

Úvod do jazyka SQL

druhý blok

Osnova

- NULL
- DISTINCT
- JOIN

Cvičení

Vypište průměrnou délku filmu v jednotlivých letech. Výsledek seřadte sestupně dle roku a u každého roku zároveň vypište celkový počet filmů.

Do výsledku nezahrnujte filmy, u kterých není zadán rok a filmy, u kterých je rok premiéry v budoucnosti.

NULL

Neexistující/nedefinovaná hodnota
Nefungují operátory =, <>, ale pouze:

`<soupec> IS NULL`

případně

`<sloupec> IS NOT NULL`

<https://posp.cz/w/muni>

Počet lidí, které mají jednotlivé fakulty v jednotlivých budovách

```
SELECT fakulta, budova,  
COUNT(*) FROM fakulty  
GROUP BY fakulta, budova
```

COUNT(*) vs COUNT(sloupec)

COUNT(sloupec) “započítává” pouze ty řádky, kde je hodnota <sloupce> NOT NULL

	A	B	
1	Kniha	Pujceno	
2	A	Osoba 1	
3	A	Osoba 2	
4	A		
5	B	Osoba 1	
6	B	Osoba 2	
7	C		
8	D		
9	E	Osoba 1	
10	E		
11	F	Osoba 2	
12	G	Osoba 3	
13			

Databáze “minidb”

Vypište počet výtisků jednotlivých knih spolu s počtem těch, které jsou zrovna vypůjčené

```
SELECT kniha, count(*) as vse, COUNT(pujceno)
as vypujceno
FROM minidb
GROUP BY kniha
```

DISTINCT

Databáze “fakulty”

Vypište jednotlivé fakulty.

```
SELECT fakulta FROM fakulty GROUP BY fakulta
```

ale i

```
SELECT DISTINCT fakulta FROM fakulty
```

DISTINCT zaručuje unikátnost zobrazené kombinace vypsaných sloupců, proto

```
SELECT DISTINCT fakulta, budova FROM fakulty
```

a

```
SELECT fakulta, budova FROM fakulty GROUP BY fakulta
```

už stejné výsledky nevrací

Databáze “fakulty”

Vypište počet budov jednotlivých fakult

```
SELECT fakulta, COUNT(DISTINCT budova) FROM  
fakulty GROUP BY fakulta
```

...bez DISTINCT by byla každá budova započítána tolikrát, kolik je v ní kateder

COUNT(*)

vs

COUNT(sloupec)

vs

COUNT(DISTINCT sloupec)

JOIN

Příspěvky

Jméno	E-mail	Příspěvek	Kampaň
Honza	jsem@posp.cz	300	hithit
Honza	jsem@posp.cz	1000	kontaktní
Tomáš	fil.hrabal@gmail.com	500	hithit
Eliška	eliska@eps.cz	300	kontaktní

Problémy?

- Pokud jeden dárce přispívá vícekrát, jeho Jméno a E-mail se budou v tabulce opakovat u každého příspěvku
- Redundance zvyšuje riziko nekonzistence – například pokud dárce změní e-mail, budete muset aktualizovat každou odpovídající řádku.
- Pokud byste chtěli aktualizovat informace o dárci (např. jeho jméno nebo e-mail), musíte tyto změny provést na více řádcích. To může být náročné na údržbu a vést k chybám.
- Nemáte přehled o unikátních donorech bez provedení deduplikace (například pomocí GROUP BY nebo deduplikace podle e-mailu).
- Chcete-li přidat další informace o donorech (např. telefon, adresa), budete muset upravit strukturu tabulky, což není optimální.
- Nemůžete evidovat potenciální donory, kteří ještě nic nedarovali
- ...

Koho by téma zajímalo víc, tak viz https://cs.wikipedia.org/wiki/Normalizace_datab%C3%A1ze

Řešení?

Dvě separátní tabulky navzájem propojitelné hodnotou v nějakém sloupci/sloupcích

Donoři

Příspěvky

ID	Jméno	Kontakt	Telefon	Poznámka
1	Honza	jsem@posp.cz	111	Nevolat před 10 dopo
2	Tomáš	fil.hrabal@gmail.com	222	
3	Eliška	eliska@frankbold.org	333	Pracuje ve FrankBold

Sloupec Donor obsahuje hodnoty ze
sloupce ID tabulky donorů

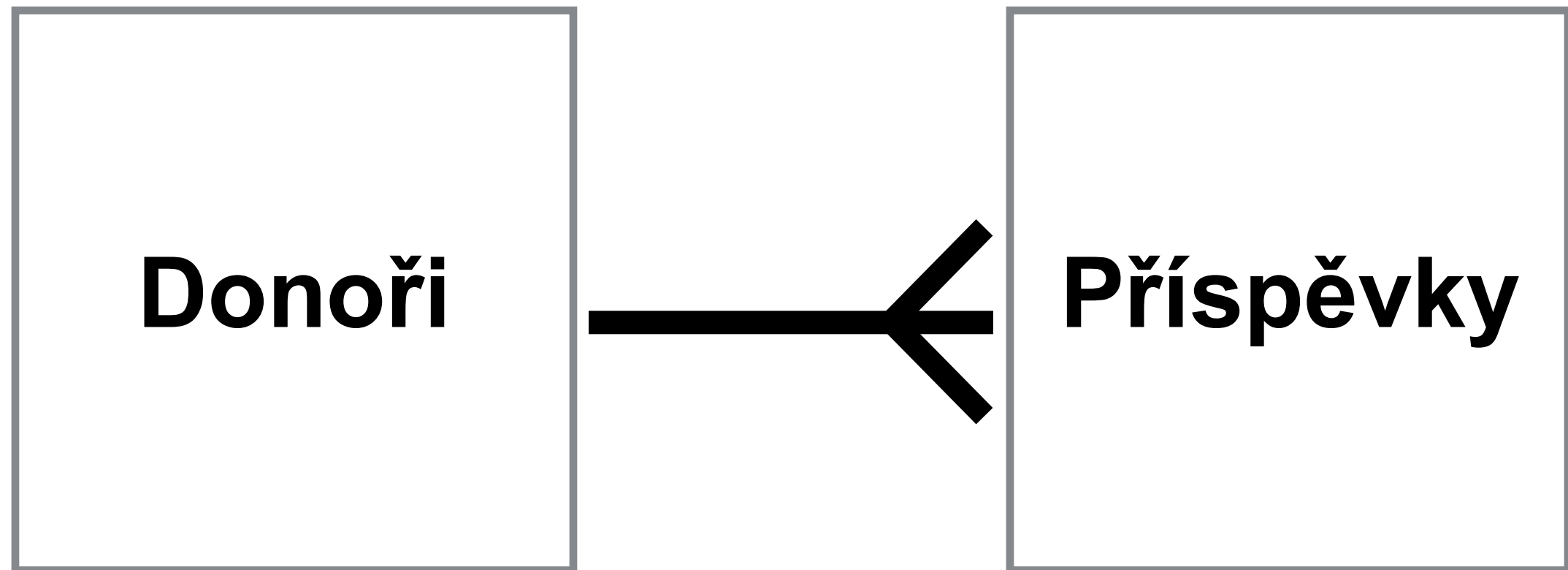
Donor	Příspěvek	Kdy	Kampaň
1	300	30. 10. 2014	hithit
1	1000	1. 11. 2014	kontaktní
2	500	1. 11. 2014	hithit
3	300	30. 10. 2014	kontaktní

