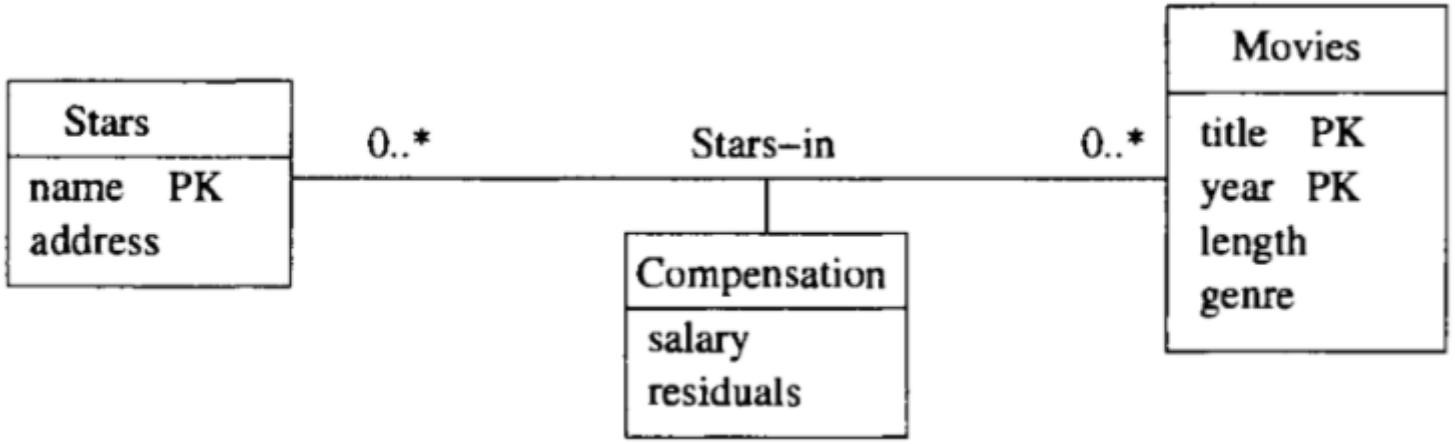
Primjer




```
Movies(title, year, length genre)
Stars(name, address)
Studios(name, address)

Stars-In(movieTitle, movieYear, starName)
Owns(movieTitle, movieYear, studioName)
```

Primjer 2

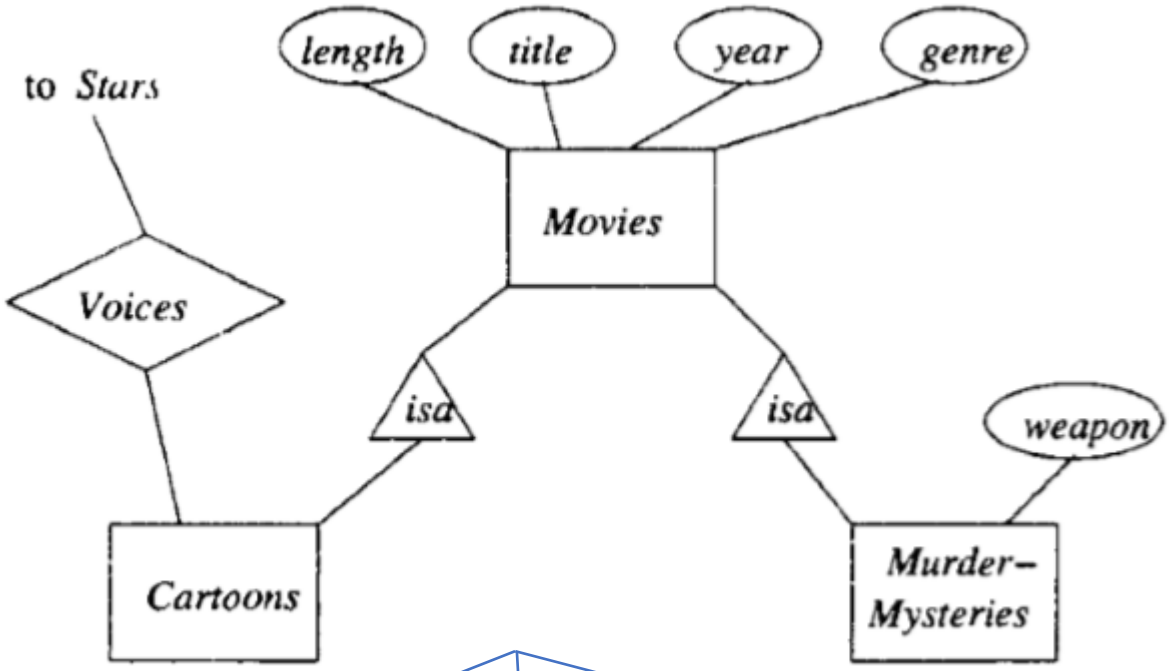



```
Stars-In(movieTitle, movieYear, starName, salary, residuals)
```

