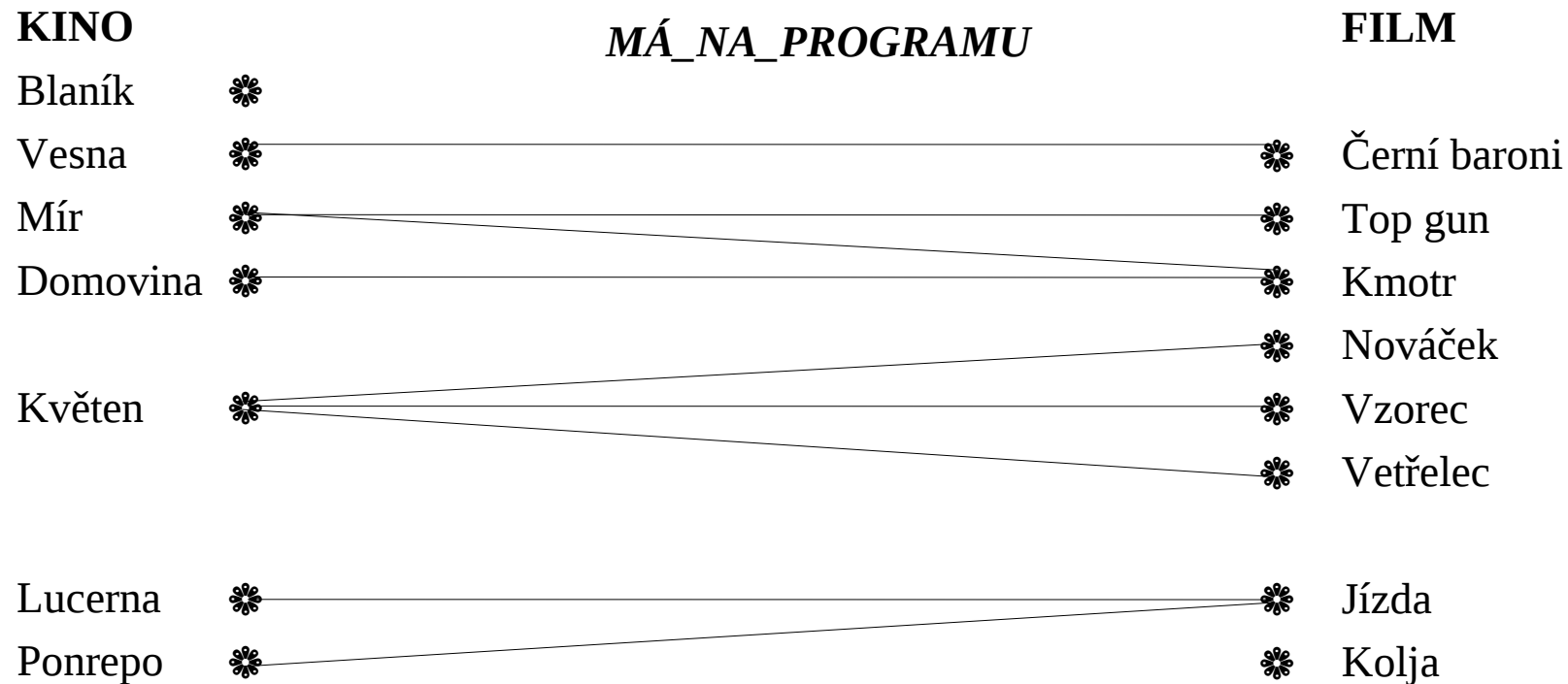


# Diagram výskytů a vztahů

Nepoužívá se pro modelování.

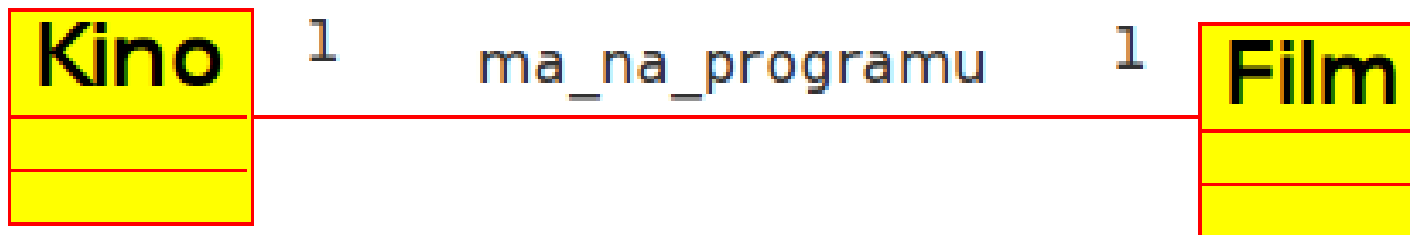
Pomůcka pro pochopení kardinalit a parcialit.



M:N

# Násobnost účasti ve vztahu (**kardinalita**) 1:1

KINO		<i>MÁ NA PROGRAMU</i>		FILM
Blaník	*			
Vesna	*	—————	*	Černí baroni
Mír	*	—————	*	Top gun
Domovina	*	—————	*	Kmotr
			*	Nováček
Květen	*		*	Vzorec
			*	Vetřelec

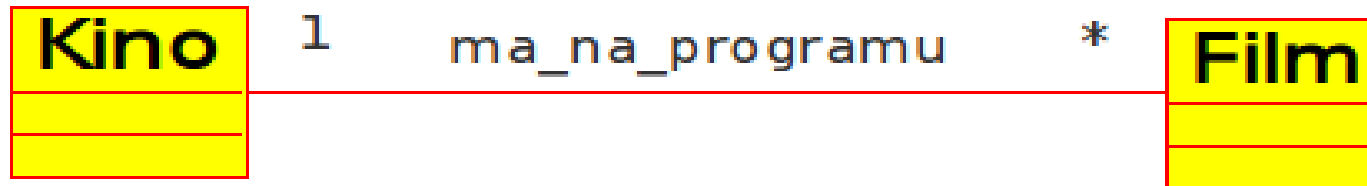
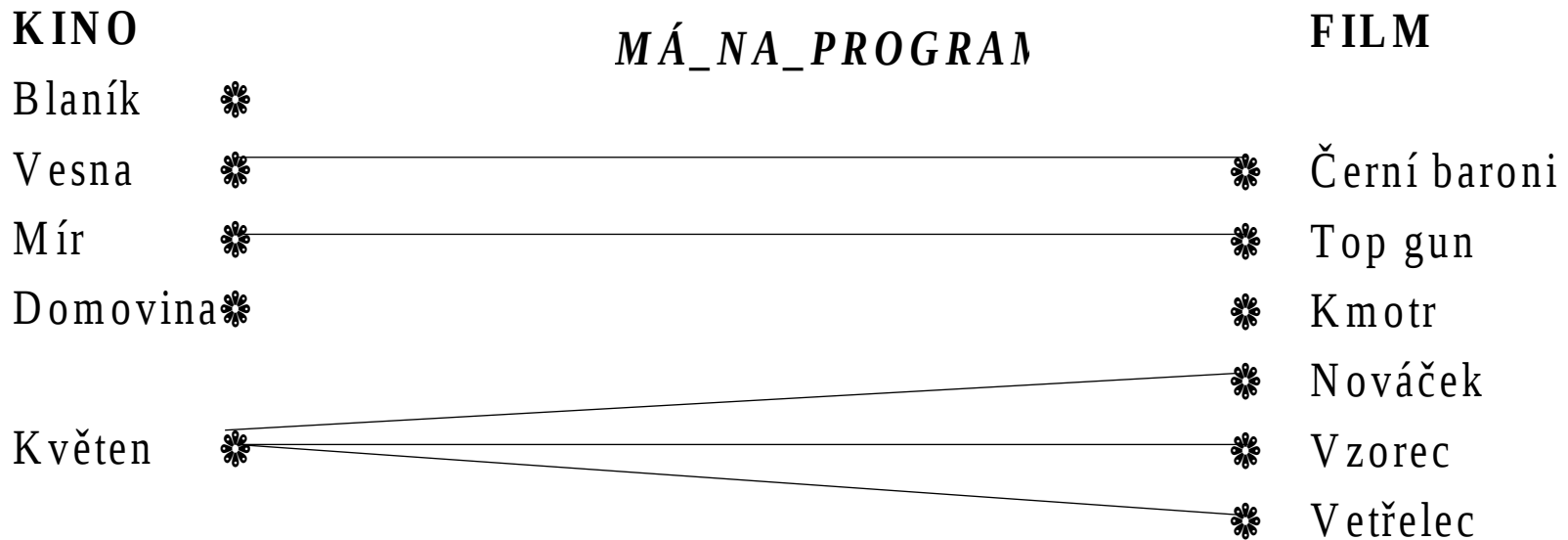