posp.cz/w/fund2

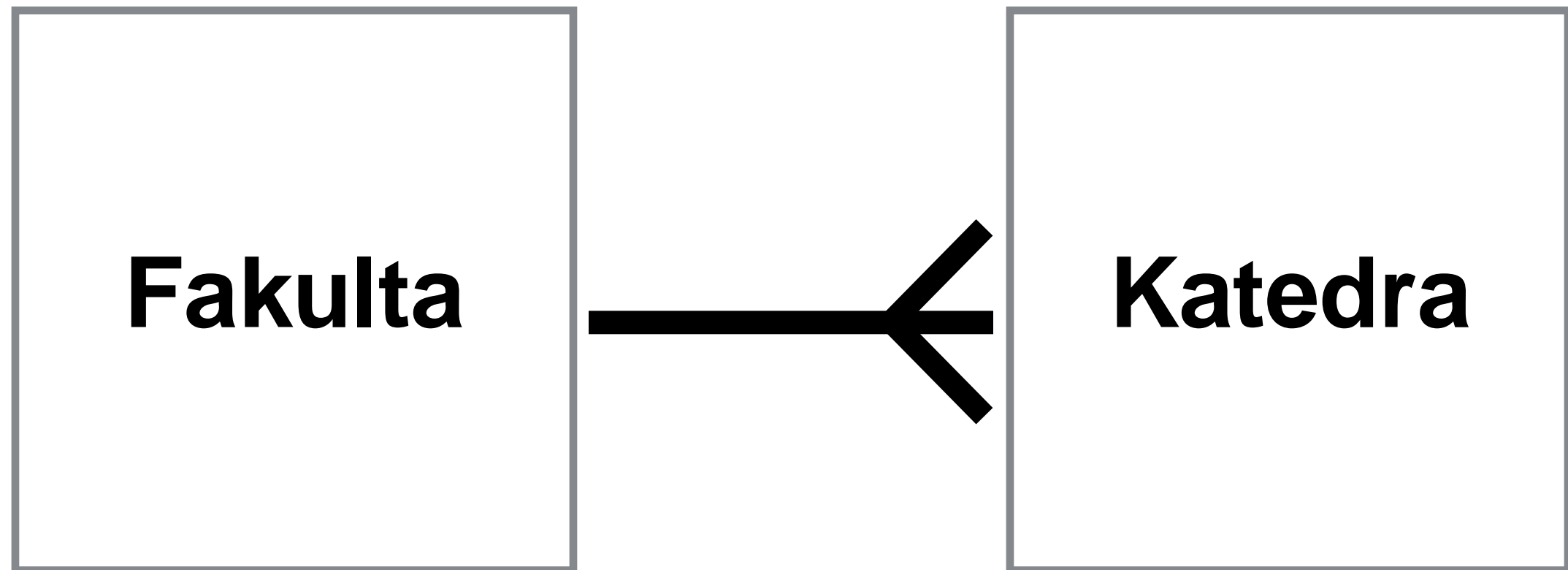
1:M

Vazba 1:M

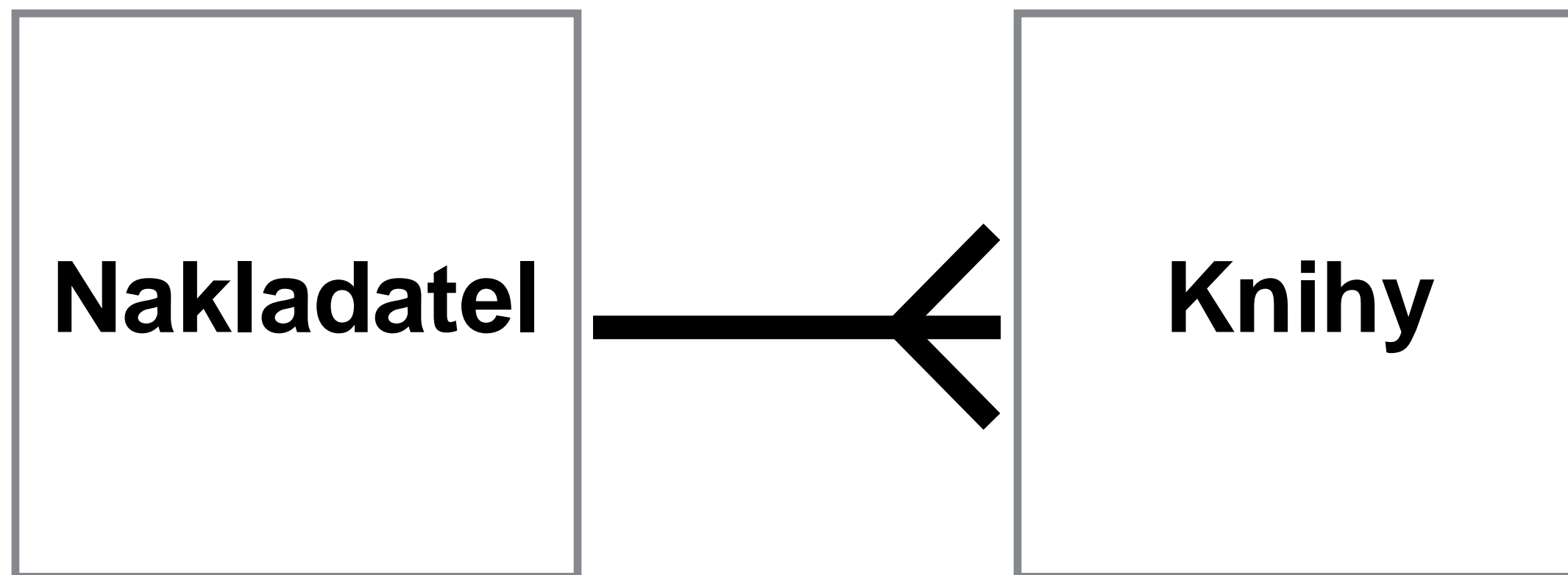
(někdy psáno "1:N", nebo 1:Many)