Prevođenje UML u relacioni model 2

- Prevođenje podklasa iz UML u relacije
  - E/R strategija, za svaku klasu E iz hijerarhije kreira se relacija koja sadrži atribute klase E zajedno sa ključevima iz svih nadklasa
  - OO strategija, za svako podstablo koje uključuje korijen hijerarhije, kreira se jedna relacija sa atributima iz svih klasa unutar tog podstabla
    - Entiteti su objekti koji pripadaju jednoj i samo jednoj klasi
  - Use null values strategija
    - Jedna relacija koja sadrži sve atribute koji se pojavljuju u hijerarhiji
    - NULL vrijednost se postavlja ako atribut ne pripada klasi

Primjer




```
Movies(title, year, length, genre)
MurderMysteries(title, year, weapon)
Cartoons(title, year)
```

```
Movie(title, year, length, genre, weapon)
```

```
Movies(title, year, length, genre)
MoviesC(title, year, length, genre)
MoviesMM(title, year, length, genre, weapon)
MoviesCMM(title, year, length, genre, weapon)
```

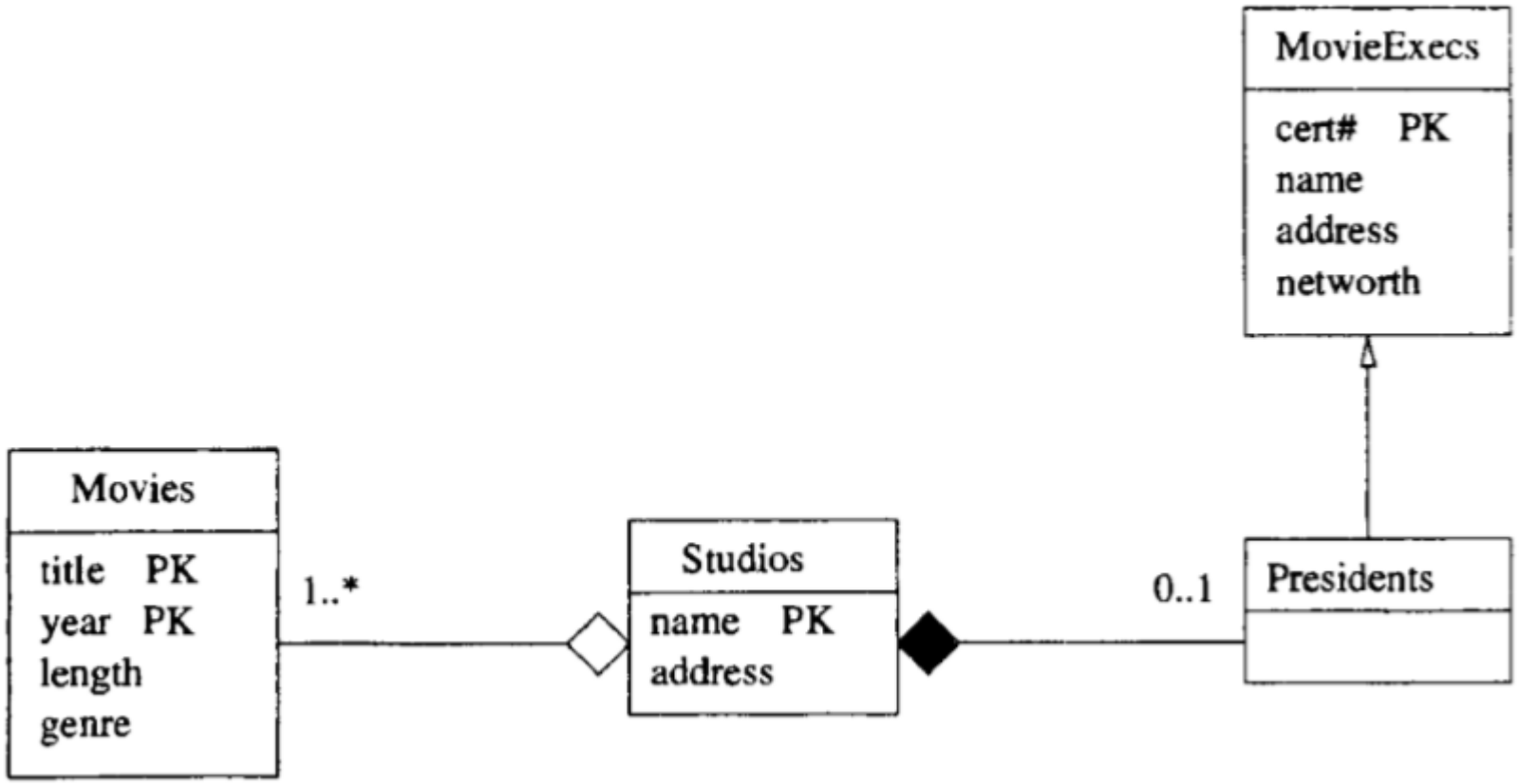
Preporuke

- Ako je hijerarhija disjoint na svakom nivou, koristi se OO strategija
- Ako je hijerarhija complete i disjoint, koristi se OO strategija a relacije kreiraju samo sa listove u hijerarhiji
- Kada je hijerarhija velika i overlapping, koristi se E/R strategija

Prevođenje UML u relacioni model 2

- Prevođenje agregacije i kompozicije
  - Ne kreiraju se posebne relacije već se dodaju ključni atributi „romb“ strane skupu atributa suprotne strane
    - za agregaciju je dozvoljeno na atributi ključevi budu NULL

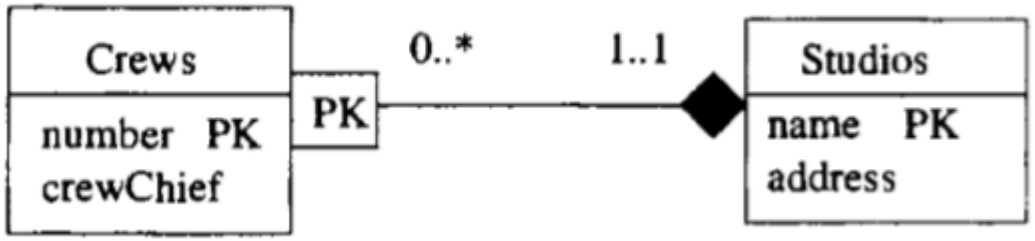
Primier




```
MovieExecs(cert#, name, address, netWorth)
Presidents(cert#, studioName)
Movies(title, year, length, genre, studioName)
Studios(name, address)
```

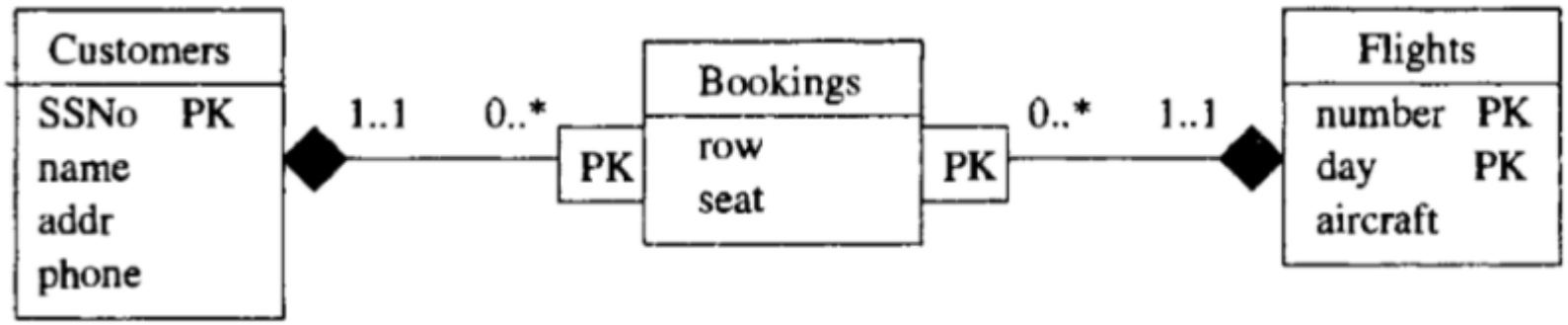
Slabi skup entiteta

- Object identity – referenca na objekat
  - Po ovoj referenci je moguće razlikovati objekte i kada imaju jednake vrijednosti svih atributa