Kino hraje nejvýše **jeden** film.

Film je na programu nejvýš **jednoho** kina.

Povinnost/nepovinnost účasti ve vztahu je diskutována dále.

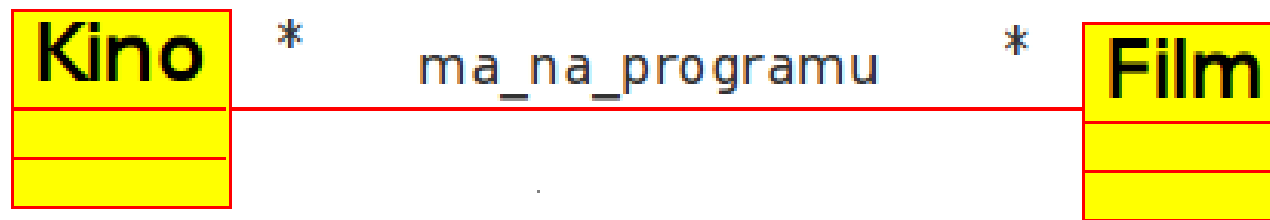
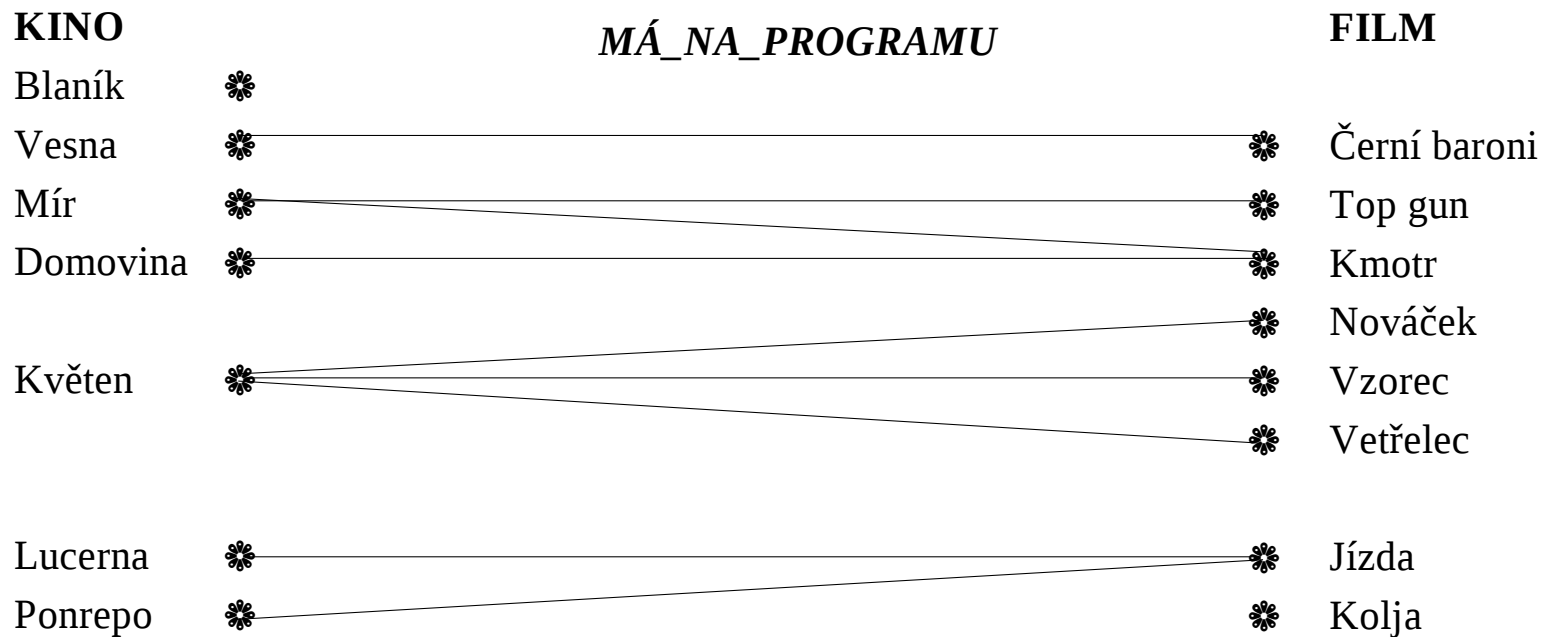
## Kardinalita vztahu 1:N



Kino může hrát **více** filmů.

Film je na programu nejvýše **jednoho** kina.

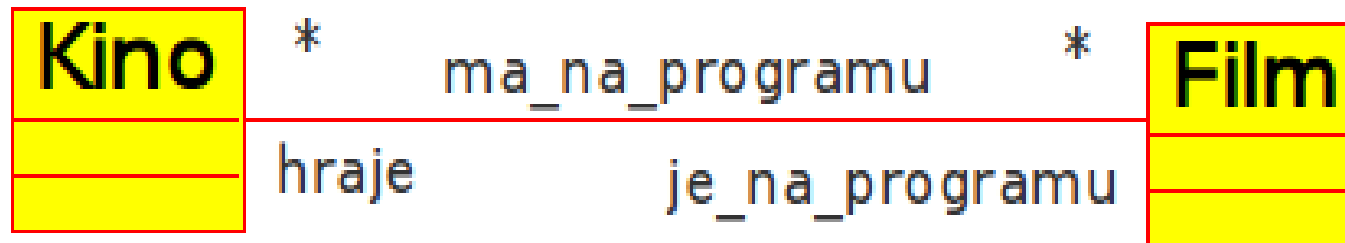
# Kardinalita vztahu M:N



Kino může hrát **více** filmů.