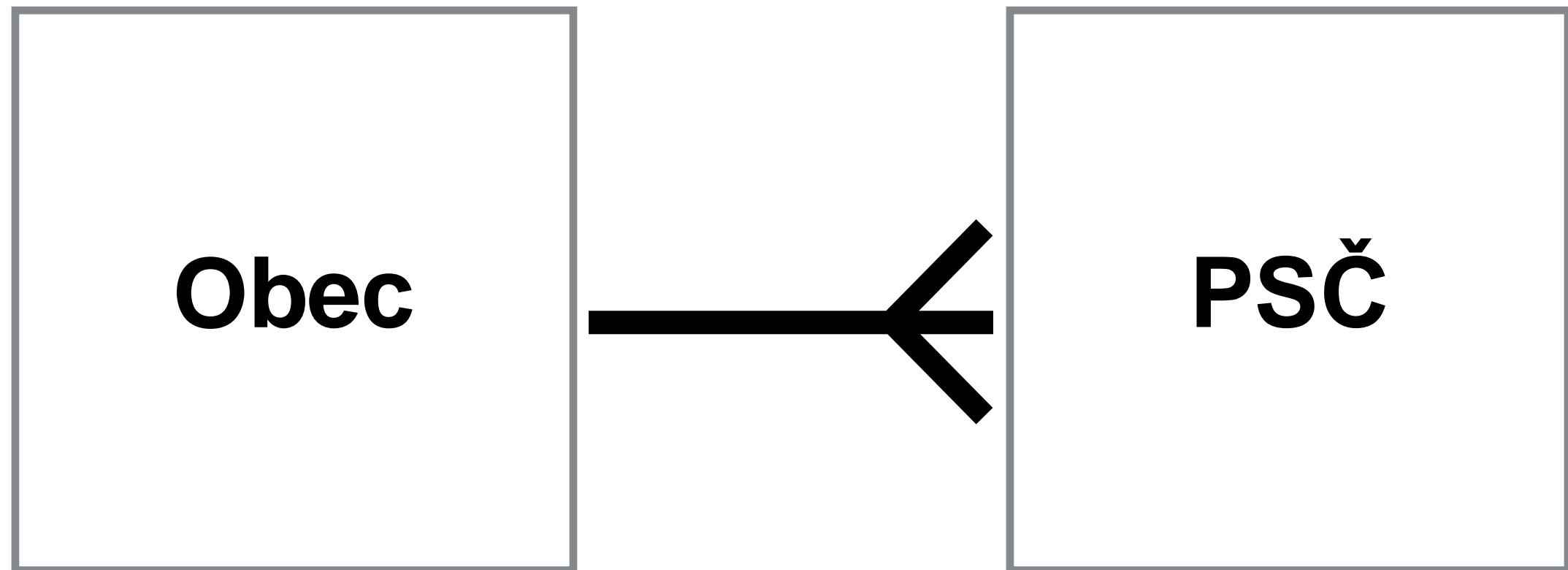
Jeden donor může dát více příspěvků,
každý příspěvek má jednoho donora

$1:M$ 

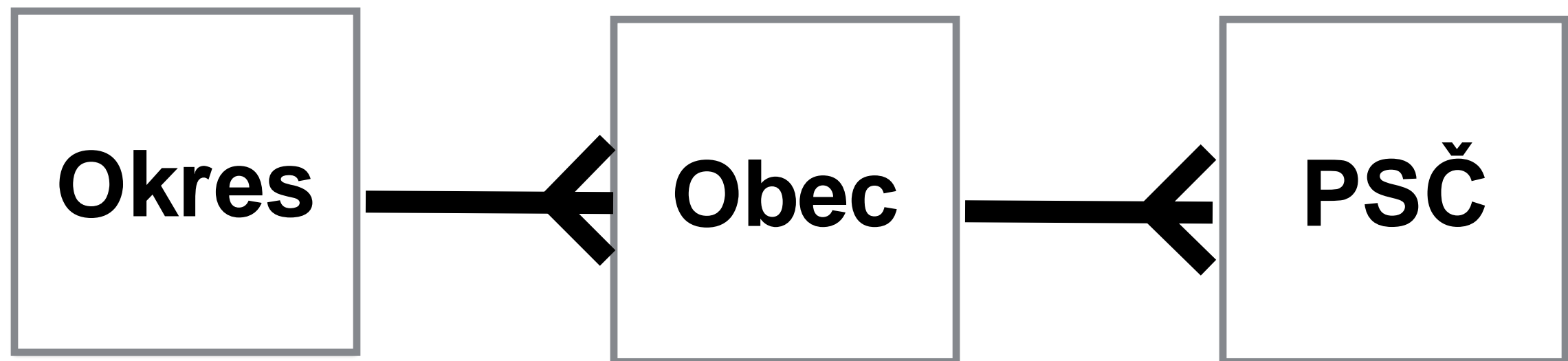
Jeden fakulta může mít více kateder,
každá katedra spadá pod jednu fakultu