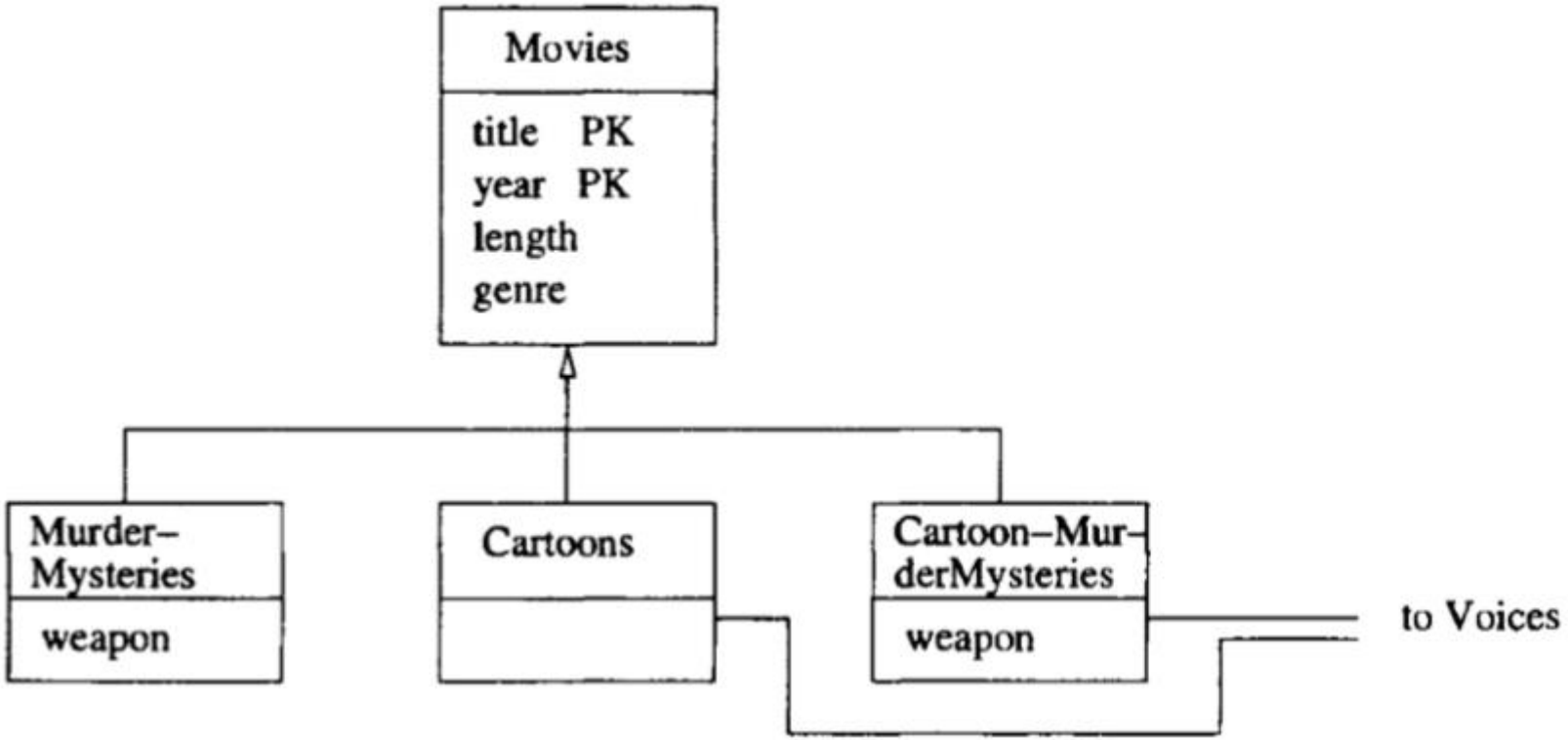



```
Studios(name, address)
Crews(number, crewChief, studioName)
```

Konvertovati sljedeće dijagrame u relacije



Konvertovati sljedeće dijagrame u relacije



Konvertovati sljedeće dijagrame u relacije