Film může být na programu **více** kin.

## *role účastníka*



### Použití:

- čitelnost schématu

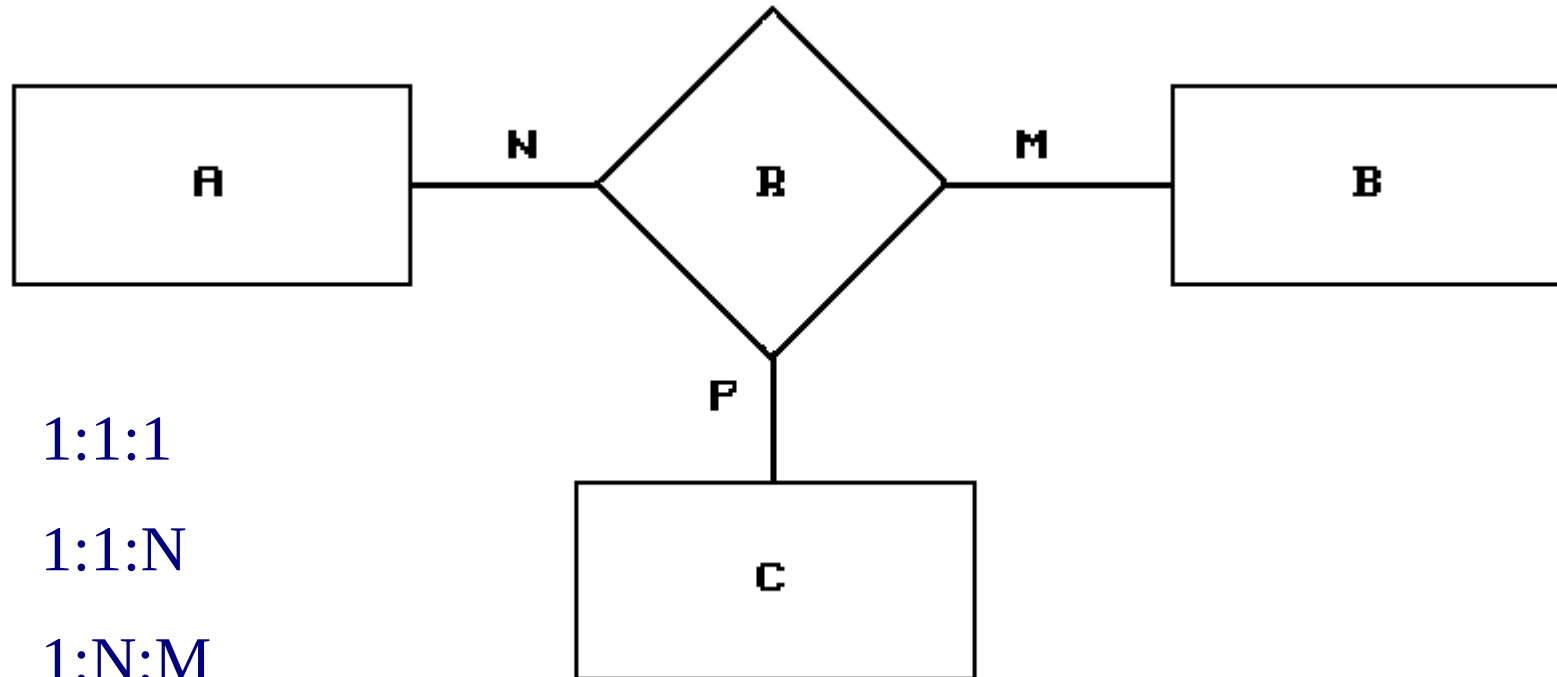
Kino hraje film. Film je na programu kina.

- implementace (v relačním modelu)

Jméno atributu, na kterém je IO cizí klíč (realizuje vztah), přebírají roli (nebo její zkratku) jako prefix. Nalomení nechtěných cyklů (viz přednáška o transformaci konc. modelu na relační).

## *N-ární vztah*

Tohle není UML!



1:1:1

1:1:N

1:N:M

M:N:P

Používá se spíše v nejvyšších úrovních abstrakce.  
Dekompozicie – vztha se změří na vztahovou entitu.

## *Povinnost/nepovinnost účasti ve vztahu (**parcialita**)*

**Povinná účast:** všechny výskyty účastníka vztahu (instance) **musí** být zapojeny do příslušného vztahu.

*Každé kino **musí** mít na programu alespoň jeden film.  
Nepřipouštíme kina, která nic nehrají.*

**Nepovinná účast:** jednotlivé výskyty účastníka vztahu (entity) **mohou , ale nemusí** být zapojeny do vztahu.

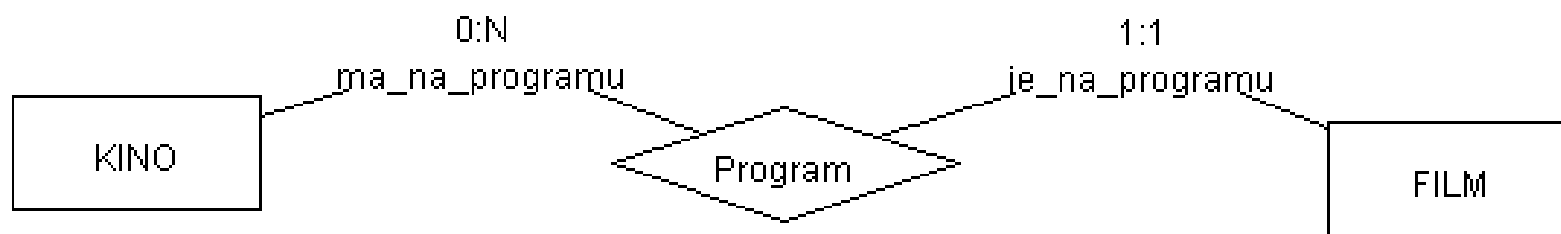
*Kino **může** být evidováno i bez programu.  
Připustíme i taková kina, která nic nehrají.*

### **Notace:**

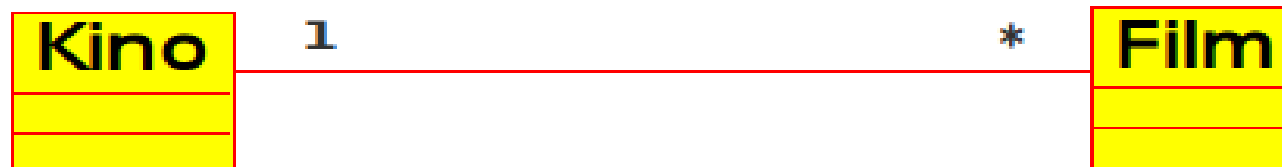
**1** (zkratka pro 1..1)  
**\*** (zkratka pro 0..\*)