1:M

Jeden nakladatel může vydat více knih,
každá kniha má jednoho nakladatele

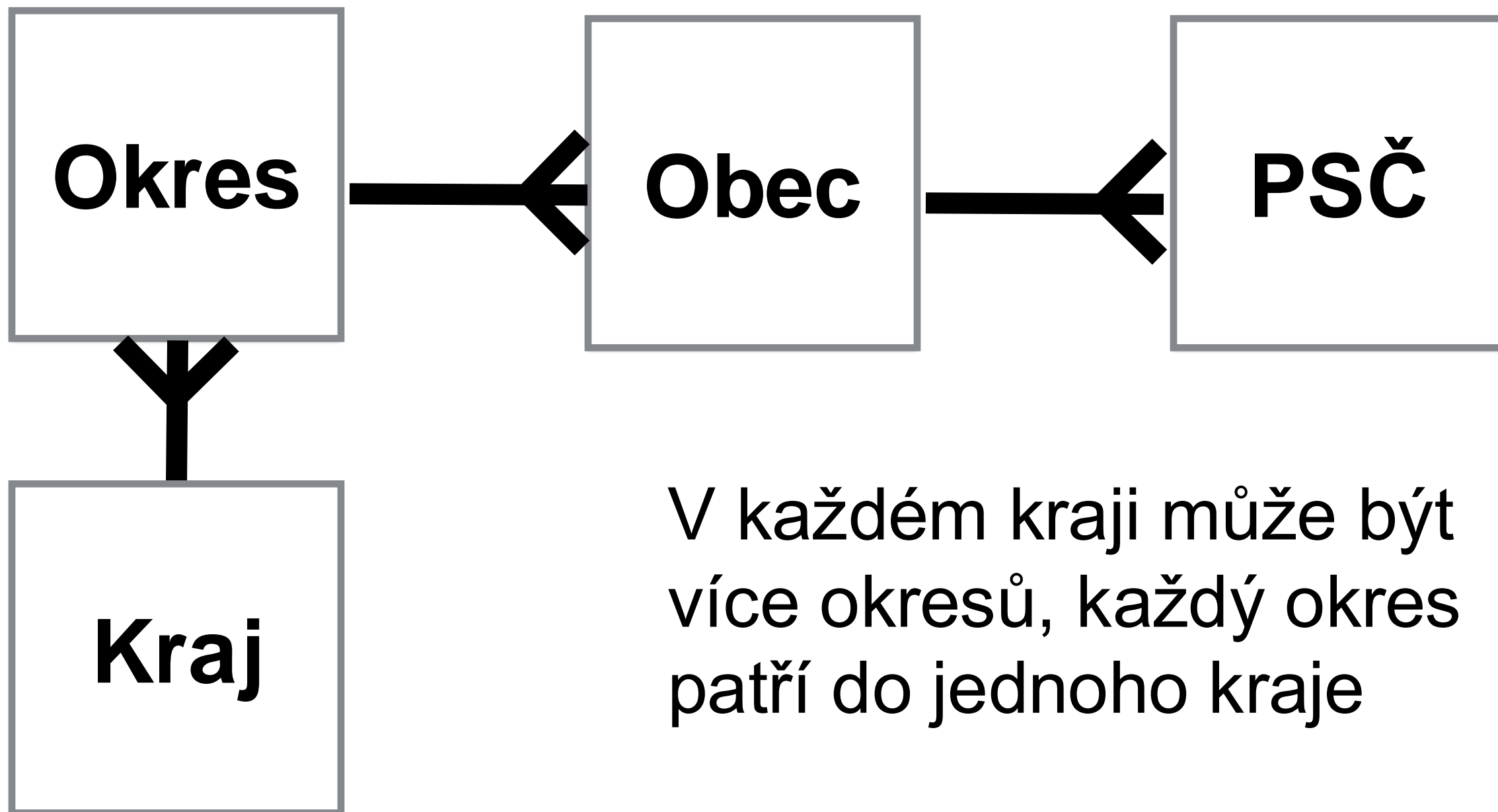
$1:M$ 

V rámci obce může být více PSČ,
každé PSČ patří do jedné obce

$1:M$ 

V každém okrese může být více obcí,
každá obec je v jednom okrese

1:M



V každém kraji může být více okresů, každý okres patří do jednoho kraje

ID	Jméno	Kontakt	Poznámka
1	Honza	jsem@posp.cz	Nevolat před 10 dopo
2	Tomáš	fil.hrabal@gmail.com	
3	Eliška	eliska@eps.cz	Pracuje v EPS

Donor	Příspěvek	Kdy	Kampaň
1	300	30. 10. 2014	hit
1	1000	1. 11. 2014	kontaktní
2	500	1. 11. 2014	hit
3	300	30. 10. 2014	kontaktní

JOIN

```
SELECT * FROM prispevky  
JOIN donori ON donori.ID = prispevky.donor
```

ID	Jméno	Kontakt	Poznámka	Donor	Příspěvek	Kdy	Kampaň
1	Honza	jsem@posp.cz	Nevolat před 10 dopo	1	300	30. 10. 2014	hithit
2	Tomáš	fil.hrabal@gmail.com		1	1000	1. 11. 2014	konktatní
				2	500	1. 11. 2014	hithit
3	Eliška	eliska@eps.cz	Pracuje v EPS	3	300	30. 10. 2014	kontaktní

Výsledek

Jméno	Kontakt	Poznámka	Příspěvek	Kampaň
Honza	jsem@posp.cz	Nevolat před 10 dopo	300	hithit
Honza	jsem@posp.cz	Nevolat před 10 dopo	1000	kontaktní
Tomáš	fil.hrabal@gmail.com		500	hithit
Eliška	eliska@eps.cz	Pracuje v EPS	300	kontaktní

JOIN vs. LEFT JOIN

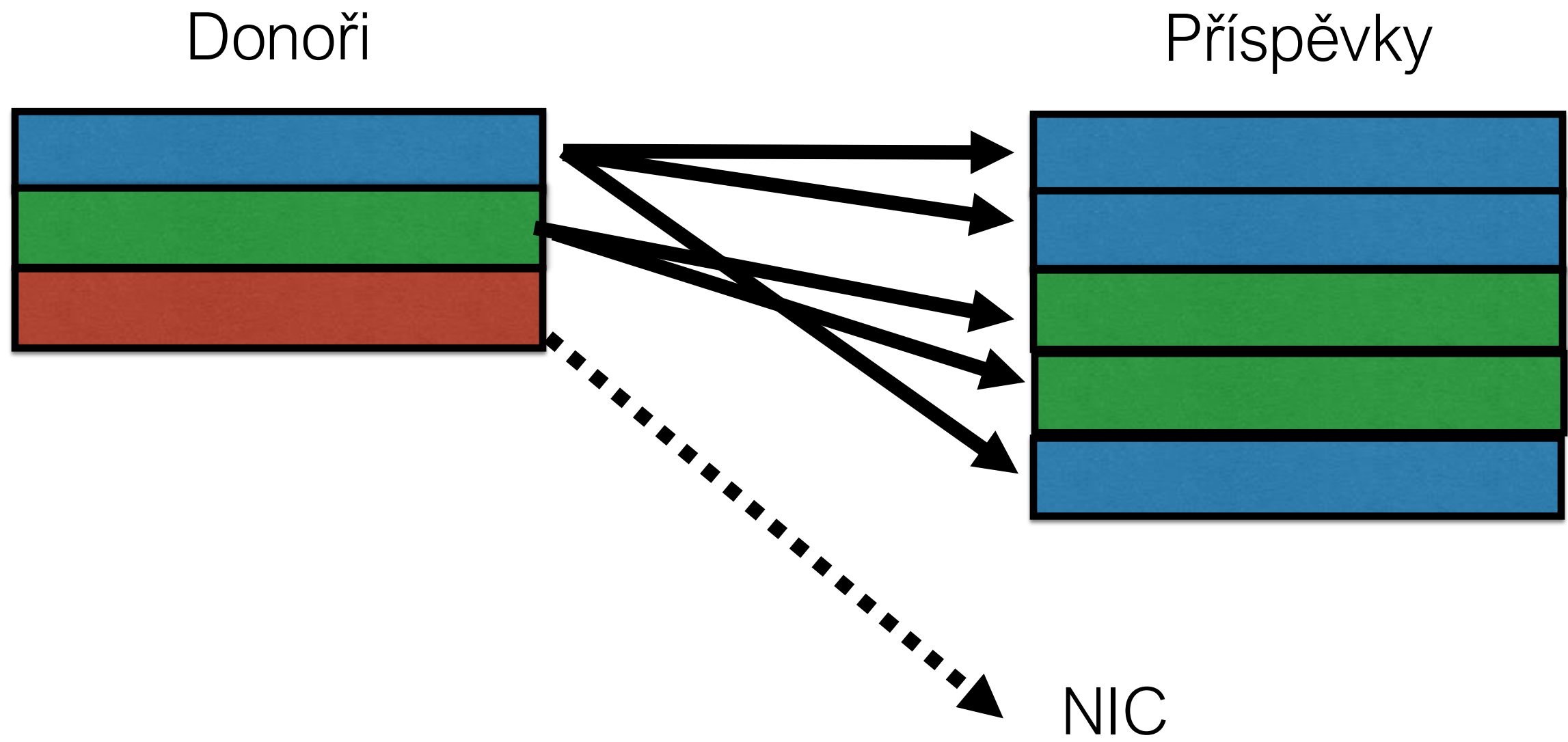
Donoři



Příspěvky



JOIN vs. LEFT JOIN



(INNER) JOIN

Donoři

Blue
Blue
Green
Green
Blue

Příspěvky

Blue
Blue
Green
Green
Blue

POZOR:
Databáze je hloupá!

Neví nic o sémantice vašich dat

POZOR: Databáze je hloupá!

Neví nic o sémantice vašich dat

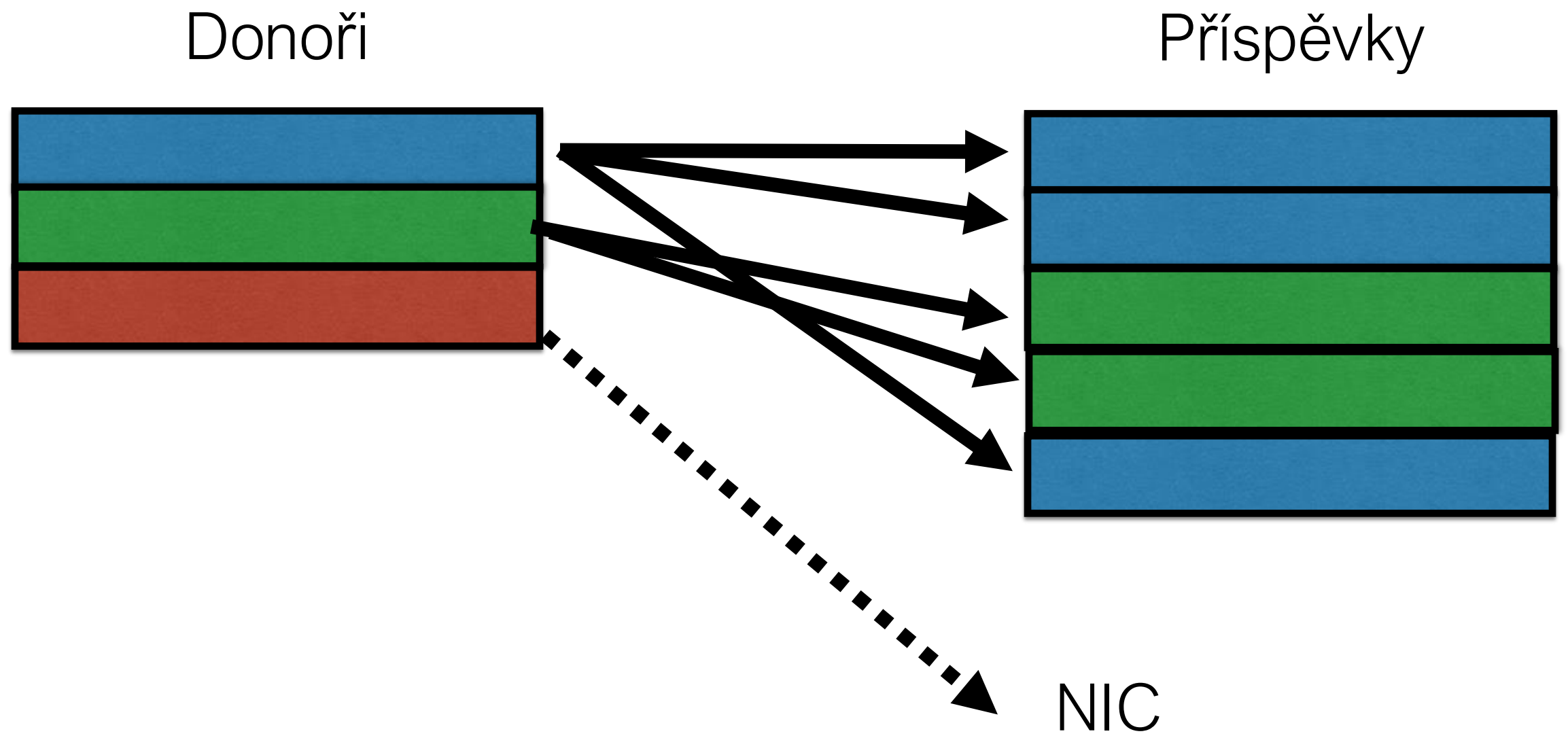
SELECT * FROM donori

JOIN prispevky

ON donori.rocnik = prispevky.castka

ID	Jméno	Kontakt	Telefon	Ročník	Poznámka
1	Honza	jsem@posp.cz	111	1984	Nevolat před 10 dpo
2	Tomáš	fil.hrabal@gmail.com	222	1985	
3	Eliška	eliska@frankbold.org	333	1983	Pracuje ve FrankBold

JOIN vs. LEFT JOIN

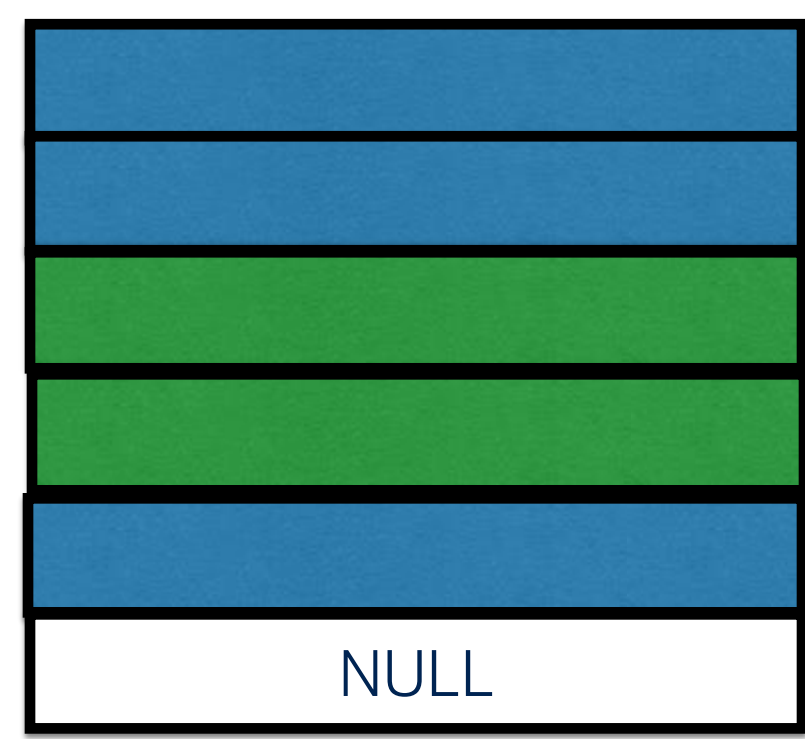


LEFT

Donoři



Příspěvky



Otázka

Kolik kdo z dárců v průběhu roku přispěl?