**0..1**  
**1..\***

*Kino může hrát více filmů (ale také žádný).  
Film je na programu právě jednoho kina.*



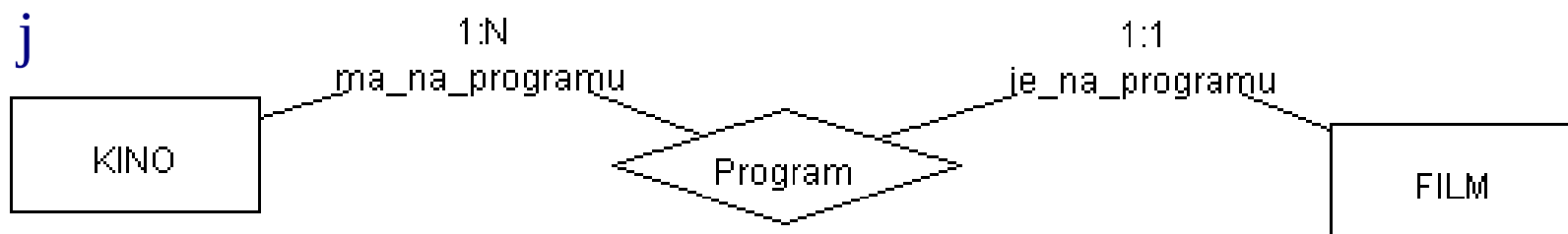
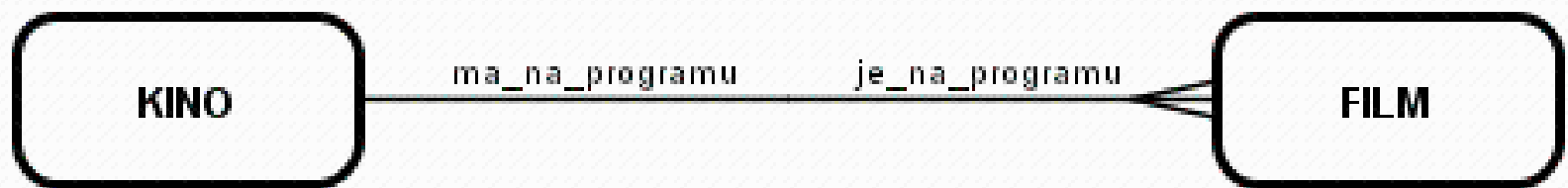
**1 = 1..1**



**\* = 0..\***



*Kino hraje alespoň jeden film (ale může více).  
Film je na programu právě jednoho kina..*



Jak lze do této databáze vložit???

# Atributy vztahu, vztah M:N

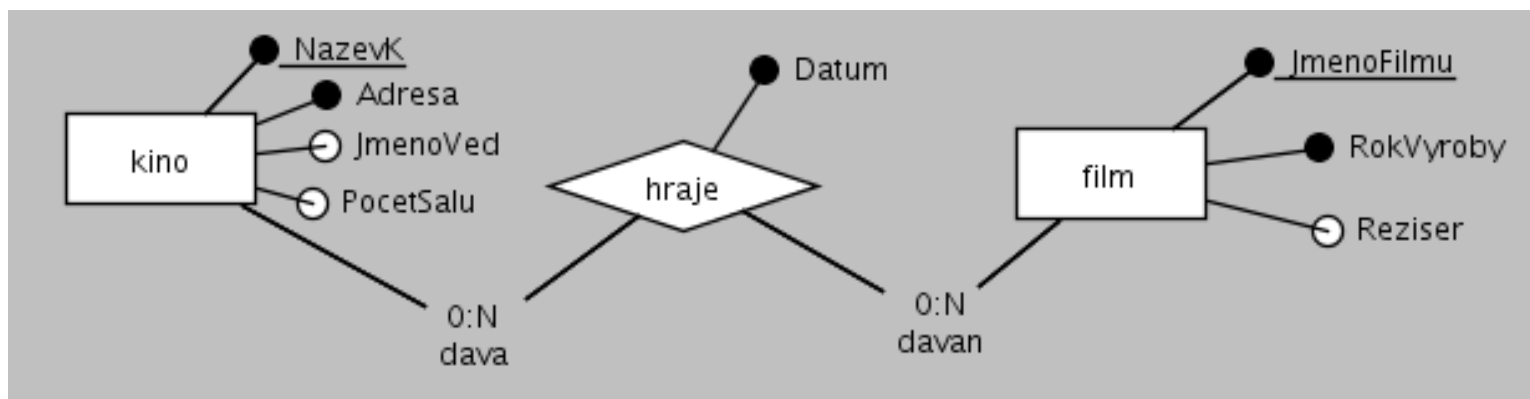
E: kino (NazevK, Adresa, JmenoVed, PocetSalu)

film (JmenoFilmu, RokVyroby, Reziser)

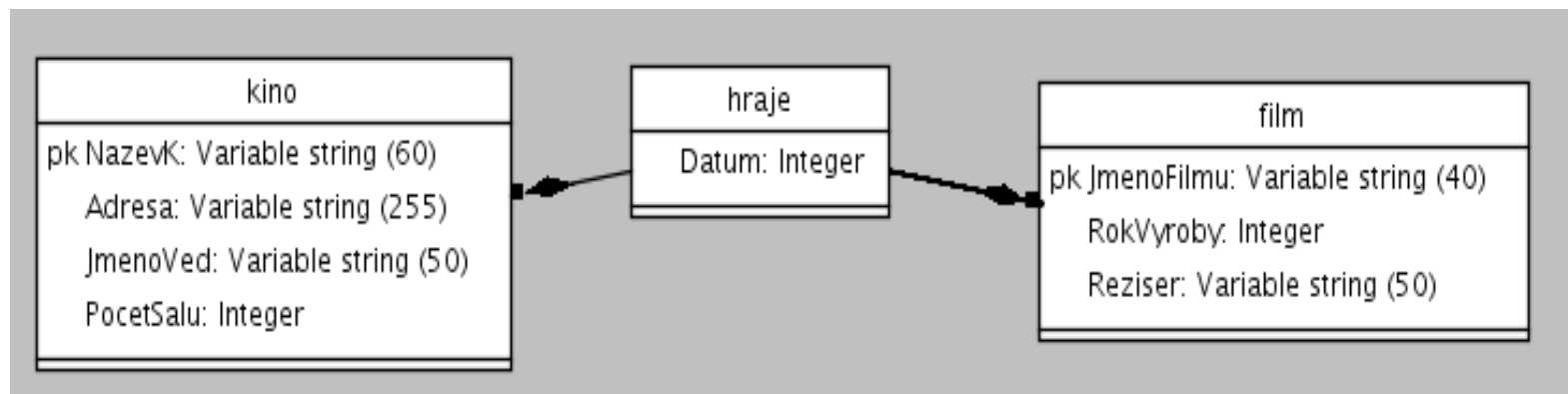
R: hraje (KINO, FILM; Datum)