Pouze ti, kteří něco přispěli:

```
SELECT donori.jmeno, SUM(castka) FROM donori  
INNER JOIN prispevky ON donori.ID = prispevky.donor  
GROUP BY donori.ID
```

Všichni

```
SELECT donori.jmeno, SUM(castka) FROM donori  
LEFT JOIN prispevky ON donori.ID = prispevky.donor  
GROUP BY donori.ID
```

U těch co nepřispěli, je hodnota sloupce SUM(castka) NULL

posp.cz/w/sport

Vypište tabulku zápasu včetně jmen týmů (z tabulky “tym”)

Pokud vícekrát připouji stejnou tabulku, musím zavést “aliasy”, jinak bude dotaz končit chybou “ambiguous column name”, databáze “neví” jaký konkrétní sloupeček myslíte. Příklad dotazu, který skončí chybou:

```
SELECT * FROM zapas
INNER JOIN tym ON tym.id_tym = domaci_tym
INNER JOIN tym ON tym.id_tym = vitezny_tym
```

Upravená funkční verze:

```
SELECT tabulka_tym_domaci.nazev as nazev_domaci, tabulka_tym_hoste.nazev as nazev_hoste,
golu_domaci, golu_hoste FROM zapas
INNER JOIN tym AS tabulka_tym_domaci ON tabulka_tym_domaci.id_tym = domaci_tym
INNER JOIN tym AS tabulka_tym_hoste ON tabulka_tym_hoste.id_tym = hostujici_tym
```

(pozn. Název aliasu je čistě na vás)

M:N

Vazba M:M

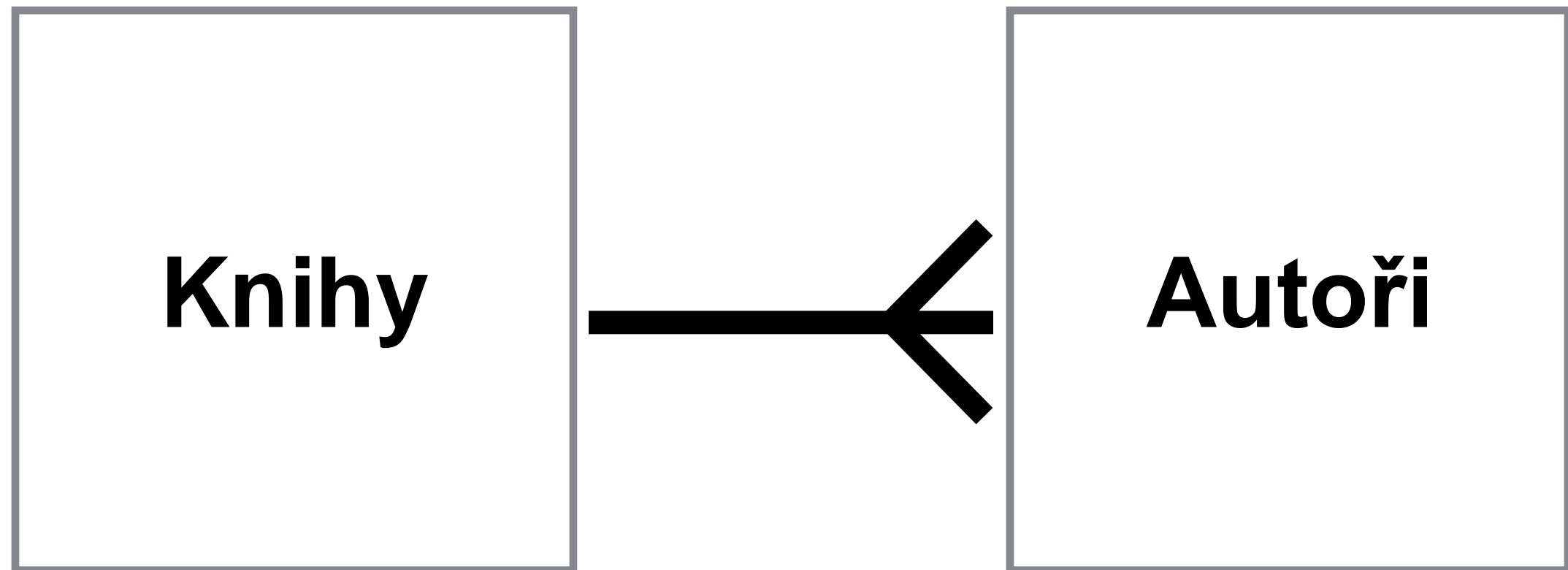
(někdy psáno "M:N", nebo Many:Many)

Knihy

Autoři

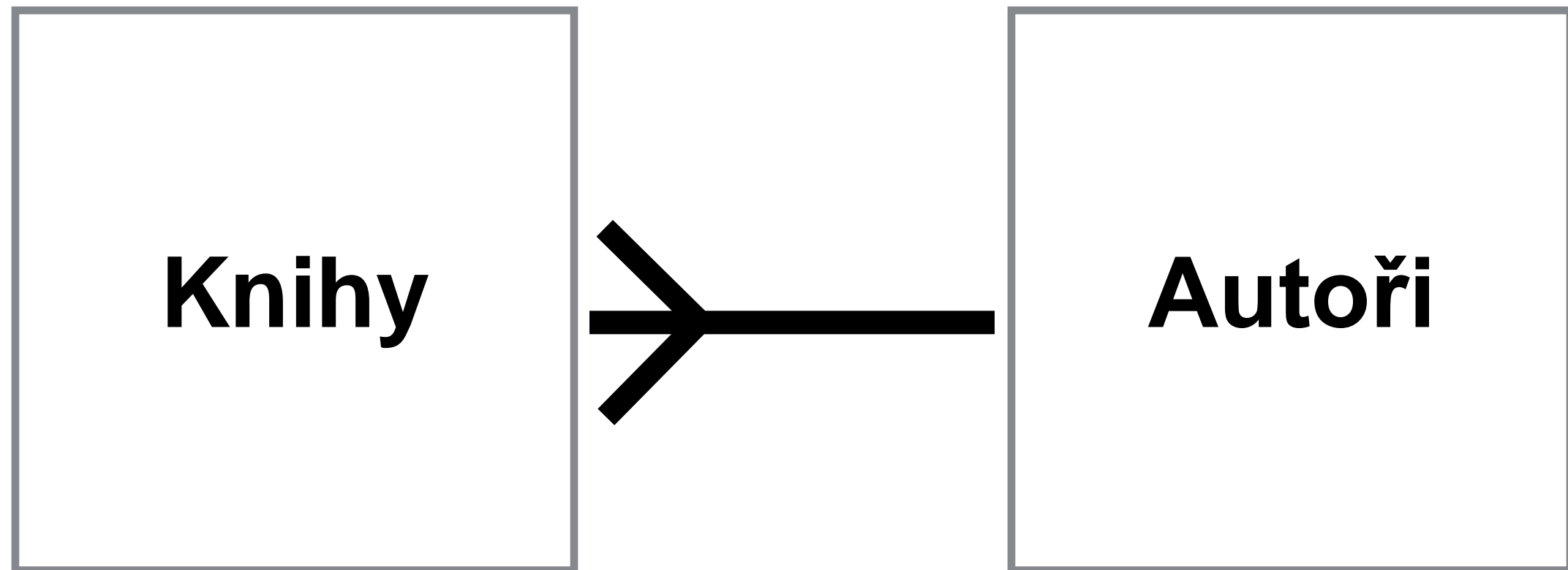
M:N

1:M nestačí, neumí zachutit, že jeden autor napsal více knih



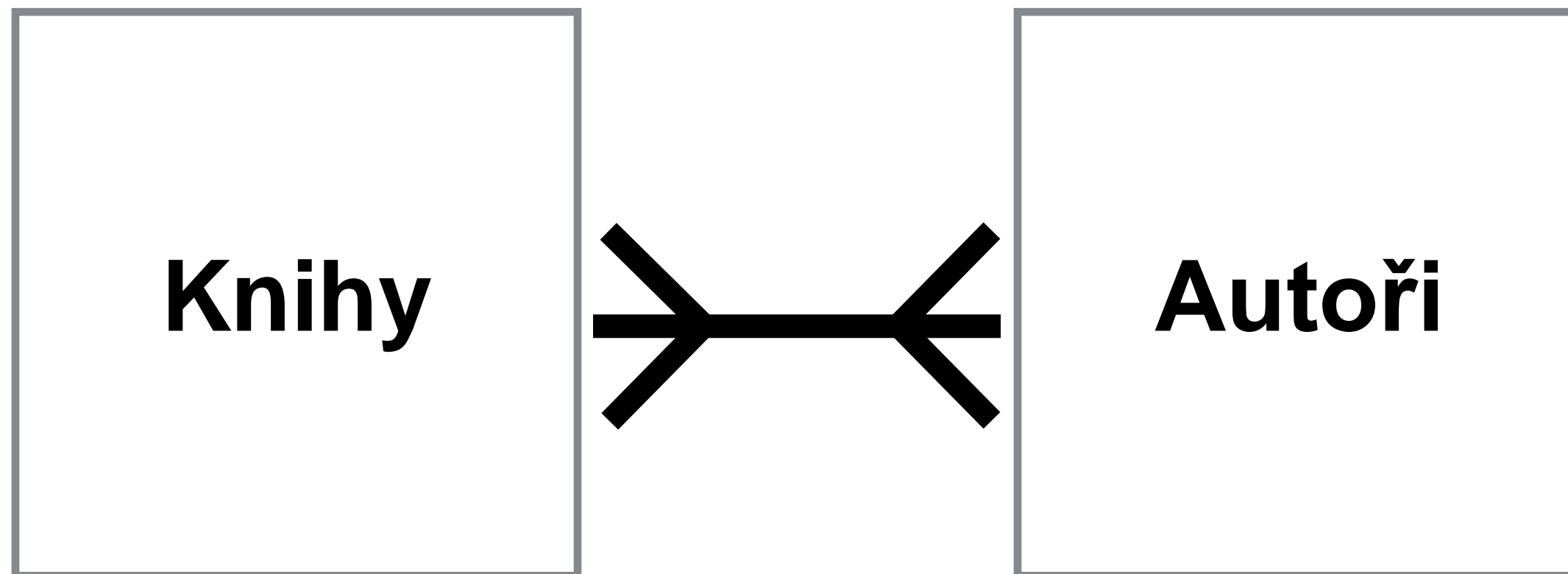
M:N

M:1 nestačí, neumí zachutit, že jednu knihu napsalo více lidí

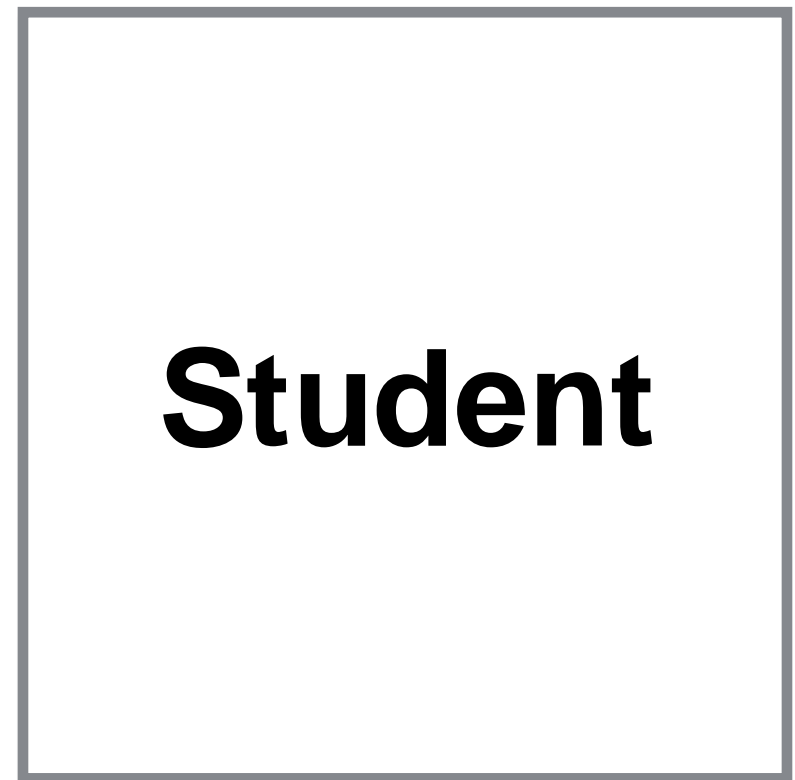
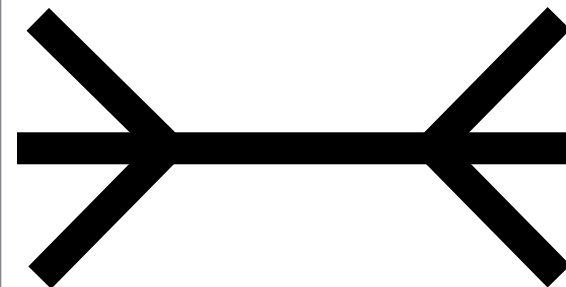
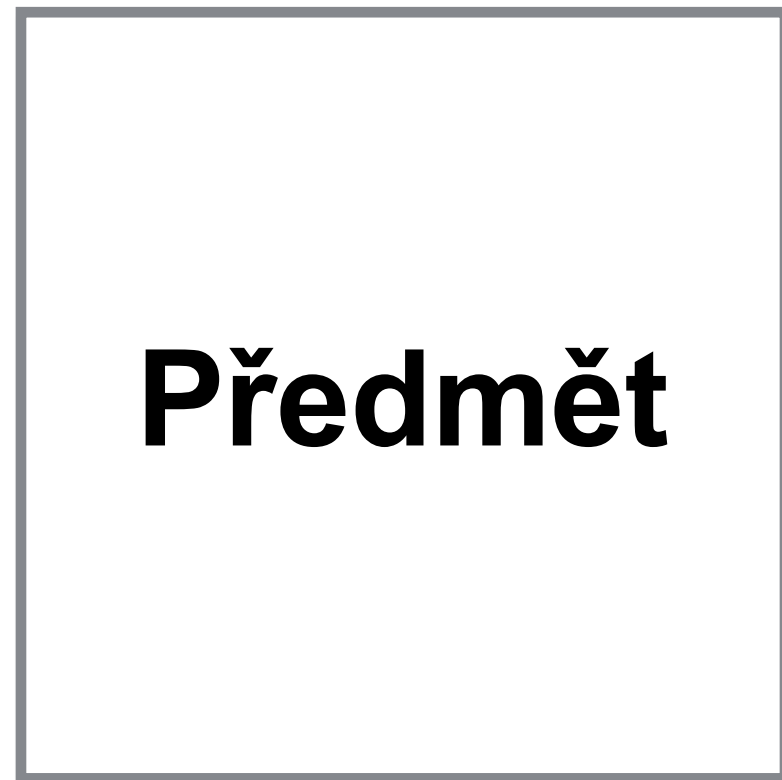