**Ale pozor!!!  
F1 v K1 max jednou!**

Chenova notace

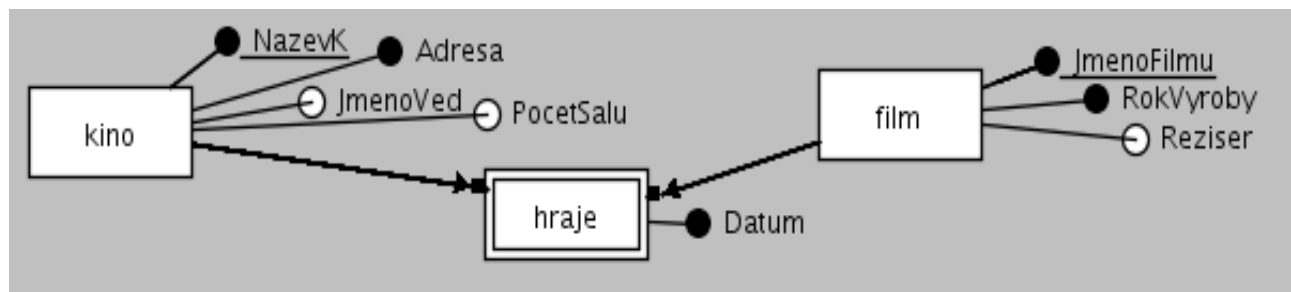
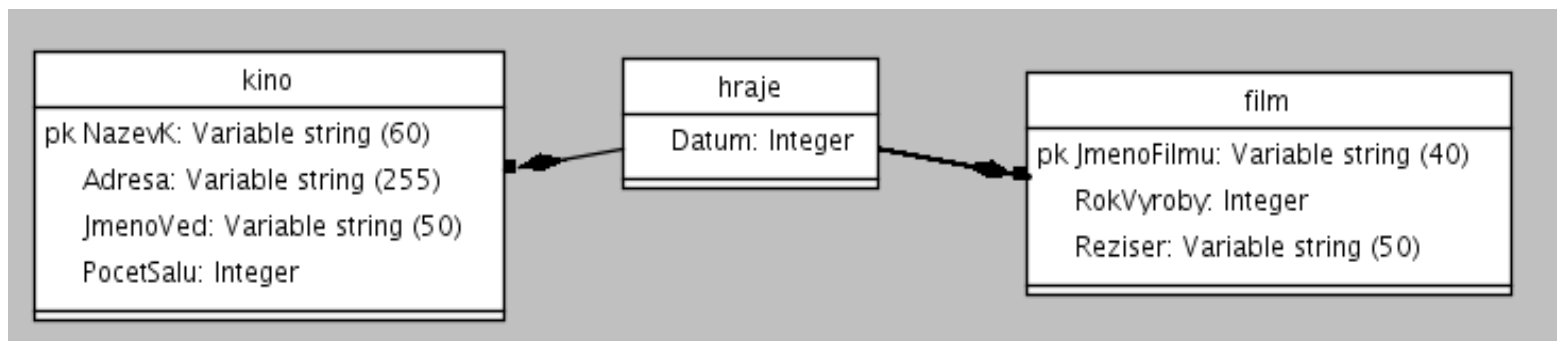
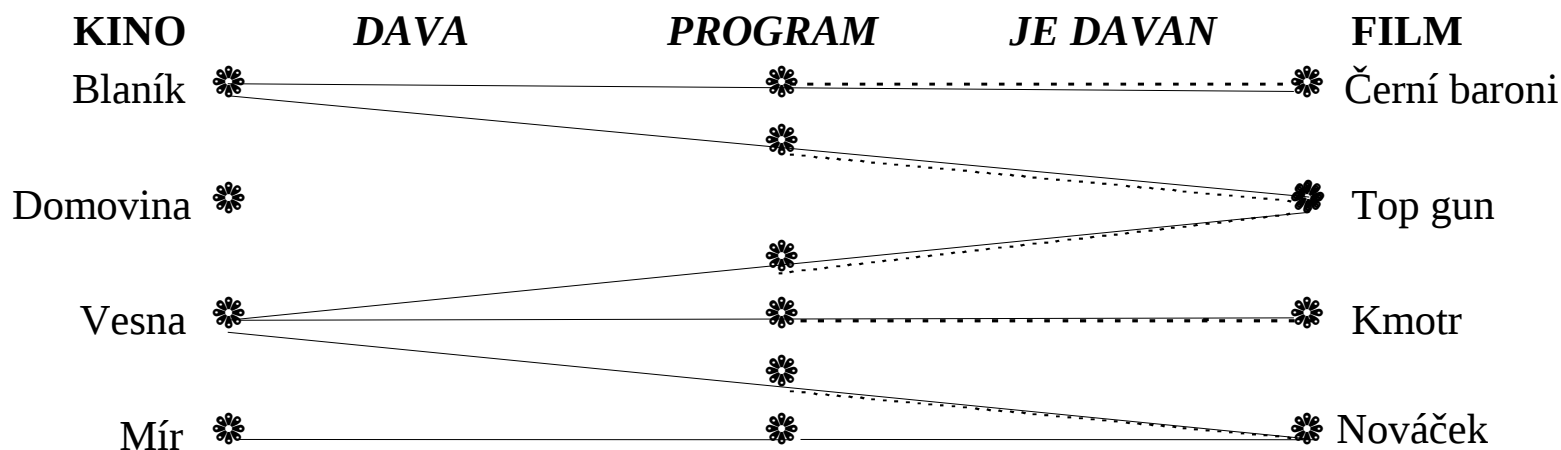