M:N

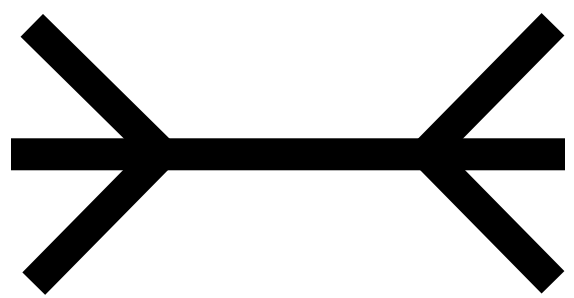
Řešením je tzv. vazební tabulka
(doporučuji si pustit záznam 2. boku od
času 2h 7min)



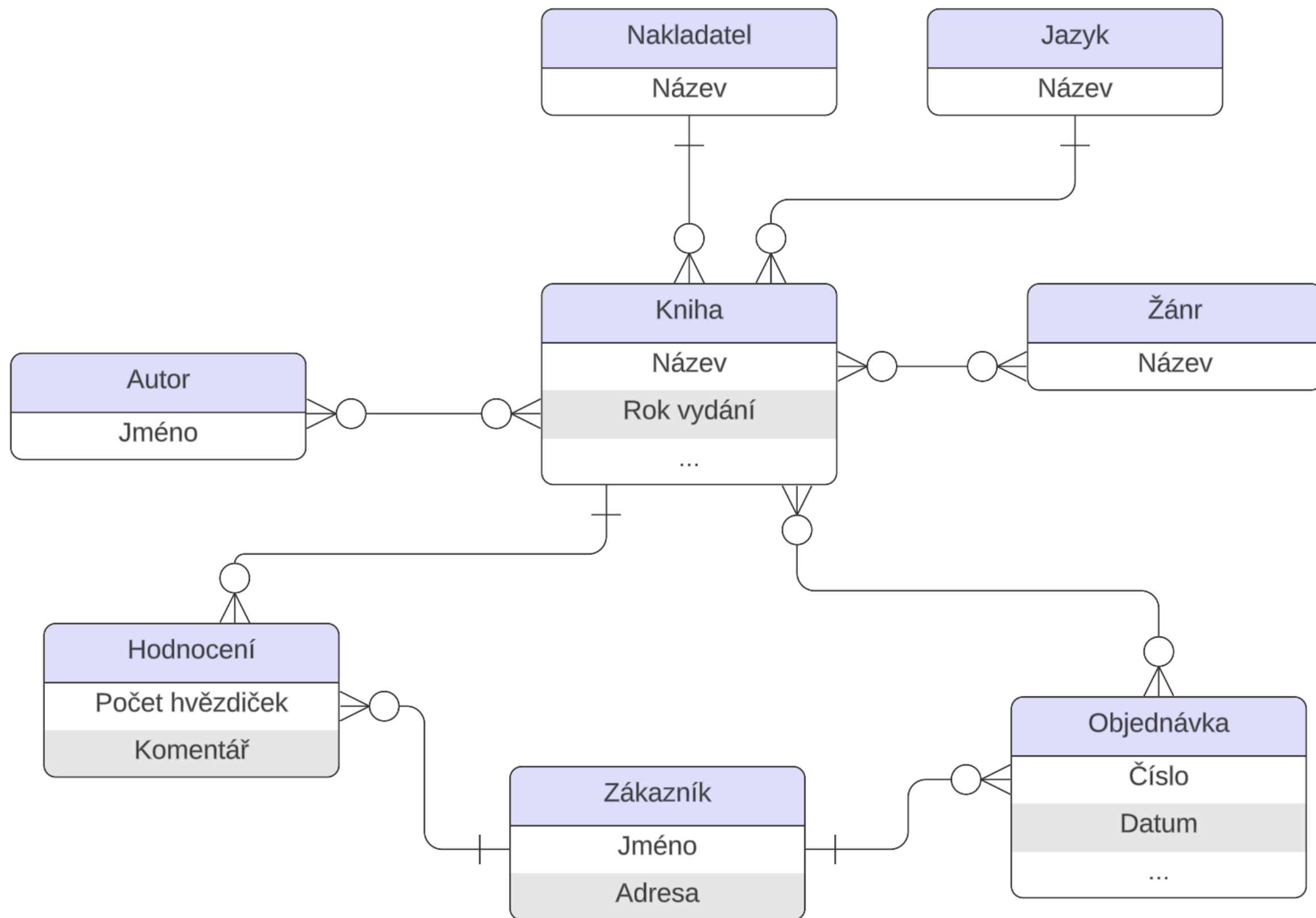
M:N



M:N



posp.cz/w/eshop



Otázka

Vypište knihy, které napsal Constance Carroll
(jedním dotazem)

```
SELECT autori.id, autori.jmeno, autori.prijmeni, knihy.nazev  
FROM autori  
INNER JOIN autori_knihy ON autori.id = autori_knihy.autori_id  
INNER JOIN knihy ON autori_knihy.knihy_id = knihy.id  
WHERE jmeno LIKE 'Constance' AND prijmeni LIKE 'Carroll'
```

autori_knihy je "vazební tabulka"