UML



**UML nepřipouští atributy u vztahů (nutná dekompozice).**

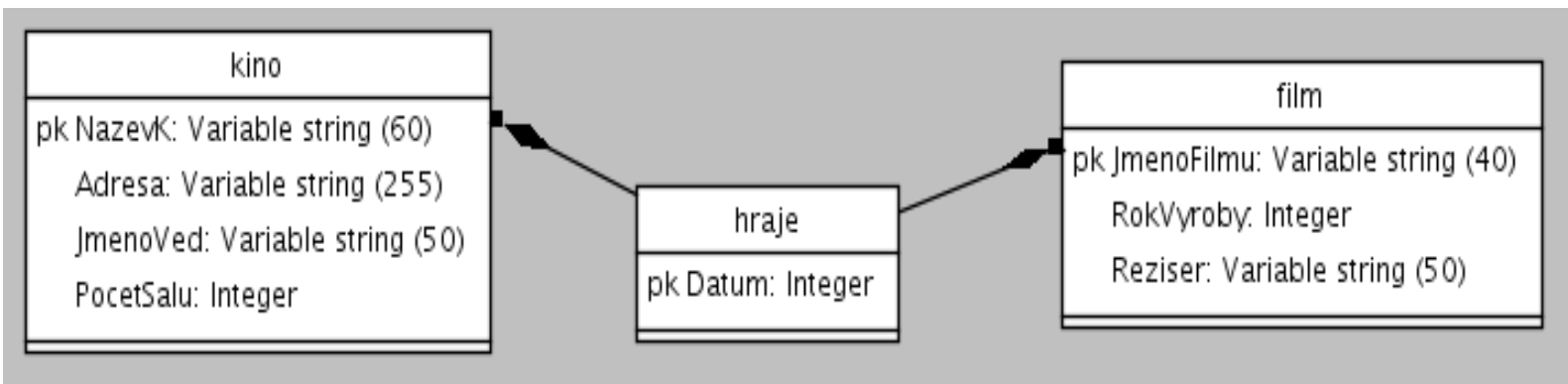
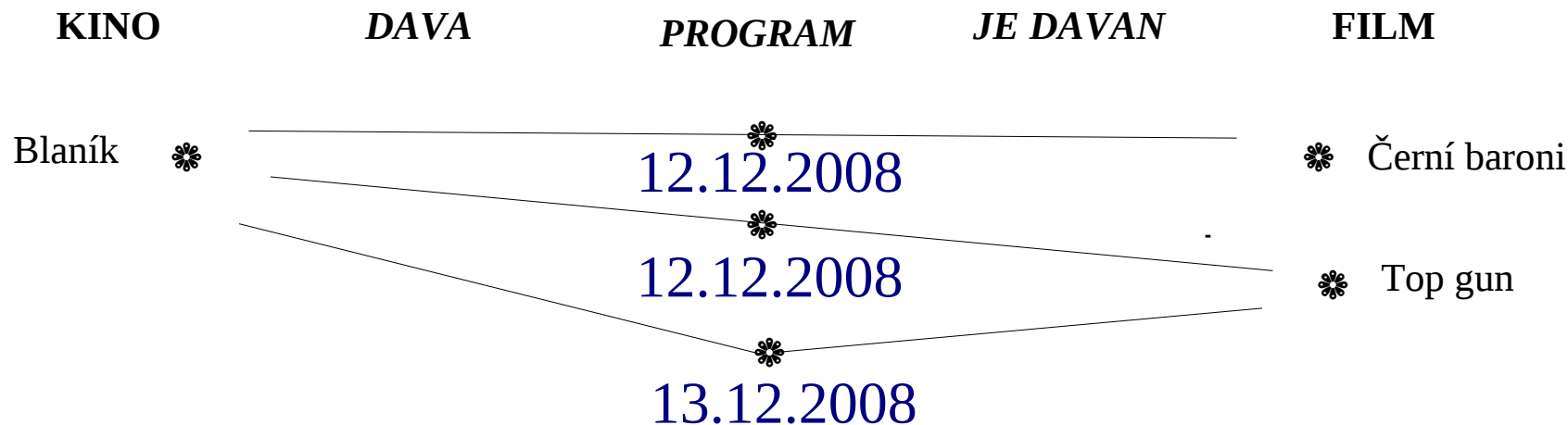
# Dekompozice vztahu M:N



**Ale pozor!!!**  
**F1 v K1 max jednou!**

Opravdu jsme to tak  
 chtěli modelovat?

*F1 v K1 vícekrát, ovšem v jiné dny.*

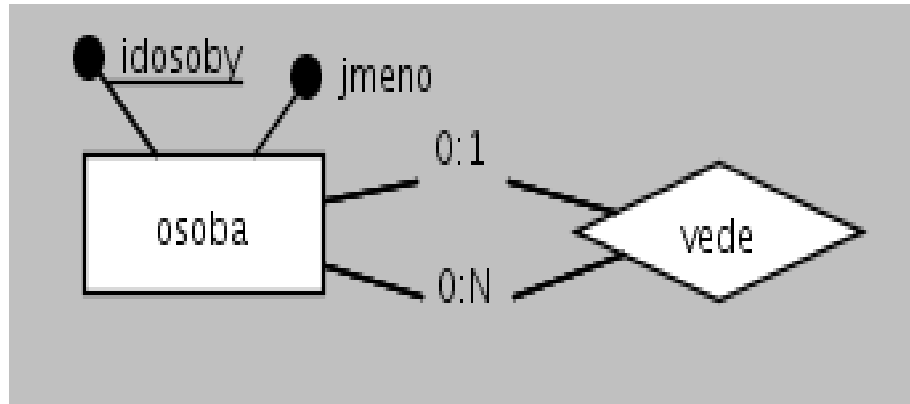


**Pozor na to, jak volíte identifikátory!!!**

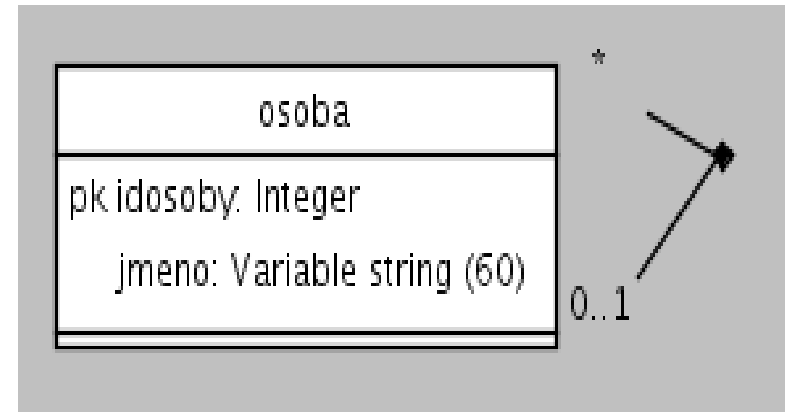
Alternativou je umělý identifikátor entity hraje.

# Rekurzivní typ vztahu

## Chenova notace



## UML



Daná OSOBA může vést mnoho OSOB

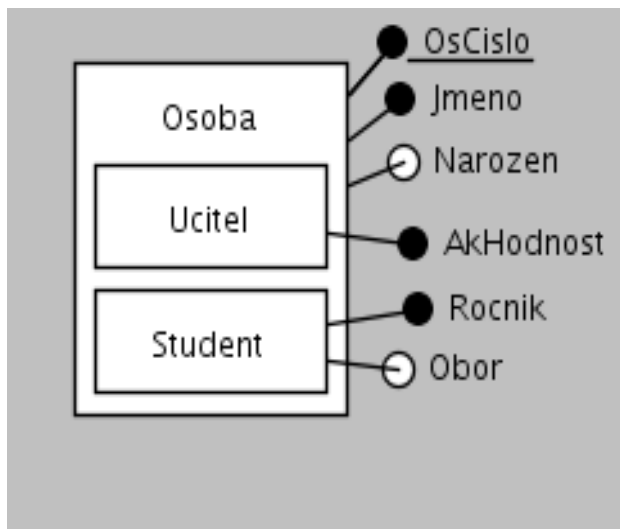
Daná OSOBA může být vedena nejvýše jednou OSOBOU

# ISA hierarchie

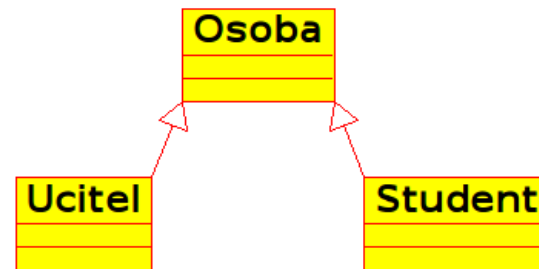
Osoba (OsCislo, Jmeno, Narozen, ... );  
Ucitel(AkHodnost) **ISA** Osoba;  
Student(Rocnik, Obor) **ISA** Osoba;

Správně pro každý nadtyp  
právě jedna instance  
podtypu!

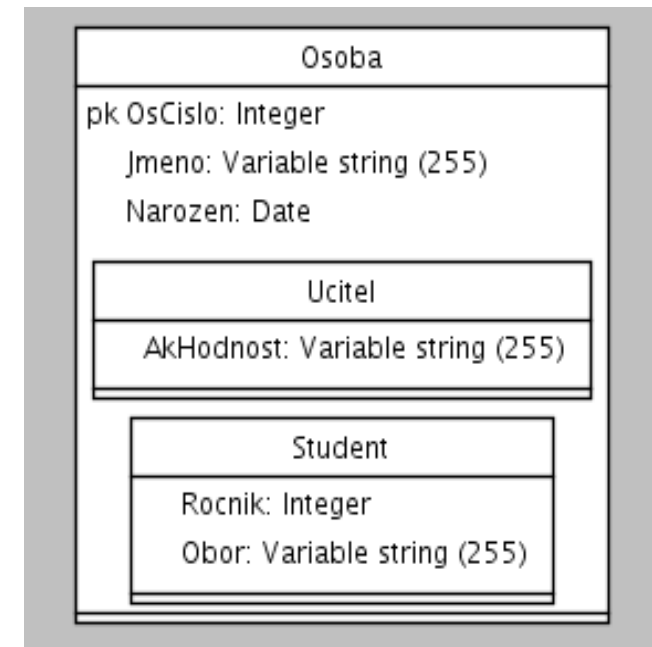
## Chenova notace



## UML



## UML (ER modelar)



Pro implementaci v relačním modelu opatrně!!!

# Sémantický relativismus

PACIENT(ROD\_ČÍS, JMÉNO, ADR, VÁHA, VÝŠKA, **POČET\_LŮŽEK**)

*Popisuje přidaný atribut vlastnost entit typu PACIENT?*

*typy entit:*

PACIENT(ROD\_ČÍS, JMÉNO, ADR, VÁHA, VÝŠKA )

POKOJ(ČÍSLO\_POKOJE, POČET\_LŮŽEK)

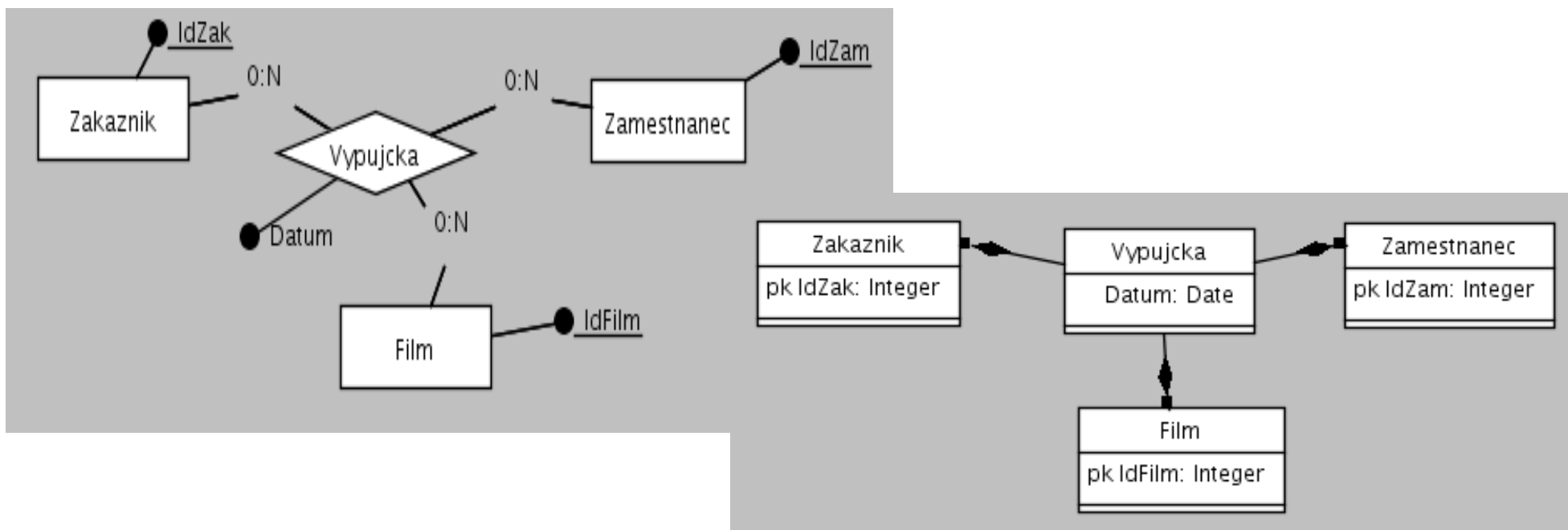
*typy vztahů:*

JE\_UMÍSTĚN\_NA(PACIENT: (1,1), POKOJ: (0,N))

*Je tím ztracena informace na kolikalůžkovém pokoji pacient leží?*

# Příklad – návrh videopůjčovny 1/4

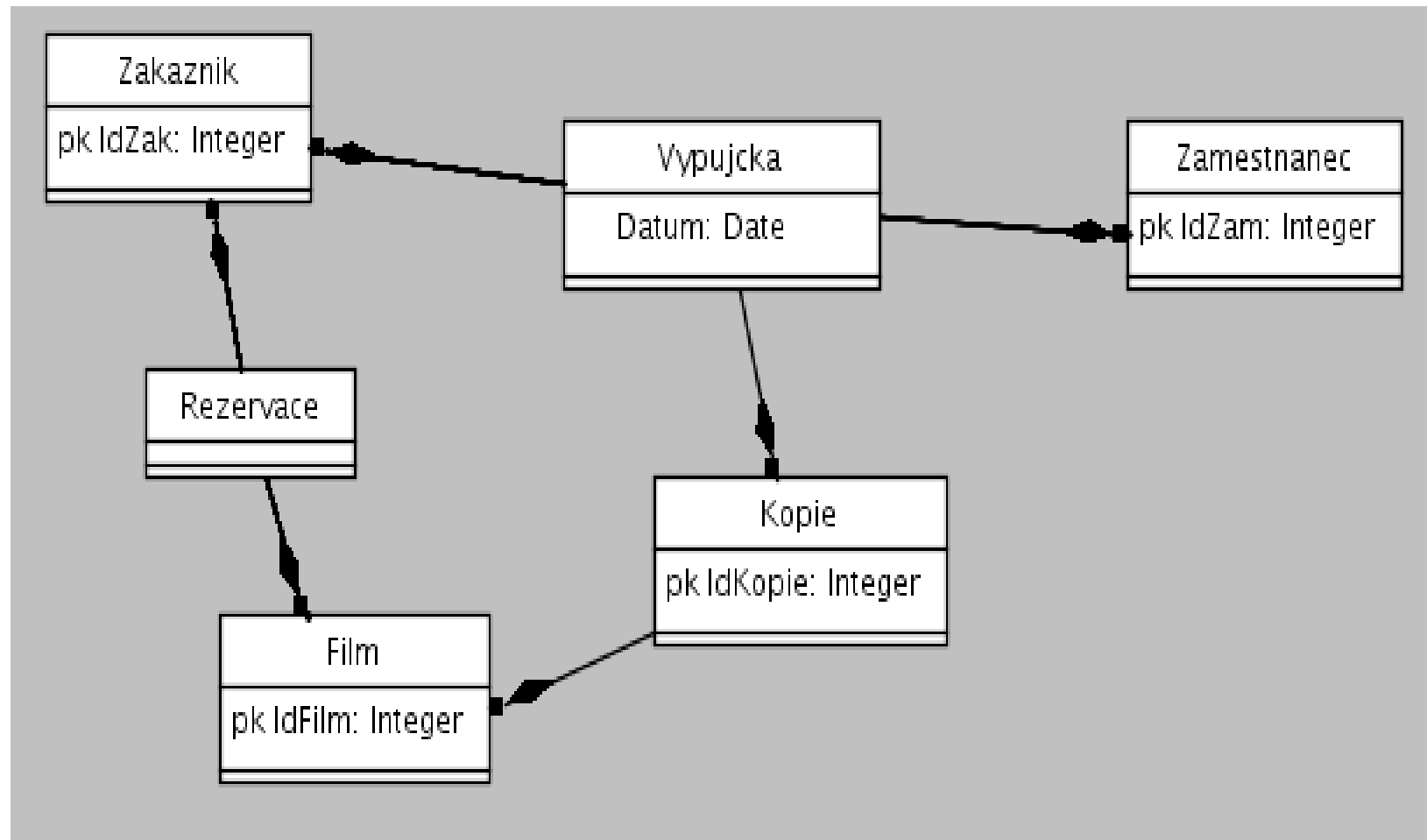
Půjčovna se rozhodla použít počítač pro evidenci výpůjček filmů. Eviduje se skutečnost, kterou lze jednou větou vyjádřit takto : ***Danému zákazníkovi je půjčen daný film daným zaměstnancem půjčovny.***



První nástřel.

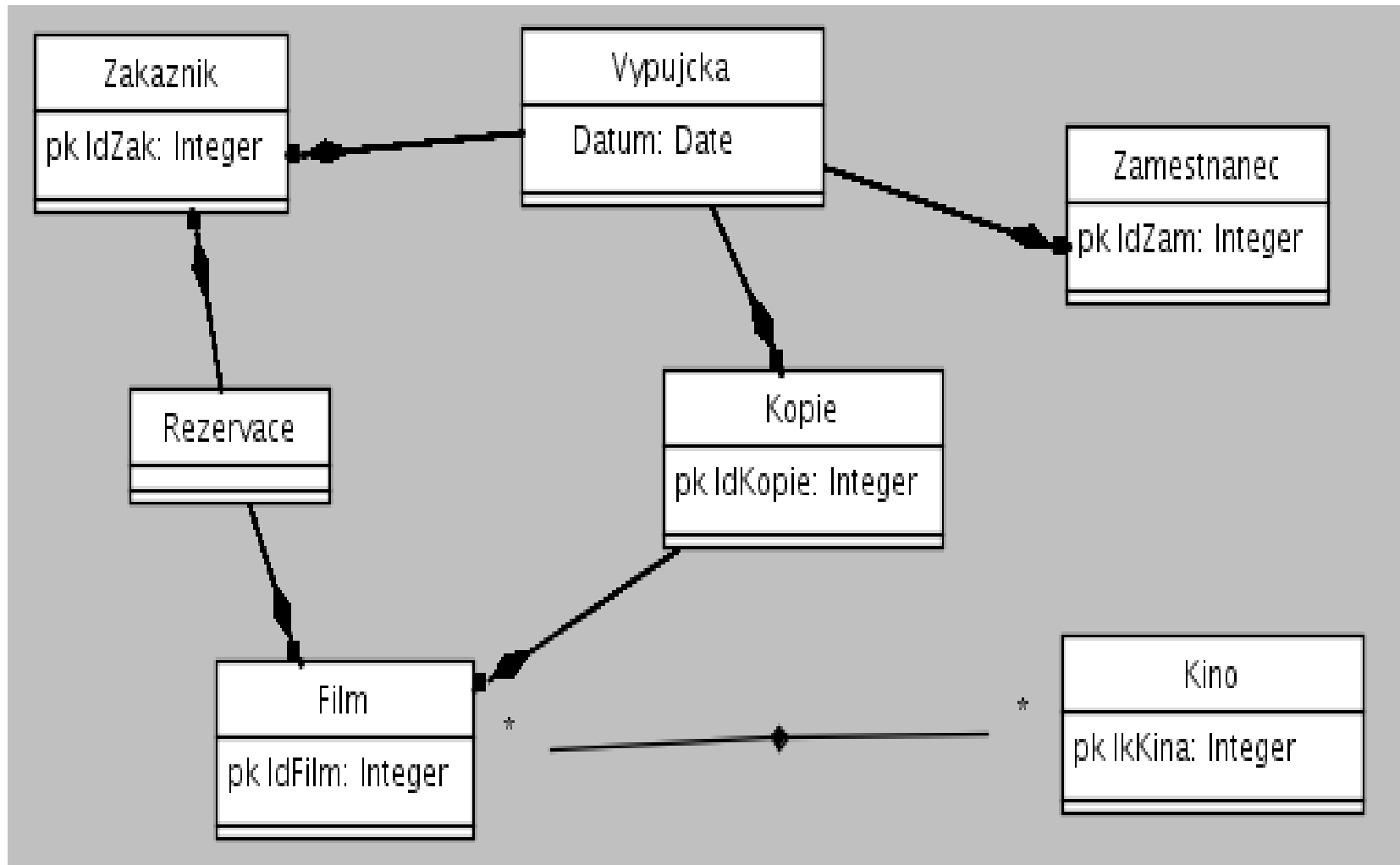


## Příklad – návrh videopůjčovny 2/3



Půjčují se kopie, rezervují se filmy.

# Příklad – návrh videopůjčovny 3/3



Chceme evidovat také programy kin.

Co když požadavek na rezervaci filmu, který půjčovna nevlastní?

# Databázové modelování– poznámky na závěr

- Notace mohou být různé.
- Při konceptuálním modelu se staráme hlavně o **popis reality**, ne o řešení v konkrétním systému.
- Používáme **vztahy mezi entitami, kardinalitu a parcialitu** nikoliv cizí klíče!
- Jakmile s v obrázku objeví **cizí klíče**, už se nejedná o konceptuální model, ale o **grafickou podobu modelu relačního!